

Integrating Collaborative Filtering and Matching-based Search for Product Recommendations

Noraswaliza Abdullah¹, Yue Xu², and Shlomo Geva³

Queensland University of Technology, School of Electronic Engineering and Computer Science, Brisbane, Australia,
¹noraswaliza@utem.edu.my, ²yue.xu@qut.edu.au, ³s.geva@qut.edu.au

Received 23 August 2012; received in revised form 20 March 2013; accepted 24 March 2013

Abstract

Currently, recommender systems (RS) have been widely applied in many commercial e-commerce sites to help users deal with the information overload problem. Recommender systems provide personalized recommendations to users and, thus, help in making good decisions about which product to buy from the vast amount of product choices. Many of the current recommender systems are developed for simple and frequently purchased products like books and videos, by using collaborative-filtering and content-based approaches. These approaches are not directly applicable for recommending infrequently purchased products such as cars and houses as it is difficult to collect a large number of ratings data from users for such products. Many of the e-commerce sites for infrequently purchased products are still using basic search-based techniques whereby the products that match with the attributes given in the target user's query are retrieved and recommended. However, search-based recommenders cannot provide personalized recommendations. For different users, the recommendations will be the same if they provide the same query regardless of any difference in their interest. In this article, a simple user profiling approach is proposed to generate user's preferences to product attributes (i.e., user profiles) based on user product click stream data. The user profiles can be used to find similar-minded users (i.e., neighbours) accurately. Two recommendation approaches are proposed, namely Round-Robin fusion algorithm (CFRRobin) and Collaborative Filtering-based Aggregated Query algorithm (CFAgQuery), to generate personalized recommendations based on the user profiles. Instead of using the target user's query to search for products as normal search based systems do, the CFRRobin technique uses the attributes of the products in which the target user's neighbours have shown interest as queries to retrieve relevant products, and then recommends to the target user a list of products by merging and ranking the returned products using the Round Robin method. The CFAgQuery technique uses the attributes of the products that the user's neighbours have shown interest in to derive an aggregated query, which is then used to retrieve products to recommend to the target user. Experiments conducted on a real e-commerce dataset show that both the proposed techniques CFRRobin and CFAgQuery perform better than the standard Collaborative Filtering and the Basic Search approaches, which are widely applied by the current e-commerce applications.

Keywords: Collaborative filtering, Recommender systems, Product search, Product recommendation, Personalization, User profiling

1 Introduction

The exponential growth of the World Wide Web (WWW) has changed how we conduct our daily activities. The WWW has become a major source of information and it continues to increase in size and use. One of the popular usages of the WWW is for online shopping, where the buying and selling of products and services are conducted electronically. Nowadays, many commercial e-commerce applications have been developed to sell products and services on the Internet. The users become overwhelmed with the vast amount of information available to them and it is a challenging task to make a final decision about which products to choose. Currently in e-commerce applications, the search-based approach is still widely applied as the common tool for users to search for products. Usually in the standard search engine for an e-commerce website, users are required to specify attribute values of the product that they are looking for as a query. Then, the search engine retrieves a set of products that have attribute values match with the user's query. Although the standard search engine is simple to implement, the search results generated by the standard search engine are not personalized as only products that have the same attribute values or match with the user's query will be displayed to the user. For different users, the recommendations will be the same if they provide the same query no matter how different their online navigation behaviour is. In addition, the user's query may not represent the user's requirements fully because the users may not know the technical details of the products that they want to purchase and thus, very often they are not able to provide accurate or sufficient information in their query to the search engine.

Recommender Systems (RS) have emerged in response to the information overload problem by providing personalized recommendations to users according to their personal interests or preferences [1]. To date, recommender systems have been widely applied by major e-commerce websites, such as Amazon (Site 1), eBay (Site 2), Netflix (Site 4) and Moviefinder.com (Site 3) [13], for recommending various products including books, music CDs or DVDs and to serve millions of consumers [1], [7], [16]. The most widely used recommender system approach is the Collaborative Filtering (CF) technique [4], [5], [17], [20]-[22], which generates personalized recommendations to a user based on the tastes and preferences of similar users. It learns user's preferences by utilizing the user's ratings or past purchase data. The user's preferences or profiles are then used to find a set of users who have similar preferences to a target user which are usually referred to as the *neighbours* of the target user. The products in which the neighbour users have shown interest are then recommended to the target user. The CF approach requires a large amount of ratings data. Therefore, it is more suitable for recommending frequently purchased products because a large amount of ratings can be collected from users when they purchase the products repetitively. However, ratings data is not always available for products that are not often purchased by users during their lifetime, such as cars and houses. Thus, the Collaborative Filtering approach is not directly applicable for recommending these kinds of products. Fortunately, the growth of e-commerce applications provides a platform to gather user's data implicitly. For example, Web search logs store user's online click or browsing history data, which contains useful information about the users and can be analysed to learn about the user's preferences. The implicit data about users can be used by the CF recommender systems when there is no explicit ratings data available as for recommending infrequently purchased products.

For the standard collaborative filtering technique, the products that are preferred by the neighbours will be used as the candidates to generate the recommendations. For recommending frequently purchased products such as books, many copies of the products are available and can be recommended to users. However, for infrequently purchased product search, there is a problem for directly recommending the products that the user's neighbours preferred as what the standard CF method does. For products such as houses or used cars, each product is usually unique, and thus products that previous users have purchased or viewed may be no longer available. Directly recommending products purchased or viewed by previous users becomes meaningless since those products may not exist anymore.

In this paper, we propose to integrate the collaborative filtering and search-based techniques to generate personalized recommendations for infrequently purchased products. In the proposed hybrid approaches, the CF approach is integrated with the search-based approach to recommend products based on the products that the neighbour users have preferred. Rather than directly recommending the neighbour user's preferred products, we propose methods to generate queries based on the neighbour user's profiles or the products viewed by the neighbour users, and then a search to the product collection by using the generated queries is conducted to find the products that may be interested by the target user.

We propose two approaches to combine the CF technique with the search based approach. One approach is to generate a query, which is called Collaborative Filtering-based Aggregated Query (CFAgQuery), by aggregating neighbour user's profiles; the query is then used to retrieve products. For the second approach, the attributes of each product preferred by the user's neighbours are used to form a new query for use by the search-based approach to retrieve similar products. The new query captures neighbour users preferences and provides more detailed content to the query which may be interested but may have been missed by the target user when she/he submits her/his query. Therefore, the product recommendations will be generated based on the new query. In these approaches, multiple queries which are derived from the products preferred by the target user's neighbours are used to retrieve products. This situation is similar to the distributed information retrieval (DIR) system where a user is allowed to

simultaneously access document collections distributed across multiple remote sites. Many different kinds of search engines can be involved in a DIR system and it performs better than an individual search engine because it aggregates the retrieval results from several search engines [14]. The ranked list of documents returned by multiple search engines must be combined in a way that optimizes the performance of the combination since the ranking assigned to documents from one collection is usually not comparable with the ranking from another collection due to the size of the collection and different ranking algorithms employed. Therefore, information fusion is an important issue in the distributed search environment that aims to combine document retrieval results from multiple search engines for improving retrieval effectiveness. As in a DIR system, a data fusion technique to merge the response must be developed for merging the result retrieved by each query. In the second approach, the Round-Robin fusion algorithm [18] is adopted to merge and rank the products retrieved from the multiple queries of each neighbour user. The second approach is therefore named CFRRobin. The two approaches will be discussed in Section 3.

