Intention-oriented Hierarchical Bundle Recommendation with Preference Transfer

Meng Tan[1], Wei Chen[1,*], Weiqing Wang[2], An Liu[1], Lei Zhao[1,*]
[1]Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, Suzhou, China
[2]Faculty of Information Technology, Monash University, Melbourne, Australia
Email: mtan23@stu.suda.edu.cn, {robertchen, anliu, zhaol}@suda.edu.cn, Teresa.Wang@monash.edu

***Abstract*—Bundle recommendation offers promotions of bundled items instead of a single one, which is a common strategy for sales revenue increase and latent customer mining. Due to the scarcity of user-bundle interactions, it is compulsory to go beyond modeling user-bundle interactions and take user-item interactions into account. Existing studies consider user-item interactions by sharing model parameters or learning representation in a multi-task manner or modeling representation based on graph neural network. However, such methods ignore the mutual influence between user preferences for items and bundles. Moreover, they fail to analyse the intentions behind users' purchase behaviors, which can be utilized to make better bundle recommendation. To overcome the drawbacks of existing studies, we propose a novel model IHBR (Intention-oriented Hierarchical Bundle Recommendation with Preference Transfer). Specifically, we consider the co-purchase and co-occurrence information within items for modeling intention-oriented hierarchical representations. Furthermore, we provide a new perspective to exploit mutual influence between user preferences for items and bundles. The experimental results obtained on two real-world datasets demonstrate that our method outperforms the state-of-the-art baselines.**

***Index Terms*—bundle recommendation, graph convolutional network, hierarchical architecture, preference transfer**

I. Introduction

Owing to the prevalence of the World Wide Web, users are faced with the issue of information overload [1]. To help users overcome the obstacle of over-choice and target their interests easily, the recommender systems have gradually been an indispensable role in information retrieval systems [2]. According to existing work, the recommender systems can be mainly categorized into CF-based recommender system [3], content-based recommender system [4], and hybrid recommender system [5].

Distinct from the traditional task of recommending an individual item to users, bundle recommendation focuses on offering a collection of items as one bundle [6] and underlines many applications, such as music playlists on Netease Cloud Music, booklists on Youshu, and game bundles on Steam. For example, the music platform Netease Cloud Music usually provides a bundle for a user in the daily recommended list, and the bundle contains a set of songs that the user may be interested in. In fact, bundle recommendation is a widely deployed strategy for its benefits on both vendors and purchasers [7]. Concretely, the purchasers are exposed to more selections related to the target interest, and the vendors can increase gross merchandise volume.

Most general recommendation algorithms generate an item suggestion for a user based on user preferences and item features extracted from user-item historical interactions [8]. Intuitively, we can treat the bundle as a virtual item and apply general recommendation solutions based on user-bundle interactions. Although technically feasible, the lack of user-bundle interactions results in the suboptimal performance for bundle recommendation [9]. Additionally, bundle recommendation is a challenging task, since it is difficult to catch the intricate relationships within items, bundles, and users. As shown in Figure 1, a bundle is composed of multiple items and a user may interact with several different items and bundles. General recommendation solutions merely concentrate on user preferences for bundles and bundle features, ignoring the fact that associations among items and bundles as well as user preferences for both bundles and items can be further exploited to enhance recommendation performance [10]. From this perspective, general recommendation solutions can be further optimized in bundle recommendation.

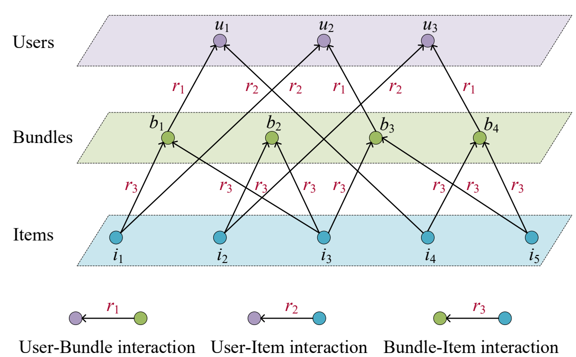


Fig. 1. An illustration of complicated relationships among users, bundles, and items.

Recently, many studies have been conducted for bundle recommendation and achieve state-of-the-art performance, such as BGCN [11], DAM [9], and BundleNet [12]. DAM and BundleNet push forward the bundle recommendation under the framework of multi-task learning. They adopt the Bayesian Personalized Ranking loss function [10] and reinforce the bundle recommendation performance by conducting item prediction and bundle prediction simultaneously. However, they

*Corresponding author

978-1-6654-1681-8/21/$31.00 ©2021 IEEE
DOI 10.1109/ICWS53863.2021.00026

not only fail to distinguish user preferences for items from user preferences for bundles well, but also ignore the mutual influence between user preference for items and bundles. Despite the great contribution made by BGCN, it models user preference for items and bundles separately but still neglects the mutual influence between user preference for items and bundles, which should originate from the associations between items and bundles. Additionally, existing work, such as [11], thinks that a user may refuse a bundle because of the existence of one disliked item. However, it overlooks the intention behind what makes a user interact with (e.g., click, purchase, or review) the bundle, which can be utilized to optimize the bundle recommendation. We observe that the intentions should arise from two aspects: the preference for a single item or multiple items in a bundle. Apparently, a user's preference for a bundle depends on the items contained in the bundle. As a result, more attention should be paid to relationships within items.

Having observed the limitations of mentioned approaches, we propose a novel model IHBR, which enables users to find the target bundle. The model tackles the problem with the following steps. (1) We hypothesize that an interaction with a single item in a bundle emphasizes more on user preference for a single item, while an interaction with multiple items in a bundle concentrates more on correlations among items in this bundle. Thus, we explore the co-purchase relationships within items based on historical user-item interactions and co-occurrence relationships within items based on bundles' composition information in order to learn better item representation based on their related items. The complicated relationships among items, bundles, and users are considered in a hierarchical way. Note that, items show different dependencies when in different bundles, we employ a self-attention mechanism for item dependency exploration. Then, our method can update the representations from bottom to top. Specifically, given the bundles' composition, we incorporate meaningful item information into bundle representation and further refine user representation by his/her historical items and bundles. (2) To better model user preference transfer between items and bundles, we design a user preference transfer paradigm in node propagation procedure between layers based on associations between items and bundles. After that, the user preferences for both items and bundles can be mutually reinforced. (3) We concatenate preference-specific user representation, preference-specific bundle representation, and their element-wise product, and feed the result into a multi-layer neural network to get the predicted score.

