Recommender Systems in E-Commerce

Conference Paper · August 2014

DOI: 10.13140/2.1.3235.5847

Asad M. Madni
University of California, Los Angeles

Recommender Systems in E-Commerce

Sanjeevan Sivapalan[1], Alireza Sadeghian[2], Hossein Rahanam[3]
Computer Science Department
Ryerson University
Toronto, Canada
Email: [1]s3sivapa@scs.ryerson.ca, [2]asadeghi@ryerson.ca, [3]hossein@ryerson.ca

Asad M. Madni
Electrical Engineering Department
University of California Los Angeles
Los Angeles, California
Email: ammadni@yahoo.com

***Abstract*— Internet is speeding up and modifying the manner in which daily tasks such as online shopping, paying utility bills, watching new movies, communicating, etc., are accomplished. As an example, in older shopping methods, products were mass produced for a single market and audience but that approach is no longer viable. Markets based on long product and development cycles can no longer survive. To stay competitive, markets need to provide different products and services to different customers with different needs. The shift to online shopping has made it incumbent on producers and retailers to customize for customers' needs while providing more options than were possible before. This, however, poses a problem for customers who must now analyze every offering in order to determine what they actually need and will benefit from. To aid customers in this scenario, we discuss about common recommender systems techniques that have been employed and their associated trade-offs.**

Keywords—E-Commerce, Recommender Systems, Online shopping, Online communications.

I. Introduction

Recommender systems were first introduced as Collaborative Filtering by its authors in which they discuss how people collaborate and filter email documents that are relevant to them and of use to their audience. The filtering process included analyses of common properties among two or more documents. Properties belonging to the documents that were analyzed included message, reply or its annotations. This was found to be more effective than simple analysis of the document's contents which many other mail systems provided. Human intervention of the filtering process led to more interesting documents being selected [1] [2].

Recommender systems allow rapid and automated customization and personalization of e-commerce sites. They allow the sites to generate more sales by tailoring to the needs of the visitors and turning them into consumers, up-selling extra products by bundling closely related things together, and increasing customer loyalty [3] [4]. Customer loyalty is achieved by showing customers that they take time to understand their needs and to learn more about them [5]. This is evident when the website structure, the products, and presentation of products changes to customers' needs and preferences. Customers revisit these websites rather a competitor's because they are accustomed to it and do not have to go through a learning process. Even if the competitor were to offer similar experience, customers will return to a site they already know.

II. Background Information

A. *Information utilized*

Information that is often used to make recommendation decisions include user demographics, item attributes, and user preferences [2] [6]. Table 1 lists common properties that are used. User demographics are attributes of users in general which can affect the results of recommendations that are made. This information includes amongst other things such as likes and dislikes of a particular gender, age group, occupation, income level, and hobbies. Item attributes are classified as either extrinsic or intrinsic. Extrinsic features cannot be easily identified by analyzing the contents automatically. Intrinsic features on the other hand are easily obtainable from the contents. Sometimes, features of items are obtained from the description of the items themselves in addition to analyzing the item. This is evident in cases such as news articles or web pages. User preferences are either a presence score, such as "likes it" or "dislikes", or numerical score indicating how much the user likes the product. In some cases, they are explicit ratings provided by users when they are asked to rate the product. The amount of time a user spends on a particular page, reading and analyzing are implicit indicators of the user's ratings [7]. While implicit indicators are more difficult to gather, they offer more information about the user that the user otherwise might not provide.

B. *Knowledge Discovery in Databases (KDD)*

KDD, also referred to as data mining is used to describe extraction of useful information from a dataset [8]. The information can either be implicit or explicit. It is used to find ways to improve efficiency of e-commerce websites by finding new ways to sell more products to customers [4] [3]. Companies that utilize KDD can find patterns in user buying behaviors, such as what time of year certain products are more likely to be bound and make recommendations on that generating millions of dollars in revenue [9]. One of the most important algorithms used in
KDD is the association rules. The rules try to associate a set of products to a different set of products in such way that the presence of one product from a set implies there is a high

Table 1 - Data used in recommendation systems [2]

Data Type	Description
Rating Data	rating scores, such as discrete multi-levels ratings and continuous rating; and latent comments, such as best, good, bad, worse
Behaviour Pattern Data	duration of browsing, click times, the links of webs; save, print, scroll, delete, pen, close, refresh of webs; selection, edition, search, copy, paste, bookmark and even download of web content
Transaction Data	purchasing date, purchase quantity, price, discounting
Production Data	for movies or music, it means actor or singer, topic, release time, price, brand and so on, while for webs or documents, it means content description using key words, the links to others, the viewed times, the topic

chance of another product being in the same set [4]. Well known associate rules include Apriori, Direct Hashing and Pruning (based on Apriori), Tree Projection algorithms, and FP-tree algorithms [4].

