Discovery of associated consumer demands: Construction of a co-demanded product network with community detection

Wu, Jiacong; Wang, Yu; Shafiee, Sara; Zhang, Dongsong

Published in:
Expert Systems with Applications

Link to article, DOI:
10.1016/j.eswa.2021.115038

Publication date:
2021

Document Version
Peer reviewed version

Citation (APA):
Discovery of associated consumer demands: Construction of a co-demanded product network with community detection

Jiacong Wu, Yu Wang, Sara Shafiee, Dongsong Zhang

PII: S0957-4174(21)00479-6
DOI: https://doi.org/10.1016/j.eswa.2021.115038
Reference: ESWA 115038

To appear in: Expert Systems with Applications

Received Date: 3 December 2020
Revised Date: 17 March 2021
Accepted Date: 10 April 2021


This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2021 Published by Elsevier Ltd.
Discovery of associated consumer demands: Construction of a co-demanded product network with community detection

Jiacong Wu*, Yu Wanga,b,*, Sara Shafieec, Dongsong Zhangd

a School of Management, Jinan University, Guangzhou, 510632, China
skywu@stu2017.jnu.edu.cn (J.W.); twygs@jnu.edu.cn (Y.W.)

b International Business School, Jinan University, Zhuhai, 519070, China
c Department of Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby 2800, Denmark
sashaf@mek.dtu.dk (S.S.)
d Belk College of Business, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC, 28223-0001, USA
dzhang15@uncc.edu (D.Z.)

* Correspondence: Email: twygs@jnu.edu.cn; Phone numbers: 0086-13570105201

Jiacong Wu: 0000-0002-2579-3372
Yu Wang: 0000-0001-7790-206X
Abstract: Some consumers have various product demands that are associated with each other. Although there are methods for discovering the associations among co-purchased products, they have limitations, including redundancy of the extracted association rules, the potential to miss novel and interesting associations among co-demanded products hidden in shopping behaviors, and neglect of several important influential factors. To give effective product recommendations, it is necessary and beneficial to discover the associations among the products co-demanded by the same consumers in a short period based on the consumers’ shopping behaviors. Therefore, this paper proposes a novel model for discovering associated consumer demands based on a co-demanded product network. First, the model identifies each consumer’s product demands and calculates their intensity based on various online shopping behaviors. Second, the model constructs a co-demanded product network based on the products demanded by the same consumers within a short period. The model also considers several important factors, neglected in the literature that can improve the detection of associations among co-demanded products, including the time interval between two product demands from the same consumers, the popularity of each demanded product, and the number of product demands from each consumer. Third, the model uses an algorithm for the detection of overlapping communities to identify the tightly connected co-demanded products within the network as communities of associated consumer demands, and ranks them based on their information density. We use a real dataset of a well-known e-commerce platform to validate the proposed model. The results show that the model can detect more modular, diverse, practical, and reliable communities of associated products compared with the existing network analysis–based market basket analysis methods.

Keywords: associated consumer demands; co-demanded products; product networks; community detection

1. Introduction

Consumer demands represent the products that consumers have purchased or are interested in purchasing; some of these demands are closely associated with one another (Chen et al., 2018). For example, consumers who want plants on their balconies are interested in purchasing not only potted plants but also beautiful flowerpots and topsoil. Two products that are or will be purchased together by the same consumers within a short period are called co-demanded products; it is reasonable to expect an association between such products. Co-demanded products that are associated with each other are defined as associated consumer demands in this paper.

Although consumers can access extensive product information online, they still need to spend much time searching and comparing the information of various products to meet their requirements, resulting in poor consumer purchase experiences (Faridizadeh et al., 2018; Kim et al., 2012). Some consumers do not even realize their associated demands, and thus, need recommendations. For example, consumers who want to purchase a tablet computer like an iPad for use in taking classroom notes probably need an Apple pencil for quick recording. Therefore, it is beneficial for e-commerce platforms and online retailers to automatically discover and take advantage of associated consumer demands to promote up-selling and cross-selling (Kim et al., 2012; Videla-Cavieres & Rios, 2014), attain higher consumer satisfaction, and improve the shopping experience of consumers via product recommendations (Smith & Linden, 2017).

The discovery of associations among products has been investigated by previous researchers...
and applied in practice for a long time. However, most previous works used the market basket analysis (MBA) method to discover associations among co-purchased products, representing the relationships among the products that the same consumers purchased in the same transaction or within a short period. The MBA method analyzes the transactional data of the consumer’s purchases to extract associations among the co-purchased products, with the aim of promoting sales through up-selling and cross-selling (Videla-Cavières & Ríos, 2014). One of the most common and well-known MBA methods is association rule mining (ARM; Agrawal et al., 1993), which investigates the pattern of products that consumers frequently co-purchase in the same transactions and extracts association rules among such products based on confidence and support measures. However, given certain levels of support and confidence, ARM algorithms may produce a huge number of association rules, with many being either redundant or simply obvious (i.e., lack of serendipity). It is often difficult to identify interesting association rules based on certain interestingness measures (Raeder & Chawla, 2011) unless there is expert knowledge that helps select appropriate measures for identifying interesting association rules in different scenarios.

To address the limitations of ARM used in MBA, some researchers have employed another method—the network analysis–based MBA method—which constructs a product network based on transactional data and detects tightly connected products as a community to analyze the relationships among products purchased together. Detecting product communities within a product network can reduce the necessity to search through a massive list of potential association rules. Moreover, it assists to uncover relationships among products that ARM may find difficult to detect, and alleviate redundancy in the discovered relationships among products by discovering larger relationships among groups of products (Raeder & Chawla, 2011). Many researchers have constructed co-purchased product networks based on the products co-purchased in the same transactions and detected communities of products with various algorithms (e.g., Qi et al., 2013; Raeder & Chawla, 2009, 2011; Videla-Cavières & Ríos, 2014).

Nowadays, online shopping is increasingly popular. Consumers no longer need to purchase many products in one transaction because they can shop online wherever they are and whenever they have an internet connection. As such, constructing a co-purchased product network based only on the products co-purchased in the same transactions can be far less effective than it once was. Accordingly, some studies that constructed a co-purchased product network considered the products co-purchased by the same consumers within a short period rather than only in the same transactions (e.g., Dhar et al., 2014; Faridizadeh et al., 2018; Kim et al., 2012; Lismont et al., 2018).

The extension of the network analysis–based MBA method still has limitations, and the main gaps in the current literature are as follows:

First, previous studies discovered associations among products based only on consumer purchase behavior; therefore, some novel, valuable, and interesting associations among co-demanded products are neglected. Consumers usually carry out various online shopping behaviors before purchasing (Xu et al., 2017). Valuable associations among products are potentially hidden in the behaviors of consumers in addition to purchase behavior.

Second, some previous studies mentioned the influences of specific factors on the associations or similarity among products, as well as on network construction and analysis, such as the time interval between two product purchases of the same consumers, the popularity of each product, and the number of products that each consumer rates (Smith & Linden, 2017; Yan et al,
However, these factors have never been considered in the existing network analysis–based MBA methods, which may significantly influence the effectiveness of determining associations among products.

To address the current research gaps, this study aimed to answer the following research questions:

1. How can we identify consumer demands and measure their intensity to indicate consumers’ likelihood of purchasing the demanded products, based on consumers’ online shopping behaviors?
2. How can we construct a co-demanded product network based on consumer demands and influential factors to discover the associations among products?
3. How can we discover associated consumer demands using a co-demanded product network and community detection?
4. How effectively can the co-demanded product network approach discover valuable and reliable associations among products compared with the co-purchased product network approach and a co-demanded product network approach that does not consider influential factors?

This study contributes to the literature in the following ways:

First, it proposes a new network analysis method based on co-demanded rather than co-purchased products, which will enable the discovery of more diverse and serendipitous associations among products. The consumer demands are identified and their intensity measured via statistical analysis of each consumer’s varied types of online shopping behaviors. Any two products co-demanded by the same consumers in a short period are connected with an edge to construct a co-demanded product network.

Second, the proposed method considers several important influential factors to improve the accuracy of the associations discovered among products, including:

- The time interval between two product demands from the same consumers. Although previous research have realized the importance of time for improving the quality of recommendations (Jia & Liu, 2015; Smith & Linden, 2017), how to consider the time factor in the network analysis-based MBA method for more accurate discovery of associations among products needs further investigation.
- The popularity of each product. Previous research on network analysis-based MBA methods neglected the fact that those higher weighted edges associated with a few very popular nodes can dominate analyses in the network (Zhang et al., 2016).
- The number of product demands from each consumer. Previous research mentioned the importance of the number of products that each consumer rates for calculation of item similarity (Yan et al., 2017). However, investigations to extend this factor for more accurate discovery of associations among products have not been considered.

Finally, the study illustrates that the proposed model can improve the diversity, accuracy and effectiveness of discovery of associations among products compared with the co-purchased product network approach or co-demanded product network approach, which does not consider influential factors. We selected a real dataset from a well-known e-commerce platform to evaluate the feasibility and validity of the proposed model, comparing it with the existing network analysis–based MBA methods. Hence, this research proposes and evaluates a novel model for the discovery of associated consumer demands based on a co-demanded product network considering
several important but neglected influential factors. The results show that the model can detect more modular, diverse, practical, and reliable communities of associated products compared with the existing network analysis-based MBA methods.

The rest of this paper is organized as follows: The next section reviews the related works. Section 3 introduces the proposed method for the discovery of associated consumer demands; it then delineates the detailed steps of the model construction and how the three influential factors are incorporated in the model to adjust the weights of the network edges. The evaluation methods and results based on a real dataset are described in sections 4 and 5, respectively. Finally, the major findings and limitations of this study, as well as future research directions, are discussed in section 6.

2. Related works

Researchers have highlighted different methods of discovering associations among consumers’ purchased products for MBA. This section introduces two streams of related works on MBA, namely, ARM-based MBA and network analysis–based MBA. The algorithms for the detection of overlapping communities in the network analysis methods are also reviewed in this section.

2.1 Association rule mining–based market basket analysis

ARM has been successfully applied in various contexts, such as customer relationship management, stock markets, health care, and census research (Pandya & Rustom, 2017; Shridhar & Parmar, 2017; Yazgana & Kusakci, 2016). It is one of the most popular methods used in MBA for extracting association rules among products co-purchased in the same transactions from large transactional datasets based on the minimum support and confidence thresholds. The Apriori algorithm (Agrawal et al., 1993) is the first ARM algorithm that iteratively uses two steps, “join” and “prune,” to find and reduce the candidate frequent itemsets and search space. However, although the Apriori algorithm reduces the size of the candidate itemsets to some extent based on an anti-monotone Apriori heuristic, the number of candidate itemsets can still be large due to the huge number of items in the database. The Frequent Pattern Growth algorithm (Han et al., 2000) improves the Apriori algorithm because it does not require candidate set generation; it represents a database in the form of a tree structure called a frequent pattern tree. This tree structure maintains the associations among itemsets and reduces the cost of searching for frequent itemsets.

