Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2018) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

International Workshop on Web Search and Data Mining (WSDM 2019)
April 29 - May 2, 2019, Leuven, Belgium

E-commerce Personalization with Elasticsearch

Konstantinos N. Vavliakis[a,b,*], George Katsikopoulos[a], Andreas L. Symeonidis[a]

[a] *Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, GR54124, Greece*
[b] *Pharm24.gr, Dafni Lakonias, GR23057, Greece*

Abstract

Personalization techniques are constantly gaining traction among e-commerce retailers, since major advancements have been made at research level and the benefits are clear and pertinent. However, effectively applying personalization in real life is a challenging task, since the proper mixture of technology, data and content is complex and differs between organizations. In fact, personalization applications such as personalized search remain largely unfulfilled, especially by small and medium sized retailers, due to time and space limitations. In this paper we propose a novel approach for near real-time personalized e-commerce search that provides improved personalized results within the limited accepted time frames required for online browsing. We propose combining features such as product popularity, user interests, and query-product relevance with collaborative filtering, and implement our solution in Elasticsearch in order to achieve acceptable execution timings. We evaluate our approach against a publicly available dataset, as well as a running e-commerce store.

Keywords: e-commerce; personalization; search; elasticsearch

1. Introduction

Over the last twenty years, e-commerce has grown swiftly in prominence and e-commerce applications are a constantly increasing segment of the retail industry. A crucial factor for the success of every e-commerce store is user experience (UX); one may summarize the e-commerce UX objective as: “to meet the exact needs of the customer, without fuss or bother”. Obviously, this cannot be successfully achieved without effectively practicing personalization on the data. Additionally, most e-commerce experts agree that a simple, easy-to-use search tool is critical for a successful e-commerce site. After all, if people can’t find your products, they can’t buy them. Search engines on e-commerce sites allow users to change the order in which the search results are presented with pre-defined sort criteria. Most online shops do not offer an option for users to implicitly or explicitly sort the results according to their

[*] Corresponding author. Tel.: 30-231-099-6349 ; fax: +30-231-099-6398.
E-mail address: kvavliak@issel.ee.auth.gr

past personal preferences or shopping behavior [6]. It is, however, intuitive to assume that taking the consumer's past behavior into account when ranking the results can be useful, e.g., by ranking items up in the list that corresponds to the consumer's typical price preferences, by listing products of the consumer's preferred brands first, or by promoting the consumables that the user has already purchased in the past. In any case, although personalization is becoming state for several web companies, it is rather challenging to effectively apply it, especially in small and medium sized organizations. That is why, while it's always been a focus of ecommerce strategizing, the promise of a personalized online shopping experience, including personalized search, remains largely unfulfilled at a commercial level, as even today it is still unclear whether personalization is consistently used in e-commerce sites, especially when looking beyond e-commerce giants such as Amazon, Ebay and Alibaba.

On the other hand, personalized search has been the focus of research communities for many years and many approaches have been proposed in academic studies. Numerous machine learning techniques have been suggested, such as deep neural networks, SVMs and decision trees, as well as a variety of statistical methods, from descriptive statistics to tf-idf, and other linguistic tools like ontologies. Nevertheless, the common ground of all these studies is that, despite some of them achieve improved search results, they do not take into account time limitations that require near real-time execution or scalability issues that are a prerequisite for applications in commercially running web systems.

In this paper we design and evaluate a personalization strategy that takes into account past user behavior, product characteristics, as well as query-product-customer relation and focuses on filtering (searching for products of a particular brand or category), as well as free text search. Our main goal is to improve search in real e-commerce environments, while at the same time ensuring that queries are executed in a timely fashion, as delays are considered a conversion killer in e-commerce environments.

The remainder of this paper is structured as follows. Related work on personalization and search systems is discussed in Section 2. Section 3 describes in detail the proposed framework, while Section 4 evaluates our approach. Finally, Section 5 summarizes work done, discusses future directions and concludes the paper.

