A Recommender System for Breast Cancer Patients

by

© Momeneh Taban, M.Sc.

A thesis submitted to the
School of Graduate Studies
in partial fulfillment of the
requirements for the degree of
Master of Science

Department of Computer Science
Memorial University of Newfoundland

June 2014

St. John's Newfoundland Canada

Abstract

An ongoing challenge in the information age is finding information relevant to a particular need. One area in which this is particularly problematic is the medical domain, where patients suffering from certain conditions seek advice on managing their health. Personalized recommendations can be useful in this context. A recommender system can assist users to locate relevant information and choose the best option that matches their needs.

This thesis developed a Breast Cancer Recommender System (BCRS) which recommends health related articles appropriate for patients confronting breast cancer. BCRS applies a hybrid algorithm which combines collaborative filtering and content based approaches to generate recommendations. Article recommendations can be categorized in four main groups: life style, emotional concerns, risk factors and treatment. To examine the quality and perceived usefulness of article recommendations, a preliminary evaluation was conducted using female medical students of Memorial University.

Acknowledgments

First and for most I would like to thank my supervisor Dr. Jeffrey Parsons for his great support of my master study and research. I should acknowledge his guidance and his constructive comments in different stages of my research.

I would like to express my thanks to Faculty of Medicine of Memorial University and particularly Dr. Wanda Parsons for their support in evaluation phase of our system. Many thanks to the Memorial University medical students who tested this system; the experiment would not be possible without their help. Furthermore, I would like to thank Mr. Nolan White for his technical support.

I would like to express my deep gratitude to my parents, Shohreh Khaleghi and Houshang Taban for their love and support in the whole life. Last but not least, I would like to thank my husband Sadegh Ekrami for his spiritual support, love and encouragement during ups and downs.

Contents

Abstract **iii**

Acknowledgments **iv**

1 Introduction **6**

1.1 Recommender System 6

1.2 Problem Statement and the Need for Health Recommenders 7

1.3 Motivation and Purpose of the Thesis 8

1.4 Contributions 9

1.5 Thesis Outline 10

2 Recommender Systems and their Applications **11**

2.1 Recommender System 11

2.2 Recommendation Techniques 13

2.2.1 Collaborative Recommenders 13

2.2.2 Content-based Recommenders 14

2.2.3 Demographic Recommenders 15

2.2.4 Knowledge based Recommenders 15

2.2.5 Hybrid Recommenders 16
2.3 Reviews of Medical/ Health Recommenders 18
2.4 BCRS and Existed Health Recommender Systems 21
2.5 BCRS and Commercial Recommenders 23
2.6 Summary 24

3 BCRS Approach **27**
3.1 System Architecture 27
3.1.1 User Web Interface 28
3.1.2 Database Structure 30
3.1.3 Algorithmic Framework of BCRS 32
3.2 BCRS's Recommendation Techniques 33
3.2.1 Step1: Selecting Neighbors of a Target User 34
3.2.2 Step 2: Finding the Average Rating of the Target Document . 37
3.2.3 Step 3: Finding Similarity between Documents 37
3.2.4 Step 4: Producing Prediction for the Target Document 39
3.2.5 Step 5 and Step 6: Selecting Top-N Recommendations 39
3.3 Implementation 41
3.4 Summary 41

4 Evaluation **42**
4.1 Pre-evaluation Steps 42
4.2 Evaluation Metrics 43
4.2.1 Coverage 44
4.2.2 Mean Absolute Error and Related Metrics 44

4.2.3 Precision and Recall 45
4.2.4 ROC Curves 45
4.2.5 Prediction Rating Correlation 46
4.2.6 Half Life Utility Metric 47
4.2.7 NDPM 47
4.3 Data Set 48
4.4 Evaluation Scenario 49
4.5 Results and Analysis 51
4.6 Summary 55

5 Summary and Suggested Future Work **56**
5.1 Summary 56
5.2 Suggested Future Work 57

Bibliography **59**

Appendix A BCRS Code **67**

List of Figures

2.1 The overview of the proposed hybrid approach. 24

3.1 The BCRS architecture and components. 28

3.2 Four major categories of the health article recommendation. 29

3.3 Database structure. 31

3.4 General recommendation algorithm. 33

3.5 Overview of the proposed approach. 40

4.1 The evaluation process. 50

4.2 MAE for 3 different approaches. 52

4.3 BCRS's technology acceptance questions. 54

List of Tables

2.1 Hybridization Methods (Adopted from [11]). 17

2.2 An overview of health recommender systems. 26

4.1 Pearson correlation for 3 different approaches. 52

4.2 Standard deviation (stdv) of perceived usefulness (PU) and perceived ease of use (PEOU) for three different approaches. 54

Chapter 1

Introduction

1.1 Recommender System

Finding "relevant" or "interesting" items or information on the web is a challenging task. The Internet offers a vast number of choices (e.g., information, products, and services) of variable quality. Filtering and choosing among many options is a complex task. Recommender Systems (RSs) have emerged to provide methods to assist users in finding appropriate information[1]. They intend to provide relevant information that is interesting and useful for users according to their personal preferences. This personalization can be based on each user's profile, user's interest judgment, or the ratings that user gave to previously seen items. In other words, an RS helps users in the decision making process; it receives item and user information as input and provides personalized recommendations as output.

Various types of RSs have been developed, and recommendation techniques have been used in many different fields. However, they have been applied mostly in the

e-commerce domain [2] with applications in other domains being limited. The use of recommender systems in other domains is also promising and provides an area for further exploration and research [3]. One area in which RSs have recently evolved, and in which further work is needed, is the medical/health field.

1.2 Problem Statement and the Need for Health Recommenders

Continued growth of the Internet and ever increasing access to medical knowledge have resulted in growing interest among individuals in managing their health data. For example, a significant increase in online search for health information was found between 2005 and 2007 in Europe [4]. A 2011 survey [5] conducted to observe Americans' use of the Internet on a daily basis showed that eight out of ten Internet users search for health information.

People are increasingly willing to obtain information about their health issues, diseases and symptoms through online sources; however, unreliable information on the Internet and uncertainty about the quality of the information is an important obstacle to its use [6]. Moreover, it is difficult for users to search among all available information they receive through search engine queries to find relevant, personalized and useful information. As a result, there is a pressing need for improved personalized retrieval of health information [7] to address the aforementioned concerns. Recommender systems have been proven to be an effective tool in various domains; it is now the time to further develop this functionality in the health field. Applying RS

techniques in the health domain can address the problem of health information overload and it also supplies the personalized health information demand. Recommender systems could extract personalized health information through either the Internet or trustworthy predefined libraries as the primary sources of the health information.

1.3 Motivation and Purpose of the Thesis

The primary motivation of this thesis is to apply recommendation methods in a health domain which is different from traditional domains like e-commerce. This thesis explores how to make recommendation methods and algorithms compatible with the features and needs of a particular health information domain. There is a huge potential in the health/medical domain; one particular area in the health context is providing individualized health information. Patients have to cope with health information overload and finding useful and trustworthy information is problematic. To address these issues the thesis develops a recommender system to generate personalized health information recommendations for users.

Among health information seekers and users, patients affected by chronic disease need support beyond the normal health care services. According to statistics provided by the Canadian Cancer Society[1] in 2012, it was estimated that in 2013, 23,800 women would be diagnosed with breast cancer and 5,100 women would die from the disease. Some studies have been done to identify breast cancer patients information needs; for instance, [8] concluded from sample of 224 women with breast cancer, the Internet is an important tool to seek information after diagnosis and treatment. Breast cancer

[1] http://www.cancer.ca/

has been declared the second most prevalent cause of cancer death among Canadian women. Thus, this is a very important domain of health information that could benefit from availability of an RS.

This thesis develops a Breast Cancer Recommender System (BCRS), which recommends health related articles appropriate for patients confronting breast cancer. The proposed system provides recommendation in four main classes: life style, emotional concerns, risk factors and treatment.

1.4 Contributions

The contributions summarized here are further discussed in the related chapters and in the conclusion section in more detail.

- Introduction of a novel domain for application of RS in health.
- Development of a novel recommender system to personalize health information system.
- An analysis of 3 different RS approaches (the proposed hybrid approach, collaborative filtering and content based filtering) in producing article recommendation in the proposed domain.
- Reduction of cold start problem using a hybrid approach to generate recommendations.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2: provides an introduction to RS and a literature review of recommender system techniques. Then some of the important previous works on RS in the health domain are introduced. In addition, differences between the health domain and traditional domains (e.g., books and movies) are explained.

Chapter 3: illustrates the main components of BCRS architecture: user interface, www server, the embedded recommender algorithm and the database. A detailed discussion of our approach and the proposed recommendation algorithm is then provided. The algorithm uses a hybrid approach that switches between two recommender techniques depending on the situation (pure collaborative technique and a hybrid content based approach). A brief explanation of the system implementation is provided in this chapter.

Chapter 4: explains the main evaluation techniques which are used in recommender system domains and describes the experiment and results. Twelve medical students were invited to use BCRS, rate the recommendations and answer some questions regarding the system usefulness. To analyze the results mean average error and Pearson correlation measures have been used. The health article recommendations of 3 different approaches (our hybrid approach, collaborative filtering and content based) are compared using these measures.

Chapter 5: finally Chapter 5 presents a summary of the thesis contributions and outlines potential areas for future work.

Chapter 2

Recommender Systems and their Applications

2.1 Recommender System

With the explosive growth of the World Wide Web, many techniques (e.g. search engines, information filtering and information retrieval) have been developed to help users find information according to their interests, preferences and needs. Recommender systems extend traditional information filtering and information retrieval techniques. The difference between RS and "search engines" or "information retrieval systems" lies in the individualization or personalization provided by former [9]. An RS is a personalized information filter which intends to find relevant items/information expected to be interesting for a particular user. In other words, an RS guides users in an individualized way to find desired and useful objects (e.g., products, services and information) by filtering the abundant and immense variety of possible options. RS

techniques improve the decision making process by finding relevant items according to a user's information profile. Different formal definitions have been proposed in the literature for the recommendation problem. One of the most cited [3] defines the problem of recommendation as follow: let C be the set of users and let S to be the set of items that can be recommended. $u : C \times S \rightarrow R$ where u denotes a utility function and R is a totally order set. Then the recommender intends to choose item $s \in S$ for each user $c \in C$ which maximizes user's utility:

$$\forall\ s \in S,\ \forall\ c \in C,\ S_c = arg\ max\ u(C, S) \tag{2.1}$$

RS techniques are widely used in industry [10] since they increase user satisfaction, user fidelity, and number of sold items. They also help service/product providers to better understand what users need. They are used by well-known e-commerce companies like Amazon, Netflix, eBay, LinkedIn and Facebook.

An RS is a software tool or application consisting of 3 major parts; the main component is a *recommendation algorithm* which uses the other two components, *background data* and *input data*, to generate recommendations. Input data refers to user profile, and user rating/user purchasing or in general user behavior model (the way that the user interacts with the system). The third component of recommender system (background data) is the information that the system has before starting the recommendation process [11]. The background data and input data can be different according to various recommendation algorithms. The role of each component is clarified in Section 2.2.2

2.2 Recommendation Techniques

Different classes of recommendation algorithms are available; following is a summary of the main techniques used by RSs.

2.2.1 Collaborative Recommenders

Collaborative filtering (CF) is the most widely used type among all RS techniques. The basic idea of CF is to compare users in order to identify users with similar tastes (neighbors). This comparison is mostly done based on the similarity of ratings which users provide for the same items; however, clustering of neighbors is sometimes according to the properties of the items the users liked in past. The CF approach generates personalized recommendations for a user according to the taste of like-minded users. This technique is called user-to-user CF; an active user will be recommended items liked by users with common interests and preferences [12].

CF can also be item-based; the basic foundation is that items that have been rated in same way are most likely to share some features[13], so if a user liked one of them in past she might like the other similar rated items. Thereby, item-to-item CF finds the similarity between a target item i with the items that user has rated and liked previously and separate k most similar items to item i, the result is the candidate set $\mathbf{C}$. Then the similarity between the target item and each of the items in $\mathbf{C}$ is calculated $\{S_{i,1}, S_{i,2}, \ldots, S_{i,k}\}$. To compute this similarity, the algorithm considers only co-rated items; thereby to find similarity between item i and item k or $S_{i,k}$, it first separates users who rated both items and then use a similarity metric. Several measures can be applied such as cosine based similarity, adjusted cosine similarity

and correlation based similarity[14]. Finally the N most similar items are selected; the next step is to predict rating for each of these items. The predicted rating r is assigned to item i by taking the average of the rating given by target user to similar items to i [14]. CF technique shows great success in movie and music domains.

2.2.2 Content-based Recommenders

Two fundamental components of the content-based approach are item profile and user profile[15]. The former refers to the item's properties, attributes or features. The focus of content-based system is on the item's features, since the similarity of items is measured based on their corresponding properties. The user profile shows the use's interests and preferences; it can be a description of user's preferences that a user provides for recommender system, or it can be extracted by analyzing the user-recommender interaction history, through purchased items, seen items or those which are already rated by user [16]. The first step of a content based algorithm is determining the best match of a user profile with an item profile. This step forms the base of recommending items that might be interesting for the user. The general method is to recommend items similar to others that already match the user's interest. However, recommendations produced in this way are mostly repetitive since the system considers the user to like the same kind of items.

As discussed earlier, a recommender system consists of 3 major parts. To better understand the role of each component, consider the content-based approach. Background data is the features of items in data set (that the system should have before beginning the recommendation process), while the input data is the ratings users

provide for recommended items. The algorithm uses the background data and the features of haighly rated items to suggest new items to the user.

2.2.3 Demographic Recommenders

Demographic techniques firstly make different categorizations of users according to their personal attributes. They then provide recommendations for users based on the particular demographic class to which they belong [11]. For instance, consider a movie recommender system with potential users from various age ranges. In order to prepare movie recommendations in such a system, age category is one of the determining factors affecting the correctness of recommendations. It is hard for this technique to show the changes of a user's preferences over the time.

2.2.4 Knowledge based Recommenders

Knowledge based (KB) recommenders generate individualized suggestions based on inferences about users' needs and preferences. Knowledge based RSs have knowledge about whether particular items meet a particular user's needs. Three different sources of knowledge are used in this method: knowledge about the user; knowledge about items; and knowledge about how to match the items with the user's needs. There exist two well-known approaches of knowledge based recommenders: case based and constraint based [17]. While the former technique measures how the users' needs match the recommendations, the latter approach uses a predefined knowledge base containing rules and explicitly defined constraints about how to match the users' needs with features of items. In comparison with other recommendation techniques, a KB

recommender has some advantages; a prominent benefit is independence between generating recommendation and users' ratings, while the content based and collaborative filtering need the users' ratings to generate suggestions. The primary disadvantage is the need for knowledge engineering; this refers to constructing knowledge-based systems based on engineering methods where the knowledge is built according to logical and human-reasoning efforts [11], [16].

2.2.5 Hybrid Recommenders

Hybrid systems use a combination of the above-mentioned techniques; this combination provides better performance with fewer disadvantages of using individual techniques. A well-known example of such recommender is the Bellkor solution[1] to the Netflix prize. Hybrid recommenders neutralize the weaknesses and combine the advantages of different techniques. Hybrid approaches can be implemented in several ways: by making content-based and collaborative-based predictions separately and then combining them; by adding content-based capabilities to a collaborative-based approach (and vice versa); or by unifying the approaches into one model [11]. Many possible combination of recommending algorithm have been recognized including: Weighted, Switching, Mixed, Feature combination, Cascade, Feature augmentation and Meta-level. Table 2.1 (taken from [11]) shows the difference between these Hybridization Methods.