In addition to the two common algorithms for ARM, a number of algorithms have been proposed to improve ARM’s efficiency (Badhon et al., 2019; Fournier-Viger et al., 2017; Ghafari & Tjortjis, 2019) and to extend the notion of association rules, including categorical association rules (Kabir et al., 2017), quantitative association rules (Can & Alatas, 2017), fuzzy association rules (Verma & Thakur, 2017), and rare association rules (Borah & Nath, 2018). Although ARM has been improved and applied to MBA for decades, it still has limitations in mining association rules from co-purchased products (Kaur & Madan, 2014; Raeder & Chawla, 2011). One of the most serious limitations is that many extracted association rules are either redundant or obvious. Here, association rules are considered redundant if they have the same support and confidence level as some more general rules (Zaki, 2000). For example, if a dataset supports the two association rules \( A \rightarrow B \) and \( AC \rightarrow B \), and such association rules have the same values in both the support and confidence measures, then the latter rule is meaningless and can be pruned due to
redundancy. One of the common solutions to this problem is to extract the maximal and closed frequent itemsets among frequent patterns to reduce the redundant association rules (Momtaz et al., 2019; Sutha & Dhanaseelan, 2017; Zaki, 2000). An itemset is maximal at $a\%$ support if none of its supersets have at least $a\%$, and an itemset is closed if none of its supersets have the same support as it has. This method’s efficiency is limited in practice because it usually prunes only a few redundant rules (Raeder & Chawla, 2011). Another solution is to filter the interesting rules from the candidate association rules based on either objective or subjective interestingness measures (Gan et al., 2018; Sethi & Shekar, 2019; Somyanonthanakul & Theeramunkong, 2020). However, such measures rank association rules ambiguously and differently, and many of them are considered a “deviation from independence”; that is, an association rule may be ranked high by one interestingness measure but ranked low by another (Raeder & Chawla, 2011). Because it is difficult to rank association rules efficiently with various measures, the effectiveness of redundant rule pruning is uncertain.

2.2 Network analysis–based market basket analysis

To address the limitations of ARM-based MBA, researchers have focused on another method—network analysis. Complex networks have been studied from several perspectives, such as belief percolation (Li & Wang, 2019), information fusion (Li et al., 2020c), and cooperation (Li et al., 2020b), to provide novel ideas and insights for understanding the structure and dynamics of complex systems. It is significant to construct a product network based on transactional data to reflect the relationships among products purchased together for MBA. We summarize two streams of network-analysis-based MBA methods in Table 1.

<table>
<thead>
<tr>
<th>Table 1. Summary of works using network analysis methods for MBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description of the method</td>
</tr>
<tr>
<td>Co-purchased product networks built based on the same transactions by the same consumers</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Co-purchased product networks built based on multiple transactions by the same consumers</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

One stream of research constructs co-purchased product networks based on the same transactions by the same consumers. Raeder and Chawla (2009, 2011) constructed a product network based on transactional data to complement ARM for MBA. In this network, it represented the products purchased by consumers as nodes, and built edges between any two products purchased together in the same transactions. An example of transactional data gathered on 1 day is

...
listed in Table 2, and a diagram of the construction of such a co-purchased product network based on the transactional data in Table 2 is shown in Figure 1(a).

<table>
<thead>
<tr>
<th>Transactions</th>
<th>Consumers</th>
<th>Date</th>
<th>Purchased products</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$u_1$</td>
<td>2020/10/1</td>
<td>$p_1, p_2$</td>
</tr>
<tr>
<td>2</td>
<td>$u_1$</td>
<td>2020/10/1</td>
<td>$p_3$</td>
</tr>
<tr>
<td>3</td>
<td>$u_2$</td>
<td>2020/10/1</td>
<td>$p_2$</td>
</tr>
<tr>
<td>4</td>
<td>$u_2$</td>
<td>2020/10/1</td>
<td>$p_3, p_4$</td>
</tr>
</tbody>
</table>

(a) Co-purchased product network built based on the same transactions by the same consumers

(b) Co-purchased product networks built based on multiple transactions by the same consumers

Figure 1. Demonstration of the construction of the two co-purchased product networks

To discover the relationships among products, tightly connected products within the network were identified as product communities using algorithms based on a modularity measure. Because communities can be arbitrarily large, they can represent relationships among co-purchased products much more expressively with less redundancy compared to the original association rules (Raeder & Chawla, 2011). Qi et al. (2013) generated a product network using a FOODMART sales dataset based on whether two products were in the same shopping basket. In addition, they analyzed the characteristics of the network, such as the degree distribution, average path lengths, and clustering coefficient. Videla-Cavieres and Ríos (2014) constructed a co-purchased product network that was the same as that constructed by Raeder and Chawla (2009), while they used algorithms for detecting overlapping communities to improve the detection of communities of co-purchased products. Different communities can be overlapping, meaning that a product may belong to more than one community. The two researchers also analyzed the stability of communities with the Jaccard index (Ríos & Videla-Cavieres, 2014).

As mentioned above, thanks to online shopping, consumers no longer need to purchase many products in one transaction. A dataset of online shopping behaviors of about 1 million consumers on Taobao.com, one of the largest and most popular e-commerce platforms in China, showed that only a small percentage of consumers bought more than one product in a single transaction. The number of products purchased in one transaction has decreased, resulting in far fewer possible associations to be discovered. Hence, constructing a product network based on co-purchased products in the same transactions may not always work. Another stream of research constructs co-purchased product networks based on multiple transactions by the same consumers in a short period. Kim et al. (2012) suggested expanding the network to include the relations among products purchased by the same consumers at different points within a time period. The products purchased by consumers are represented as nodes, and any two products purchased together by the same consumers within the specified period are connected with an edge. A diagram of the construction of such a co-purchased product network based on the transactional data in Table 2 is
shown in Figure 1(b). The original and expanded product networks differ in network characteristics and structure, and they may provide different useful product information from the same transaction data. Such an expanded product network approach has been applied to various business applications, such as demand prediction, inter-purchase time prediction, and recommendation (Dhar et al., 2014; Faridizadeh et al., 2018; Lismont et al., 2018).

2.3 Algorithms for the detection of overlapping communities in network analysis

A community in a network is defined as a group of nodes that are densely interconnected and sparsely connected to other parts of the network (Raghavan et al., 2007). In network analysis–based MBA, the tightly connected nodes of products within a network are further detected as product communities. This is done to concisely isolate associations among products and alleviate redundancy by discovering larger, more expressive relationships among groups of products (Raeder & Chawla, 2011; Videla-Cavieres & Ríos, 2014). Like clustering in data mining, community detection detects closely connected nodes with edges as communities (Faridizadeh et al., 2018). However, unlike the clustering technique, where the data are not in the form of relations, there are several relationships among the data of nodes in community detection. For example, in the detection of the communities of co-purchased products, transactional data are organized as nodes of products and relationships among them with weights if they are purchased together in the same transactions by a certain number of consumers (Videla-Cavieres & Ríos, 2014). Existing methods for community detection mainly rely on the optimization of quality functions that are defined according to the similarity between different nodes. Non-negative matrix factorization is a mainstream technology for community detection (Lee & Seung, 1999); identifiability, defined as the discrepancy between the network similarity matrix and the doubly stochastically normalized community incidence matrix, has recently been proposed to solve the limitations of this technology (Li et al., 2020a).

The detection of overlapping communities extends classic community detection by allowing the same nodes to be members of more than one community because many real-world social networks overlap, including interest networks, multimedia social networks, and biological networks (Huang et al., 2017; Najafi et al., 2016; Reihanian et al., 2018). Moreover, Xie et al. (2013) categorized algorithms for the detection of overlapping communities into the five following categories: clique percolation, line graph and link partitioning, local expansion and optimization, fuzzy detection, and agent-based and dynamical algorithms. They evaluated 14 algorithms for the detection of overlapping communities based on such criteria as normalized mutual information and omega index. The results showed that for both low- and high-overlapping-density networks, the Speaker-Listener Label Propagation algorithm (SLPA; Xie et al., 2011) offers better and relatively more stable performance than most of the other tested algorithms. Similarly, Amelio and Pizzuti (2014) classified algorithms for the detection of overlapping communities into six categories: node seeds and local expansion, clique expansion, link clustering, label propagation, other approaches, and dynamic networks. They reported the computational complexity of these categories of algorithms and showed that the label propagation approach is the most efficient category among them. Vieira et al. (2020) presented a comparative analysis of five state-of-the-art methods for the detection of overlapping communities, including CFinder (Palla et al., 2005), Bigclam (Yang & Leskovec, 2013), Demon (Coscia et al., 2014), Community Overlap PRopagation algorithm (COPRA; Gregory, 2010), and the SLPA. They were compared in terms of the structural properties of detected communities and some common
objective quality measures. This work highlights the importance of analyzing the topology of the community structure and confirms that there is not a best algorithm that solves a general community detection problem (Peel et al., 2017; Schaub et al., 2017).

3. Proposed model for the discovery of associated consumer demands based on a co-demanded product network

As stated in section 1, the discovery of associated consumer demands not only helps consumers reduce their search costs and improve their shopping experience but also assists online retailers to improve their product recommendations, stimulate consumer demands, and promote up-selling and cross-selling. However, the existing network analysis–based MBA methods neglect valuable or interesting associations among co-demanded products hidden in various shopping behaviors of consumers in addition to purchase. For example, consumers who purchase tablet computers for work or study purposes are usually interested in stylus pens for touchscreens, which can be recommended during shopping. However, as such items are not necessary or cheap, the consumers usually browse the information regarding them and add them to their favorite lists for possible future purchase. Hence, this association will be neglected if associations among products are identified based only on consumers’ purchase behaviors. Moreover, some important factors that may influence the effectiveness of the discovery of associations among products (Smith & Linden, 2017; Yan et al., 2017; Zhang et al., 2016) have not been considered in the existing network analysis–based MBA methods. To address these limitations, we propose a novel model for the discovery of associated consumer demands based on a co-demanded product network (DACCN). The model is shown in Figure 2.

![Figure 2. Proposed model for the discovery of associated consumer demands based on a co-demanded product network](image)

Although the proposed DACCN model in Figure 2 consists of two layers, the relationships among consumers are not considered in the model, and thus, the DACCN model constructs a single-layer network—a co-demanded product network—which is different from the classic multilayer networks including layers of consumers (users), knowledge, and products. Most of the classic multilayer networks are to clarify the mapping relationships among these layers for better knowledge management rather than the detection of associations among products, which is different from the aim of the proposed DACCN model. In most of these multilayer networks, the network of products is constructed based on the similarity among products or whether the products...
are mentioned, compared, purchased, or rated by the same consumers (users). However, unlike this scenario, the DACCN model represents the products demanded by the consumers as nodes and builds edges among the products co-demanded by the same consumers within a short period to construct a co-demanded product network. We consider various online shopping behaviors of consumers to build the edges among products. In addition, the weights of edges are set according to the intensity of consumer demands and further adjusted based on some important influential factors in the construction of the co-demanded product network.

