CONTENT-BASED RECOMMENDER SYSTEM FOR AN ACADEMIC SOCIAL NETWORK

VALA ALI ROHANI

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN SOFTWARE ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

UNIVERSITY OF MALAYA

KUALA LUMPUR

2014

UNIVERSITY MALAYA

ORIGINALITY LITERARY WORK DECLARATION

Name of Candidate: Vala Ali Rohani Passport No: X95385868

Registration/Matric No: WHA090022

Name of Degree: Doctor of Philosophy

Title of Thesis: CONTENT-BASED RECOMMENDER SYSTEM FOR AN ACADEMIC SOCIAL NETWORK

Field of Study: Collaborative Computing (Computer Science)

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of my work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract form, or references to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed and copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date

Subscribed and solemnly declared before.

Witness's Signature Date
Name:

Designation:

Abstract

The rapid growth of Web 2.0 applications, such as blogs and social networks creates rich online information and provides various new sources of knowledge. The situation, however, leads to a great challenge in terms of information overload among social network users. Recommender systems (RSs) alleviate this problem with a technique that suggests relevant information from the abundance of Web data by considering the user's previous preference. Collaborative and content-based are the recommendation techniques typically used in existing RSs. The content-based method is employed more widely though. Similar to the collaborative, the content-based technique suffers from the cold-start dilemma that is caused by the incapability of RSs to make reliable recommendations in situations when new items or new users are involved. Such issues have an impact on prediction accuracy in existing algorithms, and hence, a better approach is required. In this study, a new algorithm is proposed to represent an enhanced version of content-based recommender systems by utilizing social networking features. In its formulation, the algorithm considers the interests and preferences of users' friends and faculty mates in addition to users' own preferences. The algorithm exploits all interests and preferences in a hierarchy tree structure. Since no offline data on Academic Social Networks (ASNs) exists and concerning the advantages of online study benefits, a real runtime environment of ASN called MyExpert was built in order to conduct an online study to assess the four recommender algorithms. Each recommender system algorithm, including the enhanced version of the content-based recommender systems using social networking (ECSN), is later incorporated into MyExpert to propose to members of this online society the most relevant academic items including jobs, news, scholarships and conferences. By using MyExpert, the online study was carried out to collect real feedback from live interactions between users and the system. The assessment ran for 14 consecutive weeks from 7th September to 26th December, 2012. MyExpert had 920 members from 10 universities in Malaysia at the time of evaluation. Four metrics, namely precision, recall, fallout, and F1 were employed to measure the prediction accuracy of each algorithm. Although the experiment conducted presented some threats, the results indicated that the ECSN algorithm not only improves the prediction accuracy of recommendations but also resolves the cold start problem in the existing recommender systems algorithms.

Dedication

To my beloved late father
Who taught me that the truth is everything in life

To my beloved mother

In the hope that one of her wishes will come true with the accomplishment of this work

Acknowledgements

First and foremost, I offer my sincerest gratitude to Allah the Almighty for bestowing me with the opportunity and good health to complete my dissertation successfully. I would also like to thank my supervisor, Associate Professor Dr. Zarinah Mohd Kasirun, for help, advice, and mentoring throughout my graduate career. Without her support and guidance, none of the work presented in this thesis would have been possible. Moreover, I am deeply indebted to her for showing me how to successfully do research, mentor students, and communicate the research results effectively.

I am extremely grateful for the constant support from the University of Malaya Bright Spark Program, especially Dr. Sharifuddin Md Zain for his kind assistance. Dr. Abrizah and Dr. Diljit Singh, I will never forget your valuable help and advice in dealing with any difficulty during my study. My deepest gratitude goes to all staff of the faculty of Computer Science and Information Technology at University of Malaya as well. I have also had this great opportunity to meet a lot of encouraging and kind friends during my PhD studies. Dear Ghazaleh, your kind efforts in submitting online news in MyExpert during this research experiments were really helpful. Thank you so much for your support.

Last but not least, I wish to express my love and gratitude to my beloved family, and not to forget my angel, Yasamin. I just want you to know that you are everything for me. Yes darling, finally your father completed his PhD studies. After this, Inshallah, I will dedicate more time to you. Thank you.

March, 2014.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AC	:	Academia Website
C#	:	Microsoft C# Programming Language
CB	:	Campus Buddy Website
CL	:	Classmates Website
CN	:	Course Networking Website
DFD	:	Data Flow Diagram
DG	:	Digication Website
DSS	:	Decision Support Systems
ECSN	:	Enhanced Content-Based Recommender System using Social Networking
ERD	:	Entity Relationship Diagram
LK	:	LinkedIn Website
MD	:	Mendeley Website
MS SQL	:	Microsoft SQL Server
MyExpert	:	Malaysian Experts Academic Social Network
RG	:	Research Gate Website
RS	:	Recommender Systems
RSS	:	Rich Site Summary

CHAPTER 1

INTRODUCTION OF THE STUDY

1.1 Research Background

In recent years, various types of social media sites have been created to provide users with a wide range of online services. Some such websites focus on sharing photos and tubes like Flickr[1] and YouTube[2], while others are classified in the blog and wiki categories that include Blogger[3] and Wikipedia[4]. Social networks, for instance Facebook[5] and MySpace[6], have greatly succeeded in attracting internet users from all over the world. Twitter[7], a micro-blogging site, has also proven precious for its users. The rapid progression of websites with online services and vast variety of information available on them, have led to novel difficulties for users regarding access to the most relevant information through daily transactions. Users browsing websites expect to conveniently find highly relevant information based on their needs, preferences and interests. Considering this information overload phenomenon, recommender systems are now vital features in online environments. They address this issue by recommending the most relevant items according to user preferences and previous interactions with the system (Afzal & Maurer, 2011; Ricci, Rokach, & Shapira, 2011).

In light of the dramatic growth of online social networks, novel ways of communication and collaboration have emerged over the last years. More than a billion users worldwide

---

[1] http://www.Flickr.com
[2] http://www.YouTube.com
[3] http://www.Blogger.com
[4] http://www.Wikipedia.com
[5] http://www.facebook.com
[6] http://www.MySpace.com
[7] https//www.Twitter.com

utilize these for networking and accessing information and updates according to their daily needs and preferences (Cheung, Chiu, & Lee, 2011).

In this regard, the main goal of Social Recommender Systems is to mitigate the information overload experienced by social media users by recommending the most pertinent and attractive content. They are also aimed at raising the level of user engagement, adoption and participation through social media websites (Ido Guy & Carmel, 2011). Recommending content (Ido Guy, Zwerdling, Ronen, Carmel, & Uziel, 2010), tags (Lipczak, Sigurbjörnsson, & Jaimes, 2012), people (Kim, et al., 2012), and communities (Tchuente, Canut, Baptiste-Jessel, Peninou, & Sedes, 2012) usually call for personalization techniques based on the interests and preferences of a given user or group of users. Consequently, recommender systems may play a significant role in how social networks successfully ensure that their users receive suggestions with the most relevant and attractive content based on their preferences and interests (Ido Guy & Carmel, 2011).

1.2 Problem Statement

A number of more specific social networks have developed in the last decade, catering to the special needs of members. Examples of academic social networks are Academia[1], Mendely[2], LinkedIn[3], and Course Networking[4]. Members of academic social networks are interested in locating different types of scientific information via this online environment. Thus, considering the vast capacity of information in the online environments, automatic suggestions by embedded recommender systems are preferred

---

[1] http://academia.edu
[2] http://www.mendeley.com
[3] http://www.LinkedIn.com
[4] http://www.thecn.com

in contrast to self-searching for academic items of interest (de Oliveira, Lopes, & Moro, 2011; He & Chu, 2010). One means of achieving this convenience is to select the most relevant information and send it to each member in the academic social network through an academic e-newsletter (Jheng, 2011). The selection process may be accomplished by utilizing a recommender system. Higher recommender algorithm performance facilitates the suggestion of the more relevant items to target users.

According to previous research works, two paradigms - collaborative and content-based filtering - are recognized as the most prevalent methods applied in the context of recommender systems (Basilico & Hofmann, 2004; Jannach, Zanker, Felfernig, & Friedrich, 2010; Ricci, et al., 2011). Collaborative techniques recommend items to a given user that other similar users have preferred in the past, while the content-based recommender systems suggest items similar to ones the same user was keen on in previous interactions with the system. In other words, content-based methods recommend an item to a user by matching the item's characteristics with the user's preference profile (Pazzani & Billsus, 2007; Ricci, et al., 2011).

Both collaborative and content-based recommender systems have a shortcoming related with cold start (Ricci, et al., 2011). For optimal recommendations, collaborative algorithms require strict records of previous item ratings. However, in such domains new items exist with no previous rating records, and collaborative methods cannot function properly. This kind of cold start problem occurs in such conditions when new items are supposed to be recommended (Adomavicius & Tuzhilin, 2005). This issue has been mitigated to some extent by content-based recommender systems, which can predict item relevance even in the absence of prior ratings. Nevertheless, even content-based recommender systems suffer from a related version of cold start. They are unable to recommend items to new users in the absence of any history of previous interactions with the system (Schafer, Frankowski, Herlocker, & Sen, 2007). Hence, the pure content-based method poses what is known as a cold-start problem concerning recommendations to new users. For this reason, an enhanced version of the content-based algorithm is proposed in the current research, whereby social networking techniques are utilized to not only solve the cold-start problem, but to also improve the prediction accuracy of the recommendation process.

The defining attribute of the Internet today is the abundance of information and choice. As Bonhard, Sasse, and Harries (2007) pointed out in their research, recommender systems designed to alleviate this problem, have so far not been very successful, especially in social networking domains. It has been argued that recommender algorithms could be significantly improved by drawing on features from social systems (Bonhard, Harries, McCarthy, & Sasse, 2006). According to (Ma, Zhou, Liu, Lyu, & King, 2011), although recommender systems have been briefly studied in the past decade, the study of social-based recommender systems has only just started. As stated in their research, until now the majority of recommender algorithms have focused on enhancing the performance of the recommendation process without considering the social elements of decision making and advice seeking (Ma et al., 2011). More specifically, traditional recommender systems ignore social relationships among users. In real life, for instance, when we ask friends to recommend a nice restaurant, we are actually requesting verbal social recommendations (Bonhard, 2005). In another research in this context, Bonhard, Sasse, and Harries (2007) affirmed that recommender systems and social networking functionality should be integrated. Hence, in order to improve recommender systems and to provide more personalized recommendation results, the incorporation of social network information among users is necessary (Zhou, Xu, Li, Josang, & Cox, 2012). Consequently, by utilizing social networking techniques, an enhanced version of content-based recommender systems would be generated which take advantage of social network-based factors to improve the performance of the recommendation process.

1.3 Research objectives

Four objectives are taken into account in this research as follows:

1) To compare the existing techniques of recommender systems and elicit essential features of academic social networks

2) To propose an enhanced content-based recommender system using social networking techniques (ECSN)

3) To develop an academic social network as a real runtime environment for evaluating recommender algorithms

4) To evaluate the ECSN recommender system by comparing its prediction accuracy with random, collaborative and content-based recommender algorithms

1.4 Research Methodology

A suitable, well-structured method is essential to performing sound empirical research. Empirical research methods are a class of research methods in which empirical observations or data are collected in order to test a theory (Easterly & Levine, 2001). In the present study, a quantitative method is used to test the theory of whether utilizing social networking parameters can improve the performance of content-based recommender systems in academic social networks. Figure 1.1 illustrates the three-phase method applied in this research.

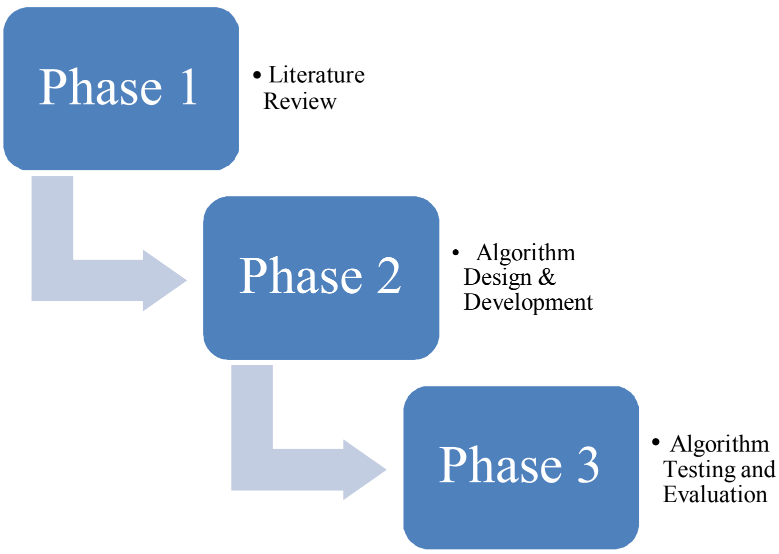


Figure 1.1: Abstract View of Research Methodology

The first phase focuses on a literature review and study of preceding research works done on recommender systems to identify the main characteristics of recommender algorithms. In this context, three different paradigms are studied: academic social networks, recommender algorithms, and recommender system evaluation methods. The description of these concepts is presented in Chapter 2.

The core focus of the second research phase centers on the design and development of three existing recommender algorithms (random, collaborative, and content-based), ECSN recommender system, and the required test environment for evaluating their performance. An ECSN recommender algorithm is designed and developed to enhance the pure content-based recommender system through social networking parameters. In this phase, MyExpert is also developed to provide an online real environment for testing the various recommender systems and comparing their accuracy in the recommendation process.

According to previous studies (Shani & Gunawardana, 2011) three methods of evaluating recommender systems exist, namely offline experiments, user studies, and online evaluation. The majority of research works in the domain of recommender systems employ the offline analysis method because it is economical and simple to implement (Adomavicius & Tuzhilin, 2005; Herlocker, Konstan, Terveen, & Riedl, 2004). In offline studies, offline datasets such as MoveiLens and GroupLens were used to test the recommender algorithms (Chen, Harper, Konstan, & Xin Li, 2010; Jung, 2011; Nguyen & Dinh, 2012; Sarwar et al., 1998). Other researchers preferred the user studies method for model evaluation in the field of recommender systems (Bambini, Cremonesi, & Turrin, 2011; Ge et al., 2010). Although the online studies is the most expensive compared to the other two and takes much more effort to implement, there are two important reasons why it has been selected for the experimental design of this research. In the first place, there was no offline data set for academic social networks with information on user feedback regarding the recommended academic items. This research focuses on enhancing the recommending process of academic items through an e-Newsletter in academic social networks. Thus, a need arose for a real academic social network with real users who receive the recommended academic items via email. By having a real online academic social network, it was possible to record the real behavior of users in their interactions with various recommender systems. Secondly, to develop the recommender algorithm in this research and compare its results with other recommender algorithms, it was necessary to access the code behind an academic social network. Consequently, a decision was made to design and develop a real academic social network (MyExpert) to serve as a runtime environment for this study. By creating MyExpert, C# programming enabled the development of random, collaborative, content-based, and ECSN recommender systems as well as running them in a real runtime environment.

During the second stage of the construction phase, three recommender algorithms (random, collaborative, and content-based) are implemented prior to the design and development of the main recommender algorithm that is proposed in this research (ECSN). A concise explanation of this phase is given in Chapter 4.

The main objective of the third research phase is to test and evaluate all four implemented recommender systems and compare their prediction accuracy. To achieve this goal, 1390 records of academic items were submitted in MyExpert, including 346 academic jobs, 339 conferences, 355 scholarships, and 350 academic news articles. As a follow-up to the data gathering schedule of this research, each of the above-mentioned recommender systems was used to send the top 10 academic items to MyExpert members over 14 consecutive weeks from 7th September to 26th December 2012. After gathering the members' feedback from the 14 weeks, Precision, Recall, Fallout, and F1 assessed the prediction accuracy of all recommender algorithms applied. The details of the evaluation process and its results are presented in Chapter 5 of this dissertation.

1.5 Theoretical Framework

The theoretical framework of this study covers several research areas. The first is Information Retrieval (IR) theory, which provides the overall context for this study (Adomavicius and Tuzhilin, 2005). Since the 1940s the problem of information storage and retrieval has attracted increasing attention (Good et. al, 1965). An IR system locates information that is relevant to a user's query. It typically searches in collections of unstructured or semi-structured data (e.g. web pages, documents, images, video, etc.). The need for an IR system occurs when a collection reaches a size where traditional cataloguing techniques can no longer cope. Similar to Moore's law of continual processor speed increase, there has been a consistent doubling in digital storage capacity every two years. With the growth of digitized unstructured information and, via high speed networks, rapid global access to enormous quantities of that information, the only viable solution to finding relevant items from these large text databases was search, and IR systems became ubiquitous (Sanderson et. al, 2012).

Another theory in this context is Cognitive Science which is an interdisciplinary effort to uncover the relationships between brains, minds, and behavior (Rich, 1979). It is the study of how the brain, a biological organ, gives rise to the mind, a functional construct. In other words, it is an interdisciplinary field encompassing psychology, neuroscience, linguistics, computer science, and mathematics. Explanation is counted as a concept which makes a connection between cognitive science and recommender systems. Explanations provide us with a mechanism for handling errors that come with a recommendation. Consider how we as humans handle suggestions as they are given to us by other humans. We recognize that other humans are imperfect recommenders. In the process of deciding to accept a recommendation from a friend, we might consider the quality of previous recommendations by the friend or we may compare how that friend's general interests compare to ours in the domain of the suggestion. However, if there is any doubt, we will ask "why?" and let the friend explain their reasoning behind a suggestion. Then we can analyze the logic of the suggestion and determine for ourselves if the evidence is strong enough (Herlocker et. al, 2000).

RSs origins can also be traced back to forecasting theory (Armstrong, 2001). A decision maker must inevitably consider the future, and this requires forecasts of certain important variables. There also exist forecasters – such as scientists or statisticians – who may or may not be operating independently of a decision maker. In the classical situation, forecasts are produced by a single forecaster, and there are several potential users, namely the various decision makers. In other situations, each decision maker may have several different forecasts to choose between (Granger & Machina, 2006).

In the mid-1990's, recommender systems emerged as an independent research area when researchers started focusing on recommendation problems that explicitly rely on the ratings structure (Goldberg, Nichols, Oki, & Terry, 1992; McSherry & Mironov, 2009). In its most common formulation, the recommendation problem is reduced to the problem of estimating ratings for the items that have not been seen by a user. This estimation is usually based on the ratings given by this user to other items and possibly on some other information as well.

1.6 The Significance of the Study

The main contribution of this research is the improved performance of the content-based recommender algorithm. As such, an enhanced content-based recommender system based on social networking techniques (ECSN) is proposed. More specifically, the ECSBN algorithm makes more accurate predictions when recommending relevant items to members of an academic social network (MyExpert[1]) compared to the random, collaborative, and content-based recommender algorithms.

Besides, in conditions where no evidence exists of previous interaction between target users and the system, social networking techniques may be very useful in identifying and recommending the most relevant items. Studying the online behavior of friends and classmates of a given user would particularly enable making rather correct predictions about their preferences and interests. Basically, this approach solves the cold start problem of the collaborative and content-based recommender systems in situations where the recommender engine is faced with new items and new users.

The present research makes an additional contribution to the field of social networking. MyExpert, as developed in this study, is now the first academic social network in Malaysia. It acts as a real runtime environment for evaluating the accuracy of the ECSN recommender system and judging its performance against other recommender algorithms. MyExperts also provides an online environment for the collaboration and sharing of academic knowledge and documents among scientists in Malaysia. Currently, more than 920 students, lecturers and researchers from 10 universities in Malaysia have joined this academic social network.

There are over 100,000 log records of transactions in MyExpert that also have the potential to be used in future research works in the recommender systems field.

[1] http://www.MyExpert.com

Numerous other studies, such as anomaly detection and knowledge-based recommender systems can draw on MyExpert datasets for research work assessment.

1.7 Thesis Organization

This dissertation is organized as follows. Subsequent to the introduction in Chapter 1, Chapter 2 explores literature on the concepts of academic social networks and recommender systems. In the first part, the reasons why academic social networks have emerged are described and the most popular examples are discussed. The second section of the literature review focuses on recommender system description. After providing some information with respect to their functionalities, the three main techniques in recommender systems are discussed briefly – the collaborative, content-based, and hybrid approaches. The final part of the chapter presents the methods for evaluating the prediction accuracy of recommender systems. In Chapter 3, the research methodology is discussed along with the target population used in this work, data collection procedure, and applied evaluation method. Chapter 4 illustrates the technical issues with this research. Two main sections present the design and implementation details of MyExpert as a runtime environment, and the recommender algorithms developed in this study. The experiment and study results are discussed in Chapter 5. Prior to the experimental design, however, the means of evaluating the recommender systems are introduced. Then the four measurements done (precision, recall, fallout, and F1), together with the results are discussed. In conclusion, Chapter 6 describes how this study contributes to current academic literature on this topic. Future works are also suggested in the final chapter. This dissertation concludes with supporting documents and data in the appendices.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the sections that follow, recommender systems and academic social networks are discussed. Since the main focus of this research is improving the performance of recommender algorithms, it is additionally essential to study the most popular recommender systems, their techniques, advantages, shortcomings along with methods of evaluating their prediction accuracy. As mentioned in the research methodology section of the previous chapter, the MyExpert academic social network is considered as runtime environment in this research for applying and evaluating recommender systems. Hence, to design and develop MyExpert, a comprehensive review of current academic social networks is first required.

2.2 Recommender Systems

"We have 6.2 million customers; we should have 6.2 million stores. There should be the optimum store for each and every customer."

-Jeff Bezos, CEO of Amazon.com™ (Bezos, March 16th 1999)

Recommender Systems (RS) are software tools and techniques for suggesting the most related items to users (Burke, 2007; Mahmood & Ricci, 2009). These suggestions may be in terms of different usages, such as books to buy, people who can be selected as friends in a social network, or online news to be read. In other words, they provide personalized recommendation based on information elicited from user profiles and item specifications (Cacheda, Carneiro, Fernández, & Formoso, 2011).

The dramatic growth of online services and the vast variety of information available on the Web have led to a number of serious difficulties for users in making correct decisions through their online transactions. Considering the phenomenon of information overload, RSs have proved to be effective and essential facilities in online environments. They address this problem by recommending new and not-yet-experienced options to users according to their current needs. To provide these coherent suggestions, RSs need various types of information on user preferences, lists of available items, and history of previous interactions between the user and the system stored in designated databases. When browsing the recommended items, the user can then provide implicit or explicit feedback that will be used to generate more relevant recommendations in future suggestions (Ricci, et al., 2011).

Pertaining to RSs, an "item" is a typical term for what is supposed to be recommended to a target user. With respect to the nature and characteristics of both "items" and "user" is crucial to implement an optimum RS in related domains. In some websites, such as Amazon.com, the RS is used for personalizing the online store for each client (Hwang, Kuo, & Yu, 2008). Ricci, et al. (2011) pointed out that in personalized RSs, users with different interests and preferences receive diverse recommendations. Besides, there are much simpler types of RSs that are non-personalized. Online magazines and newspapers usually prefer to use this kind of RSs due to their simplicity in implementation. For example they can help suggest the top ten books or CDs in some online stores to all users. While these types of non-personalized recommendations may be effective in various general instances, they are not typically addressed in RS research fields. In personalized RSs, the recommended items are presented as a ranked list. Predicting the most suitable items for users is based on user taste, preference and constraints. To accomplish this computational procedure, RSs need to generate a user profile by collecting their preferences. This eliciting process can be either explicit, e.g., by rating the products, or implicit, e.g., by analyzing user interactions. For instance, the sequence of browsing the web pages in an online store by each customer can be deemed an implicit sign of interest for items shown on visited web pages.

In a review of recommender systems, Adomavicius and Tuzhilin (2005) pointed out that RSs origins can be traced back to information retrieval (Salton, 1989), cognitive science (Rich, 1979), forecasting theories (Armstrong, 2001), approximation theory (Powell, 1981), marketing (Lilien, Kotler, & Moorthy, 1992), as well as management (Murthi & Sarkar, 2003). But more specifically, the concept of recommender systems has emerged as a novel research area in the mid-1990s when the rating structure of RSs became a center of attention for researchers in this field (Goldberg, Nichols, Oki, & Terry, 1992; McSherry & Mironov, 2009). It can be said that when user profiles came into existence as a new concept of making recommendations, previous techniques in information retrieval were improved to generate a new research field known as recommender systems. As the following facts indicate, in recent years, there has been an increasing interest in recommender systems:

1. It is becoming increasingly difficult to ignore the important role of RSs in such high-rank websites as YouTube, Yahoo, Amazon.com, Tripadvisor, Last.fm, and IMDb. Recommender systems have become one of the most essential services that the media companies are providing to their clients. For example Netflix, one of the most popular online movie rental websites, recently awarded a million dollar prize to programmers

for increasing the performance of its recommender system (Koren, Bell, & Volinsky, 2009).

2. In recent years, we can see several courses that are specifically dedicated to the RS research domain at universities and other academic centers around the world. Bedsides, workshops and tutorials on recommender systems have been very popular at computer science conferences, and also several books have recently been published on RS techniques (Jannach, et al., 2010; Tiroshi, Kuflik, Kay, & Kummerfeld, 2012).

3. Recent developments in conferences and workshops related to recommender systems have heightened the call for this research field. We can specifically mention ACM recommender systems (RecSys) established in 2007 and which have currently become among the most important annual events on recommender systems. Even in the more traditional conferences in the area of information systems, databases and adaptive systems, we frequently see the increasing growth in sessions and discussions related to RSs. ACM SIGIR Special Interest Group on Information Retrieval (SIGIR), Adaptation and Personalization (UMAP), User Modeling, and ACM's Special Interest Group on Management Of Data (SIGMOD) are some examples of such academic events.

4. There have been several calls for papers with special issues in academic journals dedicated to recommender systems research and development. Among these kinds of journals are International Journal of Web-Based Communities (2013); User Modeling and User-Adapted Interaction (2012); IEEE Intelligent Systems (2007); AI Communications (2008); International Journal of Electronic Commerce (2006); International Journal of Computer Science and Applications (2006); ACM Transactions

on Computer-Human Interaction (2005); and ACM Transactions on Information Systems (2004).

2.2.1 Recommender Systems Functionality

To expand the definition of RSs, in this section their different roles are illustrated on behalf of service providers and users. Ricci et al. (2011) list the reasons why service providers prefer to facilitate recommender systems as follows:

• **To increase the number of items sold:** Selling more items to customers is definitely one of the most important objectives for any business corporation. RSs can be effective in achieving this goal as they try to suggest items that are likely to suit the customers' needs and interests. Improving the conversion rate can be the main reason for a service provider to utilize RSs. In this way, they can increase the number of customers who accept the recommendation and find their desired items, compared to the number of simple users that just browse through the information.

• **To sell more diverse items:** Marketing, and ultimately selling new and unpopular items, is another functionality expected of a well-designed RS. What is interesting for an online merchant is to recommend items that might be hard for a customer to find using traditional information retrieval techniques.

• **To enhance user fidelity and satisfaction:** User loyalty can be majorly increased when they receive interesting and relevant recommended items by well-tuned RSs. Besides, the effectiveness and accuracy of suggested items presented in a user friendly interface leads to higher subjective system evaluation by customers. The users' previous transactions become widely utilized by RSs to compute the most relevant recommendations. Hence, the longer a user interacts with the system, the more coherent user preferences become elicited, and ultimately the more related items are recommended by the designated recommender system.

The previous issues are thus some important reasons why service providers are so interested in applying recommender systems. Considering the effective role of RSs, users may also find use for them. So it is essential to observe the requirements from the two sides and design a recommender system capable of providing appropriate suggestions for both. Herlocker (2004) identified eleven associated tasks that RSs implement concerning users' needs and expectations.

• **Find some good items:** Prepare a list of prioritized items and recommend them to target users. This prediction can be done by using previously elicited information from the user's preferences. This seems to be the most important task in many recommender systems in commercial environments.

• **Find all good items:** In some circumstances finding only some good items is not sufficient, and it is worth recommending all the items that cover user needs. Using this technique seems logical, especially when facing a rather small number of items or when the recommender system is used in a mission-critical environment, such as sensitive military or medical applications. In these scenarios, the user may take advantage of both suggestions of all related alternatives and an ordered list of items based on associated ranks.

• **Annotation in context:** A more specific duty of a RS is emphasizing items in a given existing context considering the user's long-term interactions with the system and his or her preferences.

• **Recommend a sequence:** In some special applications, the idea of recommending a sequence of correlated suggestions is preferred to merely suggesting a single item. Recommending to buy a data mining book after getting a suggestion on a database topic, suggesting a TV series, or providing a recommendation on a compilation of musical tracks are typical examples of this method in RSs (Hayes & Cunningham, 2001; Shani, Brafman, & Heckerman, 2002).

• **Recommend a bundle:** Here, the role of RSs is to recommend a package including different types of items that fit well together. In some cases, such travel plans, the item supposed to be suggested might consist of various parts like destinations, restaurants, attractions, and accommodation services located in a target area. In 2006, Ricci et al. demonstrated that users prefer to receive all these recommendations as a single travel plan.

• **Just browsing:** Sometimes web site users prefer to window shop with no intention of purchasing an item. Thus, helping the user find and browse the desired items is a task expected of RS to carry out. The adaptive hypermedia techniques can also support these kinds of transactions (Brusilovsky, 1996).

• **Find a credible recommender:** Other times, online users prefer to test a recommender system before trusting it. Hence, to assure them of the system's credibility, it is worth contriving a mechanism to allow users to fiddle with the RS and see how good it is in recommending the most related items.

• **Improve the profile:** Later in this chapter it is mentioned that users preferences can be elicited from their profiles in making better recommendations in content-based RSs. In supporting this idea, it is important to consider a capability for users to input and update their profiles.

• **Express self:** Some users mostly care about their contribution with their ratings and expressing their beliefs rather than receiving the recommendations. So in these special situations, increasing user satisfactions for this activity can be effective in improving user loyalty.

• **Help and influence others:** For a number of users in online communities, the notion that the system benefits from their contribution is an extremely good motivator that inspires them to contribute with information and rate items. Actually, their main interest is to help and influence others in finding items they are looking for.

2.2.2 Recommendation Techniques

The most important objective of recommender systems is to estimate the ratings for the items that have not been seen by a user (Adomavicius & Tuzhilin, 2005). Ultimately, after calculating the estimated rates for the yet unrated items, an ordered list of most related items can be prepared and suggested to the target user. Adomavicius and Tuzhilin (2005) represent the recommendation problem as follows: Consider C as the set of all users, and S as the set of all items supposed to be recommended to users. Both user and item space can be very large based on the application domain. Let u stand for a utility function that calculates the usefulness of item s to user:

$$u : C \times S \rightarrow R \tag{2.1}$$

In this relation, R is a totally ordered set with a non-negative integer or real numbers in a specific range. In this space, it is supposed to choose for each user $\in C$, such items $s' \in S$ that maximize u:

$$\forall c \in C,\ s \in S,\ s'_c = argmax\, u(c, s) \tag{2.2}$$

As such, each member of user space C can have a profile indicating the attributes of the user, such as age, gender, degree, etc. Similarly, the items have their own characteristics.

However, the utility function u is not defined on the whole $C \times S$ space. The main calculations in RSs based on ratings, and more specifically, it is limited to the previously rated items by the users. Consequently, the RS engine should predict the ratings of non-rated items and try to make recommendations based on this framework (Ricci, et al., 2011).

The process for predicting unknown ratings includes two main steps: 1) applying specific heuristics in the utility function and validating its performance in runtime environments, and 2) evaluating the performance of a designated utility function in optimizing the recommendation results. Ultimately, after accomplishing the estimation process, the highest rated items are recommended to users based on the applied algorithm in the previous step.

Different methods, such as machine learning, approximation theory, and various heuristics, can be used for recommending the most related items. Considering this variety, RSs are mainly classified based on their approach in rating the estimations.

In 2005, Adomavicius and Tuzhilin reviewed previous literature by Hill, Stead, et al. (1995); Rosenstein, & Furnas (1995); Resnick, et al. (1994); and Shardanand & Maes (1995), and classifed the recommender systems in three categories:

- **Collaborative recommendations:** The items are predicted based on the items that people with similar preferences and interests preferred previously;

- **Content-based recommendations:** The users' own preferences through previous interactions are considered in predicting the new items;

- **Hybrid approaches:** These methods are a combination of the collaborative and content-based methods.

In another more recent and highly cited research (Ricci, et al., 2011), three other approaches were added into the classification of recommender systems:

- **Demographic recommendations:** This approach follows the idea that users with variant demographic profiles should receive recommendations differently.

- **Knowledge-based recommendations:** In this type of recommender system, items are suggested based on specific domains of knowledge.

- **Social network-based recommendations:** The preferences and needs of the user's friends play the main role in this type of recommender system.

Next, each of the previously mentioned algorithms will be presented in more detail by revealing their advantages and shortcomings to be considered when applying them in runtime environments.

2.2.2.1 Collaborative Filtering

As Schafer et al. demonstrated in 2001, the original implementation of the Collaborative Filtering (CF) approach suggests to the active user items that similar users liked in the past. The similarity in the previous user ratings is considered for identifying the similarity in the preferences of two given users. That is why J. B. Schafer et al. (2001) referred to CF as "people-to-people correlation." The prediction process in CF systems is almost based on evaluation rather than analysis. In other words, this method categorizes information by considering the user's opinion on an item instead of the information itself. More specifically, the role of the community is highlighted in CF as it focuses on other similar users' opinions to suggest a particular item to a target user.

Hereby, we discuss the most widely used CF algorithms. A comparison study of CF algorithms was carried out by Cacheda et al. (2011), who argue that there are two main classifications for Collaborative Filtering recommender systems: memory-based (user-based) and model-based (item-based). These are applied in different application fields.

Earlier research works have been using the memory-based approach that utilizes elicited information from items previously rated by users. This method needs all items, ratings and users to be collected and stored in the memory to make recommendations. Later, to cover some shortcomings of this approach, a model-based method was developed that looks for similar items instead of making groups of similar users. In other words, it uses an offline pattern created periodically by summarizing item ratings. Both these methods will be discussed in more detail later, and their advantages and shortcomings will be compared. First of all, we need to become familiar with the User-Item matrix.

The concept of User-Item matrix is found in both memory-based and model-based collaborative filtering algorithms (Wang, De Vries, & Reinders, 2006). As shown in Figure 2.1, the user's profile can be considered a $K \times M$ user-item matrix X for K number of users and M number of items. Each element $x_{k,m} = r$ represents the value of rating that the user k has assigned to item m. For items rated $r \in \{1, \ldots, |r|\}$, and for unrated ones, $r = \emptyset$.

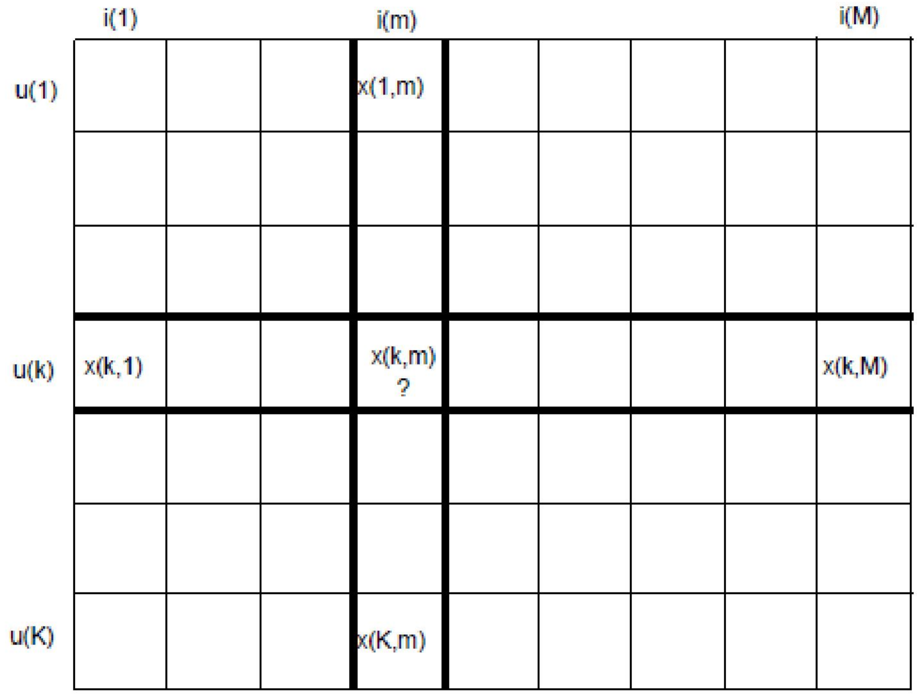


Figure 2.1: The User-Item Matrix in Recommender Systems

The user-item matrix can be manipulated in two different ways. For one, it can be represented by considering its row vectors:

$$X = [u_1, \dots, u_K], \quad u_k = [x_{k,1}, \dots, x_{k,M}], k = 1, \dots, K \qquad \textbf{(2.3)}$$

where each row vector u_k represents a user profile including all ratings assigned to the items. The memory-based collaborative filtering is based on this type of representation.

Second, it can be decomposed into column vectors:

$$X = [i_1, \dots, i_M], \quad i_m = [x_{1,m}, \dots, x_{K,m}], m = 1, \dots, M \qquad \textbf{(2.4)}$$

where each column vector i_m represents all ratings assigned to a specific item. This viewpoint leads to model-based collaborative filtering systems.

2.2.2.1.1 Memory-Based Collaborative Filtering

In the memory-based (user-based) approach, the recommender engine predicts the active user's interest for a specific item by considering the elicited information from similar users' profiles (Wang, et al., 2006).

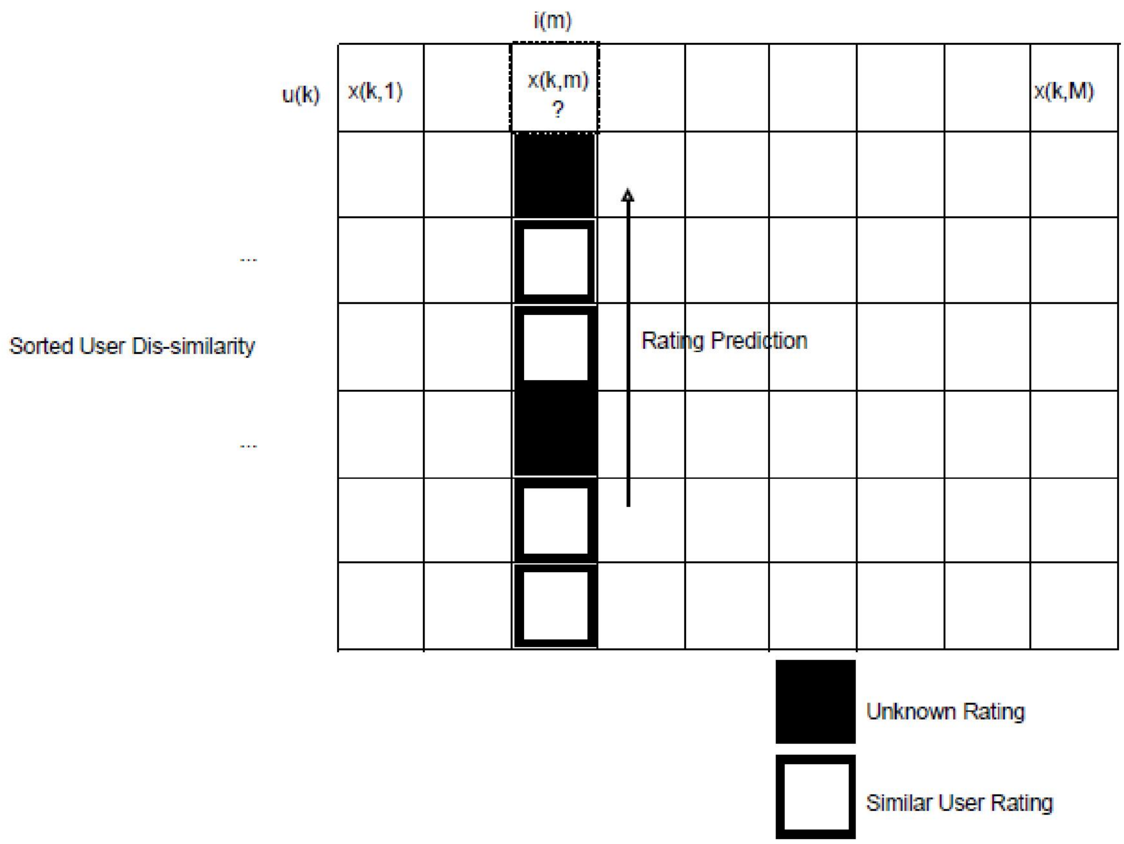


Figure 2.2: Using User Similarity to Predict the Ratings

As depicted in Figure 2.2, each row vector which represents a user profile has been sorted based on its dissimilarity towards the active user's profile. Hence, the items rated by more similar users have better chance of being recommended to the active user. A group of similar users can be generated by selecting top-N similar users $S_u(u_k)$ toward user:

$$S_u(u_k) = \{u_a | rank\ s_u(u_k, u_a) \leq N, x_{a,m} \neq \emptyset\} \quad \textbf{(2.5)}$$

where $|S_u(u_k)| = N$.

In this formula the degree of similarity between users k and a is identified by $s_u(u_k, u_a)$. According to Wang et al. (2006), the two most popular measures for calculating this kind of similarity in collaborative filtering are Cosine similarity and Pearson's correlation. Some training data can also be used to generate this ranked list of similar users (Jin, Chai, & Si, 2004). This study adopts the cosine similarity measure:

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}} \quad \textbf{(2.6)}$$

where a and b are the users whose similarity we want to measure. $r_{a,p}$ represents the rating assigned by user a to item p and P is the set of items rated by both user a and b. The last parameter is $\boldsymbol{\bar{r}_x}$, which denotes the average ratings submitted by user x. Finally, the measured similarity value will be between -1 and $+1$.

After calculating the similarity value between all pair of users, the following formula will be applied to calculate the predicted rating $pred(a, p)$ of item p by the user:

$$pred(a,p) = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \overline{r_b})}{\sum_{b \in N} sim(a,b)} \quad \textbf{(2.7)}$$

where N is the set of all users who submitted a rating to item p.

The simplicity and tangibility have made the memory-based collaborative filtering very popular in different application domains. Although they are sufficient to solve many problems in the real world, they still have shortcomings (Cacheda, et al., 2011):

Sparsity. In real-world applications, it is undeniable that even active users rate only a small subset of total items. In this situation, most of the cells in the user-item matrix remain empty. Consequently, a memory-based CF system which needs the data of the user-item matrix faces a serious problem in making a coherent recommendation. (Hofmann, 2004; Sarwar et al. 2001).

Cold Start. This problem specifically happens to new users who have not rated enough items yet. In such cases, the recommendation system cannot elicit the user's preferences and consequently, it is unable to predict the related items correctly for suggestion. To solve this problem, some RSs apply a method of forcing users to submit ratings for a minimum number of items. But even this solution has its own problems and leads to system biases. Similarly, the new items may also be affected by this kind of problem in memory-based CFs (Schein, Popescul, Ungar, & Pennock, 2002).

Shilling. Another difficulty with this method is derived from spam attacks by users who plan to mislead the RS into recommending specific items (Chirita, Nejdl, & Zamfir, 2005; S. K. Lam & Riedl, 2004). In this regard, several studies have been done in the past that point out some methods affecting both model-based (Sandvig, Mobasher, & Burke, 2008) and memory-based (Mobasher, Burke, Bhaumik, & Williams, 2007) algorithms.

Scalability. The majority of memory-based collaborative algorithms suffer from the scalability problem as they have to process large amounts of data for recommending a single item. Hence, in an online environment with a lot of items or users, these methods are not a suitable alternative, especially in real-time recommending systems (Sarwar et al. 2001).

The shortcomings with user-based collaborative systems mentioned have led to the emergence of another approach called item-based or model-based CF systems.

2.2.2.1.2 Model-Based Collaborative Filtering

The model-based CF approach takes advantage of the idea that compiling a model of user preferences may solve some of the difficulties of memory-based algorithms. This model includes precompiled information of items, users and ratings, and may be generated in several hours or days. The built model is then used for making recommendations. The main difference between this model and the previous one is considering the items instead of users to predict the ratings. As shown in Figure 2.3, the prediction can be made based on average ratings of similar items rated by the active user (Deshpande et al. 2004; Linden et al. 2003; Sarwar et al. 2001).

Similar to the memory-based method, in this algorithm sorting is done based on dissimilarity. Instead of sorting by row vectors, the items (column vectors) are sorted toward the target item. As shown in Figure 2.3, by applying this method of sorting, the rating value of items with higher degrees of similarity can be weighed stronger.

