Personalization and the Conversational Web

Konstantinos N. Vavliakis[1,2](✉), Maria Th. Kotouza[1], Andreas L. Symeonidis[1], and Pericles A. Mitkas[1]

[1] Department of Electrical and Computer Engineering,
Aristotle University of Thessaloniki, GR54124 Thessaloniki, Greece
{kvavliak,maria.kotouza,asymeon}@issel.ee.auth.gr, mitkas@auth.gr

[2] Pharm24.gr, GR23057 Dafni, Lakonias, Greece
https://issel.ee.auth.gr/
http://www.pharm24.gr

Abstract. Hyper-personalization intends to maximize the opportunities a marketer has to tailor content that fits each and every customer's wants and needs. Naturally, gathering and analyzing more data is the key to those opportunities. This is were the "*Conversation Web*" comes in, which in the near future is expected to transform to so much more than just conversational interfaces (chat-bots). In a truly Conversation Web, websites and users implicitly "*discuss*" in the form of clicks, mouse scrolls and movements, as well as page views and product purchases. Websites use this information for decoding user interests and profile and provide customized one-to-one services. In this work we proposed an integrated architecture for the conversational Web; consequently we propose a novel hybrid approach for recommendations using offline and online analysis, as well as we propose a novel personalized search strategy that takes into account the strict time performance limitations applied in e-commerce. We evaluate the proposed methods on three different datasets and we show that our personalized search approach provides considerably improvements in search results while being suitable for near real-time search in commercial environments. Regarding personalized recommendations, the proposed approach outperforms current state-of-art methods in small-medium datasets and improves performance in large datasets when combined with other methods.

Keywords: Personalization · Recommendation · Search · Elasticsearch · Conversational web · e-Commerce · RFM · Recurrent neural networks

1 Introduction

Over the last twenty years, e-Commerce has grown at an unprecedented rate all over the world and e-commerce applications have become a constantly increasing segment of the retail industry. The future of e-commerce belongs to brands who

M. J. Escalona et al. (Eds.): WEBIST 2018, LNBIP 372, pp. 56–77, 2019.
https://doi.org/10.1007/978-3-030-35330-8_4

create unique experiences that capture attention and keep customers coming back. To achieve this, companies use personalization for providing uniquely customized experiences, as talking to customers in a customized way is much more efficient than using general, uniform mass messages. Personalization techniques became increasingly popular in recent years and are considered key elements in a variety of areas, not only in e-commerce, but in movies, music, news, research articles, search queries and social tags as well. Personalization is broadly used for improving customer satisfaction, sales conversion and marketing results. A website that is not personalized usually shows exactly the same content to each visitor, irrespective of the visitors' profile, interests, preferences or behavior. As a result, only a small percentage of visitors receive an optimal user experience with this type of site. On the other hand, with personalized websites, visitors get different messages as there is no webpage duplication, and each visitor segment experiences different content that exactly fits their interest and needs. Personalization targeted to segments of users requires specific steps in order to launch an effective strategy, such as to identify audience, understand visitors, plan and create different experience for each audience.

Computing power and the use of big data has increased exponentially over the last few years, and improvements in AI-powered systems have made real-time personalized services possible. Towards this end, *"hyper-personalization"* takes personalized marketing a step further by leveraging artificial intelligence (AI) and real-time data to deliver more relevant content, product, and service information to each user on a one-to-one basis. Hyper-personalization is more involved, more complex, and more effective than personalization. It goes beyond customer data to rethinking customer interaction on a one-to-one basis, where we treat each and every customer uniquely and design a customized experience for each one. The key element for hyper-personalization is interacting one-to-one with individuals, not the customer segments they fall in. To anticipate an individual's desires at any point in time, however, requires having deep customer insight, which comes from analyzing granular and big data. Hyper-personalization identifies subtle nuances and details that profiling doesn't catch, in order to provide highly targeted and personalized products, services, promotions and content. To make this happen, it requires the ability to merge customer interactions with demographic and historical data to paint a clear, contextual picture. This leads to the next era of digital marketing; emails that change content based on where a customer is and when the email is opened. Context-aware messages and segments that are build for more relevant communications with customers, pushing only those messages they should like to receive. Except of added value, there are numerous reasons why hyper-personalization has not yet been adopted by the majority of websites, as it requires significant processing power, technical and academic expertise, as well as propose use of actionable data.

The *"Conversational web"* or conversational interfaces, also known as chatbots, is a hybrid user interface (UI) that interacts with users combining chat, voice or any other natural language interface with graphical UI elements like buttons, images, menus, videos, etc. It has recently started to be used in the context chat-bots or virtual assistants, as well as in the context of web services. On the other hand, conversational web services (CWS) refer to web services that communicate multiple times with a client to complete a single task. Conversational interfaces have emerged as a tool for businesses to efficiently provide consumers with relevant information, in a cost effective manner, as they provide ease of access to relevant, contextual information.

Next, we redefine the term Conversational Web in the context of hyper-personalization [38]. Conversational Web refers to dynamic, multiple and asynchronous interactions (implicit conversations) between users and websites. These conversations allow both sides to understand each other and communicate efficiently. We argue that only in a truly conversational system hyper-personalization is possible, as interacting one-to-one with individuals, requires listening the needs and wills of each and every individual. This is only possible within a conversational web where websites and users continuously *"discuss"* (interact). The discussion takes place in the form of clicks, mouse movement, scrolling, purchases, back or forward movements and time of each page on behalf of customers. On the other hand, websites *"hear"* customers *"talking"* and respond in the form of relevant messaging and offers that best address customer needs. Users in turn react to these responses and a new cycle of communication begins.

Hyper-personalization requires processing an over abundant of data for each individual, thus big data analysis is necessary. On the one hand, real time (online) analysis is required for dynamic adapting to each customer's needs, on the other hand offline analysis is necessary as most algorithms are both time and resource consuming tasks, thus hybrid approaches, combining both online and offline analysis are most appropriate in the new era of hyper-personalized web. In any case, although personalization is becoming more than necessary 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 e-commerce strategizing, the promise of a personalized online shopping experience, including personalized recommendations and 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, as more than half of online marketers are not sure how to implement online personalization [22].

