The Improvement of Item-based Collaborative Filtering Algorithm in Recommendation System Using Similarity Index

MAYAO

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)
UNIVERSITY UTARA MALAYSIA
2022

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

MA, YAO

calon untuk Ijazah **MASTER OF SCIENCE (INFORMATION TECHNOLOGY)**
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"THE IMPROVEMENT OF ITEM-BASED COLLABORATIVE FILTERING ALGORITHM IN RECOMMENDATION SYSTEM USING SIMILARITY INDEX"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***13 September 2021.***
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:
13 September 2021.

Pengerusi Viva: **Assoc. Prof. Ts. Dr. Mohd Hasbullah Omar** Tandatangan
(Chairman for VIVA) *(Signature)*

Pemeriksa Luar: **Assoc. Prof. Dr. Maizatul Akmar Ismail** Tandatangan
(External Examiner) *(Signature)*

Pemeriksa Dalam: **Ts. Dr. Juhaida Abu Bakar** Tandatangan
(Internal Examiner) *(Signature)*

Nama Penyelia/Penyelia-penyelia: **Ts. Dr. Mohamed Ali Saip** Tandatangan
(Name of Supervisor/Supervisors) *(Signature)*

Nama Penyelia/Penyelia-penyelia: **Dr. Azizi Bin Ab Aziz** Tandatangan
(Name of Supervisor/Supervisors) *(Signature)*

Tarikh:
(Date) ***13 September 2021***

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Penggunaan produk berteknologi tinggi yang meluas dan meningkat dalam perniagaan telah menghasilkan sejumlah besar maklumat perniagaan untuk diproses dalam banyak bidang. Oleh itu, sistem pengesyoran diperkenalkan sebagai strategi yang berkesan untuk menguruskan masalah lebihan maklumat perniagaan. Sistem ini bertujuan untuk menapis maklumat yang besar dan memberi cadangan yang sesuai kepada pengguna. Algoritma penapisan kolaboratif ialah salah satu algoritma yang digunakan dalam sistem pengesyoran. Walau bagaimanapun, algoritma penapisan kolaboratif menghadapi masalah permulaan sejuk, di mana item baharu dalam senarai beli-belah tidak dikenal pasti dan diperakui oleh sistem. Oleh itu, kajian ini mencadangkan penambahbaikan pada algoritma penapisan kolaboratif yang bertujuan untuk mengurangkan masalah permulaan sejuk dengan menggabungkan penarafan item dan atribut item dalam indeks persamaan. Prestasi algoritma yang dipertingkatkan dibandingkan dengan algoritma penapisan kolaboratif sedia ada dari segi kadar ketepatan, kadar ingatan semula dan skor F1 menggunakan dataset Movielens. Kecekapan, objektiviti dan ketepatan algoritma dalam prestasinya telah diukur. Akhirnya, keputusan eksperimen menunjukkan bahawa algoritma yang dicadangkan mendapat 15 peratus kadar ketepatan, 6 peratus kadar ingatan dan 9 peratus skor F1. Oleh itu, ia terbukti lebih berkesan dalam menangani masalah permulaan sejuk dengan menggunakan indeks persamaan baharu, dan juga boleh membuat pengesyoran pada item baharu dalam bidang berbeza dengan ketepatan yang memuaskan untuk hasil pengesyoran yang lebih baik. Secara teorinya, kajian ini menyumbang kepada penambahbaikan algoritma penapisan kolaboratif dalam sistem pengesyoran untuk mengatasi masalah permulaan sejuk dengan menganalisis lebih banyak atribut item untuk mengekstrak lebih banyak maklumat kepada algoritma. Selain itu, algoritma yang dicadangkan boleh digunakan dalam pelbagai bidang untuk pengesyoran item sejuk bagi meningkatkan kualiti sistem pengesyoran.

Kata kunci: Masalah Permulaan Sejuk, Algoritma Penapisan Kolaboratif, Sistem Pengesyoran, Pembelajaran Mesin, Indeks Persamaan.

Abstract

The extensive and increase use of high-tech product in business has generated a huge amount of business information to be processed in many fields. Thus, a recommendation system is introduced as an effective strategy to manage the business information overload problem. The system aims to filters enormous information and proposes appropriate suggestions to users. A collaborative filtering algorithm is one of the algorithms applied in the recommendation system. However, the collaborative filtering algorithm faces cold-start problem, where new items in the shopping list are not identified and recognized by the system. Hence, this study proposes an improved collaborative filtering algorithm which aims to alleviate the cold-start problem by combining the item rating and item attributes in similarity index. The performance of enhanced algorithm was compared to existing collaborative filtering algorithms in term of precision rate, recall rate and F1 score using Movielens dataset. The algorithm's efficiency, objectiveness, and accurateness towards its performances were measured. Finally, the experimental results showed that the proposed algorithm get 15 percent precision rate, 6 percent recall rate and 9 percent F1 score. Thus, it proved to be more effective in deal with cold-start problems by using new similarity index, and also can make recommendations on new items in different fields with satisfactory accuracy for better recommendation result. Theoretically, this study contributes to improve the collaborative filtering algorithm in recommendation system for overcome the cold-start problem by analysing more item attributes to extract more information to the algorithm. Besides, the proposed algorithms can be applied in many fields for cold-items recommendation and to enhance the quality of the recommendation system.

Keywords: Cold-Start Problem, Collaborative Filtering Algorithm, Recommendation System, Similarity Index.

Acknowledgement

First of all, I would like to thank my supervisors Dr. Mohamed Ali b. Saip and Dr. Azizi Bin Ab Aziz for tireless efforts, suggestions, and guidance to make this work successfully with attention and care. Moreover, I had a very enjoyable study at Universiti Utara Malaysia (UUM). Not only, it has a beautiful natural environment, and the university also has helpful staff.

As we know, 2020/2021 was a tough time for the world; it also a hard time for me; due to the spread of the Covid-19, lots of people lost their family or friend. Hoping the world back on track soon, I sincerely wish everyone good health, happiness and success.

Last but not least, I wish to express my deepest appreciation to my beloved family members and my close friends for supporting and believing in me with their unconditional support. Thank you.

Table of Contents

List of Tables

List of Figures

List of Abbreviation

CF	Collaborative Filtering
CB	Content-Based
IBCF	Item-based Collaborative Filtering
UBCF	User-based Collaborative Filtering
CS	Cold-start
RS	Recommendation system
TP	True Positive
FP	False Positive
FN	False Negative

CHAPTER ONE
INTRODUCTION

1.1 Overview

Recommendation System (RS) has become one of the important intelligence systems in many fields, which exploits the hidden information behind the users or items and make a recommendation to the users (Wei, He, Chen, Zhou, & Tang, 2017). For instance, if you like to access YouTube channel, the website will recommend videos to you based on your interest, and when you purchase items in Amazon website, you will get similar items recommendation. The information used in the recommendation system collected by the user was captured explicitly (e.g., by rating analysis and purchase history of the user) or implicitly (e.g., by collecting users' behavior and web page browsing time). Additionally, the system can utilize the user's demographic information (occupation, gender, income) and content feature for items (publication, brand, attributes). Although the recommendation system uses different information to make predictions and recommendations to users, considering accuracy, novelty, and stability is extremely important in recommendation results.

According to Schafer, Konstan, and Riedl (2011), the business needs to be equipped with multiple products that meet multiple consumer's need, which means that the business has to provide users with different products based on different preferences. For example, there are thousands of products in a store, and the users might be confused in selecting a product from such a long list. Therefore, businesses have to provide their users with accurate information about the product to make recommendations and enhance trade opportunities. Furthermore, the rapid increases of products have created enormous information when users select their preferred product. In order to solve the information overload issues, the recommendation system has been developed and proposed to support users with personalized recommendations while assisting the business to make better product management (Lu, Wu, Mao, Wang, & Zhang, 2015; Guo, Yin, Li, Ren, & Liu, 2018).

Recently, recommendation system has been applied in diverse areas such as movie recommendation, book recommendation (Amazon.com book recommendation system) and E-commerce domain (Thakkat, Varma, Vijay, Mankad & Tanwar, 2019). At the beginning of the recommendation system development, the main methods included demographic-based algorithm, content-based algorithm, and collaborative filtering algorithm (Pazzani, 1999).

Researchers suggest that a good recommendation system can process massive information about users and items, with only a second processing time to generate the recommendation result. Moreover, the system also can react immediately due to the changes in user preference and makes parallel recommendations for all users regardless of the number of purchases and ratings. Unlike other recommendation algorithms, the Collaborative Filtering (CF) algorithm can meet this challenge and be the most popular approach for recommendation systems design (Wei et al., 2017).

The concept of collaborative filtering algorithm is based on how humans have made decisions throughout history, or which are combined with our own experiences. People often making decisions depend on the experiences and knowledge that reach each of us from a relatively large group of acquaintances. In another word, CF is a method of making item predictions about the interests of a user by collecting preferences or taste information from many other similar users. In a more general sense, CF is a method to filtering the information by considers collaboration among different sources.

