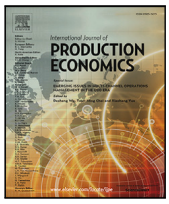
Int. J. Production Economics 280 (2025) 109487



Contents lists available at ScienceDirect

International Journal of Production Economics

journal homepage: www.elsevier.com/locate/ijpe





A personalized content-based method to predict customers' preferences in an online apparel retailer

Alireza KabirMamdouh [a,*], A. Gürhan Kök [b]

[a] Stockholm Business School, Stockholm University, Stockholm, 11419, Sweden
[b] College of Administrative Sciences and Economics, Koç University, Istanbul, 34450, Turkiye

ARTICLE INFO

Keywords:
Recommendation system
E-commerce
Apparel retailer
Item-based cold start

ABSTRACT

A critical decision for an online retailer is to select a set of products out of thousands of possible choices to present to the customers on a web page. The retailer may prefer to offer a different set for each customer because customers have heterogeneous preferences. Thus, to offer the optimal set, the retailer needs to know the customer's preferences. We propose a new personalized content-based method to comprehend customers' preferences in an online retailer based on customers' previous clicks and purchases and attributes of the products. We represent each product with an attribute vector that consists of all attributes of a product, e.g. color and brand. Then, for each customer, a score is assigned to each attribute vector based on the customer's previous preferences, representing his/her interest in that combination of attributes. We test the method using data provided by an apparel retailer. Our method outperforms benchmark methods (including Collaborative Filtering) in predicting customers' preferences (i.e. clicks and purchases) in general, and it has a strictly better performance in predicting customers' preferences over new products. Also, our method outperforms benchmark methods with a high margin in predicting the preferences of customers who are not generally interested in popular products. Finally, we implement a hybrid method consisting of all implemented methods named the Smart Selection. This method outperforms all methods in predicting clicks and purchases with a high margin. This shows that our method provides a complementary approach for Collaborative Filtering by successfully addressing the limitations of commonly used methods.

1. Introduction

Product positioning and recommendation at an online store play a crucial role in customer engagement and subsequently in sales and total revenue. An online retailer may have many available products, however, the display capacity of an e-store is limited. In that sense, an online retailer must decide on the set of products and their positions to be displayed on various points, such as the first page, landing pages, recommendation sidebars, or search query lists. Even if a retailer presents many products in a long list to increase the display capacity, several studies show that a product with a more prominent position on the page has a higher chance of being visited by a customer (Ursu, 2018; Animesh et al., 2011). This emphasizes the importance of the assortment of products selected to be presented in the first few positions. Also, an online retailer may prefer to offer a different set for each customer because customers have heterogeneous preferences, and online platforms are not restricted by a constant set of products to be presented. Thus, to offer a personalized set of products, the retailer needs to predict each customer's preferences.

Recommendation systems (RS) are one of the practical solutions for e-commerce businesses to manage the assortment of products and the order in which they are presented on the web pages. We can define RS as software tools and techniques that suggest items such as products, music, and videos to users. These methods are commonly used where there are several items to present, but there is a limited displaying capacity on each page. RS helps both the user and the retailer by recommending related items that are estimated to be interesting for the user or beneficial for the retailer and filtering unrelated ones. Two well-known types of recommendation systems are Content-based (CB) and Collaborative Filtering (CF). Content-based recommendation systems generally use pre-defined attributes of items to find the similarities among items. For example, books from the same genre or the same author can be assumed to be similar to each other. After investigating the similarity between products, the most similar items to the items preferred by the user would be recommended by CB methods (Ricci et al., 2015; Balabanović and Shoham, 1997). As a result, recommendations provided by basic CB methods are not personalized (customer-specific).

* Corresponding author.
E-mail addresses: amamdouh17@ku.edu.tr (A. KabirMamdouh), gkok@ku.edu.tr (A.G. Kök).

https://doi.org/10.1016/j.ijpe.2024.109487
Received 31 March 2024; Received in revised form 25 November 2024; Accepted 1 December 2024
Available online 6 December 2024
0925-5273/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

In contrast with CB, CF methods can provide personalized content. For example, user-based CF methods use the past interaction of items and users to detect the users that have similar preferences who are called neighbors. After determining the neighborhood, a user-based CF method recommends products preferred by the neighbors of a user to the focal user. Even though collaborative filtering methods have a positive impact on website traffic and sales, they have some limitations. Providing recommendations for new users and new items is challenging for CF methods. This is because CF methods provide recommendations based on the history of preferences, however, data is limited for new items and users. This problem is called the "cold start" problem in the literature (Masthoff, 2015). Note that such a problem is more critical for the recommendation system in online retailers with frequent flash sales. A flash sale retailer may introduce many new products every day or every week that are available for a limited time and in limited quantities. Hence, they face cold start problems more often. Thus, we are interested in a recommendation system that is able to learn customers' preferences in an e-store (specifically with a flash sales business model) and recommend new products.

In this paper, we introduce a personalized content-based method denoted as PCB in the rest of the paper. In this method, we represent each product with an attribute vector. For each product, the corresponding attribute vector consists of all explicitly known attributes of a product, e.g. color, price level, and brand. Then, we use the customers' historical interactions (clicks and purchases) with each specific product attribute to recognize the customers' preferences. In other words, we keep track of the number of times a customer has clicked or purchased a product with a specific attribute. We refer to these numbers as "subscores". We assume that the number of times that a product with a specific attribute (e.g. red products) is clicked or purchased is a positive sign of the customer's preferences for that attribute. Using these values, we provide a preference profile for each customer. Subsequently, for each customer and each attribute vector, we can assign a score of interest representing the customer's interest in that combination of attributes where the score is a function of the subscores.

Note that our method is different from classic CB methods from various perspectives as follows: First, instead of using attributes (content) to find one-to-one similarities between products, we provide a preferences profile for each customer based on the attributes. Then, instead of finding the product that is most similar to the previous products selected by the customer, we evaluate each product by comparing its attributes to the customer's profile/taste. Hence, in contrast with the classic CB method, our method can easily predict a customer's interest in a product that neither itself nor any similar products have not been presented in the store previously. Also, the CB method is known to be prone to the overspecialization problem which means that it offers very similar products without recommending any surprisingly interesting products. However, using our method, a store can recommend a higher variety of products which can solve the overspecialization problem in the practice.

We test the method by a data set provided by an apparel retailer and compare it with benchmarks such as a user-based collaborative filtering method (CF), two heuristic methods recommending popular products (PPC and PPP), and another heuristic method based on the Multinomial Logit choice model (MNL). We measure the performance of each method by its accuracy in predicting the customer's preference after a specified time. In other words, given the set of available products and the customer's history up to a specified time, the method predicts which products would be clicked or purchased by the customer after that time. Considering the average performance of each method over all customers, our method outperforms other methods in predicting clicks and purchases.

Note that our data is provided by a retailer that offers products for a very limited time (on average 9 days). Thus, we investigate the performance of each method in recommending new products as one of the main aims of the PCB method is to address the cold start problem. We define products introduced in the last 24 h as "new products" and measure the performance of the methods in recommending them. PCB has a strictly better performance in recommending new products than other methods.

We also take a deeper look at the performance of the methods for different customer segments as we observe that the performance of the benchmark methods (especially CF) has a very high variance over customer segments. Recall that CF methods recommend products based on the neighbors' preferences, assuming that neighbors have similar preferences. Thus, we segment the customers based on the number of neighbors in their neighborhood. As predicted, CF has the highest performance for customers with a higher number of neighbors and strictly lower performance than other methods for customers with a lower number of neighbors. On the other hand, although all methods have a lower performance in the latter segment relative to their grand average performance, PCB outperforms benchmarks in this segment. In another test, we define the popularity of the products based on their sales volume and divide the customers based on the average popularity of products that they generally prefer. The first group consists of customers who usually prefer popular products. In contrast, the second group consists of customers who are interested in other products. While the performance of all benchmark methods in the latter group is lower than their average, PCB has a better performance in the second segment relative to its average.

Finally, we propose a hybrid method named smart selection to select the best method for each customer out of all implemented methods based on the customer's previous actions. The smart selection method outperforms all other methods on average over all customers with a high margin. The selection method mainly uses CF and PCB which means that these methods are more suitable for such an environment and the PCB method successfully addresses the limitations of the CF method.

The remainder of this paper is structured as follows. In Section 2, we review the literature and related studies. In Section 3, we mainly discuss our data, model, and the implementation details respectively. Section 4 contains the results of tests and we discuss the hybrid smart selection method in Section 5. Finally, Section 6 concludes with suggestions for future work.

2. Literature review

Our work is related to two different streams of literature: recommendation systems and personalized assortment planning.

2.1. Recommendation system

Studies about recommendation systems mainly start in the mid-90s (Goldberg et al., 1992). The main objective of a recommendation system is to find the best set of items to be offered to a customer, based on the previously provided data. There are two well-known types of recommendation systems: Content-based and collaborative filtering (Burke, 2007). We provide a short review of each method in this section and refer readers for a vast review of the topics to handbooks provided by Ricci et al. (2015) and Shani and Gunawardana (2011), or the literature review by Park et al. (2012).

Content-based recommendation system (CB) is based on recommending items similar to the items which are previously preferred by the user. In this approach, the similarity of items is estimated based on the similarity of predefined attributes of the items (Mladenic, 1999; Van Meteren and Van Someren, 2000). Thus, the performance of CB methods strongly depends on the quantity and quality of explanatory information of items as the method needs sufficient information about each item to recognize the similarity. In that sense, CB methods are very powerful tools for recommending texts and articles as CB can easily recognize the similarities between two documents based on common words and phrases. In contrast, the information about a product in

a retail store can be very limited which makes it harder to find the similarities between products.

Another main limitation of CB recommendation is the overspecialization problem. By overspecialization, we mean that the recommendation system can get stuck recommending very similar items without recommending surprisingly interesting content (Ricci et al., 2015). The act of recommending surprisingly novel items is referred to as serendipity. For example, when a user searches for books by an author, recommending other books from the same author or same genre can seem correct, however, they are trivial recommendations as it is easy for the customer to find those books without the recommendations system. However, another book from another author and genre with thematically similar vibes can be exciting and interesting for the user. The importance of serendipity in the recommendations is mentioned in several studies (Ziarani and Ravanmehr, 2021; McNee et al., 2006; Toms et al., 2000; Ge et al., 2010). However, it is hard to define a parameter to measure the serendipity of the recommendations. Ge et al. (2010) suggest measuring the coverage and serendipity as indicators of a recommendation system's quality. They measure the serendipity based on the difference between the provided recommendations and expected recommendations while the expected recommendation is provided by a primitive prediction method. Although such a technique can be useful, we can argue that it is not a precise measurement technique as defining a basic primitive prediction method is challenging. Although defining and measuring serendipity is challenging, studies suggest different methods to provide serendipitous recommendations. For example, Iaquinta et al. (2008) propose a hybrid method of CB and serendipitous heuristics where serendipity is provided based on anomalies. In this setting, methods provide a set of items on which the system is more uncertain (i.e. ones for which the difference between the two parameters of positive and negative interest tends to be zero). Then, the method recommends a mix of trivial recommendations and uncertain items. In another interesting method, Campos and Figueiredo (2001) propose a method named MAX to increase the serendipity. In this study, instead of filtering out unrelated categories at the start, the Max method attempts to find serendipitous recommendations from other categories. Using such a mechanism, the method guarantees that the recommendations are non-trivial.