The DACCN model works as follows. First, it identifies each consumer’s product demands and calculates their intensity as indicated by the thickness of the directed dotted lines in Figure 2. In addition to purchase behavior, other online shopping behaviors also reflect consumers’ product demands to different extents, such as viewing product pages, “favoriting” a product, and adding a product to one’s shopping cart. These extensive online shopping behaviors provide valuable information for inferring and ranking consumer demands and also enable the discovery of more diverse associations among products compared to considering only consumers’ purchase behaviors. As such, the DACCN model fully considers these online shopping behaviors when identifying consumers’ product demands. Afterwards, the model calculates the intensity of consumers’ product demands based on the weighted sum of the number of various types of consumer shopping behaviors associated with the products.

Second, the DACCN model constructs a co-demanded product network based on the products co-demanded by the same consumers within a short period. In the DACCN model, the products demanded by consumers are represented by nodes, and any two products co-demanded by the same consumers within a short period are connected to each other with an edge. Moreover, the weights of the edges are determined by multiplying the intensity of the connected product demands. For example, in Figure 2, products \( p_1 \) and \( p_2 \) are co-demanded by consumers \( u_1 \) and \( u_2 \) within a short period; the nodes of \( p_1 \) and \( p_2 \) are then connected to each other with one edge. The weight of the edge is determined by the intensity of the demands from consumers \( u_1 \) and \( u_2 \) for products \( p_1 \) and \( p_2 \), indicated by the thickness of the edge. More importantly, the model considers some important ignored factors that may make the discovery of the associations among co-demanded products more effective to adjust the edges’ weights. The factors considered includes the time interval between two product demands from the same consumers, the popularity of each product, and the number of product demands from each consumer.

Finally, the DACCN model further detects and forms tightly connected co-demanded products (which are densely interconnected and sparsely connected to other parts of the network) into product communities called communities of associated consumer demands. As a product may belong to more than one community, an algorithm is used for the detection of overlapping communities. For example, products \( p_1, p_2, \) and \( p_3 \) are tightly connected, and they are detected as a community of associated consumer demands; at the same time, \( p_1 \) is tightly connected to \( p_4 \), and the two are detected as another community. The model then ranks all the product communities using a community information density measure and calculates the similarity among the consumers based on the similarity of their demanded products, which can be used for more effective product recommendations. The flowchart of the proposed DACCN model is demonstrated in Figure 3, and the details of the proposed model are discussed in the next section.
3.1 Identification of consumers’ product demands

The DACCN model assumes that online shopping behaviors can reflect consumers’ interest (i.e., demand) in certain products. There are various online shopping behaviors, such as viewing a product page (click), “favoriting” a product (favorite), adding a product to one’s shopping cart (add), and purchasing a product (buy). Examples of these occurring within a short period are
shown in Table 3 ("u" refers to consumers, "p" refers to products).

<table>
<thead>
<tr>
<th>Click</th>
<th>Favorite</th>
<th>Add</th>
<th>Buy</th>
</tr>
</thead>
<tbody>
<tr>
<td>u₁₁</td>
<td>p₁₁</td>
<td>6</td>
<td>-</td>
</tr>
<tr>
<td>u₁₂</td>
<td>p₁₂</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>u₁₃</td>
<td>p₁₃</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>u₂₁</td>
<td>p₂₁</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>u₂₂</td>
<td>p₂₂</td>
<td>6</td>
<td>-</td>
</tr>
<tr>
<td>u₂₃</td>
<td>p₂₃</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>u₃₁</td>
<td>p₃₁</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>u₃₂</td>
<td>p₃₂</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>u₃₃</td>
<td>p₃₃</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3. Examples of consumers’ online shopping behaviors within a short period

Based on these various online shopping behaviors, the DACCN model identifies consumer \( u_i \)'s demand for product \( p_j \) and calculates the intensity as follows:

\[
W(u_i, p_j) = W_c(u_i, p_j) + W_f(u_i, p_j) + W_a(u_i, p_j) + W_b(u_i, p_j)
\]  

(1)

where \( W(u_i, p_j) \) is the intensity of consumer \( u_i \)'s demand for product \( p_j \), which indicates how likely the consumer is to purchase this product. Furthermore, \( W_c(u_i, p_j) \), \( W_f(u_i, p_j) \), \( W_a(u_i, p_j) \), and \( W_b(u_i, p_j) \) are the intensity of consumer \( u_i \)'s demand for product \( p_j \), calculated based on consumer \( u_i \)'s clicking, favoriting, adding, and buying behaviors related to product \( p_j \), respectively. However, different consumers may have different purchase habits, which are reflected by the average numbers of each type of online shopping behavior carried out by consumers before purchasing. Thus, when calculating the intensity of a consumer’s demand for a product, his/her purchase habit reflected by various online shopping behaviors must be considered. Obviously, among all types of online shopping behavior, purchase or buying is the one that indicates the strongest consumer product demand intensity. Hence, DACCN model set the coefficient for the number of times a consumer carried out each type of online shopping behavior by the rate of the consumer’s conversion of the behavior into purchase based on the previous purchase records. In other words, it is the ratio of the number of product purchases to the number of times a consumer carries out a certain type of online shopping behavior. Thus, the intensity of consumer \( u_i \)'s demand for product \( p_j \) based on each type of online shopping behavior can be calculated by multiplying the number of times consumer \( u_i \) carried out a specific type of online shopping behavior for product \( p_j \) with its corresponding coefficient, as follows:

\[
W_c(u_i, p_j) = \frac{\sum_{z=1}^{n} |p_{c_{z}}|}{\sum_{z=1}^{n} |c_{z}|} \times |c_{j}|
\]

(2)
Here, \( n \) is the number of product demands from consumer \( u_i \), whereas \( |c_i| \), \( |f_i| \), \( |a_i| \), and \( |b_i| \) are the numbers of times consumer \( u_i \) carried out the online shopping behaviors of clicking, favoring, adding, and buying for product \( p_j \), respectively. Based on the example given in Table 2, the coefficient for the number of times each type of online shopping behavior was carried out by a consumer for a product is calculated, and the results are shown in Table 4.

### Table 4. Coefficient for the number of times a consumer carries out each type of online shopping behavior

<table>
<thead>
<tr>
<th></th>
<th>Click</th>
<th>Favorite</th>
<th>Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1 )</td>
<td>0.142857143</td>
<td>0.5</td>
<td>0.666666667</td>
</tr>
<tr>
<td>( u_2 )</td>
<td>0.083333333</td>
<td>0.5</td>
<td>0.666666667</td>
</tr>
<tr>
<td>( u_3 )</td>
<td>0.111111111</td>
<td>0.333333333</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Finally, DACCN model sums up the intensity calculated based on each type of shopping behavior as the intensity of consumer \( u_i \)'s demand for product \( p_j \). The value of \( W(u_i, p_j) \) can be more than 1, and the higher is the value of \( W(u_i, p_j) \), the stronger is the intensity of consumer \( u_i \)'s demand for product \( p_j \). The intensity of the consumers’ product demands based on the example in Table 3 was calculated, and the results are shown in Table 5.

### Table 5. Intensity of consumers’ product demands

<table>
<thead>
<tr>
<th></th>
<th>( p_1 )</th>
<th>( p_2 )</th>
<th>( p_3 )</th>
<th>( p_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1 )</td>
<td>3.19047619</td>
<td>2.571428571</td>
<td>-</td>
<td>2.238095238</td>
</tr>
<tr>
<td>( u_2 )</td>
<td>3.5</td>
<td>1.166666667</td>
<td>3.333333333</td>
<td>-</td>
</tr>
<tr>
<td>( u_3 )</td>
<td>3.888888889</td>
<td>-</td>
<td>1.833333333</td>
<td>2.277777778</td>
</tr>
</tbody>
</table>

With the product demands from different consumers, the similarity between two consumers can be calculated. There are many common and effective indices of similarity, such as Euclidean distance, Manhattan distance, Minkowski distance, Hamming distance, and the Jaccard index. However, not all of these indices are applicable for calculating the similarity between two consumers based on the product demands from different consumers because consumers’ product demands are sets. The Jaccard index (Jaccard, 1912), one of the most common similarity measures and a statistic used for comparing the similarity or diversity between two sets, is more applicable for calculation of the similarity between two consumers. This is expressed as follows:
where \( D_i \) and \( D_j \) are the sets of product demands from consumers \( u_i \) and \( u_j \), \( |D_i \cap D_j| \) is the number of products in the intersection of \( D_i \) and \( D_j \), and \( |D_i \cup D_j| \) is the number of products in the union of \( D_i \) and \( D_j \). A similarity value is a real value between 0 and 1. The higher the value, the more similar the two consumers are.

3.2 Construction of a co-demanded product network with adjustment of the edges' weights

Unlike the construction of the co-purchased product network described in Table 1, the DACCN model represents the products demanded by the consumers as nodes and builds edges among the products co-demanded by the same consumers within a short period to construct a co-demanded product network. We further consider various online shopping behaviors of consumers to build the edges among products, and the weights of edges are set according to the intensity of consumer demands and adjusted based on influential factors in the construction of the co-demanded product network. The difference between the construction processes of these two product networks based on the example in Table 3 is shown in Figure 4.

Some noisy edges are created because of the products coincidentally co-demanded by several consumers in the product networks. To obtain reliable associations among products, edges must be filtered using a minimum threshold; that is, an edge exists between two products only if they have been co-demanded by at least \( \alpha \) consumers. In the existing network analysis methods for MBA, Raeder and Chawla (2009) filtered edges with weights lower than 10, Kim et al. (2012) removed edges with weights lower than the average. While, Videla-Cavieiras and Ríos (2014) generated a
filtering threshold by first averaging the weights of the edges with the top three weights and then filtering the edges by different percentages of the threshold. Therefore, the DACCN model uses an edge-filtering method with the following design considerations: First, because the constructed co-demanded product network has nodes and edges with different characteristics compared with the product network constructed by Raeder and Chawla (2009), it is inappropriate to preset a particular threshold for such a network. Second, it is difficult to determine which threshold value is appropriate for the proposed model based on Videla-Caviers and Ríos’s (2014) method. The DACCN model adopts an effective filtering method like that proposed by Kim et al. (2012), in which the edges are filtered by the average number of consumers co-demanding a pair of connected products. This is expressed as follows:

\[
\alpha = \frac{\sum_{(p_i, p_j) \in E_p} |F_i \cap F_j|}{|E_p|} \tag{7}
\]

where \( E_p \) is the set of edges among all the connected products; \( p_i \) and \( p_j \) are the pairs of connected nodes of the products with an edge; \( F_i \) and \( F_j \) are the sets of consumers demanding products \( p_i \) and \( p_j \), respectively; \( |F_i \cap F_j| \) is the number of consumers in the intersection of \( F_i \) and \( F_j \), indicating the number of consumers demanding both products \( p_i \) and \( p_j \); and the numerator is the sum of the number of consumers co-demanding a pair of connected products.