Recommendation systems are implemented in many different domains. These domains can be classified to four major categories: Services [18], Context-personalized (Google news recommender [19]), E-commerce (such as Amazon and eBay) and En-

[1] http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf

Table 2.1: Hybridization Methods (Adopted from [11]).

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

tertainment (e.g., for movies: Movielens [20]). The use of recommender systems in other domains is also promising and should be explored and researched. One area in which RSs have recently emerged is the health/medical field [21]; since it is a new realm, health/medical recommenders have received limited attentions in conferences and journals. The proposed system, BCRS, can be classified as context-personalized based on the aforementioned categories of RSs. The health domain's features and needs are different from the traditional domains of RSs, as will be later discussed in Section 2.4. The next section reviews some previous researches in the medicine/health domain in which recommender systems have been used.

2.3 Reviews of Medical/ Health Recommenders

Expert systems have been widely used in many areas of the health domain; however, some features of expert systems make them inappropriate in some health-related fields. Expert systems usually address the information personalization for a set of well-known users by using a set of pre-defined rules [22]. It is also difficult for expert systems to cover the changing situation of a patient since they are dependent on predefined knowledge and well-known users; when the patterns change across time, encoded rules need to be updated manually. In addition, they do not have the capability to make efficient use of online information. Moreover, the dependency of expert systems on human experts makes it difficult to increase the number of users or resources and creates a development bottleneck [23].

The application of recommender systems in medicine/ health could become one solution to tackle the problems of expert systems and at same time solve the problem of health information overload on the Internet.

Lee et al. [24] implemented a prototype recommender system using Clinical Decision Support System (CDSS) to assist users in having a better life style. The proposed system receives user information such as user's vital signs, life style, family history and user's chronic disease as inputs and recommend meals to users according to this information and the user's preferences. For each meal, a well-being index score is calculated according to the user's preferences, meal nutrition and the health status of the user. A medical doctor can also affect the recommendation by limiting the types of meal or ingredients that are not healthy for a particular user. There is no mention in the paper about the evaluation of this system. METABO diabetes life style recom-

mender [25] is a similar platform which recommends to diabetic patients appropriate food and physical activities. It is a knowledge-based hybrid system which collects information from both patients and doctors to provide personalized suggestions. As one part of the future work, the authors intend to evaluate the system to test if the proposed methods could work in practice. Likewise, in [26] a diet recommendation service for managing and preventing heart disease is presented. Analyzing vital signs, family disease history and user's food preferences, the system is able to give individualized recommendations to users. The system considers medical constraints for users with the help of a specialist. However, no experiment has been done to evaluate the system. The authors of [27] worked on a recipe recommender system using two different strategies: content-based and collaborative filtering. Their objective is finding out the more appropriate algorithm for this recommender system. They provided an evaluation to compare existing approaches with their proposed model. They conclude that using content-based technique in their approach can achieve a high coverage and reasonable accuracy.

HSRF [28], is a Health Service Recommendation Framework which supports patients in finding proper health care services. The suggestions are provided considering the similarities between users and services. User's health status and user's location are used in the process of personalization. To evaluate their system, they simply run it and test the functionality of the system. A further evaluation is postponed in order to have a long observation of the system. In [29], a system is developed to introduce a reliable physician to target patients. The recommendation is according to the patient's health condition and users are then supposed to rate the recommendations. Home care and nursery care is another medical sub-domain in which recommender

systems have been proposed; for example, the authors of [30] developed a personalized health recommendation system (PSRS) for the purpose of home-care environment. PSRS controls the environment of patients affected by chronic disease; it records patients' daily activities and habits to construct the personal model. Using this personal model, the system has access to health care services, safety alarm and recommendable services in the home. To make the services personalized, personal models, locations and the patients health status are used. An experiment was done using three test cases to test the system's functionality according to the implemented patterns. In order to facilitate information sharing and collaboration among users and the medical team, social network is another domain in which medical RS has recently emerged. PatientsLikeMe[2] and the health social network recommender system [31] for parents of autistic children, are two examples; the main components of these RSs is finding patients with similar conditions.

Delivering proper medical information to both users and doctors is explored in another group of health/ medical recommender systems. In [32], for example, they proposed a recommender system using rule-based expert systems to process information and patient's self-reported data to suggest clinical examinations for patients or physicians. In [33] a Health Recommender System (HRS) helps users to obtain more information regarding their disease and symptoms. Their system is integrated with Personal Health Record (PHR) to provide individualized recommendations. The proposed approach extracts information from Wikipedia, then finds the most relevant information that best matches the user's PHR and user's interests. The authors indicated, they intend to experiment the recommendation accuracy using the data

[2]http://www.patientslikeme.com/all/patients

provided by Heidelberg University Hospital. The same purpose of delivering information to patients was pursued by the authors of [34]; PHE or Personal Health Explorer is a knowledge based system, in which users can do semantic search according to their PHR. As the name implies, it is an explorer (like search engines) in which users have to enter queries to find the appropriate information. The RS works behind the scene to provide the user a ranked list of articles as the output. The user has to put extra effort to find the related and interesting information among the many recommended articles. Recently, [35] recommends health videos from YouTube to users in which recommendations are personalized based on the medical terms in the titles. In addition, there has been very limited evaluation of proposed systems.

The number of works in health domain is growing, but despite the huge potential, in comparison with other domains in which recommender systems are used, it is a minor topic of conferences and journals.

2.4 BCRS and Existed Health Recommender Systems

The existing approaches of health recommenders can be categorized in 2 main groups. 1) One category provides service recommendations such as the social network recommender system [31], health service recommenders [28], [29] and home care environment services [30]. 2) The second group intends to generate personalized health information recommendations. Category 2 can be divided to 2 sub categories: a) some research aims to supply the medical/ health team with personalized information, like

drug recommendation systems [36] or health education recommender systems [23]. b) Other works are patient centric such as [27] and [33]. BCRS can be classified as a member of the latter subcategory; since it generates health article recommendations to support breast cancer patients. Some of the previous works in (b) like life style and food recommender system ([24], [37], [27] and [26]) focus on only one aspect of patients information need and they are not comprehensive systems. Due to the design of their recommender engine, their algorithm can be applied to filter the diet information or life style changes information recommendations. For example in [37], the authors applied the rule-based reasoning (knowledge-based system) and the proposed domain ontologies to recommend foods to patients. Food knowledge is integrated with physiological data to generate recommendations. Health recommender system (HRS) (discussed earlier [33]) is an example of a system that aims to integrate RS with a personal health record to address personalization. It is a system that probably can be adopted to different domains in health, but was not evaluated the quality of recommendation approaches, so it might not be feasible to apply it in different domains with different needs and features. One of the advantages and, at the same time, drawbacks of HRS is the use of ontology and semantic network to represent the knowledge. As a result it is dependent on human experts to determine the ontology for each domain and also keep the knowledge updated. Recommendation systems in general assist users in overcoming the overload of health information they receive and offer users more personalized information. In other words, they provide a method of treating patients as individuals. There exist some studies on recommender systems to diagnose cancer [38] and other work has been done to direct prostate cancer patients to credible, useful and related informative websites [39]. To the best of my knowl-

edge no academic work has been done in order to develop a comprehensive system for delivering personalized health information to patients suffering from any type of cancer.

2.5 BCRS and Commercial Recommenders

The health/ medicine domain is different from the usual domains in which recommender systems have been used. In domains like movies or books, the recommenders investigate the users' history of interaction with system to discover the users' preferences. They realize the users' interests through analyzing the overall user's past behavior (such as her purchases, clicks, ratings, seen items etc.), while in the health domain like BCRS, a user's current need is the priority. Patients would be more interested to receive good informative articles relating to their current health conditions; their past interaction with the system (like ratings) maybe less important for generating recommendation. The ratings provided by users are used just for predicting a rating for an item. In the e-commerce area, the concern is finding a way to understand if a user will purchase an item (or in general if a user wants an item or not), whereas the nature of the health recommender domain is different and requires a different kind of strategy. For example, the traditional way to find similar users in other domains is to find similar rating patterns among users, but my system intends to find similar users considering the user profiles which include demographic information. The proposed system does not rely on the users' past rating patterns to find similarities among the users or articles. Unlike traditional recommender systems, BCRS focuses on just the user's current need which we believe to enhance user

satisfaction and system usefulness. The proposed approach and the recommender algorithm are explained in detail in the next chapter. Pure collaborative filtering suffers from the "cold start" problem, or new user/ item problem [40], [3]. When a new user or item enters to a system, it is difficult to find similar users and similar items since there does not exist enough information. I propose a method of switching between pure CF and a hybrid CF technique to alleviate the shortages of cold start problem. Figure 2.1 clarifies the general hybrid approach applied in BCRS. To make the switching decision, two criteria are applied (discussed later in section 3.1.3).

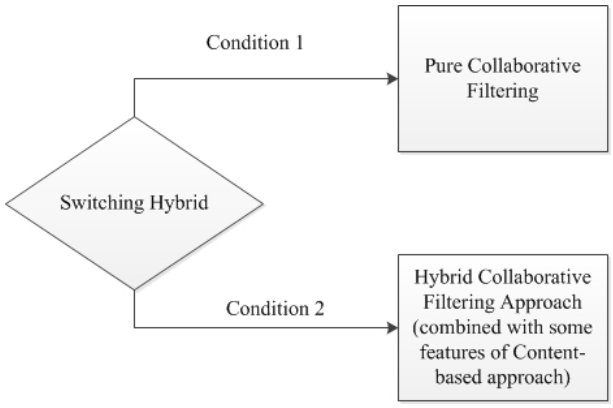

Figure 2.1: The overview of the proposed hybrid approach.

2.6 Summary

In this chapter, a brief introduction of recommender system was presented and main techniques provided by RS were reviewed. Recommender system technology has proven its usability in several domains, particularly in electronic commerce, but its application in health is new and the amount of research in this area is limited.

Some of the research on RS in the health domain were described and categorized; the overview of these works was presented in Table 2.2. BCRS is grouped as a patient-centric system that provides health information personalization.

This chapter also identified differences between the health domain like BCRS and traditional domains like books and movies. The recommender algorithm is designed to be compatible with the features of the domain. A hybrid approach is used to alleviate the cold start problem in recommender system.

Table 2.2: An overview of health recommender systems.

Name	Medical Sub-domain	Recommender Approach	Evaluation
Life Style Recommender System [24]	Food recommender	Not specified	Not specified
METABO [25]	Food/ physical activity recommender for diabetic patients	Hybrid-knowledge based	Postponed to future work
Diet Recommender System [26]	Food recommender for preventing and managing heart disease	Not specified	Not specified
Personalized Recipe Recommender System [27]	Intelligent food planing	Collaborative filtering and content-based approach	Comparing two different approaches
Health Service Recommender System (HSRF) [28]	Services	Content-based	Evaluation of feasibility and functionality of the system
Personalized Health Recommender System (PSRS) [30]	Home care/ nursery care	Collaborative filtering	Evaluation of the system functionality
Health Social Network Recommender System [31]	Information sharing	Not specified	Evaluation of text classification and related explanation
Web-based Health Recommender System [32]	Deliver information to patients	Hybrid- Knowledge based	Not specified
Health Recommender System (HRS) [33]	Deliver information to patients	Content-based	Postponed to future work
Personal Health Explorer [34]	Health search engine	Knowledge-based	Not Specified

Chapter 3

BCRS Approach

The proposed system is intended to provide health article recommendations to patients affected by breast cancer. BCRS is an online system that uses a predefined database of articles covering different topics in breast cancer. Note that the data base can be changed over time. This chapter details the BCRS architecture and the proposed recommendation approach.

3.1 System Architecture

Figure 3.1 shows the main components of BCRS and the system architecture. As can be seen, the main components are user web interface, data base and hybrid recommender. Each component is described in more details in following sections.

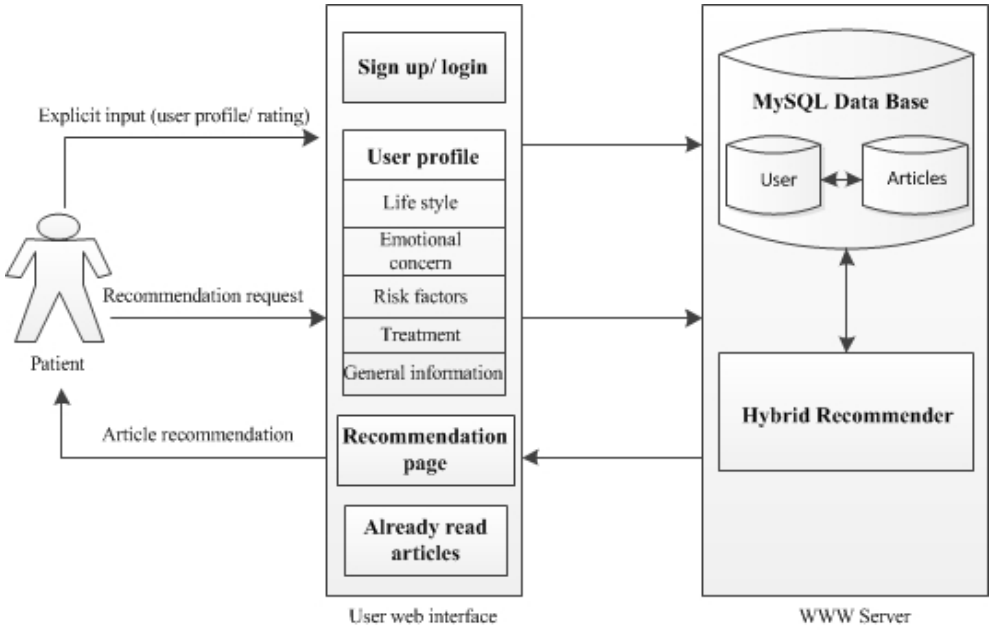

Figure 3.1: The BCRS architecture and components.

3.1.1 User Web Interface

Patients interact with the system through the user interface. The user interface consists of 3 main parts: user profile, recommendation page and the page they can see already read articles. The user interface is further connected to the data base to save and collect data. To sign up in the system, patients are required to fill in the user's profile which is one of the main elements that system uses to recommend personalized articles. BCRS provides article recommendations in four main categories (Figure 3.2) and patients have access to these categories through the recommendation page. The profile information is similar to what is shown in Figure 3.2, but it also includes general information like contact information, age, education level, height and weight. Users can benefit from relevant and useful informative articles according to their current health status. The user profiles are constructed based on analysis of

literature and factors determined to be relevant for recommending various topics. To understand a user's current health condition and information need in each category, BCRS utilizes the user's profile information.

