INTEGRATING COLLABORATIVE FILTERING AND MATCHING-BASED SEARCH FOR PRODUCT RECOMMENDATION

Noraswaliza Abdullah

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

School of Electrical Engineering and Computer Science
Faculty of Science and Engineering
Queensland University of Technology
August 2012

Dedicated to the people I love the most

Abdullah Mamat

Che Wook Mamat

Nahar Ahmad

Aisyah Nabila

Ammar Naufal

Keywords

Recommender System, User Profiling, Personalization, Implicit Feedbacks

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 them in making good decisions about which product to buy from the vast number of product choices available to them. 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 recommender system approaches. These approaches are not suitable for recommending luxurious and infrequently purchased products as they rely on a large amount of ratings data that is not usually available for such products. This research aims to explore novel approaches for recommending infrequently purchased products by exploiting user generated content such as user reviews and product click streams data. From reviews on products given by the previous users, association rules between product attributes are extracted using an association rule mining technique. Furthermore, from product click streams data, user profiles are generated using the proposed user profiling approach. Two recommendation approaches are proposed based on the knowledge extracted from these resources. The first approach is developed by formulating a new query from the initial query given by the target user, by expanding the query with the suitable association rules. In the second approach, a collaborative-filtering recommender system and search-based approaches are integrated within a hybrid system. In this hybrid system, user profiles are used to find the target user's neighbour and the subsequent products viewed by them are then used to search for other relevant products. Experiments have been conducted on a real world dataset

collected from one of the online car sale companies in Australia to evaluate the effectiveness of the proposed recommendation approaches. The experiment results show that user profiles generated from user click stream data and association rules generated from user reviews can improve recommendation accuracy. In addition, the experiment results also prove that the proposed query expansion and the hybrid collaborative filtering and search-based approaches perform better than the baseline approaches. Integrating the collaborative-filtering and search-based approaches has been challenging as this strategy has not been widely explored so far especially for recommending infrequently purchased products. Therefore, this research will provide a theoretical contribution to the recommender system field as a new technique of combining collaborative-filtering and search-based approaches will be developed. This research also contributes to a development of a new query expansion technique for infrequently purchased products recommendation. This research will also provide a practical contribution to the development of a prototype system for recommending cars.

Table of Contents

List of Figures

List of Tables

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signature: _____

Date: _____

Acknowledgements

I'm deeply grateful to my supervisor, Associate Professor Dr. Yue Xu for the guidance, patience and support she has provided throughout my time as her student. I consider myself very fortunate for having been able to work with a very considerate and encouraging supervisor like her. I owe my deepest gratitude to Nahar Ahmad, my husband, for his continued support and encouragement. Thanks also for the patience of my children, Aisyah Nabila and Ammar Naufal who experienced all of the ups and downs of my research. Finally, I would like to thank the Universiti Teknikal Malaysia Melaka and the Government of Malaysia for providing the funding which allowed me to undertake this research.

Chapter 1: Introduction

The invention of the World Wide Web (WWW) in 1990 by Tim Berners-Lee has changed how we conduct our daily activities nowadays. The WWW has become an enormous source of information and it continues to increase in size and use. People are relying more and more on the Web not only for information sourcing, but also for other usages such as communicating, banking, investing, shopping, as well as for education and entertainment purposes. 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 companies have offered their products and services over the internet by using e-commerce applications. An e-commerce website provides a huge number of product or service choices for a user to choose from which leads to an information overload problem. In this situation, the users become overwhelmed with the vast amount of information available to them and it is challenging for them to make a final choice about which products to choose. Recommender Systems (RS) have emerged in response to the information overload problem by learning from users about their interests and suggesting products that are likely to fit their needs. Therefore, the RS helps users to decide which product they would like to purchase on e-commerce sites. Nowadays, recommender systems have been widely applied by major e-commerce websites for recommending various products including books, music CDs or DVDs and for serving millions of consumers (Schafer, Konstan & Riedl, 2001). Commercial e-commerce sites include Amazon (www.amazon.com), CDNOW (www.cdnow.com), eBay (www.eBay.com),

Levis (www.levis.com), Moviefinder.com (www.moviefinder.com) and Reel.com (www.reel.com) websites (Leavitt, 2006).

1.1 BACKGROUND

The most popular recommendation systems are Collaborative Filtering (CF) and Content Based (CB) approaches. The CF approach recommends products to a new user based on the products that like-minded users have previously shown interest in. On the other hand, the CB approach recommends products that are similar to the products that the target user has liked before. Both approaches require users to provide a large amount of ratings data which shows how much the users like the products they previously owned, before accurate recommendations can be provided to them. Currently, the CF and the CB approaches are popularly applied in recommending products that are frequently purchased by users since a large amount of users' ratings or purchase history is available for use by these approaches in determining products that potential users might want to buy. In domains where products such as cars or houses are expensive and not regularly purchased by users, it is difficult to accumulate a large amount of ratings data from users. For this kind of product, standard search-based engines are still widely used as the common tool for users to search for their desired products. In this kind of search, users are required to specify product attributes as a query and the search engine will display products that match users' queries. Usually, the initial query provided by users is short, because they may not know the technical details of the products they want to buy and, thus, they are not sure what information they need to provide to the search engine. As a result, the search results are not personalized as the query does not represent a particular user's requirements accurately. Hence, a recommender system that can

predict user preferences without the availability of explicit ratings data is required to provide personalized recommendations for infrequently purchased products.

Fortunately, e-commerce data is rich and can be collected inexpensively. The actions of users in a virtual store can be collected. Information about products users look at when they browse through an e-commerce site, for example, can be easily collected. This navigation data shows products that may be of interest to a user and it is useful to recommender systems in predicting the user's preferences. In addition, with the emergence of Web 2.0, which provides a platform for users to conduct online participation, collaboration and interaction, a great deal of user generated content such as product reviews, tags and blogs is now available. These new sources of knowledge can be exploited and analysed to understand user preferences so as to provide good recommendations to the user. This research explores user reviews and online click stream data to extract knowledge about user preferences which is then used in recommending infrequently purchased products. A new recommendation technique based on query expansion is proposed that utilizes knowledge extracted from user reviews data. A hybrid recommendation approach is also proposed that combines the collaborative filtering and the search based techniques to generate more accurate recommendations. In addition, a new technique for profiling users based on the online click stream data is proposed. The generated user profiles are utilized by the proposed hybrid recommendation approach for searching relevant products to recommend based on products favoured by similar users. The proposed user profiling and recommender approaches can be adopted by e-commerce applications for recommending a greater variety of products.

1.2 RESEARCH PROBLEM AND OBJECTIVES

1.2.1 Research Problem

Current recommendation approaches rely on the explicit ratings data to make meaningful recommendations. However, explicit ratings data is not always available especially for products that are not regularly purchased by users. This research investigates how to exploit user generated content such as user reviews and user click streams data for extracting knowledge about user preferences for use in recommending infrequently purchased products. Many e-commerce websites for infrequently purchased products provides basic search functions that take a user's initial query as input and return a set of matched products to the query. Usually a user is required to provide some attributes values of the product that she or he is looking for, as a query in the search form and the products that have these attribute values are recommended to the user. In this basic search-based approach, the query provided by a user is short and does not fully represent the user's preferences. If the user's query can be expanded with knowledge about the user's preferences for product attributes, this query can be used to retrieve products that more closely satisfy the user's needs than would otherwise be possible. In this thesis, the reviews data provided by previous users is used to extract associations between attribute values of the products. These association rules can be used to expand the user's query to represent the user's requirements more precisely. In addition, the current collaborative filtering recommender approach is not directly applicable to recommending infrequently purchased products. This research also explores a novel hybrid collaborative filtering and search-based approach for generating recommendations based on profiles generated from the user click streams data. The online product click stream that is generated when a user browses products on an e-

commerce site shows products that are of interest to the user. From these viewed products, a user profile can be generated that represents the user preference for each product attribute value or how much the user likes each product attribute value. This user profile can be used by the collaborative filtering approach to find similar users without depending on the ratings data as normally applied in the standard CF approach.

To conclude, the questions raised in this research are as follows:

- What information resources can be used to extract knowledge that can be utilized by recommender systems for recommending infrequently purchased products?
- What knowledge can be extracted from those information resources and how should the knowledge be presented for use in recommending products?
- How can the extracted knowledge be used to expand a user's query in a search-based approach?
- How can the extracted knowledge be used by collaborative-filtering for recommending products?
- How should the collaborative filtering and the search-based approaches for recommending infrequently purchased products be integrated?

1.2.2 Research Objectives

The primary objectives of this research are:

1. To generate association rules between product attribute values based on user reviews data

User reviews data contains user ratings on product usage features. From this data, associations between product attribute values will be extracted by using an association rule mining technique. The associations between product attributes show other product attributes that are most likely preferred by users, based on initial attributes given by them in a query.

2. To generate user profiles from online click streams data.

   Log data contains users' browsing history, which provides information about products that each user has visited or viewed. From this information, a user's preferences for product attribute values can be generated that represent the user interests for each product attribute value.

3. To develop a query-expansion approach by using association rules generated from the user reviews data.

   Association rules between product attributes values can be used to expand a user's query, in which more attribute values that are of interest to the user are used to retrieve products that most likely meet the user preferences.

4. To develop a hybrid recommender system approach by integrating collaborative-filtering and search-based techniques using the generated user profiles.

   When a user searches for products to buy, the user may navigate from one product to another product, or perform a new search to find his or her desired products. This user browsing behaviour can be used as implicit ratings to find his/her peer users based on other users'

navigation histories from the log data. A collaborative-filtering recommender approach will use the user profiles generated from the online click streams data to find similar users or neighbour users to the target user and recommend products for him or her based on the products that the neighbour users have viewed before. For frequently purchased products like books or CDs, many copies of each product are available. Thus, products that have been viewed or purchased by previous users can still be recommended to a new user. In contrast, for infrequently purchased products like used cars, each product is unique. Thus, products viewed by previous users may have been purchased by other users and no longer available for purchase by a new user. Therefore, the search-based approach is incorporated with the collaborative filtering to find other products relevant to the neighbour users' products of interest for recommending this kind of products.

5. To conduct experiments and to evaluate the performance of the proposed approaches.

   Experiments must be conducted to evaluate the performance of the proposed approaches. The experiments involve the development of the models for the proposed approaches and the baseline approaches using the selected programming language. The testing data is used to evaluate the recommendations generated by the proposed models against the baseline models.

1.3 RESEARCH SIGNIFICANCE AND CONTRIBUTIONS

The contributions of this research include the theoretical contribution to the recommender system approach and practical contribution to the development of recommender systems for e-commerce applications. Theoretically, this research makes important contributions to personalization by exploring and exploiting new data resources and providing a novel approach to construct a new user profiling approaches by using user click stream data. Moreover, this research also contributes to the development of query expansion techniques and a new hybrid collaborative filtering and search-based recommendation technique. Both proposed approaches can enhance the current search-based approach for recommending infrequently purchased products without high involvement needed from the users.

In addition, this research will make practical contributions to the development of recommender systems for e-commerce applications. The proposed hybrid recommender system can be applied in more e-commerce applications for recommending a wider range of products. The available recommender system relies on a large amount of explicit ratings data to make meaningful recommendations and thus it is not suitable for recommending all kinds of products. This research makes significant contributions to the recommender system field as it finds new techniques for recommending products based on knowledge extracted from user reviews and user click stream data. Therefore, the proposed techniques can alleviate the current problems of recommender systems that rely on a large amount of ratings data to make meaningful recommendations.

1.4 RESEARCH METHODOLOGY

The methodology adopted by this research is the scientific method, a process for experimentation that is used to explore observations and answer questions. The scientific method involves a process of thinking through the possible solutions to a problem and testing each possibility to find the best solution. With the scientific method, project research is done with the goal of expressing a problem, proposing an answer to it (the hypothesis), designing project experimentation and performing the project experimentation to test the hypothesis. The scientific method is chosen in order to make sure the work is free from bias, inconsistencies, and unnecessary complications, as well as for creating an accurate theoretical structure. There are five major steps to be undertaken in this method:

1. **Define the question** – identify a significant problem to be solved or phenomenon to be researched.
2. **Research the topic** – involves gathering relevant information to attempt to answer the question and learning as much about the phenomenon as possible, including studying the previous studies of others in the area.
3. **Formulate the hypothesis** – propose a solution or answer to the problem or question. The proposed hypothesis must be stated in such a way that it is testable.
4. **Test the hypothesis** – test the hypothesis by conducting an experiment before it is substantiated and given any real validity.
5. **Analyse the data** – analyse the results of the project experimentation to see if the results of the experiment support or refute the hypothesis.

Reasons for experimental results that are contrary to the hypothesis are included for further testing.

After the researcher finishes these steps, the following standard practice is to make all collected data available for other researchers, so that they can confirm or refute the hypotheses and the experiment quality. This, hopefully, leads to better work based on what has been done previously.

In the scientific method, an experiment is performed through a set of actions and observations in the context of solving a particular problem or question in order to support or refute the hypotheses that were developed and help refine the researcher's work until a successful implementation is developed. The developed hypotheses will then be implemented and tested through controlled experimentation that produces results that can be reproduced if another researcher were to undertake the same experiments. In the experiment, the researcher also has control over the study. For example, in this research, many techniques or methods will be developed that involve collaborative-filtering, a search-based approach and a combination of the two techniques in a hybrid recommender system. Experimentation with the suggested techniques or methods must be undertaken to see how and when they really work, to understand their limits, and to understand how to improve them. In fact, most experiments and observations are repeated many times to test whether the hypotheses are true or false. In addition, controlled experiments will be used to test the solutions that are developed for each technique to see if they really work and what effects these solutions have on other aspects of the system.

1.5 THESIS OUTLINE

The rest of this thesis is organized as follows:

Chapter 2: This chapter contains critical and comprehensive reviews of existing research works in related research fields. It identifies and justifies the research context and gap from which the research questions were derived and pinpoints the weaknesses of the existing research works.

Chapter 3: This chapter will discuss the proposed query expansion model based on the association rules generated from the user reviews data. This chapter firstly discusses the opinion mining technique to determine the orientation of the user review. Then, this chapter will explain the rough set rule mining to generate decision rules between initial product attribute values given by target users and other attribute values that might be of interest to the target users from the user reviews. Finally, this chapter will present the proposed query expansion approach for recommending products based on the extracted decision rules.

Chapter 4: This chapter will discuss the integration of collaborative filtering and search-based approaches. Firstly, this chapter will present a user profiling approach to generate user profiles from user click stream data. Then, three proposed recommendation approaches that integrate collaborative filtering and search-based approaches by utilizing the generated user profiles will be presented.

Chapter 5: This chapter will discuss the evaluation of the proposed recommendation approaches. The discussions will start with the experiment design and methods, followed by the detailed analysis of the experiment results.

Chapter 6: This chapter concludes the thesis and draws the direction for future works.

1.6 RESEARCH OUTCOMES

There are two main outcomes of this research work, as follows:

- **Publications**

1. Braak, P. T., Abdullah, N. & Xu, Y. (2009). Improving the performance of collaborative filtering recommender systems through user profile clustering. *In the Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, 147 – 150.
2. Abdullah, N., Xu, Y. & Geva, S. (2010). Enhancement of Infrequent Purchased Product Recommendation using Data Mining Techniques. *In Artificial Intelligence in Theory and Practice III, IFIP Advances in Information and Communication Technology*, Volume 331, 57-66.
3. Abdullah, N., Xu, Y. & Geva, S. (2010). Infrequent Purchased Product Recommendation Making based on User Behaviour and Opinions in E-commerce Sites. *In the Proceeding of 10th International Conference on Data Mining Workshop(ICDMW)*, 1084-1091.
4. Abdullah, N., Xu, Y. & Geva, S. **(**2011). A Recommender System for Infrequent Purchased Products based on User Navigation and Product Review Data. *Lecture Notes in Computer Science*, Volume 6724, 13-26.

5. Abdullah, N., Xu, Y. & Geva, S.(2011). Integrating Fusion Techniques into the Collaborative Filtering Search-based Recommender Systems. *In the Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*. 343-346.

6. Abdullah, N., Xu, Y. & Geva, S. (2011). Integrating Collaborative Filtering and Search-based Techniques for Online Product Search. *In the Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW 2011)*. 711-718.

- **Prototype System**

  This research work is motivated by a need for an automatic recommender system for online car sales expressed by one of the QUT industry partners. The dataset that will be used in this research work is obtained from this company. A prototype system has been developed for this company based on the proposed approaches. This prototype system has been presented to the company and uploaded into the company's repository.

Chapter 2: Literature Review

This research deals primarily with hybrid recommender systems that combine collaborative filtering and search-based recommender approaches for recommending infrequently purchased products. The purpose of this literature review is to present an introduction to the work that has already been done in the relevant fields. It will serve as the starting point for deeper investigation into these fields and the problems that exist. In this part, recommender systems are firstly reviewed. This is followed by a review of data mining and web mining that focuses on techniques that will be used in this research, namely association rule mining and web usage mining. Finally a review about the query expansion in information retrieval will be given.

2.1 RECOMMENDER SYSTEM

Burke (2000) defined a Recommender System (RS) as a computer system that provides advice to users about items they might wish to purchase or examine. A recommender system provides individual personalization to each user by customizing its recommendations and presenting different items for each user according to her/his tastes. By selecting and providing a list of products that are likely to fit a user's needs from a large number of product choices offered by an e-commerce site, recommender systems help the user deal with information overload, reduce the user search time for interesting items, and enhance the effectiveness of the user decision making. Furthermore, recommender systems also benefit merchandisers as they can enhance sales on their e-commerce sites by converting browsers into buyers, increasing cross-selling and building consumer loyalty

(Schafer, Konstan & Riedl, 1999). Nowadays, recommendation techniques are widely used in practical applications by commercial e-commerce websites such as Amazon.com, Moviefinder.com, Reel.com, Levis.com and eBay for recommending various products such as books, CDs, movies, news, and articles to target users.

To provide a personalized set of recommendations, a recommender system incorporates a user's wishes into a user model and exploits suitable recommendation algorithms to map the user model into targeted product suggestions (Ricci & Wietsma, 2006). There are three steps involved in a recommender system: acquiring preferences from a user's input data; computing the recommendation using proper techniques; and finally presenting the recommendation results to users (Wei, Huang & Fu, 2007). A recommender system suggests products by applying data analysis techniques to various pieces of knowledge gathered from different sources, that is, from the user, from peer users of the system, from data about the items being recommended, and also from the domain of recommendation itself, for example knowledge about what requirements recommended items satisfy (Felfernig & Burke, 2008). The items of knowledge can be acquired explicitly or implicitly from the sources. Examples of items of knowledge that can be acquired explicitly from the users are demographic data, ratings data, and product requirements as stated by the user in an online form. Knowledge about users' preferences can also be acquired implicitly from the users' behaviour pattern data in the log data and also from transaction data in the database by using web mining and other web technologies (Wei et al, 2007). In addition, besides the knowledge generated by the user, knowledge can also be obtained from production data, for example, product attributes can be gathered from the product domain. Various knowledge sources used by recommender systems can be depicted in Figure 2.1.

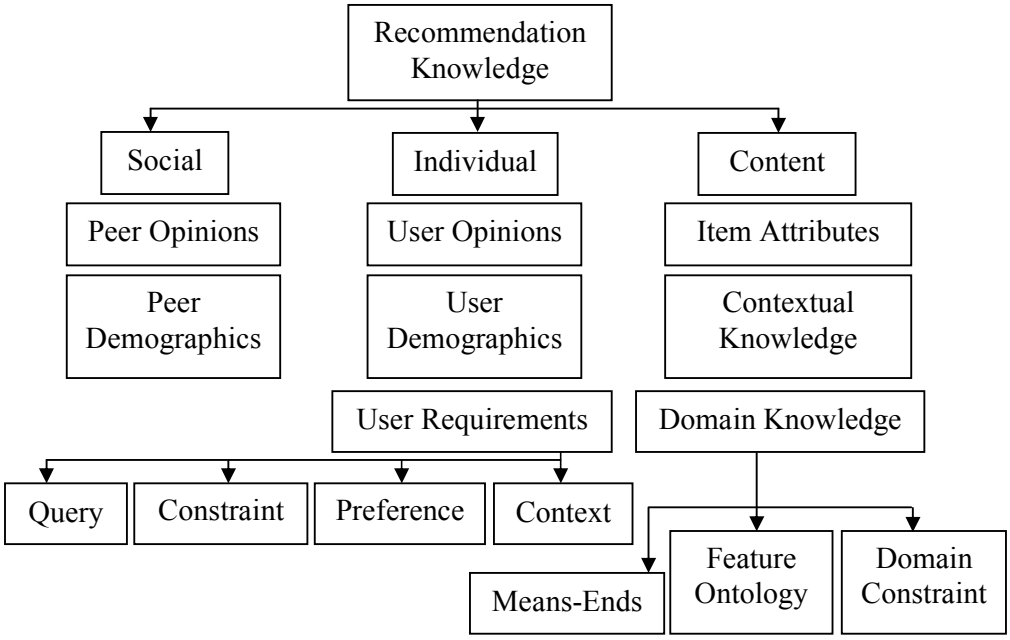


Figure 2.1: Knowledge Resources in Recommender Systems (Felfernig & Burke, 2008).

The recommendation approaches can be classified as collaborative-filtering; content-based filtering; knowledge-based and hybrid based, depending on the different set of knowledge sources and the algorithmic approach employed by the recommendation system. The collaborative-filtering approach makes use of social knowledge (e.g. user ratings) gathered from other users and is based on the correlations between the target user and other users, which are determined according to their ratings similarities. Thus, an analysis of user rating data recommends products for users according to what people with similar tastes and preferences have liked in the past. In contrast, the content-based approach is based on items correlation that is determined based on the items' attribute values and, hence, recommends products that are similar to products that the user has liked in the past. The knowledge-based recommender exploits knowledge that is not utilized in the other approaches, namely, user requirements and domain knowledge. It gathers user requirements about the target product and consults its knowledge base to reason what products match with the user requirements. The hybrid recommender system combines multiple approaches to improve the recommendation performance of a singular approach. Much research has been conducted by applying different techniques of recommender systems. The following table shows different techniques that have been used in the aforementioned recommender system approaches. Note that the table is not an exhaustive list of research in this area.

Table 2.1: Different Techniques of the Recommendation Approaches

Recommender system approach	Techniques	Research Title	Authors
Content-based	Clustering	User modelling for adaptive news access	Billsus & Pazzani(2000)
	Clustering	Content-based recommendation in e-commerce	Xu et al.(2005)
	KNN	Fab: content-based, collaborative recommendation	Balabanovic & Shoham(1997)
	Artificial Neural Network	Novelty and redundancy detection in adaptive filtering	Zhang, Callan, & Minka(2002)
	Bayesian	Content-based book recommending using learning for text categorization	Mooney & Roy(2000)
Collaborative-filtering	Web Mining	A personalized recommender system based on web usage mining and decision tree induction	Cho, Kim & Kim (2002)
	Web Mining	Application of web usage mining and product taxonomy to collaborative recommendation	Cho & Kim (2004)
	Association Rule Mining	A click stream-based collaborative filtering personalization model: towards a better performance	Kim, Atluri, Bieber, Adam, Yesha & Im(2004)
	Clustering	Eigentaste: a constant time collaborative filtering algorithm	Goldberg, Roeder, Gupta & Perkins(2001)
	KNN algorithm and improved one	Scouts, promoters, and connectors: the role of ratings in nearest neighbour collaborative filtering	Mohan, Keller & Ramakrishnan (2007)
	Maximum Entropy	A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains	Pavlov & Pennock (2002)
	Latent Semantic Analysis	Latent semantic models for collaborative filtering	Hofmann (2004)

Table 2.1: Different Techniques of the Recommendation Approaches (Continued)

Recommender system approach	Techniques	Research Title	Authors
Collaborative-filtering	Support Vector Machine	Recommender Systems using support vector machines	Min & Han (2005)
	Linear Regression	Collaborative filtering using regression-based approach	Vucetic & Obradovic (2005)
	Markov Process	An MDP-based recommender system	Shani, Heckerman & Brafman(2005)
Knowledge-based	Case-based recommendation	Feature selection methods for conversional recommender systems	Mirzadeh, Ricci & Bansal (2005)
	Case-based recommendation	Acquiring and revising preferences in a critique-based mobile recommender system	Ricci & Quang Nhat (2007)
	Case-based recommendation	Compound critiques for conversational recommender systems	Smyth, McGinty, Reilly & McCarthy (2004)
	Constraint-based recommendation	The VITA financial services sales support environment	Felfernig, Isak, Szabo & Zachar (2007)
	Constraint-based recommendation	A personalized system for conversational recommendations	Thompson, Goker & Langley(2004)
Hybrid	Clustering	A new approach for combining content-based and collaborative filters	Kim, Li, Park, Kim & Kim (2006)
	Probabilistic Model	Probabilistic models for unified collaborative and content-based recommendation in sparse data environments	Popescul, Ungar, Pennock & Lawrence (2001)
	Maximum Entropy	A maximum entropy web recommendation system: combining collaborative and content features	Jin, Zhou, & Mobasher (2005)

The earliest implementation of recommender systems was a collaborative filtering recommender system called Tapestry (Goldberg, Nichols, Oki & Terry, 1992), which was developed in the mid-1990s. Since then, recommender systems have become an important and independent research area and many recommendation technologies are developed using a broad range of statistical, machine learning and information retrieval techniques. However, collaborative-filtering and content-based approaches have received much attention from the recommender system community and have been widely applied in commercial systems for recommending simple and frequently purchased products. Little research has been done for recommending products that are more complex and infrequently purchased by users such as cars, houses, or other luxurious products or services where a large amount of ratings data is difficult to accumulate for use by a collaborative-filtering or content-based recommender system. Therefore, the current generation of recommender systems requires further improvements to make recommendation methods more effective and applicable to an even broader range of real life applications that includes recommendations pertaining to more complex types of application (Adomavicius & Tuzhilin, 2005).

In the following sections, several typical recommender systems will be reviewed in depth. Some of the issues/problems that these systems suffer from will also be presented and discussed.

2.1.1 Recommender System Approaches

2.1.1.1 Content-based Recommender System

Schafer et al. (1999) defined the content-based recommendation system as an item-to-item correlation system as it recommends items based on items with similar content items to those that a user liked before. The content-based approach has its

roots in information retrieval and information filtering (Adomavicius & Tuzhilin, 2005; Wei et al., 2007). The content-based filtering technique is based on an analysis of the content of the items and is suitable for recommending text-based items for which the content is described by keywords such as 'news' and 'articles'. Examples of such systems are the newsgroup filtering system: NewsWeeder (Lang, 1995); the web page recommender systems: Fab (Balabanovi & Shoham, 1997) and Syskill & Webert (Pazzani, Muramatsu & Billsus, 1996); the book recommender system: Libra (Mooney & Roy, 2000); as well as the funding recommendation system: ELFI (Schwab, Pohl & Koychev, 2000).

The important sources involved in this approach are the item features and the ratings that a user has given to the items. This approach is based on the past interest profile of the user where the profile is learned from the features of the previously rated items given by the user. Thus, based on the previous products that the user has indicated they preferred, the content-based recommender system suggests items a user might be interested in. The content-based problem has been tackled using a variety of information retrieval techniques such as TF/IDF, clustering, K-nearest neighbours, naïve Bayes, artificial neural network and association rule mining. However, the content-based approach still has several shortcomings due to the limited recommendations resulting from the restricted features of the products the user has rated in the past.

One of the drawbacks of the content-based recommender approach is a new user problem (Adomavicius & Tuzhilin, 2005; Felfernig & Burke, 2008; Wei et al., 2007). It is unlikely that the technique will provide accurate recommendations for a new user who only has a few ratings of products. In order to obtain good recommendations, the user must provide sufficient ratings of products before a

content-based system can really learn their preferences. However, having users rate many items is a tedious task and it is unlikely that new users will provide sufficient ratings when they are new to the system. The second shortcoming of the content-based recommendation approach is overspecialization, where only the items that have similar features to those already rated in the past are recommended to the user (Adomavicius & Tuzhilin, 2005; Wei et al., 2007). There is no chance for items that are different from what the user has seen in the past being suggested to the user. This restricts the choice of items for the users as they might like items that they have not previously seen. Finally, another weakness of the content-based recommender system is that it is designed to recommend mostly text-based items, thus, it is only able to perform recommendations in restricted domains such as web pages, news and articles. In addition, because the content is usually described by keywords, it also suffers from limited content analysis when keywords are limited to the items being recommended (Adomavicius & Tuzhilin, 2005).

Content-based recommender systems do not use social knowledge or knowledge about other users for recommending products and, thus, only items that match the content features in the user interest profile will be recommended to the user. Researchers believe that a recommendation approach using social knowledge leads to recommendation novelty or serendipity, where more unexpected or different items that are equally valuable will be recommended to the user (Schafer, Frankowski, Herlocker & Sen, 2007). The collaborative-filtering recommender systems use social knowledge for recommending products and this approach will be discussed in the following section.

2.1.1.2 Collaborative-Filtering Recommender System

The collaborative-filtering recommender system is the earliest and most successful recommendation technology (Adomavicius & Tuzhilin, 2005; Karypis, 2001). This approach automates the "word-of-mouth" paradigm that evaluates items for the target user based on the opinion of other users. The main idea of this approach is that a target user is likely to enjoy the items that other users with common interests have liked. Schafer et al. (1999) described collaborative-filtering recommender systems as people-to-people correlation recommender systems as they recommend products to a potential user based on the degree of correlation between that user and other users who have purchased the products in the past. This approach assumes that human preferences are correlated, in that a user with similar tastes will rate things similarly. Thus, explicit ratings are the typical input of this approach.

The collaborative-filtering approach only relies on collaborative knowledge sources such as collaborative opinion profiles, demographic profiles and user opinion and, thus, it does not require any other information about users or products. It can also be applied in many domains other than text-based items as in the content-based recommender systems. Collaborative-filtering systems are implemented in various domains such as in the Usenet newsgroup articles domain: Grouplens (Resnick et al., 1994); the music and musical artists domain: Ringo (Shardanand & Maes, 1995); the movies domain: Bellcore's Video Recommender (Hill, Stead, Rosenstein & Furnas 1995); the jokes domain : Jester (Goldberg et al., 2001); the books domain: Amazon.com (Linden, Brent & York, 2003); and other product domains.

A collaborative-filtering approach suffers from some drawbacks. One of the shortcomings of the collaborative-filtering recommendation approach is that it must be initialized with a large amount of user preference data in order to make

meaningful recommendations (Burke, 2000). As a consequence, this approach suffers from the cold-start problem in which a recommender is unable to make meaningful recommendations because of the lack of initial ratings when new items or new users enter the system (Adomavicius & Tuzhilin, 2005). New items cannot be recommended to any user until they have been rated by a considerable number of users. Similarly, new users must provide a substantial amount of ratings data to enable recommender systems to learn their preferences and give useful recommendations. Another shortcoming of a collaborative-filtering approach is the sparsity problem that arises when only a small number of people rate a particular item and, as a result, the item will rarely be recommended even when it receives a high rating from users (Adomavicius & Tuzhilin, 2005). Also, a collaborative-filtering approach cannot provide meaningful recommendations when the user's interests change, as the recommendation is based on past user preferences from historical ratings data that do not represent the user's current preferences (Tran, 2006).

The collaborative filtering algorithms can be classified into model-based and memory-based algorithms (Breese, Heckerman & Kadie, 1998; Adomavicius & Tuzhilin, 2005; Wei et al., 2007; Shafer et al, 2007). In the model-based algorithms, a model is learned from the collection of ratings based on the training data. Then, the model validity is checked with the testing data and finally the rating predictions of the target user's no-rating products are computed. Various statistical and machine learning techniques are used for the model-based approach, such as probabilistic models based on the Bayesian networks (Breese et al., 1998), the statistical model based on K-means clustering (Shepitsen, Gemmell, Mobasher & Burke, 2008) and the latent factor models based on matrix factorisation (Koren, 2008). The model-

based approach only requires some data from the active customer for the model to give prediction value.

On the other hand, the memory-based algorithms, also called heuristics-based algorithms, require ratings, items and users to be stored in memory and recommendation results are calculated based on the entire users' database. The recommendation process for the memory-based algorithm includes user profiling, neighbourhood formation and recommendation generation. It first builds an interest profile for a user based on the user's ratings on items that the user has purchased before, and then it makes recommendations based on the similarity between the interest profile of that user and those of the other users (Greening, 1998; Pazzani & Billsus, 1997; Resnick et al., 1994; Tran, 2006). Thus, searching for similar preferences between the active user and the other users is an important step in this recommendation approach before presenting the recommendation according to the preference of similar users (Wei et al., 2007). Cosine similarity and Pearson correlation are the most popular approaches to calculate the similarity between two users (Adomavicius & Tuzhilin, 2005).

The standard collaborative filtering methods rely entirely on the ratings data to make recommendations. Therefore a collaborative-filtering approach works best with a large amount of ratings data and is suitable for recommending frequently purchased products as its database of user preferences gets larger and larger over time when users purchase the products repetitively. This is because more and more users rate particular items and particular users eventually rate more and more items. However, this approach is not suitable for infrequently purchased items because it is not viable for a recommender system to learn a user interest profile and to accumulate a pattern of preferences between an active user and other users without a

large amount of user ratings data. Currently, more advanced profiling techniques have been used for understanding the items and user profiles for recommendation purposes. These techniques generate user or item profiles that can be used by the ratings estimation function to estimate the unknown ratings. For example, Web usage analysis based on data mining techniques has been used to discover the navigational usage patterns of users to provide better Web site recommendations. However, the techniques to make use of the navigational data have not been widely adopted in rating-based recommender systems (Adomavicius & Tuzhilin, 2005). Therefore, an interesting research problem would be to explore the navigational data to generate the user profiles and to develop recommendation methods that can utilize these profiles in order to support more complex types of recommendation applications.

Currently, the knowledge-based recommender systems approach is more popularly employed for recommending products that are more complex and infrequently purchased by users than for recommending less complex, frequently purchased items. This approach acquires users' preferences by asking the users about their product requirements and provides recommendations by reasoning what products meet the requirements using domain knowledge. The following section will discuss the knowledge-based recommender system.

2.1.1.3 Knowledge-based Recommender System

A knowledge-based recommender system uses deep knowledge about the product domain in order to provide recommendations that exactly fit the wishes of the user (Felfernig, 2005). This system is free from sparsity and from the cold starting problems of the collaborative-filtering system as its recommendations do not depend on historical data to get user preferences. A knowledge-based recommendation technique exploits knowledge of the product's domain and the user

requirements in reasoning what products meet the user requirements. It then recommends products to the user accordingly. It is suitable for recommending infrequently purchased products (that is, those for which a large amount of ratings data is difficult to accumulate over time). These kinds of products such as cars, houses, computers, and electronic equipments usually have high value and their features are highly important and are always considered by the users when they want to buy them. Therefore, a recommender system must take the user preferences for each attribute value into consideration when making recommendations for these kinds of products. The knowledge-based recommendation technique explicitly asks users about their product requirements or preferences and then consults its knowledge-base to recommend items that satisfy the user's need.

Two well known approaches to knowledge-based recommendations are case-based recommendations and constraint-based recommendations. Case-based recommendations treat recommendations as a similarity assessment problem that involves domain-specific knowledge and considerations (Felfernig & Burke, 2008). Examples of case-based recommendations are FindMe systems, which include Car Navigator systems for selecting a new car; Video Navigator and PickAFlick for choosing a rental video; Entrée for selecting a restaurant; and Kenwood for configuring a home audio system (Burke, Hammond & Yound, 1997). The FindMe systems emphasise two main features – the centrality of the example provided by the user and the tweaking process. Using the centrality of the user's example, the FindMe system recommends items that are similar to the item in which the user has expressed an interest; and using the tweaking process, it allows the user to alter the characteristics of the example to obtain recommendations that best meet their requirements (Burke, 2000). RecommenderEx uses Case-Based Reasoning Plan

Recognition approaches and Automated Collaborative-Filtering to address the knowledge representation and utilization issue for the products purchased repeatedly by a user (Prasad, 2007).

In contrast to the case-based recommendation system, a constraint-based recommendation system is viewed as a process of satisfying constraints that come from either the users or the product domain and, thus requires explicit definition of questions, product properties and constraints (Felfernig & Burke, 2008). Examples of constraint-based recommenders are VITA (Felfernig et al., 2007) and Koba4MS (Felfernig, 2005) for supporting sales dialogues between sales representatives and users who are interested in financial services domain products. Felfernig and Burke (2008) discussed research issues and techniques of constraint-based recommendations. They also gave example recommendations of web hosting services using the constraint-based technique. In order to determine the most interesting recommendations for a user, the degree of fit between an item and the given set of user requirements is calculated using the Multi-Attribute Utility Theory (MAUT). Model Based Diagnosis is used to resolve conflicts and to provide repair actions when user requirements are inconsistent with the set of compatibility constraints. A weighted majority voter is used to predict interesting values for the product attributes not specified by the user. Both the case-based recommendation and constraint-based recommendation systems must collect the user requirements, propose alternative solutions where no items fit the user's requirements and provide explanations for recommended items (Felfernig & Burke, 2008).

The knowledge-based recommender system relies heavily on domain knowledge and is suitable for recommending complex products and services that are infrequently purchased by the users. However, it requires deep knowledge

engineering and a good understanding of the product domain in order to build a knowledge base for storing important features of products that can be inferred by the recommender system for suggesting products. Building a knowledge base is not a trivial task because the knowledge has to be acquired from domain experts and the knowledge needs to be continuously maintained in order to avoid wrong recommendations (Zanker, 2008). Therefore, a recommender system that can automatically acquire users' preferences for attribute values is crucial to lessen the knowledge engineering burden for recommending infrequently purchased products. This may be achieved by integrating different recommender systems into a hybrid recommendation system. This latter recommendation system will be discussed in the following section.

2.1.1.4 Hybrid Recommender System

The hybrid recommender system integrates multiple recommender approaches to improve recommendation performance and to avoid the weaknesses of a single recommender approach (Burke, 2000). Hybrid recommender system methods can be classified into weighted, switching, mixed, feature combination, cascade, feature augmentation and meta-level recommender systems (Burke, 2002). The following table describes different types of hybrid recommendation methods.