Unlike in the existing network analysis methods for MBA, in the proposed method, the weight of an edge between two co-demanded products is determined based not only on the number of consumers co-demanding these two products but also on the intensity of the same consumers’ demands for such products. The weight of the edge between product \( p_i \) and product \( p_j \), \( W(p_i, p_j) \), is calculated by aggregating the multiplication of the intensity of the demands from the same consumers for these two products, as expressed in the following:

\[
W(p_i, p_j) = \sum_{u_k \in F_i \cap F_j} \left[ W(u_k, p_i) \times W(u_k, p_j) \right] \tag{8}
\]

Thus, the co-demanded product network is constructed as \( G_p = \{P, E_p, W(E_p)\} \).

There are some important factors that may influence the effectiveness of discovering the associations among products (Smith & Linden, 2017; Yan et al., 2017; Zhang et al., 2016). However, these factors are neglected in the existing network analysis methods for MBA based on co-purchased product networks. To discover more reliable and practical associations among products, these factors were further considered in this study to adjust the weights of the co-demanded product network’s edges. Moreover, a threshold was set to filter the edges with low weights. Below, we explain the computation of the adjustment of the edges’ weights based on the following factors: (1) the time interval between two product demands from the same consumers, (2) the popularity of each product, and (3) the number of each consumer’s product demands.

For the adjustment of the edges’ weights based on the time interval between two product demands from the same consumers, the relativity of products refers to the degree to which consumers who buy one product are unusually likely to buy the other products. Computing this relativity and how related a purchase is to another depends heavily on the two purchases’
proximity in time (Smith & Linden, 2017). Another study indicated that the time interval between consumers’ behaviors related to certain products must play an important role in determining the similarities among such products (Jia & Liu, 2015), which may extend to the reliability of the associations among them. Therefore, it is necessary to adjust the edges’ weights based on the time interval between two product demands from the same consumers. It is reasonable to expect that the reliability of the association between two products co-demanded by the same consumers will decrease quickly with increasing time intervals between two product demands. For example, two products co-demanded by the same consumers within 1 day are highly likely to be associated consumer demands, but if they are co-demanded by the same consumer over more than 1 week, even only reaching 8 days, they are probably independent of each other. Thus, the adjustment factor of the time interval $\beta_i$ is calculated by exponential function as

$$\beta_i = e^{\gamma \frac{s_{u_i} - s_{u_j}}{\lambda}}, (0 < \gamma < 1)$$

(9)

where $\gamma$ is the coefficient set based on expert knowledge and enterprises’ preferences in different scenarios. The larger the $\gamma$ is, the more products co-demanded by the same consumers within a longer period are considered to be related. The enterprise can select a smaller $\gamma$ if it prefers to detect stronger associations among products, whereas it can select a larger $\gamma$ if it prefers to detect more varied associations among products even if they are weaker. Furthermore, $|t_{w_i} - t_{w_j}|$ is the time interval between consumer $u_i$’s demands for products $p_i$ and $p_j$. As consumers usually carry out more than one shopping behavior related to a product, the time of consumer $u_i$’s demand for product $p_i$ is the average of the initial and final times that the consumer carries out online shopping behaviors for product $p_i$.

Regarding the adjustment of the edges’ weights based on the popularity of each product, the popularity of a product can be reflected by the sum of the intensity of consumers’ interest in purchasing it. Popular products tend to attract more consumers, and there are added consumer behaviors carried out in relation to them. Thus, such products are demanded by more consumers and then earn a higher number of common consumers with connected product demands compared with less popular products. Moreover, popular products will have larger edge weights in the network, and the larger weighted edges associated with a few popular products can then dominate the analysis in the network (Zhang et al., 2016). Therefore, it is essential to adjust the edges’ weights based on the popularity of each product. It is reasonable to think that, when a product is popular, the associations between it and other products are less reliable if the distraction of the product popularity is neglected. To avoid this distraction, when calculating the weight of an edge between two co-demanded products, the popularity of these two products should be divided in the denominator. A product’s popularity is determined by the number of consumers interested in purchasing it. As such, to measure this popularity, the sum of the intensity of the consumers’ demands for the product is obtained. Thus, the adjustment factor of the popularity of products $\beta_2$ to the edge between products $p_i$ and $p_j$ is calculated as
\[ \beta_i = \frac{1}{\sum_{u \in F_i} W(u_i, p_i) \times \sum_{u \in F_j} W(u_i, p_j)} \]

where \( \sum_{u \in F_i} W(u_i, p_i) \) is the popularity of product \( p_i \), calculated by summing up the intensity of the consumers’ demands for product \( p_i \), and \( \sum_{u \in F_j} W(u_i, p_j) \) is the popularity of product \( p_j \).

Regarding the adjustment of the edges’ weights based on the number of each consumer’s product demands, because there are some consumers who demand many products, some products may not truly be associated with each other despite being co-demanded by the same consumers within a short period. It can be said that the discovery of associations among the products demanded by a consumer is less reliable if the consumer has many product demands. For example, a consumer who is interested in reading, jogging, and cooking may carry out online shopping behaviors on novels, pens, running shoes, sports watches, pans, and spices in a short period. It is obvious that not all of these products are associated with each other. However, if a consumer carries out online shopping behaviors on only a few product categories within a short period, it is more likely that these products are associated with each other. Therefore, it would be better to reduce the interferences of the number of product demands from each consumer in the discovery of associations among products. The higher the number of product demands from a consumer is, the less the consumer’s contribution to the weights of the edges among the co-demanded products is. Like the calculation of item similarity based on users’ ratings in the work of Yan et al. (2017), adjustment factor \( \beta_i \) of the number of product demands from consumer \( u_i \) is calculated as follows:

\[ \beta_i = \exp(-\lg |R(u_i)|) \]

where \( |R(u_i)| \) is the number of product demands from consumer \( u_i \). The larger \( |R(u_i)| \) is, the smaller \( \beta_i \) becomes.

After calculating the adjustment factors of the time interval between two product demands from the same consumers, the popularity of each product, and the number of product demands from each consumer, the weights of the co-demanded product network’s edges are adjusted as follows:

\[
W(p_i, p_j)^* = \frac{\sum_{u \in F_i} \left[ W(u_i, p_i) \times W(u_i, p_j) \times \beta_i \times \beta_j \times \beta_2 \right] \times \beta_2}{\sum_{u \in F_i} \sum_{u \in F_j} W(u_i, p_i) \times \sum_{u \in F_j} W(u_j, p_j)}
\]

Because only the edges between two products co-demanded by less than \( \alpha \) consumers have been removed, to further filter reliable and practical associations among the co-demanded products, the average adjusted weight of the edges is set as threshold \( \delta \) following Kim et al. (2012). This is expressed as:

\[
\delta = \frac{\sum_{(p_i, p_j) \in E_p} W(p_i, p_j)^*}{|E_p|}
\]
two steps in DACCN model. This is because the edges’ weights are based not only on the number of consumers co-demanding the connected products but also on the intensity of the consumers’ demands for such products and the mentioned adjustment factors in the co-demanded product network. Filtering edges first by the number of consumers co-demanding the connected products can help save time when calculating the adjusted edges’ weights. Moreover, the filtering of edges based on their adjusted weights further guarantees the reliability of the remaining relationships among the co-demanded products. Hence, a co-demanded product network with adjusted edge weights is constructed.

3.3 Detection of overlapping communities within the co-demanded product network

After the construction of the co-demanded product network, the aim is to isolate associations among products and alleviate redundancy by discovering larger, more expressive relationships among product groups. Hence, tightly connected co-demanded products are detected as product communities (“communities of associated consumer demands”) to clearly explain the associations among the co-demanded products. As consumers may demand the same product in various scenarios, a product may belong to more than one community of associated consumer demands. For example, milk and bread may be co-demanded by consumers for breakfast, while milk and flour may be co-demanded by consumers for making a cake. Milk should be allowed to belong to the community of consumer demands associated with bread and the community of consumer demands associated with flour. Therefore, an algorithm for overlapping community detection is more suitable in the model because it allows a node to belong to more than one community. As mentioned in section 2.3, among 14 common algorithms for the detection of overlapping communities, the SLPA—an extension of the Label Propagation algorithm (LPA; Raghavan et al., 2007)—performed better and more stably for both low- and high-overlapping-density networks (Xie et al., 2013). Moreover, the computational complexity of the label propagation approach is relatively low compared with that of all other algorithms for the detection of overlapping communities (Amelio & Pizzuti, 2014). The SLPA has also been effectively applied in another study with a similar aim to ours (Videla-Cavires & Ríos, 2014). Thus, the SLPA is considered to be used in the DACCN model to detect communities of associated consumer demands based on the work of Xie et al. (2011), in the steps described below.

First, the memory of each node of demanded products is initialized with each product’s ID as a label. Then, the following steps are repeated for T times, and the loop has three steps: (1) one node of demanded products is selected as a listener; (2) each neighbor node of the listener is a speaker that sends out a label to the listener that is randomly selected from its memory and with a level of probability proportional to the label’s frequency of occurrence in its memory; and (3) the listener accepts one label from the set of labels received from the speaker nodes. Because the co-demanded product network is a weighted network, the weights of the edges between the listener and the different speakers are distinct. Therefore, for more practical and effective community detection, the listener accepts the label based not only on its popularity but also on the weights of the edges between the listener and speakers. Finally, the labels with a frequency of less than the threshold r in each node’s memory are removed, and the nodes of the demanded products with the same label are classified as belonging to a community of associated consumer demands. Hence, various communities of associated consumer demands are detected.

After the detection of overlapping communities of associated consumer demands, and in order to provide support for practitioners, such as e-commerce platforms or online retailers to
decide about the most reliable communities of associated consumer demands to meet first, the communities need to be ranked. Raeder and Chawla (2009) selected the sum of the confidence of the relationships indicated by the edges within the community as the information measure for each community and the information per node in the community as the information density measure. Like Raeder and Chawla’s (2009) work, the DACCN model defines an information density measure to rank the detected communities. Because the edges in the co-demanded product network are built based on the products co-demanded by the same consumers within a short period, the confidence level of the relationships indicated by the edges cannot be calculated. The weights of the edges reflect the closeness of the nodes of the demanded products in the co-demanded product network to each other; thus, it is a good idea to replace the sum of confidence with the sum of the weights of the edges within the community to represent the information present in the community. The community’s information density is calculated as the information per node following Raeder and Chawla (2009). This is expressed as:

\[
D(C_i) = \frac{\sum_{(p_i, p_j) \in E_i} W(p_i, p_j)}{|P_i|}
\]

(14)

where \(D(C_i)\) is the information density of community \(C_i\), \(|P_i|\) is the number of associated consumer demands in community \(C_i\), and the numerator is the sum of the adjusted weights of the edges in community \(C_i\).

