electronics

MDPI

Article

RETRACTED: A Global Structural Hypergraph Convolutional Model for Bundle Recommendation

Xingtong Liu * and Man Yuan *

School of Computer & Information Technology, Northeast Petroleum University, Fazhan Road, Daqing 163318, China

* Correspondence: lxtdyx2022@163.com (X.L.); yuanman@nepu.edu.cn (M.Y.)

Abstract: Bundle recommendations provide personalized suggestions to users by combining related items into bundles, aiming to enhance users' shopping experiences and boost merchants' sales revenue. Existing solutions based on graph neural networks (GNN) face several significant challenges: (1) it is demanding to explicitly model multiple complex associations using standard graph neural networks, (2) numerous additional nodes and edges are introduced to approximate higher-order associations, and (3) the user–bundle historical interaction data are highly sparse. In this work, we propose a global structural hypergraph convolutional model for bundle recommendation (SHCBR) to address the above problems. Specifically, we jointly incorporate multiple complex interactions between users, items, and bundles into a relational hypergraph without introducing additional nodes and edges. The hypergraph structure inherently incorporates higher-order associations, thereby alleviating the training burden on neural networks and the dilemma of scarce data effectively. In addition, we design a special matrix propagation rule that captures non-pairwise complex relationships between entities. Using item nodes as links, structural hypergraph convolutional networks learn representations of users and bundles on a relational hypergraph. Experiments conducted on two real-world datasets demonstrate that the SHCBR outperforms the state-of-the-art baselines by 11.07–25.66% on Recall and 16.81–33.53% on NDCG. Experimental results further indicate that the approach based on hypergraphs can offer new insights for addressing bundle recommendation challenges. The codes and datasets have been publicly released on GitHub.

Keywords: recommender systems; bundle recommendation; hypergraph neural networks; hypergraph

Citation: Liu, X.; Yuan, M. RETRACTED: A Global Structural Hypergraph Convolutional Model for Bundle Recommendation. *Electronics* **2023**, *12*, 3952. https://doi.org/10.3390/electronics12183952

Academic Editors: Jesús Bobadilla, David Martín Gómez, Jose M. Alcaraz Calero and Ping-Feng Pai

Received: 2 August 2023
Revised: 10 September 2023
Accepted: 13 September 2023
Published: 15 September 2023
Retracted: 22 March 2024

1. Introduction

Recommender systems have emerged as a crucial tool in the e-commerce industry, contributing significantly to improving user experiences and driving product sales with their development [1,2]. Traditional recommender systems primarily emphasize individual item recommendations, which could not meet the growing personalized needs of users. To further enhance user satisfaction, the bundle recommendation as a marketing strategy has been proposed. Based on users' purchasing behavior and the relevance of items, the bundle recommendation combines relevant items into bundles, such as music playlists [3], game bundles [4,5], and drug packages [6]. Recommending bundles containing related products not only provides users with more advantageous personalized combinations but also aids businesses in achieving profitability [7].

The essence of recommender systems lies in utilizing historical interaction information and similarity relationships between entities to predict the likelihood of interactions between users and items [8]. Traditional recommender systems only involve two types of entities: users and items. However, bundle recommendation tasks encompass three types of entities: users, items, and bundles. Therefore, the decision-making process for users to choose bundles will be more complex compared to selecting individual items. Specifically,

This paper has been retracted. A retraction notice was published on 22 March 2024 in *Electronics* **2024**, *13*, 1165. https://doi.org/10.3390/electronics13071165

users will simultaneously consider multiple items in the bundle and their combination discounts. Sometimes, even if users like all the items contained, they may not prefer the bundle because it is not a satisfactory well-matched combination. Only when users are highly satisfied with the combination of items within this bundle will they choose the bundle for consumption. Therefore, user–bundle interactions are usually sparser than user–item interactions. Moreover, the user–item-bundle relationship is much more complex than the pairwise relationship. For example, a user can interact with multiple items or bundles, and an item can exist in multiple bundles. In this case, the affinity relations are no longer dyadic (pairwise) but rather an integration of multiple binary relationships between users, items, and bundles (three types of entities). Accordingly, the aforementioned issues make bundle recommendation tasks highly challenging.

Over the past few years, graph neural networks (GNNs) [9] have become one of the research hotspots. GNNs are a type of neural network designed for processing and learning from graph-structured data. The core idea behind GNNs [10,11] is to perform message passing between neighboring nodes in the graph to learn node representations based on their local neighborhood information. This allows GNNs to capture complex and non-linear relationships between nodes in the graph. Due to their effective modeling of graph-structured data, GNNs have emerged as a cutting-edge algorithm in the field of recommender systems, finding wide applications. For example, GNNs have been applied in item recommendation [12,13], session-based recommendation [14,15], social recommendation [16,17], and bundle recommendation [18,19].

Despite the success achieved by existing solutions based on GNNs in bundle recommendation tasks, there are still some significant challenges that need to be addressed. Specifically, due to the interactions between users and items/bundles, as well as the affiliation relationships between items and bundles, it is difficult to explicitly model the multiple complex associations among three types of entities using standard graph neural networks. Furthermore, traditional graphs are limited in the sense that each edge can only connect two nodes. Consequently, a substantial number of additional nodes and edges need to be introduced to approximate higher-order associations between entities, which increases the training burden on neural networks. Finally, recommender systems generally encounter the challenge of sparse user–item interaction history data. Since bundles encompass multiple items, the user–bundle interaction history data is inherently scarcer than the user–item interaction history data. Therefore, the issue of data sparsity becomes more pronounced in bundle recommendation scenarios. The highly sparse historical interaction data make learning with neural networks difficult and unstable. This also makes it difficult for bundle recommendation models to accurately model user preferences.

Hypergraphs [20] provide a natural solution to address the above limitations. In comparison to traditional graphs, hypergraphs represent a more flexible type of graph structure. Figure 1 illustrates the differences between traditional graph structures and hypergraph structures. In traditional graphs, an edge can only connect two nodes, while a hyperedge in hypergraphs can connect multiple nodes simultaneously. Therefore, hypergraphs can more flexibly represent and capture complex relationships in real applications. Compared to traditional graph structures, hypergraph structures possess more intricate topologies and enhanced expressive capabilities. Consequently, some problems could be easier to solve with the more accurate representation provided by hypergraphs. For bundle recommendation tasks, higher-order complex associations exist among users, items, and bundles. Hypergraphs inherently possess advantages in handling such data.

In this work, we propose a novel global structural hypergraph convolutional model for bundle recommendation (SHCBR), which jointly incorporates multiple complex interactions between users, items, and bundles into a relational hypergraph. We directly connect user nodes, item nodes, and bundle nodes (three types of nodes) with a hyperedge, which explicitly models complex associations among the three types of entities without introducing additional nodes and edges. Further, we design a special matrix propagation rule that uses items as links to aggregate and update user embeddings and bundle

embeddings on relational hypergraphs. We introduce the Laplacian matrix derivation in the matrix propagation rule, which can help one to better understand the evolution process of structural hypergraph convolutional neural networks. Additionally, we design a personalized weight operation to improve the accuracy of the final recommendation results. Meanwhile, inspired by LightGCN [13], we simplify the original hypergraph convolution by removing feature transformations and non-linear activation functions, m[illegible]more suitable for bundle recommendation scenarios. Hypergraph convolution can lear[illegible]he hidden layer representation considering the high-order information thereby acquiri[illegible] more meaningful representations of users and bundles. The SHCBR fu[illegible]er integrates th acquired representations to produce recommendations based on score ra[illegible]ngs.