With the development of the CF algorithm, the item-based CF has been proved more reliable than the traditional CF algorithm (user-based CF) (Shambour, Hourani, & Fraihat, 2016). According to research, the item-based CF plays an important role in the recommendation system (Thorat, Goudar, & Barve, 2015), and is often used along with other techniques such as content-based (Lops, de Gemmis, & Semeraro, 2011), sentiment classification (Singh, Mukherjee, & Mehta, 2011) and combined with User-based CF algorithm (Thakkar et al., 2019). However, there is still suffer from drawbacks in item-based CF algorithm. When a new item appears in the system, it causes less information provided to the system to generate a recommendation, which is called the cold-start problem. Unlike other recommendation algorithms, the Collaborative Filtering (CF) algorithm can meet this challenge and be the most popular approach for recommendation systems design.

This study proposes an improvement of the item-based CF algorithm for a personalized recommendation system. The proposed algorithm explores the similarities between items in similarity calculation phase to ease the cold-start problem. The proposed algorithm will be tested against empirical data and compared with the other CF algorithm to support the capabilities of the new algorithm.

1.2 Problem Statement

The emerging internet technology has transformed data management and the ways people are buying goods and products. The increasing number of items from websites allows users to choose varieties of products from personal care to house furniture at anytime and anywhere. However, the rapid growth of information, which cause the inefficient of the users when only a few relevant products were browsed, only a few items were viewed and purchased by users (Guo et al., 2018). Hence, recommendations system has been developed to guide users in exploring and purchasing products to satisfy their needs.

During the recommendation computation, the systems utilize different sources of information to predict and make a recommendation on the fittest products to the potential users. The system will balance accuracy, novelty, diversity, and stability in the final recommendation list. Behind the recommendation, the Collaborative Filtering algorithm is the most commonly applied technology in the recommendation system, which plays an important role in the product selection process and recommendation (Karahodža et al., 2017; Kharita, Kumar, & Singh, 2018; P. Sharma & Yadav, 2020; Iwendi, Ibeke, Eggoni, Velagala, & Srivastava, 2021).

However, the Item-based CF algorithms suffer from cold-start problem (Shambour et al., 2016; Dou, Yang, & Deng, 2017; Kanakia, Eide, Shen, & Wang, 2019; Natarajan, Vairavasundaram, Natarajan, & Gandomi, 2020). The cold-start problem concerns the issue that the system cannot draw any inferences for users or items that have not yet gathered sufficient information. This is due to the lack of transactions data or browse data about the new items. Whenever a new items was added to the system, the item had not get any purchased or rated, and the system can't compute similarity to other items (Natarajan et al., 2020), which may leads the similarity calculation can't process well and the algorithm can't make new items recommendation. This may lead to unreliable and inaccurate recommendation results and cannot make any recommendation relevant to new items.

Thus, this study aims to improve the performance of the Item-based CF algorithm by reducing the items cold-start problem. Later, the proposed IBCF algorithm is evaluated to show a better recommendation result than other CF algorithms.

1.3 Research Questions

The following research questions are proposed in this study:

1. How to improve the Item-based Collaborative Filtering (IBCF) algorithm?
2. How to evaluate the improvement of the IBCF algorithm?

1.4 Research Objectives

The objectives of this study are as follows:

1. To develop an improvement of the Item-based Collaborative Filtering (IBCF) algorithm.
2. To evaluate the proposed IBCF algorithm with empirical data.

1.5 Significant of the Study

By improving the IBCF algorithm, the recommendation system will be more accurate and reliable than before, while the system has the ability to address the cold-start problem by combine the item rating and item attributes in similarity index. Moreover, the improved IBCF algorithm can applied in many fields. For example, in e-commerce domain, by improve the recommendation system, the business can gain more profit through creating more trade opportunities. In addition, the customers have a better experience, while they gain more chance to purchase potential interest products.

1.6 Research Outcomes

This study aims to provide a strong foundation for the personalized recommendation in the future, at the end of this research, the outcomes are as below:

1. The design of the improvement for Item-based Collaborative filtering algorithm to overcome the cold-start problem.
2. The implementation and evaluation of the proposed method.

1.7 Research Scope

The study focused on implement the improvement of item-based CF algorithm. In order to achieve the research objectives, empirical data from Movielens was used. These data are stored in CSV files which consist of 100,000 ratings (obtained from the Movielens open-source database) were used in this study. The data includes rating, item ID and attributes. The item-based CF algorithm was implemented in Python3.7 programming language. However, the recommendation system architectures and deep learning algorithm are out of this study’s scope.

1.8 Organization of the Study

The first chapter introduces the research overview and reveals research background that produces research questions and research objectives. This chapter also details out the problem statement that is proposed of the research objectives and discusses the IBCF algorithms of the recommendation system and the significances of this study.

The second chapter consists of the literature review that focuses on the studies on a definition of the recommendation system and different types of algorithms used in the recommendation area related to this study. Also, it covers some aspects in the cold-start solutions introduced by other researchers.

The third chapter explains the methodology of this study to achieve the objectives of this study, includes the research approach, identifies the item-based CF algorithm, experiment design, and gives a general idea on the design of the proposed method.

The fourth chapter presents the validation phase of the algorithm. This chapter also involves the data preprocessing, experiment design and the validation result phase based on the experiment.

The last chapter revisits the objectives of this study that relate to contribution in the recommendation domain. It summarizes the main contribution of this study and highlights the limitation and future work that will contribute to the recommendation system.

CHAPTER TWO
LITERATURE REVIEW

2.1 The Recommendation System

The recommendation system is an information processing engine or an information filtering system to predict user preferences. By ranking or filtering the item data, the system can provide purchase suggestion on items to users based on their interest and previous transactions history. Meanwhile, the recommendation system is an intelligent platform constructed by a massive data mining method, and it is also an implementation of machine learning. In the e-commerce area, the recommendation system has usually been used to predicting and filtering a product which the user may potentially be interested in.

At the early age of website technology, applications such as Amazon.com, YouTube, and Google have applied recommendation system (Adomavicius & Tuzhilin, 2005). Although the recommended content was processed manually, it seemed as one style of recommendation. The other method of recommendation is to integrate the information, such as bookstore or cinema. For instance, a best-seller list shows that which product was selling well. Other similar searching engines that present most relevant results are also considered a recommendation system, so the recommendation system was already embedded in people’s daily life most relevant results as a recommendation system years ago.

At the same time, traditional business trade seems too limited in time, workforce, communication, and space. While online transactions break the concept of time and space, it enables everyone to browse and purchase goods wherever and whenever. Because of the diversity of goods and products, the traditional trade method seems too limited in catching people’s taste. Meanwhile, the recommendation system can quickly formulate the psychological needs of customers.

In recent years, technology companies have broadly used the recommendation system to assist their customers with personalized recommendation services (Guo et al., 2018). For example, Amazon, YouTube and Alibaba have introduced recommendation techniques in their systems to estimate the potential preferences of customers and recommend relevant products or items to the user (Wei et al., 2017). A study founded that product recommendations largely influenced consumers' purchasing decisions and may promote sales (Zhao et al., 2014).

Nowadays, the implemented recommendation systems have become very common based on advanced technology and have been widely used in videos, books, music, news and business area (Bobadilla et al., 2013; Jiang et al., 2019; Nassar, Jafar, & Rahhal, 2020). The aim of the recommendation systems is to be built for business intelligence, which recommends appropriate items to users, while the system always processing with large amounts of item information. One of the recognized recommendation applications is Amazon’s recommendation system, which provides users with a personalized web page when they visit Amazon.com. However, technology companies are not the only one that utilizes the recommendation system to raise their service quality. Although in different applications, the system is also used in other industries, from recommending music and news to video and house.

2.1.1 What is Recommendation System

Due to social networks and technologies, data and information had a substantial growth over the past 20 years. How data becomes large-scale can be traced back to the first recorded information explosion decades ago (Menon & Hegde, 2015; Bobadilla et al., 2013). In 1944, Fremont Raider predicted that the Yale University Library's data volume in 2040 would reach 200 million volumes (Gandhi & Gandhi, 2018). The information overdue problem is more and more common in modern society. Thus, recommendation systems become one of the data processing tools. However, the recommendation system appeared in an isolated research field in the mid of 1990s (Adomavicius & Tuzhilin, 2005), when researchers began to focus on recommendations that depended entirely on rating issues. In the early concept area, the recommendation question is a tool to evaluate the rating problem of items that users did not viewed before. Fundamentally, this assessment is usually based on the ratings of other items and information. Once the system formulates the rating of the items that have not been graded, the system can recommend the item based on the highest rating.