Collaborative filtering recommendation systems (CF) recommend items based on the idea that "like-minded users" called neighbors would have similar preferences. In that sense, A CF method determines a set of users that have similar historical activities such as similar ratings for the same movies (in a movie recommendation platform) or purchasing the same product (in an online retail store) (Hill et al., 1995; Konstan et al., 1997). Then, CF predicts a customer's preferences for a product based on the preferences of his/her neighbors for that product (Ricci et al., 2015; Ning et al., 2015; Schafer et al., 2007; Su and Khoshgoftaar, 2009). One of the main limitations of CF recommendation is the cold start problem. The cold start problem happens when there are new items or new users in the system. In the user-based cold stat problem, the data about the new user's preferences is insufficient and limited. Thus, the method cannot form a proper neighborhood and predict the user's preferences for other items. Similarly, in the item-based cold start, the data about customers' preferences for the new product is limited. Thus, the method cannot predict the preferences of the customers for the new item properly. In this study, we focus on the item-based cold start as we are considering a retailer that frequently introduces new products for a limited quantity and time.

There are several studies in the literature to address the item-based cold start problem. One of the basic approaches to address the item-based cold start is using a hybrid method. Note that a new product is not significantly distinguished from other products in the quality and quantity of explanatory attributes. Thus, content-based methods (or similar methods that rely on the predefined and explicitly known attributes of items) can be useful to address the problem. For example, Saveski and Mantrach (2014) propose a hybrid recommendation approach that exploits both the attributes of the items and the similarity of the user preferences. In this method, based on the attributes of a new item, the item is recommended to the users who may be interested in the products with those attributes. They conduct an extensive experiment and show that the proposed methods outperform benchmark methods addressing the item-based cold start problem. This is similar to previous studies such as Gantner et al. (2010) that propose using both items' and user's attributes in the CF method to address the cold start problem.

Another approach that has recently been used in several studies is implementing a deep learning method (Wei et al., 2017; Volkovs et al., 2017; Vartak et al., 2017). Note that training a deep learning method requires sufficiently large data. Thus, such methods are not suitable for the retail industry in most cases. For example, Vartak et al. (2017) propose a deep learning method based on a neural network to address the item-based cold start in recommending tweets on Twitter. Generally, the panel data of a retailer is not comparable in size with the data provided on platforms such as Twitter.

The dataset for this research is provided by an online apparel retailer that sells products on flash sales campaigns. Thus, in the data, we have plenty of new products introduced every day. Also, each product is defined with a limited number of predefined attributes (which are color, brand, and price). These issues can negatively affect the performance of both CB and CF methods. Additionally, the overspecialization problem and the importance of serendipity are more crucial in the apparel environment because generally, people are not interested in purchasing nearly the same product that they have purchased previously. Thus, we propose a personalized content-based recommendation system to address these issues. Note that due to the high number of new products, a CF method would face the item-based cold start problem. However, content-based methods are effective tools to address the item-based cold start problem. We show that our method performs strongly better than other methods in predicting customers' preferences for new products in Section 4.3. Also, we show that a hybrid consisting of our method and a CF method outperforms all benchmarks in Section 5. This indicates that our method successfully addresses the limitations of the CF method.

Additionally, our method has several advantages over classic CB methods. In our method, rather than measuring one-to-one similarity between products, we provide a preference profile for each customer based on the attributes of all products that are clicked or purchased by the customer. Then, our method predicts the customer's preferences for a product (considering its combination of attributes) based on his/her profile. This approach provides advantages in addressing the issues in the data mentioned above. First, finding similarities between products is not the main aim of our method which decreases the effect of the limited number of predefined attributes. Second, using our method, we can predict customers' interest in a new product even if neither that product nor any similar product has not been presented in the store previously. Third, in this method, we are not limited to recommending the most similar products and we can propose a large variety of products which decreases the probability of facing an overspecialization problem. Finally, our method is very compatible with panel data and can be easily implemented at any online retail store.

2.1.1. Recommendation systems in apparel environment

In this section, we provide a brief review of recommendation systems implemented in apparel and fashion environments. Although the recommendation systems literature is rich and existing techniques are well-developed, most of the techniques and studies have focused on a rating-based structure. Subsequently, those methods have a strong performance in some categories such as movies, books, and articles. In contrast, the other categories such as apparel have some challenges in adopting these methods. Most of the apparel retailers either do not have any rating system or customers do not provide any rating at the web store. Note that the basic mechanism of CF methods (finding

similar customers) is domain-independent. Thus, "click or not" and "purchasing or not" decisions in apparel retailers are imperfect proxies for ratings. Hwangbo et al. (2018) present a successful implementation of the CF method in an omnichannel apparel retailer by using online click data and offline purchase data to comprehend customers' preferences. In contrast with CF methods, CB methods run based on items' attributes. Thus, any CB method should be designed compatible with the items in the environment. In an apparel setting, CB methods may be designed to find the similarity between products based on the products' predefined attributes, description text, and/or images. For example, De et al. (2022) implement a basic CB method for women's apparel. Given an apparel product, the system generates other apparel recommendations that are similar to the focal product. They find the similarity based on the weighted similarity in product attributes such as color, title, price, image, and brand. In another study, Yeruva et al. (2022) propose a content-based recommendation that recommends products based on the similarity of the customer's search query text and the title and the description of products. Lu et al. (2022) propose a CB method to recommend a complete outfit (i.e. a combination of different products, each from a different category). They use a discrete tensor factorization model to increase the efficiency of the method in data storage size and running time. Several studies propose using deep learning and neural networks to improve the performance of CB method (Tuinhof et al., 2019; Sha et al., 2016; Suvarna and Balakrishna, 2024; Leininger et al., 2020; Gharaei et al., 2021; Khalid et al., 2021). Main part of these studies suggest neural networks to extract attributes and features of products out of products' images (Tuinhof et al., 2019; Sha et al., 2016; Khalid et al., 2021). Although most of these studies report a significant increase in the precision of the recommendation as a result, higher precision can correlate with overspecialization and lower serendipity in recommendations as mentioned earlier.

In addition to these examples, two similar studies provide detailed reviews of implementations of recommendation systems in the fashion and apparel industry (Chakraborty et al., 2021; Deldjoo et al., 2023). They discuss recommendation techniques that are vastly used in apparel retailers such as visually aware CF and generative models. Also, they discuss other aspects of product recommendation in apparel and fashion environment such as pairing, outfit recommendation, and size recommendation. Finally, they compare different studies based on the differences in the datasets and the parameters that have been used in each study.

2.2. Personalized assortment planning and ranking

Our study also relates to the studies of Assortment planning (AP) and product ranking. In traditional assortment planning, the retailer must decide on the amount and assortment of items carried in each store at each period due to store capacities (Kök et al., 2015). However, several recent studies of AP consider the assortment of products for an online retailer where physical limits change to presenting and display limits. Also, the assortment for an online retailer can be both dynamic and personalized. Thus, a relatively new stream of AP studies considers personalized and dynamic assortments. One of the first studies to propose personalized assortment planning belongs to Bernstein et al. (2015). They suggest that a firm can reserve a product (by not offering the product to all customers) for specific customers who will probably be interested in that product and no other substitute for it. They show that such a personalized assortment provides a higher profit under a limited inventory in certain conditions. Golrezaei et al. (2014) define a more general model for personalized dynamic AP. They have developed a model to maximize the revenue by considering a limited inventory assumption and personalizing assortment due to the location of arriving customers. They demonstrate that this personalization approach can lead to over 10% improvements in revenue compared to a policy that treats all customers the same. They provide a worse-case performance guarantee relative to the benchmark methods. The importance of personalized dynamic assortment planning increases with the extensive focus on e-commerce. However, there are very limited studies in this field. Also, in contrast with recommendation systems where a set of recommendations is defined for each customer individually, the personalized assortment planning methods propose different assortments to groups of customers and not each customer individually. It means that the personalization levels in these two streams are different.

Several studies suggest that not only the assortment of products but also their ranking and positioning on the web pages have a significant effect on the click-through rate (CTR), and subsequently on the demand (Ursu, 2018; Animesh et al., 2011). Thus, a retailer must decide on the optimal order of the products in addition to the assortment. For example, Ghose et al. (2012) propose a method for the ranking of the hotels on a booking website. In this study, a hotel's utility is measured based on user-generated content. Subsequently, a utility-based ranking system is proposed. In a follow-up paper, Ghose et al. (2014) show that a utility-based ranking mechanism can cause an increase in search engine revenue relative to benchmark ranking methods. Interestingly, they find that ranking may also affect customers' hotel ratings independent of their base utilities. Gallego et al. (2020) show that the ranking problem under the multinomial logit (MNL) choice model is NP-hard when prices are given. They propose heuristic algorithms that, under certain assumptions, can be shown to have a performance guarantee relative to an optimal algorithm. Further, they show that when the firm can decide on the prices, optimal ranking and pricing strategy is decoupled as follows: ranking is given by descending product utility order, and pricing is given by ascending order of markups. For a similar choice model, Aouad and Segev (2021) provide a polynomial-time approximation scheme for the display optimization problem with less restrictive assumptions. From another perspective (Lei et al., 2021) study dynamic ranking and pricing while considering the fulfillment cost and inventory constraints. The study provides a near-optimal heuristic randomization solution that selects a product set to show to each customer by sampling from a joint probability distribution of prices and ranking sample space. Finally, the study provided by Feldman et al. (2022) shows the connections and similarity of product ranking and product recommendation systems. They proposed a product recommendation system for Alibaba's online marketplace based on the MNL choice model which is vastly used in assortment planning and ranking literature. They show that MNL model generates significantly higher revenue per visit compared to Alibaba's machine learning algorithm.

Similar to assortment planning literature, the majority of studies in the ranking do not propose a personalized set of products. In that sense, ranking approaches are suitable for new customers. In contrast, a recommendation system may perform better when there is sufficient data about customer's preferences.