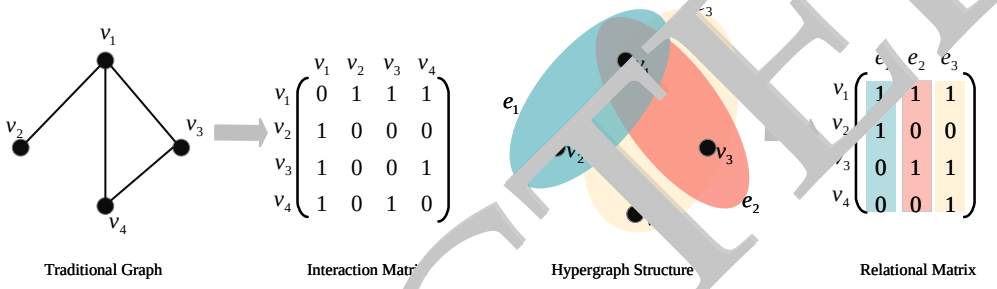

Figure 1. The differences between a tr[illegible]ditional graph and a hypergraph. In a traditional graph, every edge, represented as a line, solely con[illegible]cts two nodes. Conv[illegible]rsely, in a hypergraph, every hyperedge, denoted by a colored ellipse, can conn[illegible]more than two [illegible]odes simultaneously.

To summarize, th[illegible]rimary contrib[illegible]s of this study can be outlined as follows:

- We propose a nov[illegible]el mo[illegible]named the SHCBR, which introduces the hypergraph structure to explicit[illegible]y mo[illegible]el co[illegible]ex relationships between entities in bundle recommendation tasks. We d[illegible]irectly co[illegible]nect three types of nodes with a hyperedge without intr[illegible]g additiona[illegible] nodes and edges. By constructing a relational hypergraph c[illegible]ontainin[illegible] three types of [illegible]odes, we can explore existing information from a global p[illegible]spectiv[illegible]e. effectively alleviating the dilemma of data sparsity.
  We [illegible]sign a[illegible] matrix propagation rule and personalized weight operation in the pr[illegible]osed structural hypergraph convolutional neural network (SHCNN). Using items as [illegible]ks, we leverage efficient hypergraph convolution to learn node representations [illegible]onsidering the high-order information. Since the relational hypergraph structure naturally incorporates higher-order associations, it is enough to generate [illegible]de representations with one layer of SHCNN, further enhancing model efficiency;
- E[illegible]eriments on two real-world datasets indicate that our SHCBR outperforms existing state-of-the-art baselines by 11.07–25.66% on Recall and 16.81–33.53% on NDCG. The experimental results further validate that hypergraphs provide a novel and effective method to tackle bundle recommendation tasks.

2. Related Work

In this section, we provide a concise review of related works concerning bundle recommendation and hypergraph learning.

2.1. Bundle Recommendation

Despite the extensive research on recommender systems, few efforts have been devoted to addressing the specific challenges of bundle recommendation tasks.

Initially, some works [21,22] modeled package recommendation as a linear knapsack problem [23]. These works used integer programming techniques, which overlooked the pairwise dependencies [24] among similar items. When cross-item dependencies were modeled as hard constraints, the computation became complex, and the system could not automatically recommend best matching results to users based on their preferences. During

the same period, some studies [25,26] utilized association analysis techniques to address bundle recommendation problems. However, association rule methods primarily focused on the relationships between items, neglecting user and item feature information, which made it difficult to achieve personalized recommendations.

With the advancement of recommender systems, methods such as collaborative filtering [3,27], neural networks [28], and topic modeling [29] have been applied to bundle recommendations. For example, the embedding factor model (EFM) [3] extended the traditional factorization method to capture users' preferences. The work [27] combined matrix factorization (MF) methods for collaborative filtering with learned ratio constraints to achieve clothing package recommendation. However, matrix factorization was primarily used to model user–item interactions. It was challenging to directly model higher-order associations, such as relationships between multiple items within bundles. Furthermore, methods based on collaborative filtering often utilized information from neighboring users or items for recommendations. However, they could be affected by noisy data and outliers, which might result in inaccurate recommendations. The DAM [28] model utilized a factorized attention network to learn bundle representations in a multi-task manner. However, neural network-based bundle recommendation methods could be affected by the issue of data sparsity. Neural network methods typically involved numerous parameters and complex model architectures, and sparse training data can lead to overfitting issues. Xiong et al. [29] proposed a personalized travel package recommendation model, which utilized topic analysis to obtain activities of interest to tourists. However, topic modeling methods might not accurately capture the true meaning and attributes of items when performing topic modeling on items. This could result in inaccurate topic representations, consequently affecting the effectiveness of recommendations.

In recent years, researchers have redirected their research focus towards graph neural networks. The BGCN [18] model unified three interactions between three types of entities into a heterogeneous graph. The BGCN model used graph convolutional networks (GCN) [30] to acquire node representations for both users and bundles. The BundleNet [5] model formalized a recommendation task as a link prediction problem on the graph and addressed it by utilizing a neural network model capable of directly learning from graph-structured data. The HMBR [31] model used the graph neural network to extract multi-relation representations from various views. Additionally, an algorithm called BFTC was devised to enhance the precision of bundle representations.

Furthermore, some researchers have incorporated contrastive learning methods into GNN-based models. Contrastive learning can extract more meaningful and discriminative feature representations by learning the relationships between different entities in contrastive views. These feature representations can better capture the similarities and differences between entities, thereby enhancing the accuracy of bundle recommendation. The MIDGN [19] model utilized a graph neural network equipped with the neighbor routing mechanism to untangle the coupling between the user–item and bundle–item graphs, considering users' intents. CrossCBR [32] utilized cross-view contrastive learning to capture cross-view cooperative associations. GPCL [33] innovatively embedded each entity as a Gaussian distribution and introduced a prototypical contrastive learning method to capture more refined representations.

The review of the mentioned works indicates that a significant portion of recent research has utilized GNN to model interactions between users, items, and bundles. GNN-based models possess the advantage of integrating node features and graph topology. However, bundle recommendation involves intricate relationships among users, items, and bundles. Graph neural networks may require the design of appropriate graph structures and connections to capture these intricate relationships. Accurately modeling higher-order associations could present a challenge. On the other hand, graph neural networks exhibit high computational complexity, especially for large-scale graph data. In bundle recommendations, there might be a substantial number of nodes and edges, approximating higher-order associations among the three types of entities. This can lead to an increased

computational burden during training and inference. This brings forth the concept of the hypergraph [20]. Hypergraph is a special graph structure that can utilize hyperedges to simultaneously connect multiple nodes. In response to the challenges arising from the use of GNN, hypergraph neural networks present a natural and relative solution. The flexibility of hypergraphs makes them a novel approach for investigating bundle recommendation tasks.

2.2. Hypergraph Learning

Hypergraphs [20,34] are a specialized type of graph structure that extends traditional graphs. Hypergraphs allow a hyperedge to connect multiple nodes, forming a many-to-many relationship. Therefore, hypergraphs can more accurately represent complex relationships that traditional graphs cannot directly capture in real-world scenarios. For instance, in the knowledge graph [35], hypergraphs can directly represent relationships between one entity and multiple other entities using a single hyperedge. Such higher-order associations are useful for modeling complex semantic associations. In contrast, to approximate such higher-order associations, traditional graphs would require the introduction of a substantial number of nodes and edges.

Hypergraph learning [36] is a deep learning method based on hypergraph structures. In the past few years, hypergraph learning has gained significant attention due to its flexibility and ability to model complex data associations. Hypergraph learning was initially proposed as a label propagation method [37] for semi-supervised learning. Subsequently, Huang et al. [38] employed hypergraph learning for video object segmentation, delving into hypergraph construction methodologies. Weights had a great influence on the modeling of data correlation, and then learning the weights of hyperedges became a new research topic. Gao et al. [39] proposed a novel approach for evaluating the significance of hyperedges or subgraphs within a multi-hypergraph structure. The aim was to assign weightings that reflect their relative importance. Subsequently, the L2 regularization of weights to learn optimal hyperedge weights was proposed by [40]. Hypergraph neural networks (HGNN) have been introduced as a novel method on hypergraph structure, similar to graph neural networks [41]. Hypergraph neural networks employed the hypergraph Laplacian operator to represent the hypergraph from a spectral perspective. Hypergraph neural networks have an advantage over the current GNN methods in their ability to model and encode high-order relations within data. The authors of [42] introduced two end-to-end trainable operators recently, namely, hypergraph convolution and hypergraph attention, which can handle non-pairwise relationships modeled in high-order hypergraphs.

