Preference Identification by Interaction Overlap for Bundle Recommendation

Fei-Yao Liang[1] , **Wu-Dong Xi**[2] , **Xing-Xing Xing**[2] , **Wei Wan**[2] ,
Chang-Dong Wang[1,3,4,5*] and **Hui-Yu Zhou**[3*]
[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[2]UX Center, NetEase Games, Guangzhou, China
[3]Guangxi Key Laboratory of Digital Infrastructure, Guangxi Zhuang Autonomous Region Information Center, Nanning, China
[4]Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China
[5]Guangdong Hengdian Information Technology Co., Ltd., Guangzhou, China
liangfy25@mail2.sysu.edu.cn, {xiwudong, xingxingxing, gzwanwei}@corp.netease.com,
changdongwang@hotmail.com, 170021121@qq.com

Abstract

In the digital age, recommendation systems are crucial for enhancing user experiences, with bundle recommendations playing a key role by integrating complementary products. However, existing methods fail to accurately identify user preferences for specific items within bundles, making it difficult to design bundles containing more items of interest to users. Additionally, these methods do not leverage similar preferences among users of the same category, resulting in unstable and incomplete preference expressions. To address these issues, we propose Preference Identification by Interaction Overlap for Bundle Recommendation (PIIO). The data augmentation module analyzes the overlap between bundle-item inclusions and user-item interactions to calculate the interaction probability of non-interacted bundles, selecting the bundle with the highest probability as a positive sample to enrich user-bundle interactions and uncover user preferences for items within bundles. The preference aggregation module utilizes the overlap in user-item interactions to select similar users, aggregates preferences using an autoencoder, and constructs comprehensive preference profiles. The optimization module predicts user-bundle matching scores based on a user interest boundary loss function. The proposed PIIO model is applied to two bundle recommendation datasets, and experiments demonstrate the effectiveness of the PIIO model, surpassing state-of-the-art models.

1 Introduction

Against the backdrop of the modern digital age, recommendation systems [Xi *et al.*, 2022; Wang *et al.*, 2022; He *et al.*, 2018; Chen and Wong, 2019] have come into their own as key tools for filtering vast amounts of content and products, thereby enhancing the user experience on various online platforms. As e-commerce [Chen *et al.*, 2019b] and digital content streaming services [Zhu *et al.*, 2019] thrive, these systems have become crucial in leading users through their complex decision-making journeys. Within the array of recommendation patterns, the concept of bundle recommendation [Bai *et al.*, 2019; Zhu *et al.*, 2014; Pathak *et al.*, 2017] stands out by leveraging the inherent synergistic effects among products. Bundle recommendations, by bringing together a collection of complementary items, simplify the decision-making process for users and also probe into untapped needs by offering items that users might not have initially considered. This inclusive strategy not only boosts customer value but also strengthens user engagement and the potential for service providers to generate increased revenue. Investigating the intricacies of bundle recommendation algorithms and their effects on user satisfaction and business success is essential for the advancement of personalized recommendation systems.

In real-world bundle recommendation scenarios, a user's interaction with a bundle is often driven by interest in several items within it, rather than an interest in every item the bundle contains. For instance, in the context of music playlist recommendations, a user who interacts with a playlist of ten songs may actually be interested in only five or six of those songs. To better find bundles of interest to users, it is essential to design bundles that encompass as many items of interest to the user as possible, thereby making it easier for them to engage with the bundle. Existing methods typically consider user-bundle and user-item interactions separately, often employing GNNs to aggregate these two types of interactions independently, as seen with methods like BGCN [Chang *et al.*, 2020; Chang *et al.*, 2023; Vijaikumar *et al.*, 2020; Yu *et al.*, 2022]. Such approaches segregate user-bundle interactions from user-item interactions and fail to analyze which specific items within a bundle prompted the user interaction. Consequently, these methods cannot accurately identify a user's specific preferences for a bundle, making it challenging to design bundles that contain a maximum num-

*Corresponding authors.

ber of items that are of interest to the user. Additionally, while individual user preferences are unique and can fluctuate over short periods, the preferences among users of the same category tend to be relatively stable and long-lasting. Current methods do not take into account the similarity of preferences among users within the same category. Therefore, many effective pieces of information remain untapped in existing bundle recommendation approaches, which leads to an incomplete expression of user preferences.

To address the challenges mentioned above, we propose **P**reference **I**dentification by **I**nteraction **O**verlap for Bundle Recommendation (PIIO). Firstly, a data augmentation module is proposed, which calculates the probability of interaction for bundles that users have not yet interacted with but may be interested in, by analyzing the overlap between user-item interactions and bundle-item composition. Using a positive sampling approach, the most probable bundles are added to the user-bundle interaction dataset, allowing the recommendation system to delve deeper into the user's specific preferences for each item in a bundle. This enables more accurate identification and recommendation of bundles that contain items of interest to the user. Secondly, an autoencoder is employed to aggregate preferences of similar users, capturing not only the unique preferences of individuals but also the collective preferences of users of the same category. Such an integrated approach helps to create a more complete and accurate user preference profile, thus enhancing the recommendation system's accuracy and user satisfaction. Lastly, recommendations are generated by ranking the predicted scores.