Table 2.2: Hybridization Methods (From Burke, 2002)

Hybridization method	**Description**
Weighted	The scores of several recommendation techniques are combined together to produce a single recommendation.
Switching	The system switches between recommendation techniques depending on the current situation.
Mixed	Recommendations from several different recommenders are presented at the same time.
Feature combination	Features from different recommendation data source are thrown together into a single recommendation algorithm.
Cascade	One recommender refines the recommendations given by another.
Feature augmentation	Output from one technique is used as an input feature to another.
Meta-level	The model learned by one recommender is used as input to another.

There are many combinations of hybrid recommender systems that can be formed from the content-based, collaborative-filtering and knowledge-based recommender approaches. Table 2.3 shows some examples of hybrid recommender systems according to the hybridization approaches they employ. Content/collaborative hybrids are widely deployed because the ratings data is already available or can be inferred from data (Burke, 2002). However, this combination does not alleviate the cold-start problem as both content-based and collaborative-filtering techniques rely on a database of users ratings.

Table 2.3: Examples of the Hybrid Recommender Systems

Recommender Approaches	**Research**	**Authors**
Content-based and collaborative-filtering recommender systems	Hybrid Recommendation: Combining Content-Based Prediction and Collaborative Filtering	Rojsattarat & Soonthornphisaj (2003)
Content-based and collaborative filtering recommender systems	Hybrid Collaborative Filtering and Content-based Filtering for Improved Recommender System	Jung, Park & Lee (2004)
Collaborative filtering and knowledge-based recommender systems	Designing Recommender Systems for E-Commerce: An Integration Approach	Tran (2006)
Content-based and collaborative filtering recommender systems	A Multi-Clustering Hybrid Recommender System	Puntheeranurak & Tsuji (2007)
Content-based and collaborative filtering recommender systems	A hybrid movie recommender system based on neural networks	Christakou, Vrettos & Stafylopatis (2007)
Content-based and Collaborative filtering recommender systems	A content-collaborative recommender that exploits WordNet-based user profiles for neighbourhood Formation	Degemmis, Lops & Semeraro (2007)
Content-based and Collaborative filtering recommender systems	A Hybrid Content-Collaborative Recommender System Integrated into an Electronic Performance Support System	Iaquinta, Gentile, Lops, Gemmis & Semeraro (2007)
Collaborative filtering and knowledge-based recommender systems	A collaborative constraint-based meta-level recommender	Zanker (2008)
Collaborative filtering and knowledge-based recommender systems	REJA: A Georeferenced hybrid recommender system for restaurants	Martinez, Rodriguez & Espinilla (2009)

The knowledge-based technique seems to be a good candidate for hybridization because it does not suffer from cold-start problems. The knowledge-based technique may employ any kind of knowledge that is not being used by content-based and collaborative-filtering techniques. As a result, hybridization through knowledge-based techniques can utilise various kinds of knowledge across different sources for improving the recommender system and thus solve cold starting problems due to insufficient ratings data when new items or new users enter the system. However, it requires high involvement from domain experts to transfer the product knowledge to a knowledge base and high involvement from users to explicitly provide their requirements.

Further research is needed to explore new recommendation approaches that do not rely on user explicit ratings data or place burdens on users in acquiring their requirements in order to provide meaningful recommendations to them. Such new approaches could thus be applied for recommending products in a broader range of applications than is possible using existing recommendation approaches.

2.1.1.5 Recommender System based on Implicit Feedback

Recommender system approaches have extensively focused on processing explicit feedback or direct input from users regarding their preferences for calculating recommendations (Hu, Koren & Volinsky, 2008). However, explicit feedback data is not always available. In many practical scenarios, explicit feedback is hard to collect because of intensive user involvement. Implicit feedback such as purchase history, browsing history, and search patterns reflect users' behaviour and can be observed to infer the users' preferences. The inferred user preferences can then be utilized by recommender system algorithms to generate recommendations. However, little work has been studied in exploring how to exploit the rich user

information in community-based interactive information systems including purchasing and browsing activities, to improve the recommendation performance (Li, Hu, Zhai & Chen, 2010).

Currently, the usage of implicit feedback for recommending products has attracted new developments in recommendation algorithms that are suitable for processing implicit feedback. Kim, Yum, Song and Kim (2005) proposed a Collaborative-Filtering based recommender system that utilizes 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 one and the preference level of a product which is clicked, but not purchased, is estimated based on the probability of products that would be purchased, which is calculated based on the variables captured in the navigational data such as number of visits, length of reading time, basket placement status and suchlike.

Hu et al. (2008) 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 of a user buying 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 it for implicit feedback recommendations. Lee, Park & Park (2008) incorporate 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 one to the purchased items, a rating function is defined that computes rating values based on the launch time and purchased time of items to reflect the users' preferences to achieve better recommendation accuracy.

Li et al. (2010) proposed two ways to incorporate user information from a user's search query history, purchasing and browsing activities into collaborative filtering models. The first approach is by treating the user information as independent evidence and to linearly combine scores from different sources of user information to generate recommendations. The other is to embed the user information into the collaborative filtering model. The experiment results on a large-scale retail data set show that the user rich information such as search keywords and clickthrough data is very effective in overcoming the sparsity problem of the collaborative filtering and in helping when the neighbour-based methods have very low support from neighbours. The users' recent search query history can also be used to make recommendations because it tends to perform as well as, if not better than, the long term history.

The recommendation algorithms for processing implicit feedback are often studied independently from the domain knowledge. However, for some products, the product features are important factors for the user to consider in making decisions about the final products to buy. This thesis proposes to incorporate knowledge about product attribute values to generate user profiles from implicit feedback. To the best of the author's knowledge, there is no work that has been done so far to incorporate product attribute values for generating user profiles based on implicit feedback.

2.1.2 Recommender System in E-Commerce

The World Wide Web enables users to purchase products online via e-commerce applications. Many of the e-commerce sites only provide raw retrieval where the user enters the query through a search interface and all products that match the user's query are presented to the user. As a result, the user still faces difficulty in making decisions on which product to purchase, as there is still a range of products presented to her/him even though the system has selected the products based on

her/his query. Consequently, one of the crucial tasks of an e-commerce system is to help users make good decisions about which product to buy with minimal time and effort. An automated system to provide a more narrow selection of products for the users would be desirable in e-commerce applications to assist the users in making purchasing decisions.

Currently, recommender systems become important tools for internet marketing activities in e-commerce as they can provide a personal service for each user and support the user in product purchasing. They provide personalization on e-commerce sites by adapting product suggestions according to each user's preferences. Recommender systems also help e-commerce sites achieve mass customization by providing multiple choices of products that meet the multiple needs of multiple consumers (Schafer et al., 2001). In addition, recommender systems help e-commerce sites build users' loyalty and they also present users with products that they are interested in but had not planned on buying; hence, they encourage users to buy more products (Leavitt, 2006).

Many of the largest e-commerce websites such as Amazon.com, Apple Computer and Netflix DVD-rental are using recommender system approaches to assist their users in selecting products to purchase. In the Amazon.com site, for example, many types of recommendations are provided to assist users in making their purchasing decisions. This site provides product suggestions to users based on products that they have already purchased or rated. For example, in the book section of this site, in the *Today's Recommendation for You* feature, users receive recommendations for a list of books based on the books that they have already purchased or rated. The users' activities on the site such as book purchase or rating are compared with those of other users to recommend further books in which they

might be interested. The recommendation list thus generated recommends books that are similar to those in which the user has previously shown interest and, hence, users are exposed to books that they might have previously been unaware of. In the *Users Who Bought* feature, the system recommends books frequently purchased by other users who have purchased the selected book. Text reviews from the editorial and other users are also presented for each recommended book to provide the targeted user with more comments and opinions from others about the selected book. The average rating given by other users for the selected book is also provided to help the user choose which book to buy.

Another commercial recommender system is CDNow, a system to recommend albums of artists to users. The system identifies a group of users who like the same sets of CDs and then suggests a CD that is owned and liked by several members of the community but is not yet owned by the active user. This site also provides features similar to the *User Who Bought* feature of Amazon.com in that it recommends albums related to the album or artist preferred by the user. It also allows the user to select a particular genre of music, and the system provides a list of albums that match the selected genre. The user is also provided with a list of artists who have similar styles to the ones that the user has selected to give the user a wider choice of albums that they might be interested in. In addition, the site also recommends CDs to its users by listing the top 100 bestseller albums to be considered by them. Other e-commerce sites that support personalized recommendations for their users include Drugstore.com for recommending drugs; eBay for recommending sellers to be chosen by the user for the advertised products; and also MovieFinder.com and Reel.com, both for recommending movies to be watched by the users.

The collaborative filtering approach is not suitable for recommending products without the availability of a large amount of ratings data. Thus, a knowledge-based approach has received much attention from the recommender system community to recommend complex products or services that are infrequently purchased by the users. This approach does not depend on ratings data or purchase history to recommend products and thus can be used to provide recommendations for products where no user purchasing history data or ratings data is available. An example of a knowledge-based recommender system is the PersonalLogic system (http://www.personallogic.com), which provides recommendations for various products (for example cars, computers) and for selecting services (for example family activities, careers and graduate schools). For example, to recommend cars that might fit the user preferences, the system explicitly gathers the user's requirements such as car type and size, features that he/she prefers, price he/she can afford and also what kind of car he/she is looking for (luxury, economy). By consulting the knowledge base, cars that best match the user requirements are presented to the user.

Another example of knowledge-based recommender systems are FindMe systems, which use knowledge-based retrieval strategies for recommending various kinds of products (Burke et al., 1997). FindMe systems include Car Navigator, for suggesting a new car; Video Navigator and PickAFlick for selecting a rental video; RentMe for recommending an apartment; Entrée for finding a restaurant; and Kenwood for configuring a home audio system. Car Navigator was the first FindMe system developed. It provides an assisted-browsing system that combines searching and browsing with knowledge-based assistance to help users in selecting a new car model to buy. The system provides options for users to alter their initial preference variables by supplying four buttons to the user – sportier, roomier, cheaper, and

classier. If the user clicks one of the buttons, the whole set of search criteria is modified in one step according to the user's selected choice. This system enables the users to find a specific car that they like by gradually refining the criteria based on the previously retrieved products until they are satisfied with the suggested products.

Yet, in the current knowledge-based recommender system, the system must understand the product domain well and this requires an extensive project in knowledge acquisition which is undertaken by transferring experts' knowledge into a knowledge base. Integrating two or more recommendation approaches in a hybrid recommender system has been considered a promising way to avoid the weaknesses of each recommender system approach and also to strengthen the recommendation system performance. The Entree system is an example of a hybrid recommender system that integrates collaborative-filtering and knowledge-based recommender system approaches (Burke, 1999). In this system, the knowledge-based approach is used for initial suggestions when only a small amount of ratings data is available for use by the collaborative-filtering system. When the ratings data increases, the system moves to the collaborative-filtering approach thereby avoiding under-discriminate recommendations of the knowledge-based approach, which requires more knowledge-engineering tasks to be solved.

2.1.3 Conclusion

In conclusion, currently not many commercial e-commerce sites apply recommender systems for suggesting more complex products and services that are rarely purchased by users. Recommender systems for these kinds of products are desirable to help users make good decisions about which products they are going to buy as the products are expensive and the users only purchase the products once in a while. The knowledge-based approach, which is currently used for recommending

these kinds of products, suffers from deep knowledge engineering. On the other hand, the collaborative filtering and content based recommender systems that are popularly used to recommend simple and frequently purchased products rely on a large amount of user ratings data which is not available for infrequently purchased products. Thus, new knowledge must be exploited by recommender systems for recommending complex and infrequently purchased products and new recommendation techniques that can be employed by recommender systems are needed to exploit the new knowledge. This research will explore new knowledge from the user reviews and log data to represent users' preferences and will develop recommendation techniques that can exploit the knowledge to recommend infrequently purchased products. This research also considers an integration of collaborative-filtering and search-based approaches in a hybrid recommender system to enable product recommendations based on the preferences of similar users and to avoid user high involvement.

2.2 DATA MINING AND WEB MINING

2.2.1 Data Mining

Data mining is also known as knowledge discovery in databases (KDD). It is the set of activities used to find new, hidden or unexpected patterns or knowledge from data sources, for example from databases, text files, the web and so on. Data mining has been used by many organisations to access, analyse, summarise and interpret information intelligently and automatically (Chen & Liu, 2005). It has been applied to support various types of application domains including bioinformatics, information retrieval, adaptive hypermedia and electronic commerce. In this section,

data mining techniques will be reviewed with emphasis on the association rule mining techniques that will be used in this research. The application of association rule mining to recommender systems will be also discussed in the review.

The data mining process includes three main steps, namely pre-processing, data mining, and post-processing. The pre-processing step involves preparing data that is suitable for the mining step by cleaning the raw data in order to remove noises and also by selecting relevant attributes for the mining task. The processed data is then used by data mining algorithms to extract patterns or knowledge. Finally, in the post-processing step, the generated patterns are employed to identify useful patterns for applications. There are four major categories of data mining techniques or processing algorithms currently in use, namely classification, clustering, association rule mining, and sequential pattern mining. The association rule mining is the data mining technique that is employed in this research and will be discussed in more detail.

Association rule mining finds interesting correlations among large sets of data items. It shows sets of items that occur frequently together in a given dataset. The knowledge extracted by this data mining algorithm is in the form of if-then statements, where the "if" part is called an antecedent and the "then" part is called a consequence. Other than that, there are two numbers that express the degree of uncertainty about the rule or the strength of the rule. They are called the 'support of a rule' and 'the confidence of a rule', respectively. The 'support of a rule' is an indication of how frequently the items appear in the database. It measures the significance of the rule and it is expressed by the percentage of transactions that include all items in the antecedent and consequent parts of the rule. The 'confidence of a rule' indicates the number of times the rule have been found to be true. It

measures the degree of correlation between itemsets and it is expressed by the percentage of transactions that contain the antecedent under the condition that this transaction also contains the consequence. For example, given a rule, A -> B, the support and confidence of this rule are computed as follows:

$$Support = \frac{(A \cup B).count}{n}$$

$$Confidence = \frac{(A \cup B).count}{A.count}$$

There are two kinds of association rule mining approaches, which are based on frequent itemset generation (for example the Apriori algorithm) or based on decision tables (for example the Rough Set Theory approach). In the Apriori algorithm (Agrawal & Srikant, 1994), there are two steps involve which are:

(i) Generate all frequent itemsets –itemsets that have transaction support above minimum support, and

(ii) Generate the desired rules from the frequent itemsets - confident association rules that have confidence above minimum confidence.

A huge number of rules is often generated by this kind of association rule mining and this results in the 'interestingness' problem, which causes difficulty to the user who needs to analyze and find only useful rules (Liu, 2007).

The Rough Set concept was proposed by Zdzislaw Pawlak in the 1980's. This approach provides efficient algorithms for finding hidden patterns in data and generates sets of decision rules from the data. The Rough Set data analysis starts from a data set that is also called a decision table or an information system. A decision table contains a set of objects relating to the decision problem, with rows corresponding to objects, columns to attributes and entries in the table are attribute values. From this table, attributes can be partitioned into two classes called condition and decision classes. Decision rules induction can be performed by determining the

decision attributes values based on condition attributes values. Four steps are involved in Rough Set association rule mining and they are: data selection, data pre-processing, reduction and decision rules induction. In a data selection task, target tables, dimensions, attributes and records are selected from transaction database into data mining database. In data pre-processing, incomplete records are handled and attribute values are discretized or categorized to reduce data amounts and dimensions. In addition, one of the attributes is selected as the decision attribute in this process. In the reduction process, the information system is reduced so that it contains sets of attributes that cannot be eliminated further without losing some information from the system. Finally, decision rules are generated from the reduced information table through determining the decision attributes values based on condition attribute values.

Association rule mining has been widely applied in the collaborative filtering recommender system in order to improve the recommendations results and to solve the recommender system's problems. For example, Garcia, Romero, Ventura and Castro (2008) combined association rule mining and collaborative filtering recommender techniques to improve e-learning courses. Association rule mining was applied locally on students' usage data for an online course to extract if-then recommendations rules; and collaborative filtering recommender techniques were applied, which filter and organise recommendation priorities depending on the votes registered by experts and teachers with similar profiles. Sandvig, Mobasher and Burke (2007) introduced a robust recommendation algorithm based on the association rule mining technique to minimize the effect of profile injection attacks on a recommender system. A profile injection attack happens when multiple false users' profiles are inserted by attackers intend to bias recommendation, to promote

their products or demote the competitors' products. Frequent items sets that are generated for the association rules as abstraction from the original user profiles minimize the influence of attack as attack profiles are not directly used in recommendation. Leung, Chan and Chung (2007) proposed the CLARE (Cross-Level Association Rules) algorithm that integrates content information about domain items into collaborative filters for generating cold-start recommendations. The attributes of the domain items are considered by this algorithm to enable it to recommend cold-start items and thus the number of recommendable items can be increased.

2.2.2 Web Mining

Web mining is an application of data mining techniques that discovers and analyses useful information from the World Wide Web. The web involves three types of data: (i) web content data, which includes several types of data such as textual, image, audio, video, metadata as well as hyperlinks; (ii) the web log data regarding the users' behaviour while they browse the web; and (iii) the web structure data concerning the inter-document structure of the web (Madria, Bhowmick, Ng & Lim, 1999). Web mining is categorized into three areas of interest, namely: web content mining, web structure mining and web usage mining based on which part of the web is to be mined (Borges & Levene, 2000; Kosala & Blockeel, 2000; Madria et al., 1999). By using data mining techniques, web content mining aims to extract useful knowledge from web content data; web structure mining aims to generate a structural summary about web sites from the structure of the hyperlinks of the websites; and web usage mining aims to automatically discover patterns of usage from web servers' logs.

Web mining techniques can be used to solve information overload problems and so the techniques have been deployed in recommender system methodologies to enhance the quality of product recommendation. For example, Cho and Kim (2004) proposed a recommendation methodology based on web usage mining to analyse users' shopping behaviours on the web and collect their implicit ratings, thereby reducing the sparsity problem of the collaborative-filtering recommender system. In addition, an instance of web content mining called text mining, which applies the data mining technique to unstructured text data, has also been employed to extract useful information from user reviews of products in order to provide better recommendations. For example, Aciar, Zhang, Simoff and Debenham (2007) developed a recommender system that employed text-mining techniques to extract useful information from review comments to generate product recommendations. Text mining that utilizes consumer opinion about products to discover useful information is called opinion mining. The following section will discuss web usage mining and opinion mining in more detail.

2.2.2.1 Web Usage Mining

Web usage mining is the process of applying data mining techniques to the discovery of user behaviour while the user interacts with the Web (Kosala & Blockeel, 2000; Pierrakos, Paliouras, Papatheodorou & Spyropoulos, 2003). They described the web usage mining process in terms of basic data mining stages, which comprise four processes, namely: data collection, data pre-processing, pattern discovery and knowledge post-processing as shown in Figure 2.2.

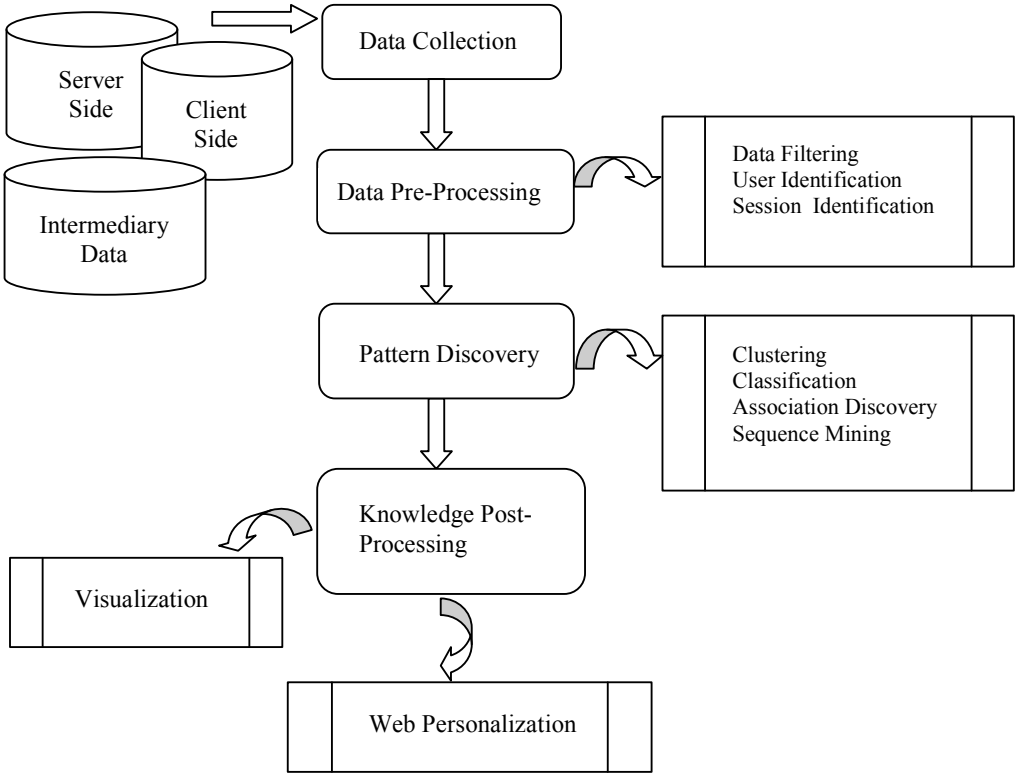


Figure 2.2: The Web Usage Mining Process (From Pierrakos et al., 2003)

In the first stage, usage data is collected from Web servers, clients that connect to a server, or intermediary sources (proxy servers and packet sniffers). Then the raw web data is processed to prepare it in a necessary form to be used for pattern discovery. The data pre-processing tasks include data filtering/cleansing, user identification and session identification. The data cleansing step involves merging the Web logs from multiple servers, removing irrelevant and redundant log entries (filenames with suffixes such as gif, jpeg, map, count.cgi, etc) and parsing of the logs. User identification and session identification are also required in data pre-processing to track individual user's behaviour at the web site from the log data. Next, the pattern discovery tasks involve the discovery of knowledge by applying

machine learning and statistical techniques such as clustering, classification, association discovery and sequence pattern discovery to the data. The knowledge extracted from the pattern discovery task is then presented in a form that is comprehensible to humans such as in a report or in a visualization form in the knowledge post-processing stage. In web personalization, the extracted knowledge is incorporated in a personalized module in order to facilitate the personalization functions.

Interesting information about user navigation patterns extracted by web usage mining can be used for various purposes such as to personalize the delivery of web content; to improve user navigation through pre-fetching and caching; to improve web design; or to improve the user satisfaction in an e-commerce site. Web usage mining can help in producing a personalized web-based system by making the system adaptive to the needs and interests of the individual user. This is due to the ability of web usage mining to construct user models that represent the interests and the behaviour of users from usage data. These user models can be used by the personalization system automatically without the involvement of any human expert and thus contribute to the development of a robust and flexible web personalization system (Pierrakos et al., 2003).

The most common applications that provide personalized content, services or items to potential consumers are recommender systems. Web usage mining has been deployed in recommender system applications to improve the scalability, accuracy and flexibility of the systems (Mobasher, Cooley & Srivastava, 2000). Examples include:

(i) EntreeC (Burke, 2000), a restaurant recommender system that manipulates navigation actions from a web server's log data as

implicit ratings to enhance the effectiveness of the collaborative-filtering algorithm in a hybrid knowledge-based/collaborative recommender system. In this system, the initial suggestions are given by the knowledge-based technique and as the system's database of ratings increases, it moves beyond the knowledge base to characterize users more precisely by using collaborative filtering technique. Two main parts of the collaborative filtering engine are the generation of ratings from the navigation actions and the computation of inter-user similarity. The positive/negative ratings or numeric ratings are used to reflect the user's original action based on the preference information the system collects from users such as entry point, exit point, browse and critique, to convert the user session into a vector of ratings. Then, two users are compared using standard collaborative filtering algorithms and return previously-unseen items in a standard collaborative filtering manner.

(ii) L-R (Ishikawa et al., 2002), a web recommender system that recommends relevant pages to the users based on the web content and the user models, which are constructed by mining the user access logs. The user models are constructed by applying classification method based on the information extracted from the Web access logs such as IP address, domain names, host names, access time, OS, browser names, and keywords. The recommendations are generated for the users based on the generated user models. The probability of the user's transition from one page to another is also calculated by using

the Web access logs and this measure is used to recommend the relevant pages to the users.

(iii) WebCF-PT (Cho & Kim, 2004), a recommendation methodology that combines web usage mining and product taxonomy, capturing implicit ratings by tracking a user's shopping behaviour on the web and applying them to enhance the recommendation quality and the system performance of the current collaborative filtering recommender systems. There are four phases of the WebCF-PT procedure which are grain specification, customer profile creation, neighbourhood formation, and recommendation generation. In the grain specification phase, similar products are grouped together using product taxonomy to reduce the product space. Then, in the customer profile creation phases, the customer profile is constructed based on three general shopping steps in Web retailers: click-through, basket placement and purchase. The weight is assigned based on the preference order between products that is {products never click} < {products only clicked through} < {products only placed in the basket} < {purchased products}. In the neighbourhood formation phase, the similarity between customers is computed and neighbourhood is formed between a target customer and a number of like-minded customers. Finally, in the recommendation generation phases, top-N recommendation is derived from the neighbourhood of customers.

2.2.2.2 Opinion Mining

Currently, it is common for e-commerce websites to enable their users to write reviews or comments about products they have purchased. The information

from the users' reviews is valuable for helping other potential users to decide whether to buy an item based on other users' experiences and opinions about a particular product. In addition, manufacturers can also gather the users' feedback from the online reviews in order to improve their products. However, with the increase in the number of users buying products, the number of reviews also grows over time and it is not possible for the users or manufacturers to read all reviews to know previous users' opinions about a certain product. In addition, some of the reviews are long, making it difficult for the users or manufacturers to recognize good and bad features about the product when deciding whether the product is worth purchasing; or for the manufacturers to decide whether the product needs to be improved. A review summarization process that can summarize whether a user provides a good or bad review about a certain product, is valuable and highly desirable for the potential users and the manufacturers to easily collect useful information about products from a large number of reviews, thereby helping them in making decisions based on the summarized information.

The idea of opinion mining and summarization is proposed by Hu and Liu (2004). Recently, automatic review mining and summarization has become a popular research topic (Ding, Liu & Yu, 2008; Hu & Liu, 2004; Popescu & Etzioni, 2005; Zhu & Balaji, 2006; Zhuang, Jing & Zhu, 2006). Review mining and summarization, also called opinion mining, aims at extracting product features on which the reviewers express their opinion and determines whether the opinions are positive or negative (Zhuang et al., 2006). The difference between opinion mining and traditional text summarization is that the former only mines features of products from the reviews and identifies whether the opinions are positive or negative; it does not rewrite any of the original sentences from the review as occurs in traditional text

summarization (Hu & Liu, 2004). The main tasks in opinion mining are (i) to find product features that have been commented on by reviewers; (ii) to identify opinion sentences in the review and their semantic orientation (whether the opinion sentence is positive, negative or neutral); and (iii) to summarize the discovered information and present it in a format that is suitable for helping users' decision making processes.

Hu and Liu (2004) proposed a model of feature-based opinion mining and summarization that uses a lexicon-based method to determine whether the expressed opinion of a product feature is positive or negative. The opinion lexicon or the set of opinion words used in this method is obtained through a bootstrapping process using the WordNet. Popescu and Etzioni (2005) improved the method based on relaxation labelling to allow the system to identify user opinions and their polarity with high precision and recall. Zhuang et al. (2006) used the same technique but implemented it in a specific domain – for analysing movie reviews. Next, Ding et al. (2008) proposed a technique that performs better than the previous methods by using the holistic lexicon-based approach. This technique deals with context dependent opinion words and aggregating multiple opinion words in the same sentence, which were the two main problems of the previous techniques.

Despite the growth in the number of online reviews and the valuable information that they can provide, little work has been done on utilizing online user review comments for creating recommendations. Wietsma and Ricci (2005) only used review comments for product descriptions and explanations about a product's recommendation, and not for recommending products. A recommender system that makes use of review comments for making recommendations was proposed by Aciar et al. (2007). They employed text mining techniques to extract useful information

from review comments and then mapped the review comments into the ontology's information structure that the recommender system can use to make recommendation. In their proposed recommender system, the product recommendation is based on the reviewer's expertise with the product and the reviewer's valuation of the product features. Thus, the ontology generated contains two main parts which are opinion quality and product quality that summarize the consumer's skill level and the experience with the product under review, respectively. To make recommendations, a user is required to input the model of the product and to select the features that they are most concern with. A set of measures such as opinion quality (OQ), feature quality (FQ), overall feature quality (OFQ), and overall assessment (OA) are used by a ranking mechanism to compute a product's rating based on the ontology data. Their method helps to overcome the cold starting problem of the current collaborative-filtering technique when the products have not been rated by enough users, by obtaining ratings of products from textual information in review comments.

Product reviews are valuable information sources to be exploited to increase users' understanding about products and to accelerate a user's decision making. The use of reviews may provide better recommendations than the use of only user ratings as they provide other users' comments and experiences about particular products as well as experts' reviews about products, which may increase users' confidence and help them in the decision making process. Therefore, user reviews should be utilized in recommender systems and more research is needed to find appropriate techniques for exploiting this useful source of information in order to improve recommendations of products to users.

2.2.3 Conclusion

With the boom of E-commerce applications nowadays, a great deal of user generated contents such as user click streams data, user reviews data, blog and tags are available to be utilized by recommender systems in order to understand users' preferences. Much research has been conducted to extract useful information from these resources by using data mining and web mining approaches to improve product recommendations. This research will explore user reviews and user click streams data to generate useful information in a form that the proposed recommender models can use. Even though some researchers have exploited knowledge extracted from user reviews data, none of them has utilized the knowledge to expand the user's query in order to represent the user preferences. The proposed approaches will make use of knowledge extracted from user reviews data to expand a user's query in order to represent the user's preferences more precisely. In addition, the proposed approaches are different from the available approaches that have utilized user click streams data to extract knowledge about the user preferences, as they will consider the main features of the products that the users are most concern with to generate user profiles from user click streams data. Therefore, the proposed approaches can be implemented to recommend infrequently purchased products as the product features are important factors for users to choose which product to buy.

2.3 INFORMATION RETRIEVAL

The increasing amount of information and the advent of computers in which to store large amounts of information provide convincing evidence of the need for Information Retrieval systems to retrieve or find useful information from the collections of information stored in the computer systems. The Information Retrieval

(IR) field was born in the 1950s to fill this necessity and over the last fifty years the field has matured significantly (Singhal, 2001). The task of IR is to retrieve unstructured records consisting of text, photographic images, audio and video files. However, with the vast volume of textual data collections on the internet, IR research has focused on retrieval of natural language text from these textual data collections (Greengrass, 2000). Given one of these textual data collections, IR looks for documents that satisfy the user's information need as expressed in a user query. A query contains formal statements of information needs that are entered into an IR system by the user. The documents are thought to be "relevant" if the documents satisfy the user's information need; and otherwise the documents are said to be "non-relevant". To facilitate the access of relevant documents for satisfying users' information needs, IR techniques focus on three basic processes, namely: the development of a representation of queries and documents; the process of comparing query and document presentations to retrieve documents most relevant to an information need;, and the evaluation of documents retrieved.

Ruthven (2008) believes a good query is the one that helps differentiate between relevant objects or non-relevant objects according to the user's information needs. However, queries provided by search engines users are usually short and composed of few keywords (Cao, Nie, Gao & Robertson, 2008; Ma, Chen, Gao & Yang, 2009; Xu, Jones & Wang, 2009). A short query usually contains insufficient information to retrieve documents that satisfy a user's information needs. Consequently, query reformulation is required to represent user information needs more precisely in order to return appropriate results to the user. Previous research has proposed query expansion (QE) techniques to deal with this problem and has shown that retrieval performance can be improved when the queries are reasonably

expanded (Chirita, Firan & Nejdl, 2007; Coa et al., 2008, Wang & Hauskrecht, 2010). The QE techniques will be discussed in more detail in the following sections.

2.3.1 Query Expansion

A query represents a user's information needs. Query expansion is the process of reformulating the user's initial query in order to improve the performance of an information retrieval system. The main aim of query expansion is to append new, meaningful terms to the initial query. Query expansion generates a better query by using additional terms to replace the original query for a new search and to increase the chance of retrieving relevant documents (Chirita et al., 2007; Chawla & Bedi, 2008; Ogilvie, Voorhees & Callan, 2009; Mu & Lu, 2010).

Current query expansion techniques can be classified into global analysis and local analysis (Cui, Wen, Nie & Ma, 2003; Bhogal, Macfarlane & Smith, 2007; Ma et al., 2009). The global analysis technique builds a thesaurus by examining word occurrence and relationships between words in the whole document set. The thesaurus is then used to obtain synonyms or words related to a user query, which are used to expand the query. On the other hand, the local analysis technique examines word occurrences and word relationships from a subset of the initial retrieval results that is returned based on the initial query and uses the selected words to expand a user's query.

Global analysis techniques include term clustering, similarity thesaurus, latent semantic indexing and ontology (Cui et al., 2003). The drawback of the global analysis technique is one of performance because it requires extensive computing resources to analyse the whole collection of documents on the given search engine collection to discover word relationships. The local analysis technique is shown by previous studies to be more effective than the global analysis technique because it is

more query-oriented (Chawla & Bedi, 2008) and focuses only on the most frequently occurring terms in the top ranked documents of the search engine. Local analysis techniques can be categorised into two approaches: i) relevance feedback which is based on the relevance judgement given by the user and ii) pseudo-relevant feedback which is based on top ranked n documents assumed to be relevant (Cui et al., 2003). The relevant feedback and pseudo-relevant feedback approaches will be discussed in the following subsections.

2.3.2 Relevant Feedback

The basic idea of the relevance feedback in information retrieval is to extract useful user feedback information from the relevant documents retrieved by a search engine based on the user's initial query (Chirita et al., 2007). The cycle of the relevance feedback approach starts with an initial query inputted by a user. Then, a user will be presented with a list of retrieval results based on her or his initial query, which would be assessed by the user to indicate those that are relevant. Based on the documents judged relevant by the user, expansion terms are extracted to generate a new query to retrieve a new set of documents for presenting to the user. Much research has been done for query expansion using Relevance Feedback (RF). Salton and Buckley (1990) did experiments on six test collections to investigate the relative performance of twelve feedback algorithms. Okabe and Yamada (2007) proposed a query expansion that uses only minimal user feedback (which is one relevant document) to get other relevant documents by using transductive learning. This approach attempted to increase the number of pseudorelevant documents from which expansion terms can be extracted. The results of their experiments show their method performs well in instances of small numbers of relevant documents being supplied by users. Yamout, Oakes and Tait (2007) developed a relevance technique called

Weight Propogation (WP), where documents judged relevant propagate positive weights to documents close by in vector similarity space, while documents judged non-relevant propogate negative weights to such neighbouring documents. The process is repeated for all the relevant and non-relevant documents and the positive and negative weights are summed for each document. Those documents with the highest weighting are retrieved as the results for the relevance feedback. Their technique improved computational time and the retrieval quality of the existing RF technique since the documents are treated as independence vectors rather than being merged by a single vector.

The success of query expansion using relevance feedback in information retrieval requires a user to have good judgement as to the relevance of the documents retrieved. If the user provides sufficient and correct feedback, this approach can achieve very good performance. This is because good quality expansion terms can only be generated from a large number of relevant documents (Ruthven, Tombros & Jose, 2001). However, users are usually unwilling to spend time assessing each document and selecting a subset of relevant documents as feedback , which explains why an automatic relevance feedback is needed where the search engine can get relevant documents from which to extract expansion terms without relying to too great an extent on user participation. As a result, Pseudo-relevance feedback becomes the commonly used approach for query expansion in IR (Cui et al., 2003) as it provides automatic local analysis, which entails no heavy burden on the user.

2.3.3 Pseudo-Relevant Feedback

In this approach, the manual process of selecting relevant documents by a user is replaced by assuming the top-ranked documents returned from the initial query are relevant and doing the relevance feedback based on this assumption. This

approach is the most effective among all the query expansion approaches (Xu & Croft, 1996). It has been found to be effective in previous Text Retrieval Conference (TREC) experiments (Cui et al., 2003; Harman & Voorhees, 2006). A key aspect of the performance of an expanded query is the 'term selection' method which includes the selection and weighting of the new terms or expanded terms (Bhogal et al., 2007). With regards to term selection, Cao et al. (2008) studied the usefulness of expansion terms and found that not all expanded terms determined by the pseudo-relevant approach based on the traditional term distribution (such as the most frequent terms) are useful. They proposed a term classification method to select only a small proportion of the useful expansion terms and utilize additional criteria such as co-occurrences of the expansion term with the original query terms and proximity of the expansion terms to the query terms (that is based on the distance of the co-occurred terms which is determined based on the minimum number of words between the two words among all co-occurrences in the documents) in their method. Lv and Zhai (2010) proposed a positional relevance model (PRM) that exploits term position and proximity evidence to assign more weight to words positioned closer to query words. The experiment results show that their method performs significantly better than other relevance models.

As the volume of data on the web becomes larger, other resources have emerged to select good candidate terms for query expansion. Billerbeck, Scholer, Williams and Zobel (2003) utilize past user queries that are associated with highly ranked documents returned by the initial query to select expansion terms. Their method shows that query associations are highly effective sources of expansions. Cui et al. (2003) proposed a query expansion method that extracts the relationships between query terms and document terms by mining user logs. These relationships

are then used to get expansion terms for new queries. Their approach has been shown as an effective way of selecting high-quality expansion terms to improve retrieval performance. Xu et al. (2009) explored the utilization of Wikipedia in pseudo-relevance feedback for query dependent expansion. In their method, the corresponding Wikipedia entity pages are used as the pseudo-relevant information instead of the top-ranked documents from the test collection. Their experiment results show that their approach outperforms a baseline relevance model.

The most recent of query expansion approaches is to use ontology to expand the original query based on the additional terms that have a semantic relationship with the original query. Ma et al. (2009) proposed a query expansion method based on the ontology-described knowledge to select expansion terms that are semantically related with the initial query. From the domain knowledge resource formalized by ontology, semantic diagraphs for combination of words in the query are generated with each query term as the first vertex based on the domain knowledge. Then, the distance between the first vertex (v_0) and each vertex (v_i) in the semantic diagraph is calculated based on the weight of the relationship between any pairs of vertexes in the nearest path between the first vertex (v_0) and the vertex (v_i). According to the threshold, the expanded terms are selected from each semantic diagraph. Finally, all the terms gotten from the semantic graphs are combined using logical operators to obtain the terms for query expansion. Expansion terms selected using their method are semantically related with the initial query and have improved retrieval results.

