Learning Multi-granularity Consecutive User Intent Unit for Session-based Recommendation

Jiayan Guo1,*, Yaming Yang2, Xiangchen Song3,*, Yuan Zhang1, Yujing Wang2, Jing Bai4, Yan Zhang1

1School of Artificial Intelligence, Peking University, China, 2Microsoft Research Lab Asia, China, 3Carnegie Mellon University, USA, 4Microsoft, China

{guojiayan,yuan.z,zyhzhy001}@pku.edu.cn
{yayaming,yujwang,jbai}@microsoft.com
{xichensong}@cmu.edu

ABSTRACT

Session-based recommendation aims to predict a user’s next action based on previous actions in the current session. The major challenge is to capture authentic and complete user preferences in the entire session. Recent work utilizes graph structure to represent the entire session and adopts Graph Neural Network (GNN) to encode session information. This modeling choice has proved to be effective and achieved remarkable results. However, most of the existing studies only consider each item within the session independently and do not capture session semantics from a high-level perspective. Such limitation often leads to severe information loss and increases the difficulty of capturing long-range dependencies within a session.

Intuitively, compared with individual items, a session snippet, i.e., a group of locally consecutive items, is able to provide supplemental user intents which are hardly captured by existing methods. In this work, we propose to learn multi-granularity consecutive user intent unit to improve the recommendation performance. Specifically, we creatively propose Multi-granularity Intent Heterogeneous Session Graph (MIHSG) which captures the interactions between different granularity intent units and relieves the burden of long-dependency. Moreover, we propose the Intent Fusion Ranking (IFR) module to compose the recommendation results from various granularity user intents. Compared with current methods that only leverage intents from individual items, IFR benefits from different granularity user intents to generate more accurate and comprehensive session representation, thus eventually boosting recommendation performance. We conduct extensive experiments on five session-based recommendation datasets and the results demonstrate the effectiveness of our method. Compared to current state-of-the-art methods, we achieve as large as 10.21% gain on HR@20 and 15.53% gain on MRR@20. Code is available at https://github.com/SpaceLearner/SessionRec-pytorch

1 This work was done when Jiayan Guo and Xiangchen Song were interning at Microsoft Research Lab Asia, Beijing, China.

CCS CONCEPTS
• Information systems → Recommender systems.

KEYWORDS
Recommender System, Session-based Recommendation, Graph Neural Networks

ACM Reference Format:
Jiayan Guo1,*, Yaming Yang2, Xiangchen Song3,*, Yuan Zhang1, Yujing Wang2, Jing Bai4, Yan Zhang1. 2022. Learning Multi-granularity Consecutive User Intent Unit for Session-based Recommendation. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM ’22), February 21–25, 2022, Tempe, AZ, USA. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3488560.3498524

1 INTRODUCTION

With the rapid growth of the amount of information on the Internet, massive products, contents, and services (which are uniformly described as items) are emerging every day. It becomes difficult for users to view all items due to the time limit. Thus Recommender Systems (RS) have played an important role in helping make efficient and satisfying choices. As is pointed out, user preference tends to be dynamic, hence short-term user history captures more accurate user intent [32]. In such scenario, session-based recommendation [10, 17, 19, 24, 30], which encapsulates a range of latest consecutive user-item interactions as sessions, draws increasing attention and leads to better performance.

Traditional session-based methods treat each session as a sequence of items sorted by click time and widely adopt Recurrent Neural Networks (RNNs) [8, 12, 26] to solve the recommendation problem. Although remarkable performance has been achieved, these methods are arguably insufficient to obtain accurate user representations in sessions and neglect complex transitions of items [37]. Instead, recent work [1, 7, 16, 29, 37, 40] utilizes graph structure to represent the session and employ Graph Neural Networks (GNNs) [11, 44] to conduct information propagation between adjacent items. Results on various academic session-based recommendation datasets demonstrate the significant superiority of GNN based methods over traditional methods.