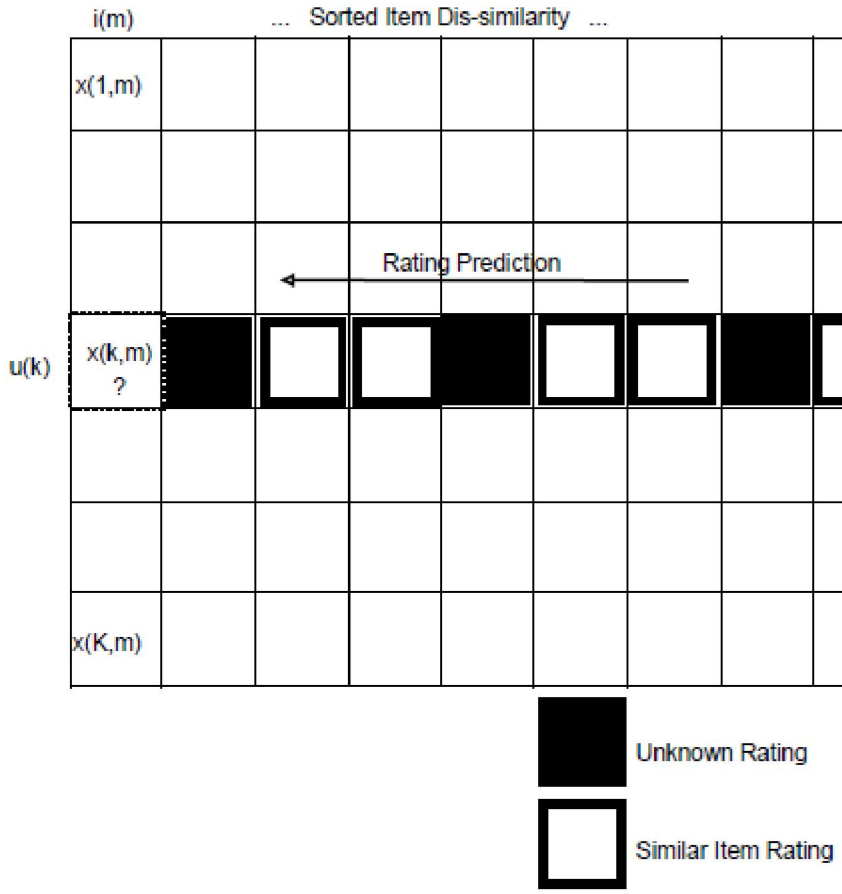


Figure 2.3: Using Item Similarity to Predict the Ratings

In item-based prediction algorithms predicting the most relevant items is based on the similarity between items:

$$\boldsymbol{sim(\vec{a},\vec{b})} = \frac{\sum_{u\in U}(r_{u,a}-\overline{r_u})(r_{u,b}-\overline{r_u})}{\sqrt{\sum_{u\in U}(r_{u,a}-\overline{r_u})^2}\sqrt{\sum_{u\in U}(r_{u,b}-\overline{r_u})^2}} \qquad \textbf{(2.8)}$$

where U indicates the set of all users who rated both item a and b. Accordingly, $\boldsymbol{r_{u,a}}$ and $\boldsymbol{r_{u,b}}$ are the rates that user u assigned to items a and b respectively.

Having similarity of available items, we can apply the following formula to predict the rating of active user u to item p:

$$\boldsymbol{pred(u,p)} = \frac{\sum_{\boldsymbol{i \in ratedItem(u)}} \boldsymbol{sim(i,p) * r_{u,i}}}{\sum_{\boldsymbol{i \in ratedItem(u)}} \boldsymbol{sim(i,p)}} \qquad \textbf{(2.9)}$$

where $sim(i,p)$ denotes the degree of similarity between each member of items rated by user u and the target item p , and $r_{u,i}$ represents the rating assigned by user u to item i.

In this context, Bayesian networks and Bayesian clustering are two methods evaluated in another research (Gong, 2010). The Bayesian clustering technique groups users with similar interests and tastes into the same class. The retrieved information by analyzing the data sets identifies the number of classes and parameters in the model. Each node in the Bayesian network model corresponds to an item in the data set. Also, the possible ratings of each item identify the state of each node.

In an early research study by Ungar & Foster (1998), clustering was applied as a pre-processing step for collaborative filtering systems which classify items and users into groups. In this method, the probability of interest to each group of items is calculated for each group of users.

As mentioned previously, data sparsity was one of the limitations in memory-based collaborative systems. Although the model-based methods deal with this to some extent, the need for tuning a large number of parameters causes serious restrictions to applying it practically.

We can count several advantages for model-based collaborative filtering methods compared to memory-based approaches. First, a model-based CF offers some added values. In addition to its predictive features, it can clarify the correlations in elicited data. Second, it needs less memory space for storing data. Third, taking advantage of the complied model, in the model-based approach the recommendations can be made very quickly. Consequently, the model-based CF systems are usually smaller, faster than, and definitely as coherent as memory-based methods. They are truly applicable in real world environments in which user profiles and interests change slowly and do not need to be updated frequently.

2.2.2.2 Content-Based Filtering

During the past thirty years, researchers have been utilizing technologies to automatically categorize information used for generating recommendations based on users' personal preferences (Herlocker, 2000). In other words, they analyzed previously rated items by a user to build a user model presenting his or her interests and preferences (Mladenic, 1999). To come up with a judgment representing the user's level of interest to a specific item, RSs try to match up the preferences retrieved from a user profile against the attributes of that item. Content-based recommendation systems have been used in a variety of domains ranging from recommending web pages, news articles, restaurants, television programs, and items for sale. Although the details of various systems differ, content-based recommendation systems share in common a means for describing the items that may be recommended, a means for creating a profile of the user that describes the types of items the user likes, and a means of comparing items to the user profile to determine what to recommend. In doing so, the profile is often created and updated automatically in response to feedback on the desirability of items that have been presented to the user (Pazzani et al. 2007). To clarify the adopted techniques in this approach, the architecture of content-based recommender systems is described in the following section.

2.2.2.2.1 Architecture of Content-Based Recommender Systems

Content-Based RSs need a well-structured framework supporting the techniques for comparing user interests with the items' specifications, and to ultimately suggest the most suitable item to a target user. Lops et al. (2011) proposed a high level architecture for content-based recommender systems as illustrated in Figure 2.4.

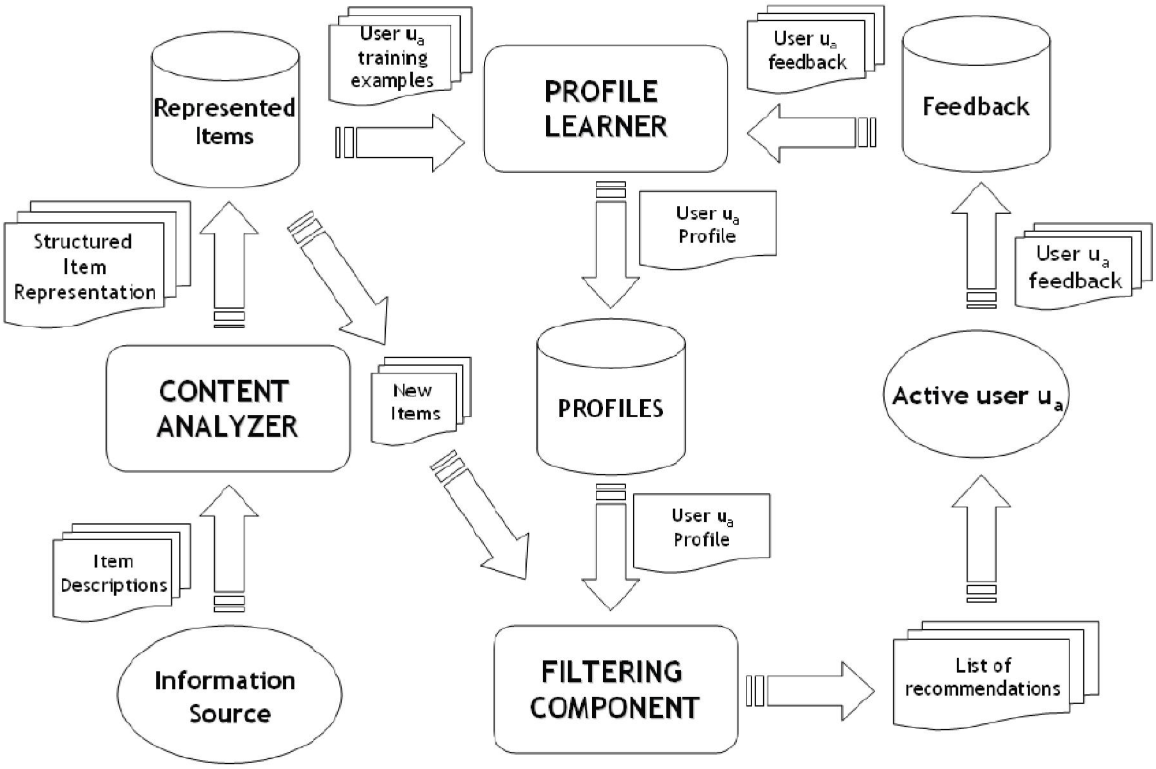


Figure 2.4: Content-Based Recommender High Level Architecture (Lops, et al., 2011)

As depicted above, this architecture has three main components:

• **CONTENT ANALYZER –**The main role of this component is to prepare the item's relevant information in a format suitable to the next steps. After applying the information retrieval techniques, the analyzed data items will be forwarded to the PROFILE LEARNER and FILTERING COMPONENT for the next level of processing.

• **PROFILE LEARNER –** This component is responsible for constructing the user profile. It usually generalizes the data related to user preferences and interests using machine learning techniques (Mitchell, 1997). For example, in a web page recommender system, the PROFILE LEARNER may utilize a relevance feedback mechanism which combines vectors of positive and negative samples of web pages to shape the user model.

• **FILTERING COMPONENT –** This component recommends the most relevant items by eliciting preferences from the user profile and comparing them with item attributes. As a result, a rank list of predicted interesting items is produced for the designated user.

More precisely, the CONTENT ANALYZER processes the item characteristics retrieved from Information Sources by applying information retrieval techniques (Chowdhury, 2010; Davenport, 2012). It produces a structured representation of items from originally unstructured formats.

As depicted in Figure 2.4, a storage called Feedback has been considered for collecting the reactions of active users (u_a) to the items. This collection is used to generate and update the profile of each u_a. Besides, the users can also provide information about their interests and preferences through their profiles explicitly. Usually the collected feedback is distinguished as two different kinds: positive information (items that the user is interested in) and negative information (items that are disliked by the user) (Holte & Yan, 1996).

Explicit and Implicit feedback are two methods that can be used in this context. In the former technique, the user is actively involved in evaluating the items, while in the latter the feedback is elicited from analyzing the user's interactions with the system in a specific duration. Three main techniques are applicable in getting feedback in the explicit method:

• **Like/Dislike –** applying a binary approach, items are rated as "relevant" or "not relevant, (Billsus & Pazzani, 1999);

• **Ratings –** pointed out by Shardanand and Maes (1995), a discrete numeric scale can be used to rate items. Alternatively, the numeric scales can be replaced by symbolic ratings such as by Syskill and Webert, where users could select hot, cold, or lukewarm when rating a web page (Pazzani et al. 1996).

• **Text Comments –** To assist the user in making decision, textual comments submitted to a specific item will be elicited and presented (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994). Amazon.com and eBay.com are two web sites which employ this method to clarify whether an item is inspired by other users in previous transactions. It is worth mentioning that applying some advanced techniques of effective computing can improve the content-based RSs' performance to carry out this process automatically (Picard, 2003).

While explicit feedback has the advantage of simple implementation, it may increase the load on the user and besides, it may not be coherent enough in exploiting the user's preferences of items. On the other hand, in implicit feedback techniques, a relevance score is assigned to a specific user action when faced with an item. These actions may be selected from a variant list of user interactions with the system such as clicking, printing, downloading, etc. Although biasing may occur in this method, it has the advantage that the user is not required to be involved directly in these rating transactions.

As mentioned previously, each active user u_a is assigned by a profile. To build this profile, the training set TR_a must be defined for each u_a.The TR_a is defined as a set of pairs (I_k, r_k) where r_k is rating submitted by u_a on item I_k. By having TR_a , the PROFILE LEARNER generates the user profile for u_a as a predictive model which will be used in subsequent stages by the FILTERING COMPONENT. Faced with a new item, the FILTERING COMPONENT compares its attributes to those in the user's profile and then predicts the degree of interest for the target active user. Finally, an ordered list of recommendations (L_a) will be generated for suggesting to u_a. After receiving the L_a by u_a, the new feedback will be recorded showing the user's feelings about the received recommended items. Meanwhile, the learning process is repeated to apply the newly gathered feedback into the user's profile. Considering the dynamic nature of user interests, the feedback-learning routine should be iterated by the PROFILE LEARNER to keep the user profile updated. It should also be considered that the user profile needs to be updated regularly by the PROFILE LEARNER as users' interests are usually subject to change with time.

2.2.2.2.2 Content-Based Filtering Algorithms

Items that are supposed to be recommended to the user are usually stored in a database table. A simple database table is shown in Table 2.1. In this table, there are records that describe three restaurants. The columns (e.g., Cuisine or Service) represent the restaurant properties. Each record is recognized by a unique identifier, or ID, in Table 2.1, which allows items with the same name to be distinguished and it serves as a key to retrieve the other attributes of the record.

Table 2.1: A sample of structured data

ID	Name	Cuisine	Service	Cost
1001	John's Pizza	Italian	Couter	Low
1002	Mary's café	French	Table	Medium
1003	Ali's Pizza House	Spanish	Table	High

The database illustrated in Table 2.1 is an example of structured data with a small number of attributes. Each item is shown by the same set of attributes, and there is a known set of values that the attributes may have. In this approach, the machine learning algorithms may be used to create a user profile from structured data.

There are also cases that need unstructured data for presenting and restoring the items. In this context, such as a text description of the restaurant, a restaurant review, or even a menu, there might be some entities that need to be represented using free text data. A simple example of unstructured data may occur in news articles (Table 2.2). In this example of unstructured data, the entire article can be treated as a large unrestricted text field.

Table 2.2: A sample of unstructured data

Lawmakers Fine-Tuning Energy Plan
SACRAMENTO, Calif. -- With California's energy reserves remaining all but depleted, lawmakers prepared to work through the weekend fine-tuning a plan Gov. Gray Davis says will put the state in the power business for "a long time to come." The proposal involves partially taking over California's two largest utilities and signing long-term contracts of up to 10 years to buy electricity from wholesalers.

Unrestricted texts, such as news articles are examples of unstructured data. Unlike structured data, there are no attribute names with well-defined values. Furthermore, the full complexity of natural language may be present in the text field including polysemous words (a word that may have several meanings) and synonyms (different words with the same meaning). For example, in the article in Table 2.7, “Gray” is a name rather than a color, and “power” and “electricity” refer to the same underlying concept.

A common approach to dealing with free text fields is to convert the free text to a structured representation. For example, each word may be viewed as an attribute, with a Boolean value indicating whether the word is in the article or with an integer value showing the number of times the word appears in the article. Many personalization systems that deal with unrestricted text use a technique to create a structured representation that originated with text search systems (Salton, 1989a). In this formalism, rather than using words, the root forms of words are typically created through a process called stemming (Porter, 1980). The goal of stemming is to create a term that reflects the common meaning behind words such as “compute,” “computation,” “computer” “computes” and “computers.” The value of a variable associated with a term is a real number that represents the importance or relevance.

From the standpoint of structured items, machine learning algorithms may be used to create a user profile from structured data (Pazzani et al. 2007). There are several such algorithms studied in previous research works. Decision trees have been used extensively in conjunction with structured data (Cho, Kim, & Kim, 2002; Cohen, 1995; Kim, Lee, Shaw, Chang, & Nelson, 2001; Kim et al., 2006). Other studies (Allan, Carbonell, Doddington, Yamron, & Yang, 1998; Billsus, Pazzani, & Chen, 2000; Yang, 1999) worked with the nearest neighbor method, which stores all of its training data in memory. In order to classify a new unlabeled item, the algorithm compares it to all stored items using a similarity function and determines the "nearest neighbor" or the *k* nearest neighbors. The methods that help users to incrementally refine queries based on previous interactions are commonly referred to as relevance feedback algorithms. The main objective of this approach is to enable users to rate documents with respect to their information needs (Manning, Raghavan, & Schütze, 2008). Some techniques that learn linear decision boundaries are referred to as linear classifiers. This algorithm has been used in some works for text classification tasks (Kivinen & Warmuth, 1997; T. Zhang & Iyengar, 2002). Another algorithm in this domain recognized by researchers is the Naïve Bayes, an exceptionally well-performing text classification algorithm frequently adopted in recent works (Nigam, McCallum, Thrun, & Mitchell, 1998; Yoshii, Goto, Komatani, Ogata, & Okuno, 2008). Next, the details of machine learning algorithms are presented as applied in a structured condition.

Decision Trees and Rule Induction

Decision tree learners such as ID3 (Quinlan, 1986) build a decision tree by recursively partitioning training data, in this case text documents, into subgroups until those subgroups contain only instances of a single class. Expected information gain is a commonly used criterion to select the most informative features for the partition tests (Yang & Pedersen, 1997).

Decision trees have been studied extensively in use with structured data such as that shown in Table 2.1. Given feedback on the restaurants, a decision tree can easily represent and learn a profile of someone who prefers to eat in expensive French restaurants or inexpensive Mexican restaurants. Arguably, the decision tree bias is not ideal for unstructured text classification tasks (Pazzani & Billsus, 1997). As a consequence of the information-theoretic splitting criteria used by decision tree learners, the inductive bias of decision trees is a preference for small trees with few tests. However, it can be shown experimentally that text classification tasks frequently involve a large number of relevant features (Joachims, 1998). Therefore, a decision tree's tendency to base classifications on as few tests as possible can lead to poor performance on text classification.

Nearest Neighbor Methods

The nearest neighbor method stores all characteristics of training data in memory. In order to classify a new item, this technique compares it to all previously stored items using a similarity function and determines the "nearest neighbor" or the k nearest neighbors. The class label or numeric score for a new item can then be derived from the class labels of the nearest neighbors. The similarity function used by the nearest neighbor method depends on the nature of the data. In case of structured data, a Euclidean distance metric is usually used. When using the vector space model, the cosine similarity metric is often used (Salton, 1989b). The cosine similarity function will not have a large value if corresponding features of two examples have small values. In contrast, in the Euclidean distance function, the same feature having a small value in two examples is treated the same as that feature having a large value in both examples. Consequently, it is appropriate for text when we want two documents to be similar when they are about the same topic, but not when they are both not about a topic.

The cosine similarity function and the vector space approach have been applied to several text classification applications (Allan, et al., 1998; Cohen & Hirsh, 1998; Yang, 1999). The Daily Learner system adopts the nearest neighbor method to create a model of the user's short term interests (Billsus, et al., 2000).

Relevance Feedback and Rocchio's Algorithm

Since the success of vector space model in document retrieval depends on the user's ability to run queries by selecting a set of representative keywords (Salton, 1989a), methods that refine queries based on previous search results have been concentrated in many researches. These methods are commonly referred to as relevance feedback. The general principle in these approaches is to allow users to rate documents returned by the retrieval system based on their preferences. This form of feedback can be used to incrementally refine the initial query. In a manner analogous to rating items, there are explicit and implicit means of collecting relevance feedback data.

Rocchio's algorithm (Rocchio, 1971) is a widely used relevance feedback algorithm that operates in the vector space model. The algorithm is based on the modification of an initial query through differently weighted prototypes of relevant and non-relevant documents. The approach forms two document prototypes by taking the vector sum over all relevant and non-relevant documents. The following formula summarizes the algorithm formally:

$$Q_{i+1} = \alpha\, Q_i + \beta \sum_{rel} \frac{D_i}{|D_i|} - \gamma \sum_{nonrel} \frac{D_i}{|D_i|} \qquad \textbf{(2.10)}$$

Here, Qi is the user's query at iteration i, and α, β, and γ are parameters that control the influence of the original query and the two prototypes on the resulting modified query. The underlying intuition of the above formula is to incrementally move the query vector towards clusters of relevant documents and away from irrelevant documents.

While this goal forms an intuitive justification for Rocchio's algorithm, there is no theoretically motivated basis for the above formula, i.e., neither performance nor convergence can be guaranteed. However, empirical experiments have demonstrated that the approach leads to significant improvements in retrieval performance (Rocchio, 1971).

Linear Classifiers

Algorithms that learn linear decision boundaries, i.e., hyperplanes separating instances in a multi-dimensional space, are referred to as linear classifiers. There are a large number of algorithms that fall into this category, and many of them have been successfully applied to text classification tasks (Lewis, Schapire, Callan, & Papka, 1996). All linear classifiers can be described in a common representational framework. In general, the outcome of the learning process is an n-dimensional weight vector w, whose dot product with an $n$dimensional instance, e.g., a text document represented in the vector space model, results in a numeric score prediction. Retaining the numeric prediction leads to a linear regression approach. However, a threshold can be used to convert continuous predictions to discrete class labels. While this general framework holds for all linear classifiers, the algorithms differ in the training methods used to derive the weight vector w. For example, the equation below is known as the Widrow-Hoff rule, delta rule or gradient descent rule and derives the weight vector w by incremental vector movements in the direction of the negative gradient of the example's squared error (Widrow & Hoff, 1960). This is the direction in which the error falls most rapidly.

$$w_{i+1,j} = w_{i,j} - 2\eta(w_i \cdot x_i - y_i)x_{i,j} \quad \textbf{(2.11)}$$

The equation shows how the weight vector w can be derived incrementally. The inner product of instance x_i and weight vector w_i is the algorithm's numeric prediction for instance x_i. The prediction error is determined by subtracting the instance's known score, y_i, from the predicted score. The resulting error is then multiplied by the original instance vector x_i and the learning rate η to form a vector that, when subtracted from the weight vector w, moves w towards the correct prediction for instance x_i. The learning rate η controls the degree to which every additional instance affects the previous weight vector.

Probabilistic Methods and Naïve Bayes

In contrast to the lack of theoretical justifications for the vector space model, there has been much work on probabilistic text classification approaches. This section describes one such example, the naïve Bayesian classifier. Early work on a probabilistic classifier and its text classification performance was reported by Maron (Maron, 1961). Today, this algorithm is commonly referred to as a naïve Bayesian Classifier (Duda & Hart, 1973).

The algorithm's popularity and performance for text classification applications have prompted researchers to empirically evaluate and compare different variations of naïve Bayes (Lewis, 1998; McCallum & Nigam, 1998). In summary, McCallum and Nigam (McCallum & Nigam, 1998) note that there are two frequently used formulations of naïve Bayes, the multivariate Bernoulli and the multinomial model. Both models share the following principles. It is assumed that text documents are generated by an underlying generative model, specifically a parameterized mixture model:

$$P(d_i \mid \theta) = \sum_{j=1}^{|C|} P(c_j \mid \theta) P(d_i \mid c_j; \theta) \qquad \textbf{(2.12)}$$

Here, each class c corresponds to a mixture component that is parameterized by a disjoint subset of θ, and the sum of total probability over all mixture components determines the likelihood of a document. Once the parameters θ have been learned from training data, the posterior probability of class membership given the evidence of a test document can be determined according to Bayes' rule:

$$P(c_j \mid d_i;\hat{\theta}) = \frac{P(c_j \mid \hat{\theta})P(d_i \mid c_j;\hat{\theta})}{P(d_i \mid \hat{\theta})} \quad \textbf{(2.13)}$$

While the above principles hold for naïve Bayes classification in general, the multivariate Bernoulli and multinomial models differ in the way $P(c_j|d_i;\hat{\theta})$estimated from training data.

In summary, after reviewing different machine learning algorithms, we can conclude that when facing to small number of structured attributes, the performance, simplicity and understandability of decision trees for content-based models are all advantages (Pazzani & Billsus, 2007). Kim, et al. (2006) adopted this method for personalizing advertisements on web pages. To provide personalized advertisements in that study, a hierarchical tree data structure was considered for storing the personal preference scores of a customer for each product category. This tree structure was implemented using a table consists of three columns [Customer ID (CID), Product Group ID (PGID), Preference Score (PS)]. Relatively, the preference scores in the preference table approach were defined as follows:

$$PS(i,j) = \propto_1 \times\ Profile(i,j) + \propto_2 \times\ Purchase(i,j) + \propto_3 \times\ Interest\ Type_1(i,j) + \cdots + \propto_{n+2} \times\ Interest\ Type_n(i,j) \quad \textbf{(2.14)}$$

where $PS(i,j)$ is the preference score of customer i for the leaf-level product category j , $Profile(i,j)$ is the profile score of customer i for the product category j , which is specified in customer i's initial profile. If customer i has specified product j as an interesting product category in his/her profile, then the values of $Profile(i,j)$ is 1, otherwise it is 0, $Purchase(i,j)$ is the number of purchases of products that belong to product category j, $Interest\ Type_k$ (k = 1, . . . , n) is the number of interest expressions of the kth type, and $\propto_k$ (k = 1, . . . , n + 2) isthe weight of each term.

In case of unstructured approach, The term frequency/inverse document frequency (TF-IDF) is one of the most popular measures for identifying keyword weights in recommender systems (Salton, 1989b; Mangina & Kilbride, 2008).

Recalling from the formula that Adomavicius and Tuzhilin (2005) provided for recommendation problem, all recommender systems have a common goal to find such items $s' \in S$ that maximizes u for each user $c \in C$:

$$\forall c \in C,\ s \in S,\ s'_c = argmax\, u(c,s) \qquad \textbf{(2.15)}$$

In content-based RSs, the utility $u(c,s)$ of item s for user c is predicted by considering the pairs of $u(c,s_i)$ which represent the ratings assigned previously by user c to other "similar" items to s called $s_i \in S$. To clarify this concept, imagine a music recommendation system that tries to suggest the most related music files to its users. In this example, the content-based RS elicits the common characteristics of previously rated music files by user c to predict the most relevant new ones to him (or her).

For more details, item profile is defined as the set of attributes characterizing items$s$, $Content(s)$. This is used to measure how appropriate the item s is to be selected for recommendation purpose. Each item may have several attributes which should be involved in item profile. One of the main issues is the "importance" of attribute $\boldsymbol{att_i}$in $\boldsymbol{item_j}$ with some weighting measures $\boldsymbol{w_{ij}}$that can be defined in several different ways (Balabanović & Shoham, 1997; Pazzani & Billsus, 1997).

The term frequency/inverse document frequency ($TF - IDF$) is one of the most popular measures for identifying keyword weights in recommender systems (Salton, 1989b). To define it, assume that N is the total number of items that are eligible to be recommended to users and that keyword k_j appears in n_i of them. Furthermore, consider $f_{i,j}$ as the number of times keyword $\boldsymbol{k_i}$ appears in document$\boldsymbol{d_j}$. Then, $\boldsymbol{TF_{i,j}}$, the term frequency (or normalized frequency) of keyword $\boldsymbol{k_i}$ in item $\boldsymbol{d_j}$, is defined as

$$TF_{i,j} = \frac{f_{i,j}}{max_z f_{z,j}} \qquad \textbf{(2.16)}$$

where the maximum is the biggest value of the frequencies $\boldsymbol{f_{z,j}}$ of all keywords $\boldsymbol{k_z}$in item$\boldsymbol{d_j}$. Although, we shouldn't count on the keywords that are repeated in many items because they will not so helpful in predicting the most relevant items. To overcome this problem, it's better to use the inverse document frequency ($\boldsymbol{IDF_i}$) in combination with $\boldsymbol{TF_{i,j}}$.

$\boldsymbol{IDF_i}$is defined as

$$IDF_i = \log\frac{N}{n_i} \qquad \textbf{(2.17)}$$

Hence, for each item $\boldsymbol{d_j}$ we can identify the $TF - IDF$ weight for keyword $\boldsymbol{k_i}$as

$$w_{i,j} = TF_{i,j} \times IDF_i \qquad \textbf{(2.18)}$$

Consequently, the content of item $\boldsymbol{d_j}$is defined as

$$Content(d_j) = (w_{1j}, \dots\ w_{kj},) \qquad \textbf{(2.19)}$$

As mentioned earlier, content-based RSs suggest the alternatives which are similar to high rated items by the target user (Cremonesi, Turrin, & Airoldi, 2011). In other words, the alternated items will be compared with ones rated by the user in the past and the best matching items are recommended. Considering these assumptions, let $ContentBasedProfile(c)$ be the profile of user c including his/her preferences and tastes. This kind of information can be elicited by analyzing the items previously viewed and rated by this specific user. In more particular, $ContentBasedProfile(c)$ can be defined as a vector of weights$(w_{c1}, \ \dots, w_{ck})$, where each weight w_{ci} represents the significance of keyword k_i to user c.

In content-based recommender algorithms, the utility function $u(c, s)$ is presented as:

$$u(c, s) = score\ (ContentBasedProfile(c), Content(s)) \qquad \textbf{(2.20)}$$

In this context, both $ContentBasedProfile(c)$ and $Content(s)$ can be defined as $TF - IDF$ vectors $\vec{w}_c$ and $\vec{w}_s$ of keyword weights (Adomavicius & Tuzhilin, 2005):

$$u(c, s) = \cos(\vec{w}_c, \vec{w}_s) = \frac{\vec{w}_c \cdot \vec{w}_s}{\|\vec{w}_c\|_2 \times \|\vec{w}_s\|_2} = \frac{\sum_{i=1}^{K} w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^{K} {w_{i,c}}^2} \sqrt{\sum_{i=1}^{K} {w_{i,s}}^2}} \qquad \textbf{(2.21)}$$

whereK is the total number of keywords in system.

In addition to the traditional techniques mostly based on information retrieval concept, there are also some other heuristics for content-based RSs. Bayesian classifiers (Mooney, Bennett, & Roy, 1998), artificial neural networks, decision trees, and clustering (Pazzani & Billsus, 1997) are some other approaches in this domain. The adopted technique for calculating the utility predictions is the origin of differences between these approaches and the others which based on information retrieval algorithms. Instead of applying some heuristics, the predictions are made based on a learned model using machine learning and statistical techniques. For example in case of some Web pages that were rated by the user as “relevant” or “irrelevant” (Pazzani and Billsus, 1997) , utilized the naive Bayesian classifier to classify unrated Web pages. More precisely, considering the set of keywords $k_{1,j}, \dots, k_{n,j}$ on the page p_j, they used the Bayesian classifier for estimating this probability that p_j belongs to a certain class C_i (e.g., relevant or irrelevant):

$$P\left(C_i \,\middle|\, k_{1,j} \& \dots \& k_{n,j}\right) \quad \textbf{(2.22)}$$

And furthermore, assuming that the keywords are independent, the above probability is changed to

$$P(C_i) \prod P\left(k_{x,j} \middle| C_i\right) \quad \textbf{(2.23)}$$

Several studies have revealed that the native Bayesian classifiers are coherent enough in recommending the most relevant items, while the keyword independence idea is not applicable in most application domains (Pazzani & Billsus, 1997).

2.2.2.2.3 Advantages and Shortcomings of Content Based Recommendations

Comparing to collaborative paradigm, the Content Based recommender systems have some advantages that are discussed in following (Lops, et al., 2011).

• **User Independence** – To build the active user profile, content-based RSs use the elicited ratings submitted by his (or her). While, collaborative recommender systems need the ratings from similar users who called the active user "nearest neighbors". Consequently, in content-based methods, it is not necessary to track the behavior of the neighbors for each active user.

• **Transparency** – In this method, the description supporting how an item included in the recommendation list can be presented to active users. This kind of clarification may be effective in increasing the users' trust on recommending mechanism. The listed features of recommended item can be used as an indicator for judging whether the RS works well. In the other side, collaborative methods work such a black box without any further description, except that we know just some unknown similar users liked the suggested item in past.

• **New Item -** Content-based RSs are effective in case of recommending new items which not yet rated in past. Consequently, they do not have the first-rater problem of collaborative Recommendation Systems.

Although the above mentioned items illustrate the bright side of content-based RSs, they have also several limitations that are discussed in following.

• **Limited Content Analysis** – The Content-based methods have an a inevitable limitation relating to the type and number of features associated with the items they suggest. It means without having a comprehensive knowledge about the items domain, we cannot expect from this recommender to suggest suitable alternatives. Because it needs enough information about the analyzed content to distinguish between the interested items and ones that the user doesn't like.

• **Over-Specialization** – This shortcoming which is also famous as serendipity phenomena, points to this fact that content-based systems tend to generate recommendations with a low level of novelty. It is derived from the nature of this method that it is not capable of suggesting some unexpected items as it just recommend items which are more relevant to the user profile. Consequently, the active user is usually supposed to receive just the items similar to those he (or she) rated in the past.

• **New User –** In case of new users with a few numbers of ratings, this method cannot recommend properly. The roots of this problem can be traced to this fact that this method needs to gather enough ratings to understand the users' interests and recommend in a coherent way.

2.2.2.3 Hybrid Approach

CF methods usually do not compete with content-based Recommendation Systems. Alternatively, they both integrate to build a more effective hybrid solution. Several successful researches in this context have been developed like MovieLens (Jung, 2011), Video Recommender (Verhoeyen, Vriendt, & Vleeschauwer, 2012) and GroupLens (Miller, Riedl, & Konstan, 2003).

The hybrid recommender systems, as the combination of recommending techniques A and B use the advantages of A to cover the shortcomings of B. For example, as mentioned previously, collaborative filtering methods that suffers from new-item problems cannot calculate the prediction for new items without any rating records. Hence, the combination of content-based and collaborative methods can be an alternative to overcome this shortcoming and gain better results (Ricci, et al., 2011)..

The combination of content-based and collaborative recommender systems into a hybrid approach can be classified in following ways:

1. implementing content-based and collaborative recommender systems separately and then combining their recommendations,
2. applying some content-based attributes into a collaborative method,
3. applying some collaborative characteristics into a content-based method, and
4. Combining both collaborative and content-based method to generate a general unifying model.

Some studied has been carried out by researchers in this context that will be described in following.

2.2.2.3.1 Combining Separate Recommenders

One solution for making a hybrid model is to implement content-based and collaborative systems separately, and then using the recommendations generated from two separate ways. In this approach, the outputs of these two method can be combined into one final recommendation list utilizing either a voting pattern (Pazzani, 1999) or a linear combination of ratings (Claypool et al., 1999). Another alternative is using one of them at any given situations as the "better" one based on some recommendation "quality" measure.

2.2.2.3.2 Adding Content-Based Features to Collaborative Algorithms

Fab system (Balabanović & Shoham, 1997) and "collaboration via content" method (Pazzani, 1999) are two samples of recommender systems that typically are based on collaborative filtering but take advantage of the user profile idea from content-based approach. As Pazzani mentioned in 1999, these profiles are used for finding the similarity between users, instead of common rated items. Applying this change to pure collaborative methods, leads to overcome some problems related to sparsity.

As another advantage of this method, we can mention to this fact that user are recommended by not only the items rated highly by similar users, but also when the items are highly related to user's profile (Balabanović & Shoham, 1997). In other study was carried out by Melville et al.(2002), the usual ratings vector in collaborative method is fortified by additional ratings calculated with content-based predictor.

2.2.2.3.3 Adding Collaborative Features to Content-Based Algorithms

The most usual method in this approach is applying some dimensionality reduction technique on a group of content-based profiles. For example, the latent semantic indexing (LSI) is used in research was carried out by Soboroff & Nicholas (1999) to create a collaborative model among a group of user profiles, where the profiles are demonstrated by term vectors. This method comes up with some improvements in performance comparing to the pure content-based method.

2.2.2.3.4 Developing a Single Unifying Recommendation Model

A considerable amount of literature has been published on this approach. (Schein, et al., 2002) and (Popescul, Pennock, & Lawrence, 2001) introduce a unified probabilistic model based on the probabilistic latent semantic analysis (Hofmann, 2001) for combining content-based and collaborative recommender systems. Another model is proposed by (Ansari, Essegaier, & Kohli, 2000) which use Bayesian mixed-effects regression models and apply Markov chain Monte Carlo methods for calculating and predicting the parameters.

Another way for augmenting the hybrid RSs is using knowledge-based techniques (Burke, 2002). In this context, case-based reasoning can make the recommenders more coherent and address some of their traditional shortcomings. For instance, we can mention to Entre´e as a knowledge-based RS that utilize some domain information about restaurants, cuisines, and foods to prepare a list of suggested restaurants to its users.

As the same as most artificial intelligence applications, the knowledge-based RSs also face to knowledge acquisition. So, it's advised to use them in applications where domain knowledge is already available in structured form (Middleton, Shadbolt, & De Roure, 2004).

Furthermore, there are several recent papers, such as (Durao& Dolog, 2010), (Ghazanfar & Prugel-Bennett, 2010), and (Porcel, Tejeda-Lorente, Martínez, & Herrera-Viedma, 2012) that carried out a comparative study on performance of the hybrid recommender systems, and traditional content-based and collaborative approaches.

2.2.2.4 Demographic Algorithms

The recommendations in this type of RSs are based on demographic attributes of the users. It follows this idea that users with variant demographic profiles should be recommended differently. There are many Websites that consider demographics in recommending products and services to their clients. For example, they consider the language, country and even the age of users for grouping them into different categories. Although these algorithms have been quite practical and popular in the marketing literature, there has been relatively little suitable research into demographic methods (Mahmood & Ricci, 2007b; Mosayebian, Keramati, &Khatibi, 2012).

2.2.2.5 Knowledge-Based Algorithms

In this type of recommender systems, items are recommended based on specific domains of knowledge regarding how the certain features of items can meet the users' preferences and needs. (Bridge, Göker, McGinty, & Smyth, 2005) and (Ricci, et al., 2006) argued that remarkable knowledge-based RSs are case-based. In these kinds of recommender systems a similarity function is used to estimate to what extent the user preferences match the recommended items. This similarity value can be considered as appropriateness of the recommendation for the user. There is another type of knowledge-based recommender systems that is called constraint-based systems. Both approaches are similar in terms of applied knowledge: user requirements are elicited; in cases that no recommendation could be proposed, repairs for inconsistent data are automatically applied; and ultimately recommendation results are generated. The root of major difference can be traced in the way recommendations are generated. Case-based solutions calculate recommendations based on similarity metrics whereas constraint-based RSs use predefined knowledge bases containing some well-defined rules for matching the users' needs with item attributes (Martinez, Barranco, Perez, & Espinilla, 2008).

2.2.2.6 Social Network-Based Algorithms

Social networking sites (SNS) are a type of virtual community that has grown extremely over the past few years. Users of this special kind of online environments can easily connect with each other, disseminate information, and share the online content through their profiles (Zhou, et al., 2012). In the past few years, the dramatic expanding of Web 2.0 and web-based applications has led to new challenges for current recommender systems. Traditional recommender systems mostly ignore social relationships among users while in our real life, when we are asking our friends for suggestion of the most interesting movies or nice digital cameras, we are actually requesting verbal social recommendations. Social recommendation is a real fact, and we always turn to our friends for recommendations. Consequently, to improve recommender systems for provide more personalized recommendation results, we need to incorporate social network information among users (Bonhard, et al., 2007).

The preferences and needs of the user's friends play the main role in community-based recommender systems. In other words, it follows the epigram "Tell me who your friends are, and I will tell you who you are" (Arazy, Kumar, & Shapira, 2009). In (2005),Smeaton et al. demonstrated that people prefer to get recommendations from their friends rather than other persons who they do not know. Results of this study, combined with the rapid growth of social networks, prove a dramatically growth of interest in community-based systems. Some recent studies refer to this special kind of

RSs as social recommender systems (Golbeck, 2006). This approach needs information about the social interactions of the users besides of the interests and preferences of their friends. Hence, the process of recommendation is in the base of ratings which submitted by the user's friends.

This approach is still in the early stages of research. For example, some recent studies in this area (Groh & Ehmig, 2007; Massa & Avesani, 2004) argued that social network based RSs are no more coherent than traditional collaborative filtering algorithms, except in some special conditions, such as cold-start situations where the users are quite new in the system and didn't assign any ratings to be compared with the others. In another research work (Seth, A., & Zhang, J., 2008) a recommender system based on a Bayesian user-model was proposed and evaluated. They used the underlying social network of blog authors and readers to model the preference features for individual users. In (2009), Guy et al. published a paper in which they described that in some situations social-network approaches come up with better results comparing with methods based on profile similarity data. He, J., & Chu, W. W. (2010) presented a new paradigm of recommender systems which can utilize information in social networks, including user preferences, item's general acceptance, and influence from social friends. They developed a probabilistic model to make personalized recommendations from such information. Furthermore, they propose to improve the performance of their system by applying semantic filtering of social networks. And also according to another relevant study by (Groh & Ehmig, 2007), the performance of recommender systems can be improved when social network data is added to traditional collaborative filtering methods.

2.2.3 Methods for Evaluating the Recommender Systems

Shaniand Gunawardana (2011) grouped the experimental design methods for evaluating recommender systems into three main categories. In the following section these experimental techniques with the potential to be used for comparing the efficiency of different recommender systems are presented. The discussion can also be traced back to other related areas, such as information retrieval and machine learning (Salzberg, 1997; Voorhees, 2002; Demšar, 2006).

First, the offline experiments are explained, and they are the easiest to conduct. As the term ‘offline’ suggests, interaction with real users is not necessary. The second method to be discussed is user studies. In this kind of evaluation, a small group of target users are selected to test the system in a controlled environment, after which they are asked to report based on their experiences. Finally, group of real users utilize the system without being aware of the experiment.

2.2.3.1 Offline Experiments

The offline experimental method involves pre-collected data sets from users who have chosen or rated items. In this way it is possible to simulate the behavior of users who would normally interact with a recommender system. To facilitate the making of reliable decisions in this method, the assumption may be that the users’ behavior in the experimental data gathering phase is sufficiently similar to their behavior in real conditions when the recommender system is deployed. The independence of offline experiments in interacting with real users is what makes this technique so appealing to researchers. In addition, it is possible to compare several candidate recommender systems at a low cost. A shortcoming with offline experiments, however, is the fact that they may only be used to answer a very limited range of questions, such as the prediction power of a recommender system.

Accordingly, the offline technique cannot be expected to directly measure the effects of recommender systems on user behavior. The approach can be helpful though when planning to filter out inappropriate recommender algorithms. To test the remaining small set of candidate algorithms, user studies or online environments may be employed at higher cost and with more effort.

The data sets utilized in offline experiments should be as similar as possible to real data that the recommender algorithm is expected to face in a runtime environment. It should be carefully considered that there is no bias in the distributions of selected items, ratings and users. For example, when manipulating large data sets, the researcher may decide to exclude users or items with low counts to reduce experiment cost. Such a situation may result in systematic bias. Hence, if necessary, randomly selecting users and items is suggested in order to reduce the amount of data (Mahmood & Ricci, 2007a).

In this method, the majority of research papers use a fixed number of hidden items or known items per sample user (called "all but n" or "given n" protocols). This method is applicable when diagnosing algorithms and illustrating in which situations they function best. If the recommender algorithm is to be tested using this approach, the question of whether we want to present the recommended items only to users who have ranked exactly *n* items, or are expected to rank more than *n* items, needs to be answered. If this is not the experimental situation we are looking for, this protocol is not reliable and the research results will be biased in predicting algorithm efficiency in real environments.

The online behavior of users should be simulated to assess the recommender algorithms via offline methods. It may thus be possible for the system to trace the recommended items and compute how the user employs or amends the recommendations (Mahmood & Ricci, 2007a). In doing so, the historical user interactions should be recorded, and some of this data should be hidden to simulate user behavior when rating items. In cases with large data sets, a simple means is to randomly select sample-test users and consider a random time moment immediately prior to a user's action. Then all sample users' selections will be hidden, right after which an attempt is made to recommend items to them. The aforementioned protocols employ user models that illustrate the behavior of users in their interactions with a specific application. More complex models can be utilized to resemble user behavior as well (Mahmood & Ricci, 2007a), but in such circumstances care should be taken when trusting the experiments to ensure the results can verify the recommender algorithms.

2.2.3.2 User Studies

In this method, a set of test users were asked to perform some routines in terms of interacting with the recommender algorithm. As the selected users carried out the assigned tests, their behavior was observed through related quantitative measurements. The adopted measurements may include the accuracy of the expected results, the portion of the task that was completed, or the time it took to perform it. In various research situations, users can be asked to answer the qualitative questions before, during, and after the task is completed. These sorts of questions may also be helpful when collecting data that is not directly observable, such as whether the designated task was easy enough to complete, or if the user found the interface enjoyable.

As an example of a user study, it is worth noting the evaluated influence of a recommender algorithm on user behavior when browsing the news items. Here the test users are asked to read a set of news that seemed interesting to them. A number of these items were related news recommendations and others merely random news without having been recommended at all. It was thus possible to check whether the recommended news items were used, or if the users preferred to read the other news. This type of study facilitates compiling data pertaining to how many times a recommendation was clicked on, and in more specific cases, eye movement is tracked to confirm whether a recommended item succeeded in catching the user's attention.

User studies have the advantage of being able to cover the widest range of questions among all three experimental approaches discussed in this section. It was mentioned earlier that by applying this method it is possible to evaluate how the recommender algorithm affects user behavior during real-life interactions with the system. In the offline approach, only a limited number of assumptions can be made, such as "the user is interested in using a relevant item recommended by a given algorithm." In terms of qualitative data, the user studies approach is the only one that allows the collection of this sort of data. Since the users are closely monitored while performing the tests, this method enables a large set of quantitative measurements to be gathered.

This user studies method, however, has some limitations as well. It is primarily too costly to conduct, and accumulating a large set of test users to carry out a sufficiently large set of tasks requires a lot of effort. Hence, it would be wise to limit the project scope to a small set of users and accordingly, a small set of tasks. It would also be preferable for each scenario to be repeated several times in order to achieve reliable results.

Furthermore, in instances when applications malfunction during particular user interactions, pilot user studies should initially be executed to prevent failed experiments. These entail limited experiments designed solely to test the applications for runtime errors and malfunctioning. As such, the results of pilot studies should not be used in the measurement computing phase.

It is worth remarking that the test users must represent the target population of users who will interact with the system in a real environment as closely as possible. For example, avid movie fans would definitely not comprise suitable test users in evaluating a movie recommender system. The reason may be that volunteers who are more interested in a subject may not be appropriate alternatives for representing the behavior of the true population in the user domain. Also, users who get paid would undeniably want to satisfy those conducting the experiment to some degree. Hence, when the test users are aware of the research objective, they may unconsciously provide evidence supporting it. To avoid this, test users should not be made aware of the goal of the experiment.

