# A PROPOSED MEMORY-BASED COLLABORATIVE FILTERING TECHNIQUE BASED ON A NEW SIMILARITY AND MADM METHODS (CF-NSMA) FOR IMPROVING THE RECOMMENDATION ACCURACY

HAEL ABDULLAH HUSSEIN AL-BASHIRI

DOCTOR OF PHILOSOPHY

UNIVERSITI MALAYSIA PAHANG

## UNIVERSITI MALAYSIA PAHANG

**DECLARATION OF THESIS AND COPYRIGHT**

Author's Full Name : Hael Abdullah Hussein Al-Bashiri

Date of Birth : 01 January 1983

Title : A Proposed Memory-Based Collaborative Filtering Technique Based On a New Similarity and MADM Methods (CF-NSMA) For Improving the Recommendation

Academic Session : SEM 1 2018/2019

I declare that this thesis is classified as:

| | | |
|---|---|---|
| ☐ | CONFIDENTIAL | (Contains confidential information under the Official Secret Act 1997)* |
| ☐ | RESTRICTED | (Contains restricted information as specified by the organization where research was done)* |
| ☑ | OPEN ACCESS | I agree that my thesis to be published as online open access (Full Text) |

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

| | |
|---|---|
| _______________ | _______________ |
| (Student's Signature) | (Supervisor's Signature) |
| _______________ | _______________ |
| New IC/Passport Number<br>07056408<br>Date: 07/01/2019 | Name of Supervisor<br>Dr. Mansoor Abdullateef Abdulgabber<br>Date: 07/01/2019 |

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

## SUPERVISOR'S DECLARATION

We hereby declare that We have checked this thesis and in our opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Doctor of Philosophy.

(Supervisor's Signature)

Full Name : Dr. MANSOOR ABDULLATEEF ABDULGABBER
Position : SENIOR LECTURER
Date :

(Co-supervisor's Signature)

Full Name : Dr. AWANIS ROMLI
Position : SENIOR LECTURER
Date :

## STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : HAEL ABDULLAH HUSSEIN AL-BASHIRI

ID Number :

Date : 7 JANUARY 2019

# A PROPOSED MEMORY-BASED COLLABORATIVE FILTERING TECHNIQUE BASED ON A NEW SIMILARITY AND MADM METHODS (CF-NSMA) FOR IMPROVING THE RECOMMENDATION ACCURACY

HAEL ABDULLAH HUSSEIN AL-BASHIRI

Thesis submitted in fulfilment of the requirements
for the award of the degree of
Doctor of Philosophy

Faculty of Computer Systems & Software Engineering

UNIVERSITI MALAYSIA PAHANG

JANUARY 2019

## ACKNOWLEDGEMENTS

There have been many people that have contributed, one way or another, to work described in this thesis. In these lines, I would like to express my sincere appreciation and gratitude to all of them.

First and foremost, I would like to thank my supervisor and mentor, Mansoor Abdullateef Abdulgabber, for offering me the opportunity of pursuing this PhD. I would not be able to complete this thesis without his efforts, encouragement, patience, and continuous work during this time. He was advising and guiding me to find the path for which this thesis is put the first step. He was the best friend more than a supervisor. A lot of time we spent together. Daily sitting tea was the beautiful moments to enjoy and discuss our tasks, it was the core of this work. Additionally, I would also like to thank my core supervisor Dr Awanis Romli, for sharing with me her knowledge and expertise throughout my PhD study. I would not be able to complete this study without her generous help and advice. Every result described in this thesis was accomplished with their helping, guidance and supporting.

Moreover, I would like first to acknowledge the Faculty of Computer Systems & Software Engineering (FSKKP) at UMP University. My graduate experience benefited greatly from the high-quality seminars that the faculty organized. Second, the Republic of Yemen and its represented Yemen embassy in Kuala Lumpur for sponsoring my research work.

I also want to send my thanks to all lab mates; they provided a friendly and cooperative atmosphere at work. Their useful feedback and insightful comments had a clear effect on the completion of this work.

I would also like to acknowledge friends who offered me their encouragement during my whole study. Hasan Kahtan, thanks for your friendship and encouragement throughout the time. Other close friends I would like to mention here Ali Al-Shatwi, Mohammed Al-Sharafi, Fadhel Hujinah, and Bashir Al-Haimi thank you for your advice and collaboration. They were fundamental in supporting me during this stressful and difficult moments.

Finally, my deepest gratitude to my beloved family for their unflagging love and unconditional support throughout my life and my studies. Mom, Dad, brother, and sisters a lot of thanks for your constant love and support. I am lucky to have this family. My wife thanks for your friendship, love, and unyielding support. My sons, Kenan & Mohammed thank you for making my time here at Malaysia a lot more fun.

I would never forget myself, without my efforts my job would have undoubtedly been more difficult.

# ABSTRAK

Saringan Kolaboratif (CF), sebagai salah satu pendekatan yang paling banyak digunakan dan paling berjaya untuk menyediakan perkhidmatan cadangan, menyediakan pengguna dengan satu set cadangan yang berkaitan dengan apa yang mereka perlukan (minat mereka). Cadangan ini akan dihasilkan berdasarkan korelasi dalam kalangan pilihan pengguna seperti kedudukan dan tingkah laku. Walau bagaimanapun, bilangan pengguna dan item yang terdapat di Internet telah meningkat secara dramatik, dan kebanyakan pengguna tidak memberikan penilaian yang mencukupi untuk item tersebut. Selain itu, peningkatan/ pertumbuhan yang besar ini menjadikan matriks penarafan pengguna-item sangat besar dan jarang. Ini dianggap sebagai masalah dalam sistem memori berasaskan CF berdasarkan memori tradisional semasa mengira hubungan antara pengguna/item menjadi sangat sukar atau mungkin menyebabkan mencari jiran yang tidak berjaya yang seterusnya membawa kepada cadangan yang lemah. Oleh itu, kunci CF berasaskan ingatan memformulasikan kaedah korelasi yang betul yang dapat mengenal pasti kejiranan yang berjaya. Sebaliknya, kaedah persamaan tradisional yang lazim tidak dapat menentukan pengguna yang sama efektif, terutamanya apabila bilangan penarafan oleh pengguna adalah kecil. Begitu juga, kaedah skor ramalan, yang menjadi tumpuan beberapa kajian juga berdasarkan tahap kepentingan yang sama. Oleh itu, kaedah ramalan masih merupakan kawasan terbuka untuk penambahbaikan untuk mendapatkan penilaian yang lebih baik dan kedudukan barang calon. Oleh itu, Teknik Memori Baharu–Berasaskan CF dicadangkan untuk meningkatkan ketepatan cadangan, Ia dipanggil CF-NSMA. Teknik ini terdiri daripada tiga langkah utama: 1- Membina matriks baharu yang dinormalizasi untuk mengatasi isu sparsiti; 2- Merumuskan ukuran persamaan yang baru, berdasarkan pengakuan keadilan dan perkadaran faktor penarafan umum untuk mencari jiran yang tepat; 3- Mengaplikasikan kaedah MADM untuk mendapatkan penilaian yang lebih baik dan senarai kedudukan item calon. Fasa-fasa ini telah direka dan dilaksanakan dengan teliti untuk menyelesaikan isu-isu yang disebut tadi. Selain itu, untuk menilai ketepatan teknik CF-NSMA, beberapa eksperimen telah dijalankan menggunakan dataset awam (MovieLens 100K, DataLens 1M penanda aras dataran). Proses penilaian dilakukan untuk mengukur ketepatan teknik yang dibangunkan dengan menggunakan Ralat Mutlak Mutlak (MAE) untuk mengukur ketepatan ramalan dan Precision, Recall dan F-measure untuk mengukur ketepatan prestasi. Metrik yang dipilih dianggap sebagai metrik yang paling biasa digunakan dalam proses penilaian ketepatan teknik CF. Hasil eksperimen menunjukkan bahawa ketepatan teknik yang dicadangkan lebih baik berbanding dengan kaedah CF berasaskan memori berasaskan biasa. Peratusan ketepatan ramalan dari segi MAE adalah kira-kira 0.76 dan 0.74 melalui 100K dan 1M masing-masing. Walaupun, peningkatan teknik CF-NSMA dari segi ketepatan prestasi adalah lebih kurang tiga kali ganda ketepatan masa, sekitar empat kali ganda dari segi penarikan semula, dan sekitar tiga kali ganda dari segi ukuran F. Kesimpulannya, kerja ini menyumbang secara signifikan kepada bidang meningkatkan ketepatan CF berasaskan ingatan dengan membangunkan fasa-fasa penting CF berasaskan memori tradisional, termasuk mewakili semula matriks penarafan, merumuskan kaedah persamaan baru dan menggantikan kaedah ramalan dengan kaedah MADM. Tambahan pula, MADM berjaya meminimumkan kesan negatif kaedah ramalan dalam menilai dan menilai item calon. Oleh itu, aplikasi MADM dengan ketara meningkatkan ketepatan CF berasaskan ingatan dan menghasilkan hasil yang lebih tepat daripada kaedah asas. Oleh itu, objektif utama kajian ini telah dicapai.

# ABSTRACT

The collaborative filtering (CF), as one of the most widely used and most successful approaches to provide service of recommendations, provides users with a set of recommendations related to what they need (their interests). These recommendations will be generated based on the correlation among the users' preferences such as ratings and behaviour. Nevertheless, the number of users and items available on the Internet has increased dramatically, and most of the users do not give enough ratings for the items. Moreover, this vast growth has made the user-item rating matrix very large and sparse. This is considered a problem in the current traditional memory-based CF recommender system because the similarity calculation process between users/items becomes very difficult or may lead to locating unsuccessful neighbours which in turn to a weak recommendation. Therefore, formulating a right similarity method to identify the successful neighborhoods is a one key of memory-based CF. Similarly, the prediction method has the same level of importance in the process of improving the CF accuracy. Unfortunately, most studies on improving the accuracy of conventional CF systems have focused solely on enhancing the similarity measure. In contrast, improving the prediction method has been somewhat neglected. Consequently, the prediction method is still an open area for improvement to get better candidate items ranking and in turn increase the accuracy of CF. In the prediction process, the system predicts a user score for each item in the candidate set and promotes the highest-rated items as recommendations. This process of evaluating and ranking candidate items is therefore quite significant to the performance accuracy of the CF. Therefore, in this work, a new memory-based Collaborative Filtering (CF) technique is proposed to address the issue of sparsity data and improve the accuracy of recommendations, it is called CF-NSMA technique. The proposed technique consists of three main steps: 1- Constructing a new normalized matrix to overcome the sparsity issue; 2- Formulating a new similarity measure, based on adopting the fairness and the proportion of common rating factors to locate the accurate neighbours; 3- Applying the MADM method to get better evaluating and ranking list of candidate items. These phases were carefully designed and implemented to solve the issues that were mentioned earlier. Moreover, to assess the accuracy of CF-NSMA technique, several experiments were conducted using a public dataset (MovieLens 100K, MovieLens 1M benchmark datasets). The evaluation process was performed to measure the accuracy of the proposed technique using Mean Absolute Error (MAE) to measure the prediction accuracy and Precision, Recall and F-measure to measure the performance accuracy. These selected metrics are considered as the most common metrics to be used in an accuracy evaluation process of the CF techniques. The result of the experiments revealed that the accuracy of the proposed technique is better compared to the common base memory-based CF methods. The prediction accuracy percentage in terms of MAE was around 0.76 and 0.74 via 100K and 1M datasets, respectively. While, the improvement of the CF-NSMA technique in terms of performance accuracy was around more than three-fold in term precision, around four-fold in term of recall, and around three-fold in term of F-measure. In conclusion, this work contributes significantly to the field of improving the accuracy of memory-based CF by developing the critical phases of traditional memory-based CF, including re-representing the rating matrix, formulating a new similarity method and replacing the prediction method with the MADM method. Furthermore, MADM successfully minimizes the negative effect of the prediction method in evaluating and ranking the candidate items and significantly improves the accuracy of memory-based CF. Therefore, the primary objectives of this research were achieved.

# TABLE OF CONTENT

# LIST OF TABLES

## LIST OF FIGURES

# LIST OF SYMBOLS

| | |
|---|---|
| N | Natural numbers |
| Max | Maximum value scale of ratings. |
| $Min$ | Minimum value scale of ratings |
| $U$ | Set of users. |
| I | Set of items |
| R | Set of ratings |
| $r_{\mathrm{i},j}$ | Rating value of user $i$ on item $j$ |
| * | Absence of ratings |
| $\vec{c}$ | A vector of category of items in the dataset. |
| $c_g^j$ | Value of item $j$ belongs into $g^{th}$ type. |
| $t_{i,g}$ | Number of items rated by user $i$ and belong to genre $g^{\mathrm{th}}$ |
| $w_{i,g}$ | Normalized value of $t_{i,g}$, which represents the ratio of user $i$ preference on category $g^{\mathrm{th}}$ |
| $\theta$ | Denominator of $Sf$, which represents the minimum size of co-rated items |
| $s_{j,i}$ | Similarity value of user $i$ with user $j$ |

## LIST OF ABBREVIATIONS

| | |
|---|---|
| RS | Recommender System |
| CF | Collaborative Filtering |
| CB | Content-Based |
| MADM | Multi-Attribute Decision Making |
| TOPSIS | Technique for Order Preference By Similarity to Ideal Solution Technique |
| BC | Bray-Curtis distance measurement |
| Ff | Fairness Factor. |
| Sf | Sigmoid Function. |
| BSF | A new similarity measurement based on merged BC, Ff and Sf. |
| CF-BSF | CF based on the combination similarity measurement (BSF) |
| CF-NSMA | CF based on the New Similarity and MADM methods. |
| CV | Cross-Validation Partition Method |
| HO | Hold Out Partition Method |
| CF-PCC | CF based on Pearson Correlation Coefficient. |
| CF-CPCC | CF based on Constrained Pearson Correlation Coefficient. |
| CF-SPCC | CF based on Sigmoid Function Based Pearson Correlation Coefficient. |
| CF-Cosine | CF based on Cosine similarity measure. |
| CF-JMSD | CF based on Jaccard and Mean Squared Difference MSD. |
| CF-NHSM | CF based on New Heuristic Similarity Model. |

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

People, nowadays, live in an age of information overload and, therefore, they face so much information. Due to that, the people will be suffered a lot of options when they need to make their decision. Therefore, they tend to seek traditional ways that may help them to alleviate this problem such as friends, newspapers, advertising, and so forth (Su & Khoshgoftaar, 2009). This method will assist them in filtering information to make their decisions. Nonetheless, the information flood has progressively become a big challenge in people's daily life (Bilge & Yargıç, 2017; Kg & Sadasivam, 2017). The flood of information makes users more confused to get the desirable alternatives. Due to they do not have sufficient knowledge for personal evaluation of these alternatives. This issue stimulated more and more researchers to develop new techniques that help users to find their interesting and valuable information in a quickly and efficiently manner (Chen, Chen, & Wang, 2015). Consequently, the researchers improve information and Internet technologies to deal with the information overload issue. This phenomenon is called Recommendation Systems (RS), which utilize information retrieval and prediction methods to suggest the most relevant items to users' preference from among tremendous amount of available items (Su & Khoshgoftaar, 2009).

In general, RS infers the knowledge about users' preference based on the users' past behaviours. This knowledge is used later to provide valuable suggestions to users (Jonathan L Herlocker et al., 1999). With these technologies, the users of the Internet can quickly retrieve their preferred information from the mass data. Therefore, the primary objective of the RS is to provide tools that generate lists of items efficiently through predicting which items will be most preferred to a user (Balabanović & Shoham, 1997; Konstan et al., 1997; Resnick et al., 1994; Ricci, Rokach, & Shapira, 2011). For this reason, the RS has been a significant application area and the focus of considerable academic and commercial interests (G. Adomavicius & A. Tuzhilin, 2005; X. W. Yang et al., 2014). These systems are widely used by many commercials and non-profit websites to suggest products that might be of interest to their customers depending on their previous purchases. Before the recommender system provides recommendations, the system firstly collects information about its users to build knowledge about the preference of users which are used later to predict the users need.

In general, based on the state-of-the-art in recommender systems, the RS is classified into three main approaches which are: Content-Based (CB); Collaborative Filtering (CF); and Hybrid approaches (G. Adomavicius & A. Tuzhilin, 2005; Bobadilla et al., 2013; Burke, 2007; Chen, Chen, & Wang, 2015; Lü et al., 2012). The hybrid approach is a combination of both approaches (Claypool et al., 1999; Kim et al., 2006). In CB, the items are recommended to the target user based on comparing his/her items' information content, the items that have been rated in the past, with the items' content in the database (Aggarwal, 2016; Balabanović & Shoham, 1997; Mooney & Roy, 2000; M. J. Pazzani & Billsus, 2007a; Van den Oord, Dieleman, & Schrauwen, 2013). Whereas, in CF, the system proposes the items based on the feedback provided comparing to similar users' preferences (Ekstrand, Riedl, & Konstan, 2011; J. L. Herlocker et al., 2004; Resnick et al., 1994; Sarwar et al., 2001; Schafer et al., 2007). For that reason, CF is the most widely used and most successful methods for the recommendation system (Zang et al., 2016; B. Zhang & Yuan, 2017). The CF is classified into model-based and memory-based models (Su & Khoshgoftaar, 2009; R. Zhang et al., 2014). In the model-based model, there is a pre-built model that is used later to predict what the users will like. Whereas, in the memory-based model, also known as a neighbour-based model, there is no need to build a learning model. It operates on the entire database of ratings provided by users to find correlations between the users/items. The memory-based model is grouped into two algorithms: user-based and item-based methods (Gediminas Adomavicius & Alexander Tuzhilin, 2005; Ricci, Rokach, & Shapira, 2011; Su & Khoshgoftaar, 2009). A brief description of the recommendation system is presented in Section 1.2.

In the CF-based approach, the feedback of users on items is collected by the system to create a user-item matrix. The feedback can be explicit or implicit (J. L.

Herlocker et al., 2004; Kelly & Teevan, 2003; Lee, Park, & Park, 2008). The explicit feedback is provided directly by users in various ways. Rating is the most common way used to represent the degree of a user preference on an item. Whereas, the implicit feedback is where the system infers the preference of users based on the previous activities of users like browsing behaviour, time spent on shopping, or past purchases. The fundamental assumption behind CF approach is that other users' preferences can be selected and gathered to generate suggestion of the target user's preference. Intuitively, it assumed that, if some users have similar preferences in the past they will most likely have similar preferences in the future (Jannach et al., 2010). Several previous studies have provided developed methods of CF recommender system to improve the recommendation. However, there is still a weakness to deal with the issue of data sparsity which considered a major obstacle facing by CF. Therefore, providing a valuable recommendation about whom the system does not have enough information about their preferences is a key challenge faced by the CF. The main reason behind data sparsity is that most users do not rate enough number of items in the database. Then, the user-item rating matrix will be usually sparse (Chen, Chen, & Wang, 2015; Ghazarian & Nematbakhsh, 2015; Gu, Yang, & Dong, 2014e; Koohi & Kiani, 2017; Patra et al., 2015; Polatidis & Georgiadis, 2016; Revankar & Haribhakta, 2015; B. Zhang & Yuan, 2017). Sparsity issue makes finding the right relationships among users difficult and sometimes impossible. Thus, it might lead to locating unsuccessful neighbours and in turn lead to weak recommendations. Furthermore, the similarity and prediction methods play an essential role in improving the accuracy of CF. Many similarity methods have been proposed to improve the accuracy of CF recommendation. Nevertheless, the accuracy still has an open area for improvement. While, the prediction method, predicts the preference score that a user would give to an item, is somewhat neglected and has not received considerable attention from researchers in the process of improving the accuracy of CF. Which is also considered on the same level of importance to improve the recommendations (Cai et al., 2014).

In this work, to improve the recommendation accuracy of memory-based CF and address the issue of sparsity data, which are mentioned earlier, a proposed memory-based CF technique will be developed. It consists of three main phases. First, constructing a new normalized user-type matrix to overcome the sparsity issue. Second, formulating a new similarity measure through adopting fairness and the proportion of common rating factors to locate the accurate neighbours. Third, adopting the MADM method instead of traditional prediction method to get a better-ranked list of candidate items. The aim of formulating similarity measure and adopting MADM method phases is to improve the accuracy of CF. Thus, in this work, MADM will be utilised to evaluate and sort the nearest neighbours' items and provide the target user with the Top-M ranking as recommended items (Ic & Yurdakul, 2010). Moreover, several experiments will be conducted using MovieLens benchmark datasets (100K & 1M) to evaluate the accuracy of the proposed technique. To measure the accuracy of the proposed CF technique specified evaluation metrics are used: MAE (Mean Absolute Error) to measure the prediction accuracy and precision, Recall and F-measure to measure the performance accuracy

## 1.2 Background

Based on the state-of-the-art in recommender systems, the RS is classified into three approaches which are: CB, CF, and hybrid approach (Burke, 2007; Chen, Chen, & Wang, 2015; Ricci, Rokach, & Shapira, 2011; Sharma & Gera, 2013). Figure 1.1 represents the structure of classification of RS approaches.

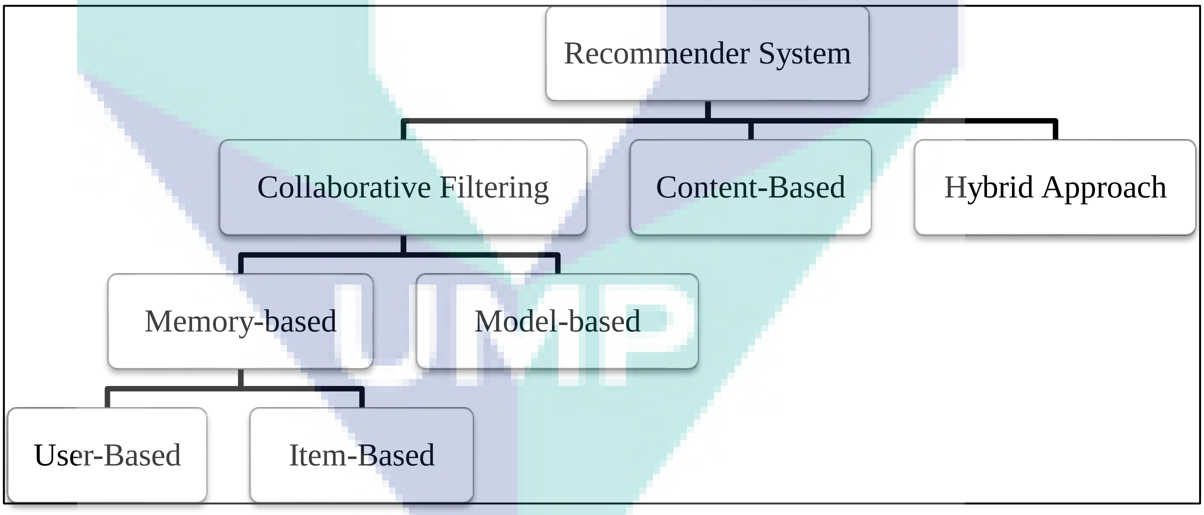



Figure 1.1 Approaches of RS.

I. Content-Based Approach

In this approach, the items will be recommended to a user based on comparing his/her items' information content, that has been rated by the user in the past, with items' information content in database (Lops, De Gemmis, & Semeraro, 2011; Mooney & Roy, 2000; M. J. Pazzani & Billsus, 2007a). The structure of content-based approach illustrated in Figure 1.2.

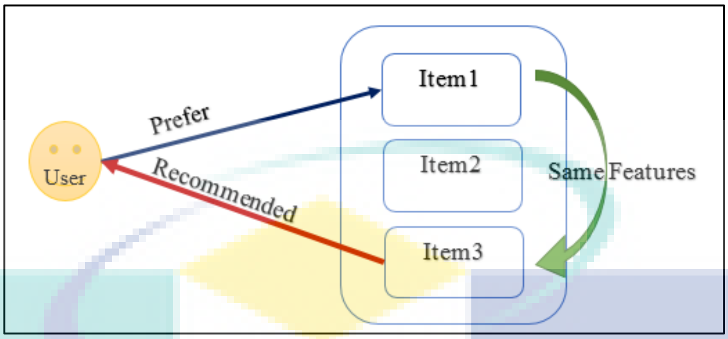



Figure 1.2 Content-based structure.
Source: Arekar, Sonar, and Uke (2015).

II. Collaborative Filtering Approach

In CF approach, the feedback provided by the users on items are collected by the system to create their preferences profiles. This feedback can be explicit or implicit feedback (J. L. Herlocker et al., 2004; Kelly & Teevan, 2003; Lee, Park, & Park, 2008). The explicit feedback will be provided directly by the users in varied ways. The rating is most common one which represent the degree of user's preference on an item. While, in the implicit feedback, the favourite of users will be inferred by the system based on the previous activities of users which can be browsing, spent time, or purchases. The idea behind this approach is that the users who have similar preferences in the past, they will have similar choices in the future (Jannach et al., 2010).

CF has a critical advantage which is not provided by content-based filtering. It can recommend items whose content is not easily analysed by automated processes. Typically, CF is classified into two primary models which are: memory-based and model-based (Su & Khoshgoftaar, 2009; R. Zhang et al., 2014). The memory-based CF model, also known as a neighbour-based, operate on entire database of ratings provided by the users on items. Memory-based can be grouped into two methods which are: user-based and item-based method (Gediminas Adomavicius & Alexander Tuzhilin, 2005; Ricci, Rokach, & Shapira, 2011; Su & Khoshgoftaar, 2009). The memory-based passes through several stages as shown in Figure 1.3. Firstly, locating the neighbours for the target user/item through computing the similarity with others based on the user-item rating matrix. The second stage is predicting process which depends on the correlation weight of selected neighbours. In this stage, the items which are rated by the neighbours and not yet rated by the target user will be collected to represent the candidate items. Next, predict the rating score which would user give to each candidate item. In the final stage, ranking the candidate items based on their prediction scores. Then, select the top elements, which may be related to the target user's taste, to represent his/her recommendations. While in the model-based, a model has previously constructed and afterward, is used to generate recommendations to a user. (G. Adomavicius & A. Tuzhilin, 2005).

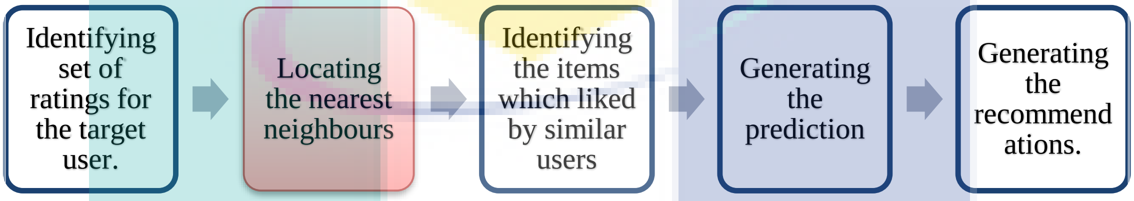



Figure 1.3 Memory-based CF method

i. User-Based Method

User-based was the first of the automated CF method (Ekstrand, Riedl, & Konstan, 2011). It was first proposed in the Group Lens Usenet article recommender (Resnick et al., 1994), also used in BellCor video and Ringo music recommender (Hill et al., 1995; Shardanand & Maes, 1995). The user-based method applies the straightforward phases of memory-based CF. Firstly, the user-to-user correlation is used to determine the most similar users for each user (Sarwar et al., 2001). And then, utilize those neighbours to make a prediction score on their items that not selected yet by the target user. Next, these items will be ranked based on its prediction score. Finally, the system generates recommendation involve the elements that have highest prediction rating. Figure 1.4 shows the basis of this method.

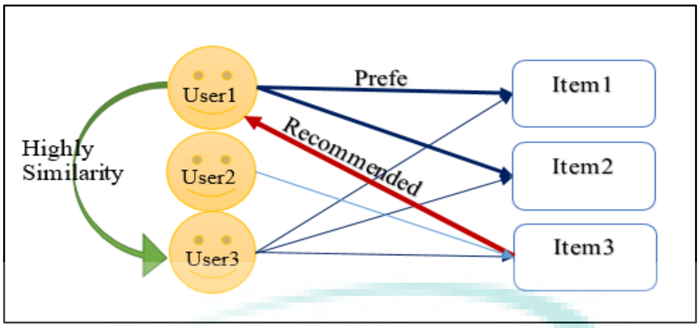



Figure 1.4 User-based CF method
Source: Arekar, Sonar, and Uke (2015).

ii. Item-Based Method

Item-based is another deployed memory-based CF method (Ekstrand, Riedl, & Konstan, 2011). In this method, the similarity computed between items by looking at how other users have evaluated items as shown in Figure 1.5. It has same steps of the user-based method, but the item-to-item correlations are used. Sarwar et al. and Karypis proposed it. (Sarwar et al., 2001).

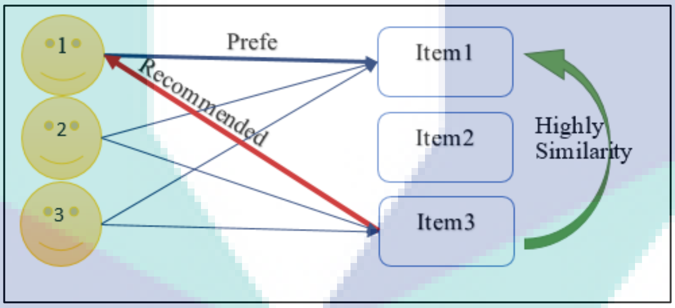



Figure 1.5 Item-based CF Method
Source: Arekar, Sonar, and Uke (2015).

III. Hybrid Approach

Several different recommender methods are combined into a hybrid recommender method (Burke, 2002). The main goal of this combination is to overcome the individual limitations. The hybrid recommender systems have been grouped into seven classes which will be discussed in Table 2.3 Section 2.3.

## 1.3 Problem Statement

Several considerable work has been done in developing methods for memory-based CF methods. However, there has been a noticeable limitation in that work. The memory-based CF technique suffers from many issues affecting its performance in term of accuracy such as sparsity, low accuracy, cold start and scalability (Y. Hu et al., 2017; Koohi & Kiani, 2017; Nguyen, Sriboonchitta, & Huynh, 2017). This work will focus to address two main issues that have a definite impact on recommendation quality: data sparsity and low accuracy.

Firstly, a key challenge for memory-based recommender systems is providing high-quality recommendations to users whom the system do not have enough information about their preferences. The user-item matrix is usually sparse because most of the users do not rate enough items or may not have used the specific item (Chen, Chen, & Wang, 2015; Gu, Yang, & Dong, 2014e; Koohi & Kiani, 2017; Liu et al., 2014; Patra et al., 2015; Polatidis & Georgiadis, 2016; Revankar & Haribhakta, 2015; B. Zhang & Yuan, 2017). Therefore, data sparsity has negatively affected the performance of memory-based as follow:

i. Finding the correlation between a pair of users or items will be difficult or impossible and making the method useless.
ii. Even, if the calculation of similarity is probable, it may lead to finding a fake relationship, which leads to a weak recommendation (Koohi & Kiani, 2017; Papagelis, Plexousakis, & Kutsuras, 2005).

Hence, building the right preferences file for users is very important, which leads to locate the right neighbours, who have similar preferences with the target user, and in turn, lead to a better recommendation.

Secondly, the accuracy of the memory-based CF technique has directly affected by the similarity and prediction methods. Therefore, improving these methods will be followed by an improvement in the accuracy. First, the similarity method plays an essential role in the process of locating the neighbours. Thus, choosing and developing suitable correlation method is a primary key of memory-based to identify the successful neighbourhoods. Finding the right neighbours mean getting more relevant items and vice versa. Recently, there are many studies focused how to improve the accuracy of memory-based system through developing the similarity measure (Cao et al., 2016; Cheng et al., 2015; El Alami, Nfaoui, & El Beqqali, 2015; Y. Hu et al., 2017; Huang & Dai, 2015; Koohi & Kiani, 2017; Patra et al., 2015; Polatidis & Georgiadis, 2016; Suryakant & Mahara, 2016; Zang et al., 2016; B. Zhang & Yuan, 2017). However, these methods still have some shortcomings in finding the neighbours that might hurt the accuracy as will be discussed in Section 2.5. Therefore, the working on similarity measures still represents an open area for development. Second, most of the studies have been improved the accuracy of memory-based just by improving similarity measure while only a few studies focused on the prediction score method which is also on the same level of importance (Cai et al., 2014; R. Zhang et al., 2014). The process of prediction method predicts the active user's score on the current item, which not rated yet, and generates the most appropriate items to the target user according to these scores. Consequently, the development of prediction method will have a positive effect on improving accuracy. Therefore, the researcher will be going to replace the prediction method by MADM method as a more effective method in the evaluating alternatives. That will be led to a new result which is expected to be better, in turn, lead to improving the accuracy.

In conclusion, two main issues will be addressed in this work as mentioned in advance: decrease the negative impact of sparsity data in memory-based CF and improve its accuracy.

i. The sparsity of data has a negative effect on the accuracy of memory-based CF. Thus, in this work, this impact will be alleviated through represent the preferences of users. The rating feedback provided by the users will be utilised to build their global preferences. The global preference will be constructed based on the classes of items that will be used as the main input to find the correlation among users.
ii. The accuracy of the memory-based CF technique has directly affected by similarity and prediction methods that are used. Therefore, improving these methods will be followed by an improvement in the accuracy. This similarity measure will be used to calculating the correlation among users. And the prediction method is used to predict preferences score of neighbours' items and not yet rated by the target user.

## 1.4 Research Objectives

The problem statement drove the researcher to have three main objectives. The main aim is to propose a new memory-based CF technique based on a new similarity and MADM methods. The proposed (CF-NSMA) technique can mitigate the negative effect of the data sparsity issue and improve the accuracy of recommendations. Thus, the objectives of this work are pointed as follow:

I. To identify the state of the art of traditional Memory-Based CF methods related to sparsity issue and recommendation accuracy.

II. To propose a new memory-based CF (CF-NSMA) technique throughout re-representing the users' preferences, formulating a new similarity measure, and adapting the MADM method.

III. To evaluate the proposed memory-based CF technique using evaluation metrics in term of accuracy.

## 1.5 Research Scope

In this work, the CF-NSMA memory-based CF technique will be proposed according to a particular context (MovieLens), and the evaluations of this technique will be on datasets that related to that context. Therefore, MovieLens, as a public dataset available and widely used in the processes of CF, will be utilised to design and develop the proposed technique. Moreover, the study research will be included the memory-based CF methods related to improve and address the accuracy and sparsity issue, respectively. Consequently, to achieve this goal, the developed technique will consist of the following phases:

i. Re-representing the users' preferences based on user-item rating matrix to alleviate the impact of sparsity data issue. This representation includes transformation the local preferences (ratings) to a global preference (type preference).

ii. Formulating a new similarity measure will utilise the global preferences as the primary input to calculate the correlations among users. This measure will consider the fairness and proportion of common rating factors to locate the accurate neighbours.

iii. Replacing the prediction method by MADM method to get a better ranking of candidate items. This method can evaluate and rank the items which rated by neighbours. In the decision matrix, the neighbours will represent the criteria, and their elements will represent the alternatives. In addition, the correlation weights between the target user and his neighbours will be utilised as main input for MADM method to describe the importance associated with each criterion.

About the evaluation phase, the evaluation will be performed to measure the accuracy of the proposed technique. A specific evaluation metrics will be used to evaluate the proposed technique in term of accuracy which are:

i. Mean Absolute Error (MAE) to measure the prediction accuracy through measuring the closeness of the actual user ratings and the predicted ratings.

ii. Precision, Recall, and F-measure metrics were used to measure the performance accuracy. Measuring if the developed technique provides the items that the user will use them.

These specified metrics are considered as most common metrics to be used in the process of accuracy evaluation of the CF techniques.

## 1.6 Significance of Study

The recommender systems have been very important applications. They are widely used in many commercials and non-profit web applications to help users to find the most interesting and valuable information for them in speedily and efficiently manner. Since the beginning of the webs, there has been an explosive growth of information on the Internet. Unfortunately, searching through this massive space of data is difficult and time-consuming. Therefore, huge efforts have been taken to present a user with the most his/her information interest as fast as possible. In today's world, we are flooded with extensive options in a wide variety of things, not just web pages. For example, there are hundreds of thousands of books, movies, songs and news articles to choose from. Recommender systems assist users who face difficulties through a massive set of choices and present items they may like. From this point, recommender systems are used as filters that suggest what they only need, for instance, recommender systems used by Amazon, Netflix, and Spotify. Although considerable work has been done in existing approaches to improve the technique of memory-based CF technique, there have been obvious limitations suggested in the literature. These limitations include and not limited to the sparsity of data matrix, representing users' preferences based on their feedback, similarity measure and prediction method which are also have clearly affecting the accuracy of recommendations. Therefore, the significance of our proposed technique can be able to alleviate the sparsity limitation. In addition, the accuracy of recommendation will be improved that help user to get a more accurate recommendation.

## 1.7 Preliminary Definitions and Terminology

The definitions, measures, parameters, notations, and sets used in the equations, will be used throughout this work, will be specified in this section. In addition, we provide a short definition of the terms which will be used in this thesis.

### 1.7.1 Formalization

The database domain for a CF system consists of U users who have preferences on I items expressed by rating $r_{u,i}$. The system collects the ratings from users on items to build the acknowledgements about users' preferences. The data represented by user-item matrix R where its rows represent the users, columns describe the items and the cells stuffed by ratings within the range [*min..max*] scale, where the unknown rating value, the user has not rated an item, will be indicated by the symbol * as shown in Table 1.1. This rating scale depends on the system which can represent in many forms. Some system uses integer-valued rating scales such 1-5 stars; others use binary scale (like/dislike).

Table 1.1 User-item matrix data structure R

| | $i_1$ | $i_2$ | ... | $i_m$ |
|---|---|---|---|---|
| $u_1$ | $r_{11}$ | $r_{12}$ | ... | $r_{1m}$ |
| $u_2$ | $r_{21}$ | * | .... | $r_{2m}$ |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| $u_n$ | $r_{n1}$ | $r_{n2}$ | ... | $r_{nm}$ |

### 1.7.2 Terminology

Finally, we specify a short definition of the terms used in this work as shown in Table 1.2:

Table 1.2 List of terminology

| Term name | Description |
|---|---|
| Recommender | The entity that generates a customized output according to users' preferences which can be a software program or a person (regular adviser). The recommender system does not guarantee to produce items which are relevant to the preference of users but may encourage users to find useful or interesting items. |
| Item | The item includes any object that has information such as the products (book, movie . . . etc.), documents (newspaper, article), services (restaurant), or persons (suggest friends in social media). |
| User's interest/ preference | An outline that shows a user view of an item. It is difficult to represent it in an objective manner; therefore, the representation, in this case, is a subjective concept. |
| Recommendations Suggestions | The final output of the recommendation system which includes in most cases one or more items. The system offers this output to users based on their interest. Moreover, to decide if the item is preferable or not to an active user there are several methods used as standards in a recommender system. |
| Ratings | The evaluation that represents the degree of a user's interest in an item. The developer of the recommender system determines the possible values of this evaluation. |
| Prediction | The predictable interest of a user on an item. The concept of prediction is diverse to the recommendation concept. In some systems can represent predictions with the actual recommendations while in other systems produce recommendations only. |
| Actual rating | The real voting of the user in a specific set of items. The user himself gives this rating according to the rating scale that used by a recommender system. |
| Predicted rating | An objective standard representing the estimated voting for a user on a specific item. The recommender system estimates this rating value which is must within a range of rating scale. |
| Similarity measure | The concept of similarity measure or similarity is fundamentally important in collaborative filtering that measures the similarity between two objects. |
| Prediction method | The specific algorithms that will be used to compute the expected ratings over a given set of items in a recommender system. |
| Accuracy Quality | The terms of recommendation accuracy or recommendation quality take the same meaning and can be divided into two principal terms each of them has own metrics. These terms are the predictive accuracy and performance accuracy. In this work, the term of accuracy will be used to evaluate the efficiency of the developed technique. |
| Performance Accuracy | The degree to which the result of recommender system conforms to the user satisfaction or near to preferences of users. That is main, measuring if the system provides the elements that the user will use them, which commonly measured by Precision, Recall and F-measure |
| Precision | The fraction of retrieved items in a recommendation list that the user would rate as useful. |
| Recall | The fraction of relevant items that are retrieved to the relevant items. |
| F-measure | F-measure metric is a combined metric of precision and recall, it gives different information, the weighted mean of precision and recall, compared to precision and recall |

Table 1.2 continued

| Term name | Description |
|---|---|
| Prediction Accuracy | The procedure that indicates at which degree the predicted voting agrees with the real voting of users. It is mean that, the ability of the system to predict a user's rating for an item. The most accurate of predictions is the best performance of the recommender system. The Mean Absolute Error (MAE) measurement is the most widely used metric to measure the predictive accuracy. |
| MAE | It calculates the difference between the actual users' ratings in the test set and predicted score to evaluate the predictive accuracy. |

In this work, the term of accuracy will be used in the evaluation phase to test how well the developed technique works. The accuracy, in this work, can be divided into two subparts which are the Prediction Accuracy and Performance Accuracy each of them has own specified metrics. The MAE measurement will be used to measure the prediction accuracy. Moreover, the performance accuracy will be measured using Recall, Precision, and F-measure.

### 1.7.3 Traditional Common Memory-based CF Methods

Table 1.3 presents the list of the popular and widely used memory-based CF methods (Liu et al., 2014; Patra et al., 2015; Polatidis & Georgiadis, 2016). They will be discussed in detail in Section 2.5, which will be used to compare the proposed technique using accuracy metrics.

Table 1.3 Similarity measures frequently used in the traditional memory-based CF.

| Tradition memory-based CF methods | Description |
|---|---|
| CF-PCC | Memory-based CF using Pearson Correlation Coefficient, Cited by 5816. |
| CF-CPCC | Memory-based CF using Constrained Pearson Correlation Coefficient, Cited by around 4000. |
| CF-SPCC | Memory-based CF using Sigmoid Function Based Pearson Correlation Coefficient, Cited by around 600. |
| CF-Cosine | Memory-based CF using Cosine similarity measure, Cited by around 4000. |
| CF-JMSD | Memory-based CF using two popular similarity methods, Jaccard and Mean Squared Difference MSD. Cited by around 203. |
| CF-NHSM | Memory-based CF using New Heuristic Similarity Model, recent method and Cited by more than 120. |

### 1.7.4 Aggregation Methods

The aggregation methods which are often used in order to calculate the prediction score for an active user *x* on item *i* of are listed in Table 1.4 (Bobadilla et al., 2011).

Table 1.4 Aggregation methods

| Algorithm | Formula |
|---|---|
| Average method | $\boldsymbol{P_{x,i}} = \boldsymbol{1}/\lvert \boldsymbol{G_{x,i}} \rvert \sum_{y \in \boldsymbol{G_{x,i}}} \boldsymbol{r_{y,i}}$, where $\boldsymbol{G_{x,i}} \neq \emptyset$ |
| Weighted sum method | $\boldsymbol{P_{x,i}} = \sum_{y \in \boldsymbol{G_{x,i}}} \boldsymbol{s(x,y)} * \boldsymbol{r_{y,i}} \Big/ \sum_{y \in \boldsymbol{G_{x,i}}} \boldsymbol{s(x,y)}$, where $\boldsymbol{G_{x,i}} \neq \emptyset$ |
| Adjusted weighted method (Deviation-From-Mean) | $\boldsymbol{P_{x,i}} = \overline{\boldsymbol{r_x}} + \sum_{y \in \boldsymbol{G_{x,i}}} \boldsymbol{s(x,y)} * (\boldsymbol{r_{y,i}} - \overline{\boldsymbol{r_y}}) \Big/ \sum_{y \in \boldsymbol{G_{x,i}}} \boldsymbol{s(x,y)}$ Where $\boldsymbol{G_{x,i}} \neq \emptyset$ |

Where $P_{x,i}$ represents the prediction which a numeric value that represents the predicted view for the target user *x* about a specific item *i* based on the similarities and ratings of his/her K neighbours. And $\boldsymbol{G_{x,i}}$ represent a set of users who are neighbours of user *x* and have rated item *i*. $\overline{\boldsymbol{r_x}}$ denotes to the average ratings of user *x*

## 1.8 Organization of Thesis

The rest of this thesis is organized as follow: Chapter Two provides an introduction to recommendation system and discusses existing memory-based CF methods that are related to this work. That chapter comes out with the finding including the limitations of current works. Chapter three presents the essential research methodology phases that help the researcher to design the structure and adopt the methodology to achieve the objectives of this thesis. Chapter four describes the main components of the proposed technique in details which including, constructing the new matrix, developing the new similarity measure, and applying the MADM method to rank the candidate items instead of the prediction method. Chapter Five presents evaluation process through making a comparison of proposed technique with common memory-based methods. Moreover, shows the positive effect of improvement using the MADM method on those common memory-based methods. Finally, this thesis will be concluded in Chapter Six. Summary of achievement and contributions of this thesis and future work also will be pointed in this chapter.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

The first objective of this thesis will be addressed in this chapter. Recently, there is a sharp increase in the number of information and products which are available in the online Webs (Arekar, Sonar, & Uke, 2015; El Alami, Nfaoui, & El Beqqali, 2015; Najafabadi et al., 2017). As an example, the millions of products which offered to the customers every day make their decisions more difficult. In this context, the purpose of Recommender System (RS) is to assist users of the Internet by suggesting a list of items as recommendations when the collections of items are very large (Das, Sahoo, & Datta, 2017; Resnick & Varian, 1997; Wang et al., 2016). The RS recommends items which considered as more relevant to the users' interest. This suggesting depends on the collecting and analysing the feedback provided by the users such as behaviour and preferences (Balabanović & Shoham, 1997; Konstan et al., 1997; Resnick et al., 1994). On the other part, the RS takes advantage of feedback from the users to create knowledge about their interest. As a feedback type, there are different forms of feedback, which are divided into two common categories of feedback: explicit and implicit (J. L. Herlocker et al., 2004; Kelly & Teevan, 2003).

In the state-of-the-art, there are several approaches have been proposed in order to provide recommendations, which are commonly classified to Content-based (CB), Collaborative Filtering (CF) and hybrid approaches as mentioned in Chapter One. In particular, we have been interested in the revision of the work related to memory-based CF. For the following reasons: CF has become one of the most widely used to provide a recommendation service in many online web systems (Huang & Dai, 2015; Nagarnaik & Thomas, 2015; Polatidis & Georgiadis, 2016). CF recommends items to the users, that they are interested in, based on their feedback. Therefore, no need to analyse the content of items and it can provide a recommendation for elements whose content is not easily analysed by automated processes (Su & Khoshgoftaar, 2009). This independence makes it applicable in several domains.(G. Adomavicius & A. Tuzhilin, 2005). Therefore, in this work, the researcher is going to propose a memory-based CF technique to alleviate the impact of sparsity issue and improve the accuracy. This proposed technique consists of three main steps. First, constructing a normalized user-type matrix to represent the global preferences of users. Second, formulating new similarity measure to locate the right neighbours. Third, adopting Multi-Attribute Decision-Making method (MADM) to evaluate and rank the candidate items instead of prediction method to get better recommendations. In this chapter, an overview of the related work on the topics of interest of this work will be presented.

The rest of this chapter is organized as follows: Firstly, a comprehensive description of the recommender system field (evolution and background) will be offered in Section 2.2. Moreover, the relevance feedback in CF recommender system will be detailed in this section. Furthermore, the main approaches categories in the recommender system will be pointed in Section 2.3. Then, in Section 2.4, CF approach will be introduced. Furthermore, in Section 2.5, the memory-based CF related work and most common challenges which face CF will be discussed, the outcome of this section is the limitations of existing works. Moreover, the MADM method will be explained in Section 2.6. Next, the widely used metrics in CF recommender system evaluation and public datasets will be presented in Section 2.7. Finally, the summary of this chapter will be concluded in Section 2.8. Figure 2.1 shows the visualisation outline of this chapter.

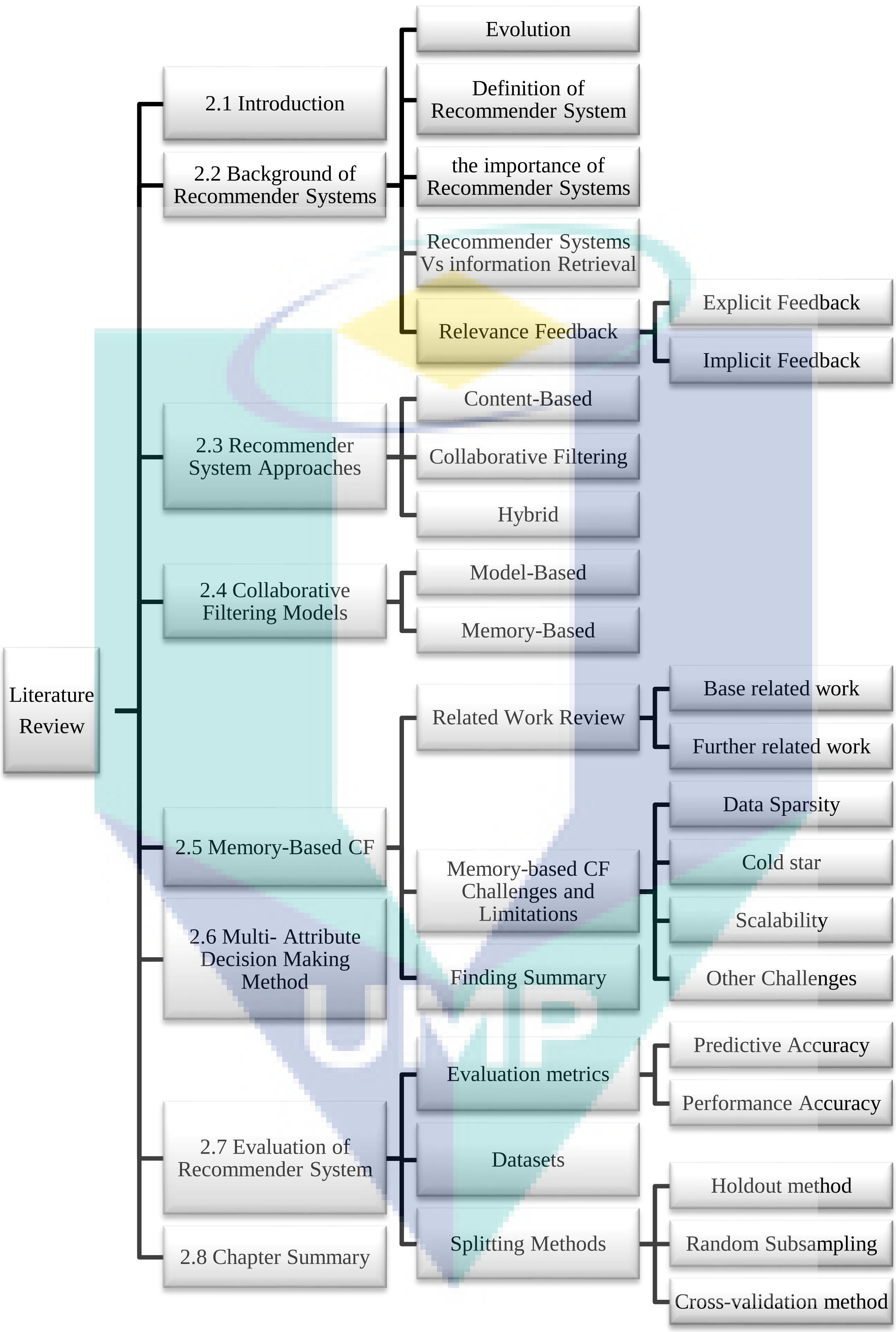



Figure 2.1 Chapter's outline visualisation

## 2.2 Background of Recommender Systems

### 2.2.1 Definition of Recommender System

RS is a computer-based system that generates a set of items/services/information as a recommendation through suggestion which items will be most interesting and valuable information for users. They gathered users' feedback information, such as ratings and past activities of users, to build their profiles preferences. Where the recommendation set contains items which not yet accessed or purchased and maybe interests of the users, and the item is the general term used to denote what the system recommends to users. For example, this item may be movie, article, web page, music, or product. This set of recommendation can assist users to make their decision, what they need/want, in quickly and efficiently manner when they have multiple alternatives (Balabanović & Shoham, 1997; Konstan et al., 1997; Resnick et al., 1994). Others have defined the recommendation as a subclass of information filtering system that tries to predict the rating or preference that a user would give to an item in the future and provide the user with the items that have a highest predicting rating (Ricci, Rokach, & Shapira, 2011).

### 2.2.2 The Importance of Recommender System

With the beginning of the web, people have been overwhelmed with information. Unfortunately, searching through this huge space of data is time-consuming and difficult, especially in case of that users do not know what they want. However, this cannot reduce the importance of the utility of the web as a significant information source. Therefore, tremendous efforts have been taken to present a user with the most relevant information related to his/her search as fast as possible. Search engines are clear examples of such efforts. Search engines propose web pages to users based on the users' requests. In today's world, we are flooded by options in a wide variety of things, not just the web pages. There are hundreds of thousands of books, movies, songs and news articles to choose from. Based on our likes and dislikes, many of these are undesirable, redundant or unrelated. RS assist users who face difficulties through a huge set of choices and present items they may like. From this point, RSs are used as filters that propose what we only need/want (Gediminas Adomavicius & Alexander Tuzhilin, 2005). Examples of popular RSs are Amazon ("Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more," 2017) and MovieLens (Najafabadi et al., 2017). On the other hand, the RS also has become one of the essential resources of tools of e-businesses to improve the selling. Many companies have been using recommendation technology to serve their customers and grow its profit such as eBay.com, Amazon.com, and Netflix.com (Schafer, Konstan, & Riedl, 1999; Sivapalan et al., 2014). To conclude, the primary goal of RS is to provide suggestions items through predicting what the user will prefer and ease their items finding and to improve the companies' sells.

### 2.2.3 Evolution

In the start of the 1990s, the CF technique was introduced as a solution to deal with information overload. Tapestry (D. Goldberg et al., 1992), is a manual e-mail filtering system developed which help users to get its relevance to them using annotations given by the user. It keeps track of messages and allows the user to provide the query for items in an information domain based on other users activates. For example, in a corporate e-mail, the statement query ("give me all the messages forwarded by Ali") retrieves all messages who Ali forwarded them.

Immediately afterwards, the automated CF was developed to identify relevant information and collect them to produce recommendations. GroupLens system is one of these automatic CF systems (Resnick et al., 1994). It can identify the relevant articles which may be interesting to the target user depends on the previous users' ratings or actions without query. The GroupLens system required just feedback (ratings or actions) from users on articles. Then, it combined this feedback with the feedback of other users to generate recommendations. Therefore, the user does not need to know about other users' opinions and what other users or items are in the system to get suggestions. After that, the interesting of CF increased among machine learning and information retrieval researchers. This led to developing several of RS for varied domains such as, BellCore Video Recommender (Hill et al., 1995) for movies, the Ringo (Shardanand & Maes, 1995) for music, Jester (K. Goldberg et al., 2001) for jokes, and iTravel (W. S. Yang & Hwang, 2013) for tourism.

At the end of the 1990s, RS applied in the domain of E-commerce. Amazon may be the most common application of RS which based on purchase history with ratings to recommend items for its users (Polatidis & Georgiadis, 2016). The primary objective of an RS in E-commerce is to increase the sales volume. The customers may buy a product if it is recommended to them. Many corporations have been using recommendation technology to serve its customers and enhance its profit (Schafer, Konstan, & Riedl, 1999; Sivapalan et al., 2014).

The techniques of recommendation have also developed after CF to include content-based methods which were extracted from information retrieval field (Balabanović & Shoham, 1997; Basu, Hirsh, & Cohen, 1998; Lops, De Gemmis, & Semeraro, 2011; Mooney & Roy, 2000; M. J. Pazzani & Billsus, 2007a). To recommend the items, these type of approaches replace the rating and history profile with actual content or features of the items. There are several applications used this technology, for instance, the PRES system (Van Meteren & Van Someren, 2000), which compares user's profile with article contents in order to generate recommendation list. The document contains a set of terms which represent the items, and similar sets of terms also identify the user profile. After generating the recommendation, the profile and content recommendations are analysed to get feedback about the recommendation if useful or not.

In the end, hybrid recommender system emerged which combine multiple techniques to produce a new recommendation technique which can avoid their limitations and gain better system performance and quality (Burke, 2002; Ricci, Rokach, & Shapira, 2011).

### 2.2.4 Recommender Systems and Information Retrieval

The capacity of computers to make advising was known in early in the history of computing. The amount of data available on the online environment has become massive and still rising at an incredibly rapid average (Cai et al., 2014; Mao et al., 2013; Wu & Zheng, 2010). On the positive side, this wealth of information may have aided the users of the Internet to find what they are looking for. On the negative side, this same wealth also makes the helpful information more problematic to be found, which called "information overload" (D. Goldberg et al., 1992). Two fundamental web technologies (information retrieval and recommendation system) have been developed to assist Internet users in overcoming the information overload problem. In the information retrieval case, presented in Figure 2.2, the system required statement query to expresses the information which they need. The Internet users will submit this query to the search system (engine). Then, the system attempts to find the most similar items in the collection of data (Baeza-Yates & Ribeiro-Neto, 1999; Chowdhury, 2010; Manning, Raghavan, & Schütze, 2008; Salton & McGill, 1986).

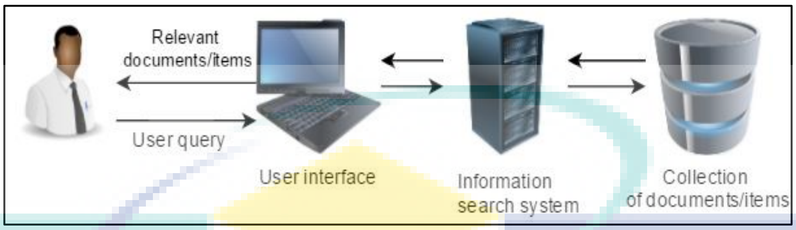



Figure 2.2 Information retrieval system

In the RS case, illustrated in Figure 2.3, the system builds knowledge about user' information needs based on their feedback, which expressed explicitly or implicitly and then, the system uses the CB, CF or combined both ways to find the items that user interest in (Konstan et al., 1998; Resnick & Varian, 1997). In a content-based filtering way, the system extracts the features of items which selected previously. Next, these features are used to identify the similar items that have not taken yet as recommendations (Balabanović & Shoham, 1997; Lops, De Gemmis, & Semeraro, 2011; Mooney & Roy, 2000; M. J. Pazzani & Billsus, 2007a; Sikka, Dhankhar, & Rana, 2012; Van Meteren & Van Someren, 2000). While, CF way uses the history of user behaviour to find the items which user may interest in (D. Goldberg et al., 1992; Resnick et al., 1994). The basic idea behind this approach is that if there are certain of users have similar preferences in the past; they tend to have the same interest in the future. For example, the user after watching movies rates them with a pre-defined scale. After that, the movie RS suggests movies that will be of interest to the user based on previous movies which already has been rated. The Netflix for movies and Last.fm for music are typical examples of CF recommender systems.

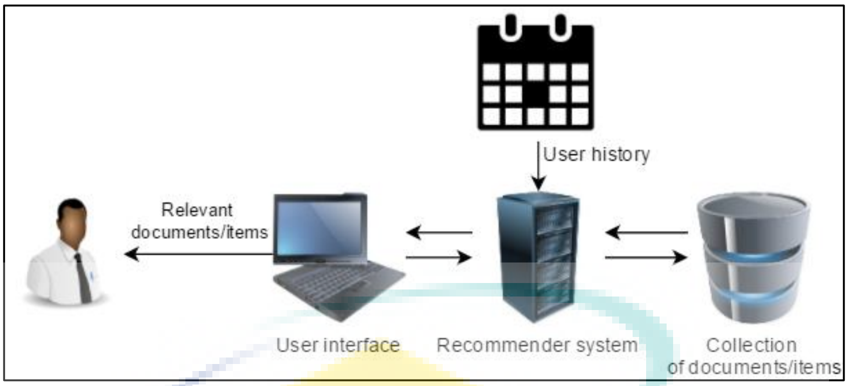



Figure 2.3 Recommendation system

### 2.2.5 Relevance Feedback

The knowledge about user interest or activities is fundamental in RS based CF. The system store behaviours and preferences of users in a transactional database. This database used later to build users' profiles and exploit the knowledge of user in order to make the recommendation more efficient. Typically, the user's profile contains the information of user's interest with a specific format that makes the recommending more efficient. Moreover, the system needs feedback provided by the user on an item, which considers the base unit in the process of creating user preference. In General, this feedback categorised into two popular types according to how it is collected from the user, as implicit feedback and explicit feedback (J. L. Herlocker et al., 2004; Kelly & Teevan, 2003; Lee, Park, & Park, 2008). Table 2.1 represents some common systems with the type of feedback used according to (Montaner Rigall, 2003).

Table 2.1 RS relevance feedback

| **Recommender system** | **Feedback type** |
|---|---|
| Bellcore Video Recom | Explicit (Ratings) |
| Krakatoa Chronicle | Explicit (Ratings), Implicit (Saving, Scrolling, Time Spent) |
| WebWatcher | Explicit (Goal Reached), Implicit (Links) |
| NewT | Explicit (Like/Dislike) |
| SIFT Netnews | Explicit (Like/Dislike) |
| Webmate | Explicit (Like/Dislike) |
| WebSail | Explicit (Like/Dislike) |
| News Dude | Explicit (Like/Dislike, |
| Tapestry | Explicit (Like/Dislike, Text Comments), Implicit(Forwarding) |
| Amalthaea | Explicit (Ratings) |
| Casmir | Explicit (Ratings) |
| Fab | Explicit (Ratings) |

Table 2.1 continued

| Recommender system | Feedback type |
|---|---|
| ifWeb | Explicit (Ratings) |
| InfoFinder | Explicit (Ratings) |
| INFOrmer | Explicit (Ratings) |
| MovieLens | Explicit (Ratings) |
| NewsWeeder | Explicit (Ratings) |
| PSUN | Explicit (Ratings) |
| Recommender | Explicit (Ratings) |
| Ringo | Explicit (Ratings) |
| Syskill & Webert | Explicit (Ratings) |
| Amazon | Explicit (Ratings), Implicit (Purchase History) |
| CDNow | Explicit (Ratings), Implicit (Purchase History) |
| LifeStyle Finder | Explicit (Ratings), Implicit (Purchase History) |
| Smart Radio | Explicit (Ratings), Implicit (Saving) |
| Anatagonomy | Explicit (Ratings), Implicit (Scrolling, Enlarging) |
| Netflix | Explicit (Ratings), implicit (Unknown) |
| GroupLens | Explicit (Ratings, Text Comments), Implicit (Time Spent) |
| WebSell | Explicit (Unknown) |
| Personal WebWatcher | Implicit (Links) |
| SiteIF | Implicit (Links) |
| LaboUr | Implicit (Links, Time Spent) |
| Let's Browse | Implicit (Links, Time Spent) |
| Letizia | Implicit (Links, Time Spent) |
| Beehive | Implicit (Mail History) |
| ACR News | Implicit (Navigation History) |
| Websift | Implicit (Navigation History) |

Source: Montaner Rigall (2003).

#### 2.2.5.1 Explicit Feedback

Traditionally, explicit feedback is obtained from users that indicating the degree preference of users on different items which is already selected by users (Jawaheer, Szomszor, & Kostkova, 2010). Then the system gathers these ratings in order to build the profiles of users' preferences. The explicit ratings usually associated with a scale of values that represent the degree of user preferences on items are numerical rating feedback (1-to-5 stars, 1-to-10 points) or simple binary feedback (like/dislike). As an example, for numerical rating, if user *x* rated the movie *a* with five scores and movie *b* with three scores. Then, we can explain this observation as that user *x* prefers movie *a* more than movie *b* (e.g., Netflix 1-5 stars' ratings with step size 0.5 and MoviePilot 0 -10 scale also with step size 0.5). In the binary case, there are two indications; like and dislike an item (e.g., YouTube up thumbs for like item and down thumbs for dislike item). In addition, most studies have focused on developing RS based on this type of user feedback. Due to its explicit feedback nature is considered higher quality than implicit feedback. (Yifan Hu, Koren, & Volinsky, 2008; Jawaheer, Szomszor, & Kostkova, 2010). Following this intuition, the system collects the ratings of all users for all items to construct the user-item matrix which are utilised for finding the relationship between users/items.

#### 2.2.5.2 Implicit Feedback

In contrast, in implicit feedback, recommender system automatically infers the preference of user from user's history actions collected by the system (Knijnenburg et al., 2012). There are different ways to obtain relevance feedback from a user. For example, in Amazon and CDNow, the system monitors the user's purchase in order to build knowledge about user interest, a navigation history in Web Watcher system, the time spent by the user on an item (e.g. GroupLens & Krakatoa Chronicle), clicks in a ranking context (K. Hofmann et al., 2014) and play counts for music or videos (Yifan Hu, Koren, & Volinsky, 2008). In most time, the implicit feedback does not reflect the real interest of the user. In another statement, the consuming time on an item/page or purchasing an item it's not mean that the user prefers this item. For example, if we consider that the system depending on spent time to indicate the user preference. And the person spent a long period watching a TV. But this inference might be wrong; the user may only have turned on the TV and then been interrupted by a phone call or busied by another activity and left the TV on.

As a final point, due to the implicit feedback does not collect directly from the user then it is not necessary to represent the preference of the user. This makes the explicit more reliable than implicit feedback. In addition, indicating explicit preference from implicit feedback is more difficult. Furthermore, (Adomavicius & Kwon, 2007; Adomavicius, Manouselis, & Kwon, 2011; Manouselis & Costopoulou, 2007), studies have introduced a different type of method for gathering feedback with multi-criteria rating system, which enables users to rate multi-dimensions of items. The item has different attributes, and these systems allow the users to evaluate each of them independently. For example, in movie RS, if the movie has three characteristics, e.g., direction, acting, and story, then the user can give three ratings. In this case, the system considers more flexible on the side of the system, but more effort on the side of the user due to the need of more information provided by the users. Eventually, most of the CF techniques have been using explicit feedback (i.e., ratings) in their scenarios. Nevertheless, the implicit feedback data also has value in scenarios of CF (Yifan Hu, Koren, & Volinsky, 2008).

## 2.3 Recommender System Approaches

The fundamental approaches which used by RS succinctly described in this section. The RS approaches are usually classified according to how recommendations are made into three approaches, which found in the most common division in the literature: (G. Adomavicius & A. Tuzhilin, 2005; Balabanović & Shoham, 1997; Bobadilla et al., 2013; Lü et al., 2012). These three approaches are Content-based (CB), Collaborative Filtering (CF) and the hybrid approach. In the first one, the system suggests the items which are like the items preferred previously by the current user (Aggarwal, 2016; Balabanović & Shoham, 1997; Mooney & Roy, 2000; M. J. Pazzani & Billsus, 2007a; Van den Oord, Dieleman, & Schrauwen, 2013). The recommendations rely on the content or features of the items in the recommendation domain. Whereas, in the second one, the system proposes the items depends on the users who have similar preferences to the current user (Ekstrand, Riedl, & Konstan, 2011; J. L. Herlocker et al., 2004; Resnick et al., 1994; Sarwar et al., 2001; Schafer et al., 2007). In this case, the feedback provided by the user utilises to find the neighbourhoods of target user/item. The hybrid approach combined both approaches (Claypool et al., 1999; Kim et al., 2006).

In the next subsections, the researcher will turn to describe each group of the recommender approaches, arguing their advantages and disadvantages. In particular, the researcher makes a particular focus on CF approach as our work based on this type of approach.

### 2.3.1 Content-based Approach

In the CB, the recommender systems construct the profile of users depends on the content of the items that rated by those users (Aggarwal, 2016; Balabanović & Shoham, 1997; Lops, De Gemmis, & Semeraro, 2011; Mooney & Roy, 2000; M. J. Pazzani & Billsus, 2007a; Van den Oord, Dieleman, & Schrauwen, 2013). Unlike CF, CB does not use the preferences of other users in order to generate recommendations. CB analyse the content of items to determine the most similar elements to those preferred by the target user (G. Adomavicius & A. Tuzhilin, 2005; Jain, Khangarot, & Singh, 2019). The items' content can be, for example, the actual content of the item in the case of text-based elements (e.g. books, news articles, Web pages); some item's attributes or features as shown in (e.g. "genre" and "actors" attributes of a movie, and "location" and "service" features of a hotels), or user-provided short characterization given to the element (e.g. comments or notes about items in a social media).

As one primary advantage of this approach, contrary to CF approach, they can treat smoothly with cold-star problem (will be discussed in the coming section). That is mean, the items which recently added into the system can be recommended to the users, even if there is no user rate it yet. However, the CB heavily reliant on the domain of recommendation, which contrasts with the generality principle in CF approaches (G. Adomavicius & A. Tuzhilin, 2005). In addition, these type of systems also depends on the availability of the correct information about the features of the elements. In this case, the system sometimes is costly to obtain this information, which is another drawback (G. Adomavicius & A. Tuzhilin, 2005). Eventually, there is a mainly suffering from specialisation in this type of methods. Due to it recommend just the items which have more similar value to items already rated by the user (G. Adomavicius & A. Tuzhilin, 2005; Chen, Chen, & Wang, 2015; Montaner Rigall, 2003).

Typically, the CB method perspectives are derived from several fields, such as information retrieval filed and semantic Web technologies. For example, web recommendation proposed early by Balabanović and Shoham (1997), based on the term-weighting model, which is information retrieval model, other cases are news recommender (Lang, 1995) and social tagging systems proposed by Cantador, Bellogín, and Vallet (2010). Moreover, semantic Web technologies utilised on recommendation system, for example, news recommendation was introduced by Cantador, Bellogín, and Castells (2008), or movie and music recommendations leveraging Linked Open Data (Ostuni et al., 2013). Another technique that used in content-based approach is Machine Learning techniques. There are several Machine Learning algorithms such as artificial neural networks, Bayesian classifiers, decision trees and clustering used for web recommendation system (M. Pazzani & Billsus, 1997). For instance, Mooney and Roy (2000) used Bayesian classifiers model to specifies the Web pages as user' preferences or not based on a set of pages previously rated by the user. This kind of recommender system approach is not within the scope of this study, so we did not go further into details.