4. Evaluation

After the discussion of the novel model for the discovery of associated consumer demands in the previous section, this section presents how the proposed model was evaluated based on a real dataset from an e-commerce platform. The detailed information on the dataset and the data that were used to evaluate the proposed model are presented, and the evaluation methods for testing the feasibility and validity of the proposed model are described.

4.1 Data

Taobao is one of the most popular e-commerce platforms in China, and a huge number of consumers shop for various product categories using this online platform every day. Thus, a dataset of user behavior data from Taobao is attainable and suitable for testing and validating the proposed model. The dataset is provided by Alimama (https://tianchi.aliyun.com/dataset/dataDetail?dataId=649). The online shopping behaviors from November 25 to December 3, 2017, of about 1 million consumers related to the products in the Taobao e-commerce platform were recorded in the dataset. These behaviors were as follows: viewing the product page (click), favoriting a product (favorite), adding a product to one’s shopping cart (add), and purchasing a product (buy). The timestamp of each behavior was also recorded. However, only the IDs of products and product categories were given in the dataset; their names were not. Detailed information on the dataset is shown in Table 6.
Table 6. Detailed information on the dataset

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumers</td>
<td>987,994</td>
</tr>
<tr>
<td>Products</td>
<td>4,162,024</td>
</tr>
<tr>
<td>Product categories</td>
<td>9,439</td>
</tr>
<tr>
<td>Behaviors</td>
<td>100,150,807</td>
</tr>
</tbody>
</table>

Because it is reasonable to expect that products are associated with each other if a certain number of the same consumers co-demanded them within a short period, we used the data from November 25–27, 2017, to construct the co-demanded product network as an example. The experiments were conducted on an Intel 4 Core i7-8650U 2.11 GHz personal computer with 16 GB of RAM using Python 3.7.6 software. Considering the calculation capability of the personal computer, we calculated data from 10,000 consumers to validate the proposed model. Consumers usually compare products in the same category before they decide to purchase one of them; although consumers carry out some online shopping behaviors related to these products during a short period, they usually demand only one product. Therefore, to discover practical associated consumer demands, product categories were made the analysis objects of the model instead of products.

4.2 Evaluation methods

4.2.1 Feasibility of the proposed model

Based on the dataset, the feasibility of the proposed model was tested following the detailed steps presented in section 3. Based on the work of Xie et al. (2011), the algorithm for the detection of overlapping communities, SLPA, produces relatively stable outputs in general when the number of iterations is greater than 20. Thus, the number of iterations was set to 20 in the evaluation. A higher filtering threshold of node’s labels means that fewer nodes will belong to more than one community; hence, there is no “optimal” choice for the selection of the filtering threshold. The enterprise can select a higher filtering threshold if it prefers to detect stronger associations among products, while it can select a lower filtering threshold if it prefers to detect more varied associations among products, which will then be weaker. Based on the work of Xie et al. (2011), the SLPA converges (i.e., producing a similar ratio of the number of detected communities to true communities) as the filtering threshold of the node’s labels varies from 0.01 to 0.1. The number of detected communities to true communities is close to 1 when the filtering threshold of the node’s labels is set to 0.1. Thus, we set the filtering threshold to 0.1 to make the number of detected communities closer to the true value, and this was selected as an example to evaluate the proposed model.

It was difficult to explain the associations among the co-demanded product categories in the detected communities of consumer demands because the dataset contained only the product category IDs and did not contain the true names of the product categories. Thus, only the numerical characteristics of the detected communities of associated consumer demands were analyzed to show the feasibility of the proposed model—the number and size distribution of the detected communities and the number of consumer demands belonging to more than one community.

We used a powerful Python module "Graph-tool" (https://graph-tool.skewed.de/) to visualize
the co-demanded product network after the detection of overlapping communities. To clearly show the popularity of each product category and the strength of the relationships among the product categories, the size of the nodes was set based on the popularity of the product categories, and the width of the edges was set based on their adjusted weights. Besides, the nodes of product categories belonging to the same communities were drawn with the same color, and the ones belonging to more than one community were drawn with more than one color to clearly show the overlapping communities. To clarify the layout of the network, the ARF spring-block layout algorithm was used for adjustment.

4.2.2 Validity of the proposed model

To validate the proposed model, both the validity of the SLPA, selected by the proposed model for the detection of overlapping communities, and the network analysis method based on the proposed co-demanded product network with adjusted weights were evaluated as described below.

To prove the superiority of SLPA, we compared it with the state-of-the-art algorithms for the detection of overlapping communities in the real dataset used in the paper. Following the work of Vieira et al. (2020), we selected CFinder (Palla et al., 2005), COPRA (Gregory, 2010), and Demon (Coscia et al., 2014). They can be briefly introduced as follows:

(1) CFinder is one of the first methods proposed in the literature for overlapping community detection, and it is considered in almost all works regarding comparing algorithms for overlapping community detection. It implements the Clique Percolation Method (CPM), which assumes that a community consists of overlapping sets of fully connected subgraphs and detects communities by searching for adjacent cliques. It is extended by CPMw, which introduces a subgraph intensity threshold for weighted networks (Farkas et al. 2007). As the constructed co-demanded product network is weighted, we used CPMw algorithms for comparison.

(2) COPRA is a common algorithm for overlapping community detection, and its validity has been proven in several studies. It introduces a belonging coefficient indicating the strength of each node’s membership in the community. Each node updates its belonging coefficients by averaging the coefficients from all its neighbors at each time step synchronously. A parameter \( v \) is also proposed to control the maximum number of communities to which a node can belong.

(3) Demon has been proven to be good at dealing with large-scale networks, which is appropriate in real-world scenarios (Coscia et al., 2014). Similar to SLPA, Demon is also based on the variation of label propagation. However, it considers an ego-network—that is, the network built around an ego node—for each node and evaluates the labels of each group shared by the node.

To evaluate the performance of the algorithms mentioned above for detecting the overlapping communities within the co-demanded product network, we select one widely used measure, modularity (Newman, 2004), as the evaluation criterion. Although there are various effective measures for evaluating community detection, such as modularity (Newman, 2004), normalized mutual information (NMI; Danon et al., 2005), and the adjusted Rand index (ARI; Hubert & Arabie, 1985), most of them need to compare the results of community detection with the real community structure. Therefore, as the real community structure of the co-demanded product network is unknown, most of these measures are not appropriate for evaluating the algorithms’
performance. Modularity is one of the most well-known measures for the assessment of community partition, and the real community structure is not needed for comparison. Therefore, it is an appropriate measure to evaluate the performance of the algorithms to detect overlapping communities within the co-demanded product network. Modularity measures the fraction of the edges in the network that connect vertices of the same type (i.e., within-community edges) minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices (Newman & Girvan, 2004). It is extended to apply to weighted networks as follows (Newman, 2004):

\[
Q = \frac{1}{2m} \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)
\]

(15)

where \( Q \) is the modularity of the weighted network, \( A_{ij} \) represents the weight of the edge between nodes \( i \) and \( j \), and \( m = \frac{1}{2} \sum_{i} A_{ii} \). In addition, \( k_i = \sum_{j} A_{ij} \) indicates the sum of the weights of the edges attached to node \( i \), and the function \( \delta(c_i, c_j) \) is 1 if nodes \( i \) and \( j \) are in the same community and 0 otherwise. The value of the modularity of the weighted network ranges from \(-0.5\) to \(1\), and the higher the value, the better the community partition. We compared the efficiency of SLPA with the algorithms mentioned above based on their time complexity as well.

To evaluate the network analysis method based on the proposed co-demanded product network with adjusted weights (CDNA), we compared it to the network analysis method with the co-purchased product network (CPN) built based on multiple transactions by the same consumers, which was commonly used in previous studies (Kim et al., 2012; Dhar et al., 2014; Faridizadeh et al., 2018; Lismont et al., 2018). The construction of this product network is shown in Table 1, and the edges within the product network are filtered by the average number of consumers co-purchasing a pair of connected products within a short period. Like the DACCN model, the SLPA is used to detect overlapping communities of associated products within the product network after network construction in this method. As mentioned above, because the number of products purchased in a single transaction has decreased due to the emergence of online shopping, the co-purchased product network built based on the same transactions may not work effectively for discovering associations among products in the context of e-commerce. Thus, it was not used for comparison in this study. To evaluate the weight adjustment of the network’s edges, the effectiveness of the network analysis method based on CDNA and that based on the co-demanded product network with unadjusted weights (CDNU) were compared. Except for the adjustment of edges’ weights, the steps of these two network analysis methods are the same. Hence, we compared the effectiveness of the network analysis methods based on these three networks from the following four perspectives: (1) the characteristics of the constructed product networks, (2) the numerical characteristics of the detected communities, (3) the stability of the detected communities, and (4) the complexity of the proposed method.

First, we compared the characteristics of the three networks (i.e., number of nodes, number of edges, degree distribution, average clustering coefficient, and network density) following Qi et al. (2013) to show their differences. We explored the probable reasons for these differences to show
the effectiveness of the proposed model. Second, the numerical characteristics of the detected communities (i.e., the modularity, number, overlapping number, and size distribution of the product communities detected based on the three networks) were compared to evaluate the performance of overlapping community partition with the network analysis methods based on the three networks. Third, to provide reliable decision support for practitioners, we calculated and compared the stability of the product communities detected based on the three networks to evaluate the performance of overlapping community partition with the network analysis methods based on the three networks.

Third, to provide reliable decision support for practitioners, we calculated and compared the stability of the product communities detected based on the three networks within a short period. It is reasonable to expect that the results of product communities detected by the same network analysis–based method within a short period will be stable if the detected product communities are reliable. Finally, we compared the efficiency of the proposed network analysis method based on CDNA, with the other two network analysis methods mentioned above based on their time complexity, space complexity, and runtime.

For the comparison of the stability of the product communities detected based on the three networks, the time window in the evaluation was set to 3 days. This is because the three network analysis methods for comparison are all based the product networks established based on the products purchased or demanded together by the same consumers within a short period. If holidays are considered in the evaluation, the time window should be shortened appropriately because more products will be co-demanded by the same consumers within the 3-day time window stemming from the discount during holidays. However, products co-demanded by the same consumers within a relatively long period may not be associated with each other. Besides, there may be some special associations among products during holidays. If part of the period selected for evaluation contains holidays, it is likely to affect the stability comparison of product communities detected based on the three networks within a short period. Since the data from November 25 to December 3 on the Chinese e-commerce website selected in this study for evaluation does not include data on holidays, it will not affect the stability comparison. Moreover, TW = \{tw_1, tw_2, ..., tw_9\} because only data for 9 days were accessible in the dataset. Communities of associated consumer demands were detected in each time window; as such, a set of communities of associated consumer demands \( C_i \) was established for each time window \( tw_i \), and each community \( C_i' \) in set \( C_i \) contained a set of co-demanded product categories \( P_i' \).

