Revisiting Bundle Recommendation

Datasets, Tasks, Challenges and Opportunities for Intent-aware Product Bundling

Sun, Zhu; Yang, Jie; Feng, Kaidong; Fang, Hui; Qu, Xinghua; Ong, Yew Soon

DOI
10.1145/3477495.3531904

Publication date
2022

Document Version
Final published version

Published in
SIGIR 2022 - Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval

Citation (APA)
Sun, Z., Yang, J., Feng, K., Fang, H., Qu, X., & Ong, Y. S. (2022). Revisiting Bundle Recommendation: Datasets, Tasks, Challenges and Opportunities for Intent-aware Product Bundling. In *SIGIR 2022 - Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2900-2911). (SIGIR 2022 - Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval). Association for Computing Machinery (ACM). https://doi.org/10.1145/3477495.3531904

Revisiting Bundle Recommendation: Datasets, Tasks, Challenges and Opportunities for Intent-aware Product Bundling

Zhu Sun
sunzhuntu@gmail.com
Institute of High Performance Computing and Centre for Frontier AI Research, A*STAR
Singapore

Jie Yang
j.yang-3@tudelft.nl
Web Information Systems
Delft University of Technology
The Netherlands

Kaidong Feng*
fengkaidong@stumail.ysu.edu.cn
School of Information Science and Engineering
Yanshan University
China

Hui Fang*
fang.hui@mail.shufe.edu.cn
Shanghai University of Finance and Economics
China

Xinghua Qu
xinghua.qu@bytedance.com
Speech and Audio Team, Bytedance AI Lab
Singapore

Yew Soon Ong
asysong@ntu.edu.sg
A*STAR Centre for Frontier AI Research and Nanyang Technological University
Singapore

ABSTRACT

Product bundling is a commonly-used marketing strategy in both offline retailers and online e-commerce systems. Current research on bundle recommendation is limited by: (1) noisy datasets, where bundles are defined by heuristics, e.g., products co-purchased in the same session; and (2) specific tasks, holding unrealistic assumptions, e.g., the availability of bundles for recommendation directly. In this paper, we propose to take a step back and consider the process of bundle recommendation from a holistic user experience perspective. We first construct high-quality bundle datasets with rich meta information, particularly bundle intents, through a carefully designed crowd-sourcing task. We then define a series of tasks that together, support all key steps in a typical bundle recommendation process, from bundle detection, completion, ranking, to explanation and auto-naming. Finally, we conduct extensive experiments and in-depth analysis that demonstrate the challenges of bundle recommendation, arising from the need for capturing complex relations among users, products and bundles, as well as the research opportunities, especially in graph-based neural methods. To sum up, our study delivers new data sources, opens up new research directions, and provides useful guidance for product bundling in real e-commerce platforms. Our datasets are available at GitHub (https://github.com/BundleRec/bundle_recommendation).

CCS CONCEPTS

• **Information systems → Recommender systems**.

*Corresponding authors

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531904

KEYWORDS

Product Bundling; User Intents; Crowd-Sourcing; Bundle Detection; Bundle Completion; Bundle Ranking

ACM Reference Format:
Zhu Sun, Jie Yang, Kaidong Feng, Hui Fang, Xinghua Qu, and Yew Soon Ong. 2022. Revisiting Bundle Recommendation: Datasets, Tasks, Challenges and Opportunities for Intent-aware Product Bundling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3477495.3531904

1 INTRODUCTION

In e-commerce, product bundling is a critical marketing strategy to support promotional campaigns, attract customers and increase sales revenue [5, 19, 25, 51, 63, 65]. It typically groups a collection of associated products that users consume *as a whole* under circumstances, e.g., limited total price [20, 22, 61], or specific intents [3, 31]. For illustration, three example bundles in the domain of Clothing, Electronic and Food are depicted in Figure 1. For example, (a) shows a bundle of party clothing, including handbags, dress, heels, necklace and earrings. Suppose that a customer is shopping online with an intent, e.g., purchasing fashion clothing for a party. In this case, they may expect such a bundle with a set of well matched fashion products, instead of items with no relationships.

As demonstrated above, bundles organize and present highly correlated products to customers and help them avoid tedious choices. It benefits both customers and sellers for at least the following reasons. (1) Bundles could help enhance user experience in different ways, e.g., alternative items could gather related products for a better comparison; whilst complementary items can broaden user horizon to help escape from monotonous choices. (2) Bundles could increase sales revenue for sellers, e.g., by exposing users to new items that they may not have considered in isolation. (3) Buying/selling bundles may cost less than buying/selling individual product separately for customers/sellers. For example, promotions and waived delivery fee are offered if the products in one order

(a) Dress up for a Party

(b) Upgrade Your Gaming Equipment

(c) Stay Safe, Stay Home

Figure 1: Running examples for bundles.

exceed a certain amount in e-commerce systems [3]. In this sense, *bundling* is a win-win solution for both customers and sellers.

Although several approaches have been recently proposed for bundle recommendation, they suffer from several inherent limitations. **1)** Most approaches are built upon noisy datasets, where bundles are defined by unverified heuristics. For instance, most studies [6, 17, 40, 69] simply treat co-purchased products as synthetic bundles, despite the fact that lots of such products are co-purchased with no common underlying intents. Others directly treat user-generated lists as bundles in specific domains such as music and books [9–11, 30, 31, 42]. Apart from being limited in those specific domains, those studies have not made an attempt to gain an understanding of the rationale behind product bundling. Some other work [14, 20, 22, 41, 55] leverages bundles defined by retailers. Obtaining such bundles is however a long, laborious and expensive process; and consequently, the sizes of such bundle datasets are limited. **2)** Existing studies are limited to specific tasks, e.g., they often directly dive into the task of bundle recommendation, with the unrealistic assumption that the historical bundles of a user are observable to the system. We argue that to support bundle recommendation, intermediate steps need to be in place such as detecting bundles from user sessions [35, 56]; furthermore, a set of auxiliary tasks, e.g., bundle auto-naming, need to be considered for successful bundle recommendation in real applications.

In this study, we take a step back and consider the process of bundle recommendation from a holistic user experience perspective. First, to better understand the rationale behind product bundling, we delicately design a crowd-sourcing task with the goal of leveraging crowd intelligence to help label potential bundles and corresponding intents hidden in the user session in three domains (Electronic, Clothing and Food). Our design draws inspiration from recent studies [35, 56] showing that users tend to explore highly-correlated (alternative or complementary) products with a common intent in a session [35, 56]. As a result, we obtain three high-quality datasets with rich meta information, particularly the bundle intents.

Using such data, we then propose several important interrelated tasks to support bundle recommendation, as illustrated in Figure 2. In reality, a user may interact (e.g., click or purchase) with multiple items in one session [35, 56], with or without common intents. An essential task is therefore *bundle detection*, aiming to efficiently detect potential bundle patterns hidden in the session. Accordingly, one subsequent task is *bundle completion*, which seeks to expand existing bundles by adding more relevant products for broader choices. Afterwards, *bundle ranking* is then needed to rank these enriched bundles based on user preferences for more accurate bundle recommendation. Meanwhile, *bundle explanation* could help interpret the results of bundle detection, completion and ranking via user intent inference, thereby enhancing system transparency and increasing user trust. With the inferred user intent, *bundle auto-naming* could help further generate attractive bundle names, e.g., 'Stay Safe, Stay Home' for the food bundle in Figure 1(c).

Lastly, to understand the research need for better bundle recommendation, we critically examine and analyze a set of state-of-the-art methods on our defined tasks through extensive experiments. We formulate and address nine research questions, ranging from the effectiveness of specific method design (e.g., graph-based neural methods) to general training strategies (e.g., pre-training). Specifically, we tailor the Apriori algorithm [2] for bundle detection at item category level and devise novel metrics for evaluation. We then adapt a number of state-of-the-arts for both bundle completion and ranking. Due to space limitation, we leave bundle explanation and auto-naming as our future study. From our experimental analysis, we draw the following conclusions: (a) pattern mining facilitates the generation of high-quality, new bundles; (b) the generative model VAE [38] performs well in bundle completion; and (c) graph-based neural methods [10] show the efficacy and necessity of capturing complex relations among users, products and bundles for bundle ranking; this also hints at the potential of graph-based methods such as hypergraph [64] for further improvement by modeling high-order connections among entities. Yet, (d) data sparsity remains a major challenge confronted by all tasks, which calls for more research, e.g., fusing side information (e.g. knowledge graph [57]) for further performance enhancement.

