

**A HYBRID APPROACH FOR PERSONALIZED
RECOMMENDER SYSTEM USING WEIGHTED TERM
FREQUENCY INVERSE DOCUMENT FREQUENCY**

REBECCA ADHIAMBO OKAKA

MASTER OF SCIENCE

(Computer Systems)

**JOMO KENYATTA UNIVERSITY OF
AGRICULTURE AND TECHNOLOGY**

2018

**A Hybrid Approach for Personalized Recommender System Using
Weighted Term Frequency Inverse Document Frequency**

Rebecca Adhiambo Okaka

**A Thesis submitted in partial fulfillment for the Degree of Master of
Science in Computer Systems in Jomo Kenyatta University of
Agriculture and Technology**

2018

DECLARATION

This thesis is my original work and has not been presented for a degree in any other university.

Signature:..... Date:.....

Rebecca Adhiambo Okaka

This thesis has been submitted for examination with our approval as University Supervisors:

Signature:..... Date:.....

Prof. Ronald Waweru Mwangi

J.K.U.A.T, Kenya.

Signature..... Date.....

Dr. George Onyango Okeyo

J.K.U.A.T, Kenya.

DEDICATION

To the Almighty God who has been my eternal rock and source of refuge, and for His word in Philippians 4:13 that kept me all through the journey of completing this work. To my late mother Achola, for all that I am or hope to be I owe it to her.

ACKNOWLEDGEMENTS

I would like to sincerely thank my supervisors, Prof. Ronald Waweru Mwangi and Dr. George Onyango Okeyo for their assistance, support and encouragement. I thank also the other members of the department, Dr. Richard Rimiru, Dr. Agnes Mindila and Dr. Kennedy Odhiambo for their helpful suggestions. In addition, I am grateful to my colleagues in the School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology for their collaboration and valuable comments provided. Special thanks to my family for their love and encouragement in all aspects of my life.

TABLE OF CONTENTS

DECLARATION.....	ii
DEDICATION.....	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	ix
LIST OF FIGURES	x
APPENDIX.....	xi
APPENDIX.....	xi
LIST OF ABBREVIATIONS	xii
ABSTRACT.....	xiii
CHAPTER ONE	1
INTRODUCTION.....	1
1.1. Background of the study	1
1.2. Research Objectives	3
1.3. Research Questions	5
1.4. Approach.....	6
1.5. Outline of the Thesis	7
CHAPTER TWO	8
LITERATURE REVIEW	8

2.1.	Introduction.....	8
2.1.1.	What is a Recommender System?	8
2.1.2.	History of Recommender Systems	8
2.1.3.	Components of a Recommender System	10
2.1.4.	Goals of Recommender Systems	11
2.1.5.	Classification of Recommender Systems.....	13
2.1.6.	Taxonomies of Recommender Systems.....	15
2.1.6.1.	Simplified Approaches	20
2.1.6.2.	Intelligent Filtering.....	21
2.1.7.	Anatomy of a Recommender System	33
2.1.8.	Recommender Systems Evaluation	35
2.1.9.	Existing Hybrid Approaches	35
2.2.	Hybrid Logistics Function.....	37
2.3.	Information Retrieval (IR).....	38
2.3.1.	The Vector Space Model	39
2.3.2.	Term Frequency Inverse Document Frequency	40
2.3.3.	Term Mapping Approaches	42
2.4.	Summary	43
2.5.	Research gap	44
CHAPTER THREE		45

METHODOLOGY	45
3.1. Introduction.....	45
3.2. The Hybrid Filtering Model	45
3.3. The Vector Space Model in the Hybrid Filtering Model	47
3.3.1. The Vector Space Model in Content based filtering	53
3.3.2. The Vector Space Model in Collaborative filtering	58
3.4. The Hybridization Process	62
3.5. Model Evaluation Metrics	64
CHAPTER FOUR.....	66
EXPERIMENTS AND RESULTS	66
4.1. Introduction.....	66
4.2. Dataset	66
4.3. Experimental Setup.....	66
4.4. Results.....	67
4.5. Discussions	74
CHAPTER FIVE	76
CONCLUSIONS AND RECOMMENDATIONS.....	76
5.1. Introduction.....	76
5.2. Summary.....	76

5.3. Conclusions.....	77
5.4. Recommendations	78
5.5. Suggestions for future work	78
REFERENCES.....	79
APPENDIX.....	87

LIST OF TABLES

Table 3.1: Sample Data for illustrating VSM retrieval process	49
Table 3.2: TF Scores for data illustrating VSM retrieval process	50
Table 3.3: IDF Scores for data illustrating VSM retrieval process.....	51
Table 3.4: TF Values of the Query in each Item.....	51
Table 3.5: IDF Values of the query in each Item.....	52
Table 3.6: TFIDF Values of the Query in each Item	52
Table 3.7: Use - Item Rating Matrix for CBF illustration	55
Table 3.8: User - Item Similarity Matrix	56
Table 3.9: Item - Item Similarity Matrix.....	56
Table 3.10: Item Similarity Scores	57
Table 3.11: User - Item Rating Matrix for CF illustration	60
Table 3.12: User - User Similarity Matrix	60
Table 3.13: Extended User - Item, User - User Matrix.....	63
Table 4.1: MAE given 100 Items.....	68
Table 4.2: MAE given 500 Items.....	69
Table 4.3: MAE given 700 Items.....	71
Table 4.4: MAE given 1200 Items.....	72

LIST OF FIGURES

Figure 2.1: Levels of Personalization in Recommender Systems	14
Figure 2.2: Taxonomy of Recommender Systems (Adomavicius et al., 2005).....	16
Figure 2.3: Taxonomy of Recommender Systems (Burke R, 2002)	16
Figure 2.4: Taxonomy of Recommender Systems (Montaner et al., 2003).....	17
Figure 2.5: Taxonomy of Recommender Systems (Schafer et al., 2001).....	19
Figure 2.6: Classification of Recommender Systems as presented in our study	20
Figure 2.7: Three steps of building a Recommender System	34
Figure 2.8: Basic Recommender System Ecosystem.....	35
Figure 3.1: The Hybrid Filtering Model	47
Figure 3.2: Weighted Hybrid Architecture	62
Figure 4.1: MAE given 100 Items	68
Figure 4.2: MAE given 500 Items	70
Figure 4.3: MAE given 700 Items	71
Figure 4.4: MAE given 1200 Items	73
Figure 4.5: Performance Summary	74

APPENDIX

Appendix: i Research Publication:	88
------------------------------------------------	-----------

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
CF	Collaborative Filtering
CBF	Content Based Filtering
CosSim	Cosine Similarity
DFD	Data Flow Diagram
eCommerce	Electronic Commerce
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HF	Hybrid Filtering
IDF	Inverse Document Frequency
IR	Information Retrieval
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Square Error
RS	Recommender System
RSS	Really Simple Syndication
SPSS	Statistical Package for Social Sciences
TF	Term Frequency
TFIDF	Term Frequency Inverse Document Frequency
UML	Unified Modeling Language
XML	eXtensible Markup Language

ABSTRACT

Recommender systems are gaining a great popularity with the emergence of eCommerce and social media on the internet. These recommender systems enable users' access products or services that they would otherwise not be aware of due to the wealth of information on the internet. Two traditional methods used to develop recommender systems are content-based and collaborative filtering. While both methods have their strengths, they also have weaknesses; such as limited content analysis, overspecialization and new user problem in content-based filtering, scalability, data scarcity and new item problem in collaborative filtering. These weaknesses leads to poor recommendation quality, but some of them can be overcome by combining two or more recommender methods to form a hybrid recommender system. This thesis deals with issues related to the design and evaluation of a hybrid approach for personalized recommender system that combines content-based and collaborative filtering methods to improve the precision of a recommendation. The content based and collaborative filtering methods use weighted Term Frequency Inverse Document Frequency to compute similarities among users and items. Experiments done using Movie Lens dataset shows that the hybrid approach for personalized recommender system overcomes the challenge of recommendation precision experienced in pure recommender systems.

CHAPTER ONE

INTRODUCTION

1.1. Background of the study

Changes in information seeking behavior can be observed globally (Gavgani, 2010). Rapid increase in blogs and websites has led to an increase in information overload and it has become extremely difficult for users to locate current relevant information. In addition, the recent increase in Rich Site Summary (RSS) also known as Really Simple Syndicate feeds used for news update has been caused by the wide use of blogs; this means that much effort is now needed to search for the much needed information from the RSS contents because of the enormous quantity of materials. With vague ideas on where to get information, users often get lost or feel uncertain when seeking information on their own, giving rise to the need for creating systems that are able to process the existing information on one side, and help users by suggesting products, services or articles that match their tastes and preferences on the other side. Recommender systems (RS) suggest items of interest to users of information systems, they are promising tools to deal with these issues.

There are lots of taxonomies of RS. They can be divided according to the fact whether the created recommendation is personalized or non-personalized (Kazienko et al., 2005). Some research distinguishes three main categories of personalized RS: collaborative filtering (CF), content-based filtering (CBF), and hybrid filtering (HF) (Adomavicius et al., 2005). In CF, a user gets recommendations of items that he or she hasn't rated or liked

before, but that were already positively rated by users in his or her neighborhood. In CBF, a user gets recommendations of items he or she had not seen or rated but similar to the ones he or she had rated or liked earlier. HF combines two or more filtering methods to overcome the limitations of each method. Adomavicius and Tuzhilin claim that these three categories are the most popular and significant recommendation methods. However, they pinpoint the shortcomings of these methods when used individually such as limited content analysis, overspecialization and the new user problem in CBF (Adomavicius et al., 2005), scalability, data scarcity and new item problem in CF. CBF tend to overspecialize because only items with a high similarity to those already rated will be suggested to the individual user, also a user first has to rate a sufficient number of items before the system is able to make accurate recommendations. Unlike CBF, CF systems exhibit the new user problem and first have to learn user preferences to make reliable recommendations. Beside the new user problem collaborative approaches also exhibit the new item problem, which means that a new item needs to be rated by a sufficient number of users in order be recommended accurately by the system.

Adomavicius and Tuzhilin also propose possible improvements on recommender systems; such as combining two or more recommender methods to gain better performance using different hybridization techniques to overcome the challenges of single recommender systems. Some of the combination methods that have been used include; weighted hybridization, switching hybrid, mixed hybrid, feature combination, feature augmentation, cascade and Meta level techniques. According to Tuzhilin (Tuzhilin et al., 2005) the combination of two or more recommender methods proceeds in four different ways; creating a unified model recommender system that brings all

approaches together, utilizing some rules of one approach into a different approach and vice versa, separate implementation of algorithms and then joining results, or developing one model that applies the characteristics of both methods.

This thesis presents a hybrid approach that uses the weighted hybridization technique which probably is the most straight forward architecture for a hybrid system – it involves separate implementation of algorithms then joining results; it is based on the idea of merging predicted ratings computed by individual recommenders to form a ranked list of items from which top items are selected and presented to the user as recommendations.

Many different approaches to recommender systems have been developed within the past few years, but the interest in this area still remains high due to the growing demand on practical applications which are able to provide personalized recommendation and deal with information overload (Gauch et al., 2007 and Adomavicius et al., 2005). Beside recommendation precision, another key consideration in computer science is computation efficiency (Koren et al., 2008). Usually a recommender system needs to deal with millions of users and items, computing rating estimations in an instant or even in real time. Under the restrictions of memory and time consumption many prediction algorithms quickly reach their limit of possible manageable data volume. In order to handle large scale datasets, further improvements on information representation and recommendation modeling need to be done.

1.2. Research Objectives

The main objective of this research is to find a general way to improve recommendation precision. The specific objectives of this research are:

- i. To study how hybrid recommender systems work.
- ii. To design a hybrid approach for recommender system.
- iii. To evaluate the hybrid approach for a personalized recommender system.

The main purpose of the personalized hybrid recommender system is to improve recommendation precision as well as provide top most relevant items to users as recommendations. Usually recommender systems assist users get information that match their tastes and preferences; by suggesting items of interest to users within a system. Generally, the main purpose of a recommender system is to reduce information overload by estimating relevance and providing personalized recommendations to target users. A recommender system deals with millions of users and items, computing rating predictions in an instant or even in real time. Different approaches of recommender systems have been described in literature and each of these approaches have shortcomings that affect the precision of the recommendation they produce; limited content analysis, overspecialization and the new user problem in content-based filtering, scalability, data scarcity and new item problem in collaborative filtering. A hybrid approach that combines content-based and collaborative filtering methods is proposed to overcome the challenges of the two methods implemented separately.

1.3. Research Questions

The major question in this research is how will the recommendation precision be improved? Other questions that rise from the specific objectives are;

- i. How do hybrid recommender systems work?
- ii. How can the hybrid approach for personalized recommender system be designed?
- iii. How can the hybrid approach design be evaluated?

Generally, in a recommender system, there exists a large number of m items $I = \{i_1, i_2, \dots, i_m\}$, which are described by a set of l attributes, $A = \{a_1, a_2, \dots, a_l\}$, where each item is described by one attribute or more, a number of n users, $U = \{u_1, u_2, \dots, u_n\}$ and for each user u , a set of rated items $IR_u = \{u_{i_1}, u_{i_2}, \dots, u_{i_n}\}$. For $u \in U$ and $i \in I$, the recommender system predicts the rating $r'_{u,i}$ called the predicted rating of the user u on the item i such that $r'_{u,i}$ is unknown. From this formulation, the predicted rating a user would give an item he or she have not seen is computed, the accuracy of the predicted rating is also computed, both the predicted rating and its accuracy determines how precise the prediction is.

The main contribution of this work is that the proposed approach is universal and can be applied to any other problem domain. Also, the system design is done using UML diagrams that can easily be modified later when additional or new requirements arise, as a result the system can be easily maintained and enhanced. UML enable the system functionalities be described at a high level of abstraction and also enforces system modularization, by splitting the system's functionality into a collection of connected

components, each component is a replaceable part of the overall system that fulfils a clear function, evolves independently, can be reused and updated by alternative components. Lastly, using hybrid approaches avoid some challenges of pure recommender systems (Adomavicius et al., 2005); the content-based and collaborative filtering methods complement each other therefore increases the efficiency of recommendations. In this case, the new user problem in content-based filtering is eliminated by collaborative filtering and the new item problem in collaborative filtering is eliminated by content-based filtering. This hybrid approach uses the most widely used effective information retrieval model, the vector space model, and a very simple efficient ranking algorithm term frequency inverse document frequency.

1.4. Approach

To overcome the challenges of pure content-based and collaborative filtering methods, a hybrid approach that combines both methods is used (Burke et al., 2007). Two part hybrid recommender systems are quite successful, but under different domains and data characteristics, different hybrid recommender systems achieve different results. Our hybrid approach uses the weighted hybridization technique based on the idea of merging items predicted ratings computed by the two individual recommenders, then presenting items with higher ratings to the user as recommendations. While in content-based filtering we explore user item features and similarities in collaborative filtering we use user behavior and similarities.

The hybrid approach adapts the vector space model in both content-based filtering and collaborative filtering, uses ranking algorithm term frequency inverse document

frequency and cosine similarity measure to find the relationships among users U , items I and attributes A .

To evaluate the hybrid system, the Movie Lens datasets that already provides different training and test samples which exhibit perfect properties for validation is used. The two disjoint datasets (training and test sets) are used to measure the difference between predicted and actual rating values. This dataset contains 100,000 anonymous ratings of 1 to 5 made by 943 users on approximately 1682 movies.

The quality of recommender systems are typically evaluated using predictive accuracy metrics, where the predicted items rating are directly compared to the actual rating a user gives an item. The most commonly used metric in literature is the Mean Absolute Error (MAE); defined as the average absolute difference between predicted rating and actual rating. Using MAE measure the proposed hybrid system is compared to individual content-based and collaborative filtering methods.