2. Related Work

Personalized search has been attracting increased attention during the last years. It was more than a decade ago when Dou et al. [2] used MSN query logs for revealing that personalized search may have significant improvement over common search and that straightforward click-based personalization strategies perform consistently and considerably well, while profile-based ones may be unstable. In a more recent advancement, Jannach et al. [6] after experimentation with various personalization techniques, concluded that considering within the recommendation process several item relevance signals such as users' general interests, their most recent browsing behavior, and current sales trends, leads to the best ranking of search results.

On the other hand two distinguished search patterns, typical and atypical search behavior are discussed by Eickhoff et al in [3]. They investigate users straying from their search profiles to satisfy information needs outside their regular areas of interest, a behavior they call atypical search. User intent was also investigated by Teevan et al. [16], where authors examined its variability using both explicit relevance judgments and large-scale log analysis of user behavior patterns. Thorsten [7] used click-through data for automatically optimizing the retrieval quality of search engines in combination with Support Vector Machine (SVM) and presented a method for learning retrieval functions that can effectively adapt the retrieval function of a meta-search engine to a particular group of users. On a different approach, Speretta and Gauch [13] explored the use of less-invasive means of gathering user information for personalized search, as they build user profiles based on activity at the search site itself. In their study, user profiles were created by classifying the information into concepts from the Open Directory Project concept hierarchy and then used to re-rank the search results by calculating the conceptual similarity between each document and the user's interests.

Recently text mining techniques have also been proposed. Yu and Mohan [18] used LDA models to discover the hidden user intents of a query, according to their conclusions, LDA-based approach provides significantly higher user satisfaction than other popular approaches. Learning to rank algorithms (LTR), such as SVMRank, RankLib, RankNet, and XGboost have all been used for improving search engine results [11], as well as BM25F [12], another popular ranking function in information retrieval, commonly used in search engines. Nevertheless, LTR requires building a judgment list, a tedious and resourceful process. In addition extensive training and evaluation is required, which requires substantial computation power, thus frequent or sudden changes in data and/or user behavior may lead to decreased performance. Ensemble methods have been proposed as well. Wu, Yan and Si [8] use different types of features, i.e., statistic features, query-item features and session features propose a stacking ensemble model that consists of different models, such as logistic regression, gradient boosted decision trees, rank SVM and a deep match model. On the other hand, Lie et al. [9] presented a cascade model in a large-scale operational ecommerce search application. Their approach modeled multiple factors of user experience and computational cost and addressed multiple types of user behavior in e-commerce search that provided a good trade-off between search effectiveness and search efficiency within operational environments in regular e-commerce environment.

Performance plays a major role in the success of any online venture. A faster web site means a better visitor experience, on the contrary a slow website will lead to a poor user experience. In this context Elasticsearch [4] was built, which is a search engine based on the Lucene library [10]. It provides an open-source, distributed, multitenant-capable full-text search engine that can be used to search all kinds of documents. It provides scalable search, has near real-time search, and supports multitenancy. As a result, it has been widely adopted by the industry leaders (e.g., Ebay, VW, IEEE and Netflix) to successfully help customers timely discover resources based on textual queries and is considered to be the most popular enterprise search engine.

SMEs have many advantages over larger organizations, one of which is the ability to change course and respond to challenges and opportunities quickly. But being able to change isn't enough, if new technologies are not affordable and applicable in real use cases. For this reason, although it is obvious that a lot of progress has been made in personalization and search systems, there is still a great need for integrated solutions that are affordable in terms of human resources and processing requirements. These solutions should, on the one hand, deliver personalized search results that improve UX and, on the other hand, be flexible enough to quickly adapt to new trends and sporadic changes in user behavior, as well as be scalable and resource efficient in terms of processing power and memory consumption.