However, most of these GNN based methods only take individual items as basic units to extract user preference and ignore the rich hidden user intents revealed by a group of consecutively adjacent items (which is modeled as intent units). Such intent units encapsulate local context-aware user preferences, thereby enabling the recommendation system to generate comprehensive and accurate recommendation results in multiple granularities. For
To better exploit CIU, we propose \textit{Multi-granularity Intent Heterogeneous Session Graph} (MIHSG). In this heterogeneous graph \cite{39}, nodes (CIUs) with different numbers of items are categorized into different groups and the transition edges among the same type of nodes capture the spatial continuity of the user-item interactions in the corresponding intent granularity. We also introduce a special type of edge to represent the transition between high order intent units and the single items, which explicitly encode the intent evolution between coarse and fine-grained granularities. Compared with existing session graphs, the proposed heterogeneous graph can extract richer user preference, which contributes to alleviate the information loss during the graph modeling process. We then apply Heterogeneous Graph Attention network (HGAT) to extract the node representations \cite{15, 21, 28, 33, 41, 43}. Since the high-level intents can provide compressed yet accurate session intent information, such information can be propagated more effectively through different intent units. The long-range dependency problem can thus be alleviated. To solve the second problem, we propose \textit{Intent Fusion Ranking} (IFR), i.e., utilizing last attention to learn session representation for each intent granularity level and compose recommendation results to predict the next item. We name our model MSGIFSR (\textit{Multi-granularity Intent Heterogeneous Session Graph and Intent Fusion Ranking} for \textit{Session-based Recommendation}). Empirically, our method significantly improves the model’s performance by enriching contextual user intent within one session and achieves state-of-the-art performance on five benchmark datasets.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{An example of hidden user intents revealed by groups of consecutive items. A session (“Flour”, “Butter”, “Eggs”) is given, “Fry Eggs”, “Breakfast” and “Bake Butter Cakes” are user intents revealed by grouping different consecutive items in the session. Items in the right are possible items the user will click next under each intent.}
\end{figure}

In summary, our contributions in this paper are as follows:

- We propose consecutive intent unit (CIU) to extract user intent from different granularities. Newly formed consecutive intent units contain more accurate user preference and provide supplementary information for recommendation.
- To better exploit CIU, we propose \textit{Multi-granularity Intent Heterogeneous Session Graph} (MIHSG) to model complex transition relationship between different granular intent units explicitly. The MIHSG can propagate information efficiently, especially in long sessions. We also propose \textit{Intent Fusion Ranking} (IFR) strategy to fully utilize the intent representation from all granularity levels to enhance the recommendation performance.
- We compare the performance of MSGIFSR against state-of-the-art baselines on five public benchmark datasets. The results show the superiority of MSGIFSR over the state-of-the-art models, i.e., as large as 10.21\% gain on HR@20 and 15.53\% gain on MRR@20.

\section{RELATED WORK}

In this section, we review the related work for session-based recommendation.

Inspired by the fact that similar users tend to buy similar items, the earliest session-based methods are mostly based on nearest neighbors \cite{3, 4, 19}. These methods need a similarity function to compute similarity scores. Then some approaches notice that exploiting sequential behavior is beneficial to predict the next item. Some of them employ Markov chain (MC) to capture the sequential signal of interactions. For example, Rendle et al. \cite{23} propose factorization of personalized Markov chains (FPMC) to capture both sequential user behavior and long-term interest. King et al. \cite{31} propose a hierarchical representation model (HRM) to improve FPMC with a hierarchical architecture. However, Markov chain-based methods are usually unable to capture more complex higher-order sequential patterns.

The rapid growth of deep learning methods brings significant performance gain on session-based recommendation. Recently proposed neural-based methods employ sequential structure neural networks like RNNs \cite{2, 9} to capture the user’s sequential behavior. For instance, Balázs et al. \cite{8} propose GRU4Rec to use the gated recurrent unit (GRU) \cite{2} to model user sequential behavior, Tan et al. \cite{26} apply data augmentation and generalized distillation to improve GRU4Rec and Li et al. \cite{12} propose to use attention mechanism to improve the representation capacity of RNNs. In addition, as sessions can be very limited as in-progress information, the neighbor information is introduced to assist in modeling ongoing sessions \cite{10, 17, 30}. For example, Jannach and Ludewig \cite{10} introduce a neighborhood session using the K-nearest neighbor (KNN) method. Unlike traditional KNNs, Wang et al. \cite{30} propose CSRM...
to incorporate neighbor sessions information via memory network [25, 36]. To learn better session representation, in NARM [12] an attention pooling method is proposed and is widely applied in subsequent neural recommendation methods [1, 14, 16, 29, 40, 42]. Specifically, these methods use global and local preference to make a recommendation. However, they still consider single items as the base recommendation units thus lack the description of in session combined information.

Graph neural network (GNN) is introduced to capture the complex transition relationships of sessions [1, 16, 29, 37, 40]. For instance, SR-GNN [37] takes transition relations between adjacent items to construct unweighted session graphs and uses gated graph neural network (GGNN) [13] to extract the transition information. Based on this work, Xu et al. [40] propose a method that uses a GGNN to extract local context information and a self-attention network (SAN) to capture global dependencies between distant positions. These GNN-based methods have shown a new and promising direction for session-based recommendation, while the constructed session graph faces a lossy session encoding problem. To alleviate such problem, Chen and Wong [1] propose lossless edge-order preserving aggregation and shortcut graph attention to efficiently aggregate information and make a recommendation. To propagate information more efficiently in long sessions, Pan et al. [16] propose to use a star node to bridge items. The newly added star node can filter out irrelevant items, thus make better recommendations on long sessions. However, all these works take single items as intent unit and ignore the consecutive combined user behavior of a group of items, thus losing the recommendation accuracy.