User experience (UX) is another crucial factor for the success of every e-commerce store. UX is connected with usability which refers to how usable and easy to use a website is. Friendly UX cannot be successfully achieved without effectively practising personalization on the search results through actionable data collected from a Conversational Web. 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 extend our previous work in providing recommendations at a conversational web [38], by extending the application of the conversational web from recommendations to personalization in general, proposing another field of application, namely personalized search. We present an integrated architecture for conversational websites and we claim that hyper-personalization is only possible in a conversational web that adapts to various user profiles feeding them with varying context. Conversational technologies can be applied to all kinds of websites, from the smallest to the biggest ones, thus there cannot be a unique fit-to-all solution, but numerous complementary personalization algorithms and techniques. We exhibit our modular architecture through two different hyper-personalized applications. In the context of the first application we present $PRCW$ (Product Recommendations for Conversation Web), a novel hybrid approach combing both offline and online recommendations using $RFMG$ (Recency-Frequency-Monetary-Gender), an extension of the popular RFM method [10]. Through $PRCW$ partial matching recommendations are combined with deep neural networks that provide improved results. In the context of the second application we present a personalization strategy that takes into account past user actions, product data, as well as the relations among queries, products and customers. We aim in improving 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. In both cases we evaluate the proposed methods on publicly available datasets, as well as in a working e-commerce site.

The remainder of this paper is organized as follows: Sect. 2 introduces related work on personalization as well as recommender and search systems. Section 3 presents in detail our framework for the Conversational Web. Section 4 introduces two novel and modular approaches for personalization, the first is discussed in Sect. 4.1 where a new approach for personalized recommendations is presented and the second one in Sect. 4.2 where our methodology for personalized search results in e-commerce is discussed. Both our approaches are evaluated in Sect. 5. Section 6 discusses the challenges and prominent open research issues and finally concludes the paper.

2 Related Work

The process of creating customized experiences for visitors to a website is the main function of web personalization. Personalization encompass several interdisciplinary techniques, with recommender systems being one of the most popular ones. Recommender systems are divided into online and offline systems. Offline recommendation systems [24] either content-based [29] or collaborative filtering based [34], they both have weaknesses. Offline line recommenders require significant training time; data updates usually require retraining the whole model and cannot take into account frequent changes in interests and profile of users.

An emerging approach in offline recommendation systems is session-based recommendation, which although was until recently a relatively unappreciated problem, in the last few years it has attracted increased interest [18]. This is because the behavior of users shows session-based traits, or users often have only one session. Recommendation systems widely use factor models [24] or neighborhood methods [34]. Factor models are hard to apply in session-based recommendation due to the absence of user profiles, while neighbourhood models, such as item to item similarity, ignore the information of the past clicks.

On the other, hand online recommender methods [41] need less processing power and do not require training, but they are less accurate than offline methods. As a result hybrid approaches [7] have been proposed that combine the advantages of online and offline recommendation methods. Preference elicitation is also a popular personalization technique. In the context of preference elicitation, questionnaires, reviewing pre-selected items, dynamic learning [32], entropy optimization [33] and latent factor models [19] have been employed. Nevertheless, preference elicitation is not always efficient and it is recommended only in specific problems [44]. Interactive systems are another popular group of methods relative to our case. In interactive systems users play an active role, they are usually based on reviews [8], constrains [14], and questionnaires [26]. A common method used in interactive systems, is when users are asked to review a predefined selection of items, in order to cope with the cold-start problem. These requirements may frustrate users.

Deep learning models, such as recurrent neural networks, have shown remarkable results [30] as they allow sequential data modeling fitting exactly to session-based date. Embedding deep learning techniques into recommender systems is gaining traction due to its state-of-the-art performance and high-quality recommendations that provide a better understanding of user's demands, item's characteristics, historical interactions and relationships between them than traditional methods do [43]. Especially recursive neural networks (RNNs) [16] model variable-length sequential data that scale to much longer sequences than other neural networks. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. In the last few years, there has been incredible success applying RNNs to a variety of problems: speech recognition, language modeling, translation, image captioning and session-based recommendations. To deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs, long short-term memory (LSTM) units were developed, as well as GRU (Gated Recurrent Unit), a variation on the LSTM.

Another interesting field that attracts increased attention during the last years is personalized search which refers to search experiences that are tailored specifically to an individual's interests by incorporating information about the individual beyond specific query provided. Several item relevance signals such as users' general interests, their most recent browsing behavior, and current sales trends, lead to improved rankings of search results [20]. Recent behavior is also a strong indicator; Bennett et al. in [3] assessed that not only short-term behavior contributes the majority of gains in an extended search session, but also long-term behavior provides substantial benefits, especially at the start of a search session and that each of them can be used in isolation or in combination to optimally contribute to gains in relevance through search personalization.

On a different approach, Teevan et al. [37] investigated user intent, with the help of authors that examined its variability using both explicit relevance judgments and large-scale log analysis of user behavior patterns. Speretta and Gauch [35] explored the use of less-invasive means of gathering user information for personalized search, they built user profiles based on activity at the search site itself. According to 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. Click-through data were used by Thorsten [21] for automatically optimizing the retrieval quality of search engines in combination with Support Vector Machine (SVM). Thorsten 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. Alternatives for incorporating feedback into the ranking process was also proposed [1], comparing user feedback with other common web search features showed that incorporating user behavior data can significantly improve ordering of top results in real web search setting.

Text mining techniques have also been proposed for personalized search. The use of LDA [5] models was proposed by Yu and Mohan [42] for discovering hidden user intents of a query, and then rank the user intents by making trade-offs between their relevance and information novelty. Based on Yu and Mohan's conclusions, the LDA model discovers meaningful user intents and the LDA-based approach provides significantly higher user satisfaction than other popular approaches.

Learning to Rank (LTR) [6] is a class of techniques that apply supervised machine learning to solve ranking problems. LTR solves a ranking problem on a list of items. The aim of LTR is to come up with optimal ordering of those items. As such, LTR doesn't care much about the exact score that each item gets, but cares more about the relative ordering among all the items. Several LTR algorithms, such as SVMRank, RankLib, RankNet, [6], XGboost [9] and BM25F [31], have been used for improving search engine results [28]. In any case, for a LTR algorithm to work, it is required building a judgment list which is a tedious and resourceful process. Moreover, extensive training and evaluation is required that need substantial computation power, thus frequent or sudden changes in data and/or user behavior may lead to decreased performance.

Using multiple learning algorithms (ensemble methods) to obtain better predictive performance have also been proposed. Wu, Yan and Si [23] proposed a stacking ensemble model that used different types of features, (i.e. statistic features, query-item features and session features) consisting of different models, such as logistic regression, gradient boosted decision trees, rank SVM and a deep match model. In a similar approach, Lie et al. [25] presented a cascade model in a large-scale operational e-commerce search application. Their approach modelled 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.