3. Data and model

3.1. Data

Our data is provided by an online apparel retailer that offers products in sets called campaigns for a limited time. Each campaign lasts 9 days on average and usually includes exclusive products (i.e. products are not repeatedly offered in different campaigns). Later, we refer to this business model as a flash sales market. Our data contains a 3-year history of the operations in the e-store including three main parts. The first data set consists of products offered in the e-store including their attributes (i.e. category, brand, color, and price). The data set includes about 780 000 unique products. The second data set consists of campaigns identifying the set of products offered in each of them and the campaigns' start and end dates. Finally, we have customers' interaction data including a list of products that are clicked or purchased by each customer. Note that we do not know the set of

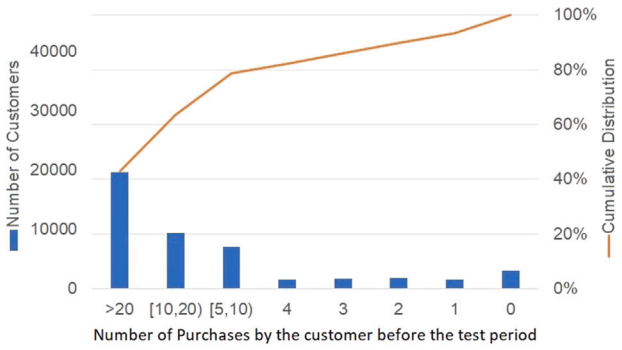


Fig. 1. Distribution of the customers based on the number of previous purchases.

products that are viewed by the customer (i.e. set of products that the customer has seen on the webpage but did not click on or purchase). In the rest of the paper, we refer to the first 80% of the time of the data set as training data and the remaining 20% time as test data or test time. The data contains about 780,000 products organized in 13,000 campaigns throughout the three years. Also, we have a record of about 80,000 customers with at least one product purchased, and out of them, 46,000 customers have purchased at least one product in the test period. Out of the latter group, only 3100 of them have no prior purchase in the data set. Fig. 1 presents the distribution of customers based on the number of previous purchases. This figure shows that more than 75% of customers have purchased more than 5 products previously which shows that the data is suitable for our analyses.

Due to the flash sales business strategy, many products are introduced every day and kept on the offer list for a short period. This means there is very limited information about customers' preferences for each product. In such a case, it is challenging to predict customer's preferences for new products. Subsequently, it is challenging for recommendation systems that rely on the prior customer–product interaction such as collaborative filtering to provide an optimal recommended product list. This problem is referred to as an item-based cold start problem. Generally, to address the item-based cold start problem, content-based methods can be used. However, content-based methods rely on the similarity of the products based on their attributes. This means that these methods require detailed predefined information about the products to configure the similarity among them. Unfortunately, the detail about each product is limited in our data set. Therefore, we propose a personalized content-based method to address both the cold-start problem and the issue with the limited number of product attributes in the next section.

3.2. Personalized content-based method

In this section, we propose a personalized content-based method to figure out customers' preferences for the current products based on their preferences over previous products in the e-store. Similar to other content-based methods, our method relies on products' attributes to find out similarities between products. We divide products' attributes into two groups based on how we extract them from the data. The first group consists of predefined attributes of each product which are mentioned in the products' data set explicitly consisting of color, brand, and price. The second group consists of operational features/attributes related to the sales or product introduction time which can easily be extracted from the other data sets. We define two operational attributes to be used in our method. The first one is the age of the product which is defined based on the time passed from the introduction of the product. This attribute plays an essential role in the store's operations as there is a lack of information about customers' demand for newly introduced products. The second operational attribute is the popularity of products which is defined based on the product's sales volume (purchase quantity).

To implement our content-based method, we represent each product with an attribute vector consisting of the product's all predefined and operational attributes. As mentioned, products' predefined attributes are category, color, brand, and price and operational attributes are popularity and age. We use discrete units to measure each attribute and feature which would help us find the similarity between products in the next steps. In that sense, we use three ordinal levels for price, popularity, and age. For instance, we represent product code 68 971 by an attribute vector $[red, Nike, expensive, bestseller, new]^T$. More formally, let us assume that each product is described with K attributes and k^{th} attribute can get L_k exclusive values. Then, let us $a(i)$, and $a_k(i)$ represent the attribute vector of product i and the value of k^{th} attribute of the product, respectively. Note that more than one product can be represented by the same attribute vector if all attributes of the products are the same. Similarly, $a_k(i)$ can be the same for different products.

We keep track of a customer's prior interactions in the e-store by measuring how many times each attribute is clicked and purchased. We define $c_{jt}^{a_k(i)}$ and $p_{jt}^{a_k(i)}$ as the number of times that all products with the corresponding attribute $a_k(i)$ are clicked and purchased by the customer j up to time t, respectively. We call $c_{jt}^{a_k(i)}$ and $p_{jt}^{a_k(i)}$ subscores in the rest of the paper. Based on these subscores, we assign a value named "score" to each attribute vector to identify the customer's interest in that combination of attributes. More formally, let us $s_{jt}^{a(i)}$ denote the score of customer j for attribute vector $a(i)$ at time t which is a function of related attribute vector and subscores as follows:

$$s_{jt}^{a(i)} = f\left(a(i), c_{jt}^{a_1(i)}, c_{jt}^{a_2(i)}, \ldots, c_{jt}^{a_K(i)}, p_{jt}^{a_1(i)}, p_{jt}^{a_2(i)}, \ldots, p_{jt}^{a_K(i)}\right) \quad (1)$$

Table 1 presents a brief overview of all notions and abbreviations used in the paper. Table 2 presents a simple example of five products, their attributes, and two customers' interactions with these products consisting of clicks and purchases. Table 3 presents the subscores calculated for all customer-attribute interactions based on the observations provided in Table 2. For example, in Table 3, the click subscore of customer 1 for the color attribute "White" is equal to 1 which means that customer 1 has clicked on exactly one product with a white color.

As mentioned, some attributes are the same in the different products, thus, the subscore of any customer for such an attribute is affected by interactions of the customer with all products that have the same attribute. For example, in Table 3, the subscore of each customer for color attribute "Blue" is calculated considering both products 43 569 and 16 697.

Let attributes color, price, and brand be represented by $k = 1$, $k = 2$, and $k = 3$, respectively. Then, Table 4 presents customers' scores for three attribute vectors based on the values presented in Table 2 and as a function of the subscores provided in Table 3. Recall that we propose PCB as a recommendation system for retailers where the number of interactions between customers and products is very limited and there are many new products. In that sense, Table 4 represents some examples showing the key potential of the PCB method in addressing the mentioned issues. As the first example, consider products 43 569 and 16 697. Although these products represent two different products, they have the same attributes and the scores for these products are equal (i.e. $s_{jt}^{a(43569)} = s_{jt}^{a(16697)}$). This helps us to find the similarities between products and use the customer's interaction history more efficiently. As the second example, consider product 12 234. Note that none of the customers have any interactions with product 12 234 which is the case for most of the new products in the e-store. However, the score function of the corresponding attribute vector of product 12 234 does not consist of all zero values as the PCB method works based on the interactions with attributes rather than products. This property of the PCB method helps us in predicting customers' preferences for new products.

Table 1
Abbreviation and notations.

Methods		
	PCB	Personalized content-Based
	RF	PCB by implementing random forest
	RFo	PCB by implementing random forest (including operational attributes)
	LF	PCB by implementing linear function
	LFo	PCB by implementing linear function (including operational attributes)
	CF	Collaborative filtering
	PPC	Popular products based on clicks
	PPP	Popular products based on purchases
	MNL	Multinomial logit
Indices		
	i	Product
	j	Customer
	k	Attribute
	t	Time
	m	Method
Notations		
	$a(i)$	Attribute vector of product i
	$a_k(i)$	kth attribute of product i
	$c_{jt}^{a_k(i)}$	Number of times that a product with attribute $a_k(i)$ is clicked by customer j by time t
	$p_{jt}^{a_k(i)}$	Number of times that a product with attribute $a_k(i)$ is purchased by customer j by time t
	$s_{jt}^{a(i)}$	Customer i's score of interest in a product with attribute $a_k(i)$ at time t
	I_{jt}^{c}	Set of products clicked by customer j until time t
	I_{jt}^{p}	Set of products purchased by customer j until time t
	N_{ij}^{c}	Binary variable indicating if product i is clicked by customer j
	N_{ij}^{p}	Binary variable indicating if product i is purchased by customer j
	A_j	Set of available products at customer j's last search session
	A_j^c	Set of products clicked by customer j's at his/her last search session
	A_j^p	Set of products purchased by customer j's at his/her last search session
	A_j^m	Set of products purchased by customer j's at his/her last search session
	$d_{jj'}$	Similarity of preferences of customers j and j'
	b_k^c	Coefficient of subscore $c_{jt}^{a_k(i)}$ in calculating score $s_{jt}^{a(i)}$
	b_k^p	Coefficient of subscore $c_{jt}^{a_k(i)}$ in calculating score $s_{jt}^{a(i)}$
	U_{mij}^{c}	Utility of product i to be clicked by customer j calculated by method m
	U_{mij}^{p}	Utility of product i to be purchased by customer j calculated by method m
	ρ_{mij}^{c}	Probability assigned by method m for product i being clicked by customer j
	ρ_{mij}^{p}	Probability assigned by method m for product j being purchased by customer j
	π_m^c	Performance of method m in predicting clicks measured by likelihood technique
	π_m^p	Performance of method m in predicting purchases measured by likelihood technique
	β_{mj}	Constant multiplier of method m to assign probabilities for customer j
	ϖ_m^c	Performance of method m in predicting clicks measured by Top5/20 technique
	ϖ_m^p	Performance of method m in predicting purchases measured by based on Top5/20 technique
	μ_i	Popularity of the products based on the sales
	$\overline{\mu}_j^t$	Average popularity of the products selected by customer j up to time t

Table 2
Examples of products, their attributes, and customers' activity history (clicks and purchases).

Product code	43569	57946	80004	16697	12234
Color	Blue	White	Red	Blue	White
Price level	Expensive	Expensive	Moderate	Expensive	Moderate
Brand	Nike	Adidas	Adidas	Nike	Adidas
Customer 1	*Clicked*	*Clicked Purchased*	–	*Clicked*	–
Customer 2	*Clicked Purchased*	–	*Clicked*	–	–

Assigning scores

In this section, we propose two heuristic methods to assign the score. First, we propose a random forest (RF) method to assign the scores. We define every customer–product interaction as an observation. In each observation, the product's attributes and the customer's related subscores are denoted as observed values, and the customer's preference is denoted as an outcome of the observed values. More formally, consider an observation where customer j has purchased product i in time t. In such an observation, $a_k(i)$, $c_{jt}^{a_k(i)}$ and $p_{jt}^{a_k(i)}$ for all $k \in K$ are given as observed values to RF where the corresponding outcome is the purchase.

To implement a random forest method in our data, we need to address two issues. First, we need to normalize the subscores such that they are comparable among customers. Thus, we normalize the click and purchase subscores by dividing them by the number of all clicks and purchases of the customer, respectively. Subsequently, all the subscores will be a number between 0 and 1.

Second, to train the RF, we need a set of observations that represents products that are not preferred by the customers. Recall that we have information on the clicks and purchases in our data, however, we assume that both clicks and purchases indicate a customer's interest in a product. Thus, we need to provide auxiliary observations where

Table 3
Examples of subscores provided based on customers' activity history.