### 2.3.2 Collaborative Filtering Approach

The term of Collaborative Filtering was coined by Goldberg in the recommender system Tapestry (D. Goldberg et al., 1992). It has become one of the most widely used approaches to provide recommendation service for users in many online web systems (Huang & Dai, 2015; Nagarnaik & Thomas, 2015; Polatidis & Georgiadis, 2016; SONG, 2018). CF approaches suggest items to the users that they interest in based on their feedback, which differs from the ones earlier that need to analyse the content of elements. This main advantage makes the CF more effective than CB approaches (Su & Khoshgoftaar, 2009). In addition, the independence of CF approaches makes them applicable in several domains. It collects information about user activities, history or/and click pattern to build a knowledge of user's preferences, which used later to provide what the user will prefer depending on his similarity with other users (G. Adomavicius & A. Tuzhilin, 2005). For example, the recommender system in amazon web collects the items that similar to those purchased by the user and their ratings to generate recommendation list (Linden, Smith, & York, 2003). As CF is the focused approach in this work, a separate heading will be designated for it.

To conclude, Table 2.2 illustrates the advantages and disadvantages of content-based and collaborative filtering approaches.

Table 2.2 Content-based vs Collaborative filtering.

| Approach | Advantages | Disadvantages |
|---|---|---|
| Content-based | The feedback providing by other users not required.<br>Has the capability to provide a recommendation for new users who do not have ratings.<br>Can recommending new items to users. | The content of items (feature extraction and representation) need to be analysed.<br>Overspecialization: The items are limited to their original feature. Therefore, the user is limited to getting items similar to those already rated in the past. |
| Collaborative filtering | No need to analyse the content of items.<br>Has the capability to provides items with different content with those previously rated.<br>Applicable to deal with any content. | It suffers from data sparsity.<br>Difficult to locate the nearest neighbours for users who do not have enough ratings.<br>Finding correlation among users depends on feedback and the similarity measure. |

### 2.3.3 Hybrid Approaches

Due to the different advantages and disadvantages of recommendation techniques, the common way in RS is to combine more than one methods in hybrid method (Burke, 2002; Passi, Jain, & Singh, 2019). The primary goal of this combination is to overcome individual limitations. Several researchers have attempted to combine CF and CB, also user-based and item-based methods, to avoid their weaknesses and enhance the quality of the system. The hybrid RS approaches grouped into seven different groups according to (Burke, 2002, 2007), as shown in Table 2.3.

Table 2.3 Hybridization Methods

| Hybridization method | Description |
| --- | --- |
| Weighted | The scores of several recommendation techniques are aggregated using a voting scheme or linear combination to generate a single recommendation. |
| Switching | This is a particular case of weighted recommenders, where the system switches among available recommendation techniques depending on the present state. One recommendation technique is switched off whenever the other is switched on. The switching decision in this type of hybrid approaches needs some criterion to be more accurate. |
| Mixed | Presents the results of different recommendation algorithms together at the same time. This method same weighting, but the results are not required united into a single list. |
| Feature combination | Collects features derived from different knowledge sources into a single recommendation algorithm. |
| Cascade | The recommendation is performed as a sequential process using different recommendation algorithms. |
| Feature augmentation | The input feature of the current algorithm was an output of other algorithms. |
| Meta-level | The model learned by one recommender is used as input to another. |

Unfortunately, the hybrid approach is not suitable to be in this research. Since hybrid is constructed to overcome the issues found in the two combined methods such as content-based & collaborative filtering. Whereas, this research aims to solve the problem of data sparsity and improve the accuracy of recommendation. Therefore, the researcher finds that using CF is sufficiently enough to be used to address these issues. However, application of hybrid approach in addressing the specified problem of this research will require more effort. Consequently, the performance will be impacted in terms of more inputs, much-complicated process and time-consuming to execute the complicated process compared to CF.

## 2.4 Collaborative Filtering Models

CF is one of the most successful approaches used in RS (Koohi & Kiani, 2017). It can be categorized into two types of models:

I. **Model-Based CF** Model**:** there is a pre-built model that is used later to predict what the user like.

II. **Memory-Based CF** Model**:** the correlations between users/items are directly used to predict the preference that a user would give to an item in the future and based on this prediction make the recommendations for users.

### 2.4.1 Model-Based CF Technique

CF technique, extract a learned model of user preferences based on the observed data (such as ratings), and afterwards, use this model to make recommendations to an active user (G. Adomavicius & A. Tuzhilin, 2005). Compared with memory-based technique, the model-based save CPU time and Memory space which make it better scalability and performance than memory-based. The most widely-used models were proposed in (Breese, Heckerman, & Kadie, 1998; Ungar & Foster, 1998) which are Bayesian network model and cluster models. Posteriorly, latent factor models were proposed in (T. Hofmann, 2004; Koren, Bell, & Volinsky, 2009). Latent factor models reduce the dimensionality of the user-item matrix. For recommendation purposes, this model uses a set of latent variables to explain the preferences of the user. The model-based approach is not related to this work, so we did not detail it.

### 2.4.2 Memory-Based CF Model

Memory-based model distinguished by their simplicity, since no need to build a learning model. This makes the implementation of memory-based and including new data easier. On the other hand, this type of methods have shortcomings, such as the performance become slow when the dataset is massive, lack of sensitivity to sparse data and may suffer from scalability issues (Ahn, 2008; Liu et al., 2014; Patra et al., 2015). In this case, pre-computing correlations can address the issue of performance and update it. For sparsity data issue, there are several studies proposed solutions to overcome this problem but not succeed yet (Koohi & Kiani, 2017; Suryakant & Mahara, 2016; B. Zhang & Yuan, 2017). Memory-based CF model is known as neighbour's methods, which are categorized into user-based and item-based according to space used to find the correlation (Su & Khoshgoftaar, 2009).

In the user-based methods, generating recommendations depending on the similarities between users (Sarwar et al., 2001). The main idea is that, for a target user, the items that preferred by similar users, target user's neighbours, can be introduced as recommendations to the target user in the future. While in the case of item-based methods, computing the similarity between items to find the most similar elements to the items rated by the active user. Memory-based find the neighbours of the target user or item using several methods and use their ratings on other elements to predict what the target user will prefer. Traditional similarity measures such as Pearson Correlation Coefficient (Resnick et al., 1994), Mean Squared Difference (Shardanand & Maes, 1995) and Cosine (Balabanović & Shoham, 1997) are the most popular similarity measures used for calculating the similarity between a pair of users or items. Similarity computation is depending on the ratings made by the pair of users on the co-rated items in case of user-based. Likewise, the item-based approach uses the ratings provided by users who rated both items to calculate the similarity between them. That is mean, the performance of these correlation methods entirely depending on the number of common ratings between a pair of users or items, which might perform unwell if there are insufficient amounts of common rating. For example, two users can be similar, even if they do not have co-rated items. Likewise, two items may similarly if there is no user evaluates both of them (Patra et al., 2015). Consequently, these similarity methods are not appropriate in the issue of data sparsity. Up to now, there are several studies which focused on this issue and proposed new similarity measures to alleviate the problem of data sparsity and enhance the accuracy. The researcher will go to discuss that in the next related work section.

On the other hand, to compute the predictions to generate recommendations for an active user, in the case of user- based, CF first utilise similarity function to calculate the similarity between users and an active user. Then, based on the similarity weights the system chooses k users who have high similarity to represent the neighbours. The system uses these similarities with their ratings to predict the degree of preference for the active user on an item which not selected yet. Typically, there are various methods used to compute predictions, such as weighted averaging functions which can be defined as shown in Table 1.4 chapter one. The next heading will discuss in detail the main issues found in the existing common memory-based CF methods.

## 2.5 Traditional Memory-Based CF Technique

In the traditional memory-based CF techniques, the recommendations are generated based on analysing the user's historical behaviour information such as users' ratings. This technique can be split up into four main steps: representing the data, finding the k-nearest neighbours, predicting the score of rating and generate recommendations. Therefore, the most critical steps are computing the similarity and the predicting score of rating.

The most widely used similarity methods in traditional memory-based CF for finding relationship between users (items) will be represented in Table 2.4 (Ahn, 2008; Choi & Suh, 2013; Kg & Sadasivam, 2017; Liu et al., 2014; Pirasteh, Hwang, & Jung, 2015; Suryakant & Mahara, 2016; Wu & Zheng, 2010; B. Zhang & Yuan, 2017).

Table 2.4 Memory-based CF similarity measures.

| **Author & year** | **Method** | **2017** | **2016** | **2015** | **2014** | **2013** | **2012** | **2011** | **2010** |
|---|---|---|---|---|---|---|---|---|---|
| Resnick et al. (1994) | PCC | 102 | 416 | 445 | 447 | 481 | 458 | 432 | 431 |
| Shardanand and Maes (1995) | CPCC & MSD | 36 | 177 | 172 | 188 | 243 | 260 | 217 | 261 |
| Jonathan L Herlocker et al. (1999) | WPCC | 58 | 259 | 264 | 303 | 249 | 269 | 224 | 212 |
| Balabanović and Shoham (1997) | Cosine | 49 | 222 | 227 | 277 | 342 | 338 | 272 | 276 |
| Sarwar et al. (2001) | ACosine | 176 | 688 | 709 | 705 | 654 | 620 | 496 | 472 |
| (J. Herlocker, Konstan, & Riedl, 2002) | SRCC | 13 | 62 | 58 | 91 | 66 | 59 | 59 | 48 |
| Ahn (2008) | PIP | 13 | 63 | 65 | 62 | 67 | 45 | 55 | 41 |
| Koutrika, Bercovitz, and Garcia-Molina (2009) | Jaccard | 5 | 17 | 15 | 12 | 12 | 7 | 12 | 7 |
| Jamali and Ester (2009) | SPCC | 29 | 117 | 100 | 117 | 86 | 60 | 36 | 27 |

Table 2.4 continued.

| Author & year | Method | 2017 | 2016 | 2015 | 2014 | 2013 | 2012 | 2011 | 2010 |
|---|---|---|---|---|---|---|---|---|---|
| Bobadilla, Serradilla, and Bernal (2010) | JMSD | 6 | 22 | 32 | 26 | 34 | 24 | 15 | 1 |
| Bobadilla, Ortega, et al. (2012) | MJD | 14 | 53 | 49 | 30 | 23 | 9 | * | * |
| Liu et al. (2014) | NHSM | 26 | 57 | 38 | 10 | * | * | * | * |
| Patra et al. (2015) | BCF | 7 | 11 | 4 | * | * | * | * | * |

From Table 2.4 the inclusion and exclusion criteria of selected comparative methods depended on the following criteria: it should be RS, CF, memory-based, user-based method, solving the sparsity issue and the number of citations is within the last six years as sub-criteria. Therefore, other similar methods which not fit the aforementioned criteria were excluded, for instance, the SRCC measure was excluded as not more popular method compare to CPCC. Additionally, PIP and MJD similarity methods were excluded because they were developed to deal with the cold-user problem only. While BCF similarity method is less popular compared to NHSM as the newest similarity method, therefore, it was excluded. Regarding MSD and Jaccard, they are combined to produce an improved similarity method called JMSD, so they also ignored. Moreover, the ACosine method was developed by Sarwar et al. (2001) as an item-based and not a user-based method, so it was excluded too. WPCC also was not included because a slightly modified version of PCC that based on constraint applied by multiplying the PCC similarity measure by a significant factor to scaling the similarity when the number of co-rated items is smaller than a given threshold. The next section will discuss the base related work and the further related work.

### 2.5.1 Base Related Work Review

As mentioned earlier, in traditional memory-based CF technique, the core stage is that how to locate the successful k-neighbours (computing the similarity). Therefore, the neighbours' formation phase is a critical stage in memory-based CF implementation, user/item-based CF method. Finding the neighbours depends on the similarity measure chosen, which use the feedback of each pair of users to find the correlations between them. Based on these relationships the most similar users to an active user are determined as much as possible. Since the quality of similarity measure has a significant impact on the accuracy of recommendation, many similarity measures have been developed in the literature. In this section, the author will introduce and analyse the shortcomings of the popular widely used similarity methods pointed in Table 2.4. On this basis, the researcher will identify the improvement factors to propose an improved memory-based CF method.

I. Pearson's Correlation Coefficient (PCC) measure

Resnick et al. (1994) applied similarity measure PCC to find the nearest neighbours. PCC is a popular measure in memory-based CF. PCC is a measure that computes the linear correlation between two objects, its outcome a value between +1 and −1, where one is the total positive correlation, 0 is no correlation, and −1 is the complete negative correlation. PCC is a statistical method used in CF to calculate the correlation between a pair of users based on co-rated of both users. The relationship calculated by the Equation 2.1 as can be shown:

$$S(x,y)^{PCC} = \frac{\sum_{i\in I}(r_{x,i} - \overline{r_x})(r_{y,i} - \overline{r_y})}{\sqrt{\sum_{i\in I}(r_{x,i} - \overline{r_x})^2}\sqrt{\sum_{i\in I}(r_{y,i} - \overline{r_y})^2}} \qquad 2.1$$

Where $(x, y)$ is the similarity between user *x* and user *y*, *I* represent a set of items which rated by both users *x* and *y*. The symbols $\overline{r_x}$ and $\overline{r_y}$ symbolize the average rating of user *x* and *y*, respectively. $r_{x,i}$ denotes to the rating value of the item *i* by the user *x*.

However, PCC similarity method has a significant influence on the sparsity of data case. Due to it is dependent over the rating matrix, therefore, the missing ratings in the rating matrix make finding the correlation between users more difficult and may lead to high/low similarity in turn to weak recommendations (Koohi & Kiani, 2017; Liu et al., 2014; Suryakant & Mahara, 2016).

II. Constrained Pearson Correlation (CPCC)

Ringo system has been developed as a new technique networked system which makes recommendations on music albums and artists to provide to its users. It requests ratings from its users with a nominal scale 1 to 7, a strong like represented by higher values in an item and a strong dislike represented by low values. While the four value represents a neutral value that is neither like nor dislike. As more users utilise this system and more information inserted the database of Ringo increased sharply. This led

Shraddhanand and Mae's to propose the CPCC which replaced the average rating variables in a PCC by the median value of the rating scale for considering the impact of positive and negative ratings (Shardanand & Maes, 1995), as it is shown in the following Equation 2.2:

$$S(x,y)^{CPCC} = \frac{\sum_{i\in I}(r_{x,i} - r_m)(r_{y,i} - r_m)}{\sqrt{\sum_{i\in I}(r_{x,i-}r_m)^2}\sqrt{\sum_{i\in I}(r_{y,i} - r_m)^2}} \quad 2.2$$

Where $r_m$ denotes to the median value of the rating scale.

However, CPCC is known to be vulnerable to data sparsity. Therefore, the similarity output of CPCC will be high/low when the number of co-rated between the pair of users very small (Ahn, 2008; Patra et al., 2015). For example, If the pair of users do not have common items the similarity value will be zero.

III. Mean Squared Difference (MSD)

Another similarity measure has been proposed by Shardanand and Maes (1995), such as MSD, which not popular and it has not seen considerable adoption. However, it cannot find the similar users, especially for users with very few ratings, which makes the opportunity of having common elements between them very small. In addition, the proportion of common rating not considered in MSD also may lead to low accuracy (Patra et al., 2015). The MSD is defined as shown in the Equation 2.3:

$$S(x,y)^{MSD} = \frac{|I_{xy}|}{\sum_{i\in I}(r_{x,i-}r_{y,i})^2} \quad 2.3$$

Where the cardinality $|\boldsymbol{I}_{xy}|$ represents the number elements which rated by user *x* and *y*.

IV. Weighted Pearson Correlation Coefficient (WPCC)

The high correlation with a few numbers of co-rated between a pair of users considered a weakness for PCC. Jonathan L Herlocker et al. (1999), in order full agreement and scaling the similarity when the number of common items is not enough, they added a significant factor ( $\min(I_{x,y},\gamma)/\gamma$) to devalue the similarity weights when the number of common items small. In their experiments, they applied the significance weights if the users have fewer than certain threshold commonly co-rated items. They used threshold $\gamma$ to determine the minimum number of co-rated items. Their experiments have shown that when $\gamma \geq 25$ the accuracy of the predicted ratings have been improved and the best result was when the value of 50 for γ. However, the proportion of common rating not considered (Polatidis & Georgiadis, 2016). Thus, the sparsity issue still has an influence on determining the most similar users who have common items bigger than the threshold. For example, if there are two pairs of users with 70 and 50 common items, respectively. Then, based on WPCC method, the significant factor value of two pairs will be 1. They just applied constraint by multiplying the similarity measure by a significant factor as shown in Equation 2.4. Where $\gamma$ represent a threshold.

$$S(x, y) = \frac{min(I_{x,y}, \gamma)}{\gamma} * s(x, y)^{pcc} \tag{2.4}$$

V. Cosine similarity measure

The Cosine, Equation 2.5, utilised to compute the similarity between two users (Balabanović & Shoham, 1997), which is a vector-space model that depending on the linear algebra approach instead of a statistical method. The users represented as vectors and the similarities are computed by the Cosine distance between a pair of rating vectors. The value of similarity ranges from 0 meaning high dissimilar to 1 meaning they have high similarity. However, if the rating vectors of the pair of users on the same line (Cosine of 0° is 1), the similarity will be set to 1 according to Cosine regardless of the difference between both users. Additionally, cannot find the correlation between a pair of users if they do not have common items. It also suffers from few co-rated issues that may impact the accuracy. Additionally, it gives high similarity even if there is a significant difference in ratings. (Patra et al., 2015; Suryakant & Mahara, 2016).

$$S(x, y)^{Cosine} = \frac{\sum_{i \in I} (r_{x,i})(r_{y,i})}{\sqrt{\sum_{i \in I} (r_{x,i})^2} \sqrt{\sum_{i \in I} (r_{y,i})^2}} \tag{2.5}$$

VI. Adjusted Cosine (ACosine)

In item-based case, finding the relationship between items using Cosine method not take into consideration the differences in rating scale between a pair of users which lead to an unacceptable result. This shortcoming addressed by the ACosine method by subtracting the corresponding user average from each co-rated pair (Sarwar et al., 2001). Item-based CF method proposed to achieve high coverage in the sparsity of data and addressing the long-scale problem in user-based technique. In this method, the scheme of calculating similarity is correlation-based, but it executed on item space instead of user space. In general, the method passes through three main steps. Started by finding the relationship between items based on the user-item matrix, using these similarities to predict the score for a user on the item, and then generate the recommendation based on the prediction scores. However, as a limitation, finding the similarity between two items become impossible if no single user who rated both items or the common rated items is small. It outputs low (high) similarity regardless of similar (significant difference in) the ratings (Patra et al., 2015; Suryakant & Mahara, 2016). The similarity between a pair of items, such as item *i* and *j*, given by Equation 2.6:

$$S(i,j)^{ACosine} = \frac{\sum_{u \in U_{ij}} (r_{u,i} - \overline{r_u})(r_{u,j} - \overline{r_u})}{\sqrt{\sum_{u \in U_{ij}} (r_{u,i} - \overline{r_u})^2} \sqrt{\sum_{u \in U_{ij}} (r_{u,j} - \overline{r_u})^2}} \quad 2.6$$

Where $U_{ij}$ a set of users who rated both items *i* and *j*.

VII. Spearman Rank Correlation Coefficient (SRCC)

Another presented similarity measure is SRCC (J. Herlocker, Konstan, & Riedl, 2002). For the Spearman correlation, the rated items by a user will be ranked such that their highest-rated item is at rank 1 and lower- rated items have higher ranks. The items with the same ratings will get the average of the ranks for their position. SRCC is like PCC but computes a measure of correlation between ranks instead of rating values. However, the SRCC give high relationship even if the ratings are similar (Ahn, 2008; Suryakant & Mahara, 2016). The relationship between user x and y can be computed using Equation 2.7

$$S(x,y)^{SRCC} = \frac{\sum_{i \in I} (rank_{x,i} - \overline{rank_x})(r_{y,i} - \overline{rank_y})}{\sqrt{\sum_{i \in I} (rank_{x,i} - \overline{rank_x})^2} \sqrt{\sum_{i \in I} (rank_{y,i} - \overline{rank_y})^2}} \quad 2.7$$

VIII. PIP (Proximity-Impact-Popularity)

Ahn (2008) introduced a heuristic measure for collaborative filtering that is called PIP similarity measure, as it is shown in Equation 2.8. Which can enhance the accuracy of RS under cold-star conditions especially when just a few of co-ratings are available, it utilised all ratings given by a set of users. The weakness of traditional CF similarity methods (PCC and Cosine Similarities) have been analysed in this study. PIP similarity measure consists of three factors of similarity that play an essential role in identifying the most similar users to an active user. These factors are proximity, impact, and popularity of the user evaluations which are local information about the evaluations, see Figure 2.4. PIP method takes into account just the local knowledge of the ratings. However, PIP has weaknesses which are: not considers the global preferences, suffers from few co-rated items issue, ignore the proportion of common rating, and the similarity computation is too much (Kg & Sadasivam, 2017).

$$S(x,y)^{PIP} = \sum_{i \in I} PIP(r_{x,i}\,, r_{y,i}) \qquad 2.8$$

Where the PIP can be defined as follows:

$PIP\ (r_{x,i}\ , r_{y,i})$ = Proximity $(r_{x,i}, r_{y,i})$ ∗ Impact $(r_{x,i}, r_{y,i})$ ∗ Popularity $(r_{x,i}, r_{y,i})$.

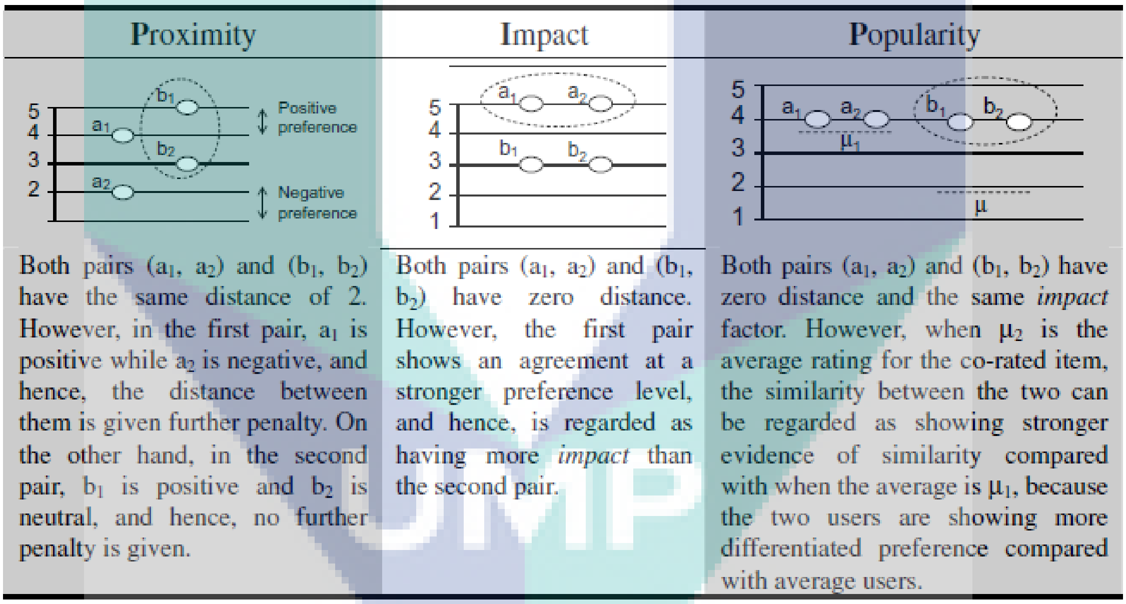



Figure 2.4 The description of the basic idea of the PIP

Source: Ahn (2008).

IX. Jaccard measure

Another measure, Jaccard similarity measurement, which was proposed by Koutrika, Bercovitz, and Garcia-Molina (2009). This measure takes into consideration only the number of co-rated ratings between a pair of users. Jaccard's main idea is that the correlation between two users depends on the number of a common rating among them. Therefore, if a pair of users has large common ratings, then they will be more similar. However, Jaccard's computation process depends only on the proportion of common rating and does not consider the absolute value of rating (Kg & Sadasivam, 2017; Patra et al., 2015). This makes it difficult to distinguish between users. Formally, the similarity between user *x* and *y* using this measure is given by Equation 2.9:

$$S(x,y)^{Jaccard} = \frac{|I_x \cap I_y|}{|I_x \cup I_y|} \tag{2.9}$$

Where $|I_x \cap I_y|$ is the number of items rated by both users *x* and *y* and $|I_x \cup I_y|$ represent the number of union items that rated by user *x* and *y*.

X. Sigmoid Function based Pearson Correlation Coefficient (SPCC)

Jamali and Ester (2009) proposed a weighted item-based similarity measure that based on the sigmoid function, which can be defined as shown in Equation 2.10. The sigmoid function addresses the problem when the number of common users who rated both items is small. It devalues the similarity value and keeps the similarity value in the range [0, 1]. In general, if the size of the set of common users who rated a pair of items is large enough, then the weight value would be 1, but for small sets of common users, the weight value would be 0.6. The denominator equals 2 in the exponential function in order to be the weight value higher than 0.9 when the set of common users is greater than 5. The disadvantage of SPCC is that, if there is a pair of users has similar ratings, they may get low similarity. For example, let the ratings of the user be represented as a vector of ratings where $u1 = (4,3,5,4)$ and $u2 = (4,3,3,4)$, it can be noted that they have very similar ratings. However, the similarity between them will be zero when computed using SPCC. Because of the output of PCC numerator is zero.

$$S(x,y)^{SPCC} = s(x,y)^{pcc} . \frac{1}{1 + exp^{(-\frac{|I|}{2})}} \tag{2.10}$$

XI. Jaccard and Mean Squared Difference measure (JMSD)

Moreover, Bobadilla, Serradilla, and Bernal (2010) joined two similarity measures, the Jaccard (Koutrika, Bercovitz, & Garcia-Molina, 2009) method and mean squared difference measure (Shardanand & Maes, 1995), to produce a new similarity measure as JMSD, which can be computed as shown in Equation 2.11. They assumed that the combined of two methods produce a new method which can enhance the performance and quality of the system. Due to the new method can avoid the limitations of each method. While the Jaccard measure considers only the proportion of common ratings between two users as similarity value between them and ignoring the actual rating value of the user, which is regarded as a drawback. The MSD only considers the actual ratings, but not considering the proportion of common ratings. It mainly addresses the shortcomings of Jaccard and MSD, but this method suffers from cold user problem. Moreover, it does not consider the credibility of the common ratings and suffers from local information and utilisation of rating problems (Patra et al., 2015)

$$S(x,y)^{JMSD} = s(x,y)^{MSD} + s(x,y)^{Jacc} \quad 2.11$$

XII. Mean–Jaccard– Difference (MJD)

Bobadilla, Ortega, et al. (2012) proposed a new similarity method that is called MJD (Mean–Jaccard– Difference), to address the new user problem. This method combined three methods: mean squared difference in rating method, Jaccard similarity measure and the difference measure of ratings between both users as shown in Equation 2.12. Where each measure of them has weight, which can be obtained based on the neural network learning. Lastly, the recommendation can be generated according to the new measure values. That used in the stage of prediction to estimate the score for a user on an item which not chosen yet.

$$S(x,y)^{MJD} = 1/6[w1 * V_{x,y}^{0} + w2 * V_{x,y}^{1} + w3 * V_{x,y}^{3} + w4 * V_{x,y}^{4} + w5 * \mu_{x,y} + w6\, s(x,y)^{Jacc}] \quad 2.12$$

Where:

$V_{x,y}^{0}$ is the number of items in which the user x and user y have voted with exactly the same score.

$V_{x,y}^{1}$ is the number of items with a difference of 1 stars in user x and user y.

$V_{x,y}^{2}$ is the number of items with a difference of 2 stars in user x and user y.

$V_{x,y}^{3}$ is the number of items with a difference of 3 stars in user x and user y.

$V_{x,y}^{4}$ is the number of items with a difference of 4 stars in user x and user y.

$\{w1, \ldots, w6\}$ represent the similarity measure weights.

$\mu_{x,y}$ is the mean squared differences Equation 2.13.

$$\mu_x = 1 - \frac{1}{\#G_{x,y}} \sum_{i \in G_{x,y}} \left(\frac{r_{x,i} - r_{y,i}}{max - min}\right)^2 \Leftrightarrow G_{x,y} \neq \emptyset, \quad \mu_x \in [0,1] \tag{2.13}$$

XIII. New Heuristic Similarity Model (NHSM)

Liu et al. (2014) analysed the drawbacks of PIP, in work (Ahn, 2008), and proposed an improved heuristic similarity model called NHSM which considered an improvement PIP measure. This model considered three factors: proximity, significance, and singularity of the user ratings. It's combined the local context information of the ratings and the global preferences of user ratings to alleviate the user cold-star problem. However, NHSM measure only considers the co-rated items to find the relationship between users using. Thus, the ratings on non-co-rated items are neglected in this method (Kg & Sadasivam, 2017). The NHSM similarity measure is as defined in Equations 2.14 - 2.22.

$$S(x,y)^{PSS} = \sum_{i \in I} PSS(r_{x,i}, r_{y,i}) \tag{2.14}$$

Where the $PSS(r_{x,i}, r_{y,i})$ the PSS value of user *x* and *y*, it is defined as follows:

$$PSS(r_{x,i}, r_{y,i}) = \text{Proximity}(r_{x,i}, r_{y,i}) * \text{Significance}(r_{x,i}, r_{y,i}) * \text{Singularity}(r_{x,i}, r_{y,i})$$

Where proximity, significance, and singularity aspects can be defined as shown in Equations 2.15, 2.16, and 2.17, respectively.

$$Proximity(r_{x,i}, r_{y,i}) = 1 - \frac{1}{1 + exp(-|r_{x,i} - r_{y,i}|)} \tag{2.15}$$

$$Significance(r_{x,i}, r_{y,i}) = \frac{1}{1 + exp(-|r_{x,i} - r_{med}| * |r_{y,i} - r_{med}|)} \tag{2.16}$$

$$Singularity(r_{x,i}, r_{y,i}) = 1 - \frac{1}{1 + exp(-|\frac{r_{x,i} - r_{y,i}}{2} - \mu_i|)} \quad 2.17$$

Moreover, they combined PSS with the Jaccard measure as a new similarity measure to alleviate the small proportion of common ratings. That is called JPSS which defined as follows, Equation 2.18:

$$S(x,y)^{JPSS} = S(x,y)^{Jacc} * S(x,y)^{PSS} * \quad 2.18$$

Further, due to the different rating preferences provided by different users. Some of them like to give high ratings and others on the contrary, like to give low ratings. To consider this behaviour preference, they built knowledge about user preference using the rating mean and standard variance as can be shown in the Equation 2.19:

$$S(x,y)^{URP} = 1 - \frac{1}{1 + exp(-|\mu_x - \mu_y| * |\sigma_x - \sigma_y|)} \quad 2.19$$

Where $\sigma_x$ and $\mu_x$ is the mean rating and standard variance of user *x*, respectively. This can be defined as follow:

$$\sigma_x = \sqrt{\sum_{i \in I_x} (r_{x,i} - \bar{r_x})^2} / |I_x| \quad 2.20$$

$$\mu_x = \sqrt{\sum_{i \in I_x} (r_{x,i})} / |I_x| \quad 2.21$$

The final formalisation combined the equations of JPSS and URP s a new similarity measure that is called improved new heuristic similarity model (NHSM). It can be defined as shown in Equation 2.22:

$$S(x,y)^{NHSM} = S(x,y)^{JPSS} * S(x,y)^{URP} \quad 2.22$$

XIV. Bhattacharyya Coefficient (BCF)

Patra et al. (2015) introduced a new similarity measure, named Bhattacharyya Coefficient in CF (BCF), to find the relationship between a pair of users in case of sparsity data. The introduced method used Bhattacharyya measure to extract the global information. The global information is used to calculate the similarity between a pair of users especially in the case of data sparsity. Bhattacharyya measure utilised for finding the relationship between two rated items which address the sparsity of data. This mean, the BCF can find the similarity between two users who do not have co-rated items. Due to that, the BCF combines two similarities global and local similarity, as it's shown in Equation 2.23. The global similarity can find the similarity between two items even there are no one user rates both items and the local similarity compute the similarity between two ratings. BCF also uses either function $loc_{cor}$ or $loc_{med}$ to calculate the similarity between two ratings where the reference scale in function one and two is the average of user rating and ratings' median, respectively. However, ignoring the differences in users' rating habits of co-rated items is considered a weakness. Moreover, it unable to find the similarity between two users when all ratings of each user have same distances from the user's median rating (in $loc_{med}$) or the user's average rating (in $loc_{cor}$).

$$S(x,y)^{BCF} = S(x,y)^{Jacc} + \sum_{i \in I_x} \sum_{j \in I_y} BC\,(i,j) loc(r_{x,i}\,, r_{y,j}) \tag{2.23}$$

Where,

$$BC(i,j) = \sum_{h=1}^{m} \sqrt{\hat{I}_h * \hat{J}_h} \tag{2.24}$$

$$loc_{car}(r_{x,i}\,, r_{y,j}) = \frac{(r_{x,i} - \overline{r_x})(r_{y,i} - \overline{r_y})}{\sigma_x \sigma_y} \tag{2.25}$$

$$loc_{med}(r_{x,i}\,, r_{y,j}) = \frac{(r_{x,i} - \overline{r_m})(r_{y,i} - \overline{r_m})}{\sqrt{\sum_{k \in I_x} (r_{x,k} - \overline{r_m})^2} \sqrt{\sum_{k \in I_y} (r_{y,k} - r_m)^2}} \tag{2.26}$$

Where $\hat{I}_h$ =#h/#i , where #i is the number of users rated the item *i*, #h is the number of users rated item *i* with rating value *h*.

### 2.5.2 Further Related Work

There were several developed methods based on the aforementioned traditional methods. They will be discussed in this subsection:

Ma, King, and Lyu (2007) presented an effective method to overcome the problematic of sparsity issue. The new algorithm depends on the information of users, items or both to predict the missing value. Prediction method determines how the prediction relies on items-based, user-based prediction or based on both through adding λ parameter which tack value in the range [0, 1]. Also, they improved the PCC measure by adding a significant weight factor. This factor devalues the correlation between two users when the number of commonly rated items, not enough. The paper presented a formula to compute the significance weight, which takes the smallest value from the number of co-rated items and the minimum threshold value, divided by minimum threshold value. The minimum number of the co-rated items will be determined by the minimum threshold value. Also, they introduced a threshold η which overcomes the shortcoming of the N-Top neighbourhood's selection method. If the correlation between the current user and the active user is bigger than the threshold value, then this user selected as a neighbour.

Bobadilla, Ortega, and Hernando (2012) improved the traditional similarity method using contextual information. It is based on the analysis of the singularity of user ratings. In their study, the voting is classified into positive and non-positive. Then, the singularity values of each user and each item are calculated. After that, it combined this singularity value with the actual users' ratings to compute the similarity value. Their philosophy says that "if 95% of users voted positively for the item, the similarity derived (for this item) between two users who belong to the 5% (very singular) must be greater than the similarity derived between two users who belong to the 95% (not very singular)". However, this method still suffers when the number of co-rated is small. Moreover, Bobadilla, Hernando, et al. (2012) proposed a new way to compute the similarity between two users based on significance. Before computing the similarity between a pair of users, they calculated three types of significances: the significance of an item, the significance of each user for providing recommendations to other users and significance of an item for a user. After that, they used one of most common measures to compute the relationship between users, which are: PCC or Cosine similarity measures according to these significances.

Choi and Suh (2013) noted that when identifying the neighbours of an active user the similarity between a target item and each of the co-rated items should be taken into consideration, and for each different target item must get a different set of neighbours. Therefore, they introduced a combination of traditional methods to give a new similarity method, which uses the item similarity between an item and the target item to weight the rating of a user on the item. They combined three different types of methods to compute the item and user correlation respectively, which are PCC, Cosine and Distance. To calculate the relationship between items they used one of the traditional similarity methods. Nevertheless, the weakness of memory-based CF similarity measures still exists such as, if the two users do not have common items. Mao et al. (2013), introduced an improved similarity measure by adding the similarity impact factor ε to traditional similarity measure to alleviate the effecting of data sparsity problem in traditional memory-based CF. The impact factor ε represents the proportion of common items which rated by a pair of users. If the pair of users does not have co-rated items, then the similarity between them giving the value as zero, this considered as a weakness for this method. To improve the prediction method, the CF predicts the score for any user on any item which not rated yet through combines the similarity between users and items which are improved using the impact factor. While, Gu, Yang, and Dong (2014a) proposed a new similarity method depending on the clustering-based, to enhance the performance of traditional collaborative filtering. Firstly, taxonomy tree metric used to cluster the users based on the social information of users. It considered as a new clustering technique that deals with numerical and categorical information of users. Then, they propose an incorporation measure to calculate the similarity between users considering the groups they belong to. As a result, the recommendation system performance is improved. A weakness of this technique, it requires finding a relationship between attributes which is difficult if the relationship not clear enough.

Newly, Huang and Dai (2015), they formulated a novel similarity method utilised to compute the association between two users based on the relationship between the target item and co-rated items and the ratio of common ratings as a very important factor. The new similarity measure called Weight Distance Model (WDM) that consider both factors.

Firstly, the proportion of the common ratings calculates using Jaccard which is one of the most common methods that used to calculate the similarity between two sets. Secondly, the relationship between the target element and the co-rated elements also is computed by Cosine or personal correlation measure. If a pair of users does not have co-rated items, then the similarity between them zero which we can consider it as a limitation of this method. Another weighted similarity measure has been proposed by Pirasteh, Hwang, and Jung (2015). The new weighting measure takes into consideration new factors in the process of finding the neighbours for an active user. First, compromise factor which assigns an asymmetric value to the similarity between users based on the number of ratings by two users on non-co-rated items. Second, they also consider the effect of behaviour with the rated item as accordance factor in computing the similarity between users. Moreover, a singularity-based similarity measure (IPIJ) is introduced by Shunpan, Lin, and Fuyongyuan (2015). This measure hypothesised that, if two users rate items that are rated by only a few users, then they should have a strong relationship compared with when they evaluate items which are rated by a lot of users. Next, the singularity values of each user replaced the similarity with singularity value to improve the PCC method. The study divided the singularity into three kinds: positive singularity, negative singularity and empty singularity. Additionally, the Jaccard measure modified based on singularity in order to consider the proportion of common items. To get the final method they combined the improved PCC and Jaccard in two different methods: simple multiplication, and linear combination. El Alami, Nfaoui, and El Beqqali (2015) introduced a new method for neighbourhood selection depends on two heuristic approaches: intersection neighbourhood and union neighbourhood. In the first method, the users who rated the same items as the target user will be selected as the nearest neighbours. While in the second one, all users who rated one item at least as the target user are selected to represent the neighbours. They employed an adjusted similarity measure that combines PCC with Jaccard similarity. It is depending on choosing neighbours who rate the same items. Due to the poor finding two users evaluate the same number of items, they added a threshold which corresponds to a minimum number of common items in order to deal with this point.

In the last two years, several studies have been proposed. Suryakant and Mahara (2016) proposed a new combined similarity measure to improve the accuracy of recommendation under data sparsity. The new measure depends on Mean Measure of

Divergence that considers the rating behaviour (Some users tend to give high ratings and other prefer to give low ratings) of the user into account. The author combined PCC, Jaccard, and Measure of Divergence to calculate the similarity between users. Another linear combination has been presented for Web service recommendation by Saranya, Sadasivam, and Chandralekha (2016). The linear combination is used to combine PCC and Jaccard measures. Moreover, a simple multiplication combination based on attributes of items was proposed by Li et al. (2016) to increase the effectiveness of current memory-based CF methods. They generated two-tier weighting similarities which are *prosim* and *prosim**. Where *prosim* is multiplication combination included similarity measures Jaccard and Cosine measure. Whereas, *prosim** used the PCC instead of Cosine. Furthermore, a new weight similarity model called NWSM is proposed by Zang et al. (2016). NWSM similarity measure takes into account the proportion of common rating, the user rating preference and the different contributions of other users to the target. To improve the accuracy of recommendation the final similarity formula is got by integrating three factors which are: PCC with influence weight (neighbourhood's rating information); Jaccard to compute the proportion of co-ratings and the mean and variance of the rating to calculate the differences of preference of each user. Moreover, Cao et al. (2016) proposed an improved memory-based CF recommendation similarity measure based on Bhattacharyya Coefficient (BC) to solve the issue of sparse data. The proposed method combined the user neighbourhood information with the item neighbourhood information to enhance the recommendation accuracy. It is described as follows: (a) Finding the nearest neighbours of items through calculating the BC similarity among items. Then, take the top N items to identify the neighbourhood N of the target item. (b) locating the nearest neighbours of users using the similarity method that be used in (Patra et al., 2015). In addition, J. Zhang et al. (2016), presented a new effective collaborative filtering method in order to decrease the impact of the data sparsity issue depends on the user preference clustering. First, groups users who have different preferences into different clusters. Then, selecting the neighbour for the active user from corresponding user clusters. They grouped users into three different groupings: the users who favour assigning high ratings allocated into optimistic user cluster, the users who prefer to give low ratings grouped into pessimistic user cluster and the users have the trend to provide reasonable ratings for items allocated into neutral user cluster. Besides, they proposed a new similarity measure to compute the correlation between users, which consider the local and global user preferences.

In 2017, there are several studies introduced to improve the accuracy of memory-based CF. Koohi and Kiani (2017), to finds the best neighbours, they proposed a new method depends on the subspace clustering technique to address the problem of data sparsity and high dimensionality. The authors extracted three items subspaces which consist of Interested, Neither Interested nor Uninterested (NIU)., and Uninterested. Then, the similarity between users will be computed by comparing all the interested, NIU and uninterested items. Moreover, Bilge and Yargıç (2017) applied two different normalization methods, z-score and decoupling normalization, onto multi-criteria preference data in order to overcome negative effects of varying rating habits of users to enhance the accuracy of multi-criteria collaborative filtering. Additionally, B. Zhang and Yuan (2017), to overcome the problem of data sparsity, presented an improved the similarity method by analysing the shortcomings of traditional memory-based CF similarity method. In the enhanced similarity method, the relationship between the users' common rating items and all items that rated by the target user is considered. In the study (Kg & Sadasivam, 2017), the authors proposed a new linear combination similarity method called weight based modified heuristic similarity measure to overcome the problem of data sparsity. In this method, the global preferences, local context of the user behaviour and proportion of common ratings between two users are considered based on PSS, Bhattacharya Coefficient, and Jaccard, respectively. Recently, in order to improve the accuracy of recommendations, Feng et al. (2018) proposed an improved similarity method. To minimize the deviation of similarity calculation, the proposed method takes three impact factors of similarity into account.

### 2.5.3 Memory-Based CF Challenges and Limitations

Undoubtedly, the present state of RS research and development has contributed to improving the satisfaction of user and the success of business in different scenarios. Nevertheless, there are still a lot of open issues, limitations, and challenges that decrease the utility of the recommendations. We review now the common challenges that are relevant to the accuracy of the system and the existing efforts which proposed to overcome these challenges and how they affect the performance of the recommendations.

#### 2.5.3.1 Data Sparsity

The sparsity of data has a significant influence on RS in term of accuracy. This data is represented in the form of user-item matrix filled by rating values provided by users on items. Due to that, the number of users and items has sharply increased which makes dimensions of matrix very high and spare. The central reason behind sparsity of data is that the number of items, which are available on the Internet, has become vast and most of the users do not rate enough number of these items and the available ratings are usually sparse. Collaborative filtering method suffers from this problem, which considered one of the major issues faced by the CF RS because it is dependent on the rating matrix. Moreover, the missing ratings in user-item matrix make finding similarities between users or items more difficult and may lead to weak recommendations (Melville, Mooney, & Nagarajan, 2002). Several methods have been proposed in order alleviate this issue. Dimensionality reduction techniques, such as Singular Value Decomposition (SVD) (Billsus & Pazzani, 1998), which works based on the principle of that the unimportant users or items should be removed to decrease the dimensionalities of the user-item matrix. These studies helped to overcome sparsity but caused some other problems. When some users or items are ignored, useful information is ignored too which lead to a decline in the quality of the recommendation. In contrast, some studies proposed new similarity measures to deal with this issue. In early, Sarwar et al. in (2001) used Item-based collaborative filtering method to achieving high coverage in the sparsity of data. In (Ma, King, & Lyu, 2007), the study proposed an effective method to predict the missing ratings which depend on information of users, items or both. Moreover, Mao, J. et al. in (2013), presented an improved similarity method through adding the similarity impact factor $\varepsilon$, $\varepsilon$ represents the ratio of co-rated items rated by a pair of users, to traditional similarity measure to alleviate the effecting of data sparsity. Patra et al. (2015) proposed a new similarity measure using Bhattacharyya coefficient to extract the global information which was more useful especially in the case of data sparsity.

Moreover, in (2014), Sharifi et al. proposed a new method for treating data sparsity problem based on non-negative matrix factorization in RS which lead to better prediction. While, Gu, Yang, and Dong (2014a) proposed a new similarity method depending on the clustering-based, in order to improve the performance of traditional CF. In addition, J. Zhang et al. (2016), presented a new effective CF method to decrease the impact of the data sparsity issue depends on the user preference clustering. Moreover, J. Zhang et al. in (2016) to reduce the effect of data sparsity issue, they depend on user preference clustering.

#### 2.5.3.2 Cold Star

Knowledge acquisition of the user interest is also having the same level of importance in RS. In order to provide appropriate recommendations, the RS requires building knowledge about the user preferences. But if this user is a new member in the system, then the system knows nothing or very little about what the user preferences or is interested in. This is commonly called as the cold-start issue (Kluver & Konstan, 2014). It is considered as a type of sparsity problem that occurs when a new user or item has just entered the system. In case of CF approach, it is problematic to suggest in case of a new user; there is insufficient available information about the user. For a new item, not enough rating values are usually available. Therefore, this may lead to a weakly and not usefulness recommendation in case of new item as well as new user (G. Adomavicius & A. Tuzhilin, 2005; Schein et al., 2002; Yu et al., 2004). There are some techniques have been proposed to deal with this characteristic. In (2002), P. Melville et al. developed Content-boosted CF method as a hybrid CF to solution the cold start problems. Another example of Model-based CF methods is TAN-ELR (Greiner et al., 2005; Su & Khoshgoftaar, 2006). As a memory-based CF method, Bobadilla, Ortega, et al. (2012) introduced a new similarity method that is called MJD (Mean–Jaccard–Difference), in order to address the cold user problem. PIP (Proximity-Impact-Popularity), is a heuristic similarity measure for memory-based CF, also proposed by Ahn (2008) to improve the accuracy of RS under cold-star conditions.

#### 2.5.3.3 Scalability

The scalability becomes a crucial issue when the number of items and users tremendously increase (Sharma & Gera, 2013). It’s more difficult to provide recommendations in real time for millions of users which creates challenges in processes of data storage and algorithm computations.

Due to that, there will not be enough computational resources to satisfy the new requirements. There are some of the RS methods deal with growing number of users and items but accompanied by an increase in computations, get expensive and sometimes leading to inaccurate results. As an example, the dimensionality reduction techniques can handle this scalability problem, but it needs some process such as, construct matrix factorisation, which is a complex and costly step. In (Sarwar et al., 2002), the researcher proposed an incremental system to reduce the cost. Also, the parallel process can improve the performance (Zhanchun & Yuying, 2012). This challenge is not related to this work and just mentioned to rich the understanding.

#### 2.5.3.4 Other Challenges

Synonym problem, which occurs when there is more than one item has the same type but with different names. The recommender systems are usually not able to discover this relation between them and take these items differently. For example, children' movie and children film are the same type but with different names. In SVD technique, the Latent Semantic Indexing (LSI) can deal with this synonymy problem but still displays problem in some situations (Sharma & Gera, 2013). In addition, the RS must remove recommendations produced by the system which the system suggests as of interest to the user, but in fact, the user does not like it. These are known as false positives (Vozalis & Margaritis, 2003). Moreover, another issue, that is called Gray Sheep, related to the CF approach, especially, when using the concept of clusters. It occurs when some users will not consistently agree or disagree with any group of people, they cannot take the benefit of CF techniques. There was an approach to reduce this problem by having a per-user approach (Claypool et al., 1999).

### 2.5.4 Finding Summary

The researcher, in the previous section, analysed the shortcomings of the common existing memory-based CF methods. As discussed earlier in this chapter, finding the relationship between a pair of users using those memory-based CF methods mostly rely on the PCC or Cosine measures and their derivatives. From related work section, we can see that most of the enhancements were based on these traditional similarity measures. Nevertheless, these improved similarity methods still have drawbacks as listed next.

i. Similarity calculation of those methods depends on the user-item rating matrix. However, the similarity computation will suffer from few co-rated items issue. If the pair of users does not have common items, the similarity value will be zero.

Additionally, output high similarity when the number of the co-rated between the pair of users very small.

ii. It is true if we say that, the rating value reflect the interest of the user. But not all of them can rate carefully (Cheng et al., 2015). This means some users may rate randomly, therefore, give an untrue rating may lead to locating unsuccessful neighbours and in turn leads to weak recommendations. Thus, need to infer the global preference of those users to address this issue.

iii. The proportion of common ratings and absolute value not considered in MSD and Jaccard, respectively, will lead to low accuracy.

iv. Additionally, all ratings provided by both users do not utilise in the process of similarity calculation.

v. Calculating the similarity in some improved methods are too much which make the computation more complicated, such as PIP, MJD and PSS methods.

vi. Most improvements are focused on developing the similarity measure to enhance the accuracy of memory-based CF while only a few studies tended to develop the prediction method which is also on the same level of importance (Cai et al., 2014).

vii. The memory-based CF mechanisms still have an open room for development that leads to enhancing the quality of recommendation (Choi & Suh, 2013; R. Zhang et al., 2014).

In memory-based CF RS, the similarity and prediction methods play an essential role in the generating recommendations process. The commonly existing similarity measures have been used in memory-based CF such as Cosine similarity, Pearson correlation coefficient to find the correlation between users. Table 2.5 presented the common used existing similarity methods in memory-based CF with their drawbacks. And the list of primary drawbacks pointed with a simple illustration next:

**Drawback A**: The few/no co-rated items in the space of user-based CF.

The similarity measure cannot find the relationship between a pair of users/items if the number of common items not enough on the side of user-based. Similarly, on item-based space, if the number of users who rated the pair of items is few. The pair of users or items can be similar if they do not have a co-rated item or no single user who rated both items, respectively. For example, let $I = (2;\ 0;\ 3;\ 0;\ 4;\ 0;\ 1;\ 0;\ 2;\ 0)$ and $J = (0;\ 3;\ 0;\ 1;\ 0;\ 4;\ 0;\ 3;\ 0;\ 2)$ represent the vectors of ratings of item $I$ and $J$. We can see there is no user rate both the items. Thus, the similarity between them cannot be computed by these measures. Therefore, the small number of ratings by user led to decrease the size of co-rated items. Consequently, therefore, find the correlation between users/items become more difficult in the case of no single common item between users or no single user who rated both items, respectively. Additionally, may lead to fake relationships if the number of co-rated items very small. For example, if the number of common items between a pair of users is exactly one, then some measures, such as PCC or Cosine measure, cannot calculate the similarity between them. The Cosine output is 1 regardless of their ratings on the item and, PCC similarity between them is either -1 or 1.

**Drawback B:** Output high/low similarity (not reliable similarity) despite similar (significant difference in) ratings. That may lead to getting an untrue nearest neighbour.

*Low similarity*: A pair of users gets low similarity, although they have similar ratings. For instance, let the ratings of the user be represented as a vector of ratings where $u1 = (4, 3, 5, 4)$ and $u2 = (4, 3, 3, 4)$, it can be noted that they have very similar ratings. However, the similarity between them will be zero when computed using SPCC. Unlike, the similarity has slight enhancement using the CPCC (0.577). However, the ACosine measure also still suffers from this problem.

*High similarity*: A pair of users can obtain high correlation regardless of the difference between ratings of both users. For example, let the rating vectors of the pair of users $u1$ and $u2$ are $(5, 3, 0, 0)$ and $(2, 1, 0, 0)$, respectively. Nevertheless, the correlation between $u1$ and $u2$ using PCC is equal 1. While the correlation between them using SPCC is also very high, that will be 0.731. If two vectors are on the same line (Cosine of 0° is 1), the similarity will be set to 1 according to Cosine regardless of the difference between both users. For example, if the rating vectors of two users *u1* and *u2* are $(1, 2, 0, 0)$ and $(2, 4, 0, 0)$, respectively. The similarity value will be 1 when the Cosine measure used.

**Drawback C:** Ignore the proportion of common ratings.

Ignoring the proportion of common ratings in the process of similarity calculation between users may lead to low accuracy. For example, if the rating vectors of the $u1$, $u2$ and $u3$ are $(4,3,5,4), (5,3,0,0)$ and $(4,3,3,4)$, respectively. According to MSD the similarity value between $u1$ and $u2$ is 0.98 and, the similarity between $u1$ and $u3$ is 0.96. But, obviously, $u1$ and $u3$ should have a stronger correlation than the correlation between $u1$ and $u2$. This is because the MSD only calculates the average difference between both users and does not take into account the ratio of common ratings.

**Drawback D:** Ignore the absolute values of ratings.

Ignoring the absolute value of ratings makes it difficult to distinguish between users. For instance, let $u1 = (4,3,5,4)$ and $u2 = (4,3,0,0)$ be the rating vectors corresponding to a pair of users. According to Jaccard measuring the similarity will yield 0.5. Whereas, if the rating vectors for $u3 = (2,1,0,0)$, then the similarity value between $u2$ and $u3$ is 1. But, in fact, the similarity between $u1$ and $u2$ should be higher than $u2$ and $u3$. This is because the computation process of Jaccard depends only on the proportion of common rating, and does not consider the absolute value of ratings.

**Drawback E:** Do not take into account the global preference of the user.

These similarities take into account only the local information of the ratings and do not consider the global preference of user ratings/behaviour such as PCC, Cosine. Sometimes, the evaluation provided by the user is not accurate, because those users rate randomly. Therefore, this leads to locate unsuccessful neighbours and in turn low accuracy.

**Drawback F:** Utilization of all ratings.

Most of the existing similarity measures do not utilise all ratings provided by the both of users. For instance, traditional memory-based CF methods depend on co-rated items such as PCC, Cosine and their derivatives methods (CPCC, ACosine, etc.). Consider the uncommon items in the process of computing similarity assist in determining the right correlation between users who have few common items.

**Drawback G:** Similarity computations, more calculation would mean time-consuming which are considered a limitation in some cases such as PIP, NHSM, and MJD.

Table 2.5 Drawbacks of existing similarity methods

| Similarity Method | A | B | C | D | E | F | G | Reference |
|---|---|---|---|---|---|---|---|---|
| | Drawbacks | | | | | | | |
| PCC | ✓ | ✓ | ✓ | | ✓ | ✓ | | (Resnick et al., 1994) |
| CPCC | ✓ | | ✓ | | ✓ | ✓ | | (Shardanand & Maes, 1995) |
| WPCC | | | ✓ | | ✓ | ✓ | | (Jonathan L Herlocker et al., 1999) |
| SPCC | | ✓ | | ✓ | ✓ | ✓ | | (Jamali & Ester, 2009) |
| Cosine | ✓ | ✓ | ✓ | | ✓ | ✓ | | (Balabanović & Shoham, 1997) |
| SRCC | | ✓ | ✓ | ✓ | ✓ | ✓ | | (J. Herlocker, Konstan, & Riedl, 2002) |
| ACosine | ✓ | ✓ | ✓ | | ✓ | ✓ | | (Sarwar et al., 2001) |
| Jaccard | | ✓ | | ✓ | ✓ | ✓ | | (Koutrika, Bercovitz, & Garcia-Molina, 2009) |
| MSD | ✓ | | ✓ | | ✓ | ✓ | | (Shardanand & Maes, 1995) |
| PIP | ✓ | | ✓ | | ✓ | | ✓ | (Ahn, 2008) |
| JMSD | ✓ | | | | ✓ | ✓ | | (Bobadilla, Serradilla, & Bernal, 2010) |
| MJD | ✓ | | | | ✓ | | ✓ | (Bobadilla, Ortega, & Hernando, 2012) |
| NHSM | | | | | | ✓ | ✓ | (Liu et al., 2014) |
| BCF | ✓ | | ✓ | | | | ✓ | (Patra et al., 2015) |

## 2.6 Multi-Attribute Decision Making Method

As mentioned before, most of the studies improved the accuracy of CF just by improving similarity measure while only a few studies focused on the prediction score models which are also on the same level of importance (R. Zhang et al., 2014). In memory-based CF, after locating the target user's neighbours, the system collects their items and predicts the rating score that the target user would give to these items in the future. Next, these items will be ranked and recommended based on their predicting score. Therefore, the prediction method plays an important role. In this section, the researcher present Multi-Attribute Decision-Making Method (MADM) method as a suitable option to rank the candidate items instead of traditional way, prediction method.

In our daily lives, there are a lot of alternatives surrounding us and make the choosing process more complicated. Consequently, the decision-making becomes very difficult when we want to rank or choose the best option, the most preferred alternative of a decision-maker, from a set of available options. Typically, there are multiple criteria used to evaluate a set of alternatives in decision-making. For example, in purchasing a car, some of the main criteria such as cost, safety, comfort, and fuel consumption should be taken into consideration. Multi-Criteria Decision Making Method (MCDM) is one of the well-known topics of decision making, and it is necessary to use decision maker's preferences to differentiate between alternatives. In the literature, MCDM can be divided into two basic approaches (Kahraman, 2008): Multi-Objective Decision Making (MODM), and Multi-Attribute Decision making (MADM). MADM problems are distinguished from MODM problems by the number of alternatives have previously determined (Hwang & Yoon, 2012). It is a well-known type that treats with decision problems under the existence of many decision criteria to evaluate and rank several decision alternatives according to their attributes. It gathers the information together with additional information from the decision maker in the decision matrix and uses MADM method to determine the final ranking of alternatives (Xu & Yang, 2001). Whereas, In MODM approach, which differs from MADM, the number of alternatives is not given. Instead, design a set of decision alternatives using a mathematical framework which provided by MODM. Each alternative once identified, is judged by how close it satisfies an objective or multiple objectives. In the MODM approach, maybe there is a large number of potential decision alternatives (Kahraman, 2008). To conclude, MADM is useful in the evaluation side while the MODM is mainly appropriate for the design/planning side (Hwang & Yoon, 2012). Therefore, the MADM method will be selected in this work to evaluate the candidate items.

Hwang and Yoon, in their book (Hwang & Yoon, 2012), described several MADM methods one of them is Technique for Order Preference by Similarity to Ideal Solution technique (TOPSIS). TOPSIS is a practical and useful technique for ranking and selection of some externally determined alternatives through distance measures (Shih, Shyur, & Lee, 2007). It was initially presented by Yoon and Hwang (1981). The main relative advantages of TOPSIS are: the best alternative can be identified quickly (Parkan & Wu, 1997); perform almost as well as Simple Additive Weighting (SAW) and better than Analytic Hierarchy Process (AHP) (Olson, 2004); the simplicity and limited number of inputs are required from decision-makers, and its output easy to understand. The only input parameter that needed is the weights values associated with the criteria (Olson, 2004). As a consequence of these advantages, in this work, the researcher finds that this technique will be more suitable to rank the candidate items compared to other techniques.

Therefore, this work will use TOPSIS to evaluate and rank the candidate items. Since it assists in the evaluating candidate items with straightforward steps regardless the size of the alternatives and the number of attributes. In addition, there is no need to get the input parameters of the criteria weights from the expert because similarity values will be used as criteria weights; this avoids the disadvantage of TOPSIS. The principle behind TOPSIS is that the alternative which has the shortest distance from the ideal solution and farthest from the negative ideal solution is the best alternative (Ishizaka & Nemery, 2013). Table 2.6 presents different MADM methods with their advantages and disadvantages.

TOPSIS technique is based on several computation steps. The idea of TOPSIS can be expressed in a series of steps. TOPSIS steps are adopted and explained in more detail in Section 4.2.4.

i. The decision alternatives should be determined.
ii. Identifying the criteria (attributes) that are related to the decision problem.
iii. Constructing a decision matrix which contains *m* alternatives associated with *n* attributes (or criteria) and filled by scores of alternatives with respect to each criterion.
iv. Normalize the row scores to construct a precedence scores matrix or normalized decision matrix. The scores in the normalization matrix should be transformed into a normalized scale.
v. Construct the weighted normalized decision matrix. For each of the attribute, give weight to reflect how important it is to the overall decision.
vi. Determine ideal and negative-ideal solutions.
vii. Calculate the separation measure between each alternative which can be computed by the n-dimensional Euclidean distance.
viii. Calculate the relative closeness to the ideal solution.
ix. Rank order alternatives by maximising the relative closeness in the previous step 6.

Table 2.6 Summary of MADM Methods

| Method | Advantages | Disadvantages |
|---|---|---|
| Multi-Attribute Utility Theory (MAUT) | Takes uncertainty into account; can incorporate preferences. | Needs a lot of input; preferences need to be precise. |
| Analytic Hierarchy Process (AHP) | Easy to use; scalable; hierarchy structure can easily adjust to fit many sized problems; not data intensive. | Problems due to the interdependence between criteria and alternatives; can lead to inconsistencies between judgment and ranking criteria; rank reversal. |
| Case-Based Reasoning (CBR) | Not data intensive; requires little maintenance; can improve over time; can adapt to changes in the environment. | Sensitive to inconsistent data; requires many cases. |
| Data Envelopment Analysis (DEA) | Capable of handling multiple inputs and outputs; efficiency can be analysed and quantified. | Does not deal with imprecise data; assumes that all input and output are exactly known. |
| Fuzzy Set Theory | Allows for imprecise input; considers insufficient information. | Difficult to develop; can require numerous simulations before use. |
| Simple Multi-Attribute Rating Technique (SMART) | Simple; allows for any type of weight assignment technique; less effort by decision makers. | The procedure may not be convenient considering the framework. |
| Goal Programming (GP) | Capable of handling large-scale problems; can produce infinite alternatives. | It's ability to weight coefficients; typically needs to be used in combination with other MCDM methods to weight coefficients. |
| ELECTRE | Takes uncertainty and vagueness into account. | Its process and outcome can be difficult to explain in layman's terms; outranking causes the strengths and weaknesses of the alternatives to not be directly identified. |
| PROMETHEE | Easy to use; does not require the assumption that criteria are proportionate. | Does not provide a clear method by which to assign weights. |
| Simple Additive Weighting (SAW) | Ability to compensate among criteria; intuitive to decision makers; the calculation is simple does not require complex computer programs. | Estimates revealed do not always reflect the real situation; result obtained may not be logical. |
| TOPSIS | Has a simple process; easy to use and program; the number of steps remains the same regardless of the number of attributes. | Its use of Euclidean Distance does not consider the correlation of attributes; difficult to weight and keep the consistency of judgment. |

Source: Velasquez and Hester (2013).

## 2.7 Evaluation of the Recommender System

This section discusses the evaluation phase. To evaluate the accuracy of the new technique, need to determine the metrics that will be used. In addition, the datasets and the methods that will be used to divide these datasets into training and testing sets also are required to be located based on the domain of development purpose. Next subsections will present these points.

### 2.7.1 Evaluation metrics

In this section, we review the process of evaluating an RS. Many evaluation metrics have been used to assess the accuracy of the new memory-based CF method. These metrics compare the accuracy of the new method with the accuracy of existing methods. The most widely used evaluation metric for prediction accuracy of CF is the Mean Absolute Error (MAE). While, the precision and recall are widely used metrics to evaluate the performance accuracy (ranked lists of returned items in information retrieval) (Bobadilla, Ortega, & Hernando, 2012; Bobadilla et al., 2013; Choi & Suh, 2013; Liu et al., 2014; Patra et al., 2015; Polatidis & Georgiadis, 2016; Su & Khoshgoftaar, 2009).

#### 2.7.1.1 Predictive Accuracy Metrics

Predictive accuracy measurements evaluate the closeness of the actual user ratings, and the RS predicted scores. Predictive accuracy rating will be presented to the user as an annotation in the context of predictive accuracy measurements such as many stars or presented as a rating in MovieLens for a movie.

Predictive accuracy can also be utilised to assess the capability of an RS to order items regarding user interest as the predicted rating values generate a ranking among the items. The commonly used metric is the Mean Absolute Error (MAE) measurement. It is used to evaluate the predictive accuracy that calculates the difference between the actual users' ratings in the test set and predicted rating (Basu, Hirsh, & Cohen, 1998; Bobadilla, Ortega, & Hernando, 2012; Gu, Yang, & Dong, 2014a; Jonathan L Herlocker et al., 1999; Patra et al., 2015; Polatidis & Georgiadis, 2016; Sarwar et al., 2001; Shani & Gunawardana, 2011; Wu & Zheng, 2010), it can be calculated using Equation 2.27.

$$MAE = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N} \qquad 2.27$$

Where N represents the number of items that have been selected for the work test and rated by the target user, $p_i$ and $r_i$ is the predicted rating and actual rating for the item $i$, respectively. Actual rating is the rating for any item provided by the target user in the past. The main features of MAE measure are easy to understand, simple in implementation and computation.

Typically, MAE is the most widely accepted measure to determine the quality of the estimations. Other commonly-used metrics related to MAE metric are Mean Squared Error and Root Mean Squared Error (RMSE) (Bobadilla et al., 2011; Shardanand & Maes, 1995). It sets more emphasis on larger absolute errors and is given by Equation 2.28. The RS should have generated its list of predictions, and the real ratings should be provided as well. The lower MAE rate corresponds to a more accurate prediction.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}|p_i - r_i|}{N}} \quad 2.28$$

### 2.7.1.2 Performance Accuracy Metrics

These measures are commonly used to assess the information retrieval systems (Chowdhury, 2010; Salton & McGill, 1986). In RS, the items are suggested based on the user interests. The user is the person who can decide if an item conforms to his taste requirements or not. Thus, interest is more inherently subjective in RS than in traditional document retrieval. Evaluating the new method needs to implement and conduct many experiments using test dataset. After getting the results of the new method; there are four possible outcomes as in the recommender matrix in Table 2.7.

Table 2.7 Classification of the possible result of a recommendation of an item to a user

| | **Recommended** | **Not Recommended** |
|---|---|---|
| **Interest** | True Positives (TP) | False Negatives (FN) |
| **Not Interest** | False Positives (FP) | True Negatives (TN) |

From Table 2.7, the numbers of test samples that belong to the user Interest and Recommended represented by true positive (TP), the numbers of the test sample belong to the user Interest and Not Recommended denoted by false negative (FN). Also, the numbers of test sample not belong to the user Interest and Not Recommended called true negatives (TN). Finally, the number of test samples not belongs to the user Interest and Recommended called false positive (FP). Based on these terms, two essential measurements are defined which are: Recall and Precision (Chowdhury, 2010; Salton & McGill, 1986). The precision P is the ratio between the recommended and interest items to the user of the total number of items recommended. It can be defined as shown in Equation 2.29.

$$Precision\ P = \frac{Total\ Number\ of\ Recommended\ and\ Interesting\ Items}{Total\ Number\ of\ Recommended}$$

$$= \frac{TP}{TP + FP}$$

The recall $R$ is the ratio between the recommended and interesting items to the user to the total number of items that are rated by the user in the test data as shown in Equation 2.30.

$$Recall\ R = \frac{Total\ Number\ of\ Recommended\ and\ Interesting\ Items}{Total\ Number\ of\ Interesting\ Items} \tag{2.30}$$

$$= \frac{TP}{TP + FN}$$

Some attempts have been taken to merge recall and precision into one measurement (Salton & McGill, 1986). One way is the F-measure or F1 measurement which combines precision and recall into one number and is given by the following Equation 2.31

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.31}$$

### 2.7.2 Datasets

Usually, RS is developed according to a particular context, and the evaluations of these systems will be on datasets that related to that context or on a local dataset. Nevertheless, there are many public datasets available and widely used in the processes of CF evaluation such as EachMovie, MovieLens, Netflix, and Jester (Bobadilla et al., 2013). These datasets formed on the structure which a new method accuracy can be compared to the accuracy of existing methods in a consistent environment. Also, can serve in the developing system as a primary testing domain when the relevant dataset not available directly. Next, the widely used datasets will be presented with their properties.

EachMovie collaborative filtering dataset is one of the early publicly available dataset ("EachMovie ", 2017). This data set is containing 2.8 M user ratings of movies which gathered by EachMovie RS. The DEC Systems Research Centre operated the EachMovie recommendation movie system. It consists of 2,811,983 ratings entered by 72,916 users for 1628 different movies.

GroupLens research operates the MovieLens movie RS which was originally based on EachMovie dataset (Najafabadi et al., 2017). It is used as a public dataset in CF systems evaluation (J. Herlocker, Konstan, & Riedl, 2002; Jonathan L Herlocker et al., 1999; Sarwar et al., 2001). There are more versions available from MovieLens dataset. First one is 100K dataset released 4/1998 which contained 100,000 ratings from 1000 users on 1700 movies. In additional, MovieLens 1M Dataset released 2/20031 with million ratings from 6000 users on 4000 movies. Finally, MovieLens 10M dataset which includes 20 million ratings and 465,000 tag applications applied to 27,000 movies by 138,000 users. These ratings provided by the user in scale ranged from 1 to 5 stars, 1-star granularity, in the older dataset and from 0.5 to 5 stars in the newer one with 0.5-star granularity.