III. Types of Recommendations

Recommender systems can be personalized, non-personalized, attribute-based, item-to-item correlation, and people-to-people correlation. Recommendations are either short-lived or long-lived depending on the implementation. The system is considered automatic if it requires minimal or no input from the active user and manual if it requires some work [3]. *Personalized recommendations* are automatic and based on the user's preferences such as favorite color, movie genre and music group. They are often compared against hand-picked products by content-providers and experts for user's preferences and tastes to provide recommendations [10]. *Non-personalized recommenders* generate recommendations based only on product ratings from other users of the system [10]. These recommendations are straight forward since they require very little effort to produce and considered automatic since user input is not required [3]. These recommendations are not short-lived since they can be applied to a variety of users. In *Attribute-based recommendations*, items can be described using various features, and attributes which are used to generate recommendations. This method is considered manual since the user must explicitly search for a certain type of product to base the recommendations on [3] [10]. These recommendations can be short-lived or not depending on how long the system remembers user's preferences for. *Item-to-item correlation recommenders* recommend items based on other items the user has displayed interest in. These recommendations are prevalent in e-commerce sites where new products are recommended based on what the user has in their shopping cart [3]. These recommendations are manual since user must have a non-empty cart, and short-lived because the user does not have full shopping carts. Association rules are most often used in this system [11]. *People-to-people correlation* system finds similarity between the active and other users in the system, recommends products other customers have purchased or rated in the past [4]. Collaborative filtering is the most commonly used approach in this system [12]. Since it requires users to have purchased or rated products in the past this method is very manual. The recommendations can last depending on system's design.

IV. Recommendation Techniques

Different algorithms and techniques are used by recommender systems to generate recommendations. The most popular ones are association rules, collaborative filtering, content-based filtering and hybrid filtering.

A. Association Rules

Association rules are used to recommend products based on their presence along with other products [4] [13]. When two products are purchased together, the presence of one item in a transaction can be used to determine the second product also being in the same transaction. This is very useful when making recommendations to new users who wish to make purchases. To define association more formally, a collection of products m products $\{P_1, P_2, P_{3,},...P_m\}$ belongs to set P. We say a transaction T from set of transactions D is a subset of P, $T \subseteq P$ such that the transaction contains products from P. Each transaction can be uniquely identified as *TID*. A transaction T contains set X, a subset of products from P and it is a subset of T. Association rules implies that there exists Y, subset of P and there is no mutual product between X. This means that whenever products from X exist in a transaction T, there is high likelihood that products from Y will also exist in the same transaction [11] [14]. Two variables, *confidence* c, and *support* s [11] are used to measure the quality of the associations made [4]. Support measures how frequent the association happens in the entire set of transactions as shown in (1) and confidence measures the frequency of both products occurring whenever one product exists in the transaction as shown in equation (2).

$$s = \frac{\text{number of transactions containing X or Y}}{\text{total number of transactions}} \tag{1}$$

$$c = \frac{\text{number of transactions containing X or Y}}{\text{number of transactions containing X}} \tag{2}$$

A major drawback is the support for a lot rules. Association rules are slower and not very effective when a lot of mining rules are used to make recommendations [14]. An e-commerce site with lot of products and transactions would have trouble scaling with multiple rules because they need to be applied to the entire database and still find recommendations for users within an acceptable timeframe.

B. Collaborative Filtering

Collaborative filtering approach uses customer details, ratings, and reviews aggregated from all the customers to build recommendations [6] [12] [8]. The strength of this approach is that it analyzes existing active customers with similar preferences and characteristics of the current customer to build the recommendations. The filtering method is achieved through a heuristic-based, a model-based method, or a hybrid model that combines characteristics from both heuristic and model-based approaches [2] [4]. The heuristic based or memory-based collaborative filtering model takes in rating data, whether product was purchased or not, and duration of viewing products to calculate the recommendations [2] [12]. Active customers whose information is used is done by selecting all the customers who are neighbors of the current customer using similarity measures including personal information, cosine metric, and jaccard coefficient for binary data [2]. Then, utilizing k-nearest neighbor classification method, prediction value is computed for each product that current customer has not viewed but the other active customers have. With the newly calculated set, recommendation is created based on products with the highest scores. There are many different algorithms and technique that can be used in heuristic based collaborative filtering includes k-nearest neighbor algorithm, web mining algorithms, decision trees, and support vector machines [2]. The model based collaborative filtering technique uses training data such as the active user's ratings and reviews to build a model using different data mining and machine learning algorithms [2] [12]. The model is then validated using the testing data and list of products and rating is predicted for them if customers have not given any rating to it yet or been exposed to it. While the heuristics based model uses the entire database and the customers to create recommendations for the active customer, the model based approach only relies on the active customer's information as the input. Techniques and algorithms from fields such as Bayesian model, clustering, association rules, artificial neural networks, linear regression, maximum entropy, latent semantic analysis, and Markov process can be used [2].