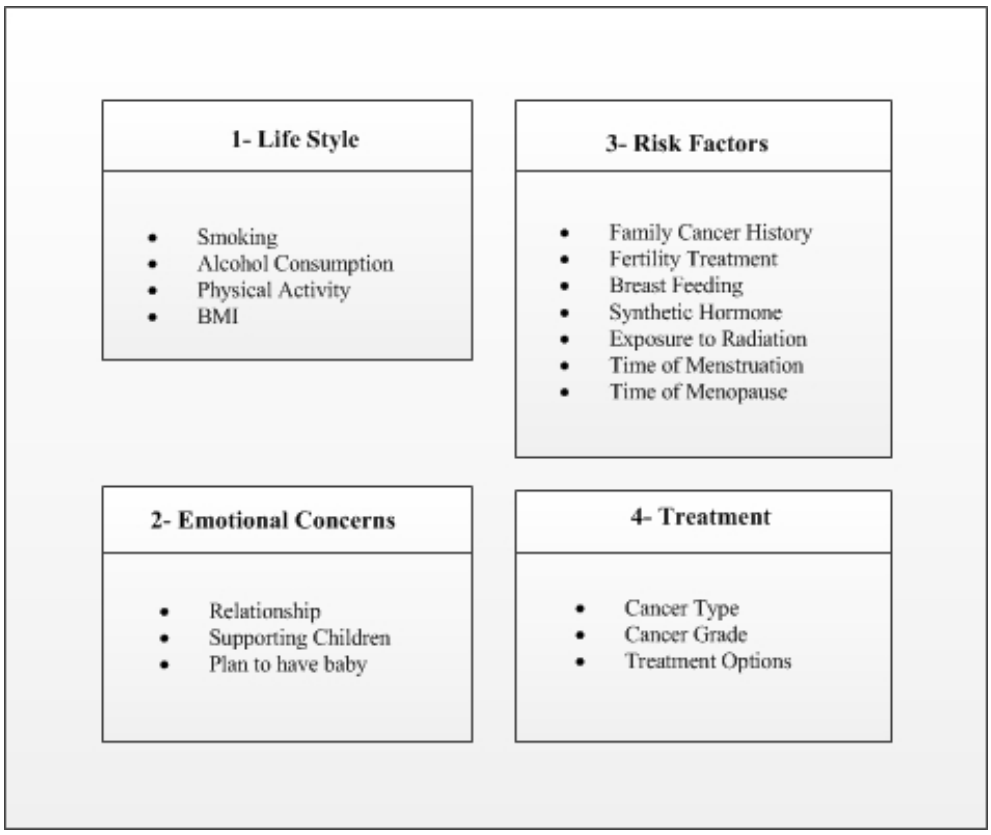

Figure 3.2: Four major categories of the health article recommendation.

The above mentioned categories address the common information needs of breast cancer patients and were selected based on some previous studies on breast cancer patients' needs of information and also classified articles in some credible websites [1] [2] [3]

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

[2] http://www.nlm.nih.gov/medlineplus/

[3] http://www.cancer.org/index

[4] [5] [6]. For example, a study on 35 women with breast cancer [41] showed that patients who are willing to receive more detailed information are more interested in receiving explanations of their diagnosis, treatment alternatives, and treatment procedures. Another report [42] concluded (from a study on 888 cancer patients) that psychology and daily living domains are the two highest information need of cancer patients. The authors of [43] conducted a study in which 228 women suffered from breast cancer participated. The results indicated that 71% of participants discussed fertility related issues with a health professional.

There exist two explicit inputs to the system: (1) the information from users' profiles through which the recommender knows users; (2) when the system generates recommendations in each of the four categories, the users provide ratings for health articles according to their usefulness.

The rating is explicit and based on a five point Likert scale: poor, fair, average, good and excellent. User profile and articles are the ways patients interact with the BCRS.

3.1.2 Database Structure

Figure 3.3 illustrates the data base components and structure.

[4] http://www.womenshealth.gov/index.html

[5] http://www.nationalbreastcancer.org/

[6] http://www.cbcf.org/ontario/Pages/default.aspx

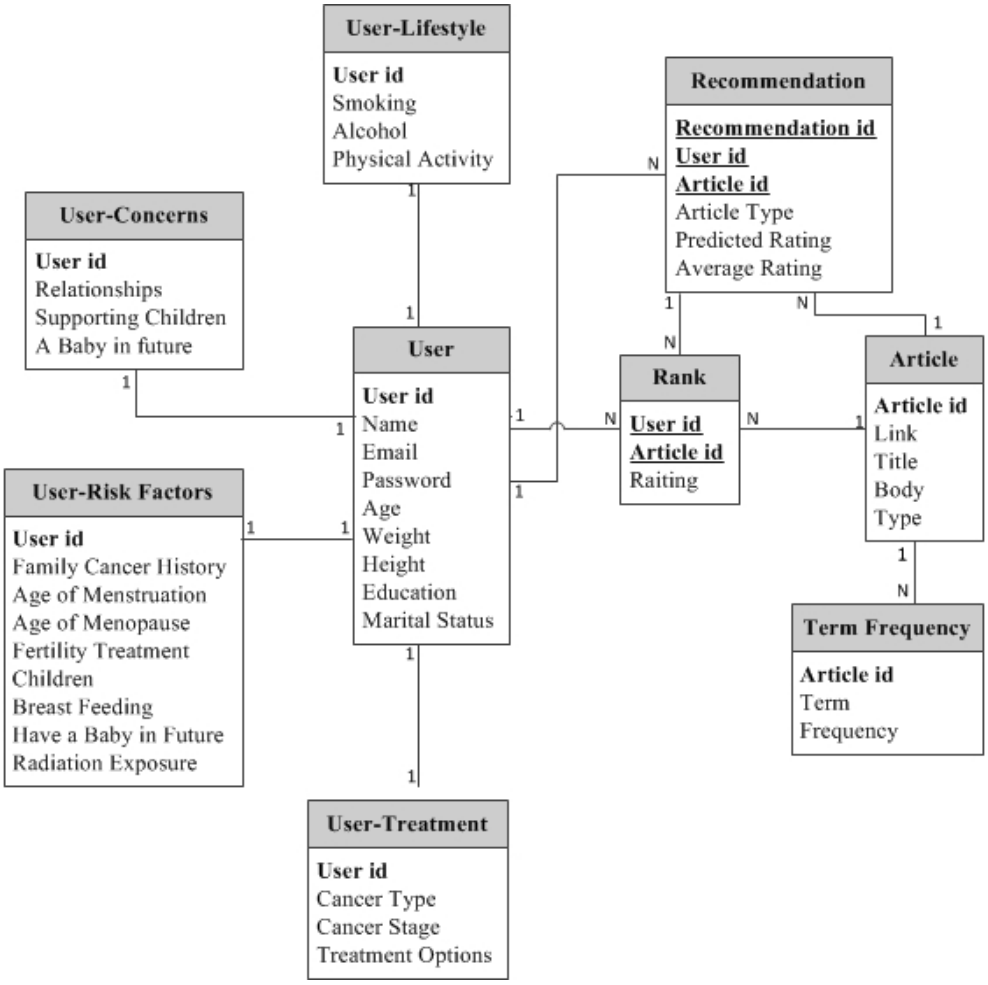

Figure 3.3: Database structure.

As can be seen, the database contains 9 tables. The user table contains the demographic profile of users; health-related user information is recorded in 4 other tables: User-Lifestyle, User-Concerns, User-Treatment and User-Risk-Factors. User profiles are used to find the similarity between users (finding neighbors). Each user can receive many recommendations and give ratings to several recommended articles. The articles have been recorded in the table Article and each article has been saved with its web resource link, title, body and the type. There exist 4 types of articles

based on 4 categories of recommendations which is mentioned in the figure 3.2 (Type1. life style, type2. emotional concerns, type3. risk factors and type 4. treatment). Each article is represented by the including terms with different term frequencies. The table, Term Frequency, stores the terms of an article and the number of their occurrence in that article which is used to calculate the similarity between health documents. Measuring the user-user similarity and document-document similarity, the BCRS is ready to generate recommendations. Each article can be suggested through several recommendations and each recommendation contains various articles. All generated article recommendations have been recorded in Recommendation table along with its predicted rating produced by the BCRS. Each article might be ranked differently by different users and the rating information have been stored in "Rank" table.

3.1.3 Algorithmic Framework of BCRS

The proposed recommender algorithm utilizes both input data and background data to provide suggestions. To recommend articles in BCRS, a hybrid technique has been used which switches between a pure collaborative approach and a hybrid collaborative filtering approach. Figure 3.4 presents the general structure of the recommendation algorithm.

The next section provides a detailed description of the recommendation algorithm.

```
00- Begin
01-   user i requests recommendation in one of the 4 fields F_k (1 ≤ k ≤ 4)
02-   documents are selected from the set of articles where sub type (document_j) = F_k. The output
      is the set of candidate docuents d_1, d_2, ..., d_j
03-     foreach user in the user set U do
04-       find similarity to user i (by comparing demographic information of user profiles)
05-       select top K similar users as neighbours N_i
06-     end for
07-     foreach document d_j do
08-       if d_j is already read at least by one of the neighbours then
09-         predict the rating for d_j, by taking the average ratings that neighbours N_i provide
            for d_j
10-       end if
11-       else if none of the neighbours have read d_j then
12-         find similarity between d_j and all documents that neighbours read in the same field
13-         predict rating for d_j based on the ratings of similar documents
14-       end if
15-       select top M documents according to their predicted ratings
16-     end for
17-   recommend top M articles to user i
18-   user i rates these articles
19- end
```

Figure 3.4: General recommendation algorithm.

3.2 BCRS's Recommendation Techniques

This section provides a detailed outline of the recommendation approach. The process can be divided into six main stages. The most challenging tasks are measuring user-user correlation and article-article correlation. All steps are summarized in Figure 3.5

3.2.1 Step1: Selecting Neighbors of a Target User

The core of collaborative filtering is finding similar users to a target user or selecting neighbors. The traditional way of calculating user similarity is finding the similar rating pattern on the set of items co-rated by users [44]. In order to calculate these rating correlations, there exist several formulas, including cosine based similarity, Pearson correlation, correlation based similarity and adjusted cosine based similarity [45]. Among them Pearson and cosine similarity measures are most popular.

After measuring the rating similarity between users, two techniques can be applied for selecting neighbors: one is top N neighbors, in which best neighbors are selected. The other one is threshold based selection in which users whose similarities exceed a certain threshold are selected as neighbors. The collaborative filtering approach usually compares users' rating on the same items to find out the users similarities. This approach is reasonable for domains in which the concern is whether a user wants an item or not. The users who showed the same interest in same items are more probable to show similar interest in the other items. In contrast, in BCRS's domain, the user's current need for individualized health information is the concern. Past behavior and past interaction history with the system would not be a reliable measure to find the users with similar taste (need). Instead, the user profile information/demographic information influences the process of finding neighbors (users with similar need), because users with similar needs are more probable to show the same interest in particular articles. For example, patients with same cancer type and risk factors are more interested to receive a specific type of information. For this reason, only the user profile and demographic information are considered to compute

the users' similarities.

However, the two above-mentioned techniques in finding neighbors can be combined; in other words, the users' rating similarity and demographic information similarity can be merged using a weighting and hybrid approach. To compute neighbors by comparing their demographic information, the following approach is applied.

As discussed earlier, the user profile information is classified in 4 main groups; each classification has a fixed number of features.

$$\begin{cases} i = 1 & \text{Life Style} \\ i = 2 & \text{Emotional Concerns} \\ i = 3 & \text{Risk Factors} \\ i = 4 & \text{Treatment} \end{cases}$$

N_{fi} $(1 \leq i \leq 4)$ is the number of features in each group:

$$\begin{cases} N_{f1} = 3 \\ N_{f2} = 3 \\ N_{f3} = 8 \\ N_{f4} = 3 \end{cases}$$

N_{f1} is the number of features in table User-Lifestyle of user profile (smoking, drinking alcohol and physical activity). The table User-Concerns consists of 3 attributes: relationship, supporting children and have a baby in future ($N_{f2} = 3$). N_{f3} and N_{f4} are respectively the number of attributes of User-Risk Factors and User-Treatment.

The users' similarity is then calculated based on the following formula:

$$
\begin{aligned}
& sim(U_x, U_y) = \frac{w_1 \times N_{f1}^{U_x \cap U_y}}{N_{f1}} + \frac{w_2 \times N_{f2}^{U_x \cap U_y}}{N_{f2}} + \frac{w_3 \times N_{f3}^{U_x \cap U_y}}{N_{f3}} + \frac{w_4 \times N_{f4}^{U_x \cap U_y}}{N_{f4}} \\
& \textstyle\sum_{i=1}^{4} w_i = 1, \\
& 0 \leq sim(U_x, U_y) \leq 1
\end{aligned}
\tag{3.1}
$$

where $N_{f1}^{U_x \cap U_y}$ is the number of same features between User x and User y in group i and w_i is the relative importance of group i in finding neighbors. Assume a user requests recommendation in lifestyle. To find the neighbors of the user, the life style category of user profile plays a more important role in compare with the other categories of user profile. As a result, w_1 should be bigger than the other coefficients: w_2, w_3 and w_4. To illustrate consider the importance of the category in which the user request recommendation as two times higher than the other categories in user profile (to find the neighbors of an active user). Having this in mind, since there are 4 categories and $\sum_{i=1}^{4} w_i = 1$, so the ratio of w_i would be $\frac{1}{5}$. For instance, if the user orders a recommendation in Treatment field, w_4 would be set to $\frac{2}{5}$ and w_1, w_2 and w_3 would be assigned $\frac{1}{5}$, so $\sum_{i=1}^{4} w_i = 1$. For this prototype system the coefficient in finding neighbors are set to be fixed numbers but further work is needed to experiment the system with assigning the dynamic amounts to the related confidants.

In the evaluation all users which receive a threshold exceeding 0.65 are considered as the neighbors of the target user. Note that this threshold is arbitrary and future work could experiment values of threshold.

3.2.2 Step 2: Finding the Average Rating of the Target Document

If the target article has been already read at least by one of the neighbors, the normal approach to predict rating of active user u for target document d_i is to compute the average rating based on the neighbors' ratings for this target article.

$$\begin{aligned} &\text{Average rating of} d_i : r_u d_i = \frac{1}{|N_u d_i|} \sum r_\gamma d_i \\ &\gamma \ \in \ N_u d_i \end{aligned} \tag{3.2}$$

where $N_u d_i$ is the number of neighbors which read d_i beforehand. γ is one of the neighbors who read document d_i and $r_\gamma d_i$ is the rating that user γ provided for document d_i.

When the document is a new one, none of the neighbors have read it before. In that case, the algorithm finds similar documents to the ones neighbor read in the same category.

3.2.3 Step 3: Finding Similarity between Documents

If the invoked article is a new one, the algorithm then discovers the articles in the same field that neighbors read in advance. The similarity of the target article and all of these articles should be calculated separately. Several models have been applied to represent a text document; among them Vector Space Model or VSM is widely used [46]. In this technique, each document d_i is represented by a vector containing the weights of all its included terms. To find the weights of each term, a widely used method in literature is TF-IDF (Term Frequency- Inverse Document Frequency) [47]. TF-IDF is a term

weighting technique that shows how important every word is in a document. TD-IDF weights all words excluding stop words (conjunctions, prepositions and pronouns) [48]. TF-IDF is composed of two main parts: a) normalization of term frequency (TF), which is the number of times each word occurs in the document divided by the number of whole words in that document, and b) inverse document frequency (IDF) refers to the logarithm of the number of the whole number of documents in the corpus divided by the number of documents containing a particular term [21]. Equation 3.3 shows how TF-IDF is computed.

$$TF - IDF(t_k, d_j) = TF(t_k, d_j).log\frac{N}{n_k} \tag{3.3}$$

where N is the number of documents in corpus and n_k is the number of documents in which term t_k appears.

As discussed in [21], cosine normalization equation (3.4) is used to make the length of all documents' vectors equal. It also put the range of weights in [0, 1] interval.

$$W_{k,j} = \frac{TF(t_k, d_j)}{\sqrt{\sum_1^{|T|} TF - IDF(t_k, d_j)^2}} \tag{3.4}$$

where $w_{k,j}$ is the weight of term k in document j.

Finally, a similarity measure is required to find the relation between two documents. As explained earlier, several similarity measures have been applied in different works; we use popular cosine similarity measures. Equation (3.5) is how the similarity

between document d_i and document d_j is computed [21] .

$$Sim(d_j, d_i) = \frac{\sum w_{k,j}.w_{k,i}}{\sqrt{\sum w_{k,j}^2}.\sqrt{\sum w_{k,j}^2}} \tag{3.5}$$

To conclude this step, each document is usually represented with its words and each term receive a score using TF-IDF formula. Documents are scanned to extract words and ignore the stop words. Afterwards, the cosine similarity can be used to find the similarity between articles.

