Knowledge-Based Systems 111 (2016) 193–206

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Recommender systems for product bundling☆

Moran Beladev*, Lior Rokach, Bracha Shapira

Information Systems Engineering, Ben-Gurion University of the Negev, Israel

ARTICLE INFO

Article history:
Received 14 February 2016
Revised 3 August 2016
Accepted 10 August 2016
Available online 13 August 2016

Keywords:
Recommender systems
Product bundling
Price bundling
E-commerce
Collaborative filtering
SVD

ABSTRACT

Recommender systems (RS) are a class of information filter applications whose main goal is to provide personalized recommendations, content, and services to users. Recommendation services may support a firm's marketing strategy and contribute to increase revenues. Most RS methods were designed to provide recommendations of single items. Generating bundle recommendations, i.e., recommendations of two or more items together, can satisfy consumer needs, while at the same time increase customers' buying scope and the firm's income. Thus, finding and recommending an optimal and personal bundle becomes very important. Recommendation of bundles of products should also involve personalized pricing to predict which price should be offered to a user in order for the bundle to maximize purchase probability. However, most recommendation methods do not involve such personal price adjustment.

This paper introduces a novel model of bundle recommendations that integrates collaborative filtering (CF) techniques, demand functions, and price modeling. This model maximizes the expected revenue of a recommendation list by finding pairs of products and pricing them in a way that maximizes both the probability of its purchase by the user and the revenue received by selling the bundle.

Experiments with several real-world datasets have been conducted in order to evaluate the accuracy of the bundling model predictions. This paper compares the proposed method with several state-of-the-art methods (collaborative filtering and SVD). It has been found that using bundle recommendation can improve the accuracy of results. Furthermore, the suggested price recommendation model provides a good estimate of the actual price paid by the user and at the same time can increase the firm's income.

1. Introduction

Recommender systems are a class of information filter applications whose main goal is to provide personalized recommendations of content and services to users. A recommender system for an e-commerce site helps users find products, such as movies, songs, books, gadgets, applications, products, and restaurants that fit their personal preferences and needs [1]. Recommender systems enhance e-commerce sales by converting browsers into buyers, exposing customers to new products, increasing cross-selling by suggesting additional products, building customer loyalty, increasing customers' satisfaction based on their purchasing experience, and increasing the likelihood of repeat visits by satisfied customers. Each of these can be translated into increased sales and higher revenue. In the age of e-commerce, it is important for firms to develop web-based marketing strategies such as product bundling to increase revenue. Product bundling refers to the practice of selling two or more goods together, packaged at a price which is below the sum of the independent prices [6]. This practice can be observed very often in the real world. For example, if a customer buys Internet access and cell phone service together from the same company, it is often sold as a package which is cheaper than buying both services independently. Generating bundles is an example of a marketing strategy aimed at satisfying consumer needs and preferences, while complementing the firm's marketing strategy on two levels, by increasing income and widening the customers' buying scope. The motivation behind using recommender systems as a platform to bundle recommendations is to expand the market to encompass new products that would not have been purchased were they not part of a bundle, and by doing so, increasing the firms' income and profits. The bundling effect can leverage and expand upon the single item recommendation by recommending top N list recommendations that include both items and bundles, providing the customer the ability to choose whether to buy a bundle or a single item.

The collaborative filtering (CF) approach is considered one of the most popular and effective techniques for building recommender systems [2]. The basic idea is to try to predict the user's opinion about different items and recommend the "best" items,

☆ Initial version of this paper was presented as a poster at the RecSys conference in 2015 [21].

* Corresponding author.
E-mail address: belachde@post.bgu.ac.il (M. Beladev).

http://dx.doi.org/10.1016/j.knosys.2016.08.013
0950-7051/

using the user's previous preferences and the opinions of other likeminded users.

CF is a very effective recommendation technique, and bundling is one of the most useful marketing strategies; therefore we suggest combining them.

The design of a RS with product bundling is more challenging than that of a RS based on single item recommendation. Whereas in a routine RS the problem is to find products that the user will like, in a RS with product bundling, we also have to deal with the associations between the products within the bundle. Moreover, the advantage for the consumer when purchasing the bundle is the associated cost savings; this results in an additional challenge – pricing the bundle in such a way that will satisfy customers and entice and convince them to buy it, while also serving the supplier's interests. Most recommendation methods are designed to provide single item recommendations and do not involve personalized price recommendation. Very few studies have been conducted in the area of combining bundling strategy with recommender systems. Previous researches did not included concrete evaluation showing that bundle purchasing can be predicted and didn't use recommender system measures such as precision and recall. Furthermore, there was no work that involved price bundling in the recommendations.

The main contribution of this research is that we introduced a novel model that integrates bundle recommendation algorithm with the recommendation systems platform. We examined the possibility of combining bundle recommendations without diminishing the prediction accuracy of state-of-the-art collaborative filtering and SVD methods (, which will be described later).

In order to recommend bundles we determined the probability that the customer would buy the bundle. The bundle purchasing probability was based on a new adjusted model which uses collaborative filtering techniques, a personalized demand graph, pricing modeling, and optimization techniques. Another contribution is that we implemented an optimization technique to determine which personal price should be offered to a user for a particular bundle. As we demonstrated, this model can increase the users' buying scope and the firm's income, and for that reason this model can help the RS industry. We evaluated our model by using three real datasets and used offline tests to evaluate the hit of our bundling recommendations and price suggestions. The results showed that in comparison to state-of-the-art item recommendation methods, our bundling recommendations can improve the precision, recall, average quantity of products purchased, and the average price paid.

The techniques and methods that we used will be described more thoroughly in the following chapters. In chapter 2 we review related work in the fields of product and price bundling. In chapter 3 we present detailed research objectives, our suggested model, and algorithm. Chapter 4 describes our experiments and results. In chapter 5 we present the conclusions, discussion and future work.

2. Related work

Recommender systems (RS) are a type of information filtering system which aims to predict the 'rating' or 'preference' a user would give an item.

This information can be obtained directly, usually based on the users' ratings for items, or indirectly, by monitoring users' behavior, such as songs heard, applications downloaded, websites visited, products purchased, and books read [1]. For the past decade, recommender systems have been investigated both by industry and academia.

The most widely used filtering algorithms presented in the literature for the recommendation task are: collaborative filtering, demographic filtering, content-based filtering, and hybrid filtering [1].

Content-based filtering makes recommendations based on user choices made in the past (e.g., in a web-based e-commerce RS, if the user has purchased comedy films in the past, the RS will likely recommend a newly released comedy that the user has not yet purchased on this website). Content-based filtering also generates recommendations using the content from objects intended for recommendation; therefore, specific content can be analyzed such as text, images, and sound.

Demographic filtering is justified on the principle that individuals with certain common personal attributes (sex, age, country, etc.) will also have common preferences. [25] presents novel approaches of user profiling for demographic recommender systems. These approaches represent alternatives for profiling users using attribute types and representations, in order to obtain a strong indication of the closeness between individuals.

Collaborative Filtering allows users to provide ratings about a set of elements in such a way that when enough information is stored on the system, recommendations can be made to each user based on information provided by other users that are thought to have the most in common with them.

The most widely used algorithm for collaborative filtering is the k-Nearest Neighbors (k-NN) which will be used in our research. In the user to user version, k-NN executes the following three tasks to generate recommendations for an active user: (1) determine k users neighbors (neighborhood) for the active user a; (2) implement an aggregation approach with the ratings for the neighborhood in items not rated by a; and (3) extract the predictions identified in step 2, and select the top N recommendations. [26] suggests a novel technique for predicting the tastes of users with an understandable probabilistic meaning based on collaborative filtering. The paper presents a new decomposition of the rating matrix which is based on factorizing the rating matrix into nonnegative matrices whose components are within the range [0,1].

Hybrid filtering uses a combination of CF with demographic filtering or CF with content-based filtering. Hybrid filtering is usually based on bioinspired or probabilistic methods such as genetic algorithms and fuzzy genetic, neural networks, Bayesian networks, clustering, and latent features (such as SVD [23]). Clustering-based recommender systems suffer from relatively low accuracy and coverage. [27] presents a new multiview clustering method to address these issues. The method iteratively clusters users from the perspectives of both rating patterns and social trust relationships. This approach demonstrates that clustering-based recommender systems are suitable for practical use.

There are two approaches in the literature for the bundling task: product bundling and price bundling [7].

- Product bundling is a design oriented approach, which helps identify which products among a feasible set of "products" should go into the bundle.
- Price bundling is a pricing oriented approach, which assumes a product portfolio and proposes the prices at which the individual items and/or bundles should be offered.

