GIMIRec: Global Interaction Information Aware Multi-Interest Framework for Sequential Recommendation

Jie Zhang  
1020041203@njupt.edu.cn  
Nanjing University of Posts and Telecommunications  
Nanjing, Jiangsu, China

Ke-Jia Chen  
chenkj@njupt.edu.cn  
Nanjing University of Posts and Telecommunications  
Nanjing, Jiangsu, China

Jingqiang Chen  
cjq@njupt.edu.cn  
Nanjing University of Posts and Telecommunications  
Nanjing, Jiangsu, China

KEYWORDS
sequential recommendation, multi-interest framework, global context extraction

ABSTRACT
Sequential recommendation based on multi-interest framework models the user’s recent interaction sequence into multiple different interest vectors, since a single low-dimensional vector cannot fully represent the diversity of user interests. However, most existing models only intercept users’ recent interaction behaviors as training data, discarding a large amount of historical interaction sequences. This may raise two issues. On the one hand, data reflecting multiple interests of users is missing; on the other hand, the co-occurrence between items in historical user-item interactions is not fully explored. To tackle the two issues, this paper proposes a novel sequential recommendation model called “Global Interaction Aware Multi-Interest Framework for Sequential Recommendation (GIMIRec)”. Specifically, a global context extraction module is firstly proposed without introducing any external information, which calculates a weighted co-occurrence matrix based on the constrained co-occurrence number of each item pair and their time interval from the historical interaction sequences of all users and then obtains the global context embedding of each item by using a simplified graph convolution. Secondly, the time interval of each item pair in the recent interaction sequence of each user is captured and combined with the global context item embedding to get the personalized item embedding. Finally, a self-attention based multi-interest framework is applied to learn the diverse interests of users for sequential recommendation. Extensive experiments on the three real-world datasets of Amazon-Books, Taobao-Buy and Amazon-Hybrid show that the performance of GIMIRec on the Recall, NDCG and Hit Rate indicators is significantly superior to that of the state-of-the-art methods. Moreover, the proposed global context extraction module can be easily transplanted to most sequential recommendation models.

CCS CONCEPTS
• Information systems → Recommender systems; Collaborative filtering.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Woodstock '18, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/10.1145/3122445.3122456

1 INTRODUCTION
The recommendation system predicts users’ preferences for items based on their historical behaviors. For example in e-commerce application, the users are often recommended some commodities based on their past purchase behaviors. Deep neural networks, especially graph neural networks (GNNs) [4, 18], have now been applied on the recommendation systems, such as GRU4Rec [6], GGNN [15] and SR-GNN [25], achieving better performance than traditional collaborative filtering methods [7, 16]. However, the graph-based methods are often with high complexity as the result of the large scale of users and items in real world [5]. The sequential recommendation has recently drawn more research attention as it starts with each user’s own interaction data, models the users’ preferences with only recent interaction behaviors and then gives personalized recommendation for each user.

The main effort of sequential recommendation is to learn the representations of users and items. Since the interests of users are usually diverse and difficult to be represented by a single low-dimensional vector, the sequential recommendation methods based on multi-interest framework have emerged [1, 2, 13]. In these methods, the number of representation vectors for each user is increased from 1 to $K$, where each vector represents one interest of the user. Multiple interests can be captured by an improved capsule network [13] or a self-attention mechanism [1] and ultimately a more personalized recommendation based on $K$ interests is provided.

Most of the existing multi-interest frameworks for sequential recommendation use only users’ recent interaction sequence as it is believed that the user’s next interaction item is more closely related to his/her recent behaviors. However, we believe that the complete historical interactions of all users (referred to as the global context in this paper) can be better leveraged to reflect the diversity of users’ interests and potential relations between items.

An example shown in Figure 1 is to illustrate how the global context affects the prediction of the 10th interactive item for each user although this item did not appear in each user’s recent interactions. Since the popcorn and wine interaction sequence can be found in
user A’s early historical sequence, the 10th interactive item for user B will be probably the wine. The cake could be recommended for user A instead of battery due to the co-occurrence of game console and cake in user C’s historical sequence. The example indicates that the user’s behaviors could be predicted with the reference of other users’ historical behaviors though different users may have different preferences.

![Figure 1: An example of users’ interaction sequences.](image)

To our knowledge, these have been some sequential recommendation models exploring users’ complete sequence to learn their preferences [17, 20, 27–29]. But they focus more on each user’s personalized behaviors but do not take the global context into consideration.