3.2.4 Step 4: Producing Prediction for the Target Document

The rating a target user u will give to the target article d_i is $r_u d_i$ that is predicted by BCRS based on the ratings that neighbors N provide for documents (d_j) that are in the same field with target document. k is the number of all documents that are the same field as target document and neighbors read them already.

$$\begin{aligned} r_u d_i &= \frac{\sum_1^n \sum_{j=1}^k r_\gamma d_j \times sim(d_i, d_j)}{\sum_1^n \sum_{j=1}^k sim(d_i, d_j)} \\ \gamma &\in N, i \neq j \end{aligned} \tag{3.6}$$

where $r_\gamma d_j$ is the rating of the neighbor γ to document j and n is the number of neighbors of the target user.

3.2.5 Step 5 and Step 6: Selecting Top-N Recommendations

After the system predicts rating for all articles, the top M rated articles are selected in the next stage. There are two approaches to select articles: (1) the articles whose predicted ratings exceed a certain threshold are selected; (2) the top M articles which

receives best predicted ratings are suggested to users. The latter approach is used by BCRS. These articles will then be suggested to the active user and will be rated by her (step 6). Ratings of the target user will be recorded to the system for further use.

Figure 3.5 represents the summary of aforementioned steps.

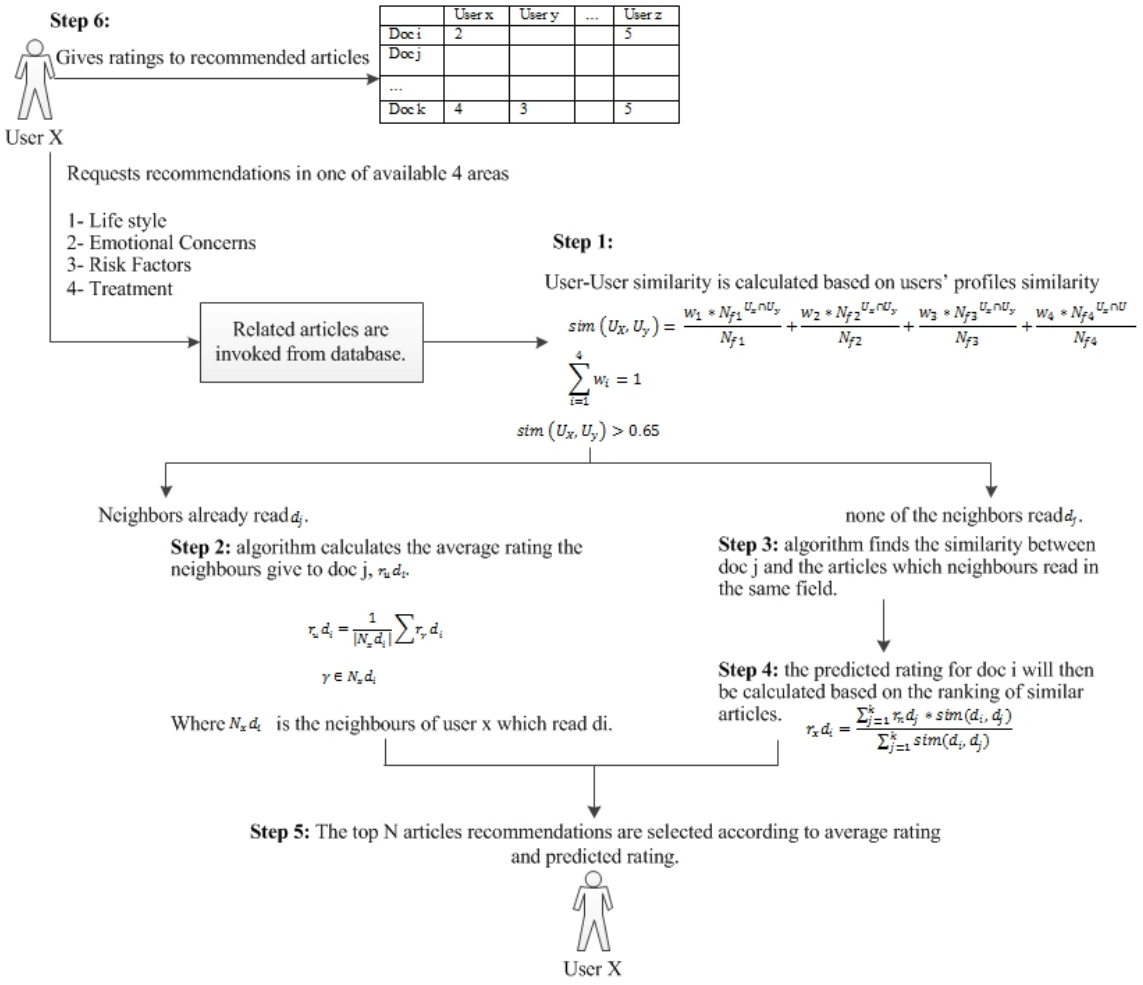

Figure 3.5: Overview of the proposed approach.

3.3 Implementation

In order to create a dynamic website, this recommender system is developed and coded in PHP and HTML are further connected to MySQL database to store the data. The BCRS's data base and program files are hosted on Memorial University webserver, which provides a secure and reliable platform for our system. The BCRS's program are presented in Appendix A.

3.4 Summary

In this chapter, the main components of the BCRS were presented: user interface, www server and the embedded recommender algorithm and database. The recommendation algorithm and all included steps were discussed in detail. The main critical steps are measuring user-user similarity and article-article similarity. The applied hybrid recommender approach switches between two recommender techniques which helps to alleviate the problem of new user/item. A brief discussion was provided regarding the implementation phase. The proposed system has been developed using PHP and HTML. Code and MySQL database are hosted on one of the Memorial University webserver.

Chapter 4

Evaluation

4.1 Pre-evaluation Steps

Recommender systems have been applied in a variety of domains to satisfy different goals; therefore, various evaluation techniques have been used to evaluate them. Our system goal is to assists patients with breast cancer to find relevant and useful health information.

The purpose of evaluation is to investigate whether the algorithm provides recommendations that matches a user's needs [10]. Before starting the evaluation of our recommender system, the user's tasks and recommendation task should be recognized in the system [49]. To this end, some pre-explanation is required.

Recommender systems assist users in two ways [9]; they can optimize an existing task or they can introduce a new task which did not already exist. In our case, the task of searching for an article already exists; instead, the system enhances the quality and speed of the task by reducing the number of bad and unrelated articles

and introducing articles relating to the patients' status that might be useful to them. This task is valuable since it automates parts of the subtasks that should be done by users.

Now the user task and recommendation task can more easily be distinguished. Considering the work done by Herlocker et. al [10], the user task in our system can be specified as "Find good items". This is because patients would be interested in receiving good health articles relating to their status.

Determining the system goal and user's task, the next step is to identify the recommendation task. The recommendation task is to connect the user task of interest and the system goal. As discussed earlier, our system generates a rating for each item which shows how a particular user is predicted to like the item, and these ratings are not visible to users (instead they are stored in data base for further analysis). Considering the approach taken by Gunawardana et. al [49], the recommendation task can be categorized as "Predicting Rating". In other words, recommendation task attempts to generate a recommendation list of N good articles based on the predicted ratings. Identifying the recommendation task and user tasks are helpful in the choice of the proper evaluation metric.

4.2 Evaluation Metrics

Recommender systems have been evaluated in various ways using different metrics according to the system goals, user tasks and recommender tasks. This section surveys a number of widely used techniques.

4.2.1 Coverage

Lack of data can put a restriction on recommenders. As a result, a recommender engine may not be able to predict a rating for every item. In the system in which the recommender task is predicting rating for items, it is not appropriate for the recommender not to cover the whole item set. In general, coverage is an evaluation metric which presents the percentage of the data set for which the system can predict rating.

4.2.2 Mean Absolute Error and Related Metrics

Mean absolute error (MAE) is classified as an accuracy metric that shows how the predicted rating for an item is different from the user's true rating. As stated in [10], most previous research has applied accuracy metrics to evaluate systems. MAE is calculated by a simple formula 4.1 which shows deviation between a user true ranking (r_i) and a predicted rating (p_i).

$$MAE = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N} \tag{4.1}$$

There exist other derived formulas from MAE like mean squared error, root mean squared error (which emphasizes large errors) and normalized mean average error [50].

MAE is widely used for recommenders that predict rating for items and the accuracy of prediction can be assessed through MAE or any of the other derived equations.

4.2.3 Precision and Recall

Precision shows the probability that a selected item is relevant, while Recall represents the probability that a relevant item will be selected. Based on this explanation, the item set is classified into relevant and non-relevant items. Equation 4.2 and 4.3 express how the precision and recall are calculated respectively [50].

$$P = \frac{N_{rs}}{N_s} \tag{4.2}$$

where N_{rs} is the number of relevant items selected and N_s is the total number of items selected.

$$R = \frac{N_{rs}}{N_r} \tag{4.3}$$

where N_r is the number of relevant items.

There have been some effort to combine these two approaches as a single formula; for example, F_1 metric have been derived by combining Precision and Recall [10].

$$F_1 = \frac{2PR}{P + R} \tag{4.4}$$

Precision and recall are suitable for tasks in which the rating is binary; the item is selected or not. They are not appropriate when a numeric scale is used for rating the items.

4.2.4 ROC Curves

Receiver Operating Characteristic or ROC curve can be considered as an alternative for precision and recall. According to [10] "ROC models the extent to which an

information system can distinguish between signal (relevance) and noise". ROC is a diagram representing two distributions; one indicates how the system predict a relevance level for each unrelated item, while the other distribution represents the probability for relevant items. The more these distribution fall apart, the better ability system has to discriminate between relevant and non-relevant items. Using this measure, a single number is calculated for the whole performance of the system. As with precision and recall, ROC works in a binary scale; this is for classification tasks in which the items are either relevant or non-relevant.

4.2.5 Prediction Rating Correlation

Two main classes for correlation measures have been used [50]: Pearson correlation and Spearman's. Pearson correlation indicates the linear relationship between two list of variables x's and y's (4.5).

$$C = \frac{\sum(x - \bar{x})(y - \bar{y})}{n \times stdev(x)stdev(y)} \tag{4.5}$$

where $\bar{x}$ and $\bar{y}$ are the mean of x's and y's and n is the number of rows in the list.

Spearman is the other candidate that is calculated the same with Pearson correlation, but shows how two different rankings agree independent of the actual values of the variables. Spearman requires the data to be ordinal and is carried out on the rank of data. Its calculation is represented in the following equation (4.6).

$$r = \frac{\sum(u - \bar{u})(v - \bar{v})}{n \times stdev(u)stdev(v)} \tag{4.6}$$

where $\bar{u}$ and $\bar{v}$ are the mean of u's and v's (ranks of variables) and n is the number

of rows in the list.

These measures cannot evaluate the accuracy of individual prediction; they are able to compare the full system ranking with the full user ranking.

4.2.6 Half Life Utility Metric

Consider the recommending scenario where a user is provided with a ranked list of recommendations. Users are not usually interested to browse the whole list. Half Life Utility is suggested for these situations to analyze the utility of the ranked list. Given a list of recommendations, the user will focus on the first items of the list and ignore the others. The probability of being selected for the items down the list will drop exponentially [49]. This measure is calculated as shown in equation (4.8).

$$R_u = \sum_j \frac{max(r_{u,j} - d, 0)}{2^{(j-1)/(\alpha-1)}} \tag{4.7}$$

where $r_{u,j}$ is the rating for item j which is provided by user u. α is the location of item in the list that might be selected with probability of 0.5 and $d, 0$ is default rating which is defined on neutral or slightly negative rating [49].

4.2.7 NDPM

Normalized Distance-based Performance Measure ($NDPM$) is used to compare two different weakly ordered rankings [45]. The equation (4.8) shows how to $NDPM$ is calculated.

$$NDPM = \frac{2C^- + C^u}{2C^i} \tag{4.8}$$

where C^- is the number of times that there exist a contradiction between user rating and system rating. C^u is the total number of times that users rank item i better than item j and the system, on the other hand, rank item i and j equally. Finally C^i is the number of preferred pairs of items in which user preferred item i to item j. $NDPM$ does not evaluate the prediction value and it only evaluates the ordering of the recommendation.

4.3 Data Set

To test BCRS, a set of articles was selected from some well-known medical web sites, including WebMD, MedlinePlus, Canadian Cancer Society, and some other credible websites. The articles were picked based on the aforementioned categories of recommendations to cover part of breast cancer patients common information needs; namely life style, risk factors, emotional concerns and treatment.

Due to sensitive nature of patient data, it was not feasible to use real breast cancer patients to test the BCRS. In the case of this domain, one of the best alternatives is to use people with significant knowledge of the domain. Medical students are familiar with patients' needs and concerns, and are in a good position to evaluate which information meet the needs of the patients.

A small evaluation was conducted with 12 medical students to check the performance of the system in predicting rating for article recommendations and to compare BCRS approach with two other common approaches.

The result of this study would be useful especially in comparing techniques with each other and understanding users' ideas about usefulness and ease of the use of the

BCRS. However, to be able to drawn a statistically significant conclusions about the system performance, a large scale evaluation of BCRS is needed.

The Lack of real patients as participants put limitation on our evaluation phase. We had to simulate the required information of 110 patients' user profiles. Furthermore, to train data we ranked 65 of the articles on behalf of the hypothetical users. The data currently contains 326 ratings over 75 users on 65 articles.

4.4 Evaluation Scenario

The study was performed online; test subjects were twelve volunteer female medical students from Memorial University. An invitation email containing the URL address of the study was sent to the Faculty of Medicine of Memorial University to distribute among medical students. Figure 4.1 shows the evaluation process. Each participant was provided with a user name and password of a hypothetical patient to log in to the system. In order to have a better picture of the patient role, they first reviewed the simulated patient's profile. Next, four categories of recommendations were provided: life style, treatment, risk factors and emotional concerns. Each participant was asked to request article recommendations in just one of these four categories (categories were varied across participants). They were asked to read the articles and rate them on a five point scale. After reading and rating all articles, they were asked to complete a questionnaire. Eight questions were developed to analyze the usefulness and easiness of use of the BCRS. The questions were designed based on the Technology Acceptance Model (TAM) [51]. TAM contains two main components: a) perceived usefulness (PU) indicates how much a system can enhance the user's job performance and b)

perceived easiness of use (PEOU) represents the extent to which using the system is easy (both considers the users' ideas as the measure). In our case, the wording was modified to assess usefulness of the articles to a patient filling the profile the participant was asked to review.

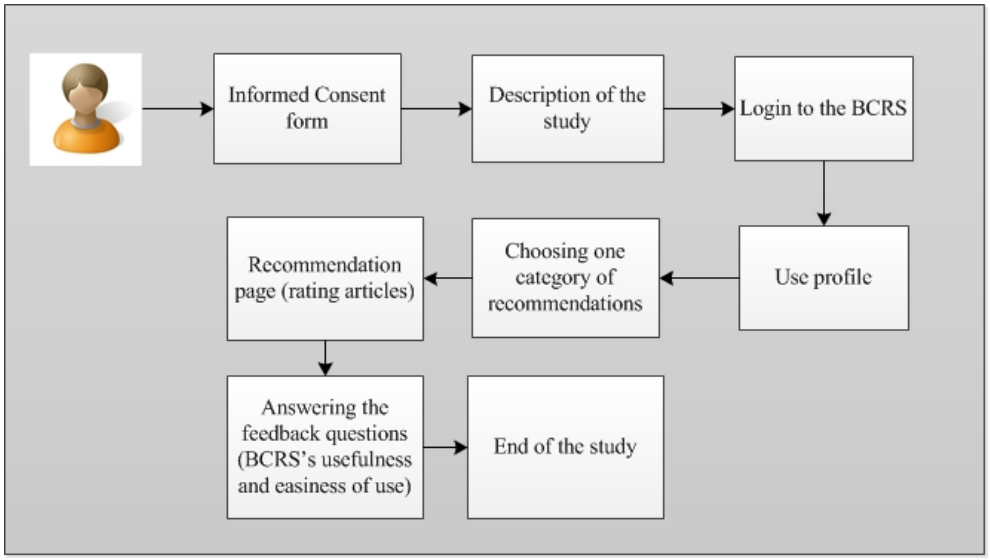

Figure 4.1: The evaluation process.

