Product Recommendation for Small-Scale Retailers

Marius Kaminskas[1], Derek Bridge[1], Franclin Foping[2], and Donogh Roche[2]

[1] Insight Centre for Data Analytics,
University College Cork, Ireland
{marius.kaminskas,derek.bridge}@insight-centre.org
[2] NitroSell Ltd.,
Cork, Ireland
{franclin.foping,donogh.roche}@nitrosell.net

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 – and demonstrate the effectiveness of the approach through two evaluation studies with real customer data.

Key words: 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 [4]. 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 [9]. 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

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

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 first two months of the evaluation period, out of 7800 products in *Retailer's #1* catalog, 2200 items were purchased, roughly 50% of them only once; for *Retailer #2*, out of 1500 products, 90 were purchased, out of them 70 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 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 recommender 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.

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 approach. Then, we describe the offline experiments conducted to validate the adopted recommendation strategy. Finally, we describe the online evaluation of the system and outline future work directions.

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 often cannot be closely tailored to the preferences of each individual user, but need to be generated in a way that would satisfy the majority of customers [6, 7]. Hence, recommendations in e-commerce are typically based on computing item-to-item similarities and using these to recommend items (products) that are similar to the ones viewed or purchased by the user [3, 8]. The core step in such approaches is reliably computing item similarity, which is often alleviated by employing data mining techniques, such as product clustering and association rule mining [10].

Cho et al. [3] used the shopping behaviour of online customers as a source of item similarity information. The authors distinguished three levels of user's involvement with an item — an item view, a basket placement, and an item purchase. Product association rules were mined for each source of information separately and then combined into a single item similarity score by giving most importance to item co-occurrence among purchases and least importance to item co-occurrence among viewed items. The authors also employed a product taxonomy to address the data sparsity problem — grouping products into categories (e.g., types of apparel) before association rule mining.

Li et al. [8] proposed modeling 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.

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. Chen et al. [2] addressed this problem by 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.

In our work, we also employ association rule (AR) mining, however we address the data sparsity problem by combining ARs with text-based item similarity. Moreover, similarly to Cho et al. [3], to cope with the limited amount of purchase data, we use product views as a source for AR mining.

3 The Approach

We observe that retail websites typically organize the product data into categories 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 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 stopword 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:

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

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).

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

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 [1].

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

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

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

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-occured 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:

$$recall = \frac{|\{p \in P \,:\, (Rel_p \cap Top_p) \neq \emptyset\}|}{|P|} \quad prec. = \frac{\sum_{p \in P} |\{i \in Top_p \,:\, i \in Rel_p\}|}{N \cdot |P|}$$

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 (in terms of page views) products 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 outperform the popularity baseline and the hybrid combination of AR and text-based techniques to outperform other methods. (Results for *Retailer #2* were analogous and are therefore omitted).

Table 1. Recall (and precision) values for *Retailer #1*.

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**)

The differences in Table 1 between the pure text-based approach and the hybrid combination of the text-based and AR-based approaches are small. This is because the AR-based approach is applicable to only 6% of *Retailer's #1* product catalog, and so its usefulness is 'lost' in the averaging of the recall 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).

Table 2. Recall (and precision) values for products covered by ARs (*Retailer #1*).

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)

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

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.

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} and August 25^{th} on *Retailer's #1* website, and between March 30^{th} and August 25^{th} on *Retailer's #2* 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's #2* 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).

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

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.

6 Conclusions and Future Work

We have proposed a 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 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.

An important future work direction is investigating alternative placement of product recommendations. In addition to displaying the recommendation panel on the product description page, recommendations could be made on the checkout page. This alternative placement poses interesting research questions: Are the same techniques applicable to recommendations when browsing and purchasing products? Should we take into account the active user's basket contents when generating recommendations?

Moreover, 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 are also interested in exploiting recommendation techniques for increasing sales diversity [5], as the current data suggests a power law distribution of product popularity for both retailers.

Finally, a user trial dedicated to recommendation perception could help understanding the effectiveness of the proposed techniques. In the current 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.

Acknowledgements

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

References

1. Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Databases (VLDB)*, volume 1215, pages 487–499, 1994.
2. Junnan Chen, Courtney Miller, and Gaby Dagher. Product recommendation system for small online retailers using association rules mining. In *Procs. of the International Conference on Innovative Design and Manufacturing*, pages 71–77, 2014.
3. Yoon Ho Cho, Jae Kyeong Kim, and Do Hyun Ahn. A personalized product recommender for web retailers. In *Systems Modeling and Simulation: Theory and Applications*, pages 296–305. Springer, 2005.
4. M Benjamin Dias, Dominique Locher, et al. The value of personalised recommender systems to e-business: a case study. In *Proceedings of the 2008 ACM Conference on Recommender systems*, pages 291–294. ACM, 2008.

5. Daniel Fleder and Kartik Hosanagar. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management science*, 55(5):697–712, 2009.
6. Michael Giering. Retail sales prediction and item recommendations using customer demographics at store level. *ACM SIGKDD Explorations Newsletter*, 10(2):84–89, 2008.
7. Jongwuk Lee, Seung-won Hwang, Zaiqing Nie, and J-R Wen. Navigation system for product search. In *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*, pages 1113–1116. IEEE, 2010.
8. Ming Li, Benjamin M Dias, et al. Grocery shopping recommendations based on basket-sensitive random walk. In *Proceedings of the 15th International Conference on Knowledge Discovery and Data mining*, pages 1215–1224. ACM, 2009.
9. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295. ACM, 2001.
10. J Schafer. The application of data-mining to recommender systems. *Encyclopedia of Data Warehousing and Mining*, 1:44–48, 2009.