The contributions of our work are summarized as follows:

- To our best knowledge, this is the first attempt to introduce user intention and user preference transfer paradigm into the bundle recommendation. And we devise a network called NPPT (**N**ode **P**ropagation based on **P**reference **T**ransfer) for capturing the mutual influence between user preferences for items and bundles.
- We analyse the user intentions and highlight the co-purchase and co-occurrence relationships within items. Next, the complicated relationships among items, bundles, and users are considered in a hierarchical way.
- The results of extensive experiments conducted on two public datasets demonstrate that the proposed IHBR outperforms the state-of-the-art baselines in terms of Recall, MRR, and NDCG. The code is available at https://github.com/IEEEICWS2021/IHBR.

II. Related Work

A. General Recommendation

Collaborative filtering [3] is a widely deployed solution in recommender systems, which models user preferences and recommends items to users based on user-item interactions. Matrix factorization [13] factorizes a user-item interaction matrix into the representation of users and items, and conducts an inner-product on them to get the prediction score. [10] is a basic pair-wise ranking algorithm under the framework of Matrix factorization. Recently, we have witnessed the prosperity of deep learning in the recommender system. [14] leverages a multi-layer perceptron to learn user-item interactions on the basis of collaborative filtering. Due to the strong power of graph neural network, it has applied to recommender system [15]. Ying et al. develop a data-efficient algorithm PinSage in [16], which combines efficient random walks and graph convolutions to generate the representation of items. Though most general recommendation models show their superiority in recommender systems, it is infeasible to apply the general recommendation models to bundle recommendation for their limitations in capturing complicated relationships among items, bundles, and users.

B. Bundle Recommendation

Standing on the advances in the recommender system, several efforts have been dedicated to the bundle recommendation. LIRE [17] takes users' previous interactions with both item lists and individual items into consideration. EFM [18] is proposed to capture user preferences over item and item lists and utilize embedding-based models to discover the associations among items and item lists. Recently, DAM [9] devises a factorized attention network for item information aggregation and jointly models user-bundle interactions and user-item interactions in a multi-task manner. Graph learning based recommender system [19] is an emerging topic and shows better performance. BGCN [11] combines the user preferences for items with bundles' composition information to capture the bundle feature. BundleNet [12] constructs a user-item-bundle tripartite graph from the historical interactions and extends the GCN model to a relational graph. However, such methods attach more importance to associations between items and bundles, ignoring the mutual influence between user preferences for items and bundles. Bai et al. propose a bundle generation network (BGN) and decompose the problem into quality/diversity parts to produce a high-quality and diversified bundle list with an appropriate bundle size in [20]. It concentrates more on bundle generation rather than bundle recommendation.

Similar to bundle recommendation that recommends multiple items concurrently, the next basket recommendation [21] organizes multiple items as a basket by temporal attributes. The next basket recommendation [22] usually can be formalized as a sequential prediction problem [23]. Obviously, sequential-based algorithms are not applicable in bundle recommendation due to the lack of sequential relationships between bundles.

C. Items in Recommendation

Recently, researches on the relationships within items have aroused considerable attention in recommender systems. Markov Chain [21] is a classic solution in modeling item relationships, it bases next item prediction on historical items. [24] explores the sequential pattern within items by the RNN. [25] considers consistency between items and proposes a consistency-aware recommendation. [26] considers both relevance and diversity of items for predicting complementary item types. [27] proposes a semi-parametric embedding framework for the item-item recommendation by using content information and behavior information. [28] focuses on pairwise correlations among items for better recommendation performance. However, little research about item relationships has been done in bundle recommendation.

For the sake of the limitations of existing studies in bundle recommendation, we propose the novel model IHBR, which contains both the item relationship modeling based on user intentions and preference transfer between items and bundles. Moreover, the complicated relationships among items, bundles, and users are considered in a hierarchical manner.

III. Preliminary

To process graph-structured data, Kipf et al. generalize convolutional operations to topological structure in [29], and the propagation rule between layers is defined as:

$$\mathbf{H}^{(l+1)} = \sigma(\mathbf{W}^{(l+1)}(\mathbf{D}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}\mathbf{H}^{(l)}) + b^{(l+1)}) \quad (1)$$

where $\mathbf{A}$ is the adjacency matrix constructed by input graph, such as G_{UI} introduced later. $\mathbf{I}$ is the identity matrix, $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is proposed for self-accessible, $\mathbf{H}^{(l)}$ is the output of $(l-1)th$ layer. If $l = 0$, $\mathbf{H}^{(0)}$ is the initial embedding matrix of nodes in graph $\mathcal{G}$. $\mathbf{W}^{(l)}$ is the trainable weight matrix, σ is the non-linear activation function ReLU [30], and $\mathbf{D}$ is the diagonal degree matrix $\mathbf{D} = \sum_j \mathbf{A}_{ij}$. To get the embedding of each node in an interaction graph, we concatenate outputs of different layers:

$$\mathbf{H} = \mathbf{H}^{(0)} \parallel \ldots \parallel \mathbf{H}^{(L)} \quad (2)$$

where L denotes the number of layers and $\parallel$ is the concatenate operation.

Suppose we have l users $U = \{u_1, u_2, \cdots, u_l\}$, m items $I = \{i_1, i_2, \cdots, i_m\}$, and n bundles $B = \{b_1, b_2, \cdots, b_n\}$. The interactions (e.g., click, purchase, and review) between users and items can be represented as edges in user-item interaction graph G_{UI}. Similarly, when a user has an interaction with a bundle, the interaction can be represented as an edge in user-bundle interaction graph G_{UB}. When an item belongs to a bundle, the composition relationship can also be represented as an edge in a bundle-item interaction graph G_{BI}.

We initialize the embedding of users, items, and bundles by randomly. Then, we can get user representation as $E^{user} = \{\mathbf{e}_u^{user} | u \in U, \mathbf{e}_u^{user} \in R^d\}$, item representation as $E^{item} = \{\mathbf{e}_i^{item} | i \in I, \mathbf{e}_i^{item} \in R^d\}$, and bundle representation as $E^{bundle} = \{\mathbf{e}_b^{bundle} | b \in B, \mathbf{e}_b^{bundle} \in R^d\}$. Here, d denotes the dimension. Next, our task is defined as recommending bundles that the target user may like.

a) Input: User embedding E^{user}, item embeddingE^{item}, bundle embedding E^{bundle}, user-item interaction graph G_{UI}, user-bundle interaction graph G_{UB}, bundle-item interaction graph G_{BI}.

b) Output: A personalized predicted score, which maps the probability that a user will be interested in a bundle.

