Journal on Data Semantics manuscript No.
(will be inserted by the editor)

Product-Seeded and Basket-Seeded Recommendations for Small-Scale Retailers

Marius Kaminskas · Derek Bridge · Franclin Foping · Donogh Roche

Received: date / Accepted: date

Abstract Product recommendation in e-commerce is a widely applied technique which has been shown to bring benefits in both product sales and customer satisfaction. In this work we address a particular product recommendation setting — small-scale retail websites where the small amount of returning customers makes traditional user-centric personalization techniques inapplicable. We apply an item-centric product recommendation strategy which combines two well-known methods – association rules and text-based similarity – for generating recommendations based on a single 'seed' product. Furthermore, we adapt the proposed approach to also recommend products based on a set of 'seed' products in a user's shopping basket. We demonstrate the effectiveness of the recommendation approach in the product-seeded and basket-seeded scenarios through online and offline evaluation studies with real customer data.

Keywords Product recommendation · Online shopping · Association rules · Text-based similarity · Hybrid approach · User study

1 Introduction

The benefits that recommender systems (RSs) can bring to e-businesses are widely recognized. In addition to direct increase of revenue, RSs have been shown to increase customer loyalty and direct customers to new items in the product catalog (Dias et al, 2008). Well-known examples of e-commerce recommenders, such as those used by Amazon[1], attract a large user community and typically rely on user-centric recommendation techniques that exploit the target user's shopping history (Sarwar et al, 2001). However, a small-scale retail setting poses additional challenges for product recommendation. Users of small-scale e-commerce websites often do not have extensive shopping history records, many customers being one-time visitors. Consequently, traditional rating-based personalization techniques (i.e., user-based or item-based collaborative filtering) are inapplicable in such settings.

In this work we propose a flexible product recommendation solution which can be applied to various product domains and which provides meaningful recommendations without relying on user profiling. We develop our approach working with two real-world websites — a party costume and accessory store which in this paper we refer to as *Retailer #1*, and a skateboarding shop which we refer to as *Retailer #2*. Both businesses are small-scale retailers, *Retailer #1*'s web site receiving a daily traffic of around 900 visits on average and *Retailer #2*'s site receiving on average 200 daily visits. For both retailers, roughly 50% of the visitors only view one product and few are returning customers. The customer-product purchase data is therefore sparse: during the evaluation period of six months, out of 7800 products in *Retailer #1's* catalog, 2739 items were purchased, roughly 40% of them only once; for *Retailer #2*, out of 1500 products, 236 were purchased, out of them 75% only once.

Since such data is not sufficient for applying *user-centric* recommendation techniques, we adopt an *item-centric* approach, by establishing a degree of *relatedness* between any

This research has been conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

M. Kaminskas · D. Bridge
Insight Centre for Data Analytics
University College Cork, Ireland
E-mail: {marius.kaminskas,derek.bridge}@insight-centre.org

F. Foping · D. Roche
NitroSell Ltd., Cork, Ireland
E-mail: {franclin.foping,donogh.roche}@nitrosell.net

[1] `http://www.amazon.com/gp/help/customer/display.html?nodeId=16465251`

two products in a retailer's product catalog. We identify two techniques for computing item relatedness – one based on textual descriptions of products, and the other based on product co-occurrence in shoppers' browsing histories. The proposed approach is based on a combination of the two techniques. Being able to compute a relatedness score for any pair of products allows us to implement a service which provides product recommendations when a user is viewing a product web page. The viewed product acts as a 'seed' or 'query' for recommending the top-*N* most related products from the catalog, which can be displayed in a recommendation panel on the product page. Furthermore, in addition to the product-seeded approach, we suggest a basket-seeded approach which generates recommendations based on products present in a user's shopping basket.

The contributions of this work are the following: a) analyzing the problem of product recommendation in the particular setting of small-scale retailers; b) suggesting a technique which is applicable to any product domain (provided that the products have text descriptions); and c) performing a user study with real customers of two retail websites.

In the following section we describe product recommendation techniques used in e-commerce. Next, we describe the implementation of the proposed product-seeded approach. Then, we describe the offline experiments conducted to validate the adopted recommendation strategy and the online evaluation of the system. Finally, we describe the basket-seeded recommendation problem and the offline evaluation of the proposed solution.

2 Related Work

A major challenge encountered when applying RS algorithms to real world e-commerce platforms is *data sparsity* — users view or purchase only a small fraction of the product catalog thus making traditional rating-based techniques difficult to apply. Moreover, user profiling in an e-commerce setting is challenging due to the lack of explicit ratings.

Due to the above challenges, e-commerce recommendations typically rely on *item-centric* techniques, i.e., rather than modeling the preferences of each individual user, the recommendation approaches work by establishing item-to-item relations and using these to recommend items (products) that are similar to the ones viewed or purchased by the users (Cho et al, 2005b; Li et al, 2009; Jannach et al, 2015). The core step in such approaches is reliably computing item relatedness, which is often alleviated by employing data mining techniques, such as association rule mining (Schafer, 2009).

Association rule (AR) mining (Agrawal et al, 1994) is a data mining technique which extracts co-occurrence patterns of item sets from a database of item groupings (e.g., product purchase transactions, term occurrences in documents, etc.). While originally applied in sales transaction analysis, AR mining has been successfully adapted to the recommender systems domain for discovering patterns of items that have common ratings (Leung et al, 2006; Sandvig et al, 2007; Gedikli and Jannach, 2010). AR-based recommendations have been shown to be robust against profile injection attacks (Sandvig et al, 2007) and to achieve accuracy levels comparable to (or even higher than) the nearest neighbour recommendation approaches (Gedikli and Jannach, 2010).

Another application area of ARs related to our work is web usage mining, where ARs are applied to discover patterns in user browsing behaviour (Carmona et al, 2012; Matthews et al, 2013). Mining ARs from web usage data has been applied to discover user navigation patterns which allows improving the design of e-commerce websites or adapting their layouts to certain types of users thus providing a personalized browsing experience (Carmona et al, 2012).

