User-Event Graph Embedding Learning for Context-Aware Recommendation

Dugang Liu[1], Mingkai He[1], Jinwei Luo[1], Jiangxu Lin[2], Meng Wang[2], Xiaolian Zhang[3], Weike Pan[1*] and Zhong Ming[1*]

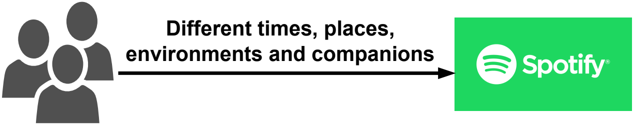
dugang.ldg@gmail.com, meng.wang@seu.edu.cn, zhangxiaolian@huawei.com, {panweike, mingz}@szu.edu.cn

[1]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

[2]Southeast University, Nanjing, China

[3]Huawei 2012 Lab, Shenzhen, China

Context-Aware Recommendation



- Previous studies have shown that users' behaviors are often influenced by the contextual information.
- Context-aware recommender systems (CARS) aim to integrate this contextual information to provide more fine-grained recommendations to users [Adomavicius et al., 2022].
- The performance of models for context-aware recommendation often heavily relies on the learning of the contextual features.

Two Sparsity Problems (1/2)

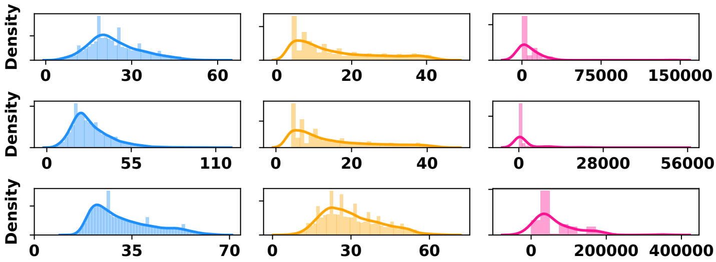


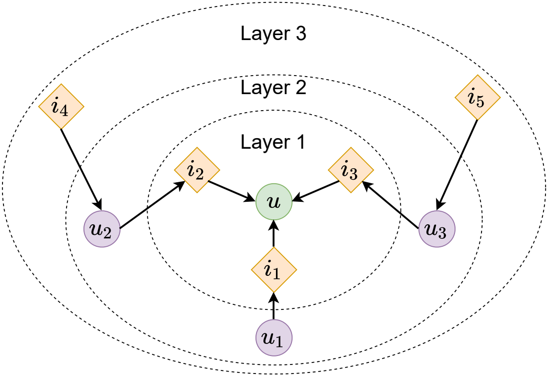
Figure: The distributions of the number of contextual features associated with each user (left column) and item (middle column), as well as the frequency statistics of the contextual features (right column), on Yelp-NC (top row), Yelp-OH (middle row) and Amazon-Book (bottom row)

Two Sparsity Problems (2/2)

- Feature sparsity: most of the contextual features have a low frequency.
    - It is often difficult for most existing methods to learn a good embedding for each of these sparse contextual features due to insufficient training examples.
- Interaction sparsity: most users (or items) are associated with only a few contextual features.
    - The existing methods may have a performance bottleneck for inactive users or unpopular items due to insufficient preference information w.r.t. the contextual features.
- As a result, the above two problems will cause the model performance to suffer from random initialization.
    - Note that, most methods for CARS focus on improving the feature interaction layer, but overlook the embedding layer.

Graph Representation vs. Data Sparsity

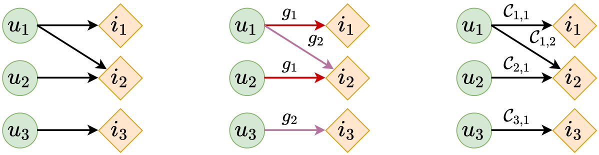
Previous works have shown that graph representation can effectively alleviate the data sparsity by learning higher-order information [Wang et al., 2019, Guo et al., 2021].



Graph Structure

However, how to better introduce the contextual information into the recommendation-oriented graph structures is not clear.

- The contextual information is pre-clustered, and the categories are used as different types of relation edges [Zhao et al., 2021].
- The contextual information is treated as different attributes of the relation edges [Wu et al., 2022].
- ...



[Zhao et al., 2021] **[Wu et al., 2022]**

Our Goal

- Design a new graph structure with contextual information, i.e., user-event graph (UEG).
- Design a new model to obtain refined embeddings for all features, i.e., user-event graph embedding learning (UEG-EL).
- The obtained refined embeddings are used for the embedding layer of a downstream recommendation model.