Therefore, this paper proposes a global interaction information aware multi-interest framework for sequential recommendation (named GIMIRec) with the main contributions as below:

- A new multi-interest sequential recommendation model is proposed, which is equipped with a global context extraction module to fully utilize the historical sequences of all users. This module can also provide a performance boost for the recommendation models based on representation learning.
- A lightweight approach is designed to calculate the global context embedding of each item, based on the weighted co-occurrence matrix fusing the constrained co-occurrence number of item pairs and their time intervals in the global context.
- Our model achieves the state-of-the-art performance on three real-world datasets without introducing any other supportive information (such as item attributes or user’s social information) other than interactions and timestamps.

2 RELATED WORK

Sequential recommendation. Traditional recommendation methods, such as matrix decomposition and collaborative filtering [7, 16], represent interaction behaviors as a user-item matrix and explore their potential relation. With the development of deep neural networks, different recommendation methods combined with deep learning have emerged, such as the neural collaborative filtering method NGCF [22]. However, it is hard for the traditional recommendation system to manage huge but sparse interaction data. To solve this problem, sequential recommendation methods are proposed. The objective of sequential recommendation is to learn each user’s preferences from the historical interaction behaviors, and then to achieve recommendation. Typical methods include GRU4Rec [6] and SASRec [10] where the recurrent neural network and the attention mechanism are used respectively. Later, the time of the interaction occurrence is also taken into account. For example, a parallel time-aware mask network is proposed in MTIN [8], which extracts a variety of time information through multiple networks to enhance the short video recommendation effect. Ti-SASRec [14] is another method incorporating time interval of every two items in sequences to further improve the performance of SASRec [10].

Multi-interest framework for sequential recommendation. User preferences may contain a variety of interests. The existing sequential recommendation systems eventually model users’ preferences as a single low-dimensional vector while the users may have more than one single interest. MIND [13] was the first to model each user with K vectors, where different vectors represent different interests. The extraction of multi-interests is similar to the clustering process and the behavior-to-interest (B2I) approach is proposed to obtain K interests of users through a capsule network. Subsequently, Cen et al. [1] proposed a multi-interest extraction method based on self-attention mechanism (ComiRec-SA) to instead of capsule network. The latest method PIMI [2] introduced the time and periodic information in users’ recent interaction sequences into the model. The self-attention mechanism was used to capture users’ multi-interests and achieved the state-of-art performance. However, all the above multi-interest based sequential recommendation models, including PIMI, do not fully utilize the historical interactions of all users, thus lacking the global context of items.

Graph neural networks. In recommendation systems, users’ historical interaction behaviors can be converted into a graph structure, where users and items are represented as nodes and their interactions are represented as edges. Graph neural networks (GNNs) [4, 18] have been proved to better learn the representations of users and items. In non-sequential recommendation models, such as NGCF [22], LightGCN [5] and DGCF [23], simplified GNNs are used on the bipartite graph composed of users and items to generate their embeddings. In sequential recommendation models such as SR-GNN [25], GNNs are also widely used to represent user’s sequence and to capture user’s preferences.

3 PROBLEM DEFINITION

Before describing the proposed model in detail, the formal definitions of sequential recommendation and multi-interest framework are given as follows:

**Definition 1: Sequential Recommendation.** Assuming that $\mathcal{U}$ represents a collection of all users and $\mathcal{I}$ represents a collection of all items. For a given user $u \in \mathcal{U}$, his/her interaction sequence $S_u$ is defined with an item sequence $I_u = \{i^u_1, i^u_2, \ldots, i^u_N\}$ and a timestamp sequence of each interaction occurrence $T_u = \{t^u_1, t^u_2, \ldots, t^u_N\}$. The task of sequential recommendation is to predict the next interactive item $i^u_{N+1}$ for the user $u$ based on $S_u$. 
Definition 2: Multi-Interest Framework. It denotes a framework designed for sequential recommendation. According to Definition 1, each user \( u \) will be represented as a \( d \)-dimensional embedding by learning from his/her interaction sequence. However, one single vector representation may not be sufficient to describe the user’s multi-interest preferences. In the multi-interest framework, \( K \) \( d \)-dimensional embeddings of each user will be learned for sequential recommendation.

4 METHOD
4.1 Overview
The main purpose of our model GIMIRec is to better capture the multiple interests of each user from the global context information, which is calculated from the historical sequences of all users. The overall model consists of the following modules.