3. Proposed Framework

Our approach takes into consideration a threefold set of features, elicited from a) products, b) users and c) queries and is augmented with information from collaborative filtering as well. A high level architectural diagram of our approach is depicted in Figure 1. All data are integrated in json files, imported in Elasticsearch. For each product i, we calculate $popularity_i$ as in Equation 1, where $buys_i$, $clicks_i$, $views_i$ are the number of buys, clicks and views for product i and $|buys|$, $|clicks|$, $|views|$ are the total number of buys, clicks and views respectively. Popularity score is usually affected more by buys, then by clicks and finally by views, thus the use of w_b, w_c and w_v.

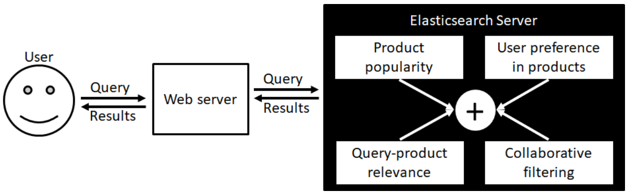

Fig. 1. High level architectural diagram of the proposed approach.

$$popularity_i = w_b \frac{buys_i}{|buys|} + w_c \frac{clicks_i}{|clicks|} + w_v \frac{views_i}{|views|} \tag{1}$$

Another factor that we take into account is the user-product relation, meaning the views, clicks and buys of each user for each product. In this case we take into account time, as recent interactions naturally are more important than historic ones. So, in case that user history is available, the user-product relevance is calculated by Equation 2, where $buys_{d,u,i}$, $clicks_{d,u,i}$, $views_{d,u,i}$ are the number of buys, clicks and views of user u for product i at day d, x is the difference in days between day d and day of search, and $|buys_u|$, $|clicks_u|$, $|views_u|$ are the total number of buys, clicks and views of user i respectively.

$$relevance_{u,i} = w_b \frac{\sum \left(buys_{d,u,i} * \left(1 + \frac{1}{1-e^{-x}}\right)\right)}{|buys_u|} + w_c \frac{\sum \left(clicks_{d,u,i} * \left(1 + \frac{1}{1-e^{-x}}\right)\right)}{|clicks_u|} + w_v \frac{\sum \left(views_{d,u,i} * \left(1 + \frac{1}{1-e^{-x}}\right)\right)}{|views_u|} \tag{2}$$

In addition, we take into account the query-product relevance, meaning the similarity between the query tokens q and the product textual description d (usually the product name) tokens as described by the Elasticsearch score function (Equation 3), where *queryNorm* is a measure for comparing queries when you are using a combination of query types, *coord* is a measurement of matching on multiple search terms, where a higher value of this measurement will increase the overall score, tf is a measure of the number of occurrences of a t in d, idf is a measurement of how frequently the search terms occur across a set of documents, and *norm* measures smaller field matches [4].

$$queryScore(q,d) = queryNorm(q) * coord(q,d) * \sum tf(t\ in\ d) * idf(t)^2 * norm(t,d))(t\ in\ q) \tag{3}$$

Finally, complementary to the statistical information described so far, we propose combining a collaborative filtering (CF) approach that we have integrated with Elasticsearch as well. As direct input from users regarding their preferences (e.g. likes, reviews, wish lists) is not available, we lack substantial evidence on which products consumers dislike. Thus, we deal only with implicit information (e.g. clicks, views, buys), and we treat positive and negative preference associated with vastly varying confidence levels. This leads to a factor model that is especially tailored for implicit feedback recommenders [5]. For improving the performance of the implicit feedback collaborative filtering model we followed the conjugate gradient approach proposed by Takács et al. [14]. Our data lie in a matrix **R** of $u \times i$ dimensions, where u is the number of users and i is the number of products. We apply matrix factorization by using the alternating least square (ALS) method [15] in order to create two new matrices: a) matrix **U** with dimensions $u \times f$ and b) matrix **V** with dimensions $f \times i$, such as $\mathbf{R} \approx \mathbf{U} \times \mathbf{V}$. The confidence level c of user u for product i is defined as $c_{ui} = 1 + ar_{ui}$ where r_{ui} is the binary representation of the interaction between user u and product i, with a being the weighting factor for interactions. Finally, we calculate the preference of each user for every product and we end up with a score for each user-product combination: $score = U_i....V^T$ where U_i is a table consisting of user i and latent factors and V^T the product-latent factors table. Combining all the above mentioned signals, ranking depends on the weighted sums of product popularity, user past behavior, query-product similarity and the collaborative filtering recommendation, according to Equation 4, where $score_i$ is a binary variable denoting whether or not a product is recommended by the CF algorithm.