Although research on hypergraph deep learning is still in its early stages, hypergraph neural networks have found widespread application across various fields [35,43–46] due to their exceptional representation capabilities. For example, in computer vision, Wang et al. [43] applied hypergraphs to visual classification tasks, using hypergraphs to describe the relations among visual features. In knowledge graph [35], hypergraph neural networks were utilized to capture higher-level correlations between entities and attributes. In the field of recommender systems, hypergraph-based recommendation models can capture complex interactions between entities and provide more accurate and diverse recommendation results. For example, the UHBR [44] model utilized hypergraphs to model complex user–item-bundle associations, thereby enhancing the efficiency and accuracy of the model. Hg-PDC [45] model proposed an algorithm based on dynamics clustering and similarity measurement in hypergraphs, thereby improving the recommendation performance. The GC–HGNN [46] model utilized the hypergraph convolutional neural network to obtain the global context information.

Hypergraph learning also comes with certain limitations in practical applications. Hypergraph learning is still a relatively new field, and there might not be enough well-established frameworks, libraries, or benchmarks available for researchers and practitioners. This can make it challenging to implement and compare different hypergraph-based models. It is worth noting that the application of hypergraph models may be more suitable for specific domains, whereas their applicability in other domains could be limited.

3. Preliminary

In this section, we briefly formulate the problem and introduce our bundle hypergraph definition.

3.1. Problem Formulation

In our research, we have a defined group of users $U = \{u_1, u_2, \ldots, u_N\}$, where the variable N is the total number of users, a defined group of items $I = \{i_1, i_2, \ldots, i_M\}$, where variable M is the total number of items, a defined group of bundles $B = \{b_1, b_2, \ldots, b_K\}$, and the variable K is the total number of bundles. Each individual bundle $b_s \in B$ is made up of a specific group of items, denoted as $b_s = \{i_{s_1}, i_{s_2}, \ldots, i_{s_{|b_s|}}\}$, with each bundle size being greater than one. In order to gain a deeper understanding of the connections among users, items, and bundles, we present three matrices: user–item matrix $X_{N\times M} = \{x_{ui}|u \in U, i \in I\}$, user–bundle matrix $Y_{N\times K} = \{y_{ub}|u \in U, b \in B\}$, and bundle–item matrix $Z_{K\times M} = \{z_{bi}|b \in B, i \in I\}$. These matrices consist of binary values, where a value of 1 indicates an observed interaction or inclusion. In the user–item matrix, an entry $x_{ui} = 1$ signifies that there is an interaction between user u and item i. Then, an entry $y_{ub} = 1$ in the user–bundle matrix indicates that user u has interacted with bundle b. Similarly, an entry $z_{bi} = 1$ in the bundle–item matrix implies that item i is a component of bundle b. From the definition provided earlier, we can formulate the issue of bundle recommendation in the following manner:

Input: Data on the interaction between users and items $X_{N\times M}$, data on the interaction between users and bundles $Y_{N\times K}$, and data on the dependency relationship between bundles and items $Z_{K\times M}$.

Output: A bundle recommendation model that calculates the likelihood of user u interacting with bundle b.

3.2. Bundle Hypergraph Definition

Hypergraphs provide an approach to obtain intricate higher-order connections. However, traditional hypergraphs struggle to represent multiple relationships in bundle recommendation tasks suitably. In response, we propose a novel data structure named the bundle hypergraph, which can more effectively capture the multiple associations among the three types of entities in bundle recommendations.

A bundle hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a mathematical structure that consists of a set of hyperedges $\mathcal{E}$ and a set of nodes $\mathcal{V}$. The hyperedges are subsets of the nodes set and can contain multiple nodes. This definition allows for a more flexible representation of relationships between nodes compared to traditional graphs. The incidence matrix $H_{ve} \in \{0, 1\}^{|\mathcal{V}|\times|\mathcal{E}|}$ represents the connection relationship between hyperedges and nodes. The entry $h(v, e)$ in the incidence matrix can be defined as:

$$h(v, e) = \begin{cases} 1, \; if \; v \in \mathcal{E}_e \\ 0, \; otherwise \end{cases} \tag{1}$$

Based on the hypergraph definition, node degrees and hyperedge degrees can be calculated. Next, we consider a node $v \in \mathcal{V}$, where $\mathcal{V}$ represents the set of nodes. In the bundle hypergraph, the degree of node v can be denoted as $d_v = \sum_{e\in\mathcal{E}} h(v, e)$, which is calculated by summing up the contributions from all of the connected hyperedges in $\mathcal{E}$ (the set of hyperedges) using the function $h(v, e)$. Similarly, for a hyperedge $e \in \mathcal{E}$, its degree can be expressed as $d_e = \sum_{v\in\mathcal{V}} h(v, e)$. It is determined by adding up the values calculated by applying the function $h(v, e)$ for each node v in $\mathcal{V}$.

4. Method

The construction of our proposed relational hypergraph is presented in Section 4.1. Following that, we elaborate on the structural hypergraph convolutional neural networks in Section 4.2. Moving on to Section 4.3, we introduce the matrix propagation rule in our

research. Lastly, a detailed explanation of the model's prediction and training process is given in Section 4.4.

4.1. Relational Hypergraph Construction

As shown in Figure 2, we construct a relational hypergraph matrix with items as links. To obtain more meaningful bundle representations, we first construct [illegible]s of relation graphs around users and bundles. The graphs that are included in this study are the interaction graph between users and items, the interaction graph between users and bundles, and the affiliation graph between bundles and items. Based on these three relation graphs, we can obtain the three interaction matrices mentioned in Section [illegible], namely, X, Y, and Z. In order to encode the input, we perform one-hot encoding for each node on the hypergraph and compress each node into a dense real-value vector. E_u, E_i, and E_b represent the embeddings of user, item, and bundle in the initial relational representation, respectively. The embedding size d remains consistent for all three matrices E_u, E_i, and E_b. We define the feature vectors of user u, item i, and bundle b in the following manner:

$$e_u = E_u^T v_u, \quad e_i = E_i^T v_i, \quad e_b = E_b^T v_b \tag{2}$$

where $E_u \in \mathbb{R}^{N \times d}$, $E_i \in \mathbb{R}^{M \times d}$, and $E_b \in \mathbb{R}^{K \times d}$. $v_u \in \mathbb{R}^N$, $v_i \in \mathbb{R}^M$, and $v_b \in \mathbb{R}^K$ are one-hot vectors of user, item, and bundle, respectively.