A simple way of checking the stability of a detected community over time is to find its most similar communities in each of the considered time windows; such similarities can show how the community has changed, which is useful for measuring community stability. Meanwhile, to determine the similarity between two consumers, we used the Jaccard index to calculate the similarity between communities of products, as these communities are also sets. Following Ríos and Videla-Cavieres (2014), we used the Jaccard index to calculate the similarity between each community in the first time window and each community in the other time windows. In the \( i^{th} \) time window, the stability of the \( k^{th} \) community in the first time window can be defined as the maximum similarity between \( C_i^k \) and \( C_i' \), which is calculated as follows:

\[
\text{stability}(C_i^k) = \max_{C_i' \in C_i} \text{similarity}(C_i^k, C_i')
\]  

\[
\text{similarity}(C_i^k, C_i') = J(C_i^k, C_i') = \frac{|P_i^k \cap P_i'|}{|P_i^k \cup P_i'|}
\]  

where \( P_i^k \) and \( P_i' \) are the sets of co-demanded product categories for the \( k^{th} \) community in the \( i^{th} \) time window and the \( i^{th} \) community in the \( i^{th} \) time window, respectively.
where $|P_k^1 \cap P_j^1|$ is the intersection of the set of product categories contained in the $k^{th}$ community in the first time window and the set of product categories contained in the $j^{th}$ community in the $i^{th}$ time window, whereas $|P_k^1 \cup P_j^1|$ is their union. To evaluate the overall stability of all the detected communities in the $i^{th}$ time window, the overall stability of the set of communities in the first time window is calculated as the average stability of the communities within the network as:

$$\text{stability}(C_i) = \frac{\sum_{C \in C_i} \text{stability}(C_i)}{|C_i|} \quad (18)$$

where $|C_i|$ is the number of communities within the network in the first time window. For clear comparison, the overall stability of $C_i$ detected in the three networks in the following time windows is presented through a line chart.

5. Results

5.1 Discovery of associated consumer demands based on a co-demanded product network

We identified the consumers’ demands for certain product categories based on consumers’ various online shopping behaviors, and we calculated their intensity levels using equations (1) to (5), along with the corresponding average timestamps. Table 7 shows 10 examples based on the results.

<table>
<thead>
<tr>
<th>Consumer ID</th>
<th>Category ID</th>
<th>Intensity of consumer demands</th>
<th>Average timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>149192</td>
<td>0.013157895</td>
<td>1511761623</td>
</tr>
<tr>
<td>11</td>
<td>154040</td>
<td>0.039473684</td>
<td>1511658734</td>
</tr>
<tr>
<td>11</td>
<td>181348</td>
<td>0.013157895</td>
<td>1511619684</td>
</tr>
<tr>
<td>11</td>
<td>815311</td>
<td>0.013157895</td>
<td>1511686919</td>
</tr>
<tr>
<td>11</td>
<td>903809</td>
<td>0.013157895</td>
<td>1511783214</td>
</tr>
<tr>
<td>11</td>
<td>982926</td>
<td>0.118421053</td>
<td>1511734815</td>
</tr>
<tr>
<td>11</td>
<td>1080785</td>
<td>0.065789474</td>
<td>1511680307</td>
</tr>
<tr>
<td>11</td>
<td>1127566</td>
<td>0.013157895</td>
<td>1511676822</td>
</tr>
<tr>
<td>11</td>
<td>1320293</td>
<td>0.013157895</td>
<td>1511704576</td>
</tr>
<tr>
<td>11</td>
<td>1813610</td>
<td>0.013157895</td>
<td>1511686748</td>
</tr>
</tbody>
</table>

Thereafter, we represented the product categories ($CID$) demanded by the consumers as nodes and built edges among such nodes if they were co-demanded by the same consumers within 3 days. The edges were filtered by the average number of consumers co-demanding a pair of connected product categories, using equation (7). We also calculated each remaining edge’s weight $W_{cd}$, time interval $t$ (unit: day) among the same consumers’ demands for product categories, multiplication of the popularity of each pair of co-demanded product categories ($W_{cd}$), and the number of consumer demands ($W_{cd}$) for further analysis. Table 8 shows 10 examples based on the results.
Table 8. Examples of the edges before weight accumulation and adjustment

<table>
<thead>
<tr>
<th>CID1</th>
<th>CID2</th>
<th>Wcc</th>
<th>t</th>
<th>Wtd</th>
<th>Wsd</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>149192</td>
<td>154040</td>
<td>0.000519391</td>
<td>1.190844907</td>
<td>12138.68828</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>149192</td>
<td>181348</td>
<td>0.00017313</td>
<td>1.6428125</td>
<td>335.6240246</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>149192</td>
<td>815311</td>
<td>0.00017313</td>
<td>0.86462963</td>
<td>1496.258343</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>149192</td>
<td>903809</td>
<td>0.00017313</td>
<td>0.24989533</td>
<td>21140.51878</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>149192</td>
<td>982926</td>
<td>0.001558172</td>
<td>0.310277778</td>
<td>65060.23193</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>149192</td>
<td>1080785</td>
<td>0.000865651</td>
<td>0.941157407</td>
<td>20283.48657</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>149192</td>
<td>1127566</td>
<td>0.00017313</td>
<td>0.981493056</td>
<td>1199.161145</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>149192</td>
<td>1320293</td>
<td>0.00017313</td>
<td>0.660266204</td>
<td>40220.90091</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>149192</td>
<td>1813610</td>
<td>0.00017313</td>
<td>0.866608796</td>
<td>1539.97972</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>149192</td>
<td>2355072</td>
<td>0.00034626</td>
<td>0.41744213</td>
<td>56182.18226</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

Using equation (12), the adjusted weights of the edges were calculated. Via equation (13), we used the average adjusted weight of the edges to filter the less reliable edges among the product categories. Table 9 shows 10 examples based on the results.

Table 9. Examples of edges with adjusted weights

<table>
<thead>
<tr>
<th>CID1</th>
<th>CID2</th>
<th>Adjusted Wcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>5064</td>
<td>2181935</td>
<td>0.001541298</td>
</tr>
<tr>
<td>5064</td>
<td>3323023</td>
<td>0.001930783</td>
</tr>
<tr>
<td>5064</td>
<td>3855599</td>
<td>0.00262126</td>
</tr>
<tr>
<td>5064</td>
<td>5002819</td>
<td>0.008503811</td>
</tr>
<tr>
<td>16219</td>
<td>5071267</td>
<td>0.002837884</td>
</tr>
<tr>
<td>21059</td>
<td>983613</td>
<td>0.037467861</td>
</tr>
<tr>
<td>21059</td>
<td>2693696</td>
<td>0.000915605</td>
</tr>
<tr>
<td>22129</td>
<td>2892802</td>
<td>0.016528788</td>
</tr>
<tr>
<td>22129</td>
<td>3123784</td>
<td>0.002690958</td>
</tr>
<tr>
<td>22129</td>
<td>4731191</td>
<td>0.002128423</td>
</tr>
</tbody>
</table>

We constructed a co-demanded product network with adjusted edge weights and used SLPA to detect overlapping communities of associated consumer demands. A total of 313 communities of associated consumer demands were detected from the co-demanded product network with adjusted edge weights, and the number of overlapping among detected communities was 154. The results show that many product categories belong to more than one community of associated consumer demands, which proves the necessity of using an algorithm for the detection of overlapping communities. The distribution of the product categories in each community of associated consumer demands is shown in Figure 5.
The communities of associated consumer demands with the top 10 numbers of product categories contained 62, 25, 19, 18, 16, 15, 14, 14, and 12 product categories. Most communities contained fewer than 10 product categories. Therefore, there was no community detected by the proposed model that contained most of the product categories, which is more realistic compared with the work of Vdela-Cavieres and Ríos (2014). Thus, the communities of associated consumer demands detected by the proposed model are more useful and practical for giving product recommendations, up-selling, cross-selling, and making further managerial decisions.

Finally, we used a powerful Python module "Graph-tool" to visualize the co-demanded product network after the detection of overlapping communities. Figure 6 visualizes the results of the detection of associated consumer demands through the detection of overlapping communities.

As can be seen from the visualized network, some product categories are particularly demanded by many consumers, and some relationships among product categories are strong. Besides benefiting from the adjusted weights of the edges among co-demanded product categories, product categories are contained in varied communities of associated consumer demands, which is closer to reality. The situation where a detected community contains most of the product categories does not really happen.
Figure 6. Visualized co-demanded product network after the detection of overlapping communities.

To provide decision support for determining the most reliable communities of associated consumer demands, we ranked the communities based on the information density measure using equation (14). The communities of associated consumer demands with the top 10 information density values are shown in Table 10.

Table 10. Communities with the top 10 information density values

<table>
<thead>
<tr>
<th>Community ID</th>
<th>Information density</th>
</tr>
</thead>
<tbody>
<tr>
<td>3576283</td>
<td>0.13650232568832288</td>
</tr>
<tr>
<td>2177033</td>
<td>0.05944024198637605</td>
</tr>
<tr>
<td>5045733</td>
<td>0.05814853281558496</td>
</tr>
<tr>
<td>739426</td>
<td>0.05144617890620543</td>
</tr>
<tr>
<td>882095</td>
<td>0.04654958431853803</td>
</tr>
<tr>
<td>2070852</td>
<td>0.04155397880120315</td>
</tr>
<tr>
<td>5120683</td>
<td>0.04074978580900884</td>
</tr>
<tr>
<td>1992078</td>
<td>0.0366733370953631</td>
</tr>
<tr>
<td>1090997</td>
<td>0.036263501402231654</td>
</tr>
<tr>
<td>2072473</td>
<td>0.03612535974598079</td>
</tr>
</tbody>
</table>
The information density value of the top community is much larger than those of the following communities, and the information density values of the latter are close to each other.