The distinction between price and product bundling is important, because each of these types of bundling entails different strategic choices and therefore different consequences for companies. Product bundling deals with the problem of choosing which products will be combined as a bundle. Price bundling deals with the problem of which price should be offered for a set of different products. Whereas price bundling is a pricing and promotional tool, product bundling is more strategic, because it creates added value. Managers can use price bundling easily, at short notice or for a limited duration, whereas product bundling is a longer-term strategy. We presented here the related work regarding this two approaches:

2.1. Product bundling

There are methods such as frequent itemsets and association rules that aim to find relationships between sets of products. Frequent itemsets play an essential role in many data mining tasks that try to identify interesting patterns within databases, such as association rules, correlations, sequences, episodes, classifiers, clusters, and many more. The frequent itemset approach tries to find sets of items that appear in many of the same baskets. Each basket consists of a set of items (an itemset) and intuitively, a set of items that appears in many baskets is said to be “frequent” [3].

The mining of association rules is one of the most popular data mining approaches. The original motivation for finding association rules came from the need to analyze supermarket transaction data by examining customer behavior based on the purchased products. Association rules describe how often items are purchased together. For example, the association rule “milk ⇒ bread (80%)” indicates that four out of five customers that bought milk also bought bread. Such rules can be useful for decisions concerning product pricing, promotions, store layout, and more.

Since their introduction in 1993 by Argawal et al. [4], the frequent itemset and association rule mining techniques have received a great deal of attention. Interest continues to this day, and within the past two decades, hundreds of research papers have been published presenting new algorithms or improvements on existing algorithms to address these mining issues more efficiently [5].

Our model deals with personal bundling recommendations as opposed to the abovementioned frequent itemset and association rules mining techniques, which are not personalized. Instead, our model relies upon the CF (collaborative filtering) method which is the most common approach for the recommendation task.

Very few studies have been conducted in the area of combining bundling strategy with recommender systems. In this section we present these studies.

The first study describes a product bundling approach in the tourism domain which presents a case model to represent a travel plan bundle along with user profiles and preferences [8]. This recommender system supports the bundling of a personalized travel plan by selecting travel products (e.g., a hotel, museum visit, or a climbing activity) and building a travel bag, which effectively represents the bundling of products. This framework also consisted of building an interactive human-machine system in which users responded to a query and the system returned suggested travel plans. When a “failure” situation occurred and no results were returned, the system suggests a new set of alternative queries that might produce satisfactory results by tightening or relaxing some of the query constraints. The model used in this paper to support personalized travel plan is a case model. The Case Base (space) is made of four components: travel wishes and constraints, travel bag (which is the bundle), user features, and reward (the rank for the travel bundle). A case is built during a human/machine interaction, and therefore it is always a structured snapshot of the interaction at a specific time.

The paper didn’t show any empirical results evaluating this system’s performance.

The second study demonstrates the feasibility of bundle recommendations using the CF recommendation method [9]. Clustering was used to find customer groups, and data mining with association rule techniques was used to find the relations between two sets of products within a transaction database. The products were classified into three categories: hot sale, general sale, and dull, and the evaluation was the hit of the bundle strategy as opposed to a specific bundle.

The third study discusses bundle optimization using a generic algorithm to maximize the compatibility of the products within the bundle (using association rules), and simultaneously satisfies both customer preferences and merchant requirements [10]. The genetic model defined chromosome structure in a way that every gene represents a particular product belonging to the merchant’s catalogue. Each gene represents a product whose reference reserves 64 bits. The fitness function is a combination of the retailer profit and the confidence of the related associative rule. They rely upon the fitness value for the evaluation, not recommender system platform measures. In our research we intend to evaluate the bundles recommendation with measures from the recommender systems field (such as precision and recall) that are based on the user behavior, i.e. whether he/she bought the items. The previous study provides the fitness value, but does not indicate whether the customer actually bought this bundle.

The last paper introduces a bundle recommendation problem (BRP) that can build a bundling effect during recommendation. Its solution is based on a set of items that maximizes some total expected reward [15]. The bundle recommendation is designed as a problem of selecting a set of k items from a given list of relevant items. More precisely, given a user u, the developed model recommends k items from a catalog containing n items whose relevancies with respect to the user u are known. For the evaluation they used offline test and online campaigns using the email. Both the offline and online results showed that the bundle recommendation algorithm can consistently improve the baseline models in terms of predefined rewards like conversions or revenues. The paper provides the whole recommendation list as a bundle and doesn’t leverage the price advantage of bundles. There is a need to take into account the price aspect and integrate dynamic pricing techniques with bundle recommendation.

Our work suggests a new approach of recommending bundles as a top N recommendations list. For the bundling task, we don’t use methods that request the user to provide information. Furthermore, we not only deal with representation of a bundle, but also show real results and the effectiveness of the bundling approach. We evaluated the recommendations in a more specific and accurate way according to a specific bundle, in contrast to some previous studies that evaluated the hit of the bundle strategy. We intend to evaluate the bundles recommendation with measures from the recommender systems field that are based on the user behavior, i.e. whether he/she bought the items. Most of the studies mentioned earlier in this section didn’t use recommender system evaluation methods to evaluate the accuracy of the bundle recommendations. In addition, each recommendation includes a personal price suggestion. The mentioned studies didn’t propose personal price recommendations for the bundles. One study has been conducted in the area of personal price recommendation. This paper [20] implements a personalized promotion as marketing tactic for increasing sales volume. The study estimated consumer’s WTP (willingness to pay) using laboratory auctions. They used Becker-DeGroot-Marschak (BDM) mechanism. Under BDM, each bidder submits a bid to purchase a product. A sales price is randomly drawn from an interval which covers all plausible bids. If that sales price is lower than a participant’s bid, then she receives the product and pays the sales price. They ran an experiment where subjects selected at least 5 best value products (from 120k skin care products in amazon.com), ranked the selected products, the system recommended the customer a list of products using Amazon’s “consumer who bought this also bought these” recommendations. Next, the customer bid at least 5 products knowing that at most one of the products will be randomly selected to go through the BDM procedure. Finally, the system ran the lottery on all subjects.

Based on the data collected they build regression models for personalized promotion. They test their model using RMSE and the seller profit metrics. They noted that it may be useful to consider bundles of goods and that personalized promotion and recommendation should be considered jointly within a unified framework, and not remain as separate problems like they did in this study.

2.2. Price bundling

In this section we describe the work done in the pricing field. The use of mixed price bundling has increased over the last few years. The effectiveness of price bundling appears to be a function of the demand in such a way as to achieve cost economies. In the mixed-joint form, a single price P_{A+B} is established for the two products purchased jointly (where $P_{A+B} < P_A + P_B$). Each product has a distribution of reservation prices (the maximum amounts buyers are expected to pay). By bundling the products together, we essentially create a new product. If the two products are independent in their demand, some customers who would buy only one of the products in the previous situation in which they were priced individually will now buy both products. The reason is that the value these customers place on one product is higher than the price of the combined two products. In economic terminology, the consumer surplus is the amount by which the individual's reservation price exceeds the actual price paid [12] presents four basic kinds of customers (segments) based on their relation to the bundle, each of which is characterized by a different set of reservation price distributions. Each of these segments needs a different pricing method. Further, [12] shows how these objectives can be achieved and examines pricing models.

The primary objective of bundling is cross-selling by appealing to customers who might purchase A or B but not both. In examining cross-selling opportunities, an important consideration is the relative unbundled demand levels for A and B. If the quantity sold of A (only) with unbundled sales is substantially greater than the quantity sold of B (only) with unbundled sales, then the strategy tends to be mixed-leader bundling, where A will be the best leader, and a reduced price for A is tied to the purchase of B. In cases in which these quantities are equal, mixed joint bundling is the appropriate strategy. For example, a sport club may have a number of customers who only purchase aerobic classes and an equal number of customers who only purchase use of the weight room. A bundling scheme that provides a discount on the total cost of the two services (i.e., mixed joint bundling) would maximize the number of cross-selling opportunities. The key to effective demand response in bundling is identifying the complementary factors among services. The complementary issue has received attention in marketing and is discussed further in this work. Complementary factors are desirable for cross-selling, because they enhance the likelihood that the current customer surplus, or any surplus created by a price reduction on the leader, is transferred to the second service. Furthermore, if B enhances the utility (reservation price) of A, the probability of buying the bundle increases. [43] presents the demand conditions required for successful cross-selling and mixed leader bundling purchasing. Additionally, [43] provides the demand conditions for customer acquisition, which focus on potential new customers who buy neither A nor B. [13] assumed that the distributions of reservation prices for each product separately and for the bundle are all normal and compares pure bundling and unbundled sales under bivariate normality. This paper shows that pure bundling is always more profitable than unbundled sales in symmetric Gaussian cases and states the conditions when mixed bundling is more profitable than the other strategies under the above assumptions. [14] suggests a mixed integer linear program in order to optimize bundle pricing. The objective function represents the firm's profit and the customers' behavior, and their surplus is treated as a constraint on the firm's objective function. To find the probability that the customer will buy a bundle at a specific price, we should look at the demand function. [16] presents the demand for cigarettes as a function of price, income, and other factors that influence taste. Both the decision to smoke (the participation decision) and the quantity smoked are of interest. The most basic result in economics is that higher prices discourage consumption of a product or service. The paper [16] shows that price has a significant negative influence on both smoking participation and on the number of cigarettes consumed by smokers, with elasticities ranging between -0.4 and -0.6. Increasing cigarette taxes will decrease both smoking participation and consumption. However, the effect of price varies widely by income group, with the greatest effect shown for individuals with low incomes and women. The price elasticity of participation for low income women is nearly -1.0, which is almost three times the elasticity for the pooled sample of women and double the magnitude for men overall. That shows that the price affects personal consumption and each user has his/her own demand function based on gender, income, and preferences. We use the results of this paper as a motivating idea for the price bundling task by finding the personal demand function of the user, which includes his price sensitivity.