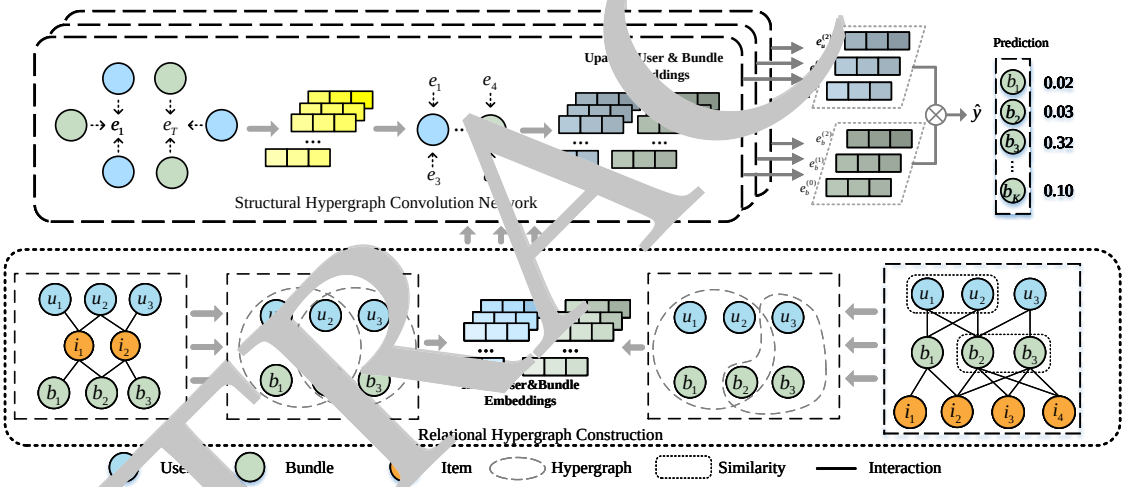

Figure 2. The SHCBR framework can be divided into two main components: in the relational hypergraph construction, we constructs a relational hypergraph matrix with items as links and obtains user/bundle initial embeddings; the structural hypergraph convolutional neural network is proposed for capturing high-order relationships of users and bundles on a relational hypergraph.

According to the definition in Section 3.1, in the application scenario of bundle recommendation, we acquire the interaction matrix X between users and items, the interaction matrix Y between users and bundles, and the affiliation matrix Z between bundles and items, in a natural manner. Given the complexity of relationships among the three types of nodes, the relevant items can serve as links between users and bundles. We construct the item association matrix as follows:

$$H_I = \begin{bmatrix} X & Z \end{bmatrix}^T \tag{3}$$

where $X \in \mathbb{R}^{|U| \times |I|}$, $Z \in \mathbb{R}^{|B| \times |I|}$, and $H_I \in \mathbb{R}^{(|U|+|B|) \times |I|}$. Similarly, using items as connectors and leveraging second-order interaction data, we can infer the similarity between users. Likewise, we infer the similarity between bundles. Highly overlapping users and bundles

aid in predicting user interests in bundles. Therefore, we further define the user–bundle adjacency matrix as follows:

$$H_{UB} = \begin{bmatrix} S_U & Y \\ Y^T & S_B \end{bmatrix} \tag{4}$$

$$S_U = X \cdot X^T, \; S_B = Z \cdot Z^T \tag{5}$$

where $H_{UB} \in \mathbb{R}^{(|U|+|B|)\times(|U|+|B|)}$. $S_U \in \mathbb{R}^{|U|\times|U|}$ and $S_B \in \mathbb{R}^{|B|\times|B|}$ denote the user similarity-overlap matrix and the bundle similarity-overlap matrix constructed based on second-order interactions, respectively.

Based on the flexibility of hypergraphs in representing and capturing complex relationships in practical problems, the relational hypergraph matrix we constructed can be defined as follows:

$$H_{SC} = \begin{bmatrix} H_{UB} & H_I \end{bmatrix} \tag{6}$$

where $H_{UB} \in \mathbb{R}^{(|U|+|B|)\times(|U|+|B|)}$, $H_I \in \mathbb{R}^{(|U|+|B|)\times|I|}$, and $H_{SC} \in \mathbb{R}^{(|U|+|B|)\times(|U|+|B|+|I|)}$. The three types of nodes are jointly incorporated into a relational hypergraph H_{SC} from a global perspective, naturally capturing higher-order connections within it.

4.2. Structural Hypergraph Convolutional Neural Networks

According to the depicted in Figure 2, we propose a structural hypergraph convolutional neural network (SHCNN), which captures the higher-order associations between entities on the hypergraph structure. Hypergraphs [40] are an extension of graphs where hyperedges can connect multiple nodes. Compared with traditional graph structures, hypergraphs have more complex connectivity, making them more suitable for bundle recommendation tasks. Hypergraph convolution, mentioned in the work by authors [42], can be perceived as a form of message passing. Knowing how to define the convolution operation and efficiently propagate information between adjacent nodes is very crucial in bundle recommendation systems. It is a natural approach to introduce more convolutional layers in order to acquire higher-order associations between entities. However, this approach accompanies the downside of imposing a substantial escalation in computational requirements. Since the relational hypergraph structure naturally incorporates higher-order associations, it enables efficient learning of node representations using only one layer of SHCNN. We further simplify the formula for hypergraph convolution operation as follows:

$$\hat{H} = D_v^{-\frac{1}{2}} H D_e^{-\frac{1}{2}} H^T D_v^{-\frac{1}{2}} \tag{7}$$

$$E^{(l+1)} = \hat{H} E^{(l)} \tag{8}$$

where $\hat{H}$ denotes relational hypergraph's Laplacian spectral normalization matrix. The output of layer l denoted as $E^{(l+1)}$.

Furthermore, we design a personalized weight operation to enhance the accuracy of learning user and bundle representations. Inspired by the hypergraph attention operator [42], we make improvements to the similarity-overlap matrix in Equation (5). Taking S_U as an example, the similarity between user u_i and user u_j can be computed in following manner:

$$S_{ij} = \frac{exp\left(\sigma\left(sim\left(u_i, u_j\right)\right)\right)}{\sum_{k\in N_i} exp(\sigma(sim(u_i, u_k)))} \tag{9}$$

where $S_{ij} \in S_U$. N_i is the neighborhood set of user u_i. $\sigma(\cdot)$ is the nonlinear activation function like LeakyReLU [47] and eLU [48], enhancing the learning ability of neural networks. The similarity function $sim(\cdot,\cdot)$ is used to calculate the similarity between two vertices. This function is presented as follows:

$$sim\left(u_i, u_j\right) = a^T \left[u_i || u_j\right] \tag{10}$$

Here, a represents a weight vector that is utilized to generate a scalar value indicating similarity. Likewise, we compute the similarity between bundles. Through the above computations, we can obtain the user similarity-overlap matrix S_U and the bundle similarity-overlap matrix S_B.

4.3. Matrix Propagation Rule

Combining the application scenario of the bundle recommendation and the SHCR model, we propose a specialized matrix propagation rule to achieve the embedding propagation of the entire relational hypergraph. In order to simulate the complex interaction logic between three types of entities, we define the relational hyperedge adjacency matrix as follows:

$$A = H_{SC} D_e^{-\frac{1}{2}} H_{SC}^T \tag{11}$$

where H_{SC} is the relational hypergraph matrix we constructed in Equation (8). The Laplacian matrix L based on the relational hypergraph matrix H_{SC} can be defined as:

$$D_v = \begin{bmatrix} D_U & \\ & D_B \end{bmatrix} \tag{12}$$

$$L = D_v^{-1/2} A D_v^{-1/2} + \begin{bmatrix} I_U & 0 \\ 0 & I_B \end{bmatrix} = \begin{bmatrix} I_U & D_U^{-1/2} A_U D_B^{-1/2} \\ D_B^{-1/2} A_B D_U^{-1/2} & I_B \end{bmatrix} \tag{13}$$

where A_U and A_B are the relational hyperedge adjacency matrix of users and bundles, respectively. A_U and A_B together constitute the relational hyperedge adjacency matrix A. D_U and D_B are the node degree matrix of users and bundles, respectively. $I_U \in \mathbb{R}^{|U| \times |U|}$ and $I_B \in \mathbb{R}^{|B| \times |B|}$ are identity matrix. Inspired by LightGCN, the purpose of adding the identity matrix is to connect own nodes.