Two Jester datasets have been collected by Jester joke RS (Polatidis & Georgiadis, 2016). The first dataset has 4.1 million continuous ratings from 73,496 users on 100 jokes collected between April 1999 and May 2003. The second dataset has over 1.7 million continuous ratings from 59,132 users on 150 jokes which are collected between November 2006 and May 2009. The rating scale is within the range [-10.00 to +10.00] for both datasets.

The Netflix dataset, constructed to support participants in the Netflix Prize, has been used as a large-scale dataset in CF systems evaluation (http://www.netflixprize.com). This dataset contains over 100 million ratings from 480 thousand users on 17 thousand movies. The ratings are on the range of scale from 1 to 5 (integer) stars.

### 2.7.3 Splitting Methods

To use those mentioned measurements to evaluate the accuracy of the new method there are several partition techniques used to divide the dataset into two independent datasets, training and testing datasets. For example, holdout (HO), and k-fold cross-validation (CV) are widely used methods (Han, Pei, & Kamber, 2011), which will be used in this work. Next subsections describe the concept of each method.

#### 2.7.3.1 Holdout Method

In this method, the dataset will be partitioned randomly into two independent datasets, training and testing datasets as shown in Figure 2.5. Training set will be used to train the model; the testing set will be used to measure the model. Typically, two-thirds of the data are taken to present the training set, and the remaining one-third is allocated as a test set (1/3 for testing, 2/3 for training). Moreover, the dataset can be divided into five parts, where the four-fifth (80%) reallocated to the training set, and the remaining one-fifth (20%) is allocated to the test set. While that, the test set is used to measure the model, the training set is used to derive the model as shown in Figure 2.6.

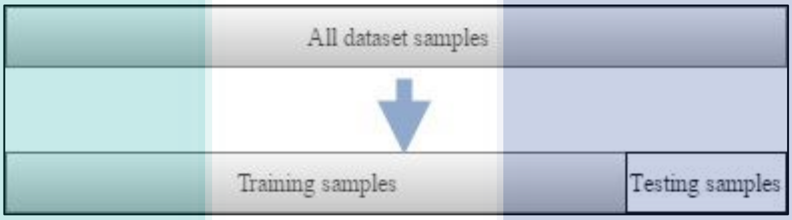



Figure 2.5 Holdout method

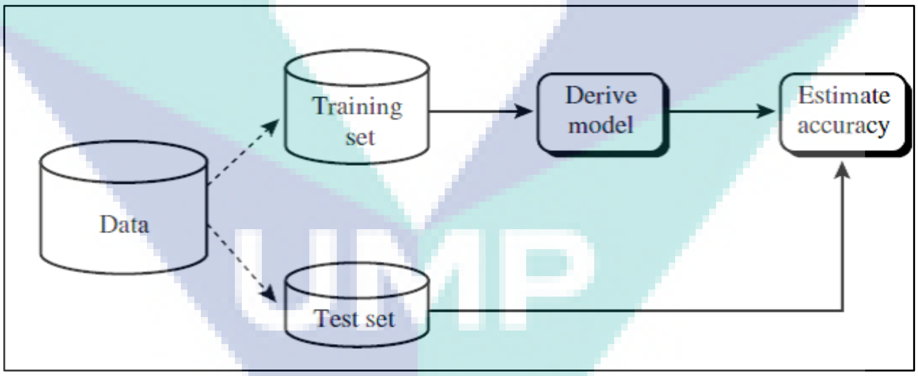



Figure 2.6 Holdout method procedure
Source: Han, Pei, and Kamber (2011)

#### 2.7.3.2 Random Subsampling/Repeated Holdout Method

In this method, the holdout method repeated k times. In each iteration, a certain fixed number of instances is randomly selected for training. The overall accuracies on the different iterations are averaged to yield an overall accuracy rate.

#### 2.7.3.3 Cross-validation Method

In cross-validation, the data set is partitioned into approximately equal $k$ subsets or folds ($D_1, D_2, \ldots, D_k$ ). Training and testing is executed $k$ times. In iteration $i$, for instance, the fold $D_i$ is used as a test set, and the other remaining folds are gathered to present the training set. To explain, in the first iteration, the subsets $D_2, \ldots, D_k$ gathered into a single set, a training set where the partition $D_1$ reserved as a test set. While, in the second iteration, the training sets are the partitions $D_1, D_3, \ldots, D_k$ and the partition $D_2$ is used as a testing set, and so on, as seen in Figure 2.7. Unlike, the holdout method, here, each partition is used $k$-1times in the training set and once for testing, it is the main advantage of the cross-validation method. That means, all records in the dataset are eventually used for both training and testing. Where, the results of $k$ trials are averaged to yield an overall result. The number of experiments depends on the number of folds, with a small number of folds, the number of experiments is reduced and vice versa.

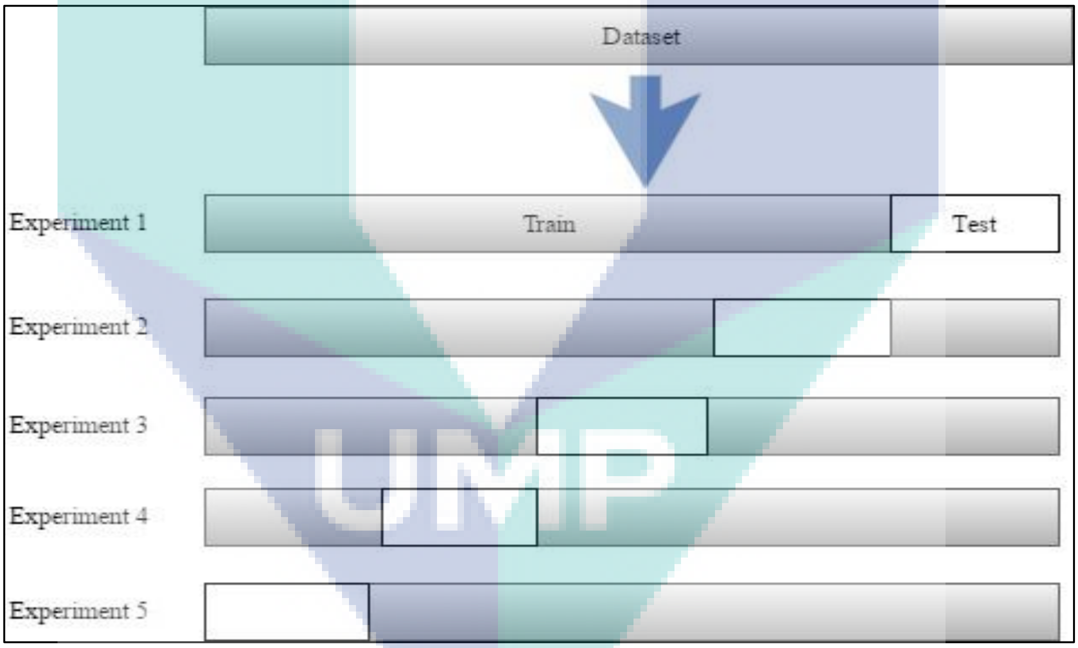



Figure 2.7 Cross-validation method procedure

## 2.8 Chapter Summary

This chapter can be summarised as follows. Firstly, the evolution and background of RS have been explained in Section 2.2. Additionally, the relevance feedback in CF RS has been presented is presented in this section. Secondly, three major approaches of RS have been demonstrated in Section 2.3, which are: content-based, collaborative filtering approach which a common approach and hybrid approach. Next, the memory-based CF, the memory-based CF related work, and most common challenges have been discussed in detail in Section 2.5. This section also came with limitations and finding. Moreover, in Section 2.6, the MADM method has been explained. Finally, this chapter ended by illustration show how to evaluate the RS in Section 2.7. Moreover, the common metrics used in CF RS evaluation, widely public used datasets and data's splitting methods have been presented in this section.

# CHAPTER 3

# METHODOLOGY

## 3.1 Introduction

The successful research passes through several fundamental and essential phases which are addressed by the researcher in an acceptable sequence. The time needed and effort for each stage are varies. In this chapter, the researcher describes the structure of the research design and methodology adopted to achieve the stipulated goals for this study. To achieve objectives of this work, Figure 3.1 presents the research framework phases that are used in achieving the defined objectives of this work. The primary aim of this work is to improve the accuracy of memory-based CF recommender system. The results expected to alleviate the aforementioned problem suffered by the memory-based CF. That will lead to improving the accuracy of memory-based CF regarding prediction accuracy and accuracy performance.

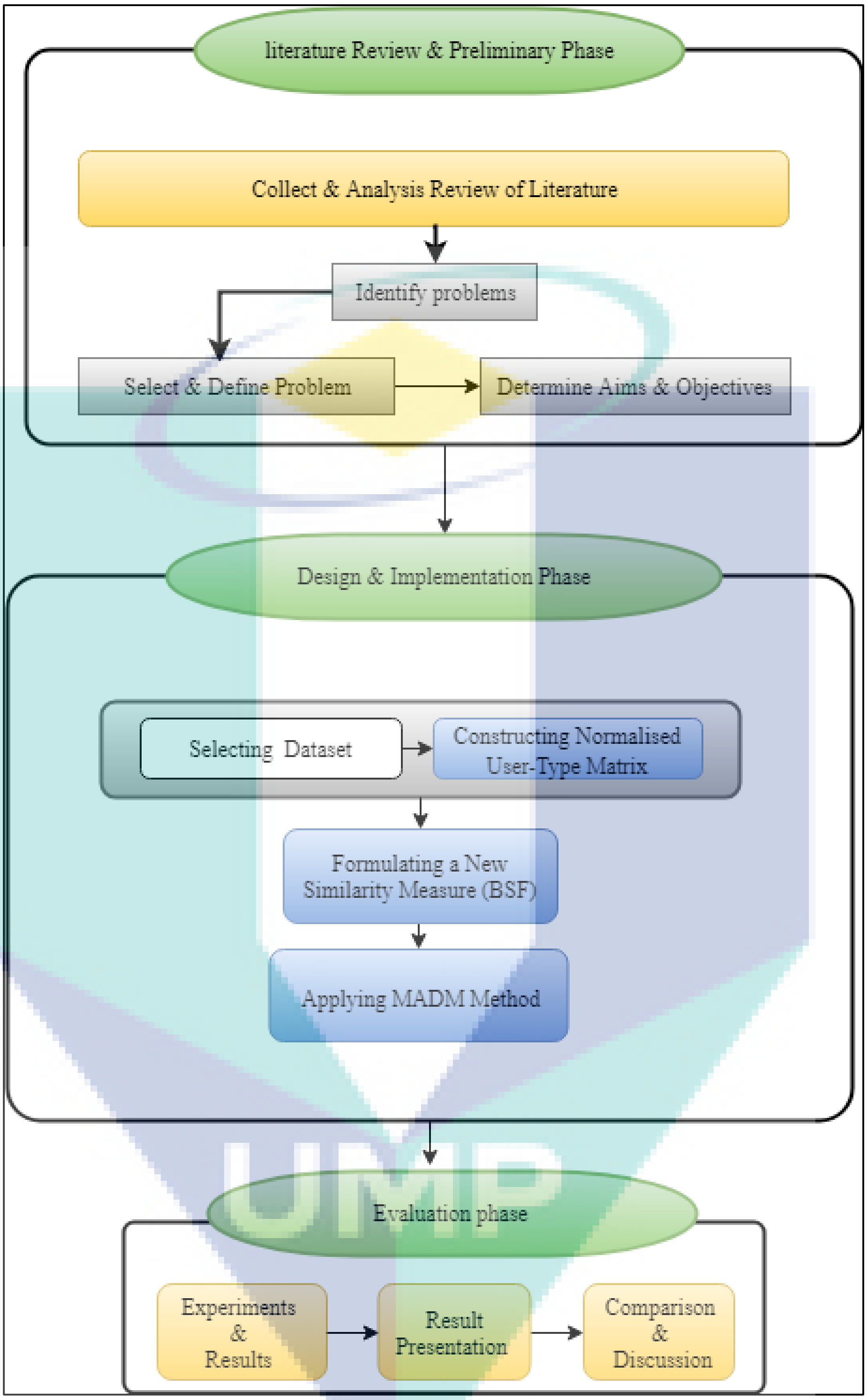



Figure 3.1 Research framework

The research framework consists of three main phases: preliminary, designing and implementation, and evaluation phase. Each phase includes subcomponents which have a brief explanation in the following subsections.

### 3.2 Literature Review and Preliminary Phase

The preliminary phase is the first phase of the proposed research framework. It is a way that the researcher set a plan before implementing his work to determine all aspects of the research. The central sub-component in this phase is collecting and analysis the literature review which is considered as one of the most critical phases of the research framework. Because, the researcher determines the limitations of existing studies and define the problem statement of this research. In addition, the objectives, significance, and scope of the study will also be determined in this component. Moreover, this phase comes out with the research Gantt chart which illustrates the predicted period to achieve this work. This phase has been clarified widely in chapter one.

The purpose of the research plan can be summarised in the following points: -

i. Provide a brief description of the research problem.
ii. Limitations of current studies that relate to the problem of research will be found.
iii. Identify the objectives of research based on the finding of present studies.
iv. Determine the procedures and steps that will be followed in addressing the problem of research.
v. Propose a way to address the problem with identifying research methods that will be followed by the researcher.
vi. Identify the dataset and its formatting needed by the researcher that will be used to measure the proposed technique

### 3.3 Design and Implementation Phase

Design steps describe the strategy followed by the researcher to address and achieve the research gap and objectives, respectively. As we see in Figure 3.1, the proposed technique consists of a sequence of steps. First of all, the dataset will be selected to be used in this work which determines the domain. Afterwards, the researcher normalizes this dataset to produce a new structure and formatting dataset matrix. This matrix will be filled by new values implicated from the user-item rating matrix to represent the global preferences of users which will address the issue of sparsity data. Next, the obtained matrix will be used as the main input for the new similarity measure to compute the correlation among users using. This new similarity measure will take into consideration two main factors to devalue of similarity when a pair of users do not have enough co-rated items. This might lead to locating the successful neighbours and therefore, the accuracy of the system will be enhanced. The first factor is the proportion of the number of items that rated by target user to the number of items that taken by both users (Fairness factor). The correlation between users will be increased as the number of ratings for each of them is close and vice versa. The second factor is the proportion of shared items also taken into account in the new similarity measure to decline the similarity weight when the number of co-rated not more enough. Finally, based on the previous result the prediction method predicts the score of items that not selected yet by the target user. Moreover, the MADM method will be adopted instead of the prediction method in traditional memory-based CF to evaluate and rank the candidate items to better recommendations as discussed in Chapter 2 and 4 Section 2.6 and 4.2.4, respectively. As a final point, the items that have the high ranking will be provided as a set of advice to the target user.

The implementation phase involves the programming and experiments. The researcher will use the C# programming language in order to execute this work. After the implementation accomplished, the selected dataset will be divided into training and testing dataset using the common splitting method to evaluate the proposed technique. More details about the design and implementation phases will be explained in Chapter Four.

### 3.3.1 Selecting Dataset

According to the context of the proposed method, the experiments of this work will be conducted on MovieLens datasets. Due to MovieLens datasets are public datasets available and widely used in the processes of memory-based CF system in order to test the new technique. Moreover, these datasets are relatively sparse and are suitable for evaluating the proposed technique.

In general, there are three versions available from MovieLens datasets.

i. First one is 100K dataset released 4/1998 which contained 100,000 ratings from 1000 users on 1700 movies.
ii. Second, MovieLens 1M dataset released 2/2003 with million ratings from 6000 users on 4000 movies.

iii. Third, 10M dataset released 1/2009 with 10 million ratings from 72,000 users on 10,000 movies.

These ratings provided by the user in scale ranged from 1 to 5 stars, 1-star granularity, in the first dataset and from 0.5 to 5 stars in the newer one with 0.5-star granularity. These datasets include three files which are: user file, item file, and rating file.

### 3.3.2 Constructing Normalized User-Type Matrix

The original user-item matrix is usually sparse, is the biggest issue facing memory-based CF recommender systems and effects on the quality of the system, since most of the users do not rate enough number of these items. Therefore, inability to locate the successful neighbours. Consequently, it may lead to weak recommendations. In order to address this issue and improve the accuracy of the memory-based CF system, the researcher creates a new matrix. This step will be illustrated in details in Section 4.2.2.

Typically, in memory-based CF, to provide recommendations to an active user the system uses his/her previous ratings to find the correlation between him/her and all users in the database. But, if the active user does not have enough ratings, this reduces the chances of having shared elements with other users, and therefore, the finding of his/her neighbours become inefficient or lead to locating unsuccessful neighbours. Thus, the researcher is going to construct the new data matrix that can alleviate from this issue; it will be derived from rating data matrix.

### 3.3.3 Formulating a New Similarity Measure (BSF)

In this section, the researcher will formulate a new similarity measure to locate the correct neighbours. This new similarity measure will use the data in the normalized user-type matrix as the primary input in the process of calculating the correlation among users. Furthermore, the new similarity measure will take into consideration the fairness and proportion of common rating factors to find the true relationships.

### 3.3.4 Applying MADM Method

In this stage, the system uses the neighbours of the active user to estimate the degree of his/her preference on an item, which not rated yet, using common aggregation methods such as average method, weighted sum method or adjusted weighted method

(Deviation-From-Mean). Instead of this way the researcher is going to employ a new way to evaluate and rank the items which will be more efficient. The MADM method will be used to assess and rank several alternatives based on various criteria. In this study, the neighbours of the active user will represent the criteria, and candidate items will represent the alternatives. The similarity weights of neighbours will be used to represent the weight according to each criterion.

## 3.4 Evaluation Phase

The researcher will use common matrices to evaluate the proposed method according to its experimental results. These results will be compared with the results of the existing memory-based CF methods. This phase contains the following steps:

Step 1: Experiments and results.

Step 2: Result presentation.

Step 3: Comparison and discussion.

In the step 2, the researcher will perform several experiments on MovieLens datasets to get the result. Holdout and Cross-validation techniques will be used to splitting the dataset into training and testing sets. Two main parameters are used as primary inputs which are: the size of neighbours and recommendation. These parameters should be specified with a fixed size. The obtained result from this step will be collected to be used in the next step (second step). In the step 2, presentation of the result obtained will be taken place by taking the average size of the recommended items for each associated neighbours' size. In the step 3, the process of evaluation will be achieved through comparing the final result with common memory-based techniques. The composition will be accomplished using specified metrics. These selected metrics considered as common measures used in the evaluation process of CF. Moreover, the evaluation will be in term of accuracy that can be divided into prediction accuracy and performance accuracy. The predictive accuracy can be defined as the ability of the proposed technique to predict a user's rating for an item not rated yet. The most one measure used to evaluate the accuracy of predicted ratings is Mean Absolute Error (MAE). MAE computes the error between the actual ratings and the predict scores. Regarding performance accuracy term, many of recommender system attempts to suggest to the users a set of items that they may want.

In this case, the researcher is not interested in whether the system predicts the ratings of items correctly but, whether the system provides the elements that the user will use it. There are two measures related to performance accuracy which are precision and recall. Furthermore, F-measure metric combines precision and recall into one number also will be used to measure the accuracy performance.

# CHAPTER 4

# CF-NSMA: PROPOSED MEMORY-BASED COLLABORATIVE FILTERING TECHNIQUE

## 4.1 Introduction

This chapter addresses the second objective. As discussed previously in Chapter Two, the users of Internet and items have been significantly increased. Meanwhile, most users do not have enough ratings. This always leads to sparse collected data which is represented in the user-item matrix. The traditional memory-based CF systems use users' rating as a source to represent their preferences. This rating some time provided randomly by the users within a determined scale and may not represent a user's information needs accurately. As a result, finding the relationship between users based on that matrix may lead to locating unsuccessful neighbours which leads to generating weak recommendations.

The proposed technique is a memory-based CF technique that uses user-item rating matrix to model the global users' preferences. To present the global users' preferences, the researcher built a new matrix, called the normalized user-type matrix. Regarding the problem of sparsity, it will be solved and in turn, improve the quality of the system with the new matrix. Moreover, a new similarity measure will be formulated to locate the neighbours based on the normalized user-type matrix inferred from the user-item matrix. The proposed technique utilises all local information (user ratings) to deduce the global information preferences. Next, the correlation between users will be computed based on these inferred preferences. As a similarity measurement, the Bray-Curtis (BC) distance measure was chosen and adjusted based on two main factors to calculate the similarity between a pair of users. These factors are the proportion of common ratings between a pair of users and the fairness factor. The ratio of common items, using a sigmoid function, is included in the new similarity measure to devalue the similarity value when the size of co-rated items of the pair users is small. When the common elements of comparative users are not as much as enough, this may lead to finding an adulterated relationship. While fairness factor can be defined as the proportion of the number of items that rated by the target user to the total number of items that taken by both users, this makes finding the relationship between users fairer. The Abbreviation of the new similarity measure is BSF (Bray & Curtis- Sigmoid function- Fairness factor). Finally, ranking the items that rated by neighbours using prediction method to evaluate our similarity method compare to existing common traditional memory-based CF methods. On the other hand, instead of the prediction method, the researcher will adopt the Multi-Attributes Decision Making (MADM) method to evaluate and rank those items.

In general, the proposed technique will be described in details in this chapter. It will be passed through four main phases aligning to the memory-based CF technique. These major phases are as follows. In the first phase, constructing the normalized user-type matrix based on the rating matrix to overcome the issue of data sparsity. In the second phase, the BSF similarity measurement will be formulated to locate the right nearest users that will lead to collect the right candidate items and in turn improve the recommendation. In the third phase, in the traditional memory-based CF the average weight regression method almost used to predict the score rating would give by the target user on a candidate item that's not taken yet. While, in this work, the MADM method will be applied instead of the prediction method. The MADM method will be used to evaluate and rank the candidate items which are not taken yet by the target user. Finally, in the last phase, the items which have the highest value ranking, M-top items (where M represents the number of recommended items), will be taken as a set of recommendations for the target user.

The rest of this chapter is organized as follows. First, the architecture of the proposed technique will be presented in Section 4.2. Each component of the system, including the input of the system, constructing the normalized user-type matrix, BSF similarity measure, neighbours formation, and MADM method will be discussed in detail in the Sections 4.2.1, 4.2.2, 4.2.3, and 4.2.4, respectively. Section 4.2.5 provides information on how the recommendations are generated. Finally, this chapter will be ended with a summary section.

## 4.2 Proposed Technique Architecture

Figure 4.1 illustrates the flow of the proposed CF technique. The system consists of five main components which are: Input, constructing the normalized user-type matrix, neighbours formation, TOPSIS technique as a useful technique in dealing with a multi-attribute problem, and the output component. Some of these components contain subcomponents as shown in Figure 4.1. Firstly, as discussed in Section 2.2.5, the ratings of users on items used as input of the system in traditional memory-based CF. Although the users' numeric ratings can be directly used in the CF system, the global preferences cannot. Therefore, the global preferences inferred by converting the numeric rating into normalized user-type, is structured. Thus, the issue of sparsity data suffered by traditional memory-based CF can be alleviated with this new structured data. Secondly, converting the rating matrix to normalized user-type matrix passes through three steps. The first step is grouping items depending on their types; each item can belong at least to one or more types in the same time, computing the frequency of times' ratings per each user regarding each type. The outcome of the grouping and computing frequent times process is a new structured data called time matrix (user-type time matrix). In the next step, normalizing the time matrix to produce normalized user-type matrix to better representing users' preferences. Thirdly, BSF similarity measurement is formulated to locate the successful neighbours, most similar users, which can enhance the accuracy of the system as will be shown in Chapter 5. Then, the items liked by neighbours are identified as a set of candidate items. Fourthly, the system employs the MADM method to evaluate and rank these candidate items. The neighbours' ratings and their similarity weights will be considered as input in MADM method to better ranking. Applying MADM will lead to a significant improving the accuracy of the system as will be presented in Section 5.3. Therefore, the neighbours' ratings on candidate items and their similarities weight with the target user will be utilised as the central part of the input to the MADM method. Finally, the output of the system will be a set of items that have the highest ranking and may be most preferred to the active user. Each component of the proposed technique will be described in detail in the remainder of this chapter.

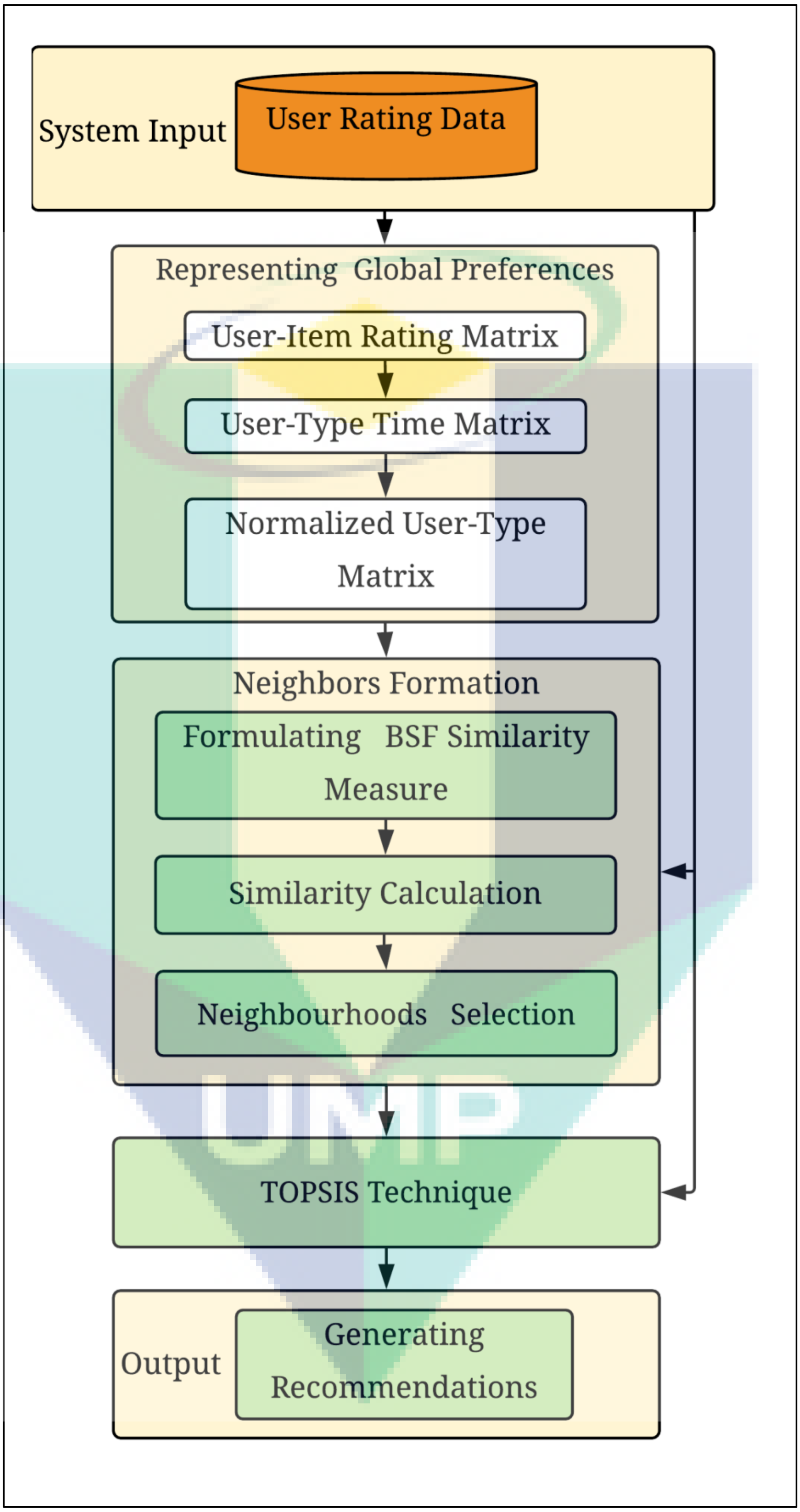



Figure 4.1 Proposed technique structure

### 4.2.1 System Input

The traditional memory-based CF systems use rating matrix directly as input. In this proposed memory-based CF technique, the global preference of the user, which is inferred based on the rating matrix, is the essential part that is used as a system input. Even though all ratings provided by the pair of users are utilised to represent the global preferences, the proportion of common rating items between the comparative pair of users also is included as part of the system input. Also, the percentage of the number of items that are rated by the target user to the total number of items that are rated by both users also is considered.

The normalized user-type matrix filled within the interval [0–1], these values represent the percentage of users' preferences on each type in the database. For example, if user x has three ratings on movies a, b, and c. The genres of these movies are: (Action, Comedy), (Action, Drama, and Romance), and (Comedy, and Drama), respectively. Then, the proportion of user's preferences on Action genre is $2/7 \approx 0.29$, where 7 represents the total times of ratings of user x on all genres, and 2 represents the number of rating times of the user on items that belong to genre Action, and so on. Therefore, the rating matrix is converted into the normalized user-type matrix which filled with values that represent the proportion of users' preferences/interest according to each type. It is observed that, finding the relationship between two users became easy even if there are no co-rated items between them because each user should have at least one item rated and therefore the shared types of users will be more than one types that make finding relationship more easily. Thus, a corresponding to normalized user-type matrix the issue of sparsity suffered by the memory-based CF using user-item matrix has been alleviated.

### 4.2.2 Re-representing User Preference

In this section, the transformation of rating matrix to the normalized user-type matrix will be explained. As mentioned in the previous section, the normalized user-type matrix can alleviate the impacts of the sparsity problem. While the traditional memory-based CF still suffer from this problem because it uses the rating matrix directly as input. Due to the number of users and items on the Internet are huge, and it is still growing. However, most of them not have enough ratings that make the rating matrix sparse. Moreover, some of the users rate the items randomly. In another word, some users may give inaccurate rating which leads to the issue of locating unsuccessful neighbours. Therefore, inferring the information about preferences of users to build their profile preferences correctly is required. In this work, the ratings of users used to infer the global favourites and will be represented in the normalized user-type matrix. This phase passed through three main processes which are presented as follows:

i. User-Item Rating Matrix Representation.
ii. User-Type Time Matrix Creation.
iii. Normalized User-Type Matrix Construction.

#### 4.2.2.1 User-Item Rating Matrix Representation

The user-item rating matrix is a collection of numerical ratings provided by the users on items, it can be defined in *Definition* 1. Each user should have at least one rate. Accordingly, every item might rate numerous times by numerous users. Next, some sets are defined to simplify the representation of the rating matrix.

Let the basic sets defined as follows:

*N*: Natural numbers.
*Max*: Maximum value scale of ratings.
*Min*: Minimum value scale of ratings.
$U = \{u_1, u_2, \ , u_i, \ldots, u_{n-1}, u_n\}$, set of users where $i = 1, 2, \ldots, n$. And $n$ is the total number of users in the database.
$I = \{i_1, i_2, \ , i_j, \ldots, i_{m-1}, i_m\}$, set of items where $j = 1, 2, \ldots, m$. And $m$ represents the total number of items in the database.
$R = \{r_{1,1}, r_{1,2}, \ldots, r_{i,j}, \ldots, r_{n-1,m-1}, r_{n,m}\}$ set of ratings where $min \leq r_{i,j} \leq max$ or is $*$ in case of absence of ratings.

*Definition* 1: let U defined as a set of *n* users and I is a set of *m* items that one rated within the interval [*Min*, …, *Max*]. Then, the conceptual rating matrix R is represented as shown in Table 4.1, where the rows represent the vector ratings of users and the items rating represented by the columns. Where $r_{i,j}$ represents the rating of user *i* on item *j* and the absence of ratings will be symbolled by the symbol *.

Table 4.1 Conceptual rating matrix R

| | $\boldsymbol{i_1}$ | $\boldsymbol{i_2}$ | $\boldsymbol{i_3}$ | $\cdots$ | $\boldsymbol{i_j}$ | $\cdots$ | $\boldsymbol{i_{m-1}}$ | $\boldsymbol{i_m}$ |
|---|---|---|---|---|---|---|---|---|
| $\boldsymbol{u_1}$ | * | $r_{1,2}$ | * | $\cdots$ | $r_{1,j}$ | $\cdots$ | $r_{1,m-1}$ | $r_{1,m}$ |
| $\boldsymbol{u_2}$ | $r_{2,1}$ | * | $r_{2,3}$ | $\cdots$ | * | $\cdots$ | $r_{2,m-1}$ | $r_{2,m}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $\boldsymbol{u_i}$ | $r_{i,1}$ | * | $r_{i,3}$ | $\cdots$ | $r_{i,j}$ | | $r_{i,m-1}$ | $r_{i,m}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $\boldsymbol{u_{n-1}}$ | $r_{n-1,1}$ | $r_{n-1,2}$ | * | $\cdots$ | $r_{n-1,j}$ | $\cdots$ | * | $r_{n-1,m}$ |
| $\boldsymbol{u_n}$ | $r_{n,1}$ | $r_{n,2}$ | * | $\cdots$ | $r_{n,2}$ | $\cdots$ | $r_{n,m-1}$ | * |

#### 4.2.2.2 User-Type Time Matrix Creation

To illustrate the motivation behind this step let's assume the following hypothesis. In most cases, e-commerce' users buy their commodities based on the type of colour, style, brand, etc. and this behaviour can reflect their preferences depending on the type of their purchases. Likewise, the movies' domain is classified into several types such as action, crime, comedy, documentary, etc. Thus, the kind of movies rated by the user may reflect his/her preferences. Obviously, the users who prefer to watch the documentary movies will prefer to watch this type of film more than others. With this assumption in mind, the researcher constructed the type matrix that defined in the coming *Definitions* 2, 3 and it will be filled with values that represent the rating times of each type per each user. Thus, this matrix is different from the former matrix; its dimension concerning space of items is smaller compared to the number of items in the database. Since the number of items in the database is huge while the groups of these items always will be less. For example, the movie dataset 1M MovieLens, which will be used in this work to evaluate the proposed technique, has eighteen genres while the number of whole movies is more than 3,500 movies.

To clarify, let Table 4.2 represents the rating matrix where the column type represents the type of movie. As can be seen, user $u_2$ has rated the movies $a$, b, and c. Therefore, the rating count of user $u_2$: *Action* movie type three times, *Comedy* movie type two times, *Animation* movie type one time, *Crime* movie type one time, *Drama* movie type one time and it has been captured in user-type time Table 4.4.

Table 4.2 User-item rating matrix example

| | $u_1$ | $u_2$ | $u_3$ | $u_4$ | Type |
|---|---|---|---|---|---|
| a | 4 | 3 | * | 4 | Action, Comedy, Animation |
| b | * | 3 | 5 | * | Action, Adventure, Crime |
| c | 4 | 4 | 3 | * | Action, Drama, Comedy, Animation |

*Definition* 2: let $\overrightarrow{I^j}$ used as a vector to represent the category information of item $j$, where $\overrightarrow{I^j} = (c_1^j, \quad c_2^j, \quad \dots, c_g^j, \quad \dots, \quad , c_{k-1}^j, c_k^j)$. And $\vec{c}$ vector defined as a vector to represent the category of items in the dataset, where $\vec{c} = \left(c_1, c_2, \dots, c_g, \dots, c_{k-1}, c_k\right)$. Where $k$ is the number of categories of items in that dataset and g=*1, …, k*. Then, the value of $c_g^j$ will be equal 1 if the item $j$ belongs into $g^{th}$ type and 0 otherwise as shown in Equation 4.1.

$$c_g^j = \begin{cases} 1, & j \in c_g \\ 0, otherwise \end{cases} \qquad 4.1$$

For example, from the Table 4.2 the category vector $\vec{c} =$ (Action, Comedy, Animation, Adventure, Crime, Drama), so the category information of movie $\vec{a} = (1, 1, 1, 0, 0, 0)$.

*Definition* 3: let U a collection of users (e.g., *n* users), and C is a set of items' categories in the database (e.g., *k* categories). Therefore, the preferences of users on each group presented in the vector space model by a user-type time matrix T, as shown in Table 4.3. Where the $t_{i,g}$ notes to the number of items that rated by user *i* and belong to genre $g^{\text{th}}$. Equation 4.2 used to compute the value of $t_{i,g}$.

$$t_{i,g} = \sum_{j \in I_i} c_g^j \qquad 4.2$$

Where $I_i$ represents the set of items that rated by user *i* and $c_g^j$ denotes to the value of item *j* that belong to g$^{th}$ genre.

Table 4.3 A conceptual time matrix T

| | $c_1$ | $c_2$ | $\cdots$ | $c_g$ | $\cdots$ | $c_{k-1}$ | $c_k$ |
|---|---|---|---|---|---|---|---|
| $u_1$ | $t_{1,1}$ | $t_{1,2}$ | $\cdots$ | $t_{1,g}$ | $\cdots$ | $t_{1,k-1}$ | $t_{1,k}$ |
| $u_2$ | $t_{2,1}$ | $t_{2,2}$ | $\cdots$ | $t_{2,g}$ | $\cdots$ | $t_{2,k-1}$ | $t_{2,k}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $u_i$ | $t_{i,1}$ | $t_{i,2}$ | $\cdots$ | $t_{i,g}$ | $\cdots$ | $t_{i,k-1}$ | $t_{i,k}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $u_{n-1}$ | $t_{n-1,1}$ | $t_{n-1,2}$ | $\cdots$ | $t_{n-1,g}$ | $\cdots$ | $t_{n-1,k-1}$ | $t_{n-1,k}$ |
| $u_n$ | $t_{n,1}$ | $t_{n,2}$ | $\cdots$ | $t_{n,g}$ | $t_{n,1}$ | $t_{n,k-1}$ | $t_{n,k}$ |

To demonstrate this step, based on the data in Table 4.2 and using the Equation 4.2 the result is represented in Table 4.4. That depicts an example of user-type time matrix.

Table 4.4 User-type time rating matrix example

| | $u_1$ | $u_2$ | $u_3$ | $u_4$ |
|---|---|---|---|---|
| Action Movie | 2 | 3 | 2 | 1 |
| Comedy Movie | 2 | 2 | 1 | 1 |
| Animation Movie | 2 | 2 | 1 | 1 |
| Adventure Movie | 0 | 1 | 1 | 0 |
| Crime Movie | 0 | 1 | 1 | 0 |
| Drama Movie | 1 | 1 | 1 | 0 |
| Total count | 7 | 10 | 7 | 3 |

#### 4.2.2.3 Normalized User-Type Matrix Construction

The researcher is going to normalize the matrix T to produce the normalized user-type matrix W. Normalized values between zero and one will fill it. The normalization here is necessary to standardise the range of values. For instance, if user x and y rated action movie 20 times. And the number of rating of user x more than user y. Logically, the percentage preference of user x should be larger than ratio preference of the user y on the action type. Therefore, a linear-scale is used to transforming the values in matrix T into the ratio matrix W with value $w_{i,g}$ between zero and one normalized. The ratio values will represent the global preference of user which used as input in the calculation similarity process. The normalized user-type matrix is defined by *Definition* 4.

*Definition* 4: let $\overrightarrow{T_i}$ used as a vector to represent the category information of user *i* where $\overrightarrow{T_i} = \{t_{i,1}, t_{i,2}, \ldots, t_{i,g}, \ldots, t_{i,k-1}, t_{i,k}\}$, where $t_{i,g}$ represents the ratings counts of user *i* on type *g* where $g = 1, 2, \ldots, k$. Therefore, the preferences of users on each category presented in the vector space model by normalized user-type matrix W, as shown in Table 4.5. Where the normalized value $w_{i,g}$ is the percentage of user *i* preference on category *g* which can be calculated using the normalization Equation 4.3:

$$w_{i,g} = \frac{t_{i,g}}{\sum_{g=1}^{k} t_{i,g}} \qquad 4.3$$

Where the *k* is the number of items categories/types in the database.

Table 4.5 A conceptual normalized matrix W

| | $c_1$ | $c_2$ | $\cdots$ | $c_g$ | $\cdots$ | $c_{k-1}$ | $c_k$ |
|---|---|---|---|---|---|---|---|
| $u_1$ | $w_{1,1}$ | $w_{1,2}$ | $\cdots$ | $w_{1,g}$ | $\cdots$ | $w_{1,k-1}$ | $w_{1,k}$ |
| $u_2$ | $w_{2,1}$ | $w_{2,2}$ | $\cdots$ | $w_{2,g}$ | $\cdots$ | $w_{2,k-1}$ | $w_{2,k}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $u_i$ | $w_{i,1}$ | $w_{i,2}$ | $\cdots$ | $w_{i,g}$ | $\cdots$ | $w_{i,k-1}$ | $w_{i,k}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $u_{n-1}$ | $w_{n-1,1}$ | $w_{n-1,2}$ | $\cdots$ | $w_{n-1,g}$ | $\cdots$ | $w_{n-1,k-1}$ | $w_{n-1,k}$ |
| $u_n$ | $w_{n,1}$ | $w_{n,2}$ | $\cdots$ | $w_{n,g}$ | $w_{n,1}$ | $w_{n,k-1}$ | $w_{n,k}$ |

To clarify, based on Table 4.4 and using Equation 4.3, the normalized value of user *X* on *Action* movie type is $w_{x,Action} = \frac{2}{7} \approx .29$ and it has been captured in the normalized user-type matrix Table 4.6.

Table 4.6 Normalized user-type matrix example

| | $u_1$ | $u_2$ | $u_3$ | $u_4$ |
|---|---|---|---|---|
| Action Movie | 0.29 | 0.3 | 0.29 | 0.33 |
| Comedy Movie | 0.29 | 0.2 | 0.14 | 0.33 |
| Animation Movie | 0.29 | 0.2 | 0.14 | 0.33 |
| Adventure Movie | 0 | 0.1 | 0.14 | 0 |
| Crime Movie | 0 | 0.1 | 0.14 | 0 |
| Drama Movie | 0.14 | 0.1 | 0.14 | 0 |

### 4.2.3 Neighbours Formation

Finding the most similar users to the target user requires calculating the similarity of this user with other users in the database. The formation of the neighbouring process consists of three steps: formulating the similarity measure, calculating the relationship between users and selecting top-k users to represent the neighbours. The next subsections illustrate these steps.

#### 4.2.3.1 Formulating BSF similarity measure

Section 2.5 discussed the shortcomings of the common existing similarity measures. To enhance the accuracy of memory-based CF system and overcome that weakness, a new similarity measure is proposed. The researcher followed design-implement-test cycle to develop the new similarity measure, as shown in Figure 4.2. For more details, Appendix A represents the experiments' results that conducted to formulate the developed similarity measure.

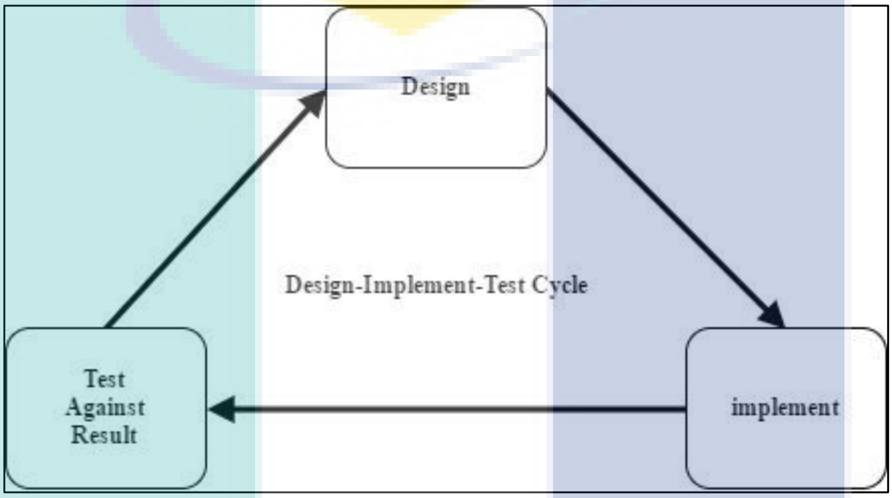



Figure 4.2 The Design-Implement-Test Cycle

*BSF similarity measure motivation*

Finding the relationship between a pair of users in the most of the traditional memory-based CF recommender system relies on the PCC or Cosine and their derivatives measures. Therefore, most of the enhancements were based on these traditional similarity measures. Nevertheless, these improved similarity measures have several shortcomings/limitations.

i. First, similarity calculation in most of those measures depends on the ratings in user-item rating matrix. However, the similarity computation suffers from few co-rated items issue. Additionally, if the pair of users does not have common items, the similarity value will be zero. The proposed method can be worked with few or no single co-rated items between the pair of users. This method presented on formulating similarity measure that utilised all ratings of users to generate the ratio matrices.

ii. Second, it is true if we say that, the rating value reflect the interest of the user. But not all of them can rate carefully. This mean, may some users give an untrue rate that leads to locating unsuccessful neighbours. For this reason, the normalized user-type matrix is created that reflects the global preferences of users regardless how to they rate items (high, low).

iii. Third, ignoring the proportion of common ratings and absolute value not considered such as MSD and Jaccard, respectively will lead to low accuracy. Unlike the proposed method that utilises all ratings provided by the user to implicate his/her preference. Moreover, it takes into consideration the proportion of common ratings.

iv. Fourth, Fairness factor is not considered in the current memory-based CF methods. Fairness is defined in this study, as the ratio of the number of items rated by the target user to the number of items that taken by both of them.

v. Fifth, calculation the similarity in some improved measures are too much which make the computation more complicated, such as PIP, MJD and PSS measures. How to reduce these computations using the new similarity measure with preserving the efficiency?

vi. Sixth, the accuracy is still an issue. Most of the studies improved the accuracy of CF just by improving similarity measure while only a few studies focused on the prediction score method which is also on the same level of importance. Therefore, employing TOPSIS as a useful technique in dealing with multi-attribute (MADM) problems it will lead to improve the accuracy.

vii. Finally, the memory-based CF mechanisms still have an open room for enhancement that leads to improvement in the accuracy of the system.

Therefore, the calculations similarity in our proposed similarity measure depends on the normalized user-type matrix. Moreover, the rating matrix information also utilised such as the proportion of common ratings and the number of rating of the target user to the total ratings by both users. Additionally, the candidate items' ratings will be used as input to MADM method. After all, the similarity measurement has been formulated with two adopted factors. In the next section, a description of proposed similarity measure is presented that aims to improve the accuracy of memory-based CF system.

*User similarity measurement based on normalized matrix*

In the beginning, adopting fairness factor to similarity measure to make the similarity more accurate. It is the proportion of the number of items that rated by target user to the number of items that taken by both users, as defined in *Definition* 5. The existing similarity does not take into consideration the percentage difference in the number of ratings for each user when finding the relationship between users. For instance, the similarity between two users who have the same number of ratings must be higher than others. Therefore, the new measurement considers the fairness factor in order to find these consistencies in the number of users' ratings. Fairness factor it can increase the value of similarity between users who have the same number of ratings.

*Definition* 5*:* if *u* represents the target user and the $|\mathrm{I_u}|$ represents the number of items rated by the user *u*. On the other hand, if we assume that, the *v* represents the compared user and $|I_v|$ is the number of rated items by user *v*. Therefore, the fairness factor Ff can defined as given in Equation 4.4:

$$Ff(u,v) = \frac{|I_u|}{|I_{u+}I_v|} \text{ , and } Ff(v,u) = \frac{|I_v|}{|I_{u+}I_v|} \qquad 4.4$$

Secondly, the number of common items is also essential. To explain, if the similarity between user $sim(u,v) = sim(u,l)$, and the number of co-rated between user *u* and *v* is bigger than *u* and *l*, then the $sim(u,v)$ should be greater than $sim(u,l)$. The new similarity will be multiplied by sigmoid function to devalue the similarity in case of a few co-ratings. Let's denominator $\theta$ use to determine the minimum size of co-rated items, and if the size of the set of common items which rated by both users is large enough (equal or bigger than minimum threshold size of co-rated items), then the sigmoid value would be bigger than 0.9, but for small of co-rated items, the value would be less 0.9. For example, if the denominator $\theta$ equal 1 and the number of common ratings of the pair of users were equal 0, then the sigmoid value would be 0.5. But, if the size of co-rated items more than 3, the sigmoid value would be greater than 0.95. That means, if more common rated items exist between users, then they are more similar. The sigmoid function ($Sf$) can be computed as shown in Equation 4.5:

$$Sf(u,v) = Sf(v,u) = \frac{1}{1 + Exp^{(-\frac{|I_{u,v}|}{\theta})}} \qquad 4.5$$

Where $|I_{u,v}|$ represents the number of items which rated by both users *u* and *v*.

In this work, the proposed similarity measure that used is Bray-Curtis (BC) distance measurement, which is introduced to compute the distance between two different sites, based on counts at each site (Bray & Curtis, 1957; M. J. Pazzani & Billsus, 2007h). The normalization is done using absolute difference divided by the summation. The output value of BC is between zero and one. A value of zero indicates a complete matching of the two data records in the n-dimensional space. In contrast, one means that the records are different (Teknomo, 2017; Viriyavisuthisakul et al., 2015). The advantage of Bray-Curtis is that the scale is easy to understand: 1 means the samples are the same, while 0 is the maximum difference that can be observed between two samples. Thus it would seem to be a proper expression to use. The BC distance performed better when compared to ten and nine distance measures in the works (Kokare, Chatterji, & Biswas, 2003; Viriyavisuthisakul et al., 2015), respectively. The general formula for computing the Bray-Curtis similarity between user *u* and *v* is defined as in the Equation 4.6:

$$Bc\ (u,v) = 1/(1 + \frac{\sum_{g=1}^{k} |w_{u,g}\ - w_{v,g}\ |}{\sum_{g=1}^{k} w_{i,g}\ + \sum_{g=1}^{k} w_{v,g}}) \qquad 4.6$$

Where *k* represents the number of items' categories in the database and the $w_{u,g}$ and $w_{v,g}$ represent the ratio rating of type *g* for user *u* and *v*, respectively.

To adopt the factors that mentioned before the Equation 4.6 was adjusted as in Equation 4.7.

$$BSF\ (u,v) = 1/(1 + \frac{\sum_{g=1}^{k} |w_{v,g}\ * cf(u,v) - w_{v,g}\ * cf(v,u)|}{\sum_{g=1}^{k} w_{v,g}\ * cf(u,v) + \sum_{g=1}^{k} w_{v,g}\ * cf(v,u)}) * sf(u,v)$$

Based on the normalized user-type matrix and using the Equations 4.4, 4.5 and 4.7 the similarity between users computed if even the pair of users does not have common item ratings, which cannot be calculated in the most traditional memory-based CF methods in such a situation. It considered a problem because the output is zero in that case. Appendix A shows the experiments that are conducted to finalize the BSF similarity measure Equation 4.7.

### 4.2.3.2 Similarity Calculation

After the similarity measurement formulated, the relationships between users in the database should be computed to determine the most similar users. The system calculates the similarity between the active user and all users in the database. The users who have highest similarity weight with the target user they make up the neighbours of his/her. Based on the BSF similarity measure of Equation 4.7 the result of similarity between users can be represented in matrix *S*, as shown in Table 4.7.

*Definition* 6: let $\overrightarrow{u_i}$ and $\overrightarrow{u_j}$ used as vectors to represent the similarity information of user *i* and user *j*, respectively. Where $\overrightarrow{u_i} = \{s_{i,1}, s_{i,2}, \dots, s_{i,j}, \dots, s_{i,n-1}, s_{i,n}\}$ and $\overrightarrow{u_j} = \{s_{j,1}, s_{j,2}, \dots, s_{j,i}, \dots, s_{j,n-1}, s_{j,n}\}$, where $i \; and \; j \; = 1, 2, \dots, n$. Therefore, the similarity values between users can be presented in the vector space model by a user-user similarity matrix S, as shown in Table 4.7. Where $s_{i,j}$ and $s_{j,i}$ represent the similarity values of user *i* with user *j* and user *j* with user *i*, respectively, and $s_{i,j} = s_{j,i}$.

Table 4.7 A conceptual similarity matrix S

| | $\boldsymbol{u_1}$ | $\boldsymbol{u_2}$ | $\cdots$ | $\boldsymbol{u_i}$ | $\cdots$ | $\boldsymbol{u_{n-1}}$ | $\boldsymbol{u_n}$ |
|---|---|---|---|---|---|---|---|
| $\boldsymbol{u_1}$ | $s_{1,1}$ | $s_{1,2}$ | $\cdots$ | $s_{1,i}$ | $\cdots$ | $s_{1,n-1}$ | $s_{1,n}$ |
| $\boldsymbol{u_2}$ | $s_{2,1}$ | $s_{2,2}$ | $\cdots$ | $s_{2,i}$ | $\cdots$ | $s_{2,n-1}$ | $s_{2,n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $\boldsymbol{u_i}$ | $s_{i,1}$ | $s_{i,2}$ | $\cdots$ | $s_{i,i}$ | $\cdots$ | $s_{i,n-1}$ | $s_{i,n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $\boldsymbol{u_{n-1}}$ | $s_{n-1,1}$ | $s_{n-1,2}$ | $\cdots$ | $s_{n-1,i}$ | $\cdots$ | $s_{n-1,n-1}$ | $s_{n-1,n}$ |
| $\boldsymbol{u_n}$ | $s_{n,1}$ | $ws_{n,2}$ | $\cdots$ | $s_{n,i}$ | $s_{n,1}$ | $s_{n,n-1}$ | $s_{n,n}$ |

*Similarity calculation running example*

Let's $\theta = 1$, meaning that, the number of co-rated items must be more than 1 rating to increase the similarity else the similarity should be decreased. Based on Table 4.6 and using the Equations 4.4,4.5, and 4.7, the similarity between user $u_3$ and $u_4$ can be computed as follows:

$$Ff(u3, u4) = \frac{2}{1+2} \approx 0.66$$

$$Ff(u4, u3) = \frac{1}{1+2} \approx 0.33$$

$$sf(u3, u4) = \frac{1}{1 + Exp^{\left(-\frac{0}{1}\right)}} = 0.5$$

$$BSF\ (u3,u4) = 1/(1 + \frac{(0.29*0.66-0.33*0.33)+\cdots+(0.14*0.66-0.0*0.33)}{(0.29*0.66+\cdots+0.0.14*0.66)+(0.33*0.33+\cdots+0.0*0.33)}) * 0.5$$

$$BSF\ (u3,u4) = 1/(1+\frac{0.3267}{0.98}) * 0.5 \approx 0.374$$

#### 4.2.3.3 Neighbourhoods Selection

After the similarity weights calculated, a subset of users should be selected to represent the neighbours and used as the main part of generating the recommendations. There are two common ways used for selection the neighbours which are: top-N neighbours (Jonathan L Herlocker et al., 1999) and similarity thresholding (Shardanand & Maes, 1995). The first way, the recommender system ranks the users according to their similarity weights to the target user. After that, the top N users who have highest similarity weights considered as neighbours for generating recommendations. In the second way, similarity thresholding, the recommender system uses a specific threshold value that determines the minimum similarity value weight. In this method, the recommender system takes all users who have bigger than minimum threshold similarity for generating recommendations. Take out the users whose similarity weight with the target user is less than the threshold value. In this work, the N-top neighbourhoods' way is used to select which users can represent the neighbours instead of the whole users because the top N users have strong relation to the target user that leads to more accurate recommendations (Jonathan L Herlocker et al., 1999).

The neighbour's formation algorithm in Figure 4.3 explains how the similarities between the users are calculated and how they are selected.

Name: Neighbours Formation
Input: Rating and normalised Matrices
Output: Similarity Matrix and K-neighbours
Body:

// Computing the similarities

$For\ i = 1\ to\ i = n$ //*i* is the target user and *n* the # of users in the database.

**$Begin$**

$For\ j = i\ to\ j = n$ //Where *j* represents the compared user

**$Begin$**

//Calculating $Ff$ of user *i* associated with the user *j* and vice versa.

$Ff\ (i,j)\ =\ I_i/|I_i + I_j|$

$Ff\ (j,i)\ =\ I_j/|I_i + I_j|$

// Computing the $Sf$ between user *i* and *j*

$Sf\ (i,j)\ =\ 1/1\ + Exp^{(-\frac{|I_{u,v}|}{\theta})}$

$Similarity\ matrix\ (i,j) =\ BSF\ (i,j)$

**$End$**

**$End$**

// Sorting Similarity matrix according each user based on similarity weights.

$Sort(Similarity\ matrix)$

//Selecting neighbours

$For\ u = 1\ to\ k$

**$Begin$**

$u_k = k^{th}$ //Selecting the top-k, $k$ is the size of neighbours.

**$End$**

//Represent the candidate items for the target user *u*

$For\ v = 1\ to\ k$

**$Begin$**

$Candidate\_items\_list =\ I_v$ //$I_v$ is a set of items that already rated by the user *v* and not rated yet by the user *u*.

**$End$**

Figure 4.3 Neighbours formation algorithm

### 4.2.4 Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

Typically, there are multiple criteria used to evaluate a set of alternatives in decision-making. For example, in purchasing a car, some of the main criteria such as cost, safety, comfort, and fuel consumption should be taken into consideration. Moreover, Multi-criteria decision making (MADM) is one of the well-known topics of decision making, and it is necessary to use decision maker's preferences to differentiate between alternatives. In the literature, MCDM problems are divided into two basic approaches (Kahraman, 2008): multi-objective decision making (MODM), and multi-attribute decision making (MADM) as discussed in Chapter 2 Section 2.6.

Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is popular one of the MADM methods which was proposed by Yoon and Hwang (1981). The main idea behind TOPSIS is that the best alternative should have the shortest Euclidean distance from the ideal solution and farthest from the negative ideal solution. Then, ranking all options and the best alternative will be in the top of the sorted list. The main advantage of this technique is that it needs only a limit number of inputs from the user and its output is easy to understand.

Hence in this work, the TOPSIS technique is applied to rank the candidate items instead of prediction rating method. It will be utilised to evaluate and sort the candidate items, the items which rated by the neighbours and not selected by the target user yet. Then, choose the top M items of sorting as a recommendation for an active user, which may be the preferable items to the target user.

Furthermore, the *k*-neighbours, their similarity weights, candidate items and the actual rating values on these items provided by the neighbours, all of these parts will be used as essential input of TOPSIS technique. Typically, TOPSIS method requires converting the problem into a decision matrix X with *m* alternatives (rows) and *n* criteria (columns). In the process of decision matrix constructing, the M candidate items and K-neighbours will be used to represent the *m* alternatives and *n* criteria, respectively. Where the $x_{i,j}$ represents the numerical outcome of the alternative $j^{th}$ with respect to $i^{th}$ criterion, which are the rating value of user $i$ on item $j$, respectively. According to the absences ratings, a threshold will be used to avoid multiplication or division problem with zero during the execution of operations of TOPSIS. Since all criteria cannot be assumed to be of equal importance, the technique needs a set of weights parameters, provided by the decision maker, which associated with the criteria. Thus, the similarity weights of neighbours proposed to be the weights that associated with the *K*-neighbours.

#### 4.2.4.1 TOPSIS Steps

Before starting the procedures of TOPSIS let define the basic sets:

- $A$ is a set of candidate items that represents the alternatives where $A = \{a_1, a_2, \;, a_j, \dots, a_{m-1}, a_m\}$, where $j = 1, 2, \dots, m$ and $m$ is the total number of candidate items.
- $C$ is a set of neighbours that represents the criteria where $C = \{c_1, c_2, \dots, c_i, \dots, c_{n-1}, c_n\}$, where $i = 1, 2, \dots, n$ and $n$ denotes the size of criteria.
- $X$ is a set of ratings where $X = \{x_{j,i} \mid j = 1, \dots, m;\ i = 1, \dots, n\}$, $x_{j,i}$ is the rating value of the $j^{th}$ alternatives/candidate item with respect to the $i^{th}$ criteria/neighbour user.
- W represents a set of weights $w = \{w_1, w_2, \dots, w_i, \dots, w_{n-1}, w_n | i = 1, 2, \dots, n\}$, where $w_i$ is the weight of the $i^{th}$ criteria/ neighbour.

Therefore, the decision matrix X which contains *m* alternatives associated with *n* criteria can be represented as shown in Table 4.8.

Table 4.8 A Conceptual decision matrix X

| | $c_1$ | $c_2$ | $\cdots$ | $c_i$ | $\cdots$ | $c_{n-1}$ | $c_n$ |
|---|---|---|---|---|---|---|---|
| $a_1$ | $x_{1,1}$ | $x_{1,2}$ | $\cdots$ | $x_{1,i}$ | $\cdots$ | $x_{1,n-1}$ | $x_{1,n}$ |
| $a_2$ | $x_{2,1}$ | $x_{2,2}$ | $\cdots$ | $x_{2,i}$ | $\cdots$ | $x_{2,n-1}$ | $x_{2,n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $a_j$ | $x_{j,1}$ | $x_{j,2}$ | $\cdots$ | $x_{j,i}$ | $\cdots$ | $x_{j,n-1}$ | $x_{j,n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $a_{m-1}$ | $x_{m-1,1}$ | $x_{m-1,2}$ | $\cdots$ | $x_{m-1,i}$ | $\cdots$ | $x_{m-1,n-1}$ | $x_{n-1,n}$ |
| $a_m$ | $x_{m,1}$ | $x_{m,2}$ | $\cdots$ | $x_{m,i}$ | $x_{m,1}$ | $x_{m,n-1}$ | $x_{n,n}$ |

For further clarification, the steps of TOPSIS will be described in a series of successive steps as follow:

**Step 1:** Construct the normalized decision matrix:

Some users prefer to give a high rating; even they do not like the item very much. However, some users tend to give low ratings, even they like the items very much. Therefore, the decision matrix includes the different criteria scale that requires being normalized, which allows comparison of the criteria; to be able to compare the measure on different ways of ratings. One way can be found for this purpose is distributive normalization. It requires that the rating values are divided by the square root of the sum of each squared alternative in a column. Thus, the value $r_{j,i}$ of the normalized decision matrix *R* can be computed using Equation 4.8:

$$r_{j,i} = \frac{x_{j,i}}{\sqrt{\sum_{j=1}^{m} x_{j,i}{}^2}}\,, \qquad j = 1 \ldots m;\; i = 1 \ldots n. \qquad 4.8$$

By applying the distributive normalization method, Equation 4.8, on matrix X the result will be presented as shown in matrix R in Table 4.9.

Table 4.9 A Conceptual normalized decision matrix R

| | $c_1$ | $c_2$ | $\cdots$ | $c_i$ | $\cdots$ | $c_{n-1}$ | $c_n$ |
|---|---|---|---|---|---|---|---|
| $a_1$ | $r_{1,1}$ | $r_{1,2}$ | $\cdots$ | $r_{1,i}$ | $\cdots$ | $r_{1,n-1}$ | $r_{1,n}$ |
| $a_2$ | $r_{2,1}$ | $r_{2,2}$ | $\cdots$ | $r_{2,i}$ | $\cdots$ | $r_{2,n-1}$ | $r_{2,n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $a_j$ | $r_{j,1}$ | $r_{j,2}$ | $\cdots$ | $r_{j,i}$ | $\cdots$ | $r_{j,n-1}$ | $r_{j,n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $a_{m-1}$ | $r_{m-1,1}$ | $r_{m-1,2}$ | $\cdots$ | $r_{m-1,i}$ | $\cdots$ | $r_{m-1,n-1}$ | $r_{n-1,n}$ |
| $a_m$ | $r_{m,1}$ | $r_{m,2}$ | $\cdots$ | $r_{m,i}$ | $r_{m,1}$ | $r_{m,n-1}$ | $r_{n,n}$ |

**Step 2:** Construct the weighted normalized decision matrix:

The set of weights W provided by the decision maker are taken into account. The weighted normalized decision matrix V can be calculated by multiplying the normalized value $r_{j,i}$ by its corresponding weights $w_i$. In this work, the similarity weights of the target user with his/her neighbours will be used to present the weights criteria associated with each user from the neighbours. For example, let *u* a target user who has *k* neighbours and $s_u$ a set of similarity weights where $s_u = \{s_{u,1}, s_{u,2}, \ldots, s_{u,i}, \ldots, s_{u,n-1}, s_{u,n} | i = 1, 2, \ldots, n\}$; where $s_{u,i}$ denotes the similarity value between the target user $u$ and $i^{th}$ neighbour. Thus, the $w_i$ weight value will be equalled $s_{u,i}$ similarity value and so on. Therefore, Table 4.10 represents the weighted normalized decision matrix *V*, which obtained by applying the Equation 4.9.

$$v_{j,i} = r_{j,i} * w_i\,, \qquad j = 1 \ldots m;\; i = 1 \ldots n. \qquad 4.9$$

Table 4.10 A Conceptual weighted normalized decision matrix V

| | $c_1$ | $c_2$ | $\cdots$ | $c_i$ | $\cdots$ | $c_{n-1}$ | $c_n$ |
|---|---|---|---|---|---|---|---|
| $a_1$ | $v_{1,1}$ | $v_{1,2}$ | $\cdots$ | $v_{1,i}$ | $\cdots$ | $v_{1,n-1}$ | $v_{1,n}$ |
| $a_2$ | $v_{2,1}$ | $v_{2,2}$ | $\cdots$ | $v_{2,i}$ | $\cdots$ | $v_{2,n-1}$ | $v_{2,n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $a_j$ | $v_{j,1}$ | $v_{j,2}$ | $\cdots$ | $v_{j,i}$ | $\cdots$ | $v_{j,n-1}$ | $v_{j,n}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| $a_{m-1}$ | $v_{m-1,1}$ | $v_{m-1,2}$ | $\cdots$ | $v_{m-1,i}$ | $\cdots$ | $v_{m-1,n-1}$ | $v_{n-1,n}$ |
| $a_m$ | $v_{m,1}$ | $v_{m,2}$ | $\cdots$ | $v_{m,i}$ | $v_{m,1}$ | $v_{m,n-1}$ | $v_{n,n}$ |

**Step 3:** Determine positive and negative ideal solutions:

The best and worst alternatives evaluations on each criterion of the normalized decision matrix V are collected to represent the ideal and negative-ideal solutions, respectively.

Let $I_1$ represents a set of positive attributes or criteria (the more is better) and $I_2$ represents a set of negative attributes or criteria (less is better). In other word, $I_1$ the criteria that associated with benefit and $I_2$ the criteria that associated with cost. For example, if you to buy a car, then the set of benefit attributes: style, reliability, fuel economy (more is better), and the set of negative attributes: cost (less is better). The positive and negative ideal solutions can defined as follow:

- Ideal solution:

$$A^* = \left\{\max\left(v_{j,i}\right) if\ c_i \in I_1 ; \min\left(v_{j,i}\right) if\ c_i \in I_2\right\} \\ = \{ v_1{}^*, \dots, v_i{}^*, \dots, v_n{}^* \}$$

- Negative-ideal solution:

$$A' = \left\{\min\left(v_{j,i}\right) if\ c_i \in I_1 ; \max\left(v_{j,i}\right) if\ c_i \in I_2\right\} \\ = \{ v_1{}', \dots, v_i{}', \dots, v_n{}' \}$$

As a result, two alternatives $A^*$ and $A'$ are created, which represent the most favourite option (ideal solution) and least favourite option (negative-ideal solution), respectively.

**Step 4:** Calculate the separation measure:

The distance for each alternative to the ideal and negative-ideal solutions for all alternatives can be calculated using the Euclidean distance measurement. Table 4.11 represents the Separation Matrix V′

- The distance of each alternative from the ideal can be computed by Equation 4.10.

$$S_j{}^* = \sqrt[2]{\sum_{i=1}^{n} (v_{j,i} - v_j{}^*)^2} \quad , j = 1, 2, \dots, m \qquad 4.10$$

- Similarly, the distance of each alternative from the negative-ideal one is given by Equation 4.11.

$$S_j' = \sqrt[2]{\sum_{i=1}^{n} (v_{j,i} - v_j')^2} \ , j = 1, 2, \dots, m \qquad 4.11$$

Table 4.11 A Conceptual separation matrix $V'$

| | $S^*$ | $S'$ |
|---|---|---|
| $a_1$ | $S_1{}^*$ | $S_1{}'$ |
| $a_2$ | $S_2{}^*$ | $S_2{}'$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $a_j$ | $S_j{}^*$ | $S_j{}'$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $a_{m-1}$ | $S_{m-1}{}^*$ | $S_{m-1}{}'$ |
| $a_m$ | $S_m{}^*$ | $S_m{}'$ |

**Step 5:** Calculate the relative closeness to the ideal solution:

For each alternative the degree of closeness with respect to the ideal solution $A^*$ can be calculated by Equation 4.12:

$$C^*{}_j = S_j'/(S_j{}^* + S_j'), \qquad 4.12$$
$$0 < C^*{}_j < 1, \ j = 1, 2, \dots, m$$

The relative closeness rating is a number between 0 and 1, with zero being the worst possible and one the best favoured the alternative. To explain, if the distance of an alternative $a_j$ from ideal solution $A^*$ smaller than negative-ideal $A'$, then $C^*{}_j$ is getting closer to 1. Whereas, if the alternative is closer to the negative-ideal $A'$ than to the ideal $A^*$, then the $C^*{}_j$ approaches 0, see Figure 4.4.

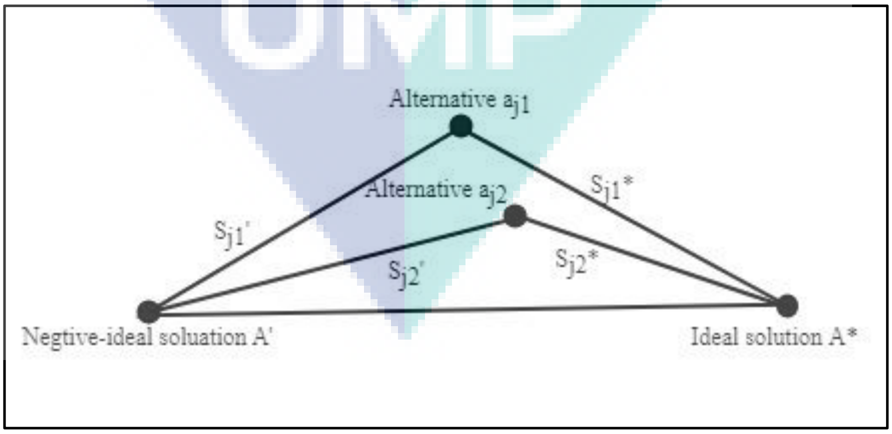



Figure 4.4 Euclidean Distances to the Ideal and Negative-Ideal Solutions

**Step 6:** Ranking the alternatives order according to the $C^*{}_j$.

Determine the preferences order by arranging the alternative in the descending order of $C^*{}_j, j = 1, 2, .., m$. Therefore, the ranks outcome using TOPSIS is with a sorted list of the alternatives.