Notations

- A set of M users $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$, and a set of J fields of user attributes $\mathcal{A} = \{\mathcal{A}^1, \mathcal{A}^2, \ldots, \mathcal{A}^J\}$.
- A set of N items $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, and a set of K fields of item attributes $\mathcal{B} = \{\mathcal{B}^1, \mathcal{B}^2, \ldots, \mathcal{B}^K\}$
- A set of R fields of contextual features $\mathcal{C} = \{\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^R\}$.
- Let $\mathcal{S} = \{(\mathbf{s}_1, y_1), (\mathbf{s}_2, y_2), \ldots, (\mathbf{s}_I, y_I)\}$ denote a set of I user-item interactions and their corresponding labels, an instance of which can be represented as,

$$\mathbf{s}_i = \left[u^i, v^i, \mathbf{A}_{u^i}, \mathbf{B}_{v^i}, \mathbf{C}^i\right], \tag{1}$$

where $u^i \in \mathcal{U}$, $v^i \in \mathcal{V}$, $\mathbf{A}_{u^i} \subset \mathcal{A}$, $\mathbf{B}_{v^i} \subset \mathcal{B}$ and $\mathbf{C}^i \subset \mathcal{C}$.

Base Model

We refer to factorization machine (FM) as the base model in this paper.

- Initial Embedding: $\mathbf{s}_i = \left[u^i, v^i, \mathbf{A}_{u^i}, \mathbf{B}_{v^i}, \mathbf{C}^i\right] \Rightarrow \mathbf{E}_{\mathbf{s}_i} = [\mathbf{e}_{u^i}, \mathbf{e}_{v^i}, \mathbf{e}_{\mathbf{A}_u}, \mathbf{e}_{\mathbf{B}_v}, \mathbf{e}_{\mathbf{C}^i}] \Rightarrow \mathbf{E}_{\mathbf{s}_i} = [\mathbf{e}'_{u^i}, \mathbf{e}'_{v^i}, \mathbf{e}_{\mathbf{C}^i}]$, where $\mathbf{e}'_u = mean\_pooling([\mathbf{e}_u, \mathbf{e}_{\mathbf{A}_u}]), \mathbf{e}'_v = mean\_pooling([\mathbf{e}_v, \mathbf{e}_{\mathbf{B}_v}])$.
- Feature Interaction: $\hat{y}(\mathbf{s}_i) = \sigma(b_g + \sum b_\star + \frac{1}{2}[(\sum \mathbf{e}_\star)^2 - \sum \mathbf{e}_\star^\top \mathbf{e}_\star])$.
- Model Training: $\mathcal{L} = -\frac{1}{|\mathcal{S}'|} \sum_{(\mathbf{s}_i, y) \in \mathcal{S}'} y_i \log \hat{y}(\mathbf{s}_i) + (1 - y_i) \log(1 - \hat{y}(\mathbf{s}_i))$, where $\mathcal{S}' = \mathcal{S} \cup \mathcal{S}^-$, and $\mathcal{S}^-$ is a set of negative instances.

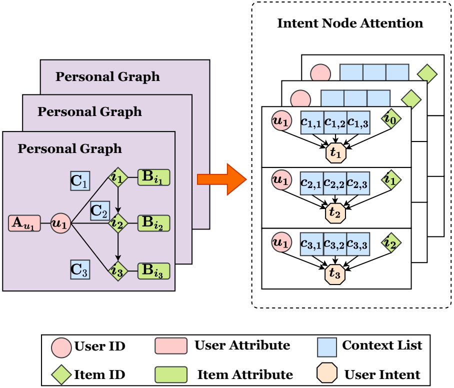
Recall that our goal is $\mathbf{E}_{\mathbf{s}_i} = [\mathbf{e}'_{u^i}, \mathbf{e}'_{v^i}, \mathbf{e}_{\mathbf{C}^i}] \Rightarrow \mathbf{P}_{\mathbf{s}_i} = [\hat{\mathbf{p}}_{u^i}, \hat{\mathbf{p}}_{v^i}, \hat{\mathbf{p}}_{\mathbf{C}^i}]$, where the latter is obtained by our UEG-EL.

Graph Construction (1/6)

As the most relevant work to ours, GCM [Wu et al., 2022] models the contextual features as edge features between users and items. We argue this may limit the recommendation performance.

- The contextual features cannot benefit from the information propagation process of graph embedding learning as users and items do.
- It is difficult to accurately capture a user's intent, i.e., to identify the subset from the current contextual features that triggers the user's interaction event.