IV. Proposed Model IHBR

In this section, we introduce the proposed model IHBR and describe the major components in detail. The basic structure of IHBR is illustrated by Algorithm 1. The model is constructed by several components: 1) Intention-oriented hierarchical representation learning, which explores the associations among items based on intention analysis and captures the hierarchical information among items, bundles, and users. 2) Node propagation based on preference transfer, which embeds structural information into node representation and designs a user preference transfer paradigm in convolution procedure. 3) Model prediction and optimization, which outputs the predicted score for bundle recommendation and optimizes the model. Figure 2 depicts the overall architecture of our model.

A. Intention-oriented Hierarchical Representation Learning

We design an intention-oriented hierarchical architecture to exhibit complicated relationships among items, bundles, and users. Firstly, we analyze the intentions behind users' interactions with bundles and pay more attention to item representation modeling. In order to obtain aggregated bundle representation, we utilize a self-attention mechanism to explore the dependencies among items. At last, we update the user representation based on items and bundles that the user has interacted with.

1) Item Representation Learning: A user's selection for a bundle can be affected by items in it. For instance, a user who interacts with an item in a certain bundle may care more about the target item rather than the bundle. As for a user who interacts with a bundle for multiple items, he/she may care more the multiple items in the bundle. Hence, we pay more attention to item representation modeling.

On one hand, we explore co-purchase relationships within items from historical user-item interactions under the circumstance of purchasing a single item in a bundle. To this end, we can get the item-item co-purchase matrix $\mathbf{M}_\alpha$ from historical user-item interactions by:

$$\mathbf{M}_\alpha = \mathrm{softmax}(\mathbf{W}_{cp}\mathbf{A}_{ui}^T\mathbf{A}_{ui}) \quad (3)$$

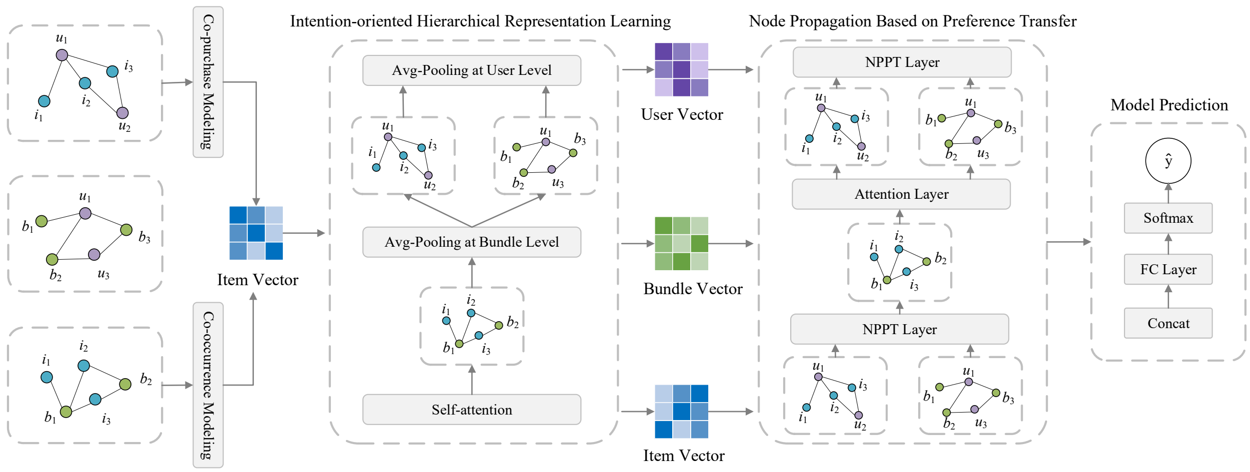


Fig. 2. Overview of the proposed model IHBR (assume dimension d=3). The IHBR takes user-item interactions, user-bundle interactions, and bundle-item interactions as inputs. It contains three components: (1) Intention-oriented hierarchical representation learning. (2) Node propagation based on preference transfer. (3) Model prediction.

Algorithm 1: Architecture of IHBR

Data: User set U, item set I, bundle set B, graphs G_{UI}, G_{UB}, and G_{BI}

Output: A personalized predicted score $\widehat{y}$

```
Initialize representations of user, item and bundle
Update item representation based on its co-purchased
  and co-occurred items with Eq.(5), (6), (7)
Model item dependency by self-attention with Eq.(8),
  (9)
Update bundle representation based on its constitute
  item with Eq.(10), (11)
Update user representation based on his/her historical
  items and bundles with Eq.(12), (13), (14)
Conduct NPPT learning on G_UI and G_UB
while layer < 2 do
    Aggregate bundle representation and item
      representation with Eq.(23), (25)
    Get attention-specific representation with Eq.(22),
      (24), (26)
    Conduct next layer NPPT learning with Eq.(27),
      (28)
end
Concatenate representations and feed the concatenated
  result into an MLP with Eq.(29), (30)
Update all parameters with loss function Eq.(31) and
  Adam optimization algorithm
```

Result: $\widehat{y}$.

where $\mathbf{W}_{cp}$ is the trainable weight matrix, $\mathbf{A}_{ui}$ is the adjacency matrix of G_{UI}.

On the other hand, we exploit co-occurrence relationships within items from bundles' composition information under the circumstance of purchasing multiple items in a bundle. Co-occurrence relationships within items depend on how users perceive those items, regardless of their superficial similarity. As such, we can get the item-item co-occurrence matrix $\mathbf{M}_{\beta}$ from existing bundles' composition information by:

$$\mathbf{M}_{\beta} = \mathrm{softmax}(\mathbf{W}_{co}\mathbf{A}_{bi}^{T}\mathbf{A}_{bi}) \tag{4}$$

where $\mathbf{W}_{co}$ is the trainable weight matrix, $\mathbf{A}_{bi}$ is the adjacency matrix of G_{BI}.

