Geoinformatica
https://doi.org/10.1007/s10707-021-00439-w

Graph neural network based model for multi-behavior session-based recommendation

Bo Yu[1,2] · Ruoqian Zhang[1] · Wei Chen[1] · Junhua Fang[1]

Received: 2 February 2021 / Revised: 11 May 2021 / Accepted: 13 May 2021 /
Published online: 29 May 2021

Abstract
Multi-behavior session-based recommendation aims to predict the next item, such as a location-based service (LBS) or a product, to be interacted by a specific behavior type (e.g., buy or click) in a session involving multiple types of behaviors. State-of-the-art methods generally model multi-behavior dependencies in item-level, but ignore the potential of discovering useful patterns of multi-behavior transition through feature-level representation learning. Besides, sequential and non-sequential patterns should be properly fused in session modeling to capture dynamic interests within the session. To this end, this paper proposes a Graph Neural Network based Hybrid Model GNNH, which enables feature-level deeper representations of multi-behavior interaction sequences for session-based recommendation. Specifically, we first construct multi-relational item graph (MRIG) and feature graph (MRFG) based on session sequences. On top of the MRIG and MRFG, our model takes advantage of GNN to capture item and feature representations, such that global item-to-item and feature-to-feature relations are fully preserved. Afterwards, each multi-behavior session is modeled by a seamless fusion of interacted item and feature representations, where self-attention and mean-pooling are used to obtain sequential and non-sequential patterns simultaneously. Experiments on two real datasets show that the GNNH model significantly outperforms the state-of-the-art methods.

Keywords Recommendation systems · Session-based recommendation · Multi-behavior modeling

1 Introduction

In the past few years, session-based recommendation [14, 19, 24, 39] has attracted widespread attention and achieved certain developments. Compared to traditional recommendation, session-based recommendation employs only user interactions during the

✉ Ruoqian Zhang
rqzhang@suda.edu.cn

Extended author information available on the last page of the article.

ongoing session, instead of all historical interactions. With the help of session-based recommendation and development of trajectory matching technique [32, 33, 35], users can be informed of their intended point-of-interests (POIs) or location-based services when browsing, and it thus plays an important role for online LBS systems. Therefore, we can customize online recommendation systems [1, 20, 21] through mining the interests of users from their behaviors. Wang et al. [43] argues that session-based recommendation captures short-term user preferences to provide timely and accurate recommendations and the key challenge is to tackle the problem of modeling user dynamic interests from few interactions within one session. Markov chain [28] is a classical example, which fails to capture the complex dependencies among items in sessions. Recent studies take advantage of deep neural networks to capture user preferences within sessions in a more effective way, such as representative models including RNN-based GRU4Rec [14] and later an improved version [39]. Moreover, attentive networks have been introduced into session-based recommendation [19, 24] and further boost the performance.

The classical session-based recommendation models mentioned above are designed for single-behavior (e.g., POI check-in with smartphones), while in fact, the majority of sessions contain multi-behavior information (e.g., browse the POIs and related reviews, etc). For such multi-behavior sessions, it is necessary to utilize *auxiliary behavior* information to improve the *target behavior* prediction performance. Although some recommendation models [7, 15, 22, 25, 40] have taken multi-behavior modeling into consideration, they are not session-aware and thus not suitable for session-based recommendation. To this end, an improved session-based recommendation model is highly demanded to capture multi-behavior information within sessions in an effective manner.

However, multi-behavior session-based recommendation is very important yet challenging. It is essential to find a mechanism to model the correlations among multiple behaviors. The state-of-the-art method is a model proposed by Wang et al. [45], which successfully incorporates the influences of auxiliary behavior for enhanced recommendation in target behavior. Specifically, it employs Graph Neural Network (GNN) [17] to learn the global item-to-item relations, such that user preferences can be largely inferred by the interacted items of the multi-behavior session. Unfortunately, multi-behavior modeling in [45] is performed in the item-level, meaning that it may not be able to fulfill its potential due to possible insufficient utilization of content information.

Although multi-behavior modeling has been considered in [45] for session-based recommendation, we argue that it is of great significance to model multi-behavior dependencies in a finer granularity. In [45], user behavior correlations are captured in the item-level only. In reality, aside from item-level dependencies, the transition patterns may also appear in feature-level, i.e., category or description text. For example, a user is likely to look for hotels and car rental services after booking a flight ticket, indicating that we should design a neural network model to capture dependencies in both item and feature perspectives simultaneously. However, feature-level dependencies can hardly be captured by existing model. Besides, implicit features from unstructured description texts are also quite crucial, but they are overlooked by state-of-the-art methods. In addition, according to [44], both sequential and non-sequential patterns are essential for capturing session-based dynamic preference. Nevertheless, long-term dependencies are neglected by multi-behavior recommendation method [45], resulting in limitations on modeling certain behaviors (e.g., click).

To solve the aforementioned limitations, we propose a Graph Neural Network based Hybrid Model, namely GNNH in short. This model enables feature-level deeper representations of multi-behavior interaction sequences for session-based recommendation. Our method constructs not only multi-relational item graph (MRIG) from behavior sequences in all sessions, but also multi-relational feature graph (MRFG) corresponding to item. Based on these two graphs, we further take advantage of GNN to perform representation learning for items and explicit features, such that both item-to-item and feature-to-feature relations in multi-behavior sessions can be fully preserved. Moreover, to incorporate the effects of implicit features, our model captures and adaptively selects essential implicit features from textual descriptions by vanilla attention, so that all useful content information can be utilized for more rational recommendation. Besides, we model multi-behavior sessions by a seamless fusion of interacted item and feature representations, in which both self-attention and mean-pooling are used to obtain sequential and non-sequential patterns simultaneously. The main contributions of this paper are summarized as follows:

- We propose a GNNH model for session-based recommendation, which is crucial for LBS systems. GNNH divides multi-behavior representation learning into item-based and feature-based to fully capture multi-behavior transition patterns in a finer granularity.
- We not only capture the explicit transition patterns between features, but also adaptively select useful implicit features derived from textual descriptions by vanilla attention, which contributes to better utilization of content information.
- To make full use of auxiliary behavior information, we propose a carefully designed session representation method, which combines mean-pooling and self-attention together to capture both sequential and non-sequential patterns.
- We carry out extensive experiments on two real datasets. GNNH achieves the best performance among strong competitors, manifesting the benefits of solving these limitations.