2.2.3.3 Online Evaluation

With regards to the main role of recommender systems, the system designer frequently attempts to influence the behavior of users. For this reason, it is important to measure the changes in user interactions with the system via different recommender algorithms. For instance, evidence that one recommender system is more efficient than another can be gathered by whether users of one system employ the recommended items more often, or whether some statistics collected from the users of one system are better than those gathered from users of another system.

Several factors may influence the effects of recommender systems. They include the user's intent (e.g. how much risk vs. how much novelty they are seeking, how specific their information needs are, etc.), the user's context (e.g. how familiar they are with the items, to what extent they trust the application, etc.), and the interface through which the recommendations are presented.

Hence, online evaluation is the strongest experimental approach for measuring the efficiency of a recommender system (Shani & Gunawardana, 2011) because the system is tested by real users in a real environment. Consequently, it is most preferable to compare a few systems online rather than limited numbers of test users that are more difficult to interpret. Owing to this advantage, several real-life systems make use of the online testing technique to compare multiple recommender algorithms (Kohavi et al. 2009). As a way of recording user interactions with variant recommender algorithms, such systems redirect a portion of the traffic to different designated recommendation engines. In running this kind of test several essentials should be considered. For one, to attain fair comparisons between recommender algorithms, random user selection is necessary to redirect users to designated algorithms. The different recommender system concepts should also be separated. If, for example, the research goal is to evaluate the algorithm's accuracy, it is important to maintain a fixed user interface. Similarly, if we want to concentrate on a better user interface, the underlying algorithm needs to be kept fixed.

A number of side effects ought to be taken into consideration as well. One can be a test recommender system that makes irrelevant suggestions and could inconvenience users and discourage them from using the real system ever again. Such situations are not acceptable -- especially in commercial environments.

In light of the above-mentioned explanations, the optimum strategy is to first run the offline experiments to ensure that the candidate approaches have the minimum number of characteristics necessary to make them eligible for experimentation, and then use the online study to evaluate the efficiency of the candidate algorithms (Herlocker et al., 2004; Shani & Gunawardana, 2011). Such gradual route reduces the inconvenience experienced by test users. Online experiments are recognized as the best method of evaluating recommender systems because they facilitate the direct measurement of the system's overall goals, e.g. user retention or long-term profit. Consequently, online experimentation is the most appropriate means of understanding how system properties influence parameters, such as recommendation accuracy and diversity, as well as recognizing the tradeoffs between these properties (Shani & Gunawardana, 2011).

In this section, three methods of evaluating recommender systems were described, namely offline experiments, user studies, and online evaluation. The majority of research works in the domain of recommender systems employ the offline analysis method for assessing the performance of recommender algorithms (Adomavicius & Tuzhilin, 2005; Herlocker, et al., 2004). In offline studies, offline datasets such as MoveiLens[1] and GroupLens[2] are used to test the recommender algorithms (Chen, et al., 2010; Jung, 2011; Nguyen & Dinh, 2012; Sarwar, et al., 1998). Other researchers prefer the user studies method for model evaluation in the field of recommender systems

---

[1] http://movielens.umn.edu
[2] http://www.grouplens.org/node/12

(Bambini, et al., 2011; Ge, et al., 2010). There are still others who favor the online testing methods (Kohavi, Longbotham, Sommerfield, & Henne, 2009).

The most significant advantages of the offline method are that it is economical and easy to implement. These characteristics have made offline techniques appropriate for conducting large evaluations on several different algorithms and datasets simultaneously. In other words, following dataset preparation, the experiment can be conducted simply by running the given algorithms.

Offline methods have two principal disadvantages. First, the set of items to be evaluated is limited due to the natural scarcity of data sets representative of the ratings. In the absence of rating records from a given user for a specific item, it is not possible to evaluate the relevance of a recommended item to that user. Secondly, objective evaluation of prediction results is challenging. None of the offline analysis methods can claim that users will certainly prefer a specific recommender system or not, either as a result of its predictions or because of other less objective parameters such as user interface aesthetics (Herlocker, et al., 2004).

2.2.4 Recommender Systems Prediction Accuracy Measurement

In the literature of recommender systems, the prediction accuracy is one of the most important items for discussion (Shani & Gunawardana, 2011). The prediction engine is the component that is considered at the base of the majority of RSs. The responsibility of this engine is to predict the user opinions about the recommended items (e.g. ratings of music files) or the probability of usage (e.g. purchase). A typical assumption in study of RSs is that the recommender system with more accurate predictions will be better from the user's point of view.

As prediction accuracy is basically independent of the user interface, it can thus be used in offline experiments. When this metric is used in a user study, it measures the accuracy of a given recommendation. It is a different issue from the prediction of user behavior without recommendations. Consequently, it is closer to the true accuracy in the real environment.

In following, we discuss three different categories of metrics for measuring the accuracy of predictions; measurements for computing the accuracy of ratings predictions, measurements for computing the accuracy of usage predictions, and finally measurements for computing the accuracy of rankings given to items.

2.2.4.1 Measuring Ratings Prediction Accuracy

In some online environments (e.g. the new versions of DVD rental service at Netflix[1]), the main goal of recommender engine is to predict the rating a user would give to a recommended item by selecting 1 to 5 stars. In other words, the accuracy of the system's predicted ratings would be calculated in this situation.

Root Mean Squared Error (RMSE) is considered as the most popular measurement for computing the accuracy of predicted ratings (Shani & Gunawardana, 2011). In this method, we calculate predicted ratings $\hat{r}_{ui}$ for a test set τ of user-item pairs (u, i) for which the true ratings r_{ui} are known. We consider the r_{ui} as known, because they are

[1]https://dvd.netflix.com/

hidden in an offline experiment approach, or because they would be obtained through a user study or online experiment. The RMSE between the actual and predicted ratings is identified by:

$$\text{RMSE} = \sqrt{\frac{1}{|\tau|}\sum_{(u,i)\in\tau}(\hat{r}_{ui} - r_{ui})^2} \quad \textbf{(2.24)}$$

And also, there is another alternative that is called Mean Absolute Error (MAE):

$$\text{MAE} = \sqrt{\frac{1}{|\tau|}\sum_{(u,i)\in\tau}|\hat{r}_{ui} - r_{ui}|} \quad \textbf{(2.25)}$$

Normalized RMSE (NRMSE) and Normalized MAE (NMAE) are two other versions of RMSE and MAE that have been normalized by the range of the ratings (i.e. $r_{max} - r_{min}$). As the functionality of normalized versions is the same as RMSE and MAE metrics, their resulting rankings are the same as the unnormalized measures.

Average RMSE and Average MAE are mostly used in case of unbalanced test sets. In this situation, the RMSE or MAE methods might be heavily suffered through the side effects of the error on a few very frequent items. If we prefer to measure the prediction error on any item, it is advised to apply MAE or RMSE for each item separately and then use the average value over all items.

2.2.4.2 Measuring Usage Prediction

In many applications, it is not expected from recommendation systems to predict the user's ratings of items. In these situations, they are supposed to recommend to users the items that they may use. For example in Netflix web site, when the user add some movies to online shopping cart, the recommender system suggests a collection of movies that may also be interesting for her. Hence, we are curious to know whether the system properly recommends an item that the user will use it (Shani & Gunawardana, 2011).

Usage prediction measurements calculate the frequency with which a recommender algorithm suggests relevant or irrelevant items. Thus as Herlocker et al. pointed out in (2004), this approach is appropriate for some practical usages such as finding appropriate items in such situations that the users have true binary preferences. As mentioned previously, these metrics are not used for directly measuring the qualifications of a recommender system to predict ratings accurately. Deviations from actual ratings are tolerated, as long as they do not lead to classification errors. In following, Precision and Recall as two particular metrics of this context are briefly discussed.

As the two most popular measurements, Precision and Recall are used for evaluating information retrieval algorithms that were proposed by (Cleverdon, Mills, & Keen, 1966) and have been used ever since. In the history of recommender systems, they have been adopted by (Billsus & Pazzani, 1998), (Basu, Hirsh, & Cohen, 1998) ,(B. Sarwar, et al., 2001), and (Shani & Gunawardana, 2011) . To calculate the precision and recall, a 2×2 table is used that is shown in Table 2.3. The item set are categorized into two classes—relevant or irrelevant. That is, if the rating scale is not already binary, we need to transform it into a binary scale. For example, the MovieLens dataset has a rating scale of 1–5 and is commonly transformed into a binary scale by converting every rating of 4 or 5 to "relevant" and all ratings of 1–3 to "not-relevant."

For calculating the precision and recall, it is needed to consider another categorization. We should separate the items into the set that was recommended to the user (selected/recommended), and the set that was not.

Consequently as shown in Table 2.3, four possible conditions may happen based on selection and usage situations:

- N_{rs} = Relevant (Used) and Selected (Recommended) Items
- N_{is} = Irrelevant (Not Used) and Selected (Recommended) Items

- N_{rn} = Relevant (Used) but Not Selected (Not Recommended) Items
- N_{in} = Irrelevant (Not Used) and Not Selected (Not Recommended) Items
- N_s = Total number of Selected (Recommended) Items
- N_n = Total number of Not Selected (Not Recommended) Items
- N_r = Total number of Relevant (Used) Items
- N_i = Total number of Irrelevant (Not Used) Items

Table 2.3: The Possible Conditions of Recommendation of Items to Users

	Selected	Not Selected	Total
Relevant	N_{rs}	N_{rn}	N_r
Irrelevant	N_{is}	N_{in}	N_i
Total	N_s	N_n	N

Considering the above presented definitions, Precision or true positive accuracy is calculated as the ratio of selected (recommended) items that are used (relevant) to the total number of selected (recommended) items (Herlocker, et al., 2004):

$$Precision = tpa = \frac{N_{rs}}{N_{rs}+N_{is}} \quad \textbf{(2.26)}$$

This is the probability that a recommended item corresponds to the user's interests and preferences.

Recall or true positive rate is calculated as the ratio of selected (recommended) items that

Are used (relevant) to the total number of used items (Herlocker, et al., 2004):

$$Recall = tpr = \frac{\boldsymbol{N_{rs}}}{\boldsymbol{N_{rs}+N_{rn}}} \quad \textbf{(2.27)}$$

This is the probability that a used (relevant) item is recommended.

As Shani and Gunawardana (2011) pointed out, Precision and recall are the most popular measurements for evaluating the prediction accuracy of recommender systems. These two metrics are inversely related.

These measurements depend on the separation of the concept of relevant and irrelevant items. The definition of “relevance” and the suitable approach to calculate it has been one of the significant sources of argument within the field of information (Harter, 1996; Voorhees, 2000). The majority of information retrieval evaluation methods have focused on an objective viewpoint of relevance, where it is defined with respect to a query, and is independent of the user. In doing so, the documents can be compared with queries to determine which documents are relevant to which queries. However, in case of recommender systems, the objective relevance makes no sense and it's not applicable. Recommender systems suggest items based on the likelihood that they will meet a given user's preferences and interest. That user is the only person who can determine if an item is suitable based on his interests or not. Consequently, in recommender systems' domain, relevance is considered as a subjective issue (Herlocker, et al., 2004).

Referring back to the definition of precision and recall, typically we can expect a trade off between these two measurements. More specifically, in case of longer recommendation lists, the recall is increased while the precision is decreased. So because of this mutual dependence it is worth to consider precision and recall in conjunction with other measurement called fallout. By the way, in some environments where the number of recommendations that can be presented to the user is preordained, the most useful measure of interest is Precision at N where N is the number of recommended items to the user (Shani & Gunawardana, 2011).

Fallout or false positive rate is measured as the ratio of selected (recommended) items that are not used (irrelevant) to the total number of not used items (Hernández del Olmo & Gaudioso, 2008):

$$Fallout = fpr = \frac{N_{is}}{N_{is}+N_{in}} \qquad \textbf{(2.28)}$$

This is the probability that an irrelevant (not used) item is recommended to the user.

2.3 Academic Social Networks

Studying the global trend in social network applications during the last decade has shown there is a need for some special sort of online social networks that enable university students, academics and scholars to showcase their research accomplishments, connect and expand their academic network. Such types of social networking website are called academic social networks. Considering their potential for collaboration, making connections, and disseminating ideas and information, academics and scholars can utilize academic social networks to improve scholarship (Szkolar, 2012). In this research, current instances of this particular kind of social network are studied and will be discussed next.

2.3.1 Introduction of Current Academic Social Networks

Academia.edu – According to its website[1] "Academia.edu is a platform for academics to share research papers. The company's mission is to accelerate the world's research." This is a popular academic social network site with over two million researchers (About Academia.edu, 2013). It is rather quick and simple to create a profile in this social network. As its most important features, scholars can upload their academic documents including resumes and publications, join scientific discussions in communities and select relevant topics to follow. Once members have registered and set up their profiles, they can find researchers with the same interests based on information submitted in their profiles. Subsequently, they are able to follow what other scholars in the same research field are working on, browse the latest research results published, and also listen to talks being given on interesting research topics. Besides, by subscribing to Rich Site

[1] http://www.Academia.edu

Summary (RSS) News, the members will be notified when anyone updates their status. As another remarkable feature, Academia.edu notifies members as soon as they are looked up on the web using specific keywords (Giglia, 2011).

Mendeley – Another successful competitor, Mendeley[1] is a free academic social network and reference manager that can help scholars to organize their research, collaborate with other researchers, and also discover the latest results in their area of research (Szkolar, 2012). Mendeley assists its members share and upload documents and encourages collaboration through motivating group features by which users can follow updates, make comments, share documents, and track progress within the groups they create. Members are able to search for papers in its crowd-source database, add papers of interest to their profile's library and write comments on peers' papers (Collaborative Features, 2013). According to an official announcement (Victor, 2012) Mendelay exceeded 2 million users in November 2012.

One of the earliest objectives of Mandeley was to develop software that automatically elicits bibliographic details from submitted publications. Thanks to this feature, researchers are not required to enter the detailed specifications of their research works by hand (Mangan, 2012). Following the free registration, users can easily set their preferences and after specifying their field of research, it is possible to develop their personal network of research contacts. The main goal of such an academic social network is finding the top members on any subject, seeing who is researching what, and staying up to date with colleagues' latest activities (Giglia, 2011).

[1] http://www.Mendeley.com

As another effective feature to enable members to have real-time insight into trends in research, Mendeley recommended scholarly publications based on their research area. Utilizing the readership metrics in the Mendeley network, members also can browse the latest statistics on top research papers, topics, authors, and journals in their academic discipline (Giglia, 2011).

ResearchGate – This scientific social networking website[1], founded in 2008, has reached 2,000,000 members (Team, 2012) who are using ResearchGate to upload their publications, share their research results, and build their worldwide reputation. According to an Economist report (Professor Facebook, 2012), the main objective of ResearchGate is to create a working and discovery network among scientists.

Each member represents their research CV in a personal profile, which contains the researcher`s preferences, educational record, projects, professional and academic experiences, publications, and contact details. Members of ResearchGate have access to a personal blog that enables them to submit interesting news and insights to a broad academic audience.

To stay connected and keep up to date, all members can follow each other`s profiles and receive the latest updates within their news feed. A messaging service additionally exists that allows users to contact each other directly. The researchers' connections are visualized through a network graph which helps users discover content on this academic social network. ResearchGate also provides the required facilities for researchers to connect through their existing profiles in other social networks such as LinkedIn, Facebook, FriendFeed, and Twitter (Giglia, 2011).

[1]https://www.researchgate.net

LinkedIn – LinkedIn[1] is known as a social network for professionals and was launched on May 5, 2003. It is currently available in nineteen languages and as of December 31, 2012 it has become the world's largest professional online network with more than 200 million members from over 200 countries and territories (About LinkedIn, 2013).

In other words, LinkedIn is truly an enormous expert database. In spite of Twitter that lacks the rich profiles, and Facebook where it is hard to find people, LinkedIn standardizes information entered by users into their profiles via predefined "Profile Headline," "Summary," "Education," "Company," and other categories. In addition to this large database of information, an effective search engine is provided that allows pinpointing the person you are looking for based on particular parameters (Schaffer, 2009).

Many professionals and experts from business domains have joined the LinkedIn social network. The connection to business environments has also helped LinkedIn play the additional role of career management media. LinkedIn members have the chance to find potential companies and recruiters, as well as be found by them.

CampusBuddy – According to the founder Mike Moradian, "CampusBuddy[2] is a social academic platform for students to connect with other classmates with information that will help them succeed in school and it allows them to basically utilize the connections with people from Facebook in an academic setting." The test site was launched in February 2008, after which a totally revamped Facebook application/website was introduced in October 2008 (Bisca, 2011).

---

[1] http://www.LinkedIn.com
[2] http://www.campusbuddy.com/

By being members of CampusBuddy, students can communicate with others from the same university and can also learn how each professor grades. To join this academic social network, students need to either register or link their Facebook accounts to the dedicated profile in CampusBuddy (Sumra, November 17, 2009). They can gain valuable information by registering to this website. Official grade distributions of 250 schools in the United States have been stored in the CampusBuddy database. The member students can effortlessly learn how difficult lecturers and courses are in their university. This feature is helpful to make better decisions in taking the most appropriate courses. The school registrars transfer this kind of information to the CampusBuddy administration team to input into the designated databases (Bisca, 2011).

Digication – The academic social network Digication[1] provides an online e-Portfolio and assessment management system for universities, k-12 schools, colleges and other professional organizations. e-Portfolios are defined as web-based platforms for lecturers, students, alumni, and professionals to publish and share their works and ideas. Some applications of e-Portolios are interactive resumes, assessments, student galleries, teaching materials assessments, and research presentations (Tochel et al., 2009).

Rhode Island School of Design (RISD) was the first place where Digication was launched in 2004 by Kelly Driscoll. The main goal of Digication is to help institutions and students to build appropriate e-Portolios to enhance the process of learning. It can also be considered a model for publishing digital content in a cost-effective and focused way (Acker, 2008).

[1] http://www.Digication.com

Classmates – Among the pioneers in social networking, Classmates was created in 1995 by Randy Conrads (Classmates.com site info, 2013). At first, it was developed to help members find classmates from kindergarten, primary school, high school, college, work and the United States military. But after years, in 2010, it began to be focused on nostalgic content such as movie trailers, high school yearbooks, photographic images, and music tracks (Bishop, 2011). Currently, it has more than 55 million registered users from over 25,000 high schools and at least 130,000 digitized yearbooks (Perez, 2012).

In order to register to Classmates.com, you need to provide your name, e-mail address, birth date and graduation year. Then you can create your personal profile and share your identifying information as well as a personal photograph. You can upgrade to a Gold member by paying a subscription fee. Gold members will be provided with additional features. With the free profile in Classmates.com you can search for other people. To contact them you need to be a Gold member. Besides, in Classmates.com, users can create reunions and events within their community and invite other members to join. It is also possible to reconnect with people from the past by e-mail and posting on message boards (Bishop, 2011).

According to user demographics of Classmates.com in the United States, the majority of members are adults, in contrast to other social networking sites such as Facebook, which has a younger average user base. Another interesting statistic shows that 85% of users are Caucasian while the share of African American members is around 8%. The remaining are Asian, Hispanic and the other races (Quantcast Audience Profile: Demographics, 2011).

CourseNetworking (TheCN) – TheCN is a new web-based product in the social networking domain with the main purpose of improving the learning process. It connects students and instructors within a classroom from around the world based on their interests and class subjects. It is a free online platform, and theCN is a simple and friendly system that enables students and instructors to share classroom materials, collaborate on assignments and stay connected worldwide. This social network was created in July of 2011 by co-inventors Ali Jafari and Indiana University. It is a private, for-profit Limited Liability Company (LLC) managed in Indianapolis, Indiana (Jafari, 2012).

Dr. Ali Jafari has stated that the primary objective of TheCN is not course management but networking that is open and free to any user all over the world. Users of this social network can share class notes, teaching materials, and collaborate with their other classmates in an online environment. "One of the biggest advantages of this new model for learning is it transforms the regular classroom into a global classroom," Jafari said. "No longer is it just you, your classmates and your teacher. Now it is you, your classmates, your teacher and your virtual classmates and teachers from all over the world. CN introduces an intercultural learning experience and offers more opportunities for educational collaboration with international universities. With this, the CN is expected to invent and introduce a totally new pedagogical framework for online learning."(Jafari, Sept. 22, 2011)

2.3.2 Feature Comparison of Current Academic Social Networks

Based on previous number of earlier research works (Rohani & Ow, 2011), academic social networks can be compared based on their general and specific features. To contrast the above-mentioned academic social networks, they were examined by navigating through their web pages and reviewing their features comprehensively. Academic social networks studied in this research are listed in Table 2.4.

Table 2.4: List of Studied Academic Social Networks

Academia (AC) LinkedIn (LK) Campus Buddy (CB) Digication (DG)	Classmates (CL) Course Networking (CN) Mendeley (MD) ResearchGate (RG)

Table 2.5 presents a list of general ASN features. Some features such as Profile Management and Friend Management are common among all of them. The Following Mechanism is provided by all except CB and DG. Searching for other members and the Rating/Recommending mechanism are not very common, only LK and CB provide them to their members. Following others and leaving offline messages are quite popular in the reviewed academic social networks. As shown in Table 2.5, most features provided by DG are not free. These kinds of characteristics that need subscription fees are recognized by a small star in Table 2.5. CN, MD, and RG provide another element for updating website information such as universities, faculties, and list of disciplines. This information is confirmed at a later date by an administration team. Some academic social networks enable their members to share their publications through their profiles. Members have the opportunity to receive relevance feedback on their research works. A point system is a way to motivate members to become more involved in social networking activities. CB and CN allocate points to their members based on their different online activities.

Table 2.5: Comparing the ASNs by Their General Features

General Features	LK	CB	DG	CL	CN	MD	RG	AC
Profile Management	✓	✓	✓*	✓	✓	✓	✓	✓
Editing Privacy Settings			✓*	✓	✓	✓	✓	
Sharing Publications			✓*		✓	✓	✓	
Personal Status Management					✓	✓		✓
File Repository Management		✓	✓*			✓	✓	
Friends Management	✓	✓	✓*	✓	✓	✓	✓	✓
Invite people to join				✓	✓	✓		✓
Forum activities	✓	✓			✓			
Following Mechanism	✓			✓	✓	✓	✓	✓
Point Mechanism		✓			✓			
Rating / Recommendation Mechanism	✓	✓						
Online Newsletter	✓			✓		✓		
Notification Emails	✓	✓			✓			✓
Offline Messaging	✓			✓	✓	✓		✓
Grouping Contacts					✓	✓		
Updating Basic Information by Members					✓	✓		✓
Search (colleges, people, professors) needs detailed description	✓	✓						

* This service needs subscription

The aforementioned facilities are classified as general features of academic social networks. Table 2.6 presents their more specific features. Almost all examined academic social networks except CB, CN, and RG provide an online dashboard for their members for illustrating the latest statistics of their activities in ASNs. A special feature is considered in CL that enables members to share memories in an online environment. MD is the only studied academic social network for Paper and Reference Management while CB and DG have paid special attention to Calendar Management and Scheduling. Course Management and e-Learning Systems have been considered in CB, DG, and CN. It is possible in CN to submit and publish e-Surveys. LK and AC inform their members about jobs related to their preferences. Universities, companies and other organizations are allowed to have a specific Webpage in LK and DG. CB is the only ASN that makes it possible for members to access official grading records in the context of course management. Finally, members of LK and CL can check who has visited their profile.

Table 2.6: Comparing the ASNs by Their Special Features

Special Features	LK	CB	DG	CL	CN	MD	RG	AC
Calendar management and scheduling		✓	✓*					
Paper Review Mechanism						✓		
Group Assignment Follow up		✓			✓			
Course Management		✓	✓*		✓			
e-learning systems			✓*		✓			
Online Survey					✓			
Reference management tools						✓		
Sharing memories				✓				
Dashboard	✓		✓*	✓		✓		✓
Posting jobs	✓							✓
Access to Official Grading Records in Context of Course Management		✓						
Class Status Update		✓			✓			
Profile Organizer	✓*					✓		
Pages for Companies or Universities	✓		✓*					
Profile Visitors	✓*			✓				

By studying the elicited features of academic social networks (Tables 2.5 and 2.6), a list of academic items is generated (Table 2.7) which are presented to members of such online social networks.

Table 2.7: List of Academic Items Presented in Academic Social Networks

• Information about Articles
• Course Materials
• Surveys
• Job Opportunities
• Course Grading Records
• Conference Information
• Scholarships
• Scientific News
• Scientific Forum Topics

2.3.3 Statistical Comparison of Current Academic Social Networks

Table 2.8 presents some statistical information on the most popular academic social networks studied in this research. According to Alexa website (Alexa Site Info, 2013) the oldest is Classmates, which was created in 1995, followed by Digication and other academic social networks, most of which were launched in 2008. The most popular academic social network among these is LinkedIn with more than 200 million members. According to Alexa Site Info (2013), it is visited mostly by users from the United States (25.2%), India (15.1%), the United Kingdom (5.0%), Spain (3.8%), France (3.6%), Canada (3.3%), Brazil (3.0%), the Netherlands (2.8%), Australia(2.4%), and Germany(1.6%).

Table 2.8: Statistical information of ASNs

ASNs	Traffic Rank	Number of Users	Page Views Per User	Average Time on Site	Creation Year
Academia.edu	3,708	> 2,200,000	2.61	2:36	2008
Mendeley	23,074	> 2,000,000	3.3	3:12	2008
ResearchGate	6,412	> 2,000,000	2.6	2:41	2008
LinkedIn	14	> 200,000,000	8.77	7:21	2008
CampusBuddy	292,369	NA	4.9	2:21	2008
Digication	249,758	NA	6.7	8:25	2004
Classmates	1,401	> 55,000,000	2.92	2:47	1995
TheCN	3,376,475	NA	2.5	3:48	2012

Statistics for Average Time on Site shows that members of LinkedIn and Digication spend more time in these websites than members of the other academic social networks. These two websites also have the highest rate of Page Views per Users with 8.77 for LinkedIN and 6.7 for Digication. LinkedIn has the best traffic rank with 14, while Classmates (1,401), Academia.edu (3,708), and ResearchGate (6,412) follow at a long distance. From this viewpoint, TheCN is the newest and has the lowest traffic rank (3,376,475) in Alexa.

2.4 Summary

With the explosion of Web 2.0 applications such as blogs, social networks, and various other types of web-based applications, the rich online information and various new sources of knowledge flood users and hence pose a great challenge in terms of information overload. Considering this phenomenon, it is essential to use recommender systems to assist users in finding the right information from an abundance of web data (Zhou, et al., 2012). The past decade has seen the rapid development of recommender systems in both research areas and online business domains. Recommender systems are defined as software tools and techniques for suggesting the most related items to users. Playing the most important roles by recommender systems in high ranked websites, conducting international conferences dedicated to this field, and recently added RS courses in famous universities are all witnesses of the increasing interest for this field of research. According to earlier research, the collaborative and content-based recommender systems were rather successful in suggesting some relevant items to target users, but they did suffer from some shortcomings such as sparsity, recommending new items, and the cold start problem for new users. More specifically, in academic social networks studied in the present research, the cold start problem is considered with more attention since the members of these online environments are recommended with new items.

Although some recent alternatives like hybrid methods, demographic algorithms, and knowledge-based approaches have been proposed to mitigate these problems, the current generation of recommender systems surveyed in this study still requires further improvements to make recommendation methods more effective. In other words, the traditional recommender systems ignored social relationships among users. But in real life, when we ask friends for recommendations of a nice restaurant we are actually requesting verbal social recommendations (Bonhard, 2005). In another research in this context, Bonhard, Sasse, and Harries (2007) stated that recommender systems and social networking functionality should be integrated. To fill this gap, Seth, et al. (2008) proposed and evaluated a recommender system based on a Bayesian user-model. They used the underlying social network of blog authors and readers to model the preference features for individual users. As a potential technique in this context, friends' preferences can be considered in addition to a given user's own preferences to improve the accuracy of predictions. Consequently, in order to improve recommender systems and to provide more personalized recommendation results, it is necessary to incorporate social network information among users (Zhou, et al., 2012).

In this chapter, we reviewed various advantages and limitations of the recommendation systems and discussed possible augmentations that can help to develop better recommendation capabilities. An overall glimpse of recommender systems is illustrated in Table 2.9. For more clarification, the commonly used techniques and related research works are listed for each category.

Table 2.9: Classification of Recommender Systems Research

Recommendation Approach	Recommendation Technique
Collaborative	**Commonly used Techniques:** • Nearest neighbor (cosine, correlation) • memory-based (user-based) • model-based (item-based) • Clustering • Graph theory • Bayesian networks • Artificial neural networks • Linear regression • Probabilistic models **Representative research examples:** • Resnick et al. (1994) • Breese et al. (1998) • Ungar& Foster, (1998) • Sarwar et al. (2001) • Schafer et al (2001) • Schein et al. (2002) • Linden et al. (2003) • Hofmann, (2004) • Lam and Riedl, (2004) • Deshpande et al. (2004) • Chirita et al. (2005) • Wang et al. (2006) • Mobasher et al. (2007) • Sandvig et al. (2008) • Cacheda et al. (2011)
Content-Based	**Commonly used Techniques:** • TF-IDF (information retrieval) • Clustering • Bayesian classifiers • Decision trees • Artificial neural networks **Representative research examples:** • Shardanand and Maes, (1995) • Holte and Yan, (1996) • Pazzani et al, (1996) • Balabanović and Shoham, (1997) • Pazzani and Billsus, (1997) • Mitchell, (1997) • Mooney et al. (1998) • Billsus and Pazzani, (1999) • Mladenic, (1999) • Herlocker, (2000) • Picard, (2003) • Adomavicius and Tuzhilin (2005)

	• Chowdhury, (2010) • Cermonesi et al. (2011) • Lops et al. (2011) • Davenport, (2012)
Hybrid	**Commonly used Techniques:** • Linear combination of predicted ratings • Various voting schemes • Incorporating one component as a part of the solution for the other • Building one unifying model **Representative research examples:** • Balabanović&Shoham (1997) • Claypool et al. (1999) • Pazzani (1999) • Soboroff& Nicholas (1999) • Ansari et al. (2000) • Hofmann, (2001) • Popescul et al. (2001) • Burke, (2002) • Melville et al (2002) • Schein et al. (2002) • Miller et al. (2003) • Middleton et al. (2004) • Durao&Dolog, (2010) • Ghazanfar&Prugel-Bennett, (2010) • Jung, (2011) • Ricci, et al. (2011) • Porcel et al. (2012) • Verhoeyen et al. (2012)
Demographic	**Commonly used Techniques:** • Demographic modeling • Segmentation • Self-organizing map (SOM) **Representative research examples:** • Mahmmod et al. (2007) • Mosayebian et al. (2012)
Knowledge-Based	**Commonly used Techniques:** • case-based modeling • constraint-based modeling **Representative research examples:** • Bridge et al. (2005) • Ricci et al. (2006) • Martinez et al. (2008)

Social Network-Based	**Commonly used Techniques:** • Social Network Models • Friends' preferences • Bayesian User-Model • Probabilistic Model • Preference Scoring **Representative research examples:** • Massa et al. (2004) • Smeaton et al. (2005) • Golbeck, (2006) • Bonhard, et al. (2007) • Groh et al. (2007) • Seth, & Zhang, J. (2008) • Arazy et al. (2009) • Guy et al. (2009) • He & Chu, W. W. (2010) • Zhou, et al. (2012)

Academic social networks are a special kind of social networking websites that enable their members to collaborate, make connections, and share their ideas and information in a web-based environment. To obtain a comprehensive picture of current academic social networks, the eight most popular samples are studied in this research, namely Academia.edu, LinkedIn, Campus Buddy, Digication, Classmates, Course Networking, Mendeley, and Research Gate.

According to Alexa website (Alexa Site Info, 2013) the oldest academic social network is Classmates, which was created in 1995, followed by Digication and other academic social networks most of which were launched in 2008. The most popular academic social network among these is LinkedIn with more than 200 million members. LinkedIn has the best traffic rank of 14 while Classmates (1,401), Academia.edu (3,708), and ResearchGate (6,412) follow it at a distance. TheCN is the newest and has the least traffic rank (3,376,475) according to Alexa.

The last part of this literature concentrates on introducing the measurements used for the evaluation of recommender systems. According to what exactly they measure, these metrics are categorized into three main groups: measuring the accuracy of rating prediction, measuring the usage prediction, and measuring the rankings. As will be discussed later, the usage prediction metrics have been used in this research to evaluate the accuracy of predictions in different recommender algorithms. The details of three-phase research methodology of this study are presented in the next chapter.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

Recommender systems play an important role in the success of social networks by mitigating the information overloading problem. The systems can suggest the most relevant and attractive items to users based on their behavior and preferences (Ido Guy & Carmel, 2011). Academic social networks are a specific type of social networks that provide web-based services to academicians and scholars. These were pointed out in the literature part of this study. In the second part of Chapter 2, the main recommender algorithms were described and compared according to their advantages and shortcomings. In addition, various means of evaluating the prediction accuracy of recommender systems were presented at the end of Chapter 2. In the present chapter, the three-phase methodology applied in this research is discussed and the target population consisting of 920 MyExpert members from 10 universities in Malaysia is introduced. The final section of this chapter provides a detailed schedule of the data collection procedure.

3.2 Research Design

A suitable, well-structured method is essential to performing sound empirical research. Empirical research methods are a class of research methods in which empirical observations or data are collected in order to test a theory (Easterly & Levine, 2001). In the present study, a quantitative method serves to test this theory of whether utilizing social networking parameters can improve the performance of content-based recommender systems in academic social networks. This section describes the three-phase method applied in this investigation (Figure 3.1).

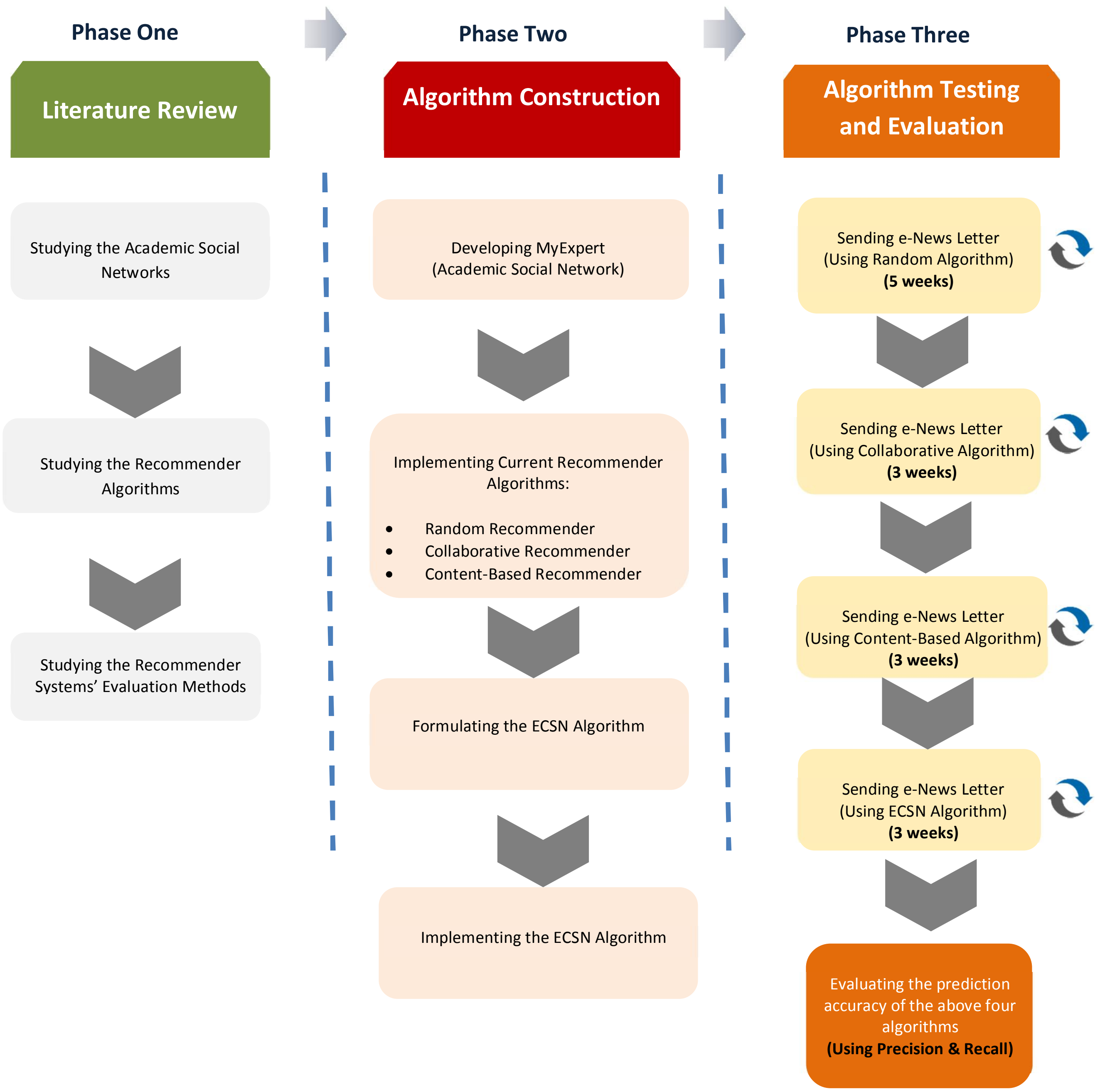


Figure 3.1: Research Methodology

The first research phase centers on a literature review and study of preceding research works done on recommender systems. This phase has three stages: studying the academic social networks, studying the recommender algorithms, and studying the recommender systems' evaluation methods.

With reference to the principal research objective that deals with improving recommender systems in academic social networks, the most popular academic social networks were first analyzed comprehensively. Eight websites of this domain were briefly examined from varying points of view. This component of the literature review has led to generating a list of general and specific features of academic social networks, as mentioned in Chapter 2.

The next stage of the first phase contains a detailed description of the most important recommender systems. According to some vastly cited researchers (Adomavicius & Tuzhilin, 2005; Ricci, et al., 2011), the collaborative and content-based methods seem to be the most prevalent recommender systems that have become origin points for other proposed algorithms in this field. The second literature review section is allocated to providing a comprehensive picture of recommender systems.

The last literature review section introduces the measurements done for analyzing recommender systems. These metrics fall into three main groups based on what exactly they measure: measuring the rating prediction accuracy, usage prediction, and rankings – all of which are discussed in the last section of Chapter 2.

The core focus of the second research phase deals with the design and development of the required test environment, recommender algorithms and the ECSN model (Figure 3.1). First, an online real environment (MyExpert) was developed for testing the various recommender systems and comparing their accuracy in recommending the most relevant items to users.

The implementation and testing of recommender systems considering all their details are among the most essential aspects of this research. As depicted in Figure 3.1, during the second stage of the construction phase, the following three recommender algorithms were initially implemented in the MyExpert environment:

- Random Recommender Algorithm
- Collaborative Recommender Algorithm
- Content-Based Recommender Algorithm

In the following stage, an ECSN algorithm was designed and formulated as the proposed recommender model to enhance the functionality of content-based algorithms. By implementing the ECSN algorithm, the final step of the second phase was accomplished.

The main objective of the third research phase is to test and evaluate all four implemented recommender systems and compare their prediction accuracy. In doing so, the online study approach was applied as the strongest experimental method (Shani & Gunawardana, 2011) for evaluating the performance of recommender algorithms. To achieve this goal, 1390 records of academic items were submitted in MyExpert, including 346 academic jobs, 339 conferences, 355 scholarships, and 350 academic news articles. As a follow-up to the data gathering schedule of this research, each of the four recommender systems examined was used to send the top 10 academic items to MyExpert members over 14 consecutive weeks.

The literature review indicates that the collaborative and content-based approaches are the most popular algorithms in recommender systems. These two essentially form a base for other recommender algorithms. For this reason, the decision to use the collaborative and content-based methods was made for recommending academic items to MyExpert members. Before applying these approaches, a series of previous ratings in the MyExpert environment were required. To collect user ratings, the random recommender algorithm was implemented and applied in MyExpert for the first five weeks of experimentation. Subsequent to gathering user ratings with the random algorithm, the collaborative and content-based algorithms were run for six weeks.

During the first 11 weeks of experiments, MyExpert users experienced three different recommender algorithms (random, collaborative, and content-based) and the ratings related to each algorithm were collected. Then the main algorithm (ECSN) was applied for recommending academic items to users. Similarly, users of MyExpert encountered the functionality of this algorithm for three consecutive weeks. The implementation particulars of all four algorithms will be provided in Chapter 4.

After gathering the members' feedback from the 14 weeks, Precision, Recall, Fallout, and F1 assessed the prediction accuracy of all recommender algorithms applied. The details of the evaluation process and its results are presented in Chapter 5.

3.3 Target Population

To take advantage of the experimental accuracy from online studies, the MyExpert academic social network was designed and developed as the runtime environment for this research. As the first academic social network in Malaysia, it has so far successfully motivated 920 academicians from 10 universities and higher-education institutes to join this social network (Figure 3.3). A log generator component was developed and embedded in MyExpert, which has generated over 80,000 user transaction records from online user-system interactions.

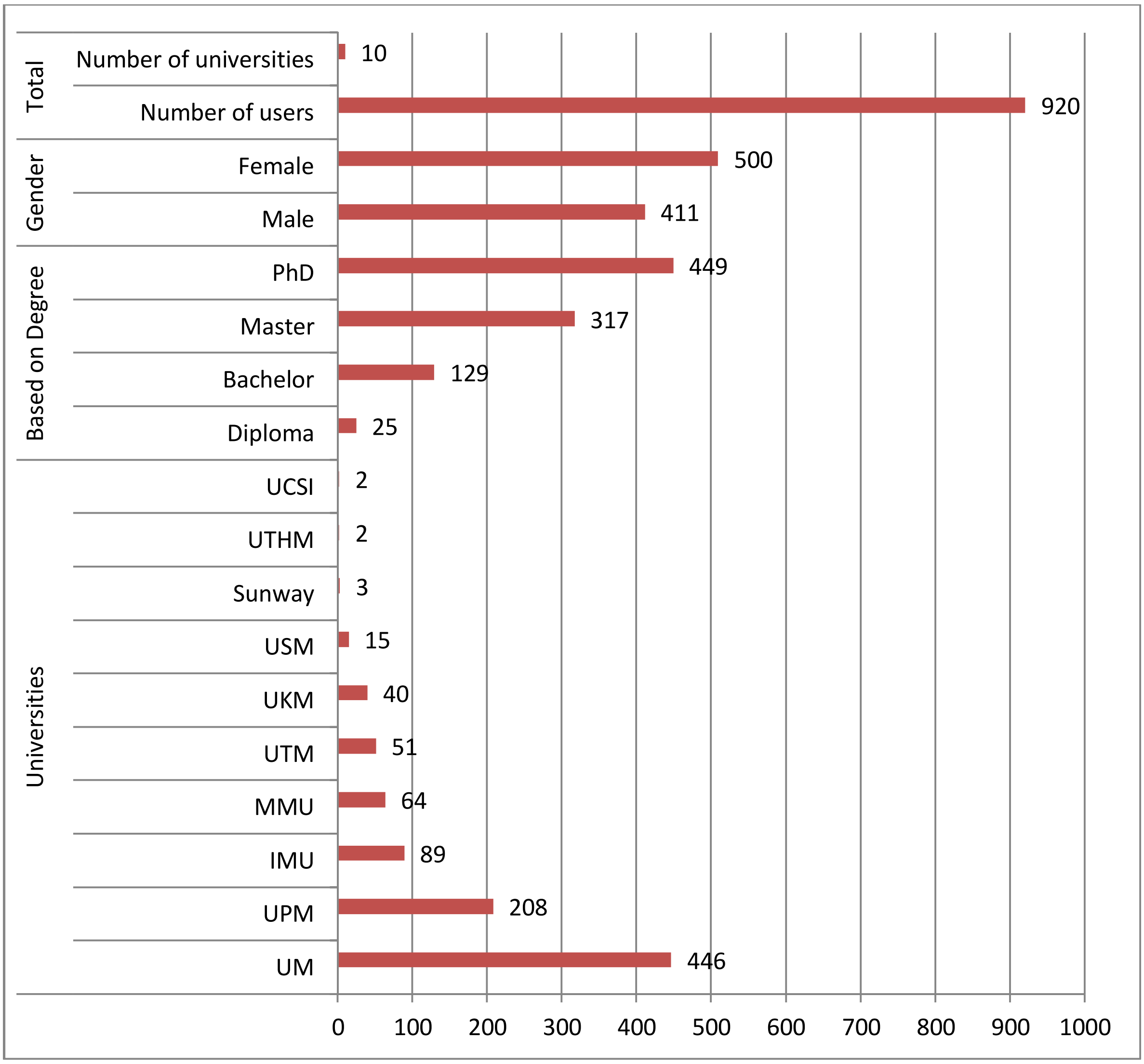


Figure 3.2: Target Population Demographics

As illustrated in Figure 3.2, the majority of MyExpert members are postgraduates (449 PhD and 317 Masters). University of Malaya (UM) is the highest ranking university in Malaysia and has contributed the most members (446) to MyExpert, followed by UPM with 208 members. Other universities, such as IMU (89 members), MMU (64 members), UTM (51 members), UKM (40 members), and USM (15 members) contributed fewer members than the first two. UCSI (2 members), UTHM (2 members) and Sunway (3 members) have the least share in MyExpert (Figure 3.3). Regarding gender, 55% of users are female and 45% are male.