		Corresponding subscores			
		Customer 1		Customer 2	
		$c_{1t}^{a_k(i)}$	$p_{1t}^{a_k(i)}$	$c_{2t}^{a_k(i)}$	$p_{2t}^{a_k(i)}$
Color	Blue	2	0	1	1
	White	1	1	0	0
	Red	0	0	1	0
Price	Expensive	3	1	1	1
	Moderate	0	0	1	0
Brand	Nike	2	0	1	1
	Adidas	1	1	1	0

Table 4
Examples of scores provided based on customers' subscores.

	Customer 1	Customer 2
$s_{jt}^{a(43569)} = s_{jt}^{a(16697)}$	$f\,(a(43569), 2, 3, 2, 0, 1, 0)$	$f\,(a(43569), 1, 1, 1, 1, 1, 1)$
$s_{jt}^{a(57964)}$	$f\,(a(57964), 1, 3, 1, 1, 1, 1)$	$f\,(a(57964), 0, 1, 1, 0, 1, 0)$
$s_{jt}^{a(12234)}$	$f\,(a(12234), 1, 0, 1, 1, 0, 1)$	$f\,(a(12234), 0, 1, 1, 0, 0, 0)$

Table 5
Examples of observations table in the RF method.

Observed values								Outcome
		Subscores (Normalized)						
Customer	Product	Color		Brand		Price-level		
ID	ID	$c_{jt}^{a_k(i)}$	$p_{jt}^{a_k(i)}$	$c_{jt}^{a_k(i)}$	$p_{jt}^{a_k(i)}$	$c_{jt}^{a_k(i)}$	$p_{jt}^{a_k(i)}$	
1	98134	0.12	0.10	0.25	0.05	0.40	0.33	Purchased
2	98134	0.05	0.08	0.00	0.00	0.10	0.20	Not-preferred
3	16347	0.30	0.30	0.10	0.13	0.20	0.25	Not-preferred

a product is not preferred by a customer. It is possible to label all available products that are not clicked or purchased as "not preferred", however, the number of products is too high which disturbs the balance between the number of positive and negative observations. Also, we are not sure whether the customer has seen all available products in the store or not. Thus, we narrow down this set as follows. Let us denote a campaign as "considered" if the customer clicked on or purchased at least one product out of that campaign. We assume that all products in the considered campaigns are viewed (seen) by the customer and if they are not clicked or purchased, we label them as "not preferred". Table 5 represents an example of how different observations are imported in RF implementation. In the table, each row represents an observation in our data that corresponds to an exclusive customer–product interaction. For example, the first row represents the preference of customer 1 over product 98 134. Let us assume the color of product 98 134 is red. Then, the subscore values in row 1 indicate that the 12% and 10% of products clicked and purchased (respectively) by the customer had red colors.

After training, RF is used to predict the probability that a product is clicked or purchased by a customer based on the corresponding subscores. We elaborate more on the implementation of the method in the next section.

As the second method for assigning the scores, we propose a linear function to assign the score as follows:

$$s_{jt}^{a(i)} = b_0 + \sum_K b_k^c . c_{jt}^{a_k(i)} + \sum_K b_k^p . p_{jt}^{a_k(i)} \tag{2}$$

In Eq. (2), b_0, b_k^c, and b_k^p are coefficients for calculating the total score. We use the maximum likelihood method to optimize the values for these coefficients by using the train data.

In the rest of the paper, we denote the random forest and linear scoring as RF and LF, respectively. Also, to isolate the effect of the operational attributes (i.e. age, and popularity) on the performance, we implement two versions of each method: In the first version, we only use each product's predefined attributes (i.e. category, brand, color, and price). In the second version, we use subscores for all operational and predefined attributes. We denote the latter versions as RFo and LFo.

Generally, content-based methods aim to find similar products based on the focal product's attribute. However, our method provides a preference profile for each customer based on the attributes of all products clicked and purchased by the customer previously. In that sense, our method estimates the customer's interest in each attribute. Thus, the method can predict customer's interest in a new combination of attributes although there exist no similar products (with almost the same attributes) in the history of the store. For example, a customer may be interested in red products and the Nike brand. However, she may have had no interaction (click or purchase) with a red Nike product previously. In such a case, we hypothesize that our method predicts the customer's preferences better than the basic content-based method. Also, rather than recommending the most similar product to the customer, our method can present a variety of products to the customer which will increase the serendipity in the recommendations.

3.3. Benchmark methods

In this section, we present four well-known methods as benchmarks. For each method, we define U_{mij}^c and U_{mij}^p as the value of product i to be clicked or purchased (respectively) by customer j calculated by method m. These values can be considered as equivalents of $s_{jt}^{a(i)}$ for each method.

First, we implement a user-based collaborative filtering method (CF). This method predicts a customer's preferences based on the preferences of other like-minded customers called neighbors. In our settings, to find the neighbors, the similarity between customers is calculated based on the number of common activities they have which is clicking on or purchasing the same product. A customer is a closer neighbor to the focal customer if they have more common activities. More formally, let $d_{jj't}$ denote the similarity between customers j and j' at time t, and I_{jt}^c and I_{jt}^p denote the set of products clicked and purchased by customer j up to time t, respectively. Then, we can calculate the similarity between any pair of customers based on the mentioned sets of products as follows:

$$d_{jj't} = \frac{|I_{jt}^c \cap I_{j't}^c|}{|I_{jt}^c| + |I_{j't}^c|} + \frac{|I_{jt}^p \cap I_{j't}^p|}{|I_{jt}^p| + |I_{j't}^p|} \tag{3}$$

Note that $I_{jt}^c \cap I_{j't}^c$ and $I_{jt}^p \cap I_{j't}^p$ are sets of products that are clicked on and purchased by both customers up to time t, respectively.

CF predicts the focal customer's interest in a specific product based on the preferences of his/her neighbors. In the implemented CF method, we use a weighted rating prediction method to predict a customer's interest in a product. In a weighted rating method, the preference of the focal customer is estimated based on the weighted average of the product's preference by her/his neighbors. This means that the preference of a closer neighbor has a higher weight on the estimated preference. More formally, let us define N_{ij}^c as a binary variable which is one if product i is clicked by customer j up to the search session and is zero otherwise. Similarly, we define N_{ij}^p as a binary variable which is one if product i is purchased by customer j up to the search session and is zero otherwise. Then, predicted preference of customer j for product i is as follows:

$$U_{mij}^c = \sum_{j' \in J} d_{jj'} . N_{ij'}^c \tag{4}$$

$$U_{mij}^p = \sum_{j' \in J} d_{jj'} . N_{ij'}^p \tag{5}$$

Second, we implement a heuristic method based on the popularity of the products. Many online retailers offer their popular products in different positions on their web pages with various names. For example, Amazon recommends popular products titled "best sellers". Thus, we implement a method that simply selects the most clicked and purchased products at the test time. We denote the method that proposes the most

Table 6
Qualitative comparison of all implemented methods.

Methods	Information	Personalized	Advantages	Limitations
RF/LF	Explicit	Yes	New products	New customer
RFo/LFo	Explicit + Implicit	Yes	New products	New customer
CF	Implicit	Yes	Unobserved values	New products + New customers
PPC/P	Implicit + Explicit	No	Unobserved values + New customer	New product
MNL	Explicit	No	New customer	Lack of explanatory data

clicked products as PPC and the method that proposes the most purchased products as PPP. Intuitively, the utility of product i for customer j is assigned equal to the products' click and purchase numbers in the PPC and PPP methods, respectively. Formally, in the PPC method:

$$U^c_{mij} = U^p_{mij} = \sum_{j' \in J} N^c_{ij'} \tag{6}$$

and in the PPP method:

$$U^c_{mij} = U^p_{mij} = \sum_{j' \in J} N^p_{ij'} \tag{7}$$

Finally, we implement a heuristic content-based method using the multinomial logit choice model (MNL). In the MNL model, we assume that each customer determines a value for a product which is a function of the product's attributes and the personal preference of the customer for the product. Formally, let us assume that the value of product i for customer j is equal to v_{ij}, Then:

$$v_{ij} = \gamma.a(i) + \epsilon_j \tag{8}$$

In Eq. (8), γ is the vector of coefficients, $a(i)$ is the attribute vector of product i, and ϵ_j represents the individual preference of each customer. In MNL models, ϵ has a Gumble distribution. We train the MNL model using the train data to optimize the values of γ. Then, we can calculate the average utility of purchasing any product in the test time as follows:

$$U^c_{mij} = U^p_{mij} = \frac{e^{\gamma.a(i)}}{\sum_{i' \in I} e^{\gamma.a(i')}} \tag{9}$$

Note that MNL does not provide personalized recommendations. In that sense, the value of U^c_{mij} and U^p_{mij} are the same for all customers for the same product.

Table 6 presents all methods implemented in this paper and compares them based on basic differences in each method. In this table, we divide the information used by each method into two different categories named Explicit and Implicit. Explicit information consists of explicitly defined attributes in the dataset such as color, brand, price, and product introduction time. On the other hand, implicit information consists of all unobserved variables that can affect customers' preferences however it is not explicitly mentioned in the data such as design elements, discounts, and advertisements. While methods such as RF, LF, and MNL run based on explicit information, methods such as PPC, PPP, and CF can indirectly predict customer's interest in some products based on the implicit information (generally based on the sales). Also, in the table, we show whether a method provides personalized recommendations or not. Finally, we present whether each method is suitable for recommending new products and recommending products to new customers.

3.4. Performance measurement

Generally, during a customer's visit to the e-store, the customer may click on several products and purchase more than one product. We name all the activity of the customer in a single visit as a search session. We identify a search session of customer j as all consecutive activities (clicks and purchases) without any idle time (i.e. a period without any clicks or purchases) of more than one hour. Then, we measure the performance of each method by its accuracy in predicting the customers' choices (i.e. clicks and purchases) in the customer's last search session if it is in the test period (i.e. last 20% of the time span of the data). To have an equal weight of the customers on the average performance of any method, we only select one search session of each customer and to have a better use of the data, we select the last search session of each customer.

Table 7
An example of the probability distribution by different methods.

	Customer 1			Customer 2		
Product code	99873	12135	20056	99873	12135	20056
Prob. distribution by PCB	0.5	0.4	0.1	0.3	0.3	0.4
Prob. distribution by CF	0.2	0.6	0.2	0.2	0.1	0.7
Customer's choice (Click)	No	Yes	No	Yes	No	No

We define products that are available at the website in the last search session of customer j as the set of available products denoted by A_j, and products that are clicked or purchased by the customer by A^c_j and A^p_j, respectively. All activities of all customers before the considered search session and the set of available products during the search session are available information for all methods.

We measure the performance of methods using two different techniques. In the first technique, we implement a likelihood-based measuring technique. In this technique, for each search session, each method assigns a probability distribution to the set of available products representing the probability that each product is going to be selected by the customer. In that sense, a method predicts a higher chance for a product to be selected by the customer out of all available products by assigning a higher probability. Then, the performance of a method is the sum of the probabilities assigned to the chosen products.