4.4. Model Prediction and Training

In the realm of hypergraph convolution, traditional methods like hypergraph neural networks [41] use the embeddings from the final layer as their ultimate representation. However, when the quantity of layers augments, the embeddings incline towards excessive smoothing, making it unreasonable to use only the last layer embeddings as the ultimate representation. In existing bundle recommendation models, BGCN [18] concatenates embeddings from all layers to merge information obtained from neighbors at different depths for prediction, potentially intensifying training challenges. Inspired by LightGCN [13], we choose to incorporate embeddings from different layers as the ultimate representation. This approach captures the information from different layers to enrich the semantics of the final representation. Next, we combine the embeddings from each layer. By merging the embeddings of each layer, we are able to generate comprehensive representations of each user and bundle:

$$e_u = \sum_{k=0}^{K} \alpha_k e_u^{(k)}, e_b = \sum_{k=0}^{K} \alpha_k e_b^{(k)} \tag{14}$$

where $\alpha_k \geq 0$ represents the significance of the embedding of the l-th layer in forming the ultimate embedding. To ensure fairness in contribution, we assign α_k as $1/(K+1)$.

Ultimately, we define the inner production of final user representations and bundle representations as our model prediction:

$$\widehat{y}_{ub} = e_u^T e_b \tag{15}$$

The ranking scores for generating recommendations rely on the final results. Observations indicate that when there is a interaction (e.g., purchase, click) between the user and the bundle, it can be inferred that this user has an interest in the bundle as a whole or in a majority of its individual items. Conversely, if a user does not interact with a bundle, it can be assumed that the user is not aware of that bundle. In order to predict bundles, we classify

bundles with interactions as positive samples and randomly choose an unobserved bundle as a negative sample. In conclusion, a pairwise framework for learning is established, and the Bayesian personalized ranking loss is applied to predict bundles:

$$L_{bundle} = (1-\alpha) \sum_{(j,e)\in R^+} \ln \sigma(\hat{y}_{je} - y_{je}) + \alpha \sum_{(j,f)\in R^-} \ln \sigma\left(\hat{y}_{jf} - y_{jf}\right) \quad (16)$$

where $R = \{(j,e,f)|(j,e) \in R^+, (j,f) \in R^-\}$. R^+ denotes bundles that has interactions with user u_j, while R^- denotes bundles that have no interactions with user u_j. We regard (j,e) as a positive sample and (j,f) as a negative sample. $\sigma(\cdot)$ denotes a sigmoid function. To prevent the model from overfitting, we add L_2 regularization into the loss function.

Subsequently, the AdamW optimizer [49] is employed to minimize the loss function, which iteratively updates neural networks' weights using the provided training data.

5. Experiment

In this section, we perform experiments on two public datasets to assess the SHCBR, with the aim of answering following research questions:

- RQ1: How does our SHCBR compare in performance to previous models?
- RQ2: How do key components influence the SHCBR performance?
- RQ3: How do different parameter settings affect the SHCBR's results?

5.1. Experiment Settings

5.1.1. Datasets and Metrics

We evaluate SHCBR and other baseline methods over two real-world datasets. The training/test/validation sets are randomly divided in a ratio of 70%/20%/10%, and the statistical data is indicated in Table 1.

Table 1. Dataset statistics.

Dataset	NetEase	Youshu
User	18,528	8039
Item	22,864	4771
Bundle	123,628	32,770
User–item	1,128,065	138,515
User–bundle	302,303	51,337
Bundle–item	1,778,065	176,667
Avg. bundle interactions	16.32	6.39
Avg. item interactions	60.88	17.23
Avg. bundle size	77.80	37.03
User–item density	0.05%	0.05%
User–bundle density	0.07%	0.13%

- **NetEase**: This is a dataset constructed using data provided by a Chinese music platform, Netease Cloud Music (http://music.163.com, accessed on 7 August 2017). As a social music software, it allows users to freely choose their favorite songs and add them to their favorites. Users can also choose to listen to playlists bundled with different songs.
- **Youshu**: This is a dataset constructed by the famous Chinese book review website Youshu (https://www.yousuu.com/, accessed on 7 August 2017). Youshu allows users to create their own booklists with different styles and types. Additionally, it can provide users with a bundle of related books.

To evaluate models' recommendation performance, two commonly-adopted ranking metrics are employed in experiments: *Recall@K* and *NDCG@K*. *Recall@K* and *NDCG@K* are metrics used to assess the performance of information retrieval and recommendation

systems. *Recall@K* measures how many of the actual target items are present among the top-K recommended items. *NDCG@K* aims to measure the ranking quality and quantity of truly relevant items within the top-K recommended items. It is worth mentioning that we have set the values of K as: {20, 40, 80}. Recall and NDCG can be calculated as below:

$$Recall@K = \frac{|D \cap B|}{D} \quad (17)$$

where K signifies that we are employing a top-K ranking approach for every user, which produce a bundle list B containing K items. D represents a bundle set that the user has previously interacted with. The symbol b_k refers to the bundle located at the k-th position in the generated bundle list B, while d denotes a bundle in the bundle set D that the user has indeed interacted with.

$$DCG@K = \sum_{k=1}^{K} \frac{2^{hit_k} - 1}{\log_2(k+1)} \quad (18)$$

$$hit_k = \begin{cases} 1, & \text{if } b_k \in D \\ 0, & otherwise \end{cases} \quad (19)$$

$$NDCG@K = \frac{DCG@K}{IDCG@K} \quad (20)$$

where *IDCG@K* is typically used as the denominator in the calculation of *NDCG@K* to compute the standardized NDCG value. *IDCG@K* is computed by arranging the truly relevant items in the ideal ranking order and then calculating their cumulative discounted gains.

5.1.2. Baselines

To demonstrate the superior performance of SHCBR, we conduct a comparison between it and the below models:

- MFBPR [4]: This is a popular MF model based on BPR loss optimization, which is widely used for implicit feedback.
- LightGCN [5]: LightGCN is a method based on graph convolutional networks that is specifically designed to handle multi-relational graphs.
- LightGCN [13]: This is an efficient and lightweight model that incorporates both graph neural networks and collaborative filtering ideas.
  BundleNet [5]: BundleNet constructs a tripartite graph consisting of users, bundles, and items, which utilizes GNN to learn node representation of entities.
- DAM [28]: DAM is a deep learning model that incorporates attention mechanisms to facilitate the acquisition of comprehensive bundle representations.
- BGCN [18]: BGCN leverages the powerful ability of GNN in learning from higher-order connections on graphs, modeling the complex relations between users, items, and bundles.
- MIDGN [19]: MIDGN designed a multi-view intent resolution graph network, using a graph neural network to separate user intent from different perspectives.

5.1.3. Implementation Details and Environment

For SHCBR, we utilize the PyTorch framework (https://pytorch.org/, accessed on January 2017) and optimize the model using the AdamW optimizer. In our experiments, we set the embedding size to 128. A batch size of 2048 is used to process both datasets. The search for the learning rate is conducted within the value set $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}\}$. Based on the experiments, it is determined that the optimal choice is 5×10^{-3}. To address the issue of overfitting, we have implemented the use of L2-norm with a specific value of 0.2. Moreover, we have also incorporated a dropout rate of 0.2 into our methodology. These measures aim to mitigate the potential negative impact of overfitting on our outcome.

To ensure experimental rigor, all experiments were performed on the same server. Our server is equipped with Nvidia GeForce RTX 2080Ti GPUs and runs on Ubuntu 14.04.2.

5.2. Performance Comparison and Analysis

We evaluate SHCBR's overall recommendation performance against existing baseline methods on two datasets. From Table 2, we can observe that SHCBR outperforms the baseline methods in terms of recommendation performance. The best-performing method, which is SHCBR, is highlighted in bold, and the strongest performing baseline is underlined. Referring to Table 2, we can make the observations and analysis as below.