3. Product and pricing bundling recommendations (PBR) model

Pervious works have shown that product bundling is feasible for recommender systems and can be very effective at increasing firms' incomes and sales. However, most previous research did not included concrete evaluation showing that bundle purchasing can be predicted and didn't use recommender system measures such as precision and recall. Furthermore, there was no work that involved price bundling in the recommendations. There was one study that deals with personalized price for items using auctions. The main goal of our research is to provide and design a detailed method that combines various techniques for the bundling task based on the recommendation systems platform, and accordingly our research will be evaluated using measures associated with this field. We designed a new personalized bundling approach for recommender systems that will be able to predict which products will be suggested as a bundle that interests the user. We verified that the algorithm provides good evaluation measures and can be used for the bundle prediction task (using recall and precision measures). We designed a new personalized pricing bundling approach that will be able to predict which price should be offered to a user for the bundle. We verified if price recommendation provides a good estimate of the actual price that the customer paid, and if price recommendation can increase the user's purchasing price and the firm's revenue.

3.1. Approach

In this section the proposed approach is presented. The general method aims at improving the seller's expected revenue, while at the same time maintaining the prediction accuracy of the CF recommendation system.

The main research method that we use is composed of the following steps:

Fig. 1 shows the process of recommending top N bundles. In order to find which top N bundles should be recommended, we use the bundling strategy of maximizing the expected revenue value of each bundle. The expected revenue function consists of the profit of the seller from selling the bundle and the probability the customer will buy the bundle at the recommended price *T. To* deter-

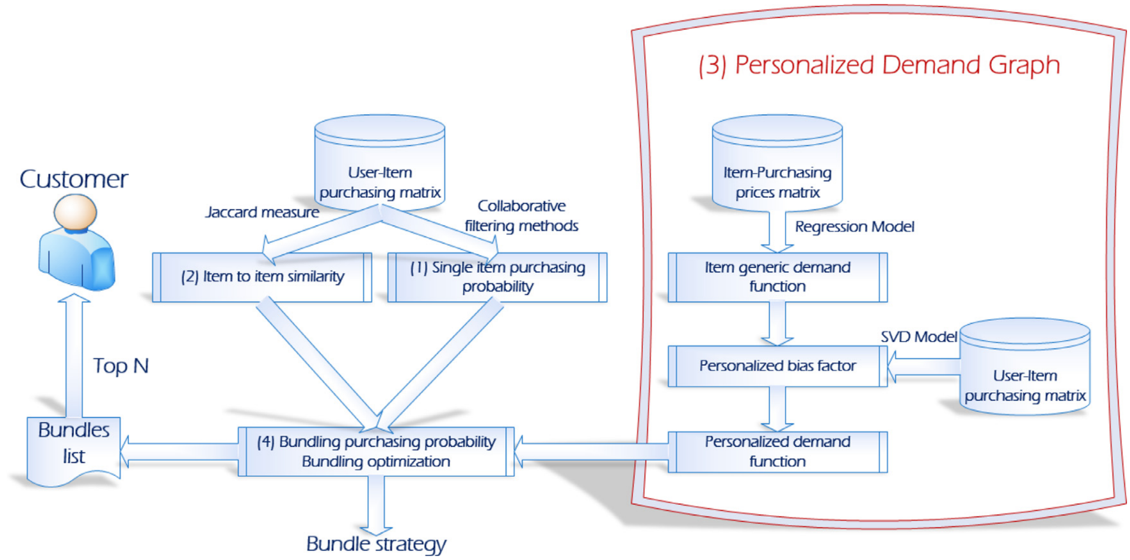

Fig. 1. The research model, methods, and techniques.

mine that probability we designed a model which consists of three parts (as indicated by the numbers (1)–(3) in Fig. 1):

1. Single item purchasing probability (using the k-NN CF method).
2. Item to item similarity using the Jaccard measure to find the dependencies of the items within the bundle.
3. Pricing element using personalized demand graph which is built by finding a generic demand graph and a personalized bias factor which represents the user's sensitivity to the price. This bias factor is predicted for each user and item using the SVD method. With this personalized demand function we can determine the probability of purchasing each product at a specific price.

The main function (indicated by the number 4 in Fig. 1) is modeled as an optimization problem aiming at identifying the bundles and prices that maximize the bundling strategy (maximizing the expected revenue or the purchasing probability) and therefore should be suggested to the user. In order to build the demand graph model we used the *user-item purchasing matrix* (which contains the following transaction information: item ID, customer ID, and purchase price) and *item-purchasing prices matrix* (which contains the following transaction information: item ID and purchase price). The model was evaluated and compared using common accuracy and performance measures on different datasets.

We first introduce our optimization problem and then the techniques and methods used to solve it.

3.2. The bundling optimization problem

In this section, we present our bundle recommendation model.

The retailer expected revenue function that we want to maximize is:

$$ExpectedRevenue = P_u(i_1, i_2, T) \cdot (T - cost_1 - cost_2) \tag{1}$$

where $P_u(i_1, i_2, T)$ is the probability that user u will purchase the bundle, which is composed of products i_1 and i_2, at price T. An assumption that differentiation in the bundle's price between customers is legitimate was made, in order to provide the user a personal recommendation of a bundle with a personal price T. $cost_1$ is the retailer's cost of product i_1, and $cost_2$ is the retailer's cost of product i_2. The proposed bundle and the price T for user u will be the one that maximizes the expected revenue:

$$(i_1, i_2, T) = argmax_{\forall i_1, i_2, T} ExpectedRevenue(i_1, i_2, T) \tag{2}$$

In order to find $P_u(i_1, i_2, T)$, we have to find the corresponding prices (C_1 of product i_1 and C_2 of product i_2) that add up to the bundle price T:

$$P_u(i_1, i_2, T) = max_{\forall c1, c2 | c1+c2=T}\ P_u(i_1 \cap i_2 \cap C_1 \cap C_2) \tag{3}$$

meaning that we have to find the prices C_1 and C_2 that maximize the probability of the user u to buy products i_1 and i_2 at those prices.

According to Bayes' law:

$$P_u(like\ i_1 \cap willing\ to\ pay\ C_1\ for\ i_1) = P_u(like\ i_1) \cdot P_u(willing\ to\ pay\ C_1\ for\ i_1 | like\ i_1) \tag{4}$$

We used the Jaccard measure to represent the products' similarity. The Jaccard coefficient measures similarity between finite sample sets and is defined as the probability of the intersection of the sample sets divided by the probability of the union of the sample sets (ranges 0 to 1) [11]. In our model we used the Jaccard similarity coefficient, since it is a measure commonly used in the shopping domain that indicates whether two products were frequently bought together [18]. Furthermore, our data model uses binary data and Jaccard is a useful measure for this kind of data. Products with a high Jaccard measure can be combined as a bundle.

According to the Jaccard measure:

$$Jaccard = J_{i_1, i_2} = \frac{P(i_1 \cap i_2)}{P(i_1 \cup i_2)} \tag{5}$$

using combinatorial mathematics, the inclusion–exclusion principle:

$$P(i_1 \cap i_2) = P(i_1) + P(i_2) - P(i_1 \cup i_2) \tag{6}$$

using Eqs. (5)+(6):

$$P(i_1 \cap i_2) = \frac{P(i_1) + P(i_2)}{1 + \frac{1}{J_{i_1, i_2}}} \tag{7}$$

using Bayes' law and Eqs. (4) and (7):

$$P_u((i_1 \cap C_1) \cap (i_2 \cap C_2)) = \frac{P_u(i_1) \cdot P_u(C_1 | i_1) + P_u(i_2) \cdot P_u(C_2 | i_2)}{1 + \frac{1}{J_{i_1, i_2}}} \tag{8}$$

The simplified assumption is that the Jaccard measure, J_{i_1, i_2}, which denotes the products' compatibility, is not affected by the price and the target user. We can't define the dependence between

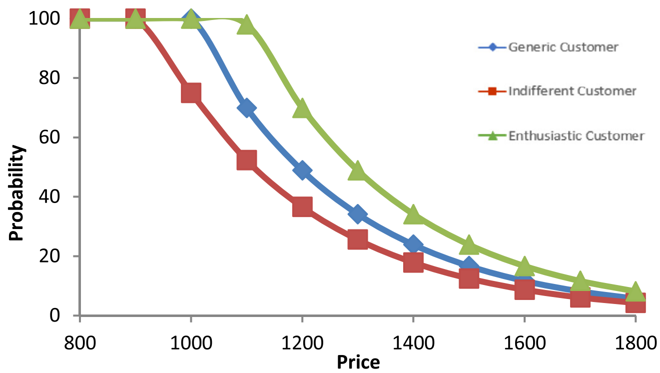

Fig. 2. A demonstration of personalized demand function for a generic customer, indifferent customer, and an enthusiastic customer.

the two products using only the target user's data. Thus, this measure is calculated in a collaborative form, using all the users that bought the both products.