Recommendation system can be used in many areas, such as recommend today’s headlines news, videos related to user's interest in YouTube, and highlight the same style of music when users use the music application. In E-commerce domain, the system can give purchase suggestions in books, foods, clothes, and other items through online shopping. It also makes content recommendations based on user’s interest in social media such as Twitter, Facebook, and Netflix. Likewise, Amazon makes a personalized recommendation for every customer (Linden, Smith, & York, 2003), based on their interests. This gives a sense as if customers walk into a shopping mall and the items are arranged by themselves, placing products that fit their preferences, with the most preferred will move to the front, and the least preferred products are in a different place. The store made a fundamental change based on the customer's interests (shows bandages to athletes and shows the baby toys to a new mother), which will improve customer satisfaction and royalty.

According to research, the input data of the recommendation system can be varied, but generally divided into three parts including item information, user information and review information.

Firstly, the item information used to describe the property of an item (Item profile), is usually varies based on related items. For instance, a book profile may include author, page number, publication data, and publishers. The item profile might include authors, keywords and title. Utilizing the item attribute and information to make a recommendation is known as content-based recommendation.

Secondly, the user information is described as the characteristic of the user, which means the user profile can be different, such as gender, age, income, interest, and vocation. In the recommendation system, making recommendations based on user information can be defined as a demographic-based recommendation.

Finally, the review information is consisting of user and item information. A simple review is a user who purchased an item and made a rating and comment, which indicates how much the user likes the item, the rating can range from 1 to 5. Depending on the system, the review may include rating, purchasing history, and browsing history.

The three sets of input can be defined as the ratings matrix, where the user has not rated will be marked with symbol question mark ('?'), which means unknown values (it might be because of the user does not purchase the product or does not give any rating for the product), and the value of the cell is the rating from the user (row) to items (column). Table 2.1 shows an example of rating matrix for three users with four different items.

Table 2.1

Ratings Matrix of recommendation system

	BOOK	**VIDEO**	**FOOD**	**FUNITURE**
User A	4	3	1	4
User B	?	2	4	?
User C	5	4	2	?

As can be seen in this Table 2.1, there are some cells are marked with symbol question mark (“?”) which indicates no rating has given to these products by the particular users. Thus, the recommendation system can predict the rating of these cells based on the different algorithms, and make a recommendation list to a user by ranking the rating score from high to low, as the predicted rating may not be the only criteria used to produce the recommendation list.

2.2 The Algorithms Used in the Recommendation System

The extremely increasing in products and services has led to information-overloaded problem. Hence, researchers in different field have explored and developed a recommendation system in order to reduce this problem. Recently, three main methods of recommendation system are widely used to solve the problem including

Demographic-based, Content-based, and Collaborative Filtering recommendations (Bobadilla et al., 2013; Thorat et al., 2015) as shown in Figure 2.1.

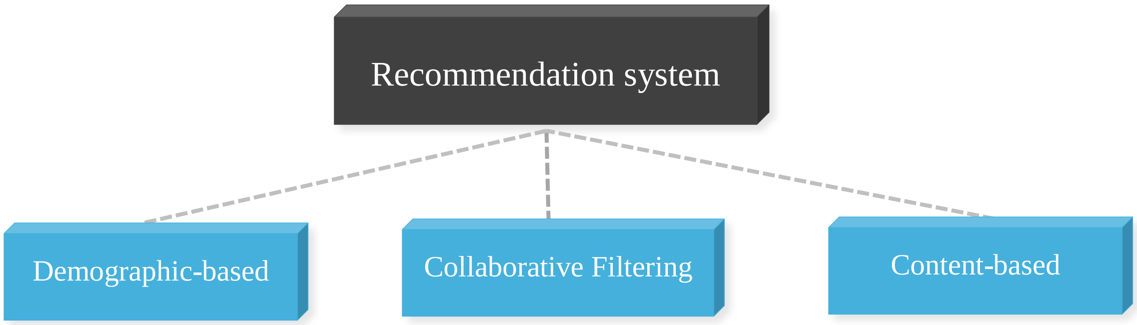


Figure 2.1 Main Methods of Recommendation System

Recommendation systems are applied in different areas by recommending items, like items, videos, interesting things, research paper, and news (Safoury & Salah, 2013). As shown in Figure 2.1, the recommendation system can consist of different methods. However, all recommendation system methods have advantages and disadvantages on a different design.

Table 2.2 shows a list of research implemented difference algorithms of recommendation system. It can be found that the demographic-based algorithm usually used in a platform contains comprehensive user information like a microblog or social website (Zhao et al., 2014). Conversely, the content-based (CB) algorithm is unable to discover the potential interest item of a user because the algorithm always tends to recommend items similar to history items, but the algorithm can overcome the cold-start problem (Son & Kim, 2017). Once the CB algorithm gets the item profile, the algorithm proceeds with the similarity between items and recommends user. At the same time, the CF algorithm is one crucial foundation algorithm in the history of recommendation systems, which has been used in many E-commerce organizations like Amazon.com (Linden et al., 2003; Linden, Smith, & York, 2017). It can discover the potential interest of the users but suffers from a well-known cold start problem (explained in Section 2.2.3), and the system tends to make a recommendation on popular items, and less popular items have fewer opportunities to be recommended. In this study, the proposed work enhances the reliability of the Item-based Collaborative Filtering algorithm in relieving the item cold-start problem.

Table 2.2

A List of the algorithms used in the recommendation system

Author	**ALGORITHM**	**type of data**	**finding**
(Sarwar, Karypis, Konstan, & Reidl, 2001)	Item-based Collaborative Filtering	MovieLens database	The results show that item-based CF algorithms can process a massive amount of data and perform well eventually
(Pazzani & Billsus, 2007)	Content-based algorithm	none	The algorithm can give a good result if the content does not contain enough information

(Zhao et al., 2014)	Demographic-based algorithm Bootstrapping algorithm	Microblogging E-commerce	With high scalability feasibility and effectiveness
(Xiao, Ai, Hsu, Wang, & Jiao, 2015)	Content-based Filtering approach Collaborative Filtering algorithm	Bing news dataset	The proposed CCF combines both the advantages of two algorithms
(Z. long Li, Huang, & Zhang, 2018)	Collaborative filtering	MovieLens dataset	The new method relieves the cold-start problem while improving the reliability of recommendation
(Gandhi & Gandhi, 2018)	& FP-Growth algorithm & Association rule mining & Collaborative filtering algorithm	MovieLens dataset	The integrated algorithm reached good scalability and computing power to make a recommendation
(Kanakia et al., 2019)	Co-citation Content-based algorithm	Microsoft Academic database	The system can handle large scale and incomplete information, while also alleviating the cold -start problem, but it shows less precision in results

(Wang, Deng, Lai, & Yu, 2019)	Innovator based Collaborative Filtering	by Alibaba Group in Ali Mobile database	The proposed method has a good performance on a recommendation result while maintaining on accuracy, novelty, and coverage

2.2.1 Demographic-based Recommendation

The Demographic-based algorithm tries to find similarity between users with the same interest, taste, gender, age, and recommend similar items from A to B (Pazzani, 1999). For instance, if person A has the same age and interests as person B, the product C and D with a high rating rated by person B will be recommended to person A as Figure 2.2.

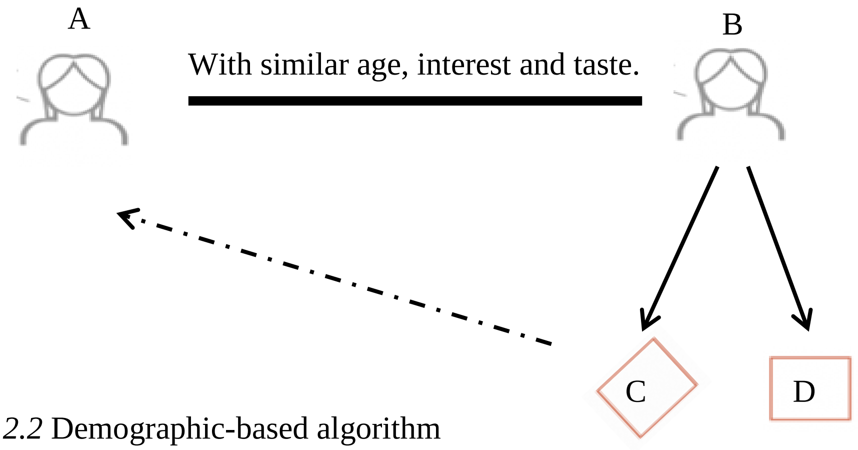


Figure 2.2 Demographic-based algorithm

Demographic-based filtering is built on a user demographic profile of user, which is trying to find the similarities among user based on demographic variables like gender, age or occupation (Thorat et al., 2015). The algorithm assumes one user might like the same product based on a similar user. When the algorithm starts making recommendations to the user, it always utilizes the user profile to calculate the similarity among users, then ranking and filtering the result to select most relevant users. After that, conduct the recommendation list depending on the user's purchases and ratings. One simple and common way to recommend is to list all the products covered by similar users and ranking the items by rating score mean, finally return top *k* items for the target user.

The advantage of Demographics-based filtering is that the computing of the process is simple, the data storage and the calculation of similarities can be processed offline. However, it creates many shortcomings. The main problem is that the recommendation result shows low reliability, which means that although the user has the same gender, age, and pattern behavior, but they have a very high probability that they have different preferences in products. Nevertheless, the algorithm cannot establish a relationship between users and products.