For the collaborative filtering technique, user preference information is essential. Usually, user previous rating data is used to profile a user's item/product preferences. In the case of lacking user rating data, user's online navigation behaviour can be utilized to create user's profiles, which are then used to identify similar users or neighbours of the target user. In this article, we propose a simple method to generate user's product interests/preferences based on user's online navigation log data.

The paper is organized as follows. First, the related work will be briefly reviewed in Section 2. Then, in Section 3 the proposed user profiling approach and the two recommendation approaches will be discussed in detail. Section 4 provides the experiments and evaluation results. Finally, the conclusion will be given in section 5.

2 Related Work

There are two kinds of products offered to users on e-commerce sites –low involvement products (LIP), such as books, videos, soap, and high involvement products (HIP), such as electronic devices, cars, and houses [19]. The Collaborative Filtering (CF) approach has been widely applied for recommending LIP because a rich source of data is available for learning user's preferences and for generating personalized recommendations according to the preference of similar users. The CF approach works best with a large amount of user preferences data and it is suitable for recommending LIP that are frequently purchased by users, as its database of user's preferences gets larger and larger over time when users purchase the products repetitively. Thus, currently, the CF approach has been widely employed by many of the commercial retail websites for recommending LIP.

The CF approach is not directly applicable for recommending HIP as the products are not frequently purchased by the users during their lifetime, and users are not able to provide ratings for products they never use. Currently, many of the e-commerce websites are still implementing the standard search-based approach for recommending HIP in which the user has to specify product attributes as the query, and the user's input is matched with the available products in the database to retrieve products that will most likely be of interest to the user. However, the user's initial query is normally short and does not fully represent the user's requirements. The query expansion approach has been proposed in [2] to expand the user's query based on the associations between the product attribute values extracted from products that have received positive reviews from the previous users. In the literature, recommendations for HIP are also resolved as a product selection problem by using approaches like Case-Based Reasoning and multi-criteria decision analysis. However, in recommending products none of these methods provide personalized recommendations, as they do not predict the user's preferences for use in product recommendations.

The current approaches for recommending HIP requires high involvement from users to provide product attributes that are of interest to them as queries. While, the CF approach, which is widely applied for LIP requires sufficient ratings or purchases history data to generate meaningful recommendations. Methods that can learn users' profiles without the availability of user's ratings or requires high involvement from the users are needed for providing personalized recommendations for the HIP.

The usage of implicit feedback for recommending products has attracted new developments in recommendation algorithms that are suitable for processing implicit feedback. [19] proposed a recommendation methodology for HIP based on the users' profiles, which are generated using the user's past purchases. Their method utilizes the specified user's multi-attributes and preferences from past purchase data for recommending products using the CF approach. Their method assumes that the user has purchased a set of products in the related product category in the past. [6] proposed to transform the implicit user observations into two paired magnitudes, namely, preference and confidence levels. Confidence scores are determined from the frequency of actions, such as the frequency that a user bought a certain item. These confidence scores are attached to the estimated preferences to indicate whether the user's preference is positive or negative. They proposed a latent factor algorithm that addresses the preference-confidence paradigm to tailor for implicit feedback recommendations. [12] incorporated temporal information, such as user purchase time and item launch time, to construct pseudo rating data from the user purchase information for collaborative filtering. Instead of simply assigning 1 to the purchased items, a rating function is defined that computes rating values based on the launch time and purchase time of items to reflect the user's preferences to achieve better recommendation accuracy. Some works derive user's preference to products by analysing user's navigational and behavioural data such as clickstream data [3], [8]-[11], [15]. [9] proposed a Collaborative Filtering based

recommender systems that utilize the preference levels of a user for a product, which are estimated from the navigational and behavioural patterns of users. The preference level of a purchased product is set to 1 and the preference level of a product that is clicked but not purchased is estimated based on the probability of a product being purchased, which is calculated based on certain variables captured in the navigational data such as number of visits, length of reading time, and basket placement status. [10] improved the work in [9] by using association rule mining to generate associations between products and further to derive user's preferences towards products. Similar to Kim's work, the work proposed in this article also generates user's product preference based on user's navigation data. However, a big difference from Kim's work is that, our approach is to generate user's preferences towards product attributes rather than products as a whole. A user may like a product as a whole, but it doesn't mean that the user likes all the features of the product. Two users may both like a product, but they may prefer different aspects of the product. Therefore, the proposed method can profile users more accurately because the profiling is at a more specific level, i.e., product attribute level.

The recommendation algorithms for processing implicit feedback are often studied independently from the domain knowledge. For HIP, the product features are an important factor for the user to consider when making a decision about the final products to buy. The current generation of recommender systems require further improvements to make recommendation methods more effective and applicable to an even broader range of real life applications, which includes recommendations pertaining to more complex types of application [1].

3 Proposed Approaches

In this section, a user profiling method based on user click stream data and the methods to generate product recommendations by combining the collaborative filtering technique and search based technique will be discussed. Before that, we first define the concepts of *product* and *user session* which will be used in this section.

- Product:** Product refers to any type of products or online services for which users can search for information or purchase. Usually a product can be described by a set of attributes which describe the characteristics of the product. For example, attributes for car domain may include *make*, *model*, *year*, *body type*, *price* and *transmission*. Each attribute can have a set of possible values. For example, if an attribute is *body type*, the possible attribute values can be *coupe*, *hatchback*, *sedan* and *wagon*. Suppose that there are n attributes A_1, A_2, \dots, A_n for a product P , each attribute A_i has a set of possible values, $\{a_{i1}, a_{i2}, \dots, a_{im_i}\}$, a product P can be represented by a vector of attribute values, i.e., $P = \langle a_1, a_2, \dots, a_n \rangle$ and $a_i \in \{a_{i1}, a_{i2}, \dots, a_{im_i}\}$, $i = 1, 2, \dots, n$.
- User Session:** A user session S represents a user's online click stream that contains a series of products viewed by the user. Let S be a set of products viewed by a user, i.e., $S = \{p^1, p^2, \dots, p^{|S|}\}$, each product can be represented as a vector of attribute values: $p^k = \langle a_1^k, a_2^k, \dots, a_n^k \rangle$, $k = 1, 2, \dots, |S|$ and $a_i^k \in \{a_{i1}, a_{i2}, \dots, a_{im_i}\}$. Each product can also be represented as a set of attribute values: $p^k = \{A_1 = a_1^k, A_2 = a_2^k, \dots, A_n = a_n^k\}$.

3.1 User Profiling

In this article, *user profiling* is defined as to generate a user profile that represents the user's preference or interest in products or product attributes. User profiles can be exploited by the recommendation generating process to recommend new potentially relevant items to users. User profiling uses a user's data that can be gathered either explicitly or implicitly from the user. Explicit data such as ratings, demographic information, reviews etc. must be provided by the user. In some circumstances, not many users are willing to provide these data. For instance, for infrequently purchased products such as cars and houses, the explicit data may not be sufficiently gathered from users as users only possess a few items during their lifetime, and thus they will not be able to give ratings for many products. It is crucial to understand a user's preferences implicitly from the user's data and provides personalized recommendations without much participation from the users. Click streams data is a kind of search log that could be collected by the search engine implicitly without user extra effort. Click stream data shows the path a user takes through a website. From the user's click streams data, a list of products that have been viewed by the user can be obtained. This online click streams data shows that the user has more interest in the viewed products compared to the other products. By analyzing the entire user's preferred products attribute values gathered from the user click streams data, the user's interests or preferences to each product attribute value can be predicted.

