Interest Driven Graph Structure Learning for Session-Based Recommendation

Huachi Zhou[1], Shuang Zhou[1], Keyu Duan[2], Xiao Huang[1] (✉), Qiaoyu Tan[3], and Zailiang Yu[4]

[1] The Hong Kong Polytechnic University, Hung Hom, Hong Kong
huachi.zhou@connect.polyu.hk, {csszhou, xiaohuang}@comp.polyu.edu.hk
[2] National University of Singapore, Singapore k.duan@u.nus.edu
[3] Texas A&M University, College Station, USA qytan@tamu.edu
[4] Zhejiang Lab, China yuzl@zhejianglab.com

Abstract. In session-based recommendations, to capture user interests, traditional studies often directly embed item sequences. Recent efforts explore converting a session into a graph and applying graph neural networks to learn representations of user interests. They rely on predefined principles to create edges, e.g., co-occurrence of item pairs in the sequence. However, in practice, user interests are more complicated and diverse than manually predefined principles. Adjacent items in the sequences may not be related to the same interest, while items far away from each other could be related in some scenarios. For example, at the end of shopping, the user remembers to purchase items associated with the one purchased at the beginning. While using predefined rules may undermine the quality of the session graph, it is challenging to learn a reasonable one that is in line with the user interest. Sessions are diverse in length, the total number of interests, etc. Signals for supervision are not available to support graph construction. To this end, we explore coupling the session graph construction with user-interest learning, and propose a novel framework - PIGR. It recognizes items with similar representations learned based on sequential behavior and preserves their interactions. Related items reside in the same induced subgraph and are clustered into one interest. A unified session-level vector is retrieved from the different granularity of interests to guide the next-item recommendation. Empirical experiments on real-world datasets demonstrate that PIGR significantly outperforms state-of-the-art baselines.

1 Introduction

Session-based recommendation has received considerable attention [17] because online users may not log in for fear of breach of privacy, making tracing the historical behaviors of users infeasible. Conventional sequential recommendation is based on rich explicit user-item interactions to reveal user preferences [5]. Session-based recommendation targets at predicting the next item choice given an anonymous sequence clicked in one session [6]. Early studies on this emerging domain mainly focus on mining actionable patterns from the chronologically ordered items. Multi-layer recurrent neural networks [8] and co-attention mechanism [3] are designed to process consecutive clicks.

Graph neural networks (GNNs) have been under broad research in session-based recommendation. One-way sequence modeling only captures adjacent dependency among consecutive items. The transition may be too sparse to effectively derive user preferences. In many scenarios, distant items might be relevant and nonadjacent dependency could reduce the overfitting brought by the sparsity of sessions. GNNs have been intensively explored to resolve mentioned problems [19]. The basic idea is to convert each click sequence into graphs to enable message passing between distant items. Along this line, advanced models are proposed to better capture collaborative signals. For example, researchers develop dual graph neural networks to exploit both global-level and local-level item transitions [18] or hypergraphs to learn the inherent dependency of items across all sessions [20]. Unanimously, these GNN-based methods express the connectivity of graph structure by manually predefined principles. A dominant heuristic principle is to use co-occurrences of item pairs as edges [2,7,22].

Despite the effectiveness, we argue that user interests are naturally far more complicated and diverse than manually predefined principles. Adjacent items may not have a strong semantic relation, while distant items not adjacent to the same pivot item might still be semantically related in some scenarios. As illustrated in Fig. 1, because the user casually clicks Airpods after wok, there is an edge from wok to Airpods in the graph based on the principle. But wok and pot are more related compared with Airpods. When GNNs recursively aggregate representations of connected items, features not in the same interest will propagate to the same node. Consequently, it may generate inaccurate summary of user interests and lead to suboptimal model performance.

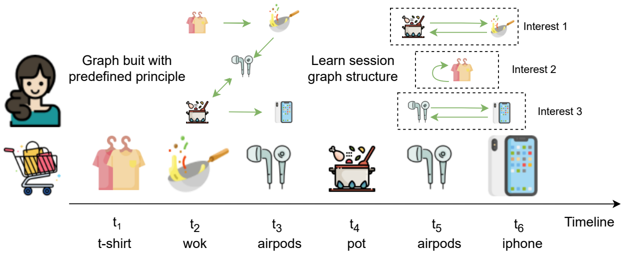

Fig. 1: A running example of two approaches of constructing the session graph, where dashed rectangle indicates one interest.