To obtain a more comprehensive item representation, we encode the co-purchase and co-occurrence relationships of an item into its representation. Concretely, given items that are co-purchased with item i, we can obtain a new representation of item i by:

$$\mathbf{e}_{i\leftarrow cp}^{item} = \sum_{j\in P(i)} \alpha_{ij}\mathbf{e}_{j}^{item} \tag{5}$$

where $P(i)$ is the co-purchased item set of item i, and α_{ij} is the corresponding attention score in $\mathbf{M}_{\alpha}$.

Similarly, we can get item vector $\mathbf{e}_{i\leftarrow co}^{item}$ of item i from co-occurred item set $O(i)$:

$$\mathbf{e}_{i\leftarrow co}^{item} = \sum_{j\in O(i)} \beta_{ij}\mathbf{e}_{j}^{item} \tag{6}$$

where β_{ij} is the corresponding attention score in $\mathbf{M}_{\beta}$.

After that, we fuse item vectors $\mathbf{e}_{i\leftarrow cp}^{item}$, $\mathbf{e}_{i\leftarrow co}^{item}$ with the original item vector $\mathbf{e}_{i}^{item}$ obtained by random initialization to get better item representation. Here, the way of summing is utilized for the same type of representation aggregation learned from different aspects, and can effectively avoid dimension expansion in later computing. The combined item representation $\mathbf{e}_{i\leftarrow i}^{item}$ is obtained by:

$$\mathbf{e}_{i\leftarrow i}^{item} = \mathbf{e}_{i}^{item} + \mathbf{e}_{i\leftarrow cp}^{item} + \mathbf{e}_{i\leftarrow co}^{item} \tag{7}$$

2) Bundle Representation Learning: Inspired by previous work [9], [11], we notice that a bundle's feature can be enriched by its constituent items. However, such methods treat each constituent item independently when aggregating them into bundle representation. They neglect the fact that each constituent item should be attached to different importance, especially when appearing with other different items. The self-attention mechanism draws global dependencies among items without regard to superficial similarity. Therefore, we utilize a self-attention mechanism to better capture the dependencies among items:

$$\mathbf{M}_{\mu} = \mathrm{softmax}\left(\frac{(\mathbf{F}\mathbf{W}^{Q})(\mathbf{F}\mathbf{W}^{K})^{T}}{\sqrt{d}}\right) \tag{8}$$

where $\mathbf{F} \in R^{m\times d}$ is the item representation matrix composed of item representation learned in Eq.(7), $\mathbf{W}^{Q}$ and $\mathbf{W}^{K}$ are projection matrices, and d is the dimension.

After that, we can get the optimized item representation $\mathbf{e}_{(i)}^{item}$ by:

$$\mathbf{e}_{(i)}^{item} = \mathbf{e}_{i\leftarrow i}^{item} + \sum_{j\in I} \mu_{ij}\mathbf{e}_{j\leftarrow j}^{item} \tag{9}$$

where I is the item set and μ_{ij} is the corresponding attention score in $\mathbf{M}_{\mu}$.

We utilize the more meaningful item representation $\mathbf{e}_{(i)}^{item}$ to better learn the bundle feature $\mathbf{e}_{b\leftarrow i}^{bundle}$:

$$\mathbf{e}_{b\leftarrow i}^{bundle} = \sigma\left(\sum_{i\in Q(b)} \frac{1}{|Q(b)|}\mathbf{e}_{(i)}^{item}\right) \tag{10}$$

where $Q(b)$ is the item set contained in the bundle b, and σ is ReLU.

Furthermore, to get a more comprehensive bundle representation $\mathbf{e}_{(b)}^{bundle}$, we unify the original bundle representation $\mathbf{e}_{b}^{bundle}$ and item-specific bundle representation $\mathbf{e}_{b\leftarrow i}^{bundle}$ by a sum operation:

$$\mathbf{e}_{(b)}^{bundle} = \mathbf{e}_{b}^{bundle} + \mathbf{e}_{b\leftarrow i}^{bundle} \tag{11}$$

3) User Representation Learning: The representation of a user is often associated with his/her interacting items and bundles. Therefore, it is advisable to derive user representations from historical interactions:

$$\mathbf{e}_{u\leftarrow i}^{user} = \sigma\left(\sum_{i\in S(u)} \frac{1}{|S(u)|}\mathbf{e}_{(i)}^{item}\right) \tag{12}$$

$$\mathbf{e}_{u\leftarrow b}^{user} = \sigma\left(\sum_{b\in T(u)} \frac{1}{|T(u)|}\mathbf{e}_{(b)}^{bundle}\right) \tag{13}$$

where σ is ReLU, $S(u)$ is the item set interacted with user u, and $T(u)$ is the bundle set interacted with user u.

To obtain the comprehensive representation for each user, we employ a sum aggregation to combine original user representation $\mathbf{e}_{u}^{user}$, item-specific user representation $\mathbf{e}_{u\leftarrow i}^{user}$, and bundle-specific user representation $\mathbf{e}_{u\leftarrow b}^{user}$. Formally, the representation of user u is calculated by:

$$\mathbf{e}_{(u)}^{user} = \mathbf{e}_{u}^{user} + \mathbf{e}_{u\leftarrow i}^{user} + \mathbf{e}_{u\leftarrow b}^{user} \tag{14}$$

Hierarchically learning in this section, we obtain the embedding matrix of l users as $\mathbf{X} \in R^{l\times d}$, m items as $\mathbf{Y} \in R^{m\times d}$, and n bundles as $\mathbf{Z} \in R^{n\times d}$, where d denotes the dimension.

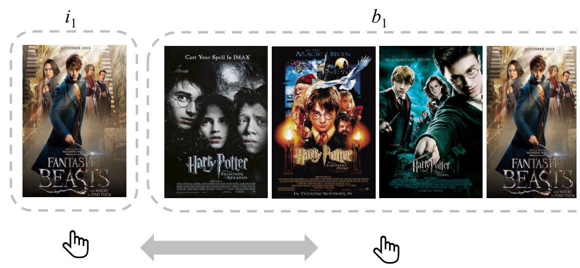


Fig. 3. An example of mutual influence between user preference for items and bundles.