Compared with the traditional machine learning method BPR, the graph-based methods LightGCN and RGCN exhibit more powerful learning capabilities. This is attributed to the advantages of graph neural networks in learning and aggregating multi-hop collaborative information. The performance of RGCN demonstrates the importance of modeling relationship between entities in personalized bundle recommendation tasks. Among the three methods, LightGCN performs the best, thanks to its simplified design of GCN, making it more suitable for bundle recommendation scenarios.

Over the past few years, there has been notable advancement in bundle recommendation research. It is worth mentioning that BundleNet constructs a user–item-bundle tripartite graph. Then, BundleNet formalizes the task of bundle recommendation as a graph-based link prediction problem. However, its performance is not satisfactory due to its overly simplistic neural network model. It has also been observed that some excellent graph neural network-based models (such as RGCN and BundleNet) perform even worse than non-graph machine learning methods like DAM, which leverages deep attention mechanisms and a multi-task manner to predict user preferences jointly. In addition, BGCN adopts graph neural networks to explicitly model complex relationships among entities. MIDGN decouples user–item and bundle–item interactions to obtain potential intentions of users and bundles from diverse perspectives. At the same time, it is also the best model among the bundle recommendation methods in the baseline. While graph-based methods are effective, hypergraphs offer a more flexible structure. Capturing complex associations between users, items, and bundles is crucial in bundle recommendation tasks, and hypergraphs have a natural advantage in handling higher-order associations. Therefore, SHCBR achieves the best results. Specifically, the performance of SHCBR on NetEase dataset is 20.18–33.33% higher than the best baseline. The performance of SHCBR on Youshu dataset is 11.07–20.37% higher than the best baseline.

Table 2. The performance of SHCBR and other baseline methods on two datasets. Bold represents the best results, while underline represents the second-best results.

Method	NetEase				Youshu			
	Recall @20	NDCG @20	Recall @40	NDCG @40	Recall @20	NDCG @20	Recall @40	NDCG @40
MFBPR	0.0355	0.0181	0.0600	0.0246	0.1959	0.1117	0.2735	0.1320
RGCN	0.0407	0.0210	0.0670	0.0280	0.2040	0.1069	0.3017	0.1330
LightGCN	0.0496	0.0254	0.0795	0.0334	0.2286	0.1344	0.3190	0.1592
BundleNet	0.0391	0.0201	0.0661	0.0271	0.1895	0.1125	0.2675	0.1335
DAM	0.0411	0.0210	0.0690	0.0281	0.2082	0.1198	0.2890	0.1418
BGCN	0.0491	0.0258	0.0829	0.0346	0.2347	0.1345	0.3248	0.1593
MIDGN	<u>0.0678</u>	<u>0.0343</u>	<u>0.1085</u>	<u>0.0451</u>	<u>0.2682</u>	<u>0.1527</u>	<u>0.3712</u>	<u>0.1808</u>
Ours	**0.0852**	**0.0458**	**0.1304**	**0.0561**	**0.3049**	**0.1838**	**0.4123**	**0.2112**
Imp%	25.66%	33.53%	20.18%	24.39%	13.68%	20.37%	11.07%	16.81%

5.3. Ablation Study of SHCBR

Next, ablation studies are conducted to analysis the effectiveness of several key designs in SHCBR. We sequentially remove some key components of SHCBR to create various

derivative models. Referring to the findings displayed in Table 3, the conclusions can be made:

Table 3. Ablation study. Bold represents the best results.

Method	NetEase		Youshu	
	Recall@20	NDCG@20	Recall@20	NDCG@
$SHCBR_{w/oHG}$	0.0801	0.0397	0.2879	0.1743
$SHCBR_{w/oHC}$	0.0797	0.0411	0.2838	0.1716
$SHCBR_{w/oMP}$	0.0832	0.0441	0.2978	0.1806
$SHCBR$	**0.0852**	**0.0458**	**0.3049**	**0.1838**

- $SHCBR_{w/oHG}$. This model removes the module of the relational hypergraph construction. In this part, we exclude the similarity overlap matrix form the user–bundle adjacency matrix. At the same time, we also eliminate the construction of the structural hypergraph matrix. We discover that SHCBR outperforms significantly $SHCBR_{w/oHG}$, demonstrating the effectiveness of the relational hypergraph construction module. Moreover, the experimental results also highlight the importance of the hypergraph structure in capturing node feature.
- $SHCBR_{w/oHC}$. This model removes the part of the structural hypergraph convolutional neural networks. Here, we replace the structural hypergraph convolutional neural networks with a simple graph convolutional neural network. It is apparent that SHCBR outperforms $SHCBR_{w/oHC}$. This is due to simple graph convolutional neural networks have limited aggregation capabilities compared to hypergraph convolutional neural networks. This demonstrates the superiority of our proposed structural hypergraph convolutional neural networks.
- $SHCBR_{w/oMP}$. This model removes the special matrix propagation rule module but retains other details of SHCBR. It can be observed that SHCBR is only slightly superior than $SHCBR_{w/oMP}$. Although SHCBR is not highly competitive compared to $SHCBR_{w/oHG}$, we can still see that the special matrix propagation rule is helpful for improving model performance.

5.4. Hyper-parameters Analysis

We further explore how hyper-parameters α and batch size influence the performance of SHCBR.

Research on hyper-parameters α. As shown in Figure 3, we analyze the loss function ratio of positive to negative samples in Equation (16) to investigate its influence on the performance of SHCBR. We compare SHCBR's *Recall*@20 and *NDCG*@20 when setting different α. We can observe that for both datasets, the highest *NDCG*@20 is achieved when the value of α is 0.4, and the highest *Recall*@20 is achieved when the value of α is 0.3. Therefore, we can draw the conclusion that the impact of negative samples is somewhat smaller compared to that of positive samples. This situation arises mainly due to the random selection strategy for negative samples.

Research on batch size. To study the impact of batch size on the model, we incrementally raise it from 128 to 8192. As shown in Figure 4, it can be observed that with the increase in batch size, the model's performance experiences an initial rapid improvement and then reaches a stable state. Based on the observed results, we adopt a 2048-size in our experiments on both datasets.

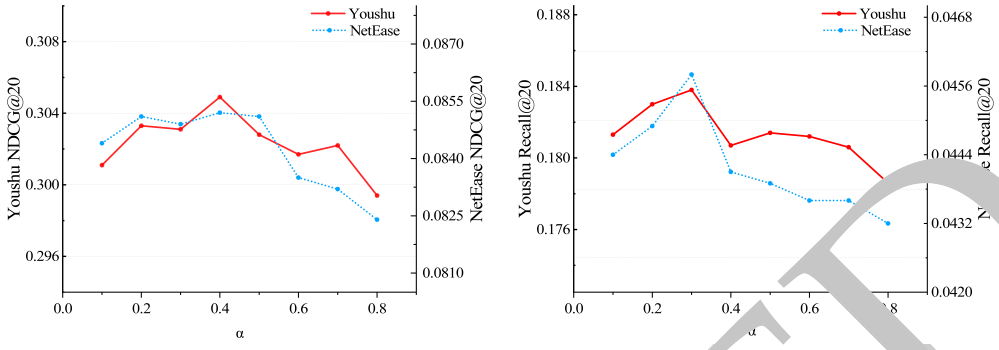

Figure 3. Impact of hyper-parameters α.