Any web user would agree that there are few things more frustrating than a slow website, as performance plays a major role in customer satisfaction. A faster website means a better visitor experience, on the contrary a slow website will lead to a poor user experience. Providing improved speed was one of the reasons Elasticsearch [17] was built. Elasticsearch is a search engine based on the Lucene library [27]. It provides an open-source, distributed, multitenant-capable full-text search engine that can be used to search all kinds of documents. Elasticsearch is distributed, which means that indices can be divided into shards and each shard can have zero or more replicas. Each node hosts one or more shards, and acts as a coordinator to delegate operations to the correct shard(s).

According to our discussion in this Section, a lot of progress has been made in personalization systems, as well as in recommendation and search systems; nevertheless, in the case of recommendation applications, there is no integrated solution that can semantically understand user's intentions and dynamically evolve based on them, while in the case of personalized search 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 A Framework for the Conversational Web

At a first glance, from a user's point a view, there is little difference between a hyper-personalized website employing conversational web technologies and a conventional website. However, as one uses more and more a conversational website, somehow things get so much easier to use, everything seems simple and intuitive both in terms of UX elements and product search. On the contrary, from the system's point of view, creating a truly conversational website involves a rather complex multi-step procedure, as we will discuss in this Section.

3.1 A Use-Case Scenario

Next we present a use case scenario of how the conversational web can augment the personalized experience of a customer Zoe, who wants to buy the new brand X1 night face cream. Zoe visits her favorite e-commerce site and performs a search using the site's search form. She clicks the third result, although the first two results are about two very popular night face creams of brand X2, Zoe is only interested in brand X1. At this point the implicit conversation between the customer and the user has already begun. The website "*listens*" that a returning customer landed using direct access (direct traffic), searching for brand X1 night cream, and has a strong preference to brand X1, rather than brand X2, so it responds with recommendations about other night creams, as well as other products of brand X1 that are commonly bought together with night creams. In addition, the site recognizes that this is a returning user, so it displays a "welcome back" greeting together with a reminder about a coupon that is expiring in the next few days. Next, Zoe adds the product to her basket and then hovers for some time over a shampoo for oily hair, but finally clicks on a brand X1 serum she noticed in a banner of the main page. These actions alone comprise four discrete messages: as the user has stated that she is actually (a) very interested in the brand X1 night cream (with intent to buy), (b) she is also interested in general for brand X1 and (c) more specifically in serums, and (d) she may need a shampoo for oily hair.

The website once again "*listens*" and responds with even more personalized search results and customized recommendations as it quickly learns the interests of the user, for example it recommends cheaper shampoos for oily hair as the ones displayed before are considered premium products and are probably too expensive. In case Zoe clicks on a cheaper shampoo the website will probably classify Zoe as a customer interested in mid-level products (at least until she starts showing interest for premium products). This is a continuous and everlasting process; the website not only adapts to better serve Zoe's interest but also learns from her behavior and the behavior of other users, aggregating this collective wisdom into actionable insights for improving the overall e-commerce UX of the site.

3.2 A Framework for the Conversational Web

In this Section we propose the overall architecture for creating a conversational website that consists of discrete modules, the behavior analysis module; the user experience analysis module; big data warehousing and the personalization module. Figure 1 depicts the proposed architecture as well as the data exchange means between subsystems [38].

Dynamic analyses of user behavior is performed by the Behavior analysis module. Data from various interdisciplinary analytical sources along with click heatmaps, scroll maps and mouse gestures are used to train models that can identify different patterns and user segments. Classification and support vector machines have provided improved performance for similar tasks in the past [36], while recently deep learning models such as recurrent neural networks have shown promising results. Semantic analysis is also required, as topic modelling and latent dirichlet allocation are useful for analyzing user's interests.

Having multiple dimensions of data, such as user experience data are necessary to understand user perspective and effectively adapt to their needs. User

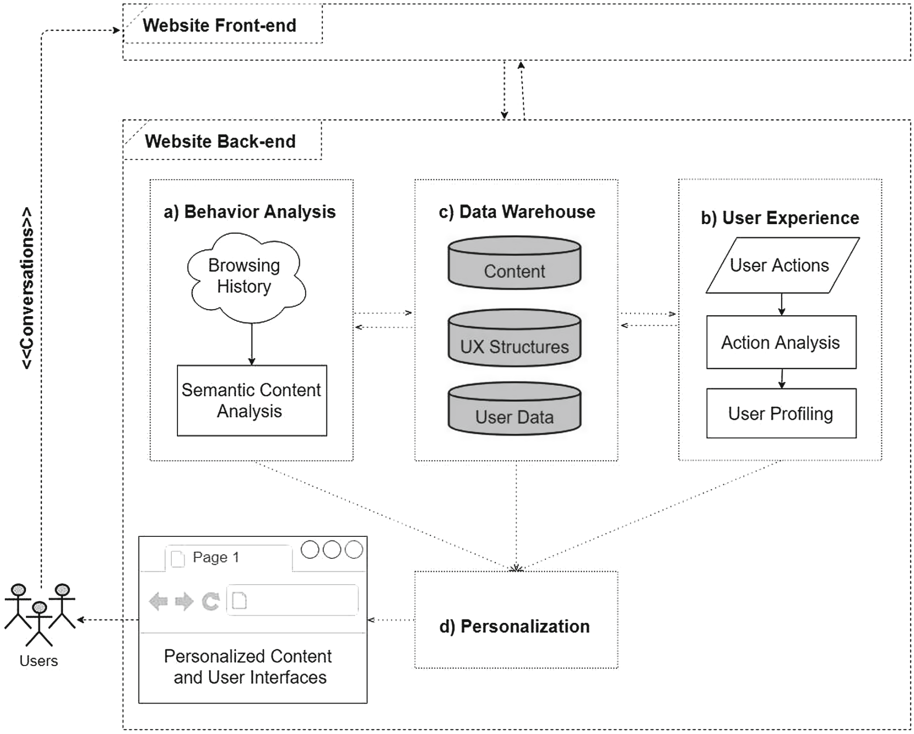

Fig. 1. System architecture in a conversational website.

experience is a multifactor parameter, as website structure, marketing, trust, interactive and information elements, colors and ease of use, all effect a person's perception about a website. All these factors are hard to be defined as they contain strongly subjective elements, key performance indicators, such as bounce rate, average time on site, conversion rate, and depth of search can provide accurate metrics for calculating user experience.

Big data warehousing is necessary for a conversational system. Due to the nature of "*conversations*", which are unstructured, continuous, lengthy and heterogeneous, data warehousing should be able to cope with big volume and high velocity data, as well as heterogeneous information, including product data, user click history, mouse movements, scroll data, e-commerce data including buys, add to cart, and favorites, visual elements and statistics about their use. On the one hand, intelligent models are required for hyper-personalization that can only be trained in offline mode and on the other hand, real-time analysis is necessary for delivering personalized services and user interfaces.