Extensive experiments conducted on two public datasets demonstrate that the PIIO model outperforms state-of-the-art methods across multiple evaluation metrics. The contributions of this paper can be summarized as follows:

- We propose a novel data augmentation module that more precisely uncovers users' specific preferences for individual items within a bundle, thereby elevating the personalization aspect of bundle recommendations.
- We introduce an autoencoder for the first time to aggregate and analyze preferences among users of the same category, constructing a more comprehensive user preference profile, which in turn facilitates more targeted and satisfying bundle recommendations.
- We conduct extensive experiments on two public datasets, and the results demonstrate that our proposed PIIO model significantly outperforms the state-of-the-art methods.

2 Related Work

2.1 Bundle Recommendation

Bundle recommendations aim to uncover user preferences for different services or product combinations, while providing personalized recommendations to those who may be interested in these bundles. BGN [Bai *et al.*, 2019] employs a feature-aware softmax and a masked beam search strategy optimized with determinantal point processes (DPPs), significantly enhancing the quality and diversity of the recommended list. CrossCBR [Ma *et al.*, 2022] introduces cross-view contrastive learning to model cooperative associations between two different views. MIDGN [Zhao *et al.*, 2022] compares user intentions decoupled from the global and local levels under a contrastive learning framework, capturing the diversity of user intentions and item associations at a finer granularity, accurately and comprehensively.

2.2 Data Augmentation

In the recommendation systems, data augmentation can alleviate issues related to data sparsity by increasing the volume of data. It can also generate or expand the characteristics of users or items, enhancing the richness of features and the accuracy of recommendations. CL4SRec [Xie *et al.*, 2022] generates self-supervised signals by manipulating user behavior sequences and uses three data augmentation techniques to construct these signals. CoSeRec [Liu *et al.*, 2021a] expands on CL4SRec by introducing two information augmentation operations to boost performance, specifically including the substitution of related items and the insertion of new items. UserBERT [Wu *et al.*, 2022] introduces a medium-hard negative sample mining framework to enhance the effects of contrastive pre-training. DuoRec [Qiu *et al.*, 2022] produces positive samples that are semantically similar yet distinct in features by applying different dropout masks. ASRep [Liu *et al.*, 2021b] and BiCAT [Jiang *et al.*, 2021] enrich sequence content by generating priority items in reverse.

3 Problem Formulation

In real-world bundle recommendation scenarios, accurately identifying which items in a bundle prompted a user to interact with it, and consequently designing bundles that include as many items of interest to the user as possible, can effectively enhance bundle recommendation performance. Existing methods consider user-bundle interactions and user-item interactions separately, isolating them from each other, and fail to analyze which specific items in a bundle led to the user's interactions with the bundle. Furthermore, preferences among users of the same category tend to be relatively stable and persistent, yet existing methods do not take into account the similarity of preferences among users of the same category. Therefore, this paper aims to incorporate these valuable pieces of information to comprehensively capture user preferences. In this paper, we propose to enrich user-bundle interactions through data augmentation and aggregate preferences among users of the same category.

4 Method

To leverage the aforementioned valuable information, including which items in a bundle led to user interactions and the preference similarity among users of the same category, we propose a novel PIIO model. In this section, we first introduce the overall architecture of the PIIO model, as shown in Figure 1. Then, we describe in detail the two modules of the PIIO model, including the data augmentation module, the preference aggregation module, and the optimization module.

4.1 Overall Architecture

As shown in Figure 1, the PIIO model consists of three modules: the data augmentation module, the preference aggregation module, and the optimization module. Firstly, a data augmentation module is designed to calculate the likelihood of a bundle not interacted by the user being interacted based on the overlap between bundle-item inclusions and user-item interactions. The bundle with the highest likelihood is added as a positive sample to enrich user-bundle interactions in a data augmentation manner. Secondly, a preference aggregation module is designed to calculate the similarity between users by leveraging the overlap in user-item interactions between different users. The top users with the highest similarity are selected as users of the same category. And an autoencoder is used to aggregate the preferences of these users of the same category. Finally, an optimization module is designed to predict the personalized score $r_{u,b}$ between the user and the entire bundle, in order to determine whether the user will interact with a particular bundle.

4.2 Data Augmentation Module

Users interact with a bundle because they are interested in several items within it. Therefore, designing bundles that contain as many items of interest to users as possible is beneficial for recommending bundles that users are interested in, thereby enhancing the performance of recommendations. In this section, we propose considering the overlap between bundle-item inclusions and user-item interactions. Based on this, we calculate the likelihood of a bundle not interacted by the user being interacted with and sample the bundle with the highest likelihood to add to user-bundle interactions. This approach helps uncover specific preferences of users towards items within the bundle.

User u's interactions with items are denoted as $A^u = \{v_1^u, v_2^u, \dots, v_n^u\}$, where n denotes the number of items user u has interacted with. Bundle b's item inclusions is denoted as $A^b = \{v_1^b, v_2^b, \dots, v_m^b\}$, where m denotes the number of items contained in bundle b. The overlap between user u's item interactions and bundle b's item inclusions is denoted as $O^{ub} = A^u \cap A^b$. The likelihood of bundle b being interacted with by the user can be denoted as $p^{ub} = \left|O^{ub}\right|$, which is the number of items contained in O^{ub}.