As discussed before, query expansion has been widely applied in information retrieval to improve the effectiveness of retrieving relevant documents. For online product search, the query-driven approach is also implemented by most of the e-commerce websites to enable users to find their products of interest. A user provides

product attribute values as the query, and the system will suggest relevant products that match the user's query. Therefore, query expansion is an approach that can potentially be applied in online product searches to represent users' information need accurately by supplementing the users' queries with additional product attribute values that they might like. This thesis proposes methods to expand a user's query for an online product search. Additional attributes to expand the user's query can be extracted from the user data such as product reviews and product click streams. The proposed query expansion method for online product search will be discussed in Chapter 3.

2.4 CHAPTER SUMMARY

This chapter reviewed the background and previous work in areas related to the thesis focus. It reviewed recommender systems, data mining and information retrieval. Most collaborative filtering methods are focused on data sets with explicit ratings. However, in many practical situations, such explicit ratings data are not available because this area of knowledge acquisition requires intensive user involvement. For example, for products that are not regularly purchased by users, it is not possible for the users to provide sufficient ratings for use by the recommender system to generate meaningful recommendations, as users buy this kind of products once in their lifetime and thus, they cannot provide ratings or feedbacks for products they never have. Thus, another area that attracts new development of recommendation techniques is the analysis of implicit feedbacks. The new user information in Web 2.0 provides new solutions to profile users and makes recommendations based on user profiles. In addition, users' browsing activities can also be easily collected when users interact with the system to find the products they

want to purchase. However, little work has been done to exploit the rich user information from this data especially to incorporate the information with domain knowledge to improve the recommendations. The following chapters will discuss the proposed recommendation approaches that utilize user reviews and user click streams data and also incorporate domain knowledge for recommending infrequently purchased products.

Chapter 3: Query Expansion based on Knowledge Extracted from User Reviews

Currently, many e-commerce sites for selling infrequently purchased products such as cars or cameras only provide a standard matching-based system for customers to search for products. The standard matching-based system provides basic search functions that take the user's initial query as input and return a set of matched products to the query. Usually, a user is required to provide some attributes values of the product that she or 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 and may be inadequate to represent the user's preferences. If the user's preferences can be predicted, based on these preferences, products that are most likely meet the user's interests can be recommended without getting more involvement from the user.

In this chapter, a query expansion method is proposed to generate a new query for a target user based on the user's preferences predicted from the online user reviews. In the proposed method, namely the Opinion Mining-based Query Expansion (OMQE), a user's preferences for product attribute values are predicted for the target user by using the association rules between attribute values based on online user reviews. The predicted attribute value preferences are used to generate a new query by expanding the initial user's query given by the target user. Instead of

using the user's initial query which has lack of information about the actual user preferences, a new query generated by the proposed method represents the user preferences more precisely and may retrieve and recommend products that best meet the user's requirements.

3.1 OPINION MINING-BASED QUERY EXPANSION (OMQE)

Nowadays, with the growth of e-commerce applications, users are given more opportunities than ever to express and share their opinions with other users in online users' reviews about products they have had experiences with. The online reviews provided by the users contain valuable information that can be utilized by recommendation approaches to understand users' preferences and to recommend products to new users. The opinions about a product provided in an online user review reflect the user's viewpoints concerning the product based on their experience of using the product. A review with a positive orientation indicates that the reviewer (that is, the user) was satisfied with the product in some aspects. This means that at least some attributes of this product were attractive to the user. By identifying these attractive attributes for each product, and based on these attributes, the products that will be of most interest to new users of the products can be determined. In this thesis, user reviews with positive orientations are used to extract association rules between product attribute values from those products that received good comments from previous users.

The association rules between product attribute values generated from user reviews can be used to predict other product attribute values that may be preferred by target users based on the initial attribute values they provide in their queries. These attribute values can be used to expand an initial query to generate a new query that

can represent their preferences more precisely. The OMQE approach applies opinion mining and rough set association mining techniques to generate association rules between attribute values of products. This OMQE approach consists of three main parts as follows:

(i) Opinion mining to detect the orientation of each review and re-present each review as a structured review.

(ii) Association rule mining to extract association rules between attribute values from those products that received positive comments.

(iii) Query expansion to expand the target user's initial query by utilizing the association rules extracted from the positive user reviews.

Figure 3.1 shows the architecture of the OMQE method and Table 3.1 shows the main procedure of the OMQE method. Section 3.2 will discuss the opinion mining technique to determine the orientation of each user review. Section 3.3 will discuss the rough set association rule mining to extract association rules between product attribute values. The query expansion method to expand the user's query by utilizing the generated association rules will be explained in section 3.4.

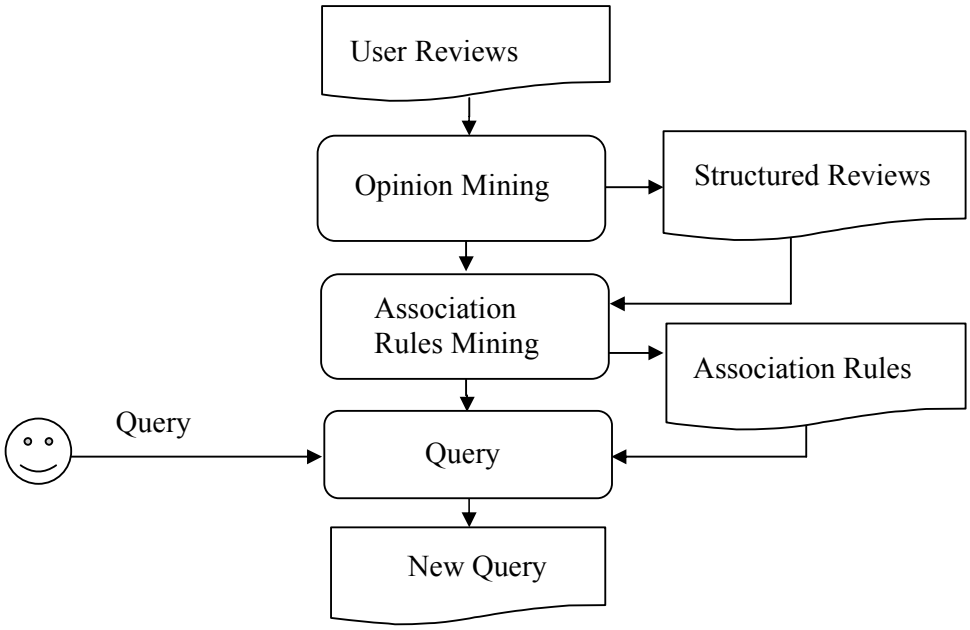


Figure 3.1: The Architecture of the OMQE Method

Table 3.1: The Main Procedure of the OMQE Method

Procedure 3.1

Begin

1. Determining user reviews' orientations
2. Determine the orientation of each opinion word for each feature
3. Determine the orientation of each feature
4. Determine the orientation of each user review
5. Generating rules between product attribute values from positive and neutral user reviews
6. Select products from positive and neutral user reviews as objects in the decision table
7. Select condition and decision attributes of the products
8. Generate rules between condition and decision attribute values using Rough Set Association Rule Mining
9. Expanding the user's initial query
10. Select candidate rules
11. Select a final rule with maximum accuracy from the candidate rules set
12. Expand the user's query with the decision attribute values of the final rule
13. Product Recommendations
14. Match product attribute values with the expanded query
15. Calculate the similarity between each product and the expanded query
16. Rank the products based on the product similarity with the expanded query
17. Select Top-N products

End

3.2 USER REVIEWS ORIENTATION DETECTION

A product P_j can be represented by two-tuple (A, F) of information. A can be described by a set of attributes representing the technical characteristics of the product defined by domain experts and each attribute can have a set of possible values. Suppose that there are m attributes $A_1, A_2, \dots, A_m$ for a product P_j, each attribute A_i has a set of possible values, $\{a_{i1}, a_{i2}, \dots, a_{im_i}\}$, and a product P_j can be represented by a vector of attribute values, i.e. $P_j =< a_1, a_2, \dots, a_m >$, $a_i \in \{a_{i1}, a_{i2}, \dots, a_{im_i}\}$, $i = 1,2, \dots, n$. In addition, F can be described by a set of usage features representing the usage performance of the product defined by domain

experts or users of the product. The usage features are usually the aspects commented upon by the users of the product. Suppose that there are n features $F_1, F_2, \ldots, F_n$ for a product P_j .

In this thesis, both the product attributes and usage features are assumed to have already been specified. For instance, for the online car search domain on which the experiments of the OMQE are conducted, examples of the car attributes A_i are "make", "model", "year", "price", "body type", "standard transmission", and so on; and the usage features F_l are "comfort practicality", "price equipment", "under bonnet", "how drives", "safety security", "quality reliability", "servicing running costs", "aesthetics styling", and so forth.

Opinion mining techniques can be used to determine the orientation of user reviews by identifying whether the users gave positive or negative opinions about the products being reviewed. In many e-commerce websites, the product features to be reviewed have been specified so that users can provide their comments and opinions on each particular feature. For reviews that are not classified according to any specific feature, opinion mining techniques can also be used to identify the product features that are addressed by each sentence in a review (Hu and Liu, 2004). In this thesis work, the sentences in each review are assumed to have been divided into groups, each of which consists of the sentences that talk about one feature of the product. Let $R = \{R_1, R_2, \ldots, R_m\}$ be a review, R_i be a set of sentences about feature f_i. Opinion mining techniques can be applied to generate the user's sentimental orientation o_i concerning feature f_i, where $o_i \in \{positive, negative, neutral\}$. Based on the sentimental orientations o_i of all the features, an overall orientation O_{all} can be determined.

This thesis adopted the approach proposed by Hu and Liu (2004) to perform the opinion mining task. Hu and Liu's work contains 3 subtasks :-(i) identifying features of the product that customers have expressed their opinions on (called *product features*); (ii) for each feature, identifying review sentences that give positive or negative opinions; and (iii) producing a summary using the discovered information. In the proposed approach, only subtask (ii) of the Hu and Liu's work is applied in which to determine the orientations of the opinion words and the features. The opinion mining task in the proposed approach includes:- (i) determining the orientation of each opinion word used to describe a feature; (ii) determining the orientation of each feature of the product reviewed by a user; and (iii) determining the orientation of each user review based on the orientations of all the features of the reviewed product.

(i) Opinion Word Orientation Detection

The user's sentimental orientation towards each feature of the product indicates whether the user likes or dislikes the product in terms of this feature. The orientation of each feature can be determined based on the orientations of all the opinion words (for example good, amazing, poor, and so on) used to describe this feature. The opinion words used by the user to express his or her opinion of a product are identified by extracting all adjectives in the review. Then, the orientation of each opinion word is identified by utilizing the adjective synonym set and antonym set in WordNet (Miller, Beckwith, Fellbaum, Gross & Miller, 1990). In WordNet, adjectives share the same orientation as their synonym and opposite orientations as their antonyms. The adjective synonym and antonym set in WordNet (Hu & Liu, 2004) and a set of seed adjectives with known orientation are utilized to

predict the orientation of a target opinion word. Let $W_i = \{w_{i1}, w_{i2}, \dots, w_{in}\}$ be the set of opinion words extracted from review R_i concerning feature f_i; let $OW_i = \{ow_{i1}, ow_{i2}, \dots, ow_{in}\}$ be the corresponding orientation of each opinion word. For each word w_{ik}, the seed adjective set and the adjective synonym and antonym sets are searched to find the opinion word's synonym or antonym with known orientation. If a synonym of the opinion word is found, ow_{ik} is set to the same orientation as the synonym and the seed list is updated. Otherwise, if the opinion word's antonym is found, its orientation is set to the opposite of the antonym and the word is added to the seed list. The process is repeated for all the target opinion words with unknown orientation and the words' orientations are identified using the updated seed list.

(ii) Feature Orientation Detection

The sentimental orientation o_i for each feature f_i can be identified by finding the dominant orientation of the opinion words in R_i through counting the number of positive opinion words with $ow_{ik} = positive$ and the negative opinion words with $ow_{ik} = negative$ as follows:

a. $total_{\mathcal{POS}}(R_i) = \sum_{p=1}^{n} positive(ow_{ip})$,

$$positive(ow_{ip}) = \begin{cases} 1, & ow_{ip} == positive \\ 0, & ow_{ip} == negative \end{cases}$$

b. $total_{\mathcal{NEG}}(R_i) = \sum_{p=1}^{n} negative(ow_{ip})$,

$$negative(ow_{ip}) = \begin{cases} 0, & ow_{ip} == positive \\ 1, & ow_{ip} == negative \end{cases}$$

If the number of positive opinion words $total_{\mathcal{POS}}(R_i)$ is larger than the number of negative opinion words $total_{\mathcal{NEG}}(R_i)$, the orientation o_i of the feature f_i is positive, otherwise negative. If the number of positive opinion

words $total_{\mathcal{POS}}(R_i)$ equals the number of negative opinion words $total_{\mathcal{NEG}}(R_i)$, the orientation o_i of the feature f_i is neutral.

$$o_i = \begin{cases} positive, & total_{\mathcal{POS}}(R_i) > total_{\mathcal{NEG}}(R_i) \\ negative, & total_{\mathcal{POS}}(R_i) < total_{\mathcal{NEG}}(R_i) \\ neutral, & total_{\mathcal{POS}}(R_i) == total_{\mathcal{NEG}}(R_i) \end{cases}$$

(iii) User Review Overall Orientation Detection

The overall orientation O_{all} indicates whether the user's opinion about the product is positive or negative. The overall orientation O_{all} of each review R can be determined based on the orientations o_i of all the features $f_1, f_2, \ldots, f_m$ by calculating the number of positive features, neutral features and negative features for the review.

a. $total_{\mathcal{POS}}(R) = \sum_{l=1}^{m} positive(o_l)$,

$$positive(o_l) = \begin{cases} 1, & o_l == positive \\ 0, & o_l == negative \\ 0, & o_l == neutral \end{cases}$$

b. $total_{\mathcal{NEG}}(R) = \sum_{l=1}^{m} negative(o_l)$,

$$negative(o_l) = \begin{cases} 0, & o_l == positive \\ 1, & o_l == negative \\ 0, & o_l == neutral \end{cases}$$

c. $total_{\mathcal{NEUT}}(R) = \sum_{l=1}^{m} neutral(o_l)$,

$$neutral(o_l) = \begin{cases} 0, & o_l == positive \\ 0, & o_l == negative \\ 1, & o_l == neutral \end{cases}$$

If the number of positive features and neutral features is more than the number of negative features, the overall orientation O_{all} for the review R is

positive, otherwise negative. If the number of positive features and neutral features is equal to the number of negative features, the overall orientation O_{all} for the review is neutral.

$$O_{all} = \begin{cases} positive, \mid (total_{POS}(R) + total_{NEUT}(R)) > total_{NEG}(R) \\ negative, \mid (total_{POS}(R) + total_{NEUT}(R)) < total_{NEG}(R) \\ neutral, \mid (total_{POS}(R) + total_{NEUT}(R)) == total_{NEG}(R) \end{cases}$$

The products with a positive orientation must possess attractive attributes or characteristics that pleased their users. Based on this idea, the products with positive or neutral orientations are selected to be used in the rough set rules mining to generate associations between product attributes values.

The detailed procedure for determining a user review's orientation is shown Table 3.2. Section 3.3 will explain the rough set association rules mining in more detail.

Table 3.2: The Detailed Procedure for Determining a User Review's Orientation

Procedure 3.2

Begin

1. For each feature in a user review
2. Extract all adjectives for each feature as opinion words
3. Determine the orientation of each opinion word
4. Search the adjective seed list and the Wordnet synonym and antonym set
5. If a synonym of the opinion word is found
6. Set the opinion word's orientation with the same orientation as its synonym
7. Else if an antonym of the opinion word is found
8. Set the opinion word's orientation with the same orientation as its antonym
9. Update the seed list
10. Determine the orientation of the feature
11. Calculate the number of positive opinion words
12. Calculate the number of negative opinion words
13. If the number of positive opinion words more than the number of negative opinion words
14. Set the feature's orientation as positive
15. Else if the number of positive opinion words less than the number of negative opinion words
16. Set the feature's orientation as negative
17. Else if the number of positive opinion words equal to the number of negative opinion words
18. Set the feature's orientation as neutral
19. Determine the orientation of the user review
20. Calculate the number of positive features
21. Calculate the number of negative features
22. Calculate the number of neutral features
23. Calculate the total number of positive and neutral features
24. If the total number of positive and neutral features more than the number of negative features
25. Set the user review's orientation as positive
26. Else if the total number of positive and neutral features less than the number of negative features
27. Set the user review's orientation as negative
28. Else if the total number of positive and neutral features equal to the number of negative features
29. Set the user review's orientation as neutral

End

3.3 ROUGH SET RULES MINING

The rough set theory was firstly introduced by Pawlak in 1985. It is a mathematical approach to deal with the inexact, uncertain and vague knowledge in data analysis. This approach can be considered as a formal framework for discovering facts from imperfect data (Walczak & Massart, 1999). It provides efficient algorithms for finding hidden patterns in data, minimal sets of data, evaluating significance of data and generating sets of decision rules from data (Pawlak, Polkowski & Skowtron, 2005). In this research, the rough set rule mining technique is employed to discover hidden patterns about attribute values of products from the positive user reviews and it expresses them as decision rules. The rough set decision rule mining technique is chosen as this technique can be used to easily select the condition and decision attributes of the rule and thus the association rule mining process can generate only the necessary rules for use in the query expansion task. The generated decision rules show the relationships between attribute values of the products that have been liked by the previous users. These rules can be used to predict the new user's preferred attribute values based on the associations between attribute values of the products that are preferred by the previous users.

Rough set data analysis starts from a data set that is called an information system which can be represented as a table. In the information system table, each row represents an object of interest, each column represents an attribute, and entries of the table are attribute values. An attribute can be a variable or an observation or a property. The object-attributes pair in the information system can be formally represented as $I = (S, T)$, where S is the universe or a non-empty finite set of objects, and T is a set of attributes for each object. In this thesis work, the rough set decision rule mining aims to generate associations between product attribute values

from a set of products that have received good comments from the previous users. Let ℙ be a set of products that has received users' reviews for a product p in ℙ, let $p.O_{all}$ be the overall orientation of the product; the set of objects for the rule mining are the products that received positive reviews, that is, $S = \{p | p.O_{all} = positive, p \in \mathrm{P}\}$. This means that each product p with the positive overall orientation is treated as an object in S in the information system $I = (S, T)$ and the attribute values of the selected products, i.e. $A, A_2, \dots, A_m$ are treated as the object's attributes, that is, $T = \{A_1, \dots, A_m\}$.

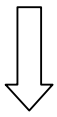
Decision rules induction requires the partitioning of the attributes into condition and decision attributes. The information system I is partitioned into two disjointed classes of attributes, called condition attributes C and decision attributes D and therefore it is often called a decision table. The decision table is denoted by $I = (S, C, D)$, where S is a set of objects, C and D are disjoint sets of condition and decision attributes, respectively. The decision table is used to study if the attributes of objects expressed in C can be expressed in terms of attributes in D. Decision rules between attribute values can be generated from the decision table through determining the decision attributes values based on the condition attributes values. The expression of a decision rule is in the form "if...then..." or in symbols $c \rightarrow d$, where c and d are called the condition and decision, respectively. The quality of the rule is indicated by its strength, which represents the number of observations or cases that accord with that rule.

In this thesis work, the attributes chosen as the condition are the product attributes that are usually provided by a user as the initial input in his or her query and the decision attributes are other attributes of the products that are not given very often as the initial user's query. For example, for the cars domain in which the

experiment is conducted in this study, three attributes that are usually provided by the users when searching for cars, that is make, model, and year are selected as the condition attributes; and three attributes, that is body type, engine size and standard transmission are selected as the decision attributes. From the decision table, decision rules between condition and decision attribute values of the products can be generated. These decision rules can be used to predict other possible attribute values of the products that the user may be interested in according to the values given by the user in the search form. For example in the car domain, the rules are used to predict the values for the decision attributes such as body type, engine size and standard transmission based on the values of the condition attributes such as make and model given in the user's query. In this study, the rough set association rule mining tool ROSETTA (Ohrn, 2000) is employed to generate association rules from the decision table generated based on the user reviews. Figure 3.2 illustrates a decision table with the car attribute values and the example of rules generated from the table after applying the rough set rule mining technique.

Logical rules generated from the decision table are used to support new decisions. The condition attributes of the selected rule specify the decision which should be made if conditions determined by the condition attributes, are satisfied. In the proposed OMQE method, the rules are used to model the relationships between the initial product attribute values given by a user in his or her query with other product attribute values that may be of interest to the user. From the rules, if the condition values are matched with the initial input given by the user in his or her query, the decision values which contain other product attribute values are used to expand the user's query in order to represent the user's preferences more precisely. Section 3.4 will discuss the query expansion method in more detail.

	Condition			Decision		
Product	**Make**	**Model**	**Year**	**BodyType**	**Engine Size**	**StdTrans mission**
P_1	Subaru	Forester	>2000_To_2005	WAGON	2.5	4A
P_2	Subaru	Forester	>2000_To_2005	WAGON	2.5	4A
P_3	Subaru	Forester	>2000_To_2005	WAGON	2.5	4A
P_4	Subaru	Forester	>2000_To_2005	WAGON	2.5	4A
P_5	Subaru	Forester	>2000_To_2005	WAGON	2.5	4A
P_6	Subaru	Forester	>2000_To_2005	WAGON	2.5	5M
P_7	Subaru	Forester	>2000_To_2005	WAGON	2.5	5M
P_8	Subaru	Forester	>2000_To_2005	WAGON	2.5	5M
P_9	Subaru	Forester	>2005	WAGON	2.5	5M
P_{10}	Subaru	Forester	>2005	WAGON	2.5	5M
P_{11}	Subaru	Forester	>2005	WAGON	2.5	5M
P_{12}	Subaru	Forester	>2005	WAGON	2.5	4A
P_{13}	Subaru	Forester	>2005	WAGON	2.5	4A
P_{14}	Subaru	Forester	>1995_To_2000	WAGON	2.5	5M
P_{15}	Subaru	Forester	>1995_To_2000	WAGON	2.5	5M



R1: Subaru, Forester, >2000_To_2005 -> Wagon, 2.5, 4A accuracy 0.625

R2: Subaru, Forester, >2000_To_2005 -> Wagon, 2.5, 5M accuracy 0.375

R3: Subaru, Forester , >2005->, Wagon, 2.5, 5M accuracy 0.6

R4: Subaru, Forester , >2005->, Wagon, 2.5, 4A accuracy 0.4

Figure 3.2: An Example of a Decision Table and the Rules Generated

3.4 QUERY EXPANSION

In information retrieval, the query expansion aims to improve the initial user's query by adding new words and phrases to the existing search terms to generate an expanded query (Cui, Wen, Nie & Ma, 2002). To represent the user's requirements more accurately, in a product search, the query expansion involves adding more attribute values that might be of user interest, to the initial product attributes values given in the user's query. The attribute values given by the user as the initial query show the features of the product that are preferred by the user. Based on these attribute values, more product attribute values that may be of interest to the user can be predicted by using association rules between attribute values which are generated from the products that have been positively reviewed by the previous users. To select a rule to expand the query, the attribute values given by a user in the initial query are matched with the condition attribute values of the rules. The rule that has the condition values matched with the initial attribute values given by the user and that has the highest accuracy, is selected to expand the user's query with the attribute values in its decision part. The accuracy of a rule is calculated based on the percentage of objects or products that match the condition part of the rule under the condition that this object also contains the decision part of the rule. For example based on objects in Figure 3.2, the accuracy calculation for rule R1 is as follows:

R1 : If Make(Subaru) and Model(Forester) and Year(>2000_To_2005) Then BodyType(Wagon) and EngineSize(2.5) and StdTransmission(4A)

8 objects match the If-part of rule R1, 5 objects are also members of the decision class in the Then-part of rule R1. Thus the accuracy of the rule is 5/8=0.625.

Let $R = \{r_1, r_2, \dots, r_{|R|}\}$ be a set of rules generated based on the user reviews, and each rule r_l can be represented by a set of attribute values $r_l =< a_1^l, a_2^l, \dots, a_k^l, a_{k+1}^l, \dots, a_m^l >$ with k condition attributes and $m - k$ decision attributes. Each rule r_l has an accuracy value acc_l to represent its strength. Let $Q^q = \{a_1^q, a_2^q, \dots, a_p^q\}$ be a user's initial query containing attribute values that are provided by the user to the search engine. Rules r_l that have the condition value a_i^l and match the user's query attribute value, i.e. $a_i^l \in Q^q,\ i = 1,2,..,k$, are selected as the candidate rules $CR = \{r_1, r_2, \dots, r_{|CR|}\}$. A rule r_l with maximum accuracy acc_l is chosen to expand the user's query. If more than one rule in the candidate rule set CR has the same maximum accuracy, acc_l, one of the rules is chosen randomly to expand the query. If there is no rules match with the query, one of the rules that has partial match with the query is selected to expand the query.

The query expansion method involves adding attribute values from the decision part of the selected rule r^s to the existing search values to generate an expanded query. Let $r^s =< a_1^s, a_2^s, \dots, a_k^s, a_{k+1}^s, a_{k+2}^s, \dots, a_m^s >$ be a selected rule for the initial query Q^q , where $a_1^s, a_2^s, \dots, a_k^s$ match the attributes in the user's query; the attribute values $a_{k+1}^s, a_{k+2}^s, \dots, a_m^s$ can be used to expand the initial query $Q^q$to generate an expanded query Q^e i.e. $Q^e = Q^q \cup \{a_{k+1}^s, a_{k+2}^s, \dots, a_m^s\}$. Table 3.3 show the detailed procedure of the query expansion.

The expanded query Q^e can represent the user's preferences more precisely than the user's initial query because it contains more attribute values of the products that the user wants to find. These attribute values are gathered from the products that are positively reviewed by the previous users, and thus, the expanded query may retrieve products that most likely satisfy the target user's requirement. Figure 3.3 illustrates an example of the query expansion process.

Table 3.3: The Detailed Procedure of the Query Expansion

Procedure 3.3

Begin

1. Select candidate rules
2. For each rule
3. Calculate the number of attribute values in the condition part of each rule that match with the attribute values in the user's initial query
4. If all attribute values in the condition part of the rule match with the attribute values in the user's initial query
5. Select the rule as a candidate rule
6. If no rule has all condition attribute values match with the user's query
7. Select candidate rules with the maximum number of attribute values match with the user's initial query
8. For each rule
9. If the rule has the maximum number of attribute values in its condition part that match the attribute values in the user's query
10. Select the rule as a candidate rule
11. Select a final rule with maximum accuracy from the candidate rules set
12. Expand the user's query with the decision attribute values of the final rule

End

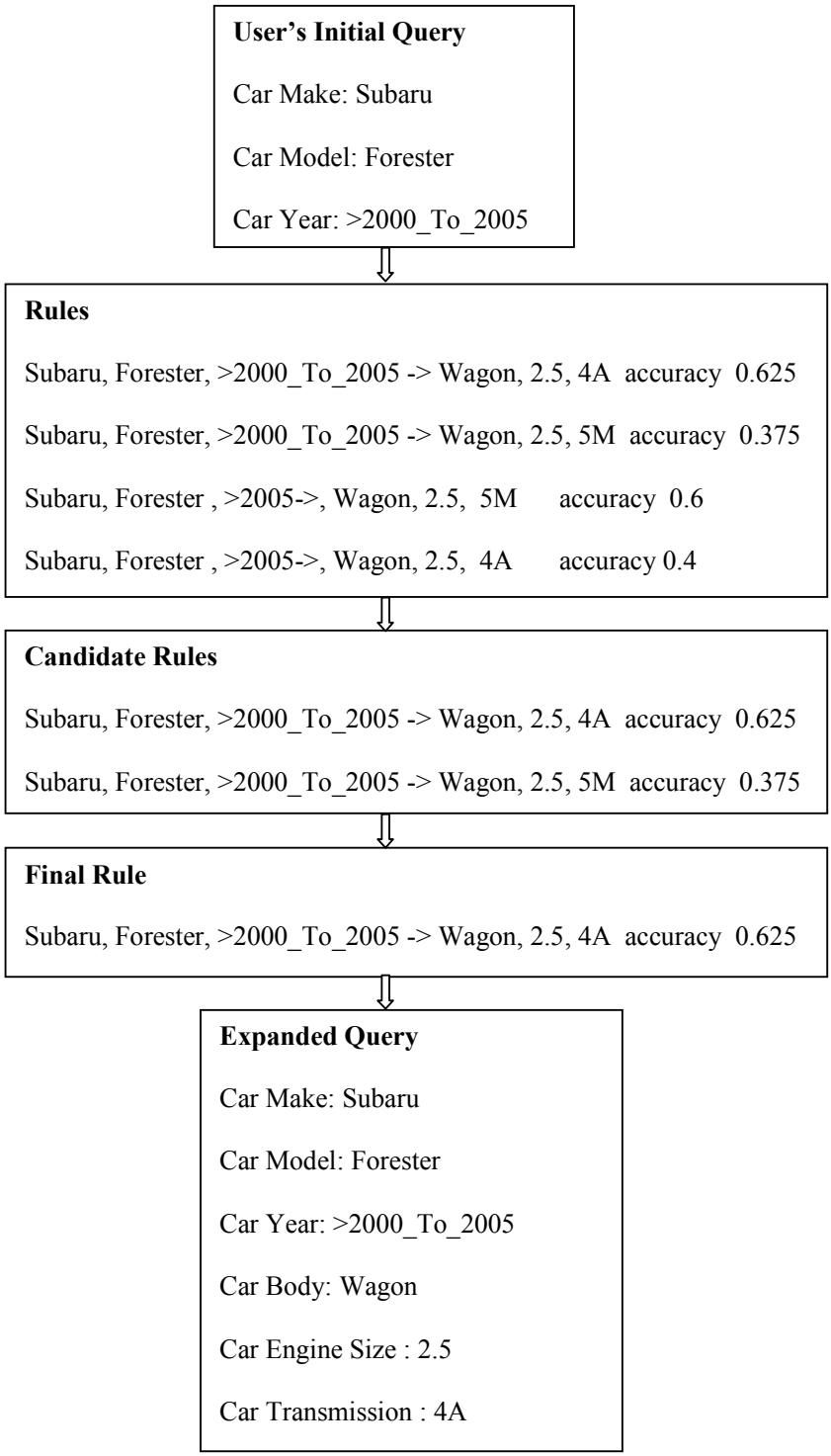


Figure 3.3: An Example of the Query Expansion Process

3.5 PRODUCT RECOMMENDATIONS

Usually there are two types of tasks that can be performed by recommender systems: rating predictions and top *N* item recommendations. The rating prediction task is to predict the rating value a target user would give to an unrated item. The top *N* recommendation task is to recommend a set of unrated/new items to the target user (Deshpande & Karypis, 2004). If explicit ratings are available for a recommender system, both tasks can be performed. However, for recommending infrequently purchased products, the ratings prediction for a product cannot be performed because there is no explicit rating available and thus, in this thesis work, the focus of recommendation-making is on recommending the top N best items to the target user.

Let U be the set of all users and DB be the set of possible products which can be recommended to the user. The problem of product recommendation is defined as predicting how much a target user u would be interested in an unseen product P_l, that is, $\mathcal{R}(u, P_l)$, in order to generate Top N of ordered products $< P_1, P_2, \dots P_N >$ to the user u, where $P_l \in DB$ and $\mathcal{R}(u, P_1) \geq \mathcal{R}(u, P_2) \geq \cdots \geq \mathcal{R}(u, P_N)$. The strength of $\mathcal{R}(u, P_l)$ is determined by ranking the Top N retrieved products based on the similarities between product P_l and the query or the user profile of the target user u.

In the product retrieval using the expanded query Q^e, a set of products $\{P_1, P_2, P_3 \dots\}$, whose attribute values $a_{jk} \in P_j$ match the attribute values $a_i^e \in Q^e$ is retrieved and also ranked, based on the similarity $sim(P_j, Q^e)$ between the products P_j and the expanded query Q^e. Let $P_j =< a_1, a_2, \dots, a_n >$ be a vector of attribute values of a product retrieved using query Q^e; the similarity value $sim(P_j, Q^e)$ between P_j and Q^e is calculated by matching each attribute value $a_l \in P_j$ with the value of $a_l^e \in Q^e$. The following equation can be used to measure the similarity between P_j and the expanded Q^e :

$$sim(P_j, Q^e) = \sum_{l=1}^{n} sim_A(a_{l,}\ a_l^e)$$

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

The products are ranked based on their similarities $sim(P_j,\ Q^e)$ values. Finally, the top-N products are selected and recommended to the user based on their rankings. Table 3.4 shows the algorithm for retrieving and ranking the products for the OMQE approach.

Table 3.4: The Algorithm for Retrieving and Ranking Products of the OMQE Approach

Algorithm 3.1

Input: An expanded query for a target user $Q^e =< a_1^e, a_2^e, \dots, a_n^e >$, a set of products in the database $DB = \{b_1, b_2, b_3, .., b_{|DB|}\}$ where $b_i =< a_{i1}, a_{i2}, \dots, a_{in} >$.

Output: A set of products $\mathcal{F} = \{P_1, P_2, \dots, P_{\Re}\}$ to be recommended

Method:

Begin

1. For each product in the database $DB,\ b_k \in$ DB
2. For each attribute value $a_{kl} \in b_k$
3. If $a_{kl} = a_l^e$
4. $sim_A(a_{kl,}\ a_l^e) := 1$
5. Else
6. $sim_A(a_{kl,}\ a_l^e) := 0$
7. $sim(b_k,\ Q^e) := \sum_{l=1}^{n} sim_A(a_{l,}\ a_l^e)$
8. Rank the products based on $sim(b_k,\ Q^e)$
9. Select top $\Re$ products from the ranked list as final products, $\mathcal{F} := \{P_1, P_2, \dots, P_{\Re}\}$

End

3.6 CHAPTER SUMMARY

The query provided by a user in an online product search is often short and does not represent the user's requirements fully. In this chapter, the OMQE approach was proposed to generate a new query for a user, by utilizing the user preferences extracted from the user review data. The user reviews provided by the previous users reflect the users' viewpoints concerning the quality of the products. The products that are positively reviewed must possess attractive attributes or characteristics that pleased their users. The orientation of each user review is determined by applying the opinion mining technique. The products that receive good comments from the users are used to extract association rules between product attribute values by applying the rough set association rules technique. These association rules are used to predict other attributes values that may be preferred by the user to expand the initial user's query. The new query generated by the proposed method may represent the user's requirements more precisely as it contains more knowledge about the features of the products that are of interest to the user. The query may retrieve more products that are not retrieved by the standard search technique as it represents more attribute values of the products than the user is looking for. Therefore, this method leads to recommendation novelty or serendipity, where more unexpected or different items that meet the users' interests will be recommended to the users. The evaluation results of the proposed OMQE recommendation approach will be given in Chapter 5.

Chapter 4: Integrating Collaborative Filtering and Search-based Recommendations

Currently in e-commerce applications, the search-based approach is still widely applied as the common tool for users to search for infrequently purchased 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 that match the user's query. The search results generated by the standard search engine are not personalized as only products that have the same attribute values or match the user's query will be displayed to the user. In addition, the users' queries may not represent the users' requirements fully because they 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. Besides, the collaborative filtering (CF) approach has been widely applied for recommending frequently purchased products on e-commerce websites. It makes recommendations based on items that similar users have shown interest in the past. The CF approach requires a large amount of ratings data from the users to make meaningful recommendations and thus it is more suitable for recommending frequently purchased products because a large amount of ratings data can be collected from the users when they buy the products repetitively.

The OMQE approach discussed in Chapter 3 aims to expand preferences of a search-based approach based on attribute values of products preferred by the user,

which are extracted from the reviews provided explicitly by him/her. However, many users do not like to provide explicit feedbacks because it requires extra efforts and time. Thus, the explicit data is not always available to be utilized by the recommender systems. Fortunately, the growth of e-commerce applications provides a platform to gather users' data implicitly. For example, web search logs store users' online click or browsing history data, which contains useful information about the users and can be analysed to learn about the users' preferences. The implicit data about users can be used by the CF recommender system when there is no explicit ratings data available - as for recommending infrequently purchased products. In this chapter, a hybrid recommender system that combines search-based and collaborative filtering approaches is proposed to utilize the implicit data for predicting user preferences.

4.1 HYBRID SEARCH-BASED AND COLLABORATIVE FILTERING RECOMMENDER SYSTEM

In the standard collaborative filtering technique, the products that are preferred by the target user's neighbours will be used as the candidates to generate the recommendations. The CF is usually applied for recommending frequently purchased products such as books and movies where many copies of the product are available and can be purchased by other users. However, for online infrequently purchased product searches, there is a problem for directly recommending the products that the user's neighbours preferred in the way that the standard CF method does. For expensive 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. For solving this problem, this thesis work proposes to integrate the collaborative filtering recommendation approach and the search-based approach by utilizing the user's online click stream data to recommend infrequently purchased products to users.

In the proposed hybrid approaches, the CF approach is integrated with the search-based approach to recommend products based on the products that similar users have preferred. Rather than directly recommending the neighbour users' preferred products, three methods are proposed to generate queries based on the neighbour users' profiles or the products viewed by the neighbour users, and then a search of the product collection by using the queries. The search is conducted to find the products that are similar to the products viewed by the neighbour users or similar to the neighbour users' profiles. From these products, the most relevant ones will be chosen and recommended to the target user. The three proposed approaches will combine the CF technique with the search based technique. One approach is to generate a query, which is called Collaborative Filtering-based Aggregated Query (CFAgQuery), by aggregating neighbour users' profiles. The query is then used to retrieve products. The second and third approaches utilize each product preferred by the user's neighbours as the basis of a new query for the search-based approach to use in retrieving similar products to the neighbour's products. The new query captures neighbour users' preferences and provides more detailed content to the query than the original query which may be of interest but may have been missed by the target user when she/he submitted her/his query. Therefore, the product recommendations will be generated based on the new query and also on the similarity between the profiles of the target user and her/his neighbours.