Graph Construction (2/6)



The graph structure adopted by GCM [Wu et al., 2022] can be viewed as a personal graph.
To address the aforementioned two sparsity problems, we first propose intent node attention (INA) to capture the user intent in each instance and generate some additional intent nodes $\mathcal{T} = \{t^1, t^2, \ldots, t^l\}$.

Graph Construction (3/6)

Suppose the length of the context features in each instance is Z. Since a user's behavior may also be influenced by the preceding behaviors rather than the context alone, we additionally consider the user's last interacted item before the current instance in INA. We use c^i_{Z+1} to denote this particular item for the sake of notational brevity.

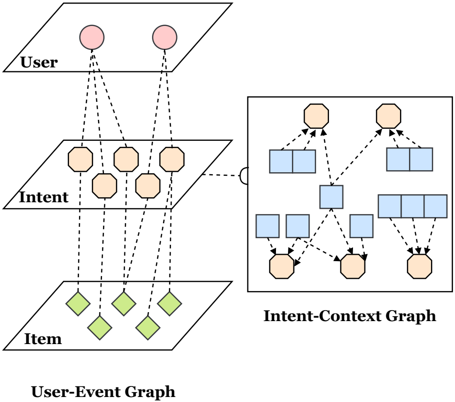
The intent node attention of an instance is computed as follows,

$$\alpha^i_z = \mathrm{Softmax}(\mathbf{W}_0^\top \mathrm{Relu}(\mathbf{W}_1 \mathbf{e}'_{u^i} + \mathbf{W}_2 \mathbf{e}_{c^i_z} + \mathbf{b}_1)). \quad (2)$$

$$\mathbf{e}_{t^i} = \sum_{z=1}^{Z+1} \alpha^i_z \mathbf{e}_{c^i_z}, \quad (3)$$

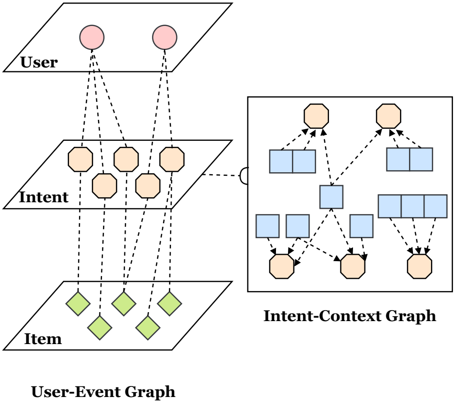
where $\mathbf{e}_{t^i}$ is the embedding representation of the intent node corresponding to this instance.

Graph Construction (4/6)



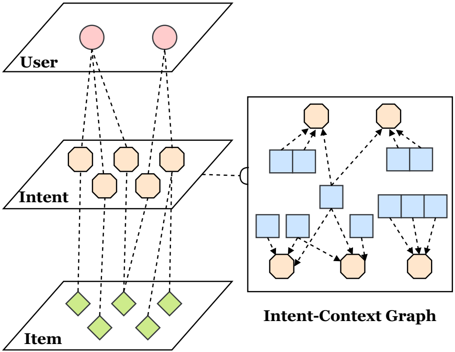
Based on the obtained intent nodes, we can then build a user-event graph (UEG) and an intent-context graph (ICG). By using contextual features to model user intent nodes, which in turn act as hubs to generate connections between users and items, we expect to obtain a better solution for context-aware recommendation.

Graph Construction (5/6)



In ICG, each intent node is associated with Z contextual features, which are used to compute this intent node in INA. Each edge between an intent node and a contextual feature has an intent attention as a weight, i.e., α_z^i. This means that if a contextual feature has a large weight in the generation of this intent node, it will get more emphasis when the information is propagated.

Graph Construction (6/6)



User-Event Graph

In addition, the user-event graph is also potentially helpful to other recommendation tasks via some extensions. For example, we can integrate user behavior types or data from different domains as a special contextual feature into the user-event graph.

User-Event Collaborative Graph Convolution

- Existing graph embedding learning techniques are not applicable to our user-event graph due to its peculiar structure, i.e., the intent nodes. Therefore, we introduce user-event collaborative graph convolution to exploit the power of the user-event graph.
- Specifically, we aim to fully exploit the pivotal role of the intent nodes to explore the connections among the users, items, and contextual features, especially to identify a focused subset of users in contextual features.
- This in turn feeds back into intent node attention to get a more accurate node embedding. We can see that the first two modules of our framework, i.e., intent node attention and collaborative graph convolution, are synergistic.