Global Context Extraction Module. The module is basically to collect the co-occurrence number and time interval for any item pair that appear one after another in each user’s historical interaction sequence for the calculation of the weighted co-occurrence matrix. Subsequently, a simplified graph convolution is used on the above matrix to obtain the global context embedding of each item.

Recent Time Interval Representation Module. Though we believe the global context will better reflect the diversity of user preferences, the user’s recent interactions directly reflect user’s recent preferences. So this module first identifies the recent interaction sequence based on the predetermined length \( L_{\text{rec}} \), and then encodes the time interval between any item pair within the sequence to obtain the time interval embedding for recent items.

Aggregation Module. For each item in the user’s recent interaction sequence, the global context embedding and the time interval embedding are merged in this module to get the hybrid embedding. For each user, a graph neural network is used to learn the final personalized embedding by aggregating the above hybrid embeddings in his/her recent interaction sequence.

Multi-Interest Extraction Module. The module extracts \( K \) interest vectors from the personalized embedding of user’s recent sequence by the self-attention mechanism [21]. For model training, one interest vector that can be best represent user’s current preference is selected from \( K \) interest vectors.

Finally, the trained model is used to predict the user’s next interest vector that can best represent user’s current preference in the test set accoring to user’s \( K \) interests.

The architecture of GIMIRec is demonstrated in Figure 2, and each module will be detailed in the following subsections.

4.2 Global Context Extraction Module
The existing multi-interest frameworks for sequential recommendation extract the user’s interests only from \( L_{\text{rec}} \) items with which user recently interacted. Historical interactions earlier than \( L_{\text{rec}} \) items are discarded.

The Global Context Extraction (GCE) Module tries to learn the global context embedding of items from all historical user interactions. Figure 3 illustrates how the GCE module processes historical interaction sequences with timestamps to get the weighted co-occurrence matrix \( \mathcal{A}' \) by taking three users A, B and C as example.

On the right-hand side of the Figure 3, a simplified graph convolution is used on \( \mathcal{A}' \) to obtain the global context embedding of each item.

Here, the weighted co-occurrence matrix \( \mathcal{A}' \) is defined as a combination of \( k \)-hop (\( k = 1, 2, 3 \)) item co-occurrence matrices. For each \( S_u \), the \( k \)-hop item pair is defined as a directed \( <\mu^u_{i1}, \mu^u_{i2}, \ldots, \mu^u_{in}, \mu^u_{nk}> \) (\( n \in \mathcal{N} \)). For example, user B in Figure 2 has four interacted items, so the 1-hop item pairs are \( <i_2, i_5>, <i_5, i_1> \) and \( <i_1, i_4> \), the 2-hop item pairs are \( <i_2, i_1>, <i_5, i_4> \) and the 3-hop item pair is \( <i_2, i_4> \).

The detailed steps of the GCE module are as follows.

Firstly, the \( k \)-hop item pairs are filtered out from the historical interaction sequences \( L_u = \{\mu^u_{i1}, \mu^u_{i2}, \ldots, \mu^u_{in} \} \) and the corresponding timestamp sequence \( T_u = \{t^u_{i1}, t^u_{i2}, \ldots, t^u_{in} \} \) \( u \in \mathcal{U} \). In order to better capture the relation between items, a time interval threshold \( L_{\text{time}} \) is set. The two items in \( S_u \) with the time interval less than \( L_{\text{time}} \) can become a co-occurrence item pair. Given the large number of items in the sequences, we only consider 1-hop, 2-hop and 3-hop item pairs to reduce the amount of calculation. For convenience, triplet collections \( R_u(k = 1, 2, 3) \) (Eq. 1) are used to save the id of two items and their time interval for three types of hop. Since the user id is no longer functional here, the superscript \( u \) is replaced with \( j \), representing one co-occurrence of the item pair \( <i_{j1}, i_{j2}> \). For different \( j \), the time interval of \( <i_{j1}, i_{j2}> \) could be different.

Figure 2: The architecture of the proposed model GIMIRec.
where $\alpha$ is supposed to be the largest and $\gamma$ is supposed to be the smallest. The identity matrix $I$ is added into $\mathcal{A}'$ in order to facilitate the graph convolution.