For the second and third approaches, multiple queries which are derived from the products preferred by a user's neighbour 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 (Montague & Aslam, 2002). 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 rankings assigned to documents from one collection are usually not comparable with rankings from another collection due to the size of the collection and different ranking algorithms employed (Zhu & Gauch, 2000). Therefore, information fusion that aims to combine document retrieval results from multiple search engines for improving retrieval effectiveness, is an important issue in the distributed search environment. As in a DIR system, a data fusion technique to merge the responses must be developed for merging the results retrieved by each query of the proposed approaches. For the second approach, the Round Robin algorithm (Si & Callan, 2003) is adopted to merge and rank the products retrieved from the multiple queries of each neighbour user. The second approach is therefore named CFRRobin. In the third approach namely CFMRRobin, the retrieved product lists from different queries are ranked before selecting the final products using the Round Robin algorithm. Figure 4.1 illustrates a general framework of the proposed hybrid collaborative filtering and search-based recommender system.

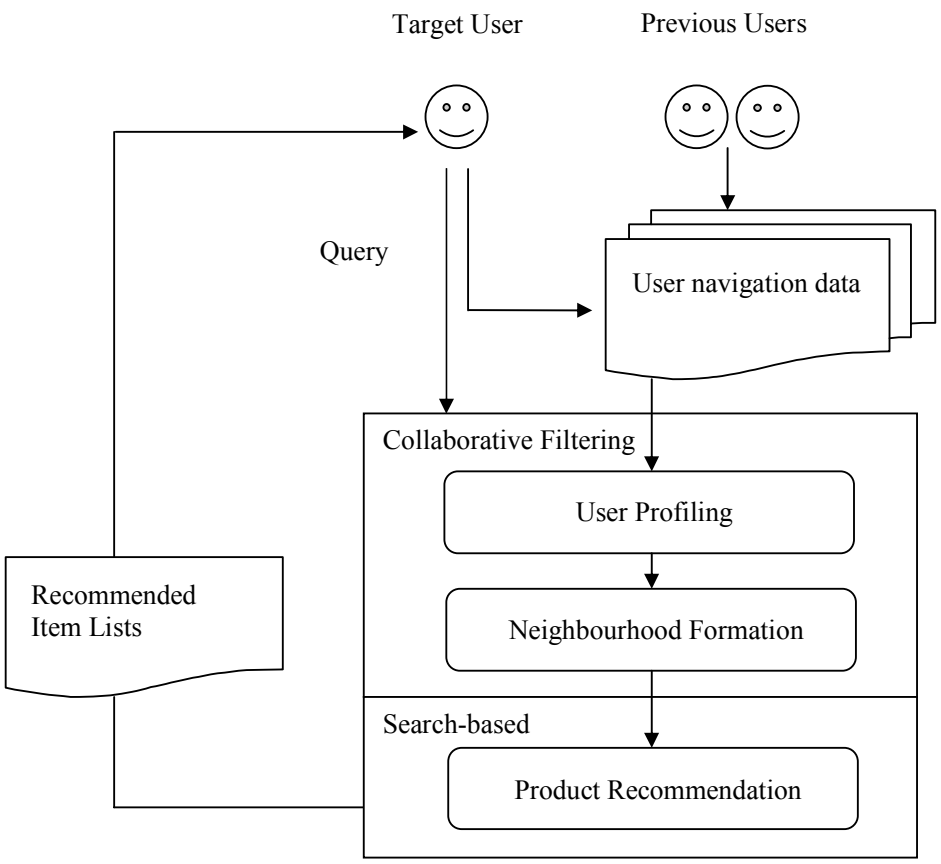


Figure 4.1: A General Framework of the Proposed Hybrid Recommender System

The proposed hybrid approaches involve two main processes which are user profiling and neighbourhood formation. The first step is to generate profiles for the target user and the previous users based on the target user's online click stream data.

After a target user has browsed a few items on an e-commerce site, the user navigation data is utilized by the CF using the user profiling method to generate the user profile or the user's preferences for product attribute values based on the products that have been viewed by the user. Then, a list of the target user's neighbours is generated based on the preference profiles of both the target user and the previous users. The next step is to search for products that might be of interest to

the target user based on the products preferred by the neighbours. While the standard CF will directly recommend the products that are preferred by the neighbours to the target user, in the CFAgQuery approach, the aggregated query generated from the products' attribute values, as favoured by the user's neighbours, is used to search for products. In the CFRRobin and CFMRRobin approaches, search-based approach is applied in the product recommendation task to search for products similar to those preferred by the user's neighbours. The next section will discuss the proposed recommender approaches in more detail.

4.2 USER PROFILING

In this section, user profiling based on the user click stream data of the proposed approaches will be discussed. Before that, the concept of "user session" that will be used throughout this chapter will be defined.

A user session is obtained from the user's product click stream when the user browses products to purchase on an e-commerce site. A user session 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: $P^k =< a_1^k, a_2^k, \dots, a_n^k >$, $k = 1,2, \dots . |S|$ and $a_i^k \in \{a_{i1}, a_{i2}, \dots, a_{im_i}\}$. Alternatively, each product also can be represented as a set of attribute values: $P^k = \{A_1 = a_1^k, A_2 = a_2^k, \dots, A_n = a_n^k\}$.

User profiling plays an important role in providing accurate product recommendations to the end user by understanding and capturing the user's needs or preferences from the user's data. The user profiling process generates a user profile that contains a representation of the user's preferences or interests that can be exploited by the recommendation generating process to recommend new potentially

relevant items to the user. User profiling uses the user's data that can be gathered either explicitly or implicitly from the user. Explicit data such as ratings, demographic information and reviews must be provided by the user and this necessity places an additional burden on the user. In some circumstances, few users are willing to provide this data. For instance, for infrequently purchased products such as cars and houses, the explicit data may not be sufficiently accumulated from users as users possesses only a few such 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 provide personalized recommendations with little participation from the users.

Click stream data is a kind of search log that can be collected by the search engine implicitly without user extra effort. Click stream data shows the path a user takes through a website. For online product searches, after providing an initial query to the search engine, a user may click on some suggested products that she or he is interested in. From the user's click stream data, a list of products that have been viewed by a user can be obtained. This online click stream data shows that the user has more interest in the viewed products compared to other products. The products that have been viewed by a user contain attribute values that attract the user. By analyzing all the user's preferred products' attribute values gathered from the user click stream data, the user's interests or preferences for each product attribute value can be predicted.

In this thesis work, a user profile is represented by a set of attribute values and their weights, which show the strengths of the user's preferences for each product attribute value. For example, in the car domain, one of the car attributes is "body type". The user preferences for the body type include the strengths of all the

possible attribute values of the body type such as “coupe”, “hatchback”, “sedan”, and “wagon”. A user profile contains the strengths of all the user’s preferences for attribute values of all the car attributes such as make, model, year, and transmission. The weight of each product attribute value in a user profile shows how much a user likes the attribute value. The user profiles can be used to retrieve products that have attribute values matching the attribute values preferred by the target user.

As mentioned above, a list of products viewed by a user in the click stream data that is generated when the user browses an e-commerce site is called a user’s session i.e. $S = \{P^1, P^2, \dots, P^{|S|}\}$. Product P^k in the user’s session can be represented as a vector of attributes $P^k =< a_1^k, a_2^k, \dots, a_n^k >$ or a transaction of attributes $P^k = \{A_1 = a_1^k, A_2 = a_2^k, \dots, A_n = a_n^k\}$, where $a_i^k \in \{a_{i1}, a_{i2}, \dots, a_{im_i}\}$ for attribute A_i. If a product is treated as a transaction of all attribute values defined as $\{a_{11} = 0, a_{12} = 0, \dots, a_1^k = 1, \dots, a_{1m_i} = 0, \dots\dots, a_{n1} = 0, a_{n2} = 0, \dots, a_n^k = 1, \dots, a_{nm_i} = 0\}$, from a set of products viewed by a user, a product transaction dataset of $|S|$ transactions can be constructed for the user, where each product in S can be represented as a vector of attribute values: i.e. $P^k =< a_{11} = 0, a_{12} = 0, \dots, a_1^k = 1, \dots, a_{1m_i} = 0, \dots\dots, a_{n1} = 0, a_{n2} = 0, \dots, a_n^k = 1, \dots, a_{nm_i} = 0 >$, $k = 1,2, \dots . |S|$ and $a_i^k \in \{a_{i1}, a_{i2}, \dots, a_{im_i}\}$. An example of the structure of a product transaction dataset is shown in Figure 4.2. The product transaction dataset represents all the possible values a_{ij}^k of all the attributes A_i for each product P^k in a user session S. The rows of the transaction set represent the products viewed by a user, and the columns of the transaction set represent the attribute values of the products viewed by the user.

	A_1				A_2				....	A_n			
	a_{11}	a_{12}	....	a_{1m_1}	a_{21}	a_{22}	....	a_{2m_2}	....	a_{n1}	a_{n2}	....	a_{nm_n}
P^1	a_{11}^1	a_{12}^1	....	$a_{1m_1}^1$	a_{21}^1	a_{22}^1	....	$a_{2m_2}^1$	....	a_{n1}^1	a_{n2}^1	....	$a_{nm_n}^1$
P^2	a_{11}^2	a_{12}^2	....	$a_{1m_1}^2$	a_{21}^2	a_{22}^2	....	$a_{2m_2}^2$	....	a_{n1}^2	a_{n2}^2	....	$a_{nm_n}^2$
P^3	a_{11}^3	a_{12}^3	....	$a_{1m_1}^3$	a_{21}^3	a_{22}^3	....	$a_{2m_2}^3$	....	a_{n1}^3	a_{n2}^3	....	$a_{nm_n}^3$
P^k	....	....	....	....	....	....	....	....	....	....	....	....	....
....	....	....	....	....	....	....	....	....	....	....	....	....	....
$P^{\|S\|}$	$a_{11}^{\|S\|}$	$a_{12}^{\|S\|}$	....	$a_{1m_1}^{\|S\|}$	$a_{21}^{\|S\|}$	$a_{22}^{\|S\|}$	....	$a_{2m_2}^{\|S\|}$	....	$a_{n1}^{\|S\|}$	$a_{n2}^{\|S\|}$	....	$a_{nm_n}^{\|S\|}$

Figure 4.2: Products Transaction Structure for a Session

From the transaction dataset, the frequency $freq(a_{ij})$ of each attribute value a_{ij} for attribute A_i can be obtained. In this thesis, a user's product interest for an attribute value is represented by the frequency of the attribute value of all 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 is formally represented as:

$$up =< ua_{11}, \dots, ua_{1m_1}, ua_{21}, \dots, ua_{2m_2}, \dots, ua_{nm_n} >$$

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

where ua_{ij} denotes the user's interest/preference to the jth value of attribute A_i. The user preference ua_{ij} for each attribute value a_{ij} is calculated based on the number of products with the attribute value a_{ij} among all products that have been viewed by a user. It shows the user preference strength for each attribute value a_{ij} of attribute A_i among all the products preferred by the user. Figure 4.3 shows an example of a transaction dataset for a user session. Assume that the user has viewed five products, that is $S = \{P^1, P^2, \dots, P^{|5|}\}$; 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 denoted as 1, otherwise 0 in the transaction dataset.

	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

Figure 4.3: An Example of the Product Transaction for a User

The user profile for this product transaction is shown below:

$$up =< 0.6,0.4,0,0,0,0,0.8,0,0.2,0,0.6,0.4 >$$

Neighbourhood formation is a key component of the collaborative filtering approach in which a set of similar users or neighbours for a target user is generated. In this thesis work, the "K-Nearest-Neighborhood" formation approach is adopted to select the top K user's neighbours based on the similarity between the target user profile and the previous user profiles. The two most popular approaches to calculate distance or similarity measure between users in collaborative recommender systems are correlation and cosine-based (Adomavisius & Tuzhilin, 2005). This thesis uses the cosine similarity method to calculate the similarity value between the two users. Let $tu =< ta_1, ta_2, \dots ta_m >$ be a target user's profile and $pu_l =< ua_{l1}, ua_{l2}, \dots, ua_{lm} >$ be a previous user's profile created, based on the user click stream data. The equation to calculate the similarity between the target user and each of the previous users is given below:

$$sim(tu, pu_l) = \frac{\sum_{j=1}^{m} ta_j \, ua_{lj}}{\sqrt{\sum_{j=1}^{m} (ta_j)^2} \sqrt{\sum_{j=1}^{m} (ua_{lj})^2}} \qquad (4.1)$$

In the neighbourhood formation, the preference similarities between the target user profile tu and the previous user pu_l are determined by calculating the users' similarities in terms of their product preferences for each product attribute value, which are ta_j and ua_{lj} in the target user profile tu and the previous user profiles pu_l respectively**.** The top- $\Re$ previous users who are highly similar to the target user are selected as the target user's neighbours $\{B_1, B_2, \ldots, B_{\Re}\}$.

In lightweight semantics approach, meaningful metadata about information resources enable intelligent resource processing resulting in advanced services such as recommendation or personalized search. The proposed user profiles is a form of lightweight metadata representation for storing the semantics about user preferences to product attribute values, which is acquired from the products users chose during the search process. The user profiles are utilized to improve the current methods of recommendation specifically for recommending infrequently purchased products.

4.3 INTEGRATING THE CF APPROACH AND SEARCH-BASED APPROACH USING QUERY AGGREGATION

In the proposed Collaborative Filtering-based Aggregated Query (CFAgQuery) approach, an aggregated user query is generated for a target user based on the preferences of the target user's similar or neighbour users. In this method, the collaborative filtering approach is applied to derive a new query for a target user based on the preferences of the similar users. Figure 4.4 illustrates a general framework of the query aggregation approach. Firstly, the product attribute preferences of the target user are generated based on the user's online click stream

data by using the user profiling method discussed in Section 4.2. Then, the neighbourhood formation is performed to find neighbour users who have similar product preferences as the target user based on the target user and the previous user profiles.

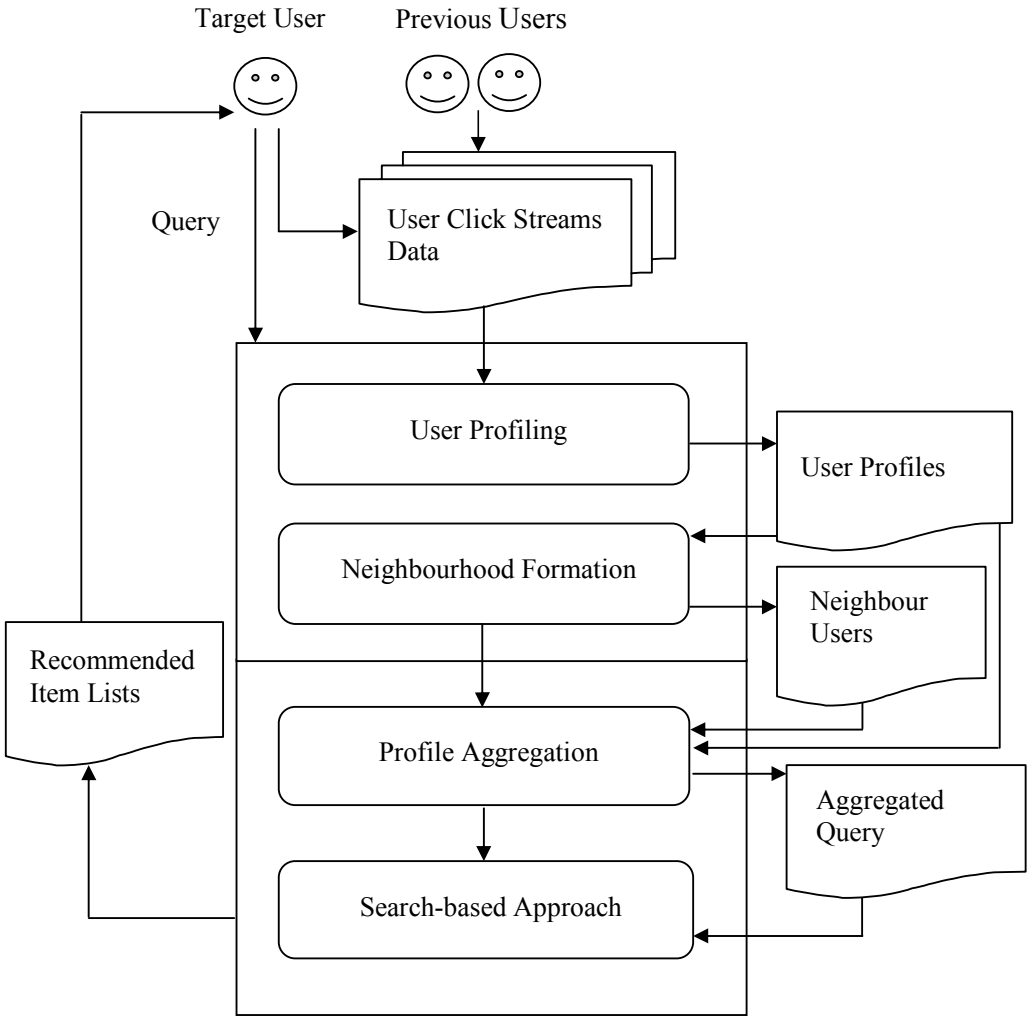


Figure 4.4: A General Framework of the CFAgQuery Approach

In the profile aggregation, the preferences of all the neighbours are aggregated to generate the target user's preferences for attribute values based on the preferences of the neighbours and the similarities between the neighbours and the target user. From the aggregated profile, the maximum value for each attribute in the aggregated profile is used to generate an aggregated query for the target user. The aggregated query contains the product attribute values that are of most interest to the similar users and may represent the target user's preferences more accurately according to the preferences of the similar users. The following sections will discuss the preferences generation, the profile aggregation and the query generation in more detail.

Let $\{B_1, B_2, \ldots., B_{\Re}\}$ be the target user's neighbours and $pu^k =< ua_{11}^k, \ldots, ua_{1m_1}^k, ua_{21}^k, \ldots, ua_{2m_2}^k, \ldots, , ua_{n1}^k, \ldots ua_{nm_n}^k >$ be the user profile of neighbour B_k which is generated using the user profiling method discussed in section 4.2. By combining pu^1, pu^2, ..., $pu^{\Re}$ for all the neighbours, an aggregated profile $pu^{ag} =< ua_{11}^{ag}, \ldots, ua_{1m_1}^{ag}, ua_{21}^{ag}, \ldots, ua_{2m_2}^{ag}, ua_{n1}^{ag}, \ldots, ua_{nm_n}^{ag} >$ can be generated for the target user u.

Each attribute value ua_{ij}^{ag} in the aggregated query is calculated using the following equation:

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

where $sim(u, B_k)$ is the similarity between u and its neighbour B_k, $sim(u, B_k)$ and can be calculated using the Equation (4.1) given in section 4.2.

$ua_{k1}^{ag}, \ldots, ua_{km_k}^{ag}$ measures the preference strength of the target user for each attribute value of attribute A_k based on the viewpoints of the target 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 $a_k^{ag} = max_{j=1}^{m_k}(ua_{kj}^{ag})$. Table 4.1 shows the algorithm of the CFAgQuery approach and Figure 4.5 illustrates the profile aggregation task of the approach where S_{B_i} refers to the sessions of neighbour user B_i.

Table 4.1: The Algorithm of the CFAgQuery Approach

Algorithm 4.1

Input: A set of user's neighbours $\{B_1, B_2, \dots., B_{\Re}\}$, neighbour profile $pu^k = < ua_{11}^k, \dots, ua_{1m_1}^k, ua_{21}^k, \dots, ua_{2m_2}^k, \dots, ua_{nm_n}^k >, k = 1, \dots, \Re$

Output: An aggregated query $AQ^u = \{a_1^{ag}, a_2^{ag}, \dots, a_n^{ag}\}$

Method:

Begin

1. For each attribute value a_{ij}
2. Calculate the preference of the target user u to a_{ij}

$$ua_{ij}^{ag} := \frac{\sum_{k=1}^{\Re}(sim(u,B_k) * ua_{ij}^k)}{\sum_{k=1}^{\Re} sim(u,B_k)}$$

3. For each attribute A_i

$$a_i^{ag} := max_{j=1}^{m_i}(ua_{ij}^{ag})$$

4. Return $\{a_1^{ag}, a_2^{ag}, \dots, a_n^{ag}\}$

End

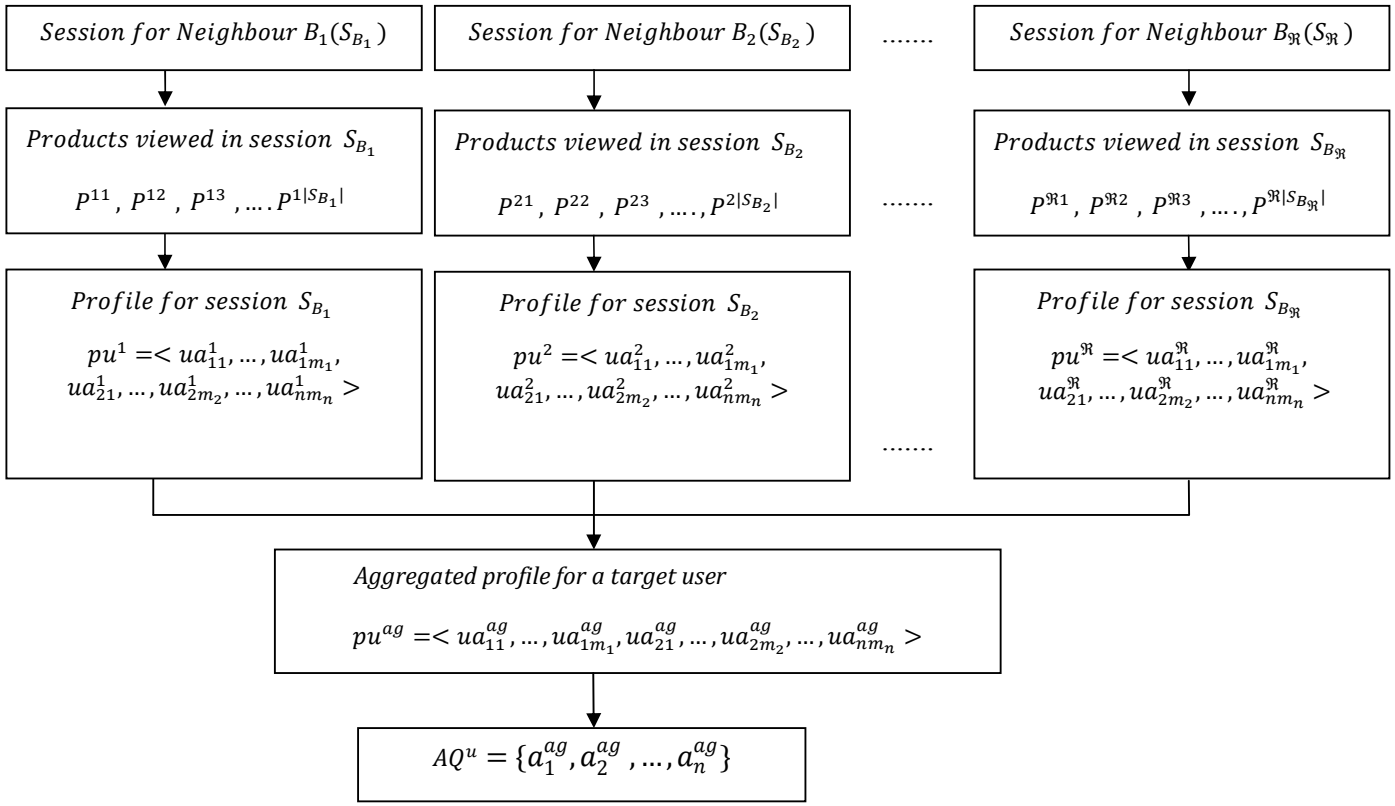


Figure 4.5: Profile Aggregation of the CFAgQuery Method

4.4 INTEGRATING THE CF APPROACH AND SEARCH-BASED APPROACH USING THE ROUND ROBIN FUSION METHOD

Figure 4.6 illustrates the framework of the CFRRobin approach. This approach uses each product of the user's neighbour 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 (Si & Callan, 2003) to generate a set of candidate products. Figure 4.7 illustrates the product retrieval and merging for each query derived from a neighbour user's preferred products to generate a candidate product set for each target user's neighbour. The products from all the candidate product sets of all the neighbours are then ranked and final products are selected for recommendation.

In the neighbourhood formation of the CF technique, a set of neighbours is generated for the target user, i.e. $\{B_1, B_2, \ldots., B_{\Re}\}$ and $S_{B_i} = \{P^{i1}, P^{i2}, \ldots, P^{i|S_{B_i}|}\}$ represents a set of products viewed by the neighbour 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. $P^{ij} =< a_1^{ij}, a_2^{ij}, \ldots, a_n^{ij} >$ are used as a query Q^{ij} to retrieve products that have similar attributes from the product database. That is, $\forall P^{ij} \in S_{B_i}$, $Q^{ij} = \{A_1 = a_1^{ij}, A_2 = a_2^{ij}, \ldots, A_n = a_n^{ij}\}$ is a query containing the attributes of a product P^{ij} that the neighbour B_i is interested in. A set of products, $\{b_1^{ij}, b_2^{ij}, \ldots\}$, whose attributes match the attributes in Q^{ij} is retrieved and also ranked based on the similarity $sim(b_k^{ij}, Q^{ij})$ between the product 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 $b_k^{ij} = \{A_1 = a_1,\ A_2 = a_2, \ldots, A_n = a_n\}$, the following method can be used to measure the similarity:

$$sim\left(b_k^{ij},\ Q^{ij}\right) = \sum_{l=1}^{n} sim_A\left(a_{l,}\ a_l^{ij}\right) \qquad (4.2)$$

$$sim_A\left(a_{l,}\ a_l^{ij}\right) = \begin{cases} 1,\ a_l == a_l^{ij} \\ 0,\ a_l \neq a_l^{ij} \end{cases}$$

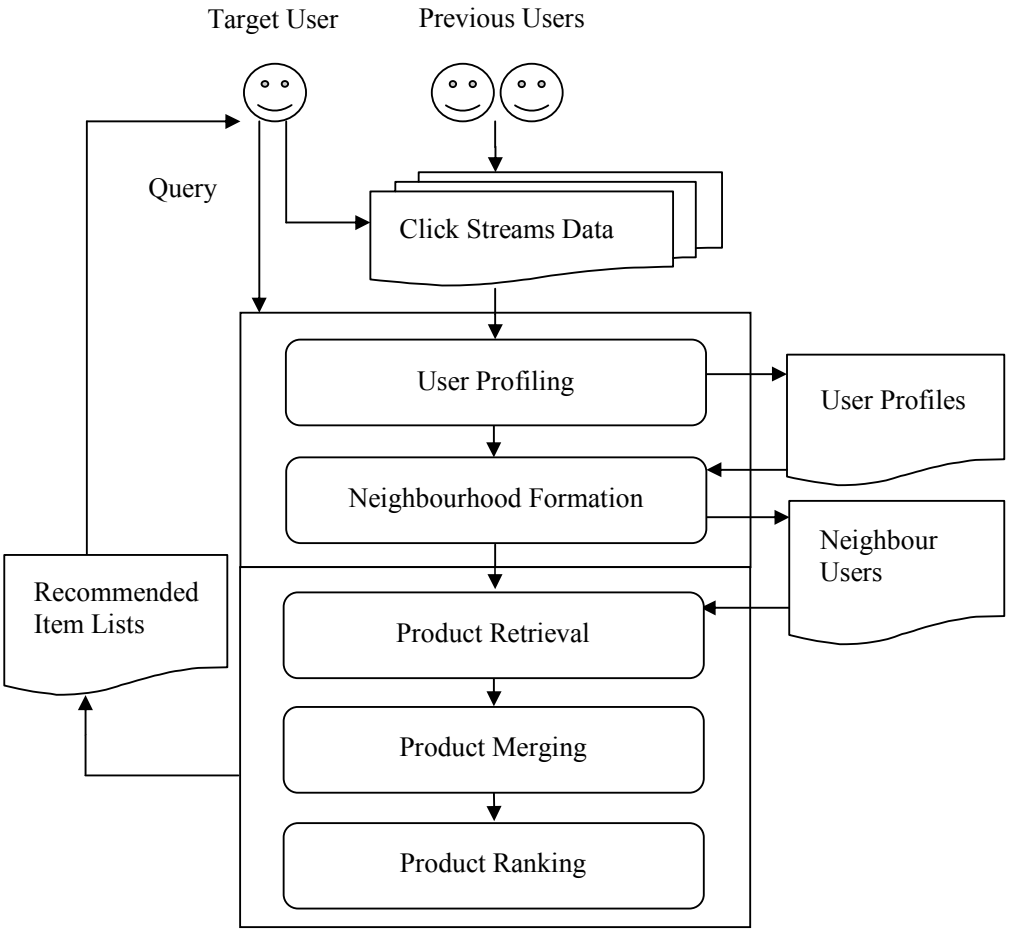


Figure 4.6: A General Framework of the CFRRobin Approach

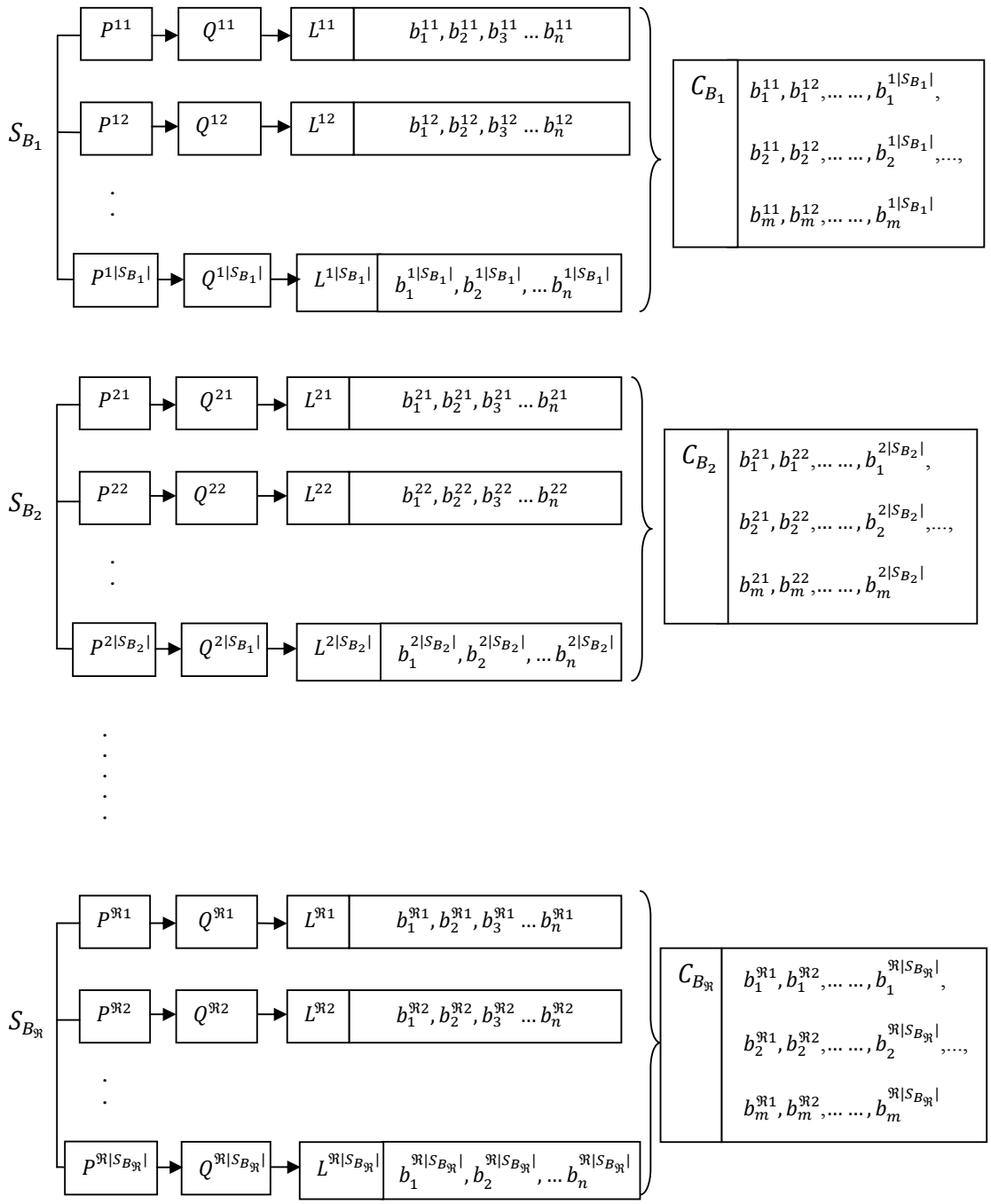


Figure 4.7: The Product Retrieval and Merging for All Neighbours

For each product P^{ij} viewed by the user neighbour B_i, i.e. $\forall P^{ij} \in S_{B_i}$, based on the similarity, a list of ranked products can be generated, $L^{ij} = < b_1^{ij}, b_2^{ij}, \ldots, b_r^{ij} >$ where $sim(b_1^{ij}, Q^{ij}) > sim(b_2^{ij}, Q^{ij}) > \cdots \ldots > sim(b_r^{ij}, Q^{ij})$. Therefore, from the neighbour B_i, $|S_{B_i}|$ lists of products are generated: L^{i1} , L^{i2}, ..., $L^{i|S_{B_i}|}$. 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 retrieved list L^{ij} is based on different query Q^{ij}, and thus the products in all the lists cannot be simply ranked based on this similarity value $sim(b_k^{ij}, Q^{ij})$ to select the products. The Round Robin method is a simple data fusion technique that is adopted to merging and selecting final products from different product lists L^{i1}, L^{i2}, ..., $L^{i|S_{B_i}|}$ retrieved by the multiple queries of each neighbour B_i. 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 $L^{i1} \cup \ldots. \cup L^{i|S_{B_i}|}$, the top N products are chosen from neighbour B_i, denoted as C_{B_i}. By combining the products in C_{B_i} for all neighbours, a set of candidate products Γ can be obtained, i.e. $\Gamma = \bigcup_{i=1}^{\Re} C_{B_i}$. Table 4.2 shows the algorithm of the CFRRobin approach.

The next section will discuss the CFMRRobin approach that ranks the retrieved product lists before selecting the candidate products of each neighbour user.

Table 4.2: The Algorithm of the CFRRobin Approach

Algorithm 4.2

Input: A set of user's neighbours $\{B_1, B_2, \ldots., B_{\Re}\}$; neighbour's session $S_{B_i} = \{P^{i1}, P^{i2}, \ldots, P^{i|S_{B_i}|}\}$ where $P^{ij} =< a_1^{ij}, a_2^{ij}, \ldots, a_n^{ij} >$; a set of products in the database $DB = \{b_1, b_2, b_3, .., b_{|DB|}\}$ where $b_i =< a_{i1}, a_{i2}, \ldots, a_{in} >$.

Output: A set of candidate products Γ

Method:

Begin

1. For each neighbour B_i
2. For each product P^{ij} in the neighbour's session S_{B_i}
   // generate a query using product attribute values
3. $Q^{ij} := \{a_1^{ij}, a_2^{ij}, \ldots, a_n^{ij}\}$
4. For each product in the database DB, $b_k \in DB$
5. For each attribute value $a_{kl} \in b_k$
6. If $a_{kl} = a_l^{ij}$ for $a_l^{ij} \in Q^{ij}$
7. $sim_A(a_{kl}, a_l^{ij}) := 1$
8. Else
9. $sim_A(a_{kl}, a_l^{ij}) := 0$
10. $sim(b_k, Q^{ij}) := \sum_{l=1}^{n} sim_A(a_{kl}, a_l^{ij})$
11. $L^{ij} :=< b_1^{ij}, b_2^{ij}, \ldots, b_r^{ij} >$ //candidate products for query Q^{ij} where $sim(b_1^{ij}, Q^{ij}) > sim(b_2^{ij}, Q^{ij}) > \ldots > sim(b_r^{ij}, Q^{ij})$
12. Apply the Round Robin method to L^{i1}, L^{i2}, ..., $L^{i|S_{B_i}|}$ and generate top N products C_{B_i} for B_i
13. Return $\Gamma := \bigcup_{i=1}^{\Re} C_{B_i}$

End

4.5 INTEGRATING THE CF APPROACH AND SEARCH-BASED APPROACH USING THE MODIFIED ROUND ROBIN FUSION METHOD

In the CFMRRobin approach, the retrieved products lists of each neighbour are sorted based on the popularity of the products retrieved for each list among all the products retrieved by all the queries of the neighbour user. The list that contains products that are more frequently retrieved by all the queries are ranked higher than the list with products that are less frequently retrieved by all the queries. Figure 4.8 illustrates a general framework of the CFMRRobin approach and Figure 4.9 shows the product retrieval and merging of the CFMRRobin approach for each query of the neighbour user.