1.5. Outline of the Thesis

The remainder of the thesis is organized as follows: First in chapter 2, some basic facts for the thesis fields; RS, HRS, IR and TFIDF are introduced. The concepts of term matching that are of special importance for this thesis are briefly explained towards the end of this chapter. In chapter 3 the proposed hybrid approach is presented. Chapter 4 contains the hybrid approach's evaluation process and results. Finally, Chapter 5 presents conclusions and further work based on the findings presented in chapter 4

CHAPTER TWO

LITERATURE REVIEW

2.1. Introduction

2.1.1. What is a Recommender System?

Recommender systems or recommendation systems (sometimes replacing "system" with a synonym such as platform or engine) are a subclass of information filtering system that seek to predict the 'rating' or 'preference' that a user would give to an item (Ricci et al., 2011), learn from a user's behavior and generate meaningful suggestions of items that might be of interest to the user.

2.1.2. History of Recommender Systems

When people want to purchase items, they have to make decisions which items to buy, likewise they also have to decide which news or book to read in open access portals or which videos to watch in a multimedia store. Their choice always depends on other users' opinions, especially in ecommerce. The older version of recommendation is "word of mouth" opinion (Kangas S, 2002). This is the method most people use when they decide to buy an item, they ask their friends who are trustworthy which item they would suggest to buy and why they should pick on that particular item but not the other.

At the beginning the recommender systems did not create a separate research area and their roots can be traced back to the cognitive science, information retrieval, forecasting theories, approximation theory and management sciences (Adomavicius et al., 2005). However the growth of internet and the rise of ecommerce solutions caused the

development of the online recommender systems and since the mid 1990s they have become an important research domain (Adomavicius et al., 2005). The reason for that was the opportunity to share the opinion between a vast numbers of people who use the internet.

The first known project in the recommendation area is the *GroupLens* (Riedl et al., 2006). The roots of this project can be traced back to the year 1992 when the main goal of the system was to explore automated collaborative filtering. After that, collaborative technique was applied in filtering the information in Usenet news (Montaner et al., 2003). *Ringo* agent was one of the first applications that provided personalized music recommendations (Shardanand et al.,1994), which became available on 1st July, 1994 (Shardanand et al.,1995), in this method the users provide ratings of musical articles, based on this the user profile, which changes overtime is created. The profile enables the generation of recommendations by use of the social filtering methods (Shardanand et al., 1995), this method can be treated as automation of the “word of mouth” recommendation. The application that utilized the *Ringo* system was *Firefly’s* system. This technology was further developed by Yahoo who signed up to use it (Kangas S, 2002). Finally this method was used by the book dealer Amazon.com that introduced the *BookMatcher* system. At the beginning the *BookMatcher* was used for book recommendations, but later on, the system started to recommend other types of items, using also other methods of recommendations (Kangas S, 2002).

Today recommender systems are one of the well-established artificial intelligence applications in modern computer science, and are being used as typical software components in eCommerce, mCommerce (Sadeh, 2002), social media and tourism

systems (Werthner et al., 1999) to recommend magazine articles, books, goods etc. The most popular ones are movies, music, news, books, research articles, search queries, social tags, and products in general. There are also recommender systems for experts, jokes, restaurants, financial services (Alexander et al., 2007) life insurance, and persons (online dating).

Although RS have been investigated and developed for many years they are still in the area of interests for many researchers. Moreover, there are still many challenges in this area. Each of the existing recommender methods suffers from shortcomings. The goal of the current research is to cope with these disadvantages by combining many approaches together (hybrid approach) and create a recommender system which will fulfill user needs and expectations.

2.1.3. Components of a Recommender System

Recommender systems are often comprised of several components namely; users, items, preferences, ratings, and neighborhoods (Ricci et al., 2010). *Items* are the things or objects that are being recommended to a user. For example, items are often products, news articles, songs or movies. These items can be characterized by their respective metadata that include relevant titles, tags, or keywords. For example, news articles can be characterized by content category, songs can be characterized by artists and genre, and movies can be characterized by genre and director. *Users* are the people who are being recommended items. They often need assistance or guidance in choosing an item within an application and use recommendation to help them make an informed and hypothetically better decision. A user model can be built over time in an effort to make

better recommendations for each particular user. This user model acts as a profile in which preferences and actions are encoded and is representative of the history of a user and their interactions with items within the RS. These interactions are known as preferences. *Preferences* can be interpreted as the user's opinion of an item in a RS and can be both explicit and implicit. Preferences are often categorized as ratings if a RS provides an interface to rate items. *Rating* is a type of explicit preference that represents a relationship between a user and an item. Every rating can describe, for example, how a user feels about certain items. An example of an explicit rating is a user rating a movie with five stars. From this rating, the RS can definitively conclude that the user likes the movie item. An implicit preference can be a user clicking on a link or skipping a video. In these examples, we can infer data from these implicit preferences and assume that the user may like an item if they click on its link, or do not like a video that they skip. *Neighborhood* relates users and their preferences and represents group of similar users. In collaborative filtering (CF) environments, which will be discussed later, neighborhoods of similar users help a RS decide on items to recommend to a user based on users with similar tastes (Ricci et al., 2011).

2.1.4. Goals of Recommender Systems

Recommender systems became an important and almost integral part of recent web sites. The aim of these systems is to help the potential users pick the appropriate item that match their specific needs, so that they can be seen as decision support systems. On the other hand, they serve as the marketing help for the ecommerce stores because they increase the attractiveness of the offer. The main goals of recommender systems are;

- To cope with information overload (Adomavicius 2005, Kazienko 2004, Montaner 2003)
- To help all users (new, frequent, and infrequent) to make decisions what items to buy, which news to read next (Terveen et al., 1997) which movie is worth watching, etc.
- To build credibility through community (Schafer et al., 2001) and maintain the loyalty of the users (Kazienko et al., 2005)
- To invite users to come back (Schafer et al., 2001)
- To enhance ecommerce sales and cross sell (Schafer et al., 2001)

The first two points show why the RS are important from the consumer point of view. First of all, they are very useful tool that help to cope with the information overload (Adomavicius 2005, Kazienko 2004, Montaner 2003). The recommender systems enable selection of a small subset of items, from millions of items, that seems to fit the users' needs and preferences. Although it is almost impossible to predict precisely the users' needs, such set of suggestions helps to limit the number of choices. Furthermore, by restricting the number of suggested items, these kinds of systems help people to make decisions, what items to buy, which news to read next (Terveen et al., 1997) or which movie is worth watching, much faster than by the use of regular search engines. The rest of the enumerated above items show that RS can be seen as the marketing tools because they enhance ecommerce sales (Schafer et al., 2001). As it was mentioned before, these systems can help people to find the items that they want to have. As a result this facilitates to convert the people who only watch to the buyers. When consumers buy

things that are recommended by the system, the additional items can be suggested in order to increase the cross sell. This leads to building and maintaining the loyalty of the customers (Kazienko et al., 2005); what is more, it encourages the customer to come back in the future. In the Internet and ecommerce where the number of competitors is very high, this feature is a crucial advantage of the recommender systems (Schafer et al., 2001).

The aim of all the goals that were pinpointed above is to satisfy the customer. Additionally, RS ought to be as high efficiency as possible in order to increase their ROI (Return on Investment). However, the recommendations not only should exist but also ought to be relevant. The problem that can appear is too high number of false-positive recommendations, which are defined as suggestions that were created for the users, although they do not suit them. In conclusion, the goals of the recommendations can be achieved only if the generated suggestions are relevant.

2.1.5. Classification of Recommender Systems

Recommender systems can be classified according to the level of personalization as *non-personalized* and *personalized* (Kazienko et al., 2005) recommender system. The former methods do not consider the characteristics and preferences of the customers, whereas the latter tightly depends on the user profile.

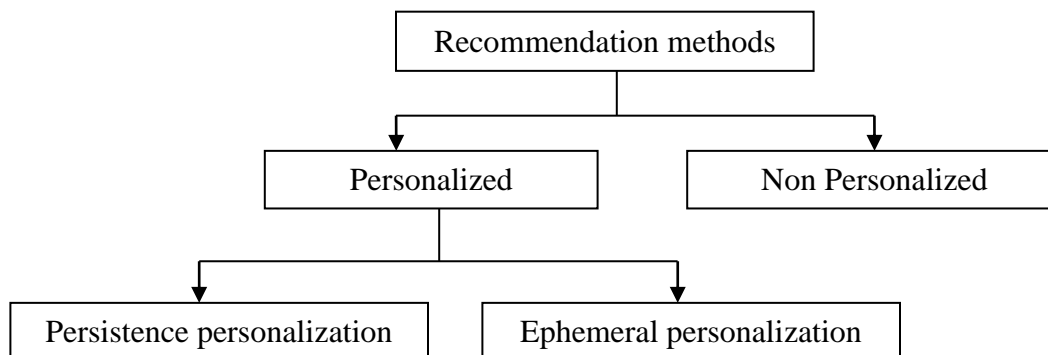


Figure 2.1: Levels of Personalization in Recommender Systems

An example of a non-personalized recommendation is the recommendation that suggests the products which were best rated in the past by all customers. In order to create this kind of recommendations, statistical methods are commonly used. This kind of recommendation depends on the policy of ecommerce and belongs to the technique where much calculation is not required. The main feature of suggestions is that they are the same for all customers. Usually, the user will find one item from the list of the most popular ones, it similar to when someone goes to a bookshop and finds a shelf with the best sellers.

According to Adomavicius and Burke, some research claims that recommender systems are those which produce personalized recommendations since their outputs help guide users to products and services that fulfill their personal needs; users find right products and services from the large amount of possible choices. As a result personalized recommender systems cope with information overload better than non-personalized recommender systems. The personalized recommendation is based on the demographic information about users or on the analysis of the past behavior of the user in order to

predict their future behavior (Schafer et al., 2001). According to (Kazienko et al., 2005) and (Schafer et al., 2001) *personalization* can be either *persistent* or *ephemeral*.

Persistent personalization is based on the previous users' behaviors that enable the creation of unique list of products for each user. The requirement in this situation is that users must log into the system in order to create their own profiles, each person on the web site sees different recommendations because the recommendations depend directly on the users' personal data.

In the *ephemeral personalization* the user profile is not necessary. In this case the recommendations are created according to the users' behaviors during a current session, their navigation and selection (Schafer et al., 2001). In this technique the recommendations are the same for all users (Kazienko et al., 2005).

2.1.6. Taxonomies of Recommender Systems

There are lots of taxonomies of recommender systems. They can be divided according to the fact whether the created recommendation is personalized or non-personalized (Kazienko et al., 2005) as discussed earlier. Some research distinguishes three main categories of RS (Figure 2.2), where all of them are personalized: *collaborative filtering*, *content based filtering*, and *hybrid methods* (Adomavicius et al., 2005). Adomavicius and Tuzhilin claim that these three categories are the most popular and significant recommendation methods. However, they pinpoint the shortcomings of those methods and propose possible improvements.

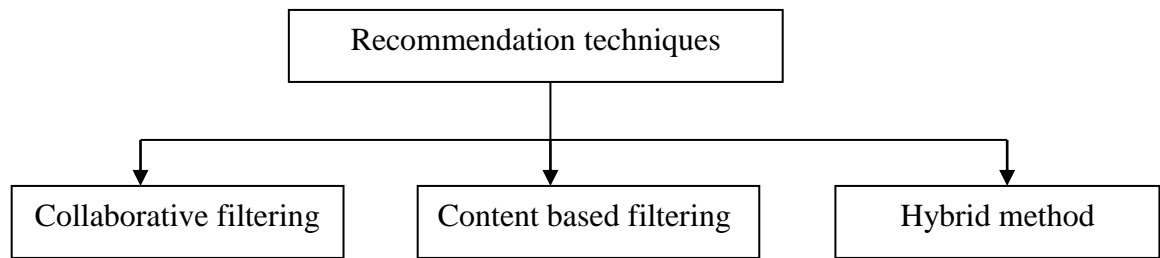


Figure 2.2: Taxonomy of Recommender Systems (Adomavicius et al., 2005)

Robin Burke distinguished five techniques of the recommendation (Figure 2.3) according to the type of a background and input data as well as the algorithm that is used to create the suggestions. The background data is the information that the system possesses before the processor recommendation begins, whereas the input data enables to create the recommendations for particular user. The input data is provided by users and directly related to the user for whom recommendations are generated. The background data is the basis that enables to distinguish the following methods of recommendation: collaborative, content based, demographic, utility based, and finally knowledge based techniques (Burke R, 2002).

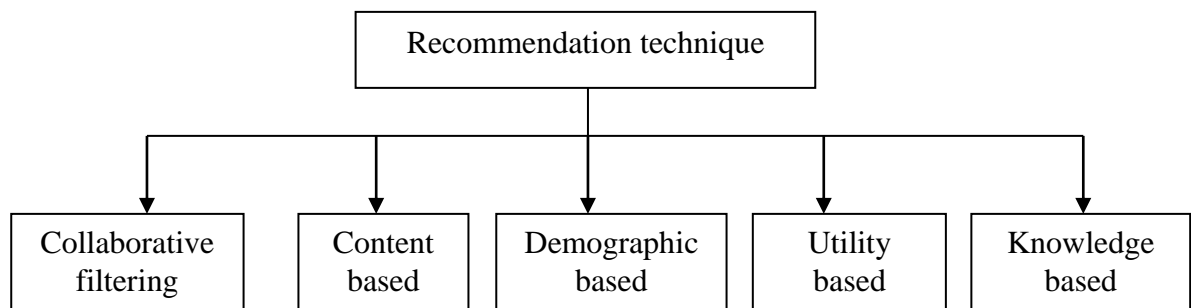


Figure 2.3: Taxonomy of Recommender Systems (Burke R, 2002)

Another research, that is worth to mention, is the taxonomy of recommender agents proposed by Miquel Montaner, Beatriz Lopez, and Lluís de la Rosa (Montaner et al., 2003). In their research, authors distinguish two main approaches to the problem of RS: spatial and functional. Furthermore, from the functional point of view, they created eight dimensions, which are the basis for further classification of the recommender agents. Five of them concern the profile creation and maintenance, and three of them users' profile exploitation.

Profile creation and maintenance are very important parts of the recommender systems, but for this master thesis, the assumption is that the input data for the recommendation technique is the appropriate user profile. Figure 2.4 presents the dimensions that characterize RS.

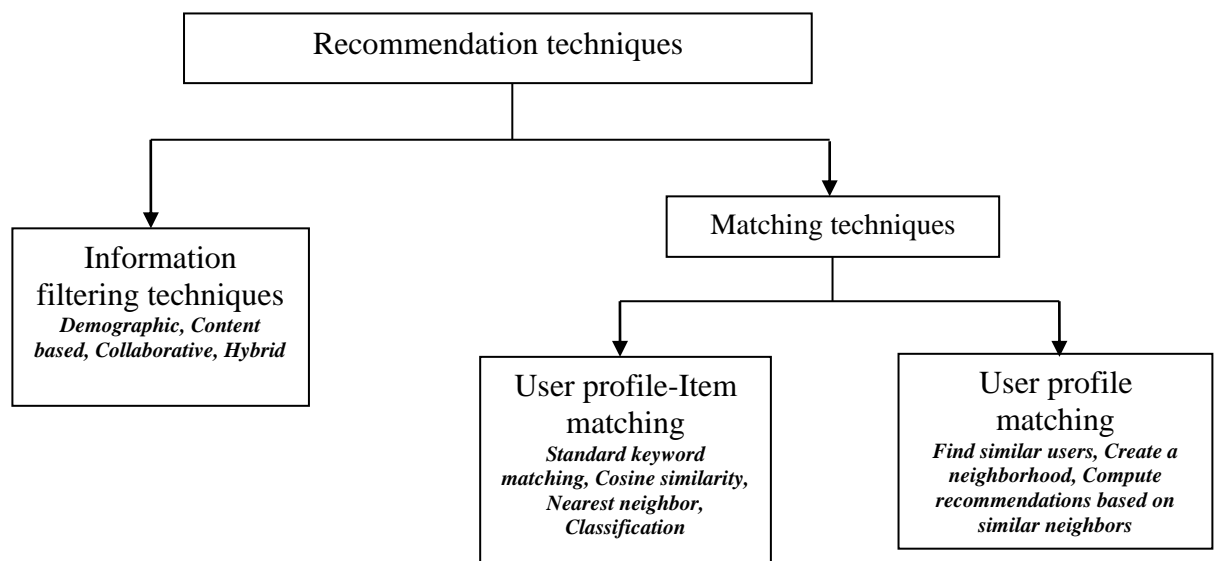


Figure 2.4: Taxonomy of Recommender Systems (Montaner et al., 2003)

The *information filtering, user profile-item matching, and user profile matching* are three main dimensions of the profile exploitation. Concerning information filtering techniques the most important techniques are demographic, content based, and collaborative filtering.