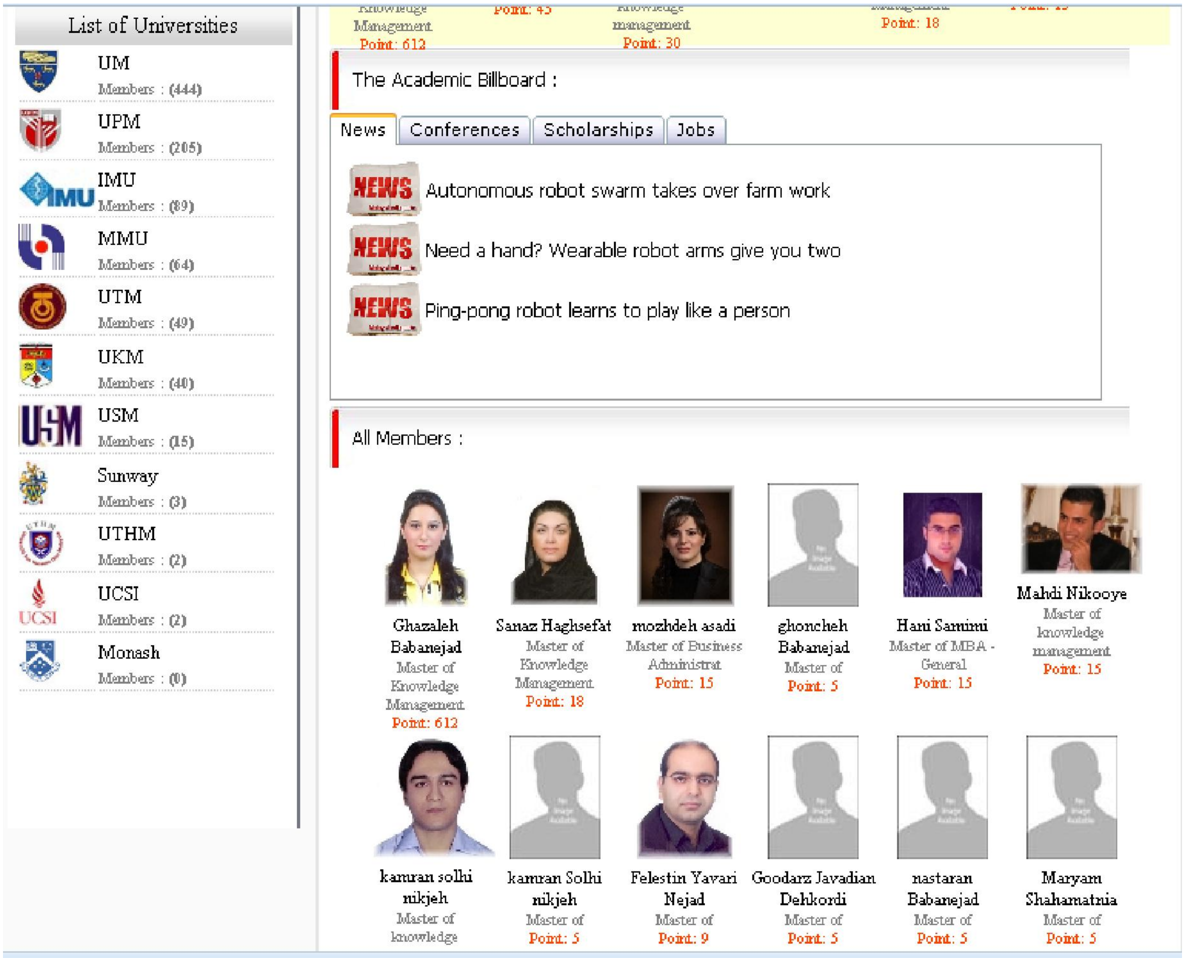


Figure 3.3: List of Universities in MyExpert

3.4 Data Collection Procedure

This research endeavors to upgrade the prediction accuracy of content-based recommender systems in academic social networks. According to the problem statement given in Section 1.2, the most relevant academic items are to be sent to MyExpert users through an e-newsletter. The users who receive item recommendations in their inbox click on the links of interest for academic items and may rate them by selecting one to five stars from the top right of the related webpage (Figure 3.4). By recording user interactions with MyExpert, relevant feedback is collected for the evaluation process.

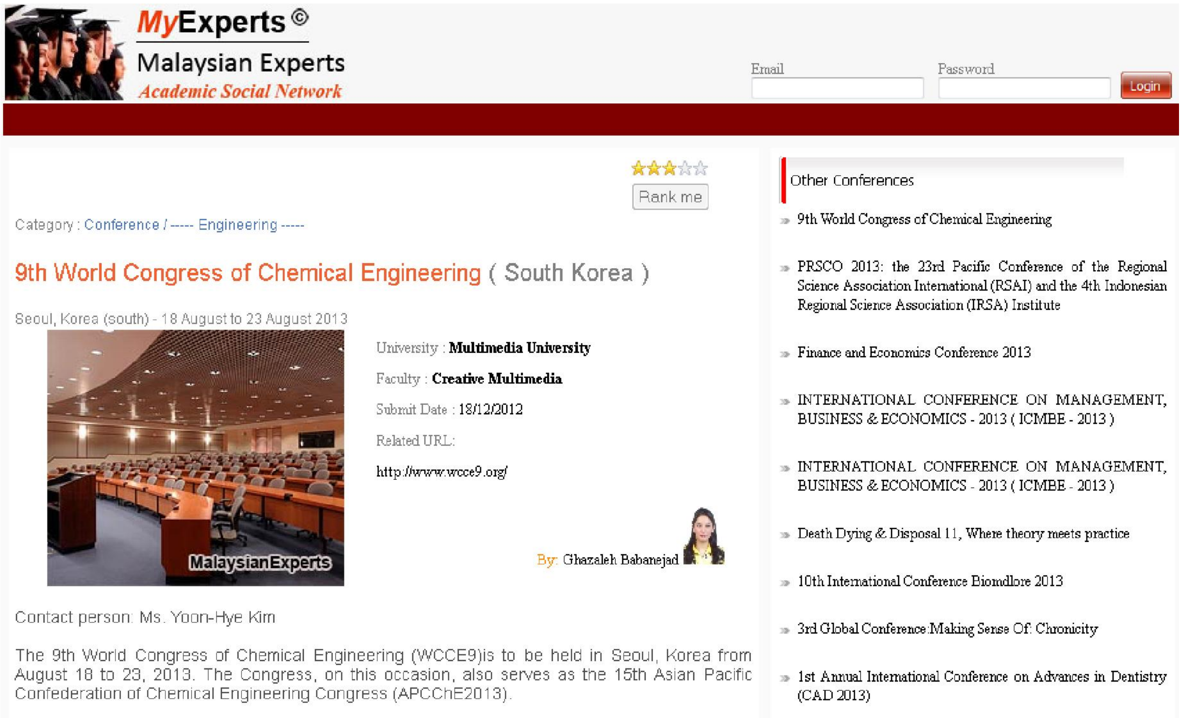


Figure 3.4: The Academic Items Webpage in MyExpert

The data collection for this research was carried out over 14 consecutive weeks from 7th September to 26th December 2012. Each week, 100 academic items including 25 news, 25 conferences, 25 scholarships, and 25 job offers were submitted to MyExpert via the online webpages designated for this purpose. After completing the weekly submission process, every MyExpert member would receive the top 10 items from a total of 100 in their email. Users expected to receive the most relevant items through MyExpert academic e-newsletter.

Throughout the 14 weeks, four recommender algorithms, namely random, collaborative, content-based, and ECSN were applied for selecting and recommending the most relevant academic items to MyExpert users. The first five weeks were dedicated to the random algorithm since the other three require previous ratings to work with. After gathering the applicable feedback with the random algorithm, the collaborative algorithm attempted to recommend the top 10 items to users. It took three weeks to collect records of user behavior for this algorithm. Immediately after, the content-based algorithm was applied for the next three weeks, from 12th November to 3rd December 2012. The final and most important component of data collection was allocated to the ECSN algorithm. Basically, this proposed algorithm was examined over the last three weeks, from 4th December to 26th Dec 2012.

3.5 Evaluation method and measurements of this study

The prediction accuracy measures are classified in three categories of Ratings Prediction Accuracy, Usage Prediction, and Ranking Measurements (Shani & Gunawardana, 2011). In some experimental environments for evaluating recommender systems, users prefer to simply click on recommended items rather than rank them. In such situations, the usage prediction measures should be considered for calculating the prediction accuracy of recommender algorithms (Herlocker, et al., 2004; B. Sarwar, et al., 2001; Shani & Gunawardana, 2011).

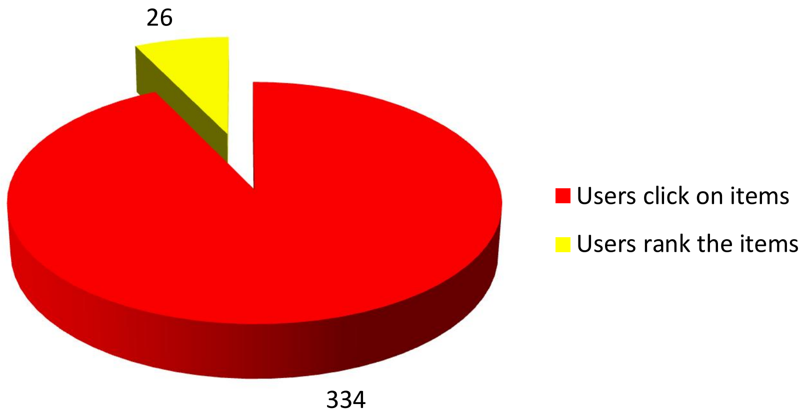


Figure 3.5: User clicks and ranks shares in the MyExpert experiment

In the 14 weeks of gathering data for this research, the academic e-Newsletters were sent to 920 MyExpert users. According to statistics (Figure 3.5), 360 persons visited the academic items while only 26 members ranked the items and the remaining 334 preferred to visit by clicking on the title links sent to their email addresses. Considering the fact that in this research 92% of users did not rank the recommended items, Usage Prediction Measures was employed instead of Ratings Prediction Accuracy and Ranking measures to assess the prediction accuracy of recommender algorithms (Herlocker, et al., 2004; B. Sarwar, et al., 2001; Shani & Gunawardana, 2011).

Consequently, Precision, Recall, Fallout, and F1 were used in this research to measure the usage prediction of each recommender algorithm. During each week of the data gathering phase, a fixed number of 10 items from 100 newly entered academic items were selected for recommendation to each MyExpert member. Also, referring to previous researchers (B. Sarwar, et al., 2001; Shani & Gunawardana, 2011), the above-mentioned metrics were computed at each recommendation list for every user, and then the average value was calculated to compare the prediction accuracy of every recommender algorithm.

The experimental results for this research work are presented based on four usage prediction measurements, namely Precision, Recall, Fallout, and F1. To further clarify the concept behind these measurements, the space of all possible items considered for recommendation to MyExpert users in this research are depicted in Figure 3.6.

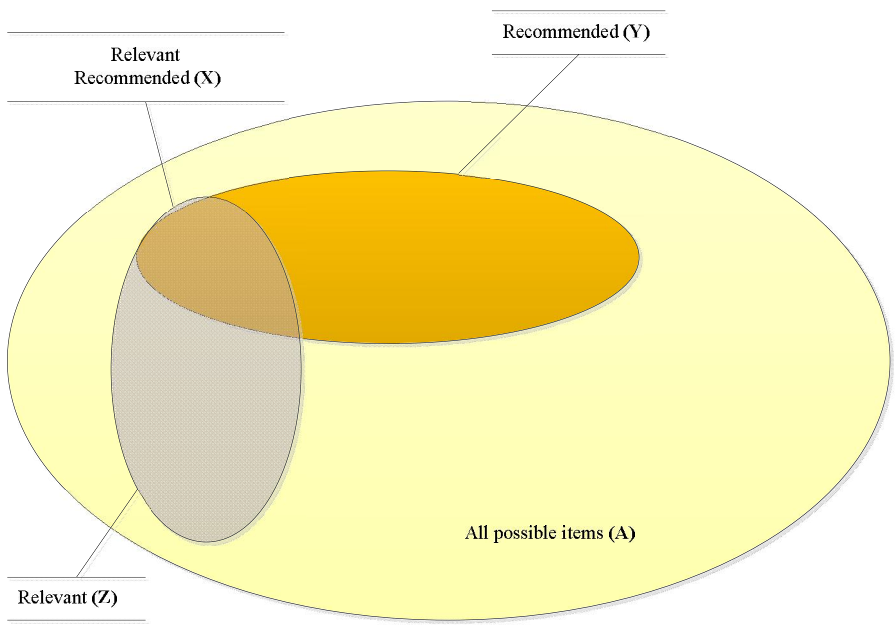


Figure 3.6: The Space of All Possible Items Considered For Recommendation to MyExpert Users

All possible items include 100 academic items consisting of academic news, conferences, jobs, and scholarships submitted to MyExpert during each week of experiments. Every implemented recommender system in this research aimed to identify the top 10 items (Y) for all members and recommend them through a MyExpert e-newsletter. Hence, in 14 weeks of experiments, 10 items (Y) were recommended to every MyExpert member from roughly 100 possible items (A). The MyExpert members had two ways of accessing their favorite academic items: by clicking on the hyperlink sent by the MyExpert e-Newsletter and by clicking on news links as other news were available at the right side of the news web page. In the former case, the viewed item appeared as Relevant Recommended (X), and in the latter case, it was shown as Relevant Not Recommended (Z-X). As per Figure 3.6, the related measurements are defined as:

$$Precision = \frac{X}{Y} \tag{3.1}$$

$$Recall = \frac{X}{Z} \tag{3.2}$$

$$Fallout = \frac{Y-X}{A-Z} \tag{3.3}$$

3.6 Summary

This research was conducted using a three-phase methodology. The first phase comprises a review of literature with focus on academic social networks, recommender algorithms, and means of evaluating recommender system performance. The next phase involves the design and development of MyExpert as a runtime environment and applying four different recommender algorithms to be evaluated in the MyExpert online environment. Upon realizing the development phase, the performance of the proposed recommender algorithm in this research (ECSN) was compared against the other three algorithms applied, i.e. random, collaborative, and content-based. This process took 14 consecutive weeks and the four measurements were precision, recall, fallout, and F1.

MyExpert members were considered the target population for conducting this research and evaluating the performance of the ECSN recommender algorithm. The population consisted of 920 students, lecturers and academicians from 10 universities in Malaysia who received the recommended academic items in their inbox through the MyExpert e-newsletter. UM (446 members), UPM (208 members), IMU (89 members), MMU (64 members), UTM (51 members), UKM (40 members), USM (15 members), Sunway (3 members), UCSI (2 members), and UTHM (2 members) were the 10 participating universities in this study.

Data collection for the current research was done throughout 14 successive weeks from 7th September to 26th December 2012. Each week, 100 academic items were submitted to MyExpert. After completing the submission process for every week of the data collection phase, the recommender systems suggested the top 10 items to users and sent them through the MyExpert academic e-newsletter. In this research, four different recommender algorithms were evaluated during the 14 weeks of gathering relevance feedback. Next chapter presents the theoretical framework of ECSN recommender algorithm. Also the details of design and implementation for all four applied recommender algorithms (Random, Collaborative, Content-based, and ECSBN) are discussed in following chapter.

CHAPTER 4

DESIGN AND CONSTRUCTION OF RECOMMENDER ALGORITHMS

4.1 Introduction

The previous chapter discussed the research methodology utilized in this study. This chapter focuses on presenting the theoretical framework, architecture, and technical issues of this research. To prepare the research environment for ECSN implementation, two components were required for design and development. The first component comprises four recommender systems, including random, collaborative, content-based and ECSN, and the second focuses on MyExpert academic social network which is needed for runtime environment of this study. The design and implementation details with respect to these elements will be presented next.

4.2 ECSN Algorithm

The ECSN recommender algorithm manages user preference scores for academic item categories in a tree data structure. To accomplish this, a hierarchy of items should be defined, as per Figure 4.1.

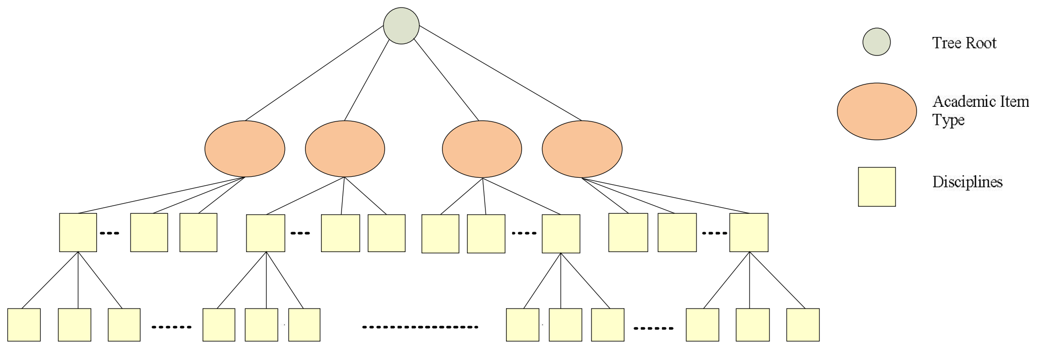


Figure 4.1: Hierarchy Tree of Academic Items Used in the ECSN Recommender Algorithm

Hence, in the ECSN algorithm user preference scores for each academic item category are stored based on the following definition:

Definition 4.1: The preference tree $PT(i)$ of user i is isomorphic to the hierarchy tree of item categories, and the set of nodes of the preference tree $PT(i)$ is as follows:

$$PT(i) = \{ (UID, ICID, PS)\},$$

where $UID, ICID,$ and PS are the user identifier, the item category identifier of the hierarchy tree, and the preference score, respectively.

Definition 4.2: The preference scores (PS) are defined as follows:

$$PS(i,j) = \propto_1 \times SelfClickScore(i,j) + \propto_2 \times SelfRankScore(i,j) + \propto_3 \times FacultyMatesScore(i,j) + \propto_4 \times FriendsScore(i,j)$$

where $\mathrm{PS(i,j)}$ is the total preference score of user i for the academic item category node j. Each element of this definition is described as follows:

$SelfClickScore(i,j)$ is the score related to clicks of a given user i for the item category node j , which is specified by counting the number of clicks for a given customer during the research experiments.

$SelfRankScore(i,j)$ is calculated by adopting the submitted rates of given user i to academic items classified in category node j.

The value of $FacultyMatesScore(i,j)$ is calculated by considering the top 3 interesting item nodes of the preference tree among members registered in the same faculty that the given user i belongs to.

The last element $FriendsScore(i,j)$, is dedicated to preferences of friends for a given member i. The strategy described above for calculating the $FacultyMatesScore(i,j)$ is adopted here for $FriendsScore(i,j)$ with the difference that it takes into account the top 3 item nodes that are most interesting among a given member's friends.

As mentioned in previous works (J. W. Kim et al., 2006), some weights may be assigned to each parameter of the formula to compute the preference scores. Accordingly, in Definition 4.2, ($\propto_k$) represents the relative weights for each element. As $SelfClickScore$ and $SelfRankScore$are the most important personal elements that should be counted for a given user i, the value of 5 is considered for $\propto_1$ and $\propto_2$. Relatively, the weight of 3 has been considered for $\propto_3$ since $FacultyMatesScore$ is less significant than the user's own preferences. Finally, $\propto_4$ is set at 1 as $FriendsScore$ has the lowest influence in Definition 4.2. The assigned weights are subjective values for considering the levels of importance among users' own preferences, their faculty mates and friends. Although the experimental results of this study indicate that these settings work well in improving the prediction accuracy of recommendations, but as the future works, even these weights might be optimized by applying some other techniques such as genetic algorithms.

After calculating the preference scores (PS) for each user i, Definition 4.3 is applied for some non-leaf nodes of the preference tree whose values are still 0.

Definition 4.3: The preference scores (PS) of a non-leaf-level product category j are defined as follows:

$$PS(i,j) = Average_{\ k \in \{k | k\ is\ a\ child\ node\ of\ product\ category\ j\}} PS(i,k)$$

In the initialization stage, the preference tree of a certain user i is initialized to 0 once the user creates a profile in the academic social network (in this study it is MyExpert):

$$PS(i,j) = 0\ _{j \in \{j | j\ is\ a\ node\ in\ preferences\ score\ tree\ structure\ for\ academic\ item\ categories\}}$$

While users open or rate the web pages of academic items, the preference score must be updated. The values given in Table 4.1 serve in updating the preference scores.

Table 4.1: Illustration of Assigned Points to Different Rates

Given Rate	Description	The Rate Value (RV)
★★★★★	Excellent	+3
★★★★☆	Good	+2
★★★☆☆	Fair	+1
★★☆☆☆	Not Bad	-1
★☆☆☆☆	Weak	-2

(1) When the given user *i* rates the academic items related to category node *j*:

$$PS(i,j).SelfRankScore = PS(i,j).SelfRankScore + RV$$

(2) When the given user *i* clicks on the academic items related to category node *j*:

$$PS(i,j).SelfClickScore = PS(i,j).SelfClickScore + 1$$

The above update procedure does not require updating of preference scores for all nodes of the tree, but rather the updating of the preference scores of nodes related to visited and rated items.

After updating $SelfRankScore$ and, the preference scores (PS) should be updated by considering faculty mates and friends of given user *i*:

$$PS(i,j).FacultyMatesScore = PS(i,j).FacultyMatesScore + Average\ (FS)$$

Where

$$FS\ \in \{Top\ 3\ Preferences\ scores\ of faculty\ mates\ for\ given\ user\ i\ \}$$

Similarly, the $FriendsScore$ value is updated as:

$$PS(i,j).FriendsScore = PS(i,j).FriendsScore + Average\ (FS)$$

Where

$$FS\ \in \{Top\ 3\ Preferences\ scores\ of friends\ for\ given\ user\ i\ \}$$

As mentioned in above formulas, Average (FS) is the average value of top 3 preferences scores which were assigned to product category j by friends or faculty mates of target user (i). In this study, top 3 scores were considered instead of all recorded scores to make the proposed algorithm more applicable in real situations facing to millions of items and users. As another reason, considering assigned weights in Definition 4.2, the preference of faculty mates and friends are mostly effective in cold-start situations when there is not enough preferences for target user. In such conditions, for making recommendations, it is preferred to find the top items which are most interesting for friends and classmates.

In each week of experiments, 100 academic items were submitted to the MyExpert academic social network. Each studied recommender algorithms aimed to select the top 10 items for each user and recommend it through an e-newsletter. For the first three algorithms, the selection process was implemented based on recommender algorithms that were studied through the literature review. In the random recommender algorithm, 10 random items were selected. The collaborative algorithm made predictions based on items that people with similar preferences and interests previously preferred (Cacheda, Carneiro, Fernández, & Formoso, 2011; Wang & Yang, 2012). For implementing the pure content-based recommendation, the preference three approach (Kim, et al., 2006) was followed. An enhanced selection process was finally used in the ECSN recommender algorithm as illustrated in Figure 4.2.

for each $\boldsymbol{u} \in \boldsymbol{U}$

{

1. Generating the ordered stack of item categories ($\boldsymbol{ItemStack}$) based on $\boldsymbol{PS}$ value computed by Definition 4.2 and 4.3.
2. $\boldsymbol{SIC} \leftarrow \boldsymbol{0}$ // Initializing the Selected Items Count (SIC) by 0
3. $\boldsymbol{SC} \leftarrow \boldsymbol{0}$ // Initializing the selected category (SC) by 0
4. $While\ (\boldsymbol{SelectedItems} < 10)$
5. {
6. $TopItemCat \leftarrow POP\ (\boldsymbol{ItemStack})$,
7. $TopItemsList \leftarrow FindNewItems(\boldsymbol{TopItemCat}, \boldsymbol{PrioritizedCount}(\boldsymbol{SC}))$
8. $SIC -= count(\boldsymbol{TopItemsList})$
9. $Adding\ TopItemsList\ to\ RecommendationList$
10. }

}

Figure 4.2: ECSN Selection Process

For each MyExpert user, the item categories are ordered according to PS value and stored in a stack data structure ($ItemStack$) such that the category with the biggest PS is accessible at the top of the stack. To produce the recommendation list for user u, the topmost category is moved to $TopItemCat$ using $POP\ (ItemStack)$. Then the newly submitted items in MyExpert (100 items per week) are searched to find academic items with the category ID $TopItemCat$. As mentioned in the research methodology, the recommendation list meant to be suggested to each user u includes the 10 most relevant items. To obtain more items with the highest PS value in this list, the $PrioritizedCount$array has been considered to identify the number of items that should be for each top-scored item category:

$$int\ PrioritizedCount = \{3,2,2,1,1,1,1,1,1,1\}$$

Based on this identified priority, the highest scored category may contain up to 3 items while the two next highest ones come with 2 items most in the recommendation list.

The others have the same value of one item. In the $while\ loop$ body, the ordered recommendation list ($TopItemsList$) is generated for each user $u \in U$.

To conclude, Definition 4.2 clearly shows that in the ECSN algorithm, the effective preferences that must be considered when finding the most relevant items to a given member are driven from three variant sources: member's own preferences, their faculty mates' interests, and friends' preferences. In this way, the cold start problem of other recommender systems addressed in the literature review section of the dissertation (Chapter 2) will be solved. Even if the recommender system does not have any records of previous transactions for a given member, friends' preference records can be considered in Definition 4.2. Furthermore, if the member has no friends to be considered for this calculation, there are definitely some faculty colleagues whose preferences would be applied in Definition 4.3 to obtain the items most relevant to the target member.

4.3 ECSN Architecture and Workflows

Figure 4.3 shows the principal workflows within the ECSN Recommender System. The architecture for this research work is identified in three main spaces: User space, Administrator space, and the ECSN recommender engine. Besides, a database is taken into account in this architecture, used for restoring and retrieving related data. The functionality of this system is briefly described subsequently based on the three different working spaces.

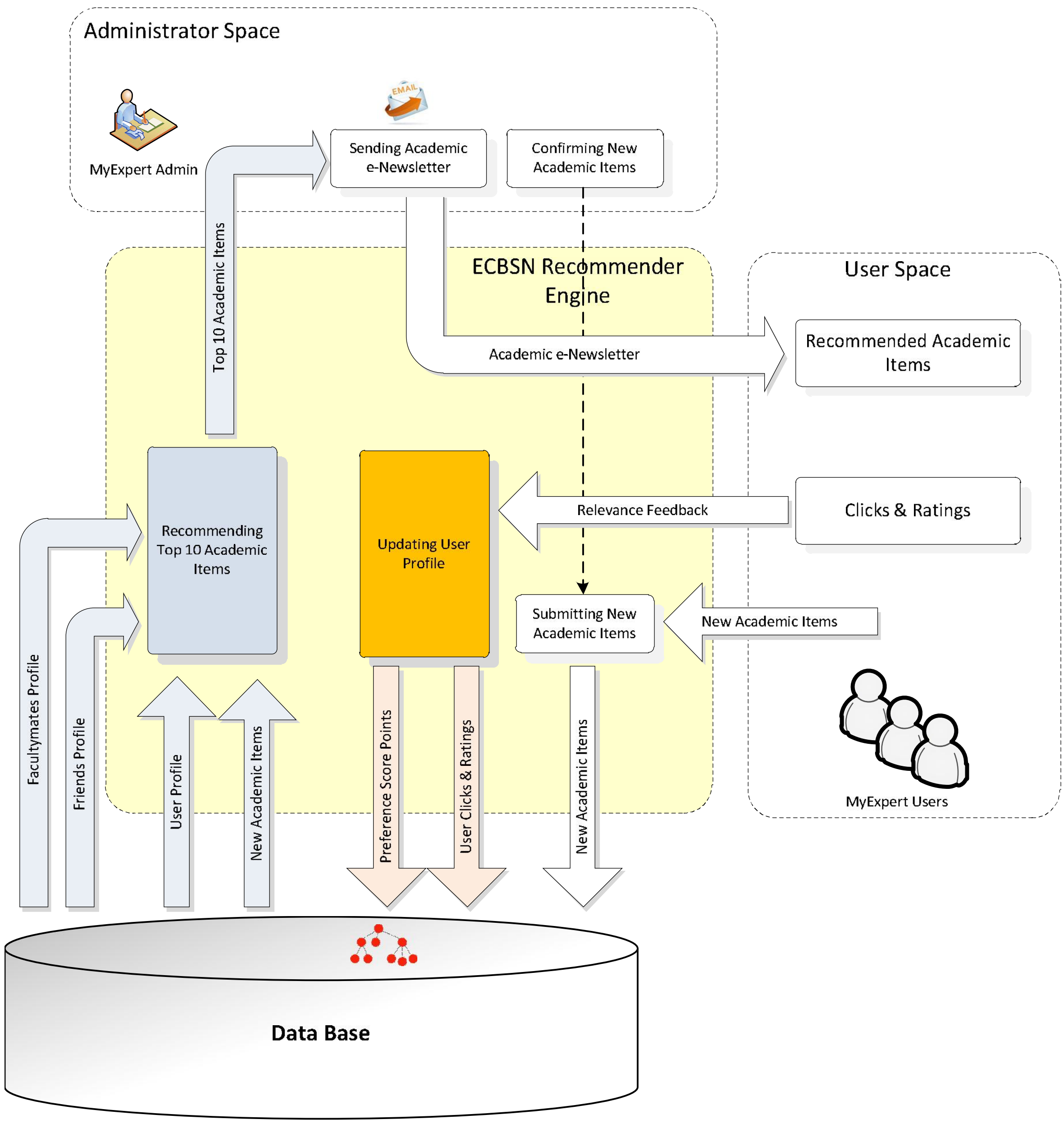


Figure 4.3: The architecture and workflows of the ECSN Recommender System

User Space:

The main goal of recommender algorithms is to provide users of online environments with the most relevant items (Ricci, et al., 2011). With reference to the experimental method selected for this research (online study), having a pool of users to interact with various recommender systems implemented in this work was essential. In response to this need, the MyExpert academic social network was developed, and it currently has more than 920 members from 10 universities in Malaysia. Some facilities are provided in MyExpert, which enable members to submit academic items through their web-based profiles. These items consist of academic news, conferences, job offers, and scholarships. As shown in Figure 4.3, the submitted items were checked by a MyExpert administrator prior to restoration to the designated database. In this research, during each week of experiments, 100 items were approved for processing by recommender algorithms in the next steps of this research work. The recommended items are fed by another process in the working space of the ECSN recommender engine, which will be briefly described at a later time.

MyExpert members received the top 10 academic items recommended by four recommending algorithms (random, collaborative, content-based, and ECSN) in every week of the data collection phase. These items were sent through the MyExpert e-Newsletter. Members who received it clicked on the title of interesting news and opened their web pages. As such, this e-newsletter served as the media for sending recommended items to MyExpert users.

An additional functionality that was considered for users of this area is clicking and rating academic items. Collecting the relevance feedback is one of the most important issues when studying recommender systems, as it gets used to evaluate the functionality of recommender algorithms (Shani & Gunawardana, 2011). Thus, MyExpert users' behavior when faced with academic items was recorded for the evaluation process. Users ranked the items based on their interest from one to five stars. One star represents non-relevance and five stars represent complete relevance. The relevance feedback gathered was applied to the Updating User Profile process, which will be elaborated in the context of the ECSN recommender engine working area.

Administrator Space:

The administrator working space provides administrative and monitoring features of an ECSN recommender system. The role of MyExpert admin comes with specially designated privileges and responsibilities of managing the ECSN. Figure 4.3 indicates that sending academic e-Newsletters and confirming new academic news are two principal features that were considered for this working area.

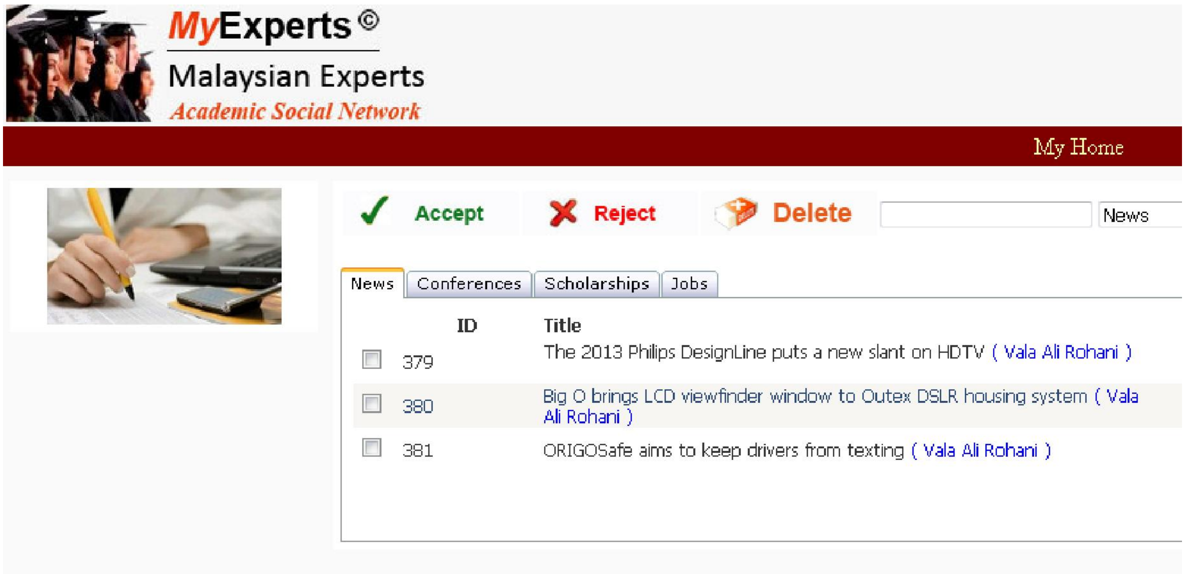


Figure 4.4: The MyExpert Webpage for Administration of Academic Items

Academic news submitted by MyExpert users need to be checked before being processed by recommender engines. For this purpose, ‘Confirming New Academic Items’ was taken into account in the administrative field to prevent the dissemination of wrong or incomplete news to MyExpert users (Figure 4.4). The administrative influence of this feature is revealed in Figure 4.3 by a dashed arrow from ‘Confirming New Academic Items’ to ‘Submitting New Academic Items.’

The second administrative facility seen in Figure 4.3 is ‘Sending Academic e-Newsletter.’ In reference to the problem statement section of this dissertation, during each week of research experiments over 14 weeks, the top 10 academic items were meant to be sent to MyExpert users. More specifically, the recommender systems’ aim was to identify the items most relevant to every MyExpert user and then compile them in an e-Newsletter. Thus, another responsibility of a MyExpert admin is to select one of the recommender algorithms based on the predefined schedule and trigger the procedure of sending e-Newsletters to members’ email. By doing this each week, all 920 MyExpert users would receive emails containing academic MyExpert e-Newsletters in their email inbox. A MyExpert e-Newsletter sample is seen in Figure 4.5.

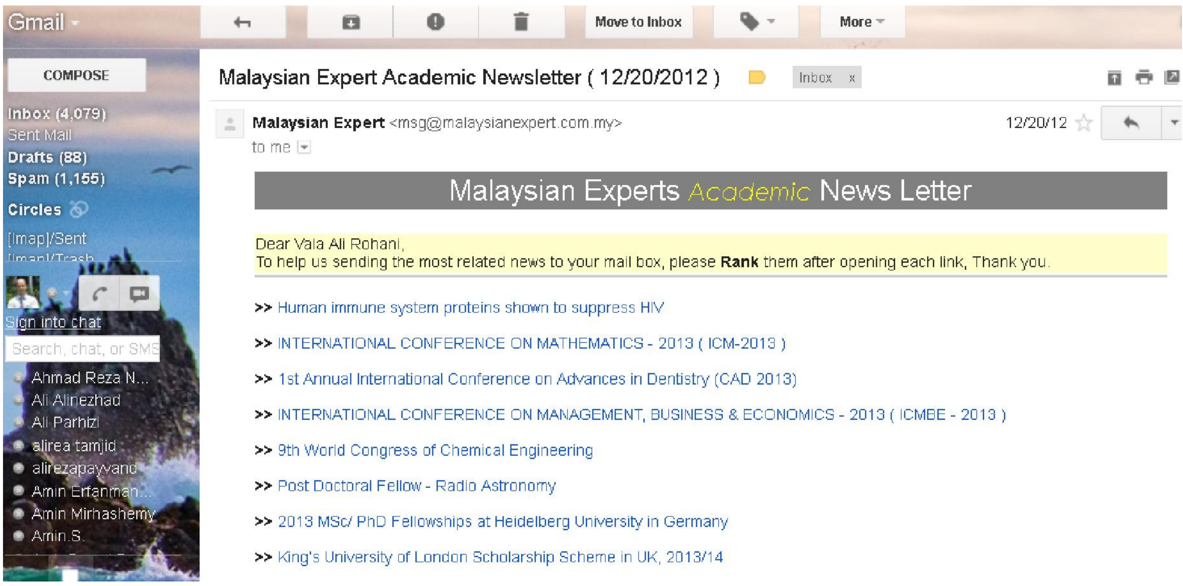


Figure 4.5: MyExpert e-Newsletter

Members receiving the e-Newsletter could click on each news link and open a related webpage to read additional details. They could then also rate the given news by clicking on designated stars (1 to 5). After describing the two administrative MyExpert features, the functionality and workflow of the main ECSN body are presented next.

ECSN Recommender Engine:

The center of Figure 4.3 depicts the heart of ECSN recommender system, namely the 'ECSN Recommender Engine.' This section of the architecture is responsible for collecting the relevance feedback from MyExpert users, generate and keep user profiles updated based on preferences elicited throughout interactions with the system, as well as apply an ECSN recommender algorithm to find the top 10 academic items among 100 submitted every study week, and finally form the weekly e-Newsletter for each MyExpert member. The two key features shown in Figure 4.3 are 'Updating User Profile' and 'Recommending Top 10 Academic Items.'

The former attribute is dedicated to creating and updating the user profiles in the ECSN system. Users of a recommender system may have greatly varying characteristics and interests. To personalize recommendations and user interactions with the systems, recommender engines exploit a range of information regarding user behavior. This concept enables users to provide the recommender system with information regarding likes and dislikes. This is an essential task in providing personalized recommendations since personalization is not possible in the absence of a convenient user model (Perez, et al., 2007). In some cases where the system has no specific knowledge of user preferences and interests, it is unable to recommend the most relevant items. The structure of this information strictly depends on the recommendation technique selected for application. For instance, in collaborative recommender algorithms user profiles are constructed as simple lists containing the ratings provided by the user on certain items. In a demographic recommender system, sociodemographic characteristics such as age, gender, education, and profession are used. In other words, user data constitutes the user profile (Fischer, 2001).

Now that the necessity for a User Profile has been clarified, the functionality of 'Updating User Profile' shall be addressed. As per Figure 4.3 the input information for this process is collected from the user's space. While MyExpert users interact with the system by clicking on, or rating, academic items, all the relevant feedback is received by this process box and re-stored in the MyExpert database. The outgoing data arrows from the 'Updating User Profile' box demonstrate that the relevance feedback is transferred into data base via two different channels: 'User Clicks & Ratings' and 'Preference Score Points.'

As mentioned earlier, the first one is the log records related to the transactions of users' click and ratings. According to Table 4.1, the pointing values which are considered for 1 to 5 stars are relatively -2, -1, +1, +2, and +3 relatively. The negative values represent the unlikeness of users regarding to the given academic items while the positives values show the user interest for items.

The second arrow, 'Preference Score Points,' signifies points of preference scores. Figure 4.1 indicates that the preference scoring tree employed in the ECSN algorithm has four levels. The second highest level has four nodes (ellipses) representing four types of academic items, namely academic news, academic jobs, conferences, and scholarships. The following two lower levels of the tree denote the disciplines to which submitted academic items belong to. It is worth mentioning that in view of the tree structure's parametric design enables both vertical and horizontal extension.

When a user clicks or rates an academic item, the related node of the given item in this hierarchy tree of preferences is established. Then the related point of that node for the given user will be updated. More precisely, user clicks and ratings of more than 2 stars cause increments of these points while ratings with one or two stars lead to decreasing assigned values of a given node. By repeating this process for each recorded user interaction with the system, all nodes of interest will obtain points relative to those that would be used for the user profile in an ECSN recommender algorithm. Higher point values for each node signify that it is more interesting in the list of academic items of the given node for the studied user. Consequent to the continuous user profile updating every week of experiments, it can be used in the next stop of this working space, that is, 'Recommending Top 10 Academic Items.'

The main ECSN recommender engine output is generated by the 'Recommending Top 10 Academic Items' feature. Again, in each week of experiments for this research, the top 10 items from 100 submissions were selected and recommended to MyExpert users through an e-Newsletter. The process box in the ECSN recommender engine is responsible for this mission. Four input arrows show all types of information required for recommending the top 10 academic items in an ECSN engine: new academic items, user profile, friends profile and faculty colleagues profile. Next, these input workflows are concisely described.

As discussed earlier, new academic items are submitted to MyExpert and confirmed by the administrator. In each week of current research experimentation, 100 approved items were submitted in a runtime environment provided for this study, namely MyExpert academic social network. The items were academic jobs, academic news, conferences, and scholarships.

Another arrow represents the user profile briefly described earlier. A hierarchy tree structure has been considered for categorizing the academic items (Figure 4.1). Maintaining updated designated database tables for this tree structure helps inform the ECSN engine regarding MyExpert users' interests and preferences based on their previous interactions with the system. By having this knowledge, the ECSN recommender systems is capable of making true decisions in selecting the top 10 academic items for each given user of the MyExpert academic social network.

However, only considering the user's own preferences causes a cold start issue in the domain of recommender systems. The origin of this shortcoming can be traced to the fact that there is no previously recorded interaction for new users of an online environment. This is a problem that some popular recommender algorithms suffer from, including the collaborative and content-based approaches (Lam, Vu, Le, & Duong, 2008; Zhang, et al., 2010). To mitigate this problem, in addition to considering the users' own preferences, the ECSN algorithm utilizes the 'Friends Profile' and 'Faculty mates Profile' too (Figure 4.3). In doing so, all transaction records of a given user's friends are analyzed by the ECSN recommend engine and the most interesting nodes in the preference tree structure of academic items would be elicited. Then the pointing value of elicited nodes will be updated in the given user's preference tree. The same process would be done regarding the faculty mates of the target user.

Now, after utilizing the 4 above-mentioned inputs, the ECSN recommender algorithm generates an order list of top 10 new academic items by matching the most interesting nodes in the tree structure of preferences and new academic items submitted in the most recent week. This process was iterated for all weeks of the research experiment when the ECSN algorithm was adopted for recommending the top 10 academic items to MyExpert users. Finally, the generated list of recommended items would be embedded in the MyExpert e-Newsletter and get sent to MyExpert users. It is worth mentioning that each user receives a different list of recommended academic items of the week, based on their updated preferences and interests in the MyExpert database.

4.4 Overall View of Adopted Recommender Systems

Now that the detailed workflows and theoretical framework of the ECSN recommender algorithm have been described, the overall view of the adopted recommender system is presented in this section. This research focuses on utilizing social networking techniques to enhance the accuracy of the content-based recommender algorithm in recommending the academic items through the e-newsletter of MyExpert academic social network. To access the practical results in an online environment and evaluate the contribution of this research, in addition to the design, development and implementation of the ECSN recommender algorithm, three more recommender systems are adopted in this study: the random, collaborative, and content-based recommender algorithms. Figure 4.6 depicts the overall architecture of the recommender systems applied in this work.