Finally, the main component of the proposed framework is the personalization module which is responsible for adapting content to a particular user according to his or her personal preferences, needs and capabilities. The personalization module dynamically integrates information data and user actions recorded from past user experiences and behavior and provides user-tailored recommendations, website user interfaces and content. Volume, velocity, and variety are key factors for effectively providing personalized experiences, thus this module must integrated hybrid solutions combining both offline and online methods.

4 Personalization via Recommendations and Search in Conversational Web

4.1 Personalized Recommendations

The Conversational Web encompass a wide variety of applications and requirements, thus there is not a universally acceptable solution that fits in every circumstance and efficiently solves any problem in product recommendation. As a result, different approaches have to be adopted that depend on the dataset attributes and the target e-commerce site, such as volume that is mainly depending on the traffic of the e-shop and the number of orders and available products. For this reason we propose ***PRCW*** (Product Recommendations for the Conversational Web), a hybrid approach for product recommendations in e-commerce sites that combines offline $RFMG$ (Recency-Frequency-Monetary-Gender) analysis and online partial matching while we also apply a deep neural model.

A successful recommendation has two prerequisites: (1) be relevant (according to user interests) and (2) be provided on time. As discussed in Sect. 1, hybrid approaches are necessary in Conversational Web, as they can provide real-time recommendations as well as support intense data processing in offline-mode. Towards this goal, we introduce a new hybrid approach using offline and online processing that combines a clustering algorithm with a rule-based method. Clustering is applied to perform consumer segmentation based on consuming behavior, using $RFMG$, a modified version of RFM modeling that combines recency, frequency and monetary with gender, whereas the proposed rule based approach combines partial matching for dealing with the problem of limited user history.

Three are the main processes (Fig. 2) included in the offline phase: (a) data preprocessing, (b) clustering via $RFMG$ analysis and (c) post-processing analysis. First, transforming raw data into an understandable format is necessary, then data cleaning and transformation should take place for smoothing noisy data and resolving the inconsistencies and missing values in the data. Reduction of the number of values via discretization is also necessary, as well as outlier detection for discovering extreme deviations.

In the retail world usually 80% of a business comes from 20% of the customers, as loyal customers are the ones that produce most of the revenue. Based on that observation, RFM (recency, frequency, monetary) [4] analysis is used to determine quantitatively which customers are the best ones by examining how recently a customer has purchased (recency), how often they purchase (frequency), and how much the customer spends (monetary). RFM is widely used for customer segmentation and has received particular attention in retail and professional services industries [13]. One approach to RFM is to assign a score

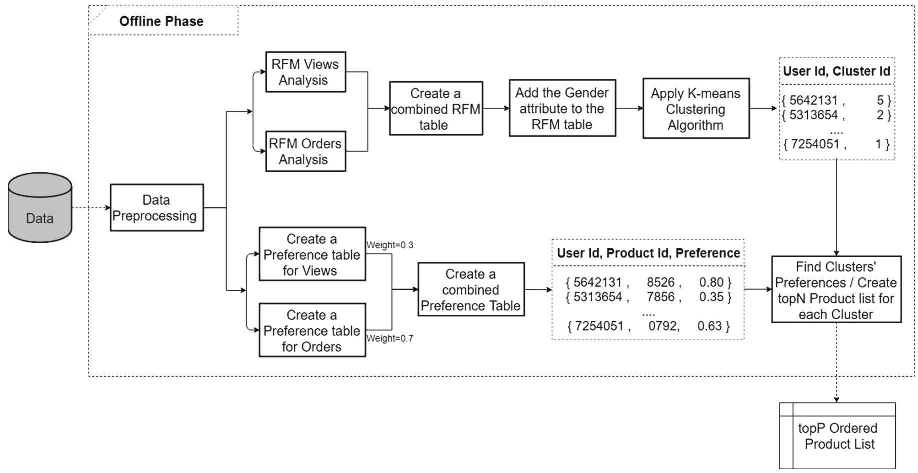
Offline Phase
Data
Data Preprocessing
RFM Views Analysis
RFM Orders Analysis
Create a combined RFM table
Add the Gender attribute to the RFM table
Apply K-means Clustering Algorithm
User Id, Cluster Id
{ 5642131 , 5 }
{ 5313654 , 2 }
....
{ 7254051 , 1 }
Create a Preference table for Views
Create a Preference table for Orders
Weight=0.3
Weight=0.7
Create a combined Preference Table
User Id, Product Id, Preference
{ 5642131 , 8526 , 0.80 }
{ 5313654 , 7856 , 0.35 }
....
{ 7254051 , 0792, 0.63 }
Find Clusters' Preferences / Create topN Product list for each Cluster
topP Ordered Product List

Fig. 2. Offline phase of PRCW.

for each dimension on a scale from 0 to 1. A formula could be used to calculate the three scores for each customer, for example, recency is the number of days that have passed since the customer last purchased (or viewed, clicked) a product, frequency is the number of purchases (or views/clicks) by the customer in the last d days and monetary is the summary of the value for all purchases (views/clicks) by the customer. In our work we also add the Gender attribute as it is highly related to e-commerce behavior (0/1 for males/females). After calculating the recency, frequency, monetary and gender values, normalization is applied.

Next, clustering is exercised on the $RFMG$ values through k-means. This leads to customer segments of similar users where customized information can be provided to them. Within-cluster sums of squares (WCSS) [40] can be used for determining the optimal number of clusters. Next for each consumer segment, a list of top-N most preferred (clicked/bought) items is fetched for every cluster.