Finally, inspired by GCN [12], LightGCN [5] and GES [30], a simple and lightweight graph convolution operation is used to obtain the global context embedding of each item (Eq. 6-7). The Laplace matrix $\mathcal{A}$ is first calculated by the obtained $\mathcal{A}'$ and its degree matrix $\mathcal{D}$ and then multiplied with the embedding lookup table of items $\mathcal{E}_{item}$ to get the global context embedding of items $\mathcal{E}_{global} \in \mathbb{R}^{I \times d}$.

$$\mathcal{A} = \mathcal{D}^{-\frac{1}{2}} \mathcal{A}' \mathcal{D}^{-\frac{1}{2}}$$ (6)

$$\mathcal{E}_{global} = \mathcal{A} \mathcal{E}_{item}$$ (7)

An example of the above calculations is illustrated in Figure 3. The GCE module only runs once in the training phase and can also be used as an independent module for any other sequential recommendation models.

### 4.3 Recent Time Interval Representation Module

After getting the global context embedding of items, our model further processes each user’s recent item sequence and timestamp sequence to learn the user’s personalized interaction preferences.

Firstly, the sequence length threshold $L_{rec}$ is set to intercept each user’s most recent interactions, which does not include the target item for training. If there are less than $L_{rec}$ items, the padding strategy [2] will be used. Eventually, each user $u$ has a fixed-length sequence $S_u = \{i_u^1, i_u^2, \ldots, i_u^{L_{rec}}\}$ and the corresponding timestamp sequence $T_u = \{t_u^1, t_u^2, \ldots, t_u^{L_{rec}}\}$.

Secondly, the time interval matrix is generated. This module resets $t_{u,k}^i$ as min($|t_u^i - t_{u,k}^i|$, $L_{time}$) by using the threshold $L_{time}$ (defined in GCE) to emphasize the user’s recent interests. The interaction time interval of every two items in the user’s recent interaction sequence $S_u$ is put into the time interval matrix $T_u \in \mathbb{R}^{L_{rec} \times L_{rec}}$, which is a symmetric matrix.

$$T_u = \begin{bmatrix}
i_u^1 & i_u^2 & \cdots & i_u^{L_{rec}} \\
i_u^2 & i_u^{L_{rec}+1} & \cdots & i_u^{2L_{rec}} \\
\vdots & \vdots & \ddots & \vdots \\
i_u^{L_{rec}} & i_u^{2L_{rec}} & \cdots & i_u^{3L_{rec}}
\end{bmatrix}$$ (8)

Thirdly, the time interval matrix $T_u$ is embedded to a tensor $T_u' \in \mathbb{R}^{L_{rec} \times L_{rec} \times d}$, where $d$ is the dimension of embedding. The time interval aware attention mechanism [2] is used to get attention scores $S_u' \in \mathbb{R}^{L_{rec} \times L_{rec}}$ for each time interval in the sequence, denoted as

$$S_u' = \text{softmax}(T_u' W_1)^T, \quad (9)$$

where $W_1$ is a trainable parameter matrix and $T_1$ represents the transposition of the matrix.

Finally, the time interval embedding of items $E_{time}'$ in the recent sequence for each user $u$ is calculated through the broadcast mechanism in TensorFlow, which combines the attention score $S_u'$ and the time interval embedding tensor $T_u'$.

$$E_{time}' = S_u' T_u' \in \mathbb{R}^{L_{rec} \times d} \quad (10)$$

Note that the item pairs whose time interval exceeds the given threshold will be discarded.

Figure 3: An example of Global Context Extraction Module.
4.4 Aggregation Module

This module aggregates the global context embeddings and the time interval embeddings of the user’s recent interactive items to get user’s personalized embedding.

Firstly, the items in each user’s recent interaction sequence are picked out from the matrix \( \mathbb{E}_{\text{global}} \) to form a global context embedding matrix \( \mathbb{E}_{\text{global}}^u \) for the user \( u \):

\[
\mathbb{E}_{\text{global}}^u = [e_1^u, e_2^u, \ldots, e_L^u] \in \mathbb{R}^{L \times d}
\]

Secondly, the time interval embeddings and the global context embeddings of each user’s recent interactive items are combined by addition to get the hybrid embeddings:

\[
\mathbb{E}_{\text{hyb}}^u = \mathbb{E}_{\text{time}}^u + \mathbb{E}_{\text{global}}^u
\]