B. Node Propagation Based on Preference Transfer

In a graph, it is desirable to learn node representation by distilling useful information from neighbors and refine the representation by stacking layers. Inspired by GCN [29], we conduct the convolutional operation on an interaction graph and get users' preference and item/bundle feature by node propagation. To better capture the mutual influence between user preferences for items and bundles, we design a user preference transfer paradigm in node propagation procedure between layers, which is the main difference between our proposed node propagation and GCN. We define the above network structure as NPPT (**N**ode **P**ropagation based on **P**reference **T**ransfer). Then, we conduct NPPT on both user-item interaction graph G_{UI} and user-bundle interaction graph G_{UB} simultaneously. The embedding matrix $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ are fed into NPPT. By aggregating information from neighbors in the $(l-1)th$ layer of NPPT:

$$\mathbf{H}_{ui}{}^{(l)} = \mathrm{NPPT}(\mathbf{A}_{ui}, \mathbf{X}^{l-1}, \mathbf{Y}^{l-1}) \tag{15}$$

$$\mathbf{H}_{ub}{}^{(l)} = \mathrm{NPPT}(\mathbf{A}_{ub}, \mathbf{X}^{l-1}, \mathbf{Z}^{l-1}) \tag{16}$$

where $\mathbf{A}_{ui}$, $\mathbf{A}_{ub}$ are adjacency matrices of G_{UI}, G_{UB}.

Then, we split $\mathbf{H}_{ui}{}^{(l)}$ to get the embedding matrix of users and items, which are denoted as $\mathbf{X}_{ui}^{l} \in R^{l\times k}$ and $\mathbf{Y}_{ui}^{l} \in R^{m\times k}$. Similarly, we split $\mathbf{H}_{ub}{}^{(l)}$ to get the embedding matrix of users and bundles, which are denoted as $\mathbf{X}_{ub}^{l} \in R^{l\times k}$ and $\mathbf{Z}_{ub}^{l} \in R^{n\times k}$. Here, k is the dimension of lth layer.

Generally, user preferences for items and bundles cannot be treated independently. As the example in Figure 3, a user who likes Fantastic Beasts and Where to Find Them may also like the series of Harry Potter. Similarly, a user who likes the series of Harry Potter may also like Fantastic Beasts and Where to Find Them. This is because both of them belong to the same bundle. Therefore, it is desirable for us to model the user preference transfer based on the composition relationships between bundles and items. We design a preference transfer paradigm in node propagation procedure to better capture the mutual influence between user preferences for items and bundles.

We combine user preferences $\mathbf{X}_{ub}^{l}$ and $\mathbf{X}_{ui}^{l}$ by an attention network to learn their corresponding importance. Taking user u as an example:

$$w_{ub} = \mathbf{v}^{T}\tanh(\mathbf{W}_{ub}\mathbf{e}_{u\_ub}^{l} + b_{ub}) \tag{17}$$

$$w_{ui} = \mathbf{v}^T \tanh(\mathbf{W}_{ui}\mathbf{e}^l_{u\_ui} + b_{ui}) \tag{18}$$

where $\mathbf{v}^T \in R^d$ is a shared attention vector, $\mathbf{W}_{ub}$ and $\mathbf{W}_{ui}$ are trainable weight matrix, and b_{ub}, b_{ui} are bias. $\mathbf{e}^l_{u\_ub}$ is the representation of user u in $\mathbf{X}^l_{ub}$ and $\mathbf{e}^l_{u\_ui}$ is the representation of user u in $\mathbf{X}^l_{ui}$.

Then, the weights can be normalized by $\mathrm{softmax}$ function:

$$\overline{w_{ub}} = \frac{\exp(w_{ub})}{\exp(w_{ui}) + \exp(w_{ub})} \tag{19}$$

$$\overline{w_{ui}} = \frac{\exp(w_{ui})}{\exp(w_{ui}) + \exp(w_{ub})} \tag{20}$$

The representation of user u can be calculated by:

$$\mathbf{e}^l_u = \overline{w_{ub}}\mathbf{e}^l_{u\_ub} + \overline{w_{ub}}\mathbf{e}^l_{u\_ui} \tag{21}$$

For simplicity, we define this attention network as:

$$\mathbf{X}^l = Attention(\mathbf{X}^l_{ub}, \mathbf{X}^l_{ui}) \tag{22}$$

where $\mathbf{X}^l$ is the user representation.

In light of bundles' composition information, we aggregate item representation learned from G_{UI} to obtain the bundle representation $\mathbf{Z}^l_{ui}$:

$$\mathbf{Z}^l_{ui} = \sigma(\mathbf{W}^l_{ui}\mathbf{A}_{bi}\mathbf{Y}^l_{ui} + b^l_{ui}) \tag{23}$$

where $\mathbf{W}^l_{ui}$ is the trainable weight matrix, b^l_{ui} is the bias, $\mathbf{A}_{bi}$ is the adjacency matrix of G_{BI}, and σ is the activation function ReLU.

Next, we can get the bundle representation $\mathbf{Z}^l$ based on the attention network:

$$\mathbf{Z}^l = Attention(\mathbf{Z}^l_{ub}, \mathbf{Z}^l_{ui}) \tag{24}$$

Similarly, each bundle has an impact on its constituent item, and we obtain the aggregated item representation $\mathbf{Y}^l_{ub}$ by:

$$\mathbf{Y}^l_{ub} = \sigma(\mathbf{W}^l_{ub}\mathbf{A}^T_{bi}\mathbf{Z}^l_{ub} + b^l_{ub}) \tag{25}$$

where $\mathbf{W}^l_{ub}$ is the trainable weight matrix, b^l_{ub} is the bias, and σ is the activation function ReLU.

Next, we combine item representations by an attention network:

$$\mathbf{Y}^l = Attention(\mathbf{Y}^l_{ub}, \mathbf{Y}^l_{ui}) \tag{26}$$

Finally, we feed $\mathbf{X}^l$, $\mathbf{Z}^l$, and $\mathbf{Y}^l$ into the next layer learning:

$$\mathbf{H}_{ub}{}^{(l+1)} = \mathrm{NPPT}(\mathbf{A}_{ub}, \mathbf{X}^l, \mathbf{Z}^l) \tag{27}$$

$$\mathbf{H}_{ui}{}^{(l+1)} = \mathrm{NPPT}(\mathbf{A}_{ui}, \mathbf{X}^l, \mathbf{Y}^l) \tag{28}$$

C. Model Prediction and Optimization

1) Interaction Prediction: In the recommendation task, the MF-based model is a common practice that simply performs interaction learning by using an inner product [13]. However, this method is limited to the modeling of linear relationships only and is not able to model non-linear relationships. To tackle the problem, we concatenate user representation, bundle representation, and their element-wise product and capture their high-order interactions by feeding the concatenated representation into a multi-layer neural network [14]:

$$\mathbf{e}_{cat} = \mathbf{e}^{(u)} \parallel \mathbf{e}^{(b)} \parallel \mathbf{e}^{(u)} \odot \mathbf{e}^{(b)} \tag{29}$$

where $\parallel$ denotes the concatenate operation and $\odot$ is the element-wise product. $\mathbf{e}^{(u)}$ and $\mathbf{e}^{(b)}$ are representation of user u and bundle b learned by Section IV-B.