The goal of the user profile-item matching is to compare the representation of the user profile (e.g. user interests or preferences) and the description of the item and as a result pick the items that are relevant for the specific customer. The examples of such techniques are presented in the Figure 2.4. The last distinguished dimension is the user profile matching that enables to find the similar users or group of users.

Another researcher, Schafer et al. considered and analyzed not only the recommendation methods, but also, similar to Burke, the input data that is delivered by the targeted customer for whom the recommendation is created, and by the rest of the customers (Schafer et al., 2001). This data serves as the input for the recommendation technique. As the result of applying the appropriate technique the targeted customer receives the suggestion of items he or she may be interested in. However, the output of applying recommendation method is not only the suggestions, but also the ratings and the prediction. The ratings are commonly used when the number of customers is small and the users know each other. In such case, it can be helpful to display the individual ratings of other customers (Schafer et al., 2001). Several RS provide the prediction of the ratings that the user would probably give to an item (Schafer et al., 2001). According to Schafer et al. the following types of the recommender systems can be distinguished: raw retrieval (also called “null recommendation”), manually selected, statistical summarization,

attribute based, item to item correlation (also called content based filtering), and user to user correlation (also called collaborative filtering).

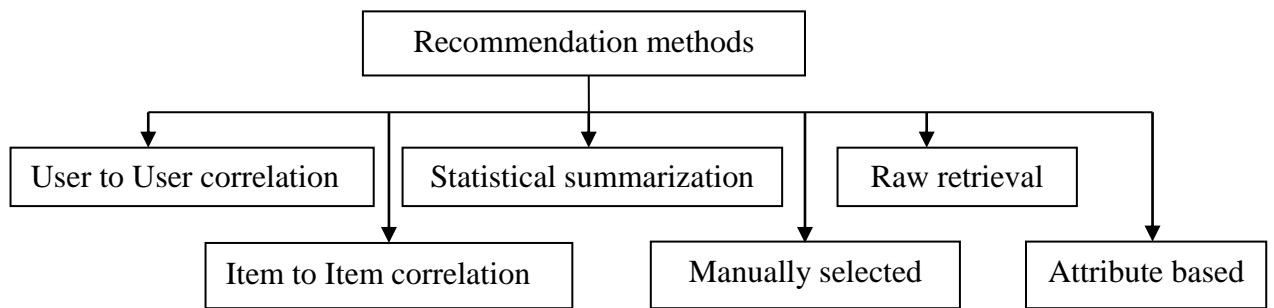


Figure 2.5: Taxonomy of Recommender Systems (Schafer et al., 2001)

There are many approaches to RS and these enumerated above create only a small part of the existing taxonomies. The classification that is the basis for the further descriptions of the recommender systems in this thesis is presented in the Figure 2.6. The most common and widely implemented methods of recommendation are enumerated and briefly described below.

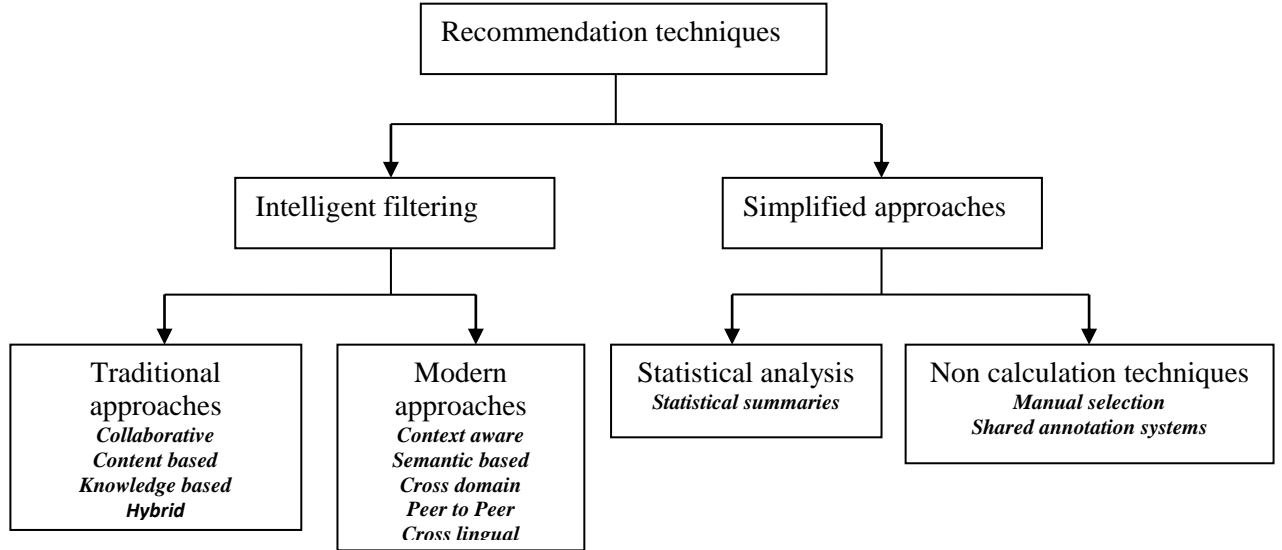


Figure 2.6: Classification of Recommender Systems as presented in our study

The recommendation methods are divided into intelligent filtering and simplified approaches. Not every recommendation method is taken into consideration in this classification. The reason for that is that the purpose of this description is not to present all existing methods of recommendation, but to discuss the vital ones.

2.1.6.1. Simplified Approaches

Simplified approaches consist of statistical analysis and non-calculation techniques. Both methods do not require complicated computations and are the non-personalized methods of recommendation.

a) Non calculation techniques

Manual selection and shared annotations systems are the examples of the non-calculation techniques. The former one is also called “human recommendation” and it is based on the suggestions made by the experts (Kazienko et al., 2004, Schafer et al., 2001). For

example in the store where movies are sold, the critics can be considered as experts. Shared annotation systems enable to exchange the opinions between the customers. They can submit their opinions about the items that they have purchased and in this way help other people to pick the items which are high quality.

b) Statistical Analysis

The statistical analysis, in contrary to the non-calculation techniques requires calculations, which are, nevertheless, not very complicated. The system provides the ratings of the items that are based on the statistical factors. Some of these factors are: the number of sold units of each of the items and average rating of the item submitted by the customers who have already bought this item (Kazienko et al., 2004). The statistics are calculated in the context of the whole community.

The main advantage of simplified approaches is that they do not require the complex calculations, while their major disadvantage is that, recommendations are the same for all the users and in consequence the suggestions are too general and not personalized. As a result it is not possible to provide the recommendations fitting to the unique preferences of a specific user.

2.1.6.2. Intelligent Filtering

Intelligent filtering techniques consist of traditional and modern techniques of recommendation. Both techniques require complicated computation and are personalized methods of recommendation.

a) Traditional Approaches

There are three main approaches to traditional recommender systems: collaborative filtering, content based filtering and hybrid approaches;

i) Collaborative Filtering (CF)

Collaborative filtering methods are based on collecting and analyzing a large amount of information on user's behavior, activities or preferences and predicting what users will like based on their similarity to others. According to (Linden et al., 2003) the idea of collaborative filtering is in finding users in a community that share appreciations. If two users have same or almost same rated items in common, they have similar tastes. Such users build a group or a neighborhood. A user gets recommendations to those items that he or she hasn't rated before, but that were already positively rated by users in his or her neighborhood. There are two methods to collaborating filtering technique namely;

User based collaborating filtering; Suggests items to a user based on what other users in his or her neighborhood liked or have rated.

Item based collaborating filtering; Suggests items to a user based on item similarity to what the user had previously liked or rated.

CF is widely used in ecommerce and online social networks (Ricci et al., 2011). Many algorithms have been used in measuring user or item similarity in CF recommender systems including k-nearest neighbor (kNN) and Pearson correlation approaches. Although collaborative filtering approach is the oldest method, it is still very effective and have the following advantages;

- **Simplicity**: Neighborhood based methods are intuitive and relatively simple to implement.

- **Justifiability:** Such methods also provide a concise and intuitive justification for the computed predictions. For example, in item based recommendation, the list of neighbor items, as well as the ratings given by the user to these items, can be presented to the user as a justification for the recommendation. This can help the user better understand the recommendation and its relevance, and could serve as basis for an interactive system where users can select the neighbors for which a greater importance should be given in the recommendation.
- **Efficiency:** One of the strong points of neighborhood based systems is their efficiency. Unlike most model based systems, they require no costly training phases, which need to be carried out at frequent intervals in large commercial applications. While the recommendation phase is usually more expensive than for model-based methods, the nearest neighbors can be pre computed in an offline step, providing near instantaneous recommendations. Moreover, storing these nearest neighbors requires very little memory, making such approaches scalable to applications having millions of users and items. **Stability:** Another useful property of recommender systems based on this approach is that they are little affected by the constant addition of users, items and ratings, which are typically observed in large commercial applications. For instance, once item similarities have been computed, an item based system can readily make recommendations to new users, without having to re-train the system. Moreover, once a few ratings have been entered for a new item, only the similarities between this item and the ones already in the system need to be computed.

However according to Sanghavi, collaborative filtering also has challenges (Sanghavi et al., 2007) as discussed below.

- **Data scarcity and the cold start problem:** In practice, many commercial recommender systems are based on large datasets. As a result, the user item matrix used for collaborative filtering could be extremely large and sparse, which brings about the challenges in the performances of the recommendation. The cold start problem is inherent to data scarcity. As collaborative filtering methods recommend items based on users' past preferences, new users will need to rate sufficient number of items to enable the system to capture their preferences accurately and thus provides reliable recommendations.
- **The scalability problem:** As the numbers of users and items grow, traditional CF algorithms will suffer serious scalability problems such as;
- ✓ **The synonyms problem:** synonymy refers to the tendency of a number of the same or very similar items to have different names or entries. Most recommender systems are unable to discover this latent association and thus treat these products differently. For example, the seemingly different items “children movie” and “children film” are actually referring to the same item. Indeed, the degree of variability in descriptive term usage is greater than commonly suspected. The prevalence of synonyms decreases the recommendation performance of CF systems.

- ✓ Grey sheep problem: Grey sheep refers to the users whose opinions do not consistently agree or disagree with any group of people and thus do not benefit from collaborative filtering. Black sheep are the opposite group whose idiosyncratic tastes make recommendations nearly impossible. Although this is a failure of the recommender system, non-electronic recommenders also have great problems in these cases, so black sheep is an acceptable failure.
- ✓ Shilling attack: In a recommendation system where everyone can give the ratings, people may give lots of positive ratings for their own items and negative ratings for their competitors. It is often necessary for the collaborative filtering systems to introduce precautions to discourage such kind of manipulations.
- ✓ The new item problem: Items that are newly inserted in the items' pool will not be recommended to the users as they have no ratings than could be used to compute the neighborhood correlation with the other items. This has for consequence that newly items never get recommended till they are rated.

ii) Content based filtering

Content based filtering works with profiles of users that are created at the beginning. A profile has information about the user and his taste. Taste is based on how users rated items. When creating a profile, the recommender system makes a survey to get initial information about the user in order to avoid the new user problem (Meyer, et al., 2009). In the recommendation process, the engine compares the items that were already positively rated by the user with the ones he didn't rate and looks for similarities. Those items that are mostly similar to the positively rated ones will be recommended to the user.

There are different algorithms of measuring similarities among items in database and their user profile. One of such approaches is the cosine similarity (Huiyi et al., 2008). Another widely used algorithm is the term frequency inverse document frequency representation also known as vector space representation.

Content- based filtering technique has the following advantages;

- User independence: Content based recommenders exploit solely ratings provided by the active user to build their own profile. Instead, collaborative filtering methods need ratings from other users in order to find the “nearest neighbors” of the active user, i.e., users that have similar tastes since they rated the same items similarly. Then, only the items that are most liked by the neighbors of the active user will be recommended;
- Transparency: Explanations on how the recommender system works can be provided by explicitly listing content features or descriptions that caused an item to occur in the list of recommendations. Those features are indicators to consult in order to decide whether to trust a recommendation. Conversely, collaborative systems are black boxes since the only explanation for an item recommendation is that unknown users with similar tastes liked that item;
- New item: Content based recommenders are capable of recommending items not yet rated by any user. As a consequence, they do not suffer from the first-rater problem, which affects collaborative recommenders which rely solely on users’ preferences to make recommendations. Therefore, until the new item is rated by a substantial number of users, the system would not be able to recommend it.

Although not a complex technique, content based filtering technique often suffer from these three problems (Giovanni, 2010);

- Limited content analysis: Content based techniques have a natural limit in the number and type of features that are associated, whether automatically or manually, with the objects they recommend. Domain knowledge is often needed, e.g., for movie recommendations the system needs to know the actors and directors, and sometimes, domain Ontologies are also needed. No content based recommendation system can provide suitable suggestions if the analyzed content does not contain enough information to discriminate items the user likes from items the user does not like. Some representations capture only certain aspects of the content, but there are many others that would influence a user's experience. For instance, often there is not enough information in the word frequency to model the user interests in jokes or poems, while techniques for affective computing would be most appropriate. Again, for Web pages, feature extraction techniques from text completely ignore aesthetic qualities and additional multimedia information. To sum up, both automatic and manually assignment of features to items could not be sufficient to define distinguishing aspects of items that turn out to be necessary for the elicitation of user interests.
- Over specialization: Content based recommenders have no inherent method for finding something unexpected. The system suggests items whose scores are high when matched against the user profile; hence the user is going to be recommended items similar to those already rated. This drawback is also called serendipity

problem to highlight the tendency of the content based systems to produce recommendations with a limited degree of novelty. To give an example, when a user has only rated movies directed by Stanley Kubrick, she will be recommended just that kind of movies. A “perfect” content-based technique would rarely find anything novel, limiting the range of applications for which it would be useful.

- New user: Enough ratings have to be collected before a content-based recommender system can really understand user preferences and provide accurate recommendations. Therefore, when few ratings are available, as for a new user, the system will not be able to provide reliable recommendations.

iii) Hybrid approaches

The motive for hybrid recommendation is the opportunity to achieve a better accuracy (Burke R, 2002). Hybrid methods combine other approaches increasing the efficiency of recommender systems. Using hybrid approaches can avoid some limitations and problems of pure recommender systems (Adomavicius et al., 2005) like the cold start problem in collaborative filtering; by enabling the results to be weighted initially towards content based filtering, then shifting the weight towards collaborative filtering as the available user data matures. The combination of two or more approaches proceeds in different ways (Tuzhilin et al., 2005);

- (i) Creating a unified recommender system that brings all other approaches together.
- (ii) Utilizing some rules of one or more approaches into a different approach and vice versa.
- (iii) Separate implementation of algorithms and joining results.