Prompted by the diverse application areas such as web usage mining and recommender systems, various extensions have been proposed for the AR mining technique. While the original AR algorithm was designed to handle Boolean data (e.g., product occurrence in a shopping basket), new AR mining algorithms were proposed to handle quantitative data (e.g., time spent viewing a web page) or temporal data (e.g., the date when a web page is accessed) (Matthews et al, 2013). In the RS domain, AR mining extensions were designed to handle quantitative user rating data (Leung et al, 2006), to detect rare item patterns, and to mine personalized ARs for each target user (Gedikli and Jannach, 2010). While the extensions may prove to be useful in our work (e.g., for promoting rare product recommendations), at this stage of the research we rely on the traditional approach to AR mining, although more advanced AR mining approaches may be considered in the future.

Another data mining method related to ARs is sequential rule (SR) mining (Cho et al, 2005a) which models customer buying behaviour over time. The method splits all customer transactions into a finite number of time periods t, clusters similar purchase transactions (i.e., product baskets) in each time to reduce data sparsity, and then represents each customer's transactions over time as a sequence of transaction clusters. Sequential rules of the form $C_1,\ldots,C_{t-1} \Rightarrow C_t$ are then mined from the transaction database in a way similar to AR mining (i.e., using confidence and support thresholds). Based on the target user's purchase behaviour prior to the current time period t the most similar sequential rule is found, and products from the transaction cluster C_t are recommended to the user.

E-commerce recommendation techniques often combine the item-centric techniques such as AR mining or SR mining with the traditional user-centric recommendation approaches like collaborative filtering (CF). For instance, Liu et al (2009) proposed a hybrid combining SR mining and CF approaches.

Given the current time period t, the SR approach was used to generate recommendations exploiting the user's purchase data from time periods $\{1,\ldots,t-1\}$, while the CF component was used to find recommendations using purchase data from the current period t. The recommendations from SR and CF components were then aggregated in a linear combination. A similar hybrid approach was adopted by Choi et al (2012) who mined the sequential rules on the individual product level (rather than mining rules on transaction clusters).

More recently, Jannach et al (2015) suggested a hybrid product recommendation strategy combining state-of-the-art recommendation approaches, such as a learning-to-rank technique for implicit user feedback (intended to capture the long-term user preferences), with item-centric approaches exploiting the user's recent browsing history (intended to capture the user's short-term shopping goals). The item-centric approaches used by the authors include recommending items based on their co-occurrence in the users' shopping sessions, recommending items that share content features (brand and category) with those recently viewed by the user, and recommending items recently viewed by the user.

Another common technique applied in e-commerce recommendation solutions is the use of product taxonomies which group products sold by the retailer into a hierarchical structure of product categories. Substituting the purchase/ browsing data of individual products with respective categories (at the desired taxonomy level) allows reducing data sparsity. This in turn enables the application of traditional recommendation techniques, such as collaborative filtering, which are difficult to apply on the original sparse data.

Cho and Kim (2004) proposed a heuristic algorithm for determining the right "grain" of product categories to be used for data sparsity reduction. The authors also distinguished three levels of user's involvement with an item — an item view, a basket placement, and an item purchase. The data, obtained from user transaction logs, was converted into implicit ratings using a weighted combination of the three types of events (most importance was given to the purchase events and the least to the item view events). The resulting rating matrix was then used for generating CF recommendations. Cho et al (2005b) later reused the same idea for AR mining — using product taxonomies to reduce data sparsity and mining product association rules for each type of user feedback (item view, basket placement, item purchase) separately. The mined ARs were combined into a single item similarity score (again giving most importance to item co-occurrence among purchases and least importance to item co-occurrence among viewed items).

Another way to handle the sparse purchase data was proposed by Li et al (2009) who modeled the grocery recommendation problem as a bipartite graph with users and items as nodes, and edges representing the purchase of an item by a user. The authors computed product similarity using transition probabilities between items in the graph (passing through the user nodes). While the first order of transition probabilities only allowed establishing similarity between items that were bought together, repeating the probability propagation resulted in higher orders of similarity. This allowed establishing similarity between items that did not appear in the same baskets but were related through common neighbours, thus alleviating the data sparsity problem.

The e-commerce research discussed above are applied to large-scale retailers, such as large online retailers (Cho and Kim, 2004; Cho et al, 2005b; Jannach et al, 2015) dealing with millions of sale transactions, thousands of active customers, and rich product taxonomies. This ensures that the applied item-centric techniques provide sufficient user and product coverage and can be effectively combined with user-centric recommendations techniques. Product recommendation for small-scale retailers is even more challenging compared to the large-scale retail setting, particularly due to the small number of returning customers and limited purchase history of individual users. Moreover, since small-scale retailers typically offer a limited product variety, their product taxonomies are overly generic and cannot be effectively used to reduce data sparsity.

To the best of our knowledge, there are few works which explicitly address the challenges of small-scale retail recommendation. Chen et al (2014) suggested combining product association rules with a number of heuristics for providing recommendations when the available data is not sufficient for association rule mining. The proposed heuristics included recommending products that are most popular among users from the target user's country, or products that are most frequently purchased in the last month.

Similarly to Chen et al (2014), we employ AR mining, however we address the data sparsity problem by combining ARs with text-based item similarity. Moreover, similarly to Cho and Kim (2004), to cope with the limited amount of purchase data, we use product views as a source for AR mining.

Finally, we note that the performance results of the different e-commerce recommenders reported in the literature are difficult to compare due to the different product datasets and evaluation protocols. Typically, e-commerce recommendations are evaluated in offline experiments, using recall and precision metrics as well as their harmonic mean — the F1 measure. The reported recall and precision values (for top-10 recommendations) usually stay below 0.1 (Cho and Kim, 2004; Cho et al, 2005b; Choi et al, 2012; Liu et al, 2009), although some works report recall values above 0.5 (Jannach et al, 2015). In our work, we compute the recall and precision metrics for the offline experiments (see Sections 4 and 7), but we observe that recall is sufficient for comparing the competing recommendation techniques, since the obtained rankings of techniques are identical when using the precision metric (and with certain evaluation methodologies precision is linearly related to recall, see Section 7).