Thirdly, a GNN [21] is used to further extract the user’s personalized representation from the recent sequence. To construct a graph, the adjacent items in the sequence are first connected and then a virtual central node \( e_c^u \in \mathbb{R}^{kd} \) is created and linked with all item nodes to capture the relation between items under less than 1-hop. The embedding of \( e_c^u \) is set to the average value of all item’s hybrid embeddings:

\[
e_c^{(0)} = \frac{1}{L_{\text{rec}}} \sum_{i=1}^{L_{\text{rec}}} \mathbb{E}_{\text{hyb}}(i)
\]

The multi-head attention mechanism is then used to aggregate the item nodes \( \mathbb{E}_{\text{hyb}} \) and the central node \( e_c^u \) in \( L_{\text{layer}} \) steps. The layer-0 aggregation is the information initialization,

\[
q_i^{(0)} = \mathbb{E}_{\text{hyb}}(i),
\]

\[
k_i^{(0)} = \text{concat}[q_i^{(0)}; e_c^{(0)}; q_i^{(0)}; e_i^{(0)}],
\]

where \( i \) represents the \( i \)-th item, and the superscript (0) represents layer-0. The neighbor node \( q_{i-1} \) and the center node \( e_c^u \) and the global context representation \( e_i^u \) are then sent into the multi-head attention mechanism at layer-\( l \) (\( l = 1, \ldots, L_{\text{layer}} \)).

\[
k_i^{(l)} = \text{concat}[q_{i-1}^{(l-1)}; e_c^{(l-1)}; q_i^{(l-1)}; e_i^{(l-1)}],
\]

\[
q_i^{(l)} = \text{MultiHeadAtt}(K = k_i^{(l)}, Q = q_i^{(l-1)}, V = k_i^{(l)}).
\]

Similarly, the central node \( e_c^{(l)} \) is aggregated as:

\[
k_l^{(l)} = \text{concat}[e_c^{(l-1)}; q_1^{(l)}; q_2^{(l)}; \ldots; q_{L_{\text{rec}}}^{(l)}],
\]

\[
e_c^{(l)} = \text{MultiHeadAtt}(K = k_l^{(l)}, Q = e_c^{(l-1)}, V = k_l^{(l)}).
\]

After the total of \( L_{\text{layer}} \) iterations, the final personalized embeddings of the items are stacked in sequential order into the matrix \( \mathbb{E}^u \), which will be used in the next module.

4.5 Multi-Interest Extraction Module

This module extracts the \( K \) interests from the final representation of recent sequence.

Research has shown that the self-attention mechanism achieves better performance than capsule network in extracting user’s multi-interests [1]. Therefore, our model also uses the self-attention mechanism with two trainable parameter matrices \( W_2 \in \mathbb{R}^{4d \times d} \) and \( W_3 \in \mathbb{R}^{K \times d} \) for user’s \( K \) interest extraction:

\[
S_2^u = \text{softmax}(W_3 \tanh(W_2 \mathbb{E}^u))
\]

\[
\mathbb{E}_{\text{interest}}^u = S_2^u \mathbb{E}^u \in \mathbb{R}^{K \times d}
\]

4.6 Model Training

To train the whole model, the optimal interest \( o^u \) of the obtained \( K \) interests of each user is selected based on the target item (i.e., the next interactive ground-truth item) (Eq. 22).

\[
o^u = \mathbb{E}_{\text{interest}}^u [\arg \max (\mathbb{E}_{\text{interest}}^u o_{\text{target}}^u)]
\]

where the \( \arg \max \) operation in TensorFlow is used on the multiplication of \( \mathbb{E}_{\text{interest}}^u \) and the embedding of the target item \( o_{\text{target}}^u \in \mathbb{E}_{\text{global}}^u \).

To be more efficient, the sampled softmax method [19] is used to train the model, with the goal of minimizing the following loss \( L(\theta) \):

\[
L(\theta) = \sum_{u \in \mathcal{U}} -\log \frac{\exp(o_{\text{target}}^u e_u^u)}{\sum_{\hat{v} \in \text{Sampled}(u)} \exp(o_{\text{target}}^u e_u^u)}
\]

The optimizer used in GIMIRec is Adam [11].

4.7 Recommendation Results

After the training, GIMIRec achieves Top-\( N \) recommendation for all users in the test set. Due to the large size of the item pool, a GPU-accelerated method Faiss [9] is used to search for \( N \) items that are most relevant to user interests. The recommendation results are given based on the \( \mathbb{E}_{\text{interest}}^u \) obtained from Eq. 21.

\[
\text{RecSet} = \text{GetTopN}(\max_{1 \leq k \leq K} (\mathbb{E}_{\text{interest}}^u o_{\text{target}}^u(k)))
\]

where \( e_u^u \) is the embedding of the predicted item, \( \mathbb{E}_{\text{interest}}^u(k) \) is the \( k \)-th interest embedding of the user \( u \).