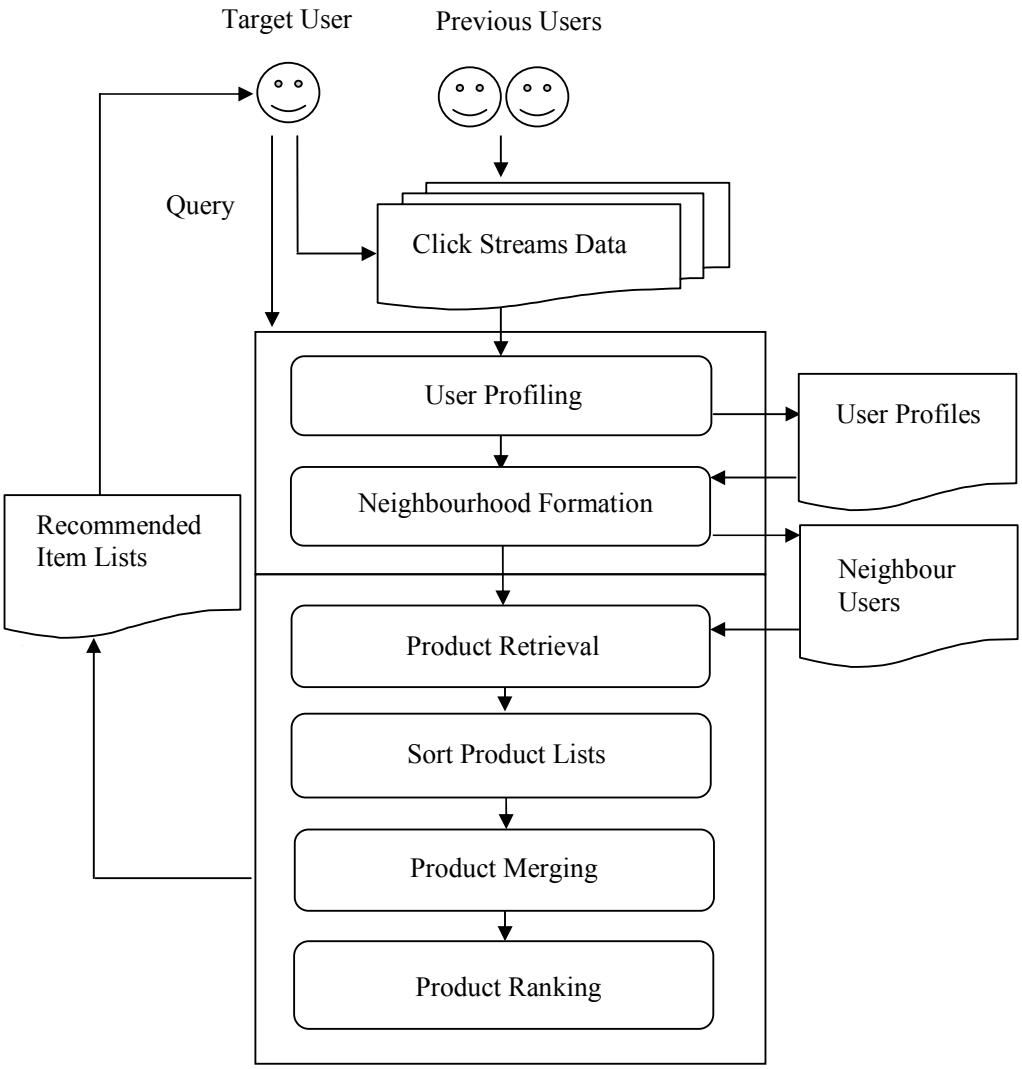


Figure 4.8: A General Framework of the CFMRRobin Approach

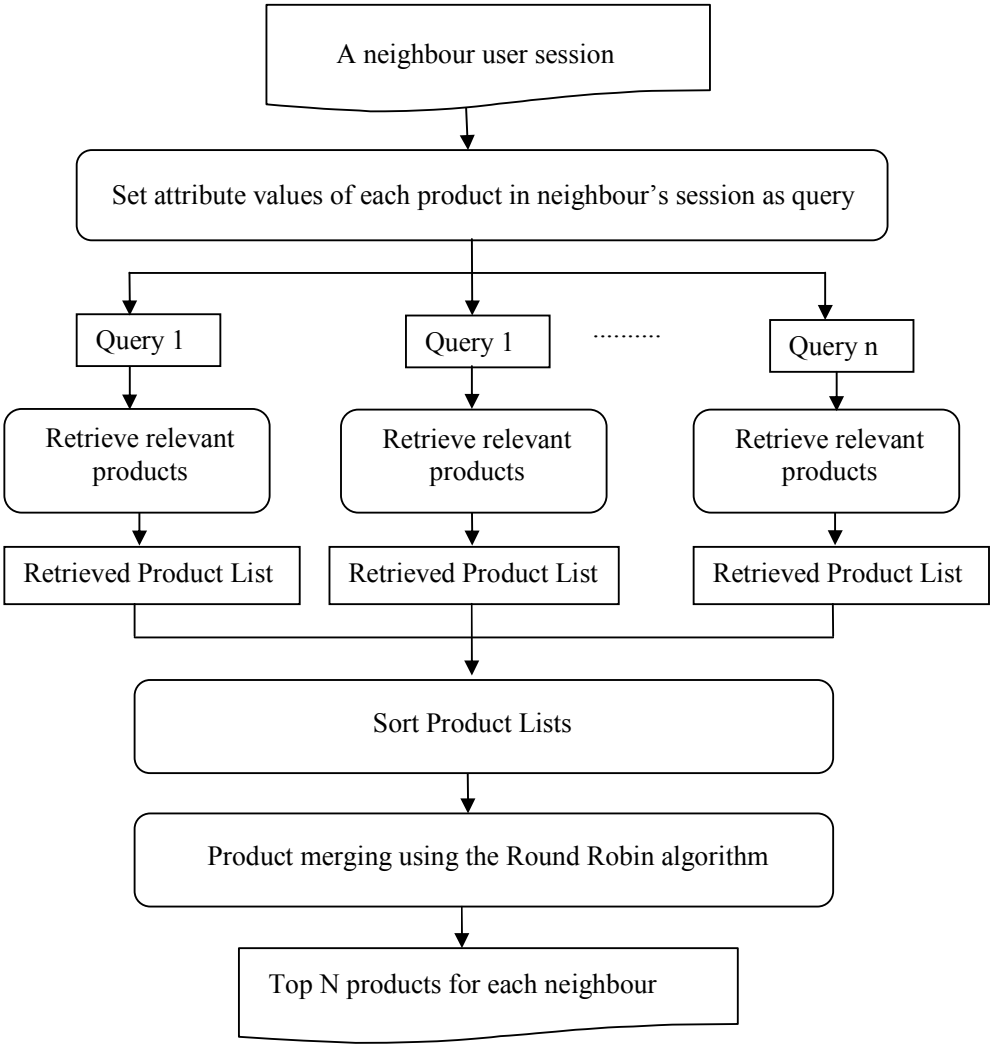


Figure 4.9: The Product Retrieval and Merging for Each Neighbour User of the CFMRRobin

For each retrieved product list of the neighbour user B_i that is: L^{i1}, L^{i2}, ..., $L^{i|S_{B_i}|}$ where $L^{ij} =< b_1^{ij}, b_2^{ij}, \ldots, b_r^{ij} >$, the list weight $lw_{L^{ij}}$ is calculated before selecting the products for each neighbour user by using the Round Robin method. To calculate the list weight $lw_{L^{ij}}$, firstly, the total number of voting $tv_{b_k^{ij}}$ received by each product b_k^{ij} in the list L^{ij} is calculated as the frequency of product b_k^{ij} exists in all the retrieved lists, i.e. $L^{i1} \cup \ldots \cup L^{i|S_{B_i}|}$.

$$tv_{b_k^{ij}} = freq\left(b_k^{ij}\right)$$

Then, the list weight $lw_{L^{ij}}$ is calculated as the total number of voting $tv_{b_k^{ij}}$ received by all products b_k^{ij} in the list L^{ij} as follows:

$$lw_{L^{ij}} = \sum_{k=1}^{n} tv_{b_k^{ij}}$$

The product lists are ranked based on the list weight $lw_{L^{ij}}$. Finally, the top *N* products are chosen for neighbour B_i, denoted as C_{B_i} from the ranked product lists $L^{i1} \cup \ldots \cup L^{i|S_{B_i}|}$ where $lw_{L^{i1}} > lw_{L^{i2}} > \cdots > lw_{L^{i|S_{B_i}|}}$ by using the Round Robin algorithm as discussed in section 4.4. A set of candidate products Γ can be obtained by combining the products in C_{B_i} for all neighbours, i.e. $\Gamma = \bigcup_{i=1}^{\Re} C_{B_i}$. Table 4.3 shows the algorithm of the CFMRRobin.

Table 4.3: The Algorithm of the CFMRRobin Approach

Algorithm 4.3

Input: A set of user's neighbours $\{B_1, B_2, \ldots, B_{\Re}\}$; neighbour's session $S_{B_i} = \{P^{i1}, P^{i2}, \ldots, P^{i|S_{B_i}|}\}$ where $P^{ij} =< a_1^{ij}, a_2^{ij}, \ldots, a_n^{ij} >$; a set of products in the database $DB = \{b_1, b_2, b_3, .., b_{|DB|}\}$ where $b_i =< a_{i1}, a_{i2}, \ldots, a_{in} >$.

Output: A set of candidate products Γ

Method:

Begin

1. For each neighbour B_i
2. For each product P^{ij} in the neighbour's session S_{B_i}
   // generate a query using product attribute values
3. $Q^{ij} := \{a_1^{ij}, a_2^{ij}, \ldots, a_n^{ij}\}$
4. For each product in the database DB, $b_k \in DB$
5. For each attribute value $a_{kl} \in b_k$
6. If $a_{kl} = a_l^{ij}$ for $a_l^{ij} \in Q^{ij}$
7. $sim_A(a_{kl}\, a_l^{ij}) := 1$
8. Else
9. $sim_A(a_{kl}, a_l^{ij})$:=0
10. $sim(b_k,\ Q^{ij}) := \sum_{l=1}^{n} sim_A(a_{kl}, a_l^{ij})$
11. $L^{ij} :=< b_1^{ij}, b_2^{ij}, \ldots, b_r^{ij} >$, where $sim(b_1^{ij},\ Q^{ij}) > sim(b_2^{ij},\ Q^{ij}) > \ldots > sim(b_r^{ij},\ Q^{ij})\}$

// Calculate the total voting for each product

12. For each product list L^{ij}
13. For each product $b_k^{ij} \in L^{ij}$
14. $tv_{b_k^{ij}} = freq(b_k^{ij})$

// Calculate the list weight

15. For each product list L^{ij}
16. For each product $b_k^{ij} \in L^{ij}$
17. $lw_{L^{ij}} = \sum_{k=1}^{n} tv_{b_k^{ij}}$
18. Sort product list L^{ij} based on $lw_{L^{ij}}$
19. Apply the Round Robin method to the sorted product lists $L^{i1}, L^{i2}, \ldots, L^{i|S_{B_i}|}$ and generate top N products C_{B_i} for B_i
20. Return $\Gamma := \bigcup_{i=1}^{\Re} C_{B_i}$

End

4.6 PRODUCT RANKING AND SELECTION

For the CFAgQuery, CFRRobin and CMFRRobin approaches, the products viewed by the user are used to derive new queries based on the products that the neighbour users have liked, before retrieving products for the target user. For the CFAgQuery approach, by doing a search of the product database, products that match the aggregated query AQ^u are retrieved as candidate products Γ for the target user. For the CFRRobin and CFMRRobin approaches, a set of candidate products Γ can be obtained by combining the products in C_{B_i} for all neighbours, i.e. $\Gamma = \bigcup_{i=1}^{\Re} C_{B_i}$. 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 $tu^u =< ta_{11}^u, \dots, ta_{1m_1}^u, ta_{21}^u, \dots, ta_{2m_2}^u, \dots, ta_{nm_n}^u >$ which is generated from the target user's online click stream data. By choosing the attribute value with the highest preference for each attribute, the target user's preferred attribute values $Q^u = \{A_1 = a_1^u,\ A_2 = a_2^u, \dots, A_n = a_n^u\}$ can be generated, where $a_k^u = max_{j=1}^{m_k}\ (ta_{kj}^u)$. Let Γ be the set of candidate products generated by the CFAgQuery , CFRRobin or CFMRRobin, $b_k \in \Gamma$ 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 (4.2). Finally, the top N products are selected as the final products $\mathcal{F} = \{b_1, b_2, \dots, b_N\}$ to be recommended from the ranked products in the candidate list. Table 4.4 shows the algorithm for the products ranking method.

Table 4.4: The Algorithm for Ranking Candidate Products

Algorithm 4.4

Input: The target user's profile
$tp^u =< ta^u_{11}, \dots, ta^u_{1m_1}, ta^u_{21}, \dots, ta^u_{2m_2}, \dots, ta^u_{nm_n} >$,
A set of candidate products $\Gamma = \{b_1, b_2, b_3, .., b_N\}$

Output: A set of final products $\mathcal{F} = \{P_1, P_2, \dots, P_{\Re}\}$

Method:

Begin

1. Generate the target user's preferred attribute values $Q^u = \{a^u_1, a^u_2, \dots, a^u_n\}$
   $a^u_k := max^{m_k}_{j=1}(ta^u_{kj})$
2. For each candidate product $b_k \in \Gamma$
3. For each attribute value $a_{kl} \in b_k$
4. If $a_{kl} = a^u_l$
5. $sim_A(a_{kl}, a^u_l) := 1$
6. Else
7. $sim_A(a_{kl}, a^u_l) := 0$
8. $sim(b_k, Q^u) := \sum^n_{l=1} sim_A(a_l, a^u_l)$
9. Rank the products based on the $sim(b_k, Q^u)$
10. Select top $\Re$ products from the ranked list as final products,
    $\mathcal{F} = \{P_1, P_2, \dots, P_{\Re}\}$

End

4.7 CHAPTER SUMMARY

In this chapter, three recommendation approaches are proposed to integrate the CF and search-based approaches by utilizing user profiles generated from user click stream data. The click stream data is generated from the user's product clicks when the user browses for products to buy and it contains valuable information that can be used to predict the user's interests or preferences for product attribute values. A user profile is generated from the user's click stream data to represent the user's preferences for each product attribute value based on the products that have been viewed by the user. The generated user profile using the proposed user profiling

method is utilised by the CFAgQuery, CFRRobin and CFMRRobin approaches to find the target user's neighbours.

In the first approach, namely the Collaborative Filtering-based Aggregated Query (CFAgQuery), a new query for the target user is created by aggregating the preferences of the user's neighbours. Instead of using the user's initial query which lacks information about the actual user preferences, a new query generated by the proposed method represents the user preferences more precisely and may retrieve and recommend products that best meet the user's requirements. In the second and third approaches namely the Collaborative Filtering-based using Round Robin (CFRRobin) and the Collaborative Filtering-based using Modified Round Robin (CFMRRobin), each product viewed by the user's neighbour is used as a query to retrieve products that are similar to the neighbour's products instead of using the initial query to retrieve products for the target user. By using the neighbours' products to search for other similar products, recommendations can be based on products that have been liked by other users with similar preferences. Some of these products may match the target user's preferences but may be overlooked by the user or not be specified in the initial query. Therefore, by using the neighbours' products as queries, a recommendation system may retrieve more products that satisfy the user's preferences than by directly recommends the neighbours' products. The evaluation results of the CFAgQuery, CFRRobin and CFMRRobin approaches will be given in Chapter 5.

Chapter 5: Experiment and Evaluation

This chapter focuses on the evaluation of the proposed recommendation methods, which include the query expansion method and the hybrid recommendation methods that integrate collaborative filtering and search-based approaches. Firstly, the experiment design and the evaluation methods will be given. Then, the results of experiments will be discussed and illustrated.

5.1 EXPERIMENT DESIGN

The experiments were conducted to see how the proposed recommendation approaches perform compared to the baseline approaches. The experiments were conducted in terms of the following hypothesis:

- **Hypothesis 1**: The user profiles generated based on user click stream data can improve the recommendation accuracy.
- **Hypothesis 2**: The integration of the collaborative filtering approach and search-based approach can generate more accurate recommendations compared to only collaborative filtering or search-based approach.
- **Hypothesis 3**: The query expansion approach using the associations between product attributes generated based on user reviews data and the proposed user profiles can improve the recommendation accuracy.

5.2 EVALUATION METHODS

5.2.1 Dataset

The proposed approach is independent from the selected domain or for infrequently purchased products. The approach involves product features or attributes as main components of the rules or profiles generated to present the user preferences. Thus it is suitable for recommending any products which have features/attributes no matter whether for frequently or infrequently purchased products as long as they have attributes. So, generally the proposed technique is applicable to any products. However, this thesis emphasizes the proposed approach for recommending infrequently purchased products because it does not rely on ratings, and thus can be used to recommend such products.

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 5,504 user reviews, 17,690 cars and 20,868 user navigation sessions. User review data contains comments provided by users for cars previously owned by them. 'Cars' data contains information about the cars available in the company's database. 'User navigation sessions' 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.

5.2.2 Evaluation Metrics

An evaluation of the proposed recommender system approaches is made to measure the levels of their performances. The evaluation involves some sort of scales or metrics to assess which of the system approaches performs better. In this thesis, 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 5.1:

  $$Precision = \frac{NM}{NR} \qquad (5.1)$$

  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 5.2:

  $$Recall = \frac{NM}{NT} \qquad (5.2)$$

  To evaluate the proposed models for online car search, *NM* is the number of retrieved cars that match the testing cars, *NR* is the number of retrieved cars, and *NT* is the number of testing cars in the testing session. The testing cars are the cars that have been viewed by a user in each session in the click streams data. These cars have some attributes that of interest to the user,

and thus the cars are considered relevant to the user's preferences. The precision and recall are calculated for each session or user and the average recall and precision for all sessions (i.e. all users) are calculated for each search model.

- **F1 Measure**

The F1 Measure was first introduced by Van Rijsbergen in 1979. The 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:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (5.3)$$

5.2.3 Experiment Environment and Framework

In order to evaluate the effectiveness of the proposed user profiling approach and the recommendation approaches, this thesis implements the proposed user profiling, recommendation approaches, and the baseline models. php was used as the programming language to implement the system. The experiments were mainly conducted on a Personal Computer equipped with an Intel® Core™ Duo p870 2.53 GHz CPU and 4.00 GB memory running a Window 7 Professional operating system.

The proposed approaches include:

- **OMQE**

This recommendation approach generates a new query by expanding the initial user's query with the user's preferred attribute values based on association rules generated from the user review data.

- **TUProfile**

This approach directly uses target user profiles as queries to retrieve relevant products. The maximum value of each attribute shows the user is more interested in this attribute value. Thus the maximum values of all product attributes are used as the query's values. The purpose of implementing this model is to examine the performance of using the proposed user profiling approach only to generate recommendations without using the aggregated query or the Round Robin based merging method.

- **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 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 their similarities with the target user profile.

- **CFMRRobin**

This recommendation approach also integrates the collaborative filtering and search-based approaches by using each product of the neighbour users as a query. This approach is different from the CFRRobin because it ranks the retrieved lists of each neighbour before using the Round Robin data fusion to merge products retrieved by multiple queries. It also ranks and

selects the final products based on their similarities with the target user's profile.

The baseline models include:

- **BS**

  Currently, many e-commerce sites for selling infrequently purchased products only provide standard search engines that retrieve products based on the initial query given by the user. The BS is the standard search-based approach that retrieves products that match the initial query.

- **CFOriginal**

  The original collaborative filtering approach is popularly used for recommending frequently purchased products. This approach finds users with similar interests to the target user and directly recommends products that are preferred by users similar to the target user.

This thesis compared the recommendation results produced by the proposed approaches with the baseline models. Figure 5.1 illustrates the framework of the evaluation experiments.

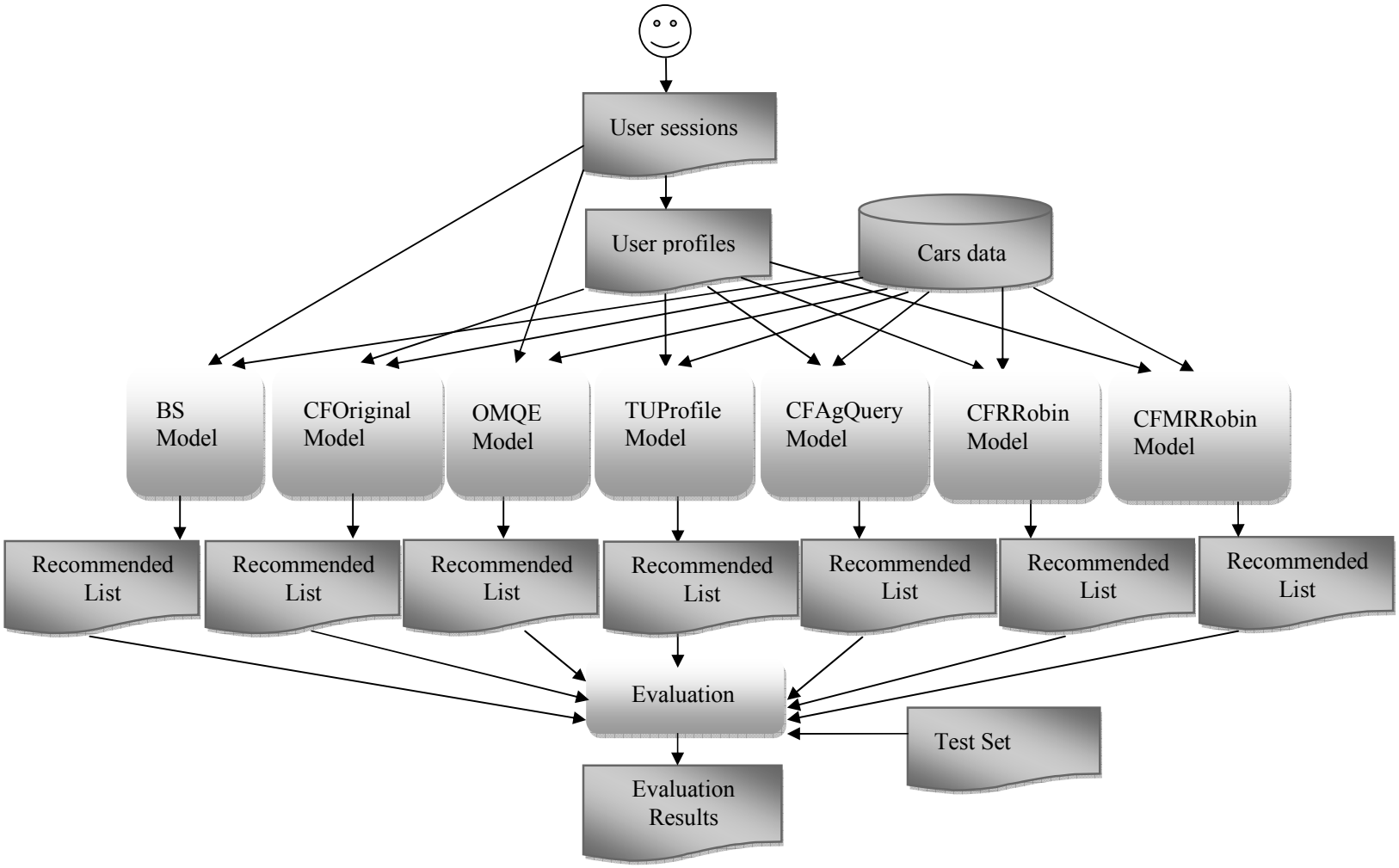


Figure 5.1: The Experiment Framework for the Evaluation

5.2.4 Experiment Setup

The user session dataset was partitioned into 5 sub datasets. 20% of user sessions from each of the sub dataset was used as a testing dataset and the remaining sessions were used as a training dataset. 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 5.2.

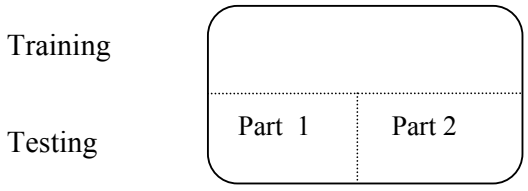


Figure 5.2: The Division of Session Dataset for the Experiment

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 the recommendations generated by the recommender models contain cars that match the cars in the Testing Part 2. For the BS and the OMQE models, the last car of each session in Testing Part 1 was used as the query to retrieve cars for these models. The CFOriginal, TUProfile, CFAgQuery, CFRRobin and CFMRRobin models generate the target user's profile based on the data in Testing Part 1 by using the user profiling method discussed in Chapter 4. Sessions in the Training dataset were considered as previous users. The cars in the training dataset were considered as the cars that this user has interest in. The training dataset was used to generate previous users' profiles which will be used to find neighbours by using the

neighbourhood formation method discussed in Chapter 4. For each experiment, there are 5 runs and thus, the average result for the 5 runs is calculated. The experiments are repeated for all the sub datasets by swapping the testing and training data. Finally, the average result for the first and second set of experiments is calculated.

The experiments are conducted to test if the proposed methods, i.e. OMQE, TUProfile, CFAgQuery, CFRRobin and CFMRRobin outperform the baseline models, i.e. BS and CFOriginal. In addition, the experiments also test the impact of using different user profiles created from different numbers of viewed products in the target user's click data for the CFOriginal, TUProfile, CFAgQuery, CFRRobin and CFMRRobin. 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 and the OMQE models, which do not utilize user profiles, only the last car is used as the query to retrieve relevant cars. Figure 5.3 illustrates the generation of different user profiles from the testing sessions. Table 5.1 lists all different runs in the experiments.

In the evaluation, the retrieved car ID is not matched with the car IDs in the testing dataset because for the car domain in which the experiments have been conducted, different car IDs may refer to different cars that have the same attributes. The purpose of product searching is to provide users with the products that meet users' requirements with respect to product attributes or features. The focus of this experiment is to recommend cars that match the attribute values preferred by the user. Thus cars with different Ids, but which have the same attributes, might be recommended.

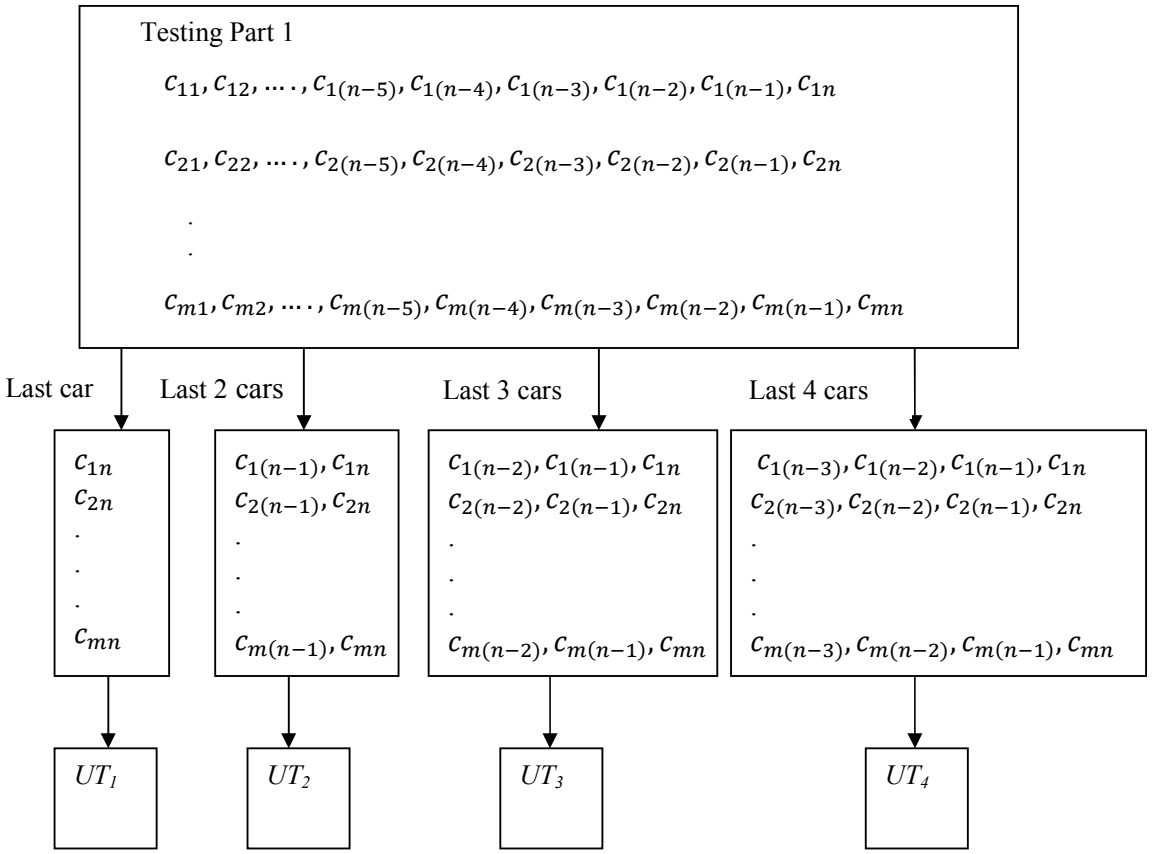


Figure 5.3: The Generation of User Profiles

Table 5.1: Different Runs of the Experiments

Models	**Runs**	**Target User Profiles**
Basic Search(BS)	BS	UT_1
Query Expansion(OMQE)	OMQE	UT_1
Using Target User Profile (TUProfile)	TUProfile1Cars	UT_1
	TUProfile2Cars	UT_2
	TUProfile3Cars	UT_3
	TUProfile4Cars	UT_4
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 (CFRRobin)	CFRRobin1Cars	UT_1
	CFRRobin2Cars	UT_2
	CFRRobin3Cars	UT_3
	CFRRobin4Cars	UT_4
Collaborative Filtering with Modified Round Robin (CFMRRobin)	CFMRRobin1Cars	UT_1
	CFMRRobin2Cars	UT_2
	CFMRRobin3Cars	UT_3
	CFMRRobin4Cars	UT_4

In this experiment, to evaluate all the models, for each session, if at least 80% of the attributes of a retrieved car match the attributes of one of the cars in the same session of the Testing Part 2, the retrieved car is considered as matching the test car. Only 80% of matched attributes values are considered in the evaluation because for products like cars, some of the attributes may have different values even though the cars have similar characteristics. For example, automatic cars may have many different standard transmission values such as 3A, 4A, 5A, 6A etc. Thus it is not necessary for all the attribute values of the car to be matched with all the attributes values of a testing car to consider the car relevant with the user requirements. In addition, the focus of this experiment is to recommend cars that match the attribute values preferred by the user. Thus cars with different Ids, but which have the same attributes, might be recommended.

5.3 RESULTS AND DISCUSSIONS

Results of recommendations based on association rules, user profiles and by combining the collaborative filtering and search-based approaches will be examined in this section. The results are compared based on the hypothesis given in section 5.1.

5.3.1 The utilization of user profiles based on user click stream for product recommendations (Hypothesis 1)

The objective of this set of experiments is to verify that the user profiles generated based on user click stream data can improve recommendation accuracy (Hypothesis 1). The experiments involve the comparison of the TUProfile model that utilises user profiles as queries and the BS that uses the original queries to search for products. The discussion of the F1 Measure, precision and recall results will be given

before the tables and figures of the results. Based on Table 5.5 and Figure 5.6, the F1 results for the TUProfile model are higher than the BS model for all the profiles. The TUProfile employs user profiles generated from the user click stream data as the users' queries, whereas the BS model uses the initial query provided by the user. These results prove that the user profiles generated from products that have been clicked by the users in the click stream data represent the users' preferences better than the original query. The user profile represents attribute values preferred by a user based on the products than have been viewed by the user, and thus can be used as a better query to retrieve products that most likely satisfy the user's interests. These results verify that **Hypothesis 1** is valid. These results also show that the TUProfile1Cars > TUProfile2Cars > TUProfile3Cars > TUProfile4Cars. These results indicate that the user profiles generated from more recent products viewed by the users are more accurate to be used as the query compared to the earlier products viewed by the users. The users have more knowledge about the products they want to buy based on the products they have looked at earlier, and thus more recent products viewed by the users can represent user preferences more accurately than the earlier viewed products.

The results in Table 5.3 and Figure 5.4 show that the precision of the TUProfile is better than that of the BS, which proves that more products that satisfy the user's needs can be recommended by using the generated user profile as a query. However, from the results illustrated in Table 5.4 and Figure 5.5, it can be seen that the TUProfile performs lower than the BS in terms of recall. The recall result for each user is calculated based on the number of products in the testing data that match the returned products. An experiment has been conducted to show the average number of returned products that match each product in the testing data and the result

of the experiment is depicted in Table 5.2. The number of products in each session may vary; only the average number of retrieved products that match for the first ten products in the testing data are shown in the table. The results of this experiment indicate that for the TUProfile, more retrieved products match the earlier products in the testing data than the later products in the testing data. The earlier products in the testing data are closer to the products that are used to generate user profiles than the later products and thus, the products may match the user's preferences better than the later products in the testing data. Therefore, the TUProfile retrieves more products that are highly match the user's preferences than the BS even though the TUProfile has lower recall than the BS. This result shows the TUProfile approach can improve the recommendation performance by presenting more products that most likely preferred by the user.

Table 5.2: The Average Number of the Retrieved Products Matching with Products in the Testing Data for the BS and TUProfile

	1st Car	**2nd Car**	**3rd Car**	**4th Car**	**5th Car**	**6th Car**	**7th Car**	**8th Car**	**9th Car**	**10th Car**
BS	5.58	3.91	2.94	2.62	1.89	1.65	1.23	0.75	0.68	0.87
TUProfile	8.43	5.95	4.63	4.07	2.54	2.49	1.80	1.42	1.44	2.12

Table 5.3: Precision Results of the TUProfile for Different Profiles

	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.379	0.383	0.382	0.380	0.380	0.381
TUProfile1Cars	0.598	0.590	0.581	0.572	0.562	0.552
TUProfile2Cars	0.546	0.538	0.530	0.522	0.514	0.507
TUProfile3Cars	0.541	0.534	0.527	0.520	0.513	0.505
TUProfile4Cars	0.526	0.519	0.512	0.505	0.498	0.492

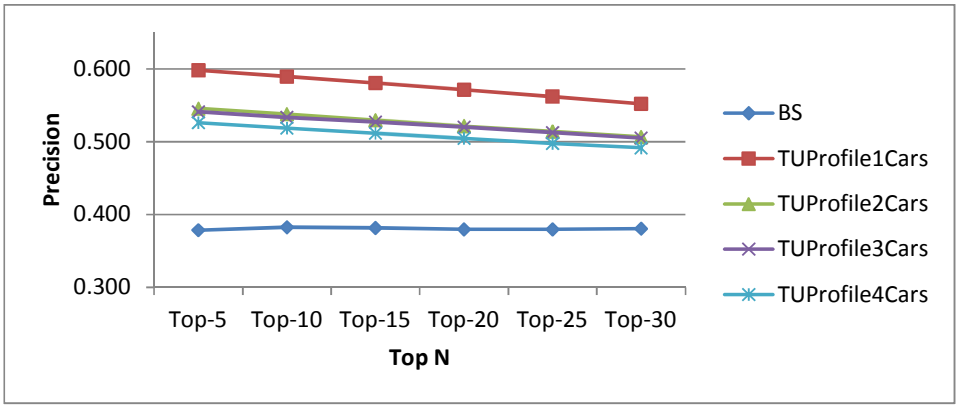


Figure 5.4: Precision Results of the TUProfile for Different Profiles

Table 5.4: Recall Results of the TUProfile for Different Profiles

	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.339	0.378	0.384	0.392	0.395	0.397
TUProfile1Cars	0.363	0.371	0.377	0.381	0.387	0.392
TUProfile2Cars	0.357	0.371	0.382	0.388	0.396	0.403
TUProfile3Cars	0.347	0.362	0.372	0.379	0.386	0.394
TUProfile4Cars	0.341	0.356	0.367	0.374	0.383	0.391

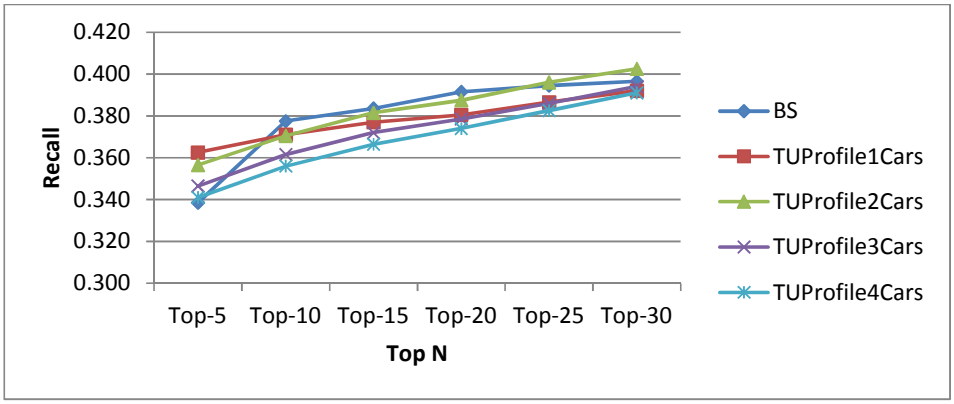


Figure 5.5: Recall Results of the TUProfile for Different Profiles

Table 5.5: F1 Measure Results of the TUProfile for Different Profiles

	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.357	0.380	0.382	0.385	0.387	0.388
TUProfile1Cars	0.451	0.455	0.457	0.457	0.458	0.458
TUProfile2Cars	0.431	0.439	0.443	0.445	0.447	0.449
TUProfile3Cars	0.422	0.431	0.436	0.438	0.440	0.443
TUProfile4Cars	0.414	0.422	0.427	0.430	0.432	0.436

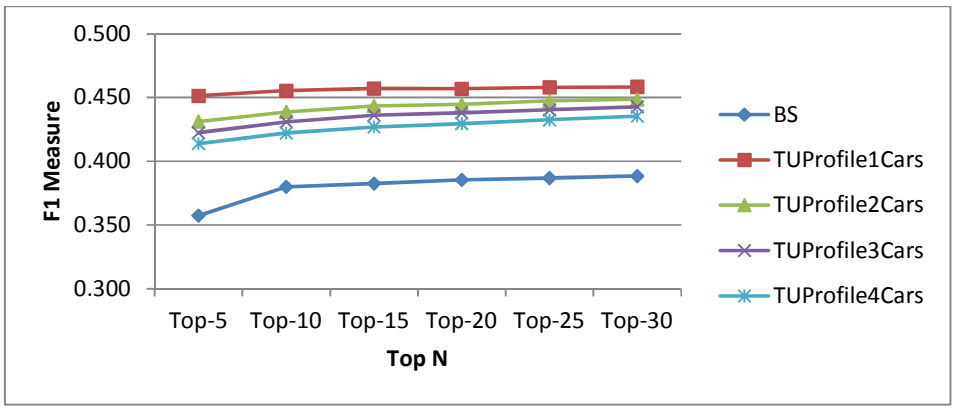


Figure 5.6: F1 Measure Results of the TUProfile for Different Profiles

5.3.2 The integration of the collaborative filtering and search-based approaches for product recommendations (Hypothesis 2)

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 2). The experiments have been conducted to compare the performance of the CFAgQuery, the CFRRobin and the CFMRRobin that integrate the collaborative filtering and search-based approach against the BS and the CFOriginal. The BS employs the standard search-based approach and the CFOriginal employs the standard collaborative filtering approach. Results of the proposed models using different user profiles will be examined in this section. The results are compared based on three categories: i) between each of the proposed models and the baseline

models for the same user profile, ii) among the proposed models for the same user profile and iii) between different profiles for the same proposed model. The results of the experiments are firstly discussed based on the tables and figures that will be given after the discussion.