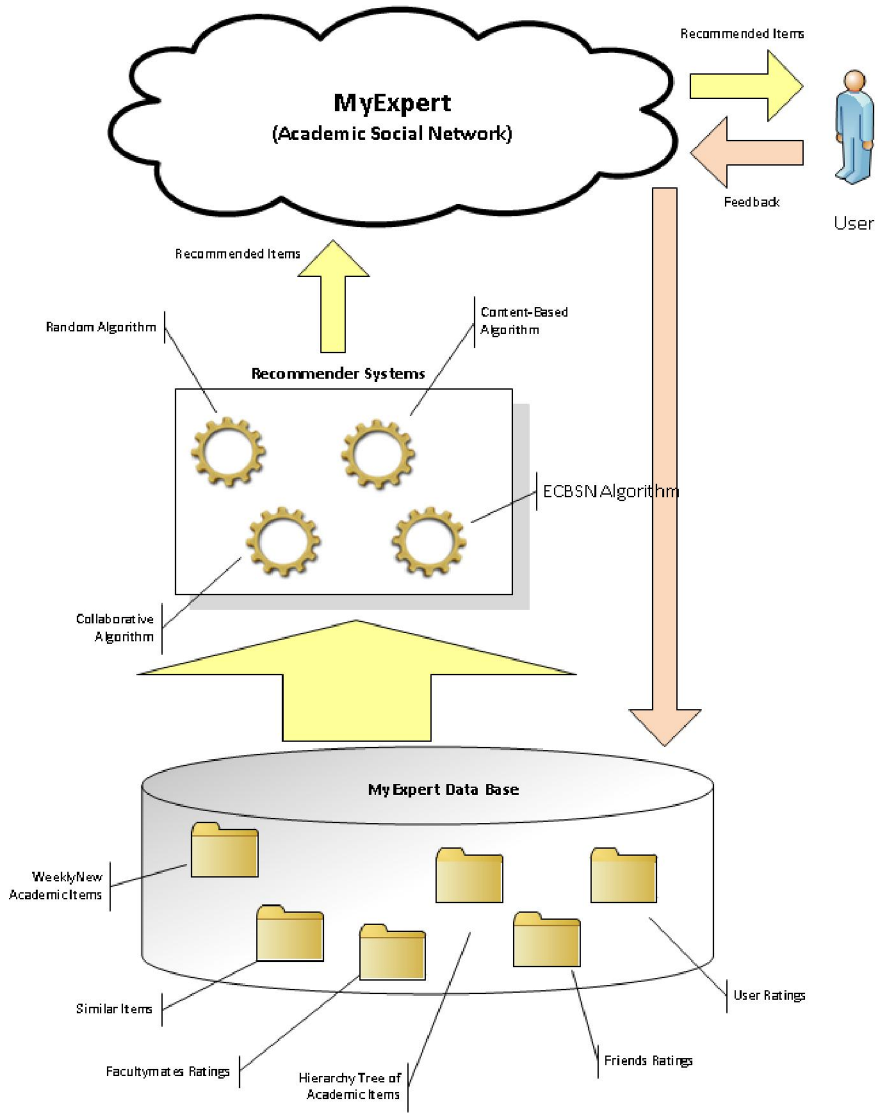


Figure 4.6: Overall View of Adopted Recommender Systems in this Research

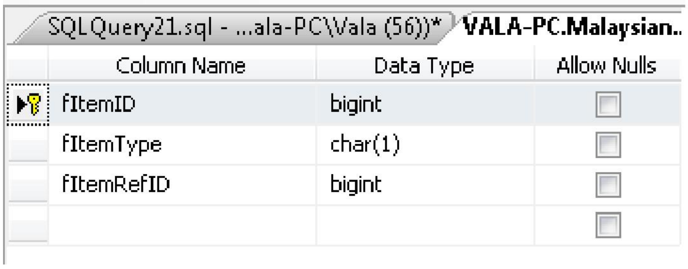
The main workflows are shown in Figure 4.6 between the coarse-grained elements of this study, namely user, MyExpert online environment, recommender system, and MyExpert data base. The MyExpert users feed the relevance feedback for this research by interacting with the MyExpert academic social network. To be exact, users' clicks and ratings submitted for academic items are restored in the database as user ratings. Besides, in conditions when MyExpert users submit new academic items through their web-based profile, these become restored in the database as weekly academic items. The recommender systems intended to select the top 10 academic items among 100 submitted each week of the research experimentation and send them to MyExpert users via e-Newsletters. To achieve this goal, the records in the MyExpert database were utilized, as highlighted in Figure 4.6.

New weekly academic items make the basic table commonly used with all recommender algorithms examined in this research (random, collaborative, content-based, and ECSN). The data related to collected relevance feedback (user ratings) is also needed for the recommendation process in all mentioned algorithms, except the random approach. Similar items are used by the collaborative filtering algorithm to make decisions in finding and recommending the most relevant items to MyExpert users. The similarity between items can be identified by considering user behavior when clicking and ranking academic items. The content-based recommender algorithm and its improved version proposed in this study, ECSN, both make use of the hierarchy tree structure of academic items. This structure categorizes all possible academic items in a tree schematic that would be scored for each user by investigating his or her interests and previous interactions with the system. The ratings belonging to friends and faculty mates are specifically used by the ECSN algorithm. By utilizing these two new paradigms of information, the ECSN recommender system can solve the cold start issue which exists in both the collaborative and content-based algorithms. The implementation details of all four mentioned recommender algorithm are described in subsequent sections of this chapter. ECSN is the enhanced version of the content-based algorithm proposed in this research, and its theoretical framework is presented in brief in next.

4.5 Implementation of Recommender Algorithms

This section briefly describes the design and implemantion of four studied recommender algorithms (random, collaborative, content-based, and ECSN). First of all, it is essential to provide a faculty for merging different types of academic items in a single table to be utilized by each recomemdner algorithm.

Four different types of academic items were submitted in MyExpert as academic news, conferences, academic jobs, and scholarships. With regards to the fact that they have different data fields based on sort of information they convey, four tables have been cosnidered in the MyExpert database for academic items, namely tNews, tJobs, tConferences and tScholarships. During each week of experiments, it was necessary to inetegrate all 100 submitted items of the week in a single table named tNewsTotal. Figure 4.7 illustrates the structure of this table.



SQLQuery21.sql - ...ala-PC\Vala (56))* VALA-PC.Malaysian..

Column Name	Data Type	Allow Nulls
fItemID	bigint	☐
fItemType	char(1)	☐
fItemRefID	bigint	☐
		☐

Figure 4.7: Table tNewsTotal for Integrating all Academic Items

To integrate all academic items into a single table, the stored procedure (spNewsAlgorithmInsertNewToNewsTotal) was designated (Figure 4.8).

```
ALTERPROCEDURE [dbo].[spNewsAlgorithmInsertNewToNewsTotal]

AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.
	SETNOCOUNTON;

	INSERTINTO tNewsTotal
	                              (fItemRefID, fItemType)
	SELECT     fID,'N'AS Expr1
	FROM         tNews
	where fNewsStatus=1

	INSERTINTO tNewsTotal
	                              (fItemRefID, fItemType)
	SELECT     fID,'J'AS Expr1
	FROM         tJobs
	where fJobStatus=1

	INSERTINTO tNewsTotal
	                              (fItemRefID, fItemType)
	SELECT     fID,'C'AS Expr1
	FROM         tConferences
	where fConfStatus=1     INSERTINTO tNewsTotal
	                              (fItemRefID, fItemType)
	SELECT     fID,'S'AS Expr1
	FROM         tScholarships
	where fSchStatus=1
END
```

Figure 4.8: Stored Procedure spNewsAlgorithmInsertNewToNewsTotal for integrating all academic items into a single table

By integrating all possible academic items into a single table, each recommender algorithm can apply its own techniques for analyzing the integrated items and find the most relevant ones to recommend to each given user. The implementation details of these techniques are given next.

4.5.1 Random Algorithm

During the first five weeks of research experiments, from 7th Sep to 16th Oct 2012, the random algorithm was adopted for selecting the top 10 academic items and recommending them to MyExpert users. Because from the beginning of experimentation we did not have any records from MyExpert member preferences and interests, the random algorithm was selected to generate these types of transactional records to be used in subsequent experiment phases by other recommender algorithms.

As a result, algorithm implementation in this case was not very complicated since it only required a mechanism to select ten random records amongst academic items submitted each week of experiments. It was primarily required, though, to transfer the academic items in a specific submission time (@fromDate to @toDate) into a temp table for further processing. A stored procedure (spTempNewsInDurationInsert) was developed in the MyExpert database for this purpose, as shown in Figure 4.9.

```
ALTERPROCEDURE [dbo].[spTempNewsInDurationInsert]
        @fromDate datetime,
        @toDate datetime
As
Begin

        SETNOCOUNTON;
        --------------------- Delete Current Records
        delete
        from tTempNewsAllinDuration

        --------------------- Insert News

        INSERTINTO tTempNewsAllinDuration
                                                    (fNewsRefID, fNewsTitle, fCountryID,
fScienceID, fNewsMemberID, fNewsCountryID, fNewsUniversityID, fNewsFacultyID,
fNewsDepartmentID, fNewsScopeID,
                                                     fNewsType)
        SELECT    fID, fNewsTitle, fCountryID, fScienceID, fNewsMemberID,
fNewsCountryID, fNewsUniversityID, fNewsFacultyID, fNewsDepartmentID,
fNewsScopeID,'N'AS Expr1
        FROM       tNews
        WHERE     (fNewsStatus = 1)and(fNewsSubmitDate >=
@fromDate)and((fNewsSubmitDate <= @toDate))


        --------------------- Insert Conferences

        INSERTINTO tTempNewsAllinDuration
                                                    (fNewsRefID, fNewsTitle, fCountryID,
fScienceID, fNewsMemberID, fNewsCountryID, fNewsUniversityID, fNewsFacultyID,
fNewsDepartmentID, fNewsScopeID,
                                                     fNewsType)
        SELECT    fID, fConfTitle, fCountryID, fScienceID, fNewsMemberID,
fNewsCountryID, fNewsUniversityID, fNewsFacultyID, fNewsDepartmentID,
fNewsScopeID,'C'AS Expr1
        FROM       tConferences
        WHERE     (fConfStatus = 1)and(fConfSubmitDate >=
@fromDate)and((fConfSubmitDate <= @toDate))
```

```sql
--------------------- Insert Jobs

INSERTINTO tTempNewsAllinDuration
                        (fNewsRefID, fNewsTitle, fCountryID,
fScienceID, fNewsMemberID, fNewsCountryID, fNewsUniversityID, fNewsFacultyID,
fNewsDepartmentID, fNewsScopeID,
                        fNewsType)
SELECT     fID, fJobTitle, fCountryID, fScienceID, fNewsMemberID,
fNewsCountryID, fNewsUniversityID, fNewsFacultyID, fNewsDepartmentID,
fNewsScopeID,'J'AS Expr1
FROM       tJobs
WHERE      (fJobStatus = 1)and(fJobSubmitDate >= @fromDate)and((fJobSubmitDate
<= @toDate))


--------------------- Insert Scholarships

INSERTINTO tTempNewsAllinDuration
                        (fNewsRefID, fNewsTitle, fCountryID,
fScienceID, fNewsMemberID, fNewsCountryID, fNewsUniversityID, fNewsFacultyID,
fNewsDepartmentID, fNewsScopeID,
                        fNewsType)
SELECT     fID, fSchTitle, fCountryID, fScienceID, fNewsMemberID,
fNewsCountryID, fNewsUniversityID, fNewsFacultyID, fNewsDepartmentID,
fNewsScopeID,'S'AS Expr1
FROM       tScholarships
WHERE      (fSchStatus = 1)and(fSchSubmitDate >= @fromDate)and((fSchSubmitDate
<= @toDate))


End
```

Figure 4.9: Stored Procedure spTempNewsInDurationInsert

After preparing all possible items in a specific time interval, ten of them were selected in random order for recommendation to each member of MyExpert academic social network. The stored procedure spNewsAlgorithmGetRandom was created for this motive (Figure 4.10).

```
ALTERPROCEDURE [dbo].[spNewsAlgorithmGetRandom]
        -- Add the parameters for tshe stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SETNOCOUNTON;

        selecttop (10)*from tTempNewsAllinDuration orderbyNEWID()
END
```

Figure 4.10: Stored Procedure spNewsAlgorithmGetRandom

Upon preparing the ten random items, they were sent to every MyExpert user through the e-Newsletter designated for this function.

4.5.2 Collaborative Algorithm

As briefly described in the literature review of this dissertation (Chapter 2), in collaborative filtering approach, prediction is done based on the items previously preferred by people with similar preferences and interests. According to a study carried out by Cacheda et al., (2011), there are two main classifications for Collaborative Filtering recommender systems, i.e. memory-based (user-based) and model-based (item-based), which are employed in different application domains.

Earlier research works have used the memory-based approach with elicited information from items previously rated by users. This method requires that all items, ratings and users be collected and stored into the memory to make recommendations. Afterward, to mitigate some of the shortcomings of this approach, a model-based method was developed that looks for similar items instead of making groups of similar users. In other words, it uses an offline pattern created periodically by summarizing item ratings. For this reason, the model-based approach was adopted for this research to utilize its advantages.

Referring to brief descriptions of model-based collaborative recommender systems (Section 2.3.2.1), the following formula is used to find the similarity between items a and b:

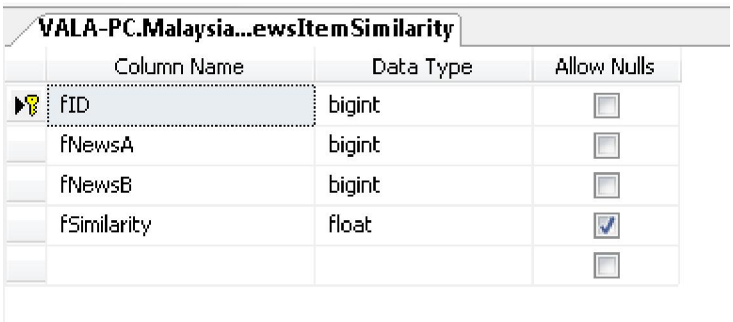
$$sim(\vec{a},\vec{b}) = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}| * |\vec{b}|} \tag{4.1}$$

More precisely, the similarity is calculated as:

$$sim(\vec{a},\vec{b}) = \frac{\sum_{u \in U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_u})^2}\sqrt{\sum_{u \in U}(r_{u,b} - \overline{r_u})^2}} \tag{4.2}$$

where U indicates the set of all users who rated both items a and b. Accordingly, $\boldsymbol{r_{u,a}}$ and $\boldsymbol{r_{u,b}}$ are the ratings assigned by user u to items a and b respectively (Jannach, et al., 2010).

In the MyExpert database, a table (tNewsItemsSimilarity) is created to store the similarity between academic items. The structure of this table is shown in Figure 4.11.


VALA-PC.Malaysia...ewsItemSimilarity

Column Name	Data Type	Allow Nulls
fID	bigint	☐
fNewsA	bigint	☐
fNewsB	bigint	☐
fSimilarity	float	☑
		☐

Figure 4.11: Table tNewsItemsSimilarity for restoring the similarity records between items

The element $\overline{r_u}$ in Equation 4.2 presents the average value of ratings for each member of MyExpert. To restore this value for each member, fAvgRate field is considered in table tProfile, which keeps the detailed information of every MyExpert user (Figure 4.12).

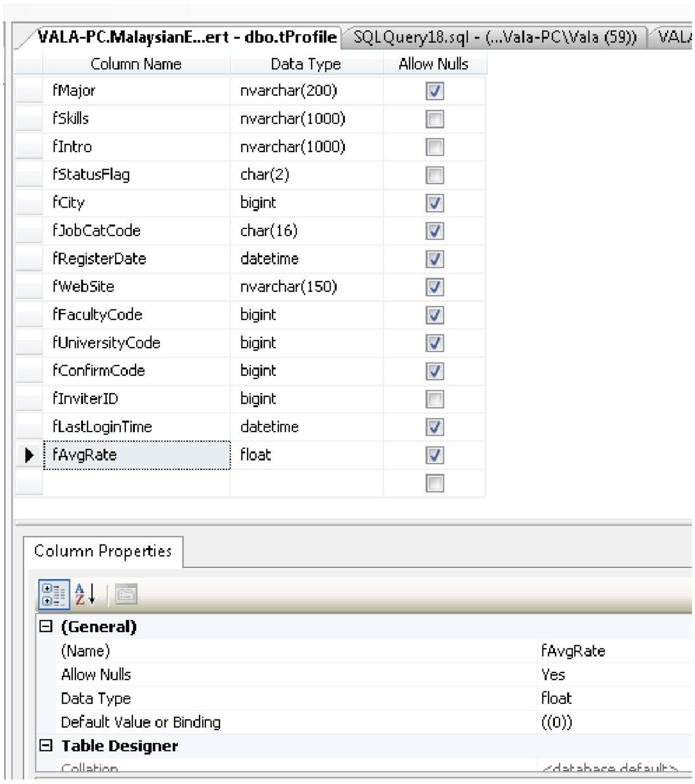


Figure 4.12: Table tProfile for restoring the detail information about MyExpert members

To calculate this value (fAvgRate), a stored procedure (spNewsAlgorithmCalculateAvgRatebyUserID) was prepared, which updates the average value of all ratings submitted by given users in the MyExpert environment (Figure 4.13).

```sql
ALTERPROCEDURE [dbo].[spNewsAlgorithmCalculateAvgRatebyUserID]
	(
		@fUserID bigint
	)
AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.

	SETNOCOUNTON;

	DECLARE @i INT
	DECLARE @totalRateCount INT
	DECLARE @sumRateValue float
	DECLARE @avgRank float

---------------------------- Get Count of Ranks by User

	select @totalRateCount=COUNT(*)
	from tNewsRanks
	where fMemberID=@fUserID

	----------------------------- Get Sum

	select @sumRateValue =SUM(fRank)
	from tNewsRanks
	groupby fMemberID
	having fMemberID= @fUserID

	--------------------------- Update fAvgRank record in user Profile

	set @avgRank =round(@sumRateValue /@totalRateCount , 5)

	update tProfile
	set fAvgRate= @avgRank
	where fMemberID=@fUserID

END
```

Figure 4.13: Stored Procedure spNewsAlgorithmCalculateAvgRatebyUserID for calculating the average value of all ratings submitted by a given member of MyExpert users

Referring back to the list of users who have ranked both items a and b, the news ranking table (tNewsRanks) should be joint to itself and a new relation should be created (Figure 4.14).

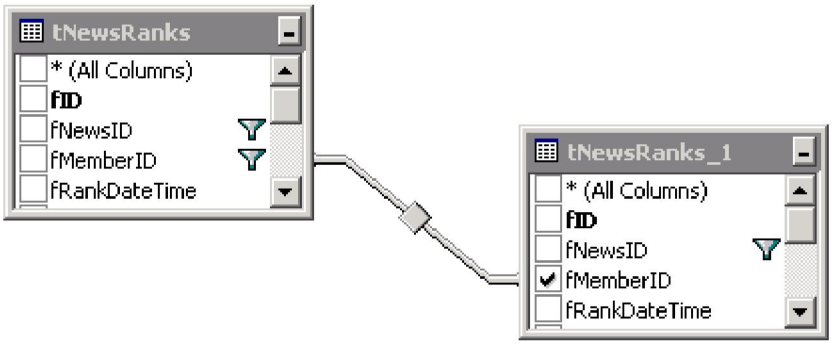


Figure 4.14: Joining tNewsRank to itself to make a relation used for finding users who ranked both items *a* and *b*

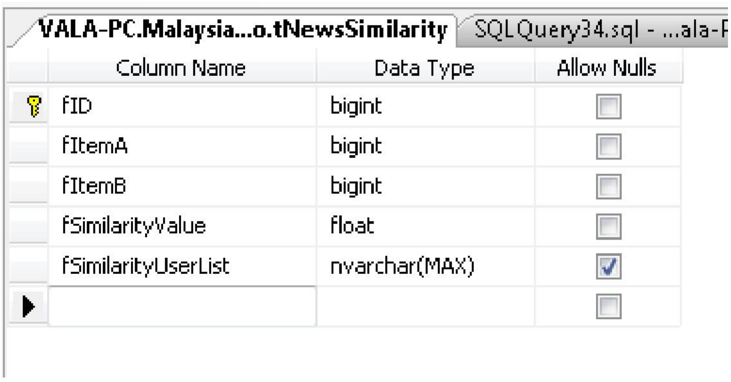
The stored procedure spNewsAlgorithmClbSimilarityItemAB , shown in Figure 4.15, is then called upon to generate the list of users who ranked the items *a* (@fItemA) and *b* (@fItemB).

```
ALTERPROCEDURE [dbo].[spNewsAlgorithmClbSimilarityItemAB]
        (
                @fItemA bigint,
                @fItemAType char(1),
                @fItemB bigint,
                @fItemBType char(1)
        )
AS
BEGIN
SETNOCOUNTON;
SELECTdistinct tNewsRanks_1.fMemberID
FROM       tNewsRanks INNERJOIN
                tNewsRanks AS tNewsRanks_1 ON tNewsRanks.fMemberID =
tNewsRanks_1.fMemberID
WHERE     (tNewsRanks.fNewsID = @fItemA)AND(tNewsRanks.fNewsType=
@fItemAType)AND(tNewsRanks_1.fNewsID = @fItemB)AND(tNewsRanks_1.fNewsType=
@fItemBType)and(tNewsRanks.fMemberID<>5)

END
```

Figure 4.15: Illustration of stored procedure spNewsAlgorithmClbSimilarityItemAB for finding the list of MyExpert users who have ranked the items *a* and *b*

To restore the similarity value between news items, tNewsItemsSimilarity is considered (Figure 4.16).



VALA-PC.Malaysia...o.tNewsSimilarity | SQLQuery34.sql - ...ala-P

Column Name	Data Type	Allow Nulls
fID	bigint	☐
fItemA	bigint	☐
fItemB	bigint	☐
fSimilarityValue	float	☐
fSimilarityUserList	nvarchar(MAX)	☑
		☐

Figure 4.16: Illustration of table tNewsItemsSimilarity

After computing the similarity of all academic items, it is possible to predict which items are the best candidates to be recommended to each given member of MyExpert. The following predictor function was used at this stage:

$$\boldsymbol{pred(u,p)} = \frac{\sum_{i \in ratedItem(u)} sim(i,p) * r_{u,i}}{\sum_{i \in ratedItem(u)} sim(i,p)} \qquad \textbf{(4.3)}$$

A table was designated (tNewsPrediction) to restore the pair (user, news) for all new academic items meant to be analyzed by the recommender engine and to be sent to users (Figure 4.17).

VALA-PC.Malaysia...o.tNewsPrediction | SQLQuery8.sql - (l...ala

Column Name	Data Type	Allow Nulls
fID	bigint	☐
fUserID	bigint	☐
fNewsID	bigint	☐
fPredictionValue	float	☐
		☐

Figure 4.17: Illustration of table tNewsPrediction

Then, for each member of MyExpert (fUserID) the top 10 items with the highest prediction value were selected to be recommended in the e-Newsletter.

To summarize, the implementation of the collaborative recommender algorithm in this research is:

1) Updating the fAvgRate in table tProfile by calling the stored procedure spNewsAlgorithmCalculateAvgRatebyUserID. This task is done by clicking on

'Calculate the Average Rate for Each User' button in the administration webpage designated for sending the MyExpert e-newsletters[1].

2) Adding the latest news records to table tNewsTotal by calling the stored procedure spNewsAlgorithmInsertNewToNewsTotal for each week of expermients.

3) Deleting all records from tNewsSimilarity, and then inserting the new records by clicking on 'Calculate the Similarity between Items' button in the mentioned administration webpage.

4) Calculating the value of similarity by clicking the 'Calculate the Similarity Value' button.

5) Deleting all previous records from the tNewsPrediction table.

6) Deleting the previous latest News ready for sending from the tNewsTempAllInDuration table.

7) Refilling tNewsTempAllInDuration by clicking the 'Transfer News in Duration to Temp' button.

8) Recalculating the prediction value by clicking the 'Calculate News Prediction Value' button.

[1] http://www.malaysianexperts.com.my/UL/adminnewsletter.aspx

9) Preparing the news for sending by clicking on ‘Preparing the News using Collaborative Method’ button. In this step, the highest ranked news for each user would be inserted to table tNewsPredictionSmall.

10) Sending the news using collaborative method by clicking on ‘Send All’ button.

The above-mentioned 10 steps were implemented for all experiments in which the collaborative recommender algorithm was supposed to be applied for recommending the most relevant academic items to MyExpert users.

4.5.3 Content-Based Algorithm

In the third phase of this research, the content-based algorithm was adopted. Referring to Chapter 2, which focuses on a review of literature in the domain of recommender systems, the users’ own preferences through previous interactions are considered in predicting new items in the content-based recommender systems. As Lops et al. (2011) pointed out in their research this approach needs a well-structured framework that supports the techniques for comparing user interests with the item specifications and to ultimately suggest the most suitable item to the target user. Considering the structured nature of academic items used in the MyExpert environment, the preference scoring structure was implemented to study and model the user profiles (De Gemmis et al., 2009; Kim et al., 2006).

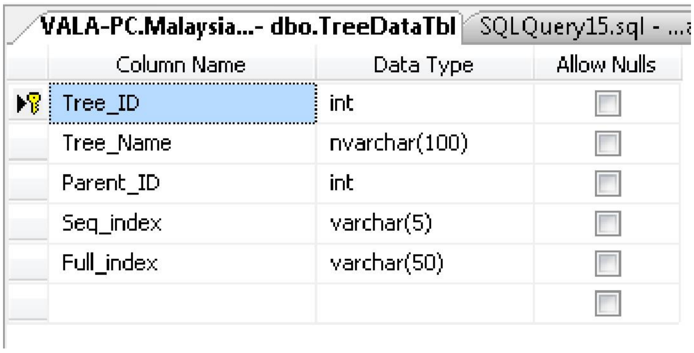
To execute the hierarchy tree structure of preference scoring, a table (TreeDataTbl) was created that re-stores the categories of academic items into three levels:

Level 1: item types (News, Conferences, Jobs, Scholarships)

Level 2: first level of disciplines (e.g., all, engineering, art, etc.)

Level 3: second level of disciplines (e.g., computer sciences, mechanical engineering, etc.)

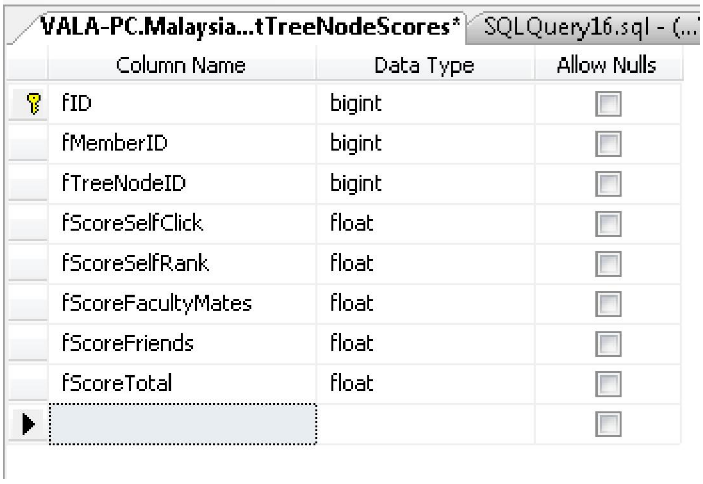
The structure of table TreeDataTbl that has 472 records is illustrated in Figure 4.18.



Column Name	Data Type	Allow Nulls
Tree_ID	int	
Tree_Name	nvarchar(100)	
Parent_ID	int	
Seq_index	varchar(5)	
Full_index	varchar(50)	

Figure 4.18: Structure of Table TreeDataTbl

Once the academic item categories are re-stored in table TreeDataTbl, they should be linked to another table (tTreeNodeScores) that plays the main role in the content-based method (Figure 4.19). For each MyExpert member, the preference scores elicited by analyzing the users' online interactions can be restored in this table.



VALA-PC.Malaysia...tTreeNodeScores* | SQLQuery16.sql - (...

Column Name	Data Type	Allow Nulls
fID	bigint	☐
fMemberID	bigint	☐
fTreeNodeID	bigint	☐
fScoreSelfClick	float	☐
fScoreSelfRank	float	☐
fScoreFacultyMates	float	☐
fScoreFriends	float	☐
fScoreTotal	float	☐
		☐

Figure 4.19: Structure of Table tTreeNodeScores

As seen in Figure 4.18, the tTreeNodeScores table re-stores the scoring data of ratings submitted by users for all visited items. From these scores, fScoreSelfClick and fScoreSelfRank are related to ratings by the given user, applied in the content-based recomemdner algorithm. Nevertheless, fScoreFacultyMates and fScoreFriends are two other scores to be used by the ECSN algorithm. After calculating these scores for each MyExpert member, the total score is restored in fScoreTotal. To make this tree, the storedprocedurespNewsAlgorithmCBTreeScoreMaking is invoked (Figure 4.20).

```
ALTERPROCEDURE [dbo].[spNewsAlgorithmCBTreeScoreMaking]

AS
BEGIN

        INSERTINTO tTreeNodeScores
                                    (fMemberID, fTreeNodeID)
        SELECT    tProfile.fMemberID, TreeDataTbl_1.Tree_ID
        FROM      tProfile CROSSJOIN
                                    TreeDataTbl AS TreeDataTbl_1
        WHERE     (tProfile.fStatusFlag ='01')
        ORDERBY tProfile.fMemberID, TreeDataTbl_1.Tree_ID

END
```

Figure 4.20: The Stored Procedure spNewsAlgorithmCBTreeScoreMaking

With regards to all the above-mentioned issues related to the database structure of the main elements in the content-based recommender systems, the sequential step approach implementation is summarized as follows:

1- Creating the table tTreeNodeScores to restore the scoring data of ratings submitted by users for all visited items. To make this tree, the stored procedure spNewsAlgorithmCBTreeScoreMaking is employed.

2- Updating the fScienceID in table tNewsRanks by executing the stored procedure spNewsAlgorithmNewsRankScienceIDUpdate.

3- Calculating the various scores for table tTreeNodeScores for each user by clicking on “Calculating the scores (Self Click).”

4- Preparing the list of new academic items submitted in the current week by pressing the “Transfer the News in Duration to Temp” button.

5- Adding the latest news records to tNewsTotal by calling the stored procedure spNewsAlgorithmInsertNewToNewsTotal.

6- Updating the fTreeNodeID for new items by invoking the stored procedure spNewsAlgorithmCBUpdateNewsNodeID.

7- Preparing the top 10 recommended items for each user. In this stage, the top 10 new items are inserted into the table tNewsPredictionCB10based by clicking on "Prepare the top 10 items in CB algorithm."

8- Sending the news by pressing the "Sending News using Content-Based algorithm" button.

The 8 steps outlined above are done for each experiment for which the content-based recommender algorithm was meant to be applied to for recommending the most relevant academic items to MyExpert users.

4.5.4 ECSN Algorithm

In the final three weeks of research experiments, from 4th Dec to 26th Dec 2012, the ECSN algorithm was adopted to make recommendations to MyExpert members. The ECSN recommender system is the enhanced version of the content-based approach, which in addition to considering the given user's own preferences and interests, takes advantage of friends and faculty mates to solve the cold start problem in pure content-based recommender systems (Figure 4.21).

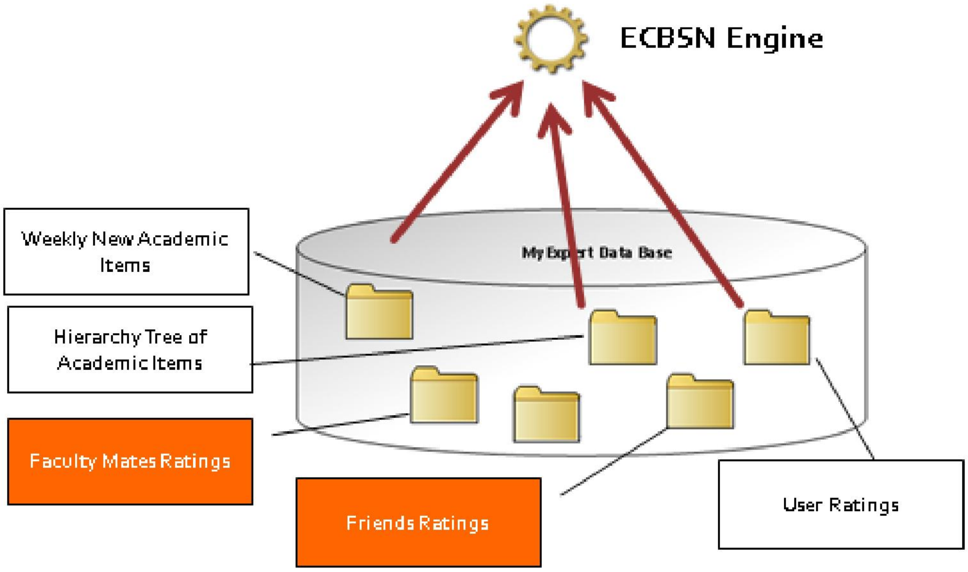


Figure 4.21: The Database Tables Used by the ECSN Recommender Engine

Besides boosting the average values of leaf nodes to higher levels, MyExpert users have to opportunity to receive an even wider variety of relevant recommendations based on their previous interactions with the system.

The same database structure utilized for the content-based recommender algorithm is also used for the ECSN algorithm. However, as stated previously, the ECSN algorithm makes use of friends and faculty mates' preferences to improve the recommendations for each member of the MyExpert academic social network. The ECSN algorithm is shown in Figure 4.22.

Begin

Initialization of Preference Scores to Zero

For each member (m) of MyExpert:

i) Update the preference scoring tree (tTreeNodeScores)

1. For all clicked and rated items by (m), calculating the self-click and self-rank scores

2. For each friends of (m), calculating the average score of top 3 items and update the designated nodes in preference scoring tree

3. For each faculty mates of (m), calculating the average score of top 3 items and update the designated nodes in preference scoring tree

4. Computing the average score value for all leaf level nodes and pushing it up to their parent nodes

ii) Selecting the top 10 items and generating the Ordered Recommended List based on Prediction Value

ii) Generating the e-Newsletter and send it to given user

End

Figure 4.22: The Algorithm of ECSN Recommender System

Besides considering the user's own preferences, the ECSN algorithm utilizes the 'Friends Profile' and 'Faculty mates Profile'. In doing so, all transaction records of a given user's friends are analyzed by the ECSN recommend engine and the most interesting nodes in the preference tree structure of academic items are elicited. Then the pointing values of the elicited nodes are updated in the given user's preference tree.

The stored procedure spNewsAlgorithmNewGetUserFacultyTopScoredNodes serves this purpose (Figure 4.23).

```
ALTERPROCEDURE [dbo].[spNewsAlgorithmNewGetUserFriendsTopScoredNodes]

(
        @memberID bigint
)

AS

BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SETNOCOUNTON;

        SELECTtop 3
tTreeNodeScores.fTreeNodeID,SUM(tTreeNodeScores.fScoreSelfClick)AS
Expr1,SUM(tTreeNodeScores.fScoreSelfRank)AS
Expr2,(SUM(tTreeNodeScores.fScoreSelfClick)+SUM(tTreeNodeScores.fScoreSelfRank))as
Total
,COUNT(tTreeNodeScores.fTreeNodeID),round(((SUM(tTreeNodeScores.fScoreSelfClick)+SU
M(tTreeNodeScores.fScoreSelfRank))/COUNT(tTreeNodeScores.fTreeNodeID)), 2)as _Avg
        FROM        tTreeNodeScores INNERJOIN
                                                        tFriend ON tTreeNodeScores.fMemberID
= tFriend.fFriendID
        WHERE    (tTreeNodeScores.fScoreSelfClick <> 0)AND(tFriend.fMemberID =
@memberID)OR
                                                        (tFriend.fMemberID =
@memberID)AND(tTreeNodeScores.fScoreSelfRank <> 0)
        GROUPBY tTreeNodeScores.fTreeNodeID
        orderby _Avg desc

END
```

Figure 4.23: The Stored Procedure spNewsAlgorithmNewGetUserFriendsTopScoredNodes

The same process is followed regarding the target user's faculty mates (Figure 4.24).

```
ALTERPROCEDURE [dbo].[spNewsAlgorithmNewGetUserFacultyTopScoredNodes]

(
        @memberID bigint
)

AS

BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SETNOCOUNTON;

        SELECT
tTreeNodeScores.fTreeNodeID,SUM(tTreeNodeScores.fScoreSelfClick)AS
Expr1,SUM(tTreeNodeScores.fScoreSelfRank)AS Expr2,

        SUM(tTreeNodeScores.fScoreSelfClick)+SUM(tTreeNodeScores.fScoreSelfRank)AS
Total,COUNT(tTreeNodeScores.fTreeNodeID)AS Expr3,

        ROUND((SUM(tTreeNodeScores.fScoreSelfClick)+SUM(tTreeNodeScores.fScoreSelf
Rank))/COUNT(tTreeNodeScores.fTreeNodeID), 2)AS _Avg,
                                        tProfile_1.fMemberID
        FROM        tProfile INNERJOIN
                                        tProfile AS tProfile_1 ON
tProfile.fFacultyCode = tProfile_1.fFacultyCode INNERJOIN
                                        tTreeNodeScores ON
tProfile.fMemberID = tTreeNodeScores.fMemberID
        WHERE    (tTreeNodeScores.fScoreSelfClick <> 0)and(tProfile.fMemberID <>
@memberID)AND(tProfile_1.fMemberID = @memberID)AND(tProfile_1.fFacultyCode
ISNOTNULL)OR
                                       (tProfile_1.fMemberID =
@memberID)AND(tProfile.fMemberID <> @memberID)and(tProfile_1.fFacultyCode
ISNOTNULL)AND(tTreeNodeScores.fScoreSelfRank <> 0)
        GROUPBY tTreeNodeScores.fTreeNodeID, tProfile_1.fMemberID
        orderby _Avg desc

END
```

Figure 4.24: The Stored Procedure

spNewsAlgorithmNewGetUserFacultyTopScoredNodes

After calculating the given user's own preferences, friends, and faculty mates, the total preference score of each elicited node needs to be calculated with the designated stored procedure (Figure 4.25).

```
ALTERPROCEDURE [dbo].[spNewsAlgorithmNewNodeUpdateScoreTotal]

AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.
	SETNOCOUNTON;

	UPDATE	tTreeNodeScores
	SET		fScoreTotal = 5*(fScoreSelfClick+ fScoreSelfRank)+
fScoreDepMates+3*(fScoreFacultyMates)

END
```

Figure 4.25: The Stored Procedure spNewsAlgorithmNewNodeUpdateScoreTotal

In light of the new database elements of the ECSN algorithm, the sequential implementation steps for the final three weeks of research experiments are:

1- Creating the table tTreeNodeScores to restore the scoring data of ratings submitted by users for all visited items. To make this tree, the stored procedure spNewsAlgorithmCBTreeScoreMaking is called on.

2- Updating the fScienceID in the table tNewsRanks by executing the stored procedure spNewsAlgorithmNewsRankScienceIDUpdate.

3- Calculating the various scores for table tTreeNodeScores for each user by clicking on “Calculating the scores (Self Click).”

4- Calculating the scores for friends of a given user by invoking the stored procedure spNewsAlgorithmNewGetUserFriendsTopScoredNodes.

5- Calculating the scores for faculty mates of a given user by invoking the stored procedure spNewsAlgorithmNewGetUserFacultysTopScoredNodes.

6- Calculating the total score by invoking the stored procedure spNewsAlgorithmNewNodeUpdateScoreTotal.

7- Computing the average score value for all leaf level nodes and pushing it up to their parent nodes by invoking the stored procedure spNewsAlgorithmNewNodeUpdateAvgScore.

8- Preparing the list of new academic items that were submitted in the current week by pressing the "Transfer the News in Duration to Temp" button.

9- Adding the new news records to tNewsTotal by calling the stored procedure spNewsAlgorithmInsertNewToNewsTotal.

10- Updating the fTreeNodeID for new items by invoking the storedprocedurespNewsAlgorithmCBUpdateNewsNodeID.

11- Preparing the top 10 recommended items for each user. In this stage, the top 10 new items are inserted into the table tNewsPredictionCB10based by clicking on "Prepare the top 10 items in ECSN algorithm."

12- Sending the news by pressing the "Sending News using ECSN algorithm" button.

The 12 steps outlined are carried out for each of the experiments for which the ECSN recommender algorithm had to be applied for recommending the most relevant academic items to MyExpert users.

4.6 Summary

This chapter illustrated the technical issues with present research. At the first section, the theoretical framework of ECSN recommender algorithm was presented. After that, ECSN architecture was shown to illustrate the workflows among proposed recommender system elements. The last part of this chapter focused on implementation of four studied recommender algorithms. 14 weeks of online experiments were conducted in this study to compare the performance of proposed recommender algorithm (ECSN) with other previous ones. The details of these experiments are presented in following chapter and the results are evaluated based on well-known measurements of Precision, Recall, Fallout, and F1.

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1 Introduction

It was discussed in Chapter 4 that the second phase of this research focused on the design and development of ECSN, and three other recommender algorithms which were studied in this research. The MyExpert academic social network was developed as a runtime environment for this study and currently has 920 members from 10 universities in Malaysia. After designing and developing the four recommender algorithms (random, collaborative, content-based, and ECSN), they were implemented to recommend the most relevant items to the members of MyExpert. Hence, the third phase of this research is dedicated to evaluating the four mentioned recommender systems along with their performance based on usage prediction measurements. This chapter provides the details of the evaluation process, which ran for 14 weeks from 7th September to 26th December, 2012.

5.2 Experimental Design of This Research

In view of all the issues discussed so far, live user experimentation was chosen as the best approach for this research, as it deals with the shortcomings of the other two methods. Besides, owing to the nature of this study, there were many more reasons for which it was necessary to prepare a real academic social network (MyExpert) to evaluate the prediction accuracy of the ECSN algorithm and compare its performance with other recommender algorithms.

Firstly, there was no offline data set for academic social networks with information on user feedback regarding the recommended academic items. As stated in the problem statement (section 1.2), this research focuses on enhancing the recommending process of academic items through an e-Newsletter. Thus, a need arose for a real academic social network with real users who receive the recommended academic items via email. By having a real online academic social network, it could be possible to record the real behavior of users in their interactions with various recommender systems.

Secondly, to develop the ECSN algorithm and compare its results with other recommender systems, it was necessary to access the code behind an academic social network. By creating MyExpert as the real runtime environment for this research work, C# programming could be utilized to develop the random, collaborative, and content-based and ECSN recommender systems and run them in MyExpert.

The third reason for choosing the online method is that earlier studies on recommender systems (Herlocker et al., 2004; Kohavi et al., 2009; Shani & Gunawardana, 2011) show that online evaluation is the strongest experimental approach for measuring the efficiency of a recommender system. In this approach, the real behavior of users is studied by collecting their relevance feedback when faced with different algorithms of recommender systems in an experimental environment.

5.3 MyExpert as Runtime Environment

To prepare the real runtime environment of this research for conducting the online experiments, MyExpert was designed and developed in this study as the first academic social network in Malaysia. It now has over 900 academicians from 10 universities in

Malaysia. Figure 5.1 illustrates the details of this process. In this research, the main objective of the recommendation process is defined as suggesting the top 10 academic items to MyExpert users in each week of experiments. Referring to Table 2.4 in Chapter 2, which presents the list of academic items presented in academic social networks, there are 9 variant items in the list. In the present study, four categories of these items were selected as follows to be used in the research experiments:

- Academic news
- Conference notifications
- Scholarship notifications
- Academic job offers.

Each week, 100 academic items comprising 25 news, 25 conferences, 25 scholarships, and 25 job offers were submitted to MyExpert through the online web pages specifically designated for this purpose. After completing the weekly submission process, MyExpert members would receive the top 10 items from a total of 100 in their email. Users expected to receive the most relevant items through the MyExpert academic e-newsletter.

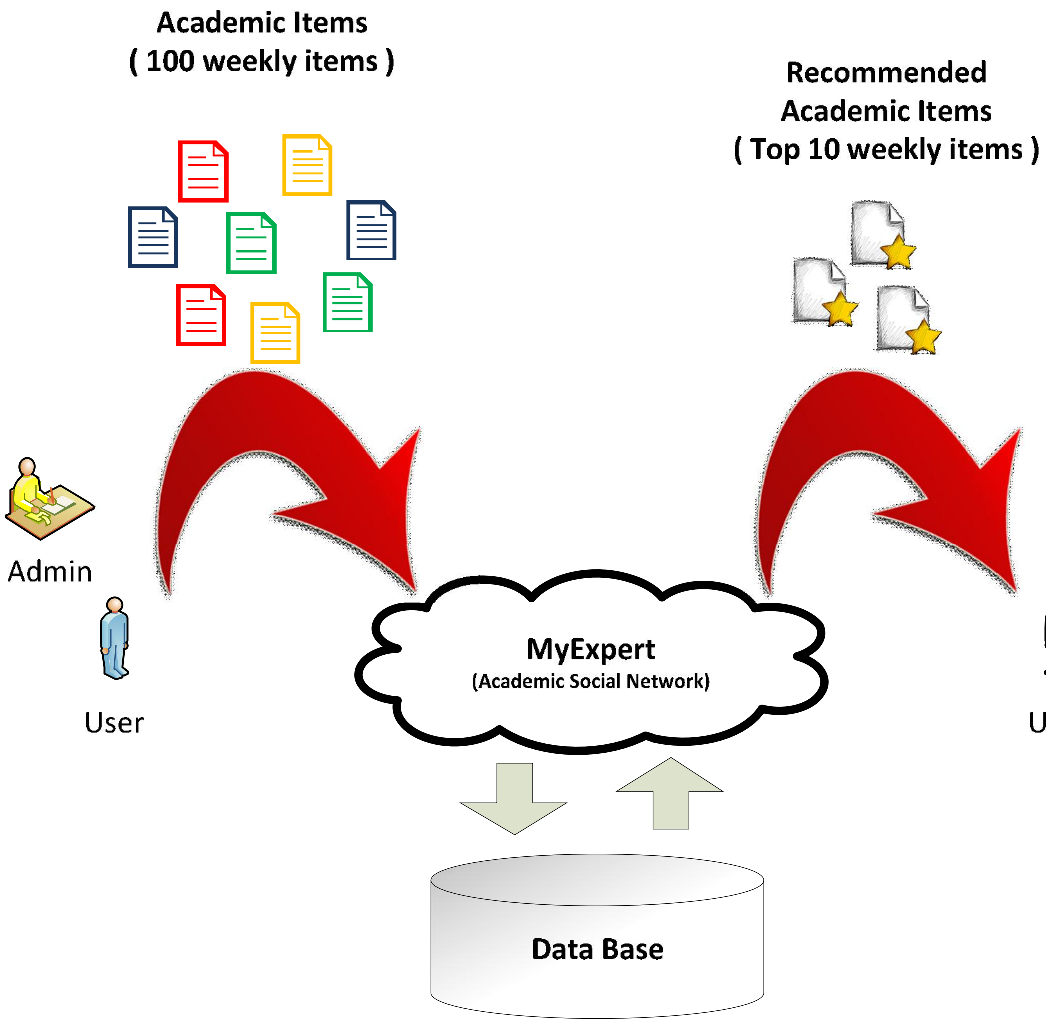


Figure 5.1: Illustration of Recommendation Process in MyExpert Environment

Consequently, this work entails the recommendation of the most relevant academic items to members of academic social networks based on their preferences.