Some works employ Hypergraphs [5] to enhance item representation [29, 38]. Xia et al. [38] take each session as a hyperedge to model cross-session item relationship, and Wang et al. [29] propose SHARE that uses a sliding window to construct a hypergraph of a single session to capture group intent. However, such hypergraph structure loses the user sequential behavior information, i.e., the sequential order of items. In addition, the last attention pooling only uses the item representation to generate session representation which can not fully utilize the user intent information from all intent unit granularity levels.

3 METHODOLOGY

In this section, we first introduce the formal definition of the general session-based recommendation problem (Section 3.1). Then we describe the proposed multi-granularity consecutive intent unit (Section 3.1). Afterwards, we elaborate on three components of the proposed model i.e. the Multi-granularity Intent Heterogeneous Session Graph (Section 3.3), the Heterogeneous Graph Attention Network (Section 3.4), and the Intent Fusion Ranking (Section 3.5).

3.1 Problem Definition

Session-based recommendation is to predict the next-item based on a sequence of clicked items. Formally speaking, let \( I = \{i_1, i_2, \ldots, i_L\} \) represent all items in the dataset. A session \( s = \{i_1, i_2, \ldots, i_L\} \) is a collection of clicked items where \( L \) is the session length and \( i_j \) is the id of the item the user clicked at position \( j \). Given \( s \), the session-based recommendation is to recommend the next most likely clicked item \( i_{L+1} \). In practice, the recommender takes \( s \) as input and predict the probability distributions of the next item \( p(i_{L+1}|s) \). Then the items with the top-K largest probability scores are recommended.

3.2 Consecutive Intent Unit

Current session-based recommendation methods consider each item separately and the higher level intent of a local session snippet are ignored. In this work, we not only learn the intent revealed by the individual items but also the combined intent from items in a consecutive fragment. We define \( v_j^k \) as a consecutive intent unit in which the items are from a consecutive fragment started from \( j \) and with length \( k \), i.e., \( v_j^k = (i_j, \ldots, i_{j+k-1}) \). The length \( k \) also represents the granularity level of the intent unit. An example is illustrated in Figure 2 where a session \( s = \{i_1, i_2, \ldots, i_4\} \) is given. The items from the original sequence denoted by \( v_1^1, \ldots, v_4^1 \) compose the level-1 consecutive intent units. The session snippets \( (i_1, i_2), (i_2, i_3), (i_3, i_4) \) are level-2 consecutive intent units denoted by \( v_2^1, v_2^2, v_2^3 \) and \( (i_1, i_2, i_3), (i_2, i_3, i_4) \) are the level-3 consecutive intent units denoted by \( v_3^1, \ldots, v_3^5 \).
Given session \( s \), we assign a learnable embedding vector \( e_j \) to represent the intent of the level-1 consecutive intent unit revealed by item \( v_j \). For the high order level-\( k \) consecutive intent units, we utilize a readout function \( \mathcal{R} \) to fuse intents of each item in the corresponding intent unit to generate the representation. Specifically, the intent of a level-\( k \) consecutive intent unit can be calculated by

\[
e^k_j = \mathcal{R}((e^1_j, ..., e^k_{j+k-1}))
\]

where \( e^k_j \) is the intent of the level-\( k \) consecutive intent unit.

We consider two kinds of readout functions i.e. set-based and sequence-based to generate representation for high level intent unit. The set-based readout functions e.g. MEAN, MAX etc. can extract order-invariant intent while the sequence-based readout functions e.g. Gate Recurrent Unit (GRU) can extract order-sensitive intent.

In this graph, each node type represents an intent granularity level. An example of MIHSG is given as illustrated in Figure 2.

In this section, we introduce how to construct the level-\( k \) consecutive intent unit. For \( j \)-th level-\( k \) consecutive intent unit the final intent can be computed by

\[
e^k_j = e^{k,\text{set}}_j + e^{k,\text{seq}}_j
\]

where \( e^{k,\text{set}}_j \) and \( e^{k,\text{seq}}_j \) are corresponding to the order-invariant and order-sensitive intent respectively.

### 3.3 Multi-granularity Intent Heterogeneous Session Graph Construction

In this section we introduce the construction of the **Multi-granularity intent unit Heterogeneous Session Graph** (MIHSG). The proposed MIHSG is composed by multiple subgraphs and each subgraph models the transitions between the intents of same level consecutive intent units. The graph constructed by the level-\( k \) consecutive intent units is defined as the level-\( k \) intent session graph. We first describe how to construct the level-\( k \) intent session graph and then introduce the proposed MIHSG.