In summary, our main contributions lie in three folds. **(1)** We introduce three new bundle datasets with enriched meta data (e.g., bundle intents) contributed by crowds, thereby facilitating future research on intent-aware product bundling. **(2)** We define a series of tasks for product bundling, from bundle detection, completion, ranking, to explanation and auto-naming, pointing to new research directions to bring forward product bundling. **(3)** We perform extensive experiments and in-depth analysis on the defined tasks, showing both research challenges and opportunities. The resulting insights can help provide guidance, and ultimately promote product bundling in real e-commerce platforms.

2 RELATED WORK

This section provides an overview of state-of-the-arts for bundle recommendation, classified into eight categories.

Constraint based Methods. Early studies minimize the cost [20] or maximize the expected reward (e.g. revenue) of a bundle [7, 69] in e-commerce. Other methods combine constraints (e.g. season, price, ratings, user preference) for travel package recommendation [41, 61, 68]. CourseRank [45] recommends course bundles subject to the constraints of degree requirements.

Figure 2: The interrelated tasks for product bundling.

Data Mining based Methods. Association rule mining is utilized in [17, 24] for bundle generation and recommendation. In [6], K-means, Apriori algorithm and SVM are adopted to form and recommend bundles. A probabilistic model is devised in [40] to describe the generative process of bundle graphs and capture user preferences and motives for bundles.

Preference Elicitation based Methods. The preference elicitation based frameworks are proposed in [16, 62] to learn utility functions for capturing user preference among various features (e.g., cost and quality) over bundles via user feedback.

Factorization based Methods. The cost-aware latent factor model is delivered in [22] to learn user cost preference for tour bundle recommendation. LIRE [42] and BBPR [46] train Bayesian ranking models to simultaneously learn user preference towards items and bundles. BBPR can further generate new bundles using a greedy annealing schedule. EFM [9] jointly factorizes user-item, user-bundle interaction matrices and item-item-bundle co-occurrence matrices, to capture user preference over items and bundles.

Sequence based Neural Methods. BGN [3] adopts a sequence generation model and integrates the masked beam search and DPP selection to produce high-quality, diversified bundles. ComEmb [34] combines product hierarchy with transaction data or domain knowledge to identify bundle candidates which are then ranked via a LSTM [52] based deep similarity model.

Attention based Methods. DAM [11] designs a factorized attention network to aggregate items in a bundle to represent the bundle and jointly models user-bundle and user-item interactions. AttList [30] aggregates items to characterize the bundle that they belong to, and then integrates bundles to estimate user preference, whereby self-attentive mechanism is used to maintain item and bundle consistency. CAR [31] combines attention-based user (general and current) preference models via a consistency-aware gating network to capture dynamic user preference over bundles.

Graph based Neural Methods. BundleNet [14] applies graph convolutional network (GCN) [58] on the user-item-bundle tripartite graph and performs both item and bundle recommendation tasks for a mutual enhancement. BGCN [10] unifies user-item, user-bundle interactions and bundle-item affinity into a heterogeneous graph, and adopts GCN to perform item- and bundle-level propagation to learn user and bundle representations with item level semantics.

Other Related Studies. Some studies recommend bundles to groups of users by aggregating user preferences within groups [47, 50]. The effect of product characteristics (e.g., popularity) on bundling strategy and its complexity have been explored in [21] and [15], respectively. There are also potential item relations for bundle generation to maintain consistency [26, 28, 37, 39, 59] and diversity [1, 44].

Despite the prosperity of current research on bundle recommendation, majority studies are built on the basis of unrealistic datasets. They directly treat either co-purchased products [40, 69] or user-generated lists [9–11, 14, 16, 30, 31, 42, 46] as synthetic bundles, ignoring the fact that (1) co-purchased products may not always reflect common intents; and (2) user-generated lists are not only limited to specific domains (e.g. music and books), but also fail to unveil the rationale behind product bundling. Studies relying on bundles pre-defined by retailers [3, 6, 14, 17, 20, 22, 41, 55] are however restricted by limited size of datasets due to the high cost on producing such bundles. Furthermore, existing approaches merely focus on specific tasks, e.g., directly diving into bundle recommendation by assuming the availability of bundles to the system. Prior to that, a series of tasks need to be in place to support bundle recommendation, e.g., detecting bundles from user sessions, and completing them for more choices. Other auxiliary tasks, e.g., bundle explanation and auto-naming are also essential to promote bundle recommendation in real e-commerce platforms.

3 CONSTRUCTING BUNDLE DATASETS

This section describes the construction process for our new bundle datasets, considering specifically the necessity of having a consistent intent in the bundle. To obtain such intent information, we design and deploy a crowdsourcing task to identify potential bundles in a user session and label them with user intents.

3.1 Crowdsourced Annotation Task

Data Preparation. Starting off with the Amazon datasets [27] widely used in recommendation research, we select three domains (i.e. Electronic, Clothing, and Food) and chronologically order the records of each user according to the timestamp information, and then divide them into different sessions by days. Due to the long history of the original data (from May 1996 to July 2014), some products may be outdated. We only consider interactions that occurred in the last year of the data, i.e., from July 2013 to July 2014. We sample 2000 sessions for each domain with lengths from 2 to 10 considering that (1) most session lengths are in the range of $[2, 10]$ (see Figure 3(a)); and that (2) longer sessions may contain noisy items and increase the difficulty of annotation. To account for the imbalance of the session length, we adopt the stratified sampling strategy: for each domain, we divide the sessions into 4 groups based on their lengths – $[2, 3]$, $[4, 5]$, $[6, 7]$, $[8, 9, 10]$, and then sample 500 sessions from each group. As a result, we obtain 2000 sessions with balanced length distribution from each domain.

Task Design and Execution. Our task asks workers to identify potential bundles from user sessions and label them with corresponding user intents. It mainly consists of three parts, shown in

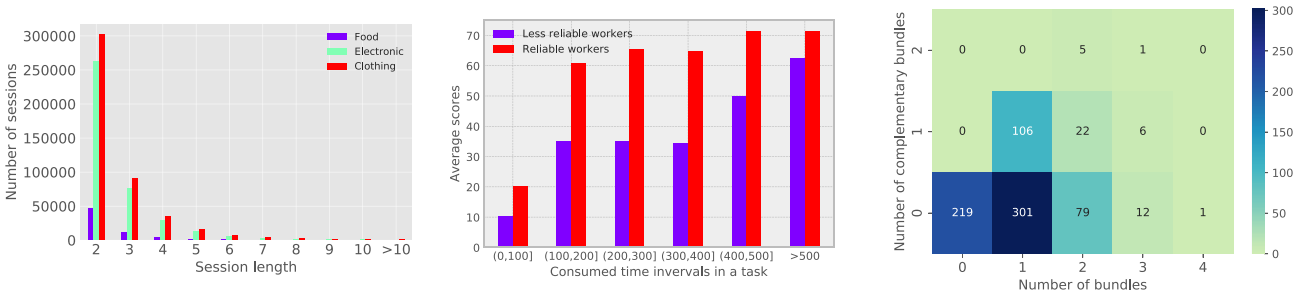
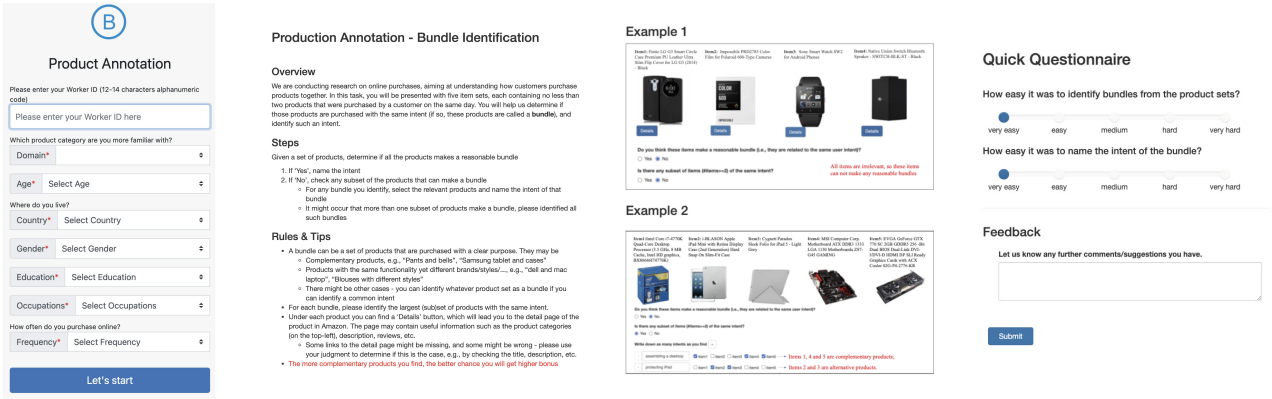


Figure 3: (a) session length distribution for the three domains; (b) score distribution on various time intervals; and (c) session distribution regarding the number of detected bundles and complementary bundles.