Let $S = \{P^1, P^2, \dots, P^{|S|}\}$ be a user session. For a product p^k in the session, $p^k = \{A_1 = a_1^k, A_2 = a_2^k, \dots, A_n = a_n^k\}$ where $a_i^k \in \{a_{i1}, a_{i2}, \dots, a_{im_i}\}$ is the value for attribute A_i , the product can be represented as a transaction of all attribute values:

$$\{a_{11}=0, a_{12}=0, \dots, a_{1k}^k=1, \dots, a_{1m_1}=0, \dots, a_{n1}=0, \dots, a_n^k=1, \dots, a_{nm_n}=0\} \quad (1)$$

This transaction represents that for the product p^k , the value of attribute A_1 is a_1^k , the value of attribute A_2 is a_2^k , etc. From a set of products viewed by a user, a product transaction dataset of $|S|$ transactions can be constructed for the user, and each product p^k in S can be represented as

$$p^k = \langle a_{11}=0, a_{12}=0, \dots, a_{1k}^k=1, \dots, a_{1m_1}=0, \dots, a_{n1}=0, \dots, a_n^k=1, \dots, a_{nm_n}=0 \rangle \quad (2)$$

From the transaction dataset, the frequency $freq(a_{ij})$ of each attribute value a_{ij} for attribute A_i can be obtained by counting the number of transactions that have $a_{ij}=1$. In this paper, we propose to represent a user's product interests by using the frequency of the attribute values of the products viewed by the user. The more frequent an attribute value, the higher the user is interested in that attribute value. A user profile can be represented as:

$$up = \langle ua_{11}, \dots, ua_{1m_1}, ua_{21}, \dots, ua_{2m_2}, \dots, ua_{nm_n} \rangle \quad (3)$$

$$ua_{ij} = \frac{freq(a_{ij})}{|S|}, \sum_{j=1}^{m_i} ua_{ij} = 1$$

where ua_{ij} denotes the user's interest/preference to the j th value of attribute A_i . It shows the user preference strength for each attribute value a_{ij} of attribute A_i among all the products preferred by the user. Table 1 shows an example of a transaction dataset for a user session. Assume that the user has viewed five products, i.e. $S = \{p^1, p^2, \dots, p^5\}$, each product p^k has three attributes, i.e. A_1, A_2, A_3 , A_1 has 4 values, i.e. $A_1 = \{a_{11}, a_{12}, a_{13}, a_{14}\}$, A_2 has 5 values, i.e. $A_2 = \{a_{21}, a_{22}, a_{23}, a_{24}, a_{25}\}$ and A_3 has 3 values, i.e. $A_3 = \{a_{31}, a_{32}, a_{33}\}$. For each product p^k , if the product has attribute value a_{ij} , the cell is assigned as 1, otherwise 0 in the transaction dataset.

The user profile for this product transaction is shown below:

$$up = \langle 0.6, 0.4, 0, 0, 0, 0, 0.8, 0, 0.2, 0, 0.6, 0.4 \rangle \quad (4)$$

Table 1: An example of the product transaction for a user

| | A_1 | | | | A_2 | | | | | A_3 | | |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | a_{11} | a_{12} | a_{13} | a_{14} | a_{21} | a_{22} | a_{23} | a_{24} | a_{25} | a_{31} | a_{32} | a_{33} |
| p^1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| p^2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| p^3 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| p^4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| p^5 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

3.2 Integrating the CF Approach and the Search-based Approach Using Query Aggregation

The standard search-based systems provide basic search function that takes the user's query as input and returns a set of matched products to the query. Usually, a user is required to provide some attributes values of the product that she/he is looking for, as a query in the search form. This query is normally short and may not reflect the user's requirements fully. In addition, many users do not have sufficient knowledge about the product they want to find and they cannot provide detailed requirements of the attributes or features of the product. Therefore, the attributes in the query may not be the right attributes to query or inadequate to represent the user's preferences. In this sub section, a new method is proposed to generate an aggregated query for a target user based on the target user's initial query and the preferences of the target user's neighbour users.

Neighbourhood formation is a key component of the collaborative filtering approach in which a set of like-minded users or neighbours for a target user is generated. The K -Nearest-Neighbourhood formation approach is adopted in this article to select the top K neighbours which are the most similar users to the target user based on their profiles.

The similarity measure can be calculated by using one of the proximity computing approaches such as cosine similarity and Pearson correlation [1].

Let $up = \langle ua_{11}, \dots, ua_{1m_1}, \dots, ua_{n1}, \dots, ua_{nm_n} \rangle$ be the target user's profile, $\{B_1, B_2, \dots, B_{\mathfrak{R}}\}$ be the target user's neighbours and $pu^k = \langle ua_{11}^k, \dots, ua_{1m_1}^k, \dots, ua_{n1}^k, \dots, ua_{nm_n}^k \rangle$ be the user profile of neighbour B_k . By combining $pu^1, pu^2, \dots, pu^{\mathfrak{R}}$ for all the neighbours, we can generate an aggregated profile $u^{ag} = \langle ua_{11}^{ag}, \dots, ua_{1m_1}^{ag}, ua_{21}^{ag}, \dots, ua_{2m_2}^{ag}, \dots, ua_{nm_n}^{ag} \rangle$ for the target user u . Each attribute value ua_{ij}^{ag} in the aggregated query is calculated using the following equation:

$$ua_{ij}^{ag} = \frac{\sum_{k=1}^{\mathfrak{R}} (sim(u, B_k) * ua_{ij}^k)}{\sum_{k=1}^{\mathfrak{R}} sim(u, B_k)} \quad (5)$$

where $sim(u, B_k)$ is the similarity between u and its neighbour B_k , which can be calculated using the cosine similarity described as below:

$$sim(u, B_k) = \frac{\sum_{i=1}^n \sum_{j=1}^{m_i} ua_{ij} * ua_{ij}^k}{\sqrt{\sum_{i=1}^n \sum_{j=1}^{m_i} (ua_{ij})^2} \sqrt{\sum_{i=1}^n \sum_{j=1}^{m_i} (ua_{ij}^k)^2}} \quad (6)$$

$ua_{k1}^{ag}, \dots, ua_{km_k}^{ag}$ measure the preference strength of the target user to each attribute value of attribute A_k based on the viewpoints of the target's user's neighbours. It is easy to prove that $\sum_{j=1}^{m_k} ua_{kj}^{ag} = 1$. By choosing the attribute value with the highest preference for each attribute, an aggregated query $AQ^u = \{A_1 = a_1^{ag}, A_2 = a_2^{ag}, \dots, A_n = a_n^{ag}\}$ can be generated, where $a_k^{ag} = \max_{j=1}^{m_k} (ua_{kj}^{ag})$. Then, by doing a search to the product database, a list of products, denoted as Γ , that match the aggregated query AQ^u are retrieved as candidate products for the target user. Figure 1 illustrates the process of generating the aggregated query.