**3.3.1 Level-\( k \) Intent Session Graph.** The level-\( k \) session graph captures the spatial continuity of the user-item interactions. It is a directed graph \( G^k = (V^k, \mathcal{E}^k) \) where each node represents the intent of a level-\( k \) consecutive intent unit and each edge connects two consecutive adjacent level-\( k \) consecutive intent units. Nodes have connection if they are adjacent in the intent session sequence. The level-1 session graph captures the fine-grained intent transition between items. As the length of intent unit increases, the session graph contains higher level transition patterns between intent units.

**3.3.2 Multi-granularity intent unit Heterogeneous Session Graph.** We merge session graphs from different granularity levels to form a unified heterogeneous session graph i.e. MIHSG. Specifically, we introduce a special edge i.e. inter-granularity edge to link the high level session graphs and the level-1 consecutive session graph. It connects intent unit with its level-1 predecessor and successor. The order of a MIHSG \( k \) is the same as the largest granularity level. An example of MIHSG is given as illustrated in Figure 2. In this graph, each node type represents an intent granularity level and two kinds of edges are considered i.e. intra-granularity edge and inter-granularity edge. The intra-granularity edge denoted by \( (v^k, \text{intra-} k, v^k) \) connects intents of the same granularity level as described in section 3.3.1 while the inter-granularity edge i.e. \( (v^1, \text{inter}, v^k) \) and \( (v^k, \text{inter}, v^1) \) models the intent transition between high level consecutive intent unit and their adjacent individual items. For example, in session \( s = \{v_1, v_2, v_1, v_3\} \), two inter-granularity edges can be constructed. They are \((v_1, \text{inter}, (v_2, v_1))\) and \((v_2, v_1), (v_2, v_1)\). This enables capturing cross granularity level intent transition patterns. We do not consider inter-granularity edge between high granularity levels to avoid redundancy.

### 3.4 Session Representation Learning

In this section, we introduce how to generate the representations for consecutive intent units and the whole session.

**3.4.1 Learning Representation of Consecutive intent units.** We employ the heterogeneous graph attention network (HGAT) to learn the representation of each consecutive intent unit. Given an directed edge \((s, e, t)\) where \( s \) and \( t \) are the source and target intent units and \( e \) is the edge. Without loss of generality, \( s \) and \( t \) can be in any granularity levels. Specially, we denote \( k_s \) and \( k_t \) as the granularity levels of \( s \) and \( t \) respectively and denote as the edge type. For a level-\( k \) MIHSG, \( k_s, k_t \in \{1, ..., K\} \) and \( \phi_e \in \{\text{inter}, \text{intra-}1, ..., \text{intra-}K\} \). For each layer, we apply the bidirectional attention to aggregate the representations of the direct in-neighbors and out-neighbors. Given an in-neighbor set i.e. \( N_{\phi_e} \), the aggregation mechanism to aggregate the in-neighbors is as following:

\[
\begin{align*}
\hat{h}^{(l+1)}_i = & \sum_{\phi_e \in \{\text{inter}, \text{intra-}1, ..., \text{intra-}K\}} \sum_{s \in N_{\phi_e}(i)} a_s^{(l)} W^{(l)} \phi_e \hat{h}^{(l)}_s \\
\hat{a}_s^{(l)} = & \text{Softmax} \left( \frac{\hat{a}_s^{(l)}}{\sigma} \right) \\
\hat{a}_s^{(l)} = & \sigma \left( \sum_{\phi_e \in \{\text{inter}, \text{intra-}1, ..., \text{intra-}K\}} W^{(l)} \phi_e \hat{h}^{(l)}_s \right)
\end{align*}
\]

Where \( W^{(l)} \in \mathbb{R}^{d \times d} \) and \( a^{(l)} \in \mathbb{R}^{2d \times 1} \) is the learnable projection weights which are not shared between different layers and edge types. \( \hat{h}_i^{(0)} = e_i, \hat{h}_t^{(0)} = e_t \) is the initial representation of intent units \( s \) and \( t \). Following [28], we also adopt LeakyReLU activation function \( \sigma() \) to increase the non-linearity.

We also adopt multi-way attention to stabilize the learning process. After getting the representation of each head, a readout function is applied to generate the output node representation.

\[
\hat{h}_i^{(l+1)} = \mathcal{R} \left( \bigcup_{i=1 \ldots H} \hat{h}_i^{(l+1)} \right)
\]

where \( H \) is the number of attention heads and \( i \) is head index. We empirically find element-wise max operation is consistently better than concatenation and mean operations.

As MIHSG is a directed graph, each node has context from both in-neighbors and out-neighbors. To extract context information from two directions, we apply the HGAT to aggregate the in-neighbors and out-neighbors information to generate node representations. Thus for each intent unit \( v \), we get two embeddings \( \hat{h}^{(l+1)}_v \) and \( \hat{h}^{(l+1)}_v \) for two directions respectively. Then the local representation of node \( v \) is the summation of node embeddings from two directions, \( \hat{v}^{(l+1)}_v = \hat{h}^{(l+1)}_v + \hat{h}^{(l+1)}_v \). The final representation of node \( v \) is the summation of the local representation and the mean of all embeddings in the session \( \tilde{h}_v^{(l+1)} = \hat{v}^{(l+1)}_v + \tilde{h}_{u \in s} \).