(a) First page **(b) Second page - part 1** **(c) Second page - part 2** **(d) Third page**

Figure 4: The three web-pages for our designed task.

Figure 4. In the beginning, each worker is asked to provide their basic information at the first page, including age, gender, country, occupation, education and online shopping frequency (Figure 4(a)). Next, the worker will then be navigated to the second page with the task overview, instructions and examples (Figures 4(b-c)), followed by five user sessions that need to be annotated. For each user session, the worker first answers the question 'do you think these items make a reasonable bundle (i.e. they are related to the same user intent)?'. If the answer is yes, the worker is asked to name the bundle intent; otherwise, a following question shows up 'is there any *subset* of items of the same intent?'. If yes, the worker is asked to select items that form the subsets (i.e., bundles) and name the corresponding intents. After finishing the annotation, the worker is asked to provide their feedback on our task (Figure 4(d)).

We launch our task on Amazon Mechanical Turk (www.mturk.com/). With a 10$ per-hour standard (higher than the minimum wage 7.5$ per hour in the US), we pay each annotation task 1$, where 0.3$ is the base reward, and the rest 0.7$ is the bonus for high-quality contributions.

Quality Control Mechanisms. A key concern of crowdsourced data annotation is the annotation quality: workers seek to maximize their monetary rewards by doing tasks as fast as possible, potentially de-prioritizing the quality of their work [18]. To deal with this issue, we adopt some widely-used quality control mechanisms, and beyond that, develop our own customized mechanisms in this study.

First, each sampled session is annotated by three workers, so that we can obtain higher quality annotations via results aggregation. Second, we use honeypot sessions to detect low-quality annotations. Within the five sessions prepared in each annotation task, we include one session with ground-truth bundles and intents available—workers who fail to provide correct annotation for this session is deemed to be less reliable. To create such ground-truth sessions, we randomly sample 90 extra sessions (10 for each bundle size) from each domain, and have three authors manually annotate them. By carefully cross-checking with each others, a consensus is reached for all the annotations w.r.t. both the presented bundles and intents. Third, we record the time that each worker spends in completing the task: those who spend too short or too long time may indicate low-quality work [67].

While helping to build the first line of defense, those quality control mechanisms might be too general. For example, it remains unclear how to best set the threshold of time elapsed for filtering annotations, and use it together with the honeypot mechanism. Besides, given a limited budget, we seek to maximize the diversity of the bundles, in terms of properties that go beyond bundle sizes. With these in mind, we decide to serially launch our task in two batches, using the first batch to get preliminary feedback and do necessary adjustment in the second batch.

3.2 Bundle Collection Procedure

At the first batch, we randomly select 300 sessions for each domain, and finally obtain the results contributed by 577 workers.

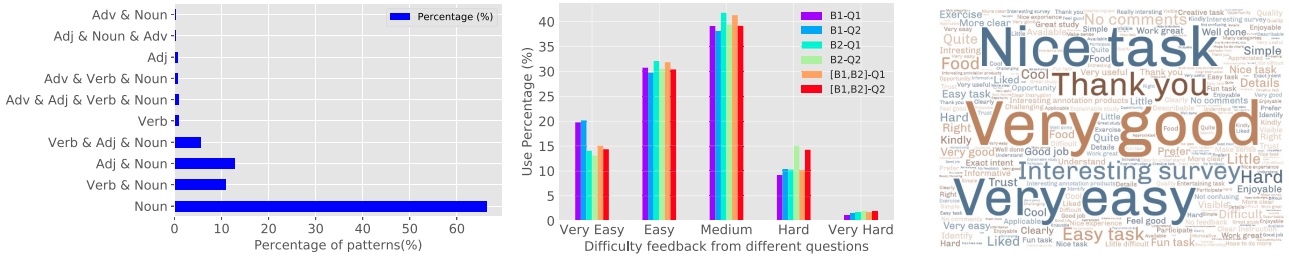


Figure 5: (a) distribution of intent wording patterns; (b) difficulty feedback for two batches; and (c) general feedback for the second batch, and we omit the results for the first batch due to space limitation, where similar trends can be observed.

Quality Assessment. To filter out low-quality annotations, we assess the annotation quality from two aspects, namely *intent coverage* and *item coverage*, denoting to what extent a worker can find out 'all intents' from a session and 'all items' for an individual intent, respectively. Formally,

$$intent\_score = \#cor\_intents/\#all\_intents * 100,$$

$$item\_score = (\#cor\_items - \#wro\_items)/\#all\_items * 100,$$

where $\#cor\_items$ and $\#wro\_items$ refer to the number of correct items and wrong items selected by the worker regarding a specific intent, respectively. We average the intent and item score to measure the reliability of each worker.

Filtering Annotations. We divide all workers from the first batch into two groups, viz., workers who provide correct answers for the honeypot sessions are considered reliable, and otherwise less reliable. Based on the above metrics, our goal is to find a sweet spot for the threshold of the time used by workers, such that we can keep as many high-quality annotations as possible – even from less reliable workers – while excluding low-quality ones.

To do so, we first manually filter out workers who provide obvious wrong answers with extremely short time consumed (e.g. $< 50s$). With our categorization, we are left with 451 workers with 277 being reliable and 174 being less reliable. Figure 3(b) depicts the average score for the two groups of workers in different time intervals (i.e., time spent in completing the task). As expected, reliable workers perform better (i.e., gain higher score) than less reliable workers across all intervals. The scores climb up as the time spent increases for all workers. Accordingly, we devise a rule to help filter high-quality annotations, viz., those from reliable workers with more than $100s$ spent and from less reliable workers with more than $500s$ spent. After the filtering, we are left with 752 sessions having valid annotations for the three domains, where 533 sessions contain bundles and the corresponding intents, and the rest 219 sessions do not contain items that can form bundles.

Item Relation Analysis. To further examine the quality of the collected data, we analyze the number of bundles detected by the workers and the relations (alternative or complementary [59]) among items inside bundles for each session. Arguably, an ideal dataset should contain more complementary items to avoid monotonous choices. As shown in Figure 3(c), we find most sessions do not contain any complementary bundles: there are only 140 (out of 752) sessions containing complementary bundles, indicating the scarcity of such bundles. To increase the number of complementary bundles under a limited budget, we investigate how to sample sessions from the remaining 1700 sessions (each domain) for our second batch.

Filtering Complementary Sessions. We use a data-driven approach to find characteristics of item pairs that can help us find more sessions with complementary items. To do so, we first manually label item relations—complementary and non-complementary—in the 533 sessions from our first batch, and then use these data to train a decision tree (DT) model to help automatically mine item relations in a session. Overall, we identify 976 pairs of complementary items and 11811 pairs of non-complementary items. Considering the unbalanced data distribution, we randomly sample an equal number of item pairs for each relation type, and use 1500 pairs to train and test the DT model. We construct five features using the meta data (i.e. category hierarchies and item images), including image distance calculated via ImageHash (pypi.org/project/ImageHash/), the number of common categories, the ratio of common categories, the depth between common category and leaf node, and the depth between root and common category. With careful parameter tuning, the DT reaches Precision, Recall, F1 and AUC on the test data (30%, cross-validation) with 52.6%, 4.3%, 8%, and 69%, respectively.

Afterwards, we use the DT to classify pair-wise item relations in the 1700 sessions for each domain: a session is classified as a complementary session as long as it contains one complementary item pair. As a result, we obtain 506, 338, and 463 complementary sessions in Electronic, Clothing, and Food, respectively. We further sample non-complementary sessions to ensure a total of 1000 sessions in each domain for the second batch while maintaining a higher ratio of complementary sessions.