The remaining parts are organized as follows: First, we review the related work regarding session-based recommendation and multi-behavior modeling in Section 2, and formulate the problem in Section 3. Then we illustrate our proposed model in Section 4. Finally, we show extensive experimental results in Section 5, and conclude our work in Section 6.

2 Related work

2.1 Session-based recommendation

Session-based recommendation aims to predict user actions on implicit feedback, where sessions are anonymous, and no explicit preferences (e.g., ratings) but only positive observations (e.g., browses or purchases) are provided [13]. Since no user profile can be constructed from history records, classical CF methods (e.g., matrix factorization) break down. In recent years, the majority of researches [14, 19, 39, 41] apply Recurrent Neural Networks for session-based recommendation and achieve promising results. For instance, Hidasi et al. [14] first proposes to model short-term preferences with Gated Recurrent Unit (GRU), and later the model [39] is enhanced by data augmentation and a method to account for temporal

shifts. Besides, attention-based methods have been introduced into recommender systems. Kang et al. [16] proposes a self-attention based sequential model, which outperforms RNN-based sequential recommendation methods. In the session-based setting, [19] utilizes attention mechanism to capture a user's sequential behavior and its main purpose. Nowadays, GNN [12, 51] has been proposed to learn representations of graph-structured data. For instance, [46] utilizes GNN to model item-to-item relations separately and locally in one behavior sequence. It's different that our work encodes representation globally through performing GNN on the graph which contains sequences from all sessions, and we consider both sequential and non-sequential patterns by utilizing mean-pooling and self-attention together.

2.2 Multi-behavior modeling

Multi-behavior recommendation aims to leverage multiple behaviors for boosting the recommendation performance on target behavior. Early studies mainly approach this task from two aspects. One category [25, 27] utilizes multi-behavior data into the sampling process and adopts the strategy of multi-sampling to reinforce the model learning process. The other category [36, 40] designs matrix factorization based model to conduct the factorization on multiple behavior matrices at the same time.

More recently, a neural network approach is proposed by [7] to learn representations for user-item interactions with different behaviors. It is a deep model for multi-task learning by regarding different types of behaviors as the cascading sequences. Though above method takes advantage of the advance in neural network based recommendation, it assumes the independence of different user-item interactions. However, it is more realistic to consider modeling sequential user behaviors in the session setting. [45] is a state-of-the-art multi-behavior session-based recommendation model, which builds item graph based on all behavior sequences from sessions. Based on the graph, it can learn global item-to-item relations and further obtain user preferences. Nevertheless, the model overlooks the feature information and we can also incorporate the transition patterns of user interactions in feature-level to further improve the performance. In addition, long-term dependencies are neglected by [45], especially for behavior sequence (e.g., click). It thus calls for an improved model that can model multi-behaviors in a finer granularity for more accurate session-based recommendation.

2.3 Point-of-interest recommendation

With the development of Location-based Social Network (LBSN) [3, 5, 34, 37], it provides a variety of location information and user check-in behavior. Therefore, we can recommend next Point-of-Interests (POI) to mobile users with the history activities. There are a number of POI-related applications that are very valuable including travel planning [31, 47], crowdsourcing task assignment [23, 30], travel time estimation [49], destination prediction [48], top-k term publish/subscribe [2, 4]. Because LBSN contains various types of implicit information, such as geographical, temporal, context and social information, that are easy to apply to MF and LDA. MF is a good algorithm to apply to implicit information or social information. Some studies [6, 8] utilized MF to quantify the importance of geographical information by finding the representative location of a vast number of POIs. Afterwards, [18] held that recommender algorithms based on probability distribution such as LDA can outperform MF in LBSN-based POI recommendation by using the frequency of items as

a measure of the user's preference in some situations. Recently, to alleviate the problem of data sparsity, [10, 11] maximized AUC by transforming the recommendation task to a classification problem. [9] explicitly utilized similarity with contextual information and incorporated global and local context to achieve good recommendation performance. In our proposed method, we additionally model transition patterns in feature level by considering both explicit features like category and other implict features.

2.4 Multi-relational graph neural network (MGNN) [45]

Existing methods for session-based behavior prediction focused on only utilizing the same type of user behavior for prediction without considering the potential of taking other behavior data as auxiliary information, especially when the target behavior is sparse but important (e.g., buying or sharing an item). Secondly, item-to-item relations are modeled separately and locally in one behavior sequence, and they lack a principled way to globally encode these relations more effectively. To overcome these limitations, MGNN proposed a novel Multi-relational Graph Neural Network for Session-based target behavior prediction. Specifically, MGNN built a Multi-Relational Item Graph (MRIG) based on all behavior sequences from all sessions, involving target and auxiliary behavior types. Based on MRIG, MGNN learned global item-to-item relations and further obtained user preferences. In the end, MGNN leveraged a gating mechanism to adaptively fuse user representations for predicting next item interacted with target behavior.

3 Problem definition

Given an anonymous session set S, each session s has an item sequence representing the *target behavior* (e.g., buy) $P^s = \left[p_1^s, p_2^s, \ldots, p_{|P^s|}^s\right]$ and another item sequence representing the *auxiliary behavior* (e.g., click) $Q^s = \left[q_1^s, q_2^s, \ldots, q_{|Q^s|}^s\right]$. The items of each sequence are arranged in the chronological order of user interaction, where $|P^s|$ and $|Q^s|$ represent the number of items in the respective sequence. Each item in the sequence can be a POI or a location-based service, which has corresponding categorical features (e.g., shop and category) and other textual features (e.g., description text and review). For example, the category of item m is represented as $c_m \in \mathcal{C}$. For the textual features corresponding to each item, we can extract keywords k_m and textual semantics t_m from text.

Moreover, we construct Multi-Relational Item Graph (MRIG) and Multi-Relational Feature Graph (MRFG) from behavior sequences in all sessions. MRIG is similar to MRFG, we formulate MRFG for example. MRFG is a feature graph $G_f = (\mathcal{V}_f, \mathcal{E}_f)$ based on all feature behavior sequences, where $\mathcal{V}_f$ is the set of nodes in the graph containing all available features of the same type (e.g., category), and $\mathcal{E}_f$ is the edge sets involving multiple types of directed edges. Each edge is a triple consisting of the head feature, the tail feature and the type of this edge. For instance, if we construct the category graph based on behaviors of buying and clicking, then an edge $(a, b, \text{buy}) \in \mathcal{E}_f$ means a user bought item in category a and subsequently bought item in category b, and an edge $(a, b, \text{click}) \in \mathcal{E}_f$ means a user clicked item in category b after clicking item in category a.