**3.4.2 Learning Representation of Whole Session.** Current methods generate the user preferences by combining the representations of
the last clicked individual item and the whole session. Since the high order granularity intents are ignored, the user preferences captured by these methods may be inaccurate or incomplete. To better utilize information of all granularity intents, we generate a separate session embedding for each level of consecutive intent units. As illustrated in Figure 2, given a session $s_i$ and the corresponding consecutive intent unit embeddings $h^k_i \in \mathbb{R}^{d_i}, k = 1, ..., K$, where $n_k$ is the number of intent units for level-$k$ and $K$ is the number of intent levels. For each level of consecutive intent units, we generate a local representation $z^k_i$ and a global representation $z^k_g$ to capture the user preferences. We take the last intent unit $h^k_n$, as $z^k_i$, and adopt a soft attention mechanism to generate $z^k_g$. To make session representation from each level captures complete user intent we compose embeddings of all intent units to generate the session embedding. Specifically, given a candidate item set $s_i$ and the corresponding context embedding, Then, we use a discriminator to re-weight the item scores to balance the focus between repeat click and exploring click. Then the scores $y$ is the combination of the two parts, by:

$$y^k = [\beta^k_1 y^k_1; \beta^k_2 y^k_2]$$

where $[;]$ is the concatenation operation. $y^k_1 \in \mathbb{R}^{|R|}$ and $y^k_2 \in \mathbb{R}^{|O|}$ are probability distributions for inner session items and outer session items respectively and we have $|R| + |O| = |I|$, $\tilde{y}^k_1 \in \mathbb{R}$ and $\tilde{y}^k_2 \in \mathbb{R}$ are the corresponding normalized probabilities. By separate normalization, we distinguish the repeat click behavior and exploration behavior.

3.5 Intent Fusion Ranking and Optimization

After generating session embeddings from different granularity levels i.e. $z^k_1$, ..., $z^k_2$, ..., $z^k_K$, we propose to utilize Intent Fusion Ranking mechanism to capture comprehensive user preferences. Specifically, we first make a separate recommendation based on each level of intents and then fuse the results to make the final recommendation.

For each intent level $k$, we adopt the inner product to calculate the similarity between candidate item intent and the level-$k$ session embedding. Specifically, given a candidate item set $l = \{v_1, v_2, ..., v_{|l|}\}$, we can calculate the similarity between the session embedding $z^k_g$ and $e^j_i$ which is the intent of item $v_i$ as following:

$$y^k_i = \langle z^k_g, e^j_i \rangle$$

where $\langle , \rangle$ is the inner product operate.

Inspired by [22, 34], we also consider the repeat click behavior and exploration behavior. Specifically, we distinguish the inner session items from outer session items and apply the Softmax normalization separately as:

$$\tilde{y}^k_i = \frac{\exp \left( y^k_i \right)}{\sum_{j=1}^{n_k} \exp \left( y^k_j \right)}$$

where $R = \{r_1, ..., r_i, ..., r_{|R|}\}$ and $O = \{o_1, ..., o_i, ..., o_{|O|}\}$ represent the inner session items and outer session items respectively and we have $|R| + |O| = |I|$, $\tilde{y}^k_1 \in \mathbb{R}$ and $\tilde{y}^k_2 \in \mathbb{R}$ are the corresponding normalized probabilities by separate normalization, we distinguish the repeat click behavior and exploration behavior.

4 EXPERIMENTS

In this section, we first describe the experimental settings including the datasets, baselines, and evaluation metrics. Then we do detailed analyses for the experimental results.
4.1 Datasets
We conduct the experiments on four real-world datasets which are commonly used in the literature of session-based recommendation.

- **Gowalla** is a check-in dataset that is widely used for point-of-interest recommendation. Following [1, 6, 22], we kept the top 30,000 most popular locations, and grouped users’ check-in records into disjoint sessions by splitting intervals between adjacent records that are longer than 1 day. The last 20% of the sessions were used as the test set.
- **Last.Fm** is widely used in many recommendation tasks. The music artist recommendation is used as the task. Following [1, 6, 22], we kept the top 40,000 most popular artists and set the splitting interval to 8 hours. Similar to Gowalla, the most recent 20% of the sessions were used as the test set.
- **Yoochoose** is a dataset that is obtained from the RecSys Challenge 2015, which contains a stream of user clicks on an e-commerce website within 6 months. We use the typical method in [1, 12, 14, 22, 37, 42] to split the dataset and use the 1/64 and 1/4 subsample of all training sessions as the training set.
- **Diginetica** is a personalized e-commerce search challenge dataset provided in CIKM Cup 2016. The dataset contains transition history which is suitable for session-based recommendation. Following [1, 6, 12, 27, 37, 42], we use the sessions in the last week for test.