Annotation Aggregation. As each session is labelled by three workers, there may be similar or contradictory labels. To aggregate the annotations, we first manually analyze 200 randomly-sampled sessions with their annotations to gain insights into worker agreement in bundle identification and intent labeling. We identify three cases: 1) same bundles (i.e., same items) with different intent expressions; 2) different yet overlapping bundles with semantically similar intents; and 3) non-overlapping bundles. For case 1, we aggregate the intents by firstly converting them into embedding vectors (using fastText: fasttext.cc) and then select the one closest to the centroid. For case 2, if one bundle is the superset of the others (in terms of the items included), we take the superset as the final bundle for the sake of completeness; if the bundles are partially overlapped, we take the overlapped items as the final bundle. For the last case, we keep the bundles as they are without any aggregation.

In the above process, semantically similar intents are detected using hierarchical clustering [43]. We use our manually-checked

Table 1: Examples for intent wording patterns.

Wording Patterns	Examples
Noun	Earrings and pendants set; Accessories for apple products;
Verb	Cooking; Baking;
Verb & Noun	Traveling and taking photos; Making a meal;
Adj & Noun	Men's jackets with different styles; High scale photography equipment;
Verb & Adj & Noun	Make hot drinks; Make digital photos;
Adv & Adj & Verb & Noun	The intent would be to upgrade Personal PC or utilizing PC more effectively.

data to find the optimal thresholds for grouping intents, which results in an accuracy of 90%, 92% and 96% for the three domains.

3.3 Resulting Datasets

After aggregation, we analyze the wording patterns of intents labelled by the workers according to POS tagging. Distribution of the wording patterns is shown in Figure 5(a), with examples of the patterns shown in Table 1. Note that we pre-process all intents by removing stopwords and conducting lemmatization. The results show that our task allows to collect a diverse set of intent patterns; in comparison, related studies such as [8] only consider two intent patterns, activity (verb) and audience (noun). A summary of the statistics of our collected datasets is given in Table 2.

We note that our data is contributed by workers with diverse background (distribution of their basic information shown in our GitHub page). We also analyze worker feedback to our annotation task. Results in Figure 5(b) shows that more than 80% of the workers believe the task has an easy or medium difficulty; only around 10% of them feel hard to complete the task. We obtain numerous positive feedback, e.g., good/interesting survey, enjoyable/fun/easy task, shown in Figure 5(c). As a final remark, we note that our original data source is Amazon where bundles can be easily identified with workers' general knowledge; we, therefore, believe the data annotated by workers is representative of real-life bundles.

4 THE PROPOSED INTERRELATED TASKS

4.1 Task Definition

As analyzed in Section 2, existing studies mainly focus on bundle recommendation. We argue that a number of tasks are essential, e.g., detecting bundles from user sessions, to support bundle recommendation. Based on our collected datasets, we propose a series of interrelated tasks (Figure 2), described below in detail.

Bundle Detection. With the boom of e-commerce, there are tremendous amount of users shopping online with e-commerce platforms (e.g. Amazon), and leaving their footprints (e.g. view, click and purchase products) everyday. Normally, during a short period (i.e. a session), users are more likely to explore highly-correlated (either alternative or complementary) products with certain intents [35, 56]. That is to say, these user generated sessions could provide rich data sources for generating high-quality and more diverse product bundles. Hence, one essential task is *bundle detection*, viz., given user session data, it aims to efficiently detect the potential bundles.

Table 2: Statistics of datasets.

	Electronic	Clothing	Food
#Users	888	965	879
#Items	3499	4487	3767
#Sessions	1145	1181	1161
#Bundles	1750	1910	1784
#Intents	1422	1466	1156
Avg. Bundle Size	3.52	3.31	3.58
#U-I Interactions	6165	6326	6395
#U-B Interactions	1753	1912	1785
Density of U-I Interactions	0.20%	0.15%	0.19%
Density of U-B Interactions	0.11%	0.10%	0.11%

Table 3: Important Notations.

Notation	Description
u, i, b, c, s, p	user, item, bundle, category, session and pattern
$b = \{i_1, \cdots i_{\|b\|}\}$	bundle b containing items $i_1, \cdots, i_{\|b\|}$
$\mathcal{B} = \{b_1, \cdots, b_{\|\mathcal{B}\|}\}$	the set of bundles
$\mathcal{C} = \{c_1, \cdots, c_{\|\mathcal{C}\|}\}$	the set of categories
$\mathcal{S} = \{s_1, \cdots, s_{\|\mathcal{S}\|}\}$	the set of sessions

Bundle Completion. Given the detected bundles, *bundle completion* aims to add more relevant products with the same intent for bundle expansion. It can serve a wide range of applications via expanding bundles in an automatic fashion. For instance, if the system detects a user browsing dress and heels in a session, which could form a bundle. With bundle completion, it can quickly further enrich it with, e.g., jewelry, to broaden user interest.

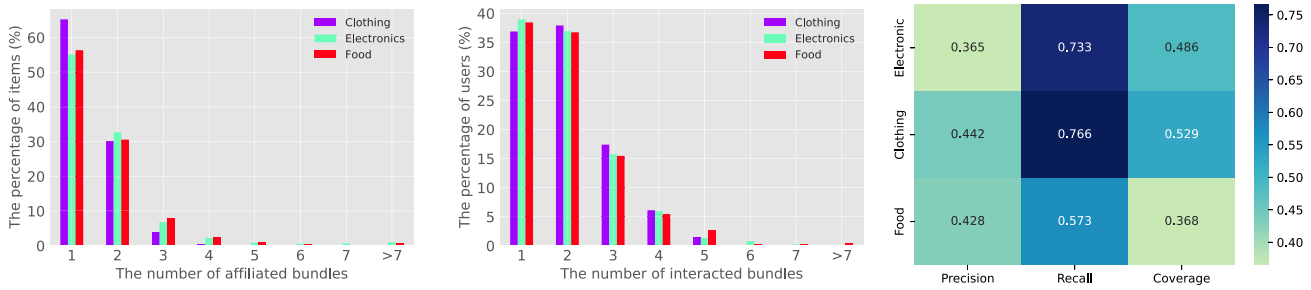
Bundle Ranking. Given the completed bundles, *bundle ranking* aims to rank these bundles according to user preference and demand for final recommendation. For instance, given a user session, several potential bundles (e.g., bundles for party clothing and cooking) may be produced via bundle detection and completion shown in Figure 2. In this case, bundle ranking will assist in sorting bundles on the basis of user interest, thus best satisfy user needs.

Bundle Explanation. One goal of *bundle explanation* is to infer intents (topics) for the completed bundles, e.g., given a bundle of products (e.g. dress and heels) in Figure 1(a), it can interpret as 'buying party clothing'. Another goal is to provide explanations for bundle recommendation. Suppose the system recommends a cooking essential bundle in Figure 1(c) to a user, it can explain as 'the user browsed steak and wine in this session'. This would help enhance system transparency and increase user trust.

Bundle Auto-Naming. Given the completed bundles and inferred intents, *bundle auto-naming* seeks to automatically generate attractive bundle names by exploiting pre-defined templates or generative language models [3]. For example, regarding the food bundle in Figure 1(c), it could generate fancy names, e.g., 'Stay Safe, Stay Home'. In this sense, bundle auto-naming would extensively save human labour on manual naming and promote product bundling in real e-commerce platforms.

4.2 Task Exploration

We now explore solutions for each defined task. Note that instead of proposing new methods, our goal is to gain a better understanding of these tasks by leveraging and adapting existing techniques, and critically examining and analyzing their strengths and weaknesses. To guide our analysis, we raise a number of research questions; by answering these questions, we aim to point out potential research



(a) Item-bundle distribution **(b) User-bundle distribution** **(c) Performance on bundle detection**

Figure 6: Item-bundle and user-bundle distribution as well as the performance on bundle detection across the three domains.

challenges and opportunities, as well as to provide insights and recommendations for future studies. Due to space limitation, we focus on the first three tasks and leave the rest for future exploration. Table 3 shows important notations.

Bundle Detection. Given a user session, bundle detection aims to discover potential bundles. This inspires us to mine frequent patterns on our collected bundles, and then perform pattern matching on user sessions to identify potential bundles. However, the data sparsity issue makes it impractical to directly mine patterns at item level. Instead, we consider item category. Specifically, we first map items in each bundle (e.g. $b_1 = \{i_1, i_2, i_3\}$) into the corresponding categories (e.g. $b_1 = \{c_1, c_2, c_3\}$), and then employ the Apriori algorithm [2] to discover frequent patterns (e.g. $p_1 = \{c_1, c_2\}$ or $p_2 = \{c_1, c_2, c_3\}$) for bundling products with the two core parameters, namely *support* and *confidence*.