(iv) Developing one model that applies the characteristics of both methods.

The following list describes several hybridization techniques that come into consideration to when merging different recommendation approaches (Burke et al., 2007);

- **Weighted Hybridization**

Perhaps the most straightforward architecture for a hybrid system is a weighted one. It is based on the idea of deriving recommendations by combining the results (predictions) computed by individual recommenders (Burke et al., 2007). Typically, empiric means are used to determine the best weights for each component. Note that content based recommenders are able to make prediction on any item, but collaborative recommender can only score an item if there are peer users who have rated it.

- **Mixed Hybridization**

In many domains it is infeasible to receive an item score by both recommenders, because either rating matrix or content spaces are too sparse. Mixed hybridization techniques generate an independent set of recommendations for each component, and join the ranked candidates before being shown to the user. However, merging the predicted items of both recommenders makes it difficult to evaluate the improvement about the individual components (Burke et al., 2007).

- **Switching Hybridization**

Some hybrid systems consist of more than two recommendation components with different underlying collaborative filtering and or content based filtering approaches. Often recommenders are ordered, and if the first one cannot produce a recommendation with high confidence, then the next one are tried, and so on. On the contrary, other

switching hybrids might select single recommenders according to the type of user or item. However, this method assumes that some reliable switching criterion is available (Burke et al., 2007).

- **Feature Combination**

Systems that follow the feature combination approach only employ one recommendation component, which is supported by a second passive component. Instead of processing the features of the contributing component separately, they are injected into the algorithm of the actual recommender (Burke et al., 2007).

- **Feature Augmentation**

The strategy of feature augmentation is similar in some ways to feature combination. But instead of using raw features from the contributing domain, feature augmentation hybrid support their actual recommender with features passed through the contributing recommender. Usually, feature augmentation recommenders are employed when there is a well engineered primary component that requires additional knowledge sources. Due to the fact that most applications expect recommendations in real time, augmentation is usually done offline. In general, feature augmentation hybrids are superior to feature combination methods, because they add a smaller number of features to the primary recommender (Burke et al., 2007).

- **Cascade Hybridization**

The concept of a cascade hybrid is akin to feature augmentation techniques. However, cascade models make candidate selection exclusively with the primary recommender (Burke et al., 2007), and employ the secondary recommender simply to refine item scores. For example, items that were equally scored by the main component might be re-ranked employing the secondary component.

- **Meta Level Hybridization**

This kind of hybrids employs a model learned by the contributing recommender as input for the actual one. Although the general schematic of Meta level hybrids reminds on feature augmentation techniques, there exists a significant difference between both approaches. Instead of supplying the actual recommender with additional features, a Meta level contributing recommender provides a completely new recommendation space (Burke et al., 2007). However, it is not always necessarily feasible to produce a model that fits the recommendation logic of the primary component.

- b) Modern Approaches**

There are modern recommender systems that have been designed to overcome some of the challenges of traditional recommender approaches. Most of these modern approaches are based on knowledge based filtering techniques. Compared to traditional approaches, knowledge based recommendations (Felfernig et al., 2010) does not rely on item ratings and textual item descriptions but in deep knowledge about the offered items. Such deep knowledge (Felfernig, et al., 2006) describes an item in more detail thus allowing different recommendation approaches. Knowledge based recommendations relies on a set

of rules (constraints) and a set of items. Depending on the given user requirements, the rules or constraints describe which items have to be recommended. The user articulates his or her requirements in terms of item property specifications which are internally represented in terms of rules or constraints; users enter their preference and receive recommendation based on the interpretation of a set of rules or constraints. Some of the modern approaches include;

i) Context aware approaches

These are recommender systems that give recommendations to users based on the information about the user environment and details of the user situation. An example is the geographical positioning system (GPS) that utilizes the COMPASS; a mobile device application that gives recommendations to users based on the location and interest of the user. They are mostly used in the tourism sector (Koolwaaij et al., 2004).

ii) Semantic based approaches

According to Elgohary, (Elgohary et al., 2010) these recommender systems incorporate semantic knowledge in their processes in order to improve recommendations quality, they employ a concept based approach to improve the user profile presentation and use standard vocabulary and ontology languages like Web Ontology Language. Ontology helps recommender systems understand how some terms relate to each other. Example, movie has a director and actor, both director and actor is a person.

iii) Cross domain based approaches

User similarities computed domain dependent. An engine creates local neighborhoods for each user according to domains, computed similarity values are sent for overall similarity

computation. The recommender system determines overall similarity, creates overall neighborhoods and makes predictions and recommendations (Ricci et al., 2007).

iv) Peer to peer approaches

These are decentralized recommender systems. Each peer can relate to a group of other peers with same interests and get recommendations from the users of that group. Recommendations can also be given based on the history of a peer. Decentralization solves the scalability problem (Ricci et al., 2010).

v) Cross Lingual approaches

These are recommender systems that let users receive recommendations to items that have description in languages they don't speak or understand. Cross lingual recommender systems break the barriers and give users an opportunity to look for items and information in other languages (Lops et al., 2010).

2.1.7. Anatomy of a Recommender System

The key component of a recommender system is *data*. Usually there are two types of data; input data that is provided by a user and is directly related to the user for whom recommendations are to be made and, the background data which is the information the system posses before the process of recommendations begin, such as previous user ratings or likes, user browsing history, user profile, items liked and already rated by other users in the neighborhood. The input data enables recommendations to be tailored to meet particular user needs. Both the input and background data serves as the basis for recommendations to users.

Once data is collected, recommender systems use *machine learning algorithms* to find similarities and affinities between items data and users' data. There are different methods available and they usually look for the distance or correlation between two vectors that is the query vector and the document vector. For this particular study cosine similarity measure will be used to find the distance between two points (query vector and the document vector) by calculating the angle between them from the origin. The smaller the distance the more similar the points are.

Lastly, *recommender logic programs* are then used to build suggestions for specific user profiles. Logic programs are programs written in a logic programming language expressing facts and rules about some problem domain. Figure 2.7 shows a simplified process of building a recommender system.

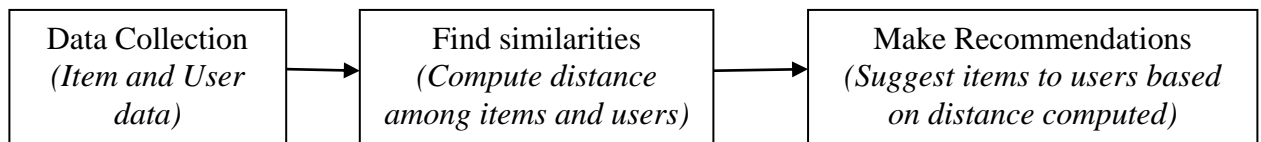


Figure 2.7: Three steps of building a Recommender System

A typical recommender system ecosystem consists of;

- A group of users
- A service
- Ability to differentiate the behavior of an individual user within the service.
- A recommendation engine that computes and stores artifacts of recommendations

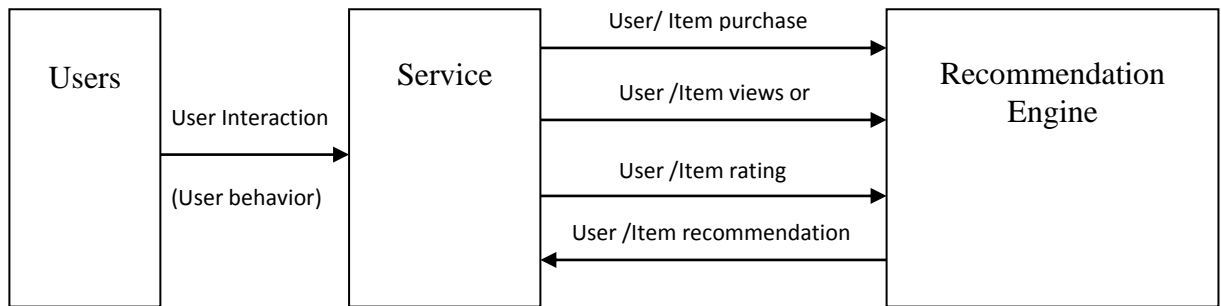


Figure 2.8: Basic Recommender System Ecosystem

2.1.8. Recommender Systems Evaluation

There are different approaches to evaluate or measure the success of recommendation algorithms. These approaches can be categorized into; *predictive accuracy metrics* such as root mean square error (RMSE), mean square error (MSE) and mean absolute error (MAE), *classification accuracy metrics* such as precision and recall, and *rank accuracy metric* such as Kendall's Tau. For this study evaluation is done using predictive accuracy metric MAE.

2.1.9. Existing Hybrid Approaches

Hybrid recommender systems combine two or more recommender systems. Depending on the hybridization approach different types of systems can be found (Burke R, 2002). There have been some works on using boosting algorithms for hybrid recommendations (Melville et al., 2002, Park et al., 2006). These works attempt to generate new synthetic ratings in order to improve recommendation quality. The personalized hybrid recommender system combines collaborative and content-based information.

Spiegel et al., 2009 proposed a framework that combines CBF, CF and demographic information for recommending information sources such as web pages or news articles. The author used home HTML pages to gather demographic information of users. The recommender system is tested on very few numbers of users and items which cannot guarantee the efficiency of the proposed system. The author does not also give an explanation on how the model is built.

Melville (Melville et al., 2002) proposed a model in which content-based algorithm is used to enhance the existing user data then the collaborative filtering is used for rating prediction. But fails to justify how both approaches combined improves prediction accuracy. Another researcher (Pazzani, 1999) used a number of collaborative filtering algorithms such as Singular Value Decomposition (SVD), Asymmetric Factor Model and neighborhood based approaches to build a recommender system. The author shows that linearly combining these algorithms increases the accuracy of prediction, but the use of all these models leads to significant increase in training time. Basu et al., 1998 used Ripper, a rule induction system, to learn a function that takes a user and movie and predicts whether the movie will be liked or disliked. They combine collaborative and content information, by creating features such as comedies liked by user and users who liked movies of genre X. They however do not show how that approach improved recommendation quality.

Several other hybrid approaches are based on traditional CF, but also maintain a content-based profile for each user. These content-based profiles, rather than co-rated items, are used to find similar users. In Pazzani's approach, each user profile is represented by a vector of weighted words derived from positive training examples using the Winnow

algorithm. Predictions are made by applying CF directly to the matrix of user profiles as opposed to the user ratings matrix. An alternative approach by; Fab (Marko et al., 1997) uses relevance feedback to simultaneously mold a personal filter along with a communal topic filter. Documents are initially ranked by the topic filter and then sent to a user's personal filter. The user's relevance feedback is used to modify both the personal filter and the originating topic filter. Good (Good et al., 1999) used collaborative filtering along with a number of personalized information filtering agents. Predictions for a user are made by applying CF on the set of other users and the active user's personalized agents. The proposed hybrid approach adapts some interesting features of the above systems; the use of collaborative and content information. It however uses the VSM, tfidf and cosine similarity measure which are very simple efficient algorithms that enable item ranking based on weights. Prediction accuracy is computed by getting the deviation of the predicted rating from the actual rating. Other works on hybrid recommender systems can be found in (Jahrer et al., 2010).

2.2. Hybrid Logistics Function

There are different functions that can be used to combine predictions for better accuracy. One such function is the sigmoid function: a bounded differentiable real function that is defined for all real input values and has a non negative derivative at each point (Han et al., 1995). The sigmoid function is a logistic function often used to fit a measured psychometric function which is a special application of a generalized linear model to psychophysical data.

Sigmoid function has been successfully and largely used in artificial neural networks (Michael, 2006). Syahrulanuar (Syahrulanuar et al., 2016) used the sigmoid function as an activation function to improve performance of implementing ANN in a field programmable gate array. Historically, a common choice of activation function is the sigmoid function since it takes a real valued input and squashes it to a range between 0 and 1. Sigmoid functions are used to map inputs into their respective outputs. Michael (Michael et al., 2010) used the sigmoid function to generate rating predictions in the range of 1 to 5 using a simple output transformation. Tan (Tan et al., 2014) found that the optimal configuration for their controller could be achieved using the sigmoid function..

In our paper we use the logistic sigmoid function due to its successes to calculate the predicted ratings at the hybrid node; the single predictions from content-based and collaborative filtering methods are related using a sigmoid function to produce single prediction value at the hybrid node.

2.3. Information Retrieval (IR)

IR is a special research field of computer science that emerged in the 1970's. It addresses the human need to automatically find or filter relevant text documents from a potentially huge collection of managed documents. For this purpose the user is expected to describe its information interest via a query; usually a search term. A related IR system is meant to return many (and if possible all) but only those documents that comply with the user's corresponding information interest. Traditionally IR systems rely on word statistics regarding managed documents. In this context, the quality of a match between a user's query and a document is determined via the frequency of certain words in the document.

Modern IR systems perform document ranking, which means that the quality of a match between a query q and a managed document d is accessed via a similarity function sc . sc returns a real valued number ≥ 0 , the larger $sc(q, d)$ the higher the degree of similarity. If $sc(q, d) = 0$ then q and d are considered completely dissimilar. At query time, the IR system efficiently retrieves those documents d_i from the collection, for which $sc(q, d_i) > 0$ hold while ignoring others. The retrieved documents are then ranked on the basis of decreasing sc values. Typically, the k best ranked documents are returned to the user (e.g., with $k = 10$). There are numerous methods (Baldi et al., 2003) to compute $sc(q, d)$. One of the oldest and widest spread frameworks for this is Salton's vector space model (VSM). When using the VSM, q and d are transformed into vectors $v(q)$ and $v(d)$ of a real valued n -dimensional vector space V , which holds as many dimensions as there exist unique terms when considering all managed documents. There also exist different ways to do this within the VSM, this thesis deals with the most popular one; the TFIDF.

2.3.1. The Vector Space Model

The vector space model (Baeza et al., 1999) is a standard algebraic model commonly used in information retrieval (IR). It treats a textual document as a bag of words, disregarding grammar and even word order. It represents both documents and queries by term sets and compares global similarities between documents and queries. It typically uses TFIDF (or a variant weighting scheme) to weigh the terms. Each document is represented as a vector of TFIDF weights. Queries are also considered as documents. Cosine similarity is used to compute similarity between document vectors and the query vector. Large similarity indicates high relevancy of documents with respect to the query. The term frequency

$TF_{t,d}$ of term t in document d is defined as the number of times that a term t occurs in a document d . It positively contributes to the relevance of d to t . The inverse document frequency IDF_t of term t measures the rarity of t in a given corpus. If t is rare, then the documents containing t are more relevant to t . IDF_t is obtained by dividing N by DF_t and then taking the logarithm of that quotient, where N is the total number of documents and DF_t is the document frequency of t or the number of documents containing t .

TFIDF weight increases with the number of occurrences within a document (term frequency) and also increases with the rarity of a term in a collection (inverse document frequency)

Cosine similarity is a standard measure estimating document similarity in the vector space model. It corresponds to the cosine of the angle between two vectors; document vector and query vector. Cosine similarity has the effect of normalizing the length of documents. The cosine measure normalizes the result of the product of the document vector and the query vector by considering their length. This prevents larger vectors from producing higher scores only because they have a bigger chance of containing similar terms.

2.3.2. Term Frequency Inverse Document Frequency

In 1972, Karen Sparck Jones published in the Journal of Documentation a paper called “A statistical interpretation of term specificity and its application in retrieval” (Sparck, 1972). The measure of term specificity first proposed in that paper later became known as inverse document frequency, or IDF; it is based on counting the number of documents in the collection being searched which contain the term in question. The intuition was that a query term which occurs in many documents is not a good

discriminator, and should be given less weight than one which occurs in few documents, and the measure was a heuristic implementation of this intuition. The intuition and the measure associated with it, proved to be a giant leap in the field of information retrieval. Coupled with TF (the frequency of the term in the document itself, in this case, the more the better), it found its way into almost every term weighting scheme.