The more items in bundle b that the user has interacted with, the more items of interest to the user are contained in the bundle b. Since users interact with bundles due to their interest in several items within them, the likelihood of bundle b being interacted with by the user increases. The higher the overlap degree, the larger p^{ub} becomes, and consequently, the higher the probability of bundle b being interacted with by user u. We calculate the interaction probability p^{ub_i} for all non-interacted bundles, where b_i belongs to $\mathcal{B}$ but has not been interacted with. The bundle with the highest probability b_{max} can be denoted as $b_{max} = max_{b_i}(p^{ub_i})$. The bundle calculated above with the highest likelihood of being interacted with by user u can be added as a positive sample to the user-bundle interactions, thereby enriching the user's preferences for bundles. Due to the limited amount of information available for users with fewer interactions with bundles, we perform data augmentation for these users. For users with more interactions, their original data will be retained because they already cover a large amount of information. For users with fewer than n_{inter} interactions with user-bundles, we randomly select a proportion of t of these users to add positive samples for them, where n_{inter} varies in different data scenarios.

4.3 Preference Aggregation Module

The interaction between users and bundles exhibits unique and transient user preferences. However, in real-world bundle recommendation scenarios, preferences among users of the same type are relatively stable and enduring. For example, a user who enjoys watching horror movies and occasionally watches a comedy movie tends to have a more stable preference for horror movies compared to comedies. Considering the preferences of users of the same category can enrich and enhance the individual preferences of users, thus more comprehensively expressing user preferences. In this section, we use autoencoder to aggregate the preferences of users of the same category.

For target user u_t, user-item interactions are denoted as $A^{u_t} = \{v_1^{u_t}, v_2^{u_t}, \dots, v_n^{u_t}\}$, where n denotes the number of items that user u_t has interacted with. The interactions of user u_i with items are denoted as $A^{u_i} = \{v_1^{u_i}, v_2^{u_i}, \dots, v_{n_i}^{u_i}\}$, where n_i denotes the number of items that user u_i has interacted with. There is an overlap of items between the user-item interactions of user u_t and the user-item interactions of user u_i. A high degree of overlap signifies a high similarity in interactions between users u_t and u_i, suggesting that they can be considered as a category of users with similar preferences. The degree of similarity between user u_t and user u_i can be denoted by the number of overlapping items in their user-item interactions as follows:

$$d^{u_t,u_i} = |A^{u_t} \cap A^{u_i}| \tag{1}$$

where d^{u_t,u_i} denotes the degree of similarity between user u_t and user u_i.

Then we calculate the similarity degree of user u_t with all users as follows:

$$D(u_t) = \{d^{u_t,u_1}, d^{u_t,u_2}, \dots, d^{u_t,u_{|\mathcal{U}|}}\} \tag{2}$$

where $D(u_t)$ denotes the set of similarity degrees between user u_t and all users, and d^{u_t,u_i} denotes the similarity degree between user u_t and user u_i.

Then we select the top k users with the highest similarity as the users of the same category as follows:

$$P_k(u_t) = \{u_i \mid u_i \in \mathcal{U} \wedge d^{u_t,u_i} \in \text{TopK}(D(u_t))\} \tag{3}$$

where $P_k(u_t)$ denotes the set of k users of the same category.

In order to optimize the representation of user preferences, aggregating the features of users of the same category is considered a strategy. However, directly concatenating the features of users of the same category may lead to an overly large feature space, bringing a burden to information processing. In view of this, it becomes necessary to adopt dimensionality reduction measures, aiming to filter out the most crucial information. During this process, we utilize autoencoder to

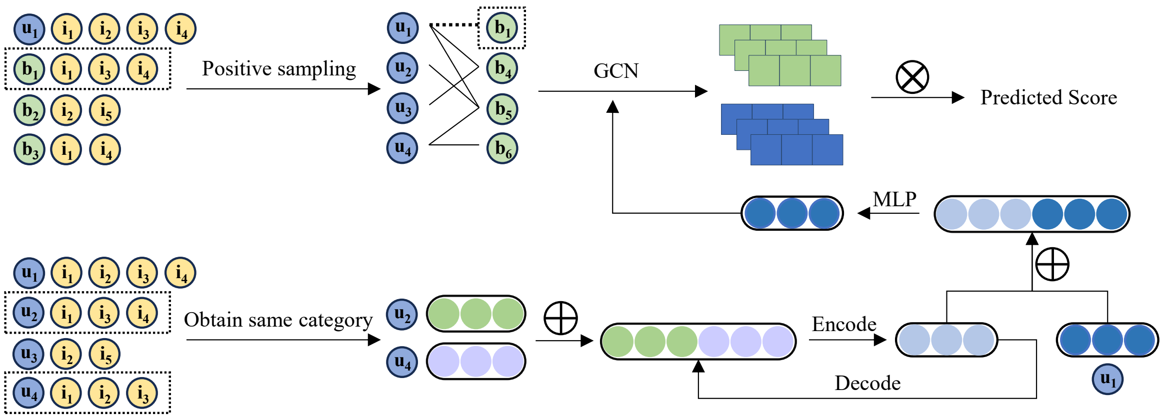

Figure 1: Overall Framework

compress the features of users of the same category, transforming them into lower-dimensional latent features. These simplified features can serve as an embedding form, integrating preferences from users of the same category more efficiently. By executing the above strategy, we not only reduce the complexity of the feature space but, more importantly, we are able to distill and retain those pieces of information that are critical for understanding and predicting user preferences. The features of users of the same category are concatenated to form the input to the autoencoder as follows:

$$\mathbf{e} = [\mathbf{e_1} \oplus \mathbf{e_2} \oplus \cdots \oplus \mathbf{e_k}] \tag{4}$$

where $\mathbf{e}$ denotes the concatenated features of users of the same category, $\mathbf{e_i}$ denotes the features of each user among users of the same category, and $\oplus$ denotes the vector concatenation operation.

The autoencoder consists of two parts: the encoder $E(\cdot)$ and the decoder $D(\cdot)$. Initially, the encoder is used to encode the concatenated features of users of the same category, compressing them into low-dimensional latent features as follow:

$$\mathbf{e_h} = E(\mathbf{e}) = \sigma_1(\mathbf{W_1 e} + \mathbf{b_1}) \tag{5}$$

where $\mathbf{e_h}$ denotes the low-dimensional latent features of users of the same category, $\mathbf{e}$ denotes the concatenated features of users of the same category, $\mathbf{W_1}$ denotes the parameter matrices, $\mathbf{b_1}$ denotes the bias vectors, and σ_1 denotes the activation function.

After encoding the concatenated features of users of the same category, the decoder $D(\cdot)$ is used to decode the low-dimensional latent features of users of the same category to obtain a reconstructed input corresponding to the original input as follow:

$$\widetilde{\mathbf{e}} = D(\mathbf{e_h}) = \sigma_2(\mathbf{W_2 e_h} + \mathbf{b_2}) \tag{6}$$

where $\widetilde{\mathbf{e}}$ denotes the reconstructed input of users of the same category, $\mathbf{e_h}$ denotes the low-dimensional latent features of users of the same category, W_2 denotes the parameter matrices, b_2 denotes the bias vectors, and σ_2 denotes the activation function.

We then train the autoencoder to find the parameter matrices and bias vectors that maximize the approximation between the original input and the reconstructed input as follows,

$$\mathbf{W_1}, \mathbf{W_2}, \mathbf{b_1}, \mathbf{b_2} = \arg \min_{\mathbf{W_1}, \mathbf{W_2}, \mathbf{b_1}, \mathbf{b_2}} < [\Delta(\mathbf{e}, \widetilde{\mathbf{e}})] > \tag{7}$$

where $\mathbf{W_1}$, $\mathbf{W_2}$ denote the parameter matrices, $\mathbf{b_1}$, $\mathbf{b_2}$ denote the bias vectors, Δ denotes the reconstruction error between the original input and the reconstructed input, and $< \cdot >$ denotes the calculation of the mean using the data of all users.

Then, we use the trained encoder $E(\cdot)$ of the autoencoder to act on the concatenated features of users of the same category, encoding to obtain the latent features of users of the same category, which serve as the aggregated features of users of the same category as follow,

$$\mathbf{e_h} = E(\mathbf{e}) = \sigma_1(\mathbf{W_1 e} + \mathbf{b_1}) \tag{8}$$

where $\mathbf{e_h}$ denotes the low-dimensional latent features of users of the same category, $\mathbf{e}$ denotes the concatenated features of users of the same category, $\mathbf{W_1}$ denotes the parameter matrices, $\mathbf{b_1}$ denotes the bias vectors, and σ_1 denotes the activation function.

Finally, we concatenate the low-dimensional latent features of users of the same category with the user's features and fuse these two types of features as the user's final features through an MLP as follows

$$\mathbf{e_{u_t}} = MLP(\mathbf{u_o} \oplus \mathbf{e_h}) \tag{9}$$

where $\mathbf{e_{u_t}}$ denotes the user u_t's final features, $\mathbf{u_o}$ denotes the user u_t's original features, and $\mathbf{e_h}$ denotes the low-dimensional latent features of users of the same category. Through the autoencoder, we have obtained features that aggregate the preferences of users of the same category, thus fully expressing the user's preferences.

4.4 Optimization

We align the final representations of users and bundles, and output the corresponding predicted ratings. Specifically, we employ the dot product to calculate the inner product between the representations of users and bundles to predict their matching scores as follows:

$$\widetilde{\mathbf{r}}_{\mathbf{u_t},\mathbf{b}} = \mathbf{e}_{\mathbf{u_t}}^{\top} \mathbf{e_b} \tag{10}$$

Datasets	Youshu	NetEase
# User	8,039	18,528
# Item	32,770	123,628
# Bundle	4,771	22,864
# User-Item	138,515	11,128,065
# User-Bundle	51,377	302,303
# Bundle-Item	176,667	1,778,838

Table 1: The statistics of the datasets.

where $\mathbf{e_{u_t}}$ denotes the representation of user u_t, $\mathbf{e_b}$ denotes the representation of bundle b, and $\widetilde{\mathbf{r}}_{\mathbf{u_t},\mathbf{b}}$ denotes the matching score between user u_t and bundle b.