5.4 Results

As seen in Chapter 4, the four recommender algorithms assessed in this research are the random, collaborative, content-based, and ECSN. In this chapter, the performance of these algorithms was evaluated based on precision, recall, fallout, and F1 as the most accepted usage prediction metrics in the domain of recommender systems (Shani &

Gunawardana, 2011). Data collection for this research ran for 14 consecutive weeks from 7th September to 26th December 2012 (Table 5.1).

Table 5.1: The Schedule for Online News Broadcasting in the Data Gathering Phase

	Date	Description
Random	7th Sep 2012	Sending Part #1 of academic news
	17th Sep 2012	Sending Part #2 of academic news
	24th Sep 2012	Sending Part #3 of academic news
	1st Oct 2012	Sending Part #4 of academic news
	9th Oct 2012	Sending Part #5 of academic news
Collaborative	17th Oct 2012	Sending Part #6 of academic news
	24th Oct 2012	Sending Part #7 of academic news
	31th Oct 2012	Sending Part #8 of academic news
Content-Based	12th Nov 2012	Sending Part #9 of academic news
	19th Nov 2012	Sending Part #10 of academic news
	25th Nov 2012	Sending Part #11 of academic news
ECSN algorithm	4th Dec 2012	Sending Part #12 of academic news
	11th Dec 2012	Sending Part #13 of academic news
	20th Dec 2012	Sending Part #14 of academic news

Throughout the 14 weeks, four recommender algorithms, namely random, collaborative, content-based, and ECSN were applied for selecting and recommending the most relevant academic items to MyExpert users. The first five weeks were dedicated to the random algorithm since the other three require previous ratings to work with. After gathering the applicable feedback with the random algorithm, the collaborative algorithm attempted to recommend the top 10 items to users. It took three weeks to collect records of user behavior for this algorithm. Immediately after, the content-based algorithm was applied for the next three weeks, from 12th November to 3rd December 2012. The final and most important component of data collection was allocated to the ECSN algorithm. Basically, this proposed algorithm was examined over the last three weeks, from 4th December to 26th Dec 2012.

5.4.1 Evaluation Results

To reiterate, four measurements (precision, recall, fallout, and F1) were selected to evaluate the prediction accuracy of recommender algorithms which were studied in this research. The results of this evaluation are presented in the next section.

Precision

Precision is one of most prevalent metrics for assessing usage prediction in recommender systems and information retrieving studies. It plays a great role in instances where some sets of best results are required out of several possible alternatives (Shani & Gunawardana, 2011).

Basically, precision is the share of top results that are relevant. In this study, the relevant items defined include academic items visited by users and that were rated with over 2 stars.

Table 5.2 illustrates four possible conditions based on the selection and usage situations.

Table 5.2: The Possible Conditions of Item Recommendation to a User

	Selected	Not Selected	Total
Relevant	N_{rs}	N_{rn}	N_r
Irrelevant	N_{is}	N_{in}	N_i
Total	N_s	N_n	N

According to the notations in Table 5.2, precision is defined as:

$$Precision = \frac{N_{rs}}{N_{rs}+N_{is}} \quad \textbf{(5.1)}$$

It can furthermore be stated that precision is the probability that a recommended item corresponds to a user's interests and preferences.

The remainder of this section focuses on result analysis based on the precision measurement.

Table 5.3: The Precision Values throughout 14 Weeks of Experiments

Recommender Algorithm	Data Gathering Series No.	Precision: Value	Precision: Graph	Average
Random	Series #1	0.217241379		Pre test
	Series #2	0.225974026		
	Series #3	0.180392157		0.1693
	Series #4	0.151785714		
	Series #5	0.175641026		
Collaborative	Series #6	0.191304348		0.2066
	Series #7	0.228571429		
	Series #8	0.2		
Content Based	Series #9	0.209850746		0.2132
	Series #10	0.215280702		
	Series #11	0.214536		
ECSN	Series #12	0.225423729		0.2477
	Series #13	0.245454545		
	Series #14	0.272340426		

Table 5.3 presents the precision values for 14 weeks of data gathering in this research. During each week of experiments and after gathering the relevant feedback from MyExpert users when faced with recommended items, the average precision value for all members involved was calculated and stated in the value column for the respective week. The first two experimental runs were considered pretest stages where MyExpert members received the recommended items through weekly e-Newsletters and, accordingly, were not included in measurements. During these two weeks, they tried rating the academic items. An upward trend in the Precision value occurred throughout the 14 weeks. The last column in Table 5.3 portrays the average precision value for each recommender algorithm. The four average values in the last column demonstrate that

the precision values of the first three algorithms slightly increased from the random (0.1693) to collaborative (0.2066) and content-based (0.2132) recommender algorithms. However, a sharp rise in the precision value of the ECSN recommender algorithm took place, reaching 0.2477 with 21% improvement compared to the content-based algorithm. It should also be noted that the ECSN method enhanced the collaborative precision by 20% and the random by 32%. Therefore, an analysis of the precision values achieved over 14 weeks of experiments by applying the four different recommender systems shows that the ECSN algorithm designed and proposed in this research work certainly contributed to the accuracy prediction in the MyExpert recommendation process.

Recall

Recall is recognized as another metric for measuring usage prediction in recommender systems and other information retrieval domains. It determines the proportion of all relevant results included in the top results (Herlocker, et al., 2004). As Sarwar et al. (2001) stated in their research, in studies where a fixed number of recommendations is suggested to each user (such as the current study in which the top 10 items were recommended to MyExpert users in every week of experiment) precision and recall can be computed at each recommendation list length N for each user. Then the average value of precision and recall can be computed for all users involved in the experiment.

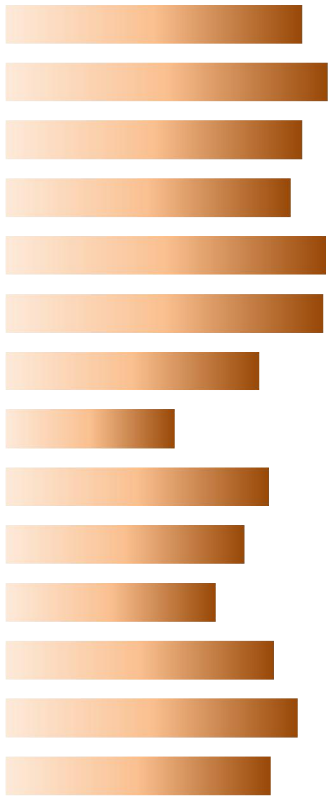
Precision and recall are inversely related. In most cases, increasing the size of the recommendation set will increase recall but decrease precision (Shani & Gunawardana, 2011). This theory is confirmed by analyzing the data in Tables 5.3 and 5.4, which present precision and recall respectively. In the present research where the top 10 academic items are included in a recommendation list the precision values range around 0.2 while recall values are roughly 0.9.

In the recommendations domain, a perfect recall score of 1.0 indicates that all excellent items were recommended in the list. Consequently, a higher precision value is better. Recall, or the true positive rate, is calculated as the ratio of selected (recommended) items used (relevant) to the total number of items used (Herlocker, et al., 2004):

$$Recall = \frac{N_{rs}}{N_{rs}+N_{rn}} \quad \textbf{(5.2)}$$

Table 5.4 shows the analysis results of this measurement for each week of experiments conducted in this research. Again, the relevance feedback gathering phase of this research took 14 weeks. During the first 5 weeks, academic items were recommended to MyExpert users by applying a random algorithm. Three more recommender algorithms (collaborative, content-based, and ECSN) were adopted in the remaining weeks.

Table 5.4: The Recall Values over 14 Weeks of Experiments

Recommender Algorithm	Data Gathering Series No.	Recall Value	Recall Graph	Average
Random	Series #1	0.97164751		Pre test
	Series #2	0.995238095		
	Series #3	0.971895425		0.9756
	Series #4	0.961309524		
	Series #5	0.993589744		
Collaborative	Series #6	0.990942029		0.9263
	Series #7	0.932738095		
	Series #8	0.855144558		
Content Based	Series #9	0.941044776		0.9175
	Series #10	0.918910914		
	Series #11	0.892632275		
ECSN	Series #12	0.946166263		0.9523
	Series #13	0.967820599		
	Series #14	0.942907801		

The evaluation results of recall measurements from 14 series of gathered relevance feedback are illustrated in Table 5.4. The recall value is measured for each series of data gathering by computing the average value of this metrics for all members who visited the web page of academic items during the specified week of experiments. Taking into account the values in the last column (Average), the overall recall value of the ECSN algorithm (0.9523) is better than both the content-based (0.9175) and collaborative approach (0.9263). In other words, the recall rates of the collaborative and content-based algorithms improved by 3% and 4%, respectively. Although the random approach presents the best recall rate (0.9756), it seems somewhat odd. This might be because during the first 5 weeks, there were not many related news compared to the following weeks. Hence, MyExpert users mostly clicked and ranked the recommended academic items, and the portion of related items not included in the recommended list decreased in the first 5 weeks. This phenomenon led to enhanced recall rates while evaluating the usage prediction of the random recommender algorithm.

To further assure the performance of the ECSN algorithm, two other metrics were applied to evaluate the accuracy prediction of all studied recommender systems in this research, namely Fallout and F1. The upcoming sections focus on analyzing the relevance feedback based on these metrics.

Fallout

Fallout, or the false positive rate, is measured as the ratio of selected (recommended) items that are not used (irrelevant) to the total number of unutilized items:

$$Fallout = \frac{N_{is}}{N_{is}+N_{in}} \quad \textbf{(5.3)}$$

In a number of research works (Shani & Gunawardana, 2011) fallout is known as the false positive rate. It is the probability that an irrelevant (not used) item will be recommended to a user. According to this definition, a lower fallout rate indicates better recommender algorithm performance. Table 5.5 presents the fallout rate for each feedback series in this research.

Table 5.5: The Fallout Values during 14 Weeks of Experiments

Recommender Algorithm	Data Gathering Series No.	Fallout		Average
		Value	Graph	
Random	Series #1	0.079655298		Pre test
	Series #2	0.07876572		
	Series #3	0.083231321		0.0844
	Series #4	0.086121657		
	Series #5	0.083744789		
Collaborative	Series #6	0.082245475		0.0810
	Series #7	0.078659818		
	Series #8	0.081821206		
Content Based	Series #9	0.078631394		0.0801
	Series #10	0.080502074		
	Series #11	0.081030192		
ECSN	Series #12	0.079236005		0.0770
	Series #13	0.077034544		
	Series #14	0.074596385		

The most striking feature in this table is the decreasing fallout rate trend through the 14 weeks of experiments. The random recommender algorithm has the lowest average fallout value (0.0844), meaning that the highest amount of irrelevant items was included in the recommendation list during the first five weeks. The average fallout rate improved slightly (0.0810) in the next three weeks when the collaborative algorithm was applied. The content-based recommender approach made it even better (0.0801) in the next stage. The best rate was in fact achieved by the ECSN recommender algorithm with a value of 0.0770. Basically, the fewest irrelevant items were included during the last three weeks when the ECSN algorithm was applied. It can be concluded that the ECSN recommender algorithm outperformed the random, collaborative and content-based algorithms by 7%, 5%, and 4% respectively.

F1 Measure

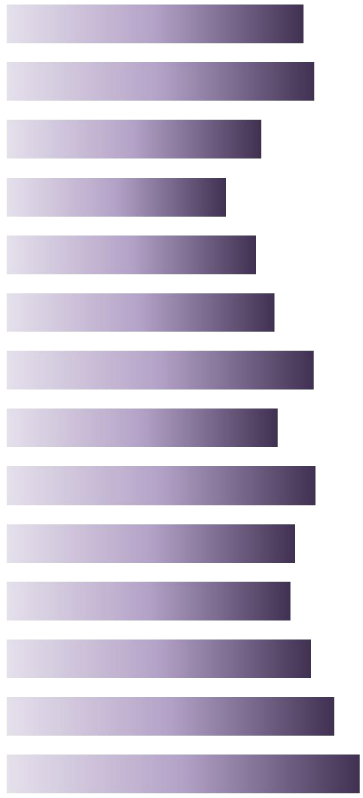
To evaluate the overall performance of a recommender algorithm it makes sense to consider precision and recall together (Herlocker, et al., 2004). Various research works have pointed out that precision and recall are inversely related and dependent on the length of the result list returned to the user (Cleverdon, et al., 1966). So under these circumstances, a vector of precision/recall pairs may describe recommender system performance. Several methods have been assessed to combine precision and recall into a single metric (Harman, 1995; Sarwar, et al., 2001). One approach is the F1 metric (Eq. 5.7) which amalgamates precision and recall into a single value. Sarwar et al. (2001) also used F1 to evaluate the performance of recommender systems in their work.

The F1-score, or F1-measure, is defined as the standard harmonic mean of precision and recall:

$$F1 = \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}} = \frac{2 \times Precision \times Recall}{Precision+Recall} \tag{5.4}$$

As shown in Eq. 5.4, the values of both precision and recall are considered when calculating the F1 score for measuring the accuracy prediction of a given recommender algorithm. Table 5.6 demonstrates the F1 scores in the fourteen weeks of experiments in this research.

Table 5.6: The F1 Values during the 14 Weeks of Experiments

Recommender Algorithm	Data Gathering Series No.	F1 Value	F1 Graph	Average
Random	Series #1	0.35509129		Pre test
	Series #2	0.36831924		
	Series #3	0.30430305		0.2883
	Series #4	0.26217532		
	Series #5	0.29851271		
Collaborative	Series #6	0.32069714		0.3373
	Series #7	0.36716702		
	Series #8	0.324181		
Content Based	Series #9	0.34317441		0.3460
	Series #10	0.34883662		
	Series #11	0.34593072		
ECSN	Series #12	0.36410063		0.3928
	Series #13	0.39159455		
	Series #14	0.42261639		

Based on the above-mentioned definition (Eq. 5.4), a higher F1 score value represents enhanced recommender algorithm performance in predicting the most relevant items to users. Table 5.6 illustrates an overall upward F1 measurement trend. The lowest value belongs to the random algorithm during the three weeks of experiments (0.2883), and a peak of 0.3928 was reached during the last three weeks when the ECSN algorithm was adopted. The F1 average score rose steadily during the first eleven series of feedback gathering, but it experienced a sharp rise when the ECSN algorithm was applied in the final three weeks. The data analysis from Table 5.6 illustrates that the ECSN algorithm promoted a significant contribution compared to the performance of previous recommender algorithms, with an improved F1 score by 26% for the random, 14% for collaborative, and 12% for the content-based algorithms.

The one-way Analysis of Variance (ANOVA) test can be used for the case of a quantitative outcome with a categorical explanatory variable that has two or more levels of treatment (Littell, 2006). As four different measurements (Precision, Recall, Fallout, and F1) were used in this study for computing the prediction accuracy of recommender algorithms, the four ANOVA test were run to examine if there were any between-group differences of means between studied recommender algorithms (Table 5.7).

Table 5.7: One-way ANOVA Tests for Examining the Difference of Means between Recommender Algorithms

a. ANOVA-Test on Precision Value

	Rnd	Col	Cont	ECSN
Rnd	1			
Col	0.03735	1		
Cont	0.04395*	0.00660	1	
ECSN	0.07847*	0.04111*	0.03452*	1

b. ANOVA-Test on Recall Value

	Rnd	Col	Cont	ECSN
Rnd	1			
Col	0.4932	1		
Cont	0.05807	0.00875	1	
ECSN	0.02330	0.02602	0.03477	1

c. ANOVA-Test on Fallout Value

	Rnd	Col	Cont	ECSN
Rnd	1			
Col	0.00346*	1		
Cont	0.00431*	0.0085	1	
ECSN	0.00741*	0.00395*	0.00310	1

d. ANOVA-Test on F1 Value

	Rnd	Col	Cont	ECSN
Rnd	1			
Col	0.04902*	1		
Cont	0.05765*	0.00863	1	
ECSN	0.10444*	0.05542*	0.04679*	1

* the mean difference is significant at the 0.05 level.

According to the results of LSD post hoc tests, the mean value of Precision is significantly different between ECSN and three other algorithms (Table 5.7.a) while this variation is not clear in terms of Recall measurement (Table 5.7.b). The differences are obvious also based on Fallout and F1 as shown in Table 5.7.c and Table 5.7.d. Consequently, in exception of Recall, the other measurements (Precision, Fallout, and

F1) show the significant difference of prediction accuracy between four studied recommender algorithms.

5.4.2 Solving the Cold Start Problem

The literature review in Chapter 2 showed that both the collaborative and content-based methods suffer from the cold start problem when new items or new users are involved (Shani & Gunawardana, 2011). The ECSN recommender algorithm proposed in this research utilized social networking features to solve this problem. Collected feedback from 14 weeks of experiments was analyzed to show how the ECSN recommender algorithm mitigated this issue.

Table 5.8 demonstrates the detailed statistics in this context. The second column of this table, Data Gathering Series No, lists 9 series of the data collection phase where three recommender algorithms (collaborative, content-based, and ECSN) were applied. The next three columns (New Items, New Users, and Existing Users) present the updated situation of the MyExpert system for each week of experiments based on the number of users and items. Finally, the last four columns represent the values of some parameters used to measure the cold start problem. The four parameters are:

totNRI_EU: Total Number of Recommended New Items with Prediction value >1 to Existing Users

totNRI_NU: Total Number of Recommended New Items with Prediction value >1 to New Users

AvgNRI_EU: Average Number of Recommended New Items with Prediction value >1 to Existing Users

AvgNRI_NU: Average Number of Recommended New Items with Prediction value >1 to New Users

To clarify the status of the cold start problem for both new items and new users, this research focused on the number of recommended new items to existing and new users. In this context, totNRI_EU assisted in investigating the new items problem while totNRI_NU concentrated on the new user perspective of the cold start problem. The average value of these measurements for each user shows to what extent the adopted recommender algorithm succeeded in solving the cold start issue.

Table 5.8: The Experimental Statistics in the Context of Cold Start

Algorithms	Data Gathering Series No.	Cold_Start Parameters						
		New Items	New Users	Existing Users	totNRI_EU	totNRI_NU	AvgNRI_EU	AvgNRI_NU
Collaborative	Series #6	100	8	864	0	0	**0**	**0**
	Series #7	99	5	869	0	0	**0**	**0**
	Series #8	98	6	875	0	0	**0**	**0**
Content Based	Series #9	99	7	882	3112	0	**3.528**	**0**
	Series #10	101	9	891	3156	0	**3.542**	**0**
	Series #11	100	8	899	3190	0	**3.548**	**0**
ECSN	Series #12	99	7	906	3720	27	**4.106**	**3.857**
	Series #13	100	5	911	3769	18	**4.137**	**3.6**
	Series #14	98	9	920	3792	34	**4.122**	**3.778**

As illustrated in Table 5.8, the collaborative recommender algorithm was not able to contribute at all regarding new users and new items in the cold start problem. Any new item was suggested even to existing MyExpert users. Thus, this method only applies when previously rated items are supposed to be recommended to existing users. The above-mentioned statistics therefore prove that the collaborative algorithm poses the cold start problem in both cases of new items and new users.

The content-based technique found an average of 3 new items for existing users. Nevertheless, it still encountered a problem with new users. As shown in Table 5.8, 15 new users were added to the MyExpert academic social network during the experimental series #6 to #8. According to column totNRI_NU (Total Number of Recommended Items with Prediction value >1 to New Users), the content-based approach was unable to recommend the new items to new users. So it is concluded that the new items part of the cold start problem was solved with an average of 3 recommended new items to existing users. However, the shortcoming related to new users remains in this approach.

The ECSN recommender algorithm could solve both the new users and new items issues with regards to the cold start problem. The statistics of experimental series #12 to #14 clearly show that roughly 4 new items were recommended to existing users. In the case of number of recommended items with prediction values higher than 1 to new users, the average value was around 3.7 based on column AvgNRI_NU. This means that not only has the cold start problem of previous recommender systems been solved by the ECSN algorithm, but it is also clear that the average number of recommended items to existing users with values above 1 improved by 15%.

Table 5.9: Cold Start Problem Situations in Different Recommender Algorithms

Recommender Algorithms	Cold Start Problem	
	New Items	New Users
Random	NA	NA
Collaborative	×	×
Content-based	√	×
ECSN	√	√

NA **:** Not Applicable
× **:** Problem exists
√ **:** Problem is solved

In conclusion, the cold start status in the four examined recommender algorithms (random, collaborative, content-based, and ECSN) is illustrated in Table 5.9. This problem is not applicable to the random recommender algorithm, as it only selected 10 random items and sent them to users. The collaborative algorithm suffers from the new item and new user problem, while the content-based approach solved the new item shortcoming but not the new user issue. As depicted in the last row of Table 5.8, both the new user and new item issues pertaining to the cold start problem were resolved by the ECSN recommender algorithm.

5.5 Summary

The third chapter of this dissertation clarified that the main objective of this work was to improve the performance of the content-based recommender system in an academic social network. To confirm the research contributions, it was necessary to compare the prediction accuracy of the proposed recommender algorithm (ECSN) against three others (random, collaborative, and content-based approaches). After designing and implementing all four algorithms, they were applied one at a time in MyExpert online social network to collect the relevance feedback of users for evaluation.

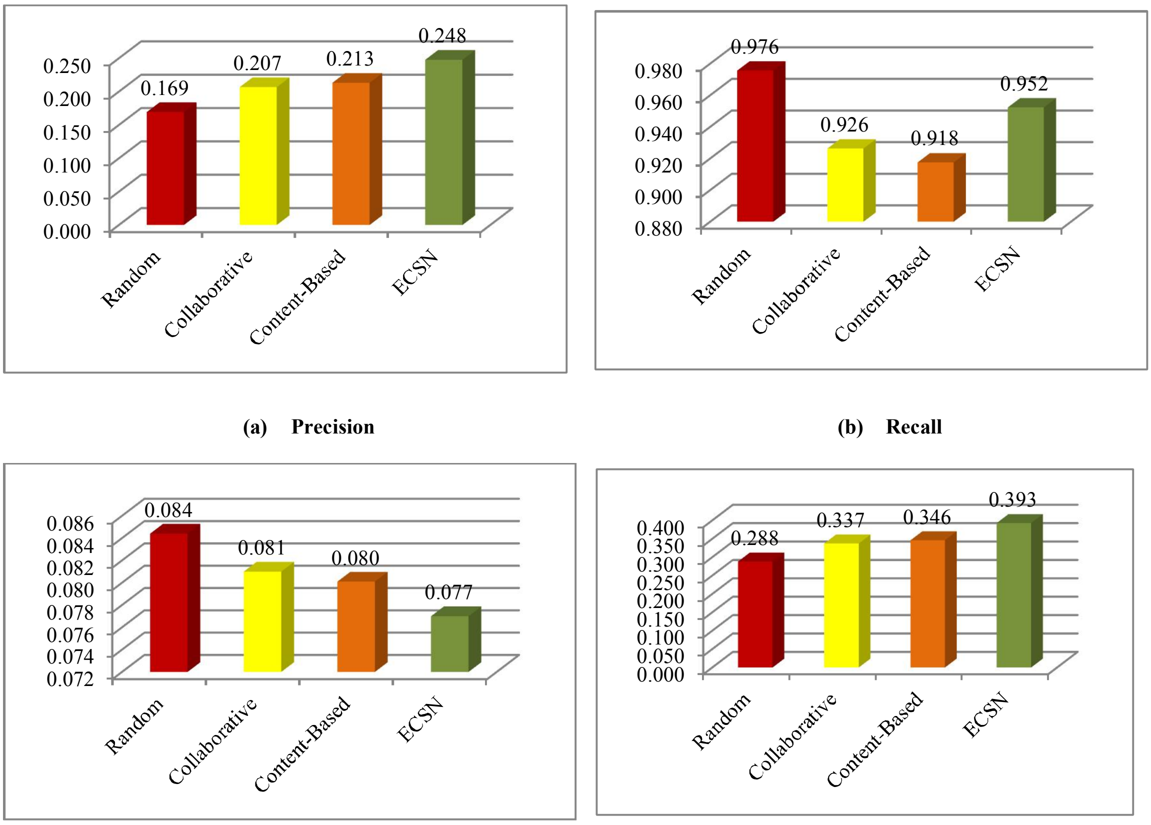
The experiments in this research ran for 14 weeks from 7th September until 26th December, 2012. In the first 5 weeks, the random algorithm was used to recommend academic items to MyExpert users. Next, the collaborative and content-based recommender systems were adopted over the following 6 weeks of data gathering. Finally, during the last 3 weeks, MyExpert members received recommendations by the ECSN algorithm to conclude the experiments for this study. In these 14 weeks of feedback collection, 1390 records of academic items were submitted to MyExpert including 346 academic jobs, 339 conferences, 355 scholarships, and 350 academic news. These items were sent to 920 MyExpert registered members from 10 universities in Malaysia.

After gathering the members' feedback from 14 weeks of online experiments, precision, recall, fallout, and F1 were measured to assess the prediction accuracy of all applied recommender algorithms. A complete view of the results based on usage prediction measurements is presented in Table 5.10.

Table 5.10: Complete View of Results based on Usage Prediction Measurements

Data Gathering Series No.	Precision		Recall		Fallout	
	Value	Graph	Value	Graph	Value	Graph
Series #1	0.217241379		0.97164751		0.079655298	
Series #2	0.225974026		0.995238095		0.07876572	
Series #3	0.180392157		0.971895425		0.083231321	
Series #4	0.151785714		0.961309524		0.086121657	
Series #5	0.175641026		0.993589744		0.083744789	
Series #6	0.191304348		0.990942029		0.082245475	
Series #7	0.228571429		0.932738095		0.078659818	
Series #8	0.2		0.855144558		0.081821206	
Series #9	0.209850746		0.941044776		0.078631394	
Series #10	0.215280702		0.918910914		0.080502074	
Series #11	0.214536		0.892632275		0.081030192	
Series #12	0.225423729		0.946166263		0.079236005	
Series #13	0.245454545		0.967820599		0.077034544	
Series #14	0.272340426		0.942907801		0.074596385	

Figure 5.2 compares the recommender algorithms examined (random, collaborative, content-based, and ECSN) based on the four stated metrics for evaluating prediction accuracy in recommender systems.



(a) Precision **(b) Recall**

(c) Fallout **(d) F1**

Figure 5.2: Comparison of Random, Collaborative, Content-Based, and ECSN Algorithms based on four Usage Prediction Metrics

Figure 5.2(a) demonstrates that the precision value experienced an upward trend throughout the 14 weeks of experiments. It started at 0.169 with the random algorithm and steadily rose to 0.207 for the collaborative and 0.213 for the content-based approach. In the last stage of experiments, the ECSN algorithm reached a peak of 0.248. The ECSN algorithm enhanced the precision value of other recommender algorithms by

32% (random, MD= 0.00741), 20% (collaborative, MD=0.00395) and 21% (content-based, MD=0.00310).

The recall value comparison for all four recommender algorithms can be seen in Figure 5.2(b). The highest rate was attained by the random (0.976) and the ECSN algorithm (0.952), while the collaborative and content-based approaches had lower recall values, at 0.926 and 0.918 respectively. Although the ANOVA test results did not show any significant differences in the case of recall metrics, and even based on this measurement, the contribution of the ECSN method is clear with an improvement over the collaborative approach by 3% and the content-based method by 4%.

The fallout values of the studied recommender algorithms are compared in Figure 5.2(c), where it is obvious that the fallout rate had a decreasing trend in the 14 weeks of experiments. Referring to the definition of fallout (Eq. 5.3), a lower fallout rate indicates better recommender algorithm performance. Thus, this diagram shows that the prediction accuracy of recommender algorithms improved from random (0.084), to collaborative (0.081), content-based (0.080), and finally the ECSN algorithm (0.077) while the fallout metric values declined steadily.

The final section of the diagram corresponds to the F1 score. As per Eq. 5.4, the values of both precision and recall were combined to calculate the F1 score for measuring the accuracy prediction of a given recommender algorithm. It is thus considered an overall metric that includes both recall and precision. Figure 5.2(d) shows the steady rise of F1 values during 14 weeks of experiments. The lowest value belongs to the random algorithm (0.288) while the peak of 0.393 corresponds to when the ECSN algorithm was applied. In other words, the ECSN recommender algorithm significantly contributed to the F1 values of the random (MD= 0.10444), collaborative

(MD=0.05542), and content-based (MD=0.04679) algorithms, by 26%, 14%, and 12% respectively.

To conclude, 14 weeks of evaluations based on the four most familiar metrics, namely precision, recall, fallout, and F1, demonstrate that the proposed recommender algorithm in this research (ECSN) successfully enhanced the prediction accuracy compared to the other studied and implemented recommender approaches.

With respect to the cold start problem, the investigation results for the collected relevance feedback from MyExpert users show that the ECSN algorithm may significantly contribute to solving this problem. In addition to addressing both new user and new item issues in the cold start context, the ECSN approach seems to improve the average value of recommended items to new users by 15% compared to the pure content-based algorithm.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

The purpose of this research was to determine how the prediction accuracy of recommender systems in academic social networks could be improved by applying an enhanced content-based algorithm utilized by social networking features (ECSN). In addition, the cold start problem of recommender systems was solved with the ECSN recommender algorithm. This chapter reflects on the aims and methods of the research and continues with a discussion of the contributions of this work. Then, future work based on this dissertation will be described.

6.1. Aims and Methods

The main objectives of the present research are as follows:

i. To elicit the techniques of recommender systems and essential features of academic social networks
ii. To propose an enhanced content-based recommender system using social networking techniques (ECSN)
iii. To develop an academic social network as a real runtime environment for evaluating recommender algorithms
iv. To evaluate the ECSN recommender system by comparing its prediction accuracy with random, collaborative and content-based recommender algorithms

The discussion that follows shows how these objectives have been achieved in this research. The first objective was to undertake an investigation on techniques and approaches in recommender systems and elicit essential features of academic social networks. In the first part of Chapter 2, the reasons why academic social networks have emerged were described and the most popular examples were discussed. The second section of the literature review focused on recommender system description. After providing some information with respect to their functionalities, the three main techniques in recommender systems were briefly presented – the collaborative, content-based, and hybrid approaches. The final part of the chapter presented the methods of evaluating the prediction accuracy of recommender systems.

The second objective entailed proposing and implementing an enhanced content-based recommender system using a social networking technique (ECSN), which improved recommendation accuracy. While the pure content-based recommender algorithm considers only the given user's preferences, the ECSN algorithm takes into account the preferences of users' friends and faculty mates. In this way, not only did the cold start problem in recommender systems get solved, but a great contribution was additionally made by the improved prediction accuracy of the recommendation process. Chapter 4 illustrated the technical issues with the ECSBN algorithm and presented the theoretical framework of this recommender algorithm.

The third objective was to design and develop MyExpert as a real runtime environment for evaluating recommender algorithms. In this study, MyExpert was employed for applying different recommender algorithms, studying user behavior, collecting relevance feedback in real conditions, and comparing the performance of the recommender systems applied against each other. The details of MyExpert design and development were presented in Chapter 4.

The final objective was to evaluate the ECSN recommender system and compare its prediction accuracy with the random, collaborative and content-based recommender algorithms. In doing so, four well-known measurements were used: precision, recall, fallout and F1. The online studies method was employed to design the research experiments and collect the relevance feedback from MyExpert academic social network users. The experiments and study results were discussed in Chapter 5

6.2. Contributions

The defining characteristic of the Internet today is the abundance of information and choice (Bonhard, et al., 2007). Considering this phenomenon, utilizing recommender systems is essential to assist users with finding the right information from a wealth of Web data (Zhou, Xu, Li, Josang, & Cox, 2012). As the first contribution, this research investigated the most popular existing recommender systems. Recommender systems are defined as software tools and techniques for suggesting the most related items to users. According to earlier research works, the collaborative and content-based recommender systems were rather successful in suggesting relevant items to target users, but they did have limitations regarding sparsity, recommending new items, and the cold start problem (Ricci, et al., 2011). Although recent alternatives like hybrid methods, demographic algorithms, and knowledge-based approaches have been proposed to address these problems, even the current generation of recommender systems surveyed in this study still requires further improvements to increase the effectiveness of recommendation methods (Pu, Chen, & Hu, 2012).

Referring to the cold start phenomenon, RSs cannot make reliable recommendations in situations when new items should be recommended or new users are expecting to receive recommendations from the system. It has been argued in previous research in this domain that recommender systems could be significantly improved by drawing on features from social systems (Bonhard, 2005). In other words, traditional recommender systems ignore the social relationships among users. But in real life, when we seek advice from our friends, we are actually requesting verbal social recommendations. In a related research in this field, Bonhard, Sasse and Harries (2007) stated that recommender systems and social networking functionality should be integrated. Hence, in order to improve recommender systems and to provide more personalized recommendation results, the incorporation of social network information among users is a must (Zhou, et al., 2012). In addition, enhancing the prediction accuracy of recommender algorithms is recognized as a cutting-edge research subject in the realm of recommender systems and information retrieval (Adomavicius & Tuzhilin, 2005).

To fill the acknowledged gap, this study proposed an enhanced version of a content-based recommender system by drawing on the social networking feature (ECSN). The ECSN recommender algorithm was applied in academic social networks to suggest the most relevant academic items to members of these online societies. In addition to considering the user's own preferences, this algorithm took advantage of the interests and preferences of the user's friends and faculty mates to provide more accurate recommendations. A hierarchy tree structure was used to store the preference scores for all academic items. When users clicked on, or rated academic items, the related node of a given item in this hierarchy tree of preferences was established, after which the related point of that node for the given user was updated. The higher point value for each node resemble that it is more interesting for studied user. Definition 4.1 in Chapter 4 identified this tree structure. Also, the mechanism for updating the scores of each node was represented by Definition 4.2. After updating the preference scores for each user by analyzing his preferences, as well as his friends' and faculty mates' preferences, the ordered list of recommended items was sent through a MyExpert e-Newsletter. The details of this process were concisely discussed in Chapter 4.

To investigate the contribution made by this research, 14 weeks of experiments (from 7th September until 26th December) were carried out by adopting the online method. In that time, the accuracy prediction of the proposed recommender algorithm (ECSN) was compared with three others (random, collaborative, and content-based approaches). These four algorithms were applied one by one in the MyExpert online social network to collect the relevance feedback of users for the evaluation process.

In the first 5 weeks of experiments, the random algorithm was used to recommend academic items to MyExpert users. Next, the collaborative and content-based recommender systems were adopted over the following 6 weeks of data gathering. Finally, during the last 3 weeks, MyExpert members received recommendations by the ECSN algorithm to conclude the experiments for this study. In these 14 weeks of feedback collection, 1390 records of academic items were submitted to MyExpert including 346 academic jobs, 339 conferences, 355 scholarships, and 350 academic news. These items were sent to 920 MyExpert registered members from 10 universities in Malaysia. After gathering the members' feedback from 14 weeks of online experiments, precision, recall, fallout, and F1 were measured to assess the prediction accuracy of all applied recommender algorithms.

According to the online experiment results, the precision value had an upward trend starting at 0.1693 with the random algorithm and steadily rising to 0.2066 for the collaborative and 0.2132 for the content-based approach. In the last stage of the experiments, the ECSN algorithm reached a peak of 0.2477. Clearly, the ECSN algorithm enhanced the precision value of other recommender algorithms by 32% (random), 20% (content-based) and 21% (collaborative).

The highest rates were attained by the random (0.976) and ECSN algorithm (0.952), while the collaborative and content-based approaches had lower recall values, at 0.926 and 0.918 respectively. The ANOVA test results did not show any significant differences in the case of the recall metrics, and even so, the contribution of the ECSN method is clear with an improvement over the collaborative technique by 3% and content-based method by 4%.

The fallout rate had a decreasing trend in the 14 weeks of experiments. Referring to the definition of fallout (Eq. 5.5), a lower fallout rate indicates better recommender algorithm performance. Thus, the diagram shows that the prediction accuracy of recommender algorithms improved from random (0.084), to collaborative (0.081), content-based (0.080), and finally ECSN (0.077), while the values of the fallout metric declined steadily.

F1 is considered an overall metric that includes both recall and precision. The results show the steady rise of F1 values during 14 weeks of experiments. The lowest value belongs to the random algorithm (0.288) while the peak of 0.393 corresponds to when the ECSN algorithm was applied. In other words, the ECSN recommender algorithm significantly contributed to the F1 values of the random (MD= 0.10444), collaborative

(MD=0.05542), and content-based (MD=0.04679) algorithms, by 26%, 14%, and 12% respectively.

To conclude, 14 weeks of evaluations based on the four most familiar metrics, namely precision, recall, fallout, and F1, demonstrate that the proposed recommender algorithm in this research (ECSN) successfully enhanced the prediction accuracy compared to the other studied and implemented recommender approaches.

Pertaining to the cold start problem, the investigation results on collected relevance feedback from MyExpert users indicate that the ECSN algorithm may significantly contribute to resolving this issue. In addition to addressing both the new user and new item issues in the cold start context, the ECSN approach could improve the average value of recommended items to new users by 15% compared to the pure content-based algorithm.

6.3. Future work

During the course of this research several potentially interesting and relevant subjects presented themselves, but to keep focused on the objectives of this study, the topics had to be abandoned. This section concisely addresses these subjects for possible future work.

Social network analysis (SNA) is a leading-edge research topic that studies the structure of social networks (Scott, 2012). It also analyzes the structure of organizations and enterprises to improve their policies in allocating resources among users and organizational nodes. As a novel topic for future research, the social network-based features utilized in this study could be combined with SNA concepts to offer a new model for enhancing organizational behaviors and recommendations (Zhou, Xu, Li, Josang, & Cox, 2012). The results of such research may assist in decision support systems (DSS).

A further question that requires more extensive research is whether it is possible to optimize the weights considered for calculating the node scores in Definition 4.2. In this research, based on the degree of importance, a weight of 5 was considered for users' own preferences, a weight of 3 for taking into account faculty mates, and finally, for applying the preferences of friends a weight of 1 was considered. Although these weights could make a significant contribution in solving the cold start problem as well as in improving the prediction accuracy of recommendations, it may be better to apply fuzzy logic or neural networks techniques to achieve more favorable weights (Zenebe et al, 2009).

Furthermore, the MyExpert academic social network developed in this study as the runtime environment for establishing online experiments has the potential to be used in future research works. Anomaly detection and community studies are two fields of related research that can benefit from this runtime environment to establish experiments and archive online results.

References

About Academia.edu. (2013). Retrieved from http://www.academia.edu/about.

About LinkedIn. (2013). Retrieved February 11, 2013 from the World Wide Web : http://press.linkedin.com/about.

Acker, S. R. (2008). Will Digital Texts Succeed? Campus Technology, 21(7), 54-56.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6), 734-749.

Afzal, M. T., & Maurer, H. (2011). Expertise Recommender System for Scientific Community. Journal of Universal Computer Science, 17(11), 1529-1549.

Alexa Site Info. (2013). retrieved February 10, 2013 from the World Wide Web : http://www.alexa.com/siteinfo/linkedin.com.

Allan, J., Carbonell, J. G., Doddington, G., Yamron, J., & Yang, Y. (1998). Topic detection and tracking pilot study final report. Paper presented at the DARPA Broadcast News Transcription and Understanding Workshop, Lansdowne, Virginia.

Ansari, A., Essegaier, S., & Kohli, R. (2000). Internet recommendation systems. Journal of Marketing research, 37(3), 363-375.

Arazy, O., Kumar, N., & Shapira, B. (2009). Improving social recommender systems. IT professional, 11(4), 38-44.

Armstrong, J. S. (2001). Principles of forecasting: a handbook for researchers and practitioners (Vol. 30): Springer.

Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. Communications of the ACM, 40(3), 66-72.

Bambini, R., Cremonesi, P., & Turrin, R. (2011). A recommender system for an iptv service provider: a real large-scale production environment. Recommender Systems Handbook, 299-331.

Basilico, J., & Hofmann, T. (2004). Unifying collaborative and content-based filtering. Paper presented at the Proceedings of the twenty-first international conference on Machine learning.

Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation. Paper presented at the Proceedings of the national conference on artificial intelligence.

Bezos, J. (March 16th 1999). [Interview]. BusinessWeek, http://www.businessweek.com/ebiz/9903/316bezos.htm,.

Billsus, D., & Pazzani, M. J. (1998). Learning collaborative information filters. Paper presented at the Proceedings of the Fifteenth International Conference on Machine Learning.

Billsus, D., & Pazzani, M. J. (1999). A hybrid user model for news story classification. Courses and lectures-international centre for mechanical sciences, 99-108.

Billsus, D., Pazzani, M. J., & Chen, J. (2000). A learning agent for wireless news access. Paper presented at the Proceedings of the 5th international conference on Intelligent user interfaces, New York, NY, USA.

Bisca, G. (2011). Interview of Mike Moradian, Founder of CampusBuddy. Retrieved February 12, 2013 from the World Wide Web : http://www.thestartupangel.com/2011/08/19/interview-of-mike-moradian-founder-of-campusbuddy/.

Bishop, T. (2011). Classmates.com brand demoted in 'Memory Lane' site launch. Retrieved February 8, 2013 from the World Wide Web (TechFlash) : http://www.bizjournals.com/seattle/blog/techflash/2011/02/classmatescom-gets-demoted-in-memory.html.

Bonhard, P. (2005). Who do trust? Combining recommender systems and social networking for better advice. Paper presented at the Proceedings of the Wokshop Beyond Personalization 2005, in conjunction with the International Conference on Intelligent User Interfaces IUI'05.

Bonhard, P., Harries, C., McCarthy, J., & Sasse, M. A. (2006). Accounting for taste: using profile similarity to improve recommender systems. Paper presented at the Proceedings of the SIGCHI conference on Human Factors in computing systems.

Bonhard, P., Sasse, M. A., & Harries, C. (2007). The devil you know knows best: how online recommendations can benefit from social networking. Paper presented at the Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it-Volume 1.

Bridge, D., Göker, M. H., McGinty, L., & Smyth, B. (2005). Case-based recommender systems. The Knowledge Engineering Review, 20(03), 315-320.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. User modeling and user-adapted interaction, 6(2), 87-129.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4), 331-370.

Burke, R. (2007). Hybrid web recommender systems. The adaptive web, 377-408.

Cacheda, F., Carneiro, V., Fernández, D., & Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. ACM Transactions on the Web (TWEB), 5(1), 2.

Chen, Y., Harper, F. M., Konstan, J., & Xin Li, S. (2010). Social comparisons and contributions to online communities: A field experiment on movielens. The American Economic Review, 100(4), 1358-1398.

Cheung, C. M. K., Chiu, P. Y., & Lee, M. K. O. (2011). Online social networks: Why do students use facebook? Computers in Human Behavior, 27(4), 1337-1343.

Chirita, P. A., Nejdl, W., & Zamfir, C. (2005). Preventing shilling attacks in online recommender systems. Paper presented at the Proceedings of the 7th annual ACM international workshop on Web information and data management.

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

Classmates.com site info. (2013). retrieved February 3, 2013 from the World Wide Web : http://www.alexa.com/siteinfo/classmates.com.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. Paper presented at the Proceedings of ACM SIGIR Workshop on Recommender Systems.

Cleverdon, C., Mills, J., & Keen, M. (1966). Factors determining the performance of indexing systems volume 1. Design.

Collaborative Features. (2013). Retrieved from http://www.mendeley.com/features/collaborate/.

Cohen, W. W. (1995). Fast effective rule induction. Paper presented at the Machine Learning International Workshop

Cohen, W. W., & Hirsh, H. (1998). Joins that generalize: Text classification using WHIRL. Paper presented at the Proceedings of KDD-98, 4th International Conference on Knowledge Discovery and Data Mining.

Cremonesi, P., Turrin, R., & Airoldi, F. (2011). Hybrid algorithms for recommending new items. Paper presented at the Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems.

De Gemmis, M., Iaquinta, L., Lops, P., Musto, C., Narducci, F., & Semeraro, G. (2009). Preference learning in recommender systems. PREFERENCE LEARNING, 41.

de Oliveira, J., Lopes, G., & Moro, M. (2011). Academic social networks. Advances in Conceptual Modeling. Recent Developments and New Directions, 2-3.

Deshpande, M., & Karypis, G. (2004). Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS), 22(1), 143-177.

Duda, R. O., & Hart, P. E. (1973). Pattern classification and scene analysis (Vol. 3): Wiley New York.

Durao, F., & Dolog, P. (2010). Extending a hybrid tag-based recommender system with personalization. Paper presented at the Proceedings of the 2010 ACM Symposium on Applied Computing.

Easterly, W., & Levine, R. (2001). What have we learned from a decade of empirical research on growth? It's Not Factor Accumulation: Stylized Facts and Growth Models. The World Bank Economic Review, 15(2), 177-219.

Fischer, G. (2001). User modeling in human–computer interaction. User modeling and user-adapted interaction, 11(1), 65-86.

Ge, Y., Xiong, H., Tuzhilin, A., Xiao, K., Gruteser, M., & Pazzani, M. (2010). An energy-efficient mobile recommender system. Paper presented at the Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.

Ghazanfar, M. A., & Prugel-Bennett, A. (2010). A scalable, accurate hybrid recommender system. Paper presented at the Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on.

Giglia, E. (2011). Academic social networks: it’s time to change the way we do research. European Journal of Physical and Rehabilitation Medicine.

Golbeck, J. (2006). Generating predictive movie recommendations from trust in social networks. Trust Management, 93-104.