#### 4.2.4.2 TOPSIS Numerical Example

Let basic sets defined as follow:

- A is a set of candidate items, where $A = \{a_1, a_2, a_3, a_4, a_5\}$.
- C is a set of neighbours, where $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$.
- X is a set of ratings where X = $\{a_{j,i} \mid j = 5, \dots, m;\ i = 1, \dots, 6\}$, $a_{j,i}$ is the rating value of the $j^{th}$ candidate item with respect to the $i^{th}$ neighbour user.
- W represents a set of weights $w = \{w_1, w_2, w_3, w_4 w_5, w_6\}$, where $w_1$ is the similarity weight of the $1^{th}$ neighbour with the target user.

Therefore, the decision matrix X which contains *5* alternatives associated with *6* criteria can be represented as shown in Table 4.12.

Table 4.12 Decision matrix X

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ |
|---|---|---|---|---|---|---|
| $a_1$ | 4 | 3 | 4 | 0 | 2 | 3.5 |
| $a_2$ | 5 | 4 | 4 | 3 | 3.5 | 0 |
| $a_3$ | 3 | 0 | 3.5 | 3.5 | 3 | 5 |
| $a_4$ | 0 | 5 | 3 | 4 | 3.5 | 0 |
| $a_5$ | 3.5 | 3.5 | 2 | 5 | 0 | 4 |

**Step 1:** Calculate the normalized decision matrix R:

$$v_{j,i} = r_{j,i} * w_i, \qquad j = 1, 2, 3, 4, 5;\ i = 1, 2, 3, 4, 5, 6$$

Table 4.13 Normalized decision matrix R

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ |
|---|---|---|---|---|---|---|
| $a_1$ | 0.5069794 | 0.380235 | 0.528655 | 0 | 0.326599 | 0.479632 |
| $a_2$ | 0.6337243 | 0.506979 | 0.528655 | 0.380235 | 0.571548 | 0 |
| $a_3$ | 0.3802346 | 0 | 0.462573 | 0.443607 | 0.489898 | 0.685189 |
| $a_4$ | 0 | 0.633724 | 0.396491 | 0.506979 | 0.571548 | 0 |
| $a_5$ | 0.443607 | 0.443607 | 0.264327 | 0.633724 | 0 | 0.548151 |

**Step 2:** Construct the weighted normalized decision matrix:

Assume that the relative importance of attributes is the weights similarity between the target user and his neighbours, where $w = \{.95, .92, .88, .83, .79, .76\}$. Then, the weighted decision matrix V is represented as shown in Table 4.14.

Table 4.14 Weighted normalized decision matrix V

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ |
|---|---|---|---|---|---|---|
| $a_1$ | 0.48163 | 0.349816 | 0.465216 | 0 | 0.258013 | 0.36452 |
| $a_2$ | 0.602038 | 0.466421 | 0.465216 | 0.315595 | 0.451523 | 0 |
| $a_3$ | 0.361223 | 0 | 0.407064 | 0.368194 | 0.387019 | 0.520743 |
| $a_4$ | 0 | 0.583026 | 0.348912 | 0.420793 | 0.451523 | 0 |
| $a_5$ | 0.421427 | 0.408118 | 0.232608 | 0.525991 | 0 | 0.416595 |

**Step 3:** Determine the positive and negative ideal solutions:

Let $I_1$ is the set of positive criteria (the more is better), then the:

- deal solution: $A^* = \left\{\max\left(v_{j,i}\right), c_i \in I_1\right\}$

  A*= {0.602038, 0.583026, 0.465216, 0.525991, 0.451523, 0.520743}.
- Negative-ideal solution: $A' = \left\{\min\left(v_{j,i}\right), c_i \in I_1\right\}$

  A'= {0, 0, 0.232608, 0, 0, 0}.

**Step 4:** Calculate the separation measure:

$$S_j{}^* = \sqrt[2]{\sum_{i=1}^{6}(v_{j,i} - v_j{}^*)^2} \; , j = 1, 2, 3, 4, 5.$$

$$S_j{}' = \sqrt[2]{\sum_{i=1}^{6}(v_{j,i} - v_j{}')^2} \; , j = 1, 2, 3, 4, 5.$$

Table 4.15 Separation matrix V'

| | $S^*$ | $S'$ |
|---|---|---|
| $a_1$ | 0.638281 | 0.877617 |
| $a_2$ | 0.573617 | 1.04876 |
| $a_3$ | 0.656013 | 0.923422 |
| $a_4$ | 0.811305 | 0.917932 |
| $a_5$ | 0.576229 | 0.921133 |

**Step 5:** Calculate the relative closeness to the ideal solution:

$$C^*{}_1 = S_1{}'/(S_1{}^* + S_1{}').$$

Table 4.16 The loseness to the ideal solution

| | $C^*$ |
|---|---|
| $a_1$ | 0.421058 |
| $a_2$ | 0.353566 |
| $a_3$ | 0.415347 |
| $a_4$ | 0.469169 |
| $a_5$ | 0.384829 |

**Step 6:** Ranking the alternatives order according to the $C^*{}_j$:

According to the descending order of C *, the preference order is:

Table 4.17 Alternatives ranking

| | $C^*$ |
|---|---|
| $a_4$ | 0.469169 |
| $a_1$ | 0.421058 |
| $a_3$ | 0.415347 |
| $a_5$ | 0.384829 |
| $a_2$ | 0.353566 |

From Table 4.17, we note that the item $a_4$ is the best alternative and the $a_2$ is the worst alternative. For example if the recommendation list size is three then the recommendation list will be contained $a_4\ a_3\ and\ a_3$.

### 4.2.5 Generating Recommendations/Output

As described before, the output that will be produced from the CF-NSMA technique is a list of sorted alternatives (candidate items). These items are sorted according to the importance measurement based on several criteria (k-neighbours). In the final phase of this technique, Top-M items will be selected to present set of suggestions. These suggestions provided to the target user as recommendations. Sometimes, the recommendation is not guaranteed to contain the items with the actual highest possible preferences of the target user but may help the user in choosing the appropriate items.

## 4.3 Implementation

The strategies of implementation of the proposed CF-NSMA technique are defined. The implementation strategy consists of several steps. Starting by selecting the suitable dataset for evaluating the proposed technique and ended by producing the recommendations. Next, the structure of chosen datasets that will be used to test the proposed technique is exemplified. Moreover, the evaluation process and the metrics that will be used in the evaluation process are specified. The holdout and k-fold cross-validation partition techniques are defined to use the selected metrics. Finally, to show the strength of the proposed similarity, a running example is presented and compared the result of other similarity methods. For more details about these points, see Appendix B.

## 4.4 Chapter Summary

In this chapter, the proposed CF-NSMA has been explained in detail. This technique consists of four major components which are: constructing the normalized user-type matrix, neighbours formation, ranking candidate items using the MADM method, and generating the recommendations. Some of these components consist of sub-components. This chapter began with a brief introduction about the primary objective of the chapter. Then, the CF-NSMA technique architecture is illustrated in a thumbnail image with a brief description. Next, the components of the technique were explained start by the inputs component. The second phase was constructing the normalized user-type matrix that passed through three steps which are described in sub-sections. In the third component, the formation of neighbours phase has been illuminated in details. This phase included three steps which are: formulating similarity measure to compute the similarity between users, calculation similarity between users, and determine the neighbours of each user based on similarity weights. After that, the steps of TOPSIS technique are clarified. TOPSIS used in this work to rank the candidate items based on ratings of neighbours for each target user. Finally, generating recommendations was the process description that ended this chapter.

# CHAPTER 5

# RESULTS AND DISCUSSION

## 5.1 Introduction

The third objective of this thesis is addressed in this chapter. Several experiments were conducted to show the preceding of the CF-NSMA technique and how well it works. This work compared the experiment's results of CF-NSMA technique, specified accuracy metrics, with popular and widely used memory-based CF methods. These methods are listed, discussed and justified in Chapter 2 Section 2.5. First, presenting the new similarity method CF-BSF (Section 4.2.3) by comparing its results with common similarity methods to prove the proposed similarity method precedes. Second, to show the notable improvement that made by the proposed technique CF-NSMA, the accuracy of CF-NSMA technique will be presented and compared to the selected memory-based CF methods. Additionally, to show the positive effect of the MADM method on the accuracy of traditional memory-based CF, the MADM method is applied on those memory-based CF methods, and its results are presented and compared without MADM method. All experiments are conducted on 100K & 1M MovieLens public datasets. The holdout and cross-validation splitting methods are used to partition these datasets into training and testing sets. The comparison was regarding accuracy using most common metrics used in the accuracy evaluation process of the CF. Finally; this chapter will be ended by the conclusion section.

## 5.2 Proposed Similarity Method (CF-BSF) Vs Traditional Similarity Methods

In this section, to show the improvement in the prediction and accuracy for the proposed similarity method (CF-BSF), several experiments were conducted on 100K & 1M MovieLens datasets. The holdout and cross-validation methods were used to partition these datasets into training and testing sets. The results will be averaged using variation number of neighbours and size of recommended items as main parameters which are: 10, 20, 30, 40, and 50. Next, the averaged result will be compared to traditional memory-based CF methods. For more details, the Appendix E & F provide a detailed result with varying size of neighbours and number of recommended items parameters. Finally, the comparison results will be presented in the bar charts to show the enhancement made by CF-BSF in terms of prediction accuracy using MAE measurement and performance accuracy using Recall, Precision, and F-Measure measures.

### 5.2.1 Prediction Accuracy

In this section, the MAE metric is used to compare the prediction accuracy. The holdout and cross-validation partition methods are applied on both datasets (100K & 1M MovieLens).

The bar chart in Figure 5.1 illustrates the MAE rate of CF-BSF compare to traditional memory-based CF methods (CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, and CF-NHSM). The size of neighbours was represented by variation size: 30, 50, 70, 100, and 150. The horizontal axis indicates the datasets and splitting methods used which are: 100K & 1M datasets using holdout and cross-validation methods, respectively. In general, there is an improvement in the MAE using the CF-BSF method. As it is presented in that bar graph, compare to all comparative similarity methods except NHSM, which has a tiny proportion improvement, the CF-BSF method has the lowest prediction accuracy in all cases. Due to the NHSM method considered three factors, one of them is the differences in rating between both of users that lead to devalue the similarity weight and in turn lead to decrease the prediction values. In contrast, the worst MAE rate was using CF-Cosine, CF_CPCC, CF_PCC, and CF_SPCC, respectively.

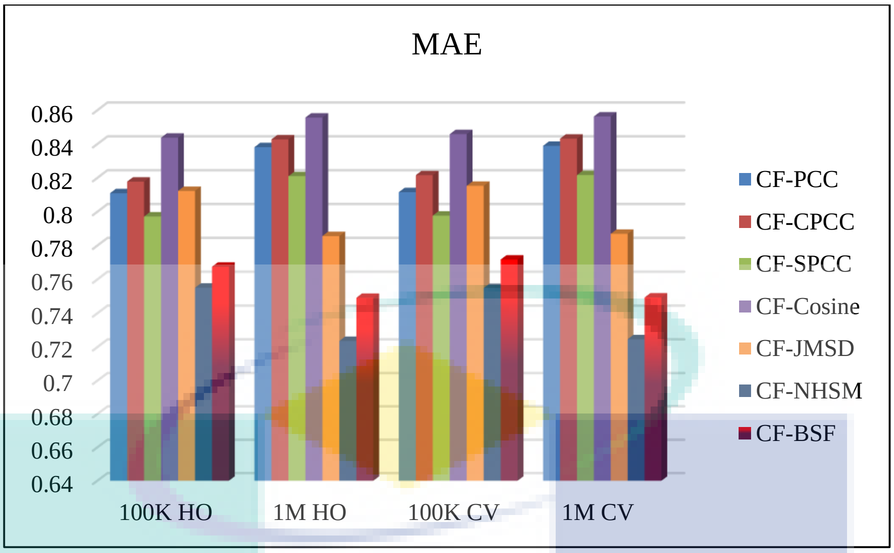



Figure 5.1 Compare MAE between CF-BSF and traditional memory-based CF methods.

### 5.2.2 Performance Accuracy

In this section, the performance accuracy of proposed similarity method CF-BSF will be measured using three commonly used metrics (Recall, Precision, and F-measure). The experiments were conducted on 100K & 1M datasets using holdout and cross-validation partition methods. The results that are shown in all Figure 5.2, Figure 5.3 and Figure 5.4 represent the averaging rats of Recall, Precision, and F-measure, respectively. The size of neighbours and recommended items take different values (10, 20, 30, 40 and 50).

#### 5.2.2.1 Recall Metric

One of metrics used to measure the performance accuracy of a recommender system is the recall measure. In this subsection, it was used to compare the performance accuracy of CF-BSF via applying holdout and cross-validation methods on 100K & 1M MovieLens datasets.

Figure 5.2 illustrates the comparison of recall metric rate between CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, CF-NHSM and the proposed CF-BSF similarity methods. The x-axis presents the datasets that are used using holdout and cross-validation splitting methods, respectively. The number of recommendations and neighbours take different values 10, 20, 30, 40, and 50. In general, the recall rate of CF-BSF was the highest overall cases. Whereas, the recall rates of CF-PCC and its derivatives methods were the lowest. According to the CF-Cosine recall rate it also was close to the CF-PCC rate. While CF-JMSD and CF-NHSM recall percentage, it was good when compared to CF-PCC and CF-Cosine methods, but still low when compared to CF-BSF. To sum up, as we can see in Figure 5.2, the proposed similarity method CF-BSF improved the accuracy in term of recall where its recall percentage was the highest.

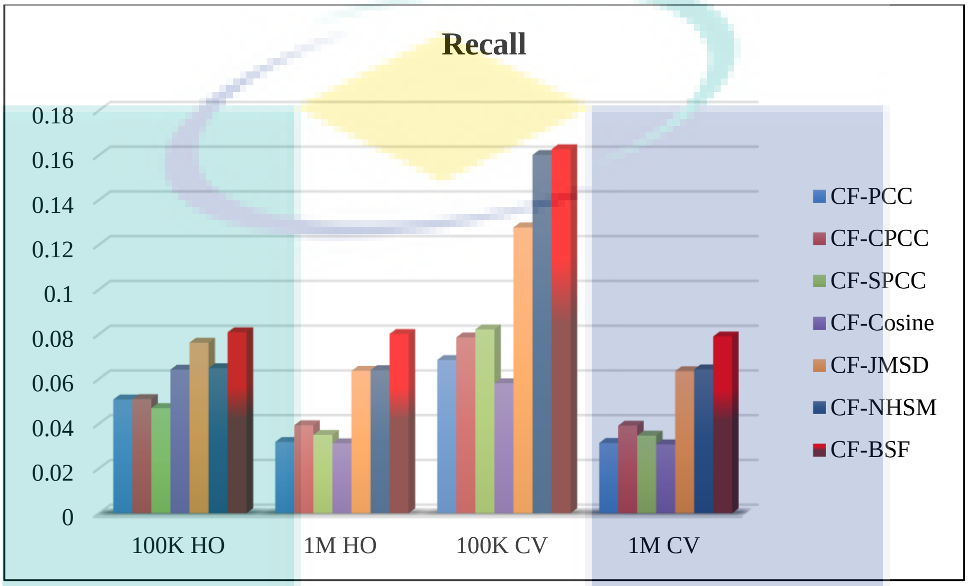



Figure 5.2 Recall rate on 100K & 1M datasets, using holdout and cross-validation.

#### 5.2.2.2 Precision Metric

The second metric will be used to measure the performance accuracy of the proposed method is the precision measurement. The holdout and cross-validation methods were applied on 100K & 1M MovieLens to compare the precision accuracy of CF-BSF with traditional memory-based CF methods. The result was finalised by averaging the results of vibration size of recommended items and neighbours.

Figure 5.3 gives comparative information about the precision rate for CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, CF-NHSM and the proposed CF-BSF similarity methods using holdout and cross-validation partition methods on 100K & 1M datasets, respectively. At first glance, it is clear that the precision rate the precision of CF-BSF was the highest over all cases. As it can be seen from the figure, the precision rates of CF-JMSD and CF-NHSM have good percentages while the other methods (CF-PCC, CF-SPCC, CF-CPCC, and CF-Cosine) have a low level of percentage. In summary, the accuracy of recommendation in term of precision has been improved using CF-BSF compare to others.

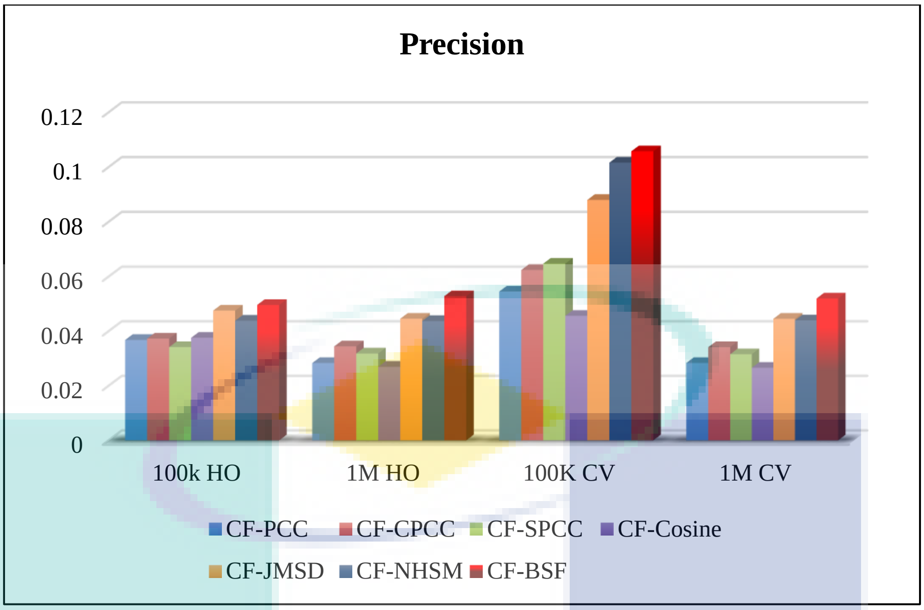



Figure 5.3 Precision rate on 100K & 1M datasets, using holdout and cross-validation.

#### 5.2.2.3 F-Measure Metric

F-measure metric is a combined metric of precision and recall, it gives different information, the weighted mean of precision and recall, compared to precision and recall. The holdout and cross-validation methods were applied on 100K & 1M MovieLens to test the F-measure value of CF-BSF. The number of recommended items and size of neighbours was equal 10, 20, 30, 40, and 50.

Figure 5.4 shows the percentage of F-measure for CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, CF-NHSM and the proposed CF-BSF similarity methods. The x-axis represents the dataset that is used (100K & 1M) with the partition methods (holdout and cross-validation). It has been observed from the graph that, for all methods, the F-measure rate of CF-BSF was the highest overall cases. Whereas, CF-NHSM and CF-JMSD have a good percentage when compared to other traditional methods (CF-PCC, CF-SPCC, CF-CPCC, and CF-Cosine). Therefore, there is a notable improvement made by the proposed method in term of F-measure.

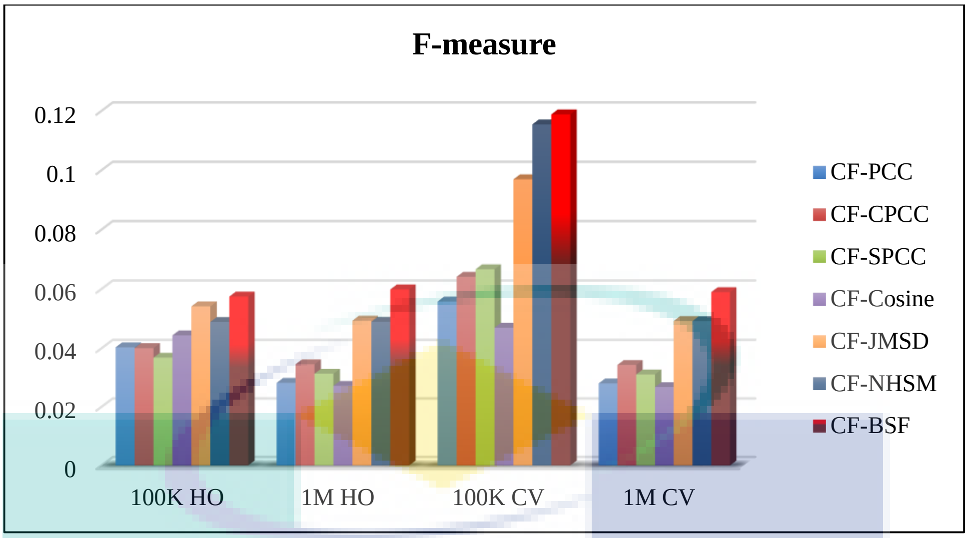



Figure 5.4 F-measure rate on 100K & 1M datasets, using holdout and cross-validation.

## 5.3 Proposed Technique (CF-NSMA) Vs Traditional Memory-Based CF Methods

In this section, the proposed technique CF-NSMA will be compared with traditional memory-based CF methods to show the preceding of its performance accuracy through conducting several experiments. Holdout and cross-validation partition methods are used on 100K & 1M MovieLens datasets. The result is presented in the bar graphs which show the notable improvement that made by CF-NSMA when compared to traditional memory-based CF methods. Three common widely metrics are used to measure the performance accuracy of CF which are a recall, precision, and F-Measure. The results that show in all the next figures represent the performance accuracy by averaging variation size of recommended items and neighbours (10, 20, 30, 40 and 50).

### 5.3.1 Recall metric

Figure 5.5 presents the comparison of recall between CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, CF-NHSM, CF-BSF and the proposed CF-NSMA technique. The horizontal axis represents two datasets (100K & 1M Movielens) with two splitting techniques. The number of recommendations and neighbours was 10, 20, 30, 40, and 50. In general, the recall rate of CF-NSMA has a significant improvement when compared to the recall rates using traditional methods. Whereas, the best recall rate of comparative methods did not exceed 0.1 with both datasets except the proportion recall of the CF based on JMSD and NHSM. Their recall percentage was around 0.14 and 0.16, respectively. Therefore, the improvement made by the proposed technique CF-NSMA in the accuracy in term of recall overcomes the traditional by around five times overall cases.

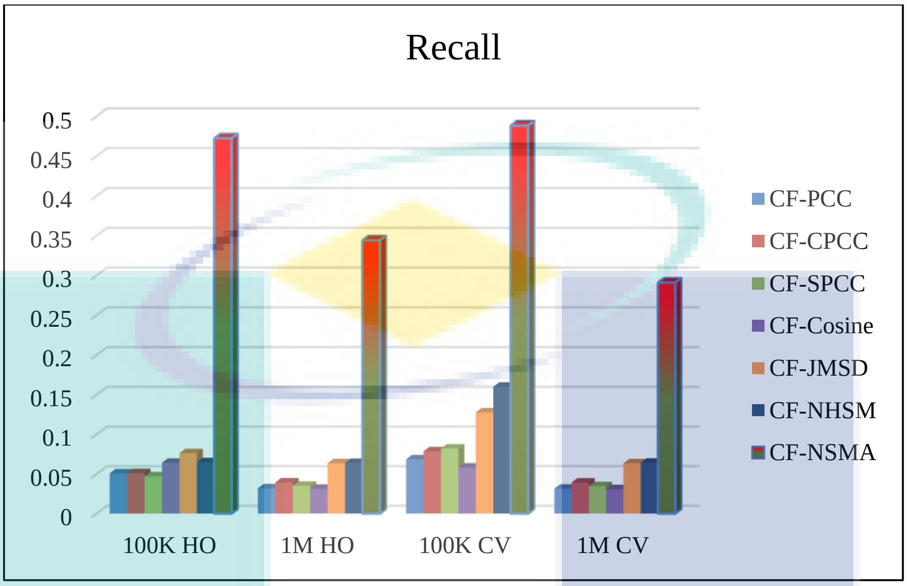



Figure 5.5 Recall comparison of traditional CF methods and CF-NSMA on 100K & 1M datasets, using holdout and cross-validation.

### 5.3.2 Precision Metric

Figure 5.6 gives information about the precision rate for CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, CF-BSF and CF-NHSM and proposed CF-NSMA technique. This comparison of precision percentage was using holdout and cross-validation partition method on 100K & 1M datasets. In general, the rate of CF-NSMA has a notable improvement compared to other methods. The precision value of CF-NSMA was more than 0.25 overall cases except in the case of the 1M dataset with cross-validation where was around 0.23. Moreover, there is a big difference between its precision rate and the traditional methods precision rate. However, it can be seen that the precision rates of all traditional methods do not exceed the 0.1. Therefore, the proposed technique CF-NSMA has a tremendous proportion precision rate when compared to other methods' rate which keeps it the highest accuracy with a significant improvement.

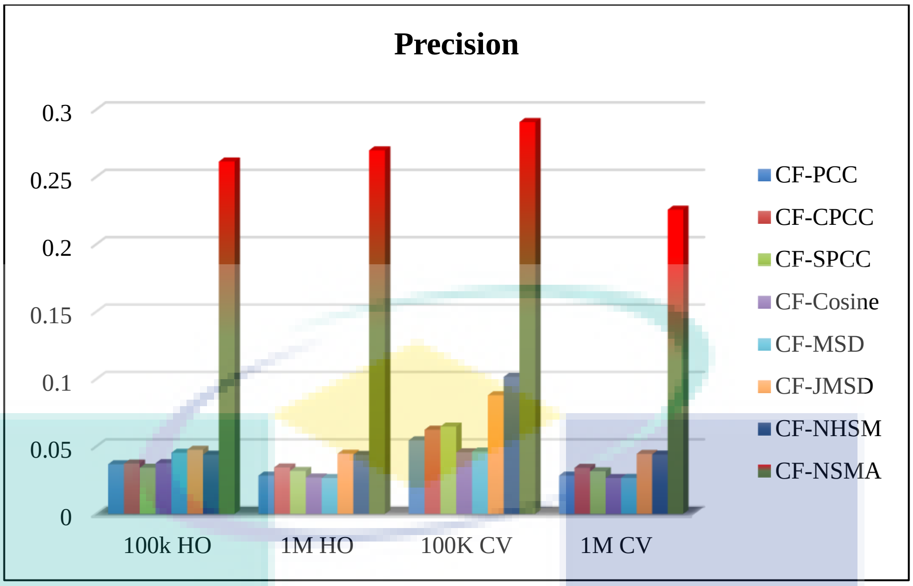



Figure 5.6 Precision comparison of traditional CF methods and CF-NSMA on 100K & 1M datasets, using holdout and cross-validation.

Figure 5.7 gives sample information about the precision rate for CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, and CF-NHSM and proposed CF-NSMA technique. The holdout and cross-validation partition methods were used on 100K & 1M datasets. The number of recommendations size is fixed by 10.

In general, the rate of CF-NSMA has a notable improvement when compared to baseline methods. The precision value was around 0.4 over all cases, except in case of 1M CV which was around 0.33. However, there is a big difference between its precision rate and the baseline methods precision rate. Moreover, it can be seen that the precision rates of all baseline methods do not exceed the 0.12 at its best. Therefore, the proposed technique has a very large proportion precision rate compare to traditional rate which keeps it the highest accuracy with a big improvement.

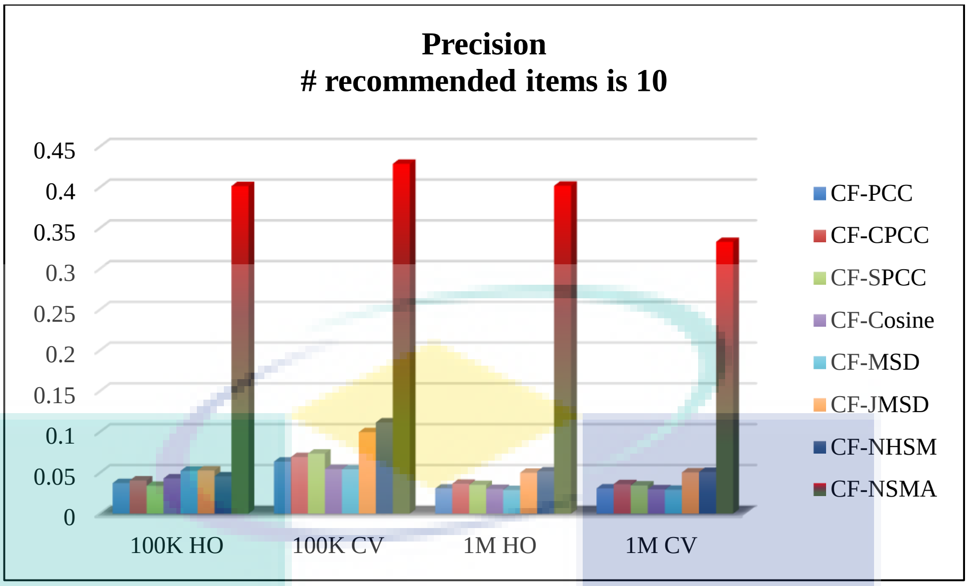



Figure 5.7 Precision comparison of CF-NSMA with baseline CF methods.

### 5.3.3 F-Measure Metric

Figure 5.8 compares the proportion of F-measure between CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, CF-NHSM, CF-BSF and CF-NSMA technique. At the onset, it is clear that the F-measure rate of CF-NSMA has a significant majority improvement compare to the rate of traditional methods. The F-measure value was over 0.24. In contrast, the highest rate of the traditional methods was approximately 0.11 for CF-NHSM and CF-JMSD. While, the maximum rate of CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, and CF-JMSD do not exceed the 0.07 in all cases. To conclude, it is still a big difference between its F-measure rate and the traditional methods F-measure rate. Thus, the proposed technique has the highest F-measure rate with approximately more than three-quarter improvement.

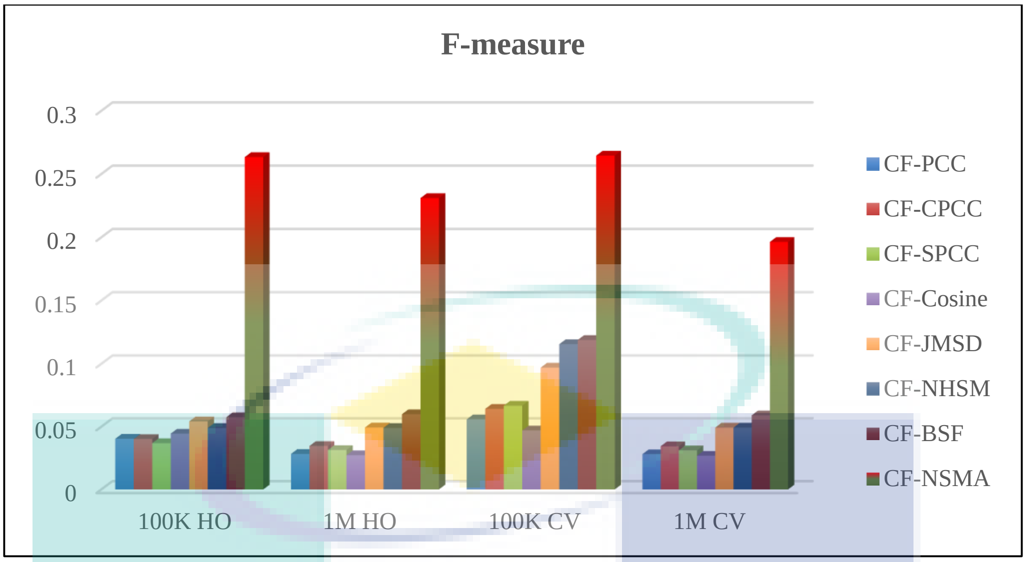



Figure 5.8 F-measure comparison of traditional CF methods and CF-NSMA on 100K & 1M datasets, using holdout and cross-validation.

## 5.4 Traditional Methods based on MADM Method

In this section, the prediction method in the traditional memory-based CF will be replaced by the MADM method to show the positive effect of MADAM on the traditional methods performance accuracy. Some experiments were conducted using holdout and cross-validation partition methods on 100K & 1M MovieLens datasets. The results were presented in the bar graphs which show the improvement that made by the MADM method when is compared to the traditional memory-based CF methods without MADM method. Three metrics were used to measure the performance accuracy. These metrics are: Recall, Precision, and F-Measure. The results that shown in all the next bar graphs illustrate the performance accuracy by averaging variation size of recommendations and neighbours (10, 20, 30, 40 and 50). For more details, see results in appendix F.

### 5.4.1 Recall Metric

Figure 5.9 compares the proportion of recall between CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, and CF-NHSM. At the onset, it is clear that the recall rate of using MADM method have a sharp improvement when compared to without MADM method. In that figure, the CF-NHSM using MADM was the highest compared to others. Moreover, all methods have a significant enhancement by around twice as much compare to its recall without MADM method. In conclusion, the improvement which made by

MADM method on traditional methods shows the importance of MADM method in improving the performance accuracy of memory-based CF.

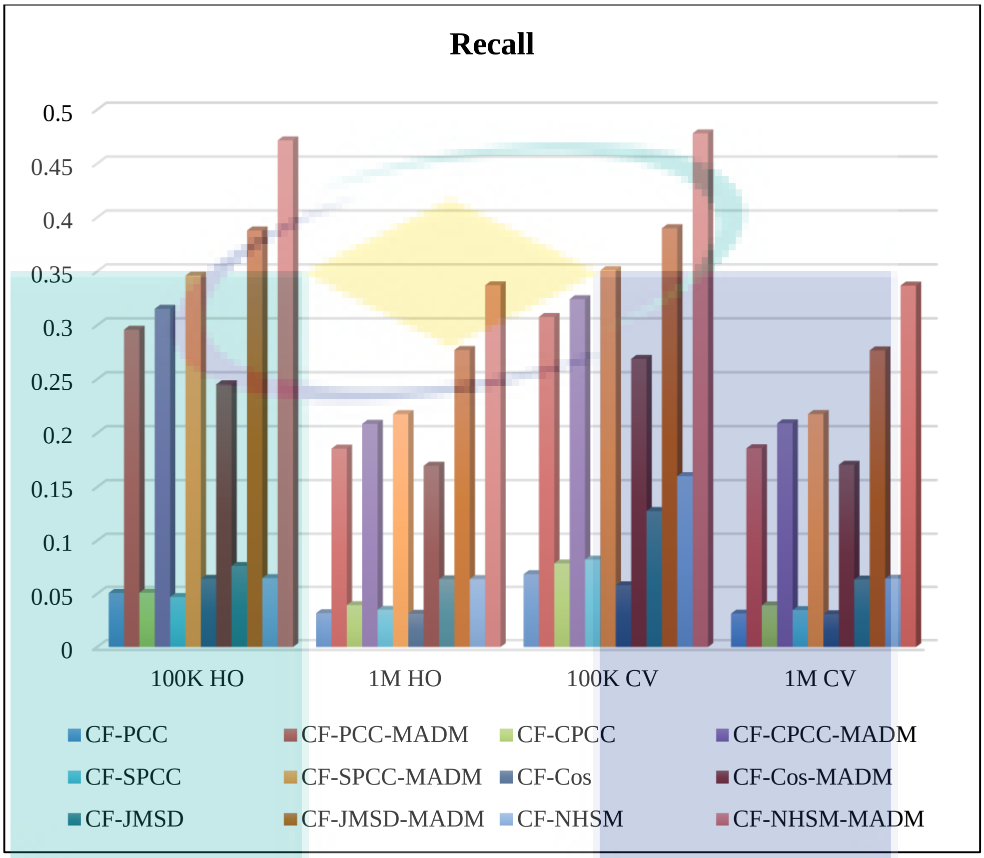



Figure 5.9 Recall comparison of traditional CF methods with and without MADM method on 100K & 1M datasets, using holdout and cross-validation.

### 5.4.2 Precision Metric

Figure 5.10 shows comparative data of precision for CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, and CF-NHSM with and without MADM method. In general, it is clear that the precision rate of CF- NHSM based on MADM has the highest rate overall cases. According to the other methods, they have a significant enhancement by around three times compare to its precision rate without MADM method. In conclusion, the improvement which made by the MADM method shows the importance of MADM method in improving the precision accuracy of memory-based CF.

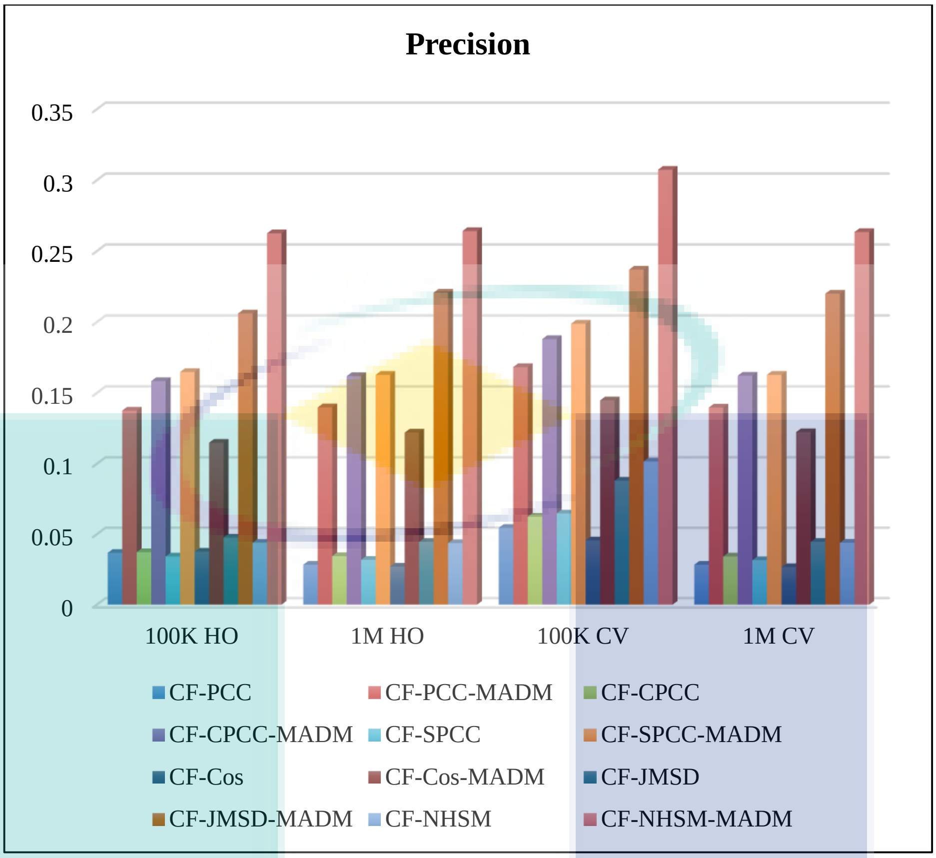



Figure 5.10 Precision comparison of traditional CF methods with and without MADM method on 100K & 1M datasets, using holdout and cross-validation.

### 5.4.3 F-measure Metric

Figure 5.11 compares the proportion of F-measure between CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, and CF-NHSM with and without-MADM technique. The comparison shows F-measure rate when holdout and cross-validation partition methods applied on 100K & 1M datasets. In general, it can be seen that all methods have a notable enhancement compare to its F-measure without MADM method. At the onset, it is clear that the recall rate of CF-NHSM using MADM has the highest rate overall cases. In conclusion, the enhancement which made by MADM method on traditional memory-based CF methods was an approximately three-quarter improvement. Thus, the result shows that the importance of the MADM method in improving the performance accuracy of memory-based CF methods.

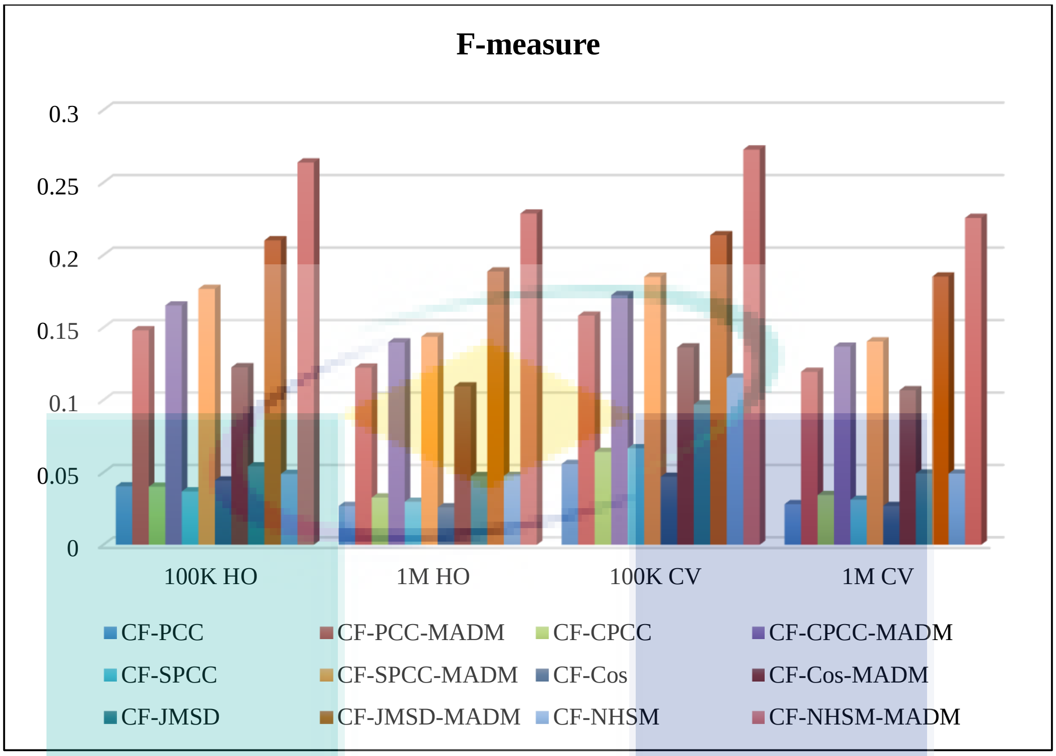



Figure 5.11 F-measure comparison of traditional CF methods with and without MADM method on 100K & 1M datasets, using holdout and cross-validation.

Generally, the RS does not guarantee that the suggested items will be relevant to the preferences of the target user, but may encourage users to find useful or interesting items. Therefore, the accuracy of the RS is affected by the user's subsequent selection from the list of recommendations. For instance, if the recommendation list contains 10 items and the user selects just four, then the accuracy will be negatively affected by the user disregarding the other six items. Thus, in this study, the experimental results above clearly show that the application of TOPSIS to the baseline methods results in better accuracy. Although the general accuracy of the proposed method is less than 0.3 in term of precision, the accuracy of all baseline methods is lower than that of the proposed method. For instance, the precision of the baseline methods does not exceed 0.1, except for NHSM, which scored was around 0.1 using 100K MovieLens. Furthermore, the maximum precision when TOPSIS was applied to NHSM reached 0.3 on the 100K. The prediction accuracy percentage in terms of MAE was around 0.76 and 0.74 via 100K and 1M datasets, respectively. While the improvement of the CF-NSMA technique in terms of performance accuracy was around more than three-fold in term precision, more four-fold in term of recall, and around three-fold in term of F-measure. The low accuracy of the baseline methods in this case is related to the prediction algorithm. The prediction algorithm produces a predicted score for all candidate items within a given range of 1–5.

Thus, there is a possibility that many items will have the same predicted score rating. Consequently, we do not know which (if either) of two items that have the same prediction score is actually more preferred by the user. This may lead to incorrect rankings and, in turn, low accuracy. However, the proposed method based on TOPSIS successfully minimizes the negative effect of the prediction algorithm in evaluating and ranking the candidate items. Thus, the application of TOPSIS significantly improves the accuracy of memory-based CF and produces more accurate results than the baseline methods.

## 5.5 Chapter Summary

In this chapter, the experiments were conducted on proposed technique (CF-NSMA) and traditional memory-based CF methods (CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, and CF-NHSM) to show the preceding of the new technique. Firstly, the results of the new similarity method CF-BSF were compared to the results of the traditional memory-based CF methods to demonstrate the importance of improvements made by CF-BSF. Secondly, as discussed in Section 5.4, the performance accuracy of proposed technique CF-NSMA has a notable improvement when compared to performance accuracy of traditional memory-based CF methods. Thirdly, the prediction method in traditional CF methods was replaced by the MADM method to show the positive effect of MADAM method on performance accuracy. The given results were presented and compared to prove the significant enhancement made by the MADM method. All experiments were conducted on 100K & 1M MovieLens public datasets using holdout and cross-validation splitting methods. The comparison was regarding prediction accuracy and performance accuracy.

# CHAPTER 6

# CONCLUSION

## 6.1 Introduction

Although considerable work has been done in developing methods for traditional memory-based CF, however, there has been a limitation. Typically, the recommendations of traditional memory-based CF system are determined fundamentally depends on the ratings provided by its users on items. Due to the expanding of the number of items and users on the Internet and the majority of those users do not give enough ratings for the items that make the user-item rating matrix very large and sparse. Therefore, finding the relationships among users or items is very difficult or may lead to locating unsuccessful neighbours and in turn to a weak recommendation. Thus, the right recommendations depend on the successful locating neighbours. This process is the core processes of memory-based CF. For the current similarity measures in the traditional memory-based CF, identifying the effective neighbours, especially when the number of ratings provided by the user is very small, still an issue. On the other hand, the prediction method in traditional memory-based CF stills an open area for improvement to better evaluate and rank for candidate items. To deal with these issues, this research was established to propose a new technique consisting of: 1- Re-representing the preferences of users through building a new matrix to overcome the sparsity issue of the user-item matrix; 2- Formulating new similarity measurement adopted the fairness and the proportion of common rating factors to find the accurate neighbours; 3- Adopting the Multi-Attributes Decision-Making (MADM) method to get a better-ranked list of candidate items that made a significant improvement in the accuracy of recommendation.

This chapter partitioned into three sections. The summarisation of thesis will be presented in section 6.1. Next, the contributions of this work will be pointed in Section

6.2. Finally, the directions for future work will be outlined at the end of this chapter in Section 6.3.

## 6.2 Summary of Thesis

The need for recommender systems has increased with the massive amount of information available online. In this thesis, the new technique was proposed and developed to improve the recommendations of traditional memory-based CF in term of accuracy. This work implemented according to essential research methodology phases which were stated in Chapter 3. Table 6.1 presents the chapters' summary of this thesis.

Table 6.1 Chapters summary

| Chapter No | Summary |
|---|---|
| Chapter 1 | This chapter introduced an introduction and a briefed background about recommender system. Moreover, the problem statement, objectives, contributions, scope of this work have been presented in this chapter. |
| Chapter 2 | Reviewed the recommender system and answered the questions what? Why? And where? In what, the recommender system is a computer-based system that generates a set of items/ services/information as a recommendation through suggestion which items will be most interesting and valuable information for users. In why, to help users to find, the most interesting, the valuable information, or most preferred. In term of where, it can use in everywhere! (Well almost!), for what product to buy (Amazon), what subject to register (smart adviser system), and so on. In how, through collecting & analysing a large amount of information on users' behaviours, activities or preferences and seeking to predict what users will they like in the future depending on the correlation of their current preferences with others' preferences.<br>Additionally, the recommender system approaches have been discussed where the memory-based CF has been explained in more detail. Next, the memory-based CF related work has been discussed to demonstrate its limitations that summarised the work findings. |
| Chapter 3 | The fundamental and essential phases (planning, designing & implementation, and evaluation phase) that were followed by the researcher have been ordered in an acceptable sequence. They helped the researcher to design the structure and adopt the methodology to achieve the objectives of this thesis. |
| Chapter 4 | The main components of the proposed technique have been described in details. These components have been summarised in three main phases as follow:<br>Transforming the user-item matrix to normalized user-type matrix to overcome the sparsity issue.<br>Formulating the new similarity measure through adopting the fairness and proportion of common rating factors to locate the accurate neighbours.<br>Adopting MADM method instead of the prediction method to get a better-ranked list of items. |

Table 6.1 continued.

| Chapter No | Summary |
|---|---|
| Chapter 5 | This chapter presented and compared the results of the proposed CF-NSMA technique with existing widely used memory-based CF methods in term of accuracy to show how well the CF-NSMA works. Specified metrics are used in the accuracy evaluation process. The final result has shown the notable improvement that made by the proposed technique. |

To conclude, the objectives of this work have been successfully achieved. For the first objective it was to identify the existing memory-based CF methods related to improving the accuracy and addressing the issue of data sparsity. Chapter 2 discussed the related work and presented the limitations in that work. These limitations related to two main issues that have a definite impact on recommendation quality: data sparsity and low accuracy.

The second objective of this research it was to propose a developed memory-based CF technique to address the issue of data sparsity and improve the recommendation accuracy. This objective has been achieved as explained in Chapter 4. The proposed technique passed through three primary mechanisms. Firstly, utilising the kinds of items and the user-item matrix to represent the global preferences of users. These preferences are used for calculating the correlation among users to find the most similar. Secondly, formulating the new similarity measure to calculate the correlation. This similarity measure considered the fairness and proportions of common items factors to locate the successful neighbours. Finally, applying MADM method instead of prediction method to better items' ranking that led to a significant improvement in the accuracy of recommendation. Moreover, the proposed method based on new similarity and TOPSIS successfully minimizes the negative effect of the prediction algorithm in evaluating and ranking the candidate items. Thus, the application of TOPSIS significantly improves the accuracy of memory-based CF and produces more accurate results than the baseline methods.

For the third objective it was to evaluate the proposed technique, achieved in Chapter 5. Several experiments have been conducted on benchmark datasets (100K & 1M Movie datasets). Specified evaluation metrics, most common metrics are used in accuracy evaluation process of the CF, are used in the evaluation process to measure the accuracy of the developed technique. Finally to cap this summary Table 6.2 shows the objectives achievement.

Table 6.2 Objectives achievement

| Objective | Description | Achievement |
|---|---|---|
| 1 | To identify the traditional Memory-Based CF methods related to sparsity issue and recommendation accuracy. | Chapter 2 & 3 |
| 2 | To propose a new memory-based CF (CF-NSMA) technique throughout re-representing the users' preferences, formulating a new similarity measure, and adapting the MADM method. | Chapter 4 |
| 3 | To evaluate the proposed memory-based CF technique using evaluation metrics in term of accuracy. | Chapter 5 |

## 6.3 Contributions

This work can contribute to the body of knowledge in the field of recommender system by sharing it in the knowledge bases. Moreover, the primary research contribution in the process relates to the development of memory-based CF consideration into each phase. The proposed technique is designed and developed to improve the recommendation accuracy and address the sparsity issue of memory-based CF. Therefore, this work aims to introduce a proposed memory-based CF technique. Three essential phases are included in the process of development. First, constructing the normalized user-type matrix that represents the global preferences of users. Second, formulating the new similarity measure to locate the accurate neighbours. Third, applying MADM method instead of prediction method to better ranking of candidate items. These essential contributions could be clarified next. Also, this work can contribute to the body of knowledge in the field by sharing it in the knowledge bases.

Firstly, the problem of data sparsity will be solved by constructing a new data matrix based on the user-item matrix. The new matrix called normalized user-type preference matrix. The normalized matrix represented the global preferences of users. A The normalized user-type matrix distinguishes from the existing one by its' low dimensions and unparsed. In this matrix, the researcher assumes that the $\overrightarrow{T_x}$ is a vector represents the category information of user *x*, and *G* the number of categories of items in the dataset. Subsequently, the $t_{x,g}$ value represents the rating counts of user *i* on type *g*, which is represented in user-rating time matrix T. Next, normalizing the T matrix to produce the normalized user-type matrix W. The value $t_{x,g}$ will be normalized to $w_{x,g}$ value using a linear-scale to transform the ratings counts value into a ratio value between zero and one. The normalized value $w_{i,g}$ is the percentage preference of user *x* on category *g*.

Secondly, a new similarity measure will be developed to improve the accuracy of the memory-based CF recommender system. In this measure, two main factors will be adopted. First, the proportion of the number of items that rated by the target user to the number of items that taken by both users will be considered to ensure the fairness when calculating the similarity between users. The similarity between users will be increased as the number of ratings for each of them is close and vice versa. Second, the proportion of co-rated items is considered in the new similarity measure to devalue the similarity when hen the number of common not more enough (less than a threshold value). This threshold is used to determine the size of co-rated.

Finally, to get better evaluating and ranking for candidate items, this research replaced the prediction method in the traditional CF methods by the MADM method. The MADM method is a useful technique for ranking and selection of some externally determined alternatives through distance measures. Therefore, in this research, the MADM method is used to rank the items which already rated by the neighbours and not yet rated by the active user. The strategy of this technique is: collecting the ratings of items that have been voted by close users to represent decision matrix, where the neighbours and their items represented the criteria that not rated by target user represented the alternatives (candidate items). Then, fill the decision matrix by the ratings of candidate items concerning each user. The corresponding similarity value is used to define the weight of criteria. The result will be a sorted list of items. This list enables the proposed technique to take top *N* items as a set of recommendations.

To conclude, the research presented in this thesis is intended to provide a newly proposed technique that assists users to find what they need. This will confirm that recommendations that would be provided will be more accurate compared to existing methods.

## 6.4 Future Work

Since the resources and other constraints are limited, therefore this work cannot be optimal without limitations. However, there are still several opportunities for extending the scope of this work. The extending directions can be taken place in the proposed technique in the future as follows:

i. The proposed CF-NSMA technique based on MADM method needs to create an *m**k* matrix for each target user, where *m* is the number of candidate items and *k* is the number of neighbours. Building this matrix for each user considered as extra time spent on computing compare traditional memory-based CF. As future work, dimension reduction techniques such as the singular value decomposition (SVD) can be employed to solve this weakness.

ii. Finding the relationship between a pair of users do not utilise the actual rating provided by both of them. Therefore, the difference between the rating of both users is not considered. This leads to obtaining a high correlation between users in some cases. How to develop the similarity measure to take into account this factor using the standard variance or the mean rating of the pair of users to decrease the similarity weight between users who have a different level of evaluation? A question needs further research to be answered.

iii. Some users rate items randomly. Thus, there will be the possibility of building incorrect users' preferences. This will lead to unsuccessful neighbours and in turn, weakens the recommendations. User reviews have to be taken into consideration in revealing preferences of the users. These identified preferences will be used as main input in the process of computing similarity among users in order to locate the successful neighbours and induce to obtain optimal recommendations set.

## REFERENCES


Adomavicius, G., & Kwon, Y. (2007). New recommendation techniques for multicriteria rating systems. *Intelligent Systems, IEEE, 22*(3), 48-55. doi: 10.1109/MIS.2007.58.

Adomavicius, G., Manouselis, N., & Kwon, Y. (2011). Multi-criteria recommender systems. *Recommender Systems Handbook* (pp. 769-803): Springer, Boston, MA. doi: https://doi.org/10.1007/978-1-4899-7637-6_25.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734-749. doi:10.1109/Tkde.2005.99

Aggarwal, C. C. (2016). Content-based recommender systems. *Recommender Systems* (pp. 139-166): Springer, Cham. doi: https://doi.org/10.1007/978-3-319-29659-3_4.

Ahn, H. J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences, 178*(1), 37-51. doi: https://doi.org/10.1016/j.ins.2007.07.024.

Amazon. (2017), Retrieved from https://www.amazon.com/

Arekar, T., Sonar, M. R., & Uke, N. (2015). A survey on recommendation system. *International Journal of Innovative Research in Advanced Engineering, 2*(1), 1-5.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval.* (Vol. 463): ACM New York Press.

Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM, 40*(3), 66-72. doi: 10.1145/245108.245124

Basu, C., Hirsh, H., & Cohen, W. (1998). *Recommendation as classification: Using social and content-based information in recommendation.* Paper presented at the 10th Conference on Innovative Applications of Artificial Intelligence (pp. 714-720).

Bilge, A., & Yargıç, A. (2017). Improving accuracy of multi-criteria collaborative filtering by normalizing user ratings. *Nadolu Üniversitesi Bilim Ve Teknoloji Dergisi A-Uygulamalı Bilimler ve Mühendislik, 18*(1), 225-237. doi: 10.18038/aubtda.273802.

Billsus, D., & Pazzani, M. J. (1998). *Learning collaborative information filters.* Paper presented at the 5th International Conference on Machine Learning, vol. 98, pp. 46-54.

Bobadilla, J., Hernando, A., Ortega, F., & Bernal, J. (2011). A framework for collaborative filtering recommender systems. *Expert Systems with Applications, 38*(12), 14609-14623. doi:10.1016/j.eswa.2011.05.021

Bobadilla, J., Hernando, A., Ortega, F., & Gutierrez, A. (2012). Collaborative filtering based on significances. *Information Sciences, 185*(1), 1-17. doi:10.1016/j.ins.2011.09.014

Bobadilla, J., Ortega, F., & Hernando, A. (2012). A collaborative filtering similarity measure based on singularities. *Information Processing & Management, 48*(2), 204-217. doi:10.1016/j.ipm.2011.03.007

Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012). A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems, 26*, 225-238. doi:10.1016/j.knosys.2011.07.021

Bobadilla, J., Ortega, F., Hernando, A., & Gutierrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems, 46*, 109-132. doi:10.1016/j.knosys.2013.03.012

Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge-Based Systems, 23*(6), 520-528. doi:10.1016/j.knosys.2010.03.009

Bray, J. R., & Curtis, J. T. (1957). An ordination of the upland forest communities of southern Wisconsin. *Ecological Monographs, 27*(4), 325-349. doi: https://doi.org/10.2307/1942268.

Breese, J. S., Heckerman, D., & Kadie, C. (1998). *Empirical analysis of predictive algorithms for collaborative filtering.* Paper presented at the 14th Conference on Uncertainty in Artificial Intelligence. pp. 43-52.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction, 12*(4), 331-370. doi: https://doi.org/10.1023/A:1021240730564.

Burke, R. (2007). Hybrid web recommender systems. *The Adaptive Web* (pp. 377-408): Springer, Berlin, Heidelberg. doi: https://doi.org/10.1007/978-3-540-72079-9_12.

Cai, G., Lv, R., Wu, H., & Hu, X. (2014). *An improved collaborative method for recommendation and rating prediction.* Paper presented at the 2014 IEEE International Conference on Data Mining Workshop (pp. 781-788). IEEE, Shenzhen, China. doi: 10.1109/ICDMW.2014.60.

Cantador, I., Bellogín, A., & Castells, P. (2008). *News@ hand: A semantic web approach to recommending news.* Paper presented at the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (pp. 279-283). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-70987-9_34.

Cantador, I., Bellogín, A., & Vallet, D. (2010). *Content-based recommendation in social tagging systems.* Paper presented at the Proceedings of the 4th ACM Conference on Recommender systems (pp. 237-240). ACM, Barcelona, Spain. doi: 10.1145/1864708.1864756.

Cao, H., Deng, J., Guo, H., He, B., & Wang, Y. (2016). *An improved recommendation algorithm based on Bhattacharyya Coefficient.* Paper presented at the IEEE International Conference on Knowledge Engineering and Applications (pp. 241-244). IEEE, Singapore, Singapore. doi: 10.1109/ICKEA.2016.7803027.

Chen, L., Chen, G. L., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction, 25*(2), 99-154. doi: https://doi.org/10.1007/s11257-015-9155-5.

Cheng, Q., Wang, X., Yin, D., Niu, Y., Xiang, X., Yang, J., & Shen, L. (2015). *The new similarity measure based on user preference models for collaborative filtering.* Paper presented at the IEEE International Conference on Information and Automation (pp. 577-582). IEEE, Lijiang, China. doi:10.1109/ICInfA.2015.7279353.

Choi, K., & Suh, Y. (2013). A new similarity function for selecting neighbors for each target item in collaborative filtering. *Knowledge-Based Systems, 37*, 146-153. doi:10.1016/j.knosys.2012.07.019

Chowdhury, G. G. (2010). *Introduction to modern information retrieval*. Facet publishing.

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

Das, D., Sahoo, L., & Datta, S. (2017). A Survey on Recommendation System. *International Journal of Computer Applications, 160*(7), 6-10.

EachMovie Collaborative Filtering Data Set. (2017), Retrieved from http://www.cs.cmu.edu/~lebanon/IR-lab/data.html.

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction, 4*(2), 81-173. doi: http://dx.doi.org/10.1561/1100000009.

El Alami, Y. E. M., Nfaoui, E. H., & El Beqqali, O. (2015). *Improving Neighborhood-Based Collaborative Filtering by a Heuristic Approach and an Adjusted Similarity Measure*. Paper presented at the International Conference on Big Data, Cloud and Applications, Tetuan, Morocco.

Feng, J., Fengs, X., Zhang, N., & Peng, J. (2018). An improved collaborative filtering method based on similarity. *PloS One, 13*(9), e0204003. doi:https://doi.org/10.1371/journal.pone.0204003.

Ghazarian, S., & Nematbakhsh, M. A. (2015). Enhancing memory-based collaborative filtering for group recommender systems. *Expert Systems with Applications, 42*(7), 3801-3812. doi:10.1016/j.eswa.2014.11.042

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM, 35*(12), 61-70. doi: 10.1145/138859.138867.

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval, 4*(2), 133-151. doi: 10.1023/A:1011419012209

Greiner, R., Su, X. Y., Shen, B., & Zhou, W. (2005). Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Machine Learning, 59*(3), 297-322. doi: 10.1007/s10994-005-0469-0

Gu, L., Yang, P., & Dong, Y. (2014a). *A Clustering-Based Similarity Measurement for Collaborative Filtering.* Paper presented at the 2$^{th}$ International Conference on Advanced Cloud and Big Data (pp. 282-287). IEEE, Huangshan, China. doi: 10.1109/CBD.2014.50.

Gu, L., Yang, P., & Dong, Y. (2014e). *An dynamic-weighted collaborative filtering approach to address sparsity and adaptivity issues.* Paper presented at the IEEE Congress on Evolutionary Computation (pp. 3044-3050), IEEE, Beijing, China. doi:10.1109/CEC.2014.6900403.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Herlocker, J., Konstan, J. A., & Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval, 5*(4), 287-310. doi: https://doi.org/10.1023/A:1020443909834.

Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999). *An algorithmic framework for performing collaborative filtering.* Paper presented at the 22$^{nd}$ Annual International ACM

SIGIR Conference on Research and Development in Information Retrieval (pp. 230-237). ACM, California, USA. doi:10.1145/312624.312682.

Herlocker, J. L., Konstan, J. A., Terveen, K., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, 22*(1), 5-53. doi: 10.1145/963770.963772

Hill, W., Stead, L., Rosenstein, M., & Furnas, G. (1995). *Recommending and evaluating choices in a virtual community of use.* Paper presented at the SIGCHI conference on Human Factors in Computing Systems (pp. 194-201). ACM, Denver, Colorado, USA. doi:10.1145/223904.223929.

Hofmann, K., Schuth, A., Bellogin, A., & De Rijke, M. (2014). *Effects of position bias on click-based recommender evaluation*. Paper presented at the European Conference on Information Retrieval(pp. 624-630). Springer, Cham. doi: https://doi.org/10.1007/978-3-319-06028-6_67.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems, 22*(1), 89-115. doi:10.1145/963770.963774

Hu, Y., Koren, Y., & Volinsky, C. (2008). *Collaborative filtering for implicit feedback datasets.* Paper presented at the 8th IEEE International Conference on Data Mining (pp. 263-272). IEEE, Pisa, Italy. doi:10.1109/icdm.2008.22.

Hu, Y., Shi, W. S., Li, H., & Hu, X. H. (2017). Mitigating Data Sparsity Using Similarity Reinforcement-Enhanced Collaborative Filtering. *ACM Transactions on Internet Technology, 17*(3), 31. doi:10.1145/3062179

Huang, B.-H., & Dai, B.-R. (2015). *A weighted distance similarity model to improve the accuracy of collaborative recommender system.* Paper presented at the 8th IEEE International Conference on Mobile Data Management (pp. 104-109). IEEE, Pisa, Italy. doi:10.1109/MDM.2015.43.

Hwang, C.-L., & Yoon, K. (2012). Multiple attribute decision making: methods and applications a state-of-the-art survey. Springer Science & Business Media.

Ic, Y. T., & Yurdakul, M. (2010). Development of a quick credibility scoring decision support system using fuzzy TOPSIS. *Expert Systems with Applications, 37*(1), 567-574. doi:10.1016/j.eswa.2009.05.038

Ishizaka, A., & Nemery, P. (2013). *Multi-criteria decision analysis: methods and software*. John Wiley & Sons.

Jain, S., Khangarot, H., & Singh, S. (2019). *Journal recommendation system using content-based filtering*. Paper presented at the Recent Developments in Machine Learning and Data Analytics (pp. 99-108). Springer, Singapore. doi: https://doi.org/10.1007/978-981-13-1280-9_9.

Jamali, M., & Ester, M. (2009). *Trustwalker: a random walk model for combining trust-based and item-based recommendation.* Paper presented at the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 397-406). ACM, Paris, France. doi:10.1145/1557019.1557067.

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

Jawaheer, G., Szomszor, M., & Kostkova, P. (2010). *Characterisation of explicit feedback in an online music recommendation service.* Paper presented at the 4th ACM conference on Recommender Systems (pp. 317-320). ACM, Barcelona, Spain. doi:10.1145/1864708.1864776.

Kahraman, C. (2008). Multi-criteria decision making methods and fuzzy sets *Fuzzy Multi-Criteria Decision Making.* Paper presented at the Fuzzy Multi-Criteria Decision Making (pp. 1-18). Springer, Boston, MA. doi: https://doi.org/10.1007/978-0-387-76813-7_1.

Kelly, D., & Teevan, J. (2003). *Implicit feedback for inferring user preference: a bibliography.* Paper presented at the ACM SIGIR Forum (pp. 18-28). ACM, New York, USA. doi:10.1145/959258.959260.

Kg, S., & Sadasivam, G. S. (2017). Modified heuristic similarity measure for personalization using collaborative Filtering Technique. *Appl. Math, 11*(1), 317-325. doi:10.18576/amis/110137

Kim, B. M., Li, Q., Park, C. S., Kim, S. G., & Kim, J. Y. (2006). A new approach for combining content-based and collaborative filters. *Journal of Intelligent Information Systems, 27*(1), 79-91. doi:10.1007/s10844-006-8771-2

Kluver, D., & Konstan, J. A. (2014). *Evaluating recommender behavior for new users.* Paper presented at the 8th ACM Conference on Recommender Systems (pp. 121-128). ACM, California, USA. doi:10.1145/2645710.2645742.

Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., & Newell, C. (2012). Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction, 22*(4-5), 441-504. doi:10.1007/s11257-011-9118-4

Kokare, M., Chatterji, B., & Biswas, P. (2003). *Comparison of similarity metrics for texture image retrieval.* Paper presented at the TENCON 2003 Conference on Convergent Technologies for the Asia-Pacific Region (pp. 571-575). IEEE, Bangalore, India. doi: 10.1109/TENCON.2003.1273228.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM, 40*(3), 77-87. doi:10.1145/245108.245126

Konstan, J. A., Riedl, J., Borchers, A., & Herlocker, J. L. (1998). *Recommender systems: A grouplens perspective.* Paper presented at the Recommender Systems: Papers from the 1998 Workshop (pp. 60-64). American Association for Artificial Intelligence Press.

Koohi, H., & Kiani, K. (2017). A new method to find neighbor users that improves the performance of Collaborative Filtering. *Expert Systems with Applications*, *83*, 30-39. doi:10.1016/j.eswa.2017.04.027

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30-37. doi:10.1109/Mc.2009.263

Koutrika, G., Bercovitz, B., & Garcia-Molina, H. (2009). *FlexRecs: expressing and combining flexible recommendations.* Paper presented the 2009 ACM SIGMOD International Conference on Management of data (pp. 745-758). ACM, Providence, Rhode Island, USA. doi:10.1145/1559845.1559923.

Lang, K. (1995). *Newsweeder: Learning to filter netnews.* Paper presented at the 12th International Conference on Machine Learning (pp. 331-339). Tahoe City, California. doi:10.1016/B978-1-55860-377-6.50048-7.

Lee, T. Q., Park, Y., & Park, Y. T. (2008). A time-based approach to effective recommender systems using implicit feedback. *Expert Systems with Applications, 34*(4), 3055-3062. doi:10.1016/j.eswa.2007.06.031

Li, W., Xu, H., Ji, M., Xu, Z., & Fang, H. (2016). *A hierarchy weighting similarity measure to improve user-based collaborative filtering algorithm.* Paper presented at the 2nd IEEE International Conference on Computer and Communications (pp. 843-846). IEEE, Chengdu, China. doi: 10.1109/CompComm.2016.7924821.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendation - Item-to-item collaborative filtering. *IEEE Internet Computing, 7*(1), 76-80. doi:10.1109/Mic.2003.1167344

Liu, H. F., Hu, Z., Mian, A., Tian, H., & Zhu, X. Z. (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems, 56*, 156-166. doi:10.1016/j.knosys.2013.11.006

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. *Recommender Systems Handbook* (pp. 73-105): Springer, Boston, MA. doi: https://doi.org/10.1007/978-0-387-85820-3_3.

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., & Zhou, T. (2012). Recommender systems. *Physics Reports, 519*(1), 1-49. doi:10.1016/j.physrep.2012.02.006

Ma, H., King, I., & Lyu, M. R. (2007). *Effective missing data prediction for collaborative filtering.* Paper presented at the roceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 39-46). ACM, Amsterdam, Netherlands. doi:10.1145/1277741.1277751.

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

Manouselis, N., & Costopoulou, C. (2007). Analysis and classification of multi-criteria recommender systems. *World Wide Web-Internet and Web Information Systems, 10*(4), 415-441. doi:10.1007/s11280-007-0019-8

Mao, J., Cui, Z., Zhao, P., & Li, X. (2013). *An improved similarity measure method in Collaborative Filtering Recommendation Algorithm.* Paper presented at the International Conference on Cloud Computing and Big Data (pp. 297-303). IEEE, Fuzhou, China. doi:10.1109/cloudcom-asia.2013.39.