The $P_u(i_1)$, $P_u(i_2)$ probabilities can be found using the collaborative filtering technique:

$$P_u(X) = \frac{\sum_{u_i \neq u_j} W_{u_i,u_j} \cdot r_{u_j}}{\sum_{u_i \neq u_j} W_{u_i,u_j}} \tag{9}$$

where W_{u_i,u_j} is the correlation between customer u_i to customer u_j. r_{u_j} is 0 if customer u_j didn't buy item x, and 1 if customer u_j bought item x. The correlation is calculated with the Jaccard measure.

To find $P_u(C_1|i_1)$, $P_u(C_2|i_2)$ we use our personal demand function presented in the next section.

3.3. Personalized demand graph

We enhance the above model by relaxing the assumption that one demand-graph suits all customers.

Namely we would like to assume that each customer has his/her own demand graph based on their preferences or business attitude. We would like to develop a heuristic for estimating the "personalized" demand graph for user u and item i using very sparse data, as is usually available in RSs.

The graph presented in Fig. 1 shows the generic customer versus an enthusiastic customer (triangle) and an indifferent one (square).

We assume that switching from one curve (e.g., the square curve) to another (e.g., the triangle curve) is done as follows:

$$P_u(C_1|i_1) = min(P_{\cdot,i_1}(C_1) \cdot \alpha_{u,i_1}\ ,\ 100\%) \tag{10}$$

where $P_{\cdot,i_1}(C_1)$ is the generic demand graph for item i_1 given price C_1, and α_{u,i_1} is the personalized bias factor for user u and item i_1.

In order to find the personal bias factor, α_{u,i_1}, we can look at the individual customer's previous purchases or the highest bid a customer assigned to an item. We compare the customer's price to the median price obtained from the generic graph. For example, if customer u purchased item i_1 at price C_1* then his/her bias factor is estimated as:

$$\alpha_{u,i_1} = \frac{0.5}{P_{\cdot,i_1}(C_1^*)} \tag{11}$$

We demonstrate this idea in the graph presented in Fig. 2. The current customer purchased the item for price $C_1* = 1300$, which, according to the generic graph, convinced only 35% of the interested population. Thus, in order to determine the personalized bias factor for this user:

$$\alpha_{u,i_1} = \frac{0.5}{0.35} = 1.42$$

We can create a bias matrix for all purchases of items by users, and the remaining values of bias factor of products that have not been purchased by the customer can be predicted using the SVD method. Given the complete matrix, we can generate the personalized demand graph of user u and item i_1 by taking the generic demand graph calculated for item i_1 and multiplying it by the predicted α .

3.4. Algorithm

In this subsection, we summarize the bundle recommendation algorithm.

The input of the algorithm is a sparse binary matrix of the users and items, where a value of 1 represents that user u purchased item i. Another input is the number of purchases at each price for specific items.

The output is a top K recommended bundles list for each user. Algorithm 1 has three parts and is described below.

1. Building the generic demand graph for each item (lines 10–11) - we build a regression model using the number of purchases in each price of the given item.
2. For each user we compute the similarity to all other users W_{u_i,u_j} (lines 13–14), using the Jaccard in order to find user to user similarity. We used the Jaccard correlation, because it is considered a good similarity function for binary sets, and it has been used in several RS studies to obtain similarity scores [22].
3. For each user we compute the item purchase probability, and the personalized bias factor for each item (lines 15–17). The bias factor can be predicted if the data exists, and if not, it can be predicted using SVD.
4. Bundling optimization (lines 18–33) - in this part the joined probability of purchasing two items is calculated. We are looking for the bundle price T that maximizes the expected revenue of selling this bundle.
   T=Min $[min(C_1),\ min(C_2)]$;T<$[max(C_1)+max(C_2)]$ (line 20) meaning that T can be at least the minimum of the prices of item 1 and item 2 that exist in the dataset, and at most, the addition of the maximum prices of item 1 and item 2 that exist in the dataset. C_1, C_2 are the prices of the items that add up to T, and we want to find the best assignment that maximizes the joined probability. Prob is the probability the user will purchase the two items at the given prices, meaning Prob=$P_u((i_1 \cap C_1) \cap (i_2 \cap C_2))$. For each bundle we save the maximum expected revenue value that can be achieved (lines 28–31).
5. Then we sort all the potential bundles by the expected revenue value and suggest the top K (lines 32–33) to the user.

We use two strategies in our model:

I. The first strategy maximizes the probability that the user will buy the bundle, meaning that we maximize $(i_1, i_2, T) = argmax_{\forall i_1, i_2, T}\ P_u(i_1, i_2, T)$, where $P_u(i_1, i_2, T) = max_{\forall c1,c2|c1+c2=T}\ P_u(i_1 \cap i_2 \cap C_1 \cap C_2)$.
II. The second maximizes the expected revenue $(i_1, i_2, T) = argmax_{\forall i_1, i_2, T} ExpectedRevenue(i_1, i_2, T)$, where $ExpectedRevenue = P_u(i_1, i_2, T) \cdot (T - cost_1 - cost_2)$.

The algorithm computational complexity consists of:

1. Single item purchasing probability (CF user to user similarity algorithm) - if user similarities are not pre-computed offline, they need to be computed at the time a recommendation is requested. In this case, there is no need for computing the whole user similarities matrix – only similarities between the active user and all of the other users or a set of training users. The cost of this computation is of the order O(mn), where m is the

Algorithm 1: Bundle Recommendation Algorithm

Data: Binary matrix of purchases of each user *u* and each item *i*. Data tuple of [item, number of purchases, price].

Construct all possible pairs of items to be potential bundle B to be recommended.

MaxP=0; //initialization

ExpectedRevenue=0; //initialization

MaxExpectedRevenue=0; //initialization

BestProposal=0; //initialization

List *bestBundles*; // list of the top bundles to be recommended

K // length of the recommendation list

Cost // The retailer's cost is x% of the price of the item. Can be modified.

Foreach item $i = i_1, i_2, \dots, i_n$ **do**

Construct a regression function to represent the item's probability *P* and price *C* correlation $P_{\cdot,i}(C)$.

Foreach user u_i **do**

Foreach user $u_j > u_i$ **do**

Compute W_{u_i,u_j} // user to user similarity

Foreach item $i = i_1, i_2, \dots, i_n$ **do**

Compute $P_{u_i}(i) = \frac{\sum_{u_i \neq u_j} W_{u_i,u_j} \cdot r_{u_j}}{\sum_{u_i \neq u_j} W_{u_i,u_j}}$

Compute / Predict $\alpha_{u_i,i}$ // find the personal bias factor

Foreach Bundle *B*=1,2,…,n composed of $\{i_1, i_2\}$ do

For (*T*=Min [min(C_1), min(C_2)];*T*<[max(C_1)+max(C_2)];*T*++)

For (C_1=0; C_1<*T*; C_1++)

$C_2 = T - C_1$;

Compute $P_{u_i}(C_1|i_1) = min(P_{\cdot,i_1}(C_1) \cdot \alpha_{u_i,i_1}, 100\%)$

Compute $P_{u_i}(C_1|i_2) = min(P_{\cdot,i_2}(C_2) \cdot \alpha_{u_i,i_2}, 100\%)$

Compute $prob = \frac{P_u(i_1) \cdot P_u(C_1|i_1) + P_u(i_2) \cdot P_u(C_2|i_2)}{1 + \frac{1}{J_{i_1,i_2}}}$

If (*Prob* > *MaxP*)

MaxP = *Prob*;

ExpectedRevenue= *MaxP* * (*T* - *cost* · min(C_1) - *cost* · min(C_2));

If (*ExpectedRevenue*> *MaxExpectedRevenue*)

MaxExpectedRevenue= *ExpectedRevenue*;

BestProposal = *T*;

bestBundles.**Add**(*B*, *MaxExpectedRevenue*)

bestBundles. **Sort by** *MaxExpectedRevenue*

bestBundles. **Take** (*K*)

number of users and n is the number of items. In order to deal with this, major e-commerce systems prefer to carry out expensive computations offline and feed the database with updated information periodically. In this way, they can provide recommendations to users quickly based on pre-computed similarities. The computation complexity of maintaining the user similarities matrix in the worst case is $O(m^2n)$ [24].

In contrast, generating a single recommendation for an active user is a two-step computation. First, we need to find the users most similar to the active user, and then we must scan items to find the ones that better match the active user's interests (according to similar users' purchases). In the worst case, this computation costs $O(n)$ when similarities are pre-computed offline [24].