Prediction by partial matching (PPM) [15] is an adaptive statistical data compression technique based on context modeling and prediction. PPM models use a set of previous symbols in the uncompressed symbol stream to predict the next symbol in the stream. PPM algorithms can also be used to cluster data into predicted groupings in cluster analysis. Figure 3 depicts our proposed online phase of our approach. The number of previous symbols, n, determines the order of the PPM model which is denoted as $PPM(n)$. Unbounded variants where the context has no length limitations also exist and are denoted as $PPM*$. If no prediction can be made based on all n context symbols a prediction is attempted with $n-1$ symbols. This process is repeated until a match is found or no more symbols remain in context. At that point a fixed prediction is made. Assuming that q_t is the state at time t, an R-order model is defined as in Eq. 1 [15], where in our problem each state is a product view. When a user views the product q_t, partial matching is applied in order to discover the pattern $<q_{t-1}, q_t>$ using data from all the users. Then the top-N products are calculated using the frequencies of the products matched. Naturally, when the order of the model R increases precision is increased but recall on the other hand is decreased.

$$P[q_t|q_{t-1}, ..., q_1] = P[q_t|t_{t-1}, ..., q_{t-R}] \tag{1}$$

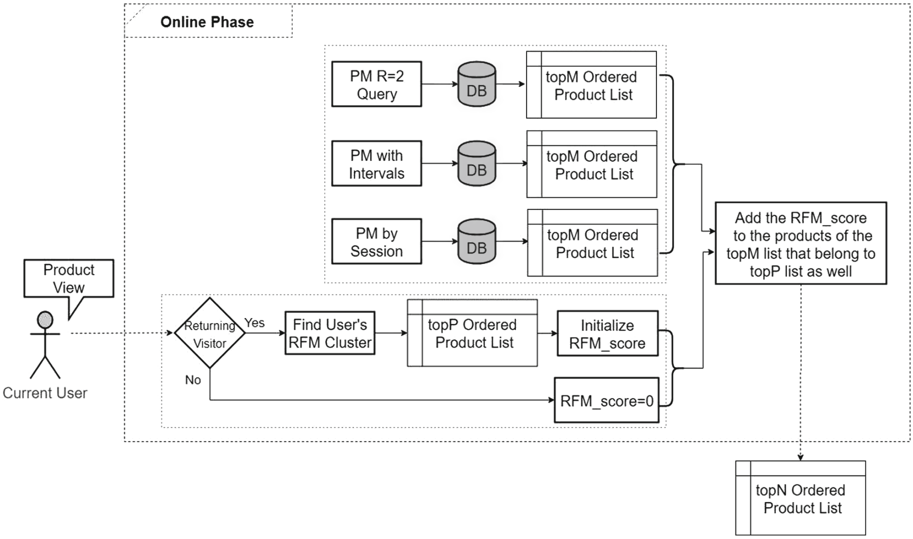
Online Phase
PM R=2 Query
DB
topM Ordered Product List
PM with Intervals
DB
topM Ordered Product List
PM by Session
DB
topM Ordered Product List
Add the RFM_score to the products of the topM list that belong to topP list as well
Product View
Current User
Returning Visitor
Yes
No
Find User's RFM Cluster
topP Ordered Product List
Initialize RFM_score
RFM_score=0
topN Ordered Product List

Fig. 3. Online phase of PRCW.

Due to the nature of the partial matching algorithm, datasets with limited data, originating from medium to small e-commerce site have an increased probability for non-matching patterns. Thus we use two variants of the partial matching procedure. The first one is called "*PM by intervals*" and looks for the pattern $<q_{t-1}, ..., q_t>$ within the history, with the restriction that the time interval between the product views q_{t-1} and $q\{t\}$ is less than a time period T. In this case, the top-N list is computed using the products that were viewed within the time period T and after the product view q_t. The second one is called "*PM by session*" and looks for the pattern $<q_{t-1}, ..., q_t>$ within the history, with the restriction that the product views q_{t-1} and q_t occur within the same session. In this case the top-N list is computed using the products that were viewed within the same session and after the product view q_t. For example, assume that a user u views the products $<p_9, p_1>$ and our history consist of 5 Sessions (Session1–Session5) as in Table 1. The top-N recommendation list using our PPM algorithm is presented in Table 2.

4.2 Personalized Search in Conversational Web

Another crucial application for providing a pleasant customer experience is personalized search that gains popularity as the demand for more relevant information is increased. Our approach for personalized search takes into consideration three sets of features, elicited from: (1) products, (2) users and (3) queries. The architectural diagram of our approach is depicted in Fig. 4. All data are integrated in json files, imported in Elasticsearch. For each product i, we calculate $popularity_i$ as in Eq. 2, 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 product views, thus the use of w_b, w_c and w_v.

$$popularity_i = w_b \frac{buys_i}{|buys|} + w_c \frac{clicks_i}{|clicks|} + w_v \frac{views_i}{|views|} \tag{2}$$

The views, clicks and buys of each user for each product are important factors that encompass hints for the user-product relation. Time is also taken into account, as recent interactions naturally are more important than historic ones. So, in case user history is available, the user-product relevance is calculated by Eq. 3, 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.

$$\begin{aligned} relevance_{u,i} = w_b \frac{\sum \left(buys_{d,u,i} * \left(1 + \frac{1}{1-e^{-x}}\right)\right)}{|buys_u|} \\ + w_c \frac{\sum \left(clicks_{d,u,i} * \left(1 + \frac{1}{1-e^{-x}}\right)\right)}{|clicks_u|} + \ w_v \frac{\sum \left(views_{d,u,i} * \left(1 + \frac{1}{1-e^{-x}}\right)\right)}{|views_u|} \end{aligned} \tag{3}$$

Query-product relevance is also taken into account, 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:

$$\begin{aligned} qScore(q,d) &= qNorm(q) * coord(q,d) \\ &* \sum \big(tf(t\ in\ d) * idf(t)^2 * norm(t,d) \big)(t\ in\ q) \end{aligned} \tag{4}$$

In Eq. 5 $qNorm$ is a measure for comparing queries when 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 and gives these more weigh [16].

Finally, by integrating 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 Eq. 5.

$$\begin{aligned} recommendationScore_{q,u,i} &= w_p * popularity_i + w_r * relevance_{u,i} \\ &+ w_q * qScore(q,d) \end{aligned} \tag{5}$$

Table 1. Example of different products views in 5 sessions.

Sessions	
Session1:	$<p_3>$ $<p_5>$ $<\mathbf{p}_1>$ $<p_2>$
Session2:	$<p_4>$ $<\mathbf{p}_9>$ $<\mathbf{p}_1>$ $<p_3>$
Session3:	$<p_6>$ $<p_4\ >$ $<p_9>$ $<p_4>$
Session4:	$<p_4>$ $<\mathbf{p}_1>$ $<p_2>$ $<p_6>$
Session5:	$<\mathbf{p}_9>$ $<\ p_2>$ $<\mathbf{p}_1>$ $<p_4>$

Table 2. Example of top-N recommendation list using the different PM algorithms.