Melville, P., Mooney, R. J., & Nagarajan, R. (2002). *Content-boosted collaborative filtering for improved recommendations.* Paper presented at the Innovative Applications of Artificial Intelligence Conference (pp. 187-192). American Association for Artificial Intelligence Press.

Montaner Rigall, M. (2003). Collaborative recommender agents based on case-based reasoning and trust. Universitat de Girona.

Mooney, R. J., & Roy, L. (2000). *Content-based book recommending using learning for text categorization.* Paper presented at the 5th ACM Conference on Digital libraries (pp. 195-204). ACM, Texas, USA. doi:10.1145/336597.336662.

Nagarnaik, P., & Thomas, A. (2015). *Survey on recommendation system methods.* Paper presented at the 2nd International Conference on Electronics and Communication Systems (pp. 1603-1608). IEEE, Coimbatore, India. doi: 10.1109/ECS.2015.7124857.

Najafabadi, M. K., Mahrin, M. N., Chuprat, S., & Sarkan, H. M. (2017). Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. *Computers in Human Behavior, 67*, 113-128. doi:10.1016/j.chb.2016.11.010

Nguyen, V. D., Sriboonchitta, S., & Huynh, V. N. (2017). Using community preference for overcoming sparsity and cold-start problems in collaborative filtering system offering soft ratings. *Electronic Commerce Research and Applications, 26*, 101-108. doi:10.1016/j.elerap.2017.10.002

Olson, D. L. (2004). Comparison of weights in TOPSIS models. *Mathematical and Computer Modelling, 40*(7-8), 721-727. doi:10.1016/j.mcm.2004.10.003

Ostuni, V. C., Di Noia, T., Di Sciascio, E., & Mirizzi, R. (2013). *Top-N recommendations from implicit feedback leveraging linked open data.* Paper presented at the 7th ACM Conference on Recommender Systems (pp. 85-92). ACM, Hong Kong, China. doi:10.1145/2507157.2507172.

Papagelis, M., Plexousakis, D., & Kutsuras, T. (2005). *Alleviating the sparsity problem of collaborative filtering using trust inferences*. Paper presented at the Trust Management (pp. 224-239). Springer, Berlin, Heidelberg. doi: https://doi.org/10.1007/11429760_16.

Parkan, C., & Wu, M. (1997). On the equivalence of operational performance measurement and multiple attribute decision making. *International Journal of Production Research, 35*(11), 2963-2988. doi: https://doi.org/10.1080/002075497194246.

Passi, R., Jain, S., & Singh, P. K. (2019). *Hybrid Approach for Recommendation System.* Paper presented at the 2nd International Conference on Data Engineering and Communication Technology (pp. 117-128). Springer, Singapore.

Patra, B. K., Launonen, R., Ollikainen, V., & Nandi, S. (2015). A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. *Knowledge-Based Systems, 82*, 163-177. doi:10.1016/j.knosys.2015.03.001

Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: the identification of interesting web sites. *Machine Learning, 27*(3), 313-331. doi:10.1023/A:1007369909943

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. *The Adaptive Web* (pp. 325-341): Springer. Berlin, Heidelberg. doi: https://doi.org/10.1007/978-3-540-72079-9_10.

Pirasteh, P., Hwang, D., & Jung, J. E. (2015). Weighted similarity schemes for high scalability in user-based collaborative filtering. *Mobile Networks & Applications, 20*(4), 497-507. doi:10.1007/s11036-014-0544-5

Polatidis, N., & Georgiadis, C. K. (2016). A multi-level collaborative filtering method that improves recommendations. *Expert Systems with Applications, 48*, 100-110. doi:10.1016/j.eswa.2015.11.023

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). *GroupLens: an open architecture for collaborative filtering of netnews.* Paper presented at the 1994 ACM

Conference on Computer Supported Cooperative Work (pp. 175-186). ACM, North Carolina, USA. doi:10.1145/192844.192905.

Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM, 40*(3), 56-58. doi:10.1145/245108.245121.

Revankar, O. S., & Haribhakta, Y. (2015). Survey on collaborative filtering technique in recommendation system. *International Journal of Application or Innovation in Engineering & Management, 4*(3), 85-91.

Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to recommender systems handbook*. Springer.

Salton, G., & McGill, M. J. (1986). Introduction to modern information retrieval. McGraw-Hill.

Saranya, K., Sadasivam, G. S., & Chandralekha, M. (2016). Performance comparison of different similarity measures for collaborative filtering technique. *Indian Journal of Science and Technology, 9*(29), 1-8. doi: 10.17485/ijst/2016/v9i29/91060.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). *Item-based collaborative filtering recommendation algorithms*. Paper presented at the 10th International Conference on World Wide Web (pp. 285-295). ACM, Hong Kong. doi: 10.1145/371920.372071.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2002). *Incremental singular value decomposition algorithms for highly scalable recommender systems*. Paper presented at the 5th International Conference on Computer and Information Science (pp. 27-28). Citeseer. doi:10.1.1.3.7894.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems *The Adaptive Web* (pp. 291-324): Springer, Berlin, Heidelberg. doi: https://doi.org/10.1007/978-3-540-72079-9_9.

Schafer, J. B., Konstan, J., & Riedl, J. (1999). *Recommender systems in e-commerce*. Paper presented at the 1st ACM Conference on Electronic commerce (pp. 158-166). ACM, Colorado, USA. doi:10.1145/336992.337035.

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). *Methods and metrics for cold-start recommendations*. Paper presented at the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 253-260). ACM, Tampere, Finland. doi:10.1145/564376.564421.

Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. *Recommender systems handbook* (pp. 257-297): Springer, Boston, MA. doi: https://doi.org/10.1007/978-0-387-85820-3_8.

Shardanand, U., & Maes, P. (1995). *Social information filtering: algorithms for automating "word of mouth"*. Paper presented at the SIGCHI Conference on Human Factors in Computing Systems (pp. 210-217). ACM Press/Addison-Wesley Publishing Co, Colorado, USA. doi:10.1145/223904.223931.

Sharifi, Z., Rezghi, M., & Nasiri, M. (2014). *A new algorithm for solving data sparsity problem based-on Non negative matrix factorization in recommender systems*. Paper presented at the 4th International Conference on Computer and Knowledge Engineering (pp. 56-61). IEEE, Mashhad, Iran. doi:10.1109/iccke.2014.6993356.

Sharma, L., & Gera, A. (2013). A survey of recommendation system: research challenges. *International Journal of Engineering Trends and Technology, 4*(5), 1989-1992.

Shih, H. S., Shyur, H. J., & Lee, E. S. (2007). An extension of TOPSIS for group decision making. *Mathematical and Computer Modelling, 45*(7-8), 801-813. doi:10.1016/j.mcm.2006.03.023

Shunpan, L., Lin, M., & Fuyongyuan. (2015). A singularity-based user similarity measure for recommender systems. *International journal of innovative computing information and control*, *11*(5), 1629-1638.

Sikka, R., Dhankhar, A., & Rana, C. (2012). A survey paper on e-learning recommender system. *International Journal of Computer Applications*, *47*(9), 27-30. doi:10.1.1.258.9486

Sivapalan, S., Sadeghian, A., Rahnama, H., & Madni, A. M. (2014). *Recommender systems in e-commerce*. Paper presented at the World Automation Congress (pp. 179-184). IEEE, Waikoloa, USA. doi:10.1109/WAC.2014.6935763.

SONG, W.-l. (2018). *News recommendation system based on collaborative filtering and SVM*. aper presented at the 3rd International Conference on Automation, Mechanical and Electrical Engineering (pp. 569-574). doi: 10.12783/dtetr/amee2018/25386.

Su, X., & Khoshgoftaar, T. M. (2006). *Collaborative filtering for multi-class data using belief nets algorithms*. Paper presented at the 18th IEEE International Conference on Tools with Artificial Intelligence (pp. 497-504). IEEE. Arlington, USA. doi:10.1109/ICTAI.2006.41.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence, 2009,* 1-19. doi: 10.1155/2009/421425

Suryakant, & Mahara, T. (2016). A New similarity measure based on mean measure of divergence for collaborative filtering in sparse environment. *Procedia Computer Science, 89,* 450-456. doi:10.1016/j.procs.2016.06.099.

Teknomo, K. (2017). Bray Curtis distance.Retrieved from http://people.revoledu.com/kardi/tutorial/Similarity/BrayCurtisDistance.html.

Ungar, L. H., & Foster, D. P. (1998). *Clustering methods for collaborative filtering*. Paper presented at the Association for the Advancement of Artificial Intelligence Workshop on Recommendation Systems (pp. 114-129). Association for the Advancement of Artificial Intelligence Press.

Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). *Deep content-based music recommendation*. Paper presented at the Advances in Neural Information Processing Systems (pp. 2643-2651).

Van Meteren, R., & Van Someren, M. (2000). *Using content-based filtering for recommendation*. Paper presented at the Machine Learning in the New Information Age Workshop (pp. 47-56).

Velasquez, M., & Hester, P. T. (2013). An analysis of multi-criteria decision making methods. *International Journal of Operations Research, 10*(2), 56-66.

Viriyavisuthisakul, S., Sanguansat, P., Charnkeitkong, P., & Haruechaiyasak, C. (2015). *A comparison of similarity measures for online social media Thai text classification*. Paper presented at the 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (pp. 1-6). IEEE, Hua Hin, Thailand. doi:10.1109/ECTICon.2015.7207106.

Vozalis, E., & Margaritis, K. G. (2003). *Analysis of recommender systems algorithms.* Paper presented at the 6th Hellenic European Conference on Computer Mathematics & its Applications (pp. 732-745).

Wang, P., Qian, Q., Shang, Z., & Li, J. (2016). *An recommendation algorithm based on weighted slope one algorithm and user-based collaborative filtering.* Paper presented at the Control and Decision Conference (pp. 2431-2434). IEEE, Yinchuan, China.doi: 10.1109/ccdc.2016.7531393.

Wu, Y., & Zheng, J. (2010). *A collaborative filtering recommendation algorithm based on improved similarity measure method.* Paper presented at the IEEE International Conference on Progress in Informatics and Computing (pp. 246-249). IEEE, Shanghai, China. doi: 10.1109/PIC.2010.5687455.

Xu, L., & Yang, J.-B. (2001). Introduction to multi-criteria decision making and the evidential reasoning approach. Manchester School of Management (pp. 1-21).

Yang, W. S., & Hwang, S. Y. (2013). iTravel: A recommender system in mobile peer-to-peer environment. *Journal of Systems and Software, 86*(1), 12-20. doi:10.1016/j.jss.2012.06.041

Yang, X. W., Guo, Y., Liu, Y., & Steck, H. (2014). A survey of collaborative filtering based social recommender systems. *Computer Communications, 41*, 1-10. doi:10.1016/j.comcom.2013.06.009

Yoon, K., & Hwang, C. (1981). TOPSIS (technique for order preference by similarity to ideal solution)–a multiple attribute decision making, w: Multiple attribute decision making–methods and applications, a state-of-the-at survey (pp. 128-140). Springer Verlag, Berlin.

Yu, K., Schwaighofer, A., Tresp, V., Xu, X. W., & Kriegel, H. P. (2004). Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering, 16*(1), 56-69. doi: 10.1109/Tkde.2004.1264822

Zang, X., Liu, T., Qiao, S., Gao, W., Wang, J., Sun, X., & Zhang, B. (2016). *A New Weighted Similarity Method Based on Neighborhood User Contributions for Collaborative Filtering.* Paper presented at the IEEE 1st International Conference on Data Science in Cyberspace (pp. 376-381). IEEE. Changsha, China. doi: 10.1109/DSC.2016.23.

Zhanchun, G., & Yuying, L. (2012). *Improving the collaborative filtering recommender system by using gpu.* Paper presented at the IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge (pp. 330-333). IEEE, Sanya, China. doi:10.1109/CyberC.2012.62.

Zhang, B., & Yuan, B. (2017). *Improved collaborative filtering recommendation algorithm of similarity measure.* Paper presented at the American Institute of Physics Conference Series Conference Proceedings (pp. 1-6). AIP Publishing. doi:10.1063/1.4982532.

Zhang, J., Lin, Y. J., Lin, M. L., & Liu, J. H. (2016). An effective collaborative filtering algorithm based on user preference clustering. *Applied Intelligence, 45*(2), 230-240. doi:10.1007/s10489-015-0756-9

Zhang, R., Liu, Q.-d., Gui, C., Wei, J.-X., & Ma, H. (2014). *Collaborative filtering for recommender systems*. Paper presented at the 2th International Conference on Advanced Cloud and Big Data (pp. 301-308). IEEE, Huangshan, China. doi: 10.1109/cbd.2014.47.

## APPENDIX A
## SIMILARITY MEASURE DERIVATIONS

This appendix illustrates how to our similarity measure formulated. The MovieLens 100K dataset is used. Holdout method also is used to partition data into training 80 % and testing 20% sets.

Firstly, need to explain Users Rating Preferences factor. Different users have different rating preferences. Some users prefer giving high ratings. Some users tend to rate low value. To reflect this behaviour preference, the mean and variance of the rating added to similarity method to test the effect of this factor. This factor can be defined as follows:

$$\mathbf{S(x\,,y)^{URP}} = \mathbf{1} - \frac{\mathbf{1}}{\mathbf{1 + exp(-|\mu_x - \mu_y| * |\sigma_x - \sigma_y|)}}$$

Where $\sigma_x$ and $\mu_x$ is the mean rating and standard variance of user *x*, respectively. Which can be defined as follow:

$$\boldsymbol{\sigma_x} = \sqrt{\sum\nolimits_{i \in I_x} (\boldsymbol{r_{x,i}} - \overline{\boldsymbol{r_x}})^2 \,/ |\boldsymbol{I_x}|}$$

$$\boldsymbol{\mu_x} = \sqrt{\sum\nolimits_{i \in I_x} (\boldsymbol{r_{x,i}}) \,/ |\boldsymbol{I_x}|}$$

Table 1 Parameters

| Parameters | Parameters descriptions |
|---|---|
| BC | Bray-Curtis distance measurement |
| Ff | Fairness factor |
| Sig | Sigmoid function |
| URP | Users rating Preferences |
| Jaccard | Jaccard similarity coefficient |
| MAE | Mean Absolute Error |
| K | Number of neighbours |
| M | Number of recommended items |

Table 2 Performance accuracy using BC similarity.

| | BC | | | |
|---|---|---|---|---|
| K | M | Recall | Precision | F-measure |
| 10 | 10 | 0.020978877 | 0.045917285 | 0.028799651 |
| | 20 | 0.043136816 | 0.046394486 | 0.044706385 |
| | 30 | 0.063842029 | 0.045705196 | 0.05327223 |
| | 40 | 0.079572578 | 0.043584305 | 0.056320288 |
| | 50 | 0.098023786 | 0.042502651 | 0.059295188 |
| 20 | 10 | 0.027056711 | 0.050583245 | 0.035255462 |
| | 20 | 0.044850016 | 0.046659597 | 0.045736915 |
| | 30 | 0.064625979 | 0.04641216 | 0.054025244 |
| | 40 | 0.080341873 | 0.043663839 | 0.056578597 |
| | 50 | 0.098268751 | 0.042820785 | 0.059649286 |
| 30 | 10 | 0.023947056 | 0.047613998 | 0.031866917 |
| | 20 | 0.04309867 | 0.045758218 | 0.044388643 |
| | 30 | 0.056172536 | 0.042382467 | 0.048312731 |
| | 40 | 0.075449471 | 0.042232238 | 0.05415285 |
| | 50 | 0.088260449 | 0.039851538 | 0.05491 |
| 40 | 10 | 0.025179004 | 0.048780488 | 0.033213968 |
| | 20 | 0.04131873 | 0.042205726 | 0.041757518 |
| | 30 | 0.054267588 | 0.03863556 | 0.045136439 |
| | 40 | 0.073167547 | 0.03947508 | 0.051282447 |
| | 50 | 0.087901004 | 0.038579003 | 0.053623228 |
| 50 | 10 | 0.025628337 | 0.0495228 | 0.033776921 |
| | 20 | 0.044142711 | 0.045546129 | 0.04483344 |
| | 30 | 0.059412762 | 0.043372216 | 0.050140851 |
| | 40 | 0.073539925 | 0.041171792 | 0.052789211 |
| | 50 | 0.088410428 | 0.039469777 | 0.054575138 |

Table 3 Performance prediction using BC similarity.

| K | MAE |
|---|---|
| 30 | 0.79330404 |
| 50 | 0.773170008 |
| 70 | 0.764332854 |
| 100 | 0.759249413 |
| 150 | 0.754547682 |

Table 4 Performance accuracy using BC similarity with Ff factor.

| | BC-Ff | | | |
|---|---|---|---|---|
| K | M | Recall | Precision | F-measure |
| 10 | 10 | 0.034536777 | 0.057370095 | 0.043117084 |
| | 20 | 0.058099513 | 0.050053022 | 0.05377694 |
| | 30 | 0.077537303 | 0.046306115 | 0.057983724 |
| | 40 | 0.103576915 | 0.045811241 | 0.063525612 |
| | 50 | 0.129031236 | 0.044623542 | 0.066313531 |
| 20 | 10 | 0.025323005 | 0.049734889 | 0.033559077 |
| | 20 | 0.057869458 | 0.052651113 | 0.055137091 |
| | 30 | 0.082319413 | 0.049346059 | 0.061703931 |
| | 40 | 0.099598741 | 0.046208908 | 0.063129048 |
| | 50 | 0.12986683 | 0.045683987 | 0.067591079 |
| 30 | 10 | 0.023469375 | 0.046871686 | 0.031277583 |
| | 20 | 0.050426002 | 0.046659597 | 0.048469741 |
| | 30 | 0.079735452 | 0.048179569 | 0.060065185 |
| | 40 | 0.098892912 | 0.046527041 | 0.063281475 |
| | 50 | 0.12354441 | 0.045302227 | 0.066294917 |
| 40 | 10 | 0.025376165 | 0.049204666 | 0.033483824 |
| | 20 | 0.05168659 | 0.04931071 | 0.050470705 |
| | 30 | 0.080315149 | 0.048957229 | 0.060832905 |
| | 40 | 0.098412156 | 0.047163309 | 0.063766829 |
| | 50 | 0.121540615 | 0.045641569 | 0.066362387 |
| 50 | 10 | 0.025177201 | 0.049204666 | 0.033310155 |
| | 20 | 0.052786506 | 0.049575822 | 0.051130811 |
| | 30 | 0.073907912 | 0.047437257 | 0.057785384 |
| | 40 | 0.094830195 | 0.045731707 | 0.061705863 |
| | 50 | 0.117919379 | 0.045153765 | 0.065302033 |

Table 5 Performance prediction using BC similarity with Ff factor.

| K | MAE |
|---|---|
| 30 | 0.804636111 |
| 50 | 0.788692057 |
| 70 | 0.778638667 |
| 100 | 0.77176836 |
| 150 | 0.766450767 |

Table 6 Performance accuracy using BC similarity with Sig factor.

| | BC-Sig | | | |
|---|---|---|---|---|
| K | M | Recall | Precision | F-measure |
| 10 | 10 | 0.02543 | 0.044221 | 0.032291 |
| | 20 | 0.046772 | 0.045228 | 0.045987 |
| | 30 | 0.067806 | 0.044892 | 0.05402 |
| | 40 | 0.087221 | 0.043929 | 0.05843 |
| | 50 | 0.10612 | 0.043436 | 0.061641 |
| 20 | 10 | 0.026258 | 0.045917 | 0.03341 |
| | 20 | 0.043496 | 0.043372 | 0.043434 |
| | 30 | 0.064816 | 0.044079 | 0.052473 |
| | 40 | 0.081004 | 0.042603 | 0.055839 |
| | 50 | 0.098872 | 0.042291 | 0.059242 |
| 30 | 10 | 0.028946 | 0.049947 | 0.036651 |
| | 20 | 0.045268 | 0.043743 | 0.044493 |
| | 30 | 0.062136 | 0.041463 | 0.049737 |
| | 40 | 0.075869 | 0.039899 | 0.052296 |
| | 50 | 0.094853 | 0.040679 | 0.056939 |
| 40 | 10 | 0.027781 | 0.047826 | 0.035146 |
| | 20 | 0.043621 | 0.043107 | 0.043363 |
| | 30 | 0.062663 | 0.042347 | 0.05054 |
| | 40 | 0.078131 | 0.041066 | 0.053835 |
| | 50 | 0.093545 | 0.040339 | 0.05637 |
| 50 | 10 | 0.027531 | 0.048674 | 0.035169 |
| | 20 | 0.048543 | 0.04666 | 0.047583 |
| | 30 | 0.062849 | 0.043054 | 0.051102 |
| | 40 | 0.078538 | 0.041622 | 0.05441 |
| | 50 | 0.094313 | 0.040954 | 0.05711 |

Table 7 Performance prediction using BC similarity with Sig factor.

| K | MAE |
|---|---|
| 30 | 0.774711 |
| 50 | 0.760113 |
| 70 | 0.753844 |
| 100 | 0.751903 |
| 150 | 0.750851 |

Table 8 Performance accuracy using BC similarity with URP factor.

| | BC-URP | | | |
|---|---|---|---|---|
| K | M | Recall | Precision | F-measure |
| 10 | 10 | 0.020007485 | 0.044645 | 0.027632 |
| | 20 | 0.041293239 | 0.044168 | 0.042682 |
| | 30 | 0.065775198 | 0.04514 | 0.053538 |
| | 40 | 0.079755529 | 0.043001 | 0.055876 |
| | 50 | 0.101921721 | 0.043563 | 0.061038 |
| 20 | 10 | 0.027060108 | 0.053977 | 0.036048 |
| | 20 | 0.044338628 | 0.047084 | 0.04567 |
| | 30 | 0.060956402 | 0.04369 | 0.050899 |
| | 40 | 0.078124078 | 0.042285 | 0.054871 |
| | 50 | 0.095504939 | 0.041527 | 0.057885 |
| 30 | 10 | 0.025656142 | 0.051326 | 0.034211 |
| | 20 | 0.044334544 | 0.046501 | 0.045392 |
| | 30 | 0.060416155 | 0.042913 | 0.050182 |
| | 40 | 0.074751002 | 0.040589 | 0.05261 |
| | 50 | 0.091133354 | 0.040339 | 0.055924 |
| 40 | 10 | 0.027715915 | 0.053977 | 0.036625 |
| | 20 | 0.04555145 | 0.047508 | 0.046509 |
| | 30 | 0.058345045 | 0.041569 | 0.048549 |
| | 40 | 0.073157494 | 0.039634 | 0.051414 |
| | 50 | 0.091077601 | 0.039427 | 0.055032 |
| 50 | 10 | 0.025935655 | 0.051644 | 0.03453 |
| | 20 | 0.046147119 | 0.045811 | 0.045979 |
| | 30 | 0.061457195 | 0.042453 | 0.050217 |
| | 40 | 0.074567282 | 0.040138 | 0.052185 |
| | 50 | 0.091716401 | 0.038897 | 0.054627 |

Table 9 Performance prediction using BC similarity with URP factor.

| K | MAE |
|---|---|
| 30 | 0.791746189 |
| 50 | 0.77271681 |
| 70 | 0.764723347 |
| 100 | 0.759143358 |
| 150 | 0.754217996 |

Table 10 Performance accuracy using BC similarity with Sig and Ff factors.

| | BC-Sig-Ff | | | |
|---|---|---|---|---|
| K | M | Recall | Precision | F-measure |
| 10 | 10 | 0.036689907 | 0.063521 | 0.046513 |
| | 20 | 0.064452098 | 0.055673 | 0.059742 |
| | 30 | 0.084775428 | 0.050619 | 0.063389 |
| | 40 | 0.108602376 | 0.047667 | 0.066254 |
| | 50 | 0.135008423 | 0.045493 | 0.068054 |
| 20 | 10 | 0.027742939 | 0.051962 | 0.036173 |
| | 20 | 0.059771181 | 0.053075 | 0.056225 |
| | 30 | 0.080833115 | 0.048427 | 0.060568 |
| | 40 | 0.101318341 | 0.046792 | 0.064018 |
| | 50 | 0.137633329 | 0.046893 | 0.069952 |
| 30 | 10 | 0.028674575 | 0.054613 | 0.037605 |
| | 20 | 0.057631096 | 0.05334 | 0.055403 |
| | 30 | 0.07815823 | 0.048074 | 0.059531 |
| | 40 | 0.110342569 | 0.047932 | 0.066833 |
| | 50 | 0.130086074 | 0.04526 | 0.067155 |
| 40 | 10 | 0.026980989 | 0.052174 | 0.035568 |
| | 20 | 0.051985235 | 0.048303 | 0.050077 |
| | 30 | 0.075981194 | 0.047119 | 0.058167 |
| | 40 | 0.099983449 | 0.04605 | 0.063057 |
| | 50 | 0.128690927 | 0.045366 | 0.067084 |
| 50 | 10 | 0.026130226 | 0.051856 | 0.03475 |
| | 20 | 0.052739153 | 0.048515 | 0.050539 |
| | 30 | 0.077702327 | 0.047755 | 0.059155 |
| | 40 | 0.101139348 | 0.046872 | 0.064057 |
| | 50 | 0.138791938 | 0.047253 | 0.070503 |

Table 11 Performance prediction using BC similarity with Sig and Ff factors.

| K | MAE |
|---|---|
| 30 | 0.786866325 |
| 50 | 0.771699049 |
| 70 | 0.763740368 |
| 100 | 0.7591266 |
| 150 | 0.756173642 |

Table 12 Performance accuracy using BC similarity with URL and Ff factors.

| | BC-URP-Ff | | | |
|---|---|---|---|---|
| K | M | 0.032393 | 0.05281 | 0.040155 |
| 10 | 10 | 0.058064 | 0.048621 | 0.052925 |
| | 20 | 0.07853 | 0.046165 | 0.058147 |
| | 30 | 0.103069 | 0.045599 | 0.063226 |
| | 40 | 0.129822 | 0.044496 | 0.066276 |
| | 50 | 0.03125 | 0.0579 | 0.040592 |
| 20 | 10 | 0.061658 | 0.055249 | 0.058278 |
| | 20 | 0.08016 | 0.049593 | 0.061277 |
| | 30 | 0.101098 | 0.046474 | 0.063676 |
| | 40 | 0.128784 | 0.04649 | 0.068318 |
| | 50 | 0.025201 | 0.049735 | 0.033452 |
| 30 | 10 | 0.053007 | 0.048409 | 0.050604 |
| | 20 | 0.077112 | 0.048356 | 0.059439 |
| | 30 | 0.096257 | 0.046501 | 0.062708 |
| | 40 | 0.133919 | 0.047953 | 0.070619 |
| | 50 | 0.023628 | 0.045281 | 0.031053 |
| 40 | 10 | 0.046806 | 0.043478 | 0.045081 |
| | 20 | 0.071726 | 0.044857 | 0.055195 |
| | 30 | 0.098299 | 0.046368 | 0.063013 |
| | 40 | 0.119319 | 0.044942 | 0.065291 |
| | 50 | 0.024475 | 0.04825 | 0.032476 |
| 50 | 10 | 0.051527 | 0.04719 | 0.049263 |
| | 20 | 0.073876 | 0.04673 | 0.057248 |
| | 30 | 0.098757 | 0.046288 | 0.063033 |
| | 40 | 0.119296 | 0.044772 | 0.065109 |
| | | 0.081093 | 0.036819 | 0.050644 |

Table 13 Performance prediction using BC similarity with URL and Ff factors.

| K | MAE |
|---|---|
| 30 | 0.8042 |
| 50 | 0.787312 |
| 70 | 0.778051 |
| 100 | 0.770984 |
| 150 | 0.766186 |

Table 1 Performance accuracy using BC similarity with Jaccard and Ff factors.

| | BC- Jaccard-Ff | | | |
|---|---|---|---|---|
| K | M | Recall | Precision | F-measure |
| 10 | 10 | 0.02392027 | 0.046872 | 0.031675 |
| | 20 | 0.05080212 | 0.047508 | 0.0491 |
| | 30 | 0.074336785 | 0.047826 | 0.058205 |
| | 40 | 0.099473331 | 0.046607 | 0.063474 |
| | 50 | 0.125745689 | 0.045642 | 0.066974 |
| 20 | 10 | 0.02511547 | 0.048038 | 0.032985 |
| | 20 | 0.052527088 | 0.048834 | 0.050613 |
| | 30 | 0.073791134 | 0.04719 | 0.057566 |
| | 40 | 0.098585541 | 0.045387 | 0.062158 |
| | 50 | 0.127213204 | 0.046681 | 0.068299 |
| 30 | 10 | 0.022349948 | 0.044539 | 0.029764 |
| | 20 | 0.051066142 | 0.047879 | 0.049421 |
| | 30 | 0.077877922 | 0.048498 | 0.059773 |
| | 40 | 0.104878203 | 0.048436 | 0.066267 |
| | 50 | 0.129836009 | 0.047253 | 0.069289 |
| 40 | 10 | 0.024225647 | 0.04666 | 0.031893 |
| | 20 | 0.052837511 | 0.048674 | 0.050671 |
| | 30 | 0.073664132 | 0.047367 | 0.057658 |
| | 40 | 0.095537274 | 0.046554 | 0.062602 |
| | 50 | 0.129149509 | 0.047338 | 0.069282 |
| 50 | 10 | 0.025345853 | 0.04878 | 0.033359 |
| | 20 | 0.051462689 | 0.049046 | 0.050225 |
| | 30 | 0.074673403 | 0.047861 | 0.058334 |
| | 40 | 0.102982252 | 0.048012 | 0.065491 |
| | 50 | 0.131398197 | 0.048399 | 0.070741 |

Table 1 Performance prediction using BC similarity with Jaccard and Ff factors.

| K | MAE |
|---|---|
| 30 | 0.815696778 |
| 50 | 0.794751377 |
| 70 | 0.781830349 |
| 100 | 0.770884895 |
| 150 | 0.761647631 |

# APPENDIX B
# EXPERIMENTS SETUP

## F.1. Introduction

The experiment & evaluation phase is required to present and evaluate how well the new technique works. Therefore, our technique is assessed against traditional memory-based CF common methods using a public movie dataset mentioned in Section 2.7.2. Moreover, in this Appendix, the main stages of implementation will be pointed. Also, the general datasets which were used for evaluation of the proposed technique are illustrated. Additionally, the structure of this evaluation and the metrics that are used to evaluate the accuracy of proposed memory-based CF will be presented. Two selected splitting techniques will be described, which are used to partition the dataset into two independent sets, a training and testing set. The strength of proposed similarity will be discussed and compared to other similarity methods based on a running example. Finally, this Appendix will be ended with a summary section.

## F.2. Implementation Strategy

In this section, the stages of implementation will be listed. The steps are displayed in a visualisation form as shown in Figure 1. This work will be passing through several stages which are:

- First, the movie will be selected as the domain to test our technique using two MovieLens datasets.
- Second, the dataset will be partitioned into two independent sets, a training set and a test set. Typically, several methods employed to divide datasets such as holdout, bootstrap and cross-validation methods. In this work, two common methods will be used to achieve this task which are: holdout and cross-validation methods.
- Third, the new matrix will be constructed, normalized user-type matrix, based on user-item rating matrix. All ratings provided by the user will be utilised to infer his/her global preferences that represent the values of the normalized user-type matrix.

- Fourth, the system going to find the relationship between all users using proposed similarity measure BcSigCf to produce similarity matrix. Moreover, the similarity between users also is computed using existing common methods which are: PCC, CPCC, SPCC, Cosine, JMSD, and NHSM. Finding the similarity between users using existing similarity measures depends on the user-item rating matrix. The proposed method depends totally on the normalized user-type with some importance variable inferred from user-item matrix such as the number of ratings per user and the number of common ratings between the pair of users.
- Fifth, according to those similarities the *K* (size of neighbours) nearest neighbours will be allocated based on similarity weights. The users who have height similarity weight will be selected as neighbours for the target user.
- Sixth, collecting the items which rated by *K* nearest neighbours and not yet taken by the target user, remove the items that already rated by the target user, to present the set of candidate items.
- Seventh, evaluating and ranking those items to determine which items may preferable to the target user. In traditional memory-based, computing the predicted value of rating which would provide by the target user on those items using adjusted weighted method. In this work, the evaluation and ranking candidate items will be done by the MADM method. Thus, the TOPSIS technique will be used to evaluate and rank the candidate items as a new proposed method for evaluating candidate items.
- Finally, in the final phase of this technique, top M (size of recommendations) items will be selected to present a set of recommendations that will be provided to the target user.

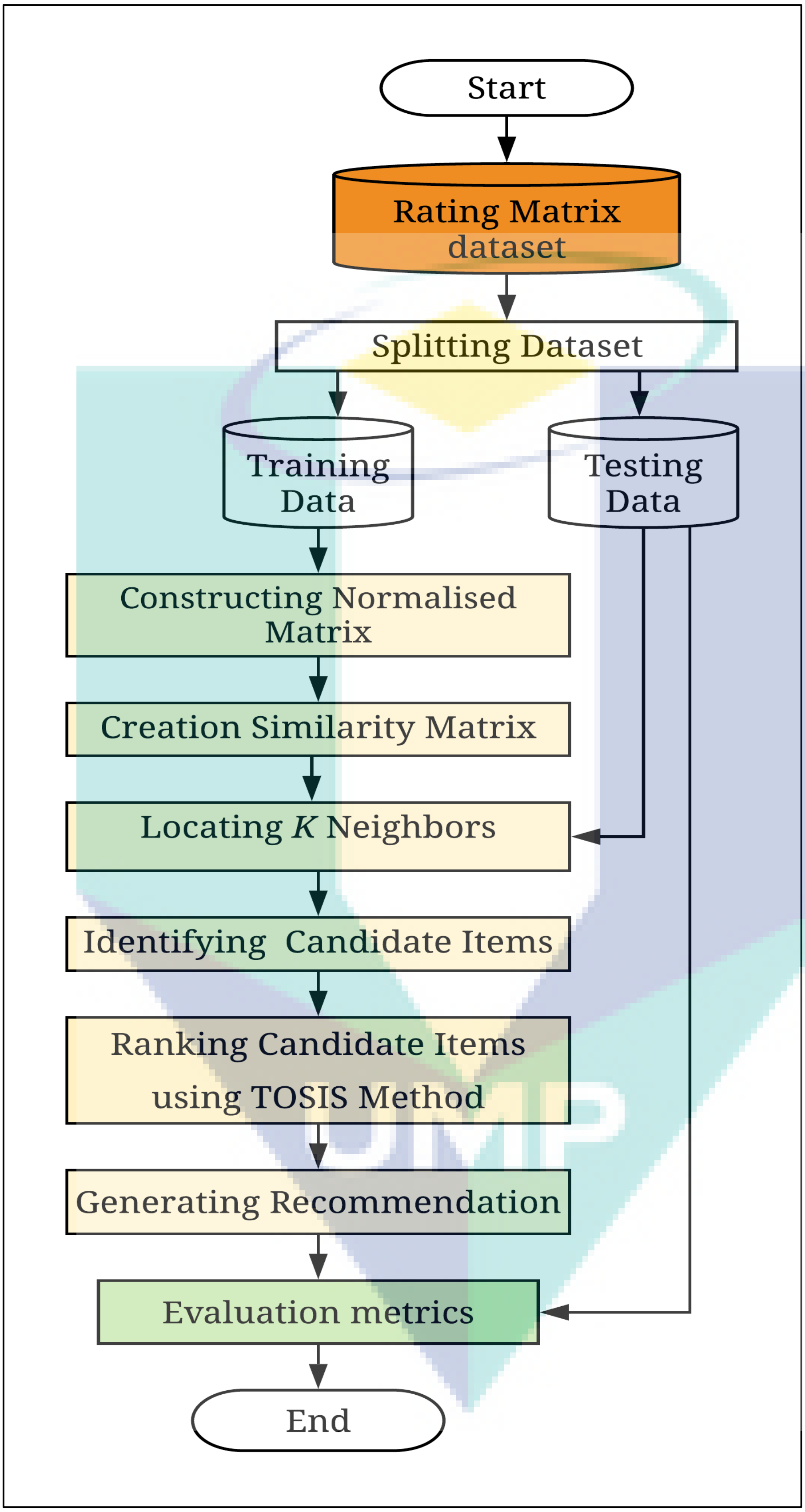



Figure 1 Implementation steps structure

### F.3. Datasets Description

The movie has been widely used application domain in memory-based CF research. Therefore, and for this reason, the movie evaluation domain is chosen to evaluate the proposed technique. There are several public movie datasets available and widely used in the processes of memory-based CF recommendation system evaluation. According to the context of the proposed method, the experiments of this work are conducted on MovieLens datasets. So, this subsection describes the properties of this datasets.

GroupLens research operates the MovieLens movie recommender system which was originally based on EachMovie dataset. There are more versions available from MovieLens dataset.

First one is a 100K dataset, released on 4/1998, which contained 100,000 ratings from 1000 users on 1700 movies. The second one, MovieLens 1M dataset released on 2/2003 with million ratings from 6000 users on 4000 movies where each user has at least 20 ratings. The rating scale is from one star (less preferable) to five stars (very preferred) as integer type in both datasets. Table 1 presents the properties of these datasets. There are three files included in these datasets. First, all ratings are contained in the file named "rating.dat" and are in the specific formatting as shown in Table.2 Second, the demographic information about the users (such as age, sex) are included in the file named "users.dat", see Table3. Finally, the information about the movie is in the file called "moies.dat" as shown in Table 4 Where Table 2, Table 3, and Table 4 represent a sampling data for rating.dat, users.dat and moies.dat, respectively. Each dataset includes a set of movies with its genre information, which are grouped into 18 different genres.

Table 1 Properties of datasets

| **Name** | **Domain** | **\|Users\|** | **\|Items\|** | **\|Ratings\|** | **Sparsity** | **\|Genres\|** | **Rating Scale** |
|---|---|---|---|---|---|---|---|
| MovieLens 100K | Movie | 967 | 4,700 | 100,00 | 0.978 % | 18 | 1-5 stars |
| MovieLens 1M | Movie | 6,040 | 3,900 | 1,000,000 | 0.9575% | 18 | 1-5 stars |

The sparsity of these datasets is derived from the ratio of empty and total entries in the user-item matrix which can be calculated using the following Equation 1:

$$sparsity = 1 - \frac{|R|}{|U| * |I|} \quad 1$$

Where:

$$|R| = number\ of\ ratings\ in\ the\ dataset.$$

$$|U| = number\ of\ users\ in\ the\ dataset.$$

$$|I| = number\ of\ items\ in\ the\ dataset.$$

For example, the sparsity of MovieLens 1M computed as follows:

$$\text{sparsity} = 1 - \frac{1{,}000{,}000}{6{,}000 * 4{,}000|} \approx 0.958$$

Table 2 Ratings Information Example

| UserID | MovieID | Ratings | Timestamp |
|---|---|---|---|
| 1 | 1 | 4 | 978300760 |
| 2 | 1 | 3 | 978302109 |
| 3 | 2 | 5 | 978299913 |
| 4 | 3 | 2 | 978298709 |
| 5 | 4 | 4 | 978302149 |

Table 3 User information Example

| UserID | Gender | Age | Occupation | Zip-code |
|---|---|---|---|---|
| 1 | M | 27 | Artist | 48067 |
| 2 | M | 35 | academic | 70072 |
| 3 | F | 45 | homemaker | 55117 |
| 4 | F | 30 | writer | 02460 |
| 5 | M | 25 | programmer | 06810 |

Table 4 Movie information Example

| MovieID | Title | Genres |
|---|---|---|
| 1 | Toy Story | Animation\|Children's\|Comedy |
| 2 | Grumpier Old Men | Comedy\|Romance |
| 3 | Waiting to Exhale | Comedy\|Drama |
| 4 | Tom and Huck | Adventure\|Children's |
| 5 | GoldenEye | Action\|Adventure\|Thriller |

For more details about the structure of 100K and 1M datasets see appendix B and C, respectively.

## F.4. Evaluation Matrices

The result ot the new method will be compared to the existing methods' result using some specific matrices that related to the specific domain to evaluate the new method. In this study, the proposed technique evaluated using the most widely used metrics in the memory-based CF recommender system. In this section, the measurements that used to compute the accuracy performance of proposed technique will be described. Two sides are used, in this study, for computing the accuracy of proposed CF-NSMA technique. These sides are: predictive accuracy and performance accuracy. Firstly, a straightforward method of measuring the predictive accuracy is to measure the Mean Absolute Error (MAE). MAE is the most widely used metric to measure the predictive accuracy of recommendations in memory-based CF research. This method calculates simply the absolute difference between each actual rating and predicted rating for all ratings of users in the test set. A lower MAE corresponds to a more accurate prediction. MAE can be defined using Equation 2 as follow:

$$MAE = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N} \tag{2}$$

Where $N$ represents the number of ratings that have been selected for the work test, $p_i$ and $r_i$ are the predicted rating and actual rating for the item $i$, respectively.

On the other side, as a performance accuracy, precision and recall metrics are used in this study. These measurements examine the accuracy of a recommender system to accurately recommend the preferable items to its users. Where, the Precision $P$ metric, is the fraction of items that rated by the user in the test set and recommended by the recommender system. In simple word, the precision metric represents the ratio of the recommended and interest items to the total number of items recommended by the system as shown in the Equation 3. Recall $R$ metric, is the fraction of rated items and recommended by a recommender system. The recall metric represents the ratio of the recommended and interesting items to whole items that are rated by the user in the test set, the recall metric can be defined using Equation 4. Where Table 5 illustrates the recommendation confusion matrix and how these metrics relate to the confusion matrix.

Table 5 Recommendation confusion matrix

| | **Rated** | **Unrated** |
|---|---|---|
| **Recommended** | TP | FP |
| **Not recommended** | FN | TN |

$$Precision = \frac{TP}{TP + FP} \quad 3$$

$$Recall = \frac{TP}{TP + FN} \quad 4$$

Moreover, the F-measure metric which has been used to evaluate recommender system. The precision and recall simplified into a single metric as shown in equation 5.

$$F - measre = \frac{2\ (Precision * Recall)}{Precision + Recall} \quad 5$$

## F.5. Splitting Techniques

To use these measurements to test the accuracy of proposed technique the holdout and k-fold cross-validation partition techniques are used. Next sections describe the concept of holdout technique and Cross-validation.

## F.6. Holdout Method

In this method, the dataset will be partitioned randomly into independent datasets, a training dataset and testing dataset. Typically, in a recommender system, the collected dataset is divided into five parts, four-fifth (80%) are taken to present the training set, and the remaining one-fifth (20%) is allocated as a testing set. In this work, this strategy of partition is applied to evaluate the CF-NSMA technique. The training set is used to construct the normalized user-type matrix which represents the global preferences of users. After that, the normalized user-type matrix is used as input part to locate the neighbours for each user through applying BSF similarity measure. While the testing set is used to calculate the accuracy of the proposed technique, see Figure 2 shows how the method been adjusted to fit our proposed.

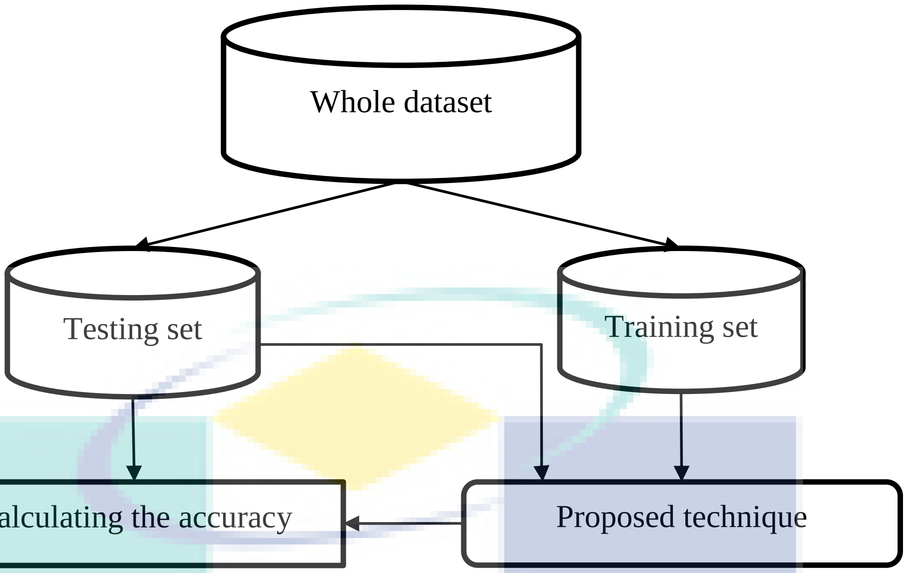



Figure 2 Holdout method

### F.7. Cross-validation Method

In $k$-fold cross-validation, the data set is partitioned into approximately equal $k$ subsets or folds ($D_1, D_2, \ldots, D_k$ ). Training and testing is executed $k$ times. In this work, the researcher partitioned the dataset into five approximately equally size folds. Each time, one of the $k$ subsets is used as a test set, and the other $k$-1 subsets are put together to form a training set. Then the average result across all $k$ trials is computed. This technique is shown in the Figure 2.7 for $k$=5.

### F.8. Experiment Environment

As an experimental work, this technique was implemented using C#.net language over visual studio 2013 environment. C#.net was selected as the language of choice due to the ease of implementation and object-oriented based programming. Moreover, the experiments executed distributed between 12 PCs in the lab FSK06. These PCs have the following properties:

- Windows: win 10 Pro.
- Processor: used Intel (R) Core(TM) i7-4790k @ 4.00 GHz.
- RAM: 16GB RAM.
- System type: 64-bit operating system, x 64-based processor.

The experiments were performed for more than 48 hours. Figure 3 presents the environment of the experiment.

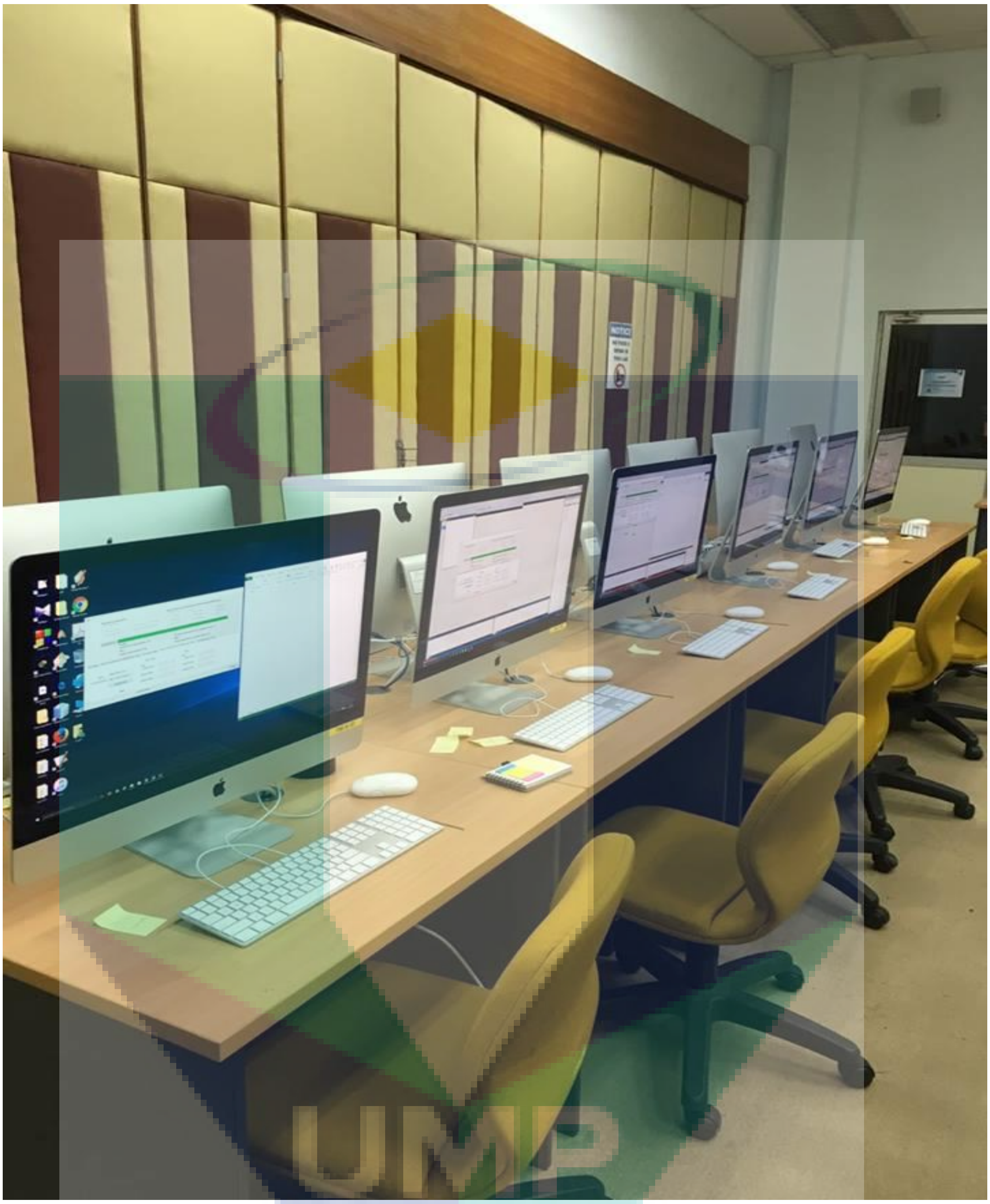


Figure 3 A simple experiment environment

### F.9. Running Example

To analyse the strengths of our proposed similarity measure, compare to the common existing similarity measures in the memory-based CF, a running example will be presented in this section. The data of this running example are developed based on previous research. Moreover, the running example will illustrate how the sparsity issue alleviated using the proposed method.

Table 6 represents an example rating matrix which contains five movies and eleven users as a given dataset. * symbol represents the absence ratings in the rating matrix and the scale rating ranged between (1) star and (5) star. Additionally, Table 7 represents the genre information of movies in the given dataset where each movie can be classified into one or more genre.

Table 6 An example user-item rating matrix

| | $movie_1$ | $movie_2$ | $movie_3$ | $movie_4$ | $movie_5$ |
|---|---|---|---|---|---|
| $U_1$ | * | 4 | 3 | 5 | 4 |
| $U_2$ | 4 | 3 | 3 | 3 | * |
| $U_3$ | * | 5 | 3 | * | * |
| $U_4$ | * | 4 | 3 | 3 | 4 |
| $U_5$ | 3 | 5 | * | 4 | 3 |
| $U_6$ | * | 2 | 1 | * | * |
| $U_7$ | 4 | * | 3 | 5 | 4 |
| $U_8$ | * | 4 | 2 | * | * |
| $U_9$ | 2 | * | * | 4 | 3 |
| $U_{10}$ | 5 | 3 | * | 3 | 3 |
| $U_{11}$ | * | * | 3 | 4 | * |

Table 7 An example genre information of movies

| Movie | Genre |
|---|---|
| $movie_1$ | Comedy, Drama, Romance, Thriller |
| $movie_2$ | Action, Crime, Thriller |
| $movie_3$ | Action, Crime, Drama |
| $movie_4$ | Action, War, Thriller |
| $movie_5$ | Action, War |

As illustrated in the Section 4.2.2, the movies' domain is classified into several types action, crime, comedy, documentary, etc. genres/types. Thus, the kind of movies rated by the user may reflect his/her preferences. Obviously, the users who prefer to watch the documentary movies will prefer to watch this type of movies more than others. Therefore, based on Table 6 and Table 7, the time and normalized metrics are constructed as shown in Table 8 and Table 9, respectively. According to the Table 7 mentioned before, we can notice there are seven genres for the movies in given dataset.

Table 8 Normalized user-type matrix, according to rating matrix in table 6

| | Comedy | Drama | Romance | Action | Crime | Thriller | War | total |
|---|---|---|---|---|---|---|---|---|
| $U_1$ | 0 | 1 | 0 | 4 | 2 | 2 | 2 | 11 |
| $U_2$ | 1 | 2 | 1 | 3 | 2 | 3 | 1 | 13 |
| $U_3$ | 0 | 1 | 0 | 2 | 2 | 1 | 0 | 6 |
| $U_4$ | 0 | 1 | 0 | 4 | 2 | 2 | 2 | 11 |
| $U_5$ | 1 | 1 | 1 | 3 | 1 | 3 | 2 | 12 |
| $U_6$ | 0 | 1 | 0 | 2 | 2 | 1 | 0 | 6 |
| $U_7$ | 1 | 2 | 1 | 3 | 1 | 2 | 2 | 12 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $U_8$ | 0 | 1 | 0 | 2 | 2 | 1 | 0 | 6 |
| $U_9$ | 1 | 1 | 1 | 2 | 0 | 2 | 2 | 9 |
| $U_{10}$ | 1 | 1 | 1 | 3 | 1 | 3 | 2 | 12 |
| $U_{11}$ | 0 | 1 | 0 | 2 | 1 | 1 | 1 | 6 |

Table 9 Normalized user-type matrix, according to rating matrix in table 5.6

| | **Comedy** | **Drama** | **Romance** | **Action** | **Crime** | **Thriller** | **War** |
|---|---|---|---|---|---|---|---|
| $U_1$ | 0 | 0.09 | 0 | 0.36 | 0.18 | 0.18 | 0.18 |
| $U_2$ | 0.076 | 0.15 | 0.076 | 0.23 | 0.15 | 0.23 | 0.076 |
| $U_3$ | 0 | 0.16 | 0 | 0.33 | 0.33 | 0.16 | 0 |
| $U_4$ | 0 | 0.09 | 0 | 0.36 | 0.18 | 0.18 | 0.18 |
| $U_5$ | 0.08 | 0.08 | 0.08 | 0.25 | 0.08 | 0.25 | 0.16 |
| $U_6$ | 0 | 0.16 | 0 | 0.33 | 0.33 | 0.16 | 0 |
| $U_7$ | 0.08 | 0.16 | 0.08 | 0.25 | 0.08 | 0.16 | 0.16 |
| $U_8$ | 0 | 0.16 | 0 | 0.33 | 0.33 | 0.16 | 0 |
| $U_9$ | 0.11 | 0.11 | 0.11 | 0.22 | 0 | 0.11 | 0.22 |
| $U_{10}$ | 0.08 | 0.08 | 0.08 | 0.25 | 0.08 | 0.16 | 0.16 |
| $U_{11}$ | 0 | 0.16 | 0 | 0.33 | 0.16 | 0.16 | 0.16 |

After the construction, the normalized user-type matrix, the similarities of users are calculated using BSF Equation 4.7. Table 10 shows the similarities values of users in the example Table 6.

For examples, let $\theta = 2$, then from Table 9, the similarity between $User_6$ and $User_9$ is computed using Equations 4.44.5, and 4.7 Section 4.2.3 and it has been captured in user-user similarities Table 10:

$$Ff(User_6, User_9) = \frac{2}{2+3} \approx 0.4$$

$$Ff(User_9, User_6) = \frac{3}{2+3} \approx 0.6$$

$$Sf(User_9, User_6) = \frac{1}{1 + Exp^{\left(-\frac{0}{2}\right)}} = 0.5$$

$$\text{BSF}(User_6, User_9) = 1/(1 + \frac{|0 * 0.4 - 0.11 * 0.6| + \cdots + |0 * 0.4 - 0.22 * 0.6|}{(0 * 0.4 + \cdots + 0.* 0.4) + (0.11 * 0.6 + \cdots + 0.22 * 0.6)}) * 0.5$$

$$\text{BSF}(User_6, User_9) = 1/(1 + \frac{0.4}{0.92}) * 0.5 \approx 0.35$$

Table 10 User-user similarities matrix

| | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ | $U_8$ | $U_9$ | $U_{10}$ | $U_{11}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $U_1$ | | 0.645 | 0.55 | 0.88 | 0.66 | 0.55 | 0.66 | 0.55 | 0.52 | 0.668 | 0.53 |
| $U_2$ | | | 0.54 | 0.65 | 0.71 | 0.6 | 0.72 | 0.54 | 0.55 | 0.7 | 0.485 |
| $U_3$ | | | | 0.55 | 0.43 | 0.73 | 0.43 | 0.73 | 0.348 | 0.44 | 0.466 |
| $U_4$ | | | | | 0.66 | 0.55 | 0.656 | 0.55 | 0.52 | 0.67 | 0.33 |
| $U_5$ | | | | | | 0.43 | 0.75 | 0.43 | 0.78 | 0.84 | 0.43 |
| $U_6$ | | | | | | | 0.43 | 0.82 | 0.348 | 0.44 | 0.466 |
| $U_7$ | | | | | | | | 0.43 | 0.68 | 0.783 | 0.49 |
| $U_8$ | | | | | | | | | 0.35 | 0.44 | 0.466 |
| $U_9$ | | | | | | | | | | 0.7 | 0.41 |
| $U_{10}$ | | | | | | | | | | | 0.39 |
| $U_{11}$ | | | | | | | | | | | |

Although several similarity measures have been proposed, they used to calculate the similarity between a pair of users and overcome the shortcomings of traditional measures such as PCC and Cosine. But still some weaknesses of their approach still exist as discussed in Section 2.5.1. Therefore, based on the example mentioned in advance, the strengths of proposed method will be described compare to most common existing similarity methods in this section.

In the beginning, in the traditional memory-based similarity methods and most developed similarity, the similarity between two users are computed based on the provided ratings by both users on common items (co-rated items). Consequently, finding the correlation between a pair of users who have not a sufficient number of co-rated items is not feasible or may lead to fake relationships. For instance, a couple of users may be similar if there is no item rated by both users. Thus, those measures are not suitable when the data is sparse. Especially in case of the pair of users do not have co-rated items, those measures can not compute the similarity between them.

Obviously, from the previously mentioned rating Table 6, the similarity between $U_8$ and $U_9$ is zero using PCC, Cosine, CPCC, MSD, Jaccard, SPCC, WPCC, JMSD and NHSM. While the similarity using proposed method BSF was approximately 0.35.

Moreover, the drawbacks related to sparsity issue are presented as follow:

- No co-rated

The small number of ratings by user leads to decrease the size of co-rated items. Therefore, the similarity measure cannot find the relationship between a pair of users/items if the number of common items not enough or there is no single item rated by both users. From our calculated rating Table 6, let *u* = (0; 4; 2; 0; 0) and *v* = (2; 0; 0; 4; 3) represent the vectors of ratings of $U_8$ and $U_9$, respectively. We can see that, there is no common item rated by both users. So the similarity between them cannot be computed using traditional and the tested similarities measures such as PCC, Cosine, CPCC, MSD, Jaccard, SPCC, WPCC, JMSD and NHSM, which is equal zero. While the similarity using proposed method BSF was approximately 0.35 as shown in Table 10.

- A small number of co-rated items

If there are a few numbers of common items between a pair of users, for example, the number of co-rated is exactly one, then the measures PCC and Cosine cannot find the correlation between them. For instance, from Table 6, $U_{10}$ and $U_{11}$ have single common item. The Cosine output is 1 regardless of their rating on the item and PCC similarity between them is -1. While the correlation equal 0.39 using the proposed similarity method BSF.

- Output low similarity despite similar

Low similarity: A pair of users get low similarity although they have similar ratings. For instance, let ratings of $U_1$and $U_4$be represented as a vector of ratings where *u*= (4, 3, 5, 4) and *v*= (4, 3, 3, 4), respectively. It can be noted that they have very similar ratings. However, the similarity between them will be zero when computed using SPCC. Also, the similarity using the CPCC, which will be 0.577, but still low. While the similarity using the proposed similarity method BSF has significant enhancement, which was 0.88 as shown in Table 10.

- Output high similarity

High similarity: A pair of users can obtain high correlation regardless of the difference between ratings of both users. For example, the rating vectors of users $U_3$ and $U_6$ are (5, 3, 0, 0) and (2, 1, 0, 0), respectively. Nevertheless, the correlation between them using PCC equals to 1. Additionally, the similarity value will be approximately equal to 1 using Cosine measure. While the correlation between them using BSF has slight enhancement, which was 0.73 as shown in the similarity Table 10.

To conclude, although our proposed similarity method does not use the actual ratings of the users on items and does not take into account the differences/ similarity in evaluations, there are a significance and slight improvement compared to traditional memory-based CF methods in case of output low/ high similarity, respectively.

- Do not take into account the proportion of common ratings.

Ignoring the proportion of common ratings in the process of calculating the similarity between users may lead to low accuracy. For example, If the rating vectors of the $U_1$, $U_3$and $U_4$are (4, 3, 5, 4), (5, 3, 0, 0) and (4, 3, 3, 4), respectively. According to MSD the similarity value between $U_1$ and $U_3$ is 2 and, the similarity between $U_1$ and $U_4$ is 1. But, obviously, $U_1$ and $U_4$ should have a stronger correlation than the correlation between $U_1$ and $U_3$. This is because the MSD only calculates the average difference between both users and does not take into account the ratio of common ratings. The proposed similarity measure overcome this shortcoming by multiple the similarity weight by the sigmoid function wieght. A sigmoid function used to devalue the similarity weight when the number of common items not enough. Therefore, the relationship between $U_1$ and $U_4$ was more strongly than $U_1$ and $U_3$. Which were, as can be seen in Table 10, 0.88 and 0.55, respectively.

- Global preference

Traditional memory-based CF similarity methods take into account only the local information of the ratings and do not consider the global preference of user ratings/ behaviour. Sometimes, the evaluation provided by the user is not accurate, because some users evaluate items randomly which may lead to low accuracy. Unlike the proposed similarity method BSF, it depends on the global preference which is represented in the normalized user-type matrix. These preferences are inferred from rating matrix utilises all ratings.

- Utilization of ratings

Most of the existing similarity memory-based CF methods do not utilise all ratings provided by the both of users. For example, PCC and Cosine similarity measures use the co-ratings between users to find the relationship between them. Unlike, the proposed method BcSigFf which uses all ratings provided by users to build their preferences that are used later as input values for proposed similarity method BSF.

- Fairness factor

The traditional memory-based CF similarities methods do not take into consideration the percentage difference in the number of ratings for each user when finding the relationship between users. For instance, the similarity between two users who have a close number of ratings should be a higher than others. Therefore, the fairness factor will be added to find these consistencies in the number of users' ratings and give a higher similarity to users with the same number of ratings. For example, the number of rating items of $U_8$, $U_{10}$ and $U_{11}$ are 2, 4 and 2, respectively. And the number of co-rated items between $U_8$ and $U_{11}$ is one. Similarly, $U_{10}$ and $U_{11}$ have a single common item. Therefore, logically, the relationship between the first pair of users must be stronger than the second pair. Because each user from the first pair has the same number of ratings, which is 2. From similarity Table 10, even though, the first and second pairs of users have the same number of co-rated items, the similarity between the first pair of users was 0.46 while the correlation between the second pair of users was 0.39. That is mean, the proposed similarity BSF has more accurate. While, the correlation between users, of both pairs, will be 1 using Cosine similarity measure because of both of them in the same line. Additionally, using PCC method, the similarity between the first pair of them will be 1 while the correlation between users of the second pair will be -1. The big differences in the relationship between them it's not logical as we can see from Table 10.

## F.10. Appendix Summary

In this Appendix, the strategy of implementation of the proposed technique was explained. The several implementation stages passed through were stated. In addition, two public selected datasets are described with its prosperities. These datasets were the MovieLens 100K and 1M dataset. Moreover, the evaluation matrices that were used to evaluate the accuracy performance of proposed technique were presented. In order use these matrices to test the accuracy of the proposed technique the partition techniques were required. Thus, next, two widely used partition techniques, holdout and k-fold cross-validation, were illustrated. The experimental environment was explained. Additionally, to analyse the strengths of our proposed similarity method compare to the common existing similarity methods in the memory-based CF, a running example was presented. The running example showed the strengths points of proposed similarity measure compare to the traditional memory-based similarity methods. From this, we conclude that the need of the proposed similarity BSF is essential to alleviate the main effects of sparsity problem that is lead to successful neighbours and in turn enhance the accuracy of recommendations as shown in this Appendix.

# APPENDIX C
# MOVIELENS 100K DATASET DESCRIPTION

SUMMARY & USAGE LICENSE
=============================================

MovieLens data sets were collected by the GroupLens Research Project
at the University of Minnesota.

This data set consists of:
* 100,000 ratings (1-5) from 943 users on 1682 movies.
* Each user has rated at least 20 movies.
* Simple demographic info for the users (age, gender, occupation, zip)

The data was collected through the MovieLens web site (movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998. This data has been cleaned up - users who had less than 20 ratings or did not have complete demographic information were removed from this data set. Detailed descriptions of the data file can be found at the end of this file.

Neither the University of Minnesota nor any of the researchers involved can guarantee the correctness of the data, its suitability for any particular purpose, or the validity of results based on the use of the data set. The data set may be used for any research purposes under the following conditions:

* The user may not state or imply any endorsement from the University of Minnesota or the GroupLens Research Group.

* The user must acknowledge the use of the data set in publications resulting from the use of the dataset (see below for citation information).

* The user may not redistribute the data without separate permission.

* The user may not use this information for any commercial or revenue-bearing purposes without first obtaining permission from a faculty member of the GroupLens Research Project at the University of Minnesota.

If you have any further questions or comments, please contact GroupLens <grouplens-info@cs.umn.edu>.

CITATION
==============================================

To acknowledge use of the dataset in publications, please cite the following paper:

F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets:
History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages.
DOI=http://dx.doi.org/10.1145/2827872


ACKNOWLEDGEMENTS
==============================================

Thanks to Al Borchers for cleaning up this data and writing the accompanying scripts.


PUBLISHED WORK THAT HAS USED THIS DATASET
==============================================

Herlocker, J., Konstan, J., Borchers, A., Riedl, J.. An Algorithmic Framework for Performing Collaborative Filtering. Proceedings of the
1999 Conference on Research and Development in Information Retrieval. Aug. 1999.

FURTHER INFORMATION ABOUT THE GROUPLENS RESEARCH PROJECT
==============================================

The GroupLens Research Project is a research group in the Department
of Computer Science and Engineering at the University of Minnesota.
Members of the GroupLens Research Project are involved in many research projects related to the fields of information filtering, collaborative filtering, and recommender systems. The project is lead
by professors John Riedl and Joseph Konstan. The project began to explore automated collaborative filtering in 1992, but is most well known for its world wide trial of an automated collaborative filtering
system for Usenet news in 1996. The technology developed in the Usenet trial formed the base for the formation of Net Perceptions, Inc., which was founded by members of GroupLens Research. Since then
the project has expanded its scope to research overall information filtering solutions, integrating in content-based methods as well as
improving current collaborative filtering technology.

Further information on the GroupLens Research project, including research publications, can be found at the following website:

http://www.grouplens.org/

GroupLens Research currently operates a movie recommender based on collaborative filtering:

http://www.movielens.org/

DETAILED DESCRIPTIONS OF DATA FILES
==============================================

```
Here are brief descriptions of the data.

ml-data.tar.gz   -- Compressed tar file.  To rebuild the u data
files do this:
                gunzip ml-data.tar.gz
                tar xvf ml-data.tar
                mku.sh

u.data     -- The full u data set, 100000 ratings by 943 users on
1682 items.
              Each user has rated at least 20 movies.  Users and
items are
              numbered consecutively from 1.  The data is randomly
              ordered. This is a tab separated list of
               user id | item id | rating | timestamp.
              The time stamps are Unix seconds since 1/1/1970 UTC

u.info     -- The number of users, items, and ratings in the u data
set.

u.item     -- Information about the items (movies); this is a tab
separated
              list of
              movie id | movie title | release date | video release
date |
              IMDb URL | unknown | Action | Adventure | Animation |
              Children's | Comedy | Crime | Documentary | Drama |
Fantasy |
              Film-Noir | Horror | Musical | Mystery | Romance |
Sci-Fi |
              Thriller | War | Western |
              The last 19 fields are the genres, a 1 indicates the
movie
              is of that genre, a 0 indicates it is not; movies can
be in
              several genres at once.
              The movie ids are the ones used in the u.data data
set.

u.genre    -- A list of the genres.

u.user     -- Demographic information about the users; this is a
tab
              separated list of
              user id | age | gender | occupation | zip code
              The user ids are the ones used in the u.data data
set.

u.occupation -- A list of the occupations.

u1.base    -- The data sets u1.base and u1.test through u5.base and
u5.test
u1.test       are 80%/20% splits of the u data into training and
test data.
u2.base       Each of u1, ..., u5 have disjoint test sets; this if
for
```

```
u2.test       5 fold cross validation (where you repeat your
experiment
u3.base       with each training and test set and average the
results).
u3.test       These data sets can be generated from u.data by
mku.sh.
u4.base
u4.test
u5.base
u5.test

ua.base    -- The data sets ua.base, ua.test, ub.base, and ub.test
ua.test       split the u data into a training set and a test set
with
ub.base       exactly 10 ratings per user in the test set.  The
sets
ub.test       ua.test and ub.test are disjoint.  These data sets
can
              be generated from u.data by mku.sh.

allbut.pl  -- The script that generates training and test sets
where
              all but n of a users ratings are in the training
data.

mku.sh     -- A shell script to generate all the u data sets
from u.data.
```

# APPENDIX D
# MOVIELENS 1M DATASET DESCRIPTION

MovieLens 1M Dataset
SUMMARY
===========================================================================

These files contain 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000.

USAGE LICENSE
===========================================================================

Neither the University of Minnesota nor any of the researchers involved can guarantee the correctness of the data, its suitability for any particular purpose, or the validity of results based on the use of the data set. The data set may be used for any research purposes under the following conditions:

* The user may not state or imply any endorsement from the University of Minnesota or the GroupLens Research Group.

* The user must acknowledge the use of the data set in publications resulting from the use of the dataset (see below for citation information).

* The user may not redistribute the data without separate permission.

* The user may not use this information for any commercial or revenue-bearing purposes without first obtaining permission from a faculty member of the GroupLens Research Project at the University of Minnesota.

If you have any further questions or comments, please contact GroupLens <grouplens-info@cs.umn.edu>.