Information Propagation for the Users

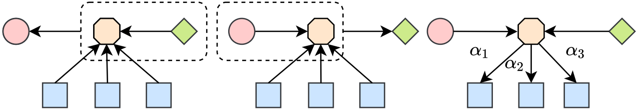


Figure: An illustration of information propagation for the nodes of users, items and contextual features in user-event collaborative graph convolution.

We use the intent nodes as a hub to propagate the information about the items and contextual features to the users, where *h* is the layer index,

$$\mathbf{p}_{u^i,i}^{(h)} = \mathbf{p}_{v^i}^{(h-1)} + \mathbf{p}_{t^i}^{(h-1)}. \tag{4}$$

$$\mathbf{p}_{u}^{(h)} = \frac{1}{\sqrt{|\{i|u^i = u\}|}} \sum_{i,u^i=u} \mathbf{p}_{u,i}^{(h)}. \tag{5}$$

$$\hat{\mathbf{p}}_u = \frac{1}{H+1} \sum_{h=0}^{H} \mathbf{p}_u^{(h)}. \tag{6}$$

Information Propagation for the Items

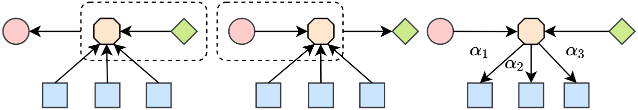


Figure: An illustration of information propagation for the nodes of users, items and contextual features in user-event collaborative graph convolution.

We use the intent nodes as a hub to propagate the information about the users and contextual features to the items, where h is the layer index,

$$\mathbf{p}_{v^i,i}^{(h)} = \mathbf{p}_{u^i}^{(h-1)} + \mathbf{p}_{t^i}^{(h-1)}. \tag{7}$$

$$\mathbf{p}_v^{(h)} = \frac{1}{\sqrt{|\{i|v^i = v\}|}} \sum_{i,v^i=v} \mathbf{p}_{v,i}^{(h)}. \tag{8}$$

$$\hat{\mathbf{p}}_v = \frac{1}{H+1} \sum_{h=0}^{H} \mathbf{p}_v^{(h)}. \tag{9}$$

Information Propagation for the Context (1/2)

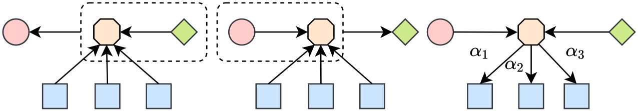


Figure: An illustration of information propagation for the nodes of users, items and contextual features in user-event collaborative graph convolution.

The information is first propagated to an intent node, and the contextual features receive different information sent by the intent nodes according to the attention distribution,

$$\mathbf{p}_{t^i,i}^{(h)} = \mathbf{p}_{u^i}^{(h-1)} + \mathbf{p}_{v^i}^{(h-1)}, \quad \mathbf{p}_{c_z^i,i}^{(h)} = \alpha_z^i \mathbf{p}_{t^i,i}^{(h)}. \tag{10}$$

$$\mathbf{p}_{c_z}^{(h)} = \frac{1}{\sqrt{|\{i | c_z^i = c_z\}|}} \sum_{i, c_z^i = c_z} \mathbf{p}_{c_z^i,i}^{(h)}. \tag{11}$$

$$\hat{\mathbf{p}}_{c_z} = \frac{1}{H+1} \sum_{h=0}^{H} \mathbf{p}_{c_z}^{(h)}. \tag{12}$$

Information Propagation for the Context (2/2)

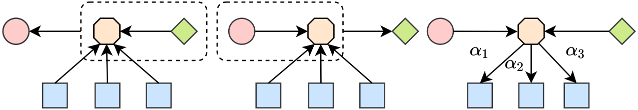


Figure: An illustration of information propagation for the nodes of users, items and contextual features in user-event collaborative graph convolution.