Then the goal of session-based target behavior prediction is to learn a model that can generate K items which are most likely to be interacted with certain user under target behavior in the next.

4 The proposed model

In this section, we will firstly present the overall architecture of our GNNH for session-based target behavior prediction. As illustrated in Fig. 1, GNNH models multi-behavior sequences in both item-level and feature-level, so that we can capture transition patterns in a finer granularity besides making full use of auxiliary behavior information.

4.1 Overview

The overall architecture of our GNNH model is shown in Fig. 1. There are four modules in the model. Specifically, in the *preprocessing module*, we construct multi-relational item graph (MRIG) and feature graph (MRFG) based on sequences from all sessions, so that complex relationships in both item and feature levels are contained. In the *item-based representation learning module*, we perform GNN on MRIG to learn the representations of items, in which global item-item relations are captured. The *feature-based representation learning module* further learns global feature-feature relations through GNN, so that multi-behavior transition patterns in a finer granularity can be fully obtained. Besides, we also combine representations of both explicit and implicit features through vanilla attention. In the *session representation learning module*, each session is modeled by a seamless fusion of interacted item and feature representations, where self-attention and mean-pooling are utilized simultaneously.

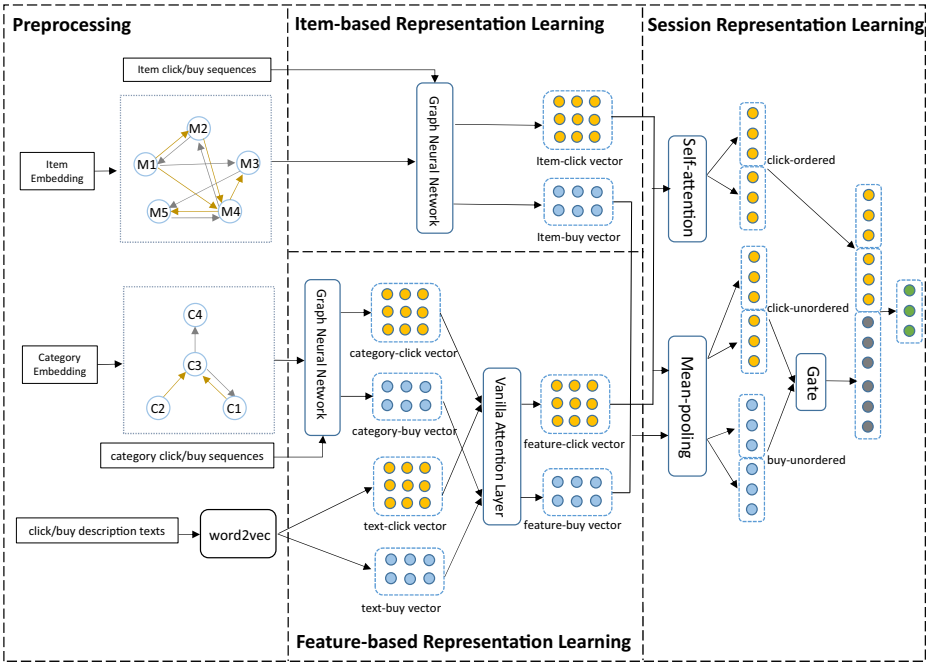


Fig. 1 The framework of our GNNH approach

4.2 Preprocessing module

4.2.1 Embedding layer

Every item and categorical feature should have its corresponding embedding vector. Take category as an example, we use embedding vector $h_{m_i^{(0)}} \in \mathcal{R}^l$ and $h_{c_j^{(0)}} \in \mathcal{R}^l$ to describe an item and a category, where l is the size of each embedding. The item embedding and the category embedding are represented by vector matrices M and C separately.

$$M = \left\{h_{m_1^{(0)}}, h_{m_2^{(0)}}, \ldots, h_{m_n^{(0)}}\right\}, \quad C = \left\{h_{c_1^{(0)}}, h_{c_2^{(0)}}, \ldots, h_{c_k^{(0)}}\right\} \tag{1}$$

For each node on item graph, we use one hot vector to describe their IDs, and then obtain the corresponding dense vector representation through applying a lookup layer on the learnable embedding matrix. For categorical features (e.g., category and shop), the same way is applied. For the textual features corresponding to each item (e.g., description text, review), we first adopt topic model to extract five topical keywords from text, and then apply Word2vector [26] to learn textual semantic representations. Finally, all topical keyword vectors are fused into a vector representation by mean-pooling.

It is worth noting that vectors in item and categorical feature matrices can be directly used as input for the graph neural network framework. Textual vector representations skip GNN and are combined with other vectors directly later.

4.2.2 Construction of multi-relational item graph MRIG

We construct an item graph based on users' historical interaction sequences, which represent multiple relationships between items. For example, if a user buys item M1, and subsequently buys item M2 in the same session, it probably means the dependency between item M1 and item M2, rather than the similarity relationship since users seldom buy similar items continually in a short period. In the same way, if a user clicks item M1, and subsequently clicks item M3, it indicates that item M1 and item M3 are similar according to user habits.

In the item graph, we take all items as nodes and each type of behavior corresponds as one directed edge, denoting different relationships between items. Specially, we browse both target and auxiliary behavior sequences from all sessions P^s and $Q^s (\forall s \in \mathcal{S})$, and collect each item m_i in the sequences as node of graph. When constructing edges from target behavior sequences, we regard each consecutive pair (m_{i-1}, m_i) as one directional edge, indicating target transition relationship. The auxiliary behavior edges are also constructed on MRIG in the same fashion.

4.2.3 Construction of multi-relational feature graph MRFG

In addition to MRIG, it's also necessary to construct a multi-relational feature graph (e.g., category graph) since we also want to capture the explicit feature-to-feature relations. Similar to MRIG, if one user clicks an item in category C3 in the e-commerce system, and subsequently clicks another item in category C1 in the same session, it is very likely that C3 and C1 are similar or even identical. In comparison, if the user purchases an item in category C2 and then purchases another item in category C3, the two categories complement with each other.

The construction of MRFG is similar to MRIG. We take all features of the same type (e.g., category) as nodes and each type of behavior corresponds as one directed edge, denoting different relationships between features. Based on the MRIG and MRFG, we will explain item-based and feature-based representation learning module separately.