Collaborative filtering is the most successful technology used in recommender systems and it is the most widely used on the internet [4]. The recommender system is split into three components: representation, neighborhood formation, and recommender generation. As described in Figure 1, in the representation, matrix *R* of size *n* x *m* is constructed for *n* customers and *m* products i n the database where $r_{i,j}$ is one if

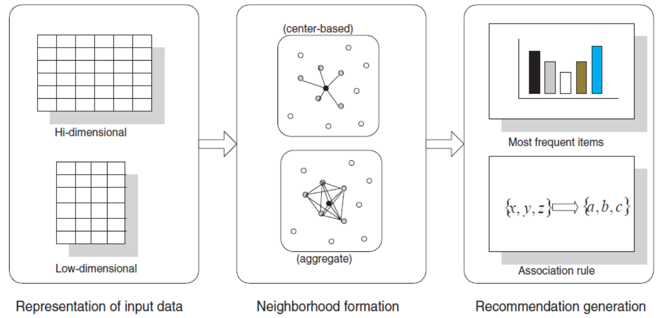

Figure 1 - Part of Recommendation Systems [4]

the i-th customer bought j-th product and zero otherwise. The matrix is called original representation [4]. Collaborative filtering has challenges with sparsity, scalability and synonymy. Synonymy occurs because similar products are labeled differently in real life, and recommender systems cannot always associate between them, and treat them as different. A reduced dimensional representation is constructed to alleviate the weaknesses. A matrix of size *n* x *k* is constructed where all values in the matrix are nonzero, which implies that each customer has had an association with the *k* product. Due to decreased size, it also helps alleviate the problem with synonymy.

The neighborhood formation forms the heart of the recommendation system. In this step, the similarities between customers are computed and used to create proximity based neighborhood between the target customer and likeminded customers [13]. For each customer *u* and *N* customers where $N = \{N_1, N_2 \ldots N_l\}$, the customer *u* does not belong to set of *N* and the similarity $sim(u, N_k)$ $\mathrm{sim}(u, N_k)$ is greater than $sim(u, N_{k+1})$ $\mathrm{sim}(u, N_k)$ with $sim(u, N_1)$ $\mathrm{sim}(u, N_k)$ being the maximum. Proximity measures can be calculated using (3) or (4).

$$\mathrm{corr}_{ab} = \frac{\sum_i (r_{ai} - \overline{r_a})(r_{bi} - \overline{r_b})}{\sqrt{\sum_i (r_{ai} - \overline{r_a})^2 \sum_i (r_{bi} - \overline{r_b})^2}} \tag{3}$$

Equation 3 calculates the correlation between two different variables in terms of how the variables are related. The correlation between user *a* and *b* is defined as the summation over *i* are over the items for which both user *a* and *b* have voted [12] [15]. The notations r_{ai} and r_{bi} represent the rating given to *i*-th item by user *a*, and user *b* respectively. r_a $\mathrm{r_a}$ and r_b represent the averages. The result is between -1 and 1 with -1 being a perfect negative correlation. In equation 4 both *a* and *b* are vectors in the m dimensional product space and the distance between them is calculated as the cosine of the angle between the two vectors. For *n* customers, a similarity matrix *S* of size *n* x *n* is computed using either one of the proximity measures.

$$\cos \vec{a}, \vec{b} = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \times \|\vec{b}\|} \tag{4}$$

There are two methods to forming a neighborhood: centre-based and aggregate neighborhood [4]. Centre based techniques form a neighborhood for a customer *c* of size *k* by selecting *l* nearest customers where both *k* and *l* are arbitrary. Aggregate neighborhood creates a neighborhood of size *l* for a customer *c* by selecting the closest customer. The rest of the *l - 1* neighbors are selected similarly. At a certain point jj, when $\vec{C}$ there are *j* neighbors in *N* and $j < l$, the centroid of N, $\vec{C}$ is calculated using (5). Then a new customer *w* who is not in *N* is selected as the $j+1^{th}$ if *w* is the closest to the centroid $\vec{C}$. The centroid is then recomputed for j +1 neighbor and continues until the number of neighbors in *N* is *l*.