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.

Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. Journal of Software, 5(7), 745-752.

Good, I. J. (1965). Speculations concerning the first ultraintelligent machine. Advances in computers, 6(31), 88.

Granger, C. W., & Machina, M. J. (2006). Forecasting and decision theory. Handbook of economic forecasting, 1, 81-98.

Groh, G., & Ehmig, C. (2007). Recommendations in taste related domains: collaborative filtering vs. social filtering. Paper presented at the Conference on Supporting Group Work: Proceedings of the 2007 international ACM conference on Conference on supporting group work.

Guy, I., & Carmel, D. (2011). Social recommender systems. Paper presented at the Proceedings of the 20th international conference companion on World wide web.

Guy, I., Zwerdling, N., Carmel, D., Ronen, I., Uziel, E., Yogev, S., & Ofek-Koifman, S. (2009). Personalized recommendation of social software items based on social relations. Paper presented at the Proceedings of the third ACM conference on Recommender systems.

Guy, I., Zwerdling, N., Ronen, I., Carmel, D., & Uziel, E. (2010). Social media recommendation based on people and tags. Paper presented at the Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval.

Harman, D. (1995). Overview of the second text retrieval conference (TREC-2). Information Processing & Management, 31(3), 271-289.

Harter, S. P. (1996). Variations in relevance assessments and the measurement of retrieval effectiveness. Journal of the American Society for Information Science, 47(1), 37-49.

Hayes, C., & Cunningham, P. (2001). Smart radio—community based music radio. Knowledge-Based Systems, 14(3), 197-201.

He, J., & Chu, W. W. (2010). A social network-based recommender system (SNRS): Springer.

Herlocker, J. L. (2000). Understanding and improving automated collaborative filtering systems. PhD Thesis, Citeseer.

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. Paper presented at the Proceedings of the 2000 ACM conference on Computer supported cooperative work.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 5-53.

Hernández del Olmo, F., & Gaudioso, E. (2008). Evaluation of recommender systems: A new approach. Expert Systems with Applications, 35(3), 790-804.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. Machine learning, 42(1), 177-196.

Holte, R., & Yan, J. (1996). Inferring what a user is not interested in. Advances in Artifical Intelligence, 159-171.

Hwang, C. S., Kuo, N., & Yu, P. (2008). Representative-based diversity retrieval. Paper presented at the ICICIC '08. The 3rd Intetnational Conference on Innovative Computing Information and Control.

Jafari, A. (2012). What is CourseNetworking. Retrieved February 15, 2013 from the World Wide Web : http://www.coursenetworking.com/pages/?p=faq.

Jafari, A. (Sept. 22, 2011). IU professor launches pioneering social network for classrooms Retrieved February 10, 2013 from the World Wide Web : http://www.coursenetworking.com/pages/?p=press.

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). Recommender systems: an introduction: Cambridge University Press.

Jheng, S. L. (2011). Enterprise Email Classification Based on Social Network Features. [Conference Publications]. International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 532 - 536

Jin, R., Chai, J. Y., & Si, L. (2004). An automatic weighting scheme for collaborative filtering. Paper presented at the Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features: Springer.

Jung, J. J. (2011). Attribute selection-based recommendation framework for long-tail user group: An empirical study on movielens dataset. Expert Systems with Applications.

Kim, J. W., Lee, B. H., Shaw, M. J., Chang, H.-L., & Nelson, M. (2001). Application of decision-tree induction techniques to personalized advertisements on internet storefronts. International Journal of Electronic Commerce, 5, 45-62.

Kim, J. W., Lee, K. M., Shaw, M. J., Chang, H.-L., Nelson, M., & Easley, R. F. (2006). A preference scoring technique for personalized advertisements on Internet storefronts. Mathematical and computer modelling, 44(1), 3-15.

Kim, Y. S., Mahidadia, A., Compton, P., Krzywicki, A., Wobcke, W., Cai, X., & Bain, M. (2012). People-to-People recommendation using multiple compatible subgroups. Paper presented at the Proceedings of the 25th Australasian joint conference on Advances in Artificial Intelligence.

Kivinen, J., & Warmuth, M. K. (1997). Exponentiated gradient versus gradient descent for linear predictors. Information and Computation, 132(1), 1-63.

Kohavi, R., Longbotham, R., Sommerfield, D., & Henne, R. M. (2009). Controlled experiments on the web: survey and practical guide. Data mining and knowledge discovery, 18(1), 140-181.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

Lam, S. K., & Riedl, J. (2004). Shilling recommender systems for fun and profit. Paper presented at the Proceedings of the 13th international conference on World Wide Web.

Lam, X. N., Vu, T., Le, T. D., & Duong, A. D. (2008). Addressing cold-start problem in recommendation systems. Paper presented at the Proceedings of the 2nd international conference on Ubiquitous information management and communication.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval Machine learning: ECML-98 (pp. 4-15): Springer.

Lewis, D. D., Schapire, R. E., Callan, J. P., & Papka, R. (1996). Training algorithms for linear text classifiers. Paper presented at the Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval.

Lilien, G. L., Kotler, P., & Moorthy, K. S. (1992). Marketing models: Prentice Hall.

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

Lipczak, M., Sigurbjörnsson, B., & Jaimes, A. (2012). Understanding and Leveraging Tag-based Relations in On-line Social Networks.

Littell, R. C. (2006). SAS for mixed models: SAS institute.

Lops, P., Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. Recommender Systems Handbook, 73-105.

Ma, H., Zhou, D., Liu, C., Lyu, M. R., & King, I. (2011). Recommender systems with social regularization. Paper presented at the Proceedings of the fourth ACM international conference on Web search and data mining.

Mahmood, T., & Ricci, F. (2007a). Learning and adaptivity in interactive recommender systems. Paper presented at the Proceedings of the ninth international conference on Electronic commerce.

Mahmood, T., & Ricci, F. (2007b). Towards learning user-adaptive state models in a conversational recommender system. Paper presented at the LWA.

Mahmood, T., & Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. Paper presented at the the 20th ACM conference on Hypertext and hypermedia New York, NY, USA.

Mangan, K. (2012). Social Networks for Academics Proliferate, Despite Some Doubts. Retrieved from http://chronicle.com/article/Social-Networks-for-Academics/131726/.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval (Vol. 1): Cambridge University Press Cambridge.

Maron, M. (1961). Automatic indexing: an experimental inquiry. Journal of the ACM (JACM), 8(3), 404-417.

Martinez, L., Barranco, M. J., Perez, L. G., & Espinilla, M. (2008). A knowledge based recommender system with multigranular linguistic information. International Journal of Computational Intelligence Systems, 1(3), 225-236.

Massa, P., & Avesani, P. (2004). Trust-aware collaborative filtering for recommender systems. On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, 492-508.

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. Paper presented at the AAAI-98 workshop on learning for text categorization.

McSherry, F., & Mironov, I. (2009). Differentially private recommender systems: building privacy into the net.

Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. Paper presented at the Proceedings of the National Conference on Artificial Intelligence.

Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2004). Ontological user profiling in recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 54-88.

Miller, B. N., Riedl, J., & Konstan, J. A. (2003). GroupLens for Usenet: Experiences in applying collaborative filtering to a social information system. From Usenet to CoWebs: Interacting With Social Information Spaces, 206-231.

Mitchell, T. M. (1997). Machine learning. WCB: McGraw-Hill Boston, MA:.

Mladenic, D. (1999). Text-learning and related intelligent agents: a survey. Intelligent Systems and their Applications, IEEE, 14(4), 44-54.

Mobasher, B., Burke, R., Bhaumik, R., & Williams, C. (2007). Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Transactions on Internet Technology (TOIT), 7(4), 23.

Mooney, R. J., Bennett, P. N., & Roy, L. (1998). Book recommending using text categorization with extracted information. Paper presented at the Proc. Recommender Systems Papers from 1998 Workshop, Technical Report WS-98-08.

Mosayebian, s., keramati, a., & khatibi, v. (2012). A retailing recommender system based on customers'demographic features using som and association rules. International journal of industrial engineering and production management (ijie) (international journal of engineering science) (persian).

Murthi, B., & Sarkar, S. (2003). The role of the management sciences in research on personalization. Management Science, 49(10), 1344-1362.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (1998). Learning to classify text from labeled and unlabeled documents. Paper presented at the Proceedings of the national conference on artificial intelligence.

Nguyen, H., & Dinh, T. (2012). A Modified Regularized Non-Negative Matrix Factorization for MovieLens. Paper presented at the Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2012 IEEE RIVF International Conference on.

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

Pazzani, M., & Billsus, D. (2007). Content-based recommendation systems. The adaptive web, 325-341.

Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 13(5), 393-408.

Pazzani, M. J., Muramatsu, J., & Billsus, D. (1996). Syskill & Webert: Identifying interesting web sites. Paper presented at the Proceedings of the national conference on artificial intelligence.

Perez, L. G., Barranco, M., & Martinez, L. (2007). Building user profiles for recommender systems from incomplete preference relations. Paper presented at the Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International.

Perez, S. (2012). Classmates.com Parent United Online Acquires School-Focused Friend Finder SchoolFeed. Retrieved February 4, 2013 from the World Wide Web :http://techcrunch.com/2012/06/11/classmates-com-parent-united-online-acquires-school-focused-friend-finder-schoolfeed/.

Picard, R. W. (2003). Affective computing: challenges. International Journal of Human-Computer Studies, 59(1), 55-64.

Popescul, A., Pennock, D. M., & Lawrence, S. (2001). Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. Paper presented at the Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence.

Porcel, C., Tejeda-Lorente, A., Martínez, M., & Herrera-Viedma, E. (2012). A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. Information Sciences, 184(1), 1-19.

Porter, M. F. (1980). An algorithm for suffix stripping. Program: electronic library and information systems, 14(3), 130-137.

Powell, M. J. D. (1981). Approximation theory and methods: Cambridge university press.

Professor Facebook. (2012). Retrieved February 11, 2013 from the World Wide Web : http://www.economist.com/node/21547218.

Pu, P., Chen, L., & Hu, R. (2012). Evaluating recommender systems from the user's perspective: survey of the state of the art. User modeling and user-adapted interaction, 1-39.

Quantcast Audience Profile: Demographics. (2011). Retrieved March 11, 2012 from the World Wide Web : http://www.quantcast.com/classmates.com/demographics.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

Rocchio, J. J. (1971). Relevance feedback in information retrieval.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. Paper presented at the Proceedings of the 1994 ACM conference on Computer supported cooperative work.

Ricci, F., Cavada, D., Mirzadeh, N., & Venturini, A. (2006). Case-based travel recommendations. Destination Recommendation Systems: Behavioural Foundations and Applications, 67-93.

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. Recommender Systems Handbook, 1-35.

Rich, E. (1979). User modeling via stereotypes. Cognitive science, 3(4), 329-354.

Rohani, V. A., & Ow, S. H. (2011). Eliciting essential requirements for social networks in academic environments. Paper presented at the Computers & Informatics (ISCI), 2011 IEEE Symposium on.

Salton, G. (1989a). Automatic Text Processing: The Transformation, Analysis, and Retrieval of: Addison-Wesley.

Salton, G. (1989b). Automatic Text Processing: The Transformation, Analysis, and Retrieval of information by computer: Addison-Wesley.

Sanderson, M., & Croft, W. B. (2012). The history of information retrieval research. Proceedings of the IEEE, 100(Special Centennial Issue), 1444-1451.

Sandvig, J. J., Mobasher, B., & Burke, R. (2008). A survey of collaborative recommendation and the robustness of model-based algorithms. IEEE Data Engineering Bulletin, 31(1), 3-13.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Paper presented at the Proceedings of the 10th international conference on World Wide Web.

Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., & Riedl, J. (1998). Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system. Paper presented at the Proceedings of the 1998 ACM conference on Computer supported cooperative work.

Schafer, J., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. The adaptive web, 291-324.

Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. Data mining and knowledge discovery, 5(1), 115-153.

Schaffer, N. (2009). What is LinkedIn and Why Should You Join? Retrieved February 12, 2013 from the World Wide Web : http://windmillnetworking.com/2009/08/26/what-is-linkedin-and-why-should-you-join/.

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. Paper presented at the Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.

Scott, J. (2012). Social network analysis: SAGE Publications Limited.

Seth, A., & Zhang, J. (2008). A social network based approach to personalized recommendation of participatory media content: ICWSM.

Shani, G., Brafman, R. I., & Heckerman, D. (2002). An MDP-based recommender system. Paper presented at the UAI'02 Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence.

Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. Recommender Systems Handbook, 257-297.

Shardanand, U., & Maes, P. (1995). Social information filtering: algorithms for automating “word of mouth”. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.

Smeaton, A. F., & Callan, J. (2005). Personalisation and recommender systems in digital libraries. International Journal on Digital Libraries, 5(4), 299-308.

Soboroff, I., & Nicholas, C. (1999). Combining content and collaboration in text filtering. Paper presented at the Proc. Int’l Joint Conf. Artificial Intelligence Workshop: Machine Learning for Information Filtering.

Sumra, H. (November 17, 2009). Web site can help students choose professors. Spartan Daily.

Szkolar, D. (2012). Social Networking for Academics and Scholars. Retrieved from http://infospace.ischool.syr.edu/2012/06/21/social-networking-for-academics-and-scholars/.

Tchuente, D., Canut, M.-F., Baptiste-Jessel, N., Peninou, A., & Sedes, F. (2012). A Community Based Algorithm for Deriving Users' Profiles from Egocentrics Networks. Paper presented at the Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on.

Team, R. D. (2012). Two Million Members, Two Million Stories. Retrieved February 11, 2013 from the World Wide Web : http://news.researchgate.net/index.php?/archives/160-Two-Million-Members,-Two-Million-Stories.html.

Tiroshi, A., Kuflik, T., Kay, J., & Kummerfeld, B. (2012). Recommender systems and the social web. Advances in User Modeling, 60-70.

Tochel, C., Haig, A., Hesketh, A., Cadzow, A., Beggs, K., Colthart, I., & Peacock, H. (2009). The effectiveness of portfolios for post-graduate assessment and education: BEME Guide No 12. Medical teacher, 31(4), 299-318.

Ungar, L. H., & Foster, D. P. (1998). Clustering methods for collaborative filtering. Paper presented at the AAAI Workshop on Recommendation Systems.

Verhoeyen, M., Vriendt, J. D., & Vleeschauwer, D. D. (2012). Optimizing for Video Storage Networking With Recommender Systems. Bell Labs Technical Journal, 16(4), 97-113.

Victor. (2012). Mendeley has two million users! To celebrate, we're releasing the Global Research Report. Retrieved February 2, 2013 from the World Wide Web :http://blog.mendeley.com/academic-life/mendeley-has-two-million-users-to-celebrate-were-releasing-the-global-research-report/.

Voorhees, E. M. (2000). Variations in relevance judgments and the measurement of retrieval effectiveness. [Article]. Information Processing & Management, 36(5), 697-716. doi: 10.1016/s0306-4573(00)00010-8

Wang, J., De Vries, A. P., & Reinders, M. J. T. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. Paper presented at the Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval.

Wang, P.-Y., & Yang, H.-C. (2012). Using collaborative filtering to support college students' use of online forum for English learning. Computers & Education.

Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. Information retrieval, 1(1-2), 69-90.

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. Paper presented at the Machine Learning International Workshop.

Yoshii, K., Goto, M., Komatani, K., Ogata, T., & Okuno, H. G. (2008). An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. Audio, Speech, and Language Processing, IEEE Transactions on, 16(2), 435-447.

Zenebe, A., & Norcio, A. F. (2009). Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems. Fuzzy Sets and Systems, 160(1), 76-94.

Zhang, T., & Iyengar, V. S. (2002). Recommender systems using linear classifiers. The Journal of Machine Learning Research, 2, 313-334.

Zhang, Z.-K., Liu, C., Zhang, Y.-C., & Zhou, T. (2010). Solving the cold-start problem in recommender systems with social tags. EPL (Europhysics Letters), 92(2), 28002.

Zhou, X., Xu, Y., Li, Y., Josang, A., & Cox, C. (2012). The state-of-the-art in personalized recommender systems for social networking. Artificial Intelligence Review, 37(2), 119-132.

Appendix A

Samples of Collected Relevance Feedbacks

Samples of Collected Relevance Feedback

from MyExpert Online Environemnt

fID	fNewsID	fMemD	fRankDateTime	fRank	fEnterDateTime	fIPAddress	fScienceID	fType
1783	78	391	2012-10-06 12:46:05.690	5	2012-10-06 12:44:19.120	210.195.121.222	1	S
1784	93	391	2012-10-06 12:46:39.797	3	2012-10-06 12:44:29.060	210.195.121.222	85	C
1785	91	391	2012-10-06 12:47:00.390	5	2012-10-06 12:46:53.750	210.195.121.222	13	C
1908	112	850	2012-10-09 23:10:11.407	5	2012-10-09 23:10:07.473	175.138.173.120	98	S
1910	151	850	2012-10-09 23:10:18.023	4	2012-10-09 23:10:14.997	175.138.173.120	13	N
1911	145	850	2012-10-09 23:10:31.617	5	2012-10-09 23:10:28.107	175.138.173.120	13	C
1916	150	865	2012-10-09 23:15:05.623	5	2012-10-09 23:14:56.587	175.138.173.120	13	N
1917	135	865	2012-10-09 23:15:26.620	4	2012-10-09 23:15:21.960	175.138.173.120	13	C
1918	143	865	2012-10-09 23:15:38.413	4	2012-10-09 23:15:34.803	175.138.173.120	13	C
1960	118	751	2012-10-10 05:21:49.727	3	2012-10-10 05:21:44.930	120.28.136.92	1	S
1961	111	751	2012-10-10 05:22:10.360	3	2012-10-10 05:22:07.263	120.28.136.92	70	J
1962	124	751	2012-10-10 05:22:38.453	3	2012-10-10 05:22:34.177	120.28.136.92	26	S
1963	111	751	2012-10-10 05:22:49.807	3	2012-10-10 05:22:46.540	120.28.136.92	70	J
1964	135	751	2012-10-10 05:23:02.333	3	2012-10-10 05:22:58.787	120.28.136.92	13	C
1965	124	751	2012-10-10 05:23:21.677	3	2012-10-10 05:23:17.803	120.28.136.92	26	C
1966	119	751	2012-10-10 05:23:36.717	3	2012-10-10 05:23:31.670	120.28.136.92	13	J
1968	108	901	2012-10-10 05:30:45.687	3	2012-10-10 05:30:03.590	175.136.162.147	13	J
1969	129	901	2012-10-10 05:31:55.760	4	2012-10-10 05:30:06.080	175.136.162.147	109	S
1971	130	901	2012-10-10 05:32:24.223	1	2012-10-10 05:32:03.497	175.136.162.147	13	C
1974	111	901	2012-10-10 05:32:48.010	5	2012-10-10 05:32:08.540	175.136.162.147	70	S
1975	137	901	2012-10-10 05:33:11.557	1	2012-10-10 05:32:10.127	175.136.162.147	13	C
2017	117	928	2012-10-10 10:29:12.190	3	2012-10-10 10:28:55.817	115.133.211.86	1	S
2019	113	928	2012-10-10 10:30:52.570	3	2012-10-10 10:30:22.363	115.133.211.86	18	J
2020	112	928	2012-10-10 10:31:13.587	1	2012-10-10 10:30:58.787	115.133.211.86	70	J
2021	134	928	2012-10-10 10:31:29.310	1	2012-10-10 10:31:23.303	115.133.211.86	70	N
2022	117	928	2012-10-10	4	2012-10-10	115.133.211.86	1	S

			10:31:55.223		10:31:35.140			
2024	113	928	2012-10-10 10:32:10.150	4	2012-10-10 10:31:59.253	115.133.211.86	18	J
2025	138	928	2012-10-10 10:32:29.803	1	2012-10-10 10:32:18.897	115.133.211.86	13	C
2026	126	928	2012-10-10 10:32:45.323	2	2012-10-10 10:32:35.357	115.133.211.86	70	J
2027	143	928	2012-10-10 10:32:55.827	1	2012-10-10 10:32:52.157	115.133.211.86	55	N
2028	127	928	2012-10-10 10:33:06.383	2	2012-10-10 10:33:00.403	115.133.211.86	26	C
2029	123	928	2012-10-10 10:33:22.753	2	2012-10-10 10:33:10.247	115.133.211.86	2	J
2030	129	928	2012-10-10 10:33:38.950	4	2012-10-10 10:33:32.043	115.133.211.86	109	S
2031	130	718	2012-10-10 10:34:01.557	3	2012-10-10 10:33:56.280	60.48.48.138	13	C
2033	131	718	2012-10-10 10:35:05.743	3	2012-10-10 10:35:03.910	60.48.48.138	95	S
2034	143	718	2012-10-10 10:35:21.523	3	2012-10-10 10:35:12.547	60.48.48.138	13	C
2035	108	718	2012-10-10 10:35:58.063	3	2012-10-10 10:35:54.060	60.48.48.138	13	J
2075	151	391	2012-10-10 20:07:49.480	2	2012-10-10 20:07:41.477	161.142.24.130	13	N
2076	128	391	2012-10-10 20:08:09.987	5	2012-10-10 20:07:55.450	161.142.24.130	2	S
2077	118	391	2012-10-10 20:08:56.933	5	2012-10-10 20:08:53.217	161.142.24.130	1	S
2078	111	391	2012-10-10 20:10:05.907	1	2012-10-10 20:10:02.707	161.142.24.130	70	S
2079	110	391	2012-10-10 20:10:13.247	1	2012-10-10 20:10:09.847	161.142.24.130	13	J
2099	116	11	2012-10-11 08:13:19.227	4	2012-10-11 08:11:11.960	5.9.226.26	117	S
2100	110	11	2012-10-11 08:13:09.920	4	2012-10-11 08:12:47.933	5.9.226.26	13	J
2101	112	11	2012-10-11 08:13:44.467	3	2012-10-11 08:13:36.897	5.9.226.26	70	J
2106	145	955	2012-10-11 12:15:53.617	1	2012-10-11 12:11:17.813	175.142.217.9	13	C
2336	130	828	2012-10-17 00:45:45.717	4	2012-10-17 00:33:39.117	120.141.131.167	5	J
2337	139	828	2012-10-17 00:45:34.500	5	2012-10-17 00:33:47.290	120.141.131.167	95	S
2346	152	323	2012-10-17 00:56:37.143	1	2012-10-17 00:56:31.230	175.136.160.114	55	C
2347	169	323	2012-10-17 00:57:04.607	1	2012-10-17 00:56:49.467	175.136.160.114	2	C
2355	139	1	2012-10-17 03:20:11.763	5	2012-10-17 03:20:06.387	60.54.40.192	95	S
2360	147	127	2012-10-17 03:27:03.287	3	2012-10-17 03:26:41.123	159.20.111.138	95	J
2384	144	850	2012-10-17 09:09:07.513	5	2012-10-17 09:08:45.700	175.138.173.120	1	S
2387	161	850	2012-10-17 09:09:56.470	4	2012-10-17 09:09:52.583	175.138.173.120	14	S
2388	140	850	2012-10-17 09:10:19.990	5	2012-10-17 09:10:15.090	175.138.173.120	70	J

2410	145	906	2012-10-17 11:00:43.107	3	2012-10-17 11:00:38.097	120.139.128.139	1	S
2449	161	391	2012-10-17 22:31:23.357	1	2012-10-17 22:24:52.083	175.142.146.247	14	S
2450	165	391	2012-10-17 22:30:21.640	1	2012-10-17 22:25:17.770	175.142.146.247	70	C
2451	156	391	2012-10-17 22:29:51.410	4	2012-10-17 22:25:34.890	175.142.146.247	55	C
2453	139	391	2012-10-17 22:32:13.247	4	2012-10-17 22:32:09.590	175.142.146.247	95	S
2492	156	11	2012-10-18 18:10:40.203	4	2012-10-18 18:08:54.833	5.9.226.26	13	S
2493	155	11	2012-10-18 18:11:50.430	4	2012-10-18 18:11:33.377	5.9.226.26	13	S
2614	116	862	2012-10-22 22:21:54.430	5	2012-10-22 22:21:46.633	175.138.173.120	70	J
2615	129	862	2012-10-22 22:22:04.353	5	2012-10-22 22:22:00.423	175.138.173.120	109	S
2616	122	862	2012-10-22 22:22:11.943	5	2012-10-22 22:22:08.130	175.138.173.120	85	S
2617	124	862	2012-10-22 22:22:22.010	5	2012-10-22 22:22:18.450	175.138.173.120	26	S
2618	124	862	2012-10-22 22:22:29.307	5	2012-10-22 22:22:26.310	175.138.173.120	26	S
2619	122	862	2012-10-22 22:22:39.940	5	2012-10-22 22:22:34.973	175.138.173.120	85	S
2620	107	862	2012-10-22 22:22:46.797	5	2012-10-22 22:22:44.593	175.138.173.120	13	J
2656	165	1	2012-10-23 17:01:07.677	4	2012-10-23 17:00:38.790	202.185.114.251	70	C
2657	161	850	2012-10-23 17:05:31.917	4	2012-10-23 17:05:11.053	175.138.173.120	14	S
2658	156	850	2012-10-23 17:05:29.000	5	2012-10-23 17:05:23.880	175.138.173.120	13	S
2659	153	850	2012-10-23 17:05:50.000	4	2012-10-23 17:05:41.770	175.138.173.120	55	S
2661	151	850	2012-10-23 17:06:05.600	4	2012-10-23 17:06:03.083	175.138.173.120	55	J
2663	164	850	2012-10-23 17:06:21.953	5	2012-10-23 17:06:18.513	175.138.173.120	70	C
2665	165	850	2012-10-23 17:06:49.147	4	2012-10-23 17:06:43.417	175.138.173.120	13	N
2669	163	850	2012-10-23 17:09:39.250	5	2012-10-23 17:09:35.333	175.138.173.120	13	N
2688	152	323	2012-10-23 17:17:21.010	1	2012-10-23 17:17:12.513	202.185.108.203	55	C
2689	155	323	2012-10-23 17:17:46.990	2	2012-10-23 17:17:36.343	202.185.108.203	13	S
2770	139	1	2012-10-23 18:11:04.660	4	2012-10-23 18:10:59.183	202.185.114.251	95	S
2775	169	1	2012-10-23 18:20:32.917	4	2012-10-23 18:20:21.390	202.185.114.251	2	C
2778	102	850	2012-10-23 18:23:03.423	5	2012-10-23 18:23:00.467	175.138.173.120	48	S
2779	69	850	2012-10-23 18:23:23.957	5	2012-10-23 18:23:22.007	175.138.173.120	26	N
2908	147	865	2012-10-23 21:51:00.230	4	2012-10-23 21:50:52.867	175.138.173.120	66	C
2910	143	865	2012-10-23	5	2012-10-23	175.138.173.120	1	S

			21:51:23.633		21:51:19.777			
2912	150	865	2012-10-23 21:51:42.617	4	2012-10-23 21:51:33.907	175.138.173.120	26	S
2913	153	865	2012-10-23 21:51:48.613	5	2012-10-23 21:51:44.080	175.138.173.120	55	S
2920	164	1	2012-10-23 23:49:33.630	5	2012-10-23 23:49:25.237	124.13.84.175	95	J
2921	164	1	2012-10-24 00:02:22.387	4	2012-10-24 00:02:15.563	124.13.84.175	1	S
2924	168	850	2012-10-24 00:06:29.410	4	2012-10-24 00:06:20.853	175.138.173.120	14	J
2925	176	450	2012-10-24 00:06:35.503	3	2012-10-24 00:06:26.637	110.159.159.111	103	C
2926	172	850	2012-10-24 00:06:42.090	5	2012-10-24 00:06:38.987	175.138.173.120	13	J
2927	178	850	2012-10-24 00:06:59.793	4	2012-10-24 00:06:54.000	175.138.173.120	85	S
2928	174	850	2012-10-24 00:07:28.210	5	2012-10-24 00:07:25.307	175.138.173.120	13	S
2929	186	850	2012-10-24 00:07:54.350	5	2012-10-24 00:07:37.073	175.138.173.120	1	S
2930	184	850	2012-10-24 00:08:06.080	4	2012-10-24 00:08:03.200	175.138.173.120	2	S
2931	178	862	2012-10-24 00:08:47.800	5	2012-10-24 00:08:41.960	175.138.173.120	102	N
2932	189	862	2012-10-24 00:08:58.320	5	2012-10-24 00:08:54.797	175.138.173.120	95	N
2933	192	862	2012-10-24 00:09:07.927	5	2012-10-24 00:09:04.533	175.138.173.120	99	N
2934	172	862	2012-10-24 00:09:18.840	5	2012-10-24 00:09:13.007	175.138.173.120	102	C
2935	178	862	2012-10-24 00:09:28.973	5	2012-10-24 00:09:26.340	175.138.173.120	99	C
2936	192	862	2012-10-24 00:09:35.540	5	2012-10-24 00:09:33.047	175.138.173.120	48	C
2937	167	862	2012-10-24 00:09:44.803	5	2012-10-24 00:09:40.913	175.138.173.120	3	J
2938	187	987	2012-10-24 00:10:13.800	3	2012-10-24 00:10:08.487	221.133.40.230	2	S
2939	178	987	2012-10-24 00:10:31.303	5	2012-10-24 00:10:27.693	221.133.40.230	85	S
2940	187	987	2012-10-24 00:10:52.487	5	2012-10-24 00:10:47.837	221.133.40.230	2	S
2941	178	862	2012-10-24 00:10:51.763	5	2012-10-24 00:10:48.527	175.138.173.120	13	J
2942	172	987	2012-10-24 00:11:00.683	5	2012-10-24 00:10:56.913	221.133.40.230	2	S
2943	169	862	2012-10-24 00:11:04.477	5	2012-10-24 00:10:57.003	175.138.173.120	70	S
2944	188	862	2012-10-24 00:11:11.433	5	2012-10-24 00:11:08.840	175.138.173.120	11	S
2945	169	987	2012-10-24 00:11:13.497	3	2012-10-24 00:11:11.183	221.133.40.230	70	S
2946	164	987	2012-10-24 00:11:26.073	5	2012-10-24 00:11:23.013	221.133.40.230	1	S
2947	180	531	2012-10-24 00:12:26.080	5	2012-10-24 00:12:18.707	210.195.250.203	99	C
2952	191	850	2012-10-24 00:15:12.380	5	2012-10-24 00:15:08.960	175.138.173.120	99	N

2955	165	908	2012-10-24 00:20:29.313	3	2012-10-24 00:20:23.313	180.74.20.142	95	S
2956	179	908	2012-10-24 00:21:01.843	4	2012-10-24 00:20:56.780	180.74.20.142	1	S
2967	179	943	2012-10-24 01:13:12.603	2	2012-10-24 01:12:09.870	119.40.118.18	103	N
2969	197	943	2012-10-24 01:14:10.870	2	2012-10-24 01:14:02.400	119.40.118.18	14	N
2970	190	943	2012-10-24 01:14:25.607	2	2012-10-24 01:14:16.927	119.40.118.18	85	C
2971	156	943	2012-10-24 01:14:55.107	2	2012-10-24 01:14:32.750	119.40.118.18	70	J
2972	160	943	2012-10-24 01:15:04.203	2	2012-10-24 01:15:00.313	119.40.118.18	70	J
2975	175	943	2012-10-24 01:15:41.463	2	2012-10-24 01:15:28.643	119.40.118.18	55	S
2976	184	943	2012-10-24 01:15:51.727	2	2012-10-24 01:15:47.650	119.40.118.18	2	S
3118	190	11	2012-10-24 04:44:24.117	4	2012-10-24 04:32:41.703	94.183.221.30	95	N
3119	163	11	2012-10-24 04:43:32.623	4	2012-10-24 04:32:53.157	94.183.221.30	70	J
3120	174	11	2012-10-24 04:41:52.287	3	2012-10-24 04:33:00.747	94.183.221.30	13	S
3122	182	11	2012-10-24 04:33:57.687	3	2012-10-24 04:33:12.927	94.183.221.30	70	S
3154	168	342	2012-10-24 09:14:57.247	3	2012-10-24 09:14:54.493	202.184.111.77	14	J
3169	180	718	2012-10-24 10:13:26.760	3	2012-10-24 10:13:21.200	60.48.48.212	11	S
3172	175	718	2012-10-24 10:15:10.957	3	2012-10-24 10:15:08.167	60.48.48.212	55	S
3180	178	718	2012-10-24 10:29:45.833	3	2012-10-24 10:29:42.680	60.48.48.212	99	C
3181	177	718	2012-10-24 10:30:30.240	3	2012-10-24 10:30:27.960	60.48.48.212	2	S
3183	162	718	2012-10-24 10:30:37.090	3	2012-10-24 10:30:35.307	60.48.48.212	70	J
3189	195	251	2012-10-24 10:49:02.353	3	2012-10-24 10:48:47.777	210.195.99.15	14	N
3190	201	251	2012-10-24 10:49:16.323	4	2012-10-24 10:49:10.533	210.195.99.15	106	N
3191	176	251	2012-10-24 10:49:25.107	4	2012-10-24 10:49:21.663	210.195.99.15	103	C
3192	179	251	2012-10-24 10:49:32.200	3	2012-10-24 10:49:30.270	210.195.99.15	99	C
3193	193	251	2012-10-24 10:49:37.483	3	2012-10-24 10:49:36.253	210.195.99.15	48	C
3194	164	251	2012-10-24 10:49:47.487	4	2012-10-24 10:49:45.173	210.195.99.15	95	J
3195	168	251	2012-10-24 10:49:53.977	3	2012-10-24 10:49:52.377	210.195.99.15	14	J
3196	171	251	2012-10-24 10:50:01.320	3	2012-10-24 10:50:00.093	210.195.99.15	13	J
3197	174	251	2012-10-24 10:50:08.187	4	2012-10-24 10:50:04.887	210.195.99.15	13	J
3198	178	251	2012-10-24 10:50:14.243	4	2012-10-24 10:50:11.393	210.195.99.15	13	J
3225	187	391	2012-10-24	4	2012-10-24	161.142.24.130	85	C

			16:53:14.827		16:52:56.870			
3226	165	391	2012-10-24 16:53:25.073	2	2012-10-24 16:53:22.250	161.142.24.130	103	J
3227	175	391	2012-10-24 16:53:34.370	3	2012-10-24 16:53:31.003	161.142.24.130	13	J
3228	164	391	2012-10-24 16:53:49.010	4	2012-10-24 16:53:44.847	161.142.24.130	1	S
3229	173	391	2012-10-24 16:53:59.377	1	2012-10-24 16:53:57.077	161.142.24.130	1	S
3231	180	391	2012-10-24 16:54:18.067	4	2012-10-24 16:54:14.343	161.142.24.130	11	S
3233	179	772	2012-10-24 17:12:01.040	3	2012-10-24 17:10:45.880	115.133.210.244	1	S
3234	185	772	2012-10-24 17:13:37.850	3	2012-10-24 17:10:59.917	115.133.210.244	70	S
3656	182	732	2012-10-29 08:16:20.503	3	2012-10-29 08:16:12.943	210.48.147.106	70	S
3657	180	732	2012-10-29 08:17:10.803	3	2012-10-29 08:16:59.993	210.48.147.106	11	S
3785	209	1	2012-10-31 02:25:29.847	4	2012-10-31 02:25:21.683	124.13.84.175	13	C
3813	194	11	2012-10-31 03:11:53.570	5	2012-10-31 03:06:09.307	94.183.223.90	55	S
3814	213	11	2012-10-31 03:14:26.077	4	2012-10-31 03:12:08.670	94.183.223.90	1	S
3862	206	850	2012-10-31 08:58:28.433	5	2012-10-31 08:58:18.057	110.159.18.79	99	N
3863	220	850	2012-10-31 08:58:45.493	5	2012-10-31 08:58:37.920	110.159.18.79	26	N
3864	181	850	2012-10-31 08:58:59.447	4	2012-10-31 08:58:55.453	110.159.18.79	13	J
3865	203	850	2012-10-31 08:59:08.090	4	2012-10-31 08:59:02.077	110.159.18.79	13	J
3866	202	850	2012-10-31 08:59:35.693	5	2012-10-31 08:59:29.817	110.159.18.79	2	S
3867	203	850	2012-10-31 08:59:51.877	4	2012-10-31 08:59:47.503	110.159.18.79	55	S
3872	194	391	2012-10-31 09:19:07.450	4	2012-10-31 09:17:24.800	161.142.24.130	55	S
3875	211	391	2012-10-31 09:18:50.413	2	2012-10-31 09:17:44.580	161.142.24.130	15	S
3876	215	391	2012-10-31 09:18:43.000	1	2012-10-31 09:17:48.363	161.142.24.130	85	N
3877	222	391	2012-10-31 09:18:35.780	1	2012-10-31 09:17:50.433	161.142.24.130	70	N
3879	185	391	2012-10-31 09:18:20.020	1	2012-10-31 09:17:55.093	161.142.24.130	109	J
3880	195	391	2012-10-31 09:19:02.167	4	2012-10-31 09:17:57.643	161.142.24.130	85	J
3881	199	391	2012-10-31 09:18:55.993	4	2012-10-31 09:18:00.890	161.142.24.130	2	J
3921	218	928	2012-10-31 11:13:11.127	3	2012-10-31 11:10:18.280	115.133.214.185	106	N
3922	182	928	2012-10-31 11:12:54.190	5	2012-10-31 11:10:24.737	115.133.214.185	13	J
3923	185	928	2012-10-31 11:12:47.787	3	2012-10-31 11:10:33.890	115.133.214.185	109	J
3924	196	928	2012-10-31 11:11:19.177	2	2012-10-31 11:10:47.853	115.133.214.185	85	J

3925	197	928	2012-10-31 11:11:04.043	5	2012-10-31 11:10:52.423	115.133.214.185	102	J
3932	213	550	2012-10-31 11:35:54.433	3	2012-10-31 11:35:50.090	210.48.147.109	85	C
3933	182	550	2012-10-31 11:36:34.397	3	2012-10-31 11:36:31.370	210.48.147.109	13	J
3934	189	550	2012-10-31 11:36:53.020	3	2012-10-31 11:36:47.870	210.48.147.109	14	J
3935	193	550	2012-10-31 11:37:11.613	3	2012-10-31 11:37:08.097	210.48.147.109	55	S
3936	209	943	2012-10-31 11:47:46.467	3	2012-10-31 11:47:30.173	119.40.118.18	13	N
3937	212	943	2012-10-31 11:48:27.990	3	2012-10-31 11:48:07.277	119.40.118.18	13	N
3938	215	943	2012-10-31 11:48:49.740	2	2012-10-31 11:48:37.803	119.40.118.18	85	C
3939	183	943	2012-10-31 11:49:13.390	4	2012-10-31 11:49:01.813	119.40.118.18	13	J
3940	195	943	2012-10-31 11:49:36.300	4	2012-10-31 11:49:21.063	119.40.118.18	85	J
3942	197	943	2012-10-31 11:49:51.083	4	2012-10-31 11:49:45.717	119.40.118.18	102	J
3943	200	943	2012-10-31 11:50:20.860	4	2012-10-31 11:49:57.307	119.40.118.18	13	J
3944	191	943	2012-10-31 11:50:37.663	4	2012-10-31 11:50:26.310	119.40.118.18	1	S
3945	199	943	2012-10-31 11:51:18.273	5	2012-10-31 11:50:46.383	119.40.118.18	1	S
3946	202	943	2012-10-31 11:51:49.270	4	2012-10-31 11:51:28.760	119.40.118.18	2	S
3947	208	862	2012-10-31 12:01:03.240	5	2012-10-31 12:00:57.397	110.159.18.79	85	N
3948	213	862	2012-10-31 12:01:14.070	5	2012-10-31 12:01:11.300	110.159.18.79	13	N
3949	226	862	2012-10-31 12:01:20.970	5	2012-10-31 12:01:18.543	110.159.18.79	70	N
3950	211	862	2012-10-31 12:01:27.670	5	2012-10-31 12:01:24.557	110.159.18.79	13	C
3951	179	862	2012-10-31 12:01:35.433	5	2012-10-31 12:01:32.640	110.159.18.79	2	J
3952	196	862	2012-10-31 12:01:42.900	5	2012-10-31 12:01:39.137	110.159.18.79	85	J
3953	197	862	2012-10-31 12:01:50.277	5	2012-10-31 12:01:47.957	110.159.18.79	102	J
3954	201	862	2012-10-31 12:01:58.667	5	2012-10-31 12:01:56.650	110.159.18.79	85	J
3955	197	862	2012-10-31 12:02:04.923	5	2012-10-31 12:02:02.200	110.159.18.79	97	S
3956	208	862	2012-10-31 12:02:11.970	5	2012-10-31 12:02:08.967	110.159.18.79	10	S
3958	208	862	2012-10-31 12:12:34.027	5	2012-10-31 12:12:30.027	110.159.18.79	10	S
3959	197	862	2012-10-31 12:12:44.607	5	2012-10-31 12:12:40.960	110.159.18.79	97	S
3960	201	862	2012-10-31 12:12:50.407	5	2012-10-31 12:12:47.793	110.159.18.79	85	J
3961	201	862	2012-10-31 12:12:56.353	5	2012-10-31 12:12:54.013	110.159.18.79	85	J
3962	201	862	2012-10-31	5	2012-10-31	110.159.18.79	85	J

			12:13:02.600		12:13:00.443			
4039	226	323	2012-11-01 12:53:08.547	5	2012-11-01 12:51:25.573	202.185.108.203	70	N
4040	202	323	2012-11-01 12:52:08.447	2	2012-11-01 12:51:40.730	202.185.108.203	2	S
4041	182	323	2012-11-01 12:53:38.547	1	2012-11-01 12:53:31.877	202.185.108.203	13	J
4184	197	918	2012-11-03 15:22:05.827	5	2012-11-03 15:22:01.270	210.195.238.170	97	S
4699	209	850	2012-11-12 01:40:36.920	4	2012-11-12 01:40:29.990	60.54.40.239	13	J
4700	222	850	2012-11-12 01:40:52.380	4	2012-11-12 01:40:47.723	60.54.40.239	13	C
4702	228	850	2012-11-12 01:41:23.170	2	2012-11-12 01:41:12.537	60.54.40.239	13	J
4704	224	850	2012-11-12 01:41:43.703	4	2012-11-12 01:41:34.803	60.54.40.239	13	S
4706	239	850	2012-11-12 01:42:06.837	5	2012-11-12 01:41:51.033	60.54.40.239	13	S
4724	220	973	2012-11-12 01:52:26.700	3	2012-11-12 01:52:08.263	175.142.219.205	70	J
4725	218	973	2012-11-12 01:53:11.923	5	2012-11-12 01:52:15.083	175.142.219.205	1	S
4737	239	11	2012-11-12 02:03:37.553	3	2012-11-12 02:02:37.967	94.183.222.58	13	S
4738	224	11	2012-11-12 07:06:07.070	3	2012-11-12 02:03:47.770	94.183.222.58	13	S
4739	222	718	2012-11-12 02:09:43.820	3	2012-11-12 02:09:35.707	118.101.201.182	13	C
4740	229	718	2012-11-12 02:09:52.987	3	2012-11-12 02:09:50.850	118.101.201.182	13	C
4741	234	718	2012-11-12 02:10:03.087	3	2012-11-12 02:10:00.943	118.101.201.182	13	C
4742	235	718	2012-11-12 02:10:10.297	3	2012-11-12 02:10:07.407	118.101.201.182	13	C
4743	209	718	2012-11-12 02:10:17.727	3	2012-11-12 02:10:15.863	118.101.201.182	13	J
4744	216	718	2012-11-12 02:10:23.647	3	2012-11-12 02:10:21.660	118.101.201.182	13	J
4745	228	718	2012-11-12 02:10:35.413	3	2012-11-12 02:10:28.753	118.101.201.182	13	J
4746	218	718	2012-11-12 02:10:59.573	3	2012-11-12 02:10:57.187	118.101.201.182	1	S
4944	206	550	2012-11-12 09:33:18.703	3	2012-11-12 09:33:15.907	210.48.147.109	14	J
4945	226	550	2012-11-12 09:33:30.133	3	2012-11-12 09:33:27.427	210.48.147.109	70	S
4963	228	710	2012-11-12 10:04:59.930	3	2012-11-12 10:04:48.460	210.186.223.161	1	S
4986	205	323	2012-11-12 10:50:01.590	2	2012-11-12 10:49:50.217	202.185.108.201	70	J
4987	220	323	2012-11-12 10:50:24.680	1	2012-11-12 10:50:09.973	202.185.108.201	70	J
5015	229	328	2012-11-12 11:51:26.853	4	2012-11-12 11:51:10.490	175.136.17.179	85	N
5027	214	391	2012-11-12 12:17:32.000	4	2012-11-12 12:16:59.960	161.142.24.130	1	S
5028	215	391	2012-11-12 12:17:41.630	1	2012-11-12 12:17:02.730	161.142.24.130	1	S