(i) Comparison between the proposed models and the baseline models for the same user profile

This section compares the results of different proposed models for the same user profile. Firstly, the F1 Measure results will be discussed based on the results given in Table 5.10, Table 5.13, Table 5.16, Table 5.19 and illustrated in Figure 5.9, Figure 5.12, Figure 5.15 and Figure 5.18. The F1 Measure results of the CFRRobin and the CFMRRobin models are better than the baseline models namely the BS and the CFOriginal for all the profiles. The CFRRobin and the CFMRRobin use each product of the neighbour users as a query to search for other relevant products. These results indicate that the performance of the recommendation approaches can be improved by using the neighbours' products as queries rather than using the initial user's query to search for products as applied by the BS approach. The results also show that the performance of the recommendations can also be improved by searching other relevant products that have the same attribute values with the neighbours' products instead of directly recommending the neighbour users' products to the target user as applied by the CFOriginal.

The F1 Measure results of the proposed CFAgQuery are better than the BS for all the profiles. The CFAgQuery generates a new query based on the neighbours' products and the similarity between the neighbour and the target user. Then, the new query is used to retrieve relevant products to

recommend. These results prove that the generated aggregated query based on the products viewed by the neighbour users is more accurate compared to the user's initial query. In addition, the F1 Measure results of the CFAgQuery are better than the CFOriginal for user profiles generated from the last car, the last 2 cars and the last 3 cars. These results demonstrate that generating a new query based on the neighbours' products to search for relevant products instead of directly recommending the neighbours' products can improve the recommendation performance. However, the F1 Measure of the CFAgQuery is lower than the CFOriginal for user profiles generated from the last 4 cars. This is because the CFOriginal performs better in terms of recall for these profiles compared to the CFAgQuery as shown in Table 5.18 and Figure 5.17. The user profiles generated from more products are more diverse, and thus there are more chances to select neighbours with diverse products. The CFOriginal directly recommends the neighbours' products and thus, more diverse products can be recommended. In contrast, the CFAgQuery generates a new query based on the neighbours' products, and this query has more focused attribute values than in the neighbours' products and thus may retrieve products with focused attribute values. Therefore, the recall for the CFAgQuery is lower than the CFOriginal especially for profiles generated from more products. The CFAgQuery performs better than the CFOriginal by considering only few most recently viewed products.

The previous discussions of the F1 results verify **Hypothesis 2** that the integration of the collaborative filtering approach and search-based approach can generate more accurate recommendations compared to only collaborative filtering or search-based approach.

Based on the results showed in Table 5.8, Table 5.11, Table 5.14, Table 5.17 and illustrated in Figure 5.7, Figure 5.10, Figure 5.13, and Figure 5.16, the precision results of the CFRRobin, the CFMRRobin and the CFAgQuery models are better than the baseline models namely the BS and the CFOriginal models. This result indicates that by using the neighbours' products to search for other relevant products as implemented by the CFRRobin and the CFMRRobin, more products that will satisfy the user's needs can be retrieved. In addition, a new query based on the neighbours' products as employed by the CFAgQuery can be used to retrieve more relevant products to the user.

The recall results discussions are based on Table 5.9, Table 5.12, Table 5.15, Table 5.18, Figure 5.8, Figure 5.11, Figure 5.14, and Figure 5.17. The results show that the CFOriginal performs the best for profiles generated from the last 2 cars, 3 cars and 4 cars. However, for profiles generated from the last car, the CFOriginal performs better than the CFRRobin, the CFMRRobin and the CFAgQuery for the top 5, top 10, and top 15 retrieved products but lower than the BS model. The BS model performs the best in terms of the recall for user profiles generated from the last car. The recall results of the BS are also better than the CFAgQuery for profiles generated from the last 2 cars, the last 3 cars and the last 4 cars. The BS also performs better than the CFRRobin and the CFMRRobin for the top 5, top 10 and top 15 retrieved products for these profiles. These results show that the recall results for the proposed models, namely the CFRRobin, CFMRRobin and CFAgQuery, are lower than the baseline models, which are the BS and the CFOriginal. The recall for each user is calculated based on the number of

products in the testing data that match the returned products. An experiment has been conducted to show the average number of returned products that match each product in the testing data and the result of this experiment is depicted in Table 5.6. As mentioned in section 5.3.1, the number of products in each session may vary, only the results of the first ten products in the testing data are shown in the table. The result of this experiment indicates that more returned products match the earlier products compared to the later products in the testing data for the CFRRobin, the CFMRRobin and the CFAgQuery approaches. On the contrary, only few retrieved products match the earlier products in the testing data for the CFOriginal and the BS approaches. The earlier products in the testing data are closer to the products used to generate user profiles, and thus might satisfy the users' preferences better than the later products in the testing data. Therefore, although products retrieved by the CFRRobin, the CFMRRobin and the CFAgQuery do not match some of the products in the testing data which result in low recalls, many of the retrieved products satisfy the products that are highly preferred by the users in the testing data, which also improve the recommendation accuracy.

Table 5.6: The Average Number of Retrieved Products Matching with the Products in Testing data for the BS, CFOriginal, CFAgQuery, CFRRobin and CFMRRobin

	1st Car	**2nd Car**	**3rd Car**	**4th Car**	**5th Car**	**6th Car**	**7th Car**	**8th Car**	**9th Car**	**10th Car**
BS	5.58	3.91	2.94	2.62	1.89	1.65	1.23	0.75	0.68	0.87
CFOriginal	1.48	1.14	0.93	0.84	0.60	0.65	0.55	0.44	0.42	0.52
CFAgQuery	8.53	6.05	4.62	4.11	2.61	2.71	1.98	1.64	1.70	2.19
CFRRobin	8.66	5.92	4.61	3.93	2.55	2.45	1.86	1.41	1.53	1.90
CFMRRobin	8.69	5.96	4.64	3.95	2.58	2.47	1.87	1.43	1.54	1.93

In addition, many users in the dataset may have diverse attribute value interests, meaning the users may look at products with different attribute values. An experiment has been conducted to identify users with diverse attribute interests and users with focused attribute interests. The similarity value for each pair of cars viewed in each session was calculated based on the cars' attribute values by using the cosine similarity function as follows:

$$sim(A, B) = \frac{\sum_{j=1}^{n} a_j b_j}{\sqrt{\sum_{j=1}^{n} a_j^2}\sqrt{\sum_{j=1}^{n} b_j^2}},$$

where A and B are two of the cars that have been viewed by a user in a session, and a_j and b_j is the value for each attribute of A and B, respectively. Then, the average attribute similarity values of all pairs of cars viewed in each session were calculated. The result of this experiment reveals that there are more users with diverse products compared to users with focused products in the testing data as shown in Table 5.7.

Table 5.7: Number of Users with Focused and Diverse Attribute Values

Focused attribute values (similarity values more or equal to 0.5)	**Diverse attribute values (similarity values less than 0.5)**
1594 user sessions	1970 user sessions

If the products in the testing data are diverse where they have different attributes from one another, 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 of most interest to the neighbour users and thus may return products that are more focused in

terms of attribute values and so might not match with other products with different attribute values in the testing data. The CFRRobin and the CFMRRobin utilise 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 and the CFMRRobin approaches to recommend more products with focused attribute values compared to the BS and the CFOriginal. On the contrary, the BS only uses a small number of attributes from the initial queries to retrieve products and thus, products that have some attribute values matching the query's attribute values can be retrieved. As a result, the products retrieved by the BS are more diverse, as the products may have different attribute values. In addition, the CFOriginal directly recommends the neighbours' products and thus more diverse products can be recommended even though the products may not highly satisfy the user's preferences. As a result, the recall results for the proposed models, which are the CFAgQuery, the CFRRobin and the CFMRRobin are lower than the BS and the CFOriginal. However, the CFRRobin, the CFMRRobin and the CFAgQuery can also recommend unexpected or different products to the users as long as the products have the attribute values that are highly preferred by the users. Moreover, the CFRRobin and the CFMRRobin perform better than the CFAgQuery because they recommend products that are similar to the products interested in 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.

(ii) Comparison between the CFRRobin, CFMRRobin and CFAgQuery models

This section compares the results of different proposed models, i.e. CFRRobin, CFMRRobin and CFAgQuery. Based on the F1 Measure results given in Table 5.10, Table 5.13, Table 5.16, Table 5.19 and illustrated in Figure 5.9, Figure 5.12, Figure 5.15 and Figure 5.18, the CFMRRobin performs the best, followed by the CFRRobin and CFAgQuery. The precision results of the CFMRRobin are also better than of the CFRRobin, whereas the precision results of the CFAgQuery is the lowest as shown in Table 5.8, Table 5.11, Table 5.14, Table 5.17 and illustrated in Figure 5.7, Figure 5.10, Figure 5.13, and Figure 5.16. The recall results also shows that the CFMRRobin performs the best, followed by the CFRRobin, while the CFAgQuery performs the worst based on Table 5.9, Table 5.12, Table 5.15, Table 5.18, Figure 5.8, Figure 5.11, Figure 5.14, and Figure 5.17. Based on the F1, precision and recall results, the CFMRRobin performs better than the CFRRobin, which shows that by sorting the products retrieved by all queries before selecting the final products using Round Robin approach helps in selecting more products that better match the user's requirements. Moreover, the CFMRRobin and the CFRRobin perform better than the CFAgQuery, which shows that the query generated based on each product viewed by the neighbour users as implemented by the CFMRRobin and CFRRobin approaches may represent user requirements better than the aggregated query generated based on the neighbour users' profiles as implemented by CFAgQuery. Therefore, the CFMRRobin and CFRRobin retrieve more products that better match the user requirements.

Table 5.8: Precision Results of Different Models for User Profile UT_1

Precision	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.379	0.383	0.382	0.380	0.380	0.381
CFOriginal1Cars	0.386	0.381	0.381	0.380	0.380	0.380
CFRRobin1Cars	0.597	0.589	0.581	0.572	0.562	0.551
CFMRRobin1Cars	0.597	0.590	0.582	0.574	0.565	0.554
CFAgQuery1Cars	0.596	0.589	0.579	0.570	0.560	0.550

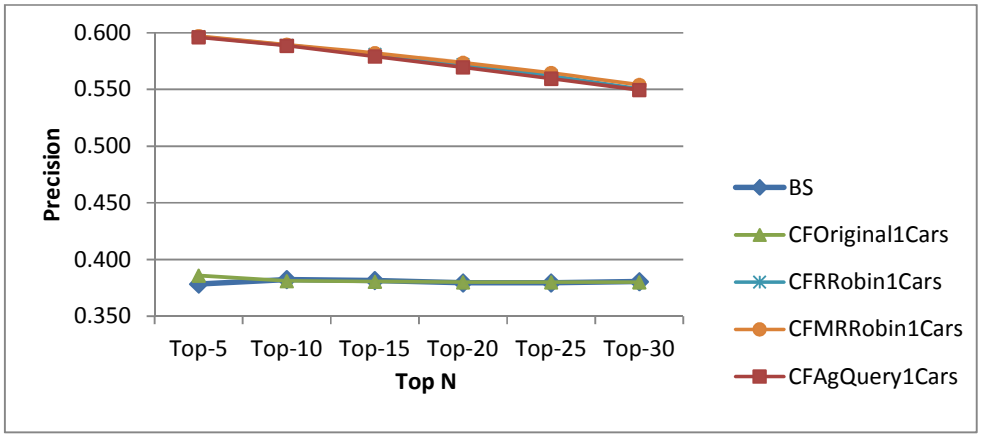


Figure 5.7: Precision Results of Different Models for User Profile UT_1

Table 5.9: Recall Results of Different Models for User Profile UT_1

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.354	0.382	0.389	0.393	0.396	0.398
CFOriginal1Cars	0.369	0.375	0.376	0.376	0.376	0.376
CFRRobin1Cars	0.360	0.368	0.374	0.379	0.385	0.390
CFMRRobin1Cars	0.361	0.369	0.375	0.380	0.385	0.389
CFAgQuery1Cars	0.361	0.369	0.376	0.382	0.387	0.393

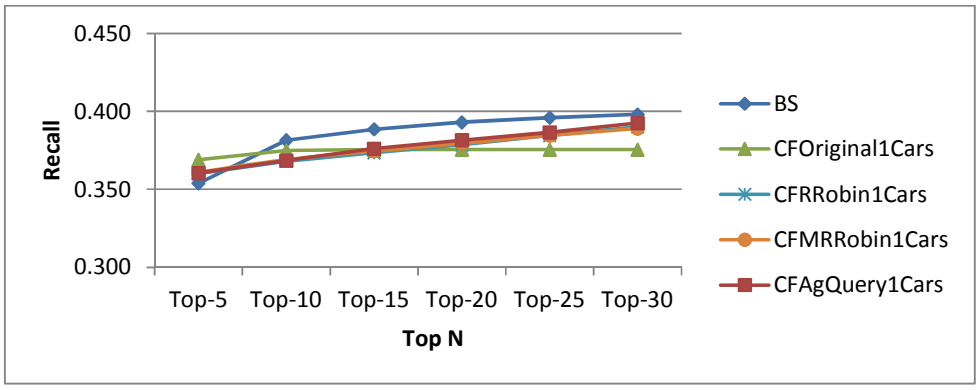


Figure 5.8: Recall Results of Different Models for User Profile UT_1

Table 5.10: F1 Measure Results of Different Models for User Profile UT_1

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.366	0.382	0.385	0.386	0.388	0.389
CFOriginal1Cars	0.377	0.378	0.378	0.378	0.378	0.378
CFRRobin1Cars	0.449	0.453	0.455	0.456	0.457	0.457
CFMRRobin1Cars	0.450	0.454	0.456	0.457	0.457	0.457
CFAgQuery1Cars	0.449	0.453	0.456	0.457	0.457	0.458

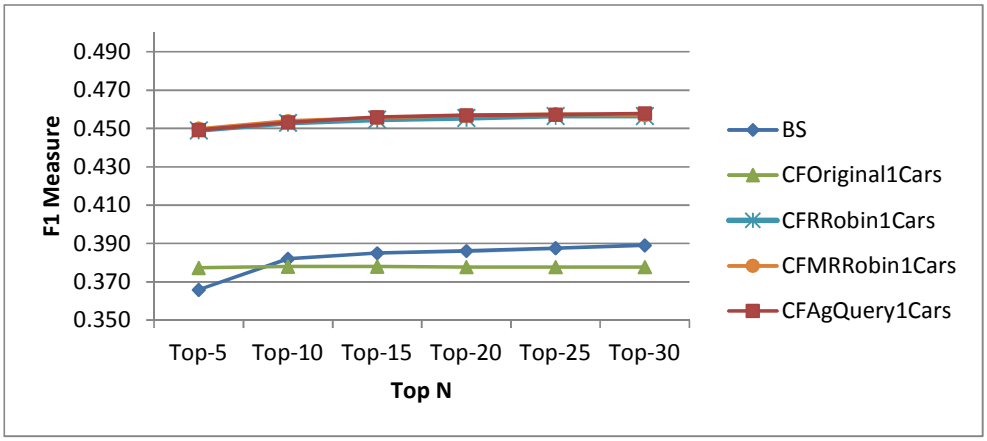


Figure 5.9: F1 Measure Results of Different Models for User Profile UT_1

Table 5.11: Precision Results of Different Models for User Profile UT_2

Precision	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.379	0.383	0.382	0.380	0.380	0.381
CFOriginal2Cars	0.427	0.416	0.414	0.413	0.413	0.412
CFRRobin2Cars	0.571	0.563	0.555	0.547	0.538	0.529
CFMRRobin2Cars	0.573	0.565	0.558	0.549	0.540	0.532
CFAgQuery2Cars	0.554	0.547	0.537	0.529	0.521	0.513

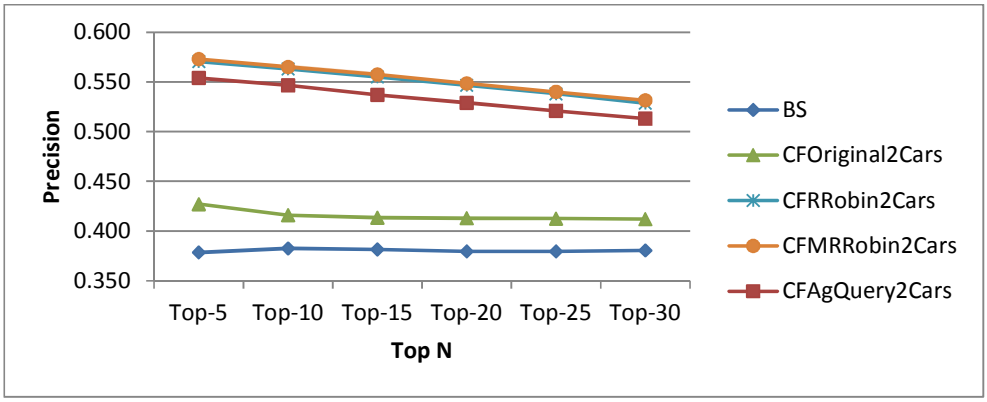


Figure 5.10: Precision Results of Different Models for User Profile UT_2

Table 5.12: Recall Results of Different Models for User Profile UT_2

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.354	0.382	0.389	0.393	0.396	0.398
CFOriginal2Cars	0.431	0.457	0.461	0.462	0.462	0.462
CFRRobin2Cars	0.364	0.378	0.390	0.414	0.425	0.435
CFMRRobin2Cars	0.365	0.379	0.391	0.414	0.425	0.435
CFAgQuery2Cars	0.352	0.364	0.375	0.382	0.387	0.394

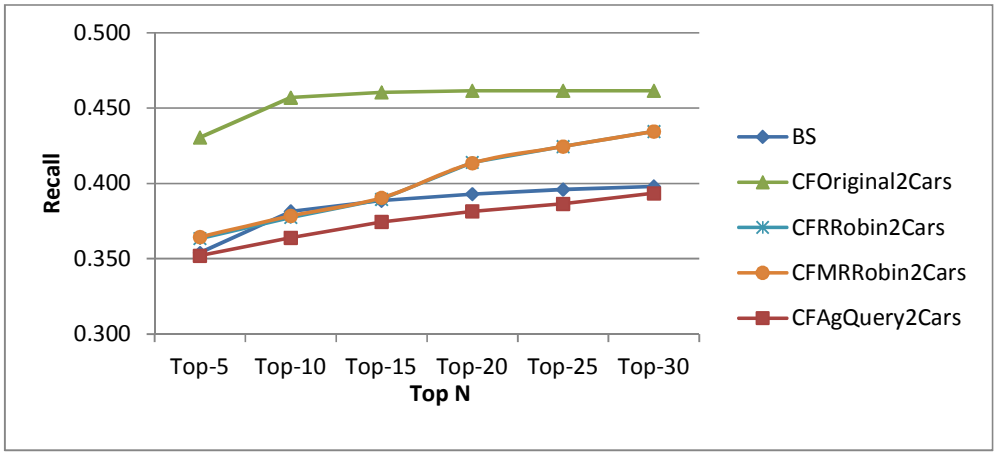


Figure 5.11: Recall Results of Different Models for User Profile UT_2

Table 5.13: F1 Measure Results of Different Models for User Profile UT_2

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.366	0.382	0.385	0.386	0.388	0.389
CFOriginal2Cars	0.429	0.436	0.436	0.436	0.436	0.435
CFRRobin2Cars	0.444	0.452	0.458	0.471	0.475	0.477
CFMRRobin2Cars	0.446	0.453	0.459	0.472	0.475	0.478
CFAgQuery2Cars	0.430	0.437	0.441	0.443	0.444	0.445

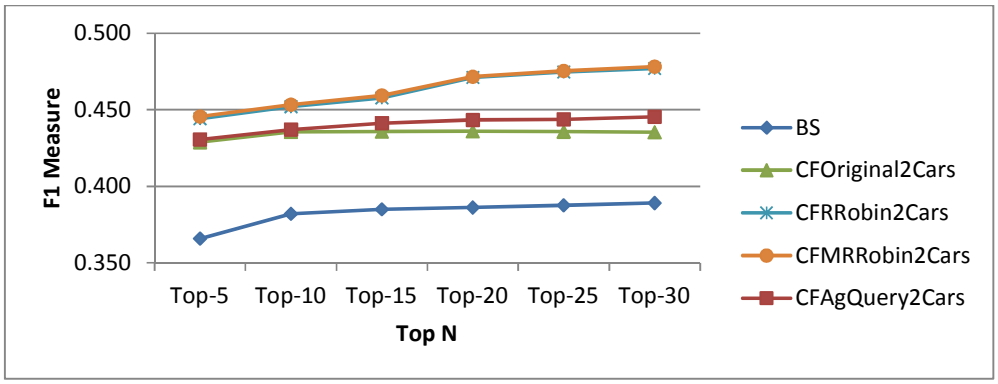


Figure 5.12: F1 Measure Results of Different Models for User Profile UT_2

Table 5.14: Precision Results of Different Models for User Profile UT_3

Precision	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.379	0.383	0.382	0.380	0.380	0.381
CFOriginal3Cars	0.424	0.413	0.409	0.408	0.407	0.407
CFRRobin3Cars	0.561	0.555	0.548	0.539	0.530	0.520
CFMRRobin3Cars	0.562	0.556	0.550	0.541	0.532	0.523
CFAgQuery3Cars	0.548	0.542	0.535	0.526	0.518	0.510

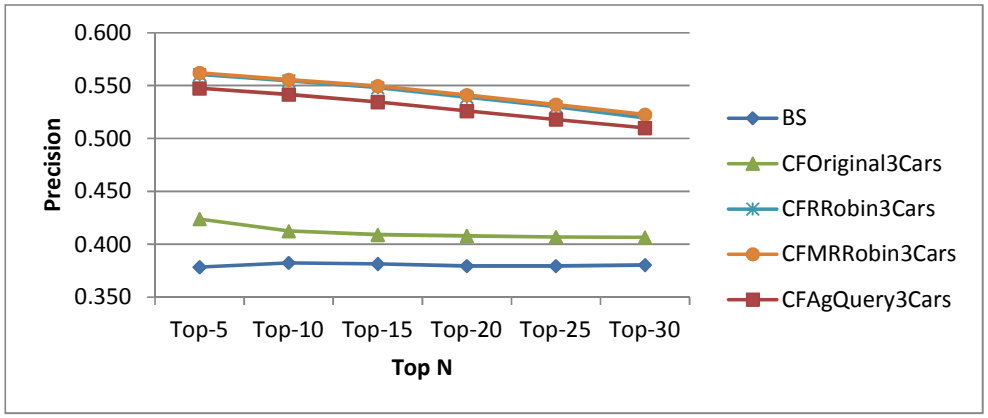


Figure 5.13: Precision Results of Different Models for User Profile UT_3

Table 5.15: Recall Results of Different Models for User Profile UT_3

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.354	0.382	0.389	0.393	0.396	0.398
CFOriginal3Cars	0.419	0.450	0.457	0.461	0.463	0.464
CFRRobin3Cars	0.356	0.369	0.380	0.401	0.413	0.422
CFMRRobin3Cars	0.357	0.370	0.382	0.401	0.412	0.423
CFAgQuery3Cars	0.344	0.357	0.366	0.373	0.379	0.386

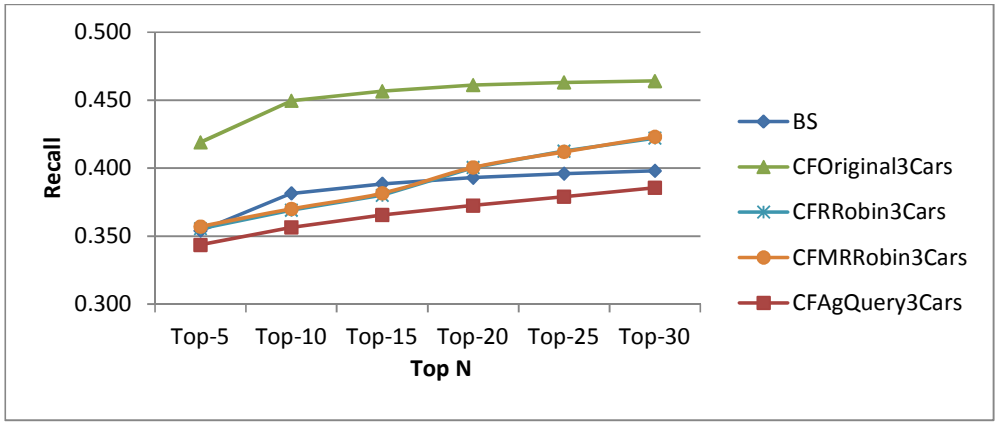


Figure 5.14: Recall Results of Different Models for User Profile UT_3

Table 5.16: F1 Measure Results of Different Models for User Profile UT_3

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.366	0.382	0.385	0.386	0.388	0.389
CFOriginal3Cars	0.421	0.430	0.431	0.433	0.433	0.433
CFRRobin3Cars	0.435	0.443	0.449	0.460	0.464	0.466
CFMRRobin3Cars	0.437	0.444	0.450	0.460	0.464	0.468
CFAgQuery3Cars	0.422	0.430	0.434	0.436	0.438	0.439

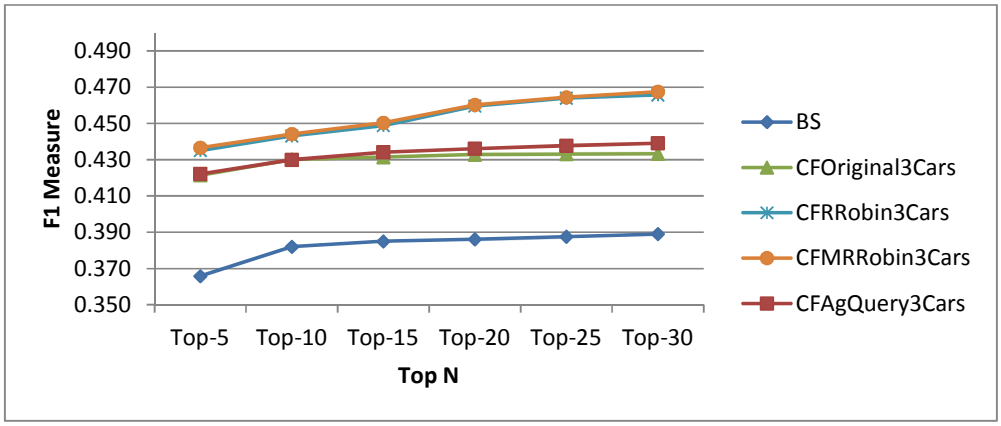


Figure 5.15: F1 Measure Results of Different Models for User Profile UT_3

Table 5.17: Precision Results of Different Models for User Profile UT_4

Precision	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.379	0.383	0.382	0.380	0.380	0.381
CFOriginal4Cars	0.424	0.412	0.408	0.407	0.406	0.405
CFRRobin4Cars	0.546	0.540	0.531	0.523	0.513	0.504
CFMRRobin4Cars	0.547	0.541	0.534	0.525	0.516	0.507
CFAgQuery4Cars	0.537	0.529	0.521	0.513	0.507	0.499

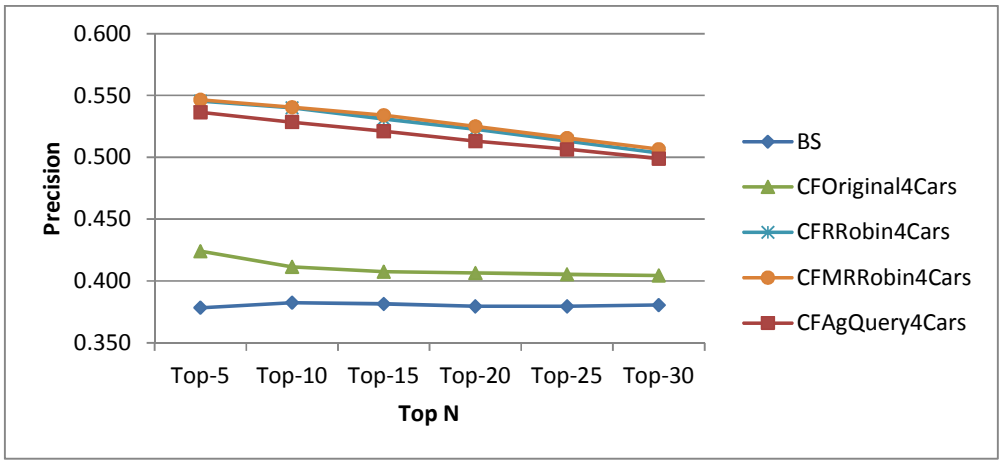


Figure 5.16: Precision Results of Different Models for User Profile UT_4

Table 5.18: Recall Results of Different Models for User Profile UT_4

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.354	0.382	0.389	0.393	0.396	0.398
CFOriginal4Cars	0.422	0.459	0.468	0.474	0.476	0.479
CFRRobin4Cars	0.352	0.367	0.379	0.403	0.418	0.429
CFMRRobin4Cars	0.353	0.368	0.380	0.403	0.417	0.430
CFAgQuery4Cars	0.341	0.354	0.365	0.370	0.377	0.384

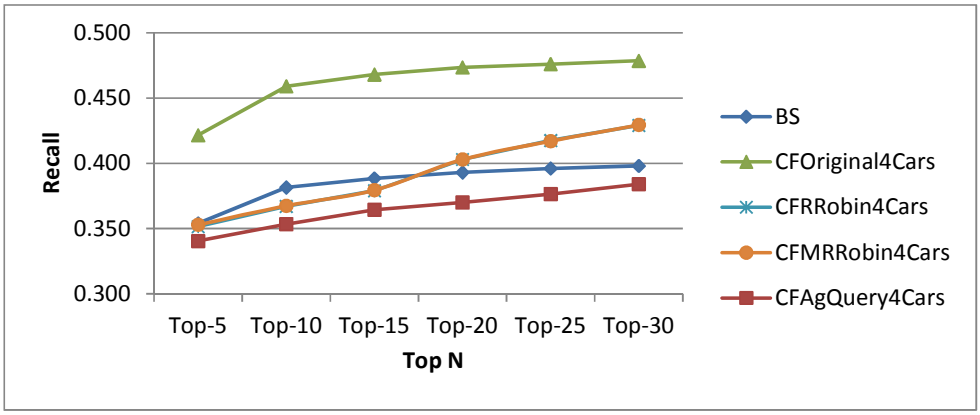


Figure 5.17: Recall Results of Different Models for User Profile UT_4

Table 5.19: F1 Measure Results of Different Models for User Profile UT_4

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.366	0.382	0.385	0.386	0.388	0.389
CFOriginal4Cars	0.423	0.434	0.436	0.437	0.438	0.438
CFRRobin4Cars	0.428	0.437	0.442	0.455	0.460	0.463
CFMRRobin4Cars	0.429	0.438	0.444	0.456	0.461	0.465
CFAgQuery4Cars	0.417	0.424	0.429	0.430	0.432	0.434

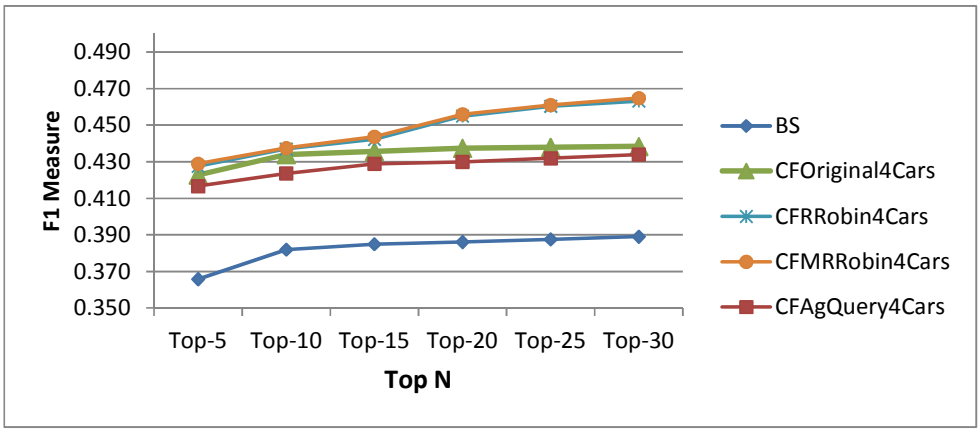


Figure 5.18: F1 Measure Results of Different Models for User Profile UT_4

(iii) Comparison of different user profiles in the same recommendation model

This section compares the results of each of the proposed models for different user profiles. Based on Table 5.22, Table 5.25, Figure 5.21 and Figure 5.24, the F1 Measure results of the CFRRobin and the CFMRRobin models for different user profiles, show that the performance of the models by using user profiles generated from the last 2 cars, the last 3 cars and the last 4 cars are in descending order. In addition, based on Table 5.28 and Figure 5.27, the F1 Measure results of the CFAgQuery is CFAgQuery1Cars > CFAgQuery2Cars > CFAgQuery3Cars > CFAgQuery4Cars, which shows that the F1 Measure results decrease when the number of 'last cars' (cars which are viewed later by a user) considered in generating user profiles increase. These results indicate that products that are recently viewed by users contribute to more accurate user profiles than earlier-viewed products. Earlier-viewed products might not really represent the user's current preferences and thus, the accuracy of the user profiles decreases when earlier viewed products are used to generate profiles rather than later viewed ones. As mentioned in section 5.3.2 (i), in the testing data, there are more user sessions that contain products with diverse attribute values, than there are sessions not containing products with diverse attribute values, which means that many users look at products with different attribute values. Therefore, generating user profiles from lists of recent products viewed by the user can represent recent user preferences more precisely. Furthermore, the F1 Measure results for the CFRRobin and the CFMRRobin using profiles generated from the last 2 cars, are the highest (i.e. for top 15, top 20, top 25

and top 30). These results show that this model performs the best when only few last products are used to generate user profiles because these profiles match the recent user preferences and also can be used to retrieve quite diverse products based on the neighbours' products.

Based on Table 5.20, Table 5.23, Table 5.26, Figure 5.19, Figure 5.22 and Figure 5.25, the precision results reveal that the profiles generated from the last product, the last 2 products, the last 3 products and the last 4 products perform in descending order for the CFRRobin, the CFMRRobin and the CFAgQuery. These results demonstrate that the more recently products viewed by the users contribute to more accurate user profiles being generated, which can represent user preferences more precisely. The users have more ideas about the products they are looking for after viewing several products. Hence, the products that are recently viewed by the users can more easily satisfy the user's needs and may better represent the user's preferences.

The recall results will be discussed based on Table 5.21, Table 5.24, Table 5.27, Figure 5.20, Figure 5.23 and Figure 5.26. The CFAgQuery has higher recall results for the target user profiles generated from less number of last products viewed by a user than more of last-viewed products. By using more of last-viewed products to generate profiles, more diverse user profiles might be generated and thus there are more chances to get neighbours with diverse products. Only the maximum value of each attribute is selected as the query's values and thus the user's query generated based on diverse products might not fully represent the user preferences. Therefore, the aggregated query generated based on diverse products might not retrieve products that satisfy the user's preferences. In contrast, the recall results for the CFRRobin

and the CFMRRobin for user profiles generated from the last product are the lowest, followed by the profiles created from the last 3 products and the last 4 products viewed. This is because the CFRRobin and the CFMRRobin use the neighbour users' products to retrieve other relevant products. By using more diverse products of the neighbour users as queries, more diverse products can be retrieved. The best recall for the CFRRobin and the CFMRRobin is for profiles created from the last 2 products. This result shows that the last 2 products can represent the recent user's preferences and can also be used to retrieve more diverse products. Therefore, by using only few recent products to generate user profiles, the CFRRobin and CFMRRobin models can retrieve more diverse products that match the user's preferences.