Method	Recall@1Next	Recall@AllNext	Precision R
PM R = 1	i2 40%	i3 20%	i4 20%
PM R = 2	i3 20%	–	–
PM by intervals	i3 20%	i4 30%	–
PM by session	i2 40%	i3 20%	i4 20%

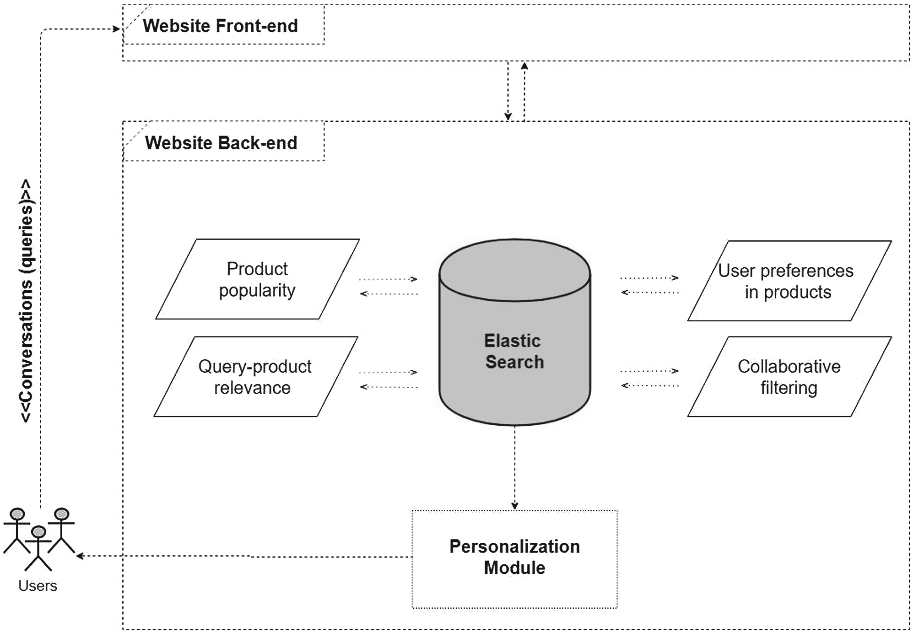

Fig. 4. The proposed architecture for personalized search.

5 Experimental Results

In this Section we evaluate the approaches described in Sects. 4.1 and 4.2 using two publicly available datasets, as well as a private dataset coming from an active e-commerce site.

5.1 Evaluation of the Recommendation Method

Our evaluation of the proposed hybrid recommendation method was performed on two different datasets. The first dataset originated from Pharm24.gr, a small-medium (in terms of traffic) retailer in Greece that provided a click stream containing data from a period of 9 months. Data from the first 7 months were used as the training set, whereas data from the last 2 months were used as the test set. Items with less than 5 views were filtered out from the training set, as well as sessions with less than two item views. Sessions with less than one item view were also removed from the test set, as well as item views that do not exist in the training set. After preprocessing, the training set contained 53,071 sessions of 875,366 events and 9,733 items, whereas the test set contains 86 sessions of 585 events and 244 items.

The second dataset is the RecSys dataset that was provided for the RecSys Challenge 2015 [2]. This dataset contains click-streams of a big e-commerce site, organized in sessions. The training set contains all but the last 10 days of the dataset, whereas the test set contains the sessions of the last 10 days. After the same preprocessing phase, the training set contains 7,802,137 sessions of 30,958,148 events and 37,331 items, while the test set contains 71,060 sessions of 217,014 events and 10,829 items. The evaluation was performed by providing the events of each session of the test set one by one and making recommendations applying the proposed algorithm to the training set.

Evaluation Metrics. To fully evaluate the effectiveness of our model we use precision and recall [12], two commonly used metrics in the field of recommender systems. Suppose that U is the set of users that are examined, $R(u)$ is the set of items recommended to user u, $V(u)$ is the set of items viewed by user u after the recommendation and $V(u, 1)$ is the first product that user u viewed after the recommendation. $PrecisionR$ (Eq. 6) is defined as the percentage of recommended items viewed by the user over the number of recommended products and $PrecisionV$ (Eq. 7) as the percentage of recommended items viewed by the user.

$$PrecisionR = \frac{\sum_u |R(u) \cap V(u)|}{\sum_u |R(u)|} \tag{6}$$

$$PrecisionV = \frac{\sum_u |R(u) \cap V(u)|}{\sum_u |V(u)|} \tag{7}$$

Recall is the percentage of users that viewed recommended items at next timestamps [12]. Three variants of recall were used: *Recall@1Next* (Eq. 8), the strictest one, takes into account only the first next view after recommendation, *Recall@AllNext* (Eq. 9), also considers all next views after recommendation, and *Recall@Positive* (Eq. 10), considers only the cases where the recommendation list has at least one item.

$$Recall@1Next = \frac{\sum_u |R(u) \cap V(u,1)|}{|U|} \tag{8}$$

$$Recall@AllNext = \frac{\sum_u |R(u) \cap V(u)|}{|U|} \tag{9}$$

$$Recall@Positive = \frac{\sum_u |R(u) \cap V(u)|}{\sum_u |R(u) \neq 0|} \tag{10}$$

Recommendation Results. Next, we present the results achieved by the $PRCW$, the RNN and the combination of them using the Pharm24.gr and the RecSys dataset. Results are presented in Tables 3 and 4 accordingly [38]. For deep model evaluation we used a GRU-based RNN model [18] for session-based recommendations, while for partial matching we used the second order model. The input of the network was the actual state of the session represented by a $1 - of - N$ encoding, where N is the number of items (a vector with 1 to the active items and 0 elsewhere), and the output was the likelihood for each item to be part of the next session. Session-parallel mini-batches and mini-batch based output sampling were used for the output.

Table 3. Results of the Pharm24 dataset using the hybrid approach for product recommendation.

Method	Recall@1Next	Recall@AllNext	Prec.R	Prec.V	Pos.Recall
PRCW	0.2880	0.5247	0.0518	0.1414	0.5247
RNN	0.1993	0.3101	0.0348	0.0936	0.3101
PRCW+ RNN	0.3901	0.6065	0.0734	0.1737	0.6065

Table 4. Results of the RecSys dataset using the hybrid approach for product recommendation.

Method	Recall@1Next	Recall@AllNext	Prec.R	Prec.V	Pos.Recall
PRCW	0.0868	0.1711	0.0273	0.0229	0.1711
RNN	0.8120	0.8886	0.0998	0.6380	0.8886
PRCW+RNN	0.8366	0.9037	0.1139	0.7069	0.9037

According to results presented in Tables 3 and 4, the RNN model could not achieve good enough results in a smaller and sparse dataset, while the proposed approach not only demanded considerable less RAM and CPU recourses, but also performed better, as *PRCW* achieved better results than RRN for the Pharm24 dataset, both in terms of Recall and Precision. On the other hand, the RNN has better performance in the RecSys dataset which contains more data both in terms of quantity and density. Nevertheless, the combination of both methods (*PRCW*+RNN) achieves improved performance in both datasets.