Note that after getting $\mathbf{p}_{c_z}^{(h)}$, we need to feed the information back to the intent node to get the embedding of the intent node at layer $h+1$,

$$\mathbf{p}_{t^i}^{(h+1)} = \sum \alpha_z^i \mathbf{p}_{c_z^i}^{(h)}. \tag{13}$$

Finally, we can obtain a set of corresponding refined embeddings,

$$\mathbf{P}_{\mathbf{s}_i} = [\hat{\mathbf{p}}_{u^i}, \hat{\mathbf{p}}_{v^i}, \hat{\mathbf{p}}_{\mathbf{C}^i}]. \tag{14}$$

Pruning the Information Propagation of Context

In practice, when the contextual features are associated with too many instances, aggregating the information of all instances may suffer from noise. Therefore, we further propose a simple but effective variant (UEG-EL-V), by pruning the information propagation of the context,

- For each context c_z, we first obtain a mean vector of the intent node embeddings for all instances associated with it.
- Then, we compute the distances of these instances from the mean vector, and use a pre-set pruning rate θ to remove those instances with larger distances.
- The idea behind this pruning operation is that the mean vector represents the majority intent among instances with this contextual feature, and instances that are farther away from it are more likely to be noise with different intents.

Complexity Analysis

- Since the refined embeddings can be obtained in offline training and be directly used for online inference, the time complexity of our UEG-EL in this case is the same as that of the base model.
- For model training, user-event collaborative graph convolution dominates the time cost, and its computational complexity is $\mathcal{O}(Z \cdot |\mathcal{G}_{ueg}| \cdot d)$, where $|\mathcal{G}_{ueg}|$ denotes the number of edges in $\mathcal{G}_{ueg}$.
- Compared with existing graph recommendation models, the complexity is linear with the number of the edges between the intent nodes and contextual features. Therefore, to speed up model training, a pre-filtering of the contextual feature set, and a pruning operation are necessary.

Datasets

Table: Statistics of the processed datasets.

Dataset	Yelp-NC	Yelp-OH	Amazon-Book
#User	6,336	5,170	44,709
#Item	13,003	12,997	46,831
#Instance	185,408	143,884	1,174,785
#User Attribute	24	24	-
#Item Attribute	68	213	24,816
#Contextual Feature	206	350	69

Baselines

- Matrix factorization (MF) [Koren et al., 2009]
- Factorization machine (FM) [Rendle et al., 2011]
- Neural factorization machines (NFM) [He and Chua, 2017]
- xDeepFM [Lian et al., 2018]
- LightGCN [He et al., 2020]
- Graph intention network (GIN) [Li et al., 2019]
- Graph convolution machine (GCM) [Wu et al., 2022]

Implementation Details

- For all baselines, we use the open-source implementation and parameter settings provided in [Wu et al., 2022], where the embedding size is set to 64, the batch size is set to 2048, the learning rate is set to 0.001, and Adam is used as the optimizer.
- For our method, we tune the number of GNN layers H in the range of $\{1, 2, 3\}$, the L2 regularization term in the range of $\{1e^{-1}, 1e^{-2}, 1e^{-3}, 1e^{-4}, 1e^{-5}\}$, and the pruning rate θ in the range of $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, and set the other parameters the same as the baselines.
- We perform grid search to tune the hyper-parameters by evaluating the summation of HR@10 and NDCG@10.
- To avoid over-fitting, we also adopt an early stopping strategy with the patience set to 50 times.

RQ1: Performance Comparison (1/3)

Table: Results on all datasets, where the best and second best results are marked in bold and underlined, respectively. Note that * indicates a significance level of $p \leq 0.05$ based on two-sample t-test between our method and the best baseline.

Dataset	Yelp-NC				Yelp-OH			
Metrics	HR@10	HR@50	NDCG@10	NDCG@50	HR@10	HR@50	NDCG@10	NDCG@50
MF	0.0384	0.1173	0.0175	0.0341	0.0429	0.1261	0.0206	0.0383
FM	0.0739	0.1804	0.0396	0.0624	0.1959	0.4201	0.1049	0.1538
NFM	0.0587	0.1477	0.0323	0.0514	0.2248	0.4836	0.1161	0.1725
xDeepFM	0.0851	0.2086	0.0458	0.0723	0.2296	0.4799	0.1218	0.1762
LightGCN	0.0499	0.1394	0.0241	0.0431	0.0518	0.1520	0.0249	0.0461
GIN	0.0866	0.2175	0.0449	0.0722	0.2304	0.4965	0.1238	0.1818
GCM	0.1042	0.2451	0.0546	0.0850	0.2584	0.5147	0.1428	0.1990
UEG-EL	**0.1067***	<u>0.2470</u>	<u>0.0570</u>	<u>0.0875</u>	<u>0.2656</u>	<u>0.5350</u>	<u>0.1481</u>	**0.2073***
UEG-EL-V	<u>0.1062</u>	**0.2476***	**0.0572***	**0.0878***	**0.2706***	**0.5354***	**0.1484***	<u>0.2063</u>

RQ1: Performance Comparison (2/3)

Table: Results on all datasets, where the best and second best results are marked in bold and underlined, respectively. Note that * indicates a significance level of $p \leq 0.05$ based on two-sample t-test between our method and the best baseline.