Table 5.20: Precision Results of the CFRRobin Model for All Profiles

Precision	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
CFRRobin1Cars	0.597	0.589	0.581	0.572	0.562	0.551
CFRRobin2Cars	0.571	0.563	0.555	0.547	0.538	0.529
CFRRobin3Cars	0.561	0.555	0.548	0.539	0.530	0.520
CFRRobin4Cars	0.546	0.540	0.531	0.523	0.513	0.504

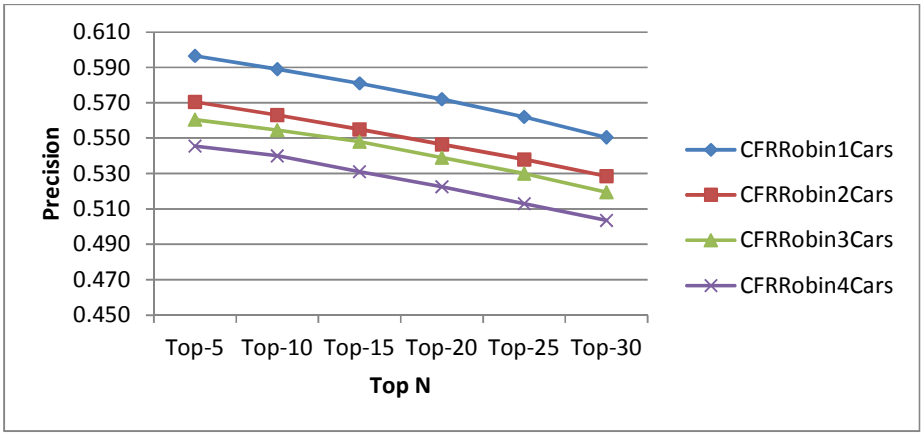


Figure 5.19: Precision Results of the CFRRobin Model for All Profiles

Table 5.21: Recall Results of the CFRRobin Model for All Profiles

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
CFRRobin1Cars	0.360	0.368	0.374	0.379	0.385	0.390
CFRRobin2Cars	0.364	0.378	0.390	0.414	0.425	0.435
CFRRobin3Cars	0.356	0.369	0.380	0.401	0.413	0.422
CFRRobin4Cars	0.352	0.367	0.379	0.403	0.418	0.429

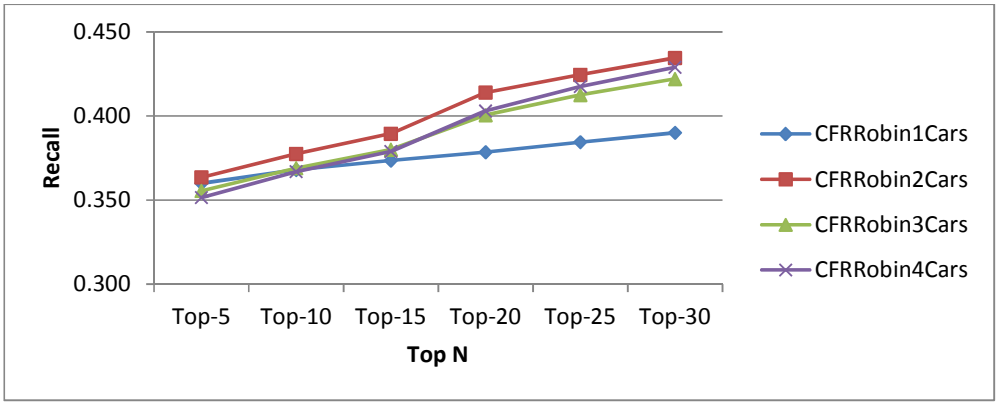


Figure 5.20: Recall Results of the CFRRobin Model for All Profiles

Table 5.22: F1 Measure Results of the CFRRobin Model for All Profiles

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
CFRRobin1Cars	0.449	0.453	0.455	0.456	0.457	0.457
CFRRobin2Cars	0.444	0.452	0.458	0.471	0.475	0.477
CFRRobin3Cars	0.435	0.443	0.449	0.460	0.464	0.466
CFRRobin4Cars	0.428	0.437	0.442	0.455	0.460	0.463

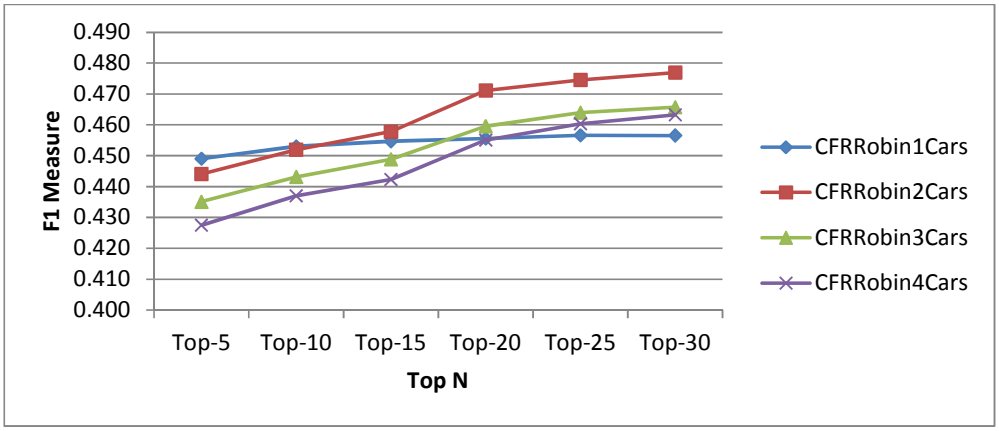


Figure 5.21: F1 Measure Results of the CFRRobin Model for All Profiles

Table 5.23: Precision Results of the CFMRRobin Model for All Profiles

Precision	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
CFMRRobin1Cars	0.597	0.590	0.582	0.574	0.565	0.554
CFMRRobin2Cars	0.573	0.565	0.558	0.549	0.540	0.532
CFMRRobin3Cars	0.562	0.556	0.550	0.541	0.532	0.523
CFMRRobin4Cars	0.547	0.541	0.534	0.525	0.516	0.507

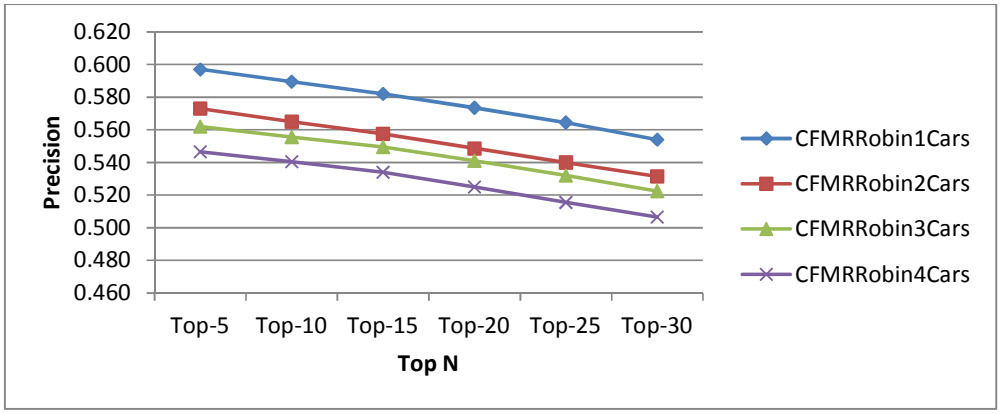


Figure 5.22: Precision Results of the CFMRRobin Model for All Profiles

Table 5.24: Recall Results of the CFMRRobin Model for All Profiles

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
CFMRRobin1Cars	0.361	0.369	0.375	0.380	0.385	0.389
CFMRRobin2Cars	0.365	0.379	0.391	0.414	0.425	0.435
CFMRRobin3Cars	0.357	0.370	0.382	0.401	0.412	0.423
CFMRRobin4Cars	0.353	0.368	0.380	0.403	0.417	0.430

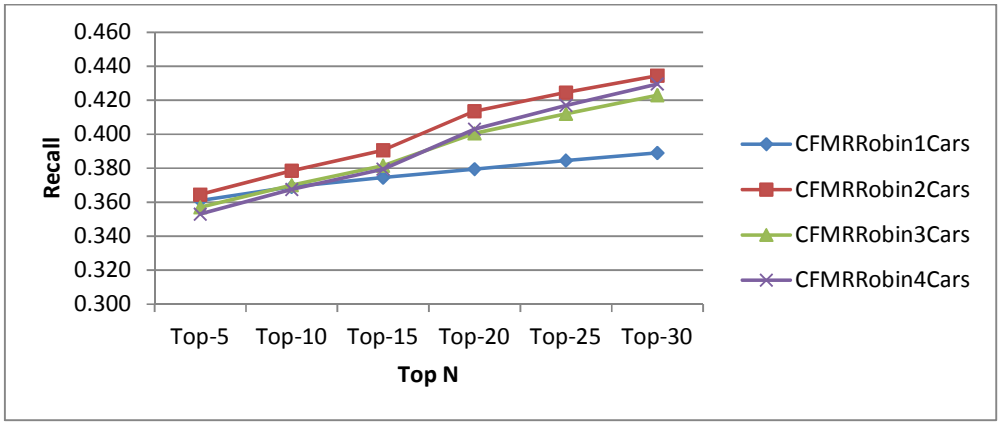


Figure 5.23: Recall Results of the CFMRRobin Model for All Profiles

Table 5.25: F1 Measure Results of the CFMRRobin Model for All Profiles

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
CFMRRobin1Cars	0.450	0.454	0.456	0.457	0.457	0.457
CFMRRobin2Cars	0.446	0.453	0.459	0.472	0.475	0.478
CFMRRobin3Cars	0.437	0.444	0.450	0.460	0.464	0.468
CFMRRobin4Cars	0.429	0.438	0.444	0.456	0.461	0.465

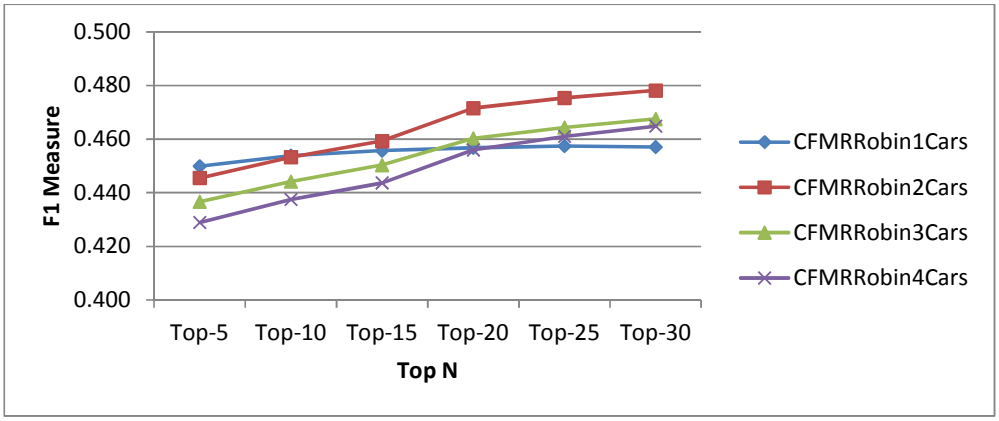


Figure 5.24: F1 Measure Results of the CFMRRobin Model for All Profiles

Table 5.26: Precision Results of the CFAgQuery Model for All Profiles

Precision	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
CFAgQuery1Cars	0.596	0.589	0.579	0.570	0.560	0.550
CFAgQuery2Cars	0.554	0.547	0.537	0.529	0.521	0.513
CFAgQuery3Cars	0.548	0.542	0.535	0.526	0.518	0.510
CFAgQuery4Cars	0.537	0.529	0.521	0.513	0.507	0.499

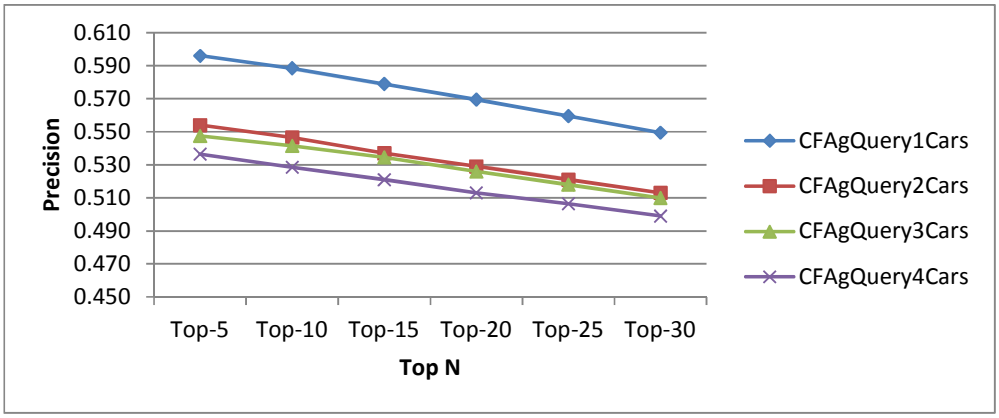


Figure 5.25: Precision Results of the CFAgQuery Model for All Profiles

Table 5.27: Recall Results of the CFAgQuery Model for All Profiles

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
CFAgQuery1Cars	0.361	0.369	0.376	0.382	0.387	0.393
CFAgQuery2Cars	0.352	0.364	0.375	0.382	0.387	0.394
CFAgQuery3Cars	0.344	0.357	0.366	0.373	0.379	0.386
CFAgQuery4Cars	0.341	0.354	0.365	0.370	0.377	0.384

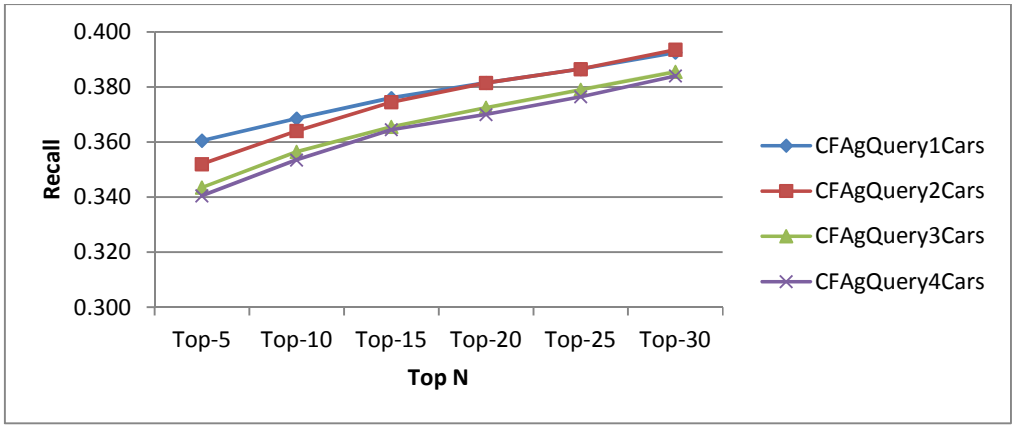


Figure 5.26: Recall Results of the CFAgQuery Model for All Profiles

Table 5.28: F1 Measure Results of the CFAgQuery Model for All Profiles

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
CFAgQuery1Cars	0.449	0.453	0.456	0.457	0.457	0.458
CFAgQuery2Cars	0.430	0.437	0.441	0.443	0.444	0.445
CFAgQuery3Cars	0.422	0.430	0.434	0.436	0.438	0.439
CFAgQuery4Cars	0.417	0.424	0.429	0.430	0.432	0.434

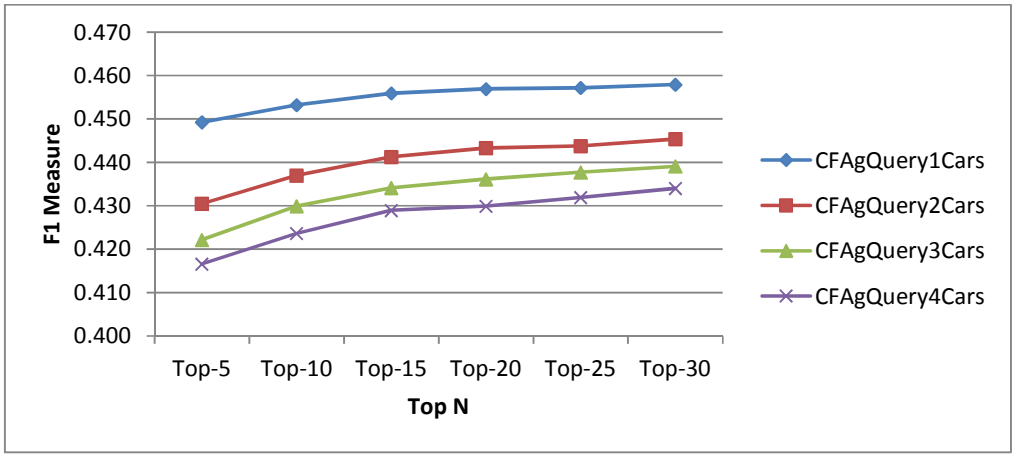


Figure 5.27: F1 Measure Results of the CFAgQuery Model for All Profiles

5.3.3 The utilization of association rules extracted from user review data and user profiles generated from user click streams data to expand a user's query (Hypothesis 3)

The objective of this set of experiments is to verify that the proposed query expansion methods that utilize association rules extracted from user reviews data and user profiles generated from user click streams data can improve the recommendation (Hypothesis 3). The experiments have also been conducted to compare the performance of the different query expansion methods:- the OMQE, the CFAgQuery, and the TUProfile. The results are compared based on two categories; i) between the OMQE model and the BS model and ii) different methods for query expansion. The results of the experiments are firstly discussed based on the tables and figures that will be given after the discussion.

(i) Comparison between the OMQE model and the BS model

The objective of this set of experiments is to verify that the query expansion approach using the associations between product attributes generated based on user review data can improve the recommendation accuracy. Based on the F1 Measure results given in Table 5.31 and Figure 5.30, the OMQE model performs lower than the BS model. This is because the recall results of the OMQE as illustrated in Table 5.30 and Figure 5.29 are much lower than for the BS. As discussed in section 5.3.2 (i), more sessions in the dataset have diverse products, which means the user looked at products with different attribute values. In the OMQE, the user's initial query is expanded with more attribute values based on the association rules between attribute values generated from user reviews. The expanded query has more attribute values than the initial and thus, may retrieve products with more

focused attribute values than the initial query. In contrast, the BS uses the original query to retrieve products in which the query only contains part of the attributes in the query expansion and thus may retrieve products with diverse attribute values. Therefore, products retrieved by the BS may match more products in the testing data which result in high recall results. However, the precision results for the OMQE, as illustrated in Table 5.29 and Figure 5.28, are better than for the BS because the OMQE uses the expanded query that can represent the user preferences more precisely and thus can retrieve more products that satisfy the user's needs. The precision results of the OMQE verify that **Hypothesis 3** is valid.

Table 5.29: Precision Results of the OMQE Model

Precision	**Top-5**	**Top-10**	**Top-15**	**Top-20**	**Top-25**	**Top-30**
BS	0.379	0.383	0.382	0.380	0.380	0.381
OMQE	0.446	0.442	0.436	0.431	0.426	0.419

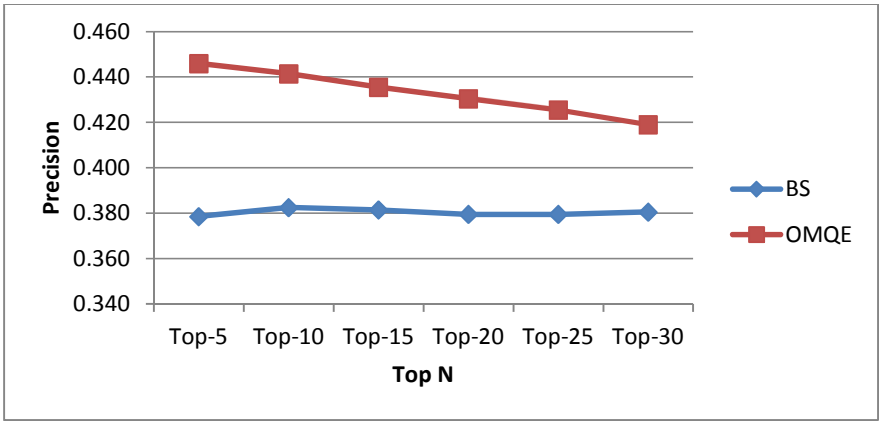


Figure 5.28: Precision Results of the OMQE Model

Table 5.30: Recall Results of the OMQE Model

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.354	0.382	0.389	0.393	0.396	0.398
OMQE	0.267	0.277	0.286	0.291	0.301	0.311

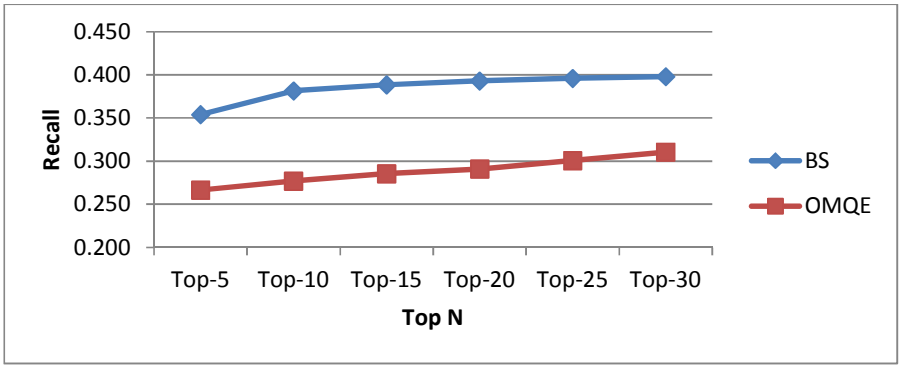


Figure 5.29: Recall Results of the OMQE Model

Table 5.31: F1 Measure Results of the OMQE Model

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
BS	0.366	0.382	0.385	0.386	0.388	0.389
OMQE	0.334	0.340	0.345	0.347	0.352	0.357

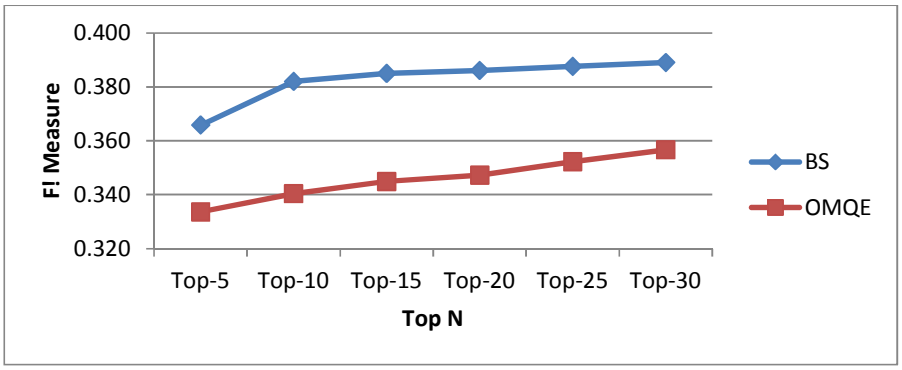


Figure 5.30: F1 Measure Results of the OMQE Model

(ii) Comparison between different methods for query expansion

The objective of this set of experiments is to compare three different proposed query expansion methods which are the OMQE, TUProfile and CFAgQuery. The OMQE method utilizes association rules extracted from user reviews data whereas the TUProfile and the CFAgQuery methods utilize the proposed user profiles generated from user click streams data to expand a user's query. Based on the F1 Measure results given in Table 5.40, Table 5.41, Table 5.42, Table 5.43 and illustrated in Figure 5.39, Figure 5.40, Figure 5.41 and Figure 5.42, the TUProfile model performs slightly better than the CFAgQuery model for profiles generated from the last product, the last 2 products and the last 3 products viewed by users, whereas the CFAgQuery performs slightly better than the TUProfile for profiles generated from the last 4 cars. On the contrary, based on the precision result given in Table 5.32, Table 5.33, Table 5.34, Table 5.35 and illustrated in Figure 5.31, Figure 5.32, Figure 5.33 and Figure 5.34, the CFAgQuery model performs better than the TUProfile model for profiles generated from the last 2 product, the last 3 products and the last 4 products viewed, whereas the TUProfile performs slightly better than the CFAgQuery for profiles generated from the last car. Furthermore, in terms of recall, the TUProfile model performs slightly better than the CFAgQuery model for all profiles based on the results given in Table 5.36, Table 5.37, Table 5.38, Table 5.39 and illustrated in Figure 5.35, Figure 5.36, Figure 5.37 and Figure 5.38. The precision results show the aggregated query generated from the neighbour users' profiles as implemented by the CFAgQuery may retrieve more products that satisfy the target user's needs. However, the recall results show the user profiles

generated from the target user's click stream data as implemented by the TUProfile may represent diverse preferences of the user and thus can retrieve more types of products that may satisfy the user's preferences. In addition, the OMQE model performs the worst among the query expansion methods in terms of F1 Measure, precision and recall results. These results demonstrate that the query expansion method that uses user profiles generated from user click streams data as implemented by the CFAgQuery and the TUProfile performs better than the query expansion method that utilizes association rules generated from user reviews data as implemented by the OMQE. This is because user profiles generated from user click streams data may represent target users' preferences more precisely compared to association rules generated from user reviews data that represent the user's preferences based on the previous users' preferences.

Table 5.32: Precision Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_1

Precision	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.446	0.442	0.436	0.431	0.426	0.419
TUProfile1Cars	0.598	0.59	0.581	0.572	0.562	0.552
CFAgQuery1Cars	0.596	0.589	0.579	0.57	0.56	0.55

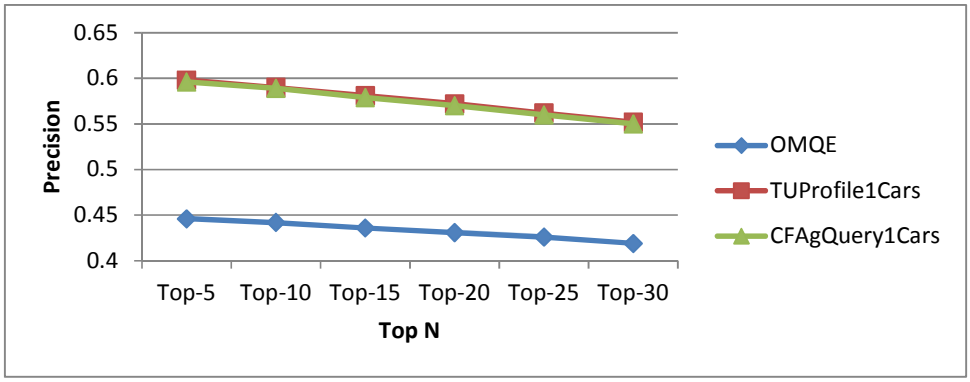


Figure 5.31: Precision Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_1

Table 5.33: Precision Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_2

Precision	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.446	0.442	0.436	0.431	0.426	0.419
TUProfile2Cars	0.546	0.538	0.53	0.522	0.514	0.507
CFAgQuery2Cars	0.554	0.547	0.537	0.529	0.521	0.513

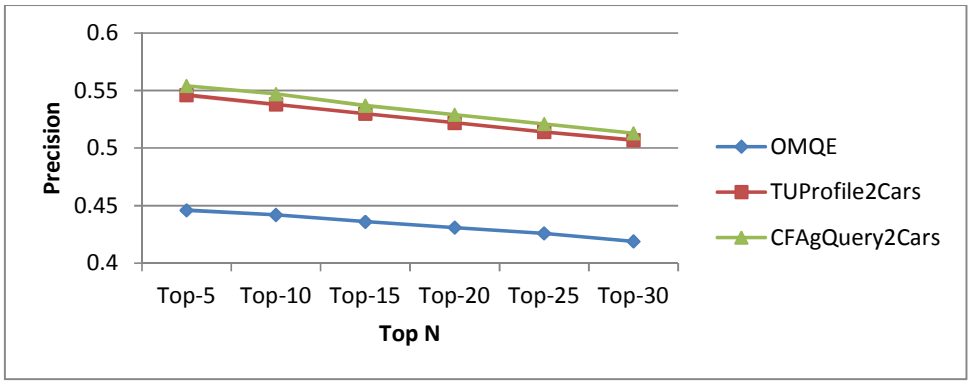


Figure 5.32: Precision Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_2

Table 5.34: Precision Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_3

Precision	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.446	0.442	0.436	0.431	0.426	0.419
TUProfile3Cars	0.541	0.534	0.527	0.52	0.513	0.505
CFAgQuery3Cars	0.548	0.542	0.535	0.526	0.518	0.51

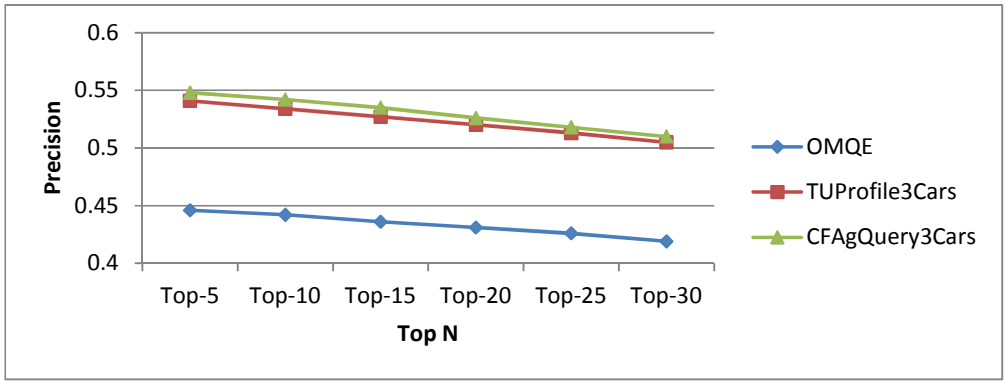


Figure 5.33: Precision Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_3

Table 5.35: Precision Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_4

Precision	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.446	0.442	0.436	0.431	0.426	0.419
TUProfile4Cars	0.526	0.519	0.512	0.505	0.498	0.492
CFAgQuery4Cars	0.537	0.529	0.521	0.513	0.507	0.499

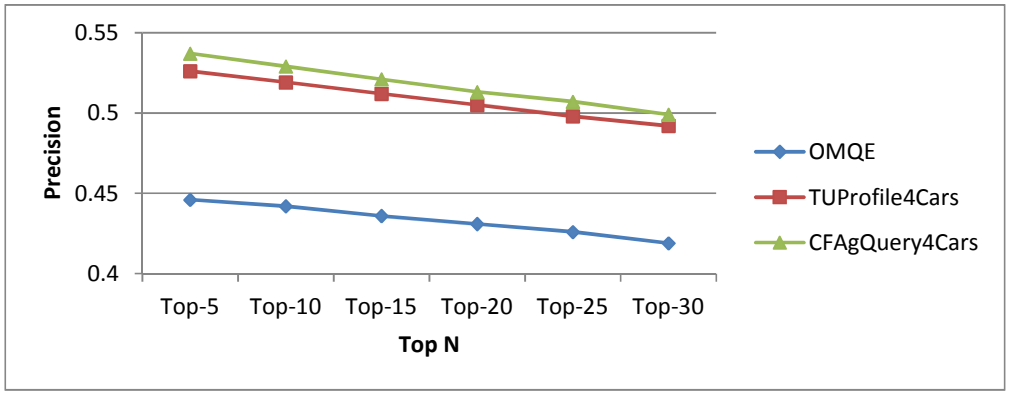


Figure 5.34: Precision Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_4

Table 5.36: Recall Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_1

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.267	0.277	0.286	0.291	0.301	0.311
TUProfile1Cars	0.363	0.371	0.377	0.381	0.387	0.392
CFAgQuery1Cars	0.361	0.369	0.376	0.382	0.387	0.393

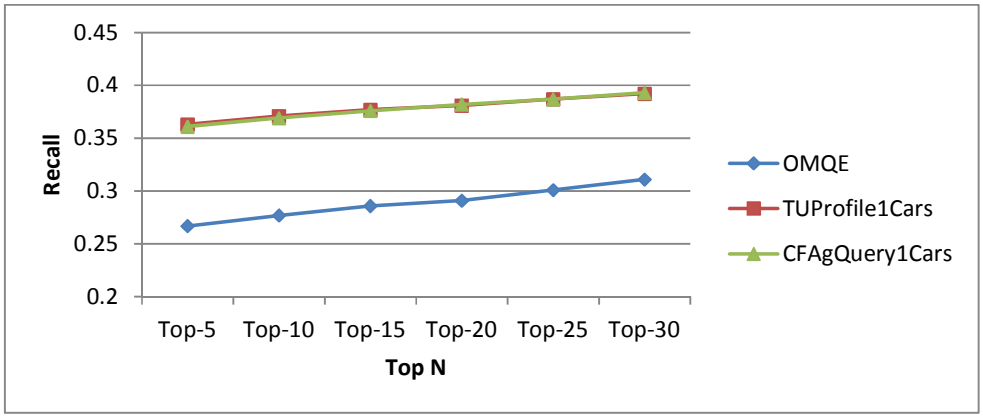


Figure 5.35: Recall Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_1

Table 5.37: Recall Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_2

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.267	0.277	0.286	0.291	0.301	0.311
TUProfile2Cars	0.357	0.371	0.382	0.388	0.396	0.403
CFAgQuery2Cars	0.352	0.364	0.375	0.382	0.387	0.394

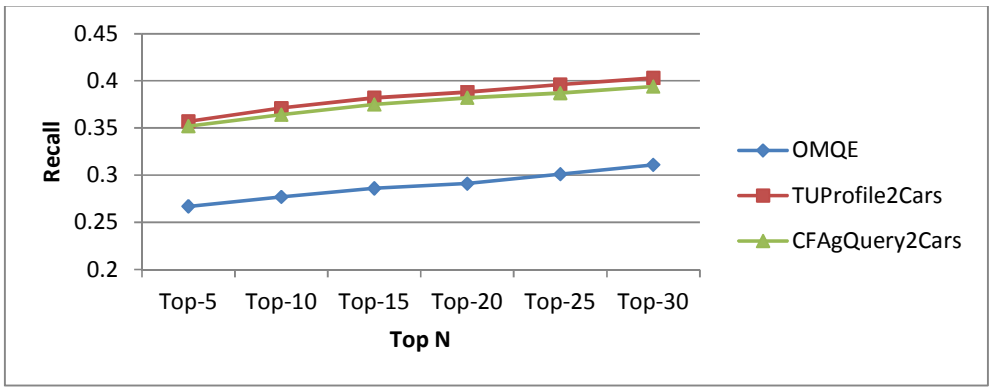


Figure 5.36: Recall Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_2

Table 5.38: Recall Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_3

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.267	0.277	0.286	0.291	0.301	0.311
TUProfile3Cars	0.347	0.362	0.372	0.379	0.386	0.394
CFAgQuery3Cars	0.344	0.357	0.366	0.373	0.379	0.386

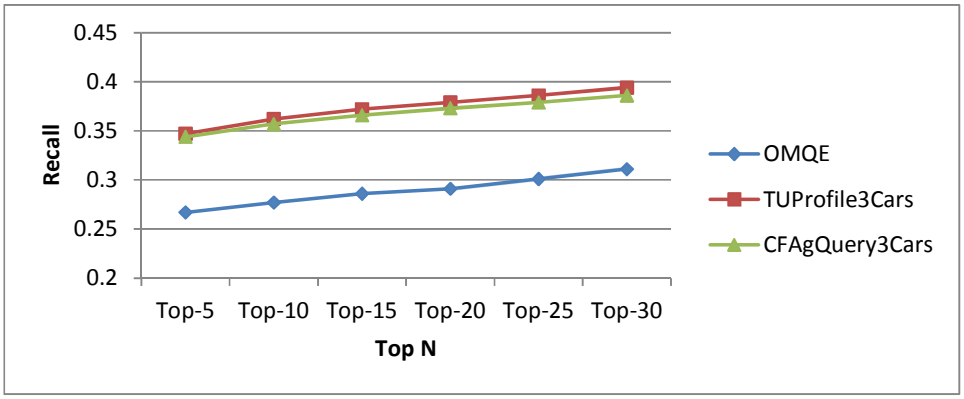


Figure 5.37: Recall Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_3

Table 5.39: Recall Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_4

Recall	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.267	0.277	0.286	0.291	0.301	0.311
TUProfile4Cars	0.341	0.356	0.367	0.374	0.383	0.391
CFAgQuery4Cars	0.341	0.354	0.365	0.37	0.377	0.384

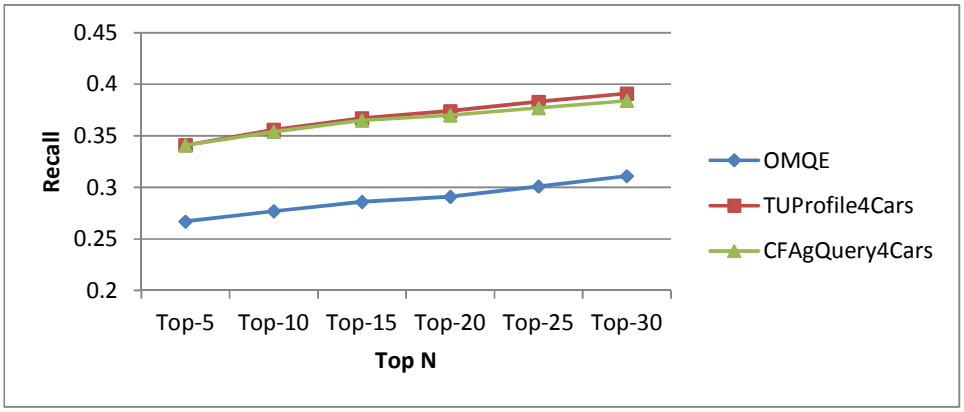


Figure 5.38: Recall Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_4

Table 5.40: F1 Measure Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_1

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.334	0.34	0.345	0.347	0.352	0.357
TUProfile1Cars	0.451	0.455	0.457	0.457	0.458	0.458
CFAgQuery1Cars	0.449	0.453	0.456	0.457	0.457	0.458

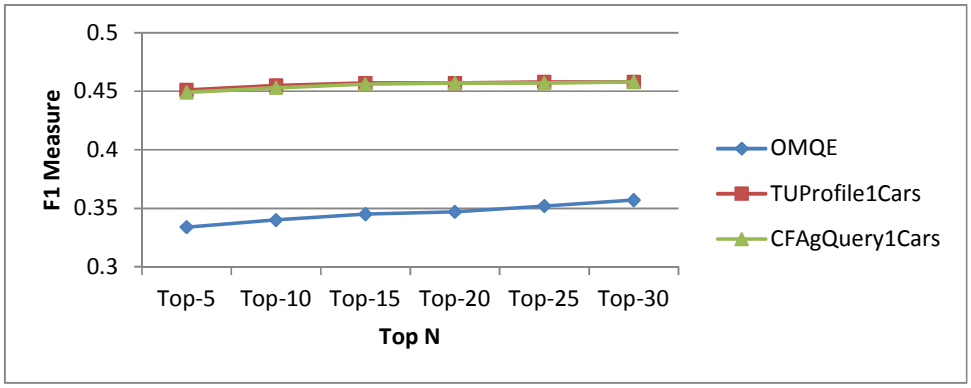


Figure 5.39: F1 Measure Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_1

Table 5.41: F1 Measure Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_2

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.334	0.34	0.345	0.347	0.352	0.357
TUProfile2Cars	0.431	0.439	0.443	0.445	0.447	0.449
CFAgQuery2Cars	0.43	0.437	0.441	0.443	0.444	0.445

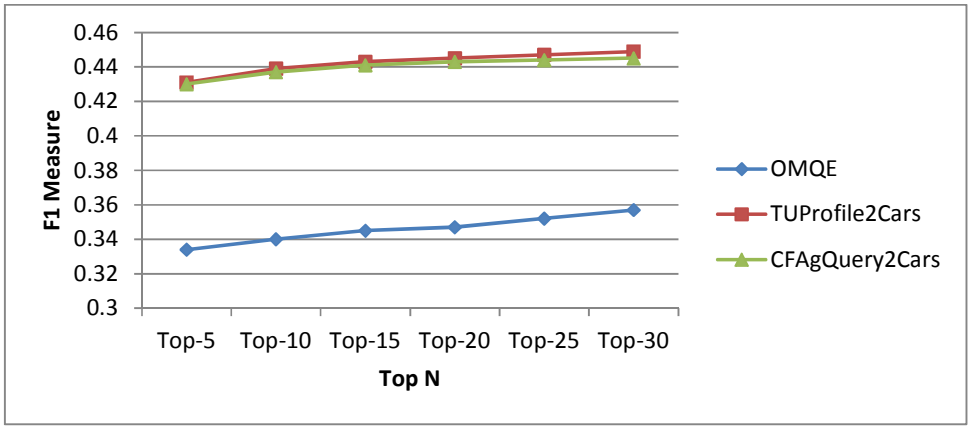


Figure 5.40: F1 Measure Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_2

Table 5.42: F1 Measure Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_3

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.334	0.34	0.345	0.347	0.352	0.357
TUProfile3Cars	0.422	0.431	0.436	0.438	0.44	0.443
CFAgQuery3Cars	0.422	0.43	0.434	0.436	0.438	0.439

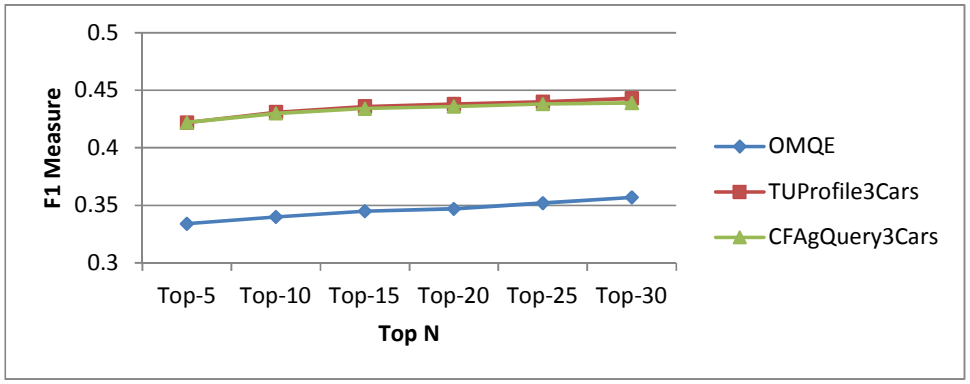


Figure 5.41: F1 Measure Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_3

Table 5.43: F1 Measure Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_4

F1 Measure	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
OMQE	0.334	0.34	0.345	0.347	0.352	0.357
TUProfile4Cars	0.414	0.422	0.427	0.43	0.432	0.436
CFAgQuery4Cars	0.417	0.424	0.429	0.43	0.432	0.434

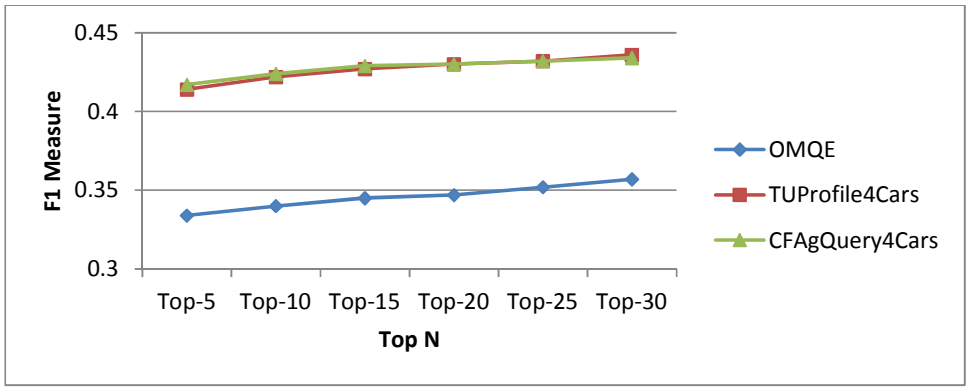


Figure 5.42: F1 Measure Results of the OMQE, TUProfile and CFAgQuery Models for User Profile UT_4

5.3.4 T-Test Evaluation

T-test evaluation has been conducted for the F1 Measure and precision results to test whether the performance of the proposed approaches are statistically significant. To compare each proposed approach with each baseline model, the p value is calculated by comparing the F1 Measure and precision values of each user of the proposed model and the base line model. If the p value is less than 0.05, the performance of the proposed approach is considered significantly improved. Table 5.44 and Table 5.45 show the T-test evaluation results for each proposed approach compared to each baseline model. The results show all the p values of each proposed approach compared to the baseline model are less than 0.05 for the F1 Measure and precision, which means all the proposed approaches significantly outperform the baseline models in regards to F1 Measure and the precision.