Motivated by the aforementioned issues, we explore the viability of coupling the session graph construction from scratch with distilling user interests. The items yearn for beneficial information from proximal nodes that share similar features and are clustered into different granularity of interests. The larger magnitude of interest is more possibly being the reason that drives the user to consume next item. However, it is a non-trivial and challenging task. First, the supervision signal indicating item node linkage is unobservable. Items belonging to the same interest are expected to be recognized and resided into the same induced subgraph.

But sequences do not have underlying intrinsic graphs that discriminate whether two items in the session should interact or not. The graph structure modification has to take on the opportunity of maximizing the prediction performance. Second, user interests are diverse. Sessions have differentiated sequence structures in terms of session length, distinct items, and interest number. Moreover, user personalized interests evolve from historical actions. The proposed solution is agnostic regarding the specifics of interest distribution for each session in advance. It is demanding to specify an appropriate number of interests for each session ahead of training.

To this end, we propose a novel method dubbed Personalized Interest Graph Recommender (PIGR) to construct session graph structure driven by distilling user interests. The model finds a reasonable graph in the absence of side information with only user-item interactions available. Our main contributions are summarized below: (1) Instead of using predefined principles, we propose a differentiable framework PIGR to enable session graph construction and adaptive interest extraction simultaneously. (2) We propose a *session graph structure learning* module to preserve the connection between similar items inferred by sequential behavior and cluster them centered around the same interest node. (3) We propose a *unified interest retrieval* module to propagate item features to the selected interest node and utilize node centrality to aggregate different granularity of interest nodes into one unified session-level vector. (4) We evaluate our model on three public datasets and the experimental results validate the superiority of PIGR.

2 Personalized Interest Graph Recommender - PIGR

Problem formulation. Session-based recommender system aims to predict the next item based on an anonymous session. A session contains a series of consecutive items sorted by clicked timestamps in ascending order. Gathering items from all sessions forms the item set $\mathcal{I}$, where $|\mathcal{I}|$ represents cardinality size. For inference, given a session s with m present items, session-based recommender system predicts the probability of item q being picked as the next item, i.e., $p(q|s)$. Among the candidate set $\mathcal{I}$, the item with the highest probability will then be selected as the next one.

Our solution. Fig. 2 provides a pipeline illustration of the PIGR framework. In detail, it is composed of two modules as follows: (i) Session graph structure learning module: It infers item similarity by taking sequential behavior into consideration and explicitly guides the session graph construction by clustering similar items centered around the same interest node; (ii) Unified interest retrieval module: It propagates features to the interest node and formulates an adaptive number of interests. Then it encodes interest nodes over the entire graph to a unified session-level vector.

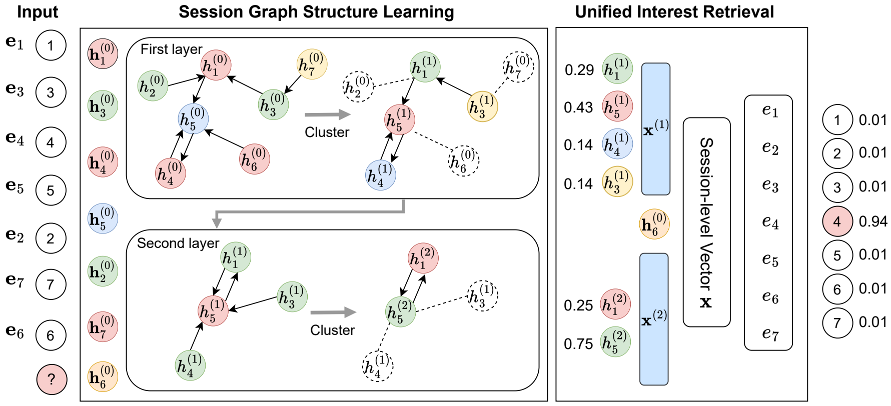

Fig. 2: The circles of the same color belong to the same interest. The first module evaluates the node similarities and converts the session to a graph. The second module propagates neighbor features along learned edges and outputs a unified session-level interest embedding.