5.2 Evaluation of the algorithm for the detection of overlapping communities within the co-demanded product network

In this section, we empirically compare the performance of SLPA (the filtering threshold of the node’s labels $r$ is set to 0.1 in the proposed model) with CFinder, COPRA, and Demon on the constructed CPNA based on the selected real dataset. For algorithms with tunable parameters, the results with various settings are reported. For CFinder, the minimum clique size $k$ ranges from 3 to 8. For COPRA, the number of maximum memberships of nodes $v$ ranges from 2 to 10. For Demon, the merging threshold $\epsilon$ ranges from 0.05 to 0.3, with an interval of 0.05. The modularity of the communities detected by these algorithms within the constructed co-demanded product network is shown in Table 11. The results of the number of detected communities and the number of overlaps among detected communities are also provided in the table.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Modularity</th>
<th>Number of communities</th>
<th>Number of overlaps</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLPA ($r=0.1$)</td>
<td>0.4820</td>
<td>313</td>
<td>154</td>
</tr>
<tr>
<td>CFinder ($k=3$)</td>
<td>0.1793</td>
<td>48</td>
<td>83</td>
</tr>
<tr>
<td>CFinder ($k=4$)</td>
<td>0.1197</td>
<td>28</td>
<td>20</td>
</tr>
<tr>
<td>CFinder ($k=5$)</td>
<td>0.0464</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>CFinder ($k=6$)</td>
<td>0.0386</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CFinder ($k=7$)</td>
<td>0.0242</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>CFinder ($k=8$)</td>
<td>0.0134</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>COPRA ($v=2$)</td>
<td>0.0476</td>
<td>46</td>
<td>0</td>
</tr>
<tr>
<td>COPRA ($v=3$)</td>
<td>0.0587</td>
<td>52</td>
<td>7</td>
</tr>
<tr>
<td>COPRA ($v=4$)</td>
<td>0.0851</td>
<td>57</td>
<td>18</td>
</tr>
<tr>
<td>COPRA ($v=5$)</td>
<td>0.1294</td>
<td>59</td>
<td>26</td>
</tr>
<tr>
<td>COPRA ($v=6$)</td>
<td>0.1808</td>
<td>58</td>
<td>57</td>
</tr>
<tr>
<td>COPRA ($v=7$)</td>
<td>0.2150</td>
<td>39</td>
<td>46</td>
</tr>
<tr>
<td>COPRA ($v=8$)</td>
<td>0.2676</td>
<td>32</td>
<td>49</td>
</tr>
<tr>
<td>COPRA ($v=9$)</td>
<td>0.2867</td>
<td>32</td>
<td>54</td>
</tr>
<tr>
<td>COPRA ($v=10$)</td>
<td>0.2894</td>
<td>32</td>
<td>56</td>
</tr>
<tr>
<td>Demon ($\epsilon=0.05$)</td>
<td>0.1540</td>
<td>21</td>
<td>60</td>
</tr>
<tr>
<td>Demon ($\epsilon=0.1$)</td>
<td>0.2341</td>
<td>22</td>
<td>66</td>
</tr>
<tr>
<td>Demon ($\epsilon=0.15$)</td>
<td>0.3719</td>
<td>24</td>
<td>86</td>
</tr>
<tr>
<td>Demon ($\epsilon=0.2$)</td>
<td>0.3754</td>
<td>26</td>
<td>107</td>
</tr>
<tr>
<td>Demon ($\epsilon=0.25$)</td>
<td>0.4250</td>
<td>31</td>
<td>149</td>
</tr>
<tr>
<td>Demon ($\epsilon=0.3$)</td>
<td>0.4308</td>
<td>40</td>
<td>193</td>
</tr>
</tbody>
</table>

Among the algorithms compared above, the modularity and number of communities detected by the SLPA are the largest, and the number of overlaps among the communities detected by the SLPA is only smaller than that detected by Demon ($\epsilon=0.3$). This indicates that the community partition by the SLPA within the constructed CDNA is better than the compared algorithms, and its efficiency in detecting overlapping communities is relatively high. For CFinder, as $k$
increases, the modularity, the number of detected communities, and the number of overlaps among detected communities decreases quickly to small values. The highest modularity attained with CFinder is just 0.1793, which is much smaller than that with the SLPA. For COPRA, as $\nu$ increases, the modularity of detected communities increases to a relatively stable value of around 0.29, and the number of detected communities and the number of overlaps among detected communities become relatively stable. However, these values are also much smaller than those attained with the SLPA. For Demon, as $\varepsilon$ increases, the modularity of detected communities also increases to a relatively large value of 0.4308, whereas the number of overlaps among detected communities increases to the highest value among the values attained with the compared algorithms. This indicates that the community partition by Demon within the constructed co-demanded product network and its efficiency in detecting overlapping communities are also good. However, the number of communities detected by Demon is much smaller than that attained with the SLPA. By comparing the size distributions of the communities detected by these two algorithms, it can be seen that Demon tends to group more nodes into a community than the SLPA does. Most communities detected by Demon contain many nodes, while most communities detected by the SLPA contain a few nodes, making the latter algorithm more practical for product recommendation. Therefore, based on the measures of modularity, the number of detected communities, and the number of overlaps among detected communities, the overall performance of the SLPA is the best among the compared algorithms, and its validity for detecting overlapping communities within the constructed CDNA is illustrated.

The time complexity of the four algorithms can be compared to evaluate their efficiency as follows: (1) The time complexity of SLPA is $O(m)$; it is linear in the number of edges $m$, and $t$ is a predefined maximum number of iterations; (2) the time complexity of CFinder is polynomial in many applications (Palla et al., 2005); (3) the time complexity of COPRA is $O(vm\log(vm/n))$ per iteration, where $\nu$ is the maximum membership of nodes and $n$ is the number of nodes; and (4) the time complexity of Demon is $O(nK^{\alpha/\nu})$, where $K$ is the maximum degree of nodes in the assumed scale free network working with, with a degree distribution of $p_k = k^{-\alpha}$. For SLPA, it produces relatively stable outputs in general when the number of iterations $t$ is greater than 20. For COPRA, the number of maximum memberships of nodes $\nu$ usually ranges from 1 to 10. For Demon, there are few super-hubs that need to be checked the entire network a few times, and the rest of the nodes add nothing to the complexity with $\alpha = 3$; while there are many high-degree nodes ending up with higher complexity, they are still subquadratic in terms of nodes with $\alpha = 2$ and $K \ll n$. Therefore, all the time complexity of SLPA, COPRA, and Demon are relative low. To test the true efficiency of the algorithms in application, we compared the runtime of these algorithms for the detection of overlapping communities within the CDNA based on the selected real dataset. The runtimes of SLPA ($r = 0.1$), CFinder ($k = 3$), COPRA ($\nu = 10$) and Demon ($\varepsilon = 0.3$) were 1.96 s, 0.98 s, 0.50 s and 0.56 s, respectively. Although the runtime of the SLPA ($r = 0.1$) was the largest among these algorithms, it is still efficient to use SLPA for the detection of overlapping communities within the constructed CDNA.

5.3 Comparison with the existing network analysis methods for MBA

5.3.1 Comparison of the characteristics of the constructed product networks

In this section, the results of the characteristics of the constructed CPN, CDNU, and CDNA
based on the selected real dataset were shown and compared. Table 12 shows the characteristics of these three networks.

Table 12. Characteristics of the three product networks in this study

<table>
<thead>
<tr>
<th></th>
<th>CPN</th>
<th>CDNU</th>
<th>CDNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>141</td>
<td>603</td>
<td>946</td>
</tr>
<tr>
<td>Number of edges</td>
<td>314</td>
<td>4132</td>
<td>2305</td>
</tr>
<tr>
<td>Average degree</td>
<td>4.454</td>
<td>13.705</td>
<td>4.873</td>
</tr>
<tr>
<td>Average clustering coefficient</td>
<td>0.440</td>
<td>0.660</td>
<td>0.210</td>
</tr>
<tr>
<td>Network density</td>
<td>0.032</td>
<td>0.023</td>
<td>0.005</td>
</tr>
</tbody>
</table>

As the objective of the network analysis methods with the three product networks is to detect reliable communities of associated products, the nodes without edges are not considered in the network construction. CPN filters edges by the average number of consumers co-purchasing a pair of connected product categories, whereas CDNU and CDNA filter edges by the average number of consumers co-demanding a pair of connected product categories and the average weight of the edges. Furthermore, in CPN, product category nodes are connected only when a certain number of the same consumers co-purchase rather than co-demand such product categories within a short period. Therefore, CPN has far fewer nodes and edges than the other two networks do, and some valuable relationships among the product categories may be neglected in this network. Compared with CDNU, the number of nodes in CDNA is higher, but the number of edges is lower. It can be inferred that the edges of some specific nodes tend to be completely filtered in CDNA, while the edges of various nodes tend to be filtered only partially in CDNU. This may be due to the adjustment of the edges’ weights based on products’ popularity. Hence, the filtering mechanism becomes more effective, and only the unreliable edges among the nodes are filtered instead of all the edges of the nodes of product categories with low popularity.

The average degrees of nodes in CPN and CDNA are similar, but they are much less than that in CDNU. Among the three networks, the degrees of only a few nodes are larger, whereas the degrees of most of the nodes are small. Furthermore, the clustering coefficient of more than one-fifth of the nodes in CDNU is equal to 1, which means that many nodes in CDNU can be easily divided into clusters. However, after the edges’ weight adjustment, the average clustering coefficient of CDNA becomes much less than that of CDNU. This is because of the complete filtration of nodes with less connection to each other when the edges’ weights are not adjusted, as opposed to the edges of various nodes being only partially filtered and the nodes maintained in the network when the edges’ weights are adjusted. Although the average clustering coefficient is higher in CDNU, some valuable relationships among specific product categories may be filtered therein. CPN’s average clustering coefficient is between the coefficients of CDNU and CDNA. Finally, because the number of nodes of CDNA is larger than those of CPN and CDNU while the number of edges is relatively small, CDNA has a much lower network density than the other networks do.

5.3.2 Comparison of the numerical characteristics of the detected communities

Based on equation (15), the modularity of communities detected in CPN, CDNU, and CDNA by the SLPA (r = 0.1) were −0.1555, −0.4733, and 0.4820. Clearly, the performance of community partition within CDNA is good, and it performs much better than those in CPN and CDNU. Both
the modularity of communities detected within CPN and CDNU were very low, even less than 0, indicating that the nodes of products within the communities detected based on these two networks are not densely interconnected and sparsely connected to other parts of the networks. The major reason for the much higher modularity attained by the network analysis method based on CDNA is that the construction of this network considers both various online shopping behaviors of consumers and the important influential factors to differentiate the edges’ weights. Oppositely, the weights of different edges in both CPN and CDNU are similar without this consideration. Hence, to some degree, it is reasonable to believe that the communities of associated products detected by the network analysis method based on CDNA are more accurate and convincing.

The numbers of communities detected for CPN, CDNU, and CDNA were 35, 67, and 313, respectively. The numbers of overlaps of detected communities within the three networks were 25, 73, and 154, respectively. Both the number and overlapping number of communities detected in CDNA were larger than those detected in the two other networks. The size distributions of the communities detected in the three networks are shown in Figures 7(a), 7(b), and 5.

As CDNU and CDNA consider various online shopping behaviors rather than only purchase behavior, the numbers of communities detected by them are much larger than that detected by CPN. However, without edge weight adjustment, most of the nodes of CPN and CDNU are contained in a certain community, leading to fewer detected communities of associated consumer demands in CDNU than in CDNA. It is difficult to explain the relationships among product categories in such a large community of associated product categories, and the reliability of these detected communities is doubtful because they do not seem aligned with reality. Although a few communities of associated consumer demands detected in CDNA had more than 20 product categories, most of the detected communities had fewer than 10 product categories. This and the variety of communities detected in CDNA make it practically useful for practitioners’ detection of product communities, which proves the advantage of the proposed model. The three networks are also visualized in Figures 8(a), 8(b), and 6.
Figure 8. Visualized networks after the community detection of CPN and CDNU
As evident in Figures 6 and 8, CPN has fewer nodes compared with CDNU and CDNA. Most of the nodes in CPN and CDNU are drawn with the same color, which means that there is a detected community containing most of the product categories in these two networks, unlike in CDNA. Especially in CDNU, few nodes of popular product categories dominate the label propagation to other nodes, seriously affecting SLPA’s detection of reliable and diverse communities of associated product categories. Therefore, the proposed model is more useful than the two other networks for detecting more diverse and practical communities of associated product categories.