As patterns are mined at category level, there may be patterns only involving one category (e.g. $p_3 = \{c_1, c_1\}$), indicating alternative products inside a bundle; meanwhile, the order of categories within each pattern does not matter, viz., we do not take into account antecedent and consequent. Consequently, we adapt the original support and confidence in the Apriori algorithm to seamlessly accommodate the unique properties in our study,

$$Sup(p = \{c_1, c_2, \cdots, c_n\}) = \frac{count(p = \{c_1, c_2, \cdots, c_n\})}{|\mathcal{B}|},$$
$$Con(p = \{c_1, c_2, \cdots, c_n\}) = \frac{count(p = \{c_1, c_2, \cdots, c_n\})}{\sum_{k=1}^{n} count(c_k)},$$

where $|\mathcal{B}|$ is the total number of bundles; $count(p)$ means the frequency of all categories within pattern p co-occurring in $\mathcal{B}$; $n \geq 2$ is the size of p, viz., the number of categories in p; $count(c_k)$ refers to the frequency of category c_k appearing in $\mathcal{B}$, and is only counted once within a bundle even if it may appear multiple times within that bundle. As such, a pattern is valid only if its *support* and *confidence* are no less than the pre-defined thresholds. With such valid patterns, we can perform pattern matching on any given user sessions for bundle detection.

In this task, we answer the following two research questions: (**RQ1**) does category-level pattern mining help detect high-quality bundles from user sessions? and (**RQ2**) does category-level pattern mining help generate new bundles?

Bundle Completion. Given partial products in a bundle as context, bundle completion seeks for more relevant products to further expand the bundle. However, one major challenge confronted is: a considerable amount of items only appear in one bundle as shown in Figure 6(a), which aggravates the data sparsity issue and increases the difficulty of this task. To combat this challenge, we design a strategy to obtain pre-trained item representations via BPRMF [48] for model initialization (details are deferred to 'The Strategy for Pre-training'). With such pre-trained item representations, we ultimately adapt four methods for this task.

(1) *ItemKNN* [49] calculates item similarity through pre-trained item representations, where the most similar items with the existing ones in the bundle are selected for completion. Note that we do not calculate item similarity via the bundle-item affinity matrix due to its high sparsity. (2) *BPRMF* [48] factorizes the bundle-item affinity matrix to learn bundle and item representations, and then selects the most similar items with the target bundle for completion. (3) *mean-VAE* [38] is inspired by VAE-CF [38], which takes the averaged representation of existing items in a bundle as the input, and then selects the items with highest probability for completion. (4) *concat-VAE* [38] is similar as mean-VAE where the only difference is the input, viz., concatenating representations of existing items in a bundle as input. Due to the varied bundle sizes, we align the input dimension by the padding operation.

Accordingly, in this task, we answer three research questions as follows: (**RQ3**) does neural-based bundle completion methods defeat traditional ones? (**RQ4**) which operation (mean-pooling or concatenation) performs better? and (**RQ5**) do pre-trained item representations help achieve better bundle completion?

Bundle Ranking. Given a number of completed bundles, bundle ranking aims to sort them based on user preference for recommendation. It is still facing the data sparsity issue, as most users only interact with a small number of bundles as shown in Figure 6(b). To ease this issue, the pre-trained item representations are also unitized for model initialization. As such, we examine the performance of six state-of-the-art methods on bundle ranking.

(1) *ItemKNN* [49] calculates bundle similarity with bundle representations, viz., the averaged representation of items in the bundle. Note that we do not calculate bundle similarity via the user-bundle interaction matrix due to its high sparsity. (2) *BPRMF* [48] factorizes the user-bundle interaction matrix to learn user and bundle representations. (3) *DAM* [11] designs a factorized attention network to aggregate item representations in a bundle for bundle representation and jointly models user-bundle and user-item interactions in a multi-task manner. (4) *AttList* [30] learns user and bundle representations via the self-attention mechanism, whereby it aggregates items to characterize the bundle that they belong to,

Figure 7: The detected bundles from sessions.

Figure 8: Case study of non-hit bundle patterns.

Table 4: Statistics of datasets for pre-training.

	#Users	#Items	#Interactions
Electronic	745,911	178,443	1,951,397
Clothing	223,571	236,387	749,423
Food	143,390	54,323	325,169

and then aggregates bundles to estimate user preference. (5) *GCN* applies the graph convolutional network [58] on the user-bundle-item tripartite graph to learn user and bundle representations. (6) *BGCN* [10] first constructs a heterogeneous graph by unifying user-item, user-bundle interactions and bundle-item affinity, and then applies propagation at both item and bundle levels to learn user and bundle representations. We do not consider EFM [9], as the source code for constructing its PMI matrix is not available and our re-implementation does not achieve promising results.

Given these methods, we answer four research questions: (**RQ6**) do neural-based bundle ranking methods outperform traditional ones? (**RQ7**) do self-attention mechanisms facilitate better bundle ranking? (**RQ8**) is it sufficient to model user-bundle interactions only for accurate bundle ranking? and (**RQ9**) does bundle ranking benefit from the pre-trained item representations?

Our Strategy for Pre-training. As observed in Table 2, the user-bundle interaction matrix is quite sparse in the three domains. To ensure the quality of learned representations, we adopt BPRMF to pre-train item representations to initialize all methods; meanwhile the bundle representation is initialized with the averaged representation of items inside this bundle. To avoid data leakage, we sample extra user-item interactions from each domain with the same time span as the bundle collection (i.e. July 2013 - July 2014) for pre-training. In particular, for each item in Table 2, we first find out all users who have interacted with it excluding those in Table 2, and then leverage these users' records to construct the user-item interaction matrix for pre-training. By doing so, the constructed interaction data covers all items but excludes all users in Table 2, therefore, do not leak any interactions in our collected data in Table 2. The statistics of datasets for pre-training across the three domains are shown in Table 4, where 80% of the interactions in each domain are treated as training data, and the rest as test data. The model is trained until optimal performance is observed on the test data regarding NDCG@10 to obtain better item representations.

5 INSIGHT, CHALLENGE AND OPPORTUNITY

This section seeks to answer the nine research questions raised in Section 4.2, thereby providing insights, and showing both research challenges as well as opportunities regarding the defined tasks.

5.1 Bundle Detection

We use the three collected datasets as shown in Table 2. For each dataset, every user session corresponds to its ground-truth bundles labelled by the workers. We divide them into training, validation and test sets with the ratio of 8:1:1, where the training set is used to mine frequent patterns with size ranging from 2 to 5; the validation set is used to tune *support* and *confidence* for best performance. Finally, we apply the valid patterns obtained from the training set to sessions in the test set for bundle detection.

We evaluate our method at both session and bundle levels. In particular, at session level, precision and recall are used to measure how many patterns have been correctly predicted for each session,

$$Precision = \frac{1}{|\mathcal{S}_t|}\sum\nolimits_{s\in\mathcal{S}_t}\frac{\#hit\_patterns}{\#predicted\_patterns},$$

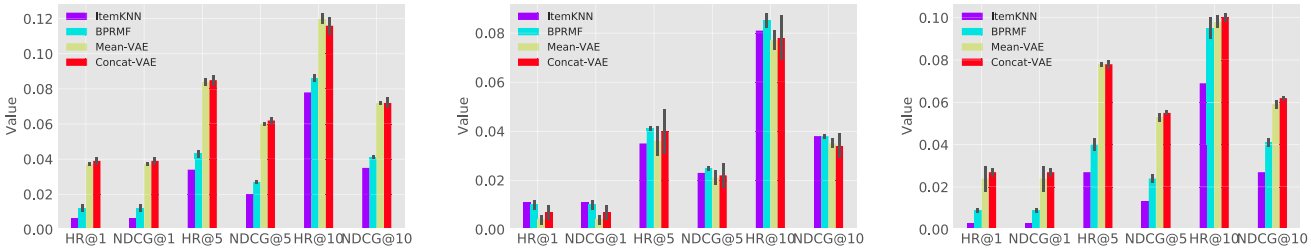
$$Recall = \frac{1}{|\mathcal{S}_t|}\sum\nolimits_{s\in\mathcal{S}_t}\frac{\#hit\_patterns}{\#ground\_patterns},$$

where $\mathcal{S}_t$ is all sessions in the test set; $\#ground\_patterns$ is the number of ground-truth patterns for each session s; and $\#hit\_bundles$ is the number of correctly predicted patterns that match ground-truth patterns (subsets included) for each session. Given a session, if multiple predicted patterns are all subsets of a same ground-truth pattern, we only retain the one with the maximum size. At bundle level, we adopt coverage shown below to measure how many categories are correctly covered by each predicted bundle pattern compared with the ground-truth pattern, where $\mathcal{B}_s$ is the set of predicted bundle patterns for s; $\#ground\_categories$ is the number of categories in the ground-truth pattern for bundle b[1].

$$Coverage = \frac{1}{\sum_{s\in\mathcal{S}_t}|\mathcal{B}_s|}\sum\nolimits_{s\in\mathcal{S}_t}\sum\nolimits_{b\in\mathcal{B}_s}\frac{\#hit\_categories}{\#ground\_categories}$$

Analysis and Insights. (**RQ1**) As reported in Figure 6(c), the higher recall indicates that our method is able to identify majority ground-truth bundles hidden in the sessions, whereas the relatively low coverage suggests that not all categories within ground-truth bundles are discovered, showing the necessity of bundle completion. Figure 7 visualizes three sessions and the corresponding detected bundles, where we can find that our method can help detect interesting patterns, e.g., ['Cleaning & Repair', 'Watches']. (**RQ2**) It

[1]Detailed parameter tuning and settings for all tasks are reported in our GitHub.