To the best of our knowledge, no works on e-commerce recommendations report results of live evaluation studies. We report the findings of online A/B experiments with two small-scale retailers in Section 5.

3 The Product-seeded Recommendation Approach

We observe that retail websites typically organize the product data into categories (i.e., taxonomies (Cho and Kim, 2004)) containing products that are similar in terms of their intended use, for instance, the product *reindeer costume* may belong to a category *animal costumes*. We exploit such grouping when evaluating our approach in an offline setting (see Section 4).

Furthermore, individual products can vary according to certain characteristics (e.g., size or colour). For instance, the product *reindeer costume* may vary by size — small or large. The item *small reindeer costume* is the actual product variant sold by the retailer. Given such an organization of products, our goal was to design a recommendation service which functions on the level of products to avoid recommending variants of the same product (e.g., recommending a small reindeer costume for users viewing a large costume of the same kind).

The proposed item-centric product-seeded recommender first computes relatedness scores for any pair of products in the retailer's catalog. Then, given a product viewed by the user, the system can obtain all scores between the viewed product and other products in the catalog, rank them according to the score, and recommend the top-*N* products to the user. The product relatedness scores can be pre-computed, since they do not depend on the user.

We view product relatedness as either item *similarity* or *complementarity* — two Christmas-themed costumes may be considered similar to each other, while a costume and a matching accessory are complementary. The proposed text-based relatedness computation approach mostly allows capturing product similarity relations, while the co-occurrence-based approach may capture both similarity and complementarity relations. Next we describe the two approaches for computing product relatedness scores.

3.1 Text-based Approach

The text-based similarity computation is a technique widely used in web mining, information retrieval, and natural language processing, since it allows estimating similarity between a pair of text documents and may be used for matching a user's query to documents, for document clustering, etc.

To compute the text-based relatedness of two products, we represent each product as a document concatenating the *name*, *keywords*, and *description* of the product taken from the retailer's database.

The text documents are then preprocessed using stop-word removal, stemming, and tokenization, converting the documents into a *bag of n-grams* representation. The collection of all product documents is then turned into a matrix of feature vectors with one row per document (i.e., a product) and one column per feature (i.e., a token). We use Python's scikit-learn package[2] for text preprocessing and building the document matrix.

Having built the document matrix, we can compute the similarity between any pair of vectors in the matrix (i.e., documents). We define the text-based relatedness score of two products as the cosine similarity between their vector representations:

$$\text{rel}_{text}(i,j) = \frac{d_i \cdot d_j}{\|d_i\| \times \|d_j\|} \tag{1}$$

where d_i and d_j are the vectors of the documents describing products i and j.

The process of text preprocessing and creating vector representations of the documents depends on a number of settings, e.g., the minimal length of terms to be considered for tokenizing the documents, the n-gram length range, etc. The optimal configuration of these settings was determined through an offline evaluation (see Section 4).

3.2 Co-occurrence-based Approach

The second technique we employ for computing relatedness scores uses association rule (AR) mining. While the general form of an AR is $(X \Rightarrow Y)$ where X and Y are sets of products and the presence of items X implies a high chance of observing items Y, we limited our analysis to rules containing one product on each side, i.e., $(i \Rightarrow j)$, where i and j are products.

Since the *purchase* transaction records of small-scale retailers typically do not provide sufficient product catalog coverage, we employ product *views* for AR mining. The underlying assumption for this method is that if two products are frequently viewed in a single user session, they are related to each other.

To get the product view data, we require a log of product pages accessed by users. We implemented and deployed user tracking functionality on the websites of the two retailers in our study. We stored the acquired data as a log of product page views attributed to permanent user session IDs. The ARs are extracted from this log using the Apriori AR mining algorithm (Agrawal et al, 1994).

For any pair of products for which there is a rule $(i \Rightarrow j)$, we define a relatedness score between products i and j,

[2] http://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

similar to the confidence of the corresponding rule:

$$\mathrm{rel}_{AR}(i,j) = \frac{|\{S_U : i,j \in S_U\}|}{|\{S_U : i \in S_U\}|} \quad (2)$$

where S_U is the set of all user sessions, and a user session is the set of all product page views accessed by the user.

Since the AR-based approach relies on actual product views, we cannot guarantee complete coverage of the product catalog. In other words, there will be products which do not appear in any rules and therefore do not have a set of related products. In fact, for the two retailers that were involved in the study, the catalog coverage was equal to 6% and 10% of the products.

Moreover, even if a product does appear in a rule, it is typically found in only a few association rules; for our two retailers, those products which appear in rules appear in only 1 to 3 association rules. Since our goal is to compute the top-N related products for any given catalog product (and for any N value), we cannot rely solely on the AR-based approach for product recommendation. However, the approach proves to be valuable when combined with the text-based approach as we show in Section 4.

3.3 Hybrid Approach

Unlike the AR-based approach, the text-based approach is able to compute the relatedness scores for any pair of products in the catalog (assuming they all have text descriptions). Therefore, we propose a hybrid combination of the two techniques: given a product, we compute k of the top-N related products by first applying the AR-based approach ($k \in [0,N]$), and then fill the remaining $N-k$ slots with the top-ranked products returned by the text-based approach. The precedence of AR-based approach over the text-based technique was chosen because the ARs are more accurate (see Section 4, Table 2) and they cover both the similarity and complementarity aspects of product relatedness.

Additionally, we have implemented a hybrid approach combining association rules with product popularity. Having computed the top-k related products with the AR-based approach, we fill the remaining $N-k$ with the most popular items (popularity estimated as the number of product views).

4 Offline Evaluation of the Product-seeded Recommendations

We use offline experiments to determine the optimal configuration of the text-based approach described above and to compare the different product recommendation approaches, using data of the two retailers. Both retailers use NitroSell eCommerce — a configurable shopping platform which provides product data and purchase transaction storage facilities.

NitroSell's platform provides a basic product recommendation panel displaying up to 8 product suggestions when a user is viewing a product page. Therefore, in our experiments, we set $N = 8$ when generating the top-N recommendations.