$$\vec{C} = \frac{1}{j}\sum_{\vec{V}\in N}\vec{V} \tag{5}$$

The final part of the recommendation system is to make the actual recommendations which is to calculate top *m* recommendations from the computed neighborhood of customers. Two prominent techniques that are used are most-frequent item recommendation, and association rule-based recommendations [4]. In *Most-Frequent Item Recommendation,* neighborhood *N* is scanned frequency count of purchases is calculated for each neighbor. All the products are then sorted according to the frequency and *m* most frequently bought products that is not purchased by the current customer are recommended [4]. In *Association Rule-Based Recommendations: L* neighbors taken into account while using association rules to generation recommendations. Association rules work by recommending a product that a neighbor bought with the presence of another product [14]. However, having a limited number of neighbors to work limits the effectiveness of the recommendations made [4].

Collaborative filtering has a major disadvantage since it requires data to exist in order to be useful. It has two major limitations which are sparsity, and scalability [2] [4]. Sparsity occurs in large e-commerce sites with low purchases [8]. In large e-commerce sites like Amazon and CDNow, active customers cannot easily purchase products such that they buy even 1% person of the store's products. A recommender system that uses nearest neighbor algorithms is ill suited to make recommendations for an active user in those sites. This is commonly known as reduced coverage. It also leads to poor recommendations due to lack of enough data [4]. Nearest neighbor algorithms grow with the number of customers and products available, thus leading to scalability issues.

C. Content-based filtering

Content-based filtering is based on being able to analyze products and find similarity with active user to recommend products. Unlike collaborative filtering or association rules, this method does not require an active database of purchase history. It is based on information retrieval, analysis and filtering [2] [6] [16] [17]. This approach is used mainly in places where content can be read or analyzed such as news articles, movies and anything with metadata attached. It also gives recommendations based on items the user has viewed in the past. The contents can be described using labels and the labels are given a weight of how well they describe the article. Using these labels and user preferences, nearest neighbor or clustering algorithms can be used to recommend other articles to the active user. However, new users with limited information and limited number of labels pose a challenge to this method. Common algorithms that are applicable include k-nearest neighbor, clustering, Bayesian, and artificial neural networks [2]. Information filtering systems are usually used with structured data that can be easily analyzed to gain insights. Vast amounts of data are usually analyzed by filtering systems to give recommendations because it is per user profile [6] [17]. The user profiles are obtained explicitly through questionnaires and forms or implicitly using behavioral information [7]. A set of attributes describing a user is computed and they are then used to make recommendations to the user. The attributes are compared with keywords describing the recommendations as mentioned. Keywords used to make recommendations are weighted using term frequency/inverse document frequency (TF-IDF) method to measure importance. Term frequency *TF* is calculated from *N* items that could potentially be recommend to user as [6] [16].

$$TF_{i,j} = \frac{f_{i,j}}{max_{z,f_{z,j}}} \tag{6}$$

where $f_{i,j}$$f_{i,j}$ is the number of times keyword k_i $k_i$appears in document d_j and computed maximum $f_{z,j}$ $f_{z,j}$ is the frequencies of all keywords k_z that appear in document d_j. Keywords that appear in many different documents are not useful when distinguishing between relevant and irrelevant documents. To do that, inverse document frequency is used [6] [16]. Inverse document frequency IDF is calculated for keyword k_i *as :*

$$IDF_i = \log\frac{N}{n_i} \tag{7}$$

Then we can simply get the weight for keyword k_i in document $k_i$$d_j$ as [6] [16]:

$$w_{i,j} = TF_{i,j} \times IDF_i \tag{8}$$

Content-based filtering systems also recommend new items based on what the user had liked previously [6]. A content based profile can be constructed for a user from their previously liked items, ratings, search keywords, and other behavioral data. This information is aggregated to create a profile for the user. These types of systems are highly dependent on the items being easy to analyze. In order for recommender systems to be able to generate recommendations, content must be structured and easy to parse. If this is not, then the item must be described manually [6]. Another problem is being able to differentiate between a bad item and a good item based on retrieved information. A bad item using same keywords as good item will also get recommended. Two other major drawbacks are lack of information about a user, and overspecialization. When a new user is introduced into the system, their preferences and profiles are not aggregated. The user would not have given enough ratings, and reviews to products. This leads to insufficient information to generate recommendations [6]. When the system is only able to recommend certain items based on user's profile, it leads to overspecialization. This is due to the user having rated a specific item, the recommender system is only able to provide recommendations for similar products. This also leads to the user never being recommended outside of their previous ratings [6]. In such cases, genetic algorithms which evolve information filtering agents to provide recommendations have been proposed. This is done by using an iterative method where previous output is used to learn and adapt dynamically [18] [19].