The class of weighting schemes known generically as TFIDF, which involve multiplying the IDF measure by a TF measure have proved extraordinarily robust and difficult to beat, even by much more carefully worked out models and theories. It has even made its way outside of text retrieval into methods for retrieval of other media, and into language processing techniques for other purposes (Aizawa, 2003).TFIDF is the most common weighting method used to describe documents in Vector Space Model (Soucy et al., 2008) and has been successfully used in IR (Salto, 1989).TFIDF is a numerical statistic that is intended to reflect how important a word is in a document corpus (Rajaman et al., 2011).It is often used as a weighting factor in IT and text mining. Variations of the TFIDF weighting scheme are used by search engines as a central tool in scoring and ranking a documents' relevance given a user query.

The reason why the measure of inverse document frequency (IDF) is often used in combination with term frequency (TF) is that, keywords that appear in many documents are not useful indistinguishing between a relevant document and a non-relevant one. Term Frequency Inverse Document Frequency has the following advantages;it is a simple and efficient algorithm for matching words in a query to documents that are relevant to that query; encoding TFIDF is straightforward making it ideal for forming the basis for more complicated algorithms and query retrieval systems; it is a simple model based on

linear algebra; its weights are not binary enabling easy document ranking based on the weights; it allows partial matching and computing a continuous degree of similarity between queries and documents.

2.3.3. Term Mapping Approaches

Stemming Algorithms: The main purpose of stemming is to reduce different grammatical terms or words to its noun, adjective, verb etc. We can say the goal of stemming is to reduce inflected word to its root form. For example if a user types “eye” in its query but the document managed only contain the term “eyes”, then the query term “eye” would be mapped to “eyes” in related documents.

Query spell correction: Typos in search terms can also lead to no matches or false matches on the document side. Spell correction procedures assess some sort of spelling distance between document terms and search terms. Reasonably close terms from the query side and the document side are then still considered as a term match. The Jaccard index of the letter based bi-gram or tri-gram sets of the terms under consideration ((Ahmed et al., 2009).e.g., the bi-gram set of the term “eye” is $A = \{\text{“ey”}, \text{“ye”}\}$ and the bi-gram set for “eyes” is $B = \{\text{“ey”}, \text{“ye”}, \text{“es”}\}$. The Jaccard-index $J(A,B) = |A \cap B|/|A \cup B|$ is a general measure for the similarity of two sets A, B and can vary between 0 (as completely dissimilar) and 1 (as identical). For the example from above, one obtains $J(A,B) = |\{\text{“ey”}, \text{“ye”}\}|/|\{\text{“ey”}, \text{“ye”}, \text{“es”}\}| = 2/3$ which indicates a rather high degree of similarity.

Query reformulation: If a user searched for “iris” but instead, only the semantically related term “eye” occurred in managed documents, then the IR system would return no results. In this case, it may help to extend the query with words that are semantically related to the given search term. There are various approaches to determine corresponding terms to be added to a query. Semantic networks and thesauruses can be used for this purpose (Munir K, 2008) as they may describe the semantic relatedness of words of a language in graph like structure.

2.4. Summary

There are changes in information seeking behavior globally (Gavgani, 2010), many people turning to the internet to seek for the much needed information, however the amount of information available in the internet is growing rapidly and its enormous quantity makes it very difficult for users to retrieve specific data (Gauch et al., 2007). Recommender systems are promising tools to deal with these issues. However pure recommender systems have shortcomings as discussed in section 2.2, such as the problem of recommendation precision. It is therefore necessary to find ways of improving recommendation precision of existing recommendation techniques. For this reason, there is need for systems that can overcome the challenges of pure recommender systems such as poor recommendation precision and address the problem of information overload, these systems should be able to process the existing web information on one side, and help users by suggesting information that match their search, tastes and preferences on the other side.

2.5. Research gap

Many different approaches to recommender systems have been developed within the past few years, but the interest in this area still remains high due to the growing demand on practical applications which are able to provide personalized recommendation and deal with information overload. Beside recommendation precision, another key consideration in computer science is computation efficiency. Usually a recommender system needs to deal with millions of users and items, computing rating estimations in an instant or even in real time. Under the restrictions of memory and time consumption many prediction algorithms quickly reach their limit of possible manageable data volume. In order to handle large scale datasets, further improvements on information representation and recommendation modeling need to be done.

CHAPTER THREE

METHODOLOGY

3.1. Introduction

As mentioned earlier the main aim of the proposed hybrid recommendation approach is to achieve better accuracy. Recommender systems suggest items of interest to users within a system and their main purpose is to reduce information overload by estimating relevance and providing personalized recommendations to the target users. However, pure recommender systems such as collaborative and content-based recommenders have challenges as discussed in the previous section. Hybrid recommender systems combine two or more pure recommender systems to increase the efficiency of recommendation. The combination of two or more pure recommender systems to form a hybrid recommender system proceeds in different ways as enumerated in chapter 2. The hybrid approach proposed for this study implements content-based and collaborative filtering techniques separately. For content-based filtering we use item features and similarities, in collaborative filtering we use user behaviors and similarities. This research focus on the prediction of unknown user item rating in content-based, collaborative and the proposed hybrid filtering approach; predicted item rating in content-based and collaborative filtering methods are merged and ranked using the weighted hybridization technique, top items are then presented to the user as recommendations.

3.2. The Hybrid Filtering Model

Figure 3.1 shows the hybrid filtering model for the proposed hybrid approach. Items database refers to the large amounts of data available on different domains, the model

implements both CBF and CF methods separately. The two methods used (CBF and CF) complement each other and contribute to each other's effectiveness (Burke R, 2002). The hybrid approach uses the vector space model (VSM) on CBF and CF methods, tfidf and cosine similarity measure to compute relationships among items and users. As with all hybrid systems, the sequence and combination of sub components can be varied to modify the functionality and behavior according to the computational requirements of the system. In this case collaborative filtering and content-based filtering methods are used to obtain separate ratings for every item. The more than one rating for each item is merged into a single value using weighted hybridization technique. Items are ranked based on their merged scores, then, the final set of items (top K) with top scores are obtained and provided to the user as recommendations.

It is also important to note that the two types of recommendation techniques (collaborative and content-based filtering) used for this thesis can be used individually in any application domain, as we can make use of social interaction, associated media resources and decisions based on rules, but the two approaches are combined to complement each other and contribute to each other's effectiveness (Montaner et al., 2003). Usually the choice of a recommendation system approach depends purely on the type of information, which is to be integrated into the system.

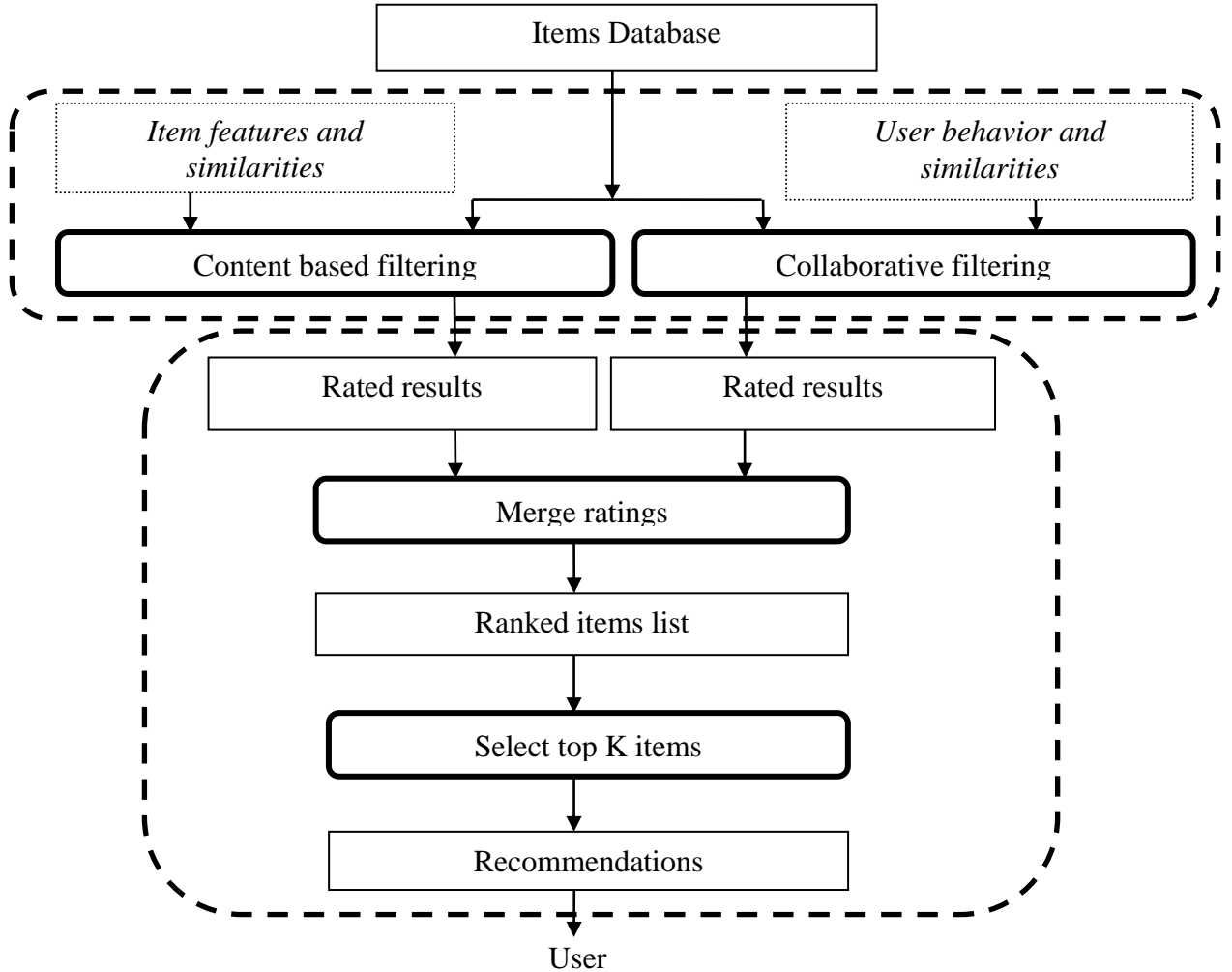


Figure 3.1: The Hybrid Filtering Model

3.3. The Vector Space Model in the Hybrid Filtering Model

The vector space model (Montaner et al., 2003) (VSM) is a standard algebraic model commonly used in information retrieval (IR). It treats a textual document as a bag of words, disregarding grammar and even word order. It represents both documents and queries by term sets and compares global similarities between documents and queries. The VSM typically uses tfidf (or a variant weighting scheme) to weight the terms. Then each document is represented as a vector of tfidf weights. Queries are also considered as

documents. Cosine similarity is used to compute similarity between document vectors and the query vector. The term frequency $TF_{t,d}$ of term t in document d is defined as the number of times that a term t occurs in a document d . Note that;

$$TF_{t,d} = 1 \text{ if } t \text{ exists in } d \quad (1)$$

$$TF_{t,d} = 0 \text{ if } t \text{ does not exist in } d \quad (2)$$

It positively contributes to the relevance of d to t . The inverse document frequency IDF_t of term t measures the rarity of t in a given corpus. If t is rare, then the documents containing t are more relevant to t . IDF_t is obtained by dividing N by DF_t and then taking the logarithm of that quotient, where N is the total number of documents and DF_t is the document frequency of t or the number of documents containing t . Formally;

$$IDF_t = \frac{\log_{10} N}{DF_t} \quad (3)$$

The TFIDF value of a term is commonly defined as the product of its TF and IDF values.

$$TFIDF_{t,d} = TF_{t,d} \times IDF_t \quad (4)$$

The TFIDF weight W , for each term in a document d is given by;

$$W_{t,d} = 1 \times \frac{\log_{10} N}{DF_t} \quad (5)$$

Generally;

$$W_{t,d} = 1 \times \frac{\log_{10} N}{DF_t} \text{ if } TF_{t,d} > 0 \quad (6)$$

$$W_{t,d} = 0, \text{ otherwise} \quad (7)$$

The following is an example showing how the vector space retrieval model works; in a collection C, of Items, consider any five Items with the following descriptions;

Table 3.1: Sample Data for illustrating VSM retrieval process

Items	Descriptions		
Item 1	Adventure	Action	Animation
Item 2	Adventure	Action	Music
Item 3	Drama	Fantasy	Animation
Item 4	Crime	Action	Children
Item 5	Adventure	Crime	Drama

Some terms used in the descriptions appear in more than two Items, others appear in only one feed. The total number of the Items is 5, so $N = 5$. We find the IDF values for each term used in the description as shown below;

$$\text{Adventure} = \log_{10} N / DF_t = \log_{10} 5/3 = 0.2218$$

$$\text{Action} = \log_{10} N / DF_t = \log_{10} 5/3 = 0.2218$$

$$\text{Animation} = \log_{10} N / DF_t = \log_{10} 5/2 = 0.3979$$

$$\text{Music} = \log_{10} N / DF_t = \log_{10} 5/1 = 0.6990$$

$$\text{Drama} = \log_{10} N / DF_t = \log_{10} 5/2 = 0.3979$$

$$\text{Fantasy} = \log_{10} N / DF_t = \log_{10} 5/1 = 0.6990$$

$$\text{Crime} = \log_{10} N / DF_t = \log_{10} 5/2 = 0.3979$$

$$\text{Children} = \log_{10} N / DF_t = \log_{10} 5/1 = 0.6990$$

Next, we find the TF scores for all the terms used in the description as shown below

Table 3.2: TF Scores for data illustrating VSM retrieval process

	Adventure	Action	Animation	Music	Drama	Fantasy	Crime	Children
Item 1	1	1	1	0	0	0	0	0
Item 2	1	1	0	1	0	0	0	0
Item 3	0	0	1	0	1	1	0	0
Item 4	0	1	0	0	0	0	1	1
Item 5	1	0	0	0	1	0	1	0

The TF scores are multiplied by the IDF values for each term used in the description and we obtain the following matrix of Items by terms;

Table 3.3: IDF Scores for data illustrating VSM retrieval process

	Adventure	Action	Animation	Music	Drama	Fantasy	Crime	Children
Item 1	0.2218	0,2218	0.3979	0	0	0	0	0
Item 2	0.2218	0.2218	0	0.6990	0	0	0	0
Item 3	0	0	0.3979	0	0.3979	0.6990	0	0
Item 4	0	0.2218	0	0	0	0	0.3979	0.6990
Item 5	0.2218	0	0	0	0.3979	0	0.3979	0

If given a query or new user profile bearing the description; “*Adventure Action Fantasy Crime*”, to find the most relevant Items from the given five, we calculate the TFIDF vector for the query and the score for each Item in the collection C relative to the query using the cosine similarity measure. When computing the TFIDF values for the query terms, the frequency is divided by the maximum frequency (3), and then multiplied with the IDF values.