$$recommendationScore_{q,u,i} = w_p * popularity_i + w_r * relevance_{u,i} + w_q * queryScore(q,d) + w_s * score_i \tag{4}$$

4. Experimental Evaluation

We evaluated our approach against the dataset provided by Diginetica [1] for the ”CIKM Cup 2016 Track 2: Personalized E-Commerce Search Challenge” as described in Table 1. The data is divided into two groups: 1) ”query-less” data, that is search engine result pages in response to the user click on some product category; and 2) ”query-full” interactions of search engine result pages returned in response to a query. The *nDCG* metric (normalized discounted cumulative gain) was employed for evaluation. *nDCG* measures ranking quality and is often used to measure effectiveness of web search engine algorithms or related applications [17]. *nDCG* is calculated using the ranking of products provided by Diginetica for each query, and then averaged over all test queries. There are three grades for relevance: 0 means irrelevant, i.e. products with no clicks; 1 stands for somewhat relevant and corresponds to the products, that were clicked by the user, and 2 is relevant, meaning products that were clicked and purchased by the user. In Equation 5, p stands for the positions up to which we calculate *nDCG*, $rating(i)$ is the score for position i and $|REL|$ the best score for p. Since we evaluate both types of queries, query-full and query-less we followed the same evaluation procedure as CIKM: the final *nDCG* value is a weighted sum of the query-full $nDCG_{qf}$ and query-less $nDCG_{ql}$ as: $nDCG = 0.2 * nDCG_{qf} + 0.8 * nDCG_{ql}$.

$$nDCG_p = \sum_{i=1}^{p} \frac{rating(i)}{log_2(i+1)} \bigg/ \sum_{i=1}^{|REL|} \frac{rating(i)}{log_2(i+1)} \tag{5}$$

The evaluation results are available in Table 2. First, we randomly ranked the results, calculated the *nDCG* values and used them as our baseline. Consequently we experimented only with the collaborative filtering algorithm to test different values of the weighting factors for interaction a. In [5] the optimal value for a was 40, so we tested for

Table 1. Dataset from CIKM Cup 2016 Track 2: Personalized E-Commerce Search Challenge.

Description	Number	Description	Number
Sessions	573,935	Searches	923,127
Clicks	1.127,764	Query-full queries	53,427
Buys	18,025	Query-less queries	849,700
Products	184,047	Registered users	232,817
Categories	1,217	Anonymous users	333,097
Views	1,235,380		

$a = 15, 30$ and 40. According to our experiments $a = 40$ achieved the best results for the query-less case, while the best result for query-less came with $a = 30$, thus it makes sense to use different a values depending on the query type. Thereafter we tested different values for the weighting factors w_r, w_p, w_q and w_s (Table 2), according to our experiments, values $w_r = 1$, $w_p = 1.5$, $w_q = 1.5$ and $w_s = 0$ gave the best results improving *nDCG* up to +42.42% when compared with the baseline. In all our experiments for calculating the $popularity_i$ we used the weights $w_b = 5$, $w_c = 3$ and $w_v = 1$, as naturally buys are more important that clicks which are more important than views.

Table 2. Evaluation Results.