4.3 Item-based representation learning module

In this section, we learn item correlations from MRIG by graph neural networks and encode them into item representations. In this way, meaningful transition patterns can be captured in item-level by modeling item sequences of each behavior type separately. Item-level representations of multi-behavior interaction sequences are enabled to contribute to the last session representation learning. Here, we will introduce how Graph Neural Network performs on MRIG as below.

4.3.1 Graph neural network

After feeding the embedding vectors of item with MRIG, we perform graph neural network to obtain comprehensive contextual representations. These representations embed transition relationships of multiple behavior globally because each node gathers neighbor information from all sessions. Since GNN in the item-based and feature-based representation learning module only differs in their inputs, we introduce GNN with MRIG briefly. Specifically, we apply GNN in a similar way as [45].

For each item node m, there are four types of neighboring node group sets according to the type and direction, namely "target-forward", "target-backward", "auxiliary-forward" and "auxiliary-backward". Take target as an example, adjacent node group set of target-forward is defined as below:

$$\mathcal{N}_{\mathrm{t}+}(m) = \left\{ m' \mid \left(m', m, \text{ target} \right) \in \mathcal{E} \right\} \tag{2}$$

Target-backward $\mathcal{N}_{\mathrm{t}-}(m)$, auxiliary-forward $\mathcal{N}_{\mathrm{a}+}(m)$ and auxiliary-backward $\mathcal{N}_{\mathrm{a}-}(m)$ can also reach in a similar fashion. To obtain the representation of target-forward group, we aggregate each item in this group by mean-pooling:

$$\mathbf{h}_{\mathrm{t}+,m}^{k} = \frac{\sum_{m' \in \mathcal{N}_{\mathrm{t}+}(m)} \mathbf{h}_{c'}^{k-1}}{|\mathcal{N}_{\mathrm{m}+}(v)|} \tag{3}$$

The remaining representations of three neighbor groups can be calculated in a similar way. Then, in order to take into account both different relations between items and original representation, we adopt sum-pooling method:

$$\mathbf{h}_{m}^{k} = \mathbf{h}_{\mathrm{t}+,m}^{k} + \mathbf{h}_{\mathrm{t}-,m}^{k} + \mathbf{h}_{\mathrm{a}+,m}^{k} + \mathbf{h}_{\mathrm{a}-,m}^{k} + \mathbf{h}_{m}^{k-1} \tag{4}$$

After K iterations, we take the node representation of last step $\mathbf{h}_m$ as the representation of the corresponding item.

4.4 Feature-based representation learning module

Inspired by [50], we argue that useful patterns cannot be revealed by item-level modeling of multi-behavior sequences only. In fact, feature-level sequences are essential to capture complex dependencies of multi-behavior interactions in a finer granularity. Note that, useful features can be *explicit features* (e.g., category and shop) and *implicit features* (e.g., features derived from description texts).

Therefore, we further propose a feature-level representation learning module to learn meaningful feature-level transition patterns globally. Specifically, we perform GNN on MRFG to capture transition relations between explicit features. Besides, we also obtain implicit feature representations from preprocessing module and combine them with explicit feature representations through vanilla attention to obtain full transition patterns between features.

For explicit features, we feed the embedding vector of feature with MRFG and then perform graph neural network which is the same as item-based representation learning module. Therefore, the final representations of category and other categorical feature such as shop after performing GNN are constructed similarly, which are denoted as $\mathbf{h}_c$ and $\mathbf{h}_s$.

4.4.1 Vanilla attention layer

Since there are various types of features corresponding to items, it is difficult to know how each feature will influence next item to be interacted with target behavior. Therefore, we employ vanilla attention to assist feature-based representation learning module to capture the session's varying interest toward features. For each item m, its features can be embedded as $\mathbf{A}_m = \left\{\mathbf{h}_{s_m}, \mathbf{h}_{c_m}, \mathbf{h}_m^{text}\right\}$, where $\mathbf{h}_{s_m}$ and $\mathbf{h}_{c_m}$ mean dense vector representation of shop and category of item m respectively after utilizing GNN, and the $\mathbf{h}_m^{text}$ denotes the textual representation of item m. We compute weight score of each feature as below:

$$\boldsymbol{\alpha}_m = \text{softmax}\left(\mathbf{W}^f \mathbf{A}_m + \mathbf{b}^f\right) \tag{5}$$

where $\mathbf{W}^f$ is $l * l$ matrices and $\mathbf{b}^f$ is l-dimensional vector. Finally, the feature representation of item m is computed as weighted sum of various feature vectors with attention scores.

$$\mathbf{f}_m = \boldsymbol{\alpha}_m \mathbf{A}_m \tag{6}$$

It is worth noting that if item m only considers one feature (e.g., category), then the feature representation of item m is $\mathbf{h}_{c_m}$.

4.5 Session representation learning module

In this section, each multi-behavior session is modeled by a seamless fusion of interacted item and feature representations, where both self-attention and mean-pooling are used to obtain sequential and non-sequential patterns simultaneously.

4.5.1 Mean-Pooling

Due to most sessions lasting for a short period, direct utilization of mean-pooling to obtain sequence representations can already achieve comparable performance especially for target behavior sequences. Therefore, our model adopts mean-pooling for both item sequences of target behavior and auxiliary behavior to capture unordered item representations. The same is true for feature representations. We denote the unordered representation of target behavior sequence P and auxiliary behavior sequence Q as $\mathbf{p}$ and $\mathbf{q}$, which are defined as follows:

$$\mathbf{p} = \frac{\sum_{i=1}^{|P|}[\mathrm{h}_{p_i}; \mathrm{f}_{p_i}]}{|P|}, \mathbf{q} = \frac{\sum_{i=1}^{|Q|}[\mathrm{h}_{q_i}; \mathrm{f}_{q_i}]}{|Q|} \tag{7}$$

where h and f denote the representation of item-level and feature-level separately.