First we find the TF values of the query in each Item as shown in the table below;

Table 3.4: TF Values of the Query in each Item

	Adventure	Action	Animation	Music	Drama	Fantasy	Crime	Children
Item 1	1	1	0	0	0	0	0	0
Item 2	1	1	0	0	0	0	0	0
Item 3	0	0	0	0	0	1	0	0
Item 4	0	1	0	0	0	0	1	0
Item 5	1	0	0	0	0	0	1	0

Next, we find the weights or the **IDF** values for the query “Adventure Action Fantasy Crime” in each Item as shown in the table below;

Table 3.5: IDF Values of the query in each Item

	Adventure	Action	Animation	Music	Drama	Fantasy	Crime	Children
Item 1	0.2218	0.2218	0	0	0	0	0	0
Item 2	0.2218	0.2218	0	0	0	0	0	0
Item 3	0	0	0	0	0	0.6990	0	0
Item 4	0	0.2218	0	0	0	0	0.3979	0
Item 5	0.2218	0	0	0	0	0	0.3979	0

The TFIDF weights of the query; “Adventure Action Fantasy Crime” in each Item is computed as shown in the table below;

Table 3.6: TFIDF Values of the Query in each Item

	Adventure	Action	Fantasy	Crime	TFIDF weights of query
Item 1	$1/3 * 0.2218$	$1/3 * 0.2218$	0	0	0.1479
Item 2	$1/3 * 0.2218$	$1/3 * 0.2218$	0	0	0.1479
Item 3	0	0	$1/3 * 0.6690$	0	0.2330
Item 4	0	$1/3 * 0.2218$	0	$1/3 * 0.3979$	0.2066
Item 5	$1/3 * 0.2218$	0	0	$1/3 * 0.3979$	0.2066

We then calculate the Length of each Item and query as shown below;

$$\text{Item 1: } \sqrt{0.2218^2 + 0.2218^2 + 0.3979^2} = 0.8415$$

$$\text{Item 2: } \sqrt{0.2218^2 + 0.2218^2 + 0.6990^2} = 1.1426$$

$$\text{Item 3: } \sqrt{0.3979^2 + 0.3979^2 + 0.6990^2} = 1.4948$$

$$\text{Item 4: } \sqrt{0.2218^2 + 0.3979^2 + 0.6990^2} = 1.3187$$

$$\text{Item 1: } \sqrt{0.2218^2 + 0.3979^2 + 0.3979^2} = 1.0176$$

$$\text{Query: } \sqrt{0.1479^2 + 0.1479^2 + 0.2330^2 + 0.2066^2 + 0.2066^2} = 0.9420$$

Finally we compute the similarity values between each Item and the query as shown below;

$$\text{CosSim(Item 1, Query)} = (0.2218 * 0.2218) + (0.2218 * 0.2218) / 0.8415 * 0.9420 = 0.1241$$

$$\text{CosSim(Item 2, Query)} = (0.2218 * 0.2218) + (0.2218 * 0.2218) / 1.1426 * 0.9420 = 0.0914$$

$$\text{CosSim(Item 3, Query)} = (0.6990 * 0.6990) / 1.4948 * 0.9420 = 0.3470$$

$$\text{CosSim(Item 4, Query)} = (0.2218 * 0.2218) + (0.3979 * 0.3979) / 1.3187 * 0.9420 = 0.1671$$

$$\text{CosSim(Item 5, Query)} = (0.2218 * 0.2218) + (0.3979 * 0.3979) / 1.0176 * 0.9420 = 0.2165$$

According to the similarity values the final order of relevance in which the Items are presented as recommendations to the user query will be; Item 3, Item 5, Item 4, Item 1, Item 2. Item 3 being the most relevant and Item 2 the least relevant.

3.3.1. The Vector Space Model in Content based filtering

Suppose a user profile is denoted by U and item profile by I. TF_{ij} is the number of times the term t_i occurs in item $I_j \in I$, and the inverse document frequency of a term $t_i \in I_j \in I$ is calculated as;

$$IDF_i = \log_{10} I \div DF_i \quad (8)$$

Where DF_i is equal to the number of items containing t_i and I is equal to the total number of items being considered. Therefore;

$$TFIDF = TF_{i,j} \times IDF_i \quad (9)$$

The TFIDF of each term is then calculated, and the vector of each user profile and item profiles are constructed based on their included terms. These vectors have the same length, so the similarity of these profiles can be calculated as;

$$Sim(U,I) = \frac{U \cdot I}{|U| \times |I|} = \frac{\sum_1^n (tfidf_U \times tfidf_I)}{\sqrt{\sum_1^n (tfidf_U^2) + \sum_1^n (tfidf_I^2)}} \quad (10)$$

The resulting similarity should range between from 0 to 1. If $Sim(U,I) = 0$, then the two profiles are independent and if $Sim(U,I) > 0$, the profiles have some similarity. Information about a set of items with similar rating patterns compared to the item under consideration is the basis for predicting the rating a user U would give an item I . The prediction formula is;

$$Pred(U, I) = \frac{\sum_1^n similarity(U, I) \times r_{U,ni}}{\sum_1^n |similarity(U, I)|} \quad (11)$$

Normally, the predicted rating of a user U for an item I in CBF is the average rating of the user on items viewed, therefore equation 11 can also be written as;

$$r'_{U,I}|CBF = \frac{\sum_1^n \text{similarity}(U,I) \times r_{U,ni}}{\sum_1^n |\text{similarity}(U,I)|} \quad (12)$$

Where $r_{U,I}$, is the average rating of a user U on items already is viewed (ni), and $r'_{U,I}|CBF$ is the predicted rating of a user on an item in CBF. The following is an example showing how to compute rating prediction in content-based filtering. The following table displays a User - Item rating matrix.

Table 3.7: Use - Item Rating Matrix for CBF illustration

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	5	4	-	4	-
User 2	5	3	4	-	5
User 3	3	4	-	2	-
User 4	4	-	3	1	3

From the table above we can also derive User – Item Similarity Matrix as displayed in the following table. Note that User – Item Similarity Matrix is computed based on User and item features and attributes.

Table 3.8: User - Item Similarity Matrix

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0.3	0.2	0.2	0.2	0.1
User 2	0.3	0.1	0.1	0.3	0.3
User 3	0.2	0.2	0.1	0.2	0.1
User 4	0.1	0.2	0.1	0.1	0.3

How do we predict the Items that will be recommended to a user like **User 2** or **User 4** who has read or liked **Item 3**? We calculate the similarity between **Item 3** and other items in the system using the item to item cosine similarity measure and obtain the following matrix;

Table 3.9: Item - Item Similarity Matrix

	Item1	Item 2	Item 3	Item 4	Item 5
Item 1	1	0.2	0.1	0.2	0.5
Item 2	0.2	1	0.8	0.5	0.8
Item 3	0.1	0.8	1	0.5	0.7
Item 4	0.2	0.5	0.5	1	0.8

The following table displays similarity scores between Item 3 and other Items;

Table 3.10: Item Similarity Scores

	Item 3
Item 1	0.1
Item 2	0.8
Item 4	0.5
Item 5	0.7

The top K items (e.g. if K = 3) similar to Item 3 are Item 2, Item 5 and Item 4. User 2 will receive Item 4 and Item 5 as recommendations and User 4 will receive Item 2 and Item 5 and in that order as recommendations if given k=2.

The computation of predicted rating a User 3 would give an Item 3 proceeds as follows from Eq.12; first we compute the average rating of User 3;

$$r_{u3,i3} = \frac{3+4+2}{3} = 3$$

$$r'_{u3,i3}|CBF = \frac{(0.1 \times 3) + (0.8 \times 3) + (0.5 \times 3) + (0.7 \times 3)}{|0.1| + |0.8| + |0.5| + |0.7|}$$

$$r'_{u3,i3}|CBF = \frac{6.3}{2.1} = 3$$

Similarly, the predicted rating a User 1 would give an Item 3 proceeds as follows; first we get the average rating of User 1 on items viewed;

$$r_{u1,i3} = \frac{5+4+4}{3} = 4.333 \approx 4$$

$$r'_{u1,i3}|CBF = \frac{(0.1 \times 4) + (0.8 \times 4) + (0.5 \times 4) + (0.7 \times 4)}{|0.1| + |0.8| + |0.5| + |0.7|}$$

$$r'_{u1,i3}|CBF = \frac{8.4}{2.1} = 4$$

As stated earlier, the predicted rating of a user u for an item i in CBF is usually the average rating of the user on items viewed, this is evident from the computations above.

3.3.2. The Vector Space Model in Collaborative filtering

The user profiles are represented as both documents and queries in an n-dimensional matrix. The weight for each term t in a user profile p is given by: $W_{i,j} = TF_{i,j} \times IDF_i$ which can also be written as;

$$W_{i,j} = TF_{i,j} \times \frac{\log_{10}P}{p_i} \tag{13}$$

$$IDF_i = \frac{\log_{10}P}{p_i} \tag{14}$$

Where, $TF_{i,j}$ is the frequency of a term t in a profile p , P is the total number of profiles, p_i is the total number of profiles containing term t and $W_{i,j}$ is the weight of the i^{th} term in a profile j . The similarity between user U_i and user U_j is calculated using cosine similarity measure. The equation for calculating the similarity is as follows;

$$\text{Sim}(U_i, U_j) = \frac{U_i \cdot U_j}{|U_i| \times |U_j|} = \frac{\sum_1^n (\text{tfidf}_{U_i} \times \text{tfidf}_{U_j})}{\sqrt{\sum_1^n (\text{tfidf}_{U_i}^2) + \sum_1^n (\text{tfidf}_{U_j}^2)}} \quad (15)$$

Again the resulting similarity should range between from 0 to 1. If $\text{Sim}(U_i, U_j) = 0$, then the two users are independent and if $\text{Sim}(U_i, U_j) = 1$, the users are similar. The information about a set of users with a similar rating behavior compared to the current user is the basis for predicting the rating a user U_i would give an item he or she has not rated. Based on the nearest neighbor of user U_i it is easy to determine the prediction of user U_i .

$$\text{Pred}(U_i, I) = \frac{\sum_1^n \text{similarity}(U_i, U_j) \times r_{U_j, I}}{\sum_1^n |\text{similarity}(U_i, U_j)|} \quad (16)$$

Where, U_j is U_i 's neighbor and $r_{U_j, I}$ is the rating of U_j on the given item. Also, given that the predicted rating of a user U_i on an item I in CF is given as $r'_{u,i}|CF$, equation 16 can therefore be written as:

$$r'_{u,i}|CF = \frac{\sum_1^n \text{similarity}(U_i, U_j) \times r_{U_j, I}}{\sum_1^n |\text{similarity}(U_i, U_j)|} \quad (17)$$

From the table below; using sample data for four users and five items rated or not yet rated by the users we explain the principles of the neighborhood approach using the cosine similarity measure.

Table 3.11: User - Item Rating Matrix for CF illustration

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	5	4	-	4	-
User 2	5	3	4	-	5
User 3	3	4	-	2	-
User 4	4	-	3	1	3

From the above table we compute user-user similarity to get the following User – User Similarity matrix.

Table 3.12: User - User Similarity Matrix

	User 1	User 2	User 3	User 4
User 1	1	0.8	0.6	0.5
User 2	0.8	1	0.4	0.7
User 3	0.6	0.4	1	0.4
User 4	0.5	0.7	0.4	1

To get the prediction of User 4 on Item 2 as an example, we proceed as follows from Eq.

17;

$$r'_{u4,i2}|CF = \frac{(0.5 \times 4) + (0.7 \times 3) + (0.4 \times 4)}{|0.5| + |0.8| + |0.4|}$$

$$r'_{u4,i2}|CF = \frac{5.7}{1.6} = 3.6 \approx 4$$

Similarly, to get the prediction a User 3 would give an Item 5, we proceed as follows;

$$r'_{u3,i5}|CF = \frac{(0.6 \times 0) + (0.4 \times 5) + (0.4 \times 3)}{|0.6| + |0.4| + |0.4|}$$

$$r'_{u3,i5}|CF = \frac{3.2}{1.4} = 2.3 \approx 2$$

A user usually has many neighbors which can be consulted for prediction rating. However most neighborhood based recommender systems limit the number of accounted neighbors because neighbors based on a small number of overlapping items tend to be a bad predictor (Agarwal et al., 2013 and Huang et al., 2012), hence the k-nearest-neighbor algorithm that only considers the top k similar users is used.

3.4. The Hybridization Process

There exist different methods for combining content-based and collaborative filtering techniques, but not all of them will lead to the same prediction accuracy. As stated earlier our hybrid filtering model combines a CBF and CF technique which uses user-item matrix and user-user matrix respectively. Figure 3.2 displays the weighted hybrid architecture.

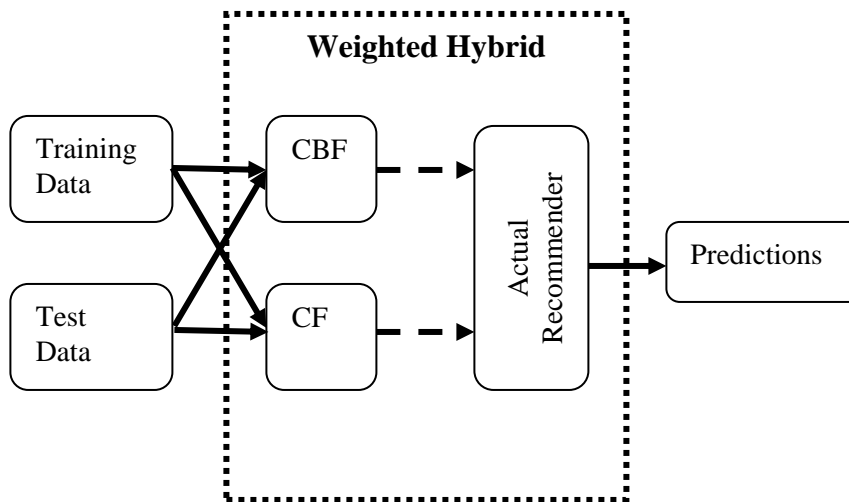


Figure 3.2: Weighted Hybrid Architecture

The hybrid filtering model is based on the idea of deriving recommendation items by combining predictions computed by each individual recommenders CBF (Eq. 12) and CF (Eq. 17), here the separate scores of an individual recommender on an item $i \in I$ recommended to a user $u \in U$ are merged into a single value (Eq. 18). Table 3.13 shows the extended User - Item and User - User matrix with sample tfidf and cosine similarity scores among users and it

Table 3.13: Extended User - Item, User - User Matrix

		Item					User profile-Attribute TFIDF					User-User cosine similarity					
		i_1	i_2	i_3	...	i_m	a_1	a_2	a_3	...	a_l	u_1	u_2	u_3	...	u_n	
User	u_1	-	-	4	...	3	0.04	0	0	...	0	1	0.1	0	...	0.2	
	u_2	4	2	-	...	5	0	0.01	0.02	...	0.02	0.1	1	0.1	...	0	
	u_3	-	-	3	...	-	0.04	0	0	...	0.02	0	0.1	1	...	0	

	u_n	-	3	-	...	-	0.04	0.01	0.02	...	0.02	0.2	0	0	...	1	
Item- Attribute tfidf	a_1	0.0	0.0	0.0	...	0.0											
	a_2	0.0	0.01	0.0	...	0.01											
	a_3	0.02	0.02	0.0	...	0.0											
											
	a_l	0.02	0.0	0.0	...	0.0											
User- Item cosine similarity	u_1	0.3	0.2	0.1	...	0.1											
	u_2	0.3	0.0	0.0	...	0.0											
	u_3	0.1	0.5	0.3	...	0.4											
											
	u_n	0.0	0.2	0.4	...	0.2											

Rated item

Unrated item

To take into account the difference in the contribution of each predictor in the final rating prediction, each predictor is assigned a parameter. Such that the resulting rating prediction $r'_{u,i}|_{HF}$ of a user u on an item i from HF is computed as follows;

$$r'_{u,i}|_{HF} = Xr'_{u,i}|_{CBF} + Yr'_{u,i}|_{CF} \tag{18}$$

Where $Xr'_{u,i}|CBF$ and $Yr'_{u,i}|CF$ are the predicted rating of an item $i \in I$ for user $u \in U$ in CBF and CF respectively.