5 EXPERIMENTS

This section introduces the experimental setup, evaluation methods and the performance comparison of the proposed GIMIRec model against six baselines, including the state-of-the-art multi-interest recommendation model. Moreover, the ablation study is carried out to verify the effectiveness of each innovation in GIMIRec.

5.1 Experiment Settings

Datasets. We conducted experiments on three real-world datasets: Amazon-Books 1, Amazon-Hybrid 1 and Taobao-Buy 2. The Amazon dataset is divided into a series of sub-datasets according to

1http://jmcauley.ucsd.edu/data/amazon/
2https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
the product category, among which Amazon-Books is the largest sub-dataset. Considering users are not only interested in books, we mixed five smaller datasets ("Electronics", "Clothing, Shoes and Jewelry", "CDs and Vinyl", "Tools and Home Improvement" and "Grocery and Gourmet Food") to form a new dataset Amazon-Hybrid. Taobao-Buy is the sub-dataset of the Taobao dataset filtering out all purchases. We discarded the users and items with less than 5 interactions and illegal timestamps. The length of the recent interaction sequence \( l_{rec} \) is set to 20 for Amazon-Books and Amazon-Hybrid, and 50 for Taobao-Buy. The statistics of the datasets is listed in Table 1.

| Dataset         | users  | items  | interactions |
|-----------------|--------|--------|--------------|
| Amazon-Books    | 603,667| 367,982| 8,898,041    |
| Amazon-Hybrid   | 306,002| 169,385| 3,351,187    |
| Taobao-Buy      | 47,140 | 82,579 | 324,003      |

We maintained the dataset division in the model validation of ComiRec [1] and PIMI [2]. In detail, we divided each dataset into a training set, a validation set, and a test set at a ratio of 8:1:1. To evaluate the model, we predicted the most recent 20% of user behaviors in the validation and test sets and used the remaining 80% to infer user multi-interest embeddings.

**Baselines.** Comparison methods include: (1) GRU4Rec [6] (2015), the first sequential recommendation method using recurrent neural networks; (2) Youtube DNN [3] (2016), a deep neural network sequential recommendation model; (3) MIND [13] (2019), the first sequential recommendation model proposing the multi-interest framework; (4) ComiRec-DR and (5) ComiRec-SA [1] (2020) use dynamic routing mechanism and self-attention mechanism respectively to extract user’s multiple interests; (6) PIMI [2] (2021), the up-to-date sequential recommendation model based on multi-interest framework. It is noted that in above models based on multi-interest framework, MIND and ComiRec introduce extra information such as user age, item category etc., which is not adopted in PIMI and our model GIMIRec.

**Evaluation Metrics.** We used three common Top-N recommendation evaluation metrics, Hit Rate @\( N \), Recall @\( N \) and NDCG @\( N \), to compare the performance of models. Hit Rate @\( N \) represents the percentage that recommendation results contain at least one ground truth item in top \( N \) position. Recall @\( N \) indicates the proportion of the ground truth items in the recommendation results. NDCG (Normalized Discounted Cumulative Gain) @\( N \) measures the ranking quality in the \( N \) items by assigning high scores to high rankings [2]. In our experiments, \( N \) is set to 20 and 50 respectively.

**Implementation Details.** GIMIRec is implemented with Python 3.8 in TensorFlow 2.4, and a CUDA compatible GPU is used on the Ubuntu 20.04 operating system. The embedding dimension \( d \) is set to 64 for all datasets. The batch size on Amazon-Book, Amazon-Hybrid and Taobao-Buy are set to 128, 128 and 256, respectively, with a dropout rate 0.1 and learning rate 0.001. In sampled softmax loss calculation, the number of samples is set to 10. Other hyperparameter settings are shown in Table 2. Finally, we iterated the training up to 1 million rounds as the previous models did.