D. Hybrid filtering

To avoid problems that exist in both content-based and collaborative filtering systems, hybrid solutions have been proposed [6]. Solutions include: implement both filtering separately and combine the results, incorporating characteristics of content-based filtering to collaborative adding characteristics of collaborative filtering to content-based filtering systems and new algorithms that incorporate both systems' techniques. Combining different recommender systems approach involves building two different recommender systems based on collaborative-based and content-based approaches. The recommendations can be separately generated and then combined linearly [20]. The algorithm assigns a weight to the generated recommendations per user based on its relevance to the user. The recommendations are then added in order to be presented to user. The second method is to use the level of confidence each system produced for the results that are more consistent with the user's past ratings and provide them to the user [21]. Many recommender systems are implemented using collaborative-based approach with content-based user profiles generated through content-based approach [6]. The profiles are then used to find similarity between users rather than items which help the system overcome some of the sparsity-related limitations. Recommendations can be generated through collaborative filtering first. They are then compared against current user profile to determine if it's interesting to the user or not and to present it [19]. Curse of dimensionality occurs when a lot of features exist per item that makes it difficult to cluster or compare them [13]. The most common approach is to use dimensionality reduction algorithm on a group of content based profiles [6]. This allows performance improvements since it reduces the amount of preferences/features that must be compared to generate the recommendations.

V. CONCLUSION AND FUTURE WORK

A. Conclusion

Recommender systems allow e-commerce sites to be highly customizable for the user and buyer. They allow companies to better understand their users, provide personalized stores, and in turn increase customer satisfaction and loyalty. They are implemented by utilizing various existing data mining tools and adapting them to current needs. Popular approaches include using association rules, collaborative filtering and content-based filtering and hybrid filtering. Recommendations using association rules are generated based on previous transactions the user has already displayed interest in. Collaborative filtering allows the active user to get recommendation based on products that users with similar interest have purchased and rated positively, and by using the active user's previous ratings and transaction history to build a model that provides a new set of similar products. Content-based filtering compares the user's personal profile and preferences with the database to find products that are of interest and align with the active user and present them. Recommendations can range from being personalized to community driven and allow for a wide range of possibilities. The recommendations are also being refreshed due to the nature of changing search history, ratings, and arrival of new products. This also poses many challenges which include cold start, handling anonymous users, creating a social recommender system that can accommodate more than one active user, handling various different data sources and scalability with increased data.

B. Future Work

Over the years, recommender systems have been extensively used in e-commerce sites but they still pose research and practical challenges including scalability, rich data, consumer-centered recommendations, anonymous users, and connecting recommenders to markets. They are used in large sites such as Amazon, where millions of products are sold, actively making recommendations to thousands of users simultaneously in real-time. The performances monitored include latency in generating recommendations, number of simultaneous requests being handled, number of consumers, number of products and vast amount of rating and review data. In order to alleviate this problem, different techniques from data mining such as dimensionality reduction and parallelism are employed. A problem faced when scaling using data mining techniques is the sparsity of ratings [2]. The recommender system is valuable when users have not rated most of the products. If different groups of users rate different categories of products, it becomes less likely the rated products will overlap and can be used to generate recommendations. Although dimensionality reduction algorithms are employed to fix this, they are ill-suited for extremely sparse data and have to be modified for recommender systems [10]. While large amount of data will slow down the system, lack of data will also hurt the ability to generate recommendations.