2.1 Session Graph Structure Learning Module

At the beginning, the model may not accurately estimate node pairwise strength with only initial embedding. The item order underlying the sequence acts as prior knowledge of graph construction and is beneficial to node connection exploration. The absolute order is designed to depict the dependency contained in the absolute position of items in the sequence while the relative pattern emphasizes the correlation between the current item and prefix items. We use the position embeddings and Gated Recurrent Units (GRU) to model the absolute and relative order of items. The hidden representation of item j processed by GRU is as follows:

$$\boldsymbol{h}_j^{(0)} = GRU(\boldsymbol{h}_{j-1}^{(0)}, [\boldsymbol{e}_j \oplus \boldsymbol{p}_j]), \tag{1}$$

where $\oplus$ is the concatenation function; $\boldsymbol{p}_j \in \mathbb{R}^d$ denotes as a trainable position vector; embedding $\boldsymbol{e}_j$ is from learnable parameter matrix $\boldsymbol{E} \in \mathbb{R}^{|\mathcal{I}| \times d}$. Then we leverage items that integrate sequential information to learn the edges. Given the hidden representations of m nodes at the k-th layer $\left[\boldsymbol{h}_1^{(k)}, \boldsymbol{h}_2^{(k)}, \ldots, \boldsymbol{h}_m^{(k)}\right]$, we measure edge strengths by the cosine similarity metric:

$$\tilde{\boldsymbol{A}}_{ij}^{(k)} = cos(\boldsymbol{h}_i^{(k)}, \boldsymbol{h}_j^{(k)}) + \epsilon \cdot \tau(\boldsymbol{w}_n \boldsymbol{h}_i^{(k)}), \tag{2}$$

where $\epsilon \sim \mathcal{N}(0, 1)$ is a scalar independently sampled from the standard normal distribution, τ is the softplus activation function, and $\boldsymbol{w}_n \in \mathbb{R}^d$ is a learnable vector shared across layers. At the early training stage, it struggles to yield satisfactory item hidden representations. Each node may not selectively determine the optimal neighbors. Therefore, we add the trainable noise which slightly disturbs neighbor weights. Empirically, a node may interact with only a sparse set of nodes. To improve computing efficiency and remove edges with low information density, we pool a sparsified adjacent matrix from the previous fully connected graph by keeping each item with top t neighbors as follows:

$$\begin{aligned} \boldsymbol{A}_{i:}^{(k)} &= Softmax(Topt(\tilde{\boldsymbol{A}}_{i:}^{(k)}, t)), \\ Topt(\tilde{\boldsymbol{A}}_{i:}^{(k)}, t)_j &= \begin{cases} \tilde{\boldsymbol{A}}_{ij}^{(k)}, \text{ if } \tilde{\boldsymbol{A}}_{ij}^{(k)} \text{ is in the top } t \text{ values} \\ -\infty, \text{ otherwise} \end{cases}. \end{aligned} \tag{3}$$

After this, we obtain a reasonable graph structure $\boldsymbol{A}^{(k)}$. In this matrix, items that select to send messages to the same neighbor are clustered into the same subgraph and the neighbor node serves as the interest node, which will inform the downstream interest extraction process.

2.2 Unified Interest Retrieval Module

The former step finds related nodes for each selected interest node. Then we aggregate neighbor representations by performing message passing strategy on the built adjacent matrix $\boldsymbol{A}^{(k)}$ and the hidden representations $\boldsymbol{H}^{(k)}$. The $(k+1)$-th step message passing is computed by:

$$\boldsymbol{H}^{(k+1)} = MLP(\boldsymbol{A}^{(k)}\boldsymbol{H}^{(k)}) + \boldsymbol{H}^{(k)}, \tag{4}$$