Next, we feed the combined representation $\mathbf{e}_{cat}$ into a multi-layer neural network. Correspondingly, the ultimate predicted score $\widehat{y}$ is given by:

$$\hat{y} = \varphi(\mathbf{W}^T\mathbf{e}_{cat} + b) \tag{30}$$

where $\mathbf{W}^T$ is the trainable weight matrix, b is the bias, and $\varphi(x) = 1/(1 + e^{-x})$ is the sigmoid function.

2) Model Optimization: In this work, the recommendation problem can be identified as a link prediction task [31]. Precisely, the predicted score indicates the probability of the existence of edges between users and bundles. Supposing that a user has interacted with a bundle, the ground-truth is set to 1, otherwise, set to 0. We utilize the cross-entropy loss function to minimize the loss between ground-truth and predicted score:

$$\mathcal{L} = -y \log \widehat{y} - (1 - y) \log(1 - \widehat{y}) \tag{31}$$

where $\widehat{y}$ is the predicted score between users and bundles. y is the ground-truth that contains observed interactions between users and bundles and unobserved interactions generated by negative sampling.

TABLE I
STATISTICS OF DATASETS

Dataset	Youshu	Steam
User	8,039	29,634
Item	32,770	2,819
Bundle	4,771	615
U-I	138,515	902,967
U-B	51,377	87,565
B-I	176,667	3,541

V. EXPERIMENTS

In this section, we conduct extensive experiments to answer the following questions:

- RQ1 Can our proposed model provide better performance compared with the state-of-the-art models?
- RQ2 How is the effectiveness of intention-oriented hierarchical representation learning?
- RQ3 Can the user preference transfer paradigm improve the bundle recommendation performance?
- RQ4 How do parameters affect the model performance?

TABLE II
PERFORMANCE COMPARISON(%)

Datasets	Youshu						Steam					
Models	Recall@5	MRR@5	NDCG@5	Recall@10	MRR@10	NDCG@10	Recall@5	MRR@5	NDCG@5	Recall@10	MRR@10	NDCG@10
BPR	53.40	35.61	40.04	66.62	38.10	44.90	97.06	74.88	80.50	99.09	73.04	79.59
NCF	58.64	38.96	43.86	77.18	40.61	47.90	97.31	73.31	79.40	99.31	72.76	79.45
SASRec	56.45	34.52	39.96	73.35	37.47	45.98	96.42	67.90	75.07	99.37	68.39	76.13
DAM	60.41	39.98	45.06	73.34	42.82	50.11	97.36	73.80	79.77	99.52	73.41	79.99
GCN-B	63.12	43.83	48.64	75.14	46.08	53.03	97.59	74.51	80.37	99.53	74.50	80.80
GCN-T	61.75	42.15	47.02	74.17	43.74	50.99	97.50	72.35	78.73	99.47	73.24	79.84
BGCN	65.34	44.56	49.73	76.62	45.97	53.30	97.60	74.65	80.47	99.53	74.66	80.92
IHBR	**68.81**	**48.01**	**53.19**	**79.04**	**49.11**	**56.27**	**98.22**	**78.54**	**83.55**	**99.65**	**78.83**	**84.09**

A. Datasets

Experiments are conducted on two public datasets Youshu [9] and Steam [32]. Youshu are collected from a Chinese book review site Youshu[1], which provides the booklists released by users. Steam are crawled from the Steam video game distribution platform[2], which provides bundled products (e.g., games, software, DLC). The statistics of datasets are presented in Table I, which contains users, items, bundles, user-item interactions (U-I), user-bundle interactions (U-B), and bundle-item interactions (B-I).

B. Baselines

- BPR [10]: BPR is a basic pair-wise ranking algorithm under the framework of matrix factorization.
- NCF [14]: NCF is a neural CF-based model, which leverages a multi-layer perceptron to learn the user-bundle interaction function.
- SASRec [33]: SASRec is a sequential model, which captures long-term semantics and makes its predictions to the next item based on relatively few actions.
- DAM [9]: DAM designs a factorized attention network for item feature aggregation and jointly models user-bundle interactions and user-item interactions in a multi-task manner.
- GCN-B: GCN-B is a variant of GCN [29], this method conducts a convolutional operation on a user-bundle interaction graph to learn the user representation and bundle representation.
- GCN-T: GCN-T is also a variant of GCN [29], such method conducts a convolutional operation on a user-item-bundle interaction graph to learn the user representation, bundle representation, and item representation.
- BGCN [11]: BGCN conducts convolutional operations on a user-bundle interaction graph and a user-item interaction graph separately, and aggregates user representation, bundle representation from such two aspects.

[1]https://www.yousuu.com
[2]https://store.steampowered.com

C. Evaluation Metrics

We conduct the experiments under the top-K recommendation and set top-K ($K \in \{5, 10\}$). We adopt the $leave$-one-out evaluation in our experiments, which has been widely used in previous studies [9], [12]. For each user, we select one of his/her interactions with bundles for testing and the remaining interactions for training. It is time-consuming for the user to rank all bundles, we randomly choose 99 bundles that never interacted with the user before from existing bundles as negative samples of the selected test bundle. Furthermore, to evaluate the performance of our IHBR, we employ several widely used ranking metrics [9], [12]: Recall, Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG). The Recall@K describes the proportion of the hits presenting within the top-K recommendation list. The MRR@K considers the average rank positions of the hits within the top-K recommendation list. The NDCG@K measures the position of the hit by assigning a higher score to the hit at a higher rank within the top-K recommendation list.

D. Implementation

Our model is implemented in PyTorch with Adam optimizer [34]. The number of layers is fixed to 2, which is optimal in most cases [19]. The batch size is set to 4096. The dropout rate and embedding size are searched in [0, 0.3, 0.5], and [16, 32, 64, 128], respectively.