In Nitrosell's platform at present, the recommendations for each product (which populate the recommendation panels) are primarily determined manually by the retailer combined with (very limited) information about product co-occurrences among purchased items. Our aim was to improve this legacy approach to recommendation.

4.1 Experimental Setup

Evaluating the proposed product relatedness computation requires a *ground truth* of product relatedness. In other words, to evaluate the relatedness scores that our algorithms compute, we need to know which products are actually related in reality. Since such information is not directly available in retailers' datasets, we approximated it with two sources of information — the co-purchased items and items belonging to the same product theme:

1. *Co-purchased* products are pairs of products that co-occurred in user baskets when they made a purchase at the online store, and these were available to us because NitroSell's platform records them in its database.
2. *Co-themed* products are related by a theme, which is manually assigned to them by the retailer, e.g., all party costumes and accessories sold during the Christmas period might be assigned a *Christmas* theme. To perform a more detailed evaluation of the recommendation approaches, we also considered two subsets of the co-themed products as distinct ground truth sources.
3. *Substitute* products belong to the same theme and the same product category. We assume a pair of such products to be substitutes for each other, e.g., two different Christmas-themed animal costumes.
4. *Complementary* products belong to the same theme, but different product categories. We assume a pair of such products to complement each other, e.g., a Christmas animal costume and a Christmas Santa costume.

The above sources of information are not available for all products in the retailers' product catalogs. Therefore, as made explicit in Table 1, we restricted the offline experiments to the products that are covered by the ground truth information and performed the experiments for each of the four product sets independently.

For each of the four types of ground truth (*Co-purchased*, *Co-themed*, *Substitute*, and *Complementary*), we denote the set of products covered by the ground truth as P and define *recall* and *precision* metrics:

$$recall = \frac{|\{p \in P \, : \, (Rel_p \cap Top_p) \neq \emptyset\}|}{|P|} \tag{3}$$

$$precision = \frac{\sum_{p \in P} |\{i \in Top_p \, : \, i \in Rel_p\}|}{N \cdot |P|} \tag{4}$$

where Rel_p is the set of products related to product p according to the ground truth, and Top_p is the set of top-N products retrieved by the product relatedness computation approach. In other words, we are measuring the ratio of products for which we can correctly recover at least one related item in the ground truth, and the average ratio of correct product recommendations in top-N.

4.2 Results

As a baseline approach for comparing against the proposed recommendation approaches, we used popularity-based product selection — for any given product, the top-8 most popular products (in terms of page views) were selected. In addition to the pure text-based approach, we used the hybrid combination of AR-based and text-based techniques, and the combination of AR-based and popularity-based methods (see Section 3.3).

For each product recommendation approach, we computed four recall and precision values — one for each type of ground truth described in the previous section. Table 1 shows the evaluation results for *Retailer #1*. The obtained results show all proposed approaches to significantly outperform the popularity baseline. The text-based approach and the hybrid combination of AR and text-based techniques significantly outperform other methods. (Results for *Retailer #2* were analogous and are therefore omitted).

While the results in Table 1 show the hybrid combination of the text-based and AR-based approaches to achieve the best performance values, the differences between the pure text-based approach and the hybrid approach are not statistically significant. This is because the AR-based approach is applicable to only 6% of *Retailer #1's* product catalog, and so its usefulness is 'lost' in the averaging of the recall and precision values for all products in the ground truth sets.

Therefore, to confirm the usefulness of the AR-based approach (hence supporting selection of the AR + text hybrid), we report the metric values for each ground truth considering only products that are covered by the ARs (Table 2).

The results show a clear advantage of the pure AR-based approach over the text-based approach. This is particularly evident for the *Co-purchased* products. We conclude that the AR-based approach can correctly identify related products for the portion of the catalog that it covers. Since these products are likely to be the most popular (most frequently viewed) in the catalog, it is essential to include the AR-based approach when generating recommendations. We therefore selected the hybrid combination of the AR-based and text-based techniques to be used in the online experiments.

5 Online Evaluation of the Product-seeded Recommendations

Having identified the best method of computing the product relatedness score, we deployed the proposed product recommender on the two retailers' websites, integrating the recommendation panel into NitroSell's platform.

The online evaluation of the recommender was conducted within an A/B testing framework: website users were randomly assigned to either group A or group B. Users in group A were shown the legacy version of the recommendation panel, while users in group B were shown the panel generated using the proposed technique — a hybrid combination of AR and text-based approaches. As we discussed in Section 4, the legacy recommendations are primarily determined manually. Therefore, the legacy version of the panel provides a non-trivial baseline for the evaluation, as we are comparing automatically generated recommendations against manually-defined ones.

5.1 Experimental Setup

To compare the effectiveness of the product recommendations in groups A and B, we identified the users by a persistent session ID. Once randomly assigned to group A or B, the users were kept in the same group for subsequent visits to the website. The experiment data was logged by recording uniquely identifiable records — *events*. Event entries consist of a number of attributes, among others:

- *eventType* defines the type of the logged event and may have the following values: {*productview*, *addtobasket*, *removedfrombasket*, *ordercomplete*}. These event types correspond to the following events, respectively: the web page for the product was viewed by the user, the product was added to the user's basket, the product was removed from the basket, and the purchase of the items in the basket was completed;
- *recommendedItems* defines the list of products that were displayed in the product recommendation panel on the product's web page (applicable to events with eventType = *productview*);
- *orderTotal* denotes the value in euros of the completed order (applicable to events with eventType = *ordercomplete*);
- *timestamp* denotes the time of the logged event.

Table 1 Recall (and precision) values for *Retailer #1*. The values marked with * are significantly better than the *Popularity* approach (using a two-proportion z-test for recall metric and a t-test for precision metric, $p < 0.001$). The values marked with ** are significantly better than both *Popularity* and *AR + pop.* approaches ($p < 0.001$)

Approach	*Co-purchased items (5020 products)*	*Same theme items (4445 products)*	*Substitutes (4170 products)*	*Complementaries (3085 products)*
Popularity	0.16 (0.022)	0.094 (0.023)	0.005 (0.001)	0.135 (0.032)
AR + pop.	0.232* (0.045*)	0.185* (0.047*)	0.112* (0.025*)	0.141 (0.033)
Text-based	0.645** (0.278**)	0.91** (0.59**)	0.83** (0.475**)	0.222** (0.053**)
AR + text	0.653** (0.284**)	0.912** (0.591**)	0.839** (0.478**)	0.222** (0.053**)

Table 2 Recall (and precision) values for products covered by ARs (*Retailer #1*). The marked values are significantly better than the *Text-based* approach (two-proportion z-test for recall metric and a t-test for precision metric, * $p < 0.01$, ** $p < 0.001$).