Formally, let us ρ^c_{mij} and ρ^p_{mij} denote the probability assigned by method m that product i is going to be clicked and purchased by customer j, respectively. Additionally, let π^c_m and π^p_m denote the performance of method m in predicting clicks and purchases, respectively. Then:

$$\pi^c_m = \frac{\sum_j (\sum_{i \in A^c_j} \rho^c_{mij})}{\sum_j (\sum_{i \in A_j} \rho^c_{mij})} \tag{10}$$

$$\pi^p_m = \frac{\sum_j (\sum_{i \in A^p_j} \rho^p_{mij})}{\sum_j (\sum_{i \in A_j} \rho^p_{mij})} \tag{11}$$

As formalized in Eqs. (10) and (11), the performance of each method in predicting clicks and purchases is equal to the average of the performance of each method over all search sessions and the performance in each search session is the summation of the probabilities assigned to products that are clicked on or purchased.

Consider Table 7 as an example of a probability distribution of two methods over three available products for two customers. In this example, PCB has an average performance of 0.35 $((0.4 + 0.3)/2)$, and CF has an average performance equal to 0.4 $((0.6 + 0.2)/2)$.

We implement two variants of the mentioned likelihood measurement technique. In the first variant, each method assigns probabilities such that for each specific search session the summation of the assigned probabilities over all available products must be less than or equal to 1. In other words, we normalize the summation of assigned probabilities for each session to 1. Note that it is possible that a customer selects several products in a search session (i.e. clicking on several products or purchasing more than one product). Then, if a method predicts a customer's all actions (all clicked products or all purchased products) in the tested search session (meaning that the method only

assigns positive probabilities to the selected products), it will receive a performance equal to exactly 1. Thus, each customer has the same weight in the average performance of each method. In other words, the performance of a method for a customer is not dependent on the number of items selected by the customer. Then, formally, we have the following equations for calculating the performance of each method:

$$\pi_m^c = \frac{\sum_j (\sum_{i \in A_j^c} \rho_{mij}^c)}{|J|} \tag{12}$$

$$\pi_m^p = \frac{\sum_j (\sum_{i \in A_j^p} \rho_{mij}^p)}{|J|} \tag{13}$$

In the second variant, the probability assigned to each product by a method is limited by 1 but the summation of the probabilities assigned over all available products may pass 1 and be equal to the number of products selected by the customer (A_j^c and A_j^p). In that sense, the average performance of each method is weighted based on the number of products selected by the customer. Thus, predicting the selected products of customers with a higher number of clicks or purchases has a higher impact on the grand average performance of each method. Formally, in this variant the performance of each method is calculated as follows:

$$\pi_m^c = \frac{\sum_j (\sum_{i \in A_j^c} \rho_{mij}^c)}{\sum_j |A_j^c|} \tag{14}$$

$$\pi_m^p = \frac{\sum_j (\sum_{i \in A_j^p} \rho_{mij}^p)}{\sum_j |A_j^p|} \tag{15}$$

We denote the first variant as EWPA (Equal Weight Probability Assignment) and the second variant as AWPA (Activity-Weighted Probability Assignment) in the rest of the paper.

In all PCB methods (i.e. RF, RFo, LF, and LFo), we assign a probability to each product proportional to the score of the corresponding attribute vector of that product as follows:

$$\rho_{mij}^c = \rho_{mij}^p = \beta_{jm} s_j^{a(i)} \tag{16}$$

In Eq. (16), β_{jm} is a constant multiplier that is calculated according to each measurement variant's constraint (i.e. probability distribution limits). Similarly, in all benchmark methods, we assign probabilities proportional to the calculated utilities U_{mij} as follows:

$$\rho_{mij}^c = \beta_{jm} U_{ijm}^c \tag{17}$$

$$\rho_{mij}^p = \beta_{jm} U_{ijm}^p \tag{18}$$

In the second performance measurement technique, we measure the performance of the methods by their efficiency in offering a limited number of products. In that sense, each method selects "Top 5" and "Top 20" products to be offered to each customer. Each method gains a positive value for each product that is in both the selected list and the customer's chosen products (clicked and purchased). Let us A_j^m denote the set of top products selected by method m to be offered/recommended to customer j. Then, the performance of each method is measured as follows.

$$\varpi_m^c = \frac{\sum_j (|A_j^m \cap A_j^c|)}{|J|} \tag{19}$$

$$\varpi_m^p = \frac{\sum_j (|A_j^m \cap A_j^p|)}{|J|} \tag{20}$$

In Eqs. (19) and (20) the performance of a method for a customer is equal to the number of products clicked and purchased (respectively) from the set of products selected by the method (i.e. A_j^m) and the overall performance of the method is the average performance for all customers.

Note that one of the main purposes of the recommendation systems is to increase the conversion rate. In that sense, in addition to measuring the accuracy in predicting customers' preferences, we should measure the conversion rate or more specifically, measure the effect of recommending different sets of products in customer abundance of the store. However, we do not have any data about the customers who have visited the store and have not clicked or purchased any product. Also, we do not have any data about the set of products presented to each customer. However, we hypothesize (without testing) that a recommendation system with a better prediction of customers' preferences will increase the conversion rate as well.

4. Results

4.1. General performance

In this section, we present the general performance of all methods in predicting the clicks and purchases in the search sessions. Table 8 presents the performance of each method (including their average on the left and standard deviation in the parentheses on the right) based on the mentioned measurement techniques.

Based on probability assignment measurement techniques (both EWPA and AWPA), RFo is the best method for predicting clicks. On the other hand, RFo, LF, and CF have similar performance values in predicting purchases. However, LF and RFo perform slightly better when measuring by EWPA and APWA, respectively.

Recall that in the AWPA measurement technique (activity-weighted probability assignment), a customer with a higher number of clicks or purchases has a higher impact on the average performance value of each method. As shown in the table, all methods have a performance value when we use the AWPA measurement technique. This indicates that methods are more precise in predicting the preferences of customers with a higher number of clicks and purchases. Several potential reasons could explain this phenomenon. First, customers with a high number of clicks and purchases during the test period likely had higher activity levels in the past as well. This historical activity enables methods such as PCB and CF to better configure and predict their preferences. Second, these customers may spend more time in the e-store, engaging with and exploring a wider variety of products. This increased interaction not only enhances their exposure to recommendations but also raises the likelihood of them discovering products that align with their preferences.

Note that the CF method has the highest performance difference when measured by EPWA and APWA. Additionally, Table 8 shows that the performance of CF has a very high variance. These results implies that the performance of CF is highly dependent on the customers. As the CF method predicts a customer's actions based on his/her neighbors, we hypothesize that the variance in the performance of CF is caused by variance in the strength of customers' neighborhoods. We analyze this issue in Section 4.4 and provide further details.

On the other hand, based on the top 5/20 (selecting a limited list) measurement techniques, PPC and PPP outperform other methods by a significant margin. Recall that we do not have the data of the products that are viewed by the customers and in that sense, we are assuming all available products as viewed. However, there is a high probability that most of the available products have not been seen by the customer. On the other hand, we know that on the retailer's website, popular products are offered regularly, which may boost the performance of PPC and PPP. We refer to this effect as "effect of viewed products". We discuss this effect in the next section in more detail.

We have a particular application of MNL model. In this application, we have defined the MNL model based on its typical implementation which is using the known attributes of a product to determine the attractiveness of the product. It seems from the strictly lower performance of the MNL model compared to the other models that these attributes do not sufficiently explain the attractiveness of the product for the customer. In other words, using three attributes for each product, MNL cannot efficiently differentiate the products. Thus,

Table 8
General performance of methods in predicting clicks and purchases.

Method	Predicting clicks				Predicting purchases			
	EWPA	AWPA	Top 5	Top 20	EWPA	AWPA	Top 5	Top 20
RF	13.3 (23.9)	22.2 (40.9)	68 (298)	243 (770)	3.3 (11.4)	5.8 (17.6)	20 (124)	93 (330)
RFo	16.3 (23.1)	25.7 (42.3)	95 (320)	300 (810)	3.9 (13.1)	5.9 (19.0)	26 (139)	123 (393)
LF	14.8 (25.4)	22.3 (44.0)	78 (316)	267 (761)	4.0 (12.6)	5.6 (18.8)	27 (141)	122 (354)
LFo	15.2 (24.2)	20.4 (40.8)	80 (350)	275 (730)	3.9 (14.1)	5.6 (19.2)	26 (133)	118 (333)
CF	15.6 (52.5)	24.0 (64.4)	82 (316)	285 (761)	3.2 (15.1)	5.7 (25.5)	19 (141)	75 (354)
PPC	12.0 (20.4)	24.5 (30.7)	278 (757)	857 (2137)	1.0 (1.6)	1.2 (2.2)	23 (160)	62 (286)
PPP	9.2 (12.4)	15.7 (18.3)	150 (397)	520 (1008)	2.1 (3.5)	3.8 (5.3)	45 (223)	138 (447)
MNL	2.9 (6.9)	4.3 (11.3)	7 (117)	39 (376)	0.6 (1.2)	1.1 (2.3)	1 (34)	6 (100)

∗ The standard deviation of each method's performance is presented in the parentheses next to the average performance.
∗∗ All values are multiplied by 1000 for ease of reading.

the MNL method generally distributes the probabilities over all available products more uniformly than other methods. As a result, the performance of MNL is not comparable to any other method in any of the measurement techniques. Therefore, we propose that the main reason for the poor performance of MNL is the lack of explanatory data (i.e. number of product attributes).

Note that there is no significant difference between LF and LFo, while, RFo performs significantly better than RF. This shows that operational attributes are more effective in the RF than linear PCB. In the rest of the paper, for ease of reading and elimination of redundancy, we present the slightly better type of each method which is RFo and LF.

4.2. Consideration set bias

Let us refer to the set of all products that are viewed by a customer as the consideration set of that customer. We know that the set of available products in the store at any time is too large. Thus, we expect the consideration set to be a very small subset of available products for most of the customers. However, our data does not include the consideration set of each customer. In such a case, methods such as PCB and CF may assign a positive probability to products that are not viewed by the customer but can be interesting for them. On the other hand, we know that the website recommends popular products regularly, which enhances the probability of these products being viewed by the customers and being in the customers' consideration set. Recall that in Table 8 PPC and PPP have the best performance in selecting the top 5/20 products. Thus, the performance of different methods can be biased.

To reduce the bias described above, we implement the following test. We select a set of top 50 products with the highest number of clicks and purchases in the test time. On average, this set has received about 50% of clicks and purchases although the set includes only about 5% of all available products. This information shows that this set of products is offered to and visited by the customers most of the time. We run a new test where we replace the available product set with the set of top 50 products. This means in both probability assigning techniques each method assigns a set of probabilities for each customer over this set (Top 50) and in the Top 5/20 technique each method selects the nominates out of this set. This reduces the chance that a method assigns a positive probability (or selects it for the top 5/20) to a product that is not been viewed by the customers. Subsequently, the mentioned bias is reduced. Results of the test are provided in Table 9.