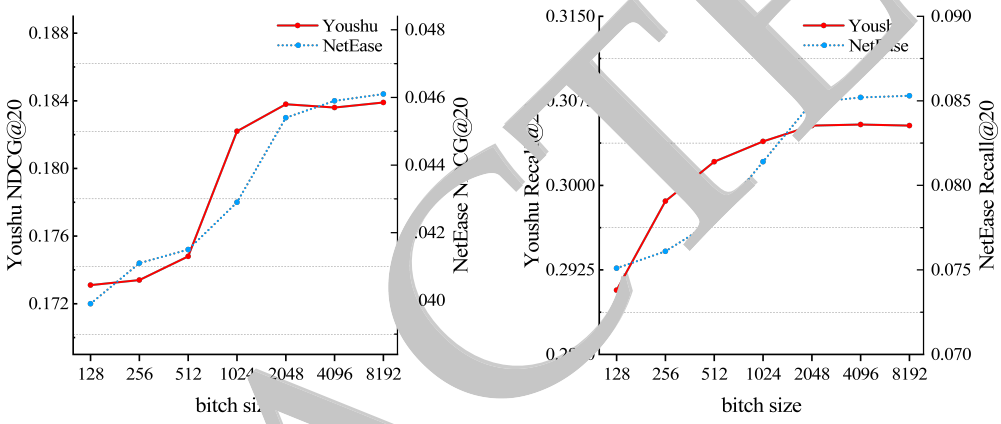

Figure 4. Impact of batch size.

5.5. Experimental Efficiency

The running cost is an important factor that needs to be considered. Lower running costs allow for faster iterations and reduce expenses. In this section, we analyze the runtime of the SHCBR model for each epoch and compared it with other models as indicated in Table 4. All experiments are performed utilizing the NetEase dataset. We choose the NetEase dataset because it has a significantly larger data interactions compared to the Youshu dataset.

Table 4. Execution cost of different method.

Method	Time for Train per Epoch	Time for Test
BundleNet	1610 s	5 s
DAM	1805 s	5507 s
BGCN	92 s	3 s
MIDGN	—	—
SHCBR	32 s	3 s

Table 4 demonstrates that our SHCBR is the most cost-effective approach, considering both the training and testing phases. The DAM model involves a large number of parameters, leading to a high optimization cost. This is primarily because the DAM model not only includes item-based factorized attention parameters but also incorporates extensive representations for users, bundles, and items. Additionally, DAM utilizes the multi-task learning method, which jointly models interactions between entities. This heavily relies on the powerful parallel computing capabilities of GPUs. However, due to limitations in GPU performance, the DAM model is less cost-effective. For BundleNet, the construction of the user–item-bundle triple graph significantly contributes to the increased runtime. Furthermore, the complexity of the graph propagation layer also results in negative effects on

time efficiency. On the other hand, BGCN, MIDGN, and SHCBR employ lightweight graph convolutional networks, which significantly improves training efficiency. For MIDGN, this model requires a substantial amount of GPU memory due to the generation of numerous representation vectors in its graph decomposition process. As indicated in Table 4, since the Nvidia RTX 2080Ti GPU ran out of memory, the running time of MIDGN model is not displayed. In contrast, our proposed SHCBR model utilizes structural[illegible] graph convolutional networks, which reduces aggregation layers and eliminates non-linear trans-formations. Additionally, we design a denser hypergraph Laplacian matrix, facilitating more efficient message propagation for convolution processing. Consequently, the SHCBR model exhibits an outstanding training time advantage.

6. Conclusions and Future Work

In this work, we investigate bundle recommendation tasks. Unlike traditional single-item recommendation tasks, bundle recommendation involves recommending a group of related items, i.e., bundles, to users. We propose a novel model named SHCBR, which jointly incorporates nodes of users, items, and bundles into a relational hypergraph from a global perspective. We utilize the flexible hypergraph structure to model multiple complex associations among three types of entities. With item nodes as links, we leverage efficient hypergraph convolution to learn representations of users and bundles while considering the high-order information, improving the quality of node representations. This modeling approach allows for better exploration of the underlying interests and associations behind user behavior, alleviating the dilemma of data scarcity. Our experiments conducted on two real-world datasets significantly illustrate the superior performance of SHCBR compared to baseline models. The experimental results further affirm that hypergraphs can offer a novel and effective approach to address bundle recommendation tasks.

This work is our initial attempt to explore the application of hypergraphs in the field of bundle recommendation. There are a number of challenging directions based on hypergraphs that warrant further exploration to address the difficulties in bundle recommendation. For example, in our experiments, hyperedges are equally weighted. When additional knowledge of the data distributions is available, theoretically, utilizing appropriate weighting mechanisms can enhance the accuracy of recommendation results. It is even possible to incorporate learnable modules within neural networks and optimize the weights using gradient descent. Combining hypergraphs with other deep learning methods is also promising. For example, the users and bundles on both sides of our relational hypergraph construction module can also serve as a suitable contrastive learning paradigm. In the future, we plan to study the integration of contrastive learning frameworks into our existing work to enhance recommendation efficiency while also alleviating the dilemma of data sparsity.

Author Contributions: X.L.: conceptualization, methodology, software, investigation, writing—original draft preparation, and writing—review and editing; M.Y.: supervision, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Heilongjiang Provincial Philosophical and Social Science Research Planning Project of China (No.19EDE334).

Data Availability Statement: The experimental codes and datasets are available at https://github.com/Lxtdzh/SHCBR-master/ (accessed on 2 August 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wu, S.; Sun, F.; Zhang, W.; Xie, X.; Cui, B. Graph neural networks in recommender systems: A survey. *ACM Comput. Surv.* **2022**, *55*, 1–37. [CrossRef]
2. Covington, P.; Adams, J.; Sargin, E. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 191–198.

3. Cao, D.; Nie, L.; He, X.; Wei, X.; Zhu, S.; Chua, T. Embedding factorization models for jointly recommending items and user generated lists. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 585–594.
4. Pathak, A.; Gupta, K.; Mcauley, J. Generating and personalizing bundle recommendations on steam. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 1073–1076.
5. Deng, Q.; Wang, K.; Zhao, M.; Zou, Z.; Wu, R.; Tao, J.; Fan, C.; Chen, L. Personalized bundle recommendation in online ga... In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, 19–23 Oct... 2020; pp. 2381–2388.
6. Zheng, Z.; Wang, C.; Xu, T.; Shen, D.; Qin, P.; Huai, B.; Liu, T.; Chen, E. Drug package recommendation v... interaction-awa... graph induction. In Proceedings of the Web Conference, Ljubljana, Slovenia, 19–23 April 2021, pp. ...84–1295.
7. Zhu, T.; Harrington, P.; Li, J.; Tang, L. Bundle recommendation in ecommerce. In Proceedings of the 37th Inte...ional A...M SIGIR Conference on Research & Development in Information Retrieval, Gold Coast, Austra...a, 6–11 Ju...y 2014; pp. ...56.
8. Su, Y.; Zhang, R.; Erfani, S.; Xu, Z. Detecting beneficial feature interactions for recommende... s...ms. In P...ceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; pp. 4357–4365.
9. Scarselli, F.; Gori, M.; Tsoi, A.; Hagenbuchner, M.; Monfardini, G. The graph neura... network mo... *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [CrossRef]
10. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. ...aph ...ral networks: A ...iew of methods and applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
11. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S. A comprehensi... rvey on grap... ural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef]
12. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.-S. Neural graph col...aborative filtering. In Proc...edings of the 42nd International ACM SIGIR Conference on Research and Development in Inform...tion Retrieval, Paris, France, 21–25 July 2019; pp. 165–174.
13. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. Lightgc... Simplifying and p...wering graph convolution network for recommendation. In Proceedings of the 43rd International ACM S...R Conference on ...esearch and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 639–648.
14. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Se...ion-based recom... ...with graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Hon... ...II, USA, 27 January–1 February 2019; pp. 346–353.
15. Wang, Z.; Wei, W.; Cong, G.; Li, X.-L.; Mao, X.-L.; Qi... M. Gl... ...ontext enhanced graph neural networks for session-based recommendation. In Proceedings of the 43rd Internatio...l AC...M SIGI... ...onference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 169–178.
16. Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E., Ta... ...; Yin, D. gr...ph neural networks for social recommendation. In Proceedings of the World Wide Web Conference, San ...rancisco, ...CA, USA, 13–...7 May 2019; pp. 417–426.
17. Guo, Z.; Wang, H. A deep graph ...ral netw...rk-based mec...anism for social recommendations. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2776–2783. [CrossRef]
18. Chang, J.; Gao, C.; He, X...an, D.; Li, Y. ...dle recommendation and generation with graph neural networks. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 232...2340. [CrossRef]
19. Zhao, S.; Wei, W.; Z...i, D...ao, X. Multi-viev...ent disentangle graph networks for bundle recommendation. In Proceedings of the AAAI Confe...nce on A...icial Intelligen...e, Virtual, 22 February–1 March 2022; pp. 4379–4387.
20. Bretto, A. Hy...rgraph theory. ...*An Introduction. Mathematical Engineering*; Springer: Cham, Switzerland, 2013; Volume 1.
21. Liu, Q.; Ge...Y.; Li, Z.; Chen, E.; X... ...i. Personalized travel package recommendation. In Proceedings of the 2011 IEEE 11th Internati...nal Conference on Data M...ning, Vancouver, BC, Canada, 11–14 December 2011; pp. 407–416.
22. Xie, M...kshmar...n, L.V.; Wood, P.T. Breaking out of the box of recommendations: From items to packages. In Proceedings of the Fourt... CM ...Confere...e on Recommender Systems, Barcelona, Spain, 26–30 September 2010; pp. 151–158
23. M...rchetti-Sp...amela, A.; ...ercellis, C. Stochastic on-line knapsack problems. *Math. Program.* **1995**, *68*, 73–104. [CrossRef]
24. Re...le, S.; Freu...thale..., C.; Gantner, Z.; Schmidt-thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **201...** arXiv:1205.2...
25. Che...n, Y.-L.; Tang, ...; Shen, R.-J.; Hu, Y.-H. Market basket analysis in a multiple store environment. *Decis. Support Syst.* **2005**, *40*, 33...-...Ref]
26. ...ng, Y.; Xiao, X.; Wang, X.; Lan, H. Customized bundle recommendation by association rules of product categories for online su...markets. In Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, Chi...18–21 June 2018; pp. 472–475.
27. Wib...owo, A.T.; Siddharthan, A.; Masthoff, J.; Lin, C. Incorporating constraints into matrix factorization for clothes package recommendation. In Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization, Singapore, 8–11 July 2018; pp. 111–119.
28. Chen, L.; Liu, Y.; He, X.; Gao, L.; Zheng, Z. Matching user with item set: Collaborative bundle recommendation with deep attention network. In Proceedings of the IJCAI, Macao, China, 10–16 August 2019; pp. 2095–2101.