2. Item to item similarity - The computation complexity of maintaining the items similarities matrix in the worst case is $O(n^2m)$. Those similarities can be pre-computed offline.
3. Personalized demand graph - the complexity of finding the bias factor, $\alpha_{u,\,i}$ for each item and user is the complexity of the SVD algorithm which is $O(\min\{mn^2,\ m^2n\})$ [25].

The optimization model - providing a recommendation to a user under the assumption that single item purchasing probability, items similarities, and personalized demand graph (1–3) were computed, is $\boldsymbol{O}(\binom{\boldsymbol{n}}{2} \cdot \boldsymbol{T}^2)$, where $\boldsymbol{n}$ is the number of items, and $\boldsymbol{T}$ is the span of the possible prices for the bundle, which in the worst case is between 0 to the sum of the maximum prices present in the DB for each of the items that are within this bundle. $\boldsymbol{O}(\binom{\boldsymbol{n}}{2})$ is the complexity of going through all of the possible pairs of items (meaning all of the optional bundles). $\boldsymbol{T}^2$ is the complexity of going through all of the possible combinations of dividing the bundle price span between the two products within the bundle.

Table 1.
Descriptive statistics of the three datasets.

		Dataset 1	Dataset 2	Dataset 3
General statistical information	Total transaction	3425	836,846	44,484
	Number of users	1000	1000	1000
	Number of items	300	300	300
	Time period	1.1.2013–12.7.2014	2.3.2012–23.7.2013	30.4.2014–31.7.2014
	Number of items purchased by an average customer **in the whole period**	3.42	836.846	44.484
	Total **price paid** by an average customer **in the whole period**	1967.6 (ILS)	3825.17 (USD)	302.6 (USD)
	Number of items purchased **together per day** by an average customer	1.434	8.288	2.598
	Total **price paid per day** by an average customer	734.14 (ILS)	37.88 (USD)	6.8 (USD)
Test set	Total transaction	136	86,125	3278
	Number of users	(Top) 20	(Top) 50	(Top) 50
	Number of items	(Top) 150	(Top) 150	(Top) 150
Train set	Total transaction	3289	750,721	41,206
	Number of users	997	1000	996
	Number of items	300	300	300

4. Evaluation

4.1. Datasets

In this section we test our model on three real datasets. Two of the datasets are from e-commerce applications that use single item recommendation methods, one of which sells Xbox games, and the other is a shopping website that sells electrical products and furniture. The third dataset is a supermarket dataset. Each of these datasets can be used to expand regular single item recommendations into bundling recommendations:

- The first is a purchase/transaction database for a large Israeli shopping website that sells electrical products and furniture; this dataset is referred to as "dataset 1." From this dataset we used the data of the top 1000 users (the most active) and top 300 products (most popular), consisting of a total of 3425 transactions. We split the data into train and test sets. The test set consists of the top 20 customers and top 150 products (meaning transactions that include both a user that is one of the top 20 users and an item that is one of the top 150 items) that yielded 136 transactions, whereas the train set consists of all the rest, i.e., 3289 transactions in the train set.
- The second dataset is a supermarket dataset from Kaggle.[1] This dataset consists of commodity purchases and is referred to as "dataset 2." We use the top 1000 customers and top 300 products in the data which comprised of 836,846 transactions. The top 50 customers and top 150 products are mapped into the test set of 86,125 transactions and 750,721 transactions to the train set.
- The third dataset contains data from Microsoft Xbox games; this dataset will be called “dataset 3.” Again, we use the top 1000 customers and top 300 products in the data made up of 44,484 transactions. The top 50 customers and top 150 products are mapped into the test set of 3278 transactions and 41,206 transactions are in the train set.

In Table 1 we summarized the descriptive statistics of the datasets and the division of the datasets into train and test sets.

We chose to use top items and users as test items/users in order to test our model on the denser part of the data distribution, so that the effect of our model would be more apparent and measurable.

[1] https://www.kaggle.com/c/acquire-valued-shoppers-challenge

As we can see, we have three different datasets: small (dataset 1), medium (dataset 3) and large (dataset 2). In Dataset 1, the customers are nor repeated visitors, meaning that they tend to buy a product once and then not to buy again. Dataset 2 represents commodities, so each customer tends to buy a lot of products. In dataset 3, the customers tend to buy a lot of Xbox games, but not as frequently as in the commodities dataset.

4.2. Experiments

We conducted offline evaluation and tested our model by comparing it to SVD and k-NN collaborative filtering. We used k-NN CF with user to user similarity using the Jaccard measure and SVD with Stochastic Gradient Descent as the optimization algorithm [23].

In order to prepare the datasets for our experiments, we organized each transaction with the following information:

- Purchase dateCustomer ID
- Item ID
- Purchase price

We used the Customer ID and Item ID data for the regular recommendation task. In order to determine whether our recommended bundle is considered a hit, we used the purchase date data in the test set. We considered a hit if the two products of the bundle have been purchased by the user within a specified time period. For dataset 1 (shopping website) and dataset 2 (supermarket), this time period was set at up to a *week*, because the purchasing patterns of commodities are characterized by a more frequent buying cycle. For dataset 3 (Xbox games) the time period was set at up to the longer time span of a *month*, because games have a less frequent buying cycle compared to commodities.

We used the purchase price data to evaluate our pricing model.

We evaluated the personal demand function, the product recommendations, and the price recommendations. In each case we describe the goal, the measures used for the evaluation, the validation procedures, and the results:

I. **The personal demand function:**
   a. **Goal** - The personal demand graph represents the probability the user will purchase an item with a given price. This probability has a strong influence on our model, and for that reason it is important to evaluate this personal function and validate that it represents the user’s preferences.

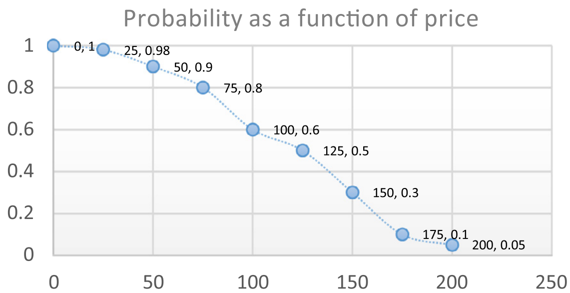

Fig. 3. Example of a demand graph for the expectancy calculation.

b **Measures** - We used the following measures to evaluate the accuracy of the personal demand graph:

i. The **RMSE** measure:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (P_i - R_i)^2}{N}}, \tag{12}$$

where, P_i is the predicted value and R_i is the actual value.

ii. The **MSE** measure:

$$MSE = \frac{\sum_{i=1}^{N} (P_i - R_i)^2}{N}, \tag{13}$$

where, P_i is the predicted value, and R_i is the actual value. Smaller MSE values indicate higher accuracy.

iii. The **expectancy** of each demand function which was compared to the actual price the user paid. The expectancy of a graph is calculated as:

$$Expectancy = \sum_{price \in P(price)=0.005}^{price=Const} price \cdot [P(price) - P(price + Const)] \tag{14}$$

where, *P(price)* is the probability the user bought the item in the given *price*, *price* decreases by *'Const'* (depending on the item's price span) beginning with the price that corresponds to the probability of 0.005, to the *'Const'* value, and *P(price+Const)* is the probability used in the previous iteration.

For example, the demand graph of user *u* for item *i* is presented in Fig. 3 with *x,y* values. *x* represents the price, and *y* represents the probability the user will buy the product at that price.

The price that corresponds to the probability of 0.005 is 200, and the 'Const' was set to 25. The expectancy is calculated as follows:

$$Expectancy = 200 \cdot 0.005 + 175 \cdot (0.1 - 0.005) + 150 \cdot (0.3 - 0.1) + 125 \cdot (0.5 - 0.3) + 100 \cdot (0.6 - 0.5) + 75 \cdot (0.8 - 0.6) + 50 \cdot (0.9 - 0.8) + 25 * (0.98 - 0.9) = 104.625$$

iv. WAPE, Weighted Absolute Percent Error measure [17], meaning:

$$WAPE = \frac{\sum |Predicted\ Price - Actual\ Price|}{\sum Actual\ price} \tag{15}$$

We decided to use this measure and not MAPE [19] since we wanted that each error will have a weight that is relative to the actual price value (where, the error of observation i is $Weight_i = \frac{Actual\ Price_i}{\sum Actual\ price}$), thus errors of high prices will have a higher importance. We used this measure without the absolute value in order to find if the predicted values are lower or higher than the actual values (in average). We call our measure WPE, which is defined as follows:

$$WPE = \frac{\sum Predicted\ Price - Actual\ Price}{\sum Actual\ price} \tag{16}$$

Table 2.
Results of (a), (b), and (c) measures on demand functions represented as linear regression models with degrees of 1,2, and 3.