Dataset	Amazon-Book			
Metrics	HR@10	HR@50	NDCG@10	NDCG@50
MF	0.0402	0.1243	0.0203	0.0382
FM	0.0587	0.1477	0.0323	0.0514
NFM	0.0808	0.1954	0.0444	0.0692
xDeepFM	0.0886	0.2119	0.0481	0.0748
LightGCN	0.0543	0.1466	0.0274	0.0473
GIN	0.0939	0.2189	0.0502	0.0774
GCM	0.0983	0.2222	0.0550	0.0819
UEG-EL	<u>0.0992</u>	<u>0.2385</u>	<u>0.0555</u>	<u>0.0857</u>
UEG-EL-V	**0.1112***	**0.2555***	**0.0623***	**0.0936***

RQ1: Performance Comparison (3/3)

- FM-based methods can capture the relationship between users, items and contexts, and achieve better performance than MF, which indicates the key role of contextual features in CARS.
- For GNN-based methods, LightGCN has a better result than MF, which shows the advantage of graph neural network. GIN mines user intent based on co-occurrence item graph and achieves a better result, indicating that the modeling of user intent is helpful. By introducing contextual features into the bipartite graph as edge features, GCM performs the best among the baselines.
- Our UEG-EL consistently outperforms all baselines. We also note that our UEG-EL has a relatively small improvement on Amazon-Book with a large number of instances and fewer contextual features, and our UEG-EL-V has a significant performance gain. This validates the effectiveness of UEG-EL and its variant.

RQ2: Ablation Study (1/3)

Table: Results of the ablation studies on all datasets, where the best results are marked in bold.

Dataset	Yelp-NC				Yelp-OH			
Metrics	HR@10	HR@50	NDCG@10	NDCG@50	HR@10	HR@50	NDCG@10	NDCG@50
UEG-EL	**0.1067**	**0.2470**	**0.0570**	**0.0875**	**0.2656**	**0.5350**	**0.1481**	**0.2073**
w/o CGC (H=1)	0.1012	0.2238	0.0544	0.0810	0.2567	0.5104	0.1398	0.1950
w/o CGC (H=2)	0.0901	0.2268	0.0474	0.0769	0.2147	0.4219	0.1158	0.1608
w/o CGC, INA (H=1)	0.0974	0.2252	0.0514	0.0788	0.2489	0.5075	0.1358	0.1920
w/o CGC, INA (H=2)	0.1042	0.2451	0.0546	0.0850	0.2584	0.5147	0.1428	0.1990

RQ2: Ablation Study (2/3)

Table: Results of the ablation studies on all datasets, where the best results are marked in bold.

Dataset	Amazon-Book			
Metrics	HR@10	HR@50	NDCG@10	NDCG@50
UEG-EL	**0.0992**	**0.2385**	**0.0555**	**0.0857**
w/o CGC (H=1)	0.0983	0.2211	0.0551	0.0818
w/o CGC (H=2)	0.0860	0.1954	0.0481	0.0717
w/o CGC, INA (H=1)	0.0817	0.1940	0.0450	0.0693
w/o CGC, INA (H=2)	0.0983	0.2222	0.0550	0.0819

RQ2: Ablation Study (3/3)

- **'w/o CGC (H=1)' vs. 'w/o CGC, INA (H=1)'.** UEG-EL without CGC beats UEG-EL without CGC and INA, indicating that our proposed INA can well capture the different user intents on each sample.
- **'w/o CGC (H=2)' vs. 'w/o CGC (H=1)'.** UEG-EL without CGC using 2-layer GNN will be weaker than 1-layer, which means that the traditional convolution method is not suitable for the case with INA. However, UEG-EL without CGC and INA still performs well when H=2, which motivates us to design a new convolution mode, i.e., our CGC.
- **UEG-EL vs. 'w/o CGC, INA (H=2)', 'w/o CGC (H=2)'.** UEG-EL achieves the best results, which means that the proposed new convolution mode CGC can help INA to capture user intent better, i.e., our UEG-EL is synergistic. In particular, CGC can enable a context node to obtain effective high-order collaborative information on the basis of INA.