As more information becomes available, algorithms and techniques must also evolve [3]. Until recently, recommendations are generally based on single value rather than combination of different data. New machine learning algorithms are emerging that solve this issue by building models based on various product attributes, and user features

[10]. However, a big challenge is posed with seasonal and temporary data. While a snow blower might be a useful recommendation in winter based on a user's search history and behavior, it is irrelevant in the summer. Temporal associations are an emerging problem that requires much more research [10]. Also, several recommenders are designed with a single user as the end consumer, and there is a lack of social recommenders [10], an example being recommending movies at theatre. Innovative algorithms that take into consideration varying perspectives and preferences of different users are needed. Another challenge is making recommendations that the user wants and finds useful since it is not always easy to tell if the recommendation was indeed useful. A possible approach is explaining the recommendations to the user in terms how the user's preference or behavior led to the recommendation and gather feedback [10]. It is an extremely difficult task to provide recommendations when the user has been browsing and purchasing anonymously [8]. A methodology was proposed that attempts to solve this by studying purchase patterns of users, and predicting purchase probability of new products [22]. In order to study purchase patterns, the user's web log can be utilized with using information such as IP address, cookies, and other session data. This information can be used to extract products the user has viewed previously. The purchase probability is calculated using associative mining rules to determine products the user might be interested. However, the shortcoming of the approach is that it is temporary. A user visiting from different browser might not be able to get same recommendations as they would get from same browser. Recommender systems are currently treated as virtual salesmen since they only give suggestions to new products, and do not actively market that product [3] [10]. The system should also take into account price-value for the user, and profits for the company. When suggesting new prices based on studying user behavior, ethical issues are raised because of price discrimination for different users [10]. It is challenging to maintain user loyalty and trust when making recommendations based on generating higher company profits [2].

References

[1] D. Goldberg, D. Nichols, B. Oki and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry," in *ACM*, New York, 1992.

[2] C.-P. Wei, M. Shaw and R. Easley, "A Survey of Recommendation Systems in Electronic Commerce," National Sun Yat-Sen University, Kaohsiung,, 2001.

[3] B. Schafer, J. Konstan and J. Riedl, "Recommender systems in e-commerce," in *ACM*, New York City, 1999.

[4] B. Sarwar, G. Karypis, J. Konstan and J. Rieldl, "Analysis of recommendation algorithms for e-commerce," in *ACM*, New York, 2000.

[5] J. Pine, Mass Customization, Boston: Harvard Business School Press, 1993.

[6] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions,* vol. 17, no. 6, pp. 734-749, 2005.

[7] M. Claypool, P. Le, M. Waseda and D. Brown, "Implicit Interest Indicators," in *ACM*, New York, 2001.

[8] A. N. Regi and R. Sandra, "A Survey on Recommendation Techniques in E-Commerce," vol. 2, no. 12, 2013.

[9] R. J. Brachman, T. Khabaza, W. Kloesgen, G. Piatetsky-Shapiro and E. Simoudis, "Mining business databases," in *ACM*, New York, 1996.

[10] B. Schafer, J. Konstan and J. Riedl, "E-Commerce Recommendation Applications," in *Applications of Data Mining to Electronic Commerce*, Minneapolis, Springer US, 2001, pp. 115-153.

[11] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in *20th International Conference on Very Large Databases*, Santiago, 1994.

[12] J. S. Breese, D. Heckerman and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Morgan Kaufmann Publishers*, San Francisco, 1998.

[13] X. Amatriain, A. Jaimes, N. Oliver and J. M. Pujol, in *Recommender Systems Handbook*, New York City, Spring US, 2011, pp. 39-71.

[14] W. Lin, S. Alvarez and C. Ruiz, "Efficient Adaptive-Support Association Rule Mining for Recommender Systems," *Data Mining and Knowledge Discovery,* vol. 6, no. 1, pp. 83-105, 2002.

[15] B. Sarwar, G. Karypis, G. Konstan and J. Rield, "Item-based collaborative filtering recommendation algorithms," in *ACM*, New York City, 2001.

[16] G. Salton, Automatic text processing: the transformation, analysis, and retrieval of information by computer, Boston: Assison-Wesley Longman Publishing Co., 1989.

[17] N. Belkin and B. Croft, "Information filtering and information retrieval: two sides of the same coin?," vol. 35, no. 12, 1992.

[18] B. Sheth and P. Maes, "Evolving agents for personalized information filtering," Orlando, 1993.

[19] M. Balabanovic and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM,* vol. 40, no. 3, pp. 66-72, March 1997.

[20] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes and M. Sartin, "Combining content-based and collaborative filters in an online newspaper," in *ACM*, Berkeley, 1999.

[21] T. Tran and R. Cohen, "Hybrid Recommender Systems for Electronic Commerce," in *AAAI Workshop*, 2000.

[22] E. Suh, S. Lim, H. Hwang and S. Kim, "A prediction model for the purchase probability of anonymous customers to support real time web marketing: a case study," *Expert Systems with Applications,* vol. 27, no. 2, pp. 245-255, 2004.