2.2.2 Content-based Recommendation

The Content-based algorithm is similar to the Demographic-based algorithm. However, the difference between the two algorithms is that the content-based algorithm identifies the similarities between products, not between the users (Pazzani

& Billsus, 2007). The algorithm tries to find items with similar attributes and properties or the relevant industry items (Figure 2.3). If a consumer purchases a laptop, the RS will make a purchase suggestion on a mouse or keyboard to the user. Another example is if a customer shows interest in the word "mobile phone" the algorithm recommends the iPhone or HUAWEI's smartphone to the user.

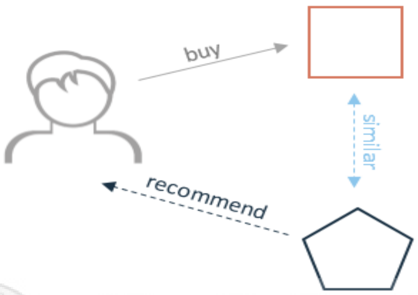


Figure 2.3 Content-based algorithm

A study (Thorat et al., 2015) showed that the content-based approach suggestions analyze the attributes of the items and the description of the items rated by users. While item attributes and user profiles play an essential role in CB algorithms, the system attempts to compute the similarities between items, recommend the best match by analyzing the most similar items relevant to the user's previous behaviors. The Content-based filtering has shown outstanding success to overcome the cold-start problem. Once the system gets a new item's profile, the algorithm can calculate the similarities to the existing items. It breaks the limitation of sparse datasets relative to a collaborative filtering algorithm and gives a strong reason to customers that “this product is similar to the one you purchased before," but it still has some drawbacks. Firstly, the system needs a hybrid models to preprocess the product information to get an Item profile manually. Secondly, this method cannot discover the potential interest items of a user because the algorithm always tends to recommend products similar to the user's history data, and with low expansibility, it needs to formulate different item profiles for a different domain.

2.2.3 Collaborative Filtering Recommendation

At the same time, the CF method assumes that "if someone A chase the same answers on a question as someone B, then A is more likely to have the same opinion as B on another question" (Linden et al., 2003). As shown in Figure 2.4, person A and B both purchased the product (triangle & pentagon), and they have a similar rating on these items, then product C which was highly rated by B will be recommended to A.

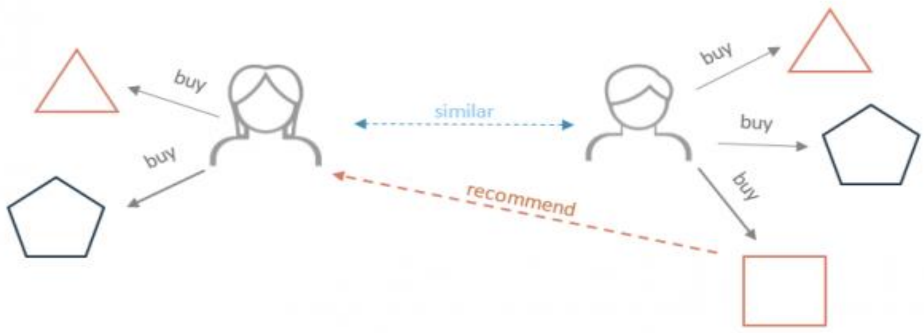


Figure 2.4 Collaborative Filtering algorithm

Unlike CB filtering methods, the CF method can discover product types that users have not seen or get interested in before. For instance, in a movie recommendation application, to make a movie recommendation to a user, a collaborative filtering algorithm attempts to find user peers with similar tastes to the user (with similar ratings for the same movie). Then, it recommends the highest peer rating movies to the user.

One advantage of collaborative filtering is that the system does not require any user or item configuration file maintenance. It can recommend products with unique attributes to the user so the algorithm can be used in many areas. The CF algorithm can also discover the user's potential interests. However, it also faces many issues, such as scalability, sparse data and cold-start issues (Pirasteh, Jung, & Hwang, 2014; Thorat et al., 2015; Li et al., 2018). Most active users rate a few items compared to all the items in the datasets, which causes the sparse data in the rating matrix. On the leading E-commerce website, transaction information is huge, but only a small number of the whole datasets is made up of the most active users rating, the “unpopular” items are rarely rated, which may result in an unreliable recommendation. The registration of new users or new items can lead to cold-start problems because there is not enough data and information for the system to work accurately.

a. Scalability

With the increase of users and items, the CF algorithm has scalability problem. For instance, the thousands of customers and millions of items available in the dataset lead to millions of ratings, which results in slow computations, thus degrading the recommendation system's performance (R. Sharma, Gopalani, & Meena, 2017). So the system needs high computation capability and sensitivity to handle such enormous data to make a recommendation, which requires higher scalability (Thorat et al., 2015).

b. Data Sparsity

The RS utilization shows commonality in modern technology companies, the RS built based on large datasets for includes all users. Therefore, the rating matrix can be very large and sparse, which may lead to inaccuracy of the recommendation result. The well-known cold-start problem is caused by the lack of information of the rating matrix, the traditional CF algorithm only utilizes the rating matrix to predict similarity between items or user (Item-based and User-based). The majority of the items are rated and viewed by a small part of users, which causes the rating matrix to be extremely sparse due to insufficient rating data, which makes algorithms unable to measure similarity among items or users. On the other hand, the CF method recommendation system utilizes the history information of the user to make a prediction, when new users and new items appear in the dataset, the system does not have enough information of users or items to allow the system to compute the similarities among them and may lead recommendation to result in low reliability

(Thorat et al., 2015).

c. Cold-start problem

The cold-start problem is a common and well-researched problem for the CF method, which new items or new users most cause. In a CF-based recommendation system, an item cannot be recommended until several users purchased and rated it. If there are fewer transactions, a CF algorithm cannot recommend the item. Because the algorithm can't compute similarity between items or users, which may cause recommendation result unreliable(Lika, Kolomvatsos, & Hadjiefthymiades, 2014). In other words, these items are seldom ever recommended to users because of insufficient utilized information. Normally the item cold-start problem means that there is a new item to add to the inventory, and there has been none or very few interactions with the item. The RS cannot make a reliable prediction because of the lack of information about the item.

2.3 What is Collaborative Filtering Algorithm

In recent years, the RS has been used in business areas to provide users with items, services, or information recommendations (Safoury & Salah, 2013), matching their preferences and interests to existing products. The RS aims to guide a user in a personalized way to find what they may be interested in, based on their historical preferences and rating information, to discover the potential items among hundreds even from thousands of products inventories.

A powerful algorithm in the recommendation system is the Collaborative Filtering (CF) method, the most extensively used approach to design recommendation systems (Thorat et al., 2015). In CF-based recommendations, for each customer, the recommendations are consisted of comparing the preferences of others who have rated the same product.

The database of the CF-based system consists of users who have previously interacted with various items (Ekstrand, Riedl, John, & Konstan, 2011). The interactions explain that a user purchased or rated an item, these are usually defined as a (User, Item, Rating) rating matrix. These rating matrixes take many forms based on a different design. Some systems use rating such as 1–5 score, while others use like or dislike, and unary ratings, such as “has purchased”.

2.3.1 The Types of Collaborative Filtering Algorithm

Collaborative Filtering algorithm can be categorized into three classes as shown in Figure 2.5 including Memory-Based, Model-Based and Hybrid (Thorat et al., 2015; Dubey, Gupta, Raturi, & Saxena, 2018).

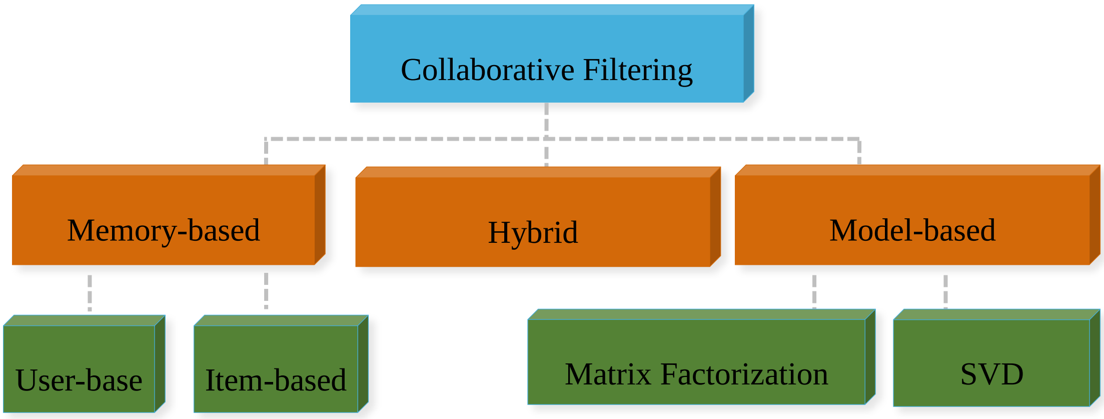


Figure 2.5 Types of Collaborative Filtering algorithm

There are two concepts in the Memory-based CF algorithm, User-based Collaborative Filtering (UBCF) algorithm and Item-based Collaborative Filtering (IBCF) algorithm. This research will focus on User-based Collaborative Filtering and IBCF and make improvement based on IBCF algorithm.