Table 5.44: F1 Measure T-Test Results of the Proposed Models

	BS	CFOrigina1Cars	CFOrigina2Cars	CFOrigina3Cars	CFOrigina4Cars
CFRRobin1Cars	7.67E-128	0.00E+00	NA	NA	NA
CFRRobin2Cars	4.29E-72	NA	0.00E+00	NA	NA
CFRRobin3Cars	1.81E-51	NA	NA	0.00E+00	NA
CFRRobin4Cars	1.36E-38	NA	NA	NA	0.00E+00
TUProfile1Cars	4.19E-137	0.00E+00	NA	NA	NA
TUProfile2Cars	3.72E-51	NA	0.00E+00	NA	NA
TUProfile3Cars	8.82E-37	NA	NA	0.00E+00	NA
TUProfile4Cars	3.95E-24	NA	NA	NA	0.00E+00
CFAgQuery1Cars	4.66E-127	0.00E+00	NA	NA	NA
CFAgQuery2Cars	1.12E-44	NA	0.00E+00	NA	NA
CFAgQuery3Cars	8.74E-34	NA	NA	0.00E+00	NA
CFAgQuery4Cars	7.61E-22	NA	NA	NA	0.00E+00

Table 5.45: Precision T-Test Results of the Proposed Models

	BS	CFOrigina1Cars	CFOrigina2Cars	CFOrigina3Cars	CFOrigina4Cars
CFRRobin1Cars	0.00E+00	0.00E+00	NA	NA	NA
CFRRobin2Cars	1.97E-168	NA	6.20E-186	NA	NA
CFRRobin3Cars	1.80E-139	NA	NA	4.95E-195	NA
CFRRobin4Cars	3.57E-112	NA	NA	NA	2.80E-161
TUProfile1Cars	0.00E+00	0.00E+00	NA	NA	NA
TUProfile2Cars	1.46E-126	NA	4.22E-108	NA	NA
TUProfile3Cars	2.50E-109	NA	NA	1.07E-120	NA
TUProfile4Cars	9.87E-85	NA	NA	NA	1.62E-98
CFAgQuery1Cars	2.72E-305	0.00E+00	NA	NA	NA
CFAgQuery2Cars	1.15E-134	NA	1.26E-139	NA	NA
CFAgQuery3Cars	7.75E-116	NA	NA	2.50E-147	NA
CFAgQuery4Cars	3.86E-92	NA	NA	NA	9.08E-122

5.4 CHAPTER SUMMARY

This chapter evaluated the effectiveness of the proposed user profiling and recommendation approaches. A real world dataset collected from one of the online car sale companies in Australia is used for conducting the experiments. The experiments involve only one dataset because the public datasets that are currently available are not suitable to be exploited by the proposed user profiling and recommendation approaches. The currently available datasets are for products that are frequently purchased by users such as books and movies, and contain only few attributes besides ratings data that is usually utilized by the standard collaborative filtering approach. The user profiling and proposed recommendation approaches are developed for infrequently purchased products. For this kind of product, usually users look at the features of the product to make decisions about which products they want to purchase. The main features or attribute values of the products are important factors for users to consider when selecting such products to buy and thus, need to be utilized by the proposed recommendation approaches. The proposed approaches can only be evaluated using datasets that contain important features or attribute values of the products to recommend.

The proposed approaches involve processing of user reviews and online click streams data to generate association rules or user profiles. However, scalability is not a big issue because the proposed approaches include offline and online processes. In the OMQE approach, association rules are generated offline and the selection of a candidate rule from the generated rules for expanding a user's query is performed online. Whereas, in the CFRRobin, CFMRRobin and CFAgQuery, previous users' profiles are generated offline and the generation of target user profile, neighbourhood formation and product recommendations are performed online. The offline processes

in the proposed approaches helps to reduce time required for recommending products and also overcome the scalability issue when using a big dataset. In addition, the proposed approaches have been implemented as a prototype system and the performance of the proposed approaches are good with the available data.

The experiment results show that association rules generated from user reviews data and user profiles generated from product click stream data can improve recommendation accuracy. In addition, the experiment results also prove that the query expansion and the integration of the collaborative filtering and search-based approaches can improve recommendation accuracy. On the other hand, the experiment results show that the recall decreases for most cases. The recall decrease is expected for big datasets because we don't really know how many of the products are actually relevant to the user query. The proposed approaches have lower recall results because of the diversity of the products viewed in user sessions that are used in the experiments. Accuracy is the focus of this research rather than diversity. The proposed approaches are developed by assuming that users have focused attribute values preferences, in that they like products that have similar attribute values. Therefore, the proposed approaches recommend products that have attribute values most preferred by the users. They retrieve more products that match with the earlier products in the testing dataset, which are closer to the products used for generating profiles and thus satisfy the user preferences more precisely.

The experiment results also show that the profiles generated from less number of 'last cars' (cars which are viewed later by a user) work better than profiles generated from more 'last cars'. This is because user profiles generated using the proposed user profiling method includes only one value for each attribute. If a user has diverse preferences for each attribute, only the attribute value that is most liked

by the user will be considered in the profile generated. The results are not good for profiles generated using more cars because the profiles do not represent diverse preferences of the users but only represent the attribute values that are most liked by the users. As a result, the proposed approaches only recommend products that are most liked by the users and not all possible products that they would like to see. However, as mentioned before, the diversity is not a target of this research. In future works, user profiles with multiple values for each attribute can be considered to represent diverse preferences of the users in order to recommend more diverse products to users. On the other hand, the results reveal that more recent products viewed by users can represent recent user preferences more precisely and generate more accurate user profiles and thus, improve the accuracy of the recommendation approaches. Finally, based on the t-test evaluation, the F1 Measure and precision results of the proposed approaches are proved to be significantly improved.

Chapter 6: Conclusions and Future Works

6.1 CONCLUSIONS

Recommender systems (RS) have been widely applied for recommending products on e-commerce sites to overcome information overload issues and help users in selecting final products to purchase. The popularly used collaborative filtering (CF) recommendation approach requires a large amount of explicit ratings data for making meaningful recommendations. However, this data is not always available as it requires high involvement from the users to provide explicit ratings of the products they already know. Consequently, recommender systems are not currently popular for recommending luxury products because the users only purchase few such items in their lifetime and thus, it is impossible for them to provide sufficient ratings for the recommender system to make meaningful recommendation. Therefore, the main focus of this thesis is to explore new information sources to extract knowledge about users' preferences and to develop recommendation approaches that can utilize the extracted knowledge for recommending such products.

Fortunately, the emergence of the Web 2.0 provides numerous user generated content such as blogs, reviews, and tags for use in understanding users' preferences. In addition, the user click stream data can be easily collected when a user browses products on an e-commerce site. This data shows products that are of interest to the user. The availability of rich user information from this data offers new possible solutions to extract users' preferences for attribute values, which can be used by recommender system approaches to find products that best match the users'

requirements. This thesis investigated user reviews and click stream data to extract knowledge for recommending infrequently purchased products. User reviews from previous users are used to generate association rules between attributes' values. These rules show other attribute values that have been liked by other users based on the known attribute values already provided by the target user. These rules are then used by the proposed OMQE recommendation approach to expand the users' queries in order to retrieve more relevant products. This thesis also investigated user click stream data to profile users for recommending infrequently purchased products. For this kind of product, users always looked at the product features when selecting a product to purchase. Based on the products viewed by a user in the click stream data, the user profiles which represent the user's preferences for each attribute of the products are generated. The generated user profile represents the weight of each attribute value of all the products that have been viewed by the user, which shows how much the user likes each attribute value.

Furthermore, this thesis also explored how to utilise user profiles generated from the proposed user profiling approaches to recommend infrequently purchased products. Four recommendation models have been proposed:

- **TUProfile**

  This model makes use of the user profile as the new user's query to retrieve relevant products. For each product attribute, the maximum value of this attribute in the user profile is selected as the attribute value of the user's query. Products that match the attribute values in the query are ranked and selected based on their similarity to the user's query.

- **CFAgQuery**

  This model employs the collaborative filtering approach to find similar or neighbour users based on the target user's profile and the previous users' profiles. Based on the products viewed by the neighbour users and their similarities with the target user's profile, a new query is generated and used to retrieve products. The products are then ranked and selected based on their similarity with the user profile.

- **CFRRobin**

  This model integrates collaborative filtering and search-based approaches to recommend products based on products that have been liked by the neighbour users. Instead of recommending products that the neighbour users have liked, this model used the products as queries to retrieve other relevant products. The products retrieved by these queries are then selected based on the Round Robin algorithm and the final products are ranked and then selected based on their similarities with the target user's profile.

- **CFMRRobin**

  This recommendation approach also integrates the collaborative filtering and search-based approaches by using each product of the neighbour users as a query. This approach is different from the CFRRobin because it ranks the retrieved lists of each neighbour before using the Round Robin data fusion to merge the retrieved products. It also ranks and selects the final products based on their similarities with the target user's profile.

The OMQE, TUProfile and CFAgQuery models aim to generate a new query that better represents the user's requirements. These models formulate a new query

based on the knowledge about the user preferred attributes values which are extracted from the user reviews or the click stream data. Besides, the CFAgQuery, the CFRRobin and the CFMRRobin models employ a collaborative filtering approach based on user profiles generated by using the proposed user profiling method for recommending products based on the neighbour users' products.

This thesis also conducted extensive evaluation experiments on a real world dataset collected from one of the leading online car sales companies in Australia. The experiment results demonstrate that integrating collaborative filtering and search-based approaches and utilising user profiles generated from user click stream data can improve the performance of the recommendation approach. The experiment results also show that utilising association rules between products' attribute values extracted from user reviews data and user profiles generated from user click streams can improve the precision results. However, the recall results for the proposed models are lower than those for the baseline models because many of the user sessions contain diverse products where these users viewed products with different attribute values. The proposed models retrieve more products that match the earlier-viewed products than the baseline models. The earlier-viewed products are closer to products used to generate user profiles than the later-viewed products and thus, the proposed approaches suggest products that most satisfy the user's preferences or are highly preferred by the user. The results of the experiments also suggest that the more recent products viewed by users can better represent the user preferences than products viewed earlier by users. This is because the users have more knowledge about the products they are looking for after viewing some products and thus, the more recent products can better represent the users' interests.

6.2 CONTRIBUTIONS

This thesis makes contributions to the web personalization and recommender systems. This thesis explores user review data available in Web 2.0 to get target user preferred product attribute values based on products that are preferred by the previous users. Currently, many works have been conducted to profile users based on user generated contents in Web 2.0. However, little work utilizes user review data to extract knowledge for better representing the user requirements. The proposed query expansion model based on association rules generated from the user review data contributes to the improvement of the searching method by representing the user's query more accurately. Thus, this thesis contributes to effectively utilising the new Web 2.0 user information resource, that is, user review data.

This thesis also contributes to web personalization as it focuses on how to profile users based on online click stream data. The user profiling approach proposed in this thesis can represent user preferences for products' attribute values based on products viewed in the online click stream data. These user profiles can be utilized by the search-based approach to represent the users' queries more accurately or can be utilized by the collaborative filtering recommendation approach to find the neighbour users. Therefore, the collaborative filtering approach can be applied to recommend products without the availability of a large amount of ratings data and can solve the sparsity or new user problems of the collaborative filtering approach.

Currently, recommender systems have been popularly developed for recommending simple products that are frequently purchased by users as explicit ratings data can be easily collected from the user. The current recommender approach is not applicable when the explicit ratings data is not available. The proposed user profiling and recommendation approaches do not depend on ratings

data to make meaningful recommendation and thus, they can be employed to recommend a wide range of products such as luxury and expensive products. Thus, this research contributes to further development of recommender system applications for all kinds of products without requiring high involvement from users.

6.3 LIMITATIONS AND FUTURE WORKS

6.3.1 Limitations

The limitations of the research in this thesis are as discussed below:

1. The user profiling and recommendation approaches proposed by this thesis require data about some features or attribute values of the products. This information is used to generate the user's preference for each attribute value. The experiments conducted only involve one dataset that contains user reviews and click stream data for online car search. This data is provided by the QUT industry partner for improving its online car sale system. It is difficult to get other suitable datasets for conducting more experiments. The available free datasets contain ratings data and only few product attributes that are only suitable to be utilized by the original collaborative filtering recommendation approach, but not suitable to test the performance of the proposed approaches. Thus, this thesis cannot test the performance of the proposed approaches for different products in order to compare the experiment results with the dataset that has been used.
2. The proposed recommendation approaches has lower recall compared to the baseline models. This is because products viewed by the users

in the click stream data are diverse, which means the users viewed products with different attribute values. The proposed approaches only recommend products that are most preferred by the users without considering other products that may have other attribute values that are of interest to the users. The proposed approaches may produce better recall results for users who have focused attribute interests, in which the products viewed have similar attribute values.

6.3.2 Future Works

This research works could be extended in the following directions:

1. Further experiments using different datasets should be conducted to examine the performance of the proposed approaches for a wider range of products. This can be done by modifying the proposed approaches to suit the available data or by getting more attribute values of the products in the available datasets.
2. In the OMQE approach, only one association rule that has the highest accuracy is used to expand the user's query. This work can be extended by considering some association rules to expand the query and using weights for different values of the same attribute to retrieve more diverse products and thus, may improve the recall of the OMQE approach.
3. The user reviews also contain user comments on product usage features. The examples of usage features for the cars domain are Comfort, Practicality, Aesthetics Styling, Under Bonnet, and Safety Security. These features can also be utilised in the OMQE approach to recommend products that most likely satisfy the user's requirements.

4. The proposed approaches assume that users have the same preferences for all the product attributes. However, this is not always true because some attributes may have more influence than other attributes on the decisions made by users. This work can be extended by finding the suitable weight value for each attribute based on the importance of the product attribute to the user. The weights of attributes can be utilised by the proposed recommendation approach to retrieve more relevant products.

5. Some products viewed by the user in the click stream data may not really be of interest to the user. This research could be extended to utilize more information from the user click stream data such as for how long the user has viewed each product. This information can be used in order to improve the results of the proposed recommendation approaches.

6. The proposed approaches generate a target user's profile from the user's online click streams data and previous users' profiles from the log data. Implicit feedback gathered from other portals within the same domain can also be used to integrate knowledge of a community of interest in order to enhance the proposed approaches. The possibilities of utilizing implicit feedback gathered from other portals to enhance the proposed approaches could include:

   - Selecting similar target users from other portals and recommending products based on products that are preferred by the similar target users from other portals. Integrating products preferred by similar users from other portals may

improve recommendations because the chances of getting more similar users from more portals are higher than from only one portal.

- Integrating products preferred by neighbour users from multiple portals and recommending products that are most preferred by the neighbour users. Recommending products preferred by neighbour users from different portals may improve the recommendation because the most similar products can be recommended to users based on products preferred by neighbour users from multiple portals.

Reference and Bibliography

Aciar, S., Zhang, D., Simoff, S. & Debenham, J. (2007). Informed Recommender: Basing recommendations on consumer product reviews. *Intelligent Systems, IEEE, 22*(3), 39-47.

Adomavicius, G. & Tuzhilin, A. (2005). 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, 17*(6), 734-749.

Agrawal, R. & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th Very Large Data Bases (VLDB) International Conference* (pp. 487–499). Santiago, Chile.

Balabanovi, M. & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. In *Communications of the ACM* (Vol. 40, pp. 66-72). New York, NY, USA: ACM.

Billerbeck, B., Scholer, F., Williams, H. E. & Zobel, J. (2003). Query expansion using associated queries. In *Proceedings of the twelfth international conference on Information and knowledge management* (pp. 2-9). ACM, New York, USA.

Billsus, D. & Pazzani, M. J. (2000). User modeling for adaptive news access. *User-Modeling and User-Adapted Interaction*, *10*(23), 147-180.

Bhogal, J., Macfarlane, A. & Smith, P. (2007). A review of ontology based query expansion. *Information Processing and Management*, 43, 866-886.

Borges, J. & Levene, M. (2000). Data mining of user navigation patterns. In *Lecture Notes in Computer Science* (Vol. 1836, pp. 92-111). London, UK: Springer Berlin /Heidelberg.

Breese, J. S., Heckerman, D. & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering, In *Proceedings of 14th Conference Uncertainty in Artificial Intelligence*.

Burke, R. (1999). Integrating knowledge-based and collaborative-filtering recommender systems. In *Proceeding of the Artificial Intelligence for Electronic Commerce: Papers from the AAAI Workshop* (pp. 69-72). Menlo Park, California: The AAAI Press.

Burke, R. (2000). Knowledge-based recommender systems. In A. Kent (Ed.), *Encyclopedia of Library and Information Science* (Vol. 69, pp. 175-186). New York: CRC Press.

Burke, R. (2002). Hybrid recommender systems: survey and experiments. *User Modeling and User-Adapted Interaction, 12*(4), 331-370.

Burke, R. D., Hammond, K. J. & Yound, B. C. (1997). The FindMe approach to assisted browsing. *IEEE Expert, 12*(4), 32-40.

Cao, G., Nie, J.-Y., Gao, J. & Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 243-250). ACM, New York, NY, USA.

Chawla, S. & Bedi, P. (2008). Query expansion using Information Scent. In *International Symposium on Information Technology* (pp. 1-8).

Chen, S. Y. & Liu, X. (2005). Data mining from 1994 to 2004: An application-orientated review. *International Journal of Business Intelligence and Data Mining, 1*(1), 4–21.

Chirita, P.A., Firan, C.S. & Nejdl, W. (2007). Personalized query expansion for the web. In *Proceedings of the 30th annual international ACM SIGIR conference on*

Research and development in information retrieval (pp. 7-14). ACM, New York, NY, USA.

Cho, Y. H. & Kim, J. K. (2004). Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce. *Expert Systems with Applications*. *26*(2), 233-246.

Cho, Y. H., Kim, J. K. & Kim, S. H. (2002). A personalized recommender system based on web usage mining and decision tree induction. *Expert System with Applications*. *23*(3), 329-342.

Christakou, C., Vrettos, S. & Stafylopatis, A. (2007). A hybrid movie recommender system based on neural networks. *International Journal on Artificial Intelligence Tools, 16*(5), 771-792.

Cleverdon, C. W., Mills, J. & Keen, M. (1966). Factors determining the performance of indexing systems. In *ASLIB Cranfield project, Cranfield*.

Cui, H. Wen, J.R., Nie, J.Y. & Ma, W.Y. (2003). Query expansion by mining user logs. *IEEE Transactions on Knowledge and Data Engineering*, *15*(4), 829 – 839.

Degemmis, M., Lops, P. & Semeraro, G. (2007). A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction, 17*(3), 217-255.

Deshpande, M. & Karypis, G. (2004). Item-Based top-N recommendation algorithms. *ACM Trans. Information Systems,* 22*(1),* 143-177.

Ding, X., Liu, B. & Yu, P., S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the International Conference on Web Search and Web Data Mining* (pp. 231-240). Palo Alto, California, USA: ACM.

Felfernig, A. (2005). Koba4MS: Selling complex products and services using knowledge-based recommender technologies. In *Proceedings of the Seventh IEEE*

International Conference on E-Commerce Technology (pp. 92-100). Munich, Germany: IEEE.

Felfernig, A. & Burke, R. (2008). Constraint-based recommender systems: technologies and research issues. In *Proceedings of the 10th International Conference on Electronic Commerce* (Vol. 342). Innsbruck, Austria: ACM.

Felfernig, A., Isak, K., Szabo, K. & Zachar, P. (2007). The VITA financial services sales support environment. In *Proceeding of the National Conference on Innovative Applications of Artificial Intelligence* (pp. 1692-1699). Vancouver, Canada: AAAI Press.

García, E., Romero, C., Ventura, S. & Castro, C. (2007). An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. *Journal of User Modeling and User-Adapted Interaction*, 19*(1), 99-132.*

Goldberg, D., Nichols, D., Oki, B., M. & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, *35*(12), 61-70.

Goldberg, K., Roeder, T., Gupta, D. & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Journal of Information Retrieval*, *4*(3), 133-151.

Greengrass, E. (2000). Information Retrieval: A Survey. *DOD Technical Report.* (Vol. 120600).

Greening, D. R. (1998). Collaborative filtering for web marketing efforts. *The AAAI Workshop on Recommender Systems WS-98-08* (pp. 53-55). Menlo Park, California: AAAI Press.

Harman, D. K. & Voorhees, E. M. (2006). TREC: An overview. *Annual Review of Information Science and Technology. 40*(1), 113-155.

Hill, W., Stead, L., Rosenstein, M. & Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 194-201). Denver, Colorado, United States: ACM Press/Addison-Wesley Publishing Co.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information System, 22*(1), 89-115.

Hu, M. & Liu, B. (2004). Mining and summarizing user reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*(pp. 168-177). Seattle, WA, USA: ACM.

Hu, Y., Koren, Y. & Volinsky, C. (2008). Collaborative Filtering for implicit feedback datasets. *In Proceeding of the Eight IEEE International Conference on Data Mining*(pp. 263 - 272). Los Alamitos, California. Washington ,Tokyo, IEEE Computer Society.

Iaquinta, L., Gentile, A. L., Lops, P., Gemmis, M. & Semeraro, G. (2007). A hybrid content-collaborative recommender system integrated into an electronic performance support system. In *Proceedings of the 7th International Conference on Hybrid Intelligent Systems* (pp. 47 – 52). Kaiserlautern, Germany.

Ishikawa, H., Nakajima, T., Mizuhara, T., Yokoyama, S., Nakayama, J., Ohta, M., et al. (2002). An intelligent web recommendation system: A web usage mining approach. In *Proceedings of the 13th International Symposium, LNCS* (Vol. 2366, pp. 342-350). Lyon, France: Springer Berlin / Heidelberg.

Jin, X., Zhou, Y. & Mobasher, B. (2005). A maximum entropy web recommendation system: combining collaborative and content features. In *Proceedings of the 11th*

ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (pp. 612-617): New York, NY, USA: ACM.

Jung, K., Park, D. & Lee, J. (2004). Hybrid collaborative filtering and content-based filtering for improved recommender system. In M. Bubak et al. (Eds.), *Computational Science* (pp. 295-302). Springer-Verlag Berlin Heidelberg.

Karypis, G. (2001). Evaluation of item-based top-N recommendation algorithms. *In Proceeding of the International Conference on Information and Knowledge Management* (pp. 247-254).

Kim, B. M., Li, Q., Park, C. S., Kim, S. G. & Kim, J. Y. (2006). A new approach for combining content-based and collaborative filters. *Journal of Intelligent Information Systems*, *27*(1), 79-91.

Kim, D. H., Atluri, V., Bieber, M., Adam, N., Yesha, Y. & Im, I., (2004). A click streams-based collaborative filtering personalization model: Towards a better performance. In *Proceedings of the 6th Annual ACM International Workshop on Web Information and data management* (pp. 88-95). Washington DC, USA: ACM.

Kim, Y. S., Yum, B.-J., Song, J. S. & Kim, S. M. (2005). Development of a recommender system based on navigational and behavioural patterns of customers in e-commerce sites. *Journal of Expert Systems with Applications*, *28*(2), 381-393.

Koren, Y. (2008). Factorization meets the neighbourhood: A Multifaceted Collaborative Filtering Model. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 426-434).

Kosala, R. & Blockeel, H. (2000). Web mining research: A survey. *ACM SIGKDD Explorations Newsletter, 2*(1), 1-15.

Lang, K. (1995). NewsWeeder: Learning to filter netnews. *In Proceedings of the 12th International Conference on Machine Learning*. San Francisco, CA, USA.

Leavitt, N. (2006). Recommendation technology: Will it boost E-Commerce?. *IEEE Computer Society*, *39*(5), 13-15.

Lee, T.Q., Park, Y. & Park, Y.T. (2008). A time-based approach to effective recommender systems using implicit feedback. *Expert System with Applications*, *34*, 3055-3062.

Leung, C. W., Chan, S. C. & Chung, F. (2007). Applying Cross-Level Association Rule Mining to cold-start recommendations. In *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops* (pp. 133-136). Silicon Valley, Canada.

Li, Y., Hu, J., Zhai, C. & Chen, Y. (2010). Improving One-Class Collaborative Filtering by incorporating rich user information. *In Proceeding of the 19th ACM international conference on Information and knowledge management* (pp. 959 - 968). ACM, New York, USA.

Linden, G. Smith, B. & York, J. (2003). Amazon.com recommendations item-to-item collaborative filtering. *Internet Computing, IEEE*, *7*(1), 76-80.

Liu, B. (2007). Web data mining: exploring hyperlinks, contents, and usage data. *ACM SIGKDD Exploration Newsletter, 10* (2). ACM, New York: Springer.

Lv, Y. & Zhai, C. (2010). Positional relevance model for pseudo-relevance feedback. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (pp. 579-586). ACM, New York, USA.

Ma, L., Chen, L., Goa, Y. & Yang, Y. (2009). Ontology based query expansion in Vertical Search Engine. In *6th International Conference on Fuzzy Systems and Knowledge Discovery* (pp. 285-289). IEEE Computer Society, Los Alamitos, CA.

Madria, S. K., Bhowmick, S., Ng, W. K. & Lim, E. P. (1999). Research issues in web data mining. In *Proceedings of the First International Conference*, *LNCS* (Vol. 1676, pp. 805-814). Florence, Italy: Springer Berlin/Heidelberg.

Martinez, L., Rodriguez, R. M. & Espinilla, M. (2009). REJA: A Georeferenced Hybrid Recommender System for Restaurants. In *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies* (pp. 187-190). Milan, Italy.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D. & Miller, K. (1990). Introduction to WordNet: An Online Lexical database. *International Journal of Lexicography (Special Issue)*, *3*(4), 235- 312.

Min, S. H. & Han, I. (2005). Recommender systems using Support Vector Machines. In *Proceeding of the 5th International Conference on Web Engineering, Lecture Notes in Computer Science* (Vol. 3579, pp. 387-393). Sydney, Australia: Springer.

Mirzadeh, N., Ricci, F. & Bansal, M. (2005). Feature selection methods for conversational recommender systems. In *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service* (pp. 772-777). ITC, Trento, Italy: IEEE.

Mobasher, B., Cooley, R. & Srivastava, J. (2000). Automatic personalization based on web usage mining. *Communications of the ACM, 43*(8), 142-151.

Mohan, B. K., Keller, B., J. & Ramakrishnan, N. (2007). Scouts, promoters, and connectors: The roles of ratings in nearest-neighbor collaborative filtering. *ACM Transactions on the Web. 1*(2), Article no. 8.

Montague, M. & Aslam, J. A. (2002). Condorcet fusion for improved retrieval. In *Proceedings of the ACM Conference on Information and Knowledge Management* (pp. 538-548). McLean, Virginia, USA:ACM.

Mooney, R. J. & Roy, L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries* (pp. 195-204). San Antonio, Texas, United States: ACM.

Mu, X. & Lu, K. (2010). Towards effective genomic information retrieval, the impact of query complexity and expansion strategies. *Journal of Information Science, 36*(2), 194-208.

Ogilvie, P., Voorhees, E. & Callan, J. (2009). On the number of terms used in automatic query expansion. *Information Retrieval. 12*(6), 666-679.

Ohrn, A. & Komorowski, J.(1997). ROSETTA: A rough set toolkit for analysis of data. In *Proceedings of the Third International Joint Conference on Information Sciences* (Vol. 3, pp. 403–407). Durham, NC.

Okabe, M. & Yamada, S. (2007). Semisupervised query expansion with minimal feedback. *IEEE Transactions on Knowledge and Data Engineering. 19*(11), 1585 – 1589.

Pavlov, D. Y. & Pennock, D. M. (2002). A Maximum Entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. In *Proceedings of 16th Annual Conference on Neural Information Processing Systems* (pp. 1441-1448). Vancouver, British Columbia, Canada: MIT Press.

Pawlak, Z., Polkowski, L. & Skowron, A. (2005). Rough Sets: An approach to vagueness. In the *Encyclopedia of Database Technologies and Applications* (pp. 575-580). Hershey, PA, Idea Group Inc.

Pazzani, M. & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, *27*(3), 313-331.

Pazzani, M. J., Muramatsu, J. & Billsus, D. (1996). Syskill & Webert: Identifying interesting web sites. In *Proceeding of The AAAI Spring Symposium on Machine Learning in Information Access* (pp. 54-61). Portland, Oregon: AAAI Press.

Pierrakos, D., Paliouras, G., Papatheodorou, C. & Spyropoulos, C. D. (2003). Web usage mining as a tool for personalization: A survey. *User Modeling and User-Adapted Interaction, 13*(4), 311-372.

Popescu, A.-M. & Etzioni, O. (2005). Extracting product features and opinions from reviews. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 9-28). London: Springer.

Popescul, A., Ungar, L. H., Pennock, D. M. & Lawrence, S. (2001). Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence* (pp. 437-444). Seattle, WA: Morgan Kaufmann Publishers Inc.

Prasad, B. (2007). A knowledge-based product recommendation system for e-commerce. *International Journal of Intelligent Information and Database system, 1*(1), 18-36.

Puntheeranurak, S. & Tsuji, H. (2007). A Multi-clustering Hybrid Recommender System. In *Proceedings of the 7th IEEE International Conference on Computer and Information Technology* (pp. 223-228). Aizu-Wakamatsu, Fukushima.

Resnick, P., Lacovou, N., Suchak, M., Bergstrom, P. & Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (pp. 175-186). Chapel Hill, North Carolina, United States: ACM.

Ricci, F. & Quang Nhat, N. (2007). Acquiring and revising preferences in a critique-based mobile recommender system. *IEEE Intelligent Systems, 22*(3), 22-29.

Ricci, F. & Wietsma, R. T. A. (2006). Product reviews in travel decision making. In *Proceedings of Information and Communication Technologies in Tourism 2006* (pp. 296-307). Lausanne, Switzerland: Springer.

Rojsattarat, E. & Soonthornphisaj, N. (2003). Hybrid recommendation: Combining content-based prediction and collaborative filtering. In J. Liu et al. (Eds.), *Data Engineering and Automated Learning* (pp. 337- 344). Springer Berlin/ Heidelberg.

Ruthven, I. (2008). Interactive information retrieval. *Annual Review of Information Science and Technology, 42*(1), 43-91.

Ruthven, I., Tombros, A. & Jose, J. M. (2001). A study on the use of summaries and summary based query expansion for a question answering task. *In 23rd BCS European annual colloquium on IR research* (pp. 1-14).

Salton, G. & Buckley, C. (1990). Improving retrieval performance by Relevance Feedback. *Journal of American Society for Information Science*, *41*(4), 288-297.

Sandvig, J. J. , Mobasher, B. & Burke, R. (2007). Robustness of collaborative recommendation based on association rule mining. In *Proceedings of the 2007 ACM Conference on Recommender Systems* (pp. 105-112). Minneapolis, MN, USA.

Schafer, J., Frankowski, D., Herlocker, J. & Sen, S. (2007). Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa & W. Nejdl (Eds.). *The Adaptive Web: Methods and Strategy for Web Recommendation: Lecture Notes in Computer Science* (Vol. 4321, pp. 291-324). New York: Springer Berlin/ Heidelberg.

Schafer, J. B., Konstan, J. & Riedl, J. (1999). Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce* (pp. 158-166). Denver, Colorado, United States: ACM.

Schafer, J. B., Konstan, J. & Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, *5*(1-2), 115-153.

Schwab, I., Pohl, W. & Koychev, I. (2000). Learning to recommend from positive evidence. In *Proceedings of the 5th international conference on Intelligent user interfaces* (pp. 241-247). New Orleans, Louisiana, United States: ACM.

Shani, G., Heckerman, D., & Brafman, R. I. (2005). An MDP-Based recommender system. *Journal of Machine Learning Research*, *6*, 1265-1295.

Shardanand, U. & Maes, P. (1995). Social information filtering: algorithms for automating word of mouth. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 210-217). Denver, Colorado, United States: ACM Press/Addison-Wesley Publishing Co.

Shepitsen, A., Gemmell, J., Mobasher, B. & Burke, R. (2008). Personalized recommendation is social tagging systems using Hierarchical Clustering. In *Proceedings of the 2008 ACM Conference on Recommender System* (pp. 259-266).

Si, L. & Callan J. (2003). A semisupervised learning method to merge search engine results, *ACM Transactions on Information Systems*, *21*(4), 457–491.

Singhal, A. (2001). Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 35 - 43.

Smyth, B., McGinty, L., Reilly, J., & McCarthy, K. (2004). Compound critiques for conversational recommender systems. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 145-151). Beijing, China.

Thompson, C. A., Goker, M. H., & Langley, P. (2004). A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, *21*, 393-428.

Tran, T. (2006). Designing recommender systems for e-commerce: an integration approach. In *Proceedings of the 8th international conference on Electronic commerce* (pp. 512-518). Fredericton, New Brunswick, Canada: ACM.

Van Rijsbergen, C. J. (1979). Information Retrieval. London, Butterworth.

Vucetic, S. & Obradovic, Z. (2005). Collaborative filtering using a Regression-Based approach. *Journal of Knowledge and Information Systems*, *7*(1), 1-22.

Walczak, B. & Massart, D.L. (1999). Tutorial Rough Sets theory. *Journal of Chemometrics and Intelligent Laboratory Systems, 47* (1), 1-16.

Wang, S. & Hauskrecht, M. (2010). Effective query expansion with the resistance distance based term similarity metric. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval.* ACM, New York, USA.

Wei, K., Huang, J., & Fu, S. (2007). A survey of e-commerce recommender systems, In *Proceedings of the Service Systems and Service Management* (pp. 1-5). Beijing: IEEE.

Wietsma, R., & Ricci, F. (2005). Product reviews in mobile decision aid systems. In *Proceedings of the Workshop Pervasive Mobile Interaction Devices* (pp. 15-18). LMU, Munich.

Xu, B., Zhang, M., Pan, Z., Yang, H., Gervasi, O., Gavrilova, M., et al. (2005), Content-based recommendation in e-commerce. In *Proceeding of International Conference on Computational Science and Its Applications, Lecture Notes in Computer Science* (Vol 3481, pp. 946-955*)*. Singapore: Springer Berlin/ Heidelberg.

Xu, J. & Croft, B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 4-11). ACM, New York, USA.

Xu, Y., Jones, G. J.F., & Wang, B. (2009). Query Dependent Pseudo-Relevance Feedback based on Wikipidea. In *Proceedings of the 32nd international ACM SIGIR*

conference on Research and development in information retrieval (pp. 59-66). ACM, New York, NY, USA.

Yamout, F., Oakes, M. & Tait, J. (2007). Relevance feedback using weight propagation compared with information-theoretic query expansion. In Amati, G., Claudio, C. & Romano, G.(Eds). *Advance in Information Retrieval (Vol.* 4425, pp. 258-270). Berlin, Heidelberg: Springer.

Zanker, M. (2008). A collaborative constraint-based meta-level recommender. In *Proceedings of the 2008 ACM conference on Recommender systems* (pp. 139-146), Lausanne, Switzerland: ACM.

Zhang, Y., Callan, J. & Minka, T. (2002). Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 81-88): Tampere, Finland: ACM.

Zhu, Z. & Balaji, V. (2006). Utility scoring of product reviews. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management* (pp. 51-57). Arlington, Virginia, USA: ACM.

Zhu, X. & Gauch, S. (2000). Incorporating quality metrics in centralized/distributed information retrieval on the World Wide Web. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 288-295). New York, USA: ACM.

Zhuang, L., Jing, F. & Zhu, X.-Y. (2006). Movie review mining and summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management* (pp. 43-50). Arlington, Virginia, USA: ACM.