5029	218	391	2012-11-12 12:17:49.440	4	2012-11-12 12:17:05.187	161.142.24.130	1	S
5030	219	391	2012-11-12 12:17:58.650	1	2012-11-12 12:17:07.287	161.142.24.130	1	S
5031	223	391	2012-11-12 12:18:05.727	4	2012-11-12 12:17:09.633	161.142.24.130	1	S
5032	227	391	2012-11-12 12:18:55.080	3	2012-11-12 12:17:12.010	161.142.24.130	1	S
5033	228	391	2012-11-12 12:19:02.783	1	2012-11-12 12:17:15.237	161.142.24.130	1	S
5034	232	391	2012-11-12 12:19:10.530	4	2012-11-12 12:17:17.803	161.142.24.130	1	S
5035	233	391	2012-11-12 12:19:17.150	4	2012-11-12 12:17:19.800	161.142.24.130	1	S
5036	234	391	2012-11-12 12:19:23.190	3	2012-11-12 12:17:21.890	161.142.24.130	1	S
5049	222	862	2012-11-12 12:51:13.320	5	2012-11-12 12:50:02.930	120.141.160.73	13	C
5050	220	918	2012-11-12 12:51:08.693	3	2012-11-12 12:50:54.823	210.195.238.170	70	J
5051	229	862	2012-11-12 12:51:28.927	5	2012-11-12 12:51:25.010	120.141.160.73	13	C
5052	234	862	2012-11-12 12:51:36.313	5	2012-11-12 12:51:33.220	120.141.160.73	13	C
5053	235	862	2012-11-12 12:51:43.940	5	2012-11-12 12:51:40.983	120.141.160.73	13	C
5054	204	862	2012-11-12 12:51:51.220	5	2012-11-12 12:51:47.907	120.141.160.73	2	J
5055	207	862	2012-11-12 12:52:00.333	5	2012-11-12 12:51:56.930	120.141.160.73	2	J
5056	209	862	2012-11-12 12:52:07.587	5	2012-11-12 12:52:05.033	120.141.160.73	13	J
5057	216	862	2012-11-12 12:52:14.703	5	2012-11-12 12:52:12.307	120.141.160.73	13	J
5058	228	862	2012-11-12 12:52:22.670	5	2012-11-12 12:52:19.937	120.141.160.73	13	J
5059	231	862	2012-11-12 12:52:28.507	5	2012-11-12 12:52:25.987	120.141.160.73	10	S
5069	231	421	2012-11-12 14:02:55.203	3	2012-11-12 14:02:30.137	119.40.120.198	99	N
5075	228	211	2012-11-12 14:35:11.020	3	2012-11-12 14:35:08.430	202.185.85.154	99	C
5090	243	850	2012-11-12 16:32:23.743	4	2012-11-12 16:32:17.800	110.159.16.163	85	C
5091	228	850	2012-11-12 16:33:06.207	5	2012-11-12 16:33:03.317	110.159.16.163	99	C
5254	222	927	2012-11-14 00:48:11.770	3	2012-11-14 00:48:05.757	210.186.217.53	99	S
5255	225	927	2012-11-14 00:48:49.700	3	2012-11-14 00:48:47.280	210.186.217.53	59	C
5256	238	927	2012-11-14 00:49:32.530	3	2012-11-14 00:49:10.207	210.186.217.53	70	N
5465	216	427	2012-11-15 15:09:42.170	3	2012-11-15 14:43:59.167	37.255.104.207	16	S
5493	219	828	2012-11-16 11:47:22.270	4	2012-11-16 11:44:43.357	120.141.214.250	95	J
5498	232	828	2012-11-16 12:00:33.670	4	2012-11-16 11:46:25.917	120.141.214.250	1	S
5633	240	391	2012-11-19	1	2012-11-19	175.136.62.14	1	S

			00:49:53.113		00:49:00.933			
5634	241	391	2012-11-19 00:49:59.140	5	2012-11-19 00:49:03.117	175.136.62.14	1	S
5636	242	391	2012-11-19 00:50:05.633	1	2012-11-19 00:49:08.093	175.136.62.14	1	S
5637	244	391	2012-11-19 00:50:19.803	1	2012-11-19 00:49:13.013	175.136.62.14	1	S
5640	246	391	2012-11-19 00:50:55.980	1	2012-11-19 00:49:18.553	175.136.62.14	1	S
5643	250	391	2012-11-19 00:51:18.420	2	2012-11-19 00:49:25.183	175.136.62.14	1	S
5645	251	391	2012-11-19 00:51:45.113	2	2012-11-19 00:49:30.197	175.136.62.14	1	S
5647	255	391	2012-11-19 00:51:52.640	5	2012-11-19 00:49:37.170	175.136.62.14	1	S
5649	258	391	2012-11-19 00:52:14.093	2	2012-11-19 00:49:44.107	175.136.62.14	1	S
5651	259	391	2012-11-19 00:52:18.920	1	2012-11-19 00:49:47.477	175.136.62.14	1	S
5842	270	970	2012-11-19 02:40:18.987	3	2012-11-19 02:39:33.023	113.210.229.23	85	C
5844	263	970	2012-11-19 02:40:42.977	3	2012-11-19 02:40:30.893	113.210.229.23	11	C
5846	260	970	2012-11-19 02:40:54.167	3	2012-11-19 02:40:49.873	113.210.229.23	8	C
5847	258	970	2012-11-19 02:41:04.113	3	2012-11-19 02:40:58.950	113.210.229.23	8	C
5849	269	970	2012-11-19 02:41:38.913	3	2012-11-19 02:41:19.230	113.210.229.23	85	C
5850	247	970	2012-11-19 02:41:52.277	3	2012-11-19 02:41:45.633	113.210.229.23	98	C
5882	241	850	2012-11-19 09:50:42.000	4	2012-11-19 09:50:05.967	60.54.40.239	13	J
5883	242	850	2012-11-19 09:50:38.533	5	2012-11-19 09:50:35.767	60.54.40.239	14	J
5884	246	850	2012-11-19 09:51:26.827	5	2012-11-19 09:51:22.800	60.54.40.239	1	S
5885	255	850	2012-11-19 09:51:34.320	4	2012-11-19 09:51:32.137	60.54.40.239	1	S
5886	236	550	2012-11-19 09:52:21.043	3	2012-11-19 09:51:37.850	210.48.147.109	14	J
5887	242	550	2012-11-19 09:52:28.360	3	2012-11-19 09:52:25.247	210.48.147.109	14	J
5888	251	550	2012-11-19 09:52:34.960	3	2012-11-19 09:52:30.623	210.48.147.109	14	J
5924	241	718	2012-11-19 12:09:15.593	3	2012-11-19 12:08:25.550	118.101.201.173	13	J
5925	241	718	2012-11-19 12:09:37.477	3	2012-11-19 12:09:34.597	118.101.201.173	1	S
5926	248	718	2012-11-19 12:11:58.107	3	2012-11-19 12:11:49.557	118.101.201.173	48	S
5927	262	718	2012-11-19 12:12:22.620	3	2012-11-19 12:12:19.593	118.101.201.173	1	S
5929	259	718	2012-11-19 12:13:14.027	3	2012-11-19 12:13:05.127	118.101.201.173	1	S
5932	256	718	2012-11-19 12:13:55.393	3	2012-11-19 12:13:40.490	118.101.201.173	11	S
5933	255	718	2012-11-19 12:14:15.777	3	2012-11-19 12:14:07.383	118.101.201.173	1	S

5934	254	718	2012-11-19 12:14:43.643	3	2012-11-19 12:14:27.390	118.101.201.173	55	S
5937	253	718	2012-11-19 12:15:24.027	3	2012-11-19 12:15:18.920	118.101.201.173	102	S
5938	249	718	2012-11-19 12:15:34.890	3	2012-11-19 12:15:32.700	118.101.201.173	85	S
5939	244	718	2012-11-19 12:15:46.297	3	2012-11-19 12:15:43.783	118.101.201.173	1	S
5940	241	718	2012-11-19 12:16:07.947	3	2012-11-19 12:16:05.093	118.101.201.173	1	S

Appendix B

List of MyExpert Stored Procedures

No.	Name of Stored Procedure
1	sp_alterdiagram
2	sp_creatediagram
3	sp_dropdiagram
4	sp_helpdiagramdefinition
5	sp_helpdiagrams
6	sp_renamediagram
7	sp_upgraddiagrams
8	spAriaGetEmails
9	spCityGet
10	spCityToState
11	spConfDelete
12	spConfGetByID
13	spConfGetByOwnerID
14	spConfGetFromTo
15	spConfGetPending
16	spConfGetPendingTop50
17	spConfGetPicname
18	spConfGetTop10
19	spConfGetTop25
20	spConfGetTop3
21	spConfInsert
22	spConfUpdateCounter
23	spConfUpdateInformation
24	spConfUpdatePicName
25	spConfUpdateStatusAccepted
26	spCountriesGetAll
27	spDegreeGet
28	spFacultyGet
29	spFacultyGetInfo
30	spFacultyToUniversity
31	spFacultyUpdateMemberCount
32	spFriendAccept
33	spFriendAcceptAndAdd
34	spFriendDeny
35	spFriendExistanceCheck
36	spFriendGetAllByInvitee
37	spFriendGetAllByInviter
38	spFriendGetAllPendingByInvitee
39	spFriendGetByID
40	spFriendGetNewCount
41	spFriendInsert
42	spFriendMyTop5
43	spFriendsCount
44	spJobDelete
45	spJobGetByID
46	spJobGetByOwnerID

47	spJobGetFromTo
48	spJobGetPicname
49	spJobGetTop10
50	spJobGetTop25
51	spJobGetTop3
52	spJobInsert
53	spJobInsertNoDeadline
54	spJobsGetPending
55	spJobsGetPendingTop50
56	spJobsUpdateCounter
57	spJobsUpdateStatusAccepted
58	spJobUpdateInformation
59	spJobUpdatePicName
60	spLastNewsUpdatesGet
61	spLastVisitorsGet
62	spLastVisitorsInsert
63	spLogin
64	spLoginFirst
65	spLogsInsert
66	spMessageGetByReciever_All
67	spMessageGetNewCount
68	spMessageInsert
69	spMessageUpdateStatus
70	spMessageUpdateStatusToNotNew
71	spNewsAlgorithmCalculateAvgRatebyUserID
72	spNewsAlgorithmCBDeleteAllPrediction10
73	spNewsAlgorithmCBGetAllItemByTreeNode
74	spNewsAlgorithmCBGetRandomRemainedItems
75	spNewsAlgorithmCBGetUserAllScoredNodes
76	spNewsAlgorithmCBMapRankToNode
77	spNewsAlgorithmCBNodeUpdateScoreDep
78	spNewsAlgorithmCBNodeUpdateScoreFaculty
79	spNewsAlgorithmCBNodeUpdateScoreSelfClick
80	spNewsAlgorithmCBNodeUpdateScoreSelfRank
81	spNewsAlgorithmCBNodeUpdateScoreTotal
82	spNewsAlgorithmCBPrediction10Insert
83	spNewsAlgorithmCBResetScores
84	spNewsAlgorithmCBTreeScoreMaking
85	spNewsAlgorithmCBUpdateNewsNodeID
86	spNewsAlgorithmClbGetAllNewsSimilarities
87	spNewsAlgorithmClbGetPredictionListByPredValue
88	spNewsAlgorithmClbGetPredValuesbyUserID
89	spNewsAlgorithmClbPredictionInsert
90	spNewsAlgorithmClbPredictionSmallInsert
91	spNewsAlgorithmClbSimilarityCalcDownPower
92	spNewsAlgorithmClbSimilarityCalcUp
93	spNewsAlgorithmClbSimilarityItemAB
94	spNewsAlgorithmClbSimilarityItemABInsert

95	spNewsAlgorithmClbSimilarityUpdateValue
96	spNewsAlgorithmEvCalculation
97	spNewsAlgorithmEvCalculation2
98	spNewsAlgorithmEvInsert
99	spNewsAlgorithmGetAllNew
100	spNewsAlgorithmGetAllNewsInDuration
101	spNewsAlgorithmGetAllNewsTotal
102	spNewsAlgorithmGetAllPrediction
103	spNewsAlgorithmGetCBbByUserID
104	spNewsAlgorithmGetClbByUserID
105	spNewsAlgorithmGetNewsPrediction_10_23
106	spNewsAlgorithmGetNewsPrediction_10_31
107	spNewsAlgorithmGetNewsPrediction_11_12
108	spNewsAlgorithmGetNewsPrediction_11_19
109	spNewsAlgorithmGetNewsPrediction_11_25
110	spNewsAlgorithmGetNewsPrediction_12_11
111	spNewsAlgorithmGetNewsPrediction_12_20
112	spNewsAlgorithmGetNewsPrediction_12_4
113	spNewsAlgorithmGetNewsRank_Nrn_Count
114	spNewsAlgorithmGetNewsRank_Nrs_Count
115	spNewsAlgorithmGetNewsRankInDuration
116	spNewsAlgorithmGetNewsRankInDuration_10_23
117	spNewsAlgorithmGetNewsRankInDuration_10_31
118	spNewsAlgorithmGetNewsRankInDuration_11_12
119	spNewsAlgorithmGetNewsRankInDuration_11_19
120	spNewsAlgorithmGetNewsRankInDuration_11_25
121	spNewsAlgorithmGetNewsRankInDuration_12_11
122	spNewsAlgorithmGetNewsRankInDuration_12_20
123	spNewsAlgorithmGetNewsRankInDuration_12_4
124	spNewsAlgorithmGetRandom
125	spNewsAlgorithmGetRatedItemsAllByUser
126	spNewsAlgorithmGetRatedItemsByUser
127	spNewsAlgorithmGetSimilarityItemAB
128	spNewsAlgorithmInsertCollaborativeRnd10
129	spNewsAlgorithmInsertNewToNewsTotal
130	spNewsAlgorithmNewGetUserFacultyTopScoredNodes
131	spNewsAlgorithmNewGetUserFriendsTopScoredNodes
132	spNewsAlgorithmNewNodeUpdateScoreTotal
133	spNewsAlgorithmNewsRankScienceIDUpdate
134	spNewsAlgorithmUpdatePredictionValue
135	spNewsDelete
136	spNewsGetByID
137	spNewsGetByOwnerID
138	spNewsGetFromTo
139	spNewsGetPending
140	spNewsGetPendingTop50
141	spNewsGetPicname
142	spNewsGetTop10

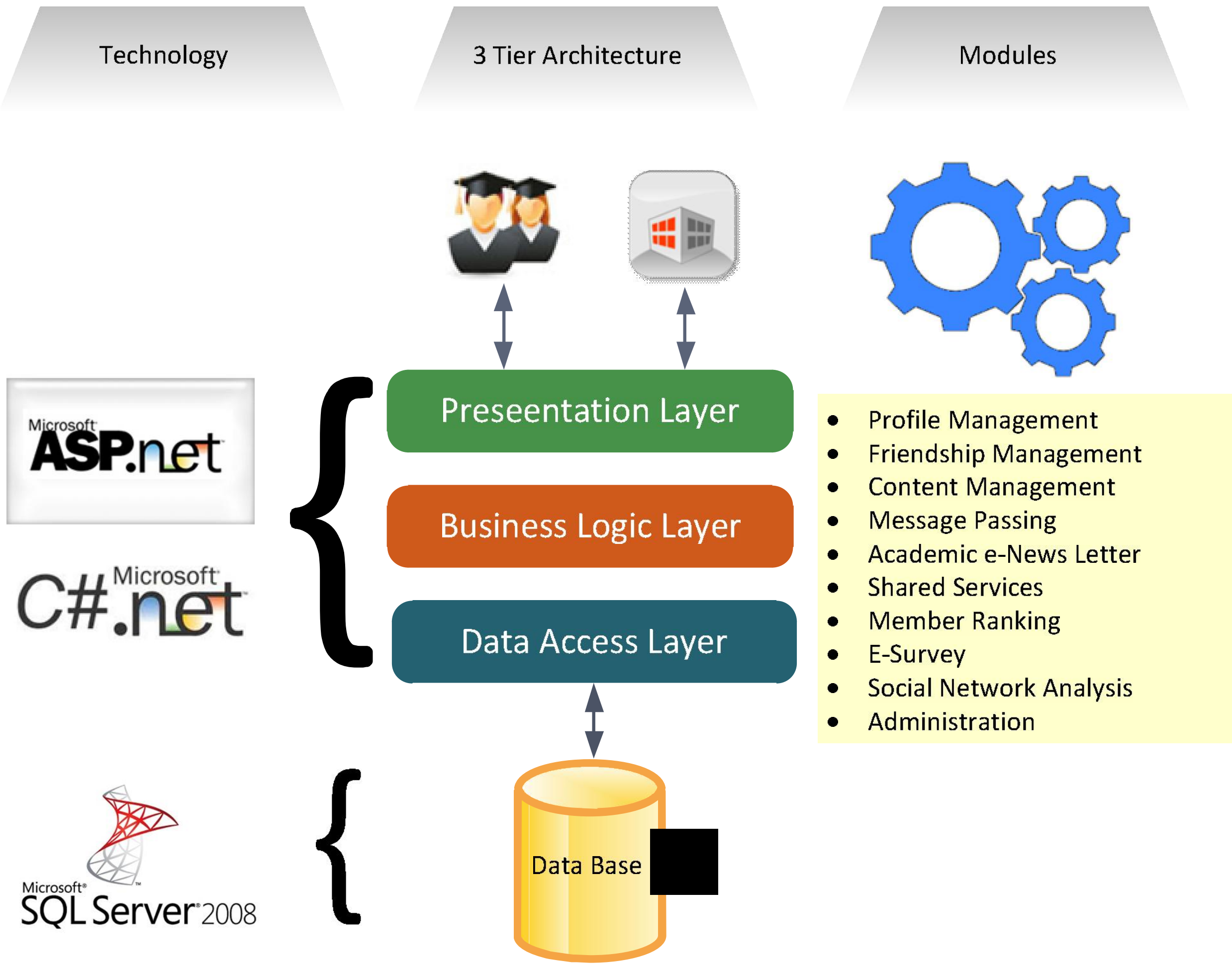
143	spNewsGetTop25
144	spNewsGetTop3
145	spNewsInsert
146	spNewsOldDelete
147	spNewsOldGetByNewsID
148	spNewsOldGetPicname
149	spNewsOldGetRecordsinDuration
150	spNewsOldGetTop20
151	spNewsOldGetTop5
152	spNewsOldInsert
153	spNewsOldUpdateInformation
154	spNewsOldUpdatePicName
155	spNewsRanksGetAllbyUser
156	spNewsRanksGetLastbyUser
157	spNewsRanksInsert
158	spNewsRankUpdate
159	spNewsTopGetRandom
160	spNewsUpdateCounter
161	spNewsUpdateInformation
162	spNewsUpdatePicName
163	spNewsUpdateStatusAccepted
164	spPointsGetBasicValue
165	spPointsInsertNew
166	spPointsUpdate1
167	spPointsUpdate2
168	spPointsUpdate3
169	spPointsUpdate4
170	spPointsUpdate5
171	spProfileExistanceCheck
172	spProfileExistanceCheckByEmail
173	spProfileGet5NewMember
174	spProfileGetAll00Flag
175	spProfileGetAllEmails
176	spProfileGetAllMemberByFaculty
177	spProfileGetAllMemberByUniversity
178	spProfileGetInfo
179	spProfileGetLastUpdates
180	spProfileGetNewMemberByUniversity
181	spProfileGetNewsByMember_10
182	spProfileGetNewsByMember_All
183	spProfileGetPicname
184	spProfileGetShortInfo
185	spProfileGetShortInfoByEmail
186	spProfileGetStatisticsInfo
187	spProfileGetTop5Member
188	spProfileGetTop6MemberInFaculty
189	spProfileGetTop6MemberInUni
190	spProfileGetTopMemberByUniversity

191	spProfileInsertLastUpdates
192	spProfileInsertNews
193	spProfilePreSignUp
194	spProfileUpdateInformation
195	spProfileUpdatePassword
196	spProfileUpdatePicName
197	spProfileUpdateStatus
198	spSchDelete
199	spSchGetByID
200	spSchGetByOwnerID
201	spSchGetFromTo
202	spSchGetPending
203	spSchGetPendingTop50
204	spSchGetPicname
205	spSchGetTop10
206	spSchGetTop25
207	spSchGetTop3
208	spSchInsert
209	spSchUpdateCounter
210	spSchUpdateInformation
211	spSchUpdatePicName
212	spSchUpdateStatusAccepted
213	spSciencesGetAll
214	spSciencesGetByParent
215	spStateGet
216	spTempNewsInDurationInsert
217	spTree_insert_tree_node
218	spTree_view_human_tree
219	spTree_view_tree
220	spUniversityGet
221	spUniversityGetFaculties
222	spUniversityGetInfo
223	spUniversityGetOrderByMembers
224	spUniversityGetTop4
225	spUniversityInsert
226	spUniversityUpdateMemberCount

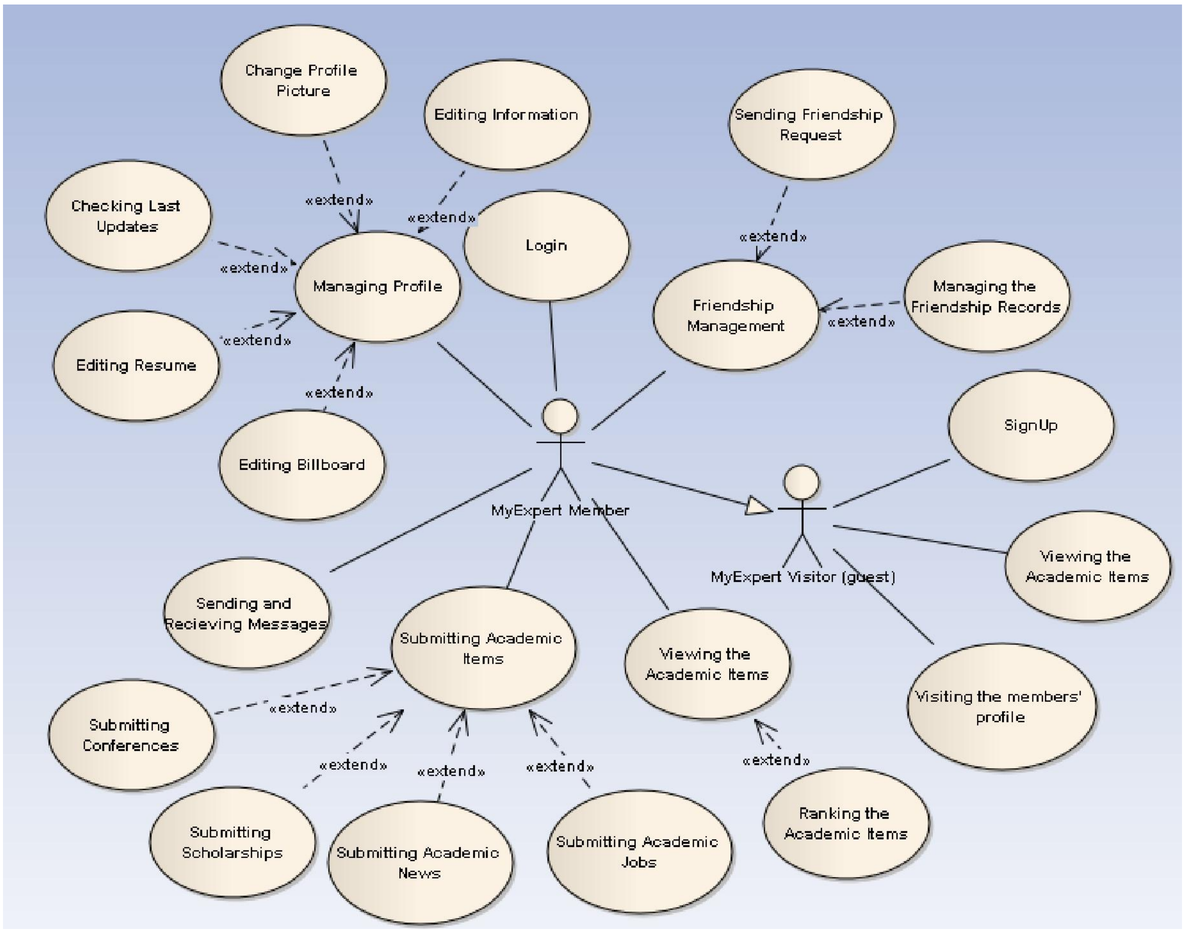
Appendix C

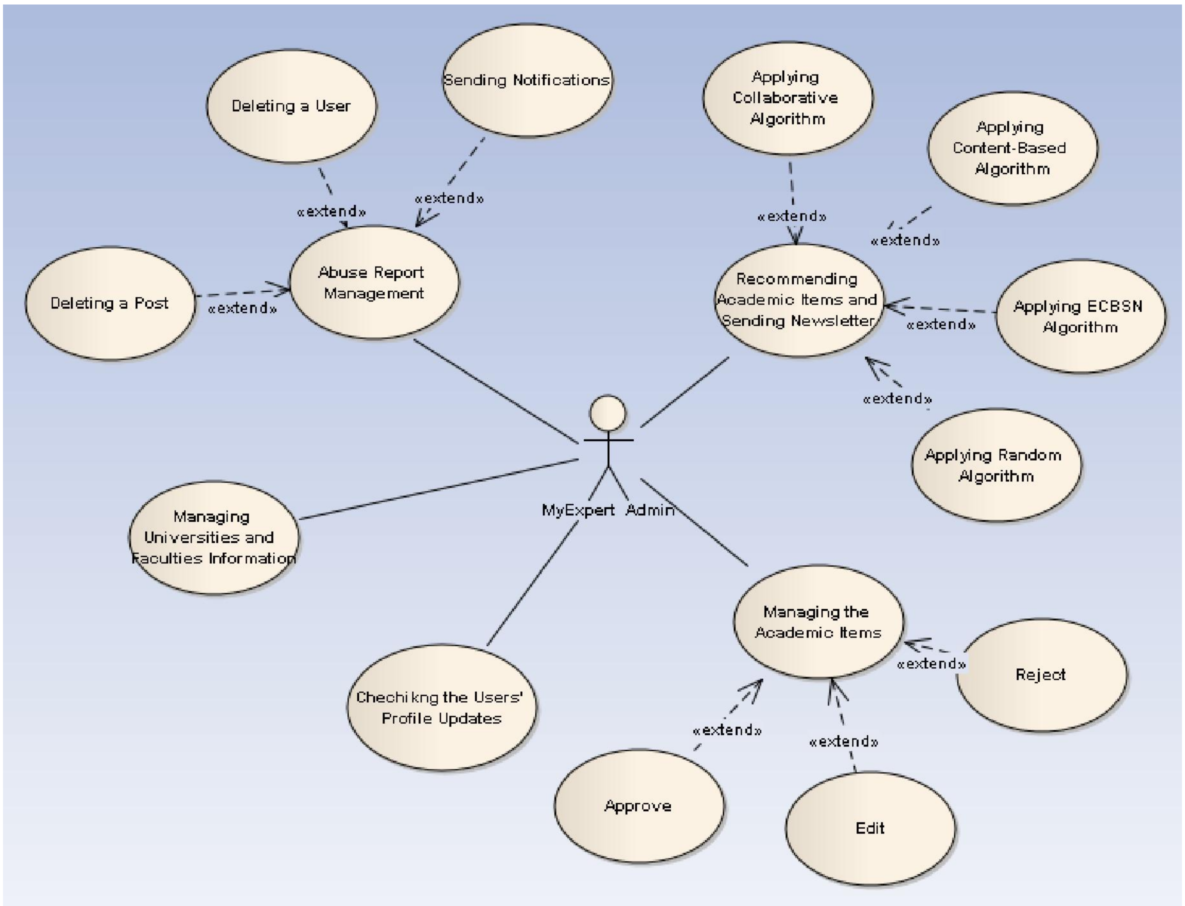
MyExpert Technical Documents

MyExpert 3-tier Programming Architecture

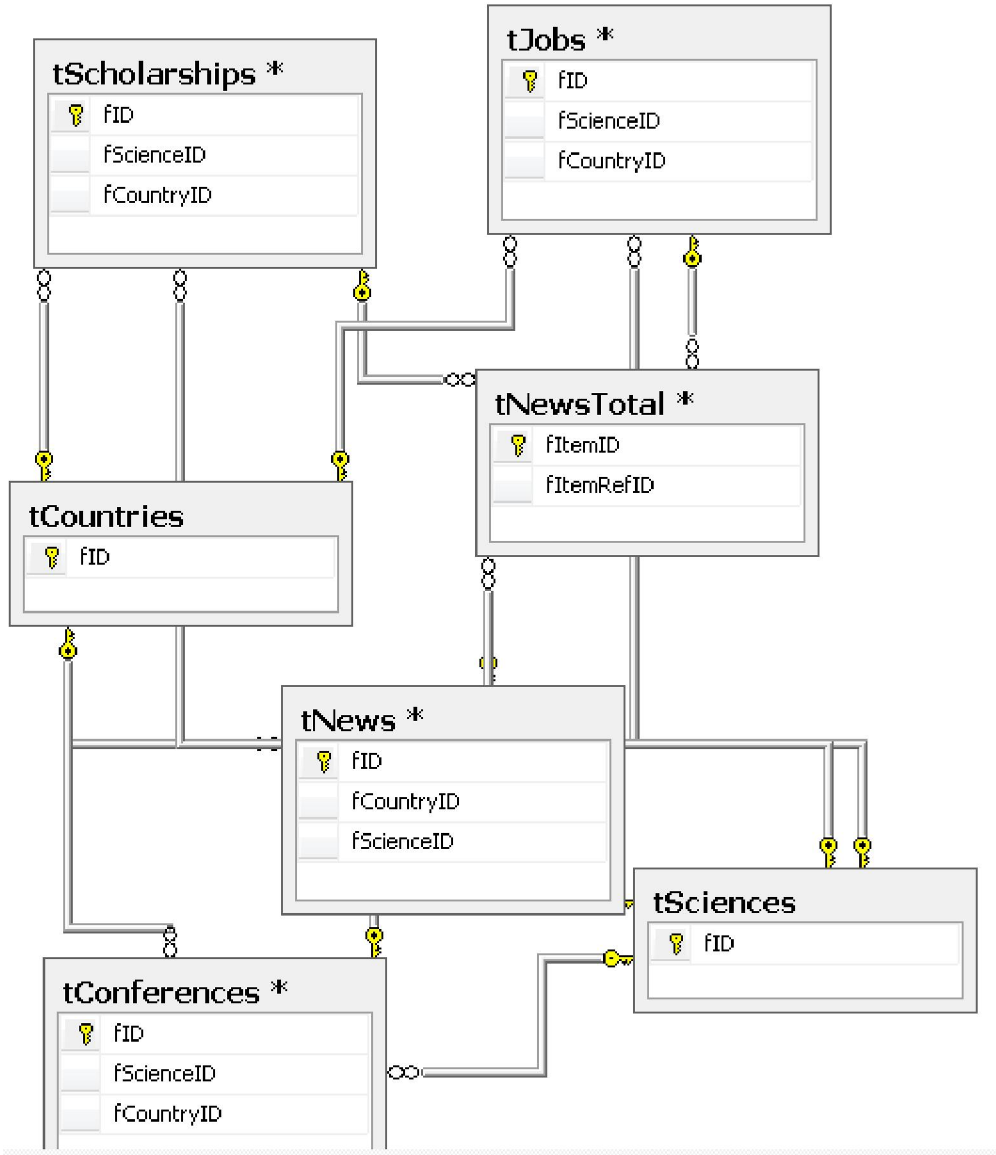


MyExpert Data Flow Diagram




Deleting a User
Sending Notifications
«extend»
«extend»
Deleting a Post
«extend»
Abuse Report Management
Applying Collaborative Algorithm
Applying Content-Based Algorithm
«extend»
«extend»
Recommending Academic Items and Sending Newsletter
Applying ECBSN Algorithm
«extend»
«extend»
Applying Random Algorithm
MyExpert Admin
Managing Universities and Faculties Information
Chechikng the Users' Profile Updates
Managing the Academic Items
«extend»
Reject
«extend»
«extend»
Approve
Edit

MyExpert Entity Relationship Diagram (ERD)



MyExpert User Interface

(Homepage)

MyExpert User Interface

(News Details Webpage)


MyExperts ©
Malaysian Experts
Academic Social Network

Email
Password
Login

Rank me

Category : Scholarship / ----- Natural Sciences -----

2013 Sacred Heart Scholarships for UK and International Students at University of Roehampton in UK (United Kingdom)

University : **Multimedia University**

Faculty : **Creative Multimedia**

Submit Date : 19/12/2012

Related URL:

http://scholarship-positions.com/2013-sa

By: Ghazaleh Babanejad

Employer: University of Roehampton and the Society of the Sacred Heart is offering this scholarship

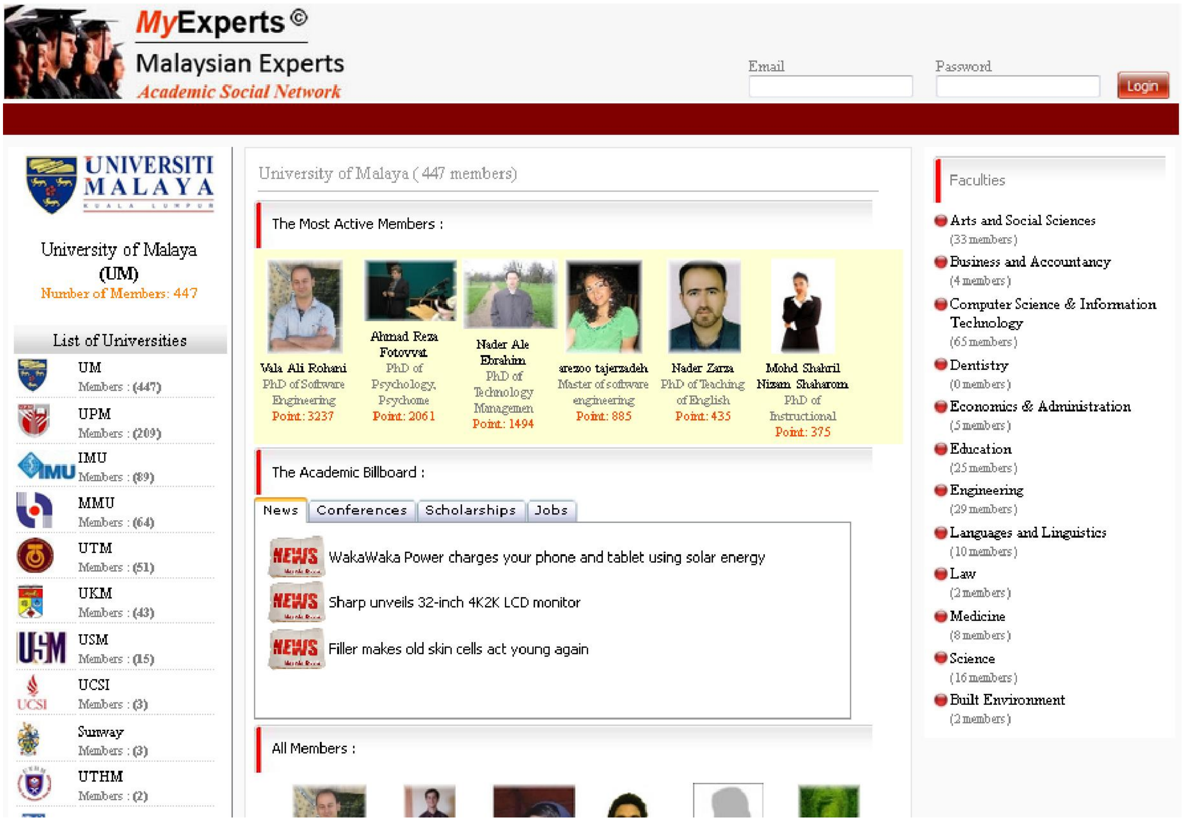
he University of Roehampton and the Society of the Sacred Heart have established a scholarship fund to encourage postgraduate studies and research in areas which promote international understanding and which enable candidates to work more effectively in their communities upon completion of their studies.The scholarships are available to academically excellent students from both within and outside the UK to follow a programme of study at the University of Roehampton. Candidates in any discipline are encouraged to apply.Applications are particularly welcome from candidates whose proposed area of study or research is broadly

Other Scholarships

- 2013 Sacred Heart Scholarships for UK and International Students at University of Roehampton in UK
- 2013 University of Bristol Home/EU and Overseas Scholarships in UK
- 2013 International Research Scholarships at University of Sussex in UK
- 2013 School Postgraduate Scholarship in Mathematics & Physical Sciences at University of Sussex in UK
- 2013 Kent Law School Taught Masters Overseas Scholarships at University of Kent in UK
- 2013/2014 Fellowships for Berlin-Based Postdoctoral Program Legal Cultures in Germany
- King's University of London Scholarship Scheme in UK, 2013/14
- 2013 PhD Scholarships at Nanyang Technological University in Singapore
- 2013 English Master's Degree Part-Fee Bursaries at University of Leicester in UK
- 2013 School of Psychology PhD Studentship at University

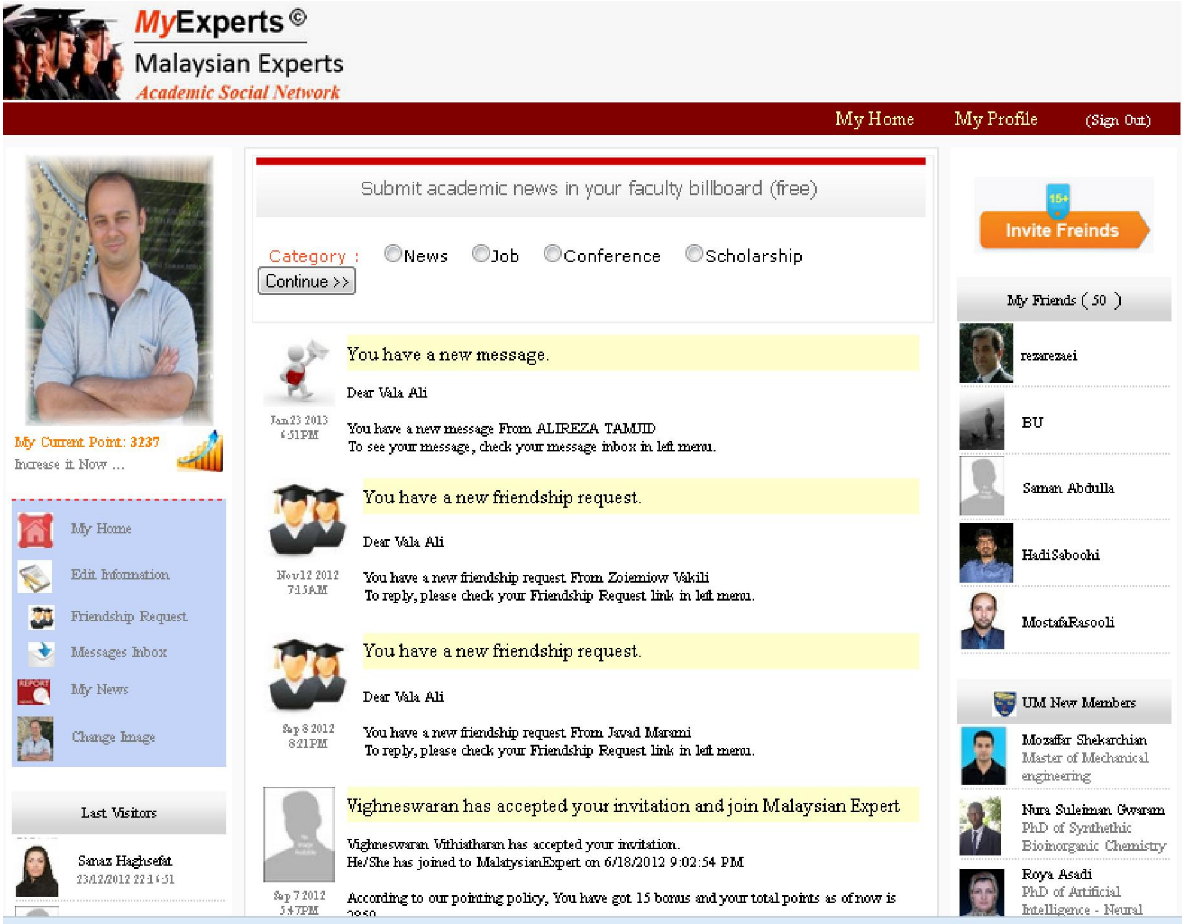
MyExpert User Interface

(Universities Webpage)



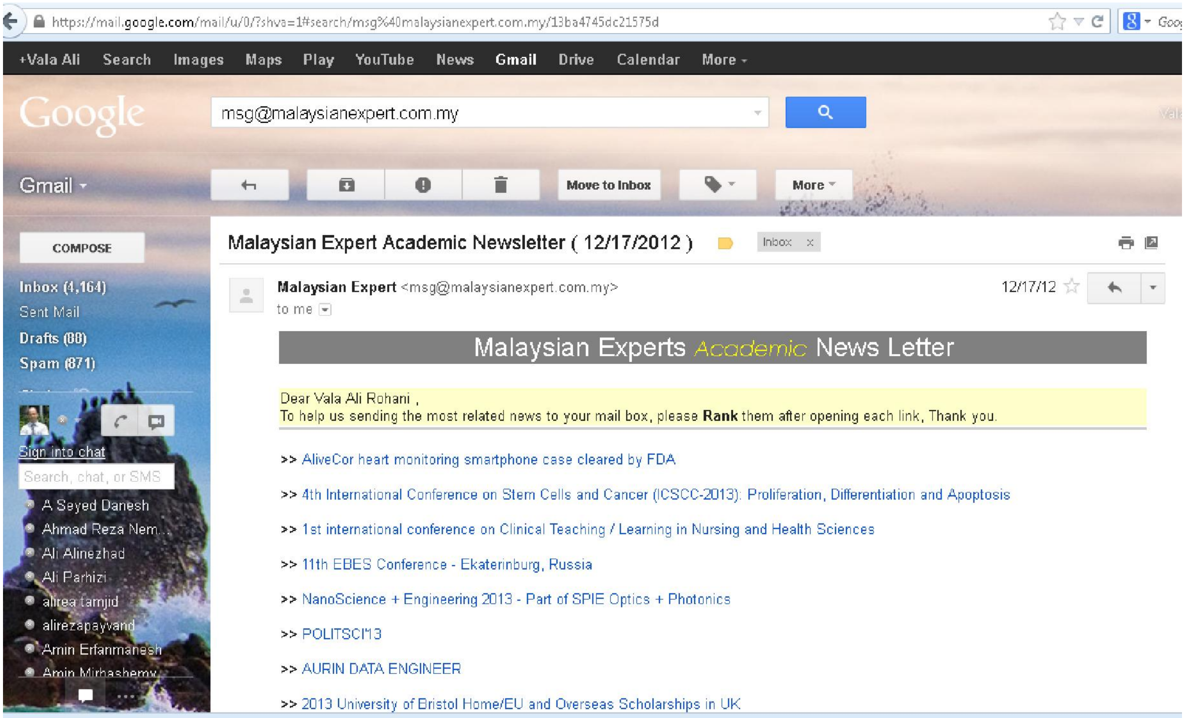
MyExpert User Interface

(Profile Management Webpage)



A Sample of MyExpert e-Newsletter

(Sent in 17th December 2012)



Appendix D

List of Publications from This Research

[1] Vala A. Rohani, Siew Hock Ow, "On Social Network Web Sites: Definition, Features, Architectures and Analysis Tools", Journal of Advances in computer research, 2 (2010), Pages 41 to 53.

[2] Vala A. Rohani, Siew Hock Ow, "A Social Network Based Solution for Integrating the Universities in Malaysia", Proceeding of the 4th International Conference on Postgraduate Education (ICPE-4 2010), Malaysia, Kuala Lumpur, 26-28 Nov 2010, ISBN 978-9675878-30-5, Pages 226 to 229. (Awarded as Best Poster Presentation)

[3] Vala A. Rohani, Siew Hock Ow, "A Framework for e-Content Generation, Management and Integration in MYREN Network", Proceeding of the IEEE 2011 International Conference on Data Engineering and Internet Technology (DEIT 2011), Bali, Indonesia, 15-17 March 2011, 978-1-4244-8581-9/11, Pages 679 to 682

[4] Vala A. Rohani, Siew Hock Ow, "Eliciting Essential Requirements for Social Networks in Academic Environments", Proceeding of 2011 IEEE Symposium on Computers & Informatics (ISCI 2011), Kuala Lumpur, Malaysia, 20-22 March 2011, 978-1-61284-690-3/11, Pages 171 to 176

[5] Vala A. Rohani, Siew Hock Ow, "Analysis of Centrality Rate in an Evolutionary Academic Social Network", Proceeding of 6th Symposium on Advances in Science

and Technology (6thSASTech), Kuala Lumpur, Malaysia, 24-25 March 2012, (Awarded as Best Poster Presentation)

[6] Vala A. Rohani, Siew Hock Ow, "A Framework for e-Content Generation, Management and Integration in MYREN Network ", Recent Progress in Data Engineering and Internet Technology, Lecture Notes in Electrical Engineering Volume 157, 2012, Pages 293 to 298

[7] Vala A. Rohani, Zarinah Mohd Kasirun, "The Emergence and Analysis of an Academic Social Network in Malaysia", 2nd Postgraduate Research Excellence Symposium (PGRES2012), Kuala Lumpur, Malaysia, Sep 2012

[8] Erfanmanesh, M., Rohani, V. A., & Abrizah, A., "Co-authorship network of scientometrics research collaboration." Malaysian Journal of Library & Information Science, 2012, 17(3), Pages 73-93

[9] Vala Ali Rohani, Zarinah Mohd Kasirun, Sameer Kumar and Shahaboddin Shamshirband, "An Effective Recommender Algorithm for Cold-start Problem in Academic Social Networks " Journal of Mathematical Problem in Engineering, 2014, (in press)

[10] Rezaei, A., Kasirun, Z. M., Rohani, V. A., & Khodadadi, T. (2013). Anomaly detection in Online Social Networks using structure-based technique. Paper presented at the International Conference on Information Science and Technology (ICIST), 2013.