E. Results and Analysis

1) Performance Comparison (RQ1): Table II shows the performance of all approaches with respect to the number of recommended items (i.e., K) on datasets Youshu and Steam. We report the best performance of compared methods. Firstly, we can observe that our proposed model achieves the best performance on both datasets. The relative improvements over the best baseline by 5.31%, 7.74%, 6.96%, and 0.64%, 5.21%, 3.83% for Recall@5, MRR@5, NDCG@5 on two datasets. When K is set to 10, the proposed model achieves 3.16%, 6.83%, 5.57%, and 0.12%, 5.59%, 3.92% performance improvements against the best baseline. General recommendation solutions, such as BPR and NCF, can not perform well in the bundle recommendation. Sequential-based recommendation solutions such as SASRec can not achieve better performance compared with bundle recommendation models, because there are not sequential relations between bundles. Models based on the neural network, such as NCF and DAM, outperform traditional method BPR for their strong power in generalization. Models based on the graph convolutional network achieve higher performance for their capability of extracting structural information. Although GCN-T extends GCN-B to a tripartite interaction graph, it performs worse than GCN-B. The reason is that GCN-T treats different types of nodes equally. Among all baselines, BGCN performs best because it considers user preferences for both items and bundles as well as bundles' composition information. However, BGCN still fails to explore the relationships within items based on the analysis of intentions behind users' purchase behaviors and ignores mutual influence between user preferences for items and bundles. As for IHBR, it explores the co-purchase and co-occurrence patterns among items and learns a more comprehensive item representation from their related items. After that, we utilize a hierarchical architecture for learning bundle representation and user representation. Furthermore, we investigate the user preference transfer from historical interactions with items and bundles, and hold the view that the user preference transfer between items and bundles should originate from the associations between items and bundles. Consequently, IHBR shows its superiority in bundle recommendation.

TABLE III
PERFORMANCE ON INTENTION-ORIENTED HIERARCHICAL REPRESENTATION LEARNING(%)

Dataset	Youshu		
Metrics	Recall@5	MRR@5	NDCG@5
noHi	65.02	45.09	50.07
noUser	66.28	46.25	51.24
noSatt	66.88	47.22	52.12
noItem	67.22	47.25	52.24
IHBR	**68.81**	**48.01**	**53.19**
Dataset	Steam		
Metrics	Recall@5	MRR@5	NDCG@5
noHi	97.46	73.24	79.38
noUser	97.60	75.17	80.87
noSatt	97.63	77.24	82.43
noItem	97.70	78.05	83.04
IHBR	**98.22**	**78.54**	**83.55**

2) Effectiveness of Intention-oriented Hierarchical Representation Learning (RQ2): Intention-oriented hierarchical representation learning is an important component in our model. Therefore, to investigate its effectiveness, we compare it with other variants, including a model without intention-oriented hierarchical representation learning named noHi, a model without user representation learning in intention-oriented hierarchical representation learning named noUser, a model without self-attention in bundle representation learning named noSatt, and a model without item representation learning in intention-oriented hierarchical representation learning named noItem. Table III depicts the recommendation performance of the four variants in comparison to our proposed model, which demonstrates that our model achieves the best performance in all cases. We also observe that: 1) Compared with noHi, our proposed model achieves better performance, which proves the effectiveness of intention-oriented hierarchical representation learning. 2) Compared with noUser, noSatt, and noItem, our proposed model shows its superiority, which proves that each level of representation learning (i.e., user, bundle, item) plays an indispensable role in intention-oriented hierarchical representation learning. 3) Compared with noSatt and noItem, our proposed model performs best. This may attribute to the fact that our model has utilized the co-purchase and co-occurrence relationships among items and explored global dependencies among items by self-attention mechanism.

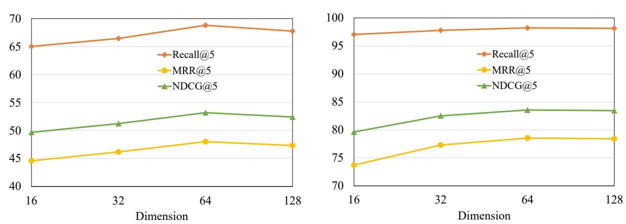
TABLE IV
PERFORMANCE ON NODE PROPAGATION BASED ON PREFERENCE TRANSFER(%)

Dataset	Youshu		
Metrics	Recall@5	MRR@5	NDCG@5
noPref	66.05	45.98	50.98
UI+UB	62.43	42.30	47.31
UIB	66.90	46.71	51.75
IHBR	**68.81**	**48.01**	**53.19**
Dataset	Steam		
Metrics	Recall@5	MRR@5	NDCG@5
noPref	97.88	77.89	83.00
UI+UB	97.49	72.24	78.67
UIB	97.82	77.89	82.97
IHBR	**98.22**	**78.54**	**83.55**

3) Effectiveness of Node Propagation Based on Preference Transfer (RQ3): To justify the contribution of node propagation based on preference transfer, we compare our model with several variants. noPref is the model without node propagation based on preference transfer. UI+UB refers to a model that conducts node propagation operations on a user-item interaction graph and a user-bundle interaction graph separately. And UIB is a model that conducts node propagation operation on a user-item-bundle tripartite interaction graph. The experimental results are shown in Table IV. Apparently, our proposed model with node propagation based on preference transfer achieves the best performance in all cases, which indicates that user preferences for both items and bundles can be mutually reinforced for bundle recommendation task. We also observe that: 1) Compared with noPref, the experimental results demonstrate the effectiveness of node propagation based on preference transfer in our model.

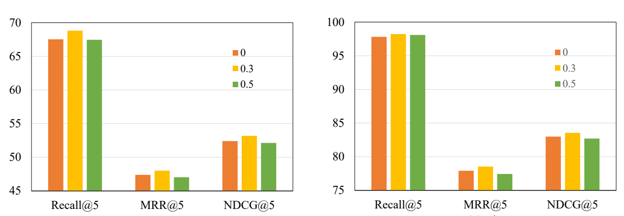
2) UI+UB achieves the poorest performance in comparison to other models. The reason is that it models user interactions with items and bundles separately and brings the problem of over-parameterization. 3) When in dataset Youshu, UIB can achieve better performance than UI+UB, but performs still worse than our proposed model. The reason is that UIB considers preference transfer under the framework of a user-item-bundle tripartite interaction graph, but treats different types of nodes equally. In dataset Steam, UIB cannot bring any improvements compared with noPref because of the tremendous imbalance among the U-I, U-B, and B-I. As for our proposed model, it can effectively overcome the above limitations and achieves the best performance on both Youshu and Steam.