CITATION
===========================================================================

To acknowledge use of the dataset in publications, please cite the following paper:

F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872

ACKNOWLEDGEMENTS
===================================================================
=============

Thanks to Shyong Lam and Jon Herlocker for cleaning up and generating the data set.

FURTHER INFORMATION ABOUT THE GROUPLENS RESEARCH PROJECT
===================================================================
=============

The GroupLens Research Project is a research group in the Department of Computer Science and Engineering at the University of Minnesota. Members of the GroupLens Research Project are involved in many research projects related to the fields of information filtering, collaborative filtering, and recommender systems. The project is lead by professors John Riedl and Joseph Konstan. The project began to explore automated collaborative filtering in 1992, but is most well known for its world wide trial of an automated collaborative filtering system for Usenet news in 1996. Since then the project has expanded its scope to research overall information filtering solutions, integrating in content-based methods as well as improving current collaborative filtering technology.

Further information on the GroupLens Research project, including research publications, can be found at the following website:

http://www.grouplens.org/

GroupLens Research currently operates a movie recommender based on collaborative filtering:

http://www.movielens.org/

RATINGS FILE DESCRIPTION
===================================================================
=============

All ratings are contained in the file "ratings.dat" and are in the following format:

UserID::MovieID::Rating::Timestamp

- UserIDs range between 1 and 6040
- MovieIDs range between 1 and 3952
- Ratings are made on a 5-star scale (whole-star ratings only)

- Timestamp is represented in seconds since the epoch as returned by time(2)
- Each user has at least 20 ratings

USERS FILE DESCRIPTION
================================================================================

User information is in the file "users.dat" and is in the following format:

UserID::Gender::Age::Occupation::Zip-code

All demographic information is provided voluntarily by the users and is
not checked for accuracy.  Only users who have provided some demographic
information are included in this dataset.

- Gender is denoted by a "M" for male and "F" for female
- Age is chosen from the following ranges:

```
	*  1:  "Under 18"
	* 18:  "18-24"
	* 25:  "25-34"
	* 35:  "35-44"
	* 45:  "45-49"
	* 50:  "50-55"
	* 56:  "56+"
```

- Occupation is chosen from the following choices:

```
	*  0:  "other" or not specified
	*  1:  "academic/educator"
	*  2:  "artist"
	*  3:  "clerical/admin"
	*  4:  "college/grad student"
	*  5:  "customer service"
	*  6:  "doctor/health care"
	*  7:  "executive/managerial"
	*  8:  "farmer"
	*  9:  "homemaker"
	* 10:  "K-12 student"
	* 11:  "lawyer"
	* 12:  "programmer"
	* 13:  "retired"
	* 14:  "sales/marketing"
	* 15:  "scientist"
	* 16:  "self-employed"
	* 17:  "technician/engineer"
	* 18:  "tradesman/craftsman"
	* 19:  "unemployed"
	* 20:  "writer"
```

MOVIES FILE DESCRIPTION
================================================================================

Movie information is in the file "movies.dat" and is in the following
format:

MovieID::Title::Genres

- Titles are identical to titles provided by the IMDB (including year of release)
- Genres are pipe-separated and are selected from the following genres:

	* Action
	* Adventure
	* Animation
	* Children's
	* Comedy
	* Crime
	* Documentary
	* Drama
	* Fantasy
	* Film-Noir
	* Horror
	* Musical
	* Mystery
	* Romance
	* Sci-Fi
	* Thriller
	* War
	* Western

- Some MovieIDs do not correspond to a movie due to accidental duplicate
entries and/or test entries
- Movies are mostly entered by hand, so errors and inconsistencies may exist

# APPENDIX E
# EXAMPLE RESULT

Table 1 Recall 100K using Holdout splitting method

| K | M | CF-PCC | CF-CPCC | CF-SPCC | CF-Cosine | CF-MSD | CF-JMSD | CF-NHSM | CF-BSF |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 10 | 0.018677 | 0.017747 | 0.011872 | 0.029609 | 0.028815 | 0.026758989 | 0.022765077 | 0.036689907 |
| | 20 | 0.032177 | 0.034843 | 0.028028 | 0.046958 | 0.05636 | 0.051895813 | 0.045474502 | 0.064452098 |
| | 30 | 0.04809 | 0.052619 | 0.047269 | 0.063936 | 0.08579 | 0.076604185 | 0.065167729 | 0.084775428 |
| | 40 | 0.06688 | 0.076056 | 0.06345 | 0.087997 | 0.105996 | 0.094350576 | 0.088542947 | 0.108602376 |
| | 50 | 0.080694 | 0.092265 | 0.079457 | 0.107958 | 0.135347 | 0.123279949 | 0.105317964 | 0.135008423 |
| 20 | 10 | 0.018886 | 0.020124 | 0.014681 | 0.022859 | 0.033221 | 0.026572383 | 0.023344412 | 0.027742939 |
| | 20 | 0.033894 | 0.037198 | 0.030095 | 0.041359 | 0.058562 | 0.051426856 | 0.043102572 | 0.059771181 |
| | 30 | 0.046843 | 0.050121 | 0.04682 | 0.071738 | 0.087438 | 0.073769702 | 0.065458025 | 0.080833115 |
| | 40 | 0.064085 | 0.071753 | 0.06268 | 0.098061 | 0.108013 | 0.104339737 | 0.086951709 | 0.101318341 |
| | 50 | 0.080364 | 0.083022 | 0.081793 | 0.123086 | 0.134693 | 0.133063288 | 0.100901866 | 0.137633329 |
| 30 | 10 | 0.019339 | 0.018302 | 0.014787 | 0.033134 | 0.038119 | 0.025427614 | 0.021784007 | 0.028674575 |
| | 20 | 0.033886 | 0.034777 | 0.029796 | 0.057756 | 0.0583 | 0.054282469 | 0.043522654 | 0.057631096 |
| | 30 | 0.050549 | 0.049382 | 0.045649 | 0.073761 | 0.08785 | 0.074506628 | 0.067134753 | 0.07815823 |
| | 40 | 0.067978 | 0.065162 | 0.062925 | 0.095154 | 0.102758 | 0.104454674 | 0.087207627 | 0.110342569 |
| | 50 | 0.084891 | 0.082583 | 0.082483 | 0.110344 | 0.123329 | 0.133232039 | 0.109641005 | 0.130086074 |
| 40 | 10 | 0.020455 | 0.018559 | 0.015309 | 0.028291 | 0.035681 | 0.026297086 | 0.022022367 | 0.026980989 |
| | 20 | 0.037889 | 0.033541 | 0.029605 | 0.051419 | 0.057898 | 0.047412598 | 0.043929633 | 0.051985235 |
| | 30 | 0.053504 | 0.047975 | 0.044192 | 0.067698 | 0.087681 | 0.070396702 | 0.066397032 | 0.075981194 |
| | 40 | 0.07174 | 0.06575 | 0.061203 | 0.079849 | 0.102814 | 0.099370839 | 0.084360134 | 0.099983449 |
| | 50 | 0.090754 | 0.07969 | 0.082834 | 0.090832 | 0.118223 | 0.128256333 | 0.105896689 | 0.128690927 |
| 50 | 10 | 0.020074 | 0.017538 | 0.016305 | 0.021297 | 0.026786 | 0.027628126 | 0.022567446 | 0.026130226 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 20 | 0.036051 | 0.035407 | 0.030101 | 0.033394 | 0.056202 | 0.049849067 | 0.044391895 | 0.052739153 |
| | 30 | 0.051393 | 0.049272 | 0.047488 | 0.042108 | 0.088899 | 0.077103181 | 0.066403278 | 0.077702327 |
| | 40 | 0.06608 | 0.064764 | 0.063955 | 0.056361 | 0.111824 | 0.09896707 | 0.085031468 | 0.101139348 |
| | 50 | 0.075494 | 0.077188 | 0.081625 | 0.069237 | 0.131554 | 0.125901763 | 0.10331408 | 0.138791938 |

Average

| | M | CF-PCC | CF-CPCC | CF-SPCC | CF-Cosine | CF-MSD | CF-JMSD | CF-NHSM | CF-BSF |
|---|---|---|---|---|---|---|---|---|---|
| | 10 | 0.019486 | 0.018454 | 0.014591 | 0.027038 | 0.032524 | 0.02653684 | 0.022496662 | 0.029243727 |
| | 20 | 0.034779 | 0.035153 | 0.029525 | 0.046177 | 0.057464 | 0.05097336 | 0.044084251 | 0.057315752 |
| | 30 | 0.050076 | 0.049874 | 0.046284 | 0.063848 | 0.087532 | 0.07447608 | 0.066112163 | 0.079490059 |
| | 40 | 0.067353 | 0.068697 | 0.062842 | 0.083485 | 0.106281 | 0.100296579 | 0.086418777 | 0.104277217 |
| | 50 | 0.08244 | 0.08295 | 0.081639 | 0.100292 | 0.128629 | 0.128746674 | 0.105014321 | 0.134042138 |

Table 2 Recall 1M dataset using Holdout

| K | M | CF-PCC | CF-CPCC | CF-SPCC | CF-Cosine | CF-MSD | CF-JMSD | CF-NHSM | CF-BSF |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 10 | 0.013223 | 0.015509 | 0.014546 | 0.012399 | 0.011529 | 0.027900988 | 0.029386847 | 0.035405133 |
| | 20 | 0.023659 | 0.028668 | 0.026093 | 0.022896 | 0.023026 | 0.050931622 | 0.049600794 | 0.062797177 |
| | 30 | 0.032786 | 0.040899 | 0.036611 | 0.032428 | 0.032164 | 0.070609327 | 0.068343764 | 0.08697086 |
| | 40 | 0.042044 | 0.052599 | 0.046285 | 0.040998 | 0.041038 | 0.088114133 | 0.085623222 | 0.108001876 |
| | 50 | 0.050873 | 0.063601 | 0.055281 | 0.049887 | 0.049281 | 0.105533256 | 0.100439295 | 0.128131897 |
| 20 | 10 | 0.012734 | 0.015451 | 0.014167 | 0.012787 | 0.011976 | 0.026982037 | 0.029425386 | 0.034406349 |
| | 20 | 0.023032 | 0.028561 | 0.025944 | 0.023428 | 0.022525 | 0.048497356 | 0.04938061 | 0.061635314 |
| | 30 | 0.032451 | 0.040677 | 0.036287 | 0.032494 | 0.031743 | 0.066886911 | 0.066936303 | 0.083140704 |
| | 40 | 0.041111 | 0.051533 | 0.045418 | 0.041142 | 0.039983 | 0.083325662 | 0.08173551 | 0.10284395 |
| | 50 | 0.049751 | 0.061422 | 0.054253 | 0.049102 | 0.04863 | 0.097634616 | 0.095727582 | 0.121029144 |
| 30 | 10 | 0.012764 | 0.015179 | 0.014292 | 0.012972 | 0.011396 | 0.025955831 | 0.029040681 | 0.034134082 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 20 | 0.022924 | 0.028214 | 0.025821 | 0.023017 | 0.022369 | 0.047669687 | 0.048071844 | 0.059825498 |
| | 30 | 0.032524 | 0.040245 | 0.035698 | 0.032023 | 0.03135 | 0.065019186 | 0.065010148 | 0.082232418 |
| | 40 | 0.041462 | 0.051292 | 0.045141 | 0.040149 | 0.040014 | 0.080667183 | 0.079536927 | 0.101750863 |
| | 50 | 0.049611 | 0.061426 | 0.054057 | 0.048341 | 0.048473 | 0.094848668 | 0.093227105 | 0.1182263 |
| 40 | 10 | 0.012906 | 0.014905 | 0.014075 | 0.012557 | 0.011977 | 0.025214823 | 0.029629741 | 0.033756211 |
| | 20 | 0.023264 | 0.027972 | 0.025022 | 0.022732 | 0.022525 | 0.046163012 | 0.048498034 | 0.059610181 |
| | 30 | 0.032556 | 0.039639 | 0.035498 | 0.031876 | 0.031596 | 0.062960222 | 0.064767309 | 0.081445346 |
| | 40 | 0.04095 | 0.050774 | 0.044803 | 0.03948 | 0.040069 | 0.078799674 | 0.078902878 | 0.100062315 |
| | 50 | 0.048734 | 0.06099 | 0.053506 | 0.047861 | 0.0486 | 0.092746362 | 0.092464355 | 0.116981473 |
| 50 | 10 | 0.012779 | 0.01488 | 0.014033 | 0.012932 | 0.011582 | 0.025126886 | 0.029167199 | 0.034453461 |
| | 20 | 0.023023 | 0.028084 | 0.025135 | 0.022582 | 0.022673 | 0.045899434 | 0.048548593 | 0.059594734 |
| | 30 | 0.03216 | 0.039639 | 0.034773 | 0.031534 | 0.031534 | 0.063045448 | 0.064431004 | 0.081282347 |
| | 40 | 0.04054 | 0.050435 | 0.044505 | 0.039105 | 0.039814 | 0.077751136 | 0.078709206 | 0.099349133 |
| | 50 | 0.048051 | 0.060475 | 0.053771 | 0.046821 | 0.048281 | 0.092311378 | 0.091912704 | 0.115544721 |

Average

| M | CF-PCC | CF-CPCC | CF-SPCC | CF-Cosine | CF-MSD | CF-JMSD | CF-NHSM | CF-BSF |
|---|---|---|---|---|---|---|---|---|
| 10 | 0.012881 | 0.015185 | 0.014223 | 0.012729 | 0.011692 | 0.026236113 | 0.029329971 | 0.034431047 |
| 20 | 0.02318 | 0.0283 | 0.025603 | 0.022931 | 0.022624 | 0.047832222 | 0.048819975 | 0.060692581 |
| 30 | 0.032496 | 0.04022 | 0.035773 | 0.032071 | 0.031677 | 0.065704219 | 0.065897705 | 0.083014335 |
| 40 | 0.041221 | 0.051326 | 0.04523 | 0.040175 | 0.040184 | 0.081731557 | 0.080901548 | 0.102401627 |
| 50 | 0.049404 | 0.061583 | 0.054173 | 0.048402 | 0.048653 | 0.096614856 | 0.094754208 | 0.119982707 |

Table 3 Recall 100K using Cross-validation splitting method

| K | M | CF-PCC | CF-CPCC | CF-SPCC | CF-Cosine | CF-MSD | CF-JMSD | CF-NHSM | CF-BSF |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 10 | 0.029646 | 0.031853 | 0.03389 | 0.025101 | 0.02336 | 0.059644416 | 0.066362815 | 0.068777521 |
| | 20 | 0.052144 | 0.058721 | 0.061953 | 0.045413 | 0.043339 | 0.103995122 | 0.119198861 | 0.126056019 |
| | 30 | 0.07137 | 0.082521 | 0.085602 | 0.061987 | 0.06058 | 0.139151178 | 0.165742513 | 0.17453211 |
| | 40 | 0.089543 | 0.102308 | 0.107267 | 0.07763 | 0.077142 | 0.173177516 | 0.20792624 | 0.217341075 |
| | 50 | 0.10708 | 0.122347 | 0.128185 | 0.091963 | 0.093538 | 0.204454814 | 0.244841789 | 0.255477592 |
| 20 | 10 | 0.028689 | 0.031165 | 0.033677 | 0.024772 | 0.023787 | 0.056662909 | 0.066114142 | 0.068101423 |
| | 20 | 0.05145 | 0.058505 | 0.061959 | 0.043753 | 0.042639 | 0.099470165 | 0.119366347 | 0.123390695 |
| | 30 | 0.071394 | 0.081903 | 0.084983 | 0.058904 | 0.059112 | 0.133995875 | 0.166515627 | 0.169769846 |
| | 40 | 0.088378 | 0.103423 | 0.105781 | 0.073556 | 0.074996 | 0.165244135 | 0.207689576 | 0.210289221 |
| | 50 | 0.105854 | 0.122117 | 0.125802 | 0.087932 | 0.090997 | 0.191836329 | 0.242821526 | 0.246430421 |
| 30 | 10 | 0.028893 | 0.031665 | 0.033646 | 0.025109 | 0.023588 | 0.055885686 | 0.065965721 | 0.068047379 |
| | 20 | 0.051121 | 0.058104 | 0.060667 | 0.043543 | 0.04181 | 0.097076058 | 0.120349348 | 0.124029043 |
| | 30 | 0.07076 | 0.081352 | 0.084208 | 0.058894 | 0.057202 | 0.130787711 | 0.166768857 | 0.168063696 |
| | 40 | 0.088279 | 0.101743 | 0.104885 | 0.075316 | 0.0722 | 0.159272992 | 0.207821817 | 0.206971777 |
| | 50 | 0.104174 | 0.120684 | 0.124991 | 0.090031 | 0.088117 | 0.185756029 | 0.242218317 | 0.242131576 |
| 40 | 10 | 0.028867 | 0.03102 | 0.0343 | 0.024471 | 0.023705 | 0.054841249 | 0.0670014 | 0.067579534 |
| | 20 | 0.050473 | 0.057033 | 0.060915 | 0.042542 | 0.041428 | 0.096860488 | 0.120014439 | 0.122289735 |
| | 30 | 0.069793 | 0.079638 | 0.084027 | 0.058243 | 0.056635 | 0.129592599 | 0.166429011 | 0.167207382 |
| | 40 | 0.086441 | 0.09987 | 0.104495 | 0.073149 | 0.071281 | 0.157884773 | 0.205920582 | 0.204885915 |
| | 50 | 0.102548 | 0.120845 | 0.124139 | 0.08757 | 0.087307 | 0.183930137 | 0.240537519 | 0.239045707 |
| 50 | 10 | 0.028018 | 0.031057 | 0.033434 | 0.024706 | 0.024026 | 0.055159545 | 0.066138645 | 0.067687266 |
| | 20 | 0.049579 | 0.057666 | 0.061768 | 0.042357 | 0.04166 | 0.095803489 | 0.118916099 | 0.121673735 |
| | 30 | 0.069478 | 0.079748 | 0.083991 | 0.057573 | 0.056589 | 0.127141881 | 0.165741947 | 0.165570971 |
| | 40 | 0.086189 | 0.099201 | 0.104283 | 0.071903 | 0.071017 | 0.155077876 | 0.2044433 | 0.203375178 |
| | 50 | 0.10182 | 0.11999 | 0.124945 | 0.085448 | 0.085752 | 0.180405165 | 0.237832259 | 0.238762231 |

| Average | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| M | CF-PCC | CF-CPCC | CF-SPCC | CF-Cosine | CF-MSD | CF-JMSD | CF-NHSM | FC-BcSigFf |
| 10 | 0.028822 | 0.031352 | 0.033789 | 0.024832 | 0.023693 | 0.056438761 | 0.066316545 | 0.068038625 |
| 20 | 0.050953 | 0.058006 | 0.061452 | 0.043521 | 0.042175 | 0.098641064 | 0.119569019 | 0.123487846 |
| 30 | 0.070559 | 0.081032 | 0.084562 | 0.05912 | 0.058024 | 0.132133849 | 0.166239591 | 0.169028801 |
| 40 | 0.087766 | 0.101309 | 0.105342 | 0.074311 | 0.073327 | 0.162131458 | 0.206760303 | 0.208572633 |
| 50 | 0.104295 | 0.121197 | 0.125613 | 0.088589 | 0.089142 | 0.189276495 | 0.241650282 | 0.244369505 |

Table 4 Recall 1M using Cross-validation method

| K | M | CF-PCC | CF-CPCC | CF-SPCC | CF-Cosine | CF-MSD | CF-JMSD | CF-NHSM | CF-BSF |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 10 | 0.012918 | 0.015414 | 0.014215 | 0.012377 | 0.012088 | 0.028151622 | 0.028959037 | 0.034941037 |
| | 20 | 0.023178 | 0.028515 | 0.025767 | 0.022377 | 0.022798 | 0.051051119 | 0.050643076 | 0.062389361 |
| | 30 | 0.032835 | 0.04083 | 0.036323 | 0.031782 | 0.032535 | 0.070581541 | 0.069418391 | 0.08625172 |
| | 40 | 0.041984 | 0.05229 | 0.046239 | 0.040747 | 0.041674 | 0.0886771 | 0.086338156 | 0.107670402 |
| | 50 | 0.05091 | 0.063058 | 0.055524 | 0.049584 | 0.05028 | 0.105496848 | 0.101337741 | 0.127727845 |
| 20 | 10 | 0.01265 | 0.014985 | 0.014027 | 0.012387 | 0.012236 | 0.026875464 | 0.028611616 | 0.034260299 |
| | 20 | 0.023071 | 0.028078 | 0.02543 | 0.022552 | 0.02278 | 0.048295374 | 0.049185111 | 0.060105876 |
| | 30 | 0.032397 | 0.040096 | 0.035637 | 0.031736 | 0.032249 | 0.066549814 | 0.066986489 | 0.082181356 |
| | 40 | 0.04114 | 0.05125 | 0.045202 | 0.040097 | 0.041125 | 0.082716064 | 0.08253295 | 0.102067534 |
| | 50 | 0.049427 | 0.061737 | 0.054029 | 0.048194 | 0.04954 | 0.09742072 | 0.096419628 | 0.120150239 |
| 30 | 10 | 0.012368 | 0.014786 | 0.013922 | 0.012677 | 0.012086 | 0.026057262 | 0.028241703 | 0.033450133 |
| | 20 | 0.022528 | 0.027724 | 0.025244 | 0.02263 | 0.022724 | 0.046777093 | 0.048521246 | 0.05870469 |
| | 30 | 0.031926 | 0.03967 | 0.035325 | 0.031702 | 0.031866 | 0.064209506 | 0.065965127 | 0.080493166 |
| | 40 | 0.040791 | 0.050963 | 0.044559 | 0.039981 | 0.040909 | 0.080048028 | 0.080843027 | 0.099794914 |
| | 50 | 0.048837 | 0.061673 | 0.053337 | 0.047747 | 0.048813 | 0.094593211 | 0.094341347 | 0.116932079 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 40 | 10 | 0.012297 | 0.014671 | 0.013791 | 0.01249 | 0.012245 | 0.025766831 | 0.028389637 | 0.033296123 |
| | 20 | 0.022535 | 0.02769 | 0.024959 | 0.022494 | 0.022594 | 0.046061859 | 0.04840763 | 0.058430032 |
| | 30 | 0.031762 | 0.039705 | 0.035188 | 0.031218 | 0.031785 | 0.063262256 | 0.065277593 | 0.079823689 |
| | 40 | 0.040187 | 0.050691 | 0.044386 | 0.039197 | 0.040412 | 0.078541827 | 0.079939741 | 0.098531331 |
| | 50 | 0.047976 | 0.06131 | 0.052926 | 0.046921 | 0.048651 | 0.092887237 | 0.093185548 | 0.11510731 |
| 50 | 10 | 0.012282 | 0.014732 | 0.01362 | 0.012391 | 0.012318 | 0.025557012 | 0.028245658 | 0.03346806 |
| | 20 | 0.02236 | 0.027702 | 0.024932 | 0.022335 | 0.022726 | 0.04551443 | 0.04825247 | 0.058176897 |
| | 30 | 0.03153 | 0.039273 | 0.03508 | 0.031145 | 0.031899 | 0.062838379 | 0.064978754 | 0.079025293 |
| | 40 | 0.039662 | 0.050492 | 0.044291 | 0.03889 | 0.040457 | 0.07772158 | 0.079492339 | 0.097467382 |
| | 50 | 0.047431 | 0.061141 | 0.052868 | 0.046253 | 0.048636 | 0.091607977 | 0.092474945 | 0.114068807 |

Average

| # Items retrieved | CF-PCC | CF-CPCC | CF-SPCC | CF-Cosine | CF-MSD | CF-JMSD | CF-NHSM | CF-BSF |
|---|---|---|---|---|---|---|---|---|
| 10 | 0.012503 | 0.014918 | 0.013915 | 0.012464 | 0.012195 | 0.026481638 | 0.02848953 | 0.03388313 |
| 20 | 0.022734 | 0.027942 | 0.025267 | 0.022478 | 0.022724 | 0.047539975 | 0.049001907 | 0.059561371 |
| 30 | 0.03209 | 0.039915 | 0.03551 | 0.031517 | 0.032067 | 0.065488299 | 0.066525271 | 0.081555045 |
| 40 | 0.040753 | 0.051137 | 0.044935 | 0.039783 | 0.040916 | 0.08154092 | 0.081829242 | 0.101106313 |
| 50 | 0.048916 | 0.061784 | 0.053736 | 0.04774 | 0.049184 | 0.096401199 | 0.095551842 | 0.118797256 |

# APPENDIX F
# EXPERIMENTS AND COMPARISON

## F.1. Introduction

This Appendix presents the experiments and results in more details. The experiments & results will be discussed and compared with the widely used traditional memory-based CF methods to show how well the new technique works. First, several denominator values of the sigmoid function are tested to determine the acceptable number of common items, when will the number of co-rated items be more enough? The determined value is used as a key input in the computation similarity process to increase the similarity weight when the number of common items more enough. Next, the performance accuracy of proposed similarity method CF-BSF is presented by comparing its results using 100K & 1M MovieLens datasets with holdout (HO) and cross-validation (CV) splitting methods. Similarly, the performance accuracy of proposed technique CF-NSMA is tested to show the improvement that made by MADAM. Moreover, to show the strength of MADAM method the results of CF-BSF and CF-NSMA are compared. Finally, this chapter will be ended by a conclusion section.

## F.2. Preliminary Experiments

In this section, to include the proportion of common ratings in the process of calculating the similarity between a pair of users, need to determine the size of co-rated items that is sufficient to increase or decrease the weight of similarity between the pair of users. In this work, the sigmoid function is used to devalue the similarity weight between the pair of users when the number of co-rated is not more enough. But it depends on its denominator value. Therefore, several experiments were conducted to find the appropriate denominator value that can improve the similarity measure and in turn to a better result. Table 1 represents the description of initial testing denominator values how to effect on sigmoid value.

Table 1 Denominator values description.

| **Denominator value $\theta$** | **Description** |
|---|---|
| 5 | The sigmoid value would be bigger 0.9 if the number of co-rated items more than ten else would be less than 0.9. In contrast, if the number of common ratings of the pair of user equal 0, then the sigmoid value would be 0.5 |

| | |
|---|---|
| 7 | The sigmoid value would be bigger 0.9 if the number of co-rated items more than 15 else would be less than 0.9 |
| 9 | The sigmoid value would be bigger 0.9 if the number of co-rated items more than 20 else would be less than 0.9 |
| 11 | The sigmoid value would be bigger 0.9 if the number of co-rated items more than 25 else would be less than 0.9 |
| 13 | The sigmoid value would be bigger 0.9 if the number of co-rated items more than 30 else would be less than 0.9 |

Experiment one: Identify the appropriate denominator value of the sigmoid function.

Technique: CF-BSF.

Input: 100K MovieLens dataset.

Denominator values testing (5, 7, 9, 11, and 13).

Size of neighbours K (10, 20, 30, 40, and 50).

Splitting method: Holdout method.

The aim: Test the input denominator values to determine the appropriate denominator value.

Metric evaluation: MAE measurement.

Description: Figure 1 illustrates the bar graph of the MAE rate for CF-BSF using holdout splitting method. In this experiment, the Movie Lens 100K is used. The number of the neighbourhood was 30, 50, 70, 100, and 150 as shown in the figure below. As is presented in that graph, there is a slight improvement in the MAE value when the number of K neighbours increases. Similarly, the MAE value has enhancement when the amount of denominator rise. Overall, it is clear that the MAE values are better when the size of neighbours is 150 overall denominator values. Whereas, as we can see, the lowest MAE rate was when the denominator equals 13. Additionally, the MAE when the denominator was equal 9 is approximately near to the best MAE rate compare to other.

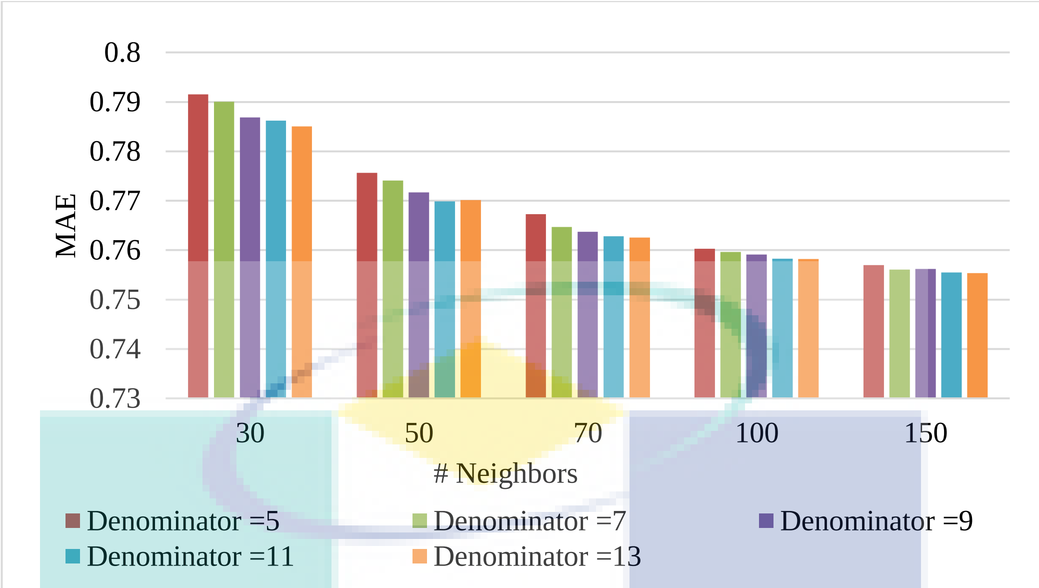



Figure 1 MAE of CF-BSF using holdout splitting method and 100K MovieLens dataset

| | |
|---|---|
| Experiment two: | Identify the appropriate denominator value of the sigmoid function. |
| Technique: | CF-BSF. |
| Input: | 100K MovieLens dataset. |
| | Denominator values testing (5, 7, 9, 11, and 13). |
| | Size of neighbours K (10, 20, 30, 40, and 50). |
| | A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout method. |
| The aim: | Test the input denominator values to determine the appropriate denominator value. |
| Metric evaluation: | Recall measurement. |
| Description: | Figure 2 shows the comparison of recall for CF-BSF using the initial denominator values. The subfigures A, B, C, D, and E represent the recall rates according to the size of neighbours 10, 20, 30, 40, and 50, respectively. Where the horizontal axis represents the size of recommendations (10, 20, 30, 40, 50). According to these subgraphs, we can see that, the recall rates, in the most cases, were the highest when the denominator value equals 9. Similarly, in some cases, the recall percentages have a good rate when the value of denominator less than 9. Unlike, the worst recall values were when the denominator bigger than 9. |

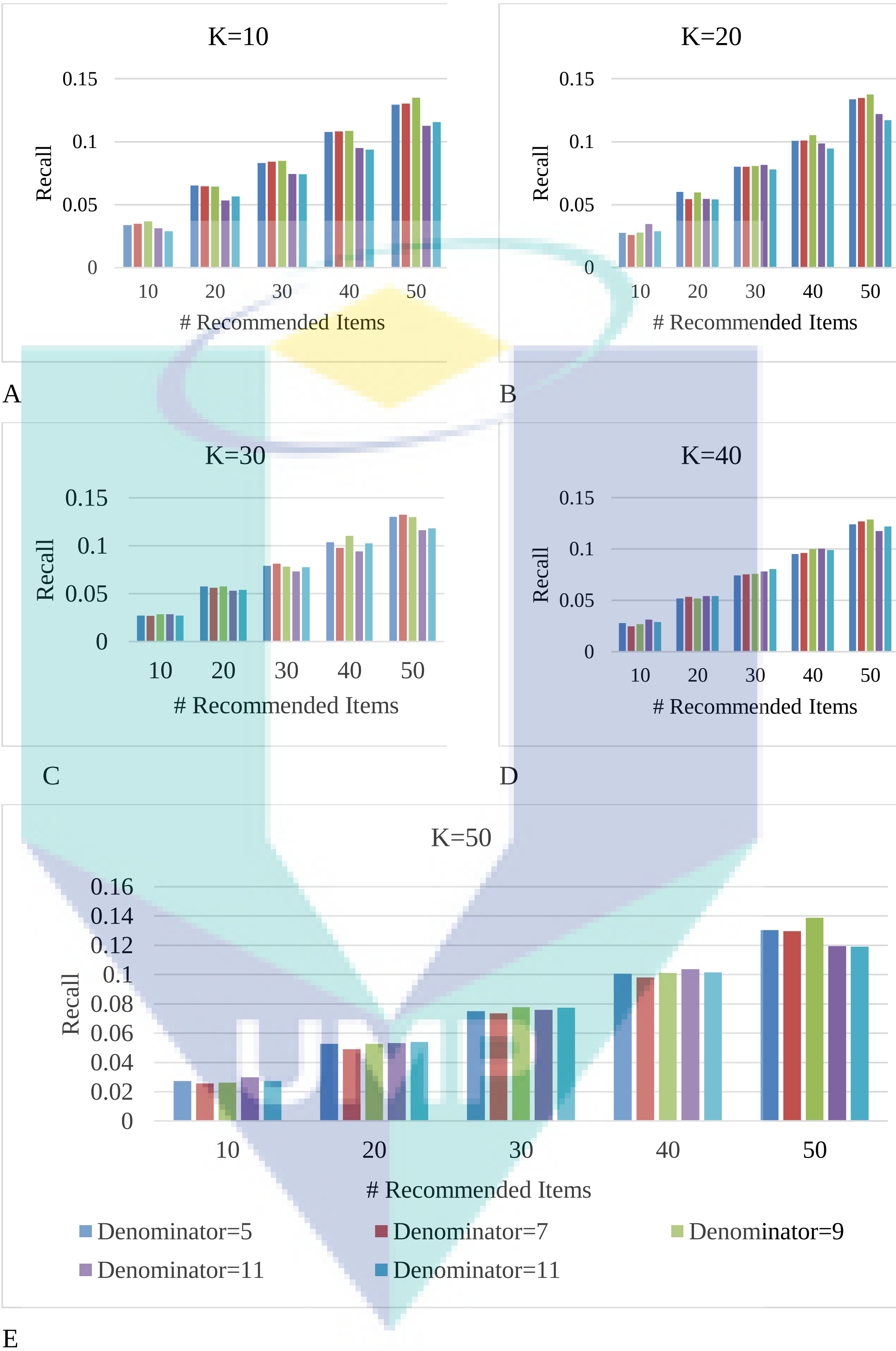
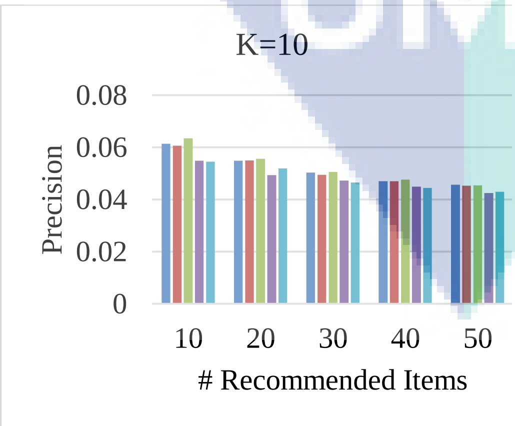
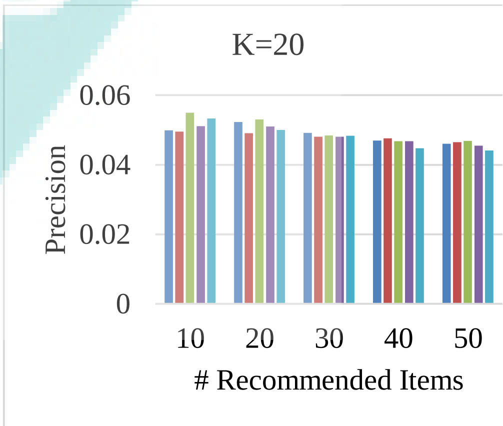



Figure 2 Recall of CF-BSF using holdout splitting method and 100K MovieLens dataset.

Experiment three: Identify the appropriate denominator value of the sigmoid function.

| | |
|---|---|
| Technique: | CF-BSF. |
| Input: | 100K MovieLens dataset.<br>Denominator values testing (5, 7, 9, 11, and 13).<br>Size of neighbours K (10, 20, 30, 40, and 50).<br>A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout method. |
| The aim: | Test the input denominator values to determine the appropriate denominator value. |
| Metric evaluation: | Precision measurement. |
| Description: | The given Figure 3 shows the precision rate of CF-BSF on five different denominator values: 5, 7, 9, 11 and 13. The subfigures A, B, C, D, and E represent the precision rates regarding the number of neighbours 10, 20, 30, 40, and 50, respectively. Where the horizontal axis represents the number of recommended items (10, 20, 30, 40, and 50).<br>As is presented in the subgraphs, when the denominator value was equal 9, the rate of precision was highest, in most cases, while the opposite is true for denominator values equal 11 and 13. Moreover, the precision rate does not exceed the precision rate of denominator value less than 9. According to these subgraphs, we can see that the highest percentage of precision was when the number of recommended items was small. Therefore, the precision rate has slightly decreased when the number of recommended items has increased.<br>To sum up, it is clear from the graph that, the rate of precision has variations, whereas with denominator equal nine almost has a higher rate in all given subgraphs. |



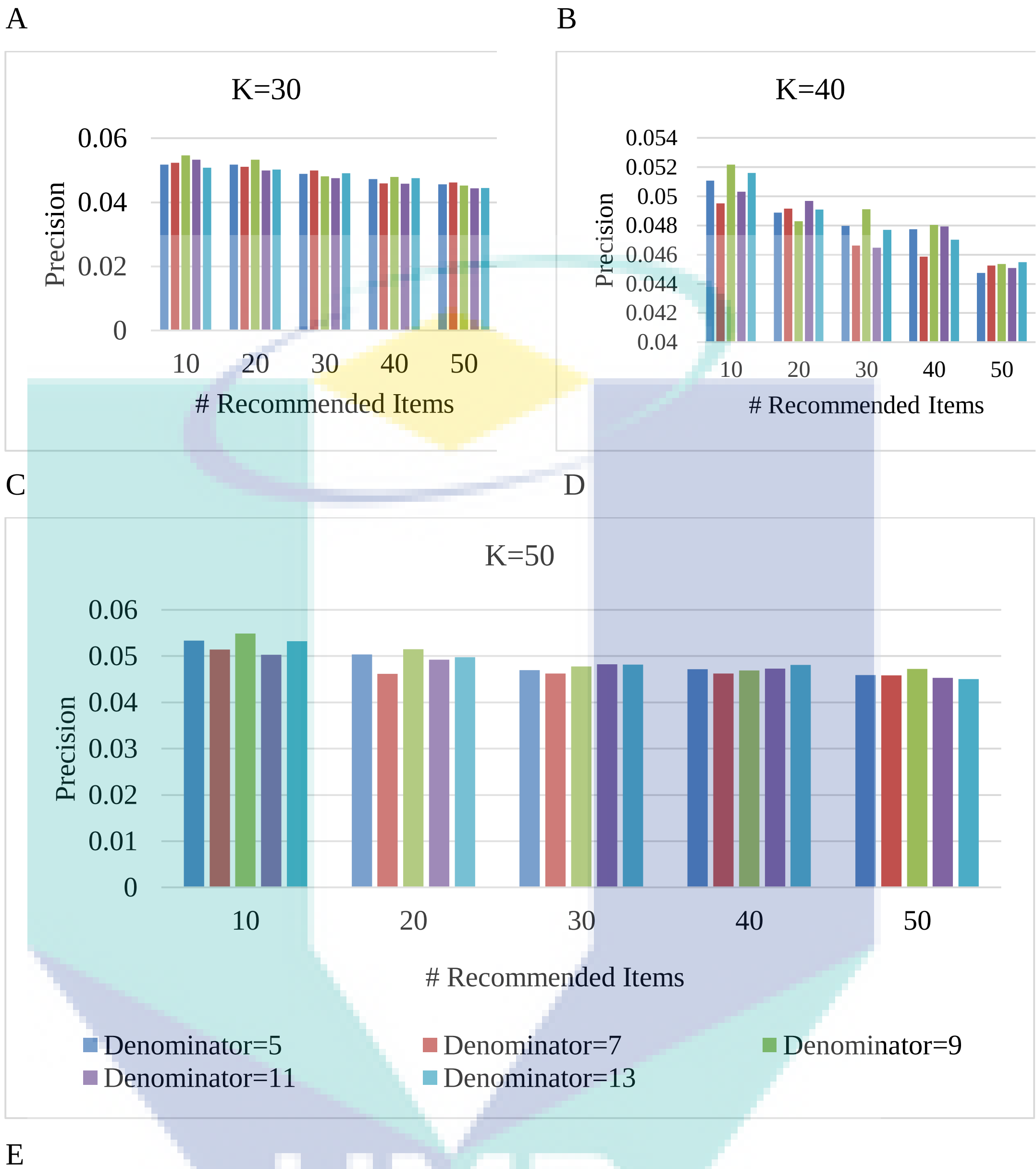


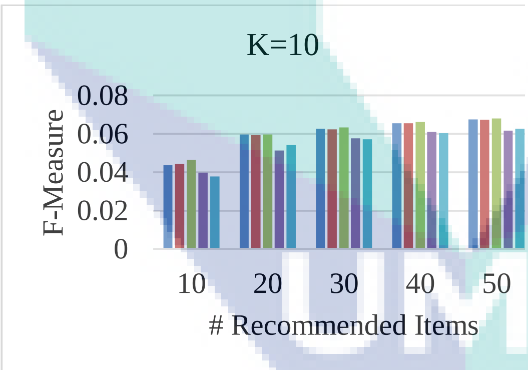

Figure 3 Precision of CF-BSF using holdout splitting method and 100K MovieLens dataset

| | |
|---|---|
| Experiment four: | Identify the appropriate denominator value of the sigmoid function. |
| Technique: | CF-BSF. |
| Input: | 100K MovieLens dataset. |
| | Denominator values testing (5, 7, 9, 11, and 13). |
| | Size of neighbours K (10, 20, 30, 40, and 50). |
| | A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout method. |
| The aim: | Test the input denominator values to determine the appropriate denominator value. |
| Metrics: | F-Measure measurement. |

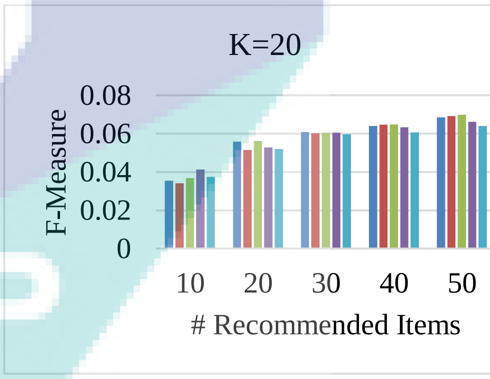
Description: The bar graphs Figure 4 depicts the F-measure rate of CF-BSF for initial denominator values. The experiment has been conducted on five different denominator values using 100K MovieLens dataset. In general, the subfigures A, B, C, D, and E represent the F-measure for a different number of neighbours 10, 20, 30, 40, and 50, respectively. Where the horizontal axis represents the number of recommended items (10, 20, 30, 40, and 50).

As is given in the illustration, the F- measure result shows that the highest rate was when the denominator value equals 9. Overall, it is clear that the worst rate was when the denominator values equal 11 and 13. While, in case of denominator value less than 9, the F-measure rate have likely the same rate of denominator value equal 9. According to these subgraphs, we can see that the highest percentage of F-measure was when the number of recommended items was 50. Therefore, the F-measure rate has slight enhancement when the number of recommended items has increased.

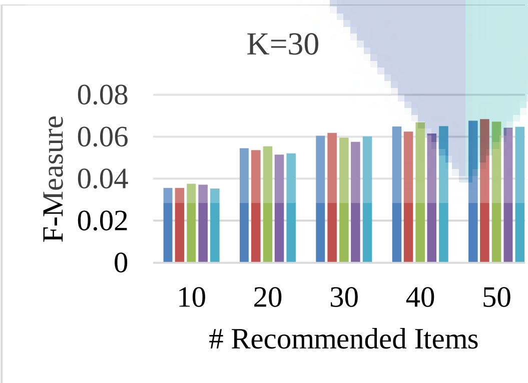
In short, it is clear from the graph that all F-measure rates using five different denominator values have a different result. Nevertheless, the rate of F-measure almost was the highest when the denominator value was 9 in all given subgraphs.

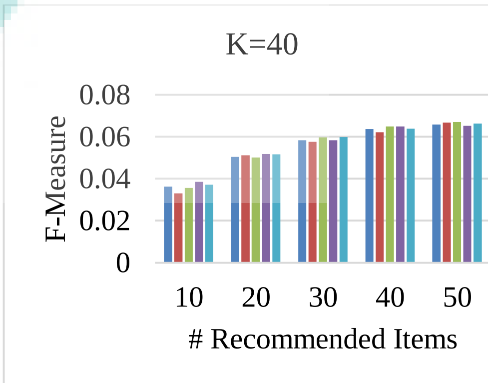



A

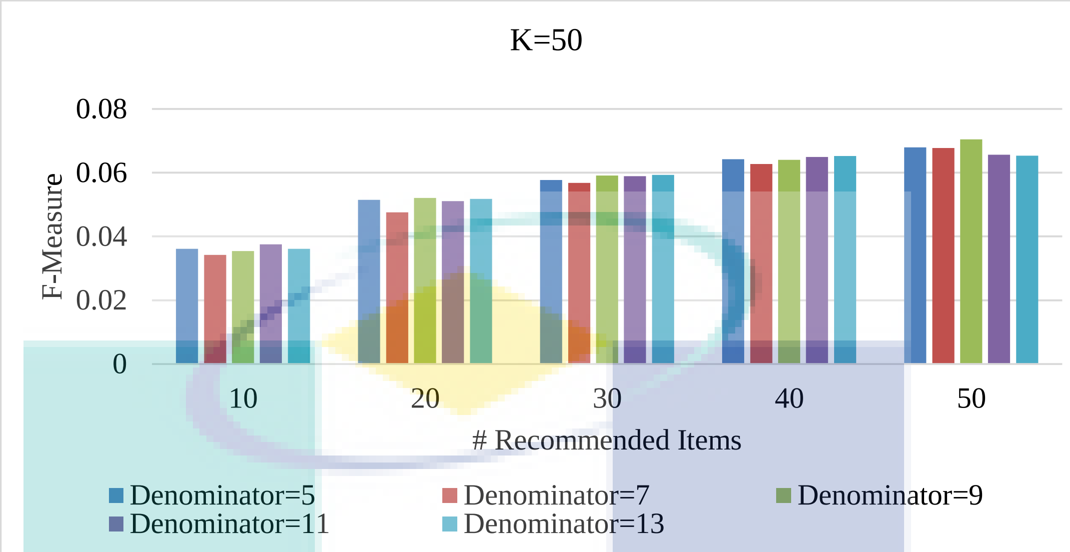



B



C D




F

Figure 4 F-measure of CF-BSF using holdout splitting method and 100K MovieLens dataset

***Observation***

Overall, from all figures that are mentioned before, Figure 1 to Figure 4, the data indicates that denominator value has effected in the prediction and performance accuracy. Additionally, the size of neighbours and number of recommended items also have an impact on both metrics. As we mentioned in the aim of these experiments, is to determine the more appropriate value of denominator. This value will be used in as a primary input for the sigmoid function to identify the right number of common rating items between users. When the number of co-rated between a pair of users not more enough the similarity weight between them will be devalued using the sigmoid function.

To sum up, although MAE rate was the best when the value of denominator bigger than 9, the recall, precision, and the F-measure rate was the worst. In contrast, the MAE rate was the worst when the value of denominator smaller than 9. Moreover, the recall, precision, and F-measure percentages were not better when compared to percentages when the denominator is 9. Therefore, as a final result, we can say that the more appropriate value for the denominator is 9.

### F.3. CF-BSF Results

In this section, four experiments are implemented on 100k & 1M MovieLens datasets to test CF-BSF technique. Holdout and cross-validation splitting methods are applied to divide these datasets into training and testing sets. Four metrics are used to present the results as shown next.

| | |
|---|---|
| Experiment one: | CF-BSF results. |
| Technique: | CF-BSF. |
| Input: | 100K & 1M MovieLens dataset.<br>Denominator value 9.<br>Size of neighbours K (10, 20, 30, 40, and 50).<br>A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout & Cross-validation methods. |
| The aim: | To present the prediction accuracy of CF-BSF using Holdout & Cross-validation for both datasets. |
| Metrics: | MAE metric. |
| Description & observation | The bar chart in Figure 5 shows the MAE for CF-BSF using two splitting techniques on two datasets. The vertical axis represents the percentage of MAE, and the horizontal axis represents the size of neighbours (30, 50, 70, 100, and 150).<br>As is given in the graph, there is a slight gradually enhancement in the MAE rate from the start (number of users was 30) to well below approximately 0.75 and 0.73 with 100k and 1M datasets, respectively, using both splitting methods. Thus, it is clear that the MAE rates with 1M dataset were better than 100K dataset in all cases. |

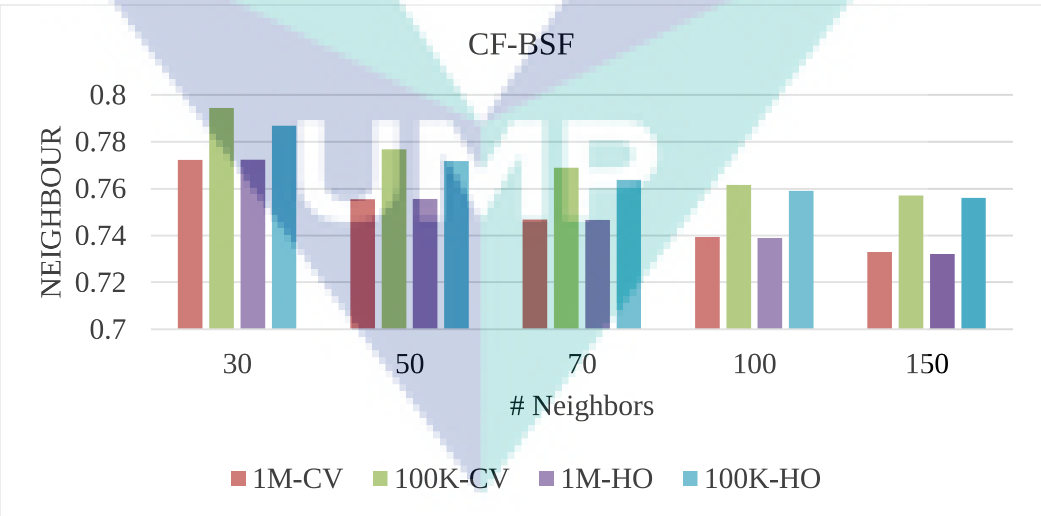
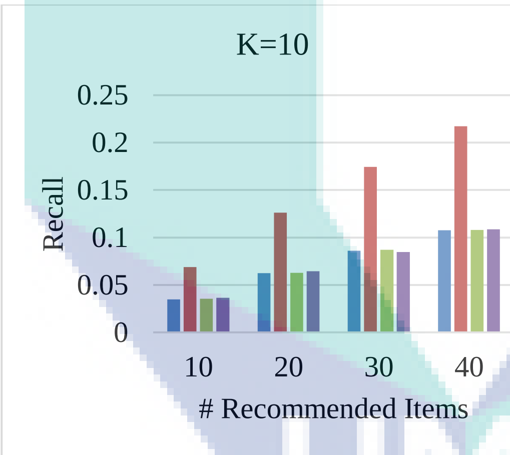


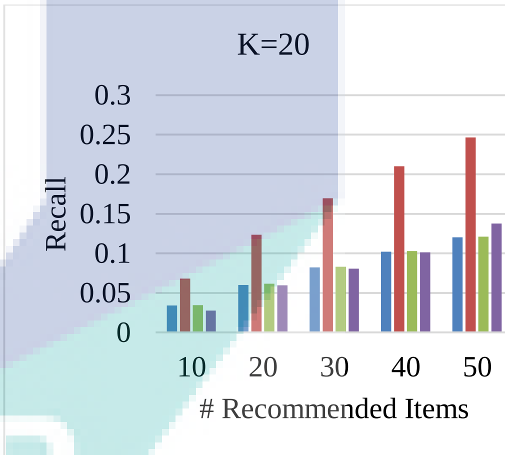

Figure 5 MAE of CF-BSF (Holdout vs Cross-Validation method) using 100K & 1M MovieLens datasets.

| | |
|---|---|
| Experiment two: | CF-BSF results. |
| Technique: | CF-BSF. |

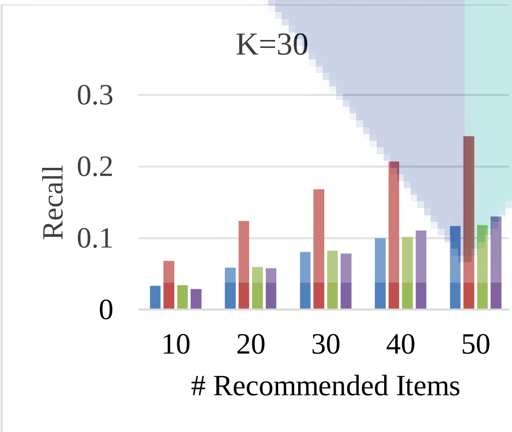
| | |
|---|---|
| Input: | 100K & 1M MovieLens dataset.<br>Denominator value 9.<br>Size of neighbours K (10, 20, 30, 40, and 50).<br>A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout & cross-validation methods. |
| The aim: | To present the performance accuracy of CF-BSF using Holdout & Cross-validation for both datasets. |
| Metrics: | Recall metric. |
| Description & observation | Figure 6 illustrates the comparison of recall for CF-BSF using Holdout & cross-validation methods on both datasets. The subgraphs A, B, C, D, and E represent the recall rates according to the size of neighbours 10, 20, 30, 40, and 50, respectively. Where the size of recommended items (10, 20, 30, 40, and 50) is presented in the horizontal axis<br>Overall, it can be seen that the recall rates of CF-BSF, using cross-validation on the 100k dataset, were the highest rates on all subgraphs. Furthermore, the recall rate rises gradually from beginning point (size of recommendation 10) to reach to the best at the end (number of recommendations is 50). |

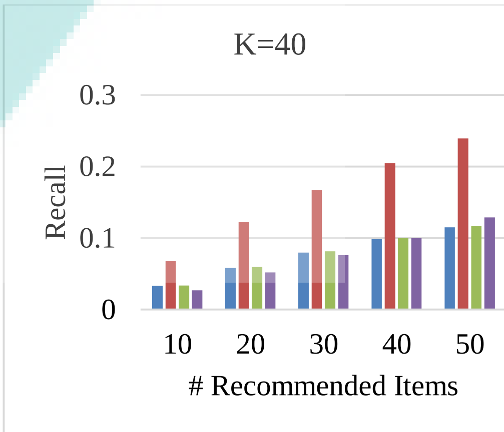



A

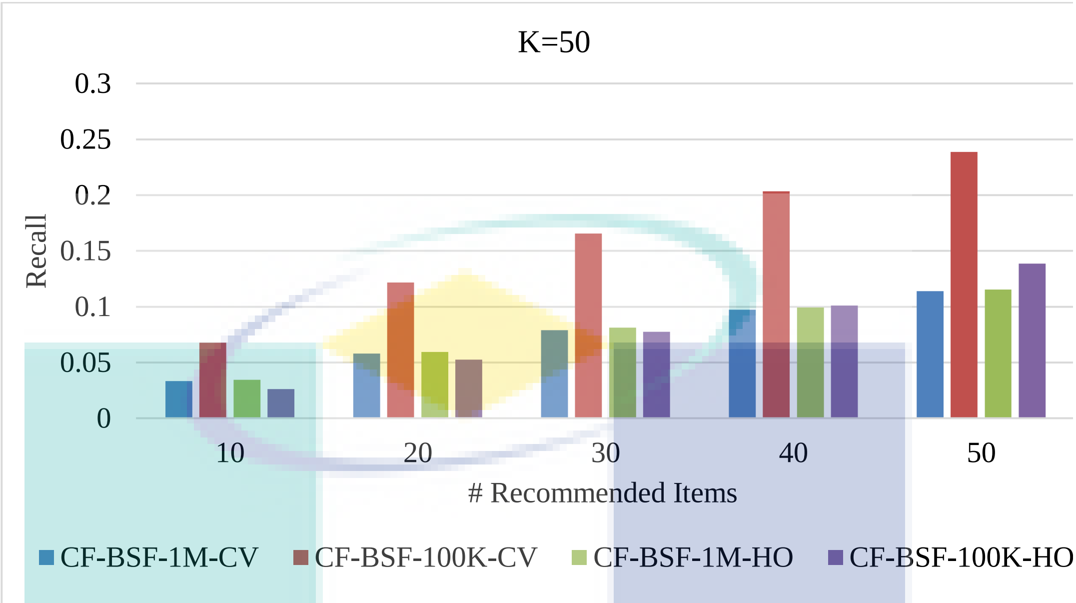



B



C D

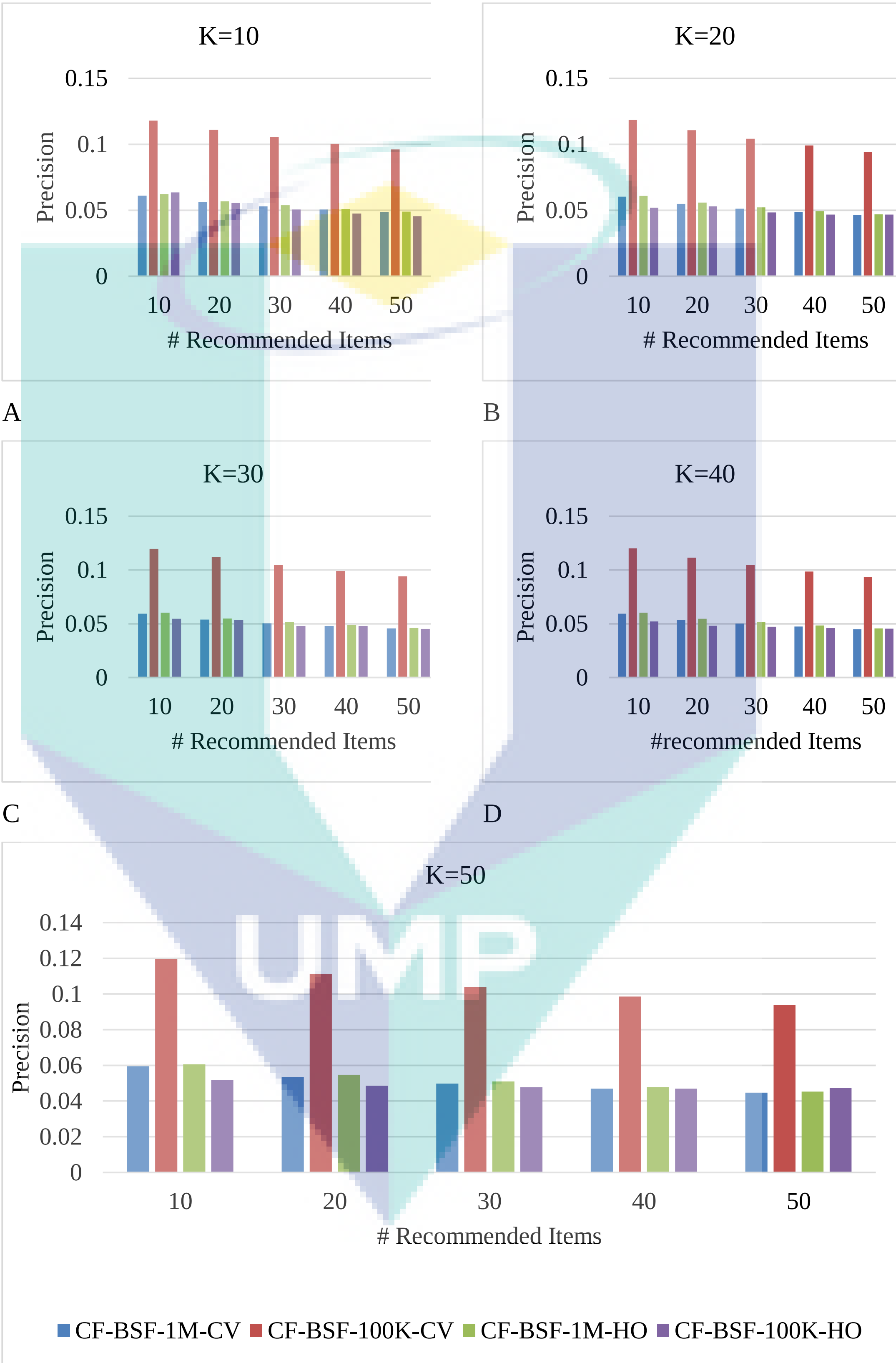



E

Figure 6 Recall of CF-BSF (Holdout vs Cross-Validation method) using 100K & 1M MovieLens datasets.

| | |
|---|---|
| Experiment three: | CF-BSF results. |
| Technique: | CF-BSF. |
| Input: | 100K & 1M MovieLens dataset.<br>Denominator value 9.<br>Size of neighbours K (10, 20, 30, 40, and 50).<br>A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout & cross-validation methods. |
| The aim: | To present the performance accuracy of CF-BSF using Holdout & Cross-validation for both datasets. |
| Metrics: | Precision metric. |
| Description & observation | The supplied bar graphs (Figure 7) compares the rate of precision for CF-BSF using Holdout & cross-validation methods on both datasets. Where the subgraphs A, B, C, D, and E represent the recall rates according to the size of neighbours 10, 20, 30, 40, and 50, respectively. Where the size of recommended items (10, 20, 30, 40, and 50) is presented in the horizontal axis<br>As a general trend, the precision influence by the size of recommendation set more than the size of neighbours. It can be seen that the precision rates of CF-BSF, using cross-validation on the 100k dataset, were the highest rates on all subgraphs. As is presented in the figure, the precision rate decreased by almost |

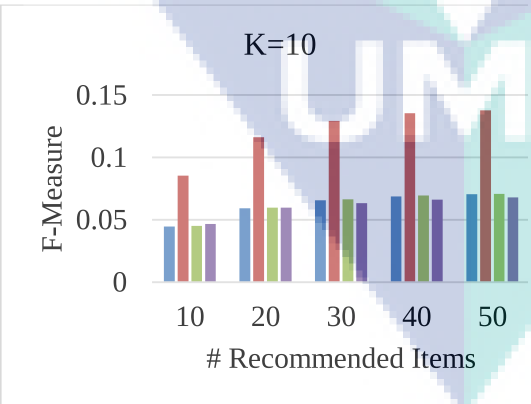
20% from beginning point (size of recommendation 10) to the end (number of recommendations is 50).

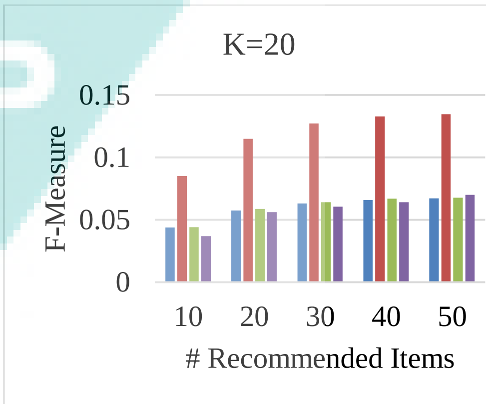


E

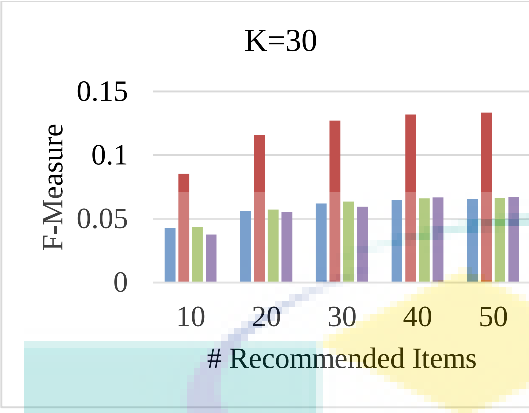
Figure 7 Precision of CF-BSF (Holdout vs Cross-Validation method) using 100K & 1M MovieLens datasets.

| | |
|---|---|
| Experiment four: | CF-BSF results. |
| Technique: | CF-BSF. |
| Input: | 100K & 1M MovieLens dataset.<br>Denominator value 9.<br>Size of neighbours K (10, 20, 30, 40, and 50).<br>A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout & cross-validation methods. |
| The aim: | To present the performance accuracy of CF-BSF using Holdout & Cross-validation for both datasets. |
| Metrics: | F-measure metric. |
| Description & observation | The bar graphs (Figure 8) enumerate the F-measure percentages of CF-BSF using Holdout & cross-validation methods on 100k and 1M datasets. Where the subgraphs A, B, C, D, and E represent the recall rates according to the size of neighbours 10, 20, 30, 40, and 50, respectively. Where the size of recommended items (10, 20, 30, 40, and 50) is presented in the horizontal axis<br>At first glance it is clear, the F-measure rates using cross-validation on 100k dataset was the highest rate on the all subgraphs overall cases. As is presented in the figure, the F-measure rate increased by around 20% from beginning point (size of recommendation 10) to the end (number of recommendations is 50) overall subgraphs. |

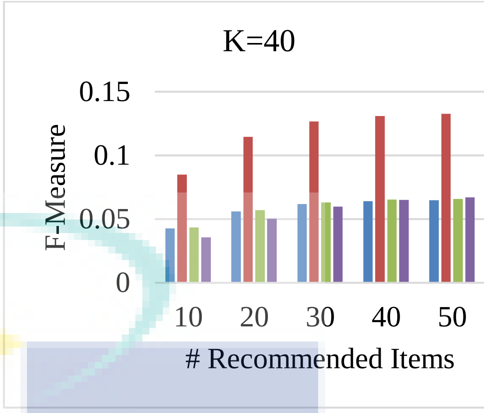


A

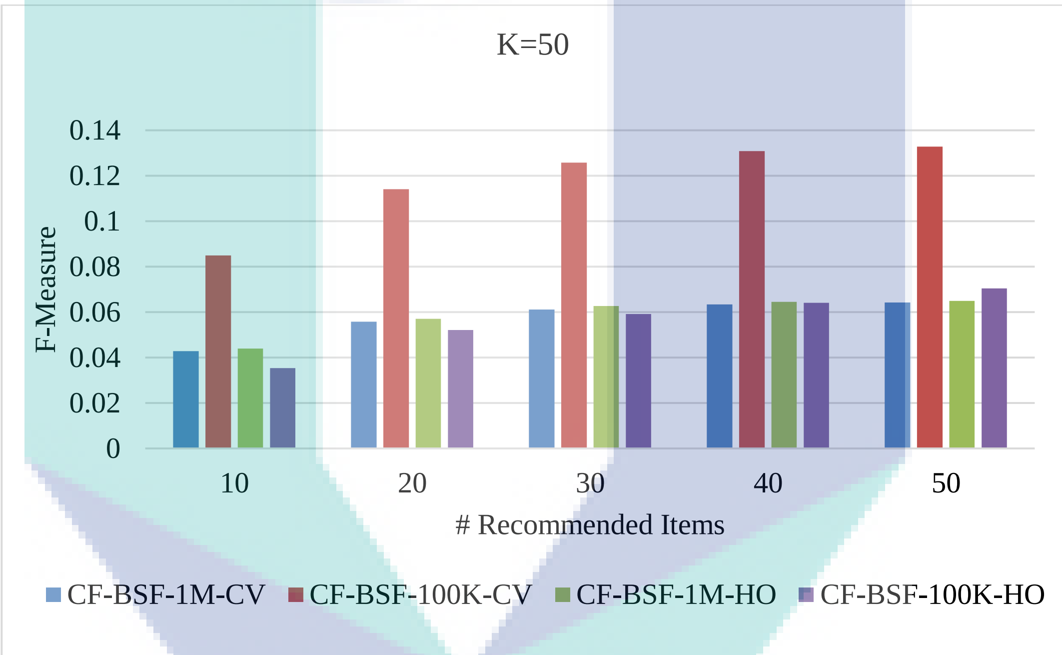



B




C

D




E

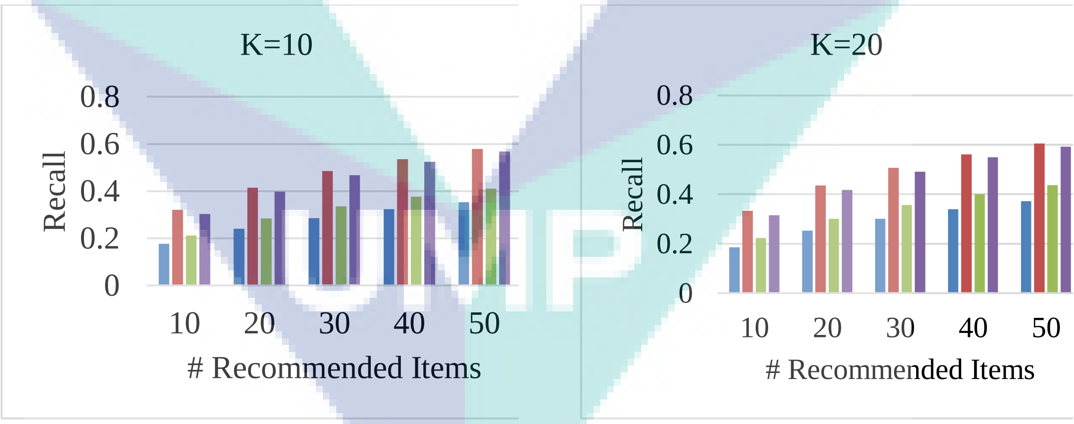
Figure 8 F-measure of CF-BSF (Holdout vs Cross-Validation method) using 100K & 1M MovieLens datasets.

### F.4. CF-NSMA Proposed Technique Results

In this section, three experiments are implemented on 100k & 1M MovieLens datasets to test the proposed CF-NSMA technique. The dataset partitioned into two sets, training and testing sets, using two splitting methods (Holdout & cross-validation). Four metrics are used to present the results as shown.