The purpose of BCRS is to generate top N article recommendations relating to the user's need. But for the purpose of this study, the recommender generator was modified to pick top 4 highly rated articles along with two low rated articles for each participant. The system was designed to recommend articles in the 4 categories of life style, emotional concerns, treatment and risk factors. Three users were assigned to receive the recommendation from each of these categories ($4 \times 3 = 12$ participants). Each participant in each category was provided with article recommendations generated by a different recommendation method: collaborative filtering, content based approach and the proposed hybrid approach. As part of this experiment, we intended

to compare the accuracy of predicted ratings produced by these three approaches. Therefore, we used MAE to calculate the deviation of actual rating from predicted rating produced by these approaches.

The other measure used to analyze data was Pearson correlation which would be useful in determining if the recommender can predict rating along the range of actual ratings. As discussed in [49], correlation metrics are good when the user task is find good items and they compare a non-binary rating generated by the system with the non-binary true rating of users.

4.5 Results and Analysis

The data provided by 12 participants was extracted from database for further analysis. As discussed earlier, MAE and Pearson correlation have been used to assess the raw data. Figure 4.2 shows the MAE for each strategy. As expected the BCRS hybrid algorithm performed better compared with the two other approaches of collaborative filtering and content-based. The lower MAE indicates the better performance of the system in prediction of ratings. The poorest performer in this domain is content-based approach. Due to the challenge of recruiting a sufficient number of participants to perform statistical tests of significance of these differences, these preliminary results can be taken only as consistent with the idea that the proposed BCRS algorithm may perform as well as or better than the alternatives. The findings are encouraging and a full study should be performed to follow up.

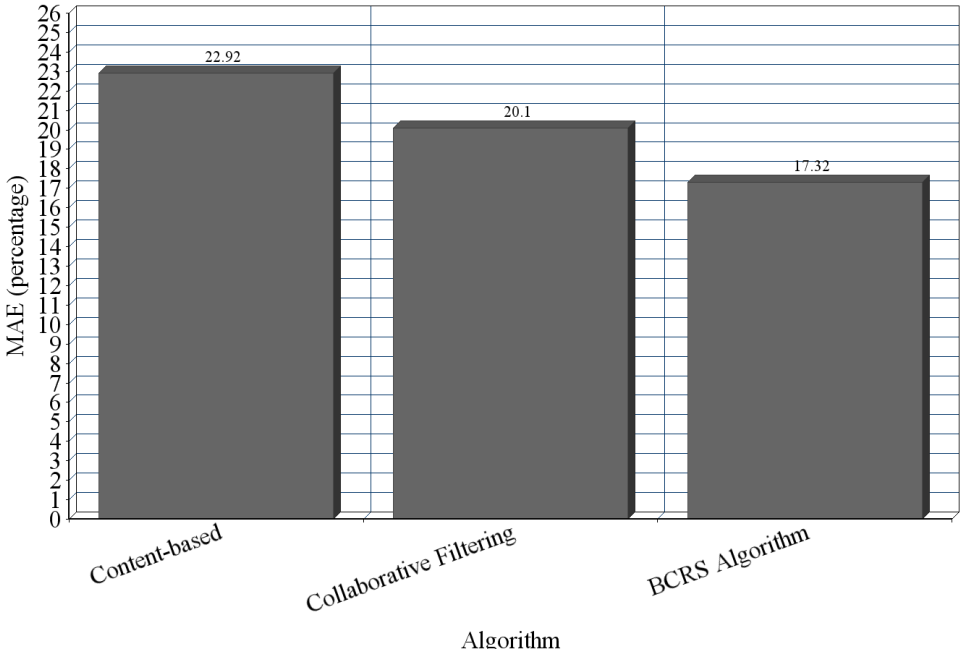

Figure 4.2: MAE for 3 different approaches.

The other measure used to analyze data is Pearson correlation that shows how recommender system is able to predict rating along with the range of true ratings provided by users. Table 4.1 summarizes the results.

Table 4.1: Pearson correlation for 3 different approaches.

Recommendation Techniques	Content-based	Collaborative Filtering	BCRS Hybrid Algorithm
Pearson Correlation Value	0.453	0.43	0.65

As can be seen, the BCRS hybrid approach works better than the other two approaches. There is a better correlation between true rating and predicted rating by the proposed hybrid approach compared with the correlation for collaborative

filtering and content-based techniques.

It was mentioned earlier that the BCRS approach works better than pure collaborative filtering or Content-based methods in dealing with the cold start problem. Although no study has been done to test it, while analyzing the data we realized that the results of recommendations of the three approaches partially validate our claim. The system was developed to recommend six articles for each user. Each of four users had been assigned to be provided by a different approach. As a result each approach should totally generate 24 article recommendations. The BCRS was able to provide 100% coverage by generating 24 articles as recommendations. In the case of collaborative filtering, the algorithm generated 22 articles (instead of 24) with the coverage of 91.6%. The content-based algorithm provided the coverage of 79.16%.

To evaluate the BCRS usefulness and easiness of use, the participants were asked to answer 8 modified questions based on TAM. Figure 4.3 is a snapshot from the feedback page of BCRS website showing the BCRS's technology acceptance questions.

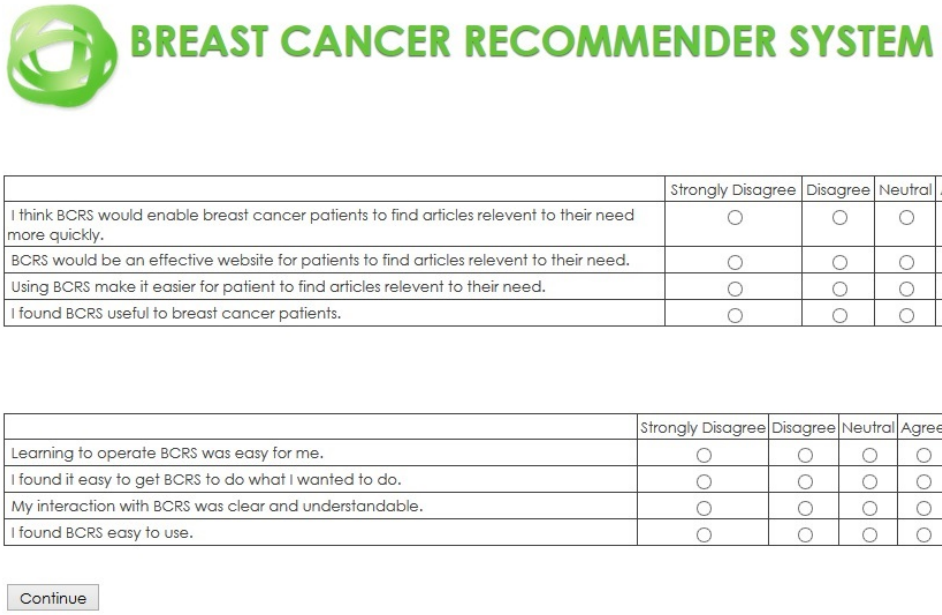

Figure 4.3: BCRS's technology acceptance questions.

The collected data from 12 participants is analyzed using standard deviation (stdv). The results for each of recommendation approaches are shown in table 4.2.

Table 4.2: Standard deviation (stdv) of perceived usefulness (PU) and perceived ease of use (PEOU) for three different approaches.

	Collaborating Filtering	Content-based	BCRS Hybrid Approach
Perceived Usefulness	0.807	0.8	0.708
Perceived Ease of Use	0.714	0.614	0.667

As can be seen the stdv of PU for BCRS is slightly lower in compare to the other two recommendation methods. The results partially show that the recommendations produced by BCRS approach seemed more useful to participants. The stdv is also

calculated for the answers of PEOU questions. However, the PEOU questions are more related to the interface and design of the system and they are not that much related to the recommendation algorithm that works behind the scene. To use the system in real world, it should be tested in future with breast cancer patients to identify the problems, needs and improvements that should be undertaken for this prototype system.

4.6 Summary

In this chapter, the main measures used in prior literature for evaluating recommenders were reviewed. We presented the evaluation scenario in which 12 medical students were participated. Due to privacy concerns and not having access to medical documents, the user profiles information was simulated. The privacy issues arise another difficulty in the system evaluation phase since the sufficient real patients were not available. As a result a preliminary study was undertaken with 12 female medical students.

The results of analyzing the data with MAE and Pearson correlation suggest that the BCRS hybrid approach performs slightly better in this domain compared with the other two approaches. However, much work remains to be done to validate and extend BCRS in practice.

Chapter 5

Summary and Suggested Future Work

5.1 Summary

Recommendation systems in general assist users with the information overload problem. They intend to recommend items that is expected to be useful or interesting for a particular user. Finding right, useful and trustworthy information on web can be problematic, especially for patients. Based on statistics provided by Canadian Cancer Society, breast cancer is the second leading cause of cancer death among Canadian women. So there is a huge potential in this area to help people suffering from breast cancer in finding the appropriate information to meet their information needs. BCRS was developed to address this concern and the system provides article recommendations for patients in 4 categories. The primary contribution of this thesis is the introduction of a novel health domain for application of RS that can provide

the basis for more RS applications and research in health area. The need for RS applications in health is also discussed earlier. A secondary contribution is related to a novel recommender system being developed to supply the patients suffering from breast cancer by providing health article recommendations relating to issues arising from their circumstances or profiles. Specifically, I introduced a hybrid algorithm that switches between two recommendation techniques based on two conditions. The proposed hybrid algorithm switches between pure collaborative approach and hybrid collaborative filtering approach regarding the novelty of the target article for the neighbors of a target user. The other contribution of this work is providing an experimental accuracy comparison of recommendations generated with three different recommending approaches: collaborative filtering, content based approach and the proposed BCRS hybrid approach. We believe the proposed algorithm reduces the problem of cold start, firstly because it is a combination of two other recommendation methods. Moreover, it does not use the similarity rating pattern to find the neighbors. No particular experiment has been done to test this, and it is postponed as part of the future work. However, analyzing the data provided in evaluation part, we realized that our system coverage was 100%, while it was 91.60% and 79.16% for collaborative filtering and content-based approach respectively. These data are consistent with claim that the BCRS approach can reduce the cold start problem.

5.2 Suggested Future Work

There is a strong opportunity for future work particularly relating to the evaluation phase. Although the experiments done by this thesis are useful particularly for

comparing different approaches and testing the accuracy of recommendations, it is essential to test BCRS with real and natural data and real users. The evaluation should be followed up by the real patients and in realistic situation.

The algorithm that was developed in this thesis helps alleviate the cold start problem. Although the results of this preliminary evaluation are only suggestive of a significant effect, further evaluation in a more comprehensive experiment is needed to be able to establish compelling evidence of this result. Another interesting direction for future work is to test the system with some metrics beyond the accuracy measures such as serendipity and novelty of recommendations [52].

Further research is also needed to experiment with the values of threshold in finding neighbors. Using different thresholds might increase the accuracy.

Bibliography

[1] S. K. Lam and J. Riedl, "Shilling Recommender System for Fun and Profit," in *13th International conference on World Wide Web*, pp. 393 – 402, Association for Computing Machinery, 2004.

[2] I. Fernadez-Tobis, I. Cantador, M. Kaminskas, and F. Ricci, "Cross-domain Recommender Systems: A Survey of The State of the Art," in *Proceedings 2nd Spanish Conference on Information Retrieval*, 2012.

[3] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender System: A Survey of the State-of-the-Art and Possible Extentions," *IEEE TRANSACTION ON KNOWLEDGE AND DATA ENGINEERING*, vol. 17, no. 6, pp. 734–749, 2005.

[4] P. Kummervold, C. Chronaki, B. Lausen, H. Prokosch, J. Rasmussen, A. S. S. Santana, and S. Wangberg, "eHealth Trends in Europe 2005-2007: a Population-based Survey," *J Med Internet Res*, 2008.

[5] S.Fox, "Pew Internet and American Life Project - Health Topics Report," *Internet*, 2011.

[6] E. Sillencea, P. Briggs, P. R. Harris, and L. Fishwick, "How do Patients Evaluate and Make Use of Online Health Information?," *Elsevier*, vol. 64, no. 9, pp. 1853–1862, 2007.

[7] M. Lpez-Nores, Y. Blanco-fern, Y. Blanco-Fernndez, J. J. Pazos-Arias, and M. I. Martn-Vicente, *Context-Aware Recommender Systems Influenced by the Users Health-Related Data*. User Modeling and Adaptation for Daily Routines, Springer London, 2013.

[8] M. J. Satterlund, K. D. McCaul, and A. K. Sandgren, "Information Gathering Over Time by Breast Cancer Patients?," *Journal of Medical Internet Research*, 2003.

[9] J. L. Herlocker and J. A. Konstan, *Understanding and Improving Automated Collaborative Filtering Systems*. PhD thesis, University of Minnesota, 2000.

[10] J. L. H. andJ. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[11] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *springer-Link, User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[12] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutirrez, "Recommender Systems Survey," *Elsevier*, vol. 46, p. 109132, 2013.

[13] A. Walker, M. M. Recker, K. Lawless, and D. Wiley, "Item-Based Collaborative Filtering Recommendation Algorithms," *10th international conference on World Wide Web, ACM*, vol. 14, no. 1, pp. 3–28, 2004.

[14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," in *10th international conference on World Wide Web*, Association for Computing Machinery, 2001.

[15] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," *Adaptive Web, Springer*, pp. 325–341, 2007.

[16] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*, pp. 1–35. Springer US, 2011.

[17] R. Burke, "Knowledge-Based Recommender Systems," *Encyclopedia of Library and Information Systems*, vol. 69, 2000.

[18] S. Abbar, M. Bouzeghoub, and S. Lopez, "Context-Aware Recommendation Systems: a Service-Oriented Approach," in *35th International Conference on Very Large Data Bases (VLDB)*, 2009.

[19] A. Das, M. Datar, A. Garg, and S. Rajaram, "Google News Personalization: Scalable Online Collaborative Filtering," in *16th international conference on World Wide Web*, Association for Computing Machinery, 2007.

[20] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. M. Sarwar, J. L. Herlocker, and J. Riedl, "Combining Collaborative Filtering with Personal Agents for Better Recommendations," in *the sixteenth national conference on Artificial*

intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence, 1999.

[21] P. Lops, M. de Gemmis, and G. Semeraro, *Content-based Recommender Systems: State of the Art and Trends*. Recommender Systems Handbook, Springer US, 2011.

[22] W. F. Velicer, J. O. Prochaska, and J. M. Bellis, "An Expert System Intervention for Smoking Cessation," *Elsevier*, vol. 18, no. 3, pp. 269–290, 1993.

[23] L. Fernndez-Luque, R. Karlsen, and L. Vognild, "Challenges and Opportunities of Using Recommender Systems for Personalized Health Education," in *the 22nd international congress of the European Federation for Medical Informatics*, 2009.

[24] S. Lee, H. Y. Byung, H. Choe, B. Y. Park, R. W. Park, P. Park, H. J. Hwang, B. M. Lee, Y. H. Lee, and U. G. Kang, "Lifestyle Recommendation System using Framingham Heart Study Based Clinical Decision Support System (CDSS)," in *World Congress on Medical Physics and Biomedical Engineering*, pp. 4016–4019, Springer Berlin, 2006.