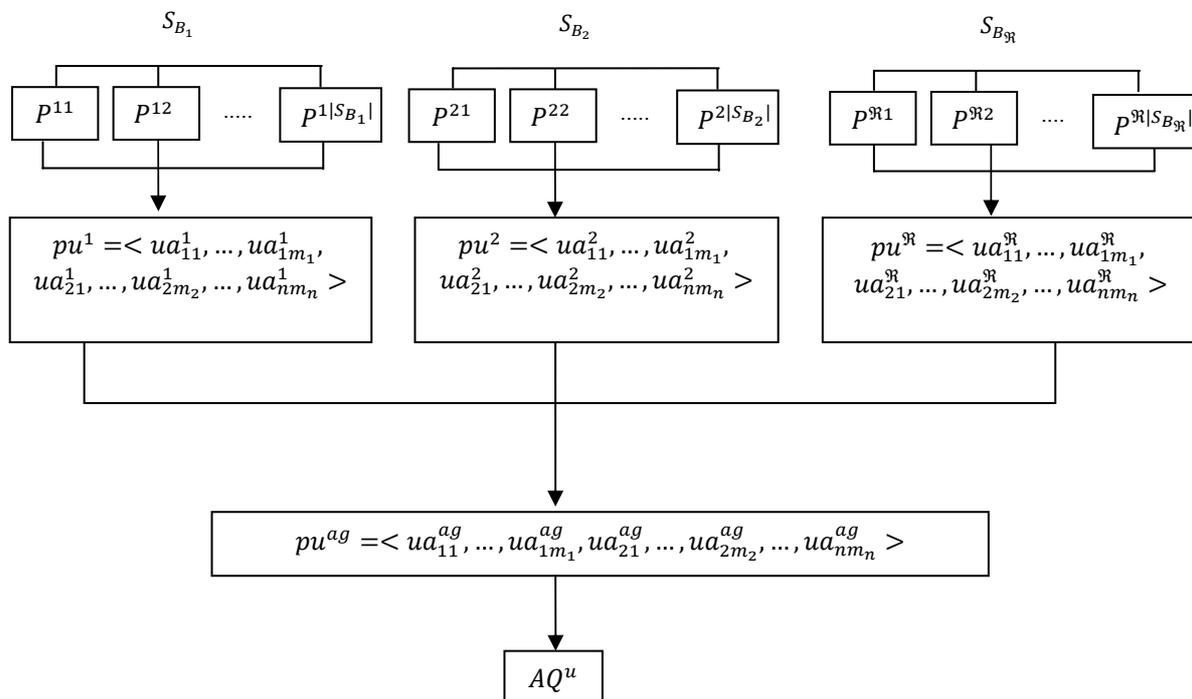


Figure 1: The profile aggregation of the CFAgQuery approach

3.3 Integrating the CF Approach and the Search-based Approach Using the Round Robin Fusion Method

This approach uses each product of the user's neighbours as a query to retrieve other relevant products. Each query retrieves a set of relevant products and the retrieved products from all the queries are merged by employing the

Round Robin method [18] to generate a set of candidate products. The products from all the candidate product sets of all the neighbours are then ranked and final products are selected for recommendation.

Let $S_{B_i} = \{p^{i1}, p^{i2}, \dots, p^{i|S_{B_i}|}\}$ be a set of products viewed by a neighbor B_i . Instead of using the products p^{ij} in S_{B_i} as the candidates for recommendations, the attribute values of each of the products, i.e. $p^{ij} = \langle a_1^{ij}, \dots, a_n^{ij} \rangle$ are used as a query Q^{ij} to retrieve products. That is, $\forall p^{ij} \in S_{B_i}, Q^{ij} = \{A_1 = a_1^{ij}, A_2 = a_2^{ij}, \dots, A_n = a_n^{ij}\}$ is a query containing the attributes of the product p^{ij} that the neighbour B_i is interested in. A set of products, $\{b_1^{ij}, b_2^{ij}, \dots\}$, whose attributes match with the attributes in Q^{ij} are retrieved and also ranked based on the similarity $sim(b_k^{ij}, Q^{ij})$ between the products b_k^{ij} and the query Q^{ij} . Generally, the attribute values are not necessarily numerical values, they can be nominal attributes. For numerical attributes, the cosine similarity can be used to measure the similarity. For nominal attributes, let $b_k^{ij} = \{A_1 = a_1, A_2 = a_2, \dots, A_n = a_n\}$, the following method can be used to measure the similarity:

$$sim(b_k^{ij}, Q^{ij}) = \sum_{l=1}^n sim_A(a_l, a_l^{ij}) \quad (7)$$

$$sim_A(a_l, a_l^{ij}) = \begin{cases} 1, & a_l == a_l^{ij} \\ 0, & a_l \neq a_l^{ij} \end{cases}$$

For each product $p^{ij} \in S_{B_i}$ viewed by the user neighbour B_i , based on the similarity, a list of ranked products can be generated, $L^{ij} = \langle b_1^{ij}, b_2^{ij}, \dots, b_r^{ij} \rangle$, where $sim(b_1^{ij}, Q^{ij}) > sim(b_2^{ij}, Q^{ij}) > \dots > sim(b_r^{ij}, Q^{ij})$. Therefore, from the neighbour B_i , $|S_{B_i}|$ lists of products are generated: $L^{i1}, L^{i2}, \dots, L^{i|S_{B_i}|}$. All the products in these lists are similar to the products preferred by B_i in terms of the product attributes. The similarity value $sim(b_k^{ij}, Q^{ij})$ for a product in different list L^{ij} is based on different Q^{ij} , and thus the products in all the lists cannot be simply ranked based on the similarity value $sim(b_k^{ij}, Q^{ij})$ to select the candidate products. The Round Robin method is a simple data fusion technique that is adopted to merging and selecting final products from $L^{i1}, L^{i2}, \dots, L^{i|S_{B_i}|}$ retrieved by the multiple queries of each neighbour B_i . By applying the Round Robin method to the lists, all the products in $L^{i1} \cup L^{i2} \cup \dots \cup L^{i|S_{B_i}|}$ can be ranked to select the candidate products for each neighbour. The Round Robin method selects a product from the top of each L^{ij} for each round, and then starts again from the top of the list for the remaining products in each L^{ij} . From the ranked products in $L^{i1} \cup L^{i2} \cup \dots \cup L^{i|S_{B_i}|}$, the top N products are chosen as the candidates generated from neighbour B_i , denoted as C_{B_i} . Thus, by combining the products in C_{B_i} for all neighbours, we obtain a set of candidate products. $\Gamma = \bigcup_{i=1}^N C_{B_i}$. Figure 2 shows the candidate product sets generated based on all the target user's neighbours.

3.4 Product Ranking

The final process is to rank the products in the candidate list Γ and to select the Top N products to recommend. The products are ranked based on the similarities between each product and the target user's interests. Let the target user's profile be $up = \langle ua_{11}, \dots, ua_{1m_1}, ua_{21}, \dots, ua_{2m_2}, \dots, ua_{nm_n} \rangle$ which is generated from the target user's online click streams data, by choosing the attribute value with the highest preference for each attribute, we can generate the target user's preferred attribute values:

$$Q^u = \{A_1 = a_1^u, A_2 = a_2^u, \dots, A_n = a_n^u\}, a_i^u = \max_{j=1}^{m_i} (ua_{ij}), i = 1, \dots, n \quad (8)$$

Let Γ be the set of candidate products generated by the CFRRobin, or CFAgQuery, $b_k \in \Gamma$ and $b_k = \{A_1 = a_1, A_2 = a_2, \dots, A_n = a_n\}$, the similarity between b_k and Q^u , denoted as $sim(b_k, Q^u)$, is used to rank the products in Γ . The similarity $sim(b_k, Q^u)$ can be calculated using Equation (6). Finally, the top N products are selected as the final products to be recommended from the ranked products.