where $MLP(\cdot)$ represents a two-layer perceptron network to integrate non-linear signal to each node and generate more expressive hidden representations. In such a manner, semantically similar node features are fused into the same coarsened node along learned edges. These coarsened nodes implicitly denote a set of clusters of multiple scales and propagation operation actually forces each node mapping to one interest then aggregate each interest. In particular, compared with efforts assigning soft cluster assignment matrix to nodes [23], we provide a general recipe to extract an adaptive number of interests without explicitly claiming ahead of training. Then we encode all interest nodes over the entire graph to the output and obtain a composite interest vector. First, the node centrality, i.e., sum of weighted in-degrees indicates the importance of each interest node in the dynamically learned graph structure formulated as $o_i^{(k+1)} = \sum_{j\in\mathcal{N}(i)} \boldsymbol{A}_{ij}^{(k+1)}$. Second, we attend interest nodes obtained from k layers with pooling to preserve the varying locality. The graph-level representation is then expressed by:

$$\boldsymbol{h} = \frac{1}{k+1}\sum_{i=0}^{k}\sum_{j=1}^{m} o_j^{(i+1)} \cdot \boldsymbol{h}_j^{(i+1)}. \tag{5}$$

The summarized interest ignores dependency contained in the linear order of items along time step. So we refine the graph-level representation with sequential information $\widetilde{\boldsymbol{h}}$, which is the last item output from Eq. (1) to learn the unified session-level vector as follows:

$$\begin{aligned} \boldsymbol{\alpha} &= \sigma(\boldsymbol{W}_\alpha[\widetilde{\boldsymbol{h}} \oplus \boldsymbol{h}]), \\ \boldsymbol{x} &= \boldsymbol{\alpha} \odot \widetilde{\boldsymbol{h}} + (\boldsymbol{1} - \boldsymbol{\alpha}) \odot \boldsymbol{h}, \end{aligned} \tag{6}$$

where σ is the sigmoid activation function; $\boldsymbol{W}_\alpha \in \mathbb{R}^{2d \times d}$ is a transformation matrix and $\boldsymbol{\alpha}$ balances the relative importance.

2.3 Training Objective

Our training target for session s is to minimize the following learning objective:

$$\mathcal{L} = \mathcal{L}_c(\hat{y}, y) + \beta \mathcal{L}_{reg}(\boldsymbol{h}, \widetilde{\boldsymbol{h}}), \tag{7}$$

where β controls the magnitude of the second loss. The $\mathcal{L}_c$ is the cross-entropy loss where $y \in \mathbb{R}^{|\mathcal{I}|}$ is the ground truth vector of session s and $\hat{y} \in \mathbb{R}^{|\mathcal{I}|}$ represents the estimated next item clicked probability. The next clicked probability concerning all items is given by: $\hat{y} = softmax(\boldsymbol{x}\boldsymbol{E}^\top)$. And we treat the recommendation task as a classification problem:

$$\mathcal{L}_c = -\sum_{i=1}^{|\mathcal{I}|} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i). \tag{8}$$

The $\mathcal{L}_{reg}$ acts as a regularization loss with data augmentation following [20]:

$$\mathcal{L}_{reg} = -\frac{1}{|\mathcal{B}|}\Big(\log \sigma(\boldsymbol{h} \odot \widetilde{\boldsymbol{h}}) + \mathbb{E}_{\hat{\boldsymbol{h}} \sim \mathbb{P}}\big(\log \sigma(1 - \boldsymbol{h} \odot \hat{\boldsymbol{h}})\big)\Big), \tag{9}$$

where σ is the sigmoid function, $|\mathcal{B}|$ is the batch size and $\hat{\boldsymbol{h}}$ is the derived embedding from $\widetilde{\boldsymbol{h}}$ with random permutation $\mathbb{P}$. The $\widetilde{\boldsymbol{h}}$ is regarded as congruent linear view of the graph. Maximizing the mutual information through regularization loss provides additional supervision signal and guarantees the interest in compliance with the sequential behaviors.

3 Experiments

In this section, we aim to answer five research questions: **RQ1**: How effective is PIGR compared with the state-of-the-art baselines? **RQ2**: How much do different components utilized by PIGR contribute to the whole model performance? **RQ3**: How is the capability of the methods in handling sessions with different lengths? **RQ4**: What is the influence of hyper-parameters on the PIGR? **RQ5**: What is the distribution of the interest number learned by PIGR?

Table 1: Detailed datasets statistics.