[25] S. Hammer, J. Kim, and E. Andre, "MED-StyleR: METABO Diabetes-Lifestyle Recommender," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010.

[26] J. H. Kim, J. Lee, J. Park, Y. H. Lee, and K. W. Rim, "Design of Diet Recommendation System for Health care Service Based on User Information," in *fourth international conference on computer science and coverage information technology*, 2009.

[27] J. Freyne and S. Berkovsky, "Intelligent Food Planning: Personalized Recipe Recommendation," in *the 15th international conference on intelligent user interfaces (IUI)*, 2010.

[28] C. Lee, M. Lee, D. Han, S. Jung, and J. Cho, "A Framework for Personalized Healthcare Service Recommendation," in *10th International Conference on e-health Networking, Applications and Services*, 2008.

[29] T. R. Hoens, M. Blanton, A. Steele, and N. V. Chawla, "Reliable medical recommendation systems with patient privacy," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 4, 2013.

[30] C. Yang, Y. K. Chang, C. P. Chang, and C. P. Chu, "A Personalized Service Recommendation System in a Home-care Environment," in *the 15th International Conference on Distributes Multimedia Systems (DMS)*, 2009.

[31] I. Song, D. Dillon, T. J. Goh, and M. Sung, "A Health Social Network Recommender System," in *the 14th International Conference, PRIMA*, pp. 361–372, Springer Berlin Heidelberg, 2011.

[32] P. Pattaraintakorn, G. M. Zaverucha, and N. Cercone, "Web Based Health Recommender System Using Rough Sets, Survival Analysis and Rule-Based Expert Systems," in *11th International Conference, RSFDGrC*, vol. 4482, pp. 491–499, Springer Berlin Heidelberg, 2007.

[33] M. Wiesner and D. Pfeifer, "Adapting Recommender Systems to the Requirements of Personal Health Record Systems," in *the 1st ACM International Health Informatics Symposium*, pp. 410–414, Springer Berlin Heidelberg, 2010.

[34] T. G. Morrell and L. Kerschberg, "Personal Health Explorer: A Semantic Health Recommendation System," in *28th International Conference on Data Engineering Workshops (ICDEW)*, pp. 55 – 59, IEEE, 2012.

[35] A. Rivero-Rodriguez, S. T. Konstantinidis, C. L. Sanchez-Bocanegra, and L. Fernandez-Luque, "A health information recommender system: Enriching YouTube health videos with Medline Plus information by the use of SnomedCT terms," in *26th International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 257 – 261, IEEE, 2013.

[36] R. C. Chen, Y. H. Huang, C. T. Bau, and S. M. Chen, "A Recommendation System Based on Domain Ontology and SWRL for Anti-Diabetic Drugs Selection," *Elsevier*, vol. 39, no. 4, pp. 3995–4006, 2012.

[37] H. Y. Kung, T. M. P. Nguyen, T. H. Kuo, C. P. Tsai, and C. H. Chen, "Intelligent Personalized Food Recommendation System Based on a Semantic Sensor Web," in *2011 International Conference in Electrics, Communication and Automatic Control Proceedings*, pp. 61–68, Springer, 2012.

[38] A. Torkaman, N. Charkari, M. Aghaeipour, and E. Hajati, "A Recommender System for Detection of Leukemia Based on Cooperative Game," in *17th Mediterranean Conference on Control and Automation, MED*, pp. 1126 – 1130, IEEE, 2009.

[39] H. Witteman, M. Chignell, and M. Krahn, "A Recommender System for Prostate Cancer Websites," in *AAMIA Annual Symposium*, p. 1177, 2008.

[40] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H. Kriegel, "Probabilistic Memory-Based Collaborative Filtering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 1, pp. 56–69, 2004.

[41] T. F. Hack, L. F. Degner, and D. G. Dyck, "Relationship Between Preferences for Decisional Control and Illness Information Among Women with Breast Cancer: A Quantitative and Qualitative Analysis," *Elsevier*, vol. 39, no. 2, pp. 279–289, 1994.

[42] R. S. Fisher, A. Girgis, A. Boyes, B. Bonevski, L. Burton, and P. Cook, "The unmet supportive care needs of patients with cancer," *Annual Report to the Nation on Status of Cancer*, vol. 88, no. 1, p. 226237, 2000.

[43] B. Thewes, B. Meiser, A. Taylor, K. Phillips, S. Pendlebury, A. Capp, D. Dalley, D. Goldstein, R. Baber, and M. Friedlander, "Fertility- and Menopause-Related Information Needs of Younger Women With a Diagnosis of Early Breast Cancer," *Journal of Clinical Ontology*, vol. 23, no. 22, 2005.

[44] J. K. J. Herlocker and J. Riedl, "An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms," *Information Retrieval*, vol. 5, no. 4, pp. 287–310, 2011.

[45] J. A. K. M. D. Ekstrand, J. T. Riedl, "Collaborative Filtering Recommender Systems," in *Foundations and Trends in Human-Computer Interaction*, vol. 4, pp. 81–173, 2011.

[46] D. L. Lee, C. Huei, and K. Seamson, "Document Ranking and the Vector-Space Model," *Software, IEEE*, vol. 14, no. 2, pp. 67–75, 2011.

[47] G. Forman, "BNS Feature Scaling: an Improved Representation Over TF-IDF for SVM Text Classification," in *the 17th ACM conference on Information and knowledge management*, pp. 263–270, 2008.

[48] W. Zhang, T. Yoshida, and X. Tangc, "A Comparative Study of TF*IDF, LSI and Multi-Words for Text Classification," *Elsevier*, vol. 38, no. 3, p. 27582765, 2011.

[49] A. Gunawardana and G. Shani, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," *The Journal of Machine Learning Research*, vol. 10, no. 12, pp. 2935–2962, 2009.

[50] G. Shani and A. Gunawardana, *Evaluating Recommendation Systems*. Recommender Systems Handbook, Springer US, 2011.

[51] V. Venkatesh and F. Davis, "A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies," *Management Science*, vol. 46, no. 2, pp. 186–204, 2000.

[52] O. Celma and P. Herrera, "A New Approach to Evaluating Novel Recommendations," in *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, (New York, NY, USA), pp. 179–186, ACM, 2008.

Appendix A

BCRS Code

Computation of document-document similarity has been coded as follow:

```
/**** Removing stop words of a document ****/
$stop_words=array('&rsquo', ' ', '&lsquo', '&sbquo',
                  '&ldquo', '&rdquo', 'nbsp','-','--' ,'a\'s', 'able',
                  'about', 'above', 'according', 'accordingly',
                  'across', 'actually', 'after', 'afterwards',
                  'again', 'against', 'ain\'t', 'all', 'allow', 'allows',
                  'almost', 'alone', 'along', 'already', 'also',
                  'although', 'always', 'am', 'among', 'amongst',
                  'an', 'and', 'another', 'any', 'anybody',
                  'anyhow', 'anyone', 'anything', 'anyway',
                  'anyways', 'anywhere', 'apart', 'appear',
                  'appreciate', 'appropriate', 'are', 'aren\'t',
                  'around', 'as', 'aside', 'ask', 'asking',
```

```
'associated', 'at', 'available', 'away',
'awfully', 'be', 'became', 'because',
'become', 'becomes', 'becoming',
'been', 'before', 'beforehand', 'behind',
'being', 'believe', 'below', 'beside',
'besides', 'best', 'better', 'between',
'beyond', 'both', 'brief', 'but', 'by',
'c\'mon', 'c\'s', 'came', 'can', 'can\'t',
'cannot', 'cant', 'cause', 'causes',
'certain', 'certainly', 'changes', 'clearly',
'co', 'com', 'come', 'comes', 'concerning',
'consequently', 'consider', 'considering',
'contain', 'containing', 'contains', 'corresponding',
'could', 'couldn\'t', 'course', 'currently',
'definitely', 'described', 'despite', 'did',
'didn\'t', 'different', 'do', 'does', 'doesn\'t',
'doing', 'don\'t','don&rsquot', 'done',
'down', 'downwards', 'during', 'each',
'edu', 'eg', 'eight', 'either', 'else',
'elsewhere', 'enough', 'entirely',
'especially', 'et', 'etc', 'even', 'ever',
'every', 'everybody', 'everyone',
'everything', 'everywhere', 'ex',
'exactly', 'example', 'except', 'far',
```

```
'few', 'fifth', 'first', 'five', 'followed',
'following', 'follows', 'for', 'former',
'formerly', 'forth', 'four', 'from', 'further',
'furthermore', 'get', 'gets', 'getting',
'given', 'gives', 'go', 'goes', 'going',
'gone', 'got', 'gotten', 'greetings',
'had', 'hadn\'t', 'happens', 'hardly',
'has', 'hasn\'t', 'have', 'haven\'t', 'having',
'he', 'he\'s', 'hello', 'help', 'hence', 'her',
'here', 'here\'s', 'hereafter', 'hereby',
'herein', 'hereupon', 'hers', 'herself', 'hi',
'him', 'himself', 'his', 'hither', 'hopefully',
'how', 'howbeit', 'however', 'i\'d', 'i\'ll',
'i\'m', 'i\'ve', 'ie', 'if', 'ignored', 'immediate',
'in', 'inasmuch', 'inc', 'indeed', 'indicate',
'indicated', 'indicates', 'inner', 'insofar',
'instead', 'into', 'inward', 'is', 'isn\'t', 'it',
'it\'d', 'it\'ll', 'it\'s', 'its', 'itself', 'just',
'keep', 'keeps', 'kept', 'know', 'knows',
'known', 'last', 'lately', 'later', 'latter',
'latterly', 'least', 'less', 'lest', 'let',
'let\'s', 'like', 'liked', 'likely', 'little',
'look', 'looking', 'looks', 'ltd', 'mainly',
'many', 'may', 'maybe', 'me', 'mean',
```

```
'meanwhile', 'merely', 'might', 'more',
'moreover', 'most', 'mostly', 'much',
'must', 'my', 'myself', 'name', 'namely',
'nd', 'near', 'nearly', 'necessary', 'need',
'needs', 'neither', 'never', 'nevertheless',
'new', 'next', 'nine', 'no', 'nobody', 'non',
'none', 'noone', 'nor', 'normally', 'not',
'nothing', 'novel', 'now', 'nowhere', 'obviously',
'of', 'off', 'often', 'oh', 'ok', 'okay', 'old',
'on', 'once', 'one', 'ones', 'only', 'onto',
'or', 'other', 'others', 'otherwise', 'ought',
'our', 'ours', 'ourselves', 'out', 'outside',
'over', 'overall', 'own', 'particular', 'particularly',
'per', 'perhaps', 'placed', 'please', 'plus',
'possible', 'presumably', 'probably', 'provides',
'que', 'quite', 'qv', 'rather', 'rd', 're', 'really',
'reasonably', 'regarding', 'regardless',
'regards', 'relatively', 'respectively', 'right',
's', 'said', 'same', 'saw', 'say', 'saying', 'says',
'second', 'secondly', 'see', 'seeing', 'seem',
'seemed', 'seeming', 'seems', 'seen', 'self',
'selves', 'sensible', 'sent', 'serious', 'seriously',
'seven', 'several', 'shall', 'she', 'should',
'shouldn\'t', 'since', 'six', 'so', 'some',
```

```
'somebody', 'somehow', 'someone', 'something',
'sometime', 'sometimes', 'somewhat',
'somewhere', 'soon', 'sorry', 'specified',
'specify', 'specifying', 'still', 'sub', 'such', 'sup',
'sure', 't', 't\'s', 'take', 'taken', 'tell', 'tends',
'th', 'than', 'thank', 'thanks', 'thanx', 'that',
'that\'s', 'thats', 'the', 'their', 'theirs', 'them',
'themselves', 'then', 'thence', 'there', 'there\'s',
'thereafter', 'thereby', 'therefore', 'therein',
'theres', 'thereupon', 'these', 'they', 'they\'d',
'they\'ll', 'they\'re', 'they\'ve', 'think', 'third',
'this', 'thorough', 'thoroughly', 'those', 'though',
'three', 'through', 'throughout', 'thru', 'thus',
'to', 'together', 'too', 'took', 'toward', 'towards',
'tried', 'tries', 'truly', 'try', 'trying', 'twice', 'two',
'un', 'under', 'unfortunately', 'unless', 'unlikely',
'until', 'unto', 'up', 'upon', 'us', 'use', 'used',
'useful', 'uses', 'using', 'usually', 'value', 'various',
'very', 'via', 'viz', 'vs', 'want', 'wants', 'was',
'wasn\'t', 'way', 'we', 'we\'d', 'we\'ll', 'we\'re',
'we\'ve', 'welcome', 'well', 'went', 'were', 'weren\'t',
'what', 'what\'s', 'whatever', 'when', 'whence',
'whenever', 'where', 'where\'s', 'whereafter',
'whereas', 'whereby', 'wherein', 'whereupon',
```

```
                    'wherever', 'whether', 'which', 'while', 'whither',
                    'who', 'who\'s', 'whoever', 'whole', 'whom',
                    'whose', 'why', 'will', 'willing', 'wish', 'with',
                    'within', 'without', 'won\'t', 'wonder', 'would',
                    'would', 'wouldn\'t', 'yes', 'yet', 'you', 'you\'d',
                    'you\'ll', 'you\'re', 'you\'ve', 'your', 'yours',
                    'yourself', 'yourselves', 'zero', 'a');
$i=0;
$query="SELECT * from `article` WHERE article_id NOT IN
        (SELECT document_ID FROM `term_frequency`)";
$result=$mysqli->query($query);
while($row= $result->fetch_array(MYSQLI_ASSOC)){
    $D_id=$row['article_id'];
    $STRING=$row['text'];
    $STRING=strip_tags($STRING);
    $STRING=addslashes($STRING);
    term_number($D_id, $STRING, $stop_words);
}

function term_number($doc_id ,$str, $stop_words){
    $str=strtolower($str);
    $str=preg_replace('/(^|\b|\s)('.implode('|', $stop_words).')
         (\b|\s|$)/i', '', $str);
    $result1=array_count_values(str_word_count($str, 1));
```

```
    foreach ($result1 as $key => $value) {
        $query="INSERT INTO term_frequency VALUES ($doc_id, '$key', $value)";
        $mysqli->query($query);
        $terms[$i]=$value;
        $ID[$i]=$doc_id;
        $frequency[$i]=$key;
        $i++;
    }
}

function TFIDF($document_ID, $stop_word, $TYPE, $General_type){
/************/
/******** Select document numbers which are
           in the $TYPE or $sub_TYPE category *********/
/************/
    if ($General_type==0){
        $q11="SELECT count(article_id) FROM article WHERE type=$TYPE";
    }elseif($General_type==1){
        $q11="SELECT count(article_id) FROM article WHERE sub_type=$TYPE";
    }
    $res=$mysqli->query($q11);
    $roww= $res->fetch_array(MYSQLI_NUM);
    $whole_document_number=$roww[0];
    $q1="SELECT term, frequency FROM `term_frequency` WHERE document_ID=$document_
```

```
    $resultt=$mysqli->query($q1);
    $counter=0;
    $j=0;
    while ($row = mysqli_fetch_array($resultt, MYSQLI_ASSOC)){
        $term=$row['term'];
        $TF1=$row['frequency'];
        $TF=$row['frequency'];
/************/
/******* Select the number of documents who has
         the exact term as the selected documents *******/
/******* Check whether the article is belong to
         which type of recommendation *******/
/************/
        if ($General_type==0){
            $query="SELECT count(frequency) FROM `term_frequency` WHERE
            term='$term' and document_ID IN
            (SELECT article_id FROM article WHERE type=$TYPE)";
        }elseif($General_type==1){
            $query="SELECT count(frequency) FROM `term_frequency` WHERE
            term='$term' and document_ID IN
            (SELECT article_id FROM article WHERE sub_type=$TYPE)";
        }
        $res1=$mysqli->query($query);
        $nk_arr1= $res1->fetch_array(MYSQLI_NUM);
```

```
        $counter++;
        $nk1=$nk_arr1[0];
        $nk=$nk_arr1[0];
        $TFIDF_others=$TF1*log($whole_document_number/$nk1);
        $TFIDF_TEMP=$TFIDF_others^2;
        $SIGMA_TFIDF=$TFIDF_TEMP+$SIGMA_TFIDF;
        $j++;
/************/
/******** Select the number of documents who has
         the exact term as the selected documents *******/
/************/
        if ($General_type==0){
            $query="SELECT count(frequency) FROM ‘term_frequency‘ WHERE
            term=’$term’ and document_ID IN
            (SELECT article_id FROM article WHERE type=$TYPE)";
        }elseif($General_type==1){
            $query="SELECT count(frequency) FROM ‘term_frequency‘ where
            term=’$term’ and document_ID in
            (SELECT article_id FROM article WHERE sub_type=$TYPE)";
        }
        $TFIDF=$TF*log($whole_document_number/$nk);
        $W_kjj=$TFIDF/sqrt($SIGMA_TFIDF);
        $W_kj[$j]=$W_kjj;
        $TERM[$j]=$term;
```

```
    }
    return array($W_kj,$TERM);
}
/**********SIMILARITY FUNCTION**************/
function SIM($doc_1,$doc_2,$type,$sub_type){
    //sub_type=5 means that the article is in a general category;
    if ($sub_type=5){
        $General_type=0; // means that the article is in a general category
    }else{
        $General_type=1; // means that the article IS NOT in a general category
    }
    $up=0;
    $Sigma_Power_j=0;
    floatval($Sigma_Power_j);
    $Sigma_Power_i=0;
    floatval($Sigma_Power_i);
    if ($General_type==0){
        $TYPE=$type;
    }elseif($General_type==1){
        $TYPE=$sub_type;
    }
    $mainString1=TFIDF($doc_1, $stop_word, $TYPE, $General_type);
    $Wkj=$mainString1[0];
    $termkj=$mainString1[1];
```

```
$mainString=TFIDF($doc_2, $stop_word, $TYPE, $General_type);
$Wki=$mainString[0];
$termki=$mainString[1];
for ($z=0; $z<=count($termkj); $z++){
    $first_document_word=$termkj[$z];
    if (array_search($first_document_word, $termki)){
        $key=array_search($first_document_word, $termki);
        $Wki_temp=$Wki[$key];
        $Wkj_temp=$Wkj[$z];
        $up_temp=$Wkj_temp*$Wki_temp;
        $power_j=$Wkj[$z];
        $power_j=pow($power_j,2);
        $power_i=$Wki[$key];
        $power_i=pow($power_i,2);
        $Sigma_Power_j=$Sigma_Power_j+$power_j;
        $Sigma_Power_i=$Sigma_Power_i+$power_i;
        $up=$up_temp+$up;  // Sigma of the upper of the equation
    }
}
$down=sqrt($Sigma_Power_j)*sqrt($Sigma_Power_i);
$up=round($up, 2);
$down=round($down, 2);
$similarity=$up/$down;
$similarity=round($similarity,2);
```

```
    return($similarity);
}
```