	Degree = 1	Degree = 2	Degree = 3
(a) Validation set			
1. Dataset 1	**0.00374**	0.0047	0.00531
2. Dataset 2	**4.281**	8.19	8.769
3. Dataset 3	**0.00233**	0.00245	0.00245
(b) Median probability			
1. Dataset 1	**0.0803**	0.0999	0.105
2. Dataset 2	**0.1162**	0.1663	0.164
3. Dataset 3	**0.0297**	0.0326	0.03265
(c) Demand function expectancy (WPE)			
1. Dataset 1	**0.00676**	0.0648	0.362
2. Dataset 2	**0.208**	0.223	0.345
3. Dataset 3	0.2907	0.1736	**0.166**

c. **Validation procedures** - In order to learn all the alphas for each user and item, we used an SVD prediction method. For datasets 1 and 3 we used 1500 iterations, and for dataset 2 we used 200 iterations (described more fully in the results section). To find the demand graph we tested a regression function with degrees of 1, 2, and 3.
We used three procedures in order to evaluate the personal demand function:

i. We defined a validation set as 20% of the existing alpha's data in order to test the predicted alphas in comparison to the actual alphas using the MSE measure.
ii. We compared the personal demand graph probability to 0.5 (median probability) of all purchased products in the test set, also using the MSE measure.
iii. We calculated the expectancy of each demand function and compared it to the actual price the user paid using the WPE measure.

d. **Results -** We built linear regression models with degrees of 1, 2, and 3. Table 2 shows the performance of the three degrees on the measures we defined in the previous section: (a) MSE based on comparing the predicted alphas in the validation set to the actual alphas (b) MSE based on comparing the personal demand graph probability to 0.5 (median probability) of all purchased products in the test set, and (c) the WPE measure based on comparing the expectancy of the demand function to the actual price the user paid.
In Table 2, we can see that in most cases linear regression with a degree of 1 has the best performance (appears in bold in Table 2). The MSE values based on evaluating the alphas of the validation set and the differences from the median probability were all the smallest utilizing a degree of 1. The WPE measure which represents the difference of the price expectancy from the actual price in percentage terms was very accurate in dataset 1 - 0.676%, in contrast to dataset 2 - 20.8% and dataset 3 (with a degree of 3) - 16.6%.
We decided to proceed with a linear regression model with a degree of 1 and predicted the missing alphas using SVD on the existing alphas matrix. The SVD learning iterations are shown in Fig. 4 for datasets 1 and 3 and in Fig. 5 for dataset 2. Dataset 2 is presented in a separate graph, since its RMSE is bigger and has a different scale.

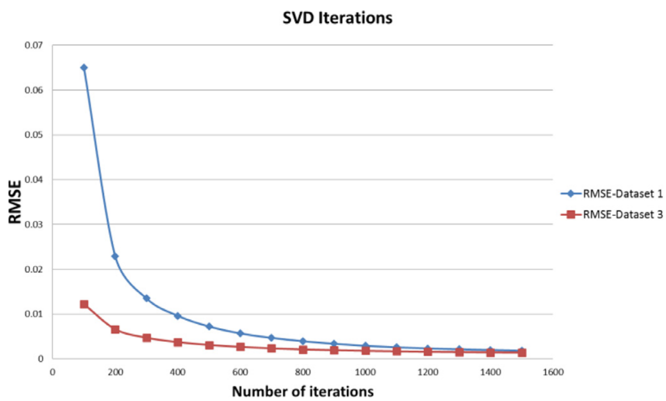

Fig. 4. Predicting alphas using SVD for datasets 1 and 3. The model improves the prediction (measured by RMSE) as the number of iterations decreases.

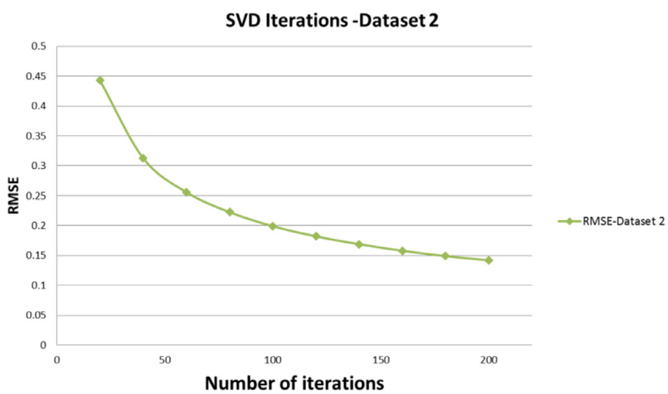

Fig. 5. Predicting alphas using SVD for dataset 2. The model improves the prediction (measured by RMSE) as the number of iterations decreases.

Table 3
Results of (a), (b), and (c) measures on the demand function represented as a linear regression model with a degree of 1.

	Dataset 1	Dataset 2	Dataset 3
(a) Validation set	0.00374	4.281	0.00233
(b) Median probability	0.0803	0.1162	0.0297
(c) Demand function expectancy (WPE)	0.00676	0.208	0.2907

As we can see, for datasets 1 and 3 the RMSE measure converges to a very small value. For dataset 2 the alphas are harder to predict, because the commodities dataset does not reveal significant personal patterns. The patterns are generic and represent most of the population. For that reason, the RMSE in this dataset is higher compared to the other datasets. After 200 iterations the RMSE value tends to increase; thus, we stopped iterating after 200.
Table 3, summarizes the results according to regression function with a degree of 1 which had the best performance.
As we can be observed in Table 3, datasets 1 and 3 have very low MSE values for (a) and (b) measures, confirming that the alpha prediction is suitable, as is the personal demand graph for each user. As mentioned before, in dataset 2 the MSE values are much higher because of the generic character of the data.
The WPE measure for comparing the demand function expectancy to the actual price (c) for dataset 1 is 0.6%, which means that the expectancy price was on average 0.6% greater than the actual price. For dataset 2 the WPE was 20.8%, and for dataset 3 the WPE was 29.07%. These results mean that the expectancy prices for datasets 2 and 3 are 20%-30% greater than the actual price, which is good since we want to recommend higher prices (in order to increase revenue). Moreover, the increase of 20% in the price is logical and not exaggerated. The 0.6% WPE measure for dataset 1 shows that the expectancy of the demand graph is very accurate (compared to the actual price). We believe this is due to the fact that the Israeli shopping website already manipulated the price so that it was as high as possible. Thus, the accuracy here appears to be very close to the actual price.

I. **The product bundling recommendations:**
  a. **Goal** - Our main objective is to recommend personal bundles that the user will purchase. We would like to evaluate if our recommendations provide a good prediction of the products the user will buy in the future and compare the results to state-of-the-art k-NN CF and SVD methods.
  b. **Validation procedure -** Our evaluation is based on a comparison of the top 5 bundles to the top 5 items recommended by k-NN CF and SVD algorithms. If a recommended bundle was purchased in the test set within the established time period, we consider it a hit. In this way the maximum a bundle recommendation list can achieve is 5 hits. We chose to use a fixed number of 5 as our recommendation size, since we deal with bundles (meaning 10 items) and a higher number of recommendations can exhaust the user.
  c. **Measures** - We used the following measures:
    i. **Precision** - the ratio between recommended items that were actually purchased by the user (TP) to all the items that were recommended to the user by the system (those that were purchased-TP and those that were not truly purchased by the user-FP). This measure represents the probability that a recommended item will be purchased:

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

    ii. **Recall** - the ratio between recommended items that were actually purchased (TP) to the total number of purchased items by the user (those that were recommended-TP and those that were not recommended-FN). This measure represents the probability that a purchased item will be recommended:

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

    iii. **Average quantity** that was recommended and purchased - in order to compare this measure with the bundle recommendation list and item recommendation list. We define this as follows: if a bundle was purchased it is valued at 1 quantity, and if one item within a bundle was purchased it is valued at 0.5 quantity. In this way the maximum that the bundle list can achieve is a quantity of 5 units. If an item within the item recommendation list was purchased, it is valued at 1 quantity. In this way the top 5 items list can achieve a maximum quantity of 5.
    iv. **Average price** that was paid for the recommended and purchased products - in order to compare this measure with the bundle recommendation list and item recommendation list, we define this as follows: if a bundle was purchased it is valued at 1 quantity $*$ the price that was paid in the test set for this bundle, and if one item within a bundle was purchased, it is valued at 0.5 quantity $*$ the price that was paid in the test set for this item.

Table 4.
Results of bundling recommendation for dataset 1. Bundle recommendation is considered a hit if the two products were purchased within a *week*.

	Precision	Recall	Average quantity	Average price
k-NN CF	0.027	0.012	0.133	12.133
SVD	0.013	0.033	0.067	70.533
Bundling – strategy I	0.074	0.1	0.4	469.133
Bundling – strategy II	0.058	0.087	0.3	457.467

Table 6
Results of the bundling recommendation for dataset 2. Bundle recommendation is considered a hit if the two products were purchased within a *week*.