Datasets	Items num	Training num	Test num	Avg length	Length range
LastFM	24,699	799,884	206,723	17.26	Long
Gowalla	57,995	1,064,565	323,593	7.13	Medium
Yoochoose	17,390	312,527	91,428	4.24	Short

3.1 Experimental Setup

Datasets Processing. To study the property of the proposed framework PIGR, we conduct experiments on three real-world datasets LastFM[5], Gowalla[6] and Yoochoose[7] with different average length ranges. We summarize detailed dataset statistics in Table 1. Following previous experimental protocol [24], we filter sequences whose lengths are smaller than 2 in each dataset. Similar to [2], we use the most recent 20% of the original sequences as test sets and leave the rest as training set. And we split the last 20% subset of training set to tune hyper-parameters. Moreover, we apply the segmentation preprocessing technique to each sequence. For an anonymous sequence with elements $[s_1, s_2, \ldots, s_l]$, we generate a series of subsequence and label pairs for model input, i.e., $[[s_1], [s_2]]$, $[[s_1, s_2], [s_3]]$, ..., $[[s_1, s_2, \ldots, s_{l-1}], [s_l]]$.

Table 2: Overall performance comparison w.r.t. Recall@N and NDCG@N scores on the three benchmark datasets where p-value <0.01.

Methods	LastFM		Gowalla		Yoochoose	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
STAMP	22.53	9.95	35.38	19.57	59.35	32.10
MIND	25.73	15.74	25.13	13.55	54.86	27.30
Comirec-SA	16.48	6.95	27.06	16.81	53.46	26.24
CO-SAN	25.92	11.59	48.25	28.32	71.02	40.20
FGNN	25.59	11.65	47.19	27.76	67.14	36.72
GC-SAN	28.27	13.89	51.67	31.22	69.11	38.41
SR-GNN	25.47	12.20	49.29	29.51	68.51	38.32
LESSR	28.33	13.93	52.50	32.82	70.04	40.18
DHCN	27.35	12.47	52.79	31.04	69.52	38.73
PIGR w/ SI	29.02	13.86	52.70	31.78	71.21	40.64
PIGR w/ FG	28.75	13.18	51.18	31.65	70.25	39.20
PIGR	**31.07**	**14.53**	**54.73**	**33.54**	**71.81**	**40.92**
Improvement(%)	**+9.7%**	**+4.3%**	**+3.67%**	**+2.19%**	**+1.11%**	**+1.79%**

Experimental Settings. We consider the following representative methods to compare with PIGR. (i) To verify the usefulness of modeling distant item transition, two sequential models (STAMP [11], COSAN [12]) and two multi-interest models (MIND [9], Comirec-SA [1]) are included; (ii) To prove the superiority of learning personalized interest graph structure, GNN-based models

[5] http://ocelma.net/MusicRecommendationDataset/index.html

[6] https://snap.stanford.edu/data/loc-gowalla.html

[7] https://www.kaggle.com/datasets/chadgostopp/recsys-challenge-2015

(FGNN [14], GC-SAN [22], SR-GNN [19], LESSR [2], DHCN [20]) with predefined principles are included. Besides, we also incorporate two variants of PIGR to verify our motivation. PIGR with fixed-graph (PIGR w/ FG) removes Eqs. (1), (2), (3) and constructs a fixed graph using popular co-occurrence rules. PIGR with static interest (PIGR w/ SI) excludes Eqs. (4), (5), (6) and employs the self-attention technique to capture static long-term and short-term interests. For the implementation details, we implement PIGR with Pytorch, where the learning rate is set to 0.0005 and the batch size is set to 512. The Adam optimizer is adopted. We apply the grid search strategy following [20] to tune hyper-parameters based on the validation performance. Each method is independently run five times and reported the average performance. And we adopt two standard evaluation metrics **Recall@N** and **NDCG@N** to measure model performance.