### 5.2 Comparisons and Analysis

**Recommendation performance.** The recommendation results of all methods on three datasets are shown in Tables 3 and 4, corresponding to the value of \( N=20 \) and \( N=50 \). To be fair, all models adopt the same parameter settings. The result shows that the GIMIRec model achieves the best performance on all evaluation indicators for all datasets. The performance of GIMIRec is significantly better than MIND and ComiRec even though there is no extra information introduced. Compared to the most advanced PIMI method, GIMIRec has great advantages, especially for Amazon data-sets. For Amazon-Hybrid, the performance improvement is the highest. The possible reason is that the multiple interests of users can be better represented from a rich variety of items. For Taobao-Buy, the performance improvement is not as good as that for Amazon datasets due to the shorter time span of dataset and sparser user interactions.

The result fully verifies the effectiveness of integrating the global context information.

**Ablation study.** We conduct a series of ablation studies to observe the impact of parameters settings in the GCE module on the results. GIMIRec is compared with three GIMIRec variants, GIMIRec_−INT (no Interval, Number and Threshold), GIMIRec_−IN (no Interval and Number) and GIMIRec_−I (no Interval). Among them, GIMIRec_−I changes GIMIRec by setting \( a=0 \) and \( b=1 \) in Eq. 2, indicating that the interaction interval information is not used when calculating the global context. GIMIRec_−IN further changes GIMIRec_−I in Eq. 3 by setting \( q^k_{ij} = 1 \), indicating that neither time intervals nor the co-occurrence number of item pairs are used. The difference of GIMIRec_−INT and GIMIRec_−IN is that the former does not set the time interval threshold \( l_{time} \). The experimental results in Table 5 show that the integration of the number of interactions and time intervals in the global context extraction module in GIMIRec is effective and can further improve the performance. The use of the time interval threshold \( l_{time} \) is valid to filter nearby item pairs.

**Impact of the number of interests \( K \).** Different from the general sequential recommendation, the multi-interest framework learns \( K \) vectors for each user. Therefore, we discussed the impact of \( K \) on the performance. The experimental results are listed in Table 6. It shows that the recommendation result of modeling multiple vectors for user preferences is better than that of modeling only a single vector. Moreover, the optimal \( K \) value of the model relies on the dataset. The recommendation performance is the best when \( K = 4 \) for two Amazon datasets. However, for Taobao-Buy, most of the recommendation results are the best when \( K = 8 \). The possible reason is that Taobao-Buy has more product categories, which could reflect more interests of users.

### Table 2: Hyperparameter Settings

| Dataset          | \( a \) | \( b \) | \( \alpha \) | \( \beta \) | \( \gamma \) | \( K \) | \( l_{time} \) |
|------------------|--------|--------|-------------|------------|-----------|------|-------------|
| Amazon-Books     | 0.65   | 0.35   | 4.5         | 2          | 1         | 4    | 64          |
| Amazon-Hybrid    | 0.5    | 0.5    | 5           | 2.5        | 1         | 4    | 64          |
| Taobao-Buy       | 0.6    | 0.4    | 5           | 3          | 1         | 8    | 7           |
with all three matrices is better than that of only 1-hop matrix. The weight of 1-hop, 2-hop and 3-hop respectively. Table 7 lists the recommendation result is optimal when the proportion of $\alpha$, $\beta$ and $\gamma$, to set the co-occurrence weight of 1-hop, 2-hop and 3-hop respectively. Table 7 lists the recommendation results of GIMIRec on Amazon-Hybrid under different combinations of three parameters. It shows that the result with all three matrices is better than that of only 1-hop matrix. The recommendation result is optimal when the proportion of $\alpha$, $\beta$ and $\gamma$ is 5:2.5:1. The result is consistent with expectation, that is, the 1-hop relation is the most important.

**Impact of hyperparameters $a$ and $b$.** The $a$ and $b$ in Eq. 2 determine how the GCE module uses the time interval of historical interactions. The value of $a$ (the weight of time interval) has been proved to be effective in above ablation experiments. In Table 8, we experimented different combinations of $a$ and $b$ in Amazon-Hybrid.
Table 7: Impact of hyperparameters $\alpha$, $\beta$ and $\gamma$. The best in each column is bolded and the second best is underlined.