	Precision	Recall	Average quantity	Average price
k-NN CF	0.404	0.017	2.02	18.894
SVD	0.476	0.02	2.38	23.811
Bundling – strategy I	0.487	0.02	2.86	44.959
Bundling – strategy II	0.408	0.016	2.26	53.854

If an item within the item recommendation list was purchased it is valued at 1 quantity $*$ the price that was paid in the test set for this item. In this way we can evaluate if the bundle recommendation helped at increasing income.

d. **Statistical significance** - To determine the confidence level of each measure comparison (precision, recall, quantity, and price) we use a Paired T Test which is a test of the null hypothesis that the difference between the two methods we compared has a mean value of zero; the alternative hypothesis is that the bundling models have a greater value than k-NN CF and SVD methods. We calculate the P-value of each test. We used a paired test, because all the methods are tested on the same population.

e. **Results** - We compared the two bundling strategies: I- maximizing the probability; and II- maximizing the expected revenue, to state-of-the-art item recommendation methods (k-NN CF and SVD), using top 5 recommendations, precision, recall, average quantity, and average price measures. The results are shown in Tables 4, 6, and 8. The P-values for the T-tests are shown in Tables 5, 7, and 9. As observed from the results, the null hypothesis that the two methods perform equally is rejected ($p < 0.05$).

As we can see in Table 4, the results for dataset 1 show that all the measures- precision, recall, average quantity, and average price - were improved by recommending the top 5 bundles instead of the top 5 items. Despite this, we can see that the values of the measures are small, because this dataset is hard to predict since the customers are not regular customers, meaning that they tend to buy a product once and not buy again. The number of transactions per customer is very small. Moreover, we can see that the bundling recommendations increased the average price paid for the recommended and purchased products of the top 5 recommendations, which means that our bundling model can increase the buying scope by recommending the most profitable products together. The first bundling strategy of maximizing the probability has better performance in comparison to strategy II of maximizing the expected revenue.

In Table 5, we can see that the p- values of the precision and average quantity measures of strategy I in comparison to all other methods were below to 0.05, providing that the bundling model with strategy I has significantly the best precision and average quantity performance. The recall and average price measures of the bundling model with strategy I were significantly better than the collaborative filtering method and strategy II but not significantly better than the SVD method.

As we can see in Table 6, the results of dataset 2 show that all the measures- precision, recall, average quantity and average price - were improved by recommending the top 5 bundles instead of the top 5 items with the bundle model that maximizes the probability.

Bundling with strategy II (maximizing the expected revenue) yields the highest average price, since this method aims at maximizing the revenues and therefore chooses the expensive products. In Table 7, we can see that the p-values of all measures of bundling strategy I compared to the k-NN CF were below 0.05; thus, this bundling model significantly outperforms the k-NN CF. Bundling with strategy I did not outperform SVD significantly in terms of precision and recall, probably because SVD has better performance than the k-NN CF for this dataset, and the bundling model is based on the k-NN collaborative filtering method. We assume this is the case and leave for future research the issue of whether our bundling model will perform significantly better using SVD techniques rather than the items recommendation SVD.

Strategy I performed significantly better than strategy II for all measures except the average price, a measure in which strategy II performed better. The reason might be that since strategy II maximizes the revenues, it provides recommendations of more profitable products. Strategy II was significantly superior to k-NN CF in the average quantity and average price measures and significantly outperformed SVD only for the average price.

As we can see in Table 8, the results of dataset 3 show that results for all measures - precision, recall, average quantity and average price - were improved when recommending the top 5 bundles instead of the top 5 items with the bundle model that maximized the probability. The results for the k-NN CF method were very close to the bundling model, since our model uses k-NN CF to find the single item purchasing probability. SVD method had the worst performance with this dataset.

In Table 9, we can see that the p-values for all the measures comparing the bundling model to SVD were below 0.05, meaning that the bundling significantly outperforms

Table 5
P -values of T-test for dataset 1, with null hypothesis that the means of the two methods are equal and an alternative hypothesis that the mean of the first method is greater than the second method.

First method	Second method	Precision	Recall	Average quantity	Average price
Bundling – strategy I	**k-NN CF**	0.0189*	0.0135*	0.017*	0.034*
Bundling – strategy I	**SVD**	0.0179*	0.105	0.0222*	0.067
Bundling – strategy II	**k-NN CF**	0.048*	0.0242*	0.041*	0.0375*
Bundling – strategy II	**SVD**	0.0329*	0.143	0.036*	0.072
Bundling – strategy I	**Bundling – strategy II**	0.037*	0.043*	0.0413*	0.042*

Table 7
P-values of T-test for dataset 2, with null hypothesis that the means of the two methods are equal and an alternative hypothesis that the mean of the first method is greater than the second method.

First method	Second method	Precision	Recall	Average quantity	Average price
Bundling – strategy I	**k-NN CF**	0.0039**	0.0416*	2.35E−06**	1.132E−14**
Bundling – strategy I	**SVD**	0.215	0.365	0.00011**	8.411E−09**
Bundling – strategy II	**k-NN CF**	0.350	0.801	0.049*	3.573E−12**
Bundling – strategy II	**SVD**	0.978	0.986	0.670	9.2E−08**
Bundling – strategy I	**Bundling – strategy II**	1.11E−16**	2.05E−09**	2.532E−12**	0.994

Table 8.
Results of the bundling recommendation for dataset 3. Bundle recommendation is considered a hit if the two products were purchased within a *month*.

	Precision	Recall	Average quantity	Average price
k-NN CF	0.283	0.022	1.413	7.42
SVD	0.017	0.002	0.087	0.686
Bundling – strategy I	0.296	0.166	1.946	9.092
Bundling – strategy II	0.197	0.114	1.283	37.273

SVD. Strategy I performed significantly better than strategy II for all measures except the average price. This is consistent with the former results for the same reason. The recall and average quantity measures of the bundling model implementing strategy I were significantly higher than collaborative filtering, but the precision and the average price were not.

In conclusion, in most cases product bundling improves the recommendations for all measures. Bundling recommendations increase the number of purchases and the income. For all datasets the first strategy of maximizing the probability led to higher results in terms of precision, recall, and average quantity than the strategy of maximizing the expected revenue, while maximizing the expected revenue in most datasets provides better results in terms of the average price actually paid for the top 5 recommendations. In some cases our bundling model was significantly better than the k-NN CF and SVD methods.

III. **The price bundling recommendations:**

a. **Goal** - Our second objective of the research is to suggest a price recommendation for each bundle. We would like to evaluate the accuracy of the recommended bundle price and the ability of this price suggestion to leverage (increase) the buying scope and the firm's income.

b. **Validation procedure** - Our evaluation consists of comparing the recommended price to the actual price the user paid in the test set using the WPE measure that we mentioned in Eq. (16), meaning:

$$WPE\_R = \frac{\sum Recommended\ Price - Actual\ Price}{\sum Actual\ price}$$

Table 10.
Results of the price bundling recommendation for dataset 1. WPE_R measures for the recommended price versus the actual price and WPE_M for the mean price versus the actual price.

	Recommended price (WPE_R)	Mean price (WPE_M)
Bundling – strategy I	0.014	0.013
Bundling – strategy II	0.018	0.013

The mean price is compared to the actual price the user paid in the test set using WPE measure, meaning:

$$WPE\_M = \frac{\sum Mean\ Price - Actual\ Price}{\sum Actual\ price}$$

where, the mean price is the sum of the mean prices of the bundle products in the dataset.

In this evaluation phase we compared the WPE_R and WPE_M.

c. **Results -** Tables 10–12 show the WPE_R measure of the difference between the recommended bundle price and the actual price paid by the user in comparison to the WPE_M measure of the difference between the mean price of the two products in the dataset and the actual price paid by the users.

As can be observed in Table 10, the WPE_R measures of the recommended prices were on average 1.4% and 1.8% higher than the actual price, and the mean price was on average 1.3% higher than the actual price. Thus, the recommended prices of the bundling models are accurate in comparison to the actual price and provide a small improvement in the amount paid for each recommendation, and thus may increase revenue. The price bundling recommendation is very accurate since, as we mentioned before, the shopping website already manipulates the prices before recommending them to the customers. Thus, the accuracy here is more than adequate and very close to the actual price

As we can see in Table 11, the WPE_R measures for the recommended prices were on average 42.1% and 51.1% higher than the actual price, and the WPE_M of the mean price was on average 28.5% and 52% higher than the actual price. The recommended prices of the bundling models are higher in comparison to the actual price and do not provide a good estimation of the actual prices. The same is true for the mean price estimation which is higher than the actual price

Table 9
P-values of T-test for dataset 3, with null hypothesis that the means of the two methods are equal and an alternative hypothesis that the mean of the first method is greater than the second method.

First method	Second method	Precision	Recall	Average quantity	Average price
Bundling – strategy I	**k-NN CF**	0.317	4.21E−06**	0.0202*	0.1938
Bundling – strategy I	**SVD**	2.91E−13**	4.24E−07**	3.61E−12**	1.19E−10**
Bundling – strategy II	**k-NN CF**	0.923	6.37E−05**	0.558	2.80E−05**
Bundling – strategy II	**SVD**	7.19E−09**	3.75E−06**	2.91E−08**	1.63E−07**
Bundling – strategy I	**Bundling – strategy II**	0.000202**	0.0071**	0.00037**	0.999

Table 11
Results of the price bundling recommendation for dataset 2. WPE_R measures of the recommended price versus the actual price and WPE_M of the mean price versus the actual price.