As predicted, by reducing the bias, PPC/P methods lose their advantage and get outperformed by other methods even when we measure by selecting the top 5 or top 20 products. Note that on average, a product in this set has received 10 times more clicks and purchases than a product in the available products set. This is the main reason for higher performance values in Table 9 in comparison with the previous section. Note that limiting the available set is less effective for methods such as RFo, CF, and PPP/C. This is mainly because these methods can easily recognize popular products and distribute higher probability to these products. It means that even without limiting the set, they recommend these products more (i.e. each method sets a higher probability for a product in this set). In contrast, LF has the highest performance increase. As a result, LF has the best performance in predicting both clicks and purchases in all measurement types in this set. The significant increase in the performance of LF shows that probably some of the products selected from the available products set have not been viewed by the customer. Considering the high performance of LF in this test, we propose that using LF as a recommendation system will expand the variety of the products offered to the customer. Subsequently, it can increase the sales of less-promoted products.

4.3. New products

Recall that due to the flash sales strategy, the retailer offers several new products every day. As mentioned, predicting customers' interest in new products is more challenging due to the lack of information about these products. Thus, we are interested in the performance of each method in recommending new products. We define a product as new if it is introduced in the last 24 h. Then, we measure the accuracy of each method in predicting clicks and purchases of new products.

Table 10 presents the performance of each method in this test. As we expect, LF and RFo outperform other methods by large margins showing that LF and RFo are the best methods to recommend new products in our setting. We conclude that our method is an efficient way to address the cold start problem. In that sense, we propose that using our method as a recommendation system will help a retailer boost the customers' interaction with new products and subsequently increase the sales of these products. It means that PCB methods are advantageous tools to be used in similar environments where a retailer has a high portion of new products.

4.4. Customer-based segmentations

In this section, we provide further analyses of the performance of each method over different customer groups. Thus, we present the results based on the EWPA measurement technique where each customer has an equal weight on the overall performance of each method.

Neighborhood segmentation

We have observed that the performance of CF has a very high variance in the customers. Note that predictions in the CF method are based on the neighborhood of the focal customer. Thus, we hypothesize that the variance in the performance of CF is caused by variance in the strength of customers' neighborhoods. First, to measure the strength of the neighborhood of a customer, we propose the following parameters: The number of neighbors, the number of neighbors that have at least two occasions where they have selected the same product, the average nearness of the neighborhood, and the number of the neighbors with a higher nearness level than a threshold α ($\alpha = 0.8, 0.7, 0.5$). Then, to test our hypotheses, we implemented several linear regressions where we assigned the performance of the CF method as the dependent variable

Table 9
Performance of the methods in predicting clicks and purchases in the set of top 50 popular products.

Method	Predicting clicks				Predicting purchases			
	EWPA	AWPA	Top 5	Top 20	EWPA	AWPA	Top 5	Top 20
RF	64 (50.9)	84 (61.7)	306 (613)	1043 (1567)	15.1 (28.8)	19.1 (34.2)	75 (280)	287 (700)
RFo	66 (48.9)	87 (66.1)	401 (667)	1104 (1700)	16.9 (27.4)	22.0 (37.5)	79 (289)	276 (854)
LF	79 (51.1)	99 (75.2)	453 (678)	1192 (1801)	18.2 (25.5)	23.6 (35.0)	84 (260)	301 (782)
LFo	77 (52.1)	96 (74.4)	420 (714)	1133 (1500)	18.0 (28.0)	23.1 (34.8)	83 (290)	304 (841)
CF	60 (125)	94 (170)	318 (1100)	1074 (2531)	14.7 (56.4)	21.8 (75.1)	66 (408)	310 (1080)
PPC	38 (48.8)	50 (71.9)	283 (757)	889 (2137)	5.7 (3.6)	7.5 (5.5)	23 (162)	90 (300)
PPP	31 (30.9)	42 (58.4)	170 (427)	561 (1092)	12.1 (8.6)	16.6 (10.5)	48 (241)	151 (460)
MNL	27 (9.8)	39 (20.4)	130 (269)	390 (927)	3.6 (2.0)	5.2 (2.9)	21 (45)	78 (130)

∗ The standard deviation of each method's performance is presented in the parentheses next to the average performance.
∗∗ All values are multiplied by 1000 for ease of reading.

Table 10
Performance of each method in predicting customers' preferences in new products.

Method	Predicting clicks				Predicting purchases			
	EWPA	AWPA	Top 5	Top 20	EWPA	AWPA	Top 5	Top 20
RF	10.3 (20.9)	12.4 (24.0)	55 (210)	193 (728)	6.6 (19.9)	7.8 (21.9)	35 (167)	107 (501)
RFo	12.1 (22.2)	14.3 (25.3)	65 (245)	211 (737)	7.1 (24.7)	9.2 (27.3)	40 (192)	120 (543)
LF	11.8 (25.1)	14.0 (26.6)	67 (289)	203 (800)	7.8 (26.4)	9.5 (28.0)	44 (240)	139 (607)
LFo	11.6 (39.9)	13.8 (26.5)	64 (270)	198 (789)	7.2 (26.7)	8.9 (27.5)	41 (231)	131 (592)
CF	3.6 (19.4)	4.1 (20.7)	20 (160)	86 (494)	1.7 (10.1)	2.5 (11.2)	11 (112)	47 (305)
PPC	0.4 (1.2)	0.5 (1.2)	0.0 (0.0)	0.0 (0.0)	0.1 (0.4)	0.1 (0.4)	0.0 (0.0)	0.0 (0.0)
PPP	2.2 (7.5)	2.5 (8.0)	39 (240)	108 (555)	1.0 (3.4)	1.2 (3.6)	28 (183)	63 (348)
MNL	1.3 (6.4)	1.7 (6.7)	7 (45)	39 (102)	0.4 (1.2)	0.6 (1.3)	1 (12)	7 (88)

∗ The standard deviation of each method's performance is presented in the parentheses next to the average performance.
∗∗ All values are multiplied by 1000 for ease of reading.

in all regressions and each of the parameters above as the independent variable in each exclusive regression. As a result, we find that the number of neighbors and the number of neighbors that have at least two similar activities have significant effects on the CF performance.

To illustrate the variance in the performance of methods, we order customers based on the mentioned parameters (i.e. the number of neighbors and the number of neighbors that have at least two similar activities) and partition them into ten deciles. We present the performance of each method in the first and the last decile of customers as two extreme cases compared to the grand average performance. We denote the first decile as "Top" where the number of the neighbors for each customer is higher and we denote the last decile as "Bottom" where the number of the neighbors for each customer is lower, and we these two segments with the grand average denoted as "Average". Results are presented at Tables 11 and 12.

We observe that all methods have a lower performance level at the bottom segment. Recall that customers in the bottom segment have fewer neighbors than other segments. This means that either these customers are new or their preferences are incompatible with the majority of the customers. These reasons cause all methods to have a challenge in predicting their preferences. However, the variance in CF's performance is noticeable in these tables. While CF outperforms all methods in the top decile in both predicting clicks and purchases by a high margin, it has a weak performance in the bottom decile. For example, in Table 11 CF is outperformed by all methods predicting purchases in the bottom decile, and in Table 12 it has lower performance in predicting clicks than RFo, PCB, PPC, and PPP methods in that decile.

Note that the performance of CF in predicting purchases for the bottom decile is almost zero (rounded to tenths). After investigating the reason behind that, we found that the bottom segment has no neighbors and subsequently, CF sets equal probabilities on all available products which results in a performance so close to zero. On the other hand, RFo and PCB are the best methods to predict clicks and purchases, respectively, for the bottom segment in both tables.

Table 11
Effect of the number of neighbors on the performance.

Method	Predicting clicks			Predicting purchases		
	Bottom	Average	Top	Bottom	Average	Top
RFo	12.0 (14.8)	16.3 (23.1)	16.8 (20.9)	1.9 (2.3)	3.9 (13.1)	4.2 (14.3)
LF	9.9 (12.3)	14.8 (25.4)	14.8 (26.6)	2.1 (2.4)	4.0 (12.6)	4.6 (13.8)
CF	9.8 (62.5)	15.6 (52.5)	18.5 (31.6)	0.0 (0.0)	3.2 (15.1)	5.2 (13.4)
PPC	9.6 (17.5)	12.0 (20.4)	14.7 (25.2)	0.8 (1.4)	1.0 (1.6)	1.2 (2.1)
PPP	6.6 (10.2)	9.2 (12.4)	12.7 (15.6)	1.3 (2.1)	2.1 (3.5)	3.2 (4.6)
MNL	2.7 (14.2)	2.9 (6.9)	3.0 (5.6)	0.6 (1.2)	0.6 (1.2)	0.6 (1.3)

∗ The standard deviation of each method's performance is presented in the parentheses next to the average performance.
∗∗ All values are multiplied by 1000 for ease of reading.

Table 12
Effect of structure of neighborhood on the performance.

Method	Predicting clicks			Predicting purchases		
	Bottom	Average	Top	Bottom	Average	Top
RFo	11.9 (15.6)	16.3 (23.1)	16.8 (21.2)	1.8 (2.4)	3.9 (13.1)	4.7 (14.5)
LF	11.6 (12.8)	14.8 (25.4)	15.6 (28.4)	2.0 (2.3)	4.0 (12.6)	4.8 (13.1)
CF	6.1 (60.1)	15.6 (52.5)	22.2 (37.1)	1.8 (12.9)	3.2 (15.1)	7.0 (16.3)
PPC	10.0 (19.8)	12.0 (20.4)	13.8 (22.6)	1.0 (1.5)	1.0 (1.6)	1.2 (2.0)
PPP	7.6 (10.9)	9.2 (12.4)	13.2 (16.5)	1.7 (2.6)	2.1 (3.5)	3.6 (5.1)
MNL	2.9 (10.2)	2.9 (6.9)	2.9 (6.0)	0.6 (1.1)	0.6 (1.2)	0.8 (1.4)

∗ The standard deviation of each method's performance is presented in the parentheses next to the average performance.
∗∗ All values are multiplied by 1000 for ease of reading.

Preference segmentation

Similar to CF, PPC/P methods have a high performance variance in the customers. Thus, we define a parameter based on the popularity of products purchased by the customers. Let μ_i denote product i's popularity which is equal to the number of times that the product is purchased totally. Then, we define $\overline{\mu}_j^t$ as customer j's average interest in popular products at time t which is equal to the average popularity of products that the customer has purchased up to that time. Formally:

$$\overline{\mu}_j^t = \frac{\sum_{i \in I_{jt}^p} \mu_i}{|I_{jt}^p|} \tag{21}$$

Table 13
Effect of customers' average preference popularity on the performance.