3.2 Comparison with Baselines (RQ1 & RQ2)

Table 2 summarizes all methods performance in terms of Recall@20 and NDCG@20 scores on three datasets. We have the following observations. First, the traditional sequential methods generally behave worse than the GNN models. These cases confirm the necessity of modeling distant item transition in sessions and the power of graph neural networks. Second, the performance of graph neural network competitors is inferior to PIGR. These methods construct the session graph based on the manually predefined principles and may easily introduce unnecessary edges in the sequence. As the session length extends, the relationships among items are more complex than predefined principles. Therefore, PIGR outperforms GNN models by a large margin. Besides, PIGR extracts diverse interests from the learned session graph and consistently outperforms PIGR w/ SI. And PIGR takes advantage of the learnable graph-structured information and achieves better performance than PIGR w/ FG. These results suggest that a promising direction is to learn a personalized graph structure from the session to extract an adaptive number of interests.

3.3 The Influence of Session Length (RQ3)

Table 4: Length range definition on Gowalla dataset.

Length range	Min length	Max length	Number
Long	15	200	114,540
Medium	6	14	86,339
Short	1	5	122,714

To specifically explore previous baselines performance on sessions in different length ranges, we partition the prediction results of test sessions on Gowalla dataset following the definition in Table 4 in line with each session length. There is a relatively balanced sequence length distribution on Gowalla dataset and it could fairly manifest all models capability of coping with different length ranges. The separate results are reported in Table 3. First, graph

Table 3: Performance comparison with different session length ranges in terms of Recall@N and NDCG@N scores on Gowalla dataset.

Methods	Long		Medium		Short	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
STAMP	34.91	19.83	39.08	22.91	33.62	17.34
MIND	20.72	10.44	28.36	15.84	27.86	15.45
Comirec-SA	26.79	16.19	27.85	17.49	26.85	17.04
CO-SAN	51.80	29.80	49.98	29.06	45.36	27.10
FGNN	45.03	24.65	46.98	26.92	44.41	26.77
GC-SAN	54.54	32.07	54.03	32.87	47.70	29.47
SR-GNN	51.81	30.22	51.01	30.62	45.95	28.15
LESSR	55.73	33.97	55.26	34.90	47.91	30.48
DHCN	56.35	32.10	54.52	31.93	48.68	29.51
PIGR	**58.43**	**35.26**	**57.30**	**35.37**	**49.82**	**30.89**

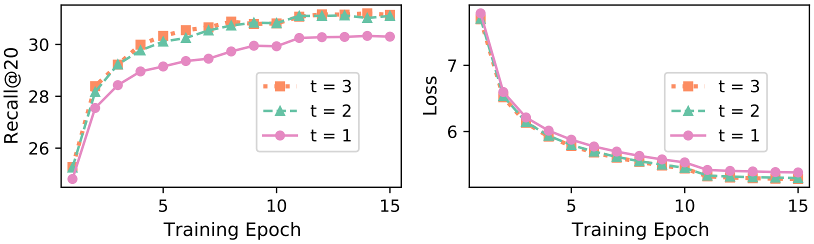

Fig. 3: Empirical training loss and corresponding Recall@20 score with different neighbor number t.

neural network models perform equally well or considerably better than the sequential models. It proves the importance of depicting item topological dependency. Second, PIGR has better improvement in the long sessions than in the short sessions and medium sessions. Since the user's complete preference is much more diverse in the long sessions, manually designing the graph structure is not an appropriate choice. It demonstrates the superiority of PIGR handing session lengths in different ranges.

3.4 Hyper-parameter Sensitivities (RQ4)

To evaluate the impact of hyper-parameters, we conduct two groups of hyper-parameter sensitivity experiments. In the first group, to study the impact of the neighbor number t on the training convergence rate, we draw Fig. 3, which characterizes the empirical training loss and performance in terms of Recall@20 scores curve over epoch on LastFM dataset. We have the following observations. First, they do not exhibit a faster convergence rate considering Recall scores