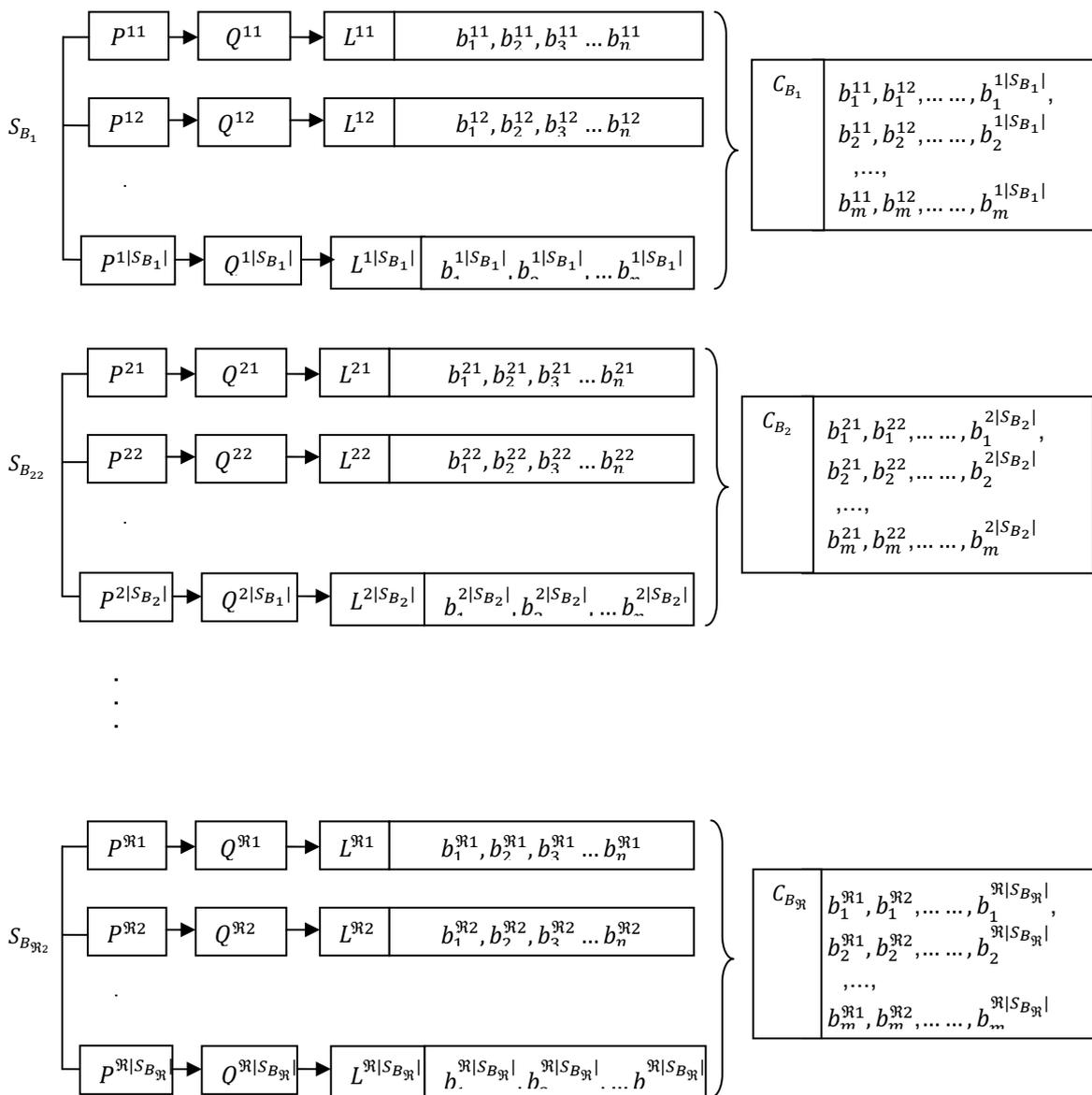


Figure 2: The product retrieval and merging for the CFRRobin approach

4 Experiments and Evaluation

This section focuses on the evaluation of the proposed user profiling and recommendation models. Firstly, the experiment design and the evaluation methods will be given. Then, the results of experiments will be discussed and illustrated.

4.1 Experiment Design

The experiments were conducted to see how the proposed user profiling and recommendation approaches perform by comparing to the baseline approaches. The experiments were conducted in order to verify the following working hypothesis:

H1: The integration of collaborative filtering approach and the search-based approach can generate more accurate recommendations compared to only collaborative filtering or search-based approach.

4.1.1 Datasets

A case study has been conducted for the car online selling domain. Data was collected from a well known company in Australia that sells cars online. The dataset contains 17,690 cars and 20,868 user navigation sessions. Cars data

contains information about the cars available in the company's database. User navigation session data is generated from the company's website search log by which each user session is generated from a sequence of cars viewed by a user. In the experiment, each session will be divided into two parts in which each part must contain at least 2 cars. Thus, only sessions with at least four viewed cars are selected for the experiments. The final dataset contains 3564 user sessions.

4.1.2 Evaluation Metrics

The evaluation uses some metrics to assess which of the system performs better. In this paper, Precision, Recall and F1 Measurement metrics are used to evaluate the performance of the proposed models.

- *Precision and Recall:* Precision and recall that are proposed by Cleverdon et al. in 1966 are the most popular metrics used for evaluating information retrieval systems. Precision measures the ability of the system to present only those items that are relevant, and it can be seen as the measure of exactness. Precision is defined as the ratio of the retrieved items that are relevant (NM) and the number of all retrieved items (NR) shown in Equation 9:

$$Precision = \frac{NM}{NR} \quad (9)$$

Recall measures the ability of the system to present all the relevant items and it can be seen as the measure of completeness. Recall is defined as the ratio of the retrieved items that are relevant (NM) and the number of items that should be returned (NT) shown in Equation 10:

$$Recall = \frac{NM}{NT} \quad (10)$$

To evaluate the proposed models for online car search, NM is the number of retrieved cars that match with the testing cars, NR is the number of retrieved cars, and NT is the number of testing cars in the testing session. The precision and recall are calculated for each session or user and the average recall and precision for all sessions (i.e. all users) were calculated for each search model.

- *F1 Measure:* The F1 Measure was first introduced by Van Rijsbergen in 1979. F1 metric is used to provide a general overview of the overall performance. The F1 measure combines the recall and precision results with an equal weight in the following form:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (11)$$

4.1.3 Experiment Setup

In order to evaluate the effectiveness of the proposed user profiling approach and the recommendation approaches, this paper implements the proposed user profiling, recommendation approaches, and the baseline models. php was used as the programming language to implement the system. The proposed approaches include:

- *CFAgQuery:* This recommendation approach integrates the collaborative filtering and search-based approaches. It generates an aggregated query based on products viewed by the neighbour users. Then, the retrieved products are ranked based on the similarity to the target user profile.
- *CFRRobin:* This recommendation approach integrates the collaborative filtering and search-based approaches by using each product of the neighbour users as a query. This approach implements the Round Robin data fusion to merge products retrieved by multiple queries and ranks the final products based on the similarity to the target user profile.

The baseline models include:

- *BS:* BS refers to the Basic Search approach that retrieves products that match with the user's query. Currently, many e-commerce sites for selling infrequently purchased products only provide standard search engines that retrieve products based on the query given by the user.
- *CFOriginal:* CFOriginal refers to the Original Collaborative filtering approach which is popularly used for recommending products. This approach finds users with similar interests with the target user and directly recommends products that are preferred by the similar users to the target user.

The user session dataset was partitioned into 5 sub datasets. Each of them (20% of user sessions) was used as a testing dataset and the remaining part was used as training data. Each session in the testing dataset was further

divided into two parts evenly. As a result, the session dataset contains three parts – Training, Testing Part 1 and Testing Part 2, as illustrated in Figure 3.