(a) Electronic (b) Clothing (c) Food

Figure 9: Performance on bundle completion, where the short line on each bar denote the standard deviation.

Table 5: Performance on bundle ranking, where the best results are highlighted in bold; *w/o pre* means without pre-training.

	Metrics	ItemKNN	BPRMF	BPRMF$_{w/o\ pre}$	DAM	DAM$_{w/o\ pre}$	AttList	AttList$_{w/o\ pre}$	GCN	GCN$_{w/o\ pre}$	BGCN	BGCN$_{w/o\ pre}$
Electronic	HR@1	0.032±0.000	0.026±0.005	0.001±0.002	0.013±0.009	0.001±0.001	0.051±0.045	0.022±0.012	0.133±0.029	0.067±0.021	**0.179±0.042**	0.065±0.016
	HR@5	0.084±0.000	0.088±0.015	0.014±0.005	0.076±0.027	0.005±0.005	0.134±0.092	0.069±0.032	0.300±0.038	0.211±0.043	**0.388±0.052**	0.176±0.025
	HR@10	0.154±0.000	0.141±0.013	0.043±0.013	0.152±0.035	0.015±0.008	0.171±0.107	0.130±0.038	0.382±0.038	0.316±0.050	**0.522±0.044**	0.282±0.035
	NDCG@1	0.032±0.000	0.026±0.005	0.001±0.002	0.013±0.009	0.001±0.001	0.051±0.045	0.022±0.012	0.133±0.029	0.067±0.021	**0.179±0.042**	0.065±0.016
	NDCG@5	0.058±0.000	0.057±0.009	0.008±0.003	0.044±0.018	0.003±0.002	0.094±0.068	0.046±0.021	0.219±0.032	0.141±0.031	**0.287±0.045**	0.122±0.017
	NDCG@10	0.081±0.000	0.074±0.007	0.017±0.005	0.068±0.018	0.006±0.003	0.106±0.073	0.066±0.024	0.245±0.031	0.174±0.032	**0.331±0.042**	0.156±0.018
Clothing	HR@1	0.021±0.000	0.011±0.004	0.002±0.002	0.017±0.021	0.000±0.000	0.030±0.025	0.027±0.009	0.182±0.015	0.081±0.022	**0.255±0.033**	0.058±0.011
	HR@5	0.074±0.000	0.055±0.009	0.015±0.006	0.075±0.054	0.001±0.002	0.085±0.049	0.073±0.014	0.336±0.020	0.233±0.052	**0.447±0.047**	0.151±0.016
	HR@10	0.139±0.000	0.107±0.013	0.048±0.013	0.167±0.066	0.002±0.002	0.138±0.066	0.135±0.024	0.431±0.024	0.350±0.037	**0.568±0.059**	0.242±0.021
	NDCG@1	0.021±0.000	0.011±0.004	0.002±0.002	0.017±0.021	0.000±0.000	0.030±0.025	0.027±0.009	0.182±0.015	0.081±0.053	**0.255±0.033**	0.058±0.011
	NDCG@5	0.049±0.000	0.033±0.006	0.008±0.004	0.046±0.037	0.000±0.001	0.057±0.037	0.050±0.009	0.262±0.017	0.158±0.037	**0.354±0.038**	0.104±0.013
	NDCG@10	0.070±0.000	0.049±0.006	0.018±0.006	0.075±0.041	0.001±0.001	0.074±0.042	0.070±0.012	0.293±0.018	0.196±0.044	**0.394±0.043**	0.134±0.014
Food	HR@1	0.055±0.000	0.016±0.004	0.001±0.001	0.012±0.011	0.001±0.002	0.058±0.045	0.025±0.011	0.170±0.022	0.066±0.023	**0.201±0.064**	0.053±0.025
	HR@5	0.133±0.000	0.081±0.012	0.019±0.007	0.077±0.039	0.003±0.004	0.122±0.079	0.089±0.029	0.336±0.026	0.205±0.067	**0.405±0.066**	0.161±0.024
	HR@10	0.188±0.000	0.141±0.019	0.059±0.014	0.163±0.046	0.004±0.004	0.169±0.094	0.149±0.040	0.439±0.026	0.314±0.085	**0.526±0.053**	0.251±0.028
	NDCG@1	0.055±0.000	0.016±0.004	0.001±0.001	0.012±0.011	0.001±0.002	0.058±0.045	0.025±0.011	0.170±0.022	0.066±0.023	**0.201±0.064**	0.053±0.025
	NDCG@5	0.095±0.000	0.049±0.007	0.010±0.003	0.043±0.025	0.002±0.002	0.091±0.064	0.057±0.020	0.257±0.020	0.136±0.044	**0.306±0.064**	0.106±0.023
	NDCG@10	0.113±0.000	0.068±0.008	0.022±0.005	0.071±0.026	0.002±0.002	0.107±0.068	0.076±0.023	0.291±0.020	0.171±0.049	**0.345±0.060**	0.134±0.022

Table 6: Challenges and opportunities for the defined tasks.

Tasks	Challenges	Opportunities
Detection	• Sparse item-bundle affinity records • Detect bundles with consistent yet diverse items • Detect and evaluate new bundles	• Use side information, e.g., category hierarchy [53], to detect bundles with different granularities • Use techniques, e.g., self-attention, to maintain item consistency within bundles; and e.g., DPP [3], to maintain item diversity within bundles • Use semi-supervised learning [55] for bundle detection • Debug noise items [66] within sessions to form bundles • Design crowd-sourcing tasks for new bundle evaluation
Completion	• Sparse item-bundle affinity records • Complete bundles with new items • Learn accurate context representations for bundles	• Use data augmentation techniques [36] to address the sparsity issue • Leverage side information, e.g., knowledge graph [52] for completion with (new) items • Design e.g., multimodal systems [32, 54], to consider item category, title and images, ect. • Use techniques, e.g., CVAE [33], to consider bundle intents as conditions
Ranking	• Sparse user-bundle interaction records • Rank new bundles for users • Complex relations among users, items and bundles • Learn accurate user, item and bundle representations	• Leverage user-item interaction records as complements • Use meta-learning [12] to transfer knowledge from multiple source domains to the target domain • Use graph-structured data (e.g., bipartite graph [58], knowledge graph [52, 57] and hypergraph [64]) to represent the relations among users, items and bundles • Use techniques, e.g., GCN [58], hypergraph convolution and attention [4] to model high order relations • Use self-supervised learning on graphs [60] to reinforce representation learning via self-discrimination

is easy to infer only partial predicted patterns match the ground-truth according to the results on Precision. This naturally leads us to ask a question: *are the predicted but non-hit bundle patterns reasonable*? To this end, we randomly sample 200 such patterns from each domain where three examples are visualized in Figure 8. Through a careful manual check, we find that 79.5%, 92.5%, and 73.5% of them are reasonable bundles in Electronic, Clothing and Food, respectively. This demonstrates such pattern mining indeed facilitates the generation of high-quality new bundles.