RQ3: Effectiveness of UEG-EL-V (1/2)

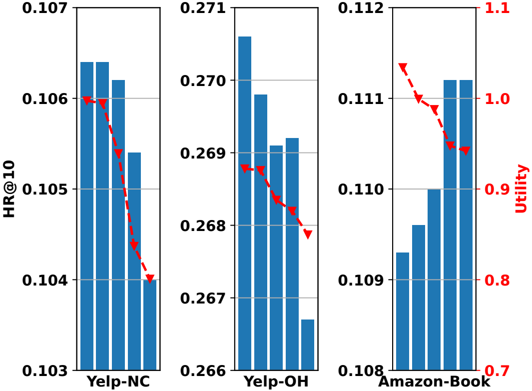
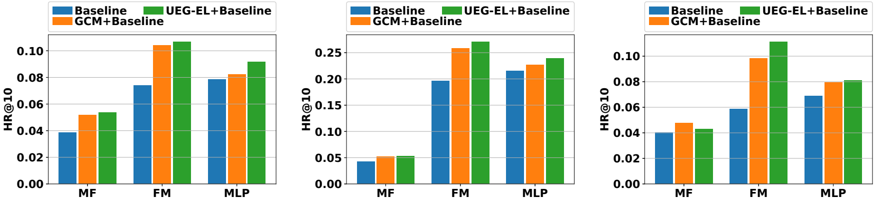


Figure: Recommendation performance of UEG-EL-V and the ratio of its computation time compared with that of UEG-EL, where the values on the x-axis (i.e., the pruning rates) are 0.1, 0.3, 0.5, 0.7 and 0.9, respectively.

RQ3: Effectiveness of UEG-EL-V (2/2)

- We can see that as the pruning rate increases, the effect of our UEG-EL-V on Yelp-NC and Yelp-OH gradually decreases. The reason is that they have a relatively small number of samples, causing the pruning operation to exclude a lot of context and our UEG-EL-V can thus not learn the appropriate high-order collaborative information well.
- On Amazon-Book, the performance improves as the pruning rate increases, indicating that our proposed pruning method can effectively alleviate the influence of noise in aggregating information for contextual features when a large number of instances are available.
- On all datasets, we find that when the pruning rate increases, the evaluation time becomes shorter, which is a merit of a typical pruning strategy in improving the efficiency.

RQ4: Compatibility Evaluation of UEG-EL (1/2)



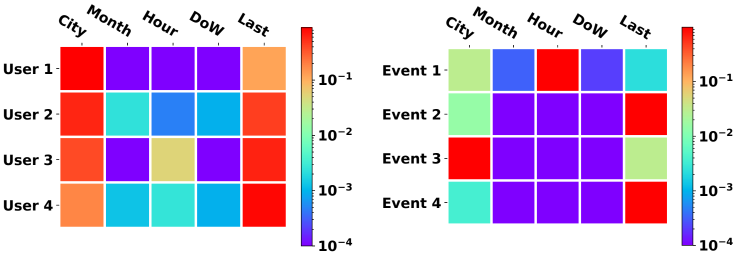
(a) Yelp-NC (b) Yelp-OH (c) Amazon-Book

Figure: Recommendation performance of GCM and our UEG-EL with different downstream models, i.e., MF, FM and MLP, on the three datasets.

RQ4: Compatibility Evaluation of UEG-EL (2/2)

- Our UEG-EL outperforms the base downstream model that uses the initial embedding vector of user, item and context in all cases and outperforms GCM in most cases. This suggests that our UEG-EL can be used as a general framework to improve the performance of different downstream recommendation models.
- Note that 'UEG-EL+MF' does not exceed 'GCM+MF' on Amazon-Book. The reason is that MF only uses the refined vectors of the users and items, and if we do not use the refined vectors of the context in a downstream model, the user intent in our INA may not be learned well. In contrast, both our 'UEG-EL+FM' and 'UEG-EL+MLP' exceed 'GCM+FM' and 'GCM+MLP' when the refined vectors of the context are used.

RQ5: In-depth Analysis of UEG-EL (1/4)



(a) global user intent of the four users

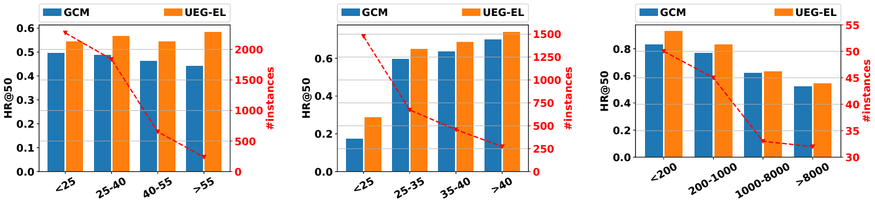
(b) fine-grained intent of user 3

Figure: Visualization of the attention of our UEG-EL w.r.t. some users in Yelp-OH.