We optimize our model based on the User Interest Boundary (UIB) [Zhuo *et al.*, 2022] loss function. To ascertain whether a user is interested in an unseen bundle, the UIB loss provides a personalized decision boundary, which is formulated as $b_{u_t} = \mathbf{W_3 e_{u_t}}$, where $\mathbf{e_{u_t}}$ denotes the representation of user u_t, b_{u_t} denotes the decision boundary, and $\mathbf{W_3}$ denotes the parameter matrices.

The final loss can be denoted as follows:

$$\mathcal{L} = \sum_{(u,b_{ob})\in\mathcal{R}^+} \Phi(b_u - \widetilde{r}_{u,b_{ob}}) + \alpha \sum_{(u,b_{un})\in\mathcal{R}^-} \Phi(\widetilde{r}_{u,b_{un}} - b_u) \tag{11}$$

where $\mathcal{R}^+$ denotes the set of observed interactions, $\mathcal{R}^-$ denotes the set of unobserved interactions, $\widetilde{r}_{u,b_{ob}}$ denotes the predicted score for observed interactions, $\widetilde{r}_{u,b_{un}}$ denotes the predicted score for unobserved interactions, and b_u denotes the decision boundary.

5 Experiments

5.1 Experimental Settings

Datasets

Building upon previous work [Chang *et al.*, 2020; Ma *et al.*, 2022; Zhao *et al.*, 2022], we conduct extensive experiments on two datasets, Youshu and NetEase. These datasets include user-bundle historical interactions, user-item historical interactions and bundle-item inclusions. The two datasets exhibit distinct statistical characteristics due to the different application domains. In addition, they vary in size and sparsity, particularly in the average number of items per bundle within each dataset. The statistical data for these two datasets are presented in Table 1.

Evaluation

To evaluate the bundle recommendation task, we use two common metrics: Recall@k (R@k) and Normalized Discounted Cumulative Gain@k (NDCG@k) to assess the performance of our PIIO model. The higher the values of these metrics, the better the performance of the model. In our experiments, we set k to 20 and 40. During the testing phase, we utilize all bundles as the candidate set and measure the ranking of positive samples among all bundles. For each user's positive sample in the test set, we calculate the average score across all users to provide a comprehensive evaluation of our PIIO model.

Baselines

In our experiments, we compare our proposed PIIO model with the baseline models:

- **MF-BPR** [Rendle *et al.*, 2009] optimizes representations of users and items through BPR loss and predicts interactions between users and bundles.
- **LightGCN** [He *et al.*, 2020] constructs a user-bundle interaction graph and leverages GCNs to learn information from high-order interactions, predicting user-bundle interactions.
- **SGL** [Wu *et al.*, 2021] enhances mutual information using contrastive learning, extending LightGCN and achieving better performance.
- **DAM** [Chen *et al.*, 2019a] aggregates item embeddings within a bundle using an attention mechanism to obtain bundle features.
- **BundleNet** [Deng *et al.*, 2020] builds a unified user-bundle-item tripartite graph and employs neural network models to learn directly on the graph-structured data.
- **BGCN** [Chang *et al.*, 2020] constructs two separate item-view and bundle-view graphs, using neural networks to learn representations on both graphs.
- **CrossCBR** [Ma *et al.*, 2022] models cooperative associations between two different views through cross-view contrastive learning.
- **MIDGN** [Zhao *et al.*, 2022] disentangles the user's intent coupled with inter-bundle items at a global level and disentangles the user's intent coupled with items within each bundle at a local level.
- **GPCL** [Liu *et al.*, 2023] embeds each user/bundle/item as a Gaussian distribution rather than a fixed vector, capturing contextual information and alleviating sampling bias through a prototypical contrastive learning module.
- **CoHeat** [Jeon *et al.*, 2024] addresses the highly skewed distribution of bundle interactions with a popularity-based merging method.

Parameter Settings

We employ the the Xavier [Glorot and Bengio, 2010] and Adam [Kingma and Ba, 2015] methods to train our model. For the general parameters across both datasets, we set the embedding size to 64 and the learning rate to 0.005. For the Youshu dataset, we set the batch size to 1024 and the autoencoder's hidden size to 256. For the NetEase dataset, we set the batch size to 2048 and the autoencoder's hidden size to 128. In the Data Augmentation module, for users with fewer than 2 interactions with user-bundles in the Youshu dataset, we randomly select a proportion t of these users to add positive samples and tune t in {0.1, 0.2, 0.3, 0.4, 0.5}. For the NetEase dataset, we apply this approach to users with fewer than 16 interactions with user-bundles, and also tune t in the same values. In the preference aggregation module, we select the k most similar users as the same category users for user u

and tune k in $\{20, 30, 40, 50, 60\}$. For the baseline models, if the optimal parameter settings in the original paper are reasonable and conform to standards, we directly use the results reported in the original paper for comparison. Otherwise, we will use the results obtained after reproduction. Our model is trained end-to-end on an NVIDIA GeForce RTX 3090 GPU without any pre-training.