Following [1, 12, 14, 18, 20, 22, 37, 42], we filter out sessions of length 1 and items appearing less than 5 times. We adopt the data augmentation method described in in [12, 14, 22] to process the dataset. The statistics of these datasets are described in Table 1.

| Dataset  | #clicks | #train sessions | #test sessions | #items | length ≥ 5 | length > 5 | Average length |
|----------|---------|-----------------|----------------|--------|------------|------------|---------------|
| Gowalla  | 716,835 | 675,561         | 2,837,644      | 55,898 | 239,483    | 203,793    | 239,483       |
| Last.FM  | 1,122,788 | 3,835,706     | 369,859        | 55,898 | 3,733,254  | 136,267    | 1,738,734     |
| yoochoose | 425,956 | 29,310          | 38,615         | 16,766 | 29,618     | 29,618     | 29,618        |
| yoochoose1/64 | 557,248 | 1,336,909     | 55,898         | 16,766 | 289,490    | 289,490    | 289,490       |
| yoochoose1/4 | 8,326,407 | 51,267       | 55,898         | 16,766 | 4,234,915  | 4,234,915  | 4,234,915     |

4.2 Baselines
We consider three kinds of methods as our baselines: the conventional Nearest Neighbors (NN) based methods, Neural-based sequence methods and GNN based methods. The description of the baselines are as follows:

- **Item-KNN** [3] is a NN based method which recommends items that are similar to the previous items in current session and cosine similarity is adopted to measure the similarity between two items.
- **GRU4Rec** [8] employs gated recurrent unit [2] to model the sequential behaviour of items in a session.
- **NARM** [12] adopts GRU to extract sequence information and employs attention mechanism to capture user preferences.
- **SR-GNN** [37] uses gated graph neural networks to capture the transition relations between items.
- **GC-SAN** [40] applies a self-attention layer after the graph neural network module to integrate contextual information.
- **NISER+** [7] utilizes dropout and $l_2$ norm to alleviate over-fitting and long-tail effect.
- **SGNN-HN** [16] uses highway network to reduce over-smooth and long-range dependency problems.
- **LESTR** [1] introduces two kind of session graphs to solve the information loss and long-range dependency problem.
- **SHARE** [29] proposes utilizing hypergraph and attention network to exploit the contextual windows to model session-wise item representations.

4.3 Experimental Setup
For all methods, we employ grid search to find the best hyper-parameters. We random split 10% samples from the training set as the validation set and the hyper-parameters which achieve best performance on the validation set are selected. Specifically, for GNN-based models, the number of layers is searched in $\{1, 2, 3, 4, 5\}$. For the proposed MSGIFSR, we also search the best intent unit granularity level $K$ in $\{1, 2, 3, 4, 5, 6, 7\}$. We use Adam to optimize the model while set learning rate and weight decay to $4 \times 10^{-5}$ and $5 \times 10^{-4}$ respectively. We follow [7, 16, 37] to decay the learning rate every 3 epochs with a 0.1 rate. We fix the embedding dimension to 256 and the batch size to 512. For all models, we compare the HR@20 (Hit Rate) and MRR@20 (Average Reciprocal Ranking) metrics.

4.4 Performance Comparison
In this section, we compare our methods with the state-of-the-art baselines to validate the effectiveness. The empirical results of all baselines are shown in Table 2. From the results, we have the following important observations:

Firstly, neural network based methods are significantly better than conventional methods (e.g. Item-KNN), proving the neural network based models are capable of capturing complex sequential patterns for making recommendations. However, among them, the GRU4Rec achieves inferior performance and the reason is that in GRU4Rec only sequential information is leveraged. In contrast, NARM achieves competitive performance because it exploits both of the sequential information and global preferences. Moreover, we find that GNN-based models [1, 7, 16, 29, 37, 40] are better than previous methods. The GNN-based methods demonstrate the superiority of session graphs in representing transition relationships between different items. In current GNN based methods, SGNN-HN achieves better performance since it introduces a star node to help filter out irrelevant items and adopt high-way network to relieve the over-fitting problem.

Secondly, our method MSGIFSR outperforms all baselines by a large margin, indicating that the session-based recommendation can benefit from our proposed consecutive intent units. We attribute such significant gain to the following reasons: 1) the proposed MIHSG can exploit the intents from multiple granularity levels and model the complex transitions between different user intents; 2) the proposed IFR module can integrate the intents of different granularity levels to capture more accurate and complete user preferences. In addition, we find that MSGIFSR has larger gains on datasets which contain relative long sessions e.g. Last.FM.