To compute the value for each parameter, a function $S(n)$ that gives the weight of a user's rating n ($n=|R_u|$) is used. The sigmoid function satisfies these constraints for $S(n)$.

The parameters X and Y can be computed using the sigmoid function as follows;

$$X = 1 - \frac{1}{1+e^{-n}} \quad (19)$$

$$Y = \frac{1}{1+e^{-n}} \quad (20)$$

These parameters X and Y, represent the weight confidence levels given to CBF and CF respectively. The resulting rating predictions of items from the hybrid approach are ranked based on their prediction scores, from the ranked items list the top scoring set of items (top k items) are selected and provided to the user as recommendations.

3.5. Model Evaluation Metrics

There are several metrics by which a RS can be evaluated and interpreted for accuracy: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). MAE and RMSE are known as predictive accuracy or statistical accuracy metrics because they represent how accurately a RS estimates a user's preference for an item. In our experiments, we evaluate the different datasets using MAE evaluation metrics. MAE evaluates how well the RS can predict a user's rating for an Item based on a scale from one to five stars which can be

converted to a tfidf score between zero and one. **MAE** is calculated by averaging the absolute deviation of a user's predicted score and actual score.

The formula for MAE is:
$$\mathbf{MAE} = \frac{\sum_i^n |s_i - p_i|}{n} \quad (21)$$

RMSE is calculated by finding the square root of the average squared deviations of a user's predicted score and actual score.

The formula for RMSE is:
$$\mathbf{RMSE} = \sqrt{\frac{\sum_i^n (s_i - p_i)^2}{n}} \quad (22)$$

Where in both formulas for MAE and RMSE, **n** is the total number of items, **i** is the current item, **s_i** is the actual score a user expressed for item **i**, and **p_i** is the RS's predicted score a user has for **i**.

Since RMSE squares the deviations and MAE only sums the deviations, RMSE will weight larger deviations more than MAE. In our study, we provide MAE evaluation on the RS algorithms. The smaller MAE the more accurate a RS.

CHAPTER FOUR

EXPERIMENTS AND RESULTS

4.1. Introduction

This chapter presents the experimental set up, evaluation results and discussions.

4.2. Dataset

The MovieLens (<http://www.grouplense.org>) 100k dataset was used. This data was collected by the GroupLens Research Project at the University of Minnesota during a seven-month period between 19th September 1997 and 22nd April 1998. The MovieLens is used mainly because it is publicly available and has been used in many hybrid recommender systems and therefore considered a good benchmark for this purpose. This dataset contains 943 users, 1682 movie items and 100000 ratings. Each user rates a minimum of 20 movies using integer values 1 to 5 and not all movies are rated by all users. There are 19 movie genres. A movie can belong to more than one genre. A binary value of 1 and 0 is used to indicate whether a movie belongs to a specific genre or not. The dataset is split into 5 subsets, each having (80%) training and (20%) test sets.

4.3. Experimental Setup

This experiment was done using R for Windows, the MovieLens datasets that already provides different training and test samples which exhibit perfect properties for validation is used. The two disjoint datasets (training and test sets) are used to measure the difference between predicted and actual rating. For content-based filtering we use the movie features and for collaborative filtering we use the movie ratings. In this

experiment, 5-fold cross validation was performed on sub datasets 1 to 5 provided by MovieLens 100k dataset, 80% training data and 20% test data on each sub dataset. The performance measure was done using prediction accuracy metric: Mean Absolute Error (MAE), which is used to represent how accurately a RS estimates a user's preference for an item. MAE is calculated by averaging the absolute deviation of a user's predicted score and actual score. The smaller the MAE the more precise the RS is.

$$MAE = \frac{\sum_i^n |s_i - p_i|}{n} \quad (23)$$

Where, n is the total number of items, i is the current item, s_i is the actual score a user expressed for item i , and p_i is the RS's predicted score a user has for i .

4.4. Results

Even though the 5 sub data sets used have almost the same number of users and items, they have different rating patterns therefore a standard number of users and items were used for experiment across all the datasets. Results presented here in tables and graphs are the MAE average across all the sub data sets given the specified number of users and items.

Table 4.1: MAE given 100 Items

No of Users	Filtering Methods		
	CF	CBF	HF
100	0.3686	0.3828	0.3433
350	0.3374	0.3632	0.3162
500	0.3398	0.3659	0.3161
800	0.3258	0.3555	0.3081

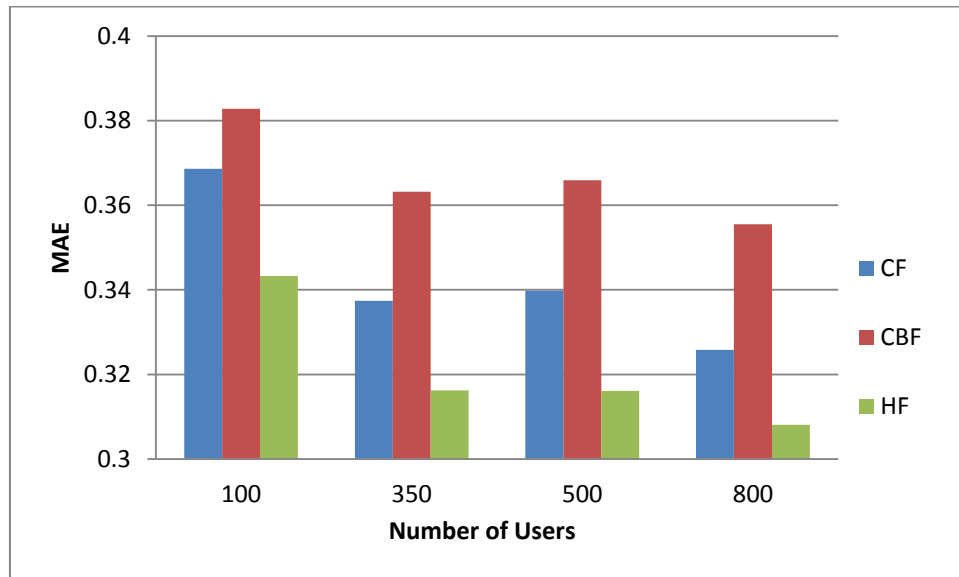


Figure 4.1: MAE given 100 Items

Table 4.1 and Figure 4.1 shows the MAE average across all the five sub datasets in the MovieLens 100K dataset; given 100 items and 100, 350, 500, 800 users respectively. The HF technique achieved a greater prediction accuracy compared to CF and CBF; this could

be due to the fact that CF and CBF complement each other, thus eliminating their challenges when implemented as pure recommender systems leading to greater achievement at the hybrid level. HF performed 7 % and 14% better than CF and CBF respectively.

Results displayed in Table 4.1 and Figure 4.1 above also reveals that prediction accuracy improves in situations where there are more users and fewer items compared to situations where the number of users is equivalent to or almost the same as the number of items.

Table 4.2: MAE given 500 Items

No of Users	Filtering Methods		
	CF	CBF	HF
100	0.3396	0.3588	0.3043
350	0.3110	0.3560	0.2998
500	0.3016	0.3544	0.2954
800	0.2971	0.3519	0.2953

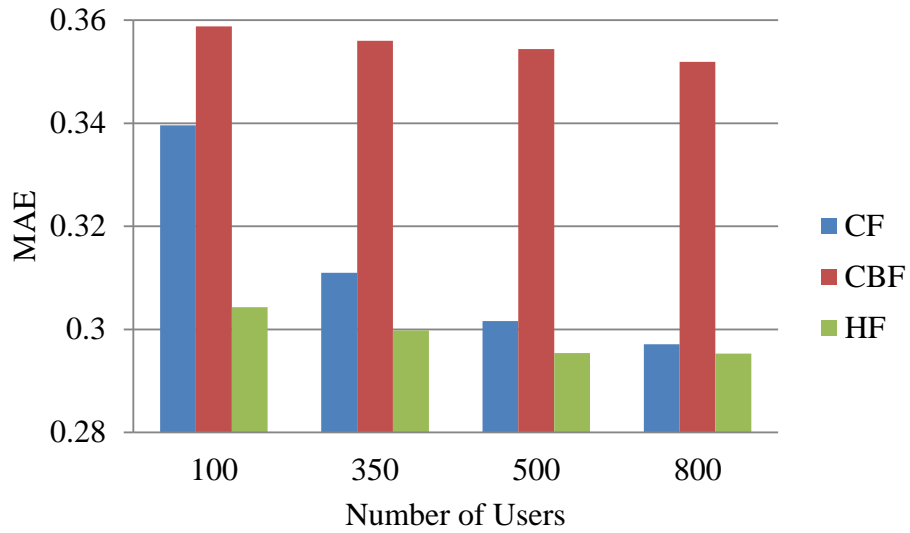


Figure 4.2: MAE given 500 Items

Table 4.2 and Figure 4.2 shows the MAE average across all the five sub datasets in the MovieLens 100K dataset, given 500 items and 100, 350, 500, 800 users respectively. Again, the HF technique achieved greater prediction accuracy than CF and CBF, HF performing 5 % and 18% better than CF and CBF respectively. The prediction accuracy of HF improves as the number of users increases, a tendency that can also be observed in CF and CBF, this proves that the two methods used complement each other perfectly well leading to better prediction accuracy at the hybrid level.

Table 4.3: MAE given 700 Items

No of Users	Filtering Methods		
	CF	CBF	HF
100	0.3326	0.3554	0.3029
350	0.3324	0.3690	0.3167
500	0.3203	0.3676	0.3122
800	0.3020	0.3564	0.2986

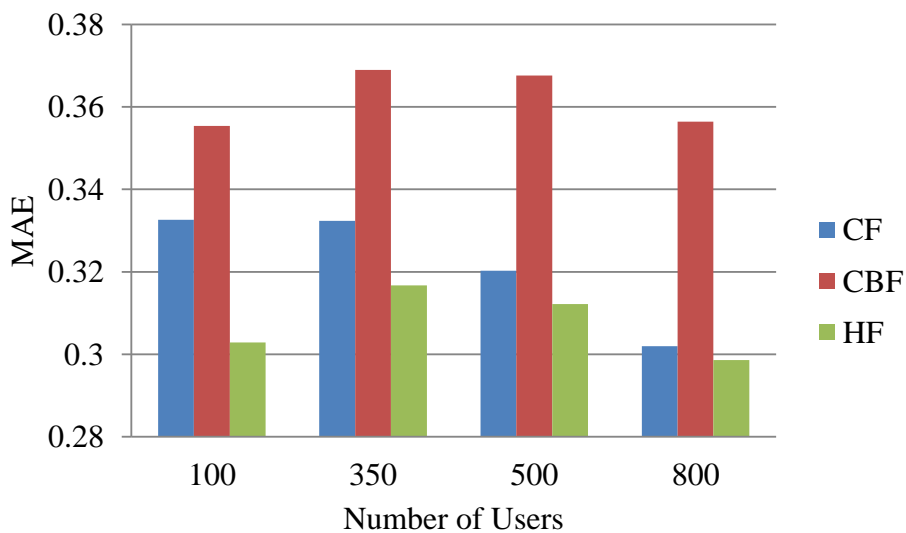


Figure 4.3: MAE given 700 Items

Table 4.3 and Figure 4.3 shows the MAE averages across all the five sub datasets in the MovieLens 100K dataset, given 700 items versus 100, 350, 500 and 800 users respectively. HF performed 5 % and 18% better than CF and CBF respectively. HF

technique achieved almost equal prediction accuracy in situations where there are fewer users many items and many users many items, revealing that the number of items contributes greatly to how accurate a recommender system can be. From the MAE averages for both CF and CBF, it is worth noting that CF contributed more towards the final prediction accuracy.

Table 4.4: MAE given 1200 Items

No of Users	Filtering Methods		
	CF	CBF	HF
100	0.3374	0.3539	0.3050
350	0.3386	0.3859	0.3245
500	0.3294	0.3745	0.3145
800	0.2997	0.3442	0.2883

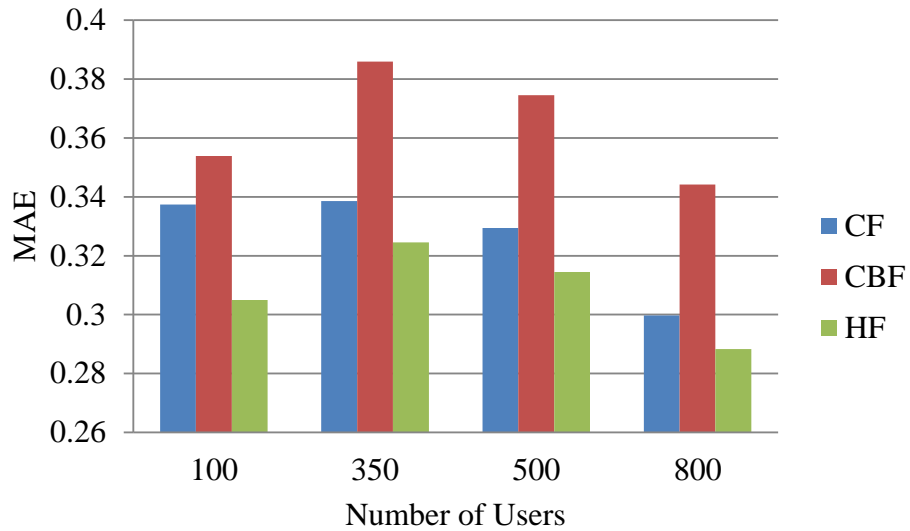


Figure 4.4: MAE given 1200 Items

Table 4.4 and Figure 4.4 shows the MAE averages across all the five sub datasets in the MovieLens 100K dataset, given 1200 items and 100, 350, 500, 800 users respectively. HF performed 6 % and 18% better than CF and CBF respectively. Although the results exhibit the same tendency, generally, prediction accuracy is poor where the number of items is equivalent to or almost the same as the number of users, this can be seen in results shown in Tables 4.1, 4.2 and 4.3 as well as in Figure 4.1, 4.2 and 4.3.

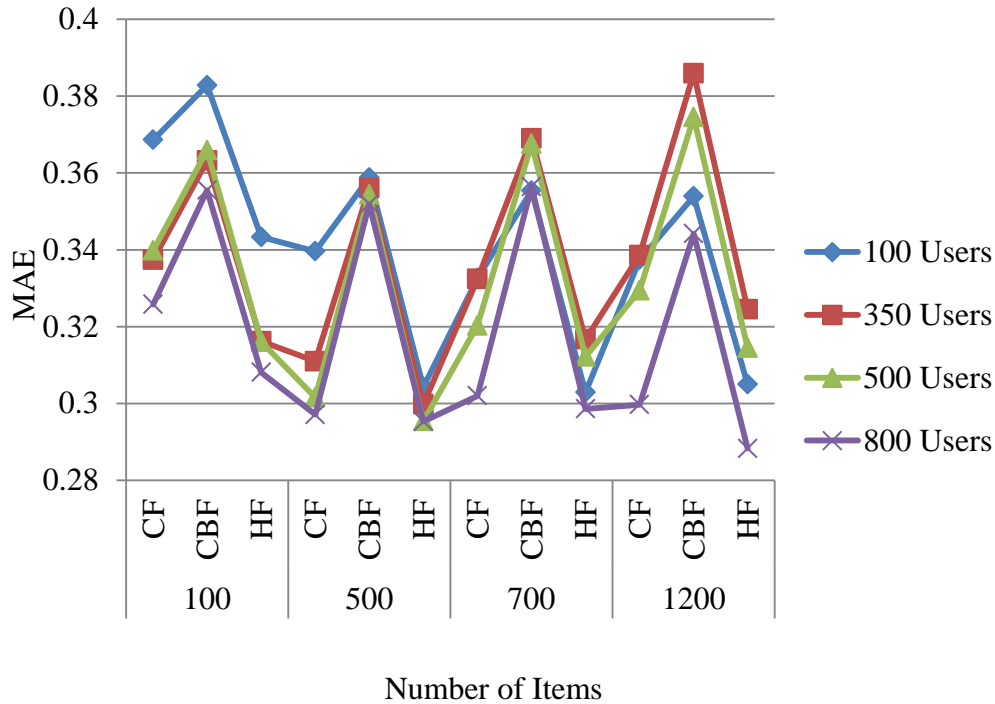


Figure 4.5: Performance Summary

4.5. Discussions

Across all of the evaluations, in Figure 4.5, results show that the hybrid filtering approach achieves better prediction accuracy than each of the traditional filtering methods (CF and CBF) implemented separately. Collaborative filtering and content-based filtering performing on average 6% and 17% worse than the hybrid approach respectively; the hybrid approach achieves an average MAE of 0.3084 whereas collaborative and content-based filtering achieve 0.3258 and 0.3622 respectively.