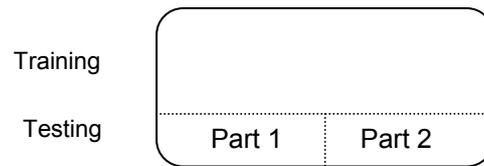


Figure 3: The division of session dataset for the experiment

Sessions in the Testing dataset were considered as target users and the cars listed in Testing Part 1 in each session were considered as cars viewed by the target user of that session. Besides, Testing Part 2 was used as testing data to evaluate whether or not the recommended cars generated by the recommender models match the cars in the Testing Part 2. For the BS, the last car of each session in Testing Part 1 was used as the query to search for cars. The CFOOriginal, CFAgQuery, and CFRRobin models generate the target user's profile based on the data in Testing Part 1 by using the user profiling method discussed in Section 3.1. Sessions in the Training dataset were considered as previous users. Training dataset was used to generate previous user's profiles which will be used to find neighbours by using the neighbourhood formation method. For each experiment, there will be 5 runs. Finally, the average result for the 5 runs will be calculated.

The experiments were conducted to test if the proposed methods, i.e. CFAgQuery, and CFRRobin outperform the baseline models, i.e. BS and CFOOriginal. In addition, the experiments also test the impact of using different user profiles created from different amount of viewed products in the target user's click data for the CFOOriginal, CFAgQuery, and CFRRobin. The purpose is to investigate whether the most recently viewed cars are more important than previously viewed cars in generating accurate user profiles. To this end, four user profiles named UT_1 , UT_2 , UT_3 and UT_4 are generated using the last viewed car, the last 2 viewed cars, the last 3 viewed cars and the last 4 viewed cars by the target user, respectively. For the BS model which does not utilize user profiles, only the last car is used as the query to retrieve relevant cars. Table 2 lists all different runs in the experiments. Figure 4 illustrates the generation of different user profiles from the testing sessions.

The purpose of product searching is to provide users with the products that meet user's requirements on product attributes or features. In this experiment, to evaluate all the models, for each session, if at least 80% of the attributes of a retrieved car match with the attributes of one of the cars in the same session of the Testing Part 2, the retrieved car was considered as matching with the testing car. The focus of this experiment is to recommend cars that match the attribute values preferred by the user. Thus cars with different IDs but have the same attributes might be recommended.

4.2 Results and Discussion

Results of recommendations based on user profiles and by combining the collaborative filtering and search-based approaches will be examined and compared.

The objective of this set of experiments is to verify that the integration of collaborative filtering and search-based approaches can generate more accurate recommendations compared to only collaborative filtering or search-based approach (Hypothesis). For each run of the experiment, the results for recommending different number of top N cars are recorded. The precision results are given in Table 3 to Table 6, the recall results are given in Table 7 to Table 10, and the Table 11 to Table 14 are the results of F1 measure.

The precision results show that our proposed approaches, CFRRobin and CFAgQuery, perform better than the two baseline models BS and CFOOriginal. The precision results for the CFAgQuery and CFRRobin models are quite similar for the profile generated from the last car. For the profiles generated from more cars, the precisions for the CFAgQuery model are better than the precisions for the CFRRobin model.

Table 2: Different runs of the experiments

| Models | Runs | Target User Profiles |
|--|-----------------|----------------------|
| Basic Search(BS) | BS | UT_1 |
| Original Collaborative Filtering (CFOriginal) | CFOriginal1Cars | UT_1 |
| | CFOriginal2Cars | UT_2 |
| | CFOriginal3Cars | UT_3 |
| | CFOriginal4Cars | UT_4 |
| Collaborative Filtering with Query Aggregation (CFAgQuery) | CFAgQuery1Cars | UT_1 |
| | CFAgQuery2Cars | UT_2 |
| | CFAgQuery3Cars | UT_3 |
| | CFAgQuery4Cars | UT_4 |
| Collaborative Filtering with Round Robin (CFRRRobin) | CFRRRobin1Cars | UT_1 |
| | CFRRRobin2Cars | UT_2 |
| | CFRRRobin3Cars | UT_3 |
| | CFRRRobin4Cars | UT_4 |
| | CFMRRRobin2Cars | UT_2 |
| | CFMRRRobin3Cars | UT_3 |
| | CFMRRRobin4Cars | UT_4 |

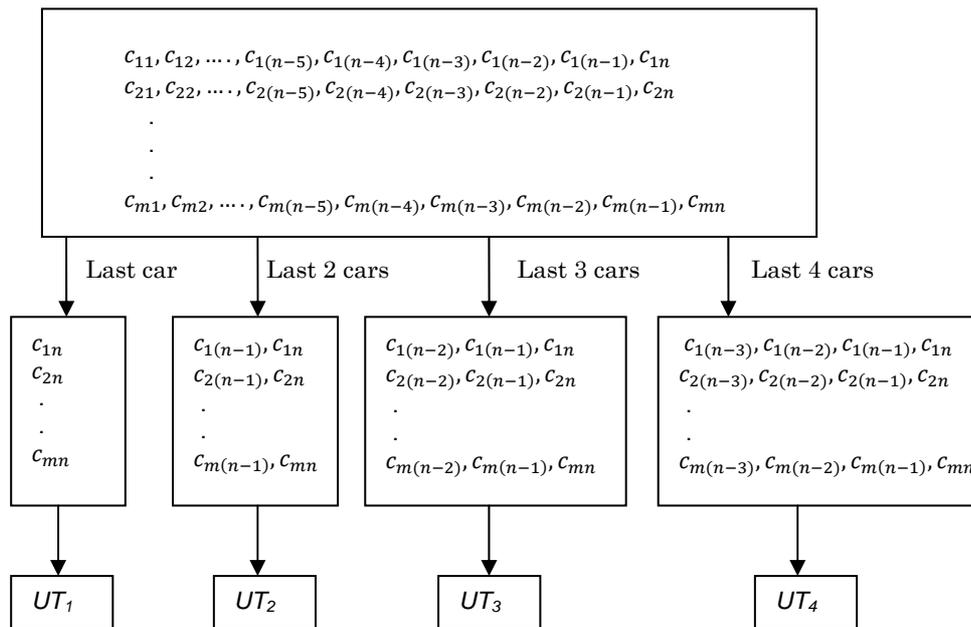


Figure 4: The generation of user profiles

Table 3: Precision results of different models for user profile UT_1

| | Top5 | Top 10 | Top 15 | Top 20 | Top 25 | Top 30 |
|-----------------|------|--------|--------|--------|--------|--------|
| BS | 0.41 | 0.43 | 0.43 | 0.42 | 0.42 | 0.42 |
| CFOriginal1Cars | 0.08 | 0.07 | 0.07 | 0.06 | 0.05 | 0.05 |
| CFRRRobin1Cars | 0.64 | 0.64 | 0.62 | 0.61 | 0.60 | 0.59 |
| CFAgQuery 1Cars | 0.64 | 0.63 | 0.62 | 0.61 | 0.60 | 0.59 |

Table 4: Precision results of different models for user profile UT_2

| | Top 5 | Top10 | Top 15 | Top 20 | Top 25 | Top 30 |
|-----------------|-------|-------|--------|--------|--------|--------|
| BS | 0.41 | 0.43 | 0.43 | 0.42 | 0.42 | 0.42 |
| CFOriginal2Cars | 0.08 | 0.06 | 0.05 | 0.05 | 0.05 | 0.04 |
| CFRRobin2Cars | 0.60 | 0.60 | 0.60 | 0.58 | 0.57 | 0.56 |
| CFAgQuery2Cars | 0.61 | 0.61 | 0.60 | 0.59 | 0.59 | 0.58 |