4.5.2 Self-Attention

There are multi-behavior interactions in the majority of sessions, in which more interactions in auxiliary behavior sequence than target behavior sequence. According to [44], sequential patterns are essential for capturing dynamic preference in long sequences. Therefore, it is necessary to take the position information of auxiliary sequence into consideration to distinguish different items or features in the sequence. We adopt the self-attention network with multi-head attention proposed by [42] to aggregate all previous items' embedding and features' embedding in auxiliary behavior sequence into ordered representation. For simplicity, we define the whole self-attention mechanism as:

$$\mathbf{O}_s = SAN([\mathrm{h}_q; \mathrm{f}_q]) \tag{8}$$

We take the last dimension of $\mathbf{O}_s$ as the final ordered representation, and t denotes t-th line of matrices.

$$\mathbf{o}_s = \mathbf{O}_{s_t} \tag{9}$$

4.5.3 Fusion

We argue that auxiliary sequences contribute differently when building an integrated representation. For instance, some users who have no purchase purpose at first often click on certain items randomly, and then suddenly find items want to buy. So the majority of click item sequences generated in this way are not very helpful for the final purchase. Nevertheless, the large part of items clicked by users with the explicit purchase purpose are related to next items to buy. Therefore, we should distinguish between the representation of target behavior sequence and the representation of auxiliary behavior sequence through gating mechanism. To measure the respective importance of target and auxiliary behavior for the final representation, we use the gating mechanism:

$$\alpha = \sigma\left(\mathbf{W}_g[\mathbf{p}; \mathbf{q}]\right) \tag{10}$$

Then, we get the unordered representation of the current session through weighted summation.

$$\mathbf{o}_u = \alpha \cdot \mathbf{p} + (1 - \alpha) \cdot \mathbf{q} \tag{11}$$

For the ordered representation $\mathbf{o}_s$ obtained from self-attention, we only consider auxiliary sequences. Afterwards, we concatenate ordered and unordered representations and project them into a fully-connected layer.

$$\mathbf{o}_{us} = [\mathbf{o}_u; \mathbf{o}_s]\,\mathbf{W}_{us} + \mathbf{b}_{us} \tag{12}$$

where $\mathbf{W}_{us} \in \mathcal{R}^{4l \times l}$, $\mathbf{b}_{us} \in \mathcal{R}^{l}$. $\mathbf{o}_{us}$ is the final representation of session.

4.6 Objective function

In this section, we want to maximize the prediction probability of the actual next item interacted with target behavior within the current session.

Therefore, we first calculate the recommendation score of each item $m \in \mathcal{M}$ through current session representation $\mathbf{o}_{us}$ and item representation $\mathbf{e}_m$. Then softmax function is applied to normalize scores over all items to get the probability distribution $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{o}_{us}^{\top}\mathbf{W}\mathbf{e}_m) \tag{13}$$

Table 1 Basic statistics of the datasets

Dataset	Yoochoose	Tmall
#items	52740	569658
#sessions	201961	52379
#categories	339	6352
Average length of target	3.31	1.23
Average length of auxiliary	8.56	10.71
#training	163005	44876
#validation	12985	2501
#test	25971	5002

Then, we adopt the cross-entropy loss as the optimization objective function, which is defined as:

$$\mathcal{L} = -\sum_{i=1}^{m} \mathbf{y}_i \log\left(\hat{\mathbf{y}}_i\right) + (1 - \mathbf{y}_i) \log\left(1 - \hat{\mathbf{y}}_i\right) \tag{14}$$

where $\mathbf{y}$ denotes the one-hot representation of the ground-truth item.

5 Experiment

In this section, we conduct extensive experiments on real-word datasets to answer the following research questions:

RQ1 How does our proposed GNNH model perform compared with competitive baselines?
RQ2 How does feature information affect GNNH's performance of the next item prediction?
RQ3 How do different ways of fusion affect the recommendation performance of our approach?

5.1 Datasets

We evaluate the proposed model on two real datasets, i.e., Yoochoose and Tmall. They consist of user behavior records for e-commerce, which are frequently used in recommendation studies. The statistical information of datasets are shown in Table 1.

- **Yoochoose** [1] It can be obtained from RecSys Challenge 2015, consisting of six months of click and buy streams gathered from an e-commerce website. The user behavior sequences in the dataset are already segmented into sessions and all users are anonymized.
- **Tmall** [2] It is a publicly available dataset, which is based on transactional data provided by Alibaba. It contains four types of behaviors that a user can take. In this paper, we only take click and purchase for evaluation. According to the user behaviors recorded in the platform lasting for one month, we need to split the dataset into sessions. Therefore, we set a maximum time range one day for each session sequence and remove user information.

[1] http://2015.recsyschallenge.com/challege.html
[2] https://tianchi.aliyun.com/competition

In both datasets, we treat buying behavior as the *target behavior*, and regard behavior of click as the *auxiliary behavior*. The method of handling datasets is the same as [45]. Behaviors of each session are organized in a chronological order. We take the first 6/7 of datasets as the training data, and use 1/3 of the remaining data as the validation data to determine the optimal hyper-parameter settings. Therefore, we can obtain the sequences of target and auxiliary behaviors that interact with items, and feature sequences can also be achieved according to the item sequences. Here, feature refers to category because of the limitations of datasets. To avoid the auxiliary input already sees the labels, we only keep the clicked items before the target item that is also bought by the user. In addition, MRIG and MRFG utilized throughout the experiments are constructed only based on sequences from training data.

5.2 Baselines

To evaluate the performance of our proposed model GNNH, we choose state-of-the-art and closely related work: (1) Traditional recommendation methods including POP and Item-KNN. (2) RNN-based methods including GRU4Rec and NARM. (3) Attention-based method STAMP. (4) Multi-behavior graph-based method MGNN and our extension to the MGNN method MGNN+. They are briefly described as follows.

- **POP:** This baseline always recommends the most popular items based on occurrence frequency in the training set.
- **Item-KNN** [29]**:** In the baseline, items similar to the existing items based on cosine similarity are recommended.
- **GRU4Rec** [14]**:** It is an RNN based deep learning model for session-based recommendation, which consists of GRU units.
- **NARM** [19]**:** It employs attention mechanism to capture main purpose from hidden states obtained by RNN and combines that with the sequential behavior as final representation.
- **STAMP** [24]**:** This model captures a user's general interest based on the long-term memory of session context and the user's current interest based on the short-term memory of the last clicks in the session.
- **MGNN** [45]**:** It is a state-of-the-art multi-behavior session-based recommendation model, which performs GNN on multi-relational item graph to learn global item-to-item relations.
- **MGNN+:** It is our extension of MGNN method, which replaces the input of original model with combined vector of item and its features.