5.2 Overall Performance

The performance comparison results of the proposed PIIO model with the baseline model on the Youshu and NetEase datasets are shown in Table 2. We use bold to denote the best scores and underlined font to denote the second-best scores. We observe the experimental results and make the following key analyses.

The traditional recommendation system baseline models (MF-BPR, LightGCN, SGL) perform poorly, but LightGCN performs better than MF-BPR, indicating that using graph convolution to learn user-bundle interactions is effective. SGL further incorporates contrastive learning to enhance mutual information, and its performance is improved compared to LightGCN, which demonstrates the importance of contrastive learning. Cross-view bundle recommendation models (CrossCBR and MIDGN) learn user preferences from different views and model the correlations between views through contrastive learning, thus accurately capturing user intent and achieving significant improvements compared to traditional recommendation system baseline models. GPCL embeds user/bundle/item as Gaussian distributions and captures context information through prototype contrastive learning, mitigating the sampling bias issue and achieving the best performance among baseline models on the Youshu dataset. CoHeat is the most recent model that addresses the highly skewed distribution of bundle interactions based on a popularity merging approach and effectively learns latent representations through curriculum learning, achieving the best performance among baseline models on the NetEase dataset.

On both the Youshu and NetEase datasets, the proposed PIIO model outperforms baseline models in most metrics. Particularly, on the Youshu dataset, the PIIO model surpasses all baseline models across all metrics. The improvement in the recall metric indicates that the bundles recommended by the PIIO model can more effectively cover the bundles of interest to users, thereby enhancing user satisfaction and user experience. The increase in the NDCG metric suggests that the PIIO model not only identifies bundles that users may be interested in but also aligns the order of the recommended list more closely with users' actual preferences, placing the most interesting bundles at the top of the recommendation list, which increases user trust and improves the conversion rate of user interactions. We attribute the improvements in these two metrics to the data augmentation module and the preference aggregation module proposed in this paper. The data augmentation module uncovers the reasons why users interact with certain bundles by considering the degree of overlap between bundle-item inclusion and user-item interactions, thereby identifying which items within the bundles lead to the interaction of the users. The preference aggregation module identifies users of the same category based on their interactions with items and aggregates the preferences of users of the same category using an autoencoder. These two modules ensure the completeness of user preferences, thereby enhancing the effectiveness of the recommendations.

5.3 Ablation Study

To analyze how each module of the PIIO model influences recommendation performance, we conduct ablation study on both the Youshu and NetEase datasets. Specifically, we prepare the following three variants of the PIIO model for performance comparison:

- PIIO-BASE: The basic framework without the data augmentation and the preference aggregation module.
- PIIO-DA: The PIIO model without the preference aggregation module, retaining only the data augmentation module.
- PIIO-PA: The PIIO model without the data augmentation module, retaining only the preference aggregation module.

The comparison results are presented in Table 3. From Table 3, we make the following key observations as follow. All three variants exhibit a drop in performance across all metrics, with the exception of the R@40 metric on the Youshu dataset. The PIIO-BASE variant performs the worst, which indicates that both the data augmentation module and the preference aggregation module are effective individually. Compared to the PIIO model, the PIIO-DA variant shows a decrease in performance, suggesting that the preference aggregation module's use of an autoencoder to aggregate the preferences of users of the same category assists in enhancing the model's performance. The performance of the PIIO-PA variant also decreases, indicating that the data augmentation module effectively mines out which items in the bundle are of interest to the users and thereby interacts with the bundle, better representing user preferences and contributing to model performance improvement. Thus, the ablation study reveals that both the data augmentation module and the preference aggregation module are instrumental in boosting the PIIO model's ability to recommend bundles.

5.4 Hyperparameter Analysis

In this section, we use the Youshu dataset to study the impact of hyperparameters within the model on performance. Initially, we delve into the effect of the hyperparameter t in the Data Augmentation module, which decides the proportion of users that will undergo data augmentation. Moreover, we analyze the function of the hyperparameter k in the preference aggregation module, which determines the number of users to be aggregated as the same category for the target user u.

To assess the influence of the hyperparameter t on model performance, we tune the value of t in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, with the model's performance variations displayed in Figure 2. From the experimental results, we observe that the model's performance remains stable with changes in t, indicating a strong robustness of the data augmentation module.