| $\alpha$ | $\beta$ | $\gamma$ | Amazon-Hybrid @20 | Amazon-Hybrid @50 | Recall | NDCG | Hit Rate | Recall | NDCG | Hit Rate |
|---|---|---|---|---|---|---|---|---|---|---|
| 1:0:0 | 1:1:0 | 3:2:1 | 4:2:1 | 4:5:2:1 | 5:2:1:1 | 5:3:1 | 6:2:1 | 7:0:1 | 8:0:1 |
| 6.248 | 6.916 | 6.872 | 6.550 | 6.986 | 7.033 | 6.729 | 6.986 | 7.033 | 7.033 |
| 0.000 | 0.000 | 0.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 9.599 | 10.603 | 10.498 | 10.014 | 10.679 | 10.750 | 12.385 | 10.228 | 10.750 | 10.750 |
| 11.905 | 13.189 | 13.016 | 12.326 | 13.251 | 13.304 | 12.868 | 12.670 | 13.248 | 13.248 |
| 10.183 | 11.291 | 11.323 | 10.614 | 11.420 | 11.642 | 11.326 | 12.294 | 11.326 | 11.326 |
| 15.251 | 16.803 | 16.864 | 15.844 | 16.972 | 17.262 | 16.777 | 15.953 | 19.349 | 19.349 |
| 18.617 | 20.637 | 20.679 | 19.290 | 20.839 | 21.186 | 20.735 | 19.349 | 20.124 | 20.124 |

Table 8: The impact of hyperparameters $a$ and $b$.

| $a$ | $b$ | Amazon-Hybrid @20 | Amazon-Hybrid @50 | Recall | NDCG | Hit Rate | Recall | NDCG | Hit Rate |
|---|---|---|---|---|---|---|---|---|---|
| 0.3 | 0.7 | 7.009 | 10.657 | 12.232 | 11.489 | 17.045 | 20.836 |
| 0.4 | 0.6 | 6.696 | 10.228 | 12.670 | 11.075 | 16.550 | 20.254 |
| 0.5 | 0.5 | 7.033 | 10.750 | 13.304 | 11.642 | 17.262 | 21.186 |
| 0.6 | 0.4 | 7.022 | 10.677 | 13.248 | 11.534 | 17.139 | 20.986 |
| 0.7 | 0.3 | 5.645 | 10.072 | 12.382 | 10.758 | 16.673 | 19.545 |

Table 9: The impact of the time threshold $L_{time}$.

| $L_{time}$ | Amazon-Hybrid @20 | Amazon-Hybrid @50 | Recall | NDCG | Hit Rate | Recall | NDCG | Hit Rate |
|---|---|---|---|---|---|---|---|---|
| 32 | 6.984 | 10.601 | 13.225 | 11.379 | 16.911 | 20.784 |
| 64 | 7.033 | 10.750 | 13.304 | 11.642 | 17.262 | 21.186 |
| 128 | 6.977 | 10.590 | 13.026 | 11.331 | 16.934 | 20.702 |
| 256 | 6.856 | 10.470 | 13.039 | 11.307 | 16.812 | 20.660 |

Time complexity analysis of the GCE module. In GCE, we conducted a lightweight graph convolution instead of a complete GCN. The overall time complexity of the latter is $O(|\mathcal{V}|dK + |\mathcal{E}|d^2K)$ [24, 26], where $|\mathcal{I}|$ represents the number of items, $|\mathcal{V}|$ represents the number of interactions (edges), $d$ is the embedding dimension and $K$ is the depth (i.e., the number of layers) of GCN. However, in the GCE module, only 1, 2 and 3-hop relations are calculated, the process of graph convolution is simplified and only one convolution layer is used. As a result, the time complexity of GCE is reduced to $O(|\mathcal{V}|d)$.

6 CONCLUSION

This paper proposes a novel multi-interest framework for sequential recommendation, which leverages the historical interactions of all users to extract global context information, so as to better learn users’ multiple interests and provide better personalized recommendations. Specifically, the co-occurrence number of item pairs and their time intervals from all interaction sequences are first used to learn the global context embedding of items based on a simplified graph convolution operation. Then, the global context embedding is fused with the recent time interval embedding to get the personalized embedding for each user. Finally, a multi-interest framework is used to learn $K$ interest vectors for each user to provide recommendations. Extensive experiments verify that the potential information contained in the historical sequences can greatly benefit sequential recommendation. Since the proposed module does not use any external information and has lightweight calculations, it can be easily transplanted to any sequential recommendation model. In the future, we will further explore the multi-interest framework by focusing on the non-linear relationship between multiple interests of users and fine-grained representation of items.
[30] Tianyu Zhu, Leilei Sun, and Guoqing Chen. 2021. Graph-based Embedding Smoothing for Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering* PP (04 2021), 1–1. https://doi.org/10.1109/TKDE.2021.3073411