When looking into the results of Tables 3 and 4, one can better witness the differences between the algorithms and datasets. Bigger datasets have improved chances to get better recommendations, due to the larger amount of information that contain, and achieve worse results at the *PrecisionR* metric, as there are too many products in the dataset. On the other hand, smaller datasets have shorter sessions and achieve worse results at the *PrecisionV* metric. Deep learning can perform exceptionally well, as long as there are enough data and processing power to feed the neural network. Nevertheless, the proposed method *PRCW* works better on smaller datasets. In any case combining both *PRCW* and RNN delivers the best results in both datasets, which leads us to the conclusion that both methods deliver useful results that should be combined for optimal performance.

5.2 Evaluation of the Personalized Search Solution

The proposed personalized search approach is evaluated against a dataset provided by Diginetica[1] for the "CIKM Cup 2016 Track 2: Personalized E-Commerce Search Challenge", which contains information for more than 500,000 sessions, 1,000,000 clicks, 900,000 searches, 18,000 products and 1,000 categories (Table 5). The data are 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

Table 5. 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		

[1] http://diginetica.com/.

response to a query. Further information regarding the dataset and its characteristics is online available [11].

$nDCG$ measures ranking quality and is often used to measure effectiveness of web search engine algorithms or related applications [39]. In our case $nDCG$ is calculated by employing 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, that represents products with no clicks, 1 stands for somewhat relevant and corresponds to the products which were clicked by the user and 2 is relevant meaning products that were clicked and purchased by the user. In Eq. 11, p stands for the positions up to which we calculate $nDCG$, $rating(i)$ is the score for position i and $|REL|$ is 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: $nDCG = 0.2 * nDCG_{qf} + 0.8 * nDCG_{ql}$.

$$nDCG_p = \sum_{i=1}^{p} \frac{rating(i)}{log_2(i+1)} \Bigg/ \sum_{i=1}^{|REL|} \frac{rating(i)}{log_2(i+1)} \tag{11}$$

The evaluation results are available in Table 6. 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 [24] the optimal value for a was 40, so we tested for $a = 15, 30$ 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 6), 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

Table 6. Evaluation results.

Description	nDCG	Impr.	nDCG$_{\mathrm{ql}}$	Impr.	nDCG$_{\mathrm{qf}}$	Impr.
Baseline	0.242	–	0.220	-	0.334	–
$w_p = 0, w_q = 0, w_s = 1, a = 15$	0.325	34.0%	0.313	42.5%	0.372	11.5
$w_p = 0, w_q = 0, w_s = 1, a = 30$	0.325	34.0%	0.313	42.6%	0.372	11.6%
$w_p = 0, w_q = 0, w_s = 1, a = 40$	0.325	34.0%	0.372	69.5%	0.313	−6.1%
$w_r = 1.5, w_p = 3, w_q = 3$	0.343	41.6%	0.336	52.7%	0.375	12.3%
$w_r = 3, w_p = 1.5, w_q = 1.5$	0.343	41.6%	0.336	52.8%	0.388	16.3%
$w_r = 1, w_p = 1.5, w_q = 1.5$	0.344	41.9%	0.335	52.3%	0.382	14.5%
$w_r = 1.5, w_p = 1, w_q = 1$	0.346	41.6%	0.335	52.6%	0.388	16.3%
$w_r = 1, w_p = 1, w_q = 1, w_s = 0$	0.345	42.4%	0.337	53.41%	0.379	13.5%
$w_r = 1, w_p = 1.5, w_q = 1.5$	0.344	41.9%	0.337	53.2%	0.375	12.3%

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.

6 Conclusion

Delivering individualized experiences is at the heart of converting a business's generic audience into loyal customers. Hyper-personalization helps organizations realize granularity of customer data to gain a deeper customer connection and build a loyal customer base. In order to do so, the application of qualitative tools and frameworks is needed, in order to collect meaningful omnichannel data in real-time. Hyper-personalization is possible only in a truly Conversation Web. The Conversation Web is far more than just chatbots and conversational web services, it is a new type on Web where implicit and explicit conversation between websites and users are continuous.

In this paper we presented a generic framework for the conversational web that can provide hyper-personalized services, such as product recommendation, personalized search, UI/UX personalization, as well as individual messages and promos per customer. We presented two methods for hyper-personalization, one for product recommendations and one for search. Finally, we evaluated these methods on different datasets.

Future work includes working on better integrating the various personalization methods in a way that they can interact and learn from each other. Deeper integration of our hybrid approach with RNNs is also worth investigating in the near future. Moreover, privacy concerns that arise from collecting such a large amount of customer data is an open issue. Finally, we plan to work on improving the integration of both our methods 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 like to thank George Katsikopoulos for his valuable help with the personalized search experiments and Kostas Nikolaros for his useful feedback regarding user search behavior.

References

1. Agichtein, E., Brill, E., Dumais, S.: Improving web search ranking by incorporating user behavior information. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2006, pp. 19–26. ACM, New York, NY, USA (2006). https://doi.org/10.1145/1148170.1148177
2. Ben-Shimon, D., Tsikinovsky, A., Friedmann, M., Shapira, B., Rokach, L., Hoerle, J.: RecSys challenge 2015 and the YOOCHOOSE dataset. In: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, pp. 357–358. ACM, New York, NY, USA (2015). https://doi.org/10.1145/2792838.2798723