To evaluate the impact of the hyperparameter k on model performance, we adjust the value of k in $\{20, 30, 40, 50, 60\}$,

Models	Youshu				NetEase			
	R@20	N@20	R@40	N@40	R@20	N@20	R@40	N@40
MF-BPR	0.1959	0.1117	0.2735	0.1320	0.0355	0.0181	0.0600	0.0246
LightGCN	0.2286	0.1344	0.3190	0.1592	0.0496	0.0254	0.0795	0.0334
SGL	0.2568	0.1527	0.3537	0.1790	0.0687	0.0368	0.1058	0.0467
DAM	0.2082	0.1198	0.2890	0.1418	0.0411	0.0210	0.0690	0.0281
BundleNet	0.1895	0.1125	0.2675	0.1335	0.0391	0.0201	0.0661	0.0271
BGCN	0.2347	0.1345	0.3248	0.1593	0.0491	0.0258	0.0829	0.0346
CrossCBR	0.2813	0.1668	0.3785	0.1938	0.0842	**0.0457**	0.1264	0.0569
MIDGN	0.2682	0.1527	0.3712	0.1808	0.0678	0.0343	0.1085	0.0451
GPCL	0.2882	0.1713	0.3963	0.2007	0.0833	0.0441	0.1270	0.0557
CoHeat	0.2804	0.1646	0.3781	0.1905	0.0847	0.0455	0.1302	0.0565
PIIO	**0.3008**	**0.1778**	**0.4060**	**0.2064**	**0.0848**	0.0451	**0.1310**	**0.0573**

Table 2: Comparison results of different models in terms of Recall and NDCG. The best and the second best scores are denoted in bold and underlined fonts respectively.

Methods	Youshu				NetEase			
	R@20	N@20	R@40	N@40	R@20	N@20	R@40	N@40
PIIO-BASE	0.2952	0.1756	0.4048	0.2053	0.0825	0.0436	0.1277	0.0556
PIIO-DA	0.2946	0.1758	**0.4094**	0.2063	0.0839	0.0447	0.1274	0.0561
PIIO-PA	0.2977	0.1767	0.4080	0.2063	0.0846	0.0443	**0.1319**	0.0568
PIIO	**0.3008**	**0.1778**	0.4060	**0.2064**	**0.0848**	**0.0451**	0.1310	**0.0573**

Table 3: Ablation study of the PIIO model and its three variants on Youshu and NetEase. Bold scores denote the highest results of all variants.

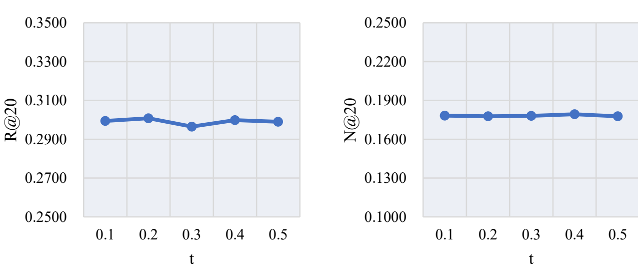

Figure 2: Parameter analysis on t.

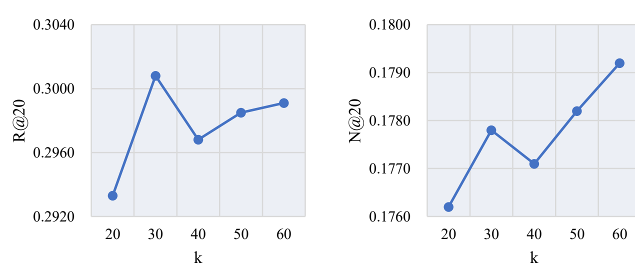

Figure 3: Parameter analysis on k.

and the model's performance variations are shown in Figure 3. As k increases, the performance of the model generally improves, maintaining an overall good level. We believe this is because the more similar users a user has, the more stable and enduring their preferences tend to be, thus, after aggregation, a richer user preference profile is obtained, which in turn enhances model performance.

6 Conclusion and Future Work

In this paper, we propose a novel PIIO model to improve the personalized effects of bundle recommendation systems. By delving deeper into user-bundle interactions and integrating the preferences of users of the same category, our model can more accurately identify items that users are interested in and generate bundle recommendations that are both more targeted and satisfying. Experimental results demonstrate that our model surpasses current state-of-the-art methods.

Looking ahead, there are directions for further exploration in future research. Firstly, as user preferences may change over time, it would be valuable to investigate how to dynamically capture these evolving preferences. Secondly, incorporating multimodal information can be explored. Lastly, enhancing the explainability of recommendations can strengthen users' trust in the recommendation system. By pursuing these future lines of inquiry, we aim to further refine bundle recommendation systems and enhance both user experience and recommendation effectiveness.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (2024YFA1011900), by Open Project Program of Guangxi Key Laboratory of Digital Infrastructure (GXDIOP2024011), and by the Guangxi Key R&D Program (2024AB08144), and by Support Scheme of Guangzhou for Leading Talents in Innovation and Entrepreneurship (2019017). This was also supported by the UX Center, NetEase Games.

References

[Bai *et al.*, 2019] Jinze Bai, Chang Zhou, Junshuai Song, Xiaoru Qu, Weiting An, Zhao Li, and Jun Gao. Personalized bundle list recommendation. In *WWW*, pages 60–71, 2019.

[Chang *et al.*, 2020] Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. Bundle recommendation with graph convolutional networks. In *SIGIR*, pages 1673–1676, 2020.

[Chang *et al.*, 2023] Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. Bundle recommendation and generation with graph neural networks. *IEEE TKDE*, 35(3):2326–2340, 2023.