It is worth noting that the above baselines except MGNN and MGNN+ are designed for single-behavior modeling. To make the comparsion more impartial, we revise these methods by modeling target behavior sequences and auxiliary behavior sequences of both items and features respectively, and then fuse these four sequences with the proposed gating mechanism as ours.

5.3 Evaluation metrics

To evaluate the performance of each model, we apply three widely used common metrics, i.e., Hit-ratio (H@K), Mean Reciprocal Rank (M@K) and Normalized Discounted Cumulative Gain (N@K). H@K is the proportion of cases having the desired item amongst the top-k items in all test cases. M@K is the average of reciprocal ranks of the desired items. It

is equivalent to Mean Average Precision (MAP). N@K takes the position of correctly recommended items into consideration. In our experiments, we choose K = 100 to illustrate different results of H@K, M@K and N@K.

5.4 Parameter settings

We implement our proposed model based on Tensorflow. The dimension of item and feature embedding is set to 128 and both vectors are initialized with zero. We use dropout [38] with drop ratio $p = 0.2$. For the hyper-parameters of the Adam optimizer, we also set the default value. To speed up the training and converge quickly, GNN is ensured to run in mini-batch size of 64, while the depth of GNN on item graph is set to 2 and the depth of GNN on category graph is set to 1. In self-attention module, the number of blocks is set to 2 and number of heads is set to 2. All baselines are tested on different forms of attention computation formulas for best results.

5.5 Performance comparsion (RQ1)

Table 2 shows the performance comparison between our model and the adopted baselines. Comparing the performance of all methods over two datasets, we find that the results of Yoochoose are generally better than Tmall. This may result from differences in data sparsity.

Firstly, the performances of traditional methods such as POP and Item-KNN are not competitive. These results suggest the importance of taking the user's behavior into consideration in session-based recommendation tasks rather than solely basing on similarity.

Secondly, all of the neural network baselines significantly outperform conventional models, proving the effectiveness of deep learning technology in this field. GRU4Rec, NARM, and STAMP are standard sequential recommendation models for session, and their performances are better than non-sequential models. This demonstrates the necessity of considering sequential information in next item recommendation. Moreover, SR-GNN, GC-SAN and HetGNN achieve better results than previous methods, because these methods utilize the Graph Neural Network to learn high-order transition information. Different from

Table 2 Evaluation results of all methods

Method	Yoochoose			Tmall		
	H@100	M@100	N@100	H@100	M@100	N@100
POP	6.095	0.2529	1.2231	0.928	0.1074	0.3165
Item-KNN	15.286	1.9415	4.4040	1.861	0.1392	0.4381
GRU4Rec	19.340	2.5373	5.5919	2.447	0.1650	0.5382
NARM	18.835	2.5997	5.5975	2.392	0.1643	0.5484
STAMP	20.481	2.3516	5.6944	2.595	0.1647	0.5609
SR-GNN	21.383	2.6935	6.1397	2.637	0.1669	0.5750
GC-SAN	19.875	2.5719	5.6929	2.747	0.1811	0.5957
HetGNN	24.168	2.9956	6.8852	2.981	0.1834	0.6549
MGNN	28.632	3.6564	8.2722	3.114	0.1821	0.6617
MGNN+	28.982	3.9470	8.5771	3.276	0.1894	0.6784
GNNH	29.138	4.5997	9.2342	3.409	0.2004	0.7583

previous two method, HetGNN constructs a heterogeneous graph with multiple types of nodes and edges, and further improves the performance.

Thirdly, MGNN is the best baseline of multi-behavior session-based recommendation, which learns global item-to-item relations. However, this method doesn't consider feature transition and only models non-sequential patterns of user behaviors. To make the comparison fairer, MGNN+ improves the performance than MGNN by concatenating item representations and feature representations together as input of graph, so we just perform graph representation propagation on item graph, which is not the best method to consider feature information.

Finally, regardless of the datasets and the evaluation metrics, our proposed GNNH achieves the best performance. Compared with MGNN+, our method models both item-level sequences and feature-level sequences, and additionally considers sequential patterns of auxiliary behavior. This result shows the effectiveness of our GNNH model.

5.6 Impact of feature information (RQ2)

The paper [45] has already demonstrated that considering the auxiliary behavior in the model indeed boosts the performance of session-based recommendation. In our GNNH method, we further take multiple types of feature into consideration. However, due to the limitation of available datasets, we simplify our proposed model with only a category feature. In the Table 3, methods with "w/o c" denote removing category information from the full version. It is shown that our GNNH method still performs better than other models in the setting. Moreover, by comparing each method in Table 2 with its "w/o c" version, we can find that when category information is utilized, performance of each method improves in a certain scale. This demonstrates that considering category information is indeed meaningful. We could expect further improvements of performance if we consider more categorical features and implicit features.

Furthermore, we want to test the impact of different depths of GNN, including both depth settings on item graph and feature graph. From the Fig. 2, we can infer the effectiveness of graph neural network module. When both depth values are 0, it means removing the graph neural network module from our model, and replacing it with randomly initialized item and feature embedding. We can see that the performance becomes significantly better when depth of GNN on item graph grows from 1 to 2 and depth of GNN on feature graph grows from 0 to 1, showing that it is indispensable to model high-order relations between items and features through GNN. When the number of depth increases continually, the performance would be worse since the representations of items and features might become less distinguishable. We can also find out that the best value of item and feature is different.

Table 3 Results of not using category information

Method	Yoochoose			Tmall		
	H@100	M@100	N@100	H@100	M@100	N@100
GRU4Rec(w/o c)	19.114	2.5292	5.5830	2.423	0.1645	0.5281
NARM(w/o c)	18.775	2.5819	5.5813	2.266	0.1549	0.5387
SR-GNN(w/o c)	21.262	2.6892	6.1232	2.582	0.1596	0.5562
GNNH (w/o c)	**28.815**	**4.2519**	**8.7829**	**3.214**	**0.1974**	**0.7123**

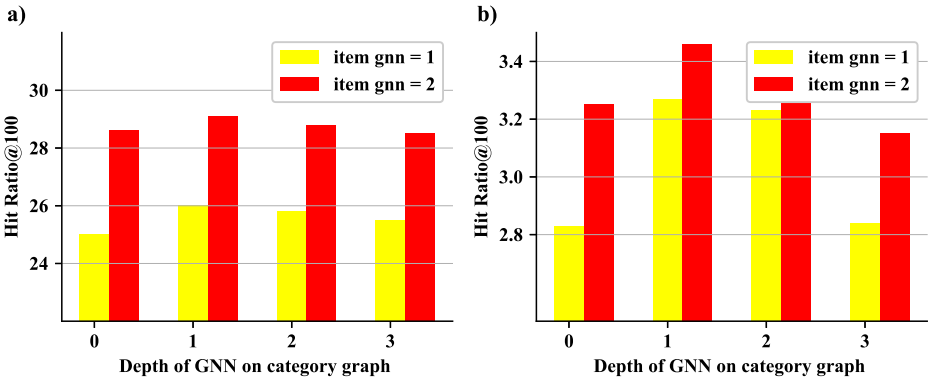


Fig. 2 Results of our model with different depths of GNN

This is due to the fact that the number of items is much larger than that of features, and higher-order relations between features might be meaningless.