Method	Predicting clicks			Predicting purchases		
	Bottom	Average	Top	Bottom	Average	Top
RFo	16.9 (25.5)	16.3 (23.1)	16.0 (24.1)	4.0 (9.2)	3.9 (13.1)	3.9 (9.4)
LF	16.6 (22.4)	14.8 (25.4)	14.8 (21.8)	4.3 (12.8)	4.0 (12.6)	4.0 (12.5)
CF	8.8 (30.3)	15.6 (52.5)	15.5 (37.8)	0.6 (4.2)	3.2 (15.1)	4.0 (14.0)
PPC	12.0 (18.2)	12.0 (20.4)	10.4 (19.6)	0.8 (1.2)	1.0 (1.6)	1.1 (1.8)
PPP	6.6 (9.8)	9.2 (12.4)	9.7 (13.5)	1.2 (1.8)	2.1 (3.5)	2.5 (4.1)
MNL	2.9 (14.0)	2.9 (6.9)	2.9 (4.9)	0.6 (1.1)	0.6 (1.2)	0.6 (1.1)

∗ The standard deviation of each method's performance is presented in the parentheses next to the average performance.
∗∗ All values are multiplied by 1000 for ease of reading.

We implement several linear regressions and assign the performance of each method as the dependent variable in an exclusive regression and $\overline{\mu}_j^t$ as the independent variable in all regressions. We found that $\overline{\mu}_j^t$ has a significant effect on the performance of PPC/P method as well as several other methods. To illustrate the performance of methods in different customer segments, we divide all customers into ten ordinal deciles based on $\overline{\mu}_j^t$. We mention the first and the last decile of customers as "Top" and "Bottom", respectively (where the top decile contains customers with the highest average popularity of preferences). The performance of each method in the bottom and top customer deciles in comparison with the grand average denoted by "Average" is provided in Table 13.

In contrast with other methods, our methods (especially LF) have better performance at the bottom segment (customers who do not prefer popular products generally). We can conclude our methods are able to capture idiosyncratic preferences better than basic methods as they utilize information on the level of customer-attribute interaction. Interestingly, the effect of $\overline{\mu}_j^t$ on the performance of CF is even higher than PPC and PPP methods. We observe that CF has a challenge in predicting both clicks and purchases of the bottom decile.

5. Extension: Hybrid method

Recall that we show that the following three parameters have a significant impact on the performance of each method: (i) the average popularity of products selected by a customer, (ii) the number of neighbors, and (iii) the number of neighbors with more than one common preferred product. Thus, we are motivated to implement a hybrid method by selecting the best method for each customer based on these parameters. To do so, we implement an RF which decides on the method to provide the predictions. We denote this RF hybrid method as "smart selection". Table 14 presents the percentage of using each method for prediction in smart selection. Table 15 presents the performance of smart selection in comparison to all other methods. As predicted, smart selection outperforms all methods in all tests using the best one in each customer segment. These results show that the parameters defined previously such as the number of neighbors, and $\overline{\mu}_j^t$ are practically useful to have better predictions. We observe that CF is used in more than half of the cases. After CF, our content-based methods (i.e. LF and RF) are used around one-third of the time. This is aligned with our results that show CF has high performance when the neighborhood is constructed well, otherwise, LF and RF perform better.

As mentioned earlier, we propose the PCB methods to cover the limitations of other methods specifically CF. Although the general results presented in Section 4.1 show very similar performance levels for the CF and PCB methods, the results in this section indicate that these methods are complementary to each other and PCB can successfully address the limitation of the CF method.

Table 14
Smart selection of the methods.

	RFo/LF	CF	PPC/PPP
Predicting clicks	34%	55%	11%
Predicting purchases	36%	55%	9%

6. Conclusions and future research

In this article, we propose a novel personalized content-based method (PCB) to predict customers' preferences for products in an online apparel retailer. Our method is a useful tool to select a subset of products at the e-store to be recommended in various points such as the first page, sidebar recommendations, and search pages. We have developed this method considering a flash seller online apparel retailer that offers sets of products in a limited quantity and time. Generally in such an environment, due to the short period of product availability in the store, the number of interactions between customers and products and subsequently information about the customers' preference for the products is very limited. The limited interactions of customers with the new products make it challenging for some of the recommendation systems such as collaborative filtering as it is harder to comprehend the customer's preferences and provide useful recommendations. This problem is referred to as the "item-based cold start" problem. One of the practical solutions to address the cold start problem is using a content-based method. However, the performance of content-based methods depends on the quality and detail of products' explanatory data, especially the quantity and quality of explicitly recorded attributes of each product in the data such as color, brand, and price. However, in our case, the explanatory information about the product in our dataset provided by the retailer is also limited. Thus, we propose a personalized content-based recommendation system to address both item-based cold start and limited explanatory information (the number of explicitly recorded attributes of each product) to predict customers' preferences. We use the number of interactions (clicks and purchases) of each customer with each specific attribute to provide a preference profile for the customer. Then, we represent each product with a corresponding attribute vector consisting of all explicitly known attributes of a product, e.g. color, price level, and brand. Finally, for each customer and each attribute vector, we can assign a score of interest to represent the customer's interest in that combination of attributes. Each score is a function of the number of times that each attribute in the vector is clicked or purchased by the customer. This method has several advantages over classic CB methods. Rather than measuring one-to-one similarity between products, our method estimates the customer's interest in each attribute; Thus, the method can predict the customer's interest in a new combination of attributes although there exist no similar products (with almost the same attributes) in the history of the store. Also, rather than recommending the most similar product to the customer, our method can present a variety of products to the customer which will increase the serendipity in the recommendations.

We test our method using a data set provided by an apparel retailer and compare it with benchmarks such as a collaborative filtering method (CF), two heuristic methods recommending popular products (PPC and PPP), and another heuristic method based on the multinomial logit choice model (MNL). We measure the performance of each method by its accuracy in predicting each customer's preferences in the test time. This means that given the "available products set" and customers' history up to the test time, the method predicts which products would be clicked or purchased by the customer after that time. Results show that our method outperforms other methods in predicting products that are clicked and purchased by the customers with a slight margin in comparison with the CF Method.

Recall that the mentioned apparel retailer has a flash sale business strategy which means that every day many new products are introduced in the e-store. Also, we know that a large number of new products

Table 15
General performance of the smart selection method in comparison with other methods.

Method	Predicting clicks				Predicting purchases			
	EWPA	AWPA	Top 5	Top 20	EWPA	AWPA	Top 5	Top 20
Smart	19.8 (28.2)	28.6 (58.1)	114 (389)	354 (827)	6.0 (13.3)	9.7 (21.4)	40 (150)	156 (360)
RF	13.3 (23.9)	22.2 (40.9)	68 (298)	243 (770)	3.3 (11.4)	5.8 (17.6)	20 (124)	93 (330)
RFo	16.3 (23.1)	25.7 (42.3)	95 (320)	300 (810)	3.9 (13.1)	5.9 (19.0)	26 (139)	123 (393)
LF	14.8 (25.4)	22.3 (44.0)	78 (316)	267 (761)	4.0 (12.6)	5.6 (18.8)	27 (141)	122 (354)
LFo	15.2 (24.2)	20.4 (40.8)	80 (350)	275 (730)	3.9 (14.1)	5.6 (19.2)	26 (133)	118 (333)
CF	15.6 (52.5)	24.0 (64.4)	82 (316)	285 (761)	3.2 (15.1)	5.7 (25.5)	19 (141)	75 (354)
PPC	12.0 (20.4)	24.5 (30.7)	278 (757)	857 (2137)	1.0 (1.6)	1.2 (2.2)	23 (160)	62 (286)
PPP	9.2 (12.4)	15.7 (18.3)	150 (397)	520 (1008)	2.1 (3.5)	3.8 (5.3)	45 (223)	138 (447)
MNL	2.9 (6.9)	4.3 (11.3)	7 (117)	39 (376)	0.6 (1.2)	1.1 (2.3)	1 (34)	6 (100)

∗ The standard deviation of each method's performance is presented in the parentheses next to the average performance.
∗∗ All values are multiplied by 1000 for ease of reading.

causes the cold start issue for the CF and PP methods. Thus, we define products introduced in the last 24 h as "new products" and measure the performance of the methods in recommending them. As expected, we found that PCB has a strictly better performance in recommending new products. Thus, we propose our method as an effective method to address the cold-start problem.

Also, we notice that the performance of the benchmark methods has a very high variance over different customer segments. To understand one of the main reasons behind this variance, one must know that CF methods recommend products based on the preferences of like-minded customers named neighbors, assuming that neighbors have similar preferences. Thus, the performance of the CF method is highly dependent on the strength of the neighborhood. We show that the variance in the performance of the CF method is correlated to the number of neighbors in their neighborhood. To measure the effect of this factor on the performance of each method, we divide customers into ten ordinal deciles where the last decile is a set of customers with the lowest number of neighbors. Although all method has a lower performance in the last decile, PCB has a higher performance than other methods with a high margin.

Another factor that affects the performance of the several methods is the popularity of products that customers generally prefer. We define the popularity of each product by the number of times that the product is purchased. Again, to measure the effect of this factor on the performance of each method, we divide customers into ten ordinal deciles. The first decile consists of customers who usually prefer popular products. In contrast, the last deciles consist of customers who are interested in other products. While the performance of all methods in the first decile is almost similar to the grand average, CF, PPC, and PPP have a strictly lower performance in the last decile. In contrast, PCB has a higher performance in the last decile than the grand average and outperforms all benchmarks.

Finally, we propose a hybrid method named "smart selection" to select the best method for each customer based on the customer's data including the number of neighbors and average popularity of the products previously purchased by the customer. The smart selection method outperforms all other methods on average over all customers. In our test, smart selection uses the CF and PCB methods to predict customers' preferences for the majority of the customers. This shows that CF and PCB are complementary methods and PCB can successfully address the weak points of the CF method.

As a summary, PCB is a recommendation system designed specifically for online retailers. The method relies on the interactions in the online retailing environment such as clicks and purchases rather than parameters such as ratings and feedback which are generally used by other recommendation systems. Also, PCB has a high performance in addressing the cold start problem and recommending new products which is suitable for retailers who frequently change their product assortments. Note that many methods proposed by literature to solve the cold start problem require big data and a huge history of customer action, however, PCB performs well in addressing the cold start problem even with normal-size data of a retailer. Although our method has not been implemented as a recommendation system in an online retailer, the results indicate a potential for a successful performance in the case of implementation. Besides the implementation of the method as a recommendation system, we propose two following ideas to extend this research.

Product views without click or purchase

We hypothesize that if a product has been offered many times to a particular customer, but the customer has not clicked on the product or purchased it, this could be counted as a negative sign of interest. In other words, in contrast with clicks and purchasing which are positive feedback of interest, viewing without clicking or purchasing can be assumed to indicate negative feedback. It is important to account for this negative feedback and to differentiate between products that have been offered sufficiently many times and those that have been not offered. This would give a higher chance to the rarely offered products to be offered more. To test this hypothesis, we need a data set including the set of products viewed by each customer. Then, we can define $v_{jt}^{a_k(i)}$ as the number of times that a product with an attribute $a_k(i)$ is viewed by the customer j up to time t. Then, the score function would be updated as follows:

$$s_{jt}^{a(i)} = f(a(i), v_{jt}^{a_1(i)}, v_{jt}^{a_2(i)}, \dots, v_{jt}^{a_K(i)}, c_{jt}^{a_1(i)}, c_{jt}^{a_2(i)}, \dots, c_{jt}^{a_K(i)}, p_{jt}^{a_1(i)}, p_{jt}^{a_2(i)}, \dots, p_{jt}^{a_K(i)}) \quad (22)$$

Then, based on our hypothesis in the linear form of our method the coefficient for viewing subscores must be negative and in the random forest method viewing subscores must have a negative impact on the score.