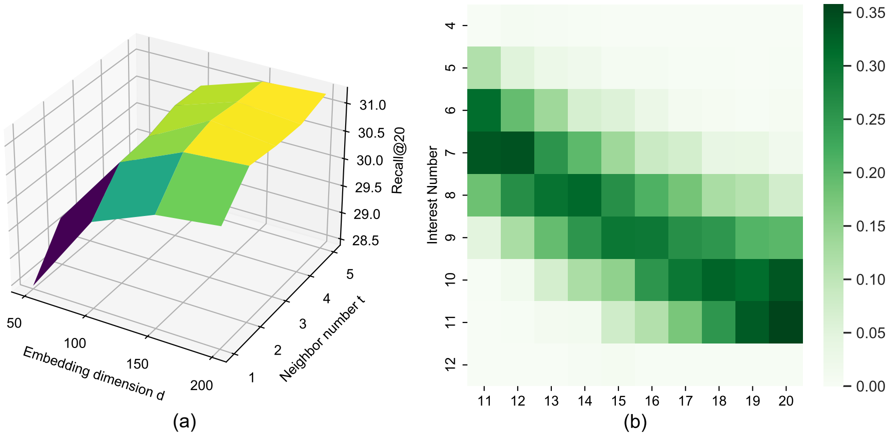

Fig. 4: (a) Test performance under joint impacts of embedding dimension d and neighbor number t. (b) Heat map of extracted interest probability distribution among session lengths.

ranging t from 1 to 3. Remarkably, the loss curve illustrates that growing neighbor number does not turn the model to converge to a better global minimum. Overall, these observations collectively indicate that increasing neighbor number enhances the model generalization ability and does not impact the convergence rate.

In the second group, we examine joint effects of two hyper-parameters: neighbor number t and embedding dimension d. The prediction results on LastFM dataset are drawn in three-dimensional map in Fig. 4(a). We observe that the performance is consistently better while continuously increasing embedding dimensions. And the model achieves substantial improvement when t and d equals 2 and 150 respectively and marginal improvement with larger values.

3.5 Adaptability Analysis (RQ5)

To prove that PIGR extracts an adaptive number of interests, we select all sessions whose lengths are longer than ten on Yoochoose dataset and calculate the extracted interest number. The visualization is shown in Fig. 4(b), where each cell represents the occurrence ratio of different interest numbers among the same session length. We observe that PIGR extracts different interest numbers within the same session length. And the assigned probability mass is not all concentrated in one cell, proving that PIGR learns an adaptive number of interests.

4 Related Work

Session-based Recommendations. Li et al. [10] incorporate an attention mechanism to calculate each item score to user current interest. Then Yuan et al. [24] incorporate α-entmax technologies to filter redundant items. Wu

et al. [19] introduce graph gated neural networks to fully explore topological properties of the sequence. Pan et al. [13] add star nodes to link nonadjacent items. However, several works think that current sequences only focus on explicit item dependencies in a single session, ignoring implicit global information between sessions. Wang et al. [18] propose to construct a global item-item graph based on each pair of item occurrences in all training sessions. To capture dynamic user preferences. Qiu et al. [15] propose to utilize a sample reservoir to store valuable samples while Zhou et al. [25] propose to capture temporal information.
Multi-interest Recommendations. A user's sequence may display different users' intents. A next item choice may be due to the influence of multiple interest factors. Cen et al. [1] propose two multi-interest extraction mechanisms: self-attention and dynamic routing. Similarly, Li et al. [9] leverage capsule network to model multi-interests of users at Tmall. Xiao et al. [21] propose a Transformer-based framework to capture diverse interests expressed by the user behaviors. Tan et al. [16] only select the most related k interests from the prototype pool for each user. Cho et al. [4] set up K general proxy to encode general interests shared by multiple sessions.

5 Conclusions and Future Work

In this paper, we propose a novel framework named PIGR for session-based recommendations. Compared with existing GNN-based solutions, PIGR learns a reasonable graph structure from the session instead of directly extracting the co-occurrence of item pairs as edges. This learning process is driven by extracting adaptive interest for each session. Items with similar representations learned by sequential behavior will be clustered to center around the same interest node. The different granularity of interests at each layer can be retrieved as one unified session-level vector of the user. Extensive experiments demonstrate that PIGR achieves significant performance improvement over the state-of-the-art baselines on real-world datasets. Our future work is to explore the applicability of embedding item multiple attributes to graph structure learning.

Acknowledgments. The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 25208322) and Key Research Project of Zhejiang Lab (No.2022PG0AC03).