### Table 1: Statistics of datasets

| Dataset  | #clicks | #train sessions | #test sessions | #items | length ≥ 5 | length > 5 | Average length |
|----------|---------|-----------------|----------------|--------|------------|------------|---------------|
| Gowalla  | 716,835 | 675,561         | 2,837,644      | 55,898 | 239,483    | 203,793    | 239,483       |
| Last.FM  | 1,122,788 | 3,835,706     | 369,859        | 55,898 | 3,733,254  | 136,267    | 1,738,734     |
| yoochoose | 425,956 | 29,310          | 38,615         | 16,766 | 29,618     | 29,618     | 29,618        |
| yoochoose1/64 | 557,248 | 1,336,909     | 55,898         | 16,766 | 289,490    | 289,490    | 289,490       |
| yoochoose1/4 | 8,326,407 | 51,267       | 55,898         | 16,766 | 4,234,915  | 4,234,915  | 4,234,915     |
The reason is that MSGIFSR captures the user preferences in a hierarchy way i.e. low level of intent unit to capture the fine grained intents while the high level intent unit to capture complex intents. Such a design can help propagate information between different intents more effectively and efficiently. We also notice that our proposed methods obtain relatively larger gains on Gowalla and Yoochoose1/64. From the statistics, we can learn that Gowalla and Yoochoose1/64 have relative small number of items and sessions. In such situation, our proposed high level consecutive intent units can help extract richer user intent, thus boosting the recommendation performance.

### 4.5 Ablation Study

In this section, we conduct ablation study to evaluate the contribution of each key components. Specifically, in each experiment we remove one of the components and the performance change indicates the importance of the removed component. The results are shown in Table 3.

#### 4.5.1 Effectiveness of Multi-granularity Intent Heterogeneous Session Graph.

To study the impact of MIHSG, we train models without out intra-granularity edges (intra-E), without inter-granularity edges (inter-E), and without MIHSG (without both types of edges) respectively and compare the performance. From Table 3, we find that either removing intra-granularity edges or inter-granularity edges harms HR@20 and MRR@20. It indicates that both kinds of transition relationships are important to capture the sequential information. By removing MIHSG, the transition information is completely lost and the context awareness is largely restricted since the sequence info has been removed. Meanwhile, by doing so, the propagation path for long dependency is cut off, which further aggravates the information loss. Besides, removing intra-granularity edges has a larger impact than removing inter-granularity edges.

This is because the transitions between the intent units within the same granularity serve as the foundation of the user preference description, and the inter-granularity edges can also benefit from it.

#### 4.5.2 Impact of Intent Fusion Ranking Module and RENorm.

The intent fusion ranking module composes recommendation results from all intent granularity levels. To study the effectiveness of IFR, we train a reduced model only taking the results from the level-1 intent granularity to make recommendations. As shown in Table 3, both of HR@20 and MRR@20 decays significantly, which means that information from higher intent granularity levels is complementary to the original level-1 intent granularity item sequence. We also train another reduced model without RENorm and find that worse MRR@20 are reported on all four datasets. It indicates that distinguishing repeat click behaviors and exploration behaviors is beneficial. By comparing the effectiveness of IFR module and RENorm module, it is observed that the IFR module brings larger gains on HR@20 while RENorm module impacts the MRR@20 more. An explanation is that the IFR module can help to recommend more correct items which favor HR and the RENorm aims to obtain strictly correct order, which leads to better MRR.

### 4.6 In-depth Analysis

#### 4.6.1 Performance on Different Session Lengths.

We study relation of the performance and session lengths. We firstly split the test set into long sessions and short sessions. Following [16, 37], sessions with length larger than 5 are long sessions while remain sessions are short ones. After that, we compare our method, i.e., MSGIFSR with intent unit granularity level 3 and Mean-GRU readout function with two state-of-the-art baselines NISER+ and SGNN-HN, and report the result on the four datasets. All GNN layers are set to 1 for fair comparisons. The results are shown in Figure 3.

From the figure, we find that NISER+ performs worse than SRGNN-HN. It is because SRGNN-HN applies star node to filter out irrelevant items. Among the three datasets, MSGIFSR achieves the best result on both long and short sessions, indicating the effectiveness of MSGIFSR. Moreover, we observe that the baseline methods suffer from the long-range dependency problem, especially for the long sessions, and the relative gains of MSGIFSR to...
baselines in long sessions are larger than in short ones. The reason is that the proposed high-level intent unit granularity can model more accurate user intent from a combined perspective, i.e., the overall meaning of a group of consecutive items. And, the information of the former intent units can be efficiently transported to the later ones through MIHSG. Thus, the long-dependency problem can be well handled.