3. Bennett, P.N., et al.: Modeling the impact of short- and long-term behavior on search personalization. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2012. pp. 185–194. ACM, New York, NY, USA (2012). https://doi.org/10.1145/2348283.2348312
4. Birant, D.: Data mining using RFM analysis. In: Knowledge-Oriented Applications in Data Mining, chap. 6. Funatsu, Kimito, Rijeka (2011). https://doi.org/10.5772/13683
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003). http://dl.acm.org/citation.cfm?id=944919.944937
6. Burges, C., et al.: Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning, ICML 2005, pp. 89–96. ACM, New York, NY, USA (2005). https://doi.org/10.1145/1102351.1102363
7. Burke, R.: Hybrid recommender systems: survey and experiments. User Modeling User-Adap. Inter. **12**(4), 331–370 (2002). https://doi.org/10.1023/A:1021240730564
8. Chen, L., Pu, P.: Critiquing-based recommenders: survey and emerging trends. User Modeling User-Adap. Inter. **22**(1), 125–150 (2012). https://doi.org/10.1007/s11257-011-9108-6
9. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 785–794. ACM, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939785
10. Chen, Y.L., Kuo, M.H., Wu, S.Y., Tang, K.: Discovering recency, frequency, and monetary (RFM) sequential patterns from customers' purchasing data. Electron. Commer. Res. Appl. **8**(5), 241–251 (2009). https://doi.org/10.1016/j.elerap.2009.03.002. Special Issue: Marketing and Electronic Commerce
11. CIKM Cup organizing committee: Cikm cup 2016 track 2: Personalized e-commerce search challenge (2016). https://competitions.codalab.org/competitions/11161#learnthedetails-data2. Accessed 15 Jan 2019
12. Davis, J., Goadrich, M.: The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning, ICML 2006, pp. 233–240. ACM, New York, NY, USA (2006).https://doi.org/10.1145/1143844.1143874
13. Fader, P.S., Hardie, B.G., Lee, K.L.: RFM and CLV: using iso-value curves for customer base analysis. J. Mark. Res. **42**(4), 415–430 (2005). https://doi.org/10.1509/jmkr.2005.42.4.415
14. Felfernig, A., Friedrich, G., Jannach, D., Zanker, M.: Developing constraint-based recommenders. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 187–215. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_6
15. Gellert, A., Florea, A.: Web prefetching through efficient prediction by partial matching. World Wide Web **19**(5), 921–932 (2016). https://doi.org/10.1007/s11280-015-0367-8
16. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
17. Gormley, C., Tong, Z.: Elasticsearch: The Definitive Guide, 1st edn. O'Reilly Media Inc., Sebastopol (2015)
18. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. CoRR abs/1511.06939 (2015). http://arxiv.org/abs/1511.06939

19. Huang, S.L.: Designing utility-based recommender systems for e-commerce: evaluation of preference-elicitation methods. Electron. Commer. Rec. Appl. **10**(4), 398–407 (2011). https://doi.org/10.1016/j.elerap.2010.11.003
20. Jannach, D., Ludewig, M.: Investigating personalized search in e-commerce. In: FLAIRS Conference, pp. 645–650. AAAI Press (2017)
21. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2002, pp. 133–142. ACM, New York, NY, USA (2002). https://doi.org/10.1145/775047.775067
22. Saleh, K.: Online shopping personalization - statistics and trends (2018). https://www.invespcro.com/blog/online-shopping-personalization/. Accessed 15 Jan 2019
23. Kong, D.: Personalized feature based re-ranking method for ecommerce search at cikm cup 2016. Technical report, CIKM Cup (2016)
24. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009). https://doi.org/10.1109/MC.2009.263
25. Liu, S., Xiao, F., Ou, W., Si, L.: Cascade ranking for operational e-commerce search. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2017, pp. 1557–1565. ACM, New York, NY, USA (2017). https://doi.org/10.1145/3097983.3098011
26. Mahmood, T., Ricci, F.: Improving recommender systems with adaptive conversational strategies. In: Proceedings of the 20th ACM Conference on Hypertext and Hypermedia, HT 2009, pp. 73–82. ACM, New York, NY, USA (2009). https://doi.acm.org/10.1145/1557914.1557930
27. McCandless, M., Hatcher, E., Gospodnetic, O.: Lucene in Action, Second Edition: Covers Apache Lucene 3.0. Manning Publications Co., Greenwich (2010)
28. Palotti, J.: Learning to rank for personalized e-commerce search at CIKM cup 2016. Technical report CIKM Cup (2016)
29. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_10
30. Quadrana, M., Karatzoglou, A., Hidasi, B., Cremonesi, P.: Personalizing session-based recommendations with hierarchical recurrent neural networks. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, 27–31 August 2017, pp. 130–137 (2017). https://doi.acm.org/10.1145/3109859.3109896
31. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009). https://doi.org/10.1561/1500000019
32. Rubens, N., Kaplan, D., Sugiyama, M.: Active learning in recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P. (eds.) Recommender Systems Handbook, pp. 735–767. Springer, Boston (2011). https://doi.org/10.1007/978-0-387-85820-3_23
33. Salimans, T., Paquet, U., Graepel, T.: Collaborative learning of preference rankings. In: Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys 2012, pp. 261–264. ACM, New York, NY, USA (2012). https://doi.acm.org/10.1145/2365952.2366009
34. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, WWW 2001, pp. 285–295. ACM, New York, NY, USA (2001). https://doi.org/10.1145/371920.372071

35. Speretta, M., Gauch, S.: Personalized search based on user search histories. In: The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2005), pp. 622–628, September 2005. https://doi.org/10.1109/WI.2005.114
36. Sun, A., Lim, E.P., Ng, W.K.: Web classification using support vector machine. In: Proceedings of the 4th International Workshop on Web Information and Data Management, WIDM 2002, pp. 96–99. ACM, New York, NY, USA (2002). https://doi.acm.org/10.1145/584931.584952
37. Teevan, J., Dumais, S.T., Liebling, D.J.: To personalize or not to personalize: Modeling queries with variation in user intent. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, pp. 163–170. ACM, New York, NY, USA (2008). https://doi.org/10.1145/1390334.1390364
38. Vavliakis, K.N., Kotouza, M.T., Symeonidis, A.L., Mitkas, P.A.: Recommendation systems in a conversational web. In: Proceedings of the 14th International Conference on Web Information Systems and Technologies, WEBIST 2018, Seville, Spain, 18–20 September 2018, pp. 68–77 (2018). https://doi.org/10.5220/0006935300680077
39. Wang, Y., Wang, L., Li, Y., He, D., Liu, T.Y., Chen, W.: A theoretical analysis of NDCG type ranking measures. CoRR abs/1304.6480 (2013)
40. Witten, D.M., Tibshirani, R.: A framework for feature selection in clustering. J. Am. Stat. Assoc. **105**(490), 713–726 (2010)
41. Ying, Y., Feinberg, F., Wedel, M.: Leveraging missing ratings to improve online recommendation systems. J. Mark. Res. **43**(3), 355–365 (2006). http://www.jstor.org/stable/30162410
42. Yu, J., Mohan, S., Putthividhya, D.P., Wong, W.K.: Latent Dirichlet allocation based diversified retrieval for e-commerce search. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM 2014, pp. 463–472. ACM, New York, NY, USA (2014). https://doi.org/10.1145/2556195.2556215
43. Zhang, S., Yao, L., Sun, A.: Deep learning based recommender system: a survey and new perspectives. CoRR abs/1707.07435 (2017)
44. Zhao, X., Zhang, W., Wang, J.: Interactive collaborative filtering. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, CIKM 2013, pp. 1411–1420. ACM, New York, NY, USA (2013). https://doi.acm.org/10.1145/2505515.2505690