The best results were achieved where we had the highest number of users and items respectively (Table 4.4), where MAE for the Hybrid system is 0.2883, content-based and

collaborative methods have MAE of 0.3442 and 0.2997 respectively. Poor results were achieved where we had the least number of users and items respectively (Table 4.1), where MAE for the Hybrid system is 0.3433, content-based and collaborative methods have MAE of 0.3828 and 0.3686 respectively. From the results, it can be seen that collaborative filtering contributes greatly in the results of this approach more so where there are large numbers of items; its performance becomes better with increasing number of items and users respectively, but does not perform as well with large number of users and small number of items, this can also be the reason why collaborative filtering methods are greatly used compared to other pure filtering methods. On the other hand, content-based filtering does not make much contribution to this approach, as its performance worsens as the number of items increases and in cases where there are small number of users and items respectively. However content-based and collaborative filtering methods overcome the limitations of each other leading to greater prediction accuracy at the hybrid node.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1. Introduction

This chapter presents the summary of work done, conclusions, recommendations and possible further work in this research area.

5.2. Summary

There exists a number of different hybrid approaches in literature. In this thesis a hybrid approach that combines content-based and collaborative filtering methods is used to improve recommendation accuracy. Both methods use the effective information retrieval model the vector space model, a very simple efficient ranking algorithm, the term frequency inverse document frequency and cosine similarity measure to find the relationships among users, items and attributes. The cosine similarity measure normalizes the length of the document vector as well as the product of the document and query vectors, preventing larger vectors from producing higher scores simply because they have a bigger chance of containing similar items. The hybrid approach used the weighted hybridization technique that merged the predicted ratings of individual recommenders using the sigmoid function at the hybrid node, the sigmoid function is a logistic function often used to fit a measured psychometric function. Here it is used to map the two separate predictions from the content based approach and collaborative approach into a single prediction at the hybrid node.

The evaluation of the hybrid approach was done using real data, the MovieLens dataset in R for Windows language, its performance was measured using the accuracy metric mean absolute value.

From the experiments, the hybrid system achieved better prediction accuracy compared to a single content-based and single collaborative based recommenders; therefore it is proof that a combination of content and collaborative information helps improve recommendation precision.

5.3. Conclusions

During this research, the following objectives were achieved;

- i. To study how hybrid recommender systems work.
- ii. To design the hybrid approach for personalized recommender system.
- iii. To evaluate the hybrid approach for personalized recommender system.

In chapter 2, literature describes how hybrid recommender systems work, furthermore different types of recommender systems, their strengths and weaknesses are also discussed in the literature. Three main recommender techniques that formed the basis for our thesis; collaborative filtering, content-based filtering and hybrid filtering are also discussed into details in the literature. In chapter 3, the hybrid filtering model for the proposed hybrid approach is discussed. The adaptation of the VSM in the hybrid model is illustrated; adaptation of the VSM in both content-based and collaborative filtering techniques. The hybridization process is also presented. In chapter 4, the hybrid filtering approach is evaluated. The MovieLens 100k data set is used to evaluate the hybrid

recommender system in experiments done using R. The MovieLens data is used mainly because it is publicly available and has been used in many hybrid recommender systems and therefore considered a good benchmark for this study. The MAE prediction accuracy is used to evaluate the performance of the hybrid system; MAE measures the deviation of a user's predicted score on an item from the actual score; the smaller the deviation the greater the accuracy, the average accuracy measures across all the five sub data sets of MovieLens 100k data set, given different number of items and users are presented in chapter 4.

5.4. Recommendations

It should be noted that, the hybrid system is universal, and also because of its good performance, it can be applied to perform recommendation tasks in different recommendation domains.

5.5. Suggestions for future work

The possible future work related to this study is first to test the efficiency of this approach to other larger datasets like MovieLens 1M and 10M datasets. Secondly, to explore the possibilities of experimenting with other variants of TFIDF, similarity measures and the vector space model to see how well they perform in this kind of hybrid recommender environment.

REFERENCES

- Adomavicius G., Tuzhilin A. (2005). Toward the next generation of recommender systems: A survey of the state of the art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17, 734-749.
- Agarwal D., Chen B., Elango P., & Ramakrishnan R. (2013). Content recommendation on web portals. *Commun. ACM*, 56:92–101.
- Ahmed F., Nürnberger A., & Luca E.W.D. (2009). Revised N-gram based automatic spelling correction tool to improve retrieval effectiveness. *Res. J. Comput. Sci. Comput. Eng. Appl. (Polibits)*; 40:39–48.
- Aizawa, A. (2003). An information-theoretic perspective of tfidf measures: *Information Processing and Management*, 39, 45–65.
- Alexander F., Klaus I., Kalman S., & Peter Z. (2007). The VITA Financial Services Sales Support Environment, *AAAI*, 1692-1699.
- Baeza Y. R., & Ribeiro N. B. (1999). *Modern Information Retrieval*. Addison-Wesley.
- Baldi P., Frasconi P., & Smith P. (2003). *Modeling the Internet and the Web: Probabilistic Methods and Algorithms*, Wiley and Sons, 1; East Grinstead, UK.

- Basu C., Hirsh H., & Cohen C. (1998). Recommendation as classification: Using social and content-based information in recommendation. *In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 714–720.
- Bell R., Koren Y., & Volinsky Ch. (2007). Chasing \$1,000,000: How we won the Netflix Progress Prize. *ASA Statistical and Computing Graphics Newsletter*, 18(2):4–12.
- Burke R. D. (2002). Hybrid Recommender Systems: Survey and Experiments, User Modelling and User Adapted Interaction, 12(4), 331-370.
- Burke R., & Robin D. (2007). Hybrid Web Recommender Systems: The Adaptive Web, Methods and Strategies of Web Personalization, Springer. 4321, 377–408.
- Elgohary A., Nomir H., Sabek I., Samir M., Badawy M., & Yousri N.A. (2010). Wiki-rec: A semantic based recommender system using awaikipedia as an ontolog. *10th International Conference on Intelligent Systems Design and Applications*.
- Facebook, (2010). Pandora Lead Rise of Recommendation Engines – TIME: TIME.com. 27 May 2010. Retrieved 1 June 2015.
- Felfernig A., Gerhard H., Dietman J., & Markus Z. (2010). Developing Constraint-based Recommender. *Recommender Systems Handbook*. Springer, 187 – 121.
- Felfernig A., Jeran M., Ninaus G., Reinfrank F., & Reiterer S. (2013). Multimedia Services in Intelligent Environments. Springer; Heidelberg, Germany. *Toward the Next Generation of Recommender Systems: Applications and Research Challenges*, 81–98.

- Felfernig A., & Shchekotykhin K. (2006). Debugging User Interface Descriptions of Knowledge Based Recommender Applications. *In ACM International Conference of Intelligent User Interfaces. Sydney, Australia*, 234 – 241.
- Gauch S., Aravind C., & Alessandro M., (2007). User Profiles for Personalized Information Access: The Adaptive Web, Springer, 4321, 54–89.
- Gavvani V.Z. (2010). Health Information Need and Seeking Behavior of Patients in Developing Countries' Context; an Iranian Experience. *Proceedings of the 1st ACM International Health Informatics Symposium (IHI '10), ACM; New York, New York, NY, USA*, 575–579
- Giovanni S., Marco G., & Lops P. (2010). Content Based Recommender Systems: *Recommender Systems Handbook, Springer*, 73 -100.
- Good N., Schafer J. B., Konstan J.A., Borchers A., Sarwar B., Herlocker J., & Riedl J. (1999). Combining collaborative filtering with personal agents for better recommendations: *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, 439–446.
- Group Lens, (2014). Movie Lens [Online]. Retrieved from: <http://grouplens.org/datasets/movielens>
- Han J., & Morag C. (1995). The Influence of Sigmoid Function Parameters on the speed of back propagation Learning: *From Natural to Artificial Neural Computation. San Francisco*, 195 201.

- Huang Z., Lu X., Duan H., & Zhao C. (2012). Collaboration-based medical knowledge recommendation. *Artif. Intell. Med*, 55,13–24.
- Huiyi T., Junfei G., & Young L. (2008). E-learning Recommender System. *Proceedings of the 2008 International Conference on Computer Science and Software Engineering*, 430 – 433.
- Jahrer M., Toscher A., & Legenstein R. (2010). Combining predictions for accurate recommender systems: *Proceedings of the SIGKDD conference*. New York, NY, USA: ACM, 693-702.
- Jason M.A., Paul N.B., & Anton T. (2007). Combining Personalized Agents to Improve Content Based Recommendations, CMU, LT1, 7 –15.
- Kangas S. (2002). Collaborative Filtering and Recommender Sstems, *Research report*, 35.
- Kazienko P., & Kiewra M. (2004). Personalized Recommendation of Web Pages. *Intelligent Technologies for Inconsistent Knowledge Processing. Advanced Knowledge International. Adelaide. South Australia*, 163 -183.
- Kazienko P., & Kołodziejki P. (2005). WindOwls: Adaptive Systems for the Integration of Recommendation Methods in eCommerce, *Springer Verlag*, 218 – 224.
- Koolwaaij J., Pokraev S., Van S.M., & Koolwaaij, J. (2004). Context-aware recommendations in the mobile tourist application compass. In *Nejdl, W., and De, P.,Bra, editors, Adaptive Hypermedia*, Springer Verlag, 235–244.

- Koren, Y., (2008). Tutorial on recent progress in collaborative filtering: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys, 333–334.
- Linden G, Brent S & Jeremy Y. (2005). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing*, 7(1).
- Lops P., Marco G., & Giovanni S. (2010). Content Based Recommender Systems: *Recommender Systems Handbook*, Springer, 73 -100.
- Marko B., Yoav S. & Fab. (1997). Content-based, collaborative recommendation. *Communications of the Association for Computing Machinery*, 40 (3):66–72.
- Melville P., Mooney R., Nagarajan R. (2001). Content-boosted collaborative filtering for improved recommendations. *18th National Conference on Artificial Intelligence (AAAI-02)*, 187-193.
- Melville, P., Mooney R., & Nagarajan R. (2001). Content-boosted collaborative filtering. *In ACM SIGIR Workshop on Recommender Systems*.
- Meyer F., Candillier L., Fessant F., & Jack K. (2009). State of the art recommender systems.
- Michael J., Andreas T., & Robert L. (2010). Combining predictions for an accurate recommender system: *KDD, Washington DC, USA*.
- Michael N. (2006). Artificial Intelligence: A Guide to Intelligent Systems In Artificial Intelligence, 2nd Edition, *Addison Wesley*, 165 – 170.

- Montaner, M., Lopez, B. & De la Rosa J.L. (2003). A Taxonomy of Recommender Agents on the Internet. *Artificial Intelligence Review, Kluwer Academic Publisher, 19*, 285 – 330.
- Munir K. (2008). Ontology Assisted Query Reformulation Using the Semantic and Assertion Capabilities of OWL-DL Ontologies: *Proceedings of the 2008 International Symposium on DB Engineering Applications (IDEAS '08), ACM; New York, NY, USA*, 81–90.
- Park S. T., Pennock D., Madani O., Good N., & DeCoste D. (2006). Naïve filterbots for robust cold start recommendations: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 669-705.
- Pazzani M. J. (1999). A framework for collaborative, content based and demographic filtering. *Artificial Intelligence*, 13(5-6), 393-408.
- Pazzani, Michael J., & Daniel B. (2007). Content-Based Recommendation Systems. *The Adaptive Web, Springer Verlag, Berlin*, 4321(10), 325–341.
- Rajaraman A.; & Ullman J. D. (2011). Data Mining. *Mining of Massive Datasets (PDF)*, 1–17.
- Ricci F., Lior R., & Bracha S. (2011). Introduction to Recommender Systems Handbook. *Recommender Systems Handbook, Springer*, 1-35.
- Ricci F., Lior R., Bracha S., & Paul B.K. (2010). Recommender Systems Handbook. *Springer*.
- Ricci F., Lior R., & Bracha S. (2011). Introduction to Recommender Systems Handbook. *Recommender Systems Handbook. New York: Springer*, 1-35.

- Ricci F., & Nguyen Q.N. (2007). Acquiring and revising preferences in a critique-based mobile recommender system. *IEEE Intelligent Systems* 22(3), 22–29.
- Riedl J.T., Herlocker J.L., Konstan J.A., & Terveen L.G.(2006). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1), 5–53.
- Sadeh N.M. (2002). mCommerce: Technologies, Services and Business Models. *Wiley*.
- Salton G. (1989). Automatic Text Processing, *Addison Wesley*.
- Schafer J.B., Konstan J.A., & Riedl J. (2001). eCommerce Recommendation Applications. *Data Mining and Knowledge Discovery*, 115 – 152.
- Schafer J. B., Ben J., Dan F., Jon H., & Shilad S., (2007). Collaborative Filtering Recommender Systems. *Springer Verlag, Berlin, Germany. The Adaptive Web*, 4321(9), 291–324.
- Shanghack L., Jihoon Y., & Sung Y. P. (2007). Discovery of Hidden Similarity on Collaborative Filtering To Overcome Sparsity Problem, *Discovery Science*.
- Shardanand U. (1994). Social Information Filtering for Music Recommendation. *Massachusetts Institute of Technology*.
- Shardanand U., & Maes P. (1995). Social Information Filtering: Algorithms for Automating “Word of Mouth”. *Proc. Conf. Human Factors in Computing Systems*.
- Soucy P., & Guy W.M. (2008) Beyond TFIDF weighting for text categorization in the Vector Space Model.

- Sparck J.K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28,11 – 21.
- Spiegel S., Kunegis J., & Li F. (2009). Hydra: A hybrid recommender system, cross-linked rating and content information in CIKM-CNIKM, 75-80.
- Syahrulanuar N., Rohani A.B., Abdullah E., & Saifudin R. (2016). Two Step Implementation of Sigmoid function for Artificial Neural Network in Field Programmable Gate Array. *ARPN Journal of Engineering and Applied Sciences*, 11,4882 – 4887.
- Tan T.G., Teo J., & Patricia A. (2014). A Comparative investigation of non linear activation functions in neural controllers for search based game: *AI Engineering, Artificial Intelligence Review*, 41(1), 1- 25.
- Terveen L., Hill W., Amento B., McDonald D., & Creter J. (1997). PHOAKS: A system for sharing recommendations. *Communications of the ACM*, 59 – 62.
- Tuzhilin A., & Adomavicius G. (2005). Towards the next generation of recommender systems. A survey of the state of the art and possible extensions. *IEEE Trans Knowl Data Eng*, 17, 734 – 749.
- Werthner H., & Klein S. (1999). Information Technology and Tourism, a Challenging Relationship. *Springer*.

APPENDIX

Research Publication: <http://ijcat.com/archives/volume5/issue12/ijcatr05121006.pdf>