5.2 Bundle Completion

For each domain, we split the collected bundles in Table 2 into training, validation and test sets with the ratio of 8:1:1. Our current study considers the scenario, viz., removing one item from each bundle and deem the rest as context. Consequently, the goal is to complete the missing item based on the context for each bundle as accurate as possible. For a solid evaluation, we expand the test set by taking one different item from a bundle as the missing one,

and the rest as context each time. As such, one test bundle b will be expanded to $|b|$ samples. To improve test efficiency, for each test bundle, we pair 99 randomly-sampled items with its missing (ground-truth) item, and then rank the 100 items for completion. We adopt the widely-used Hit@N and $NDCG@N$ as evaluate metrics. Generally, higher metric values indicate better results. We test each method ten times, and report the results with $(mean \pm SD)$.

Analysis and Insights. Figure 9 presents the results. Overall, as the size of the candidate list (N) increases, the values of HR and NDCG consistently go up for all methods across the three domains. This is natural as larger N enhances the probability of correct items being selected for completion. (**RQ3**) Comparing different methods, the factorization-based BPRMF outperforms the memory-based ItemKNN, whilst being defeated by the neural-based VAE in most cases, suggesting the superiority of the generative model VAE on capturing bundle context for more accurate completion. (**RQ4**) Regarding the two VAE variants, concat-VAE performs slightly better than mean-VAE, which indicates concatenation can preserve more useful information for bundle completion. (**RQ5**) Due to space limitation, Figure 9 only reports the results with pre-trained representations, which far exceed those without pre-trained representations.

5.3 Bundle Ranking

We use the annotated user-bundle interaction data in the three domains in Table 2. Leave-one-out is adopted as the evaluation protocol, viz., for each user, we randomly hold-out one bundle for validation, one bundle for test, and the rest for training. To boost test efficiency, we pair 99 randomly-sampled bundles that are not interacted by the target user with the ground-truth bundle, and rank the 100 bundles for recommendation. Hit@N and $NDCG@N$ are the evaluation metrics. We test each method ten times and report the results in the format of $(mean \pm SD)$[2].

Analysis and Insights. Table 5 shows the results, where interesting findings can be noted. (**RQ6**) ItemKNN, though simple, outperforms BPRMF and even the attention-based DAM. This is in line with the conclusion drawn in [13], where traditional recommenders may defeat neural models with well-tuned parameters. (**RQ7**) Regarding attention-based methods, AttList achieves better performance than DAM due to (a) it learns hierarchical user preference via both item- and bundle-level aggregation; and (b) it adopts the self-attention mechanism to refine the item/bundle representation by fusing the consistency of neighboring items/bundles before aggregation. (**RQ8**) Graph-based neural methods (GCN and BGCN) perform better than others, and BGCN consistently achieves the best performance. It helps confirm (a) the necessity of capturing complex relations among users, items and bundles with the heterogeneous graph; and (b) the efficacy of convolutional propagation on better representation learning. Such a compatible combination (i.e. graph as data structure and GCN as technical support) greatly assists in mitigating the scarcity of user-bundle interactions. (**RQ9**) Generally, pre-training helps ensure better performance, for example, BPRMF outperforms $\text{BPRMF}_{w/o\ pre}$, which is consistent with the conclusion drawn in existing studies [29].

[2]In the current study, we choose to focus on offline evaluation. Future exploration on online evaluation, e.g., user engagements, can use other metrics such as CTR [23].

5.4 Discussion

Table 6 presents the challenges and opportunities for the defined tasks. As shown, data sparsity is one major challenge confronted by all tasks. To this end, we perform pattern mining at category level for bundle detection; recurrently sample items to expand test set for bundle completion; and design a pre-training strategy for model initialization. While, the presence of such a challenge also brings in new opportunities for other potential solutions, such as fusing side information (e.g. category hierarchy [53] and knowledge graph [52, 57]), or employing semi-supervised learning [55] for further enhancement. Beyond that, the necessity of capturing complex relations among users, products and bundles for bundle ranking indicates room for further exploration, e.g., hypergraph [64] may help better model the high order connections between users and bundles transit by items for boosted performance.

6 CONCLUSION AND FUTURE WORK

This paper seeks for more fundamental contributions in the area of bundle recommendation. We first delicately design a crowdsourcing task to construct high-quality bundle datasets with rich meta data, particularly the bundle intents, thus facilitating a deeper understanding of the rationale behind product bundling. With such data, we then define a series of interrelated tasks, and perform extensive experiments where we critically examine and analyze the strengths and weaknesses of the state-of-the-art. We draw insights from the experiments and point out both challenges and opportunities for future research. In a nutshell, our study delivers new data sources, opens up new research directions and provides useful guidance to promote product bundling in real-world platforms.

For the future work, we plan to explore possible solutions for bundle explanation and auto-naming; meanwhile we also intent to extend the datasets in a larger scale with more domains involved.

ACKNOWLEDGMENTS

We greatly acknowledge the support of National Natural Science Foundation of China (Grant No. 71771141 and 72192832), and the Natural Science Foundation of Shanghai (Grant No. 21ZR1421900). This research is also supported by the Centre for Frontier AI Research, Institute of High Performance Computing, A*STAR, and the TU Delft AI Lab Design@Scale.

REFERENCES

[1] Gediminas Adomavicius and YoungOk Kwon. 2011. Improving Aggregate Recommendation Diversity Using Ranking-based Techniques. *TKDE* 24, 5 (2011), 896–911.

[2] Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast Algorithms for Mining Association Rules. In *VLDB*, Vol. 1215. 487–499.

[3] Jinze Bai, Chang Zhou, Junshuai Song, Xiaoru Qu, Weiting An, Zhao Li, and Jun Gao. 2019. Personalized Bundle List Recommendation. In *WWW*. 60–71.

[4] Song Bai, Feihu Zhang, and Philip HS Torr. 2021. Hypergraph Convolution and Hypergraph Attention. *Pattern Recognition* 110 (2021), 107637.

[5] Yannis Bakos and Erik Brynjolfsson. 1999. Bundling Information Goods: Pricing, Profits, and Efficiency. *Management Science* 45, 12 (1999), 1613–1630.

[6] Arash Beheshtian-Ardakani, Mohammad Fathian, and Mohammadreza Gholamian. 2018. A Novel Model for Product Bundling and Direct Marketing in E-commerce based on Market Segmentation. *Decision Science Letters* 7, 1 (2018), 39–54.

[7] Moran Beladev, Lior Rokach, and Bracha Shapira. 2016. Recommender Systems for Product Bundling. *KBS* 111 (2016), 193–206.

[8] Adrian Boteanu, Emily Dutile, Adam Kiezun, and Shay Artzi. 2020. Subjective Search Intent Predictions using Customer Reviews. In *CHIIR*. 303–307.

[9] Da Cao, Liqiang Nie, Xiangnan He, Xiaochi Wei, Shunzhi Zhu, and Tat-Seng Chua. 2017. Embedding Factorization Models for Jointly Recommending Items and User Generated Lists. In *SIGIR*. 585–594.

[10] Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Bundle Recommendation with Graph Convolutional Networks. In *SIGIR*. 1673–1676.

[11] Liang Chen, Yang Liu, Xiangnan He, Lianli Gao, and Zibin Zheng. 2019. Matching User with Item Set: Collaborative Bundle Recommendation with Deep Attention Network. In *IJCAI*. 2095–2101.

[12] Yudong Chen et al. 2021. Curriculum Meta-learning for Next POI Recommendation. In *KDD*. 2692–2702.

[13] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *RecSys*. 101–109.

[14] Qilin Deng, Kai Wang, Minghao Zhao, Zhene Zou, Runze Wu, Jianrong Tao, Changjie Fan, and Liang Chen. 2020. Personalized Bundle Recommendation in Online Games. In *CIKM*. 2381–2388.

[15] Ting Deng, Wenfei Fan, and Floris Geerts. 2013. On the Complexity of Package Recommendation Problems. *SIAM J. Comput.* 42, 5 (2013), 1940–1986.

[16] Paolo Dragone, Giovanni Pellegrini, Michele Vescovi, Katya Tentori, and Andrea Passerini. 2018. No More Ready-Made Deals: Constructive Recommendation for Telco Service Bundling. In *RecSys*. 163–171.

[17] Yan Fang, Xinyue Xiao, Xiaoyu Wang, and Huiqing Lan. 2018. Customized Bundle Recommendation by Association Rules of Product Categories for Online Supermarkets. In *DSC*. 472–475.

[18] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. 2015. Understanding Malicious Behavior in Crowdsourcing Platforms: The Case of Online Surveys. In *CHI*. 1631–1640.

[19] Gary J Gaeth, Irwin P Levin, Goutam Chakraborty, and Aron M Levin. 1991. Consumer Evaluation of Multi-product Bundles: An Information Integration Analysis. *Marketing Letters* 2, 1 (1991), 47–57.