Approach	*Co-purchased items (670 products)*	*Same theme items (577 products)*	*Substitutes (547 products)*	*Complementaries (367 products)*
Text-based	0.578 (0.141)	0.792 (0.174)	0.713 (0.158)	0.065 (0.01)
AR-based	0.706** (0.18**)	0.811 (0.187)	0.815* (0.188*)	0.068 (0.01)

A user *session* is defined as the set of events attributed to the same session ID value. Each session can belong to only one experiment group.

5.2 Performance Metrics

For each experiment group, we computed a number of performance metrics to compare the user behavior and the effectiveness of product recommendations in the two groups. The following metrics were used in the evaluation:

– The click-through rate for the product recommendation panel, which we define as the ratio of product page views which originated from a click on a recommended product over the total number of product page views:

$$\frac{|\, e \in E_G \,:\, \text{eventType=}\textit{productview} \,\&\, \text{productId} \in R_G \,|}{|\, e \in E_G \,:\, \text{eventType=}\textit{productview} \,|}$$

where E_G is the set of all events in the target experiment group ($G = \{A, B\}$) and R_G is the set of all product IDs found in the *recommendedItems* attribute values among events that occurred before $e.timestamp$ in the same session.

– The average number of product page views per session:

$$\frac{|\, e \in E_G \,:\, \text{eventType=}\textit{productview} \,|}{|S_G|}$$

where S_G is the set of *sessions* in group G. This metric corresponds to the average session length which is a common performance metric in e-commerce.

– The average number of completed orders per session:

$$\frac{|\, e \in E_G \,:\, \text{eventType=}\textit{ordercomplete} \,|}{|S_G|}$$

which corresponds to the *conversion rate* — another common performance metric for e-commerce systems.

We note that the definition above of a *recommendation click* is not strict — it does not require the user to immediately click on a recommended product, but includes product page views of the recommended item that occur later in the session. The rationale behind this is that even if users do not directly click on the recommendation, they may be driven to search for it later. A stricter definition of the recommendation click is one where we consider only product page view events whose product ID was among the recommendations in the *previous* session event. We report results for both relaxed and strict definitions.

5.3 Results

The results that we present here come from running the online experiment between March 3^{rd} 2015 and August 25^{th} 2015 on *Retailer #1's* website, and between March 30^{th} 2015 and August 25^{th} 2015 on *Retailer #2's* website. Prior to analyzing the collected data, we filtered the log to exclude duplicate events (which may occur when refreshing a webpage) and to discard user sessions that either contain no product page views, do not begin with a product page view, or consist of one event only (this indicates customers being redirected from third party shopping platforms).

The remaining data amounts to 7850 (8158) unique user sessions in group A (B resp.) for *Retailer #1*, and 1516 (1627) user sessions in group A (B resp.) for *Retailer #2*.

We first measured the recommendation panel *click-through rate* for the two websites. For *Retailer #1*, the results show a rate of 0.05 for group A and 0.1 for group B (using the strict definition of the recommendation click), and 0.16 (0.25) for group A (B resp.) using the relaxed definition. The numbers for *Retailer #2* data are 0.07 (0.19) for the strict definition and 0.17 (0.37) for the relaxed definition in groups A (B resp.). Both retailers show consistency in the results — the users are more likely to click on a recommended product when it is generated using the proposed approach. We also observe that users are more likely to click on the recommendation panel on *Retailer #2's* website. This can be explained by the different placement of the panel on the two websites: *Retailer #1* displays the panel at the bottom of the page, therefore preventing some users from seeing the panel without scrolling, while *Retailer #2* displays recommendations on the side of the screen, making them more visible to the users.

The *average session length* for both retailers is slightly higher for group B: for *Retailer #1* the sessions had an average of 6.4 page views in group A and 6.9 in group B; for *Retailer #2* the values are 4.9 (6.8) for groups A (B resp.).

The *conversion rate* for both retailers showed no difference between the experiment groups: for *Retailer #1* both groups showed an average of 0.14 orders per session, for *Retailer #2* an average of 0.04 orders per session.

To further analyze the purchase data in the two experiment groups, we restricted our analysis to users who clicked the recommendation panel at least once during their interaction with the website. Tables 3 and 4 present the number of completed orders and total revenue (in euros) among all recorded sessions, and among sessions that contain a recommendation click (*SD* – strict definition, *RD* – relaxed definition).

For *Retailer #1*, the total revenue numbers are approximately equal in both groups. But, when restricting the analysis to user sessions that contain a recommendation click, the total revenue is higher for group B, due to the fact that this group contains more sessions with recommendation clicks. For *Retailer #2*, the total revenue is higher for group B — both for all the user sessions, and for sessions containing a recommendation click.

To summarize, we observe that the product recommendation panel in both websites is not frequently noticed by the users. This may be influenced by the visibility of the panel, so alternative placement strategies may be explored in the future. However, among users who click on the recommendations, the number of completed orders and total revenue are higher in group B. This leads us to believe that the proposed recommendation approach brings benefit to the retailers.

5.4 Comparison of the Components of the Hybrid Recommender

To understand which component of the hybrid recommender – the AR-based or the text-based – produces more attractive recommendations, we analyzed their click-throughs separately.