5.7 Impact of different ways of fusion (RQ3)

In this section, we conduct experiments to analyze the contribution of self-attention blocks to the model. In the Fig. 3, using 'mp' denotes we only utilize mean-pooling, while using 'mp + sa' means we also consider self-attention simultaneously. Here, the datasets are divided into several portions according to the length of auxiliary behavior sequences. From Fig. 3, when the length of auxiliary behavior sequences is in the range from 3 to 5, the improved effect of utilizing self-attention is insignificant since the representations of short sequences can already be captured well by mean-pooling. However, the performance of GNNH with self-attention module would be improved in a larger scale as the length of auxiliary behavior sequences increases from 8 in the experiment on two datasets. This demonstrates that self-attention module could extract more information of user's current interest because of assigning different importance weights to the items or features in long

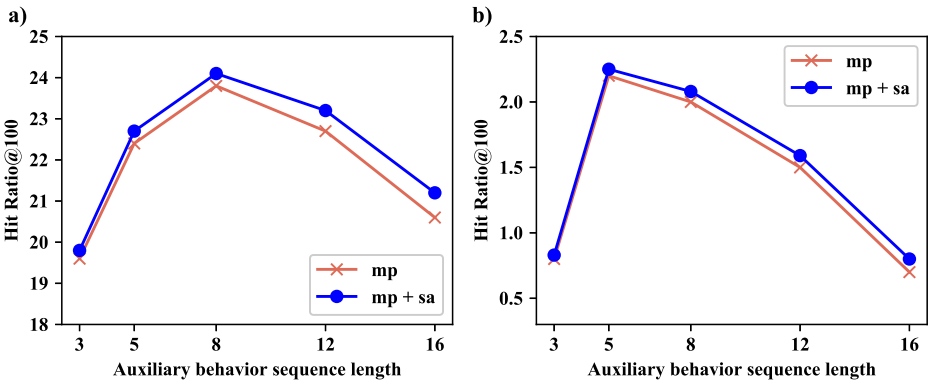


Fig. 3 Results of our model with different ways of fusion

sequence. Moreover, we also try to test the fusion on target sequence. We can find that simple mean-pooling can already achieve comparable performance while retaining low complexity.

6 Conclusion

In this paper, we propose a novel graph neural network based hybrid model (GNNH), which enables feature-level deeper representations of multi-behavior interaction sequences for session-based recommendation. Specifically, we observe that user interactions in feature-level also contribute to target behavior prediction. Therefore, we construct both MRIG and MRFG and capture global item-to-item and feature-to-feature relations through GNN. In addition, we also employ a seamless fusion of interacted item and feature representations to obtain both sequential and non-sequential patterns. Finally, extensive experiments on two real datasets show that our proposed method significantly outperforms the state-of-the-art methods.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant Nos. 61872258, 61802273, Major project of natural science research in Universities of Jiangsu Province under grant number 20KJA520005.

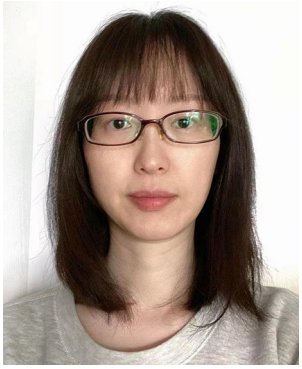
References

1. Auer P (2002) Using confidence bounds for exploitation-exploration trade-offs. JMLR, pp 397–422
2. Chen L, Shang S, Jensen CS, Xu J, Kalnis P, Yao B, Shao L (2020) Top-k term publish/subscribe for geo-textual data streams. VLDB, pp 1101–1128
3. Chen L, Shang S, Yang C, Li J (2020) Spatial keyword search: a survey. GeoInformatica, pp 85–106
4. Chen L, Shang S, Zhang Z, Cao X, Jensen CS, Kalnis P (2018) Location-aware top-k term publish/subscribe. In: ICDE, pp 749–760
5. Chen X, Xu J, Zhou R, Zhao P, Liu C, Fang J, Zhao L (2020) S^2r-tree: a pivot-based indexing structure for semantic-aware spatial keyword search. GeoInformatica, pp 3–25
6. Cheng C, Yang H, King I, Lyu MR (2012) Fused matrix factorization with geographical and social influence in location-based social networks. In: AAAI
7. Gao C, He X, Gan D, Chen X, Feng F, Li Y, Chua T (2019) Neural multi-task recommendation from multi-behavior data. In: ICDE, pp 1554–1557
8. Gao R, Li J, Li X, Song C, Chang J, Liu D, Wang C (2018) STSCR: Exploring spatial-temporal sequential influence and social information for location recommendation. Neurocomputing, pp 118–133
9. Han P, Li Z, Liu Y, Zhao P, Li J, Wang H, Shang S (2020) Contextualized point-of-interest recommendation. In: IJCAI, pp 2484–2490
10. Han P, Shang S, Sun A, Zhao P, Zheng K, Kalnis P (2019) Auc-mf: point of interest recommendation with auc maximization. In: ICDE, IEEE, pp 1558–1561
11. Han P, Shang S, Sun A, Zhao P, Zheng K, Zhang X (2021) Point-of-interest recommendation with global and local context. IEEE Transactions on Knowledge and Data Engineering
12. Han P, Yang P, Zhao P, Shang S, Liu Y, Zhou J, Gao X, Kalnis P (2019) Gcn-mf: disease-gene association identification by graph convolutional networks and matrix factorization. In: SIGKDD, pp 705–713
13. He X, Zhang H, Kan M, Chua T (2016) Fast matrix factorization for online recommendation with implicit feedback. In: SIGIR, pp 549–558
14. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D (2016) Session-based recommendations with recurrent neural networks. In: ICLR
15. Jin B, Gao C, He X, Jin D, Li Y (2020) Multi-behavior recommendation with graph convolutional networks. In: SIGIR, pp 659–668
16. Kang W, McAuley JJ (2018) Self-attentive sequential recommendation. In: ICDM, pp 197–206
17. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: ICLR