a. User-based Collaborative Filtering

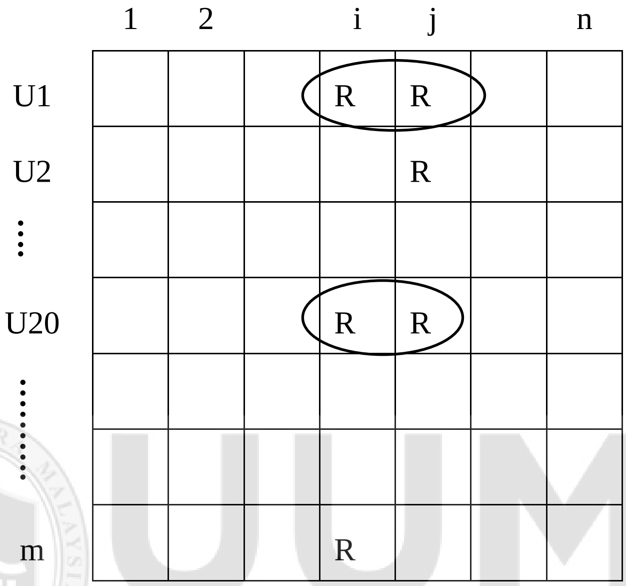
User-based Collaborative Filtering will find a similar user based on ratings and then predicts the user's rating on another item according to others' rating. The basic concept of the UBCF algorithm is to find similar users (neighbors) for the chosen user. In other words, the algorithm tries to find the greatest similarities in the whole user dataset. After comparing the similarity of the user to others, the system chooses the top k similar neighbors based on user similarity. Finally, the algorithm predicts the rating of the item that the user never viewed before, based on the rating history of neighbors and get the recommended results (Dou et al., 2017).

b. Item-based Collaborative Filtering

As the number of users increases, the UBCF suffers from scalability problem. In order to overcome this drawback, Sarwar, Karypis, Konstan and Riedl (2001) introduced a new method called the Item-based Collaborative Filtering (IBCF) algorithm. The Item-based Collaborative Filtering was developed to improve recommendation result and overcome scalability problem. Since then, IBCF has been popular and widely used in many fields by the giant company such as YouTube to make recommendations. A report estimated that 30 percent of Amazon's page views were generated from the IBCF recommendation approach (Smith & Linden, 2017).

Unlike the UBCF matching the user to a similar user, IBCF finds similar items based on the ratings, then finds most similar items and makes predictions. In order to calculate the most similar items on a targeted item, the algorithm draws from a similar-items matrix by finding two items that are both rated by the same customers, which means formulating an item to item matrix by iterating through all item pairs and computing a similarity score for each pair.

Alternatively, as shown in Figure 2.6, the Item similarity matrix is computed by looking into co-rated items only. For items *i* and *j* (each column present one item), the similarity value is computed by looking into them. Each of these co-rated pairs is obtained from different users (each row present one user). For this example, they come from user 1(U1) and user 20 (U20).



	1	2		i	j		n
U1				R	R		
U2					R		
⋮							
U20				R	R		
⋮							
m				R			

Figure 2.6 Rating Matrix

Once the system gets an Item-item similarity matrix, the algorithm will find the most similar items to the target item based on each of the user’s purchases and ratings and then recommends the most similar or correlated items. This computation is very simple and easy to implement because algorithm calculations rely only on ratings.

There are two keys step for the IBCF algorithm: similarity calculation and the prediction generation (Dou et al., 2017), making the IBCF recommendation results better than other methods .

a. Item Similarity Calculation

The item similarity calculation phase in the IBCF algorithm is to compute the similarity between every two items, the concept of calculating the similarity between two items *I* and *J* is first to find the users who have rated both of these items and then to applying a similarity formulation approach to compute the similarities.

There are several different approaches to compute the similarity between items. For example, there these approaches are Cosine and Pearson correlation similarities (Bobadilla et al., 2013; Dou et al., 2017).

1. Cosine-based Similarity

One of the most common methods to determine similarity is cosine similarity computation. Amazon recommendation system uses cosine similarity formulation to find the most similarity items between every two items and to generate the item-item similarity matrix (Sayyed, Argiddi, & Apte, 2013), the cosine similarity formulation between vector x and vector y as shown below :

$$\mathrm{COS}_{\mathrm{SIM(x,y)}} = \frac{x \cdot y}{\|x\|^2 * \|y\|^2} = \frac{\sum x_i y_i}{\sqrt{\sum x_i{}^2}\sqrt{\sum y_i{}^2}} \qquad (2.1)$$

2. Pearson Correlation Similarity

The other method to determine similarity is the Pearson correlation coefficient which is still the most popularly used to measure the similarities between two items (Thakkar et al., 2019), the Pearson correlation formulation between vector x and vector y as shown below :

$$\text{Pearson}_{\text{SIM(x,y)}} = \frac{(x-\bar{x})\cdot(y-\bar{y})}{\|(x-\bar{x})\|^2 * \|(y-\bar{y})\|^2} = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2}\sqrt{\sum(y_i-\bar{y})^2}} \quad (2.2)$$

b. Predict Recommendation

After the Item-item similarity matrix is obtained, the next step is to make the rating prediction of the target user. After the forward steps, the system obtains a similar item table and finds the most similar items based on the similarity measurement. The calculation of items *I* is predicted by computing the weighted sum of the ratings given by the user on the item similar to *I*. Each rating is weighted by the corresponding similarity between items *I* and *J*. Basically, this approach tries to capture how users rate similar items. The weighted sum is scaled by the sum of the similarity terms to make sure the prediction is within the predefined range, as shown in Figure 2.7, the target item predict rating will be computed as:

(4.5*0.7+3*0.4+5*0.8)/(0.7+0.4+0.8) = 4.4

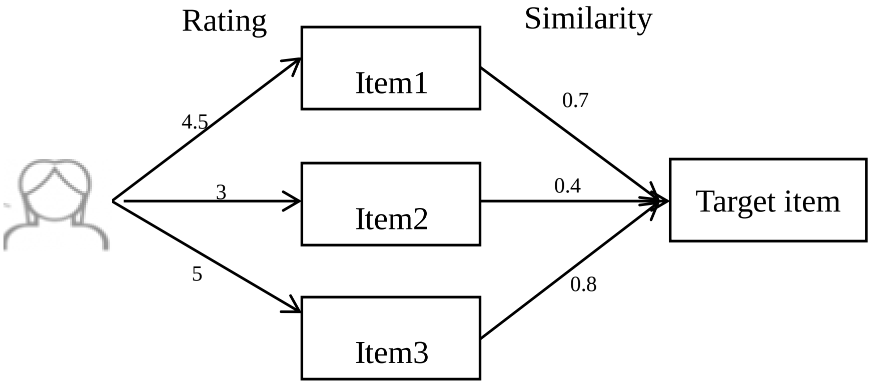


Figure 2.7 Predict calculation

2.3.2 The Previous and Current Improvement of Cold-Start Problems

As previously discussed in Section 2.2.3, the CF algorithm suffers from the cold-start problem caused by insufficient information about the new item/user. To overcome the cold-start problem, Zhu, Lin and He (2020) proposed a novel method for dealing with the cold-start problem, where item attributes are exploited to improve active learning in the recommendation system. At the same time, other research suggest utilizing new user demographic data to provide recommendations to avoid the cold-start problem (Safoury & Salah, 2013). On the other hand, Wang, Deng and Lai (2019) introducing the concept of innovators who can discover cold items without the help of the system. Therefore, cold items can be captured in the recommendation list via innovators. Some research also tends to combine the item attribute with the user's rating to overcome the cold-start (CS) problem (Z. long Li et al., 2018; Pirasteh et al., 2014).

According to Wei et al. (2017), they build a model combining the CF algorithm and deep learning neural network to ease the cold-start problem, the neural network part aims to extract the item features, and the CF model utilizes the time dynamics of the user preferences and item features, in order to consider the content information into prediction phase for cold-start items. Their experiment shows that the proposed model is effective for the cold-start problem. The author also points out that their model can be applied to other domain for the recommendation.

According to Wang (2019), the CF algorithm has been widely used to build recommendation systems because it can share collaborative insights and experiences from other users. However, these methods easily fall into the Matthew effect. This means that popular items have more chance to get recommended, and the cold items (less popular item) have become less and less popular. In this case, the result in the performance is severely degraded when the system is looking for cold items, and users may be interested, but the items are undiscoverable. Therefore, the research proposes a new CF algorithm, called Innovator-based CF, which can recommend cold items to users. By introducing innovators, who are special users, the cold items can be discovered by themselves without the help of the system. Therefore, cold items can be captured in the recommendation list via innovators to balance surprise and accuracy. Zhu et al. (2020) propose a method for overcoming the item cold-start problem, where items' attributes are exploited to improve active learning methods in the recommendation system. Firstly, the model will be trained to predict rating based on users' ratings and item attributes. Second, select a small portion of users to rate a new item. Thirdly, the prediction model is retrained by adding feedback ratings. Finally, unselected users' ratings are predicted by the re-trained model.