[Chen and Wong, 2019] Tianwen Chen and Raymond Chi-Wing Wong. Session-based recommendation with local invariance. In *ICDM*, pages 994–999, 2019.

[Chen *et al.*, 2019a] Liang Chen, Yang Liu, Xiangnan He, Lianli Gao, and Zibin Zheng. Matching user with item set: Collaborative bundle recommendation with deep attention network. In *IJCAI*, pages 2095–2101, 2019.

[Chen *et al.*, 2019b] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *SIGIR*, pages 765–774, 2019.

[Deng *et al.*, 2020] Qilin Deng, Kai Wang, Minghao Zhao, Zhene Zou, Runze Wu, Jianrong Tao, Changjie Fan, and Liang Chen. Personalized bundle recommendation in online games. In *CIKM*, pages 2381–2388, 2020.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[He *et al.*, 2018] Jia He, Rui Liu, Fuzhen Zhuang, Fen Lin, Cheng Niu, and Qing He. A general cross-domain recommendation framework via bayesian neural network. In *ICDM*, pages 1001–1006, 2018.

[He *et al.*, 2020] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, pages 639–648, 2020.

[Jeon *et al.*, 2024] Hyunsik Jeon, Jong-eun Lee, Jeongin Yun, and U Kang. Cold-start bundle recommendation via popularity-based coalescence and curriculum heating. In *WWW*, pages 3277–3286, 2024.

[Jiang *et al.*, 2021] Juyong Jiang, Yingtao Luo, Jae Boum Kim, Kai Zhang, and Sunghun Kim. Sequential recommendation with bidirectional chronological augmentation of transformer. *CoRR*, abs/2112.06460, 2021.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Liu *et al.*, 2021a] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S. Yu, Julian J. McAuley, and Caiming Xiong. Contrastive self-supervised sequential recommendation with robust augmentation. *CoRR*, abs/2108.06479, 2021.

[Liu *et al.*, 2021b] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S. Yu. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. In *SIGIR*, pages 1608–1612, 2021.

[Liu *et al.*, 2023] Zhao-Yang Liu, Liucheng Sun, Chenwei Weng, Qijin Chen, and Chengfu Huo. Gaussian graph with prototypical contrastive learning in e-commerce bundle recommendation. *CoRR*, abs/2307.13468, 2023.

[Ma *et al.*, 2022] Yunshan Ma, Yingzhi He, An Zhang, Xiang Wang, and Tat-Seng Chua. Crosscbr: Cross-view contrastive learning for bundle recommendation. In *KDD*, pages 1233–1241, 2022.

[Pathak *et al.*, 2017] Apurva Pathak, Kshitiz Gupta, and Julian J. McAuley. Generating and personalizing bundle recommendations on *Steam*. In *SIGIR*, pages 1073–1076, 2017.

[Qiu *et al.*, 2022] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *WSDM*, pages 813–823, 2022.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[Vijaikumar *et al.*, 2020] M. Vijaikumar, Shirish K. Shevade, and M. Narasimha Murty. GRAM-SMOT: top-n personalized bundle recommendation via graph attention mechanism and submodular optimization. In *ECML PKDD*, pages 297–313, 2020.

[Wang *et al.*, 2022] Chang-Dong Wang, Wu-Dong Xi, Ling Huang, Yin-Yu Zheng, Zi-Yuan Hu, and Jian-Huang Lai. A BP neural network based recommender framework with attention mechanism. *IEEE TKDE*, 34(7):3029–3043, 2022.

[Wu *et al.*, 2021] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *SIGIR*, pages 726–735, 2021.

[Wu *et al.*, 2022] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. Userbert: Pre-training user model with contrastive self-supervision. In *SIGIR*, pages 2087–2092, 2022.

[Xi *et al.*, 2022] Wu-Dong Xi, Ling Huang, Chang-Dong Wang, Yin-Yu Zheng, and Jian-Huang Lai. Deep rating and review neural network for item recommendation. *IEEE TNNLS*, 33(11):6726–6736, 2022.

[Xie *et al.*, 2022] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *ICDE*, pages 1259–1273, 2022.

[Yu *et al.*, 2022] Zhouxin Yu, Jintang Li, Liang Chen, and Zibin Zheng. Unifying multi-associations through hypergraph for bundle recommendation. *KBS*, 255:109755, 2022.

[Zhao *et al.*, 2022] Sen Zhao, Wei Wei, Ding Zou, and Xianling Mao. Multi-view intent disentangle graph networks for bundle recommendation. In *AAAI*, pages 4379–4387, 2022.

[Zhu *et al.*, 2014] Tao Zhu, Patrick Harrington, Junjun Li, and Lei Tang. Bundle recommendation in ecommerce. In *SIGIR*, pages 657–666, 2014.

[Zhu *et al.*, 2019] Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. DAN: deep attention neural network for news recommendation. In *AAAI*, pages 5973–5980, 2019.

[Zhuo *et al.*, 2022] Jianhuan Zhuo, Qiannan Zhu, Yinliang Yue, and Yuhong Zhao. Learning explicit user interest boundary for recommendation. In *WWW*, pages 193–202, 2022.