Description	*nDCG*	*nDCGImpr.*	$nDCG_{ql}$	$nDCG_{ql}$ Impr.	$nDCG_{qf}$	$nDCG_{qf}Impr.$
Baseline	0.242856	-	0.220042	-	0.334116	-
$w_r = 0, w_p = 0, w_q = 0, w_s = 1, a = 15$	0.325426	34.00%	0.313623	42.53%	0.372638	11.53
$w_r = 0, w_p = 0, w_q = 0, w_s = 1, a = 30$	0.325513	34.04%	0.313675	42.55%	0.372863	11.60%
$w_r = 0, w_p = 0, w_q = 0, w_s = 1, a = 40$	0.325544	34.05%	0.372923	69.48%	0.313699	-6.11%
$w_r = 1.5, w_p = 3, w_q = 3, w_s = 0.75$	0.343913	41.61%	0.336081	52.73%	0.375240	12.31%
$w_r = 3, w_p = 1.5, w_q = 1.5, w_s = 0.75$	0.343913	41.61%	0.336124	52.75%	0.388671	16.33%
$w_r = 1, w_p = 1.5, w_q = 1.5, w_s = 1$	0.344620	41.90%	0.335158	52.32%	0.382457	14.47%
$w_r = 1.5, w_p = 1, w_q = 1, w_s = 0$	0.346330	41.61%	0.335746	52.58%	0.388671	16.33%
$w_r = 1, w_p = 1, w_q = 1, w_s = 0$	0.345870	42.42%	0.337566	53.41%	0.379080	13.46%
$w_r = 1, w_p = 1.5, w_q = 1.5, w_s = 0$	0.344670	41.92%	0.337002	53.15%	0.375346	12.34%

Finally, we tested the proposed solution live in a real e-commerce store, Pharm24.gr which is a well-known online pharmacy in Greece with a few hundred thousand visitors per month. Pharm24.gr, just like many more small-medium e-commerce retailers, although considerably smaller than the global e-commerce giants, has enough traffic so a small increase in conversion rate, led by improved user experience and generated by ameliorating the search engine can lead to a substantial increase in revenue. For this reason we integrated the proposed solution in the existing Elasticsearch engine and run an A/B experiment, comparing our solution with the default search behavior that was sort "by popularity". According to our experiment the click-through rate increased by 4.7%, while conversion rate increased on average by 1.62%. We also performed a stress test analysis with 5000 clients querying our server within 1 minute duration using a popular online stress test tool (https://loader.io/). In this test (Figure 2) the total response time (including server connection time, ssl handshake and content download) was $224ms$, while the average net query time was $16ms$. Although these are preliminary results, they are a good indication that the proposed system can be applied in real-life scenarios, achieving improved results and meeting the strict performance restrictions of e-commerce stores.

5. Conclusion

In this paper we presented a novel approach for near real-time personalized e-commerce search, suitable for commercial environments. Our goal was not only to provide improved personalized results, but to do so within the limited accepted time frames required for online browsing. We proposed a novel system combining different features, such as product popularity, user interests, query-product relevance, augmented by a collaborative filtering technique. We integrated our solution in Elasticsearch and experimented in a publicly available dataset, as well as in an e-commerce

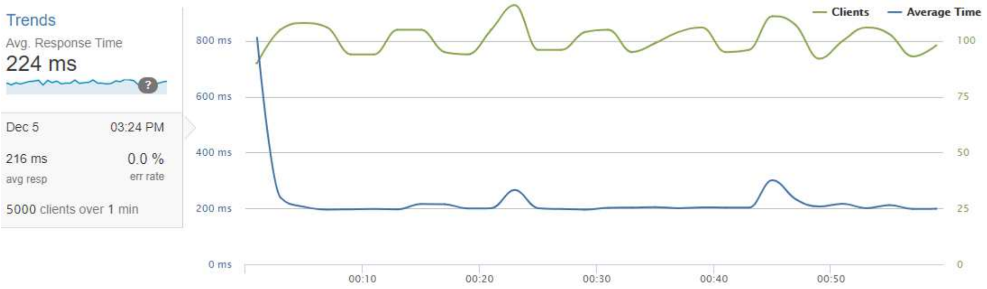

Fig. 2. Stress test results

store through A/B testing. Our experiments showed that the proposed system, not only achieved improved results, but also is suitable for near real-time search in commercial environments. Future work includes further experimentation on the optimal weighting factors, as well as improving the integration of our method with Elasticsearch.