Experiment one: CF-NSMA results.

| | |
|---|---|
| Technique: | CF-NSMA. |
| Input: | 100K & 1M MovieLens dataset.<br>Denominator value 9.<br>Size of neighbours K (10, 20, 30, 40, and 50).<br>A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout & cross-validation methods. |
| The aim: | To test the performance accuracy of CF-NSMA using Holdout & Cross-validation for both datasets. |
| Metrics: | Recall metric. |
| Description & observation | Figure 9 illustrates the comparison of recall for CF-NSMA using Holdout & cross-validation methods on both datasets. The subgraphs A, B, C, D, and E, represent the recall rates according to the size of neighbours 10, 20, 30, 40, and 50, respectively. Where the size of recommended items (10, 20, 30, 40, 50) is presented in the horizontal axis<br>Overall, it can be seen that the recall rates using 100k dataset were the highest rates on all subgraphs for both splitting datasets methods. Moreover, the recall rate rose slightly from start point (size of recommendation is 10) to reach the highest at endpoint (number of recommendations is 50). With 100k dataset, the recall rate using cross-validation a slightly better than the holdout. Unlike, in the 1M dataset, the holdout rate was better than cross-validation by approximately 5%. |



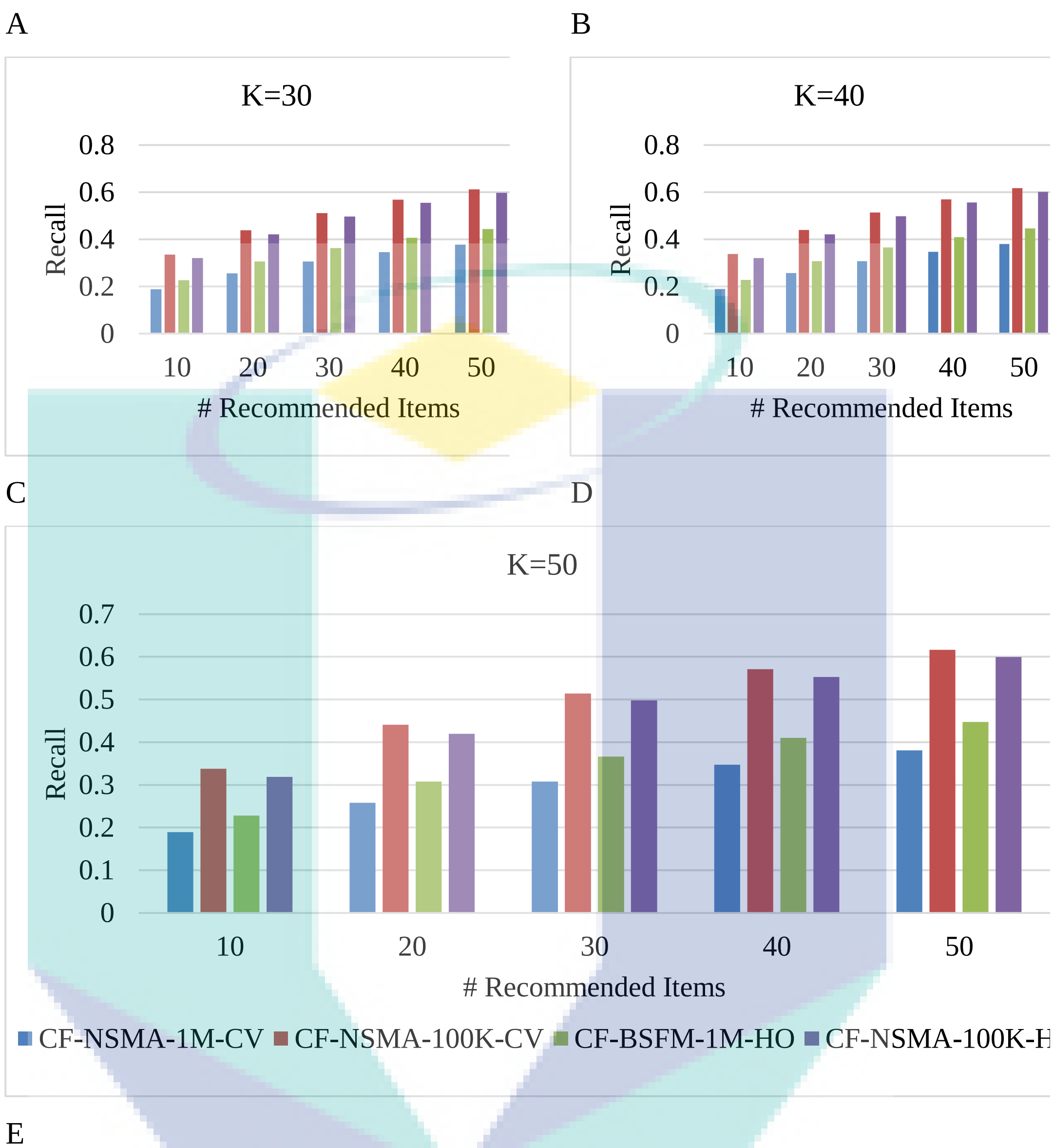


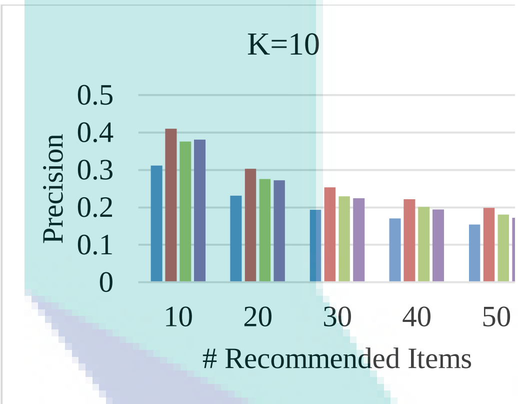

Figure 9 Recall of CF-NSMA (Holdout vs Cross-Validation method) using 100K & 1M MovieLens datasets

| | |
|---|---|
| Experiment three: | CF-NSMA results. |
| Technique: | CF-NSMA. |
| Input: | 100K & 1M MovieLens dataset. |
| | Denominator value 9. |
| | Size of neighbours K (10, 20, 30, 40, and 50). |
| | A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout & cross-validation methods. |
| The aim: | To present the performance accuracy of CF-NSMA using Holdout & Cross-validation for both datasets. |
| Metrics: | Precision metric. |

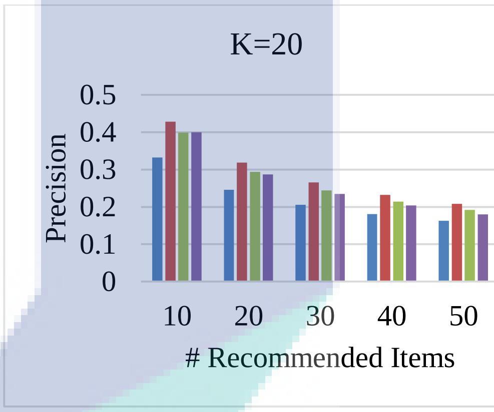
Description & observation

The presented bar charts (Figure 10) presents information about the precision rate of CF-NSMA using Holdout & cross-validation methods on both datasets. Where the subgraphs A, B, C, D, and E represent the precision rates according to the size of neighbours 10, 20, 30, 40, and 50, respectively. Where the size of recommended items (10, 20, 30, 40, and 50) is presented in the horizontal axis. In general, the size of recommendation set has more influence than the size of neighbours on the precision.

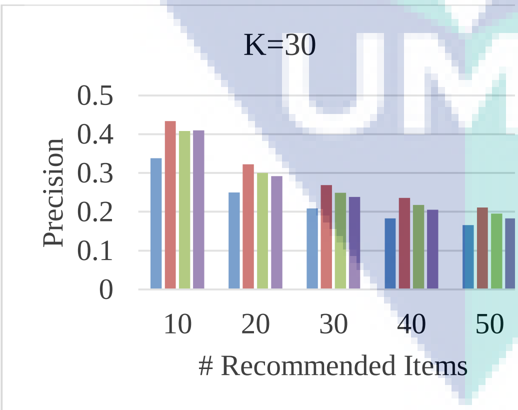
As it is observed, the precision rates were the highest when the number of recommended items was 10. In contrast, the worst rate was when the number of recommended items was 50 with whole cases. Therefore, we can see that there is a gradual decrease in the rate by around 20% from beginning point (size of recommendation 10) to the end (number of recommendations is 50) in all bar charts.

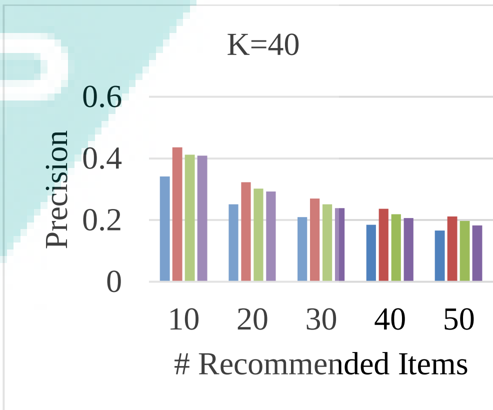



A

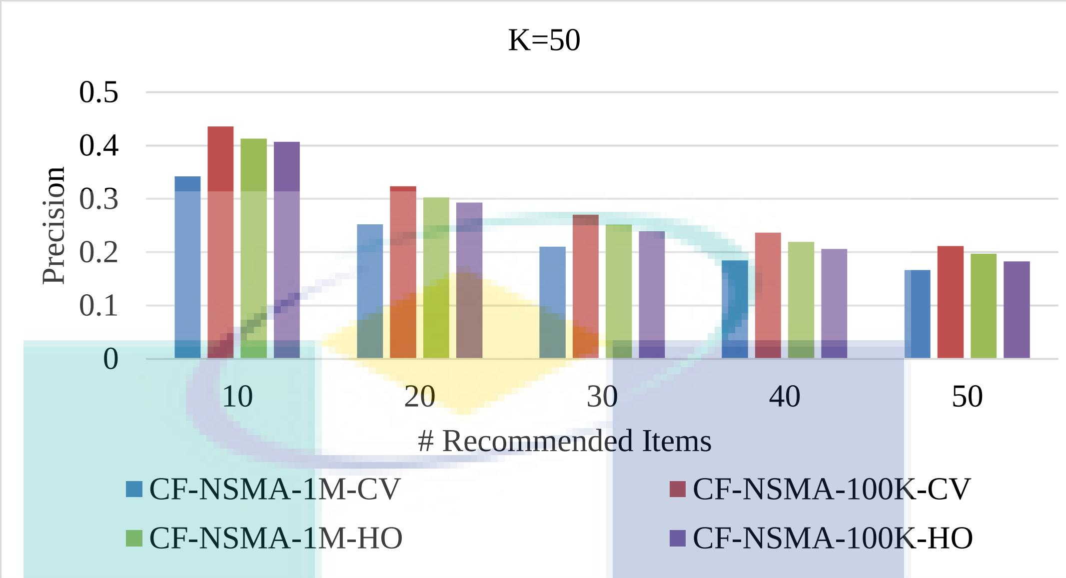



B



C D




E

Figure 10 Precision of CF-NSMA (Holdout vs Cross-Validation method) using 100K & 1M MovieLens datasets.

| | |
|---|---|
| Experiment four: | CF-NSMA results. |
| Technique: | CF-NSMA. |
| Input: | 100K & 1M MovieLens dataset.<br>Denominator value 9.<br>Size of neighbours K (10, 20, 30, 40, and 50).<br>A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout & cross-validation methods. |
| The aim: | To compare the performance accuracy of CF-NSMA using Holdout & Cross-validation for both datasets. |
| Metrics: | F-measure metric. |
| Description & observation | The bar graphs (Figure 11) enumerate the F-measure percentages using Holdout & cross-validation methods on 100k and 1M datasets using proposed CF-NSMA. The subgraphs A, B, C, D, and E, represent the F-measure rates regarding the size of neighbours 10, 20, 30, 40, and 50, respectively. Where the size of recommended items (10, 20, 30, 40, and 50) is presented in the horizontal axis.<br>As is presented in the subgraphs, the F-measure rates have a slight change in all subgraphs overall cases. It decreased by around 5% from beginning point (size of recommendation 10) to the end (size of recommendations is 50) overall subgraphs. |

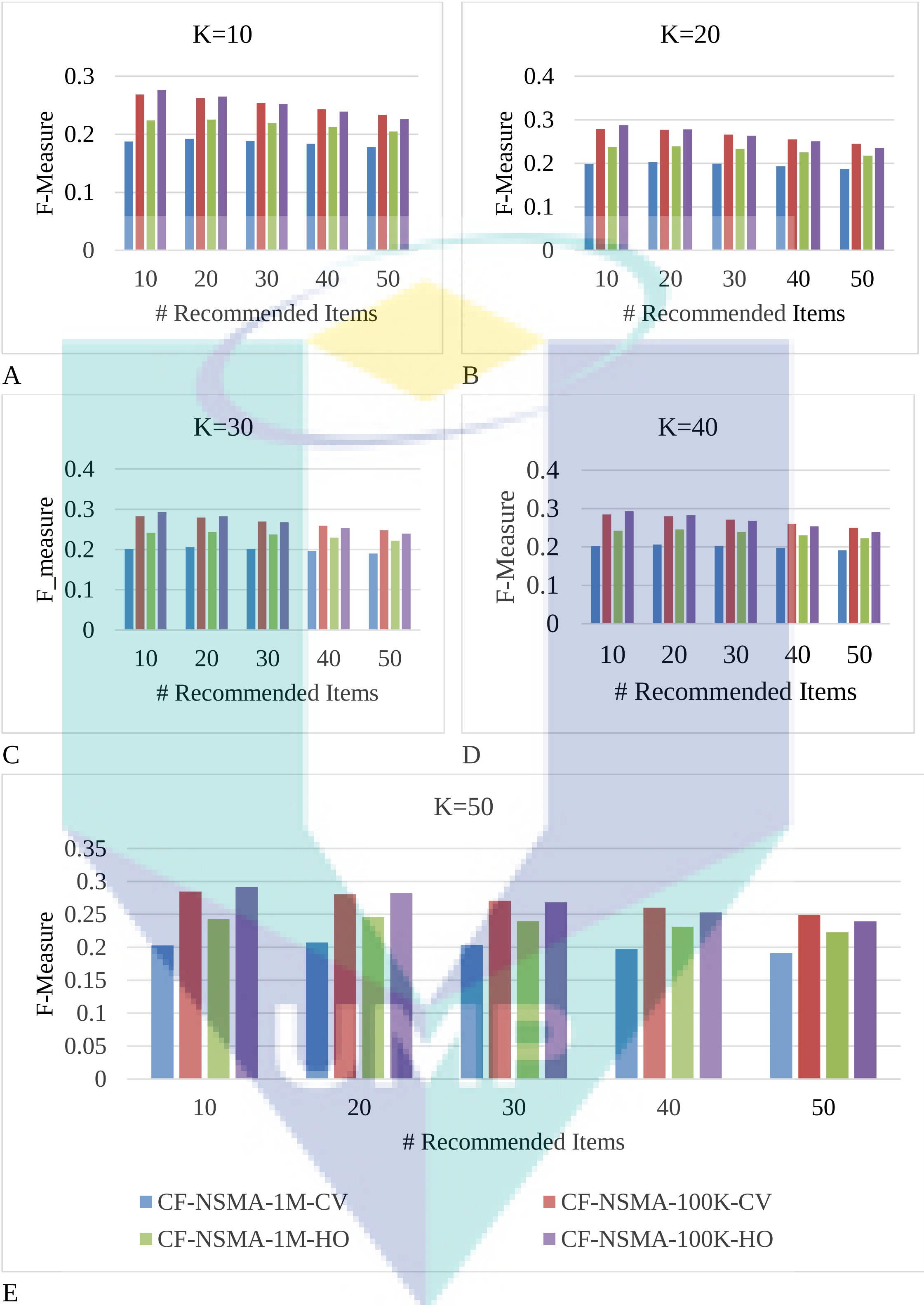
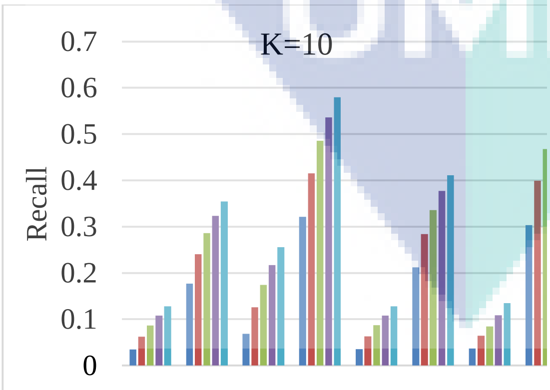


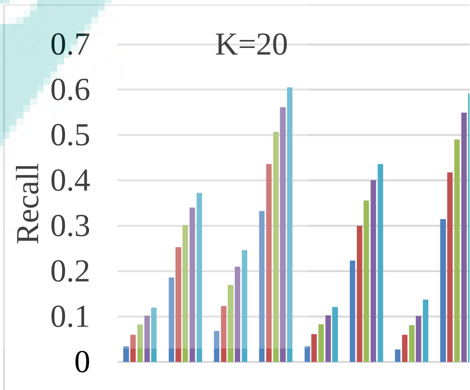

Figure 11 F-measure of CF-NSMA (Holdout vs Cross-Validation method) using 100K & 1M MovieLens datasets.

**F.5. Comparison CF-NSMA VS CF-BSF**

In this section, the results were presented in previous sections will be compared to show the enhancement that made by the MADM method. The comparison will be between CF-NSMA and CF-BSF in term of performance accuracy. Three metrics are used to make this comparison which are: recall, precision and F-measure.

| | |
|---|---|
| Experiment one: | Comparison CF-NSMA vs CF-BSF. |
| Technique: | CF-NSMA & CF-BSF. |
| Input: | 100K & 1M MovieLens dataset.<br>Denominator value 9.<br>Size of neighbours K (10, 20, 30, 40, and 50).<br>A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout & cross-validation methods. |
| The aim: | Test the impact of MADAM on performance accuracy. |
| Metrics: | Recall metric. |
| Description & observation | Figure 12 illustrates the comparison of recall between CF-NSMA and CF-BSF. Holdout & cross-validation methods are used on both datasets. The subgraphs A, B, C, D, and E, represent the recall rates according to the size of neighbours 10, 20, 30, 40, and 50, respectively. The horizontal axis represents a composite term (technique-dataset-splitting method). And the legend represented the number of recommended items (10, 20, 30, 40, and 50).<br>Overall, it can be seen that there is a significant improvement in the performance of recall using MADAM over all cases by around 25%. In both techniques, whenever the size of recommendation has increased the recall rates has a gradual improvement on all subgraphs. As can be seen in the given figure, the highest recall percentages were using CF-NSMA on 100K dataset. |



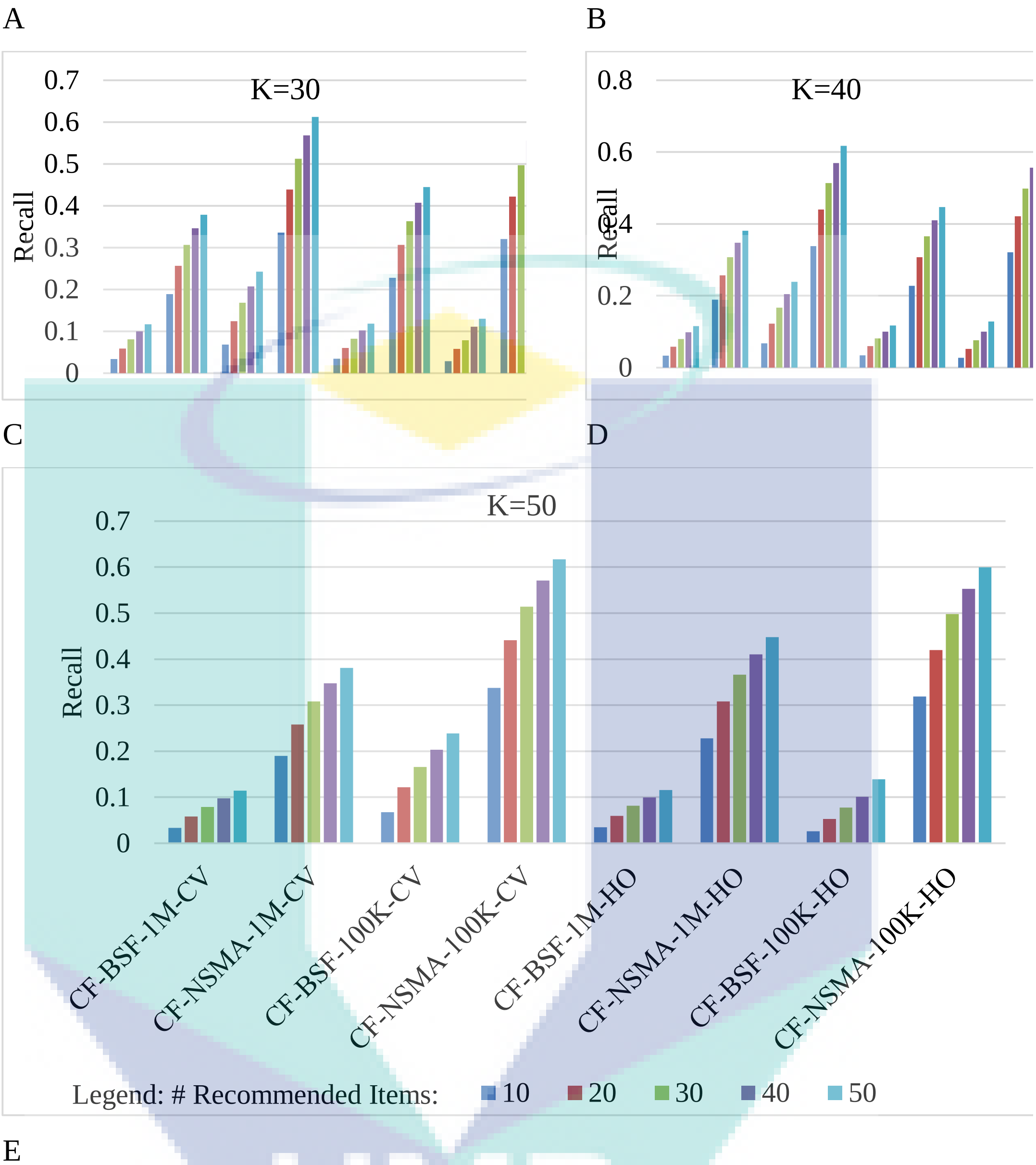
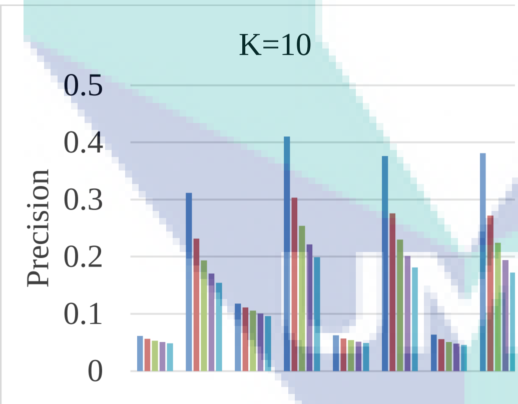
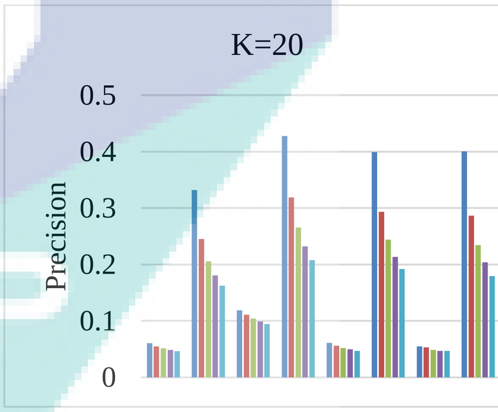



Figure 12 Recall comparison between CF-BSF and CF-NSMA (Holdout vs Cross-Validation method) using 100K & 1M MovieLens datasets.

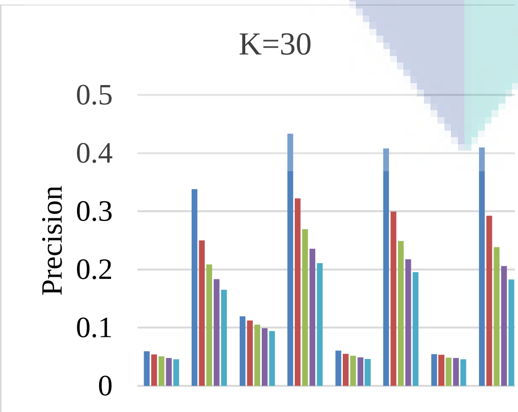
| | |
|---|---|
| Experiment two: | Comparison CF-NSMA vs CF-BSF. |
| Technique: | CF-NSMA & CF-BSF. |
| Input: | 100K & 1M MovieLens dataset.<br>Denominator value 9.<br>Size of neighbours K (10, 20, 30, 40, and 50).<br>A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout & cross-validation methods. |
| The aim: | Test the impact of MADAM on performance accuracy. |
| Metrics: | Precision metric. |
| Description & observation | Figure 13 illustrates the comparison of precision between CF-NSMA and CF-BSF. Two splitting methods are applied on 100K & 1M. The subgraphs A, B, C, D, and E, represent the precision rates according to the size of neighbours 10, 20, 30, 40, and 50, respectively. The horizontal axis represents a composite term (technique-dataset-splitting method). And the legend represented the number of recommended items (10, 20, 30, 40, and 50).<br>In general, there is a significant improvement in the performance of precision using MADAM over all cases by approximately 30%. In both techniques, it can be seen that whenever the size of recommendation has increased the precision rates has a gradual decrease in all subgraphs. To conclude, the highest precision percentages were in the CF-NSMA with ten recommended items. |

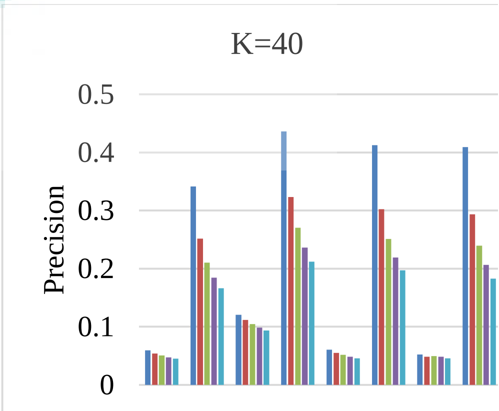



A

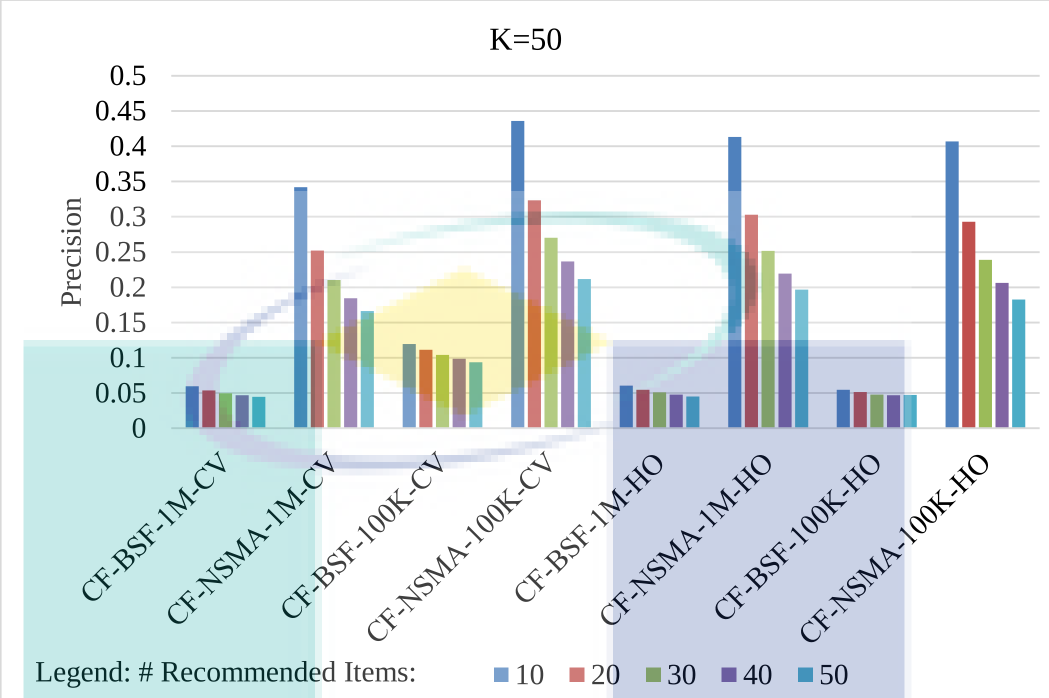



B



C D

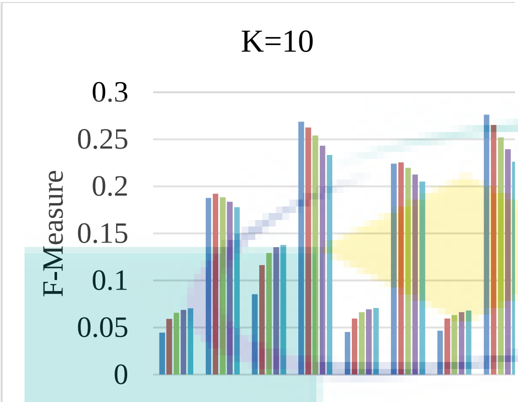



E

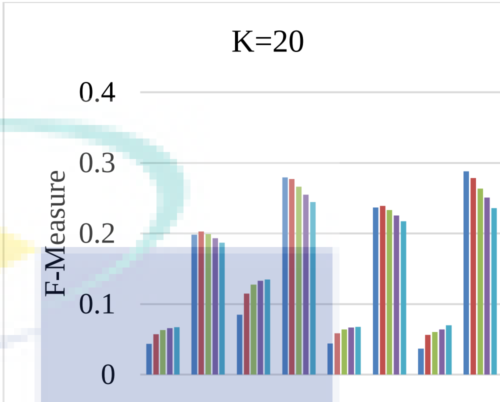
Figure 13 Precision comparison between CF-BSF and CF-NSMA (Holdout vs Cross-Validation method) using 100K & 1M MovieLens datasets.

| | |
|---|---|
| Experiment three: | Comparison CF-NSMA vs CF-BSF. |
| Technique: | CF-NSMA & CF-BSF. |
| Input: | 100K & 1M MovieLens dataset.<br>Denominator value 9.<br>Size of neighbours K (10, 20, 30, 40, and 50).<br>A number of recommended items (10, 20, 30, 40, and 50). |
| Splitting method: | Holdout & cross-validation methods. |
| The aim: | Test the impact of MADAM on performance accuracy. |
| Metrics: | F-measure metric. |
| Description & observation | The bar graphs (Figure 14) compare F-measure for CF-NSMA and CF-BSF the using Holdout & cross-validation methods on 100k and 1M datasets. The subgraphs A, B, C, D, and E, represent the F-measure rates according to the size of neighbours 10, 20, 30, 40, and 50, respectively. Where the subgraphs A, B, C, D, and E represent the F-measure rates according to the size of neighbours 10, 20, 30, 40, and 50, respectively. The horizontal axis represents a composite term (technique-dataset-splitting method). And the legend represented the number of recommended items (10, 20, 30, 40, and 50). |

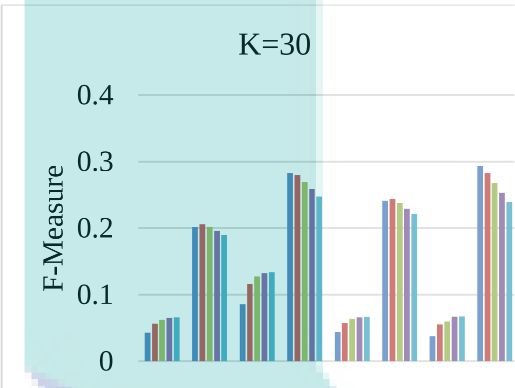
As is presented in the subgraphs, the F-measure rate of CF-NSMA has a noticeable improvement in all subgraphs overall cases. It enhanced by around 20% compare to CF-BSF.



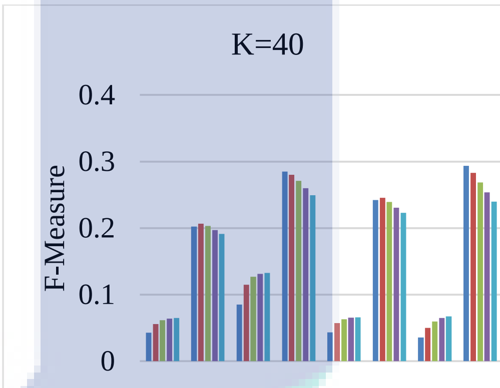

A



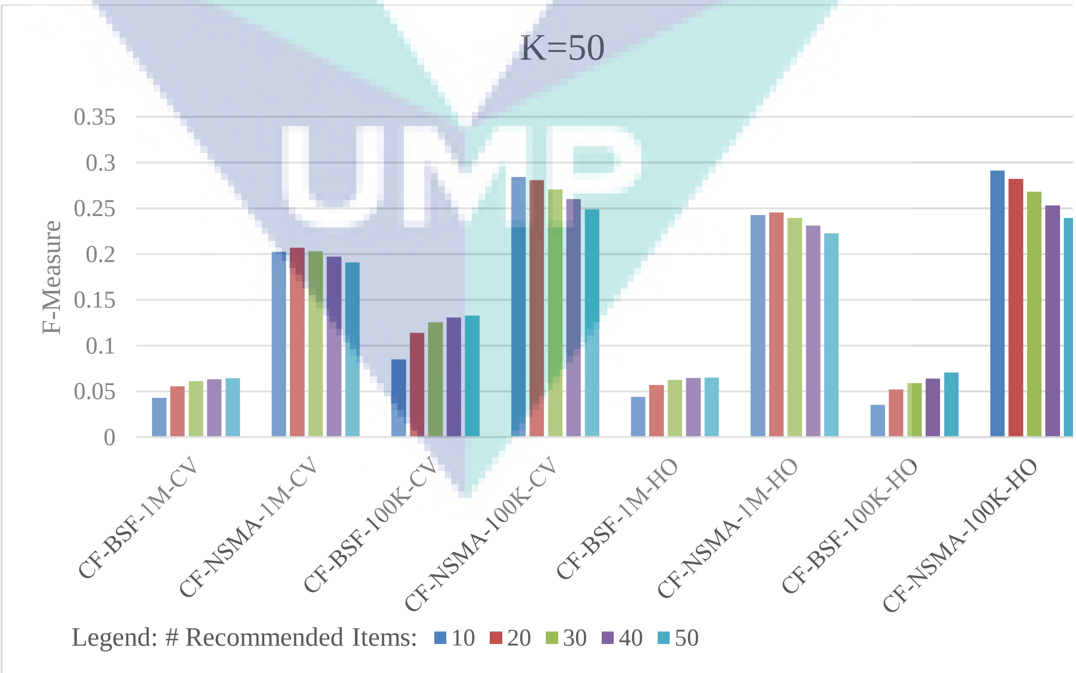

B




C




D



E

Figure 14 F-Measure comparison between CF-BSF and CF-NSMA (Holdout vs Cross-Validation method) using 100K & 1M MovieLens datasets.

## F.6. Comparison with common memory-based CF

In this part, to show the preceding of the new technique, several experiments were conducted, and the results will be compared with the widely used traditional memory-based CF methods using performance accuracy metrics. First, presenting the new similarity method CF-BSF by comparing its results with traditional similarity methods to prove the proposed similarity method precedes. Second, to show the notable improvement that made by the proposed technique CF-NSMA, the performance accuracy of CF-NSMA technique will be presented and compared to the traditional memory-based CF methods. Additionally, to show the positive effect of MADAM method on performance accuracy for traditional memory-based CF, the results of traditional memory-based CF methods using MADAM will be presented and compared with CF-NSMA technique. All experiments were conducted on 100K & 1M MovieLens public datasets. The holdout and cross-validation splitting methods were used to partition the dataset into training and testing sets. The comparison was regarding prediction and performance accuracy. Finally, this appendix will be ended by the conclusion section.

## F.7. Proposed Similarity Method Vs Traditional Similarity Methods

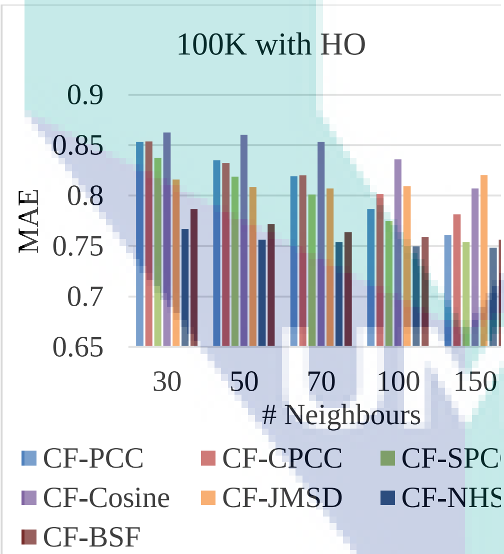
In this section, to show the improvement in prediction and performance accuracy results for the proposed similarity method (CF-BSF), it will be compared to traditional memory-based CF methods. The experiments were conducted on 100K & 1M MovieLens datasets, and the holdout and cross-validation methods were used to partition these datasets into training and testing sets. The results were presented in bar charts and line graphs which show the enhancement that made by CF-BSF in terms of prediction accuracy (MAE) and performance accuracy (Recall, Precision, and F-Measure).

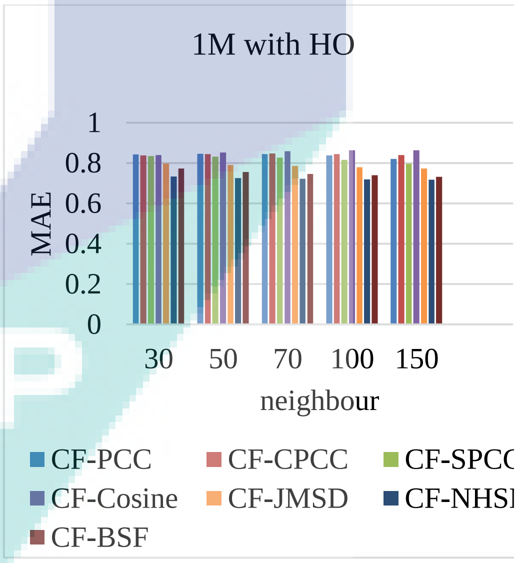
### I. Prediction Accuracy

The MAE metric will be used to compare the prediction accuracy. The holdout and cross-validation methods were applied on both datasets (100K & 1M MovieLens).

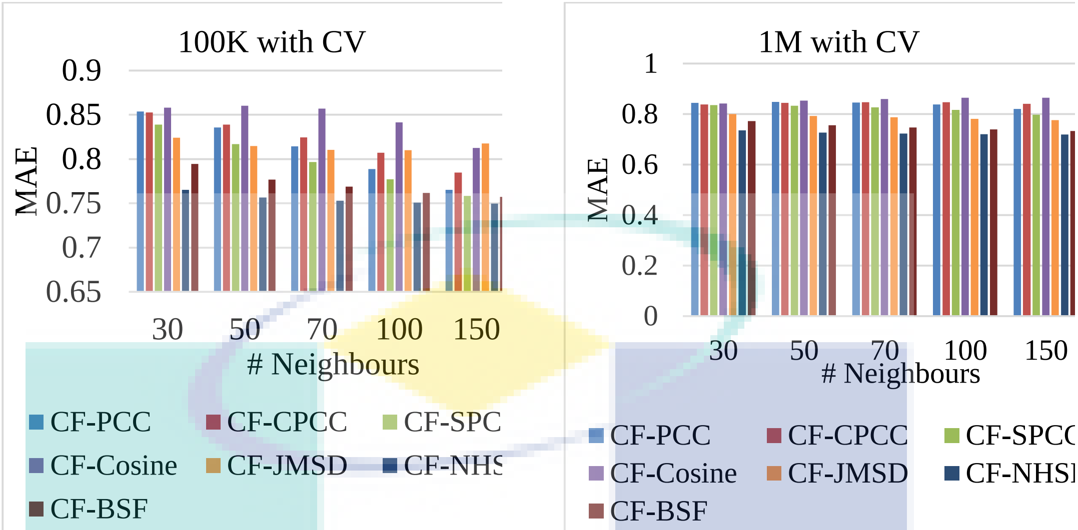
The bar charts in Figure 15 illustrate the MAE rate of CF-BSF compare to traditional memory-based CF methods (CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, and CF-NHSM). The size of neighbours was presented on the horizontal axis which has variation size: 30, 50, 70, 100, and 150, as shown in Figure 15. The subgraphs A, B, C, and D indicate MAE percentage for traditional memory-based CF methods and CF-BSF using holdout and cross-validation methods on 100K & 1M datasets, respectively.

In general, there is a slight improvement in the MAE value when the number of neighbours increases with the 1M dataset. Whilst, the MAE value has notable enhancement with100K dataset. Therefore, it is clear that the MAE values were the lowest when the size of neighbours is 150 overall cases. As it is presented in that bar graphs, compare to all comparative similarity methods except NHSM, which has a very small proportion improvement, the CF-BSF method has the lowest prediction accuracy in all cases. In contrast, the worst MAE rate was using CF-Cosine, CF_CPCC, and CF_SPCC, respectively.



A B




C D

Figure 15 Compare MAE between CF-BSF and traditional memory-based CF methods

### II. Performance Accuracy

In this section, the performance accuracy of proposed similarity method CF-BSF will be measured using three main metrics (Recall, Precision, and F-measure). The experiments were conducted on 100K & 1M datasets using holdout and cross-validation partition techniques. The results that shown in all Figure 16 to Figure 27 next represent the averaging of various size of neighbours (10, 20, 30, 40 and 50).

#### *i. Recall Metric*

One of metrics used to measure the performance accuracy of a recommender system is the recall measure. In this subsection, it was used to compare the performance accuracy of CF-BSF via applying holdout and cross-validation methods on 100K & 1M MovieLens.

Figure 16 to Figure 19 illustrate the comparison of recall metric results between CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, CF-NHSM and the proposed CF-BSF similarity methods. Where Figure 16 & Figure 17 present the comparison recall rate when the holdout method was used with 100K & 1M datasets, respectively. And Figure 18 & Figure 19 show the recall percentage comparison when the cross-validation was used on 100K & 1M datasets, respectively. The horizontal axis presents the number of recommendations (10, 20, 30, 40, and 50).

In general, for all methods, the rate rose gradually to reach to the highest rate when the number of recommendations was 50. As it can be seen from those graphs, the recall rate of CF-BSF was the highest overall variation number of recommended items in all figures. Whereas, the recall rates of PCC and its derivatives methods were the lowest. According to the CF-Cosine recall in Figure 16, it was good, but it decreases to be bad in the rest figures. To sum up, the recall rate improves as the number of recommended items increases.

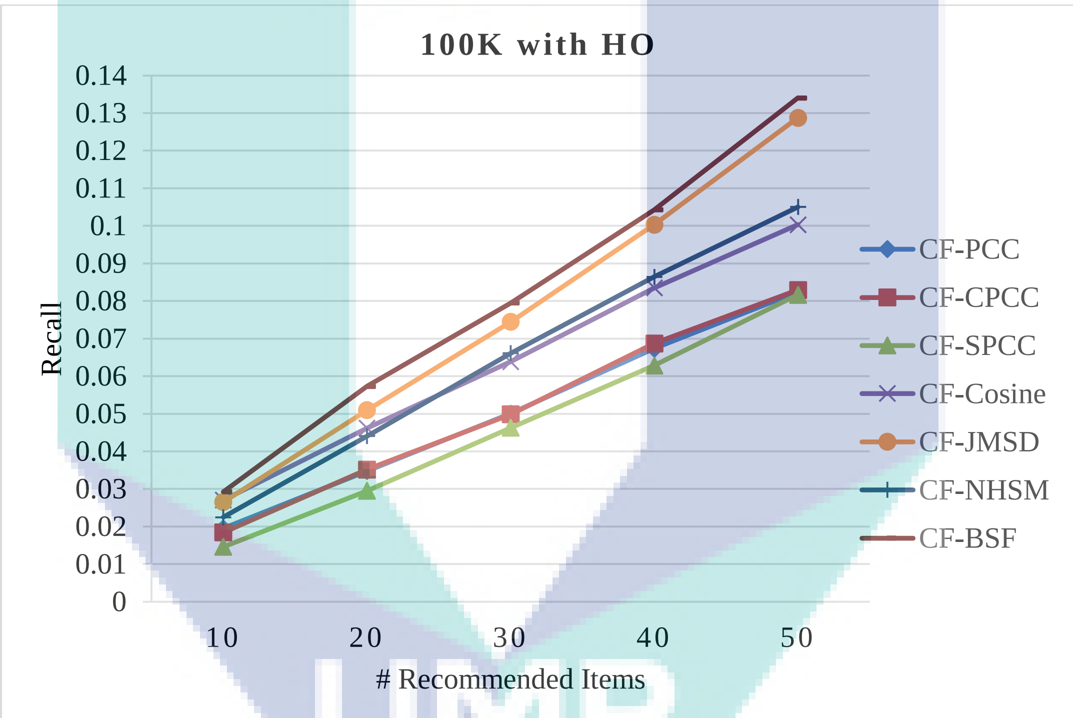



Figure 16 Recall measure vs various number of recommendations on 100K, Holdout.

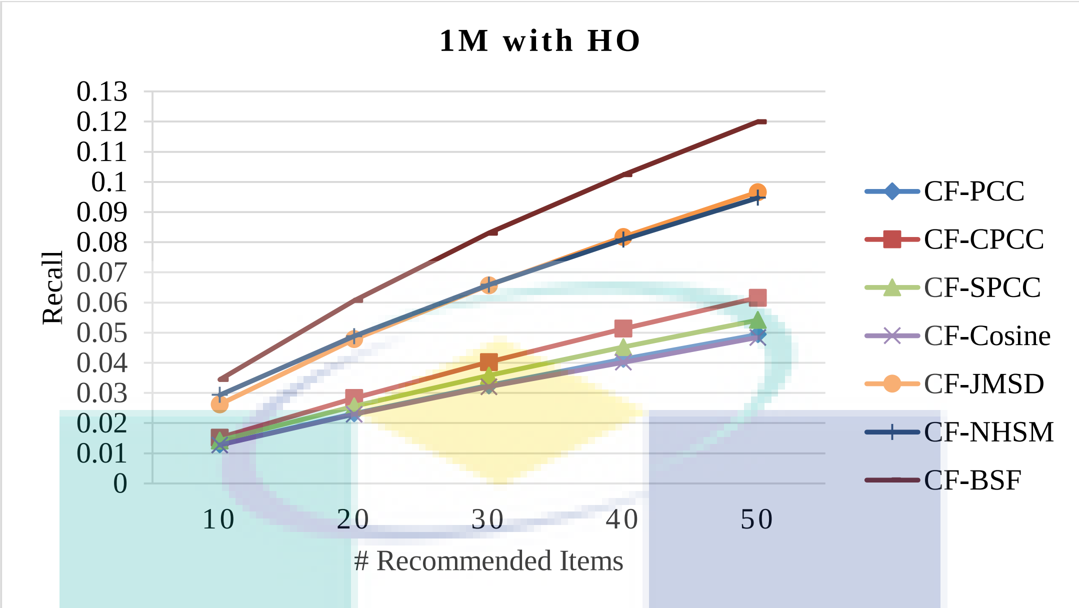



Figure17 Recall measure vs various number of recommendations on 1M, Holdout.

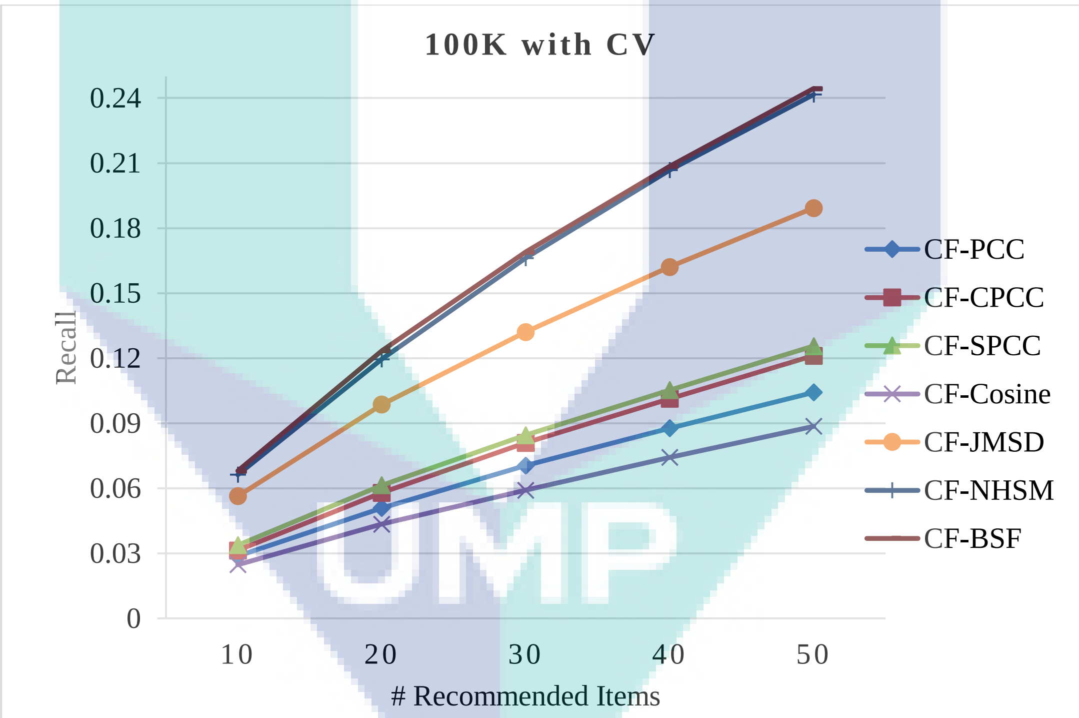



Figure 18 Recall measure vs various number of recommendations on 100K, Cross-validation.

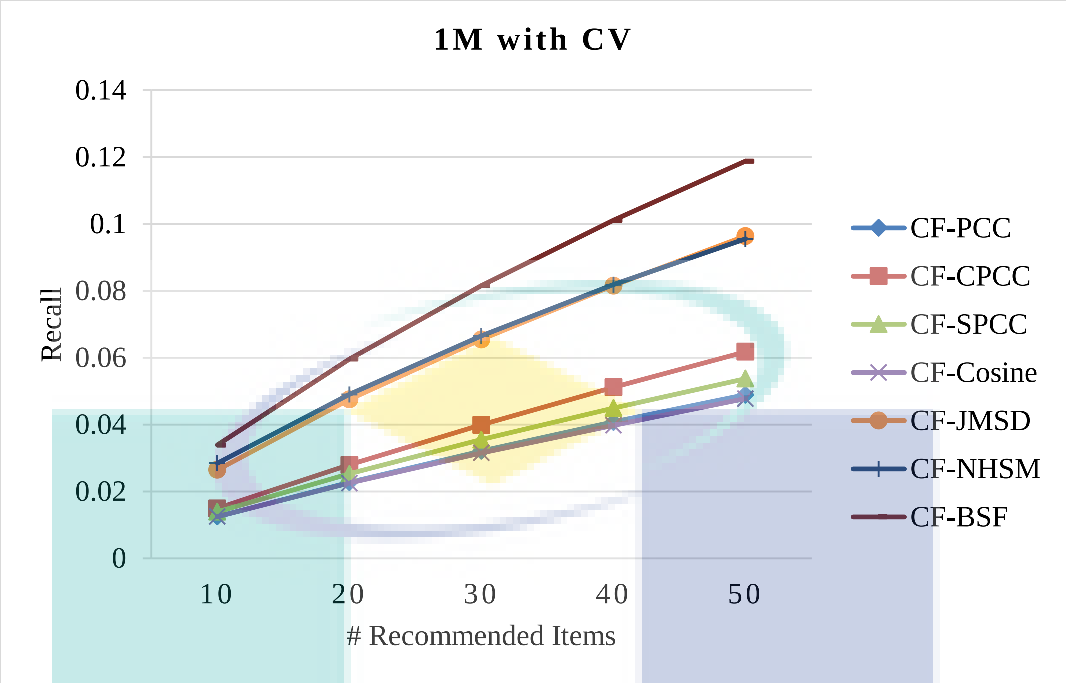



Figure 19 Recall measure vs various number of recommendations on 1M, Cross-validation.

*ii. Precision Metric*

The second metric will be used to measure the performance accuracy of a recommender system is the precision measurement. The holdout and cross-validation methods were applied on 100K and 1M MovieLens to compare the precision accuracy of CF-BSF with traditional memory-based CF methods.

The graphs (Figure 20 to Figure 23) give comparative information about the precision rate for CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, CF-NHSM and the proposed CF-BSF similarity methods using holdout and cross-validation partition methods on 100K & 1M datasets, respectively. Where, the number of recommended was equal 10, 20, 30, 40, and 50 items and is presented on the x-axis.

At first glance, it is clear that the precision rate, for all methods, declines from start point, when the number of recommendations was 10, to reach to lowest rate when the number of the recommendations was 50 in all graphs. As it can be seen from the graphs, regarding CF-BSF method, the rate of precision at least dropped gradually by around 0.01 from start point to the end in all figures. Likely, the precision rate of CF-JMSD and CF-NHSM have an approximately same dropped percentage. While, the other methods (CF-PCC, CF-SPCC, CF-CPCC, and CF-Cosine) have a low level of percentage with a little less plummeted overall variation number of recommended items. In summary, the precision of CF-BSF was the highest in all graphs over all cases.

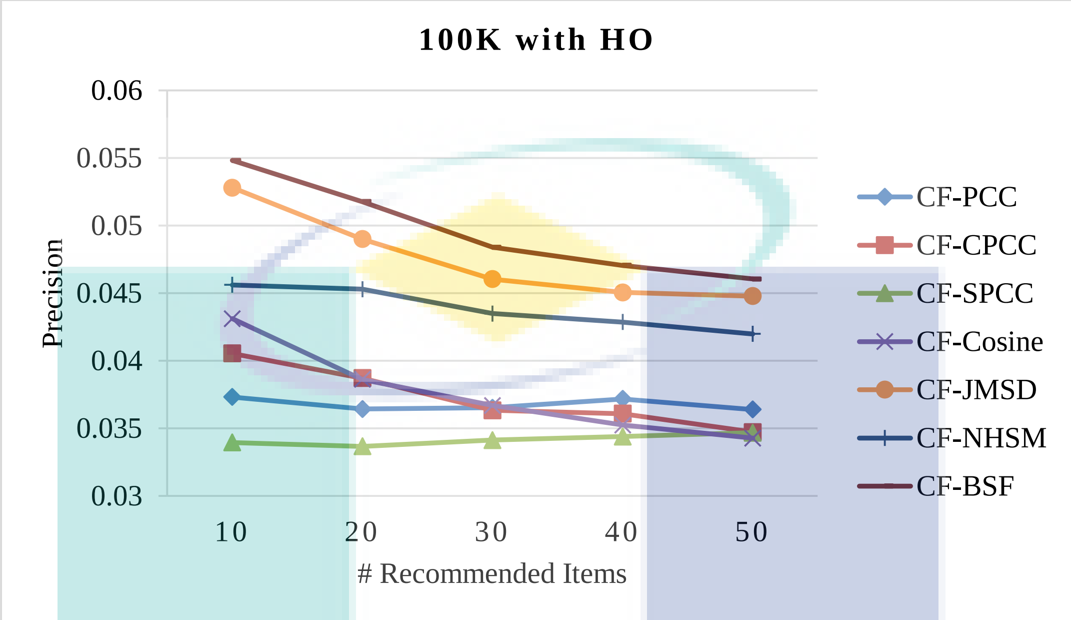



Figure 20 Precision measure vs various number of recommendations on 100K, Holdout.

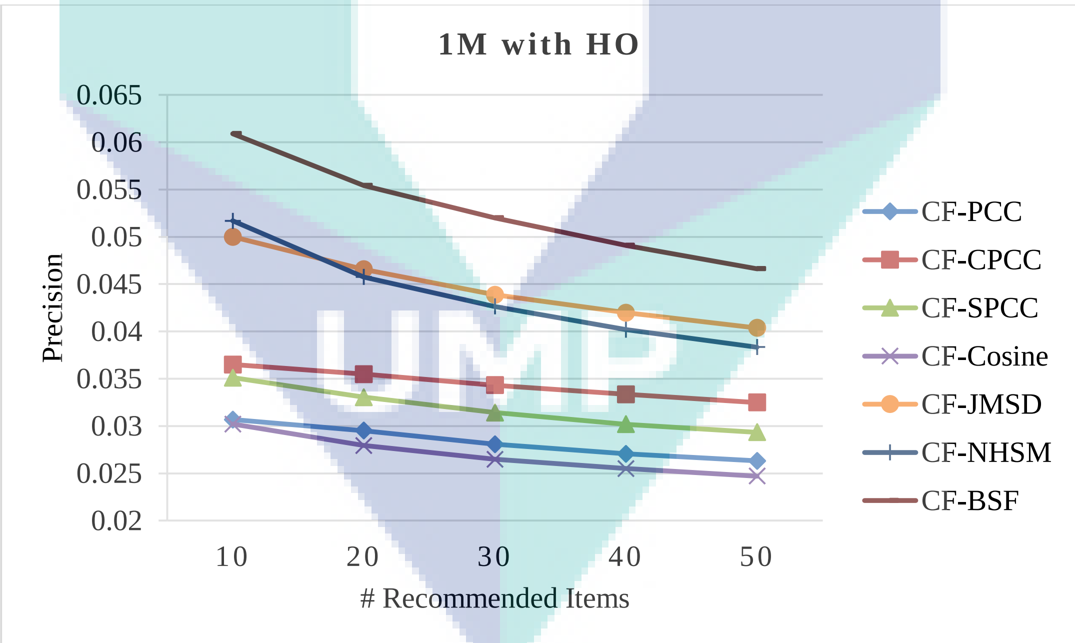



Figure 21 Precision measure vs various number of recommendations on 1M, Holdout.

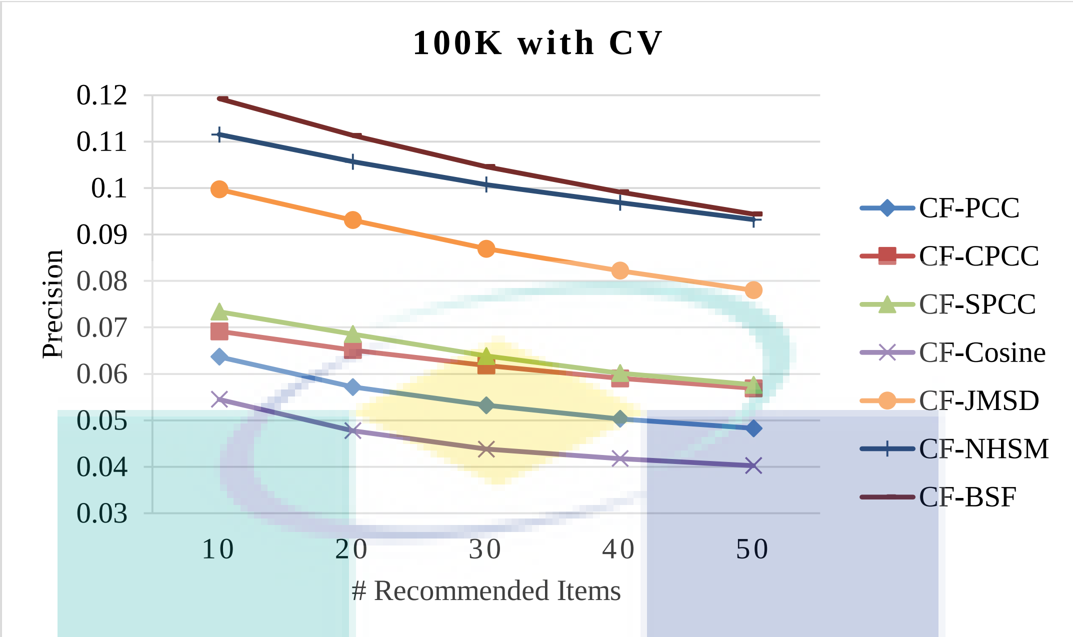



Figure 22 Precision measure vs various number of recommendations on 100K, Cross-validation.

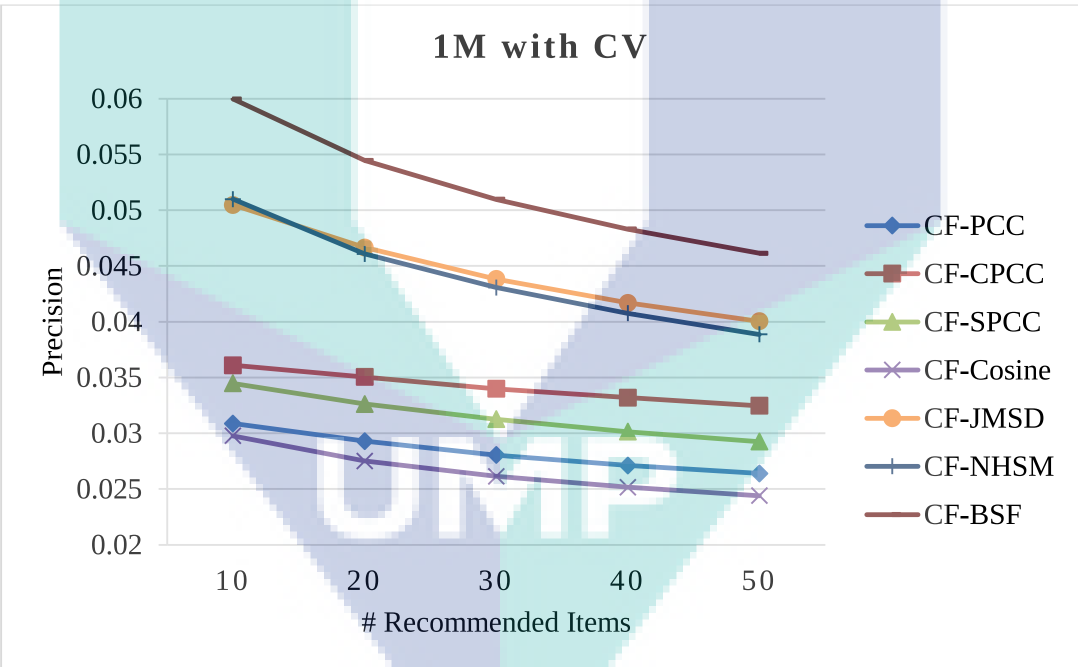



Figure 23 Precision measure vs various number of recommendations on 1M, Cross-validation.

*iii. F-Measure Metric*

F-measure metric is a combined metric of precision and recall, it gives different information, the weighted mean of precision and recall, compared to precision and recall.

The holdout and cross-validation methods were applied on 100K & 1M MovieLens to find the F-measure value of CF-BSF.

Figure 24 to Figure 27 show the percentage of F-measure for CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, CF-NHSM and the proposed CF-BSF similarity methods using holdout and cross-validation partition methods on 100K & 1M datasets, respectively. The number of recommended items are presented on the x-axis.

It has been observed from the graphs that, for all methods, there is a significant rise in the F-measure percentage in all methods from start point when the size of recommended items was 10 to 30 however after that it raises slightly within the next two sizes of recommendations unto maximum percentages. To conclude that, the F-measure rate of CF-BSF better than CF-NHSM & CF-JMSD by around 0.05 in all figures, where CF-NHSM & CF-JMSD have the highest rate when compared to other traditional methods (CF-PCC, CF-SPCC, CF-CPCC, and CF-Cosine). Therefore, the F-measure rate of CF-BSF was the highest in all figures overall cases.

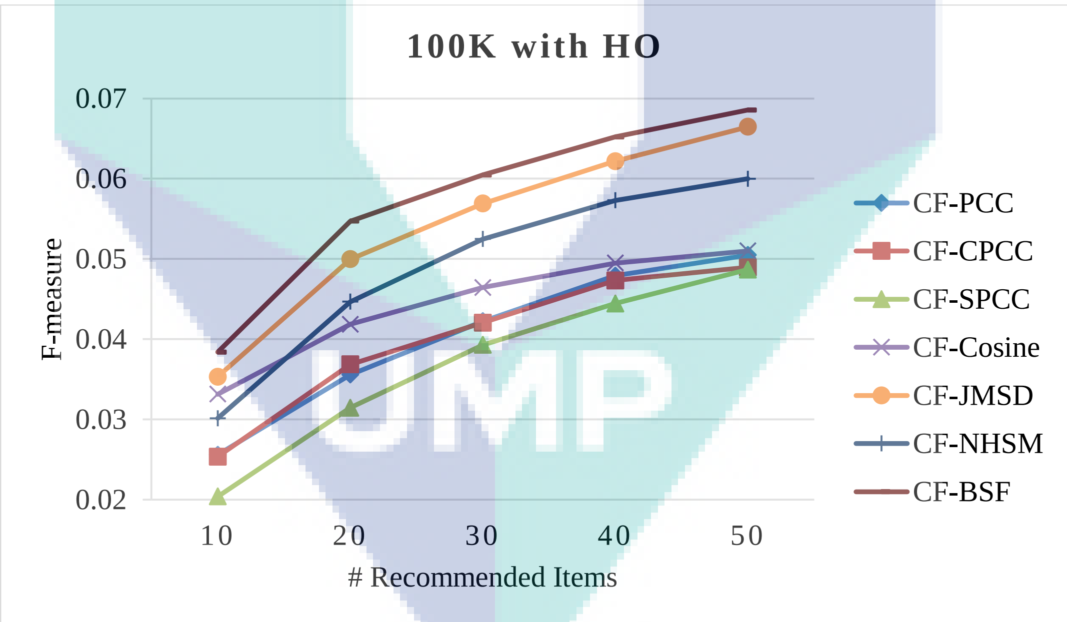



Figure 24 F-measure measure vs various number of recommendations on 100K, Holdout.

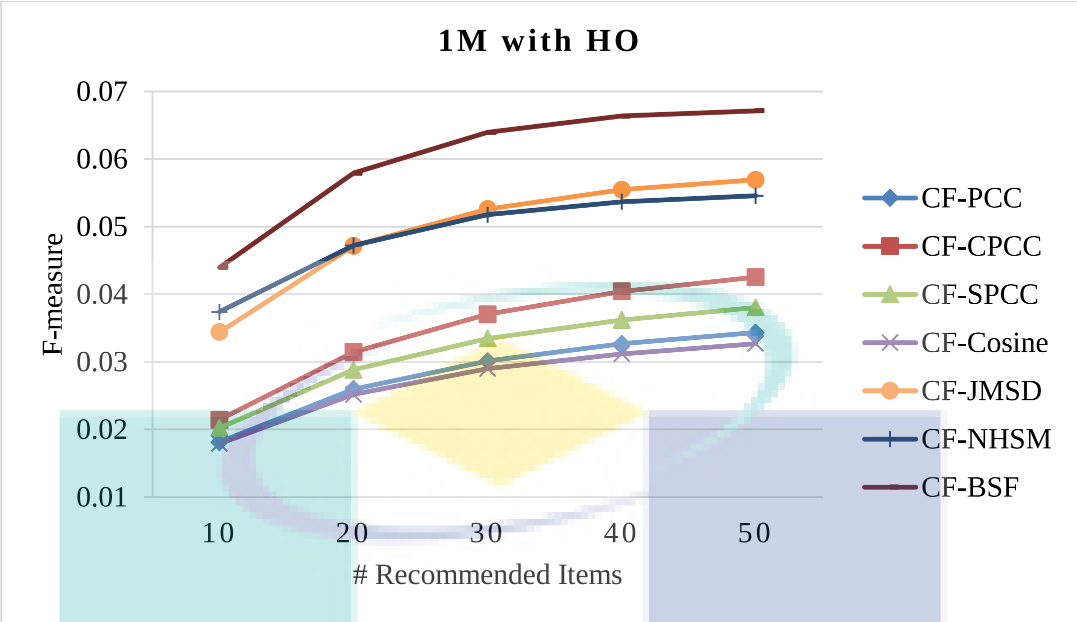



Figure 25 F-measure measure vs various number of recommendations on 1M, Holdout.

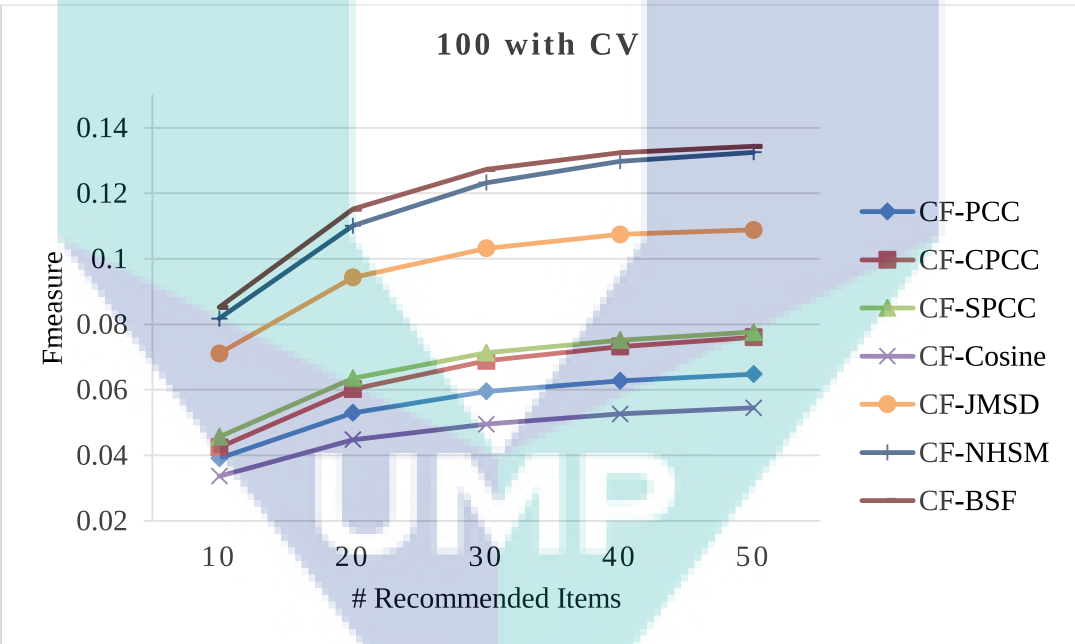



Figure 26 F-measure measure vs various number of recommendations on 100K, Cross-validation.