Meanwhile, combining CF algorithm with content-based algorithm is one significant methods to overcome cold-start problem, which takes advantage of both Content-based and Collaborative filtering algorithm (Pal, Parhi, & Aggarwal, 2018; Zhu et al., 2020; Fernández, Formoso, Cacheda, & Carneiro, 2020; Fan, Wu, Parvin, Beigi, & Pho, 2021). By combining two algorithms, the RS could utilize both content information (e.g., item attributes) and initial user ratings are valuable for seizing users' preferences on a new item. In a new item situation, the information about user tastes is certain, while user tastes about items are particularly important. Which means the user have a profile for each item, based on these item profiles, a content-based technique searches for items similar to those rated by the user. This way, overcoming cold-start problem with the new item problem searching for items similar to user tastes.

Based on previous research, to ease the cold-start problem in the recommendation system, further steps are needed to process cold item information, but it shows that these methods are time-consuming and costly, which means they need to track on user-behaviors to get feedback and to update their model. So in this study, the proposed method to alleviate the item cold-start problem in the IBCF algorithm is to make predication directly combine the item attributes in the item similarity calculation phase (Z. long Li et al., 2018), which no needs to re-train the model, in order to provide extended information for an algorithm to generate the prediction which eases the item cold-start problem.

2.3.3 The Collaborative filtering (CF) algorithm used for the RS

Today, the world-renowned recommendation system is developed by Amazon (Sayyed et al., 2013), which is known for its personalization and recommendation. The system helps customers discover hidden products that might fit their interest. Nearly 20 years ago (Linden et al., 2003) Amazon.com recommended more than a million items to millions of customers. With the success of the recommendations, collaborative filtering algorithms have been extended by most networks and other organizations, which is challenged by other algorithms and other techniques and used in different areas to improve the diversity and discovery of recommendations.

The recommendation system utilizes data analysis to assist customers to find the items in which they have a potential interest. Item recommendations can be made in different methods, like demographic-based method, best-selling items or predict rating based on customers past purchase habit. CF is the most successful recommendation technique (Sarwar et al., 2001). The CF recommendation is the earliest proposed and widest used method in the recommendation system (X. Li & Li, 2019). The algorithm not only finds out what user is interested in but also explore out the implicit information behind the data. As time goes by, the effect of the recommendation system can be improved significantly. Therefore, the collaborative filtering recommendation is one of the most popular recommendation technologies in the electronic commerce recommendation system.

A study (R. Sharma et al., 2017) shows several challenges to CF algorithm recommendation. One challenge is to improve the system scalability of the CF algorithm. Another challenge is to improve the quality of recommendations to users. From a particular perspective, these two challenges are conflicting because the shorter the time that the algorithm spent, the worse the scalability and the quality. An item-based alternative method CF algorithm has been developed to solve such a problem. The disadvantage of the traditional CF algorithm (UBCF) is that it has to compute the similarities between users frequently, which is very dynamic and computationally expensive. The item-based CF algorithm overcomes this shortcoming by discovering the relationships between items based on collaborative concepts rather than user relationships. And make calculation by looking for items with similar ratings from other users because it is easier to calculate the relationships between items. Also, the item-based CF algorithm can provide the same accuracy as the traditional CF algorithm while reducing calculation complexity.

The CF algorithm has been used to build recommendation systems, which are the essential method and underlying. While demographic-based algorithms show low accuracy on recommendations, it has a customer's potential interest that cannot be captured by the content-based approach. The main reason for this research is to improve the accuracy and reliability of the Item-based collaborative filtering algorithm. However, the weakness of the CF method (the cold-start problem, sparse data, scalability) has been discussed previously. In order to relieve the cold-start problem, this paper introduces an item similarity calculation function, in which a similarity calculation is performed to calculate the correlation between items through the CF algorithm. Therefore, the new item has enough information to generate sufficient information that can be captured and recommended to the user in the recommendation list in order to verify the effectiveness of the new CF algorithm.

2.4 The Evaluation Metrics used in Recommendation system

The important thing in order to fulfill research objectives and requirement of methods are the evaluation metrics. The function of evaluation is to predict the quality of the proposed algorithm used in recommendation system. The researches (Fan et al., 2021; Zhu et al., 2020) utilize Mean Absolute Error (MAE) among the actual and predicted ratings as evaluation metrics as 2.3, where I represents item I, J represents items J,

R represents actual rating, while $\check{R}$ represents predicted rating.

$$\text{MAE} = \frac{\sum_i \sum_j |R_{IJ} - \check{R}_{IJ}|}{Testsize} \qquad (2.3)$$

Precision and recall are computed from a 2×2 table, such as the one shown in Table 2.3. The item set must be separated into two classes relevant or not relevant. That is, if the rating scale is not already binary, need to transform it into a binary scale. For example, the MovieLens dataset has a rating scale of 1–5 and is commonly transformed into a binary scale by converting every rating of 4 or 5 to “like” and all ratings of 1–3 to “don’t like.” For precision and recall, we also need to separate the item set into the set that was returned to the user (selected/recommended), and the set that was not.

Table 2.3 *Confusion Matrix*

	Like	Don’t like
Predict like	True Positive	False Positive
Predict don’t like	False Negative	True Negative

1. Precision is defined as the number of correctly recommended items (i.e. the number of preferred items existing in the recommendation list) divided by the number of all recommended items.
2. Recall is defined as the number of correctly recommended items divided by the total number of items which should be recommended.

The F1-score metric considers both Precision and Recall rate, calculated from the precision and recall of the test, where the precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive. The highest possible value of an F1-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if either the precision or the recall is zero.

Based on above discussion, the evaluation metrics used in this research are precision rate, recall rate and F1-score, which aims to evaluate the effective and quantity of the proposed algorithm in making trade-off between new item recommendation and accuracy.

2.5 Summary

This chapter explained the review of the recommendation system and collaborative filtering algorithm used in the recommendation. The chapter also discusses the previous and current solution of the cold-start problem. In order to design a more effective IBCF, a new similarity calculation phase is proposed to overcome the cold-start problem. In addition, this chapter also discusses the important role played by recommendation system in the E-commerce area. The next chapter will explain the methodology applied in this study.

CHAPTER THREE
RESEARCH METHODOLOGY

3.1 Introduction

This chapter describes the research methodology applied in this study to realize the objectives related to improve the item-based collaborative filtering algorithm in recommendation system to reduce cold-start problem. The enhancement of algorithm is proposed to make a better product recommendation. The main contribution of this research is to utilize more eigenvector to overcome cold-item problem on item recommendation phase. The research procedures in Figure 3.1 show the research phases, activities, methods, and outcomes for each phase.

3.2 Preliminary Study

This section discusses a preliminary study phase of this study. The phase begins with problems identification based on the previous research. The literature review was conducted to analyze the existing research in a recommendation domain in order to identify issues that need further investigation. This phase identified that Item-Based Collaborative Filtering (IBCF) algorithm suffers from many drawbacks, which may lead to the recommendation system's inaccuracy when cold-start problem occurs. Thus, the outcome of this phase answered the first research objective for this study.