As mentioned above, for *Retailer #1*, we obtained a recommendation click-through rate of 0.1 (using the strict definition of a recommendation click — a page view of a product which was among the recommendations on the *previous* page): out of 56020 product page views, 5855 resulted from clicks on recommendations. Out of these 5855 recommendations, 1143 were generated by the AR-based technique and 4712 by the text-based technique. However, due to the low catalog coverage of ARs (see Section 3.2), recommendations generated by the AR-based technique are less frequently available compared to the text-based recommendations. A proper comparison requires that we normalize the recommendation click frequencies by the number of times AR-based (text-based resp.) recommendations were displayed to users. In this case, we obtained click-through rates of 0.084 for the AR-based approach and 0.046 for the text-based approach. Therefore, AR-based recommendations are almost twice as likely to be clicked on when displayed to users.

Furthermore, if we consider only the recommendation click-throughs that led to the clicked product being placed in a basket, we obtain a normalized click-through rate of 0.02 (0.008) for the AR-based (text-based resp.) recommendations. If we consider only the recommendation clicks that led to the clicked product being purchased, we obtain normalized click-through rates of 0.009 (0.003) for the AR-based (text-based resp.) recommendations.

For *Retailer #2*, we obtained a recommendation click-through rate of 0.19: out of 11054 product page views, 2068 resulted from clicks on recommendations. Of these, 59 were generated by the AR-based technique and 2009 by the text-based technique. The normalized click-through rates for the two techniques are 0.09 (0.05) for the AR-based (text-based resp.) recommendations. Considering only the recommendation clicks that led to the clicked product being placed in a basket, we obtained 0.006 (0.003), and considering only the recommendation clicks that led to the clicked item being purchased, we obtained 0.002 (0.0007).

The presented results indicate that while the AR-based recommendations are less commonly available than the text-based recommendations (due to the low catalog coverage of ARs), when displayed to users they are more likely to be clicked on and can therefore be considered more attractive to customers. This finding confirms the offline evaluation result that showed better performance of the AR-based approach compared to the text-based approach (see Section 4, Table 2) and validates the decision to give the AR-based approach precedence over the text-based technique in the hybrid combination (Section 3.3).

6 The Basket-seeded Recommendation Approach

Having validated the effectiveness of the hybrid solution for product-seeded recommendation, we have begun an investigation of a different product recommendation problem — generating recommendations based on the contents of the user's basket. In this section we present an early-stage analysis of a recommendation solution whose input is a set of

Table 3 An analysis of completed orders for *Retailer #1*.

	Num. of sessions		*Num. of orders*		*Total revenue*	
Group	*A*	*B*	*A*	*B*	*A*	*B*
All sessions	7850	8158	1067	1114	38907	39720
SD sessions	1713	2910	340	536	14814	20758
RD sessions	2545	3655	606	737	24688	28016

Table 4 An analysis of completed orders for *Retailer #2*.

	Num. of sessions		*Num. of orders*		*Total revenue*	
Group	*A*	*B*	*A*	*B*	*A*	*B*
All sessions	1516	1627	62	71	4735	6258
SD sessions	329	689	19	28	1285	2832
RD sessions	458	753	27	34	2168	3181

products that are present in the target user's shopping basket and whose output is a set of recommendations of other products the user may be interested in. The e-commerce platform may exploit such basket-aware recommendations in a number of ways:

- Recommendations may be displayed at checkout time, allowing users to update the contents of their baskets;
- Recommendations may be displayed after the user has added a product to the basket, suggesting additional items to purchase;
- Recommendations may be provided after the user has removed an item from the basket, to suggest potential replacements for the removed product;
- Recommendations may be sent to the user together with a reminder of an "abandoned" basket.

So far, we have implemented and evaluated three techniques for basket-seeded recommendations that take in a set of products present in the user's basket (i.e., query items) and return a top-N list of recommended products:

- The *product-related* approach is based on the product-seeded approach described in Section 3. Since the product-seeded approach generates top-N recommendations given a *single* 'seed' product, we adapt it to recommend top-N products for a *set* of 'seed' products (i.e., the query items in the basket) using a *rank aggregation* technique (Dwork et al, 2001). Given a set of query products Q (i.e., a basket), for each product $q \in Q$, we obtain a ranked list σ_q consisting of top-N recommendations returned by the product-seeded approach. We denote an item's i position in the ranked list as $\sigma_q(i) \in [1,N]$. Then, given the set of $|Q|$ ranked lists, we compute the rank aggregation score for each distinct item i found in the rankings:

$$\text{rank aggregation}(i) = \sum_{q \in Q} N - \sigma_q(i) \tag{5}$$

Finally, we rank the items according to their rank aggregation scores and recommend the top-N items.

- The *basket-related* approach is based on association rules mined from the basket data. Unlike the *co-occurrence-based* approach for product-seeded recommendations (see Section 3.2), in this approach we do not restrict AR mining to rules containing one product on each side ($i \Rightarrow j$), but consider all rules whose antecedent is a subset of the query items Q and whose consequent is a single product. This approach is similar to the adoption of ARs for collaborative filtering proposed by Sandvig et al (2007).

Algorithm 1: The generation of recommendation lists for the basket-related approach. (We use '++' to designate list concatenation.)

```
Data: a set of query products $Q$ (i.e, a basket); a set of
      association rules $A$
Result: a set of recommendation lists $RecLists$

$RecLists \longleftarrow []$;
for $l \longleftarrow len(Q)$ to 1 do
    $L_l \longleftarrow []$;
    $Subsets_l \longleftarrow \{S : S \subset Q \wedge |S| = l\}$;
    for $S \in Subsets_l$ do
        for $i \in \{p : `S \Rightarrow p' \in A\}$ do
            $L_l \longleftarrow L_l ++ [i]$;
        end
        $RecLists \longleftarrow RecLists ++ [L_l]$;
    end
end
return RecLists;
```

Given the set of query items Q, we check for ARs of the form $S \Rightarrow i$, where $S \subseteq Q$, and add i to the list of recommendations. A separate list is constructed for each length value $l = |S|$. The algorithm for obtaining the set of all recommendation lists for each l value is shown in Alg. 1. Having obtained the set of recommendation lists for all non-empty subsets of Q, we compute the aggregated score for each distinct item i present in the lists similarly to the rank aggregation described above, but without taking the item's rank value into account:

$$\text{aggregated score}(i) = \sum_{L \in RecLists} L(i),$$
$$\text{where} \quad L(i) = \begin{cases} 1, & \text{if } i \in L \\ 0, & \text{otherwise} \end{cases} \qquad (6)$$

We then rank the items according to their aggregated scores and recommend the top-N items.