	Recommended price (WPE_R)	Mean price (WPE_M)
Bundling – strategy I	0.421	0.285
Bundling – strategy II	0.511	0.520

Table 12.
Results of the price bundling recommendation for dataset 3. WPE_R measures for the recommended price versus the actual price and WPE_M for the mean price versus the actual price.

	Recommended price (WPE_R)	Mean price (WPE_M)
Bundling – strategy I	0.049	0.02
Bundling – strategy II	0.051	0.031

and therefore also doesn't provide a good estimation. The commodity dataset does not contain unique customer's purchasing patterns, and therefore, as shown before, the personal demand graph doesn't provide good predictions. This might be the reason for the poor performance of the recommended prices. Despite this, the recommended prices can guarantee an increase of ~50% in the price customers would pay for the products they are interested in.

As we can see in Table 12, the WPE_R measures for the recommended prices were on average 4.9% and 5.1% higher than the actual price, and the WPE_M of the mean price was on average 2% and 3.1% higher than the actual price. The recommended prices of the bundling model are accurate in comparison to the actual price and provide a 5% improvement in the price for the top 5 recommendations, and thus increase the revenues. Recommending a bundling price that is 5% higher is reasonable and not exaggerated, which means that customers will buy at this price, and the firm's income will increase.

The results show that the prices for the recommended bundles are higher (on average) than the actual prices paid by the customer. This doesn't mean that the bundle's price is higher than the sum of the prices of the individual products that are part of the bundle. The algorithm goes through a span of prices that begins with the minimum price of one of the products that are part of the bundle to the sum of the maximum price of those products. Thus, if a customer actually paid a lower price for both of the products it means that he/she didn't pay the maximum prices that appeared in the database for the two products. Moreover, the consumer may have bought these products as part of a promotion.

In conclusion, the recommended prices of dataset 1 and 3 provide good estimates of the actual prices and can guarantee an increase in income. Dataset 2 yielded the poorest results, because it doesn't contain unique purchasing patterns and the personal demand graph is not accurate. Moreover, the dataset dealing with commodities contains products with low prices, and therefore no range exists for manipulating the prices.

5. Conclusions and discussion

In this work we presented a novel product and price bundling recommendation model that aims at increasing customers' buying scope and the firm's income. The proposed method uses a collaborative filtering method, personalized demand graph, Jaccard similarity measure, and expected revenue/purchasing probability optimization in order to provide top N bundles recommendations with a personalized recommended price for each bundle.

The main contribution of our method is that unlike most recommendation methods that are designed to provide single item recommendations and do not involve personalized price recommendation, our model uses the recommender system platform in order to implement a bundling strategy and utilizes an optimization technique to determine the optimal price for the bundle for individual users. As we demonstrated, this model leverages the single item recommendation method, combining with it bundle recommendations, which can increase the users' buying scope and the firm's income, thereby enhancing the value provided by RSs.

In our evaluation, it was shown that our personalized demand graphs were accurate and represented the customers' buying preferences. For dataset 2 the personalized demand graphs were harder to predict, since the commodities dataset does not reveal significant personal patterns. The patterns are generic and represent most of the population. In addition, we showed that, in comparison to state-of-the-art item recommendation methods, the bundling recommendations can improve precision, recall, the average quantity of products actually purchased, and the average price paid.

The first strategy of maximizing the probability led to higher results in terms of precision, recall, and average quantity than the strategy of maximizing the expected revenue. However, maximizing the expected revenue for most datasets provided better results for the average price actually paid for the top 5 recommendations, since this method chooses the most profitable products. In some cases our bundling model was significantly better than the k-NN CF and SVD methods.

Furthermore, the personalized recommended price for datasets 1 and 3 was up to 5% higher than the actual price, meaning that we provide a good estimation of the actual price, while increasing the potential payment for our recommended bundles. As mentioned, dataset 2 yielded the poorest results, because it doesn't contain unique personal purchasing patterns and there was no range to manipulate the price.

This work raises new questions and additional research directions in the field of product and price bundling recommendations. Therefore, we intend to extend this research in the following directions:

Online A/B testing. Our evaluation method was offline. The bundling method has a much higher effect when offered online to users. Evaluating the online customers' response to the suggested bundles and their prices can increase the research contribution.

Extending the model to more than two products. Our model deals with bundles of only two products. There is a need to see how it can be expanded to more than two products.

Handling quantity data. The algorithm presented in this paper was performed on systems that consist of binary data that represents buying events. There is a need to extend the model to use quantity data that represents how much of each item the customer has bought. This quantity can intensify the preference the customer has for a specific product.

Using data features. Our algorithm didn't use the items' features in order to decide which items should be offered together. Future work can include extending our model to use the items' characteristics, which could improve the bundle recommendations.

Bundling of the same product. Our bundling model offers two different products as a bundle, but in some domains we can extend our model to offer a bundle of the same product twice with a discount (for example, 1+1 campaigns).

Acknowledgement

Our gratitude goes to Microsoft corporate, for making this research feasible for us by providing an adequate data that was used in the experiments part.

References

[1] J. Bobadilla, et al., Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.
[2] R. Burke, Hybrid recommender systems: survey and experiments, UMUAI 12 (4) (2002) 331–370.
[3] J. Leskovec, A. Rajaraman, J. Ullman, 2012. Mining of massive datasets. Chapter 6.
[4] R. Agrawal, T. Imielinski, A.N. Swami, Mining association rules between sets of items in large databases, in: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 22, 1993, pp. 207–216.
[5] B. Goethals, Survey on Frequent Pattern Mining, Univ. of Helsinki, 2003.
[6] M. Reisinger, 2004. The effects of product bundling in duopoly.
[7] G. Tellis, S. Stremersch, Strategic bundling of products and prices: a new synthesis for marketing, J. Mark. 66 (2002) 72.
[8] F. Ricci, et al., ITR: a case-based travel advisory system, in: Advances in Case-Based Reasoning, Springer Berlin Heidelberg, 2002, pp. 613–627.
[9] L. Guo-rong, Z. Xi-zheng, Collaborative filtering based recommendation system for product bundling, Management Science and Engineering, International Conference on, IEEE, 2006.
[10] C. Birtolo, et al., Searching optimal product bundles by means of GA-based engine and market basket analysis, IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), Joint. IEEE, 2013.
[11] A. Campagna, R. Pagh, Finding associations and computing similarity via biased pair sampling, Knowl. Inf. Syst. 31 (3) (2012) 505–526.
[12] J.P. Guiltinan, The price bundling of services: a normative framework, J. Mark. (1987) 74–85.
[13] R. Schmalensee, Gaussian demand and commodity bundling, J. Bus. (1984) S211–S230.
[14] W. Hanson, R.K. Martin, Optimal bundle pricing, Manag. Sci. 36 (2) (1990) 155–174.
[15] T. Zhu, et al., Bundle recommendation in ecommerce, in: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, 2014.
[16] J. Hersch, Gender, income levels, and the demand for cigarettes, J. Risk Uncertainty. 21 (2-3) (2000) 263–282.
[17] J. Chase, W. Charles, Innovations in business forecasting, J. Bus. Forecasting 33 (3) (2014) 22.
[18] C-Y. Tsai, C-C. Chiu, A purchase-based market segmentation methodology, Expert. Syst. Appl. 27 (2) (2004) 265–276.
[19] J. Tayman, D.A. Swanson, On the validity of MAPE as a measure of population forecast accuracy, Popul. Res. Policy Rev. 18 (4) (1999) 299–322.
[20] Q. Zhao, et al., E-commerce recommendation with personalized promotion, in: Proceedings of the 9th ACM Conference on Recommender Systems, 2015.
[21] M. Beladev, B. Shapira, L. Rokach, Recommender systems for product bundling, in: Poster Proceedings of ACM RecSys 2015, 2015.
[22] D. Ben-Shimon, L. Rokach, G. Shani, B. Shapira, Anytime algorithms for recommendation service providers, ACM Trans. Intell. Syst. Technol. (TIST) 7 (3) (2016) 43.
[23] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 8 (2009) 30–37.
[24] M. Papagelis, I. Rousidis, D. Plexousakis, E. Theoharopoulos, Incremental collaborative filtering for highly-scalable recommendation algorithms, in: Foundations of Intelligent Systems, 2005, pp. 553–561.
[25] M.Y.H. Al-Shamri, User profiling approaches for demographic recommender systems, Knowl.-Based Syst. 100 (2016) 175–187.
[26] A. Hernando, J. Bobadilla, F. Ortega, A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model, Knowl.-Based Syst. 97 (2016) 188–202.
[27] G. Guo, J. Zhang, N. Yorke-Smith, Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems, Knowl.-Based Syst. 74 (2015) 14–27.