Here is the BCRS' hybrid recommendation code:

```
require_once('../user.php'); // User similarity function page is called
$similar_users=similarity($_SESSION['ID'],$_GET['Type']);
// Calling the function of similarity which is to detect the neighbors
$ID=$_SESSION['ID']; // $ID stores the user_id who wants the recommendation
$TYPE=$_GET['Type']; // $TYPE stores the category of recommendation
                        that user is looking for

/******** Determine a new suggestion at each time user login **********/
$query="SELECT MAX(`set_id`) FROM `suggestions` WHERE `user_id`=".$ID."";
$res = $mysqli->query($query);
$row = $res->fetch_array(MYSQLI_NUM);
$set_id=$row[0];
if ($set_id==''){
$set_id=1;
}else{
$set_id++;
}
/******** Determine a new suggestion at each time user login **********/
/************************/
```

```
if ($TYPE==3){ // Lifestyle recommendation
    $query="SELECT * FROM user_lifestyle WHERE userid=$ID";
    $result=$mysqli->query($query);
    $row = $result->fetch_array(MYSQLI_ASSOC);
    $smok=$row['smoking'];
    $alcohol=$row['alcohol'];
    $activity=$row['activity'];
    $query="SELECT * FROM user WHERE userid=$ID";
    $result=$mysqli->query($query);
    $row = $result->fetch_array(MYSQLI_ASSOC);
    $weight=$row['weight'];
    $height=$row['height'];
    $BMI=$weight/($height/100)^2;  // BMI Calculation
    $i=0;   // The number of problems user may have (initialized!)
    if ($smok=="yes"){
        $sub_type[$i]=1;
        $i++;
    }
    if ($alcohol=="yes"){
        $sub_type[$i]=2;
        $i++;
    }
    if ($activity=="no"){
```

```
    $sub_type[$i]=3;
    $i++;
    }
    if ($BMI<18.5 || $BMI>25){
        $sub_type[$i]=4;
        $i++;
    }
}elseif($TYPE==2){  // Treatment recommendation
    $query="SELECT * FROM user_treatment WHERE userid=$ID";
    $result=$mysqli->query($query);
    $row = $result->fetch_array(MYSQLI_ASSOC);
    $cancer_type=$row['cancer_type'];
    $cancer_stage=$row['cancer_stage'];
    $used_treatment=$row['used_treatment'];
    $i=0; // The number of problems user may have (initialized!)
    if ($cancer_type!=""){
        $sub_type[$i]=1;
        $i++;
    }
    if ($cancer_stage!=""){
        $sub_type[$i]=2;
        $i++;
    }
    if ($used_treatment!=""){
```

```
        $sub_type[$i]=3;
        $i++;
    }
}elseif($TYPE==4){ // Emotional Concerns recommendation
    $query="SELECT * from user_concerns where userid=$ID";
    $result=$mysqli->query($query);
    $row = $result->fetch_array(MYSQLI_ASSOC);
    $relation=$row['relation'];
    $support_child=$row['support_child'];
    $have_child=$row['have_child'];
    $i=0;  // The number of problems user may have (initialized!)
    if ($relation=="yes"){
        $sub_type[$i]=1;
        $i++;
    }
    if ($support_child=="yes"){
        $sub_type[$i]=2;
        $i++;
    }
    if ($have_child=="yes"){
        $sub_type[$i]=3;
        $i++;
    }
}
```

```
if (empty($sub_type)){  // Recommendation in general category
    $query="SELECT * FROM article WHERE
    type=$TYPE AND sub_type!=1 AND sub_type!=2
    AND sub_type!=3 AND sub_type!=4 AND article_id NOT IN
    (SELECT articleid FROM rank WHERE userid=$ID)";
    // Select related articles in the category type that are new
    and user did not ranked before. sub_type!=1,2,3,4 means
    that the article is not in sub-categories
    recommendation($query, $TYPE, 5, $ID, $similar_users, $limit, $set_id);
    // Recommendation function is called with some elements
}else{
    // The type is in Risk Factors, Life style,
    Emotional Concerns or Treatments.
    for ($k=1;$k<=$i;$k++){
        if ($i==1){
            $limit=6;
        }elseif($i==2){
            $limit=3;
        }elseif($i==3){
            $limit=2;
        }elseif($i>3){
            $limit=1;
        }
        $subType=$sub_type[$k-1];
```

```
        $query="SELECT * FROM article WHERE
        type=$TYPE AND sub_type=$subType AND article_id NOT IN
        (SELECT articleid FROM rank WHERE userid=$ID)";
        // Select related articles in the category type that
        are new and user did not ranked before.
        recommendation($query, $TYPE, $subType, $ID, $similar_users,
        $limit, $set_id);
        // Recommendation function is called with some elements
    }
}

/******************* Recommendation Function *********************/
function recommendation($query, $TYPE, $subType, $ID, $similar_users,
$limit, $set_id){
    $result=mysqli_query($mysqli, $query);
    while($row = mysqli_fetch_array($result, MYSQLI_ASSOC)){
        $arr[]=array('article_id'=>$row['article_id'], 'article_text'=>
        $row['text'], 'article_title'=>$row['title']);
    }
    extract ($arr);
    require_once('../doc.php'); // Document similarity function page is called
    foreach ($arr as $key=>$one){ // Going over all articles one by one
        $article_id=$one['article_id'];
        //  Article that user did not ranked before
```

```
$COUNT_OTHERS_WHO_RANKED=0; // Initialize the variable
$RANK=0; // Initialize the variable
$Top=0;
$Down=0;
foreach($similar_users as $key=>$value){
    // Go over all similar users one by one to check
    whether they have read the article before or not!
    $USER_OTHER = $key;
    // Look for other neighbors ranking to check whether
    they have ranked the article or not
    if ($TYPE==1 || $TYPE==2){
        if ($value>=0.75){
        // Threshold of similarity for users in Type 1 and 2
            $query1="SELECT rank FROM rank WHERE
            articleid=$article_id AND userid=$USER_OTHER";
            $result1 = mysqli_query($mysqli, $query1);
            if ($row1 = mysqli_fetch_array($result1, MYSQLI_ASSOC)){
                $RANK+=$row1['rank']; // The summation of all rankings
                $COUNT_OTHERS_WHO_RANKED++;
                // The number of people who ranked
            }
        }
    }elseif($TYPE==3 || $TYPE==4){
        if ($value>=0.60){
```

```
            // Threshold of similarity for users in Type 3 and 4
            $query1="SELECT rank FROM rank WHERE
            articleid=$article_id AND userid=$USER_OTHER";
            $result1 = mysqli_query($mysqli, $query1);
            if ($row1 = mysqli_fetch_array($result1, MYSQLI_ASSOC)){
                $RANK+=$row1['rank']; // The summation of all rankings
                $COUNT_OTHERS_WHO_RANKED++;
                // The number of people who ranked
            }
        }
    }
}

if ($RANK!=0){ // If the article has been ranked before  --- GOOD RANKS
    $AVG_RANK=$RANK/$COUNT_OTHERS_WHO_RANKED;
    $AVERAGE_RANKING[$i]=$AVG_RANK;
    $query1="INSERT INTO `mt0454_bcsr`.`suggestions`
    (`user_id`,`article_id` ,`article_type` ,`article_sub_type`
    ,`avg_rank` ,`partition` ,`check`, `set_id`) VALUES
    (".$ID.", ".$article_id.",'".$TYPE."','".$subType."', ".$AVG_RANK.",
    0, 0, ".$set_id.")";
    // partition=0 means that the article is belong to first part
    $result1 = $mysqli->query($query1);
    $ARTICLE_IDS[$i]=$article_id;
```

```
    $AVGR=$RANK/$COUNT_OTHERS_WHO_RANKED;
    $ranking_others[$i]=array($article_id=>$AVGR);
    $RANKING[$i]=$AVGR;
}else{   // The article hasn't seen before.
    foreach($similar_users as $key=>$value){
        // go over all similar users one by one to check whether
        they have read the article before or not!
        $USER_OTHER = $key;// select similar users one by one
        // Go over all similar users one by one to check whether
        they have read the article before or not!
        if ($TYPE==1 || $TYPE==2){
            if ($value>=0.75){
                $query2="SELECT rank, articleid FROM `rank` WHERE
                 userid=$USER_OTHER AND articleid IN
                 (SELECT article_id FROM `article` WHERE
                 type=$TYPE and sub_type=$subType)";
                 // Select neighbors' articles based on the TYPE
                 of the recommendation
                 $result4 = mysqli_query($mysqli, $query2);
                 while
                 ($row2 = mysqli_fetch_array($result4, MYSQLI_ASSOC)){
                     $rank=$row2['rank'];
                     $A_id=$row2['articleid'];
                     $start=microtime(true);
```

```
                $similarity_articles=
                SIM($article_id,$A_id,$TYPE,$subType);
                // Check the similarity of the article with
                other articles in that category--onebyone
                $Top+=$rank*$similarity_articles;
                $Down+=$similarity_articles;
            }
        }
    }elseif($TYPE==3 || $TYPE==4){
        if ($value>=0.60){
            $query2="SELECT rank, articleid FROM ‘rank‘ WHERE
            userid=$USER_OTHER AND articleid IN
            (SELECT article_id FROM ‘article‘ WHERE
            type=$TYPE and sub_type=$subType)";
            // Select neighbors’ articles based on the TYPE
            of the recommendation
            $result4 = mysqli_query($mysqli, $query2);
            while
            ($row2 = mysqli_fetch_array($result4, MYSQLI_ASSOC)){
                $rank=$row2[’rank’];
                $A_id=$row2[’articleid’];
                $similarity_articles=
                SIM($article_id,$A_id,$TYPE,$subType);
                // Check the similarity of the article with
```

```
                        other articles in that category--onebyone
                        $Top+=$rank*$similarity_articles;
                        $Down+=$similarity_articles;
                    }
                }
            }
        }
        $ESTIMATE=$Top/$Down;
        $query1="INSERT INTO `mt0454_bcsr`.`suggestions`
        (`user_id` ,`article_id` ,`article_type` ,`article_sub_type`,
        `avg_rank` ,`partition` ,`check`, `set_id`) VALUES
        (".$ID.", ".$article_id.",'".$TYPE."','".$subType."', ".$ESTIMATE.",
        1, 0, ".$set_id.")";
        $result2 = mysqli_query($mysqli, $query1);
        $ranking_others[$i]=array($article_id=>$ESTIMATE);
        $RANKING[$i]=$ESTIMATE;
    }
    $i++;
}
/*********Sorting Articles according to its ratings achieved*********/
krsort($ranking_others);   // sort ranking from hiest to lowest
rsort($RANKING);
$value= current($ranking_others[0]);
$x=0;
```

```
    $counter=0;
    while($RANKING[$x]==true){
        $counter++;
        if ($counter>5){
            break 1;
        }
        $p=0;
        while ($ranking_others[$p]==true){
           $value= current($ranking_others[$p]);
           if ($value==$RANKING[$x]){
               $ARTICLES[$x]=key($ranking_others[$p]);
               if ($p==0){
                   array_shift($ranking_others);
                }else{
                    unset($ranking_others[$p]);
                }
                $x++;
                break 1;
            }
            $p++;
        }
    }
}
```

```
/********Sorting Articles according to its ratings achieved*********/
$limit=6;
$partition=0;
$NUM=1;
$queryyyy="SELECT count(distinct(‘article_sub_type‘)) FROM ‘suggestions‘
WHERE  user_id=$ID AND set_id=$set_id";
$resssss=$mysqli->query($queryyyy);
$rowwww = $resssss->fetch_array(MYSQLI_NUM);
$count=$rowwww[0];
echo "<div align=’center’ style=’padding-right:200px ;
padding-left:200px ;’>";
echo "<table align=’center’ cellpadding=’100px;’ cellspacing=’100px;’
style=’padding-left:100px ;’>";
for ($m=0;$m<$limit;$m++){
    if ($limit==0){
        break 1;
    }
    if ($m==2 || $m==3){
        $partition=1;
    }else{
        $partition=0;
    }
    if ($count==1){ // User has one problems in her profile
        if ($m==0){
```

```
            $query="SELECT `article_id` FROM `suggestions` WHERE
            `check`=0 AND `partition`=0 AND user_id=$ID AND
            set_id=$set_id ORDER BY `avg_rank` DESC LIMIT 1";
        }elseif($m<4){
            $query="SELECT `article_id` FROM `suggestions` WHERE
            `check` =0 AND  user_id=$ID AND `partition`=$partition
            AND set_id=$set_id ORDER BY `avg_rank` DESC LIMIT 1 ";
        }else{
            $query="SELECT `article_id` FROM `suggestions` WHERE
            `check` =0 AND  user_id=$ID AND set_id=$set_id
            ORDER BY `avg_rank` ASC LIMIT 1 ";
        }
    }elseif ($count==2){
        // User has two problems in her profile
        if ($m==0){
            $query="SELECT `article_id` FROM `suggestions` WHERE
            `check`=0 AND `partition`=0 AND user_id=$ID AND
            set_id=$set_id ORDER BY `avg_rank` DESC LIMIT 1";
        }elseif($m<4){
            $query="SELECT `article_id` FROM `suggestions` WHERE
            `check` =0 AND  user_id=$ID AND `partition` =$partition
            AND set_id=$set_id AND `article_sub_type` NOT IN
            (SELECT `article_sub_type` FROM suggestions WHERE
            `article_id` =$article_id AND set_id=$set_id) ORDER BY
```

```
        ‘avg_rank‘ DESC LIMIT 1 ";
    }else{
        $query="SELECT ‘article_id‘ FROM ‘suggestions‘ WHERE
        ‘check‘ =0 AND  user_id=$ID AND set_id=$set_id  ORDER BY
        ‘avg_rank‘ ASC LIMIT 1 ";
    }
}elseif($count=3){
    // User has three problems in her profile
    if ($m==0){
        $query=" SELECT ‘article_id‘ FROM ‘suggestions‘
        WHERE ‘check‘=0 AND ‘partition‘=0 AND
        user_id=$ID AND set_id=$set_id ORDER BY
        ‘avg_rank‘ DESC LIMIT 1";
    }elseif($m<3){
        $query="SELECT ‘article_id‘ FROM ‘suggestions‘ WHERE
        ‘check‘=0 AND  user_id=$ID AND ‘partition‘ =$partition
        AND set_id=$set_id AND ‘article_sub_type‘ NOT IN
        (SELECT ‘article_sub_type‘ FROM suggestions WHERE
        user_id=$ID AND ‘check‘=1 AND set_id=$set_id) ORDER BY
        ‘avg_rank‘ DESC LIMIT 1 ";
    }elseif($m==3){
        $query="SELECT ‘article_id‘ FROM ‘suggestions‘ WHERE
        ‘check‘=0 AND  user_id=$ID AND ‘partition‘=$partition
        AND set_id=$set_id ORDER BY ‘avg_rank‘ DESC LIMIT 1 ";
```

```
    }else{
        $query="SELECT ‘article_id‘ FROM ‘suggestions‘ WHERE
        ‘check‘=0 AND  user_id=$ID AND set_id=$set_id
        ORDER BY ‘avg_rank‘ ASC LIMIT 1 ";
    }
}elseif($count>3){
    // User has more than two problems in her profile
    if ($m==0){
    $query="SELECT ‘article_id‘ FROM ‘suggestions‘ WHERE
    ‘check‘=0 AND ‘partition‘=0 AND user_id=$ID AND
    set_id=$set_id ORDER BY ‘avg_rank‘ DESC LIMIT 1";
    }elseif($m<4){
        $query="SELECT ‘article_id‘ FROM ‘suggestions‘ WHERE
        ‘check‘ =0 AND  user_id=$ID AND ‘partition‘ =$partition
        AND set_id=$set_id AND ‘article_sub_type‘
        NOT IN (SELECT ‘article_sub_type‘ FROM suggestions WHERE
        user_id=$ID AND ‘check‘=1 AND set_id=$set_id)
        ORDER BY ‘avg_rank‘ DESC LIMIT 1 ";
    }else{
        $query="SELECT ‘article_id‘ FROM ‘suggestions‘ WHERE
        ‘check‘ =0 AND  user_id=$ID AND set_id=$set_id
        ORDER BY ‘avg_rank‘ ASC LIMIT 1 ";
    }
}
```

```
$result=$mysqli->query($query);
if($row11 = $result->fetch_array(MYSQLI_ASSOC)){
$article_id=$row11['article_id'];
$query1= "SELECT * FROM `article` WHERE `article_id`=$article_id";
$result1=$mysqli->query($query1);
$row1 = $result1->fetch_array(MYSQLI_ASSOC);
$previous_id=$article_id;
$article_text=$row1['text'];
$title=$row1['title'];
$query4="UPDATE `suggestions` SET `check`=1 where
`article_id`=$article_id AND user_id=$ID AND set_id=$set_id";
$mysqli->query($query4);
$shorting=strip_tags($article_text);
$short_text=substr($shorting, 0, 250);
$short_text=$short_text."...";
if ($i==0){
    echo "<tr>";
}
$query2="Select * from rank where articleid=$article_id and userid=$ID";
$result2=$mysqli->query($query2);
if ($row2 = $result2->fetch_array(MYSQLI_ASSOC)){
    echo "<td> $title <br/> Your ranking is: ".$row2['rank']."  <hr/></td>";
}else{
    echo "<td align='center'>
```

```
<b>$title</b>

<br/>

$short_text

<br/>

<a href='#' class='topopup".$NUM."'>Read More</a>

<div id='toPopup".$NUM."'>

<div class='close".$NUM."'></div>

<span class='ecs_tooltip".$NUM."'>

Press Esc to close <span class='arrow'></span>

</span>

<div id='popup_content".$NUM."'> <!--your content start-->

<p>$article_text</p>

</div>

</div>

<!--  <div class='loader".$NUM."'></div> --!>

<div id='backgroundPopup".$NUM."'></div>

</td>

</tr>

<tr>

<td>

<div id='contact_form".$NUM."'>

<form name='contact'>

Excellent

<input type='radio' id='rank".$NUM."' name='rank[]' value='5' />
```

```
    Good
    <input type='radio' id='rank".$NUM."' name='rank[]' value='4' />
    Average
    <input type='radio' id='rank".$NUM."' name='rank[]' value='3' />
    Fair
    <input type='radio' id='rank".$NUM."' name='rank[]' value='2' />
    Poor
    <input type='radio' id='rank".$NUM."' name='rank[]' value='1' />
    <input type='hidden' name='articleid' id='articleid".$NUM."'
    value='".$article_id."'/>
    <input type='hidden' name='j' id='j".$NUM."'  value='".$NUM."'/>
    <input name='submit'  type='button' class='button'
    id='submit_btn' value='Send' />
    </form>
    </div>
    <br/>
    <br/>
    <hr/>
    </td>";
    $NUM++;
}
$i++;
if ($i==0){
    echo "</tr>";
```

```
    }
    if ($i>0){
        $i=0;
    }
    }
}
echo "</table></div>";
```