RQ5: In-depth Analysis of UEG-EL (2/4)

- We can find that the users' intents are different, e.g., user 1 and 2 pay more attention to the context '*city*', while user 3 and 4 pay more attention to the last interacted items. The temporal context '*month*', '*hour*' and '*day_of_week* (DoW)' are often of low values, but user 3 is sensitive to '*hour*'.
- we can see that this user has different intents in different events. For event 1, the user is more inclined to interact at a certain moment, which is likely to be a habitual behavior. For events 2 and 4, the user chooses to interact because it is an item similar to the most recently interacted ones. For event 3, it is likely because of an item that the user interacts with in a certain city.
- In general, different users have different intents in different contexts, and a user's intents for the contexts vary in different interactions, which justifies the necessity of our proposed INA.

RQ5: In-depth Analysis of UEG-EL (3/4)



(a) u-c interaction (b) i-c interaction (c) context frequency

Figure: Recommendation performance of our UEG-EL with different user-context (u-c) interaction levels, item-context (i-c) interaction levels, and context frequencies on Yelp-OH.

RQ5: In-depth Analysis of UEG-EL (4/4)

- We can find that our UEG-EL has a significant improvement on all groups. In particular, when a user associates more contextual features, our UEG-EL has a better result since the proposed INA can effectively identify the attention subset of this user.
- Our UEG-EL has a more significant improvement on item groups associated with fewer contextual features.
- We can see a bigger boost with our UEG-EL on groups with lower frequencies.
- The above results demonstrate that our UEG-EL can effectively alleviate these two key challenges in CARS.

Conclusions

- We propose a novel user-event graph embedding learning (UEG-EL) framework to address two sparsity challenges in existing context-aware recommendation models.
- Our UEG-EL includes three modules, i.e., a graph construction module for obtaining the user-event graph, a user-event collaborative graph convolution module for refining the embeddings of all features, and a recommendation module to improve the performance of some existing context-aware recommendation model using the refined embeddings.
- Finally, we conduct extensive experiments on three real-world datasets to verify the effectiveness and compatibility of our solution.

Thank You!

- We thank the anonymous reviewers for their expert and constructive comments and suggestions.
- We thank the support of National Natural Science Foundation of China (NSFC) Nos. 61836005 and 62172283.
- We thank Mr. Wei Guo for his helpful discussions.

- Codes and slides are available at: `https://github.com/dgliu/KDD22_UEG`
- If you have any questions, please feel free to contact us.

Adomavicius, G., Bauman, K., Tuzhilin, A., and Unger, M. (2022).
Context-aware recommender systems: From foundations to recent developments.
In *Recommender Systems Handbook*, pages 211–250.

Guo, W., Su, R., Tan, R., Guo, H., Zhang, Y., Liu, Z., Tang, R., and He, X. (2021).
Dual graph enhanced embedding neural network for CTR prediction.
In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 496–504.

He, X. and Chua, T.-S. (2017).
Neural factorization machines for sparse predictive analytics.
In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 355–364.

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., and Wang, M. (2020).
LightGCN: Simplifying and powering graph convolution network for recommendation.
In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 639–648.

Koren, Y., Bell, R., and Volinsky, C. (2009).
Matrix factorization techniques for recommender systems.
Computer, 42(8):30–37.

Li, F., Chen, Z., Wang, P., Ren, Y., Zhang, D., and Zhu, X. (2019).
Graph intention network for click-through rate prediction in sponsored search.
In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 961–964.

Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., and Sun, G. (2018).
xDeepFM: Combining explicit and implicit feature interactions for recommender systems.
In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1754–1763.

Rendle, S., Gantner, Z., Freudenthaler, C., and Schmidt-Thieme, L. (2011).

Fast context-aware recommendations with factorization machines.
In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 635–644.

Wang, X., He, X., Wang, M., Feng, F., and Chua, T.-S. (2019).
Neural graph collaborative filtering.
In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174.

Wu, J., He, X., Wang, X., Wang, Q., Chen, W., Lian, J., and Xie, X. (2022).
Graph convolution machine for context-aware recommender system.
Frontiers of Computer Science, 16(6):1–12.

Zhao, Y., Wang, X., Chen, J., Wang, Y., Tang, W., He, X., and Xie, H. (2021).
Time-aware path reasoning on knowledge graph for recommendation.
ACM Transactions on Information Systems.