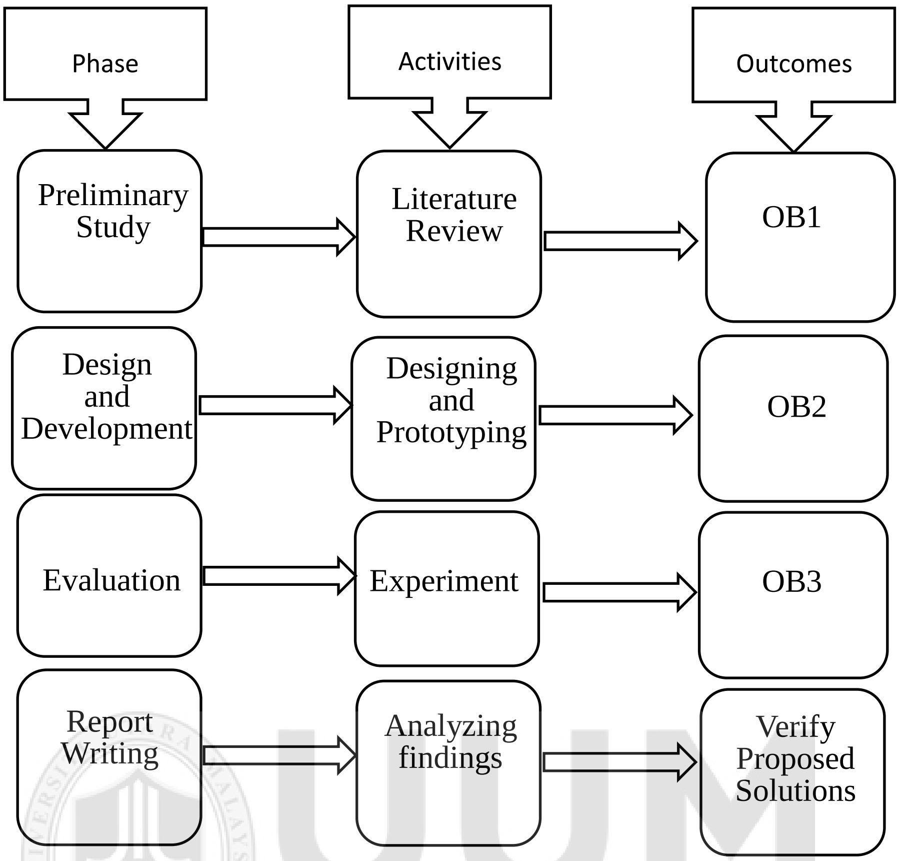


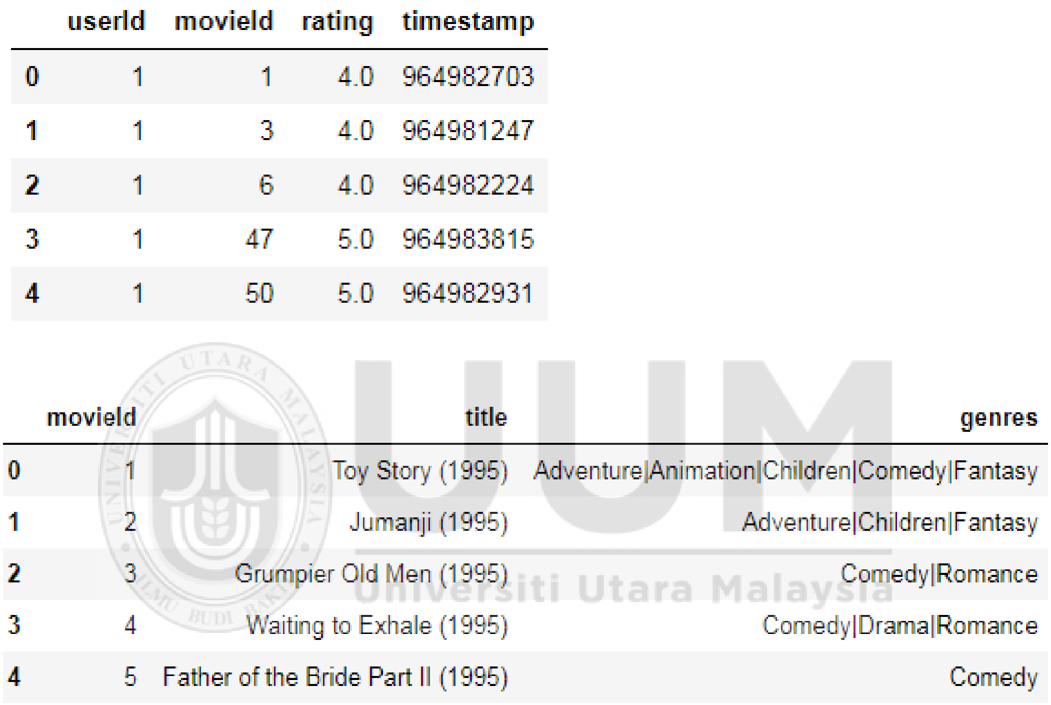
Figure 3.1 Research design

3.3 Design and Development

The Item-Based Collaborative Filtering (IBCF) algorithm concept was inspired by Sarwar et.al. (2001). which has given a clear concept of the algorithm. In this study, the improvement of the IBCF algorithm is based on the work proposed by Li, Huang and Zhang (2018) and the implementation of the algorithm was using Python Version 3.7.4 on Jupyter Notebook 6.0.3 IDE.

3.3.1 Data Collection

The datasets used in this study were collected from the MovieLens dataset provided by the Minnesota University Study Group (https://grouplens.org/datasets/movielens/). The dataset contains 100,000 ratings from 610 users on 9724 movies, and there are at least 20 ratings for each user, and the dataset also include information about the item. The dataset pattern is shown in *Figure 3.2*.



	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

Figure 3.2 Sample of the data

3.3.2 Identify the Item-based Collaborative Filtering Algorithm

The implementation of the previous steps helps the process of making a feasible plan. Moreover, most of the related work mentioned in Chapter Two was considered to gain a thorough understanding of Item-based CF algorithms. The existing recommendation system helps guide instructions about the research design principle to determine the appropriate methods for algorithm testing. Nowadays, E-commerce organizations need to provide various products that meet the multiple needs of multiple consumers. It is said by the CEO of Amazon that “If I have 3 million customers on the Web, I should have 3 million stores on the Web” (Schafer et al., 2001) to realize a personalized recommendation. The collaborative filtering algorithm is an important foundation, despite suffering from many drawbacks. Nonetheless, the traditional Item-based CF algorithm cannot make a recommendation on new items. This study aims to enhance the Item-based CF algorithm by adding a new function by calculating the similarity between item attributes, which involves a new parameter in the classification function to give information to the algorithm based on the new item’s attributes. The original Item-based CF algorithm design as below:

1. First, build the rating matrix, which rows length equal to the user’s quantity and columns as to item’s quantity.

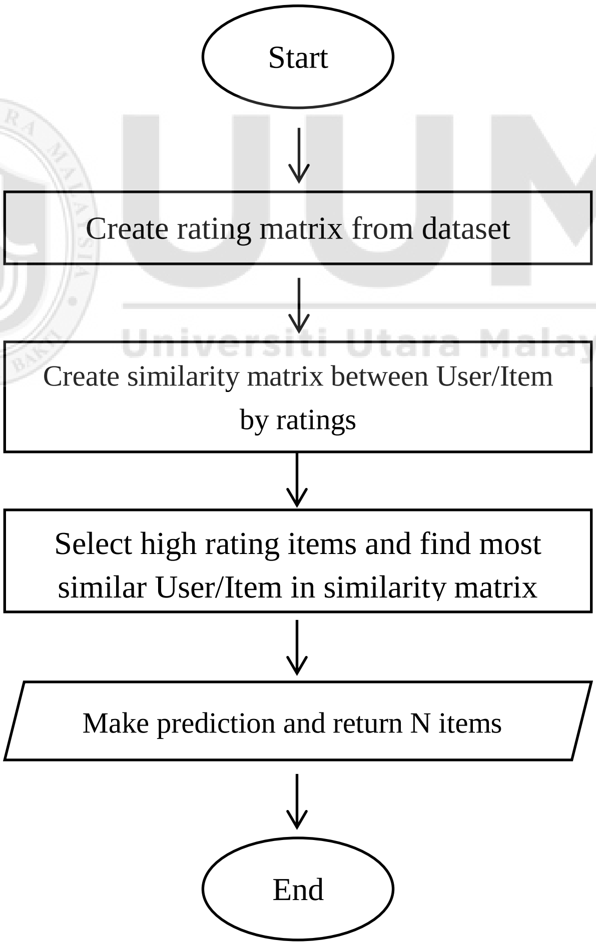
2. Then, build an item similarity matrix and assign it by calculating the similarities between the two items using the Pearson correlation coefficient formulation.

3. After similarity calculation, a similarity matrix is produced. The final step is to find the items which had user ratings similar to the original dataset (user’s

rating >= 4), use weighted value to predict the rating of target users or totalize the similarity matrix by horizontal stack, and returning to the TOP k similarity items to make a recommendation list.

3.3.3 The Design of New Similarity Function

In traditional CF algorithm, both Item-based and User-based CF algorithm need ratings to calculate the similarities between items or users, the algorithm flowchart is as shown in *Figure* 3.3.



Start
Create rating matrix from dataset
Create similarity matrix between User/Item by ratings
Select high rating items and find most similar User/Item in similarity matrix
Make prediction and return N items
End

Figure 3.3 The Flowchart of Traditional CF Algorithm

Figure 3.3 shows the flowchart of the traditional CF algorithm. This algorithm starts with inputting the dataset and creating a rating matrix, then calculating the similarities between User/Item by utilizing these ratings to create similarity matrix, finding the user's preference, and returned *N* items before making a recommendation.