Computation of user-user similarity has been coded as follow:

```
function similarity($userid,$rec_type){
    $q1="select * from `user_concerns` where userid=$userid";
    $result1=$mysqli->query($q1);
    $row1= $result1->fetch_array(MYSQLI_NUM);
    $relation=$row1[1];
    $support_child=$row1[2];
    $have_child=$row1[3];
    $q2="select * from `user_lifestyle` where userid=$userid";
    $result2=$mysqli->query($q2);
    $row2= $result2->fetch_array(MYSQLI_NUM);
    $smoking=$row2[1];
    $alcohol=$row2[2];
    $activity=$row2[3];
    $q3="select * from `user_generalinfo` where userid=$userid";
    $result3=$mysqli->query($q3);
```

```
$row3= $result3->fetch_array(MYSQLI_NUM);
$pfcancer_history=$row3[1];
$menstruation=$row3[2];
$menopause=$row3[3];
$fertility_treatment=$row3[4];
$have_children=$row3[5];
$breast_feeding=$row3[6];
$firstchild_born=$row3[7];
$plan_baby=$row3[8];
$synthetic_hormones=$row3[9];
$radiation=$row3[10];
$q6="select * from `user_treatment` where userid=$userid";
$result6=$mysqli->query($q6);
$row6= $result6->fetch_array(MYSQLI_NUM);
$cancer_type=$row6[1];
$cancer_grade=$row6[2];
$used_treatment=$row6[3];
$used_t1=$row6[4];
$used_t2=$row6[5];
$used_t3=$row6[6];
$used_t4=$row6[7];
$used_t5=$row6[8];
$q4="select age,weight,height from `user` where userid=$userid";
$result4=$mysqli->query($q4);
```

```
$row4= $result4->fetch_array(MYSQLI_NUM);
$age=$row4[1];
$weight=$row4[2];
$height=$row4[3];
$BMI=$weight/($height/100)^2;  //BMI Calculation
/*************** TYPE OF BMI ****************/
if ($age<=30){
    $AGE_TYPE=1;
}elseif($age>30 && $age<=39){
    $AGE_TYPE=2;
}elseif($age>39 && $age<=49){
    $AGE_TYPE=3;
}elseif($age>49 && $age<=59){
    $AGE_TYPE=4;
}elseif($age>59 && $age<=69){
    $AGE_TYPE=5;
}elseif($age>69 && $age<=79){
    $AGE_TYPE=6;
}elseif($age>79){
    $AGE_TYPE=7;
}
if ($BMI<18.5){
    $BMI_TYPE="Underweight";
}elseif($BMI>18.5 && $BMI<24.9){
```

```
        $BMI_TYPE="Normal weight";
    }elseif($BMI>25 && $BMI<29.9){
        $BMI_TYPE="Overweight";
    }elseif($BMI>30){
        $BMI_TYPE="Obesity";
    }
/*************** TYPE OF BMI ****************/
    $q4="select userid from `user`";
    $checker=1;
    $i=0;
    if($result = mysqli_query($mysqli, $q4)){
        while($row4 = mysqli_fetch_array($result, MYSQLI_NUM)){
            $USERID=$row4[0];
            $N1=0;
            $N2=0;
            $N3=0;
            $N4=0;
            if ($userid!=$USERID ){
                $q1="select * from `user_concerns` where userid=$USERID";
                $result1=$mysqli->query($q1);
                $row1= $result1->fetch_array(MYSQLI_NUM);
                $relation_other=$row1[1];
                $support_child_other=$row1[2];
                $have_child_other=$row1[3];
```

```
$q2="select * from ‘user_lifestyle‘ where userid=$USERID";
$result2=$mysqli->query($q2);
$row2= $result2->fetch_array(MYSQLI_NUM);
$smoking_other=$row2[1];
$alcohol_other=$row2[2];
$activity_other=$row2[3];
$q3="select * from ‘user_generalinfo‘ where userid=$USERID";
$result3=$mysqli->query($q3);
$row3= $result3->fetch_array(MYSQLI_NUM);
$pfcancer_history_other=$row3[1];
$menstruation_other=$row3[2];
$menopause_other=$row3[3];
$fertility_treatment_other=$row3[4];
$have_children_other=$row3[5];
$breast_feeding_other=$row3[6];
$firstchild_born_other=$row3[7];
$plan_baby_other=$row3[8];
$synthetic_hormones_other=$row3[9];
$radiation_other=$row3[10];
$q6="select * from ‘user_treatment‘ where userid=$USERID";
$result6=$mysqli->query($q6);
$row6= $result6->fetch_array(MYSQLI_NUM);
$cancer_type_other=$row6[1];
$cancer_grade_other=$row6[2];
```

```
$used_treatment_other=$row6[3];
$used_t1_other=$row6[4];
$used_t2_other=$row6[5];
$used_t3_other=$row6[6];
$used_t4_other=$row6[7];
$used_t5_other=$row6[8];
$q4="select age,weight,height from `user` where
userid=$USERID";
$result4=$mysqli->query($q4);
$row4= $result4->fetch_array(MYSQLI_NUM);
$age_other=$row4[1];
$weight_other=$row4[2];
$height_other=$row4[3];
if ($age_other<=30){
    $age_other_TYPE=1;
}elseif($age_other>30 && $age_other<=39){
    $age_other_TYPE=2;
}elseif($age_other>39 && $age_other<=49){
    $age_other_TYPE=3;
}elseif($age_other>49 && $age_other<=59){
    $age_other_TYPE=4;
}elseif($age_other>59 && $age_other<=69){
    $age_other_TYPE=5;
}elseif($age_other>69 && $age_other<=79){
```

```
    $age_other_TYPE=6;
}elseif($age_other>79){
    $age_other_TYPE=7;
}
$BMI_other=$weight_other/($height_other/100)^2;
// BMI Calculation
/*************** TYPE OF BMI ****************/
if ($BMI_other<18.5){
    $BMI_TYPE_OTHER="Underweight";
}elseif($BMI_other>18.5 && $BMI<24.9){
    $BMI_TYPE_OTHER="Normal weight";
}elseif($BMI_other>25 && $BMI<29.9){
    $BMI_TYPE_OTHER="Overweight";
}elseif($BMI_other>30){
    $BMI_TYPE_OTHER="Obesity";
}
/*************** TYPE OF BMI ****************/
// Emotional Concern
if ($relation==$relation_other){
    $N4++;
}
if($support_child==$support_child_other){
    $N4++;
}
```

```
if($have_child==$have_child_other){
    $N4++;
}
// lifestyle category
if ($smoking==$smoking_other){
    $N3++;
}
if($alcohol==$alcohol_other){
    $N3++;
}
if ($activity==$activity_other){
    $N3++;
}
if ($BMI_TYPE==$BMI_TYPE_OTHER){
    $N3++;
}
// Risk Factor
if ($pfcancer_history==$pfcancer_history_other){
    $N1++;
}
if ($menstruation==$menstruation_other){
    $N1++;
}
if ($menopause==$menopause_other){
```

```
    $N1++;
}
if ($fertility_treatment==$fertility_treatment_other){
    $N1++;
}
if ($have_children==$have_children_other){
    $N1++;
}
if ($breast_feeding==$breast_feeding_other){
    $N1++;
}
if ($plan_baby==$plan_baby_other){
    $N1++;
}
if ($synthetic_hormones==$synthetic_hormones_other){
    $N1++;
}
if ($radiation==$radiation_other){
    $N1++;
}
if ($AGE_TYPE==$age_other_TYPE){
    $N1++;
}
// Treatment category
```

```
if ($cancer_type==$cancer_type_other){
    $N2++;
}
if ($cancer_grade==$cancer_grade_other){
    $N2++;
}
if ($used_t1==$used_t1_other){
    $N2++;
}else if ($used_t2==$used_t2_other){
    $N2++;
}else if ($used_t3==$used_t3_other){
    $N2++;
}else if ($used_t4==$used_t4_other){
    $N2++;
}else if ($used_t5==$used_t5_other){
    $N2++;
}
// Initializing the coefficient according to
// the type of recommendation
if ($rec_type==1){
    $W1=2/5;
    $W2=1/5;
    $W3=1/5;
    $W4=1/5;
```

```
}elseif($rec_type==2){
    $W1=1/5;
    $W2=2/5;
    $W3=1/5;
    $W4=1/5;
}elseif($rec_type==3){
    $W1=1/5;
    $W2=1/5;
    $W3=2/5;
    $W4=1/5;
}elseif($rec_type==4){
    $W1=1/5;
    $W2=1/5;
    $W3=1/5;
    $W4=2/5;
}
$similarity=($W1*$N1)/10+($W2*$N2)/3+($W3*$N3)/4+($W4*$N4)/3;
$i++;
$array1=array($USERID=>$similarity);
if($checker==1){
    $sim=$array1;
}else{
    $sim=$sim + $array1;
}
```

```
                $checker++;
            }
        }
    }
    arsort($sim, SORT_NUMERIC);
    $j=0;
    return ($sim);
}
```