29. Xiong, H.; Liu, Z. A situation information integrated personalized travel package recommendation approach based on TD-LDA model. In Proceedings of the 2015 International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC), Nanjing, China, 30 October–1 November 2015; pp. 32–37.
30. Kipf, T.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
31. Sun, J.; Wang, N.; Liu, X. IMBR: Interactive Multi-relation Bundle Recommendation with Graph Neural Network. In Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications, Dalian, China, 24–26 Nov[illegible]er 2022; pp. 460–472.
32. Liu, Z.; Sun, L.; Weng, C.; Chen, Q.; Huo, C. Gaussian Graph with Prototypical Contrastive Learning in E-Commerce Bu[illegible]e Recommendation. *arXiv* **2023**, arXiv:2307.13468.
33. Ma, Y.; He, Y.; Zhang, A.; Wang, X.; Chua, T. CrossCBR: Cross-view contrastive learning for bundle r[illegible]mmendation. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, W[illegible]shington DC, USA, 14–18 August 2022; pp. 1233–1241.
34. Yadati, N.; Nitin, V.; Nimishakavi, M.; Yadav, P.; Louis, A.; Talukdar, P. NHP: Neural hypergr[illegible]h link pr[illegible]diction. In [illegible]dings of the 29th ACM International Conference on Information & Knowledge Managemen[illegible], V[illegible]ual Eve[illegible], 19–23 Oct[illegible]ber 2020; pp. 1705–1714.
35. Xu, Y.; Zhang, H.; Cheng, K.; Liao, X.; Zhang, Z.; Li, Y. Knowledge graph embeddi[illegible]g with entity [illegible]ributes us[illegible]g hypergraph neural networks. *Intell. Data Anal.* **2022**, *26*, 959–975. [CrossRef]
36. Gao, Y.; Zhang, Z.; Lin, H.; Zhao, X.; Du, S.; Zou, C. Hypergraph learning: Meth[illegible]ds a[illegible]ractices. *IEEE T[illegible]s. Pattern Anal. Mach. Intell.* **2020**, *44*, 2548–2566. [CrossRef]
37. Zhou, D.; Huang, J.; Schölkopf, B. Learning with hypergraphs: Clusterin[illegible]assification, a[illegible]mbedding. *Adv. Neural Inf. Process. Syst.* **2006**, *19*.
38. Huang, Y.; Liu, Q.; Metaxas, D. Video object segmentation by hyp[illegible]rgraph cut. In Proceedin[illegible]of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–2[illegible] June 2009; pp. 1738–1745.
39. Gao, Y.; Wang, M.; Tao, D.; Ji, R.; Dai, Q. 3-D object retrieval and r[illegible]ognition with hyper[illegible]graph analysis. *IEEE Trans. Image Process.* **2012**, *21*, 4290–4303. [CrossRef] [PubMed]
40. Gao, Y.; Wang, M.; Zha, Z.; Shen, J.; Li, X.; Wu, X. Visual-textual joi[illegible]levance lea[illegible]ng for tag-based social image search. *IEEE Trans. Image Process.* **2012**, *22*, 363–376. [CrossRef] [PubMed]
41. Feng, Y.; You, H.; Zhang, Z.; Ji, R.; Gao, Y. Hypergraph [illegible]networks. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA[illegible]15–20[illegible]2019; pp. 3558–3565.
42. Bai, S.; Zhang, F.; Torr, P. Hypergraph convolution and [illegible]ypergraph a[illegible]ntion. *Pattern Recognit.* **2021**, *110*, 107637. [CrossRef]
43. Wang, M.; Liu, X.; Wu, X. Visual classification by ℓ_1-hy[illegible]graph modeling. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 2564–2574. [CrossRef]
44. Yu, Z.; Li, J.; Chen, L.; Zheng, Z. U[illegible]fying mu[illegible]i-association th[illegible]ough hypergraph for bundle recommendation. *Knowl.-Based Syst.* **2022**, *255*, 109755. [CrossRef]
45. Wang, Z.; Chen, J.; Rosas, F.; Zhu, T. [illegible]framework for personalized recommendations via user preference and dynamics clustering. *Exp[illegible] Syst. Appl.* **2022**, *204*, 117552. [CrossRef]
46. Peng, D.; Zhang, S. GC[illegible]GNN: A global-[illegible]text supported hypergraph neural network for enhancing session-based recommendation. *Electron. Co[illegible]mer[illegible]s. Appl.* **2022**, *52*, [illegible]129. [CrossRef]
47. Maas, A.; Hannu[illegible]A.; Ng, [illegible]ectifier nonlin[illegible]arities improve neural network acoustic models. In Proceedings of the International Conference o[illegible]Machine Learn[illegible](ICML 2013), Atlanta, GA, USA, 16–21 June 2013; p. 3.
48. Clevert, D.; Unterthiner, T.; Hoch[illegible]er, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* **2015**, arXiv:15[illegible]1.07289.
49. Losh[illegible]I.; Hut[illegible]F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
50. Wang, X.; [illegible]Y.; Liu, J.; W[illegible]H. Relational graph neural network with neighbor interactions for bundle recommendation service. In Proceedin[illegible]of the 2021 IEEE International Conference on Web Services (ICWS), Chicago, IL, USA, 5–10 September 2021; pp[illegible]67–172.