Correlations between attributes

Note that in the proposed PCB method, we do not consider any correlation between different attribute types (e.g. brand and color). However, a customer's preference for an attribute may be dependent on the other attributes of the product. For instance, a customer may prefer white Nike shoes but prefer red Adidas ones, which means that his/her preference for color is dependent on the product's brand. To consider this effect we must evaluate the customer's interaction with each exclusive combination of attributes rather than each attribute separately. However, there are many possible combinations. Remember that each product is defined with K attributes and each attribute can get L_k different values (e.g. red, blue), then set of all possible attribute vectors, A, would have a size of $\prod_K L_k$. Each member of the set (attribute vector) $a(i)$, is an exclusive combination of attributes (e.g. $[Shoe, Red, Nike, Expensive, \dots]^T$). Some of these combinations may not correspond to any existing product. For example, maybe there is no red cheap Adidas product, however, the attribute vector can be assumed. The high number of possible vectors implies enormous interaction of customers and products to calculate the score for each

attribute vector. Although it seems inefficient to account for all correlations among attributes, an interesting extension to this method can be an investigation of the attributes that have the highest and most significant correlation with each other. This may cause a better prediction of customers' preferences and a higher performance in recommending products.

CRediT authorship contribution statement

Alireza KabirMamdouh: Writing – review & editing, Writing – original draft, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **A. Gürhan Kök:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Data curation, Conceptualization.

Data availability

The authors do not have permission to share data.

References

Animesh, A., Viswanathan, S., Agarwal, R., 2011. Competing "creatively" in sponsored search markets: The effect of rank, differentiation strategy, and competition on performance. Inf. Syst. Res. 22 (1), 153–169.

Aouad, A., Segev, D., 2021. Display optimization for vertically differentiated locations under multinomial logit preferences. Manage. Sci. 67 (6), 3519–3550.

Balabanović, M., Shoham, Y., 1997. Fab: Content-based, collaborative recommendation. Commun. ACM.

Bernstein, F., Kök, A.G., Xie, L., 2015. Dynamic assortment customization with limited inventories. Manuf. Serv. Oper. Manag. 17 (4), 538–553.

Burke, R., 2007. Hybrid web recommender systems. In: The Adaptive Web.

Campos, J., Figueiredo, A.D.d., 2001. Searching the unsearchable: Inducing serendipitous insights. In: Proceedings of the Workshop Program At the Fourth International Conference on Case-Based Reasoning. ICCBR.

Chakraborty, S., Hoque, M.S., Rahman Jeem, N., Biswas, M.C., Bardhan, D., Lobaton, E., 2021. Fashion recommendation systems, models and methods: A review. In: Informatics. vol. 8, MDPI, p. 49, 3.

De, U.C., Banerjee, S., Rath, M.K., Swain, T., Samant, T., 2022. Content based apparel recommendation for E-commerce stores. In: 2022 3rd International Conference for Emerging Technology. INCET, IEEE, pp. 1–6.

Deldjoo, Y., Nazary, F., Ramisa, A., Mcauley, J., Pellegrini, G., Bellogin, A., Noia, T.D., 2023. A review of modern fashion recommender systems. ACM Comput. Surv. 56 (4), 1–37.

Feldman, J., Zhang, D.J., Liu, X., Zhang, N., 2022. Customer choice models vs. machine learning: Finding optimal product displays on Alibaba. Oper. Res. 70 (1), 309–328.

Gallego, G., Li, A., Truong, V.-A., Wang, X., 2020. Approximation algorithms for product framing and pricing. Oper. Res. 68 (1), 134–160.

Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., Schmidt-Thieme, L., 2010. Learning attribute-to-feature mappings for cold-start recommendations. In: 2010 IEEE International Conference on Data Mining. IEEE, pp. 176–185.

Ge, M., Delgado-Battenfeld, C., Jannach, D., 2010. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In: Proceedings of the Fourth ACM Conference on Recommender Systems. pp. 257–260.

Gharaei, N.Y., Dadkhah, C., Daryoush, L., 2021. Content-based clothing recommender system using deep neural network. In: 2021 26th International Computer Conference, Computer Society of Iran. CSICC, IEEE, pp. 1–6.

Ghose, A., Ipeirotis, P.G., Li, B., 2012. Designing ranking systems for hotels on travel search engines by mining user-generated and crowdsourced content. Mark. Sci. 31 (3), 493–520.

Ghose, A., Ipeirotis, P.G., Li, B., 2014. Examining the impact of ranking on consumer behavior and search engine revenue. Manage. Sci. 60 (7), 1632–1654.

Goldberg, D., Nichols, D., Oki, B.M., Terry, D., 1992. Using collaborative filtering to weave an information tapestry. Commun. ACM.

Golrezaei, N., Nazerzadeh, H., Rusmevichientong, P., 2014. Real-time optimization of personalized assortments. Manage. Sci..

Hill, W., Stead, L., Rosenstein, M., Furnas, G., 1995. Recommending and evaluating choices in a virtual community of use. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Hwangbo, H., Kim, Y.S., Cha, K.J., 2018. Recommendation system development for fashion retail e-commerce. Electron. Commer. Res. Appl. 28, 94–101.

Iaquinta, L., De Gemmis, M., Lops, P., Semeraro, G., Filannino, M., Molino, P., 2008. Introducing serendipity in a content-based recommender system. In: 2008 Eighth International Conference on Hybrid Intelligent Systems. IEEE, pp. 168–173.

Khalid, M., Keming, M., Hussain, T., 2021. Design and implementation of clothing fashion style recommendation system using deep learning. Rom. J. Inform. Technol. Autom. Control/Rev. Rom. Inform. Autom. 31 (4).

Kök, A.G., Fisher, M.L., Vaidyanathan, R., 2015. Assortment planning: Review of literature and industry practice. In: Retail Supply Chain Management. Springer, pp. 175–236.

Konstan, J.A., Miller, B.N., David Maltz, J.L., Herlocker, L.R.G., Riedl, J., 1997. GroupLens: Applying collaborative filtering to usenet news. Commun. ACM.

Lei, Y., Jasin, S., Uichanco, J., Vakhutinsky, A., 2021. Joint product framing (display, ranking, pricing) and order fulfillment under the multinomial logit model for E-commerce retailers. Manuf. Serv. Oper. Manag..

Leininger, L., Gipson, J., Patterson, K., Blanchard, B., 2020. Advancing performance of retail recommendation systems. SMU Data Sci. Rev. 3 (1), 6.

Lu, Z., Hu, Y., Yu, C., Jiang, Y., Chen, Y., Zeng, B., 2022. Personalized fashion recommendation with discrete content-based tensor factorization. IEEE Trans. Multimed. 25, 5053–5064.

Masthoff, J., 2015. Group recommender systems: Combining individual models. In: Recommender System Handbook.

McNee, S.M., Riedl, J., Konstan, J.A., 2006. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In: CHI'06 Extended Abstracts on Human Factors in Computing Systems. pp. 1097–1101.

Mladenic, D., 1999. Text-learning and related intelligent agents: A survey. IEEE Intell. Syst. Appl..

Ning, X., Desrosiers, C., Karypis, G., 2015. A comprehensive survey of neighborhood-based recommendation methods. In: Recommender Systems Handbook. Springer, pp. 37–76.

Park, D.H., Kim, H.K., Choi, I.Y., Kim, J.K., 2012. A literature review and classification of recommender systems research. Expert Syst. Appl. 39 (11), 10059–10072.

Ricci, F., Rokach, L., Shapira, B., 2015. Recommender systems: Introduction and challenges. In: Recommender Systems Handbook. Springer, pp. 1–34.

Saveski, M., Mantrach, A., 2014. Item cold-start recommendations: learning local collective embeddings. In: Proceedings of the 8th ACM Conference on Recommender Systems. pp. 89–96.

Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S., 2007. Collaborative filtering recommender systems. In: The Adaptive Web. Springer, pp. 291–324.

Sha, D., Wang, D., Zhou, X., Feng, S., Zhang, Y., Yu, G., 2016. An approach for clothing recommendation based on multiple image attributes. In: Web-Age Information Management: 17th International Conference, WAIM 2016, Nanchang, China, June 3-5, 2016, Proceedings, Part I 17. Springer, pp. 272–285.

Shani, G., Gunawardana, A., 2011. Evaluating recommendation systems. In: Recommender Systems Handbook. Springer, pp. 257–297.

Su, X., Khoshgoftaar, T.M., 2009. A survey of collaborative filtering techniques. Adv. Artif. Intell. 2009.

Suvarna, B., Balakrishna, S., 2024. Enhanced content-based fashion recommendation system through deep ensemble classifier with transfer learning. Fash. Text. 11 (1), 24.

Toms, E.G., et al., 2000. Serendipitous information retrieval. In: DELOS. Citeseer.

Tuinhof, H., Pirker, C., Haltmeier, M., 2019. Image-based fashion product recommendation with deep learning. In: Machine Learning, Optimization, and Data Science: 4th International Conference, LOD 2018, Volterra, Italy, September 13-16, 2018, Revised Selected Papers 4. Springer, pp. 472–481.

Ursu, R.M., 2018. The power of rankings: Quantifying the effect of rankings on online consumer search and purchase decisions. Mark. Sci. 37 (4), 530–552.

Van Meteren, R., Van Someren, M., 2000. Using content-based filtering for recommendation. In: Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop. vol. 30, pp. 47–56.

Vartak, M., Thiagarajan, A., Miranda, C., Bratman, J., Larochelle, H., 2017. A meta-learning perspective on cold-start recommendations for items. Adv. Neural Inform. Process. Syst. 30.

Volkovs, M., Yu, G., Poutanen, T., 2017. Dropoutnet: Addressing cold start in recommender systems. Adv. Neural Inform. Process. Syst. 30.

Wei, J., He, J., Chen, K., Zhou, Y., Tang, Z., 2017. Collaborative filtering and deep learning based recommendation system for cold start items. Expert Syst. Appl. 69, 29–39.

Yeruva, S., Sathvika, A., Sruthi, D., Reddy, D.Y., Krishna, G., 2022. Apparel recommendation system using content-based filtering. Int. J. Recent Technol. Eng. (IJRTE) 11, 46–51.

Ziarani, R.J., Ravanmehr, R., 2021. Serendipity in recommender systems: A systematic literature review. J. Comput. Sci. Tech. 36, 375–396.