18. Kotzias D, Lichman M, Smyth P (2019) Predicting consumption patterns with repeated and novel events. TKDE, pp 371–384
19. Li J, Ren P, Chen Z, Ren Z, Lian T, Ma J (2017) Neural attentive session-based recommendation. In: CIKM, pp 1419–1428
20. Li L, Chu W, Langford J, Schapire RE (2010) A contextual-bandit approach to personalized news article recommendation. In: WWW, pp 661–670
21. Li L, Lu Y, Zhou D (2017) Provable optimal algorithms for generalized linear contextual bandits. coRR
22. Li Y, Xu J, Zhao P, Fang J, Chen W, Zhao L (2020) Atlrec: an attentional adversarial transfer learning network for cross-domain recommendation. JCST, pp 794–808
23. Liu A, Wang W, Shang S, Li Q, Zhang X (2018) Efficient task assignment in spatial crowdsourcing with worker and task privacy protection. GeoInformatica, pp 335–362
24. Liu Q, Zeng Y, Mokhosi R, Zhang H (2018) STAMP: Short-term attention/memory priority model for session-based recommendation. In: SIGKDD, pp 1831–1839
25. Loni B, Pagano R, Larson MA, Hanjalic A (2016) Bayesian personalized ranking with multi-channel user feedback. In: Recsys, pp 361–364
26. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Nips, pp 3111–3119
27. Rendle S, Freudenthaler C, Gantner Z (2009) Schmidt-thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: UAI, pp 452–461
28. Rendle S, Freudenthaler C (2010) Schmidt-thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: WWW, pp 811–820
29. Sarwar BM, Karypis G, Konstan JA, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: WWW, pp 285–295
30. Shang S, Chen L, Jensen CS, Wen J, Kalnis P (2018) Searching trajectories by regions of interest. In: ICDE, pp 1741–1742
31. Shang S, Chen L, Wei Z, Jensen CS, Wen J, Kalnis P (2017) Collective travel planning in spatial networks. In: ICDE, pp 59–60
32. Shang S, Chen L, Wei Z, Jensen CS, Zheng K, Kalnis P (2017) Trajectory similarity join in spatial networks. VLDB, pp 1178–1189
33. Shang S, Chen L, Wei Z, Jensen CS, Zheng K, Kalnis P (2018) Parallel trajectory similarity joins in spatial networks. VLDB, pp 395–420
34. Shang S, Chen L, Zheng K, Jensen CS, Wei Z, Kalnis P (2019) Parallel trajectory-to-location join. TKDE, pp 1194–1207
35. Shang S, Ding R, Zheng K, Jensen CS, Kalnis P, Zhou X (2014) Personalized trajectory matching in spatial networks. VLDB, pp 449–468
36. Singh AP, Gordon GJ (2008) Relational learning via collective matrix factorization. In: SIGKDD, pp 650–658
37. Song X, Xu J, Zhou R, Liu C, Zheng K, Zhao P, Falkner N (2020) Collective spatial keyword search on activity trajectories. GeoInformatica, pp 61–84
38. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. JMLR, pp 1929–1958
39. Tan YK, Xu X, Liu Y (2016) Improved recurrent neural networks for session-based recommendations. In: Recsys, pp 17–22
40. Tang L, Long B, Chen B, Agarwal D (2016) An empirical study on recommendation with multiple types of feedback. In: SIGKDD, pp 283–292
41. Tuan TX, Phuong TM (2017) 3d convolutional networks for session-based recommendation with content features. In: Recsys, pp 138–146
42. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: NIPS, pp 5998–6008
43. Wang S, Cao L, Wang Y (2019) A survey on session-based recommender systems. coRR
44. Wang S, Hu L, Wang Y, Cao L, Sheng QZ, Orgun MA (2019) Sequential recommender systems: Challenges, progress and prospects. In: IJCAI, pp 6332–6338
45. Wang W, Zhang W, Liu S, Liu Q, Zhang B, Lin L, Zha H (2020) Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In: WWW, pp 3056–3062
46. Wu S, Tang Y, Zhu Y, Wang L, Xie X, Tan T (2019) Session-based recommendation with graph neural networks. In: AAAI, pp 346–353
47. Xu J, Gao Y, Liu C, Zhao L, Ding Z (2015) Efficient route search on hierarchical dynamic road networks. DPD, pp 227–252
48. Xu J, Zhao J, Zhou R, Liu C, Zhao P, Zhao L (2021) Predicting destinations by a deep learning based approach. TKDE, pp 651–666

49. Xu S, Zhang R, Cheng W, Xu J (2020) Mtlm: a multi-task learning model for travel time estimation. GeoInformatica
50. Zhang T, Zhao P, Liu Y, Sheng VS, Xu J, Wang D, Liu G, Zhou X (2019) Feature-level deeper self-attention network for sequential recommendation. In: IJCAI, pp. 4320–4326
51. Zhang Y, Liu Y, Han P, Miao C, Cui L, Li B, Tang H (2020) Learning personalized itemset mapping for cross-domain recommendation. In: IJCAI, pp 2561–2567

Bo Yu is a postgraduate student in Soochow University. His research interests mainly include data mining, recommendation systems.



Ruoqian Zhang is a faculty member of school of computer science and technology, Soochow university. Her research interests include databases and data mining.

Wei Chen is currently a lecture in School of Computer Science and Technology, Soochow University. His research interests include data mining and spatial-temporal database.

Junhua Fang is an associate professor in Soochow University. His research interests include distributed stream processing, cloud computing and spatio-temporal database.

Affiliations

Bo Yu[1,2] · Ruoqian Zhang[1] · Wei Chen[1] · Junhua Fang[1]

Bo Yu
20194227029@stu.suda.edu.cn

Wei Chen
robertchen@suda.edu.cn

Junhua Fang
jhfang@suda.edu.cn

[1] Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, Suzhou, China

[2] Neusoft Corporation, Shenyang, China