Note that in this approach we do not address rule redundancy (Kotsiantis and Kanellopoulos, 2006) — a rule $R : A \Rightarrow i$ may be 'covered' by a more general rule $R' : B \Rightarrow i$ where $B \subset A$. When aggregating the score for product i, we consider both rules R and R', which has the not-inappropriate effect of boosting the score of products that appear in more specific rules. A more sophisticated handling of the association rules may be investigated in the future work.

- The *hybrid* approach combines the *product-related* and the *basket-related* techniques in a way similar to that of the hybrid approach for product-seeded recommendation (see Section 3.3). Similarly to the ARs mined from user browsing data for the product-seeded recommendation problem, the basket ARs do not cover the full product catalog, therefore, the *basket-related* approach may not always generate N recommendations for a given set of query products. To account for this, we compute k of the top-N recommendations using the basket-related approach ($k \in [0,N]$), and then fill the remaining $N-k$ slots with the top-ranked products returned by the *product-related* approach.

  The precedence of the basket-related technique over the product-related technique in the hybrid combination was chosen because the basket ARs were shown to provide a more reliable source of information (see Section 7.2, Table 5).

7 Offline Evaluation of the Basket-seeded Recommendations

At this stage of the research, we evaluate the three basket-seeded recommendation techniques in an offline setting, using the user basket data collected during the online experiments described in Section 5. The main goal of the offline experiments is to identify promising techniques prior to conducting time-costly online experiments on retailers' websites.

7.1 Experimental Setup

The offline evaluation methodology is based on splitting the set of product baskets into *train* baskets and *test* baskets. The train baskets are used for training a recommendation approach. Then, each test basket is further split into a set of *query products* and a single *target product*. For each test basket, its query products are used as input for the recommender which outputs a ranked list of top-N recommended products ($N = 8$ was used in all the experiments). The recommendations are then compared against that basket's target product. The performance metric which captures the accuracy of the evaluated approach is defined as:

$$recall = \frac{|\{b \in B \,:\, target_b \in Top_p\}|}{|B|} \qquad (7)$$

where B is the set of all baskets used in the evaluation, $target_b$ is the target product for basket b, and Top_b is the set of products recommended by the evaluated approach for basket b.

Since the adopted evaluation methodology uses only one target product ($target_b$ in the equation above), the corresponding precision values are proportional to those of the recall metric (i.e., $precision = recall/N$). We therefore omit the precision metric in our experiments and report only recall values.

We note that an evaluation procedure like this may be biased toward complementary items (e.g., party costumes and their accessories) rather than substitute items (e.g., alternative versions of the same party costume), as user baskets are more likely to contain items that complement each other. This may hurt the performance of approaches that are geared towards recommending substitute items. To account for this bias, in the future we intend to measure the distribution of complementary/substitute items among recommendations generated by the different approaches (see Section 8).

7.2 Results

We ran the offline experiments using the three basket-seeded recommendation approaches on basket data collected during the online experiment run between March 3^{rd} 2015 and August 25^{th} 2015 as described in Section 5. We considered the basket data of all completed orders containing at least two products. This resulted in 1602 baskets for *Retailer #1* (mean size of a basket is 6.13, $SD = 4.19$) with 2553 distinct products present in the baskets, and 79 baskets for *Retailer #2* (mean size of a basket is 3.12, $SD = 2.19$), with 175 distinct products present in the baskets. Since the basket data for *Retailer #2* is not sufficient for association rule mining, we conducted the experiments only for *Retailer #1*.

As described in Section 7.1, we split the basket data into train and test baskets. All reported results were obtained using a 5-fold cross validation (CV) with an 80-20% split, i.e., each CV fold containing 1282 randomly selected train baskets and 320 test baskets.

Table 5 Recall values for *Retailer #1* obtained using the 5-fold cross-validation. In each column, the values marked with ** are significantly better than the other two approaches (two-proportion z-test, $p < 0.001$). The value marked with * is better than the worst-performing approach ($p < 0.001$)

Approach	*Baskets covered by the ARs* *(*181.2 *baskets per fold on average)*	*All test baskets* *(*320 *test baskets per fold)*
Product-related	0.22 ($SD = 0.029$)	0.19 ($SD = 0.018$) *
Basket-related	0.25 ($SD = 0.019$)	0.12 ($SD = 0.01$)
Hybrid	0.33 ($SD = 0.014$) **	0.24 ($SD = 0.024$) **

We note that, of the evaluated recommendation approaches, the *product-related* (and consequently, the *hybrid*) approach can generate recommendations for any set of query products. Conversely, the *basket-related* approach, which relies on ARs mined from basket data, cannot recommend any items if no subset of the query items is found among the ARs. In fact, when mining the ARs from 1282 train baskets during the cross-validation, we observed an average of 258 products covered by the ARs (out of the 2553 distinct purchased products). Therefore, to ensure a fair comparison of the approaches, we initially restricted the evaluation to test baskets for which the *basket-related* approach can recommend at least one item. This resulted in 181.2 test baskets per CV fold on average ($SD = 10.72$). Then, a second set of experiments was run on the full set of test baskets.

Table 5 shows the mean *recall* values obtained across the 5 cross-validation folds. In the first set of experiments, the basket-related approach obtains slightly higher recall than the product-related approach. While we could not confirm the statistical significance of this difference ($p = 0.16$ in a two-proportion z-test), we believe this result indicates a pattern similar to the relationship between *text-based* and *co-occurrence-based* product-seeded recommendation approaches described in Section 4.2. We can see that when available, the approach which relies on association rules (i.e., the basket-related approach) generates more reliable (or at least equally good) recommendations compared to the approach which uses text similarity.

However, due to the low catalog coverage of the association rules, the basket-related approach loses to other approaches in the experiments with the full set of test baskets. Therefore, an additional source of product relatedness is required to generate recommendations for any user basket. The results show that in each set of the experiments, the hybrid approach achieves the highest recall, indicating that the product-related and basket-related approaches successfully complement each other.