Acknowledgements

This work was partially funded by an IKY scholarship funded by the "Strengthening of Post-Academic Researchers" Act from the resources of the OP "Human Resources Development, Education and Lifelong Learning" with Priority Axes 6,8,9 and co-funded by the European Social Fund ECB and the Greek government. The authors would also like to thank Kostas Nikolaros of Pharm24.gr for his valuable feedback regarding user search behavior.

References

[1] CIKM Cup organizing committee, 2016. Cikm cup 2016 track 2: Personalized e-commerce search challenge. URL: https://competitions.codalab.org/competitions/11161#learnthedetails-data2. [Online; accessed 5-December-2018].

[2] Dou, Z., Song, R., Wen, J.R., 2007. A large-scale evaluation and analysis of personalized search strategies, in: WWW '07: Proceedings of the 16th international conference on World Wide Web, ACM Press. pp. 581–590.

[3] Eickhoff, C., Collins-Thompson, K., Bennett, P.N., Dumais, S., 2013. Personalizing atypical web search sessions, in: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, ACM, New York, NY, USA. pp. 285–294.

[4] Gormley, C., Tong, Z., 2015. Elasticsearch: The Definitive Guide. 1st ed., O'Reilly Media, Inc.

[5] Hu, Y., Koren, Y., Volinsky, C., 2008. Collaborative filtering for implicit feedback datasets, in: 2008 Eighth IEEE International Conference on Data Mining, pp. 263–272.

[6] Jannach, D., Ludewig, M., 2017. Investigating personalized search in e-commerce, in: FLAIRS Conference, AAAI Press. pp. 645–650.

[7] Joachims, T., 2002. Optimizing search engines using clickthrough data, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA. pp. 133–142.

[8] Kong, D., 2016. Personalized Feature based Re-Ranking Method for Ecommerce Search at CIKM Cup 2016. Technical Report. CIKM Cup.

[9] Liu, S., Xiao, F., Ou, W., Si, L., 2017. Cascade ranking for operational e-commerce search, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA. pp. 1557–1565.

[10] McCandless, M., Hatcher, E., Gospodnetic, O., 2010. Lucene in Action, Second Edition: Covers Apache Lucene 3.0. Manning Publications Co., Greenwich, CT, USA.

[11] Palotti, J., 2016. Learning to Rank for Personalized E-Commerce Search at CIKM Cup 2016. Technical Report. CIKM Cup.

[12] Robertson, S., Zaragoza, H., 2009. The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr. 3, 333–389.

[13] Speretta, M., Gauch, S., 2005. Personalized search based on user search histories, in: The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), pp. 622–628.

[14] Takács, G., Pilászy, I., Tikk, D., 2011. Applications of the conjugate gradient method for implicit feedback collaborative filtering, in: Proceedings of the Fifth ACM Conference on Recommender Systems, ACM, New York, NY, USA. pp. 297–300.

[15] Takács, G., Tikk, D., 2012. Alternating least squares for personalized ranking, in: Proceedings of the Sixth ACM Conference on Recommender Systems, ACM, New York, NY, USA. pp. 83–90.

[16] Teevan, J., Dumais, S.T., Liebling, D.J., 2008. To personalize or not to personalize: Modeling queries with variation in user intent, in: Proc. of the 31st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, ACM, New York, NY, USA. pp. 163–170.

[17] Wang, Y., Wang, L., Li, Y., He, D., Liu, T.Y., Chen, W., 2013. A theoretical analysis of ndcg type ranking measures. Journal of Machine Learning Research 30.

[18] Yu, J., Mohan, S., Putthividhya, D.P., Wong, W.K., 2014. Latent dirichlet allocation based diversified retrieval for e-commerce search, in: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, ACM, New York, NY, USA. pp. 463–472.