Table 5: Precision results of different models for user profile UT_3

| | Top5 | Top 10 | Top 15 | Top 20 | Top 25 | Top 30 |
|-----------------|------|--------|--------|--------|--------|--------|
| BS | 0.41 | 0.43 | 0.43 | 0.42 | 0.42 | 0.42 |
| CFOriginal3Cars | 0.10 | 0.09 | 0.08 | 0.07 | 0.06 | 0.06 |
| CFRRobin3Cars | 0.59 | 0.59 | 0.58 | 0.57 | 0.55 | 0.54 |
| CFAgQuery3Cars | 0.60 | 0.60 | 0.59 | 0.59 | 0.58 | 0.56 |

Table 6: Precision results of different models for user profile UT_4

| | Top5 | Top 10 | Top 15 | Top 20 | Top 25 | Top 30 |
|-----------------|------|--------|--------|--------|--------|--------|
| BS | 0.41 | 0.43 | 0.43 | 0.42 | 0.42 | 0.42 |
| CFOriginal4Cars | 0.12 | 0.09 | 0.08 | 0.07 | 0.06 | 0.06 |
| CFRRobin4Cars | 0.54 | 0.54 | 0.53 | 0.52 | 0.51 | 0.50 |
| CFAgQuery4Cars | 0.58 | 0.58 | 0.58 | 0.57 | 0.57 | 0.55 |

Table 7: Recall results of different models for user profile UT_1

| | Top 5 | Top 10 | Top 15 | Top 20 | Top 25 | Top 30 |
|--------------------|-------|--------|--------|--------|--------|--------|
| BS | 0.27 | 0.30 | 0.30 | 0.30 | 0.30 | 0.30 |
| CFOriginal1Cars | 0.06 | 0.09 | 0.11 | 0.13 | 0.14 | 0.17 |
| CFRRobin1Cars | 0.28 | 0.28 | 0.29 | 0.30 | 0.30 | 0.31 |
| CFAgQuery 1Cars | 0.27 | 0.28 | 0.29 | 0.29 | 0.29 | 0.30 |

The recall results show that, averagely the CFRRobin and CFAgQuery models outperform the CFOriginal model and achieve very close results as the BS model. For profiles created from the last 2 and 3 cars, both the CFRRobin and CFAgQuery models perform better than the baseline models, BS and the standard CF. However, for profiles created from the last one car (Table 6) or the last 4 cars (Table 9), the recall results for the CFRRobin and CFAgQuery models are not always better than the BS model.

An investigation has been conducted to identify the users with diverse attribute interests and users with focused attribute interests. It was found that, many users in the testing data may have diverse interests, in which the users may look at products with different attribute values. The similarity value for each pair of cars viewed in each session was calculated based on the car's attribute values. , the average attribute similarity values of all pairs of cars viewed in each session were calculated. The result of this investigation reveals that there are more users with diverse products (1970 users whose car similarity less than 0.5) compared to the users with focused products in the dataset (1593 users whose car similarity more than or equal to 0.5). If the products in the testing data are diverse where they have different attributes from each other, not all products with different attribute values can be retrieved by the proposed approaches. The CFAgQuery generates a new query based on the attribute values that are most interested by the neighbour users and therefore may return products that are more focused in terms of attribute values and thus, might not match with other products with different attribute values in the testing data. The CFRRobin utilizes each product of the neighbours' products as a query to retrieve other similar products to the neighbours' products and the final products are selected by matching the retrieved products with the user's profile. Thus, there are more possibilities for the CFRRobin approach to recommend more products with focused attribute

values compared to the BS. On the contrary, the BS only uses small number of attributes from the initial queries to retrieved products and thus, products that have some attributes values match with the query's attribute values can be retrieved. As a result, the products retrieved by the BS may be a bit more diverse. Moreover, the CFRRobin performs better than the CFAGQuery because this approach recommends products that are similar to the products interested by the neighbour users and do not use a single query to retrieve products as implemented by the CFAGQuery. Thus, it recommends more diverse products compared to the CFAGQuery. Therefore, in practice, if the business model needs to promote various products to users, CFRRonin would be a better approach to use than CFAGQuery. Moreover, if user data, e.g., user searching logs or user product ratings, is available, it is preferable to use the proposed approaches CFAGQuery and CFRRobin or the standard collaborative filtering approach rather than use the basic search algorithm. However, for a newly created Website which doesn't have much data about users, the basic search algorithm might be better to use as the CFOriginal, CFAGQuery and CFRRobin approaches rely on previous user data to generate recommendations.

Table 8: Recall results of different models for user profile UT_2

| | Top5 | Top 10 | Top 15 | Top 20 | Top 25 | Top 30 |
|-----------------|------|--------|--------|--------|--------|--------|
| BS | 0.27 | 0.30 | 0.30 | 0.30 | 0.30 | 0.30 |
| CFOriginal2Cars | 0.06 | 0.07 | 0.09 | 0.10 | 0.12 | 0.13 |
| CFRRobin2Cars | 0.29 | 0.30 | 0.31 | 0.33 | 0.34 | 0.35 |
| CFAGQuery2Cars | 0.29 | 0.30 | 0.30 | 0.31 | 0.31 | 0.31 |

Table 9: Recall results of different models for user profile UT_3

| | Top5 | Top 10 | Top 15 | Top 20 | Top 25 | Top 30 |
|-----------------|------|--------|--------|--------|--------|--------|
| BS | 0.27 | 0.30 | 0.30 | 0.30 | 0.30 | 0.30 |
| CFOriginal3Cars | 0.09 | 0.11 | 0.11 | 0.12 | 0.13 | 0.14 |
| CFRRobin3Cars | 0.29 | 0.30 | 0.30 | 0.32 | 0.34 | 0.34 |
| CFAGQuery3Cars | 0.28 | 0.29 | 0.30 | 0.30 | 0.32 | 0.32 |

Table 10: Recall results of different models for user profile UT_4

| | Top5 | Top 10 | Top 15 | Top 20 | Top 25 | Top 30 |
|-----------------|------|--------|--------|--------|--------|--------|
| BS | 0.27 | 0.30 | 0.30 | 0.30 | 0.30 | 0.30 |
| CFOriginal4Cars | 0.12 | 0.12 | 0.13 | 0.14 | 0.14 | 0.16 |
| CFRRobin4Cars | 0.27 | 0.30 | 0.30 | 0.33 | 0.34 | 0.36 |
| CFAGQuery4Cars | 0.28 | 0.29 | 0.29 | 0.29 | 0.31 | 0.31 |

The F1 Measure results of both the CFRRobin model and the CFAGQuery are better than the two baseline models, BS and CFOriginal, for all the profiles. These results can verify the *Hypothesis* that the integration of collaborative filtering approach and search-based approach can generate more accurate recommendations compared to only collaborative filtering or search-based approach.