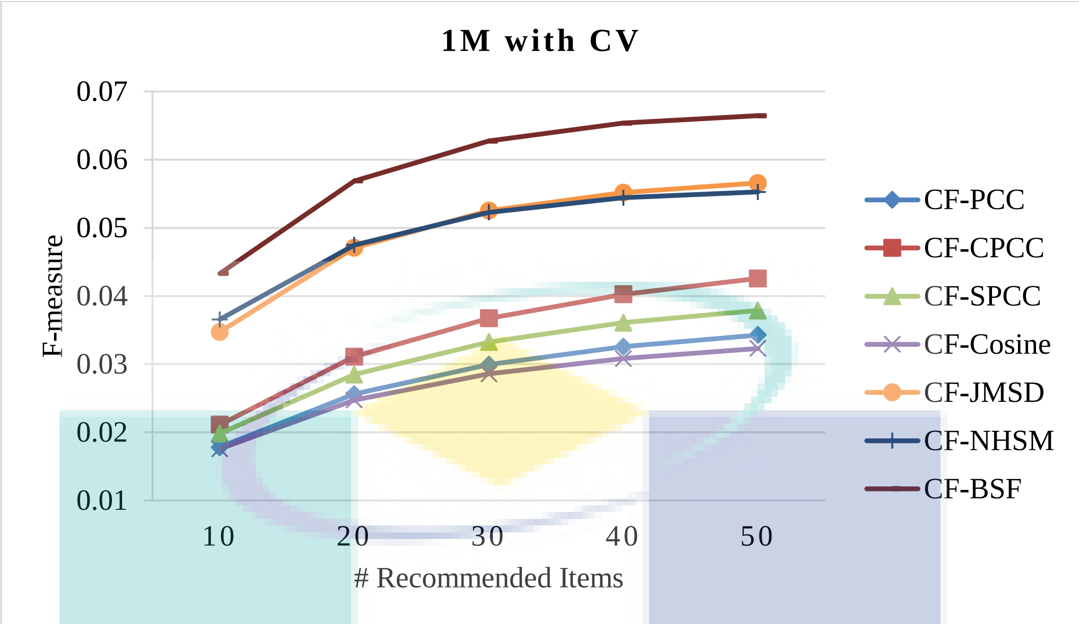



Figure 27 F-measure measure vs various number of recommendations on 1M, Cross-validation.

### F.8. CF-NSMA Proposed Technique Vs Traditional Memory-Based CF Methods

In this section, the proposed technique CF-NSMA will be compared with traditional memory-based CF methods to show the preceding of its performance accuracy through conducting some experiments. Holdout and cross-validation partition methods were used on 100K & 1M MovieLens datasets. The results were presented in the line graphs which show the notable improvement that made by CF-NSMA when compared to traditional memory-based CF methods. Three metrics were used to measure the performance accuracy which are Recall, Precision, and F-Measure. The results that shown in all figures represent the performance accuracy by averaging variation size of neighbours (10, 20, 30, 40 and 50).

#### I. Recall metric

Figure 28 to Figure 31 present the comparison of recall between CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, CF-NHSM and proposed CF-NSMA technique. Figure 28 and 29 represent the recall when holdout partition method was used on 100K & 1M datasets, respectively. While, Figure 30 and Figure 31 demonstrate the percentage of recall when cross-validation partition method was used on 100K & 1M datasets, respectively. The number of recommendations was 10, 20, 30, 40, and 50 which represented on the horizontal axis.

In general, the rate of CF-NSMA rose significantly to reach to the highest rate when the number of recommendations was 50 in all figures. Unlike the traditional methods, it increased slowly. As can be seen from those graphs, the recall rate of CF-NSMA has a significant improvement when compared to the rate of traditional methods. Which was around 0.25 and 0.3 when the size of recommendation is 10 and has risen to over 0.35 and 0.55 with 1M & 100K datasets, respectively. Whereas, the best recall rate of traditional methods did not exceed 0.06 and 0.13 with both datasets in all figures except Figure 30. In this figure, the recall of CF-NHSM and CF-JMSD are around 0.24 and 0.18, respectively, when the number of recommended items was 50.

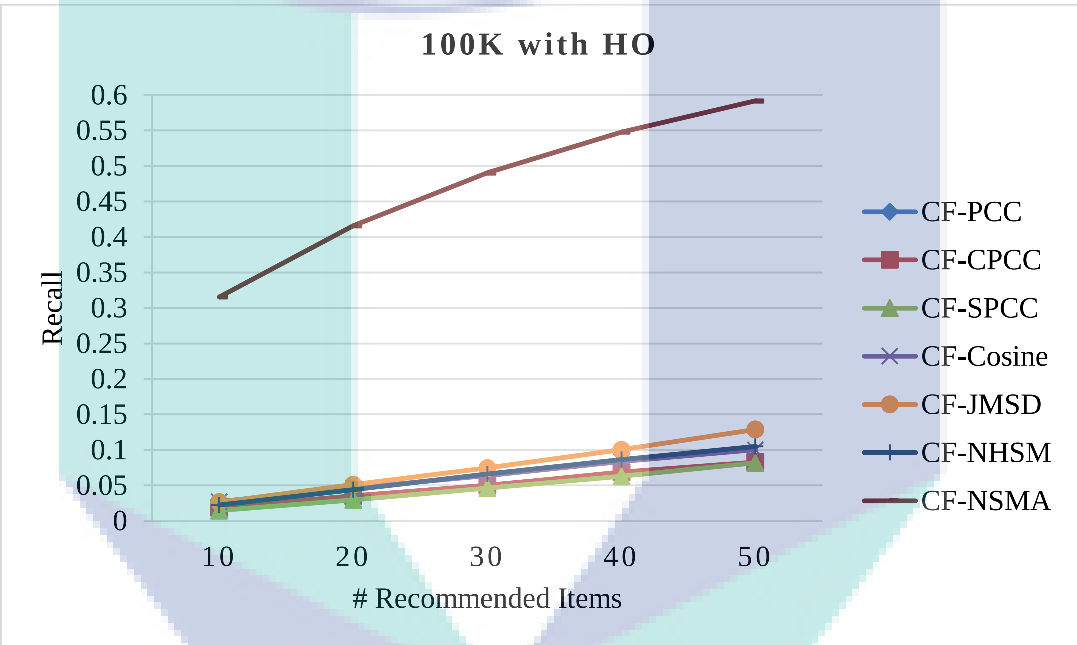



Figure 28 Recall measure vs various number of recommendations on 100K, Cross-validation.

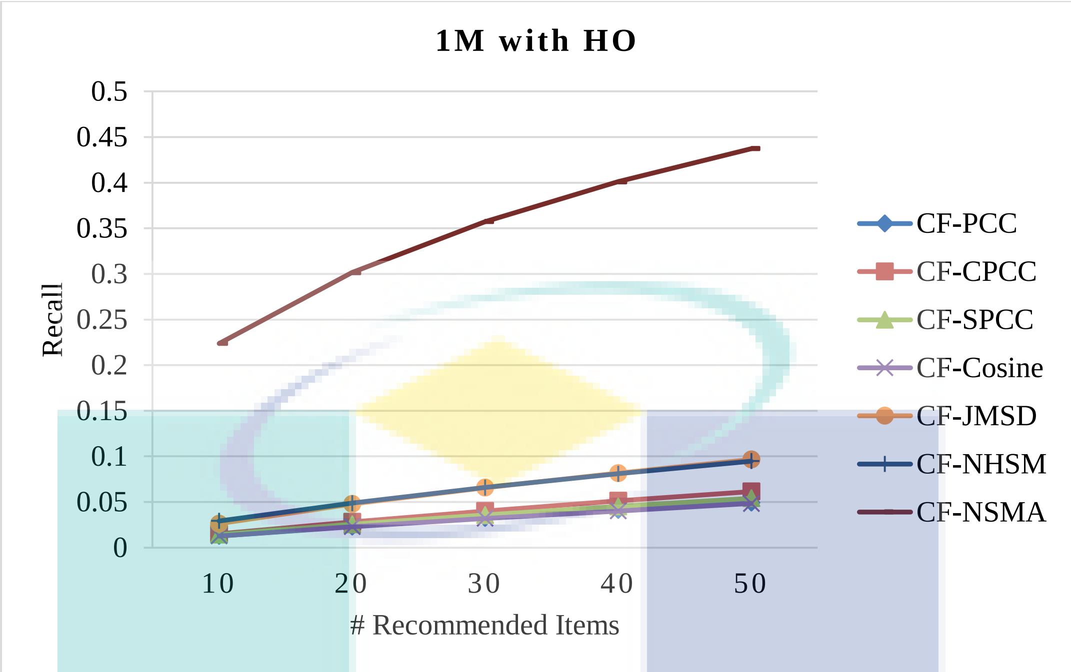



Figure 29 Recall measure vs various number of recommendations on 1M, Holdout.

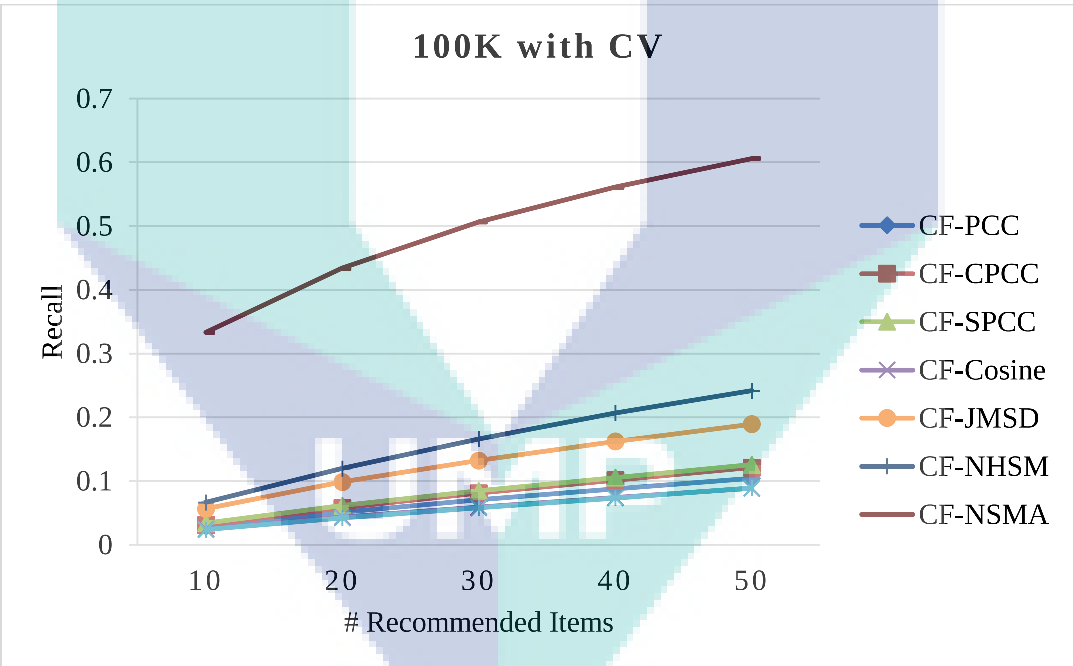



Figure 30 Recall measure vs v number of recommendations on 100K, Cross-validation.

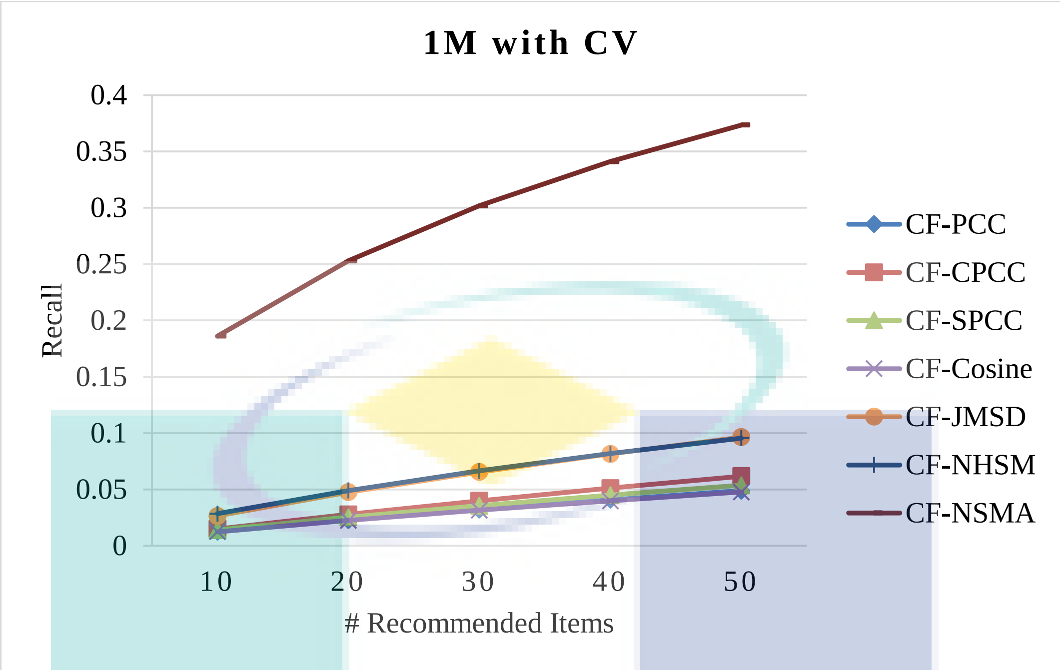



Figure 31 Recall measure vs various number of recommendations on 1M, Cross-validation

### II. Precision Metric

Figure 32, 33, 34, and 35 give information about the precision rate for CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, and CF-NHSM and proposed CF-NSMA technique. Where Fig. 32 and 33 depict the comparison of precision when holdout partition method was used on 100K & 1M datasets, respectively. While Fig. 34 and 35 express the comparison of precision percentage when cross-validation partition method was used on 100K & 1M datasets, respectively. The horizontal line present variation number of recommendations size which was 10, 20, 30, 40, and 50.

In general, the rate of CF-NSMA has a notable improvement when compared to traditional methods. The precision value was around 0.4 when the number of recommended items was 10, and dropped to around 0.20 when the number of recommendations increased to 50 in all line graphs over all cases. However, there is a big difference between its precision rate and the traditional methods precision rate. Moreover, it can be seen that the precision rates of all traditional methods do not exceed the 0.12 at its best. Therefore, the proposed technique has a very large proportion precision rate compare to traditional rate which keeps it the highest accuracy with a big improvement.

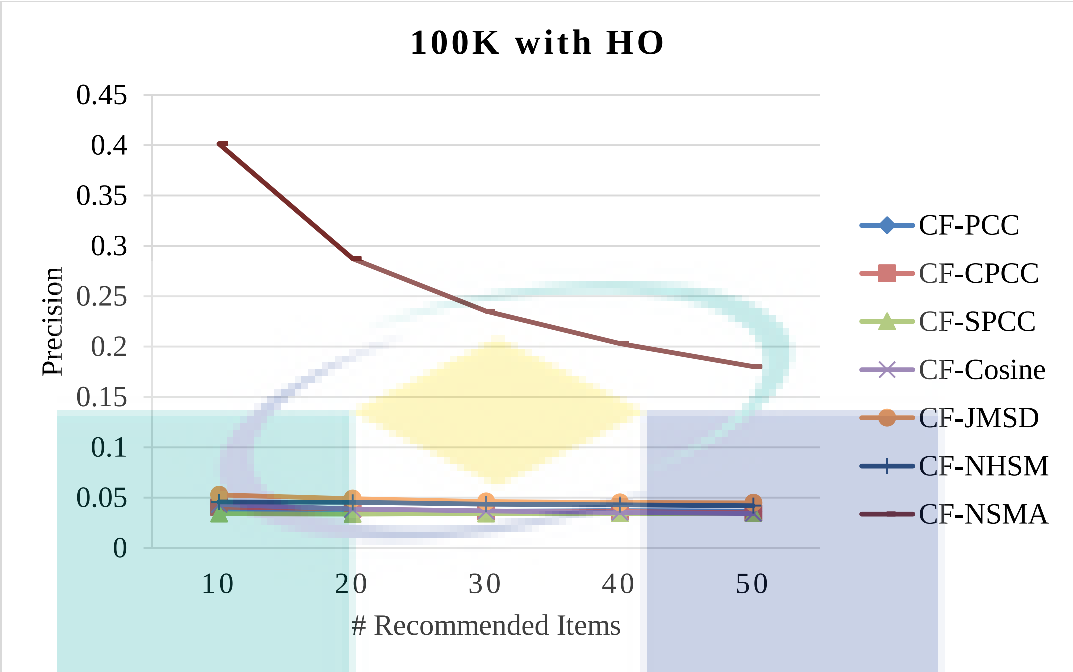



Figure 32 Precision measure vs various number of recommendations on 100K, Holdout.

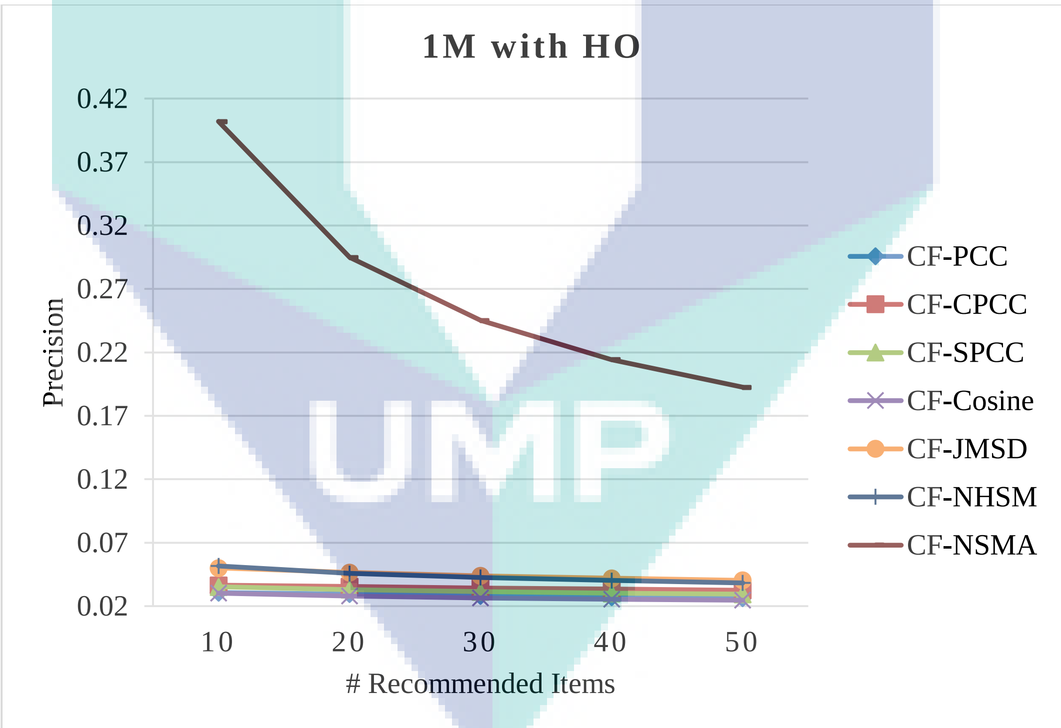



Figure 33 Precision measure vs various number of recommendations on 1M, Holdout.

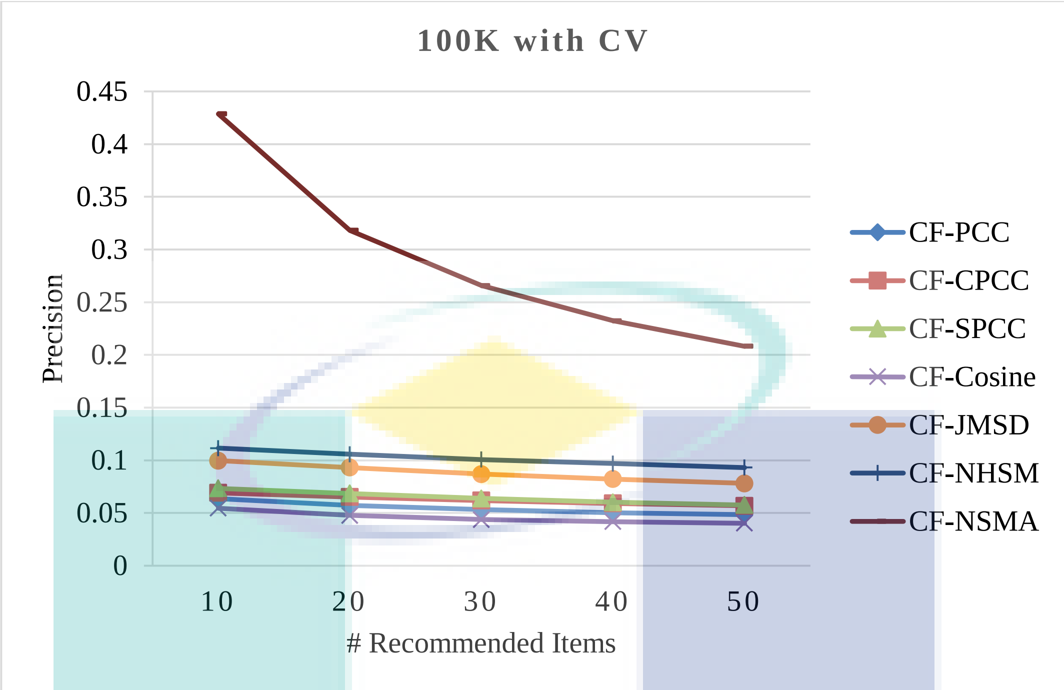



Figure 34 Precision measure vs various number of recommendations on 100K, Cross-validation.

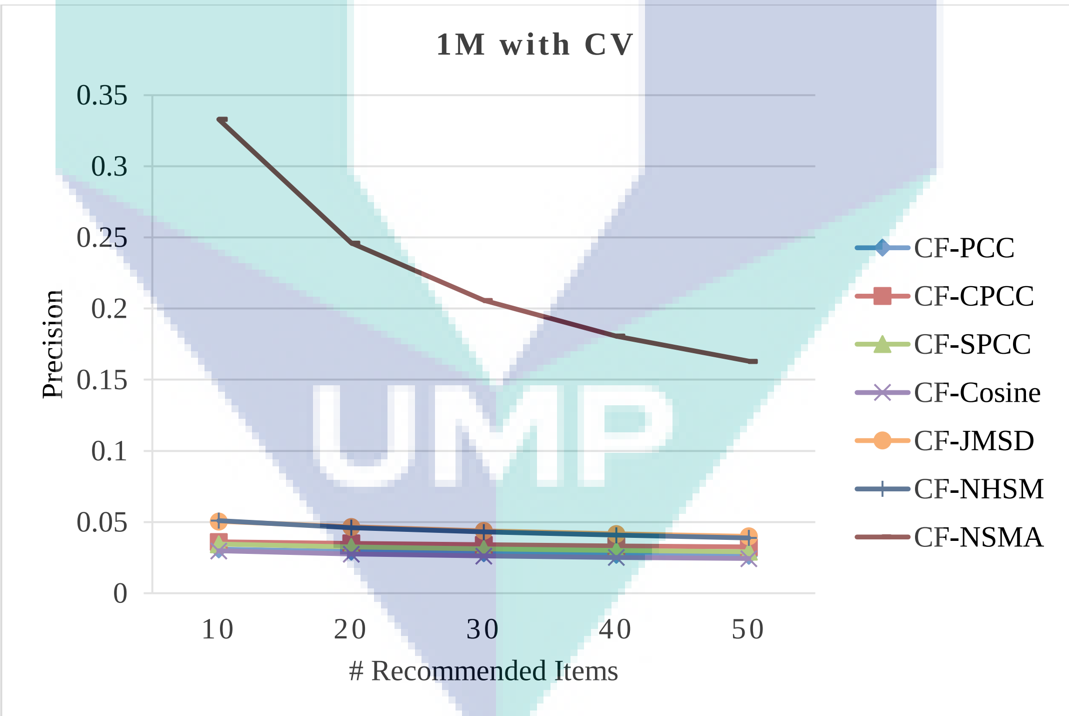



Figure 35 Precision measure vs various number of recommendations on 1M, Cross-validation

### III. F-Measure Metric

Figure 36 to Figure 39 compare the proportion of F-measure between CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, CF-NHSM and CF-NSMA technique. Figure 36 & Figure 37 indicate the comparison of F-measure when holdout splitting method was used on 100K & 1M datasets, respectively. While, Figure 38 and Figure 39 show the comparison of F-measure rate when cross-validation partition method was used on 100K & 1M datasets, respectively. The horizontal line present variation number of recommendations size which was 10, 20, 30, 40, and 50.

At the onset, it is clear that the F-measure rate of CF-NSMA has a significant majority improvement compare to the rate of traditional methods. The F-measure value was over 0.24 when the number of recommended items was 10 in all figures except Figure 39 which was around 0.20. Then, the rate of CF-NSMA dropped by approximately 0.04 and 0.02 when the size of recommendations increased to 50 with 100K and 1M datasets used, respectively. However, it is still a big difference between its F-measure rate and the traditional methods F-measure rate. In contrast, from Figure 38 the highest percentage of the traditional methods was approximately 0.1 & 0.13 for CF-NHSM and CF-JMSD, respectively. While, the maximum rate of CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, and CF-JMSD do not exceed the 0.07 in all figures. To conclude, the proposed technique has the highest F-measure rate with approximately more than three-quarter improvement.

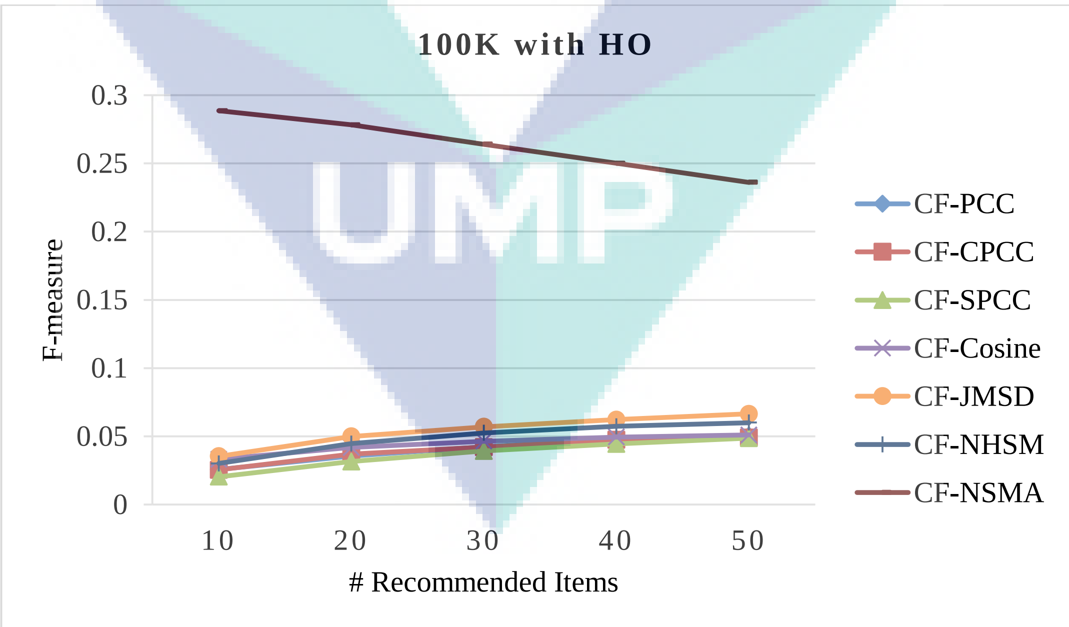



Figure 36 F-measure measure vs various number of recommendations on 1M, Holdout.

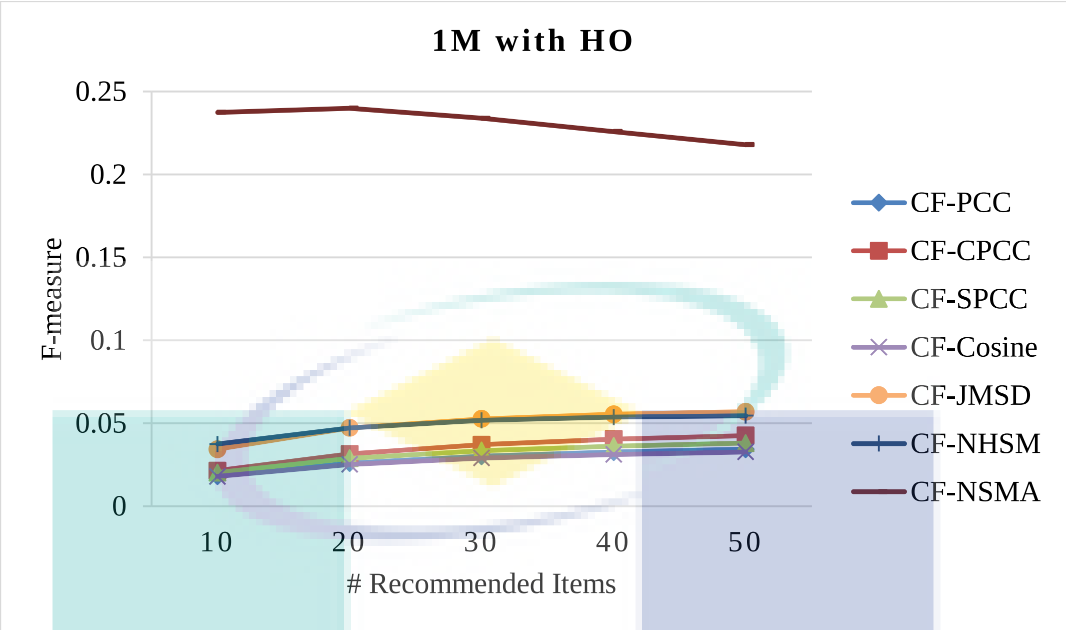



Figure 37 F-measure measure vs various number of recommendations on 1M, Holdout.

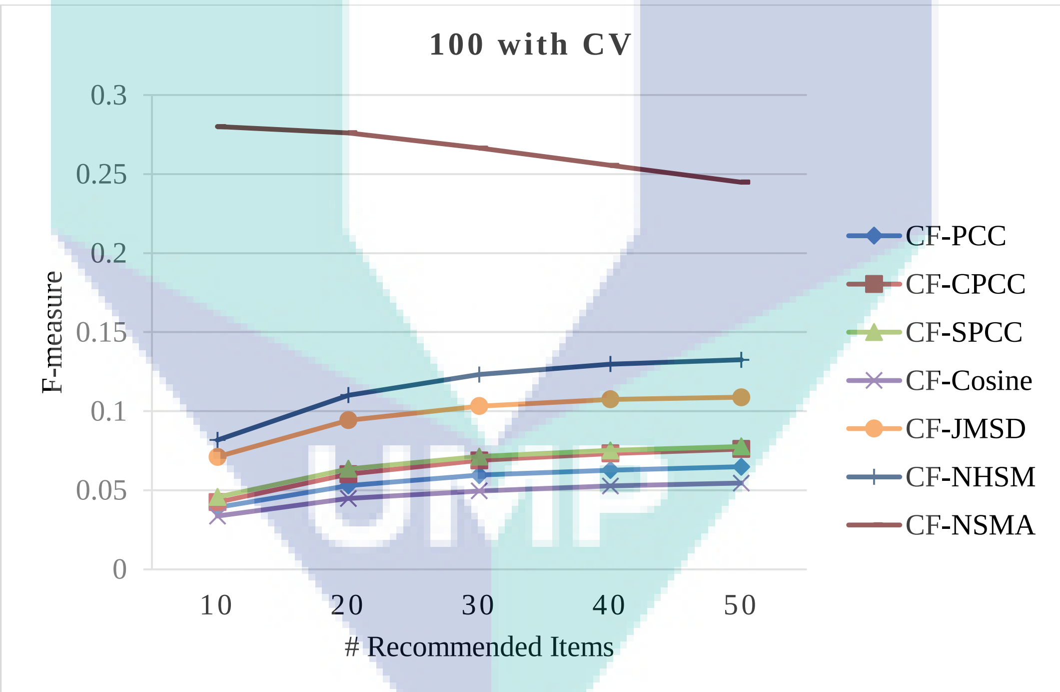



Figure 38 F-measure measure vs various number of recommendations on 100K, Cross-validation.

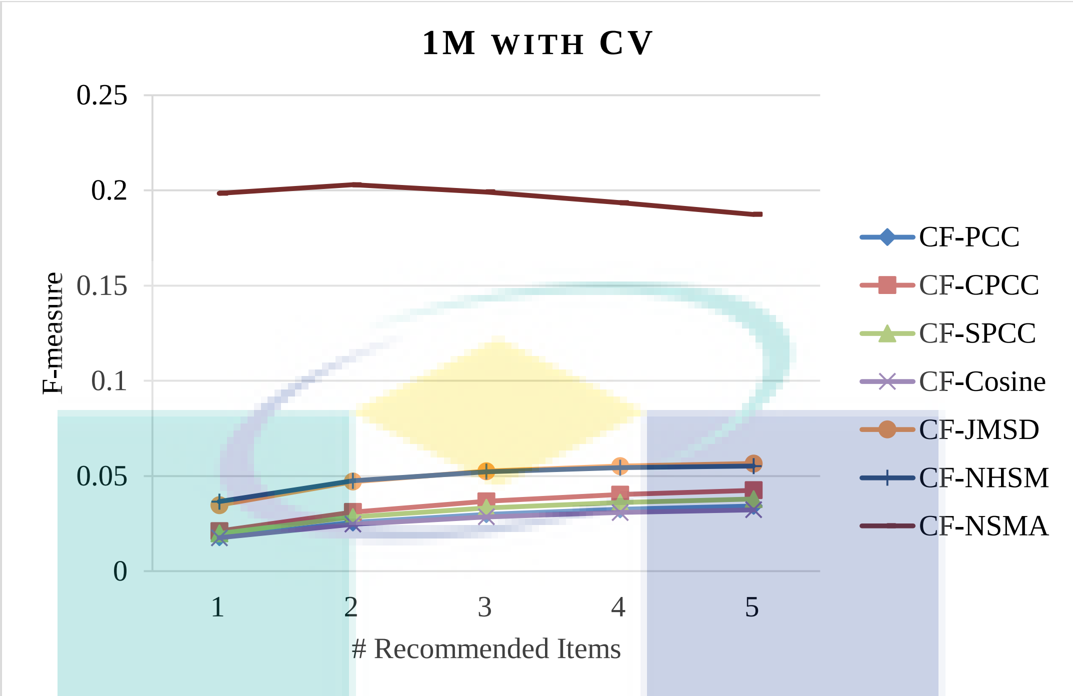



Figure 39 F-measure measure vs various number of recommendations on 1M, Cross-validation

### F.9. Proposed Technique Vs Traditional Methods Using MADAM

In this section, the prediction method in the traditional memory-based CF will be replaced by the MADM method to show the positive effect of MADAM on the traditional methods performance accuracy. Some experiments were conducted using holdout and cross-validation partition methods on 100K & 1M MovieLens datasets. The results were presented in the line graphs which show the improvement that made by MADM when is compared to the traditional memory-based CF methods. Three metrics were used to measure the performance accuracy (Recall, Precision, and F-Measure). The results that shown in all line graphs illustrate the performance accuracy by averaging variation size of neighbours (10, 20, 30, 40 and 50). For more details, see results in appendix E.

#### I. Recall Metric

Figure 40 to Figure 43 compare the proportion of recall between CF-PCC-MADM, CF-CPCC-MADM, CF-SPCC-MADM, CF-Cosine-MADM, CF-JMSD-MADM, CF-NHSM-MADM and CF-NSMA technique. Where Figure 40 & Figure 41 present the comparison of recall when holdout splitting method was applied on 100K & 1M datasets, respectively. While, Figure 42 & 43 indicate the comparison of recall rate when cross-validation partition method was applied on 100K & 1M datasets, respectively. The horizontal line present variation size of recommendations (10, 20, 30, 40, and 50).

At the onset, it is clear that the recall rate of CF-NSMA and CF-NHSM-MADM have approximately the same rate which was the highest rate in all figures over all cases except Figure 43. In that figure the CF-NHSM-MADM was a little better than CF-NSMA and, therefore was the highest. In general, the recall rose gradually to reach to the highest rate when the number of recommendations was 50. Moreover, all methods have a significant enhancement by around twice as much compare to its recall without MADM method. However, in all figures, we can see that the CF-NSMA recall exceeds all methods recall except Fig. 7.29, which was less than CF-NHSM-MADM rate by around 0.05

In conclusion, the enhancement which made by MADM method on traditional methods shows the importance of MADM method in improving the performance accuracy of memory-based CF.

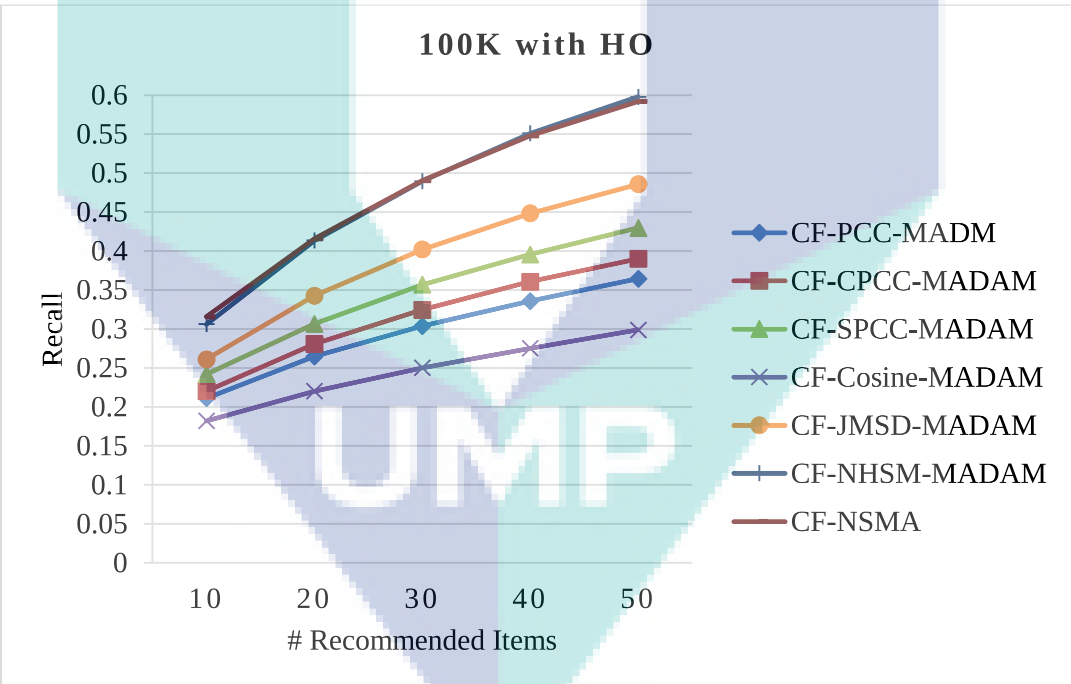



Figure 40 Recall measure vs various number of recommendations on 100K, Holdout.

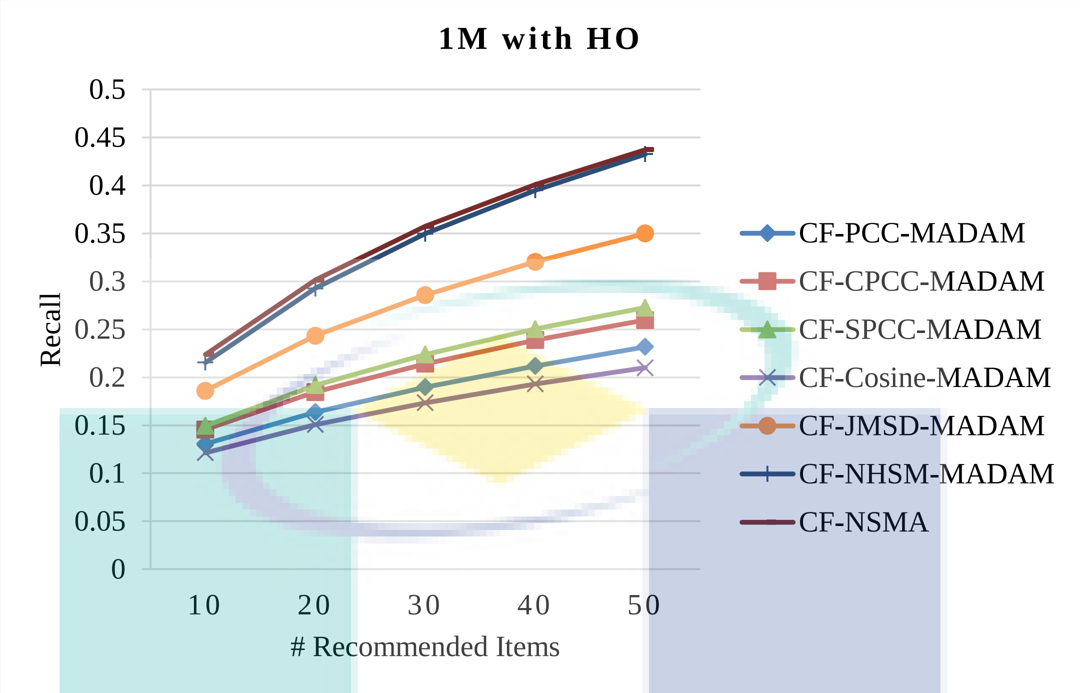



Figure 41 Recall measure vs various number of recommendations on 1M, Holdout.

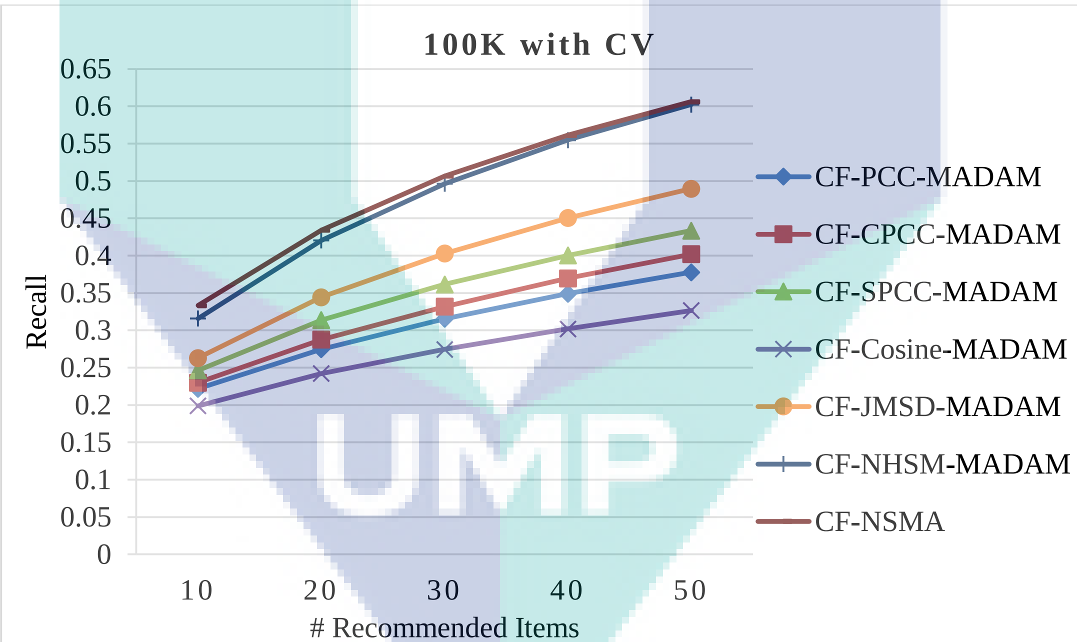



Figure 42 Recall measure vs various number of recommendations on 100K, Cross-validation.

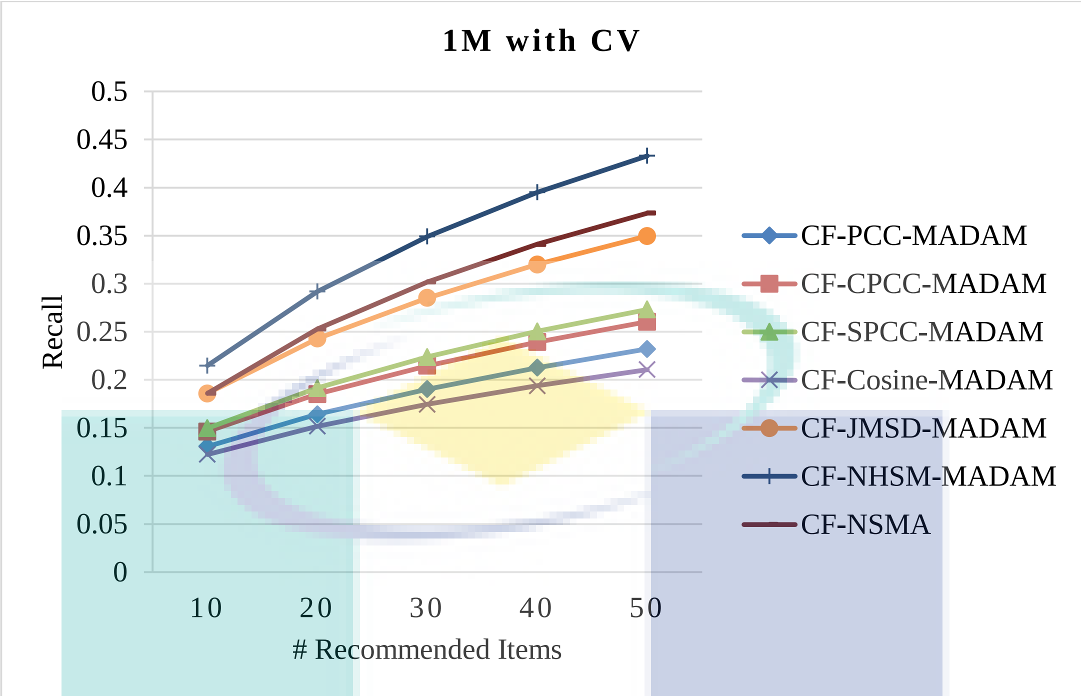



Figure 43 Recall measure vs various number of recommendations on 1M, Cross-validation

## II. Precision Metric

Figure 44 to Figure 47show comparative data of precision for CF-PCC-MADM, CF-CPCC-MADM, CF-SPCC-MADM, CF-Cosine-MADM, CF-JMSD-MADM, CF-NHSM-MADM and CF-NSMA technique. Where Figure 44 & Figure 45 indicate the comparison of precision when holdout splitting method was applied on 100K & 1M datasets, respectively. While, Figure 46 and Figure 47 illustrate the comparison of precision rate when cross-validation partition method was applied on 100K & 1M datasets, respectively. The horizontal line present variation size of recommendations (10, 20, 30, 40, and 50).

In general, it is clear that the precision rate of CF-NSMA has approximately the highest rate in the figures Figure 44 & Figure 45 overall cases. While, in Figure 46 & Figure 47 the CF-NHSM-MADM has a little more rate than CF-NSMA by around 0.02 and 0.04, respectively. According to the other methods, they have a significant enhancement by around more 0.05 compare to its precision rate without MADM method. It can be seen that the precision decreased gradually to reach to the lowest rate when the number of recommendations was 50 in all figures. In conclusion, the enhancement which made by MADM method on traditional methods shows the importance of MADM method in improving the precision accuracy of memory-based CF.

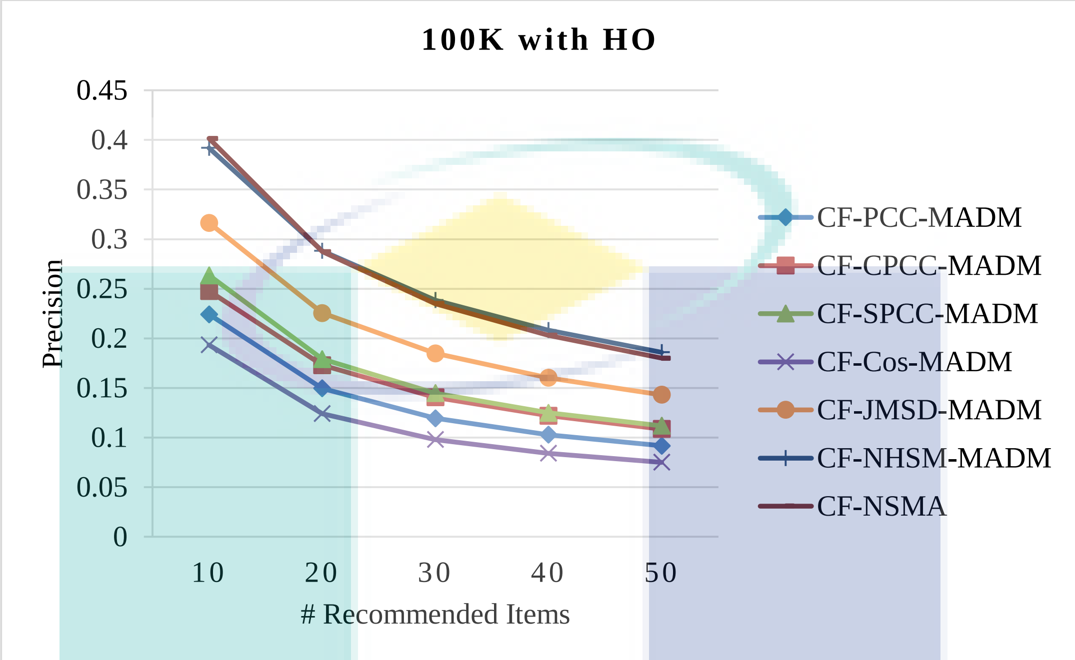



Figure 44 Precision measure vs various number of recommendations on 100K, Holdout.

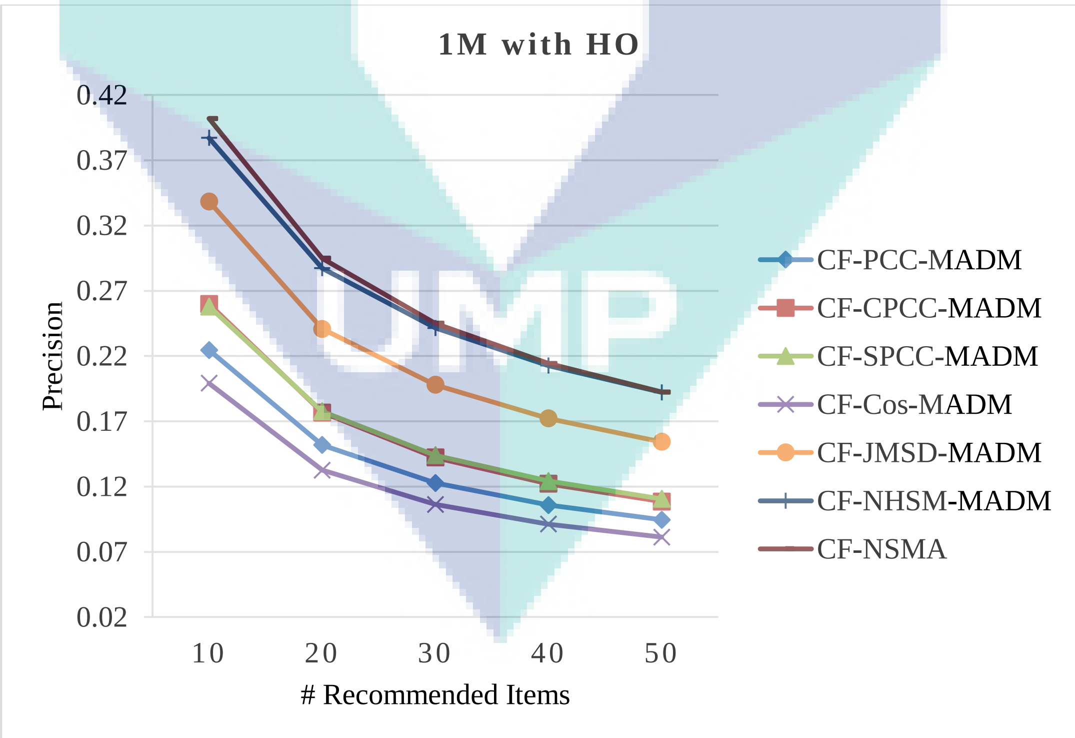



Figure 45 Precision measure vs various number of recommendations on 1M, Holdout.

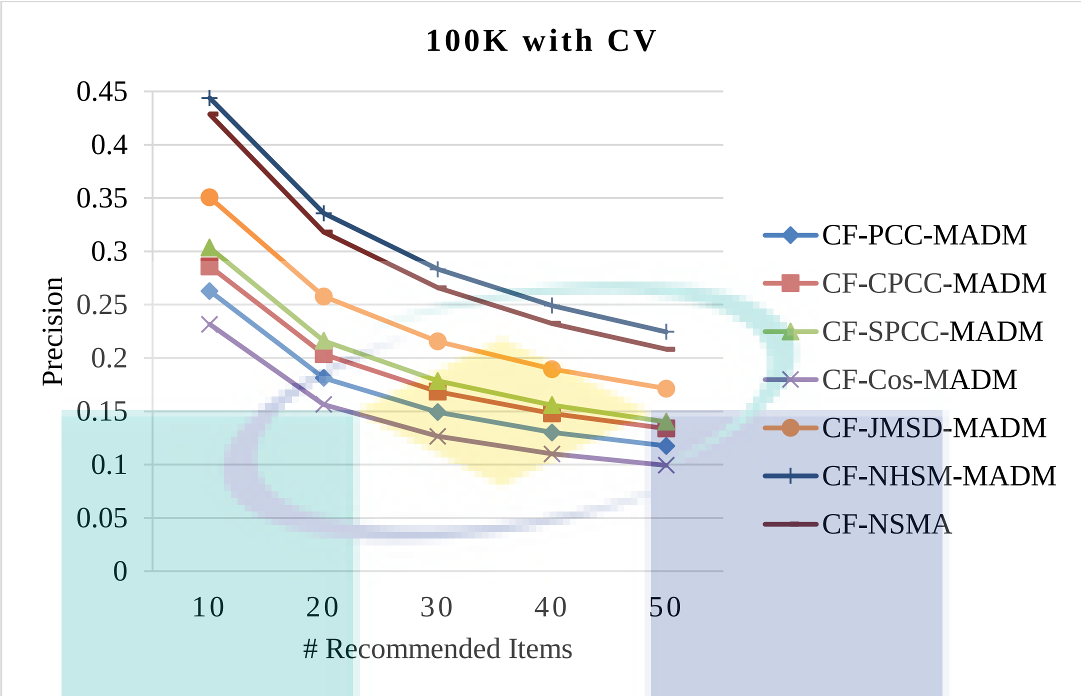



Figure 46 Precision measure vs various number of recommendations on 100K, Cross-validation.

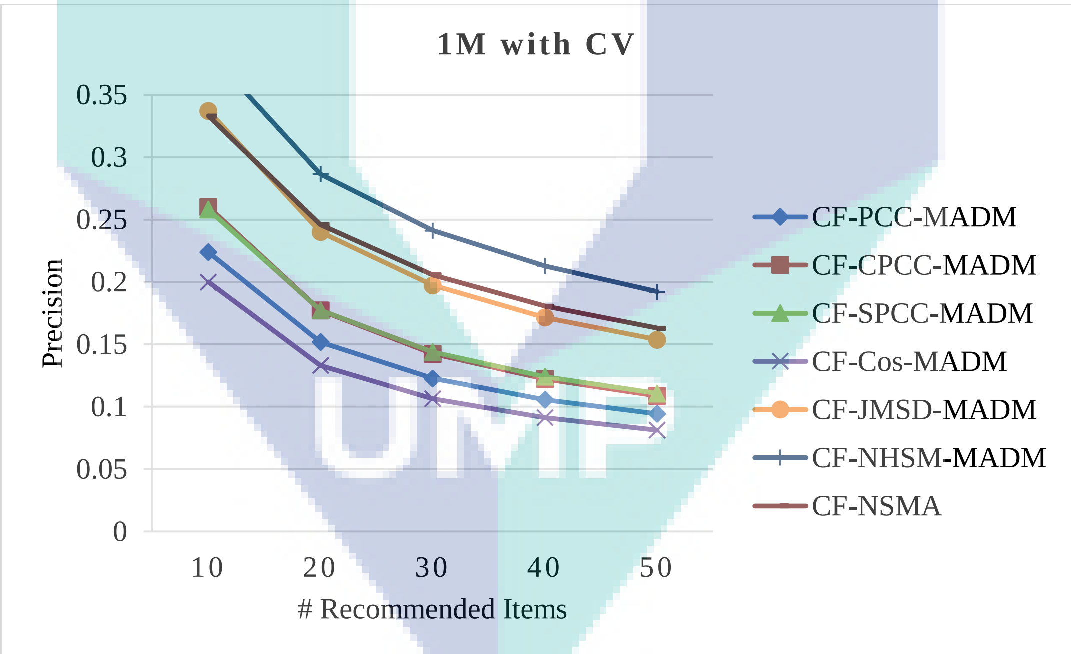



Figure 47 Precision measure vs various number of recommendations on 1M, Cross-validation.

### III. F-measure Metric

Figure 48 to Figure 51 compare the proportion of F-measure between CF-PCC-MADM, CF-CPCC-MADM, CF-SPCC-MADM, CF-Cosine-MADM, CF-JMSD-

MADM, CF-NHSM-MADM and CF-NSMA technique. Figure 48 & Figure 49 indicate the comparison of F-measure when holdout splitting method was used on 100K & 1M datasets, respectively. While, Figure 50 & Figure 51 show the comparison of F-measure rate when cross-validation partition method was used on 100K & 1M datasets, respectively. The horizontal line presents the F-measure rate using various number of recommendations size which was 10, 20, 30, 40, and 50.

In general, it can be seen that the F-measure has slightly dropped when the number of recommendations increases in all figures. Moreover, we can see that; all methods have a notable enhancement compare to its F-measure without MADM method.

At the onset, it is clear that the recall rate of CF-NSMA and CF-NHSM-MADM have an approximately same rate which was the highest rate in all figures over all cases except Figure.51 In this figure, the CF-NHSM-MADM was a little better than CF-NSMA and, therefore was the highest. In conclusion, the enhancement which made by MADM method on traditional memory-based CF methods shows the importance of MADM method in improving performance accuracy of memory-based CF methods. The F-measure rate enhanced by approximately more than three-quarter improvement.

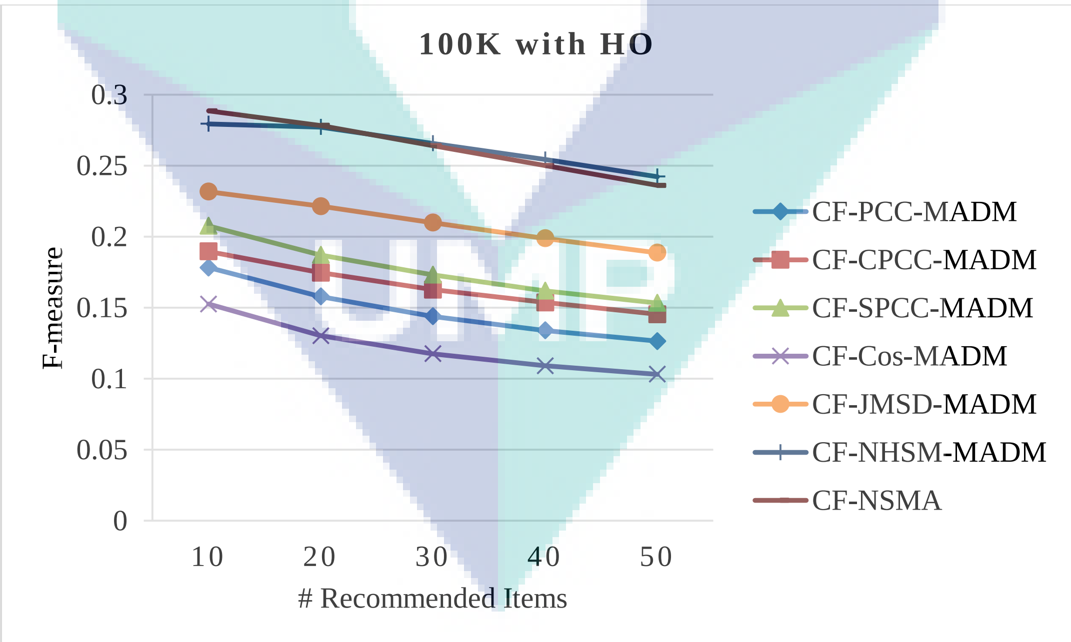



Figure 48 F-measure vs various number of recommendations on 100K, Holdout.

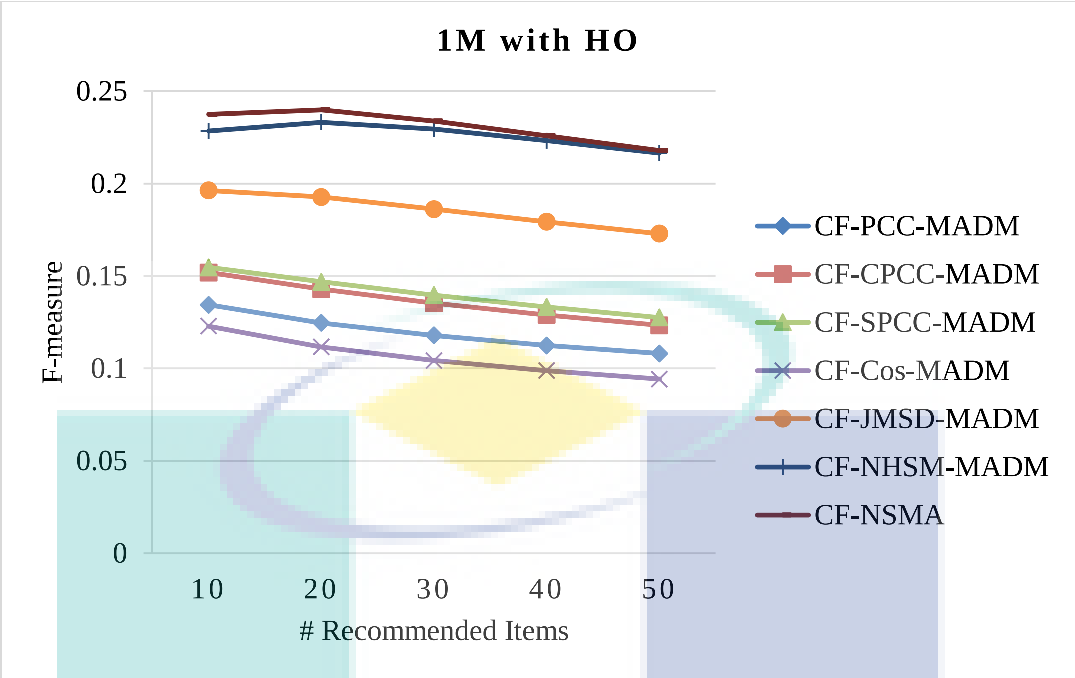



Figure 49 F-measure vs various number of recommendations on 1M, Holdout.

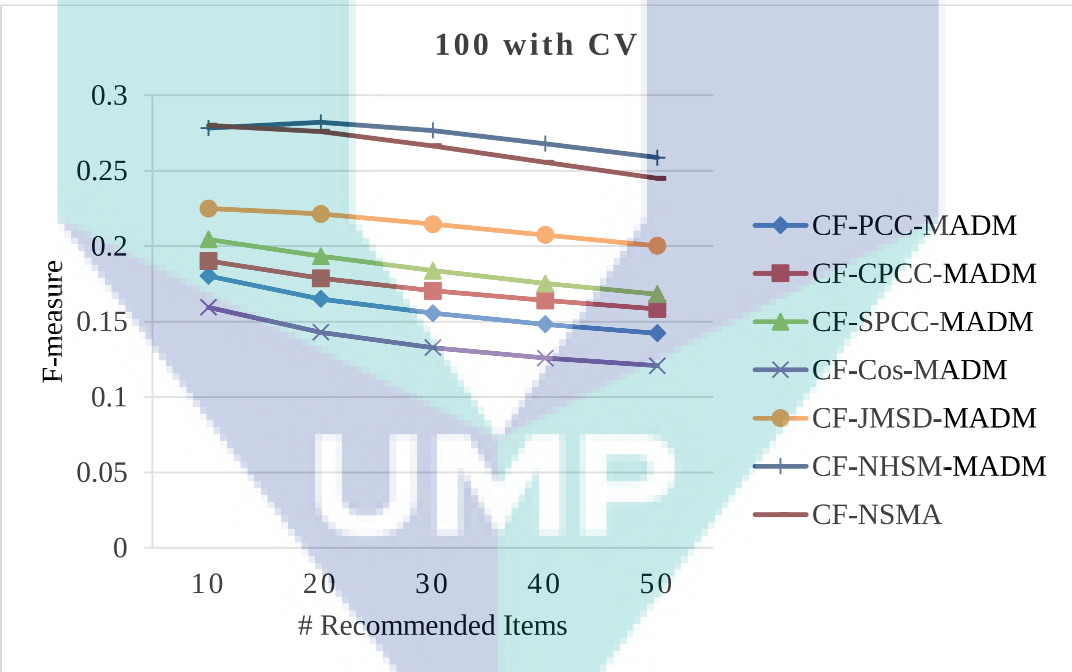



Figure 50 F-measure vs various number of recommendations on 100, Cross-validation.

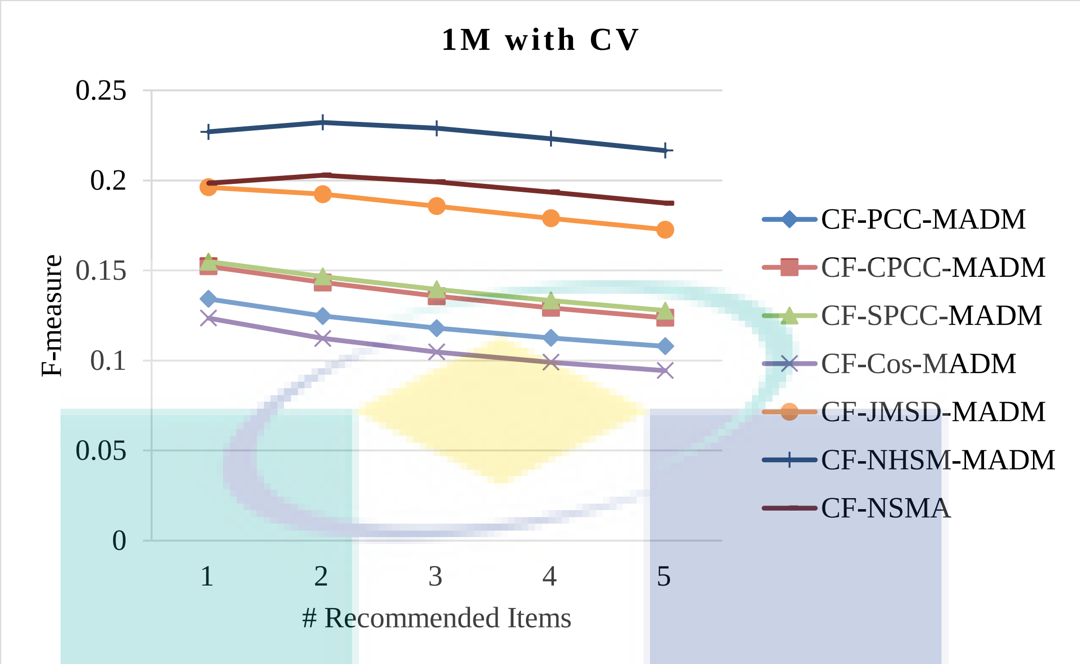



Figure 51 F-measure vs various number of recommendations on 1M, Cross-validation.

## F.10. Appendix Summary

To summarise, the experiments and its results are presented which shown how well the proposed technique works. Firstly, the initial values of sigmoid function denominator are tested, and the appropriate value was identified. This value was used as primary input in the computation similarity process to up the similarity weight when the number of common items more enough (the minimum number of co-rated items). Next, two public datasets (100K & 1M MovieLens) are partitioned into triaging and testing sets, using holdout and cross-validation splitting methods, to implement CF-BSF technique. After that, the outputs were presented and compared according to dataset and splitting method in terms of prediction and performance accuracy. Likely, the performance accuracy of proposed technique CF-NSMA was tested which shown the improvement that made by MADAM. Moreover, the results of CF-BSF and CF-NSMA were compared. As discussed before, the performance accuracy of CF-NSMA has a significant improvement. Therefore, the main objective of replacing prediction method by MADAM method is achieved.

In this part, the experiments were conducted on proposed technique (CF-NSMA) and traditional memory-based CF methods (CF-PCC, CF-CPCC, CF-SPCC, CF-Cosine, CF-JMSD, and CF-NHSM) to show the preceding of the new technique. Firstly, the results of new similarity method CF-BSF were compared to results of the traditional memory-based CF methods to show the importance improving made by CF-BSF. Secondly, the performance accuracy of CF-NSMA has a notable improvement when compared to performance accuracy of traditional memory-based CF methods. Thirdly, the prediction method in traditional CF methods was replaced by the MADM method to show the positive effect of MADAM method on performance accuracy. The given results were presented and compared with CF-NSMA to show the significant enhancement. All experiments were conducted on 100K & 1M MovieLens public datasets using holdout and cross-validation splitting methods. The comparison was regarding prediction and performance accuracy.