4) Analysis of Parameters (RQ4): To validate parameter sensitivity, we first set the embedding dimension (i.e., d) to 16, 32, 64, and 128. Seen from Figure 4, the performance of our proposed model shows an increasing tendency when setting d from 16 to 64 because the model can retain more useful information during the increase of dimension. When d is set to 128, the performance of the model decreases slightly for overfitting. Then, we vary the dropout rate from 0 to 0.5 in Figure 5. Our model with 0.3 performs best because it can effectively alleviate over-parameterized issues. When the dropout rate is set to 0.5, its performance drops on a small scale for neglecting some significant information.



(a) Youshu (b) Steam

Fig. 4. Performance on embedding dimension(%)



(a) Youshu (b) Steam

Fig. 5. Performance on dropout rate(%)

VI. Conclusions and Future Work

In this paper, we propose a novel model IHBR to address the problem of bundle recommendation. To effectively investigate the intentions behind users' interactions with a bundle, both co-purchase and co-occurrence relationships among items are explored to obtain a better item representation from their related items. Furthermore, a hierarchical architecture is deployed for feature aggregation. To be more specific, we employ a self-attention mechanism for capturing item dependency, then update the representation from bottom to top. Finally, in order to catch the mutual influence between user preferences for items and bundles, we devise a preference transfer paradigm in convolutional learning based on both user-item and user-bundle interaction graphs. The experimental results on two real-world datasets demonstrate the superiority of the proposed model. In the future, we can tackle the problem in a heterogeneous graph, where each type of entity is associated with different traits. Apart from this, the meta-path can be taken into account for capturing contextual information.

VII. Acknowledgment

This work was supported by the Major Program of the Natural Science Foundation of Jiangsu Higher Education Institutions of China under Grant Nos.19KJA610002 and 19KJB520050, and the National Natural Science Foundation of China under Grant No.61902270.

References

[1] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[3] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–45, 2014.

[4] T. Di Noia, R. Mirizzi, V. C. Ostuni, D. Romito, and M. Zanker, "Linked open data to support content-based recommender systems," in *I-SEMANTICS*, 2012, pp. 1—-8.

[5] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[6] T. Zhu, P. Harrington, J. Li, and L. Tang, "Bundle recommendation in ecommerce," in *SIGIR*, 2014, pp. 657–666.

[7] M. Li, X. Bao, L. Chang, Z. Xu, and L. Li, "A survey of researches on personalized bundle recommendation techniques," in *ML4CS*, 2020, pp. 290–304.

[8] R. Qiu, Z. Huang, J. Li, and H. Yin, "Exploiting cross-session information for session-based recommendation with graph neural networks," *ACM Transactions on Information Systems (TOIS)*, vol. 38, no. 3, pp. 1–23, 2020.

[9] L. Chen, Y. Liu, X. He, L. Gao, and Z. Zheng, "Matching user with item set: Collaborative bundle recommendation with deep attention network." in *IJCAI*, 2019, pp. 2095–2101.

[10] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009, pp. 452–461.

[11] J. Chang, C. Gao, X. He, D. Jin, and Y. Li, "Bundle recommendation with graph convolutional networks," in *SIGIR*, 2020, pp. 1673–1676.

[12] Q. Deng, K. Wang, M. Zhao, Z. Zou, R. Wu, J. Tao, C. Fan, and L. Chen, "Personalized bundle recommendation in online games," in *CIKM*, 2020, pp. 2381–2388.

[13] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *NIPS*, 2008, pp. 1257–1264.

[14] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*, 2017, pp. 173–182.

[15] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 833–852, 2019.

[16] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *KDD*, 2018, pp. 974–983.
[17] Y. Liu, M. Xie, and L. V. Lakshmanan, "Recommending user generated item lists," in *RecSys*, 2014, pp. 185–192.
[18] D. Cao, L. Nie, X. He, X. Wei, S. Zhu, and T.-S. Chua, "Embedding factorization models for jointly recommending items and user generated lists," in *SIGIR*, 2017, pp. 585–594.
[19] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, 2017, pp. 1024–1034.
[20] J. Bai, C. Zhou, J. Song, X. Qu, W. An, Z. Li, and J. Gao, "Personalized bundle list recommendation," in *WWW*, 2019, pp. 60—-71.
[21] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010, pp. 811—-820.
[22] Z. Liu, M. Wan, S. Guo, K. Achan, and P. S. Yu, "Basconv: Aggregating heterogeneous interactions for basket recommendation with graph convolutional neural network," in *SDM*, 2020, pp. 64–72.
[23] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *SIGIR*, 2015, pp. 403—-412.
[24] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *CIKM*, 2017, pp. 1419—1428.
[25] Y. He, Y. Zhang, W. Liu, and J. Caverlee, "Consistency-aware recommendation for user-generated item list continuation," in *WSDM*, 2020, pp. 250–258.
[26] J. Hao, T. Zhao, J. Li, X. L. Dong, C. Faloutsos, Y. Sun, and W. Wang, "P-companion: A principled framework for diversified complementary product recommendation," in *CIKM*, 2020, pp. 2517—-2524.
[27] P. Hu, R. Du, Y. Hu, and N. Li, "Hybrid item-item recommendation via semi-parametric embedding," in *IJCAI*, 2019, pp. 2521–2527.
[28] D.-T. Le, H. W. Lauw, and Y. Fang, "Correlation-sensitive next-basket recommendation," in *IJCAI*, 2019, pp. 2808–2814.
[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017, pp. 1–14.
[30] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *AISTATS*, 2011, pp. 315–323.
[31] X. Li and H. Chen, "Recommendation as link prediction: a graph kernel-based machine learning approach," in *JCDL*, 2009, pp. 213–216.
[32] A. Pathak, K. Gupta, and J. McAuley, "Generating and personalizing bundle recommendations on steam," in *SIGIR*, 2017, pp. 1073–1076.
[33] W. Kang and J. McAuley, "Self-attentive sequential recommendation," in *ICDM*, 2018, pp. 197–206.
[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015, pp. 1–15.