References

1. Cen, Y., Zhang, J., Zou, X., Zhou, C., Yang, H., Tang, J.: Controllable multi-interest framework for recommendation. In: KDD. pp. 2942–2951 (2020)
2. Chen, T., Wong, R.C.W.: Handling information loss of graph neural networks for session-based recommendation. In: KDD. pp. 1172–1180 (2020)
3. Chen, W., Cai, F., Chen, H., de Rijke, M.: A dynamic co-attention network for session-based recommendation. In: CIKM. pp. 1461–1470 (2019)

4. Cho, J., Kang, S., Hyun, D., Yu, H.: Unsupervised proxy selection for session-based recommender systems. In: SIGIR. pp. 327–336 (2021)
5. Fang, H., Zhang, D., Shu, Y., Guo, G.: Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. TOIS pp. 1–42 (2020)
6. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)
7. Huang, C., Chen, J., Xia, L., Xu, Y., Dai, P., Chen, Y., Bo, L., Zhao, J., Huang, J.X.: Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation. In: AAAI. pp. 4123–4130 (2021)
8. Jannach, D., Ludewig, M.: When recurrent neural networks meet the neighborhood for session-based recommendation. In: RecSys. pp. 306–310 (2017)
9. Li, C., Liu, Z., Wu, M., Xu, Y., Zhao, H., Huang, P., Kang, G., Chen, Q., Li, W., Lee, D.L.: Multi-interest network with dynamic routing for recommendation at tmall. In: CIKM. pp. 2615–2623 (2019)
10. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: CIKM. pp. 1419–1428 (2017)
11. Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: Stamp: short-term attention/memory priority model for session-based recommendation. In: KDD. pp. 1831–1839 (2018)
12. Luo, A., Zhao, P., Liu, Y., Zhuang, F., Wang, D., Xu, J., Fang, J., Sheng, V.S.: Collaborative self-attention network for session-based recommendation. In: IJCAI. pp. 2591–2597 (2020)
13. Pan, Z., Cai, F., Chen, W., Chen, H., de Rijke, M.: Star graph neural networks for session-based recommendation. In: CIKM. pp. 1195–1204 (2020)
14. Qiu, R., Li, J., Huang, Z., Yin, H.: Rethinking the item order in session-based recommendation with graph neural networks. In: CIKM. pp. 579–588 (2019)
15. Qiu, R., Yin, H., Huang, Z., Chen, T.: Gag: Global attributed graph neural network for streaming session-based recommendation. In: SIGIR. pp. 669–678 (2020)
16. Tan, Q., Zhang, J., Yao, J., Liu, N., Zhou, J., Yang, H., Hu, X.: Sparse-interest network for sequential recommendation. In: WSDM. pp. 598–606 (2021)
17. Wang, S., Cao, L., Wang, Y., Sheng, Q.Z., Orgun, M.A., Lian, D.: A survey on session-based recommender systems. CSUR pp. 1–38 (2021)
18. Wang, Z., Wei, W., Cong, G., Li, X.L., Mao, X.L., Qiu, M.: Global context enhanced graph neural networks for session-based recommendation. In: SIGIR. pp. 169–178 (2020)
19. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: AAAI. pp. 346–353 (2019)
20. Xia, X., Yin, H., Yu, J., Wang, Q., Cui, L., Zhang, X.: Self-supervised hypergraph convolutional networks for session-based recommendation. In: AAAI. pp. 4503–4511 (2021)
21. Xiao, Z., Yang, L., Jiang, W., Wei, Y., Hu, Y., Wang, H.: Deep multi-interest network for click-through rate prediction. In: CIKM. pp. 2265–2268 (2020)
22. Xu, C., Zhao, P., Liu, Y., Sheng, V.S., Xu, J., Zhuang, F., Fang, J., Zhou, X.: Graph contextualized self-attention network for session-based recommendation. In: IJCAI. pp. 3940–3946 (2019)
23. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling (2018)
24. Yuan, J., Song, Z., Sun, M., Wang, X., Zhao, W.X.: Dual sparse attention network for session-based recommendation. In: AAAI. pp. 4635–4643 (2021)
25. Zhou, H., Tan, Q., Huang, X., Zhou, K., Wang, X.: Temporal augmented graph neural networks for session-based recommendations. In: SIGIR. pp. 1798–1802 (2021)