We denote the model SHARE-IFR since it utilizes the proposed \( L \) granularity levels. It is because hypergraph only considers the group levels are among 1 to 7. The results are shown in Figure 4. We find that MSGIFSR with higher level intent granularity is consistently better than that with only single items. For Diginetica, in the beginning, the performance of MSGIFSR improves as higher level of intent granularity is incorporated into the MIHSG. Then the performance gain becomes stable as the level of granularity increases. We observe similar phenomenon in Yoochoose, Gowalla. For Last.FM, the performance of MRR@20 has a drop when intent granularity level increase to 4 and 5 and then becomes better for \( L = 6 \). It indicates that for the datasets with a long average session length, we need to incorporate higher-level granularity of intent to capture the user preference. We also find that as the granularity goes coarser, the performance becomes stable. That is because sessions are usually not so long and keep increasing the granularity level brings no more useful information.

4.6.3 Comparisons with Hypergraph. We replace MIHSG with other topological structures like hypergraph [29] to test the effectiveness of MIHSG. We then apply a hypergraph attention neural network to perform message passing operations on such hypergraph while keeping the remaining settings the same as our model. We denote the model SHARE-IFR since it utilizes the proposed IFR module for fairness. The representations of hyperedges are considered as high-level granularity intent. The intent-granularity levels are among 1 to 7. The results are shown in Figure 4. We find that MIHSG performs better than hypergraph across different granularity levels. It is because hypergraph only considers the group intent information of items and ignores the sequential behavior of them, which may cause information loss problem. In contrast, MIHSG can capture such information by using a recurrent neural network like GRU. Moreover, we also find that SHARE-IFR is not stable as the intent granularity level increases. An explanation is that the hypergraph attention neural network conducts message passing by two steps, i.e., from nodes to hyperedges and then from hyperedges to nodes. This makes information transported between two hyperedges inefficiently and destabilizes the performance when multiple hyperedges are involved. While heterogeneous GAT makes message passing for nodes with different intent granularity levels directly, hence avoid such phenomenon.

Figure 3: Comparison on different lengths of sessions.

(a) Results on HR@20
(b) Results on MRR@20

Figure 4: Impact of intent unit Granularity Levels

5 CONCLUSION

In this paper, we study the session-based recommendation problem and proposed MSGIFSR, a novel model that extracting different in-session intent granularity information. It is observed that varied consecutive combined intent granularity provides richer user preference, and our proposed MIHSG successfully captures complex preference transition relationships among multi-level consecutive intent units and long-range dependencies by modeling intra- and inter-granularity intent unit edges. In addition, the intent unit encoder mechanism considering both the order-variant and order-invariant relations of a group of items which are good at representing intent unit meanings. Last but not least, the ablation study demonstrates the Intent Fusion Ranking module successfully incorporates recommendation results from all intent unit levels. We will extend our work in a streaming session-based setting and reduce the time complexity and popularity bias for recommendations in the future work.

Acknowledgments. Jiayan Guo and Yan Zhang are supported by National Key Research and Development Program of China under Grant No. 2018AAA0101902, and NSFC under Grant No. 61532001.
A APPENDICES

A.1 Hyper-parameter Study

We study the effects of different readout functions when extracting high-level granularity of intent from a group of items. Specifically, we compare order-invariant readout functions (MEAN, MAX), order-variant readout functions (GRU) and the combination of both operations (sum the outputs). The intent granularity length is set to 3 in all experiments. As shown in Figure 4, for Diginetica and Gowalla the readout function with both order-variant operation and order-invariant operation achieves the best results. However, for Yoochoose1/64, GRU achieves the best result, indicating that sequential behavior is critical for predicting the next item. As for Last.FM, MAX+GRU achieves the best in HR@20, since max pooling can filter out irrelevant items and GRU preserves sequential information thus help make recommendation on long sessions.

Table 4: Impact of Readout function

|                     | Diginetica | Gowalla | Last.FM | Yoochoose1/64 |
|---------------------|------------|---------|---------|---------------|
|                     | HR@20 | MRR@20 | HR@20 | MRR@20 | HR@20 | MRR@20 | HR@20 | MRR@20 |
| MEAN                | 56.87   | 19.91   | 55.89   | 28.66   | 26.95   | 10.46   | 72.73   | 33.35   |
| MAX                 | 56.82   | 19.84   | 55.88   | 28.57   | 26.99   | 10.55   | 72.60   | 33.38   |
| GRU                 | 56.81   | 19.94   | 56.03   | 28.75   | 26.95   | 10.60   | 72.85   | 33.48   |
| MEAN+GRU            | 56.90   | 20.03   | 56.00   | 28.74   | 27.01   | 10.65   | 72.80   | 33.42   |

We also study the performance of different HGAT layers, and the results are shown in Figure 5. From the figure, we find that deep models, e.g., depth > 3, suffer from over-fitting problems and achieve worse performance. Therefore, stacking more layers is not an effective method to capture long-range dependencies. However, we can alternatively extend the intent granularity level to relieve the burden of long-range dependencies.

Figure 5: Results on different number of HGAT layers.