[20] Robert Garfinkel, Ram Gopal, Arvind Tripathi, and Fang Yin. 2006. Design of a Shopbot and Recommender System for Bundle Purchases. *DSS* 42, 3 (2006), 1974–1986.

[21] Xinyu Ge, Yousha Zhang, Yu Qian, and Hua Yuan. 2017. Effects of Product Characteristics on the Bundling Strategy Implemented by Recommendation Systems. In *ICSSSM*. 1–6.

[22] Yong Ge, Hui Xiong, Alexander Tuzhilin, and Qi Liu. 2014. Cost-aware Collaborative Filtering for Travel Tour Recommendations. *TOIS* 32, 1 (2014), 1–31.

[23] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *IJCAI*. 1725–1731.

[24] Liu Guo-rong and Zhang Xi-zheng. 2006. Collaborative Filtering based Recommendation System for Product Bundling. In *ICMSEM*. 251–254.

[25] Ward Hanson and R Kipp Martin. 1990. Optimal Bundle Pricing. *Management Science* 36, 2 (1990), 155–174.

[26] Junheng Hao, Tong Zhao, Jin Li, Xin Luna Dong, Christos Faloutsos, Yizhou Sun, and Wei Wang. 2020. P-Companion: A Principled Framework for Diversified Complementary Product Recommendation. In *CIKM*. 2517–2524.

[27] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*. 507–517.

[28] Ruining He, Charles Packer, and Julian McAuley. 2016. Learning Compatibility Across Categories for Heterogeneous Item Recommendation. In *ICDM*. 937–942.

[29] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. 173–182.

[30] Yun He, Jianling Wang, Wei Niu, and James Caverlee. 2019. A Hierarchical Self-Attentive Model for Recommending User-Generated Item Lists. In *CIKM*. 1481–1490.

[31] Yun He, Yin Zhang, Weiwen Liu, and James Caverlee. 2020. Consistency-Aware Recommendation for User-Generated Item List Continuation. In *WSDM*. 250–258.

[32] Murium Iqbal, Adair Kovac, and Kamelia Aryafar. 2018. A Multimodal Recommender System for Large-Scale Assortment Generation in E-commerce. In *Proceedings of ACM SIGIR Workshop on eCommerce*.

[33] Diederik P Kingma and Max Welling. 2014. Auto-encoding Variational Bayes. In *ICLR*.

[34] Pigi Kouki, Ilias Fountalis, Nikolaos Vasiloglou, Nian Yan, Unaiza Ahsan, Khalifeh Al Jadda, and Huiming Qu. 2019. Product Collection Recommendation in Online Retail. In *RecSys*. 486–490.

[35] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-Based Recommendation. In *CIKM*. 1419–1428.

[36] Yang Li et al. 2019. Context-aware Attention-based Data Augmentation for POI Recommendation. In *ICDEW*. 177–184.

[37] Zhi Li, Bo Wu, Qi Liu, Likang Wu, Hongke Zhao, and Tao Mei. 2020. Learning the Compositional Visual Coherence for Complementary Recommendations. In *IJCAI*. 3536–3543.

[38] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW*. 689–698.

[39] Yusan Lin, Maryam Moosaei, and Hao Yang. 2020. OutfitNet: Fashion Outfit Recommendation with Attention-Based Multiple Instance Learning. In *WWW*. 77–87.

[40] Guannan Liu, Yanjie Fu, Guoqing Chen, Hui Xiong, and Can Chen. 2017. Modeling Buying Motives for Personalized Product Bundle Recommendation. *TKDD* 11, 3 (2017), 1–26.

[41] Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. 2011. Personalized Travel Package Recommendation. In *ICDM*. 407–416.

[42] Yidan Liu, Min Xie, and Laks V.S. Lakshmanan. 2014. Recommending User Generated Item Lists. In *RecSys*. 185–192.

[43] Fionn Murtagh and Pedro Contreras. 2017. Algorithms for Hierarchical Clustering: An Overview, II. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7, 6 (2017), e1219.

[44] Katja Niemann and Martin Wolpers. 2013. A New Collaborative Filtering Approach for Increasing the Aggregate Diversity of Recommender Systems. In *KDD*. 955–963.

[45] Aditya Parameswaran, Petros Venetis, and Hector Garcia-Molina. 2011. Recommendation Systems with Complex Constraints: A Course Recommendation Perspective. *TOIS* 29, 4 (2011), 1–33.

[46] Apurva Pathak, Kshitiz Gupta, and Julian McAuley. 2017. Generating and Personalizing Bundle Recommendations on Steam. In *SIGIR*. 1073–1076.

[47] Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2016. Recommending Packages to Groups. In *ICDM*. 449–458.

[48] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. 452–461.

[49] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *WWW*. 285–295.

[50] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2017. Fairness in Package-to-Group Recommendations. In *WWW*. 371–379.

[51] Stefan Stremersch and Gerard J Tellis. 2002. Strategic Bundling of Products and Prices: A New Synthesis for Marketing. *Journal of Marketing* 66, 1 (2002), 55–72.

[52] Zhu Sun, Qing Guo, Jie Yang, Hui Fang, Guibing Guo, Jie Zhang, and Robin Burke. 2019. Research Commentary on Recommendations with Side Information: A survey and Research Directions. *ECRA* 37 (2019), 100879.

[53] Zhu Sun, Jie Yang, Jie Zhang, and Alessandro Bozzon. 2017. Exploiting both Vertical and Horizontal Dimensions of Feature Hierarchy for Effective Recommendation. In *AAAI*, Vol. 31.

[54] Quoc-Tuan Truong, Aghiles Salah, and Hady Lauw. 2021. Multi-Modal Recommender Systems: Hands-On Exploration. In *RecSys*. 834–837.

[55] Hen Tzaban, Ido Guy, Asnat Greenstein-Messica, Arnon Dagan, Lior Rokach, and Bracha Shapira. 2020. Product Bundle Identification Using Semi-Supervised Learning. In *SIGIR*. 791–800.

[56] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. 2019. Modeling Multi-purpose Sessions for Next-Item Recommendations via Mixture-Channel Purpose Routing Networks. In *IJCAI*. 3771–3777.

[57] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *KDD*. 950–958.

[58] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*. 165–174.

[59] Zihan Wang, Ziheng Jiang, Zhaochun Ren, Jiliang Tang, and Dawei Yin. 2018. A Path-constrained Framework for Discriminating Substitutable and Complementary Products in E-commerce. In *WSDM*. 619–627.

[60] Jiancan Wu et al. 2021. Self-supervised Graph Learning for Recommendation. In *SIGIR*. 726–735.

[61] Min Xie, Laks VS Lakshmanan, and Peter T Wood. 2010. Breaking Out of the Box of Recommendations: from Items to Packages. In *RecSys*. 151–158.

[62] Min Xie, Laks VS Lakshmanan, and Peter T Wood. 2014. Generating Top-k Packages via Preference Elicitation. *VLDB Endowment* 7, 14 (2014), 1941–1952.
[63] Manjit S Yadav. 1994. How Buyers Evaluate Product Bundles: A Model of Anchoring and Adjustment. *Journal of Consumer Research* 21, 2 (1994), 342–353.
[64] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. 2019. Revisiting User Mobility and Social Relationships in LBSNs: a Hypergraph Embedding Approach. In *WWW*. 2147–2157.
[65] De-Nian Yang, Wang-Chien Lee, Nai-Hui Chia, Mao Ye, and Hui-Ju Hung. 2012. On Bundle Configuration for Viral Marketing in Social Networks. In *CIKM*. 2234–2238.
[66] Jie Yang et al. 2019. Scalpel-CD: Leveraging Crowdsourcing and Deep Probabilistic Modeling for Debugging Noisy Training Data. In *WWW*. 2158–2168.
[67] Jie Yang, Judith Redi, Gianluca Demartini, and Alessandro Bozzon. 2016. Modeling Task Complexity in Crowdsourcing. In *HCOMP*. 249–258.
[68] Markus Zanker, Markus Aschinger, and Markus Jessenitschnig. 2010. Constraint-based Personalised Configuring of Product and Service Bundles. *International Journal of Mass Customisation* 3, 4 (2010), 407–425.
[69] Tao Zhu, Patrick Harrington, Junjun Li, and Lei Tang. 2014. Bundle Recommendation in Ecommerce. In *SIGIR*. 657–666.