Table 11: F1 Results of different models for user profile UT_1

| | Top5 | Top 10 | Top 15 | Top 20 | Top 25 | Top 30 |
|-----------------|-------|--------|--------|--------|--------|--------|
| BS | 0.33 | 0.35 | 0.35 | 0.35 | 0.35 | 0.35 |
| CFOriginal1Cars | 0.069 | 0.079 | 0.086 | 0.082 | 0.074 | 0.077 |
| CFRRobin1Cars | 0.39 | 0.39 | 0.40 | 0.40 | 0.40 | 0.41 |
| CFAGQuery1Cars | 0.38 | 0.39 | 0.40 | 0.39 | 0.39 | 0.40 |

Table 12: F1 Results of different models for user profile UT_2

| | Top5 | Top 10 | Top 15 | Top 20 | Top 25 | Top 30 |
|-----------------|------|--------|--------|--------|--------|--------|
| BS | 0.33 | 0.35 | 0.35 | 0.35 | 0.35 | 0.35 |
| CFOriginal2Cars | 0.07 | 0.06 | 0.06 | 0.07 | 0.07 | 0.06 |
| CFRRobin2Cars | 0.39 | 0.40 | 0.41 | 0.42 | 0.43 | 0.43 |
| CFAgQuery2Cars | 0.39 | 0.40 | 0.40 | 0.41 | 0.41 | 0.40 |

Table 13: F1 Results of different models for user profile UT_3

| | Top5 | Top 10 | Top 15 | Top 20 | Top 25 | Top 30 |
|-----------------|------|--------|--------|--------|--------|--------|
| BS | 0.33 | 0.35 | 0.35 | 0.35 | 0.35 | 0.35 |
| CFOriginal3Cars | 0.09 | 0.10 | 0.09 | 0.09 | 0.08 | 0.08 |
| CFRRobin3Cars | 0.39 | 0.40 | 0.40 | 0.41 | 0.42 | 0.42 |
| CFAgQuery3Cars | 0.38 | 0.39 | 0.40 | 0.40 | 0.41 | 0.41 |

Table 14: F1 Results of different models for user profile UT_4

| | Top5 | Top 10 | Top 15 | Top 20 | Top 25 | Top 30 |
|-----------------|------|--------|--------|--------|--------|--------|
| BS | 0.33 | 0.35 | 0.35 | 0.35 | 0.35 | 0.35 |
| CFOriginal4Cars | 0.12 | 0.10 | 0.10 | 0.09 | 0.08 | 0.09 |
| CFRRobin4Cars | 0.36 | 0.39 | 0.38 | 0.40 | 0.41 | 0.42 |
| CFAgQuery4Cars | 0.38 | 0.39 | 0.39 | 0.38 | 0.40 | 0.40 |

5 Conclusion

In this article, we investigated the methods for recommending infrequently purchased products by integrating collaborative filtering techniques and search-based techniques. We utilize user's online click stream data to learn user's preferences for creating user's profiles. Two methods are proposed, which are the CFRRobin and CFAgQuery. Both methods generate the target user's product preferences (i.e., the target user profile) based on the target user's neighbours' preferences using a frequent-based technique. Instead of directly recommending the products that the user's neighbours have liked as implemented in the standard collaborative filtering, the proposed methods search the product dataset by using the generated target user's profile as a query for products which match the target user's preferences. The experiment results show that the proposed methods perform better than the Basic Search (BS), and the Original Collaborative Filtering (CFOriginal) models.

For future work, instead of using frequent-based technique to create user's profiles, we intend to apply model-based techniques such as probabilistic latent semantic analysis to learn the user's preferences and to create the user's profiles in order to improve the recommendations generated by the proposed methods.

Websites List

Site 1: Amazon
www.amazon.com

Site 2: eBay
www.eBay.com

Site 3: Movie Finder
moviefinder.com.au/

Site 4: Netflix
<https://www.netflix.com>

References

- [1] G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, 2005.
- [2] N. Abdullah, Y. Xu, S. Geva, and M. Looi, Enhancement of infrequent purchased product recommendation using data mining techniques, in *Proceedings of International Federation for Information Processing, World Computer Congress, Brisbane, Australia, 2010*, pp. 57-66.
- [3] M. Claypool, P. Le, M. Wased, and D. Brown, Implicit interest indicators, in *Proceedings of the 6th International Conference on Intelligent User Interfaces, Santa Fe, New Mexico, USA, 2001*, pp. 33-40.
- [4] Y. Ge, H. Xiong, A. Tuzhilin, and Q. Liu, Collaborative filtering with collective training, in *Proceedings of the 5th ACM Conference on Recommender Systems, New York, USA, 2011*, pp. 281-284.
- [5] M. Gori and A. Pucci, Itemrank: A random-walk based scoring algorithm for recommender engines, in *Proceedings of the 20th International Joint Conference on Artificial Intelligence, San Francisco, USA, 2007*, pp. 2766-2771.
- [6] Y. Hu, Y. Koren, and C. Volinsky, Collaborative filtering for implicit feedback datasets, in *Proceeding of the 8th IEEE International Conference on Data Mining, Washington, USA, 2008*, pp. 263-272.
- [7] T. Iwata, Kazumi Saito, and Takeshi Yamada, Recommendation method for improving customer lifetime value, *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 9, pp. 1254-1263, 2008.
- [8] D. Kelly and J. Teevan, Implicit feedback for inferring user preference: A bibliography, *ACM Special Interest Group on Information Retrieval Forum*, vol. 37, no. 2, pp. 18-28, 2003.
- [9] Y. S. Kim, B.J. Yum, J. S. Song, and S. M. Kim, Development of a recommender system based on navigational and behavioural patterns of customers in e-commerce sites, *Journal of Expert Systems with Applications*, vol. 28, no. 2, pp. 381-393, 2005.
- [10] Y. S. Kim and B.J. Yum, Recommender system based on click stream data using association rule mining, *Journal of Expert Systems with Applications*, vol. 38, no. 10, pp. 13320-13327, 2011.
- [11] J. Lee, M. Podlaeck, E. Schonberg, and R. Hoch, Visualization and analysis of click stream data of online stores for understanding web merchandising, *Data Mining and Knowledge Discovery*, vol. 5, no. 1/2, pp. 59-84, 2001.
- [12] T.Q. Lee, Y. Park, and Y.T. Park, A time-based approach to effective recommender systems using implicit feedback, *Expert System with Applications*, vol. 34, no. 4, pp. 3055-3062, 2008.
- [13] N. Leavitt, Recommendation technology: Will it boost e-commerce?, *IEEE Computer Society*, vol. 39, no. 5, pp. 13-15, 2006.
- [14] M. Montague and J. A. Aslam, Condorcet fusion for improved retrieval, in *Proceedings of the 11th ACM International Conference on Information and Knowledge Management, New York, USA, 2002*, pp. 538-548.
- [15] R. Rafter and B. Smyth, Passive profiling from server logs in an online recruitment environment, in *Proceedings of the 17th International Joint Conference on Artificial Intelligence, Seattle, Washington, USA, 2001*, pp. 35-41.
- [16] J. B. Schafer, J. Konstan, and J. Riedl, E-commerce recommendation applications, *Data Mining and Knowledge Discovery*, vol. 5, no. 1-2, pp. 115-153, 2001.
- [17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, Analysis of recommendation algorithms for e-commerce, in *Proceedings of the 2nd ACM Conference on Electronic Commerce, Minneapolis, Minnesota, USA, 2000*, pp. 158-167.
- [18] L. Si and J. Callan, A semisupervised learning method to merge search engine results, *ACM Transactions on Information Systems*, vol. 21, no. 4, pp. 457-491, 2003.
- [19] K. Srikumar and B. Bhasker, Personalized product selection in internet business, *Journal of Electronic Commerce Research*, vol. 5, no. 4, pp. 216-227, 2004.
- [20] X. Su and T. M. Khoshgoftaar. (2009, January) A survey of collaborative filtering techniques, *Advances in Artificial Intelligence*. [Online]. Available: <http://www.hindawi.com/journals/aai/2009/421425/>.
- [21] J. Wang, A. P. De Vries, and M. J. T. Reinders, Unified relevance models for rating prediction in collaborative filtering, *ACM Transactions on Information Systems*, vol. 28, no. 3, pp. 1-42, 2008.
- [22] H. Ye, A personalized collaborative filtering recommendation using association rules mining and self-organizing, *Journal of Software*, vol. 6, no. 4, pp. 732-739, 2011.