5.3.3 Comparison of the stability of the detected communities

Figure 9 shows the overall stability of $C_i$ detected in the three networks in the following 3-day time windows as a line chart.

![Figure 9. Comparison of the overall stability of the communities detected in the three networks (time window=3 days)](image)

Although the communities detected in CPN had the highest overall stability at first, it rapidly decreased and fluctuated from 0.15 to 0.2. The overall stability of communities detected in CDNA was slightly lower than that in CPN initially, but it decreased more slowly and became stable at around 0.25. The tendency of the overall stability of communities detected in CDNU was like that in CDNA, but the overall stability of communities detected in CDNU was always the lowest in the observed time windows. Therefore, the communities of associated product categories detected by the proposed model are more reliable than those detected via the two other networks.

To compare the overall stability of product communities detected based on the three networks in a short period more convincingly, we supplemented two experiments by setting the time windows to 2 days and 4 days. The results are shown in Figures 10 and 11.
Figure 10. Comparison of the overall stability of the communities detected in the three networks (time window=2 days)

Figure 11. Comparison of the overall stability of the communities detected in the three networks (time window=4 days)

Similar to the results with the time window=3 days, when the time window=2 or 4 days, the communities detected in CPN still had the highest overall stability at first, but it rapidly decreased. In contrast, the overall stability of communities detected in CDNA was slightly lower than that in CPN initially, but it decreased more slowly and became stable with a value higher than that in CPN. The tendency of the overall stability of communities detected in CDNU was like that in CDNA, but that in CDNU was always the lowest during the observed time windows. These results support the view that the communities of associated product categories detected by the proposed model are more reliable than those detected by the network analysis methods based on the two other networks.

5.3.4 Comparison of the complexity of the network analysis methods

In this section, we compare the efficiency of the proposed network analysis method based on CDNA with the network analysis methods based on CPN and CDNU according to the time complexity, space complexity, and runtime of these methods. These network analysis methods are composed of network construction and the detection of overlapping communities. Because all of these network analysis methods use the same algorithm—the SLPA—for community detection,
and the time complexity of the SLPA was shown in section 5.2, in this section, we compare the time complexity of network construction of these methods. For the construction of CPN, any two nodes of products purchased together by each consumer within a short period need to be matched when constructing CPN. Therefore, if there are $m$ products purchased together by a consumer within a short period, $\frac{m(m-1)}{2}$ matches are required when constructing a CPN based on the consumer; thus, the time complexity of the construction of CPN is $O(m^2)$. For the construction of CDNU and CDNA, any two nodes of products demanded together by each consumer within a short period need to be matched when constructing CDNU and CDNA. Therefore, if there are $n$ products demanded together by a consumer within a short period, $\frac{n(n-1)}{2}$ matches are required when constructing CDNU based on the consumer. However, three important influential factors are considered in the construction of CDNA, and three adjustment factors need to be further calculated. Therefore, $\frac{3n(n-1)}{2}$ times of calculation is required when constructing CDNA based on the consumer. The time complexity of the construction of both CDNU and CDNA is $O(n^2)$.

For the construction of CPN, as mentioned above, if there are $m$ products purchased together by a consumer within a short period, $\frac{m(m-1)}{2}$ matches are required when constructing CPN based on the consumer. In each match, the variable updates once by appending the matched products and the weight of the edge between them. Therefore, the space complexity of the construction of CPN is $O(m^2)$. Similarly, the space complexity is $O(n^2)$ for both CDNU and CDNA.

To empirically compare the efficiency of the proposed network analysis method based on CDNA with the network analysis methods based on CPN and CDNU, the runtime of these methods based on the selected real dataset was compared. The runtimes of the network analysis methods based on CPN, CDNU, and CDNA were 8 s, 1363 s, and 3846 s, respectively. Whether two products are demanded together by a consumer within a short period is determined by whether the consumer has performed at least one online shopping behavior on both these products within a short period, including clicking, favoriting, adding, and buying. Hence, the runtimes of the network analysis methods based on CDNU and CDNA are significantly longer than that of the network analysis methods based on CPN because there were many more products demanded together than purchased together by the same consumers within a short period. Limited by the calculation capability of the personal computer used for the experiment, the runtimes of the network analysis methods based on CDNU and CDNA were relatively long. However, with the rapid development of computers’ calculation capability, it is worth sacrificing some time to detect more modular, diverse, practical, and reliable communities of associated products by the proposed network analysis method based on CDNA.

6. Discussion and conclusions

Nowadays, various consumer demands are mutually associated, and consumers will be more satisfied if these demands can be discovered and products to meet these demands recommended to them during their online shopping experience. Several studies have also shown the importance of discovering associations among products. However, there are limitations in the previous studies,
including redundancy of the extracted association rules, the potential to miss novel and interesting associations among co-demanded products hidden in shopping behaviors, and neglect of several important influential factors. To address these limitations, this research proposed and evaluated a novel model for more diverse, accurate and practical discovery of associated consumer demands based on a co-demanded product network called the DACCN model.

6.1 Major findings

Some major study findings are summarized below. First, in the context of e-commerce, apart from purchase behavior, many other associations among products are hidden in the various online shopping behaviors of consumers. The number of detected communities of associated product categories based on the co-demanded product network is higher than that based on the co-purchased product network. Second, the time interval between two product demands from the same consumers, the popularity of each product, and the number of product demands from each consumer significantly influence the characteristics of the constructed product networks. With the adjustment of these factors to the edges’ weights in the constructed product network, more nodes of product categories but fewer edges among them will remain after filtering; which concludes in that strong associations among more product categories to be discovered. Third, the performance of the detected communities of associated product categories could be improved by the proposed DACCN model in terms of modularity, size distribution, and stability. Compared with other network analysis methods, the modularity of communities detected by the proposed network analysis method based on CDNA is significantly higher than the modularity detected by the network analysis methods based on CPN and CDNU. Hence, this indicates that the performance of community partition within CDNA is much better than those within CPN and CDNU. Considering the size distribution of detected communities, there is no community detected by the proposed model containing most of the product categories, which provides more practical implications for practitioners. Moreover, the communities of associated product categories detected by our proposed model are more reliable than those detected by the network analysis methods based on CPN or CDNU. Finally, both the time complexity and space complexity of the network analysis methods based on CPN, CDNU, and CDNA are $O(m^2)$, $O(n^2)$, and $O(n^2)$ respectively. The runtimes of the network analysis methods based on CDNU and CDNA were relatively long because there were many more products demanded together than purchased together by the same consumers within a short period.

6.2 Contributions to research

This study contributes to research from three different perspectives. First, via network analysis–based MBA methods, this study utilized various online shopping behaviors of consumers to construct a co-demanded product network for the more effective discovery of associations among products. With the rapid development of e-commerce and the popularization of online shopping, the shopping behaviors and habits of consumers have changed, and these various shopping behaviors reflect consumer demands. Considering these behaviors to construct a co-demanded product network will discover more diverse and interesting associations among products, which may result in extended studies on the discovery of associations among co-purchased products. Second, some neglected but important factors that may influence the effectiveness of discovering associations among co-demanded products are considered in the proposed DACCN model including the time interval between two product demands from the same
consumers, the popularity of each product, and the number of product demands from each consumer. Based on some relevant previous studies that have explained the effects of these factors on the discovery of associations among products (Smith & Linden, 2017; Yan et al., 2017; Zhang et al., 2016), these factors were further used to adjust the weights of the relationships among products in the proposed DACCN model for a more accurate and effective discovery of associated consumer demands. Finally, the proposed DACCN model was compared with the network analysis methods based on CPN and CDNU to evaluate its effectiveness based on various criteria. The modularity, numbers, and size distributions of the detected communities based on the three networks were compared, and a novel measure of overall stability of detected communities was proposed to evaluate the DACCN model.

6.3 Contributions to practice

This study makes several practical contributions. We used a real dataset of a well-known e-commerce platform in China to test and validate the proposed model. The obtained results show that the proposed DACCN model is feasible and can detect more modular, diverse, practical, and reliable communities of associated products compared with the existing network analysis methods. Therefore, because the proposed model is based on a product network constructed based on consumers’ product demands, it provides practitioners such as e-commerce platforms and online retailers, with a more useful and practical method of discovering associated consumer demands. Hence, enterprises can use the proposed model to generate better product recommendations for each consumer in e-commerce platforms, based on the ranked associated consumer demands and the calculated similarity among consumers. Some studies have also mentioned the utility of combining associated products and consumers’ historical preferences for giving product recommendations (Faridizadeh et al., 2018; Videla-Cavieres & Rios, 2014), which can stimulate and satisfy consumers’ associated demands in various scenarios through up-selling and cross-selling for higher consumer satisfaction and more profits.

6.4 Limitations and future work

In this study, the proposed DACCN model was shown to be able to detect more modular, diverse, practical, and reliable communities of associated products based on the measures of modularity, number, size distribution, and overall stability of the detected communities. However, the true meaning of each product community detected by the proposed DACCN model has not been analyzed and compared with those detected by other methods because the attainable dataset contains only the IDs of products and product categories. Hence, this research could be made more convincing if the difference between the product communities detected by different network analysis methods were discussed based on their real meaning. Therefore, in future work, the proposed model should be applied by practitioners to further evaluate the meaning and usefulness of the communities of associated consumer demands detected by the model. Besides, the proposed model has provided an improved method of detecting communities of associated consumer demands and also proposed an approach to calculate the similarity among consumers and rank the detected communities based on the information density measure. Considering the popularity of e-commerce and online communities, interactions among consumers are also becoming more frequent, and this may significantly affect consumer preference for products (Li et al., 2019). It will be interesting and valuable to further investigate the interaction among consumers and combine different approaches described in this study to generate more useful product
recommendations for consumers in the future.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant numbers 71772075, 71672074] and the Science and Technology Program of Guangdong Province [grant number 2020A1414010229].

References


community detection methods from the perspective of the structural properties. Applied Network Science, 5(1), 1–42.


• Figures 6 and 8 have been replotted by “Graph-tool” with a better presentation
• Gaps and contributions of the paper have been clarified and highlighted
• The English language presentation has been refined
Jiacong Wu: Conceptualization; Data curation; Formal analysis; Methodology; Writing - original draft; Writing - review & editing.

Yu Wang: Project administration; Supervision; Resources; Funding acquisition.

Sara Shafiee: Writing - review & editing.

Dongsong Zhang: Writing - review & editing.