8 Conclusions and Future Work

We have proposed a product-seeded recommender that is a hybrid combination of two techniques of which the AR-based approach provides higher-quality recommendations but which, due to data sparsity (i.e., few products being purchased/viewed together), cannot provide recommendations for all products in the catalog. Therefore, a second technique – the text-based approach – is a necessary complement when generating recommendations for the full product catalog.

The obtained evaluation results lead us to believe that the proposed approach results in a more attractive recommendation panel, since the users are more likely to click on it compared to the legacy version of the panel. We also conclude that recommendation placement is essential, since users are more likely to click on recommendations if they are clearly visible on the website and less likely to click on them if scrolling is required. The results also showed that among users who engage with product recommendations, the number of completed orders and total revenue are higher compared to the legacy version of the recommender. Moreover, the proposed recommendation approach does not require manual input from the retailers compared to the legacy version of the recommendation panel in both websites.

Moreover, having validated that ARs provide a more valuable source of product relatedness information compared to the text-based similarity, we may investigate new hybrid solutions (e.g., combining manual recommendations with the AR-based approach). Another possibility is to exploit external sources of information, such as existing product taxonomies, to enrich the text descriptions of products and to improve the quality of the text-based relatedness computation.

We have also adapted the product-seeded approach to generate basket-seeded recommendations. Through an offline evaluation we demonstrated that a hybrid combination of product-related recommendations and association rules mined from basket data provides the best quality basket-seeded recommendations.

An important future work direction is investigating alternative techniques for basket-seeded recommendations, for instance adapting techniques from grocery shopping recommendations (Li et al, 2009). It is also important to consider additional metrics when comparing the basket-seeded recommendations, e.g., ones that measure the distribution of complementary/substitute items among recommendations generated by the different approaches. Once these other basket-seeded approaches have been evaluated in offline experiments, the next step would be to choose one or more of them to evaluate in an online setting, as we did for the product-seeded recommendations.

For both product-seeded and basket-seeded recommendations, we are also interested in investigating the beyond-accuracy aspects of product recommendation, such as diversity (i.e., ensuring that the items in a recommendation list are not too similar to each other), novelty (i.e., promoting recommendations of less popular items) and coverage (i.e., increasing the portion of the catalog for which recommendations are generated) (Fleder and Hosanagar, 2009) as these recommendation qualities are known to positively influence user satisfaction with recommendations (Knijnenburg et al, 2012).

Finally, user trials dedicated to recommendation perception could help understanding the effectiveness of the proposed techniques. In the current online experiments, the users were not aware that they were part of an experiment. Actively gathering their feedback about the product recommendations could help us obtain important insights.

References

Agrawal R, Srikant R, et al (1994) Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Databases (VLDB), vol 1215, pp 487–499

Carmona CJ, Ramírez-Gallego S, Torres F, Bernal E, del Jesus MJ, García S (2012) Web usage mining to improve the design of an e-commerce website: Orolivesur. com. Expert Systems with Applications 39(12):11,243–11,249

Chen J, Miller C, Dagher G (2014) Product recommendation system for small online retailers using association rules mining. In: Procs. of the International Conference on Innovative Design and Manufacturing, pp 71–77

Cho YB, Cho YH, Kim SH (2005a) Mining changes in customer buying behavior for collaborative recommendations. Expert Systems with Applications 28(2):359–369

Cho YH, Kim JK (2004) Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce. Expert systems with Applications 26(2):233–246

Cho YH, Kim JK, Ahn DH (2005b) A personalized product recommender for web retailers. In: Systems Modeling and Simulation: Theory and Applications, Springer, pp 296–305

Choi K, Yoo D, Kim G, Suh Y (2012) A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. Electronic Commerce Research and Applications 11(4):309–317

Dias MB, Locher D, et al (2008) The value of personalised recommender systems to e-business: a case study. In: Proceedings of the 2008 ACM Conference on Recommender systems, ACM, pp 291–294

Dwork C, Kumar R, Naor M, Sivakumar D (2001) Rank aggregation methods for the web. In: Proceedings of the 10th international conference on World Wide Web, ACM, pp 613–622

Fleder D, Hosanagar K (2009) Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. Management science 55(5):697–712

Gedikli F, Jannach D (2010) Neighborhood-restricted mining and weighted application of association rules for recommenders. In: Web Information Systems Engineering–WISE 2010, Springer, pp 157–165

Jannach D, Lerche L, Jugovac M (2015) Adaptation and evaluation of recommendations for short-term shopping goals. In: Proceedings of the 9th ACM Conference on Recommender Systems, ACM, pp 211–218

Knijnenburg BP, Willemsen MC, Gantner Z, Soncu H, Newell C (2012) Explaining the user experience of recommender systems. User Modeling and User-Adapted Interaction 22(4-5):441–504

Kotsiantis S, Kanellopoulos D (2006) Association rules mining: A recent overview. GESTS International Transactions on Computer Science and Engineering 32(1):71–82

Leung CWk, Chan SCf, Chung Fl (2006) A collaborative filtering framework based on fuzzy association rules and multiple-level similarity. Knowledge and Information Systems 10(3):357–381

Li M, Dias BM, et al (2009) Grocery shopping recommendations based on basket-sensitive random walk. In: Proceedings of the 15th International Conference on Knowledge Discovery and Data mining, ACM, pp 1215–1224

Liu DR, Lai CH, Lee WJ (2009) A hybrid of sequential rules and collaborative filtering for product recommendation. Information Sciences 179(20):3505–3519

Matthews SG, Gongora MA, Hopgood AA, Ahmadi S (2013) Web usage mining with evolutionary extraction of temporal fuzzy association rules. Knowledge-Based Systems 54:66–72

Sandvig JJ, Mobasher B, Burke R (2007) Robustness of collaborative recommendation based on association rule mining. In: Proceedings of the 2007 ACM Conference on Recommender Systems, ACM, pp 105–112

Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, ACM, pp 285–295

Schafer J (2009) The application of data-mining to recommender systems. Encyclopedia of Data Warehousing and Mining 1:44–48