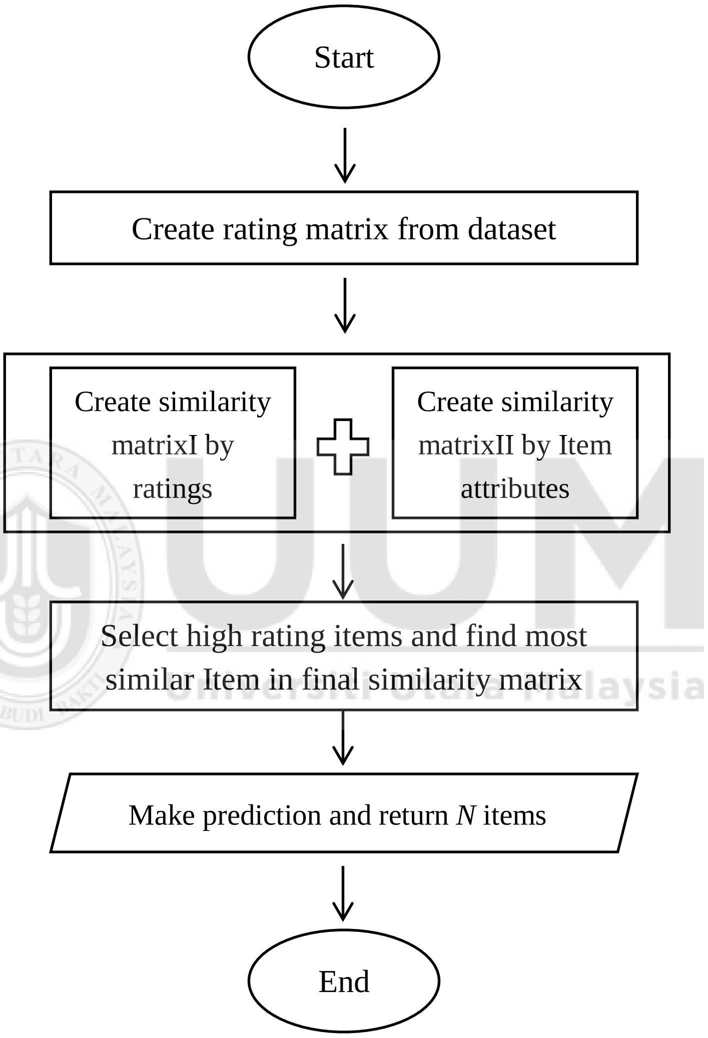

Start
Create rating matrix from dataset
Create similarity matrixI by ratings
Create similarity matrixII by Item attributes
Select high rating items and find most similar Item in final similarity matrix
Make prediction and return *N* items
End

Figure 3.4 The Proposed IBCF Algorithm

Figure 3.4 shows a proposed flowchart where it involves a new calculation processing phase and displaying ways to discover the similarity between existing items and new items. It illustrates the new flowchart of the IBCF algorithm that can extract other information that calculates the similarity between old items and new items. The new calculation processing phase could ease the cold-start problem by utilizing item attributes and make a recommendation based on new items.

3.3.4 Item Attribute Calculation

The attributes of the items provide the user's interest and preferences, and the user's interest can be obtained by analyzing the attribute characteristics of the items. In normal situation, item attributes always include color, item class, size, brand, and other relevant attributes. Assuming that an item has *n* attributes, the matrix of item attributes shown in Table 3.1.

Table 3.1

The matrix of item attributes

Items	**Attributes**			
	Attribute I	**Attribute II**		**Attribute…**
Item1	0	1	…..	0
Item2	1	0	…..	1
…	0	0	…..	0
Item10	0	0	…..	1
…..	1	1	…..	0

Table 3.1 illustrates the items and attributes of the items; if the item has the corresponding attribute, the value will be 1 otherwise 0. As mentioned previously, Li, Huang and Zhang (2018) proposed an attribute similarity formulation as 3.1. While this study adopted a different attribute similarity formulation as 3.2, the similarity calculation formulation can be defined as follow:

$$\text{ASim}\left(I_i, I_j\right) = \frac{N^{I_i \cap I_j}}{N - N^{I_i \cup I_j}} \tag{3.1}$$

$$\text{ASim}\left(I_i, I_j\right) = \frac{N^{I_i \cap I_j}}{N^{I_i \cup I_j}} \tag{3.2}$$

Where N represents all attributes in the dataset, $N^{I_i \cap I_j}$ represents the number of attributes both I_i and I_j have, while $N^{I_i \cup I_j}$ represents all attributes belong to I_i and I_j. In Li et al (2018) research explanation, $N - N^{I_i \cup I_j}$ represents the total attributes set which neither belong to I_i nor I_j, so the equation 3.1 means the attributes both belongs to I_i and I_j divide the attributes neither belong to I_i nor I_j. On the contrary, the equation 3.2 which also knows as intersection-over-union between I_i and I_j, which means the attributes both belong to I_i and I_j divide all attributes belong to I_i and I_j, which is more reasonable and intuitive than previous research.

For instance, if two items have the following attributes, as shown in *Table* 3.2. The similarity attributes calculation between Item1 and Item2 is $\text{Sim}\left(I_i, I_j\right) = \frac{2}{7}$.

Table 3.2

Example of attributes matrix

	A1	A2	A3	A4	A5	A6	A7	A8	A9
Item1	0	1	0	1	0	1	1	0	0
Item2	1	1	0	0	1	1	0	1	0

But for some occasion, vector A={1,0,0,0,0}, B={1,0,0,0,0}, C={1,0,1,1,1}, D={1,0,1,1,1}, the attribute similarity between A and B, C and D are both 1, but the latter apparently has higher similar than former, and a weighted function can be described as follow:

$$\omega = \ln(1 + (N^{I_i \cap I_j})) \tag{3.3}$$

The final attribute similarity can be:

$$\text{ASim}(I_i, I_j) = \frac{N^{I_i \cap I_j}}{N^{I_i \cup I_j}} * \ln(1 + (N^{I_i \cap I_j})) \tag{3.4}$$

For vector A={1,0,0,0,0}, B={1,0,0,0,0}, C={1,0,1,1,1}, D={1,0,1,1,1}, the attribute similarity between A and B is 1*ln(2)=0.7, C and D is 1*ln(5) = 1.6, which has a better interpretability for attribute similarity than before, and for $N^{I_i \cap I_j}$=0 the ln(1+0) is 0, so it also gives a good explanation to 0 similarity

3.3.5 Similarity Calculation

To combine the similarity calculation between $\text{Pearson}_{\text{SIM}(I_i, I_j)}(Psim)$ and $\text{Attribute}_{\text{SIM}(I_i, I_j)}(Asim)$. The computation of the final similarity can be defined as:

$$\text{Final Sim}(I_i, I_j) = \frac{Psim^2 + Asim^2}{Psim + Asim} \tag{3.5}$$

Based on the final similarity formulation, the recommendation system can recommend new items, which means that if there is no rating about the item, the Psim tend to be 0, and the final similarity will be calculated totally by Asim calculation. In contrast, the old item can make a recommendation on two similarity calculation, which enhances the accuracy of the recommendation.

3.4 Evaluation

The Item-based Collaborative filtering algorithm was improved and implemented in the algorithm design phase to ensure better recommendation outcome. Chapter 2 has listed the previous CF algorithms by providing reasons for drawbacks of the typical item-based CF algorithm. The proposed experimental design is to validate the proposed algorithm, since the traditional item-based CF algorithm worked based on user ratings, if there is a new item that has none or fewer ratings provided to the algorithm, the algorithm cannot work efficiently. In this research, an enhanced similarity calculation phase by utilizing item attributes is proposed, which can build the similarity matrix by combining the user's ratings with item attributes. This kind of model can alleviate the cold-start problem, as mentioned. Because the cold-start problem caused by a lack of information the algorithm on new items, after the new similarity calculation, the new function can provide enough information for the algorithm to make a recommendation that relieves the cold-start problem. The experimental design will discuss in the next chapter, which provides strong evidence of this research relieving the cold-start problem because the proposed IBCF algorithm can achieve almost the same performance as traditional IBCF while making recommendations on new items.

3.5 Analyzing Findings

Finally, Chapter Four will analyze the experiment result, which validates the outcome of this study. The IBCF algorithm will be improved through the proposed method and evaluated on the Movielens dataset, and the evaluation metrics are precision rate, recall rate, F1 score, the performance evaluation between IBCF, IRACF, and proposed IBCF will be compared in the experiment phase.

3.6 Summary

This chapter explains the method in detail to achieve the objectives of the study. It starts by introducing the overview of the research procedures divided into four major phases: study literature review, algorithm design. The experimental design and evaluation phases will be explained in the next chapter.

CHAPTER FOUR
EXPERIMENT DESIGN AND VALIDATION

4.1 Introduction

In this chapter, an experiment aims to evaluate the proposed IBCF algorithm. It is followed by three experiments that produce validation results. In the experiment phase, the proposed IBCF algorithm was compared to the traditional Item-based CF and IRACF from Li et al (2018). The results were analyzed and discussed in this chapter.

4.2 Data Preprocessing

In preprocessing phase, the dataset was designed in two forms including normal datasets (includes user ID, item ID and rating) and cold-item datasets (randomly drop different percentage of the items). This study aims to relieve the cold-start problem to item-based CF algorithm, which means comparing the different algorithms within the existing item and cold-item dataset to achieve the designed research objective.

4.3 Experiment Design

This section presents the experiment design and metrics of evaluation performance on different CF algorithm. This is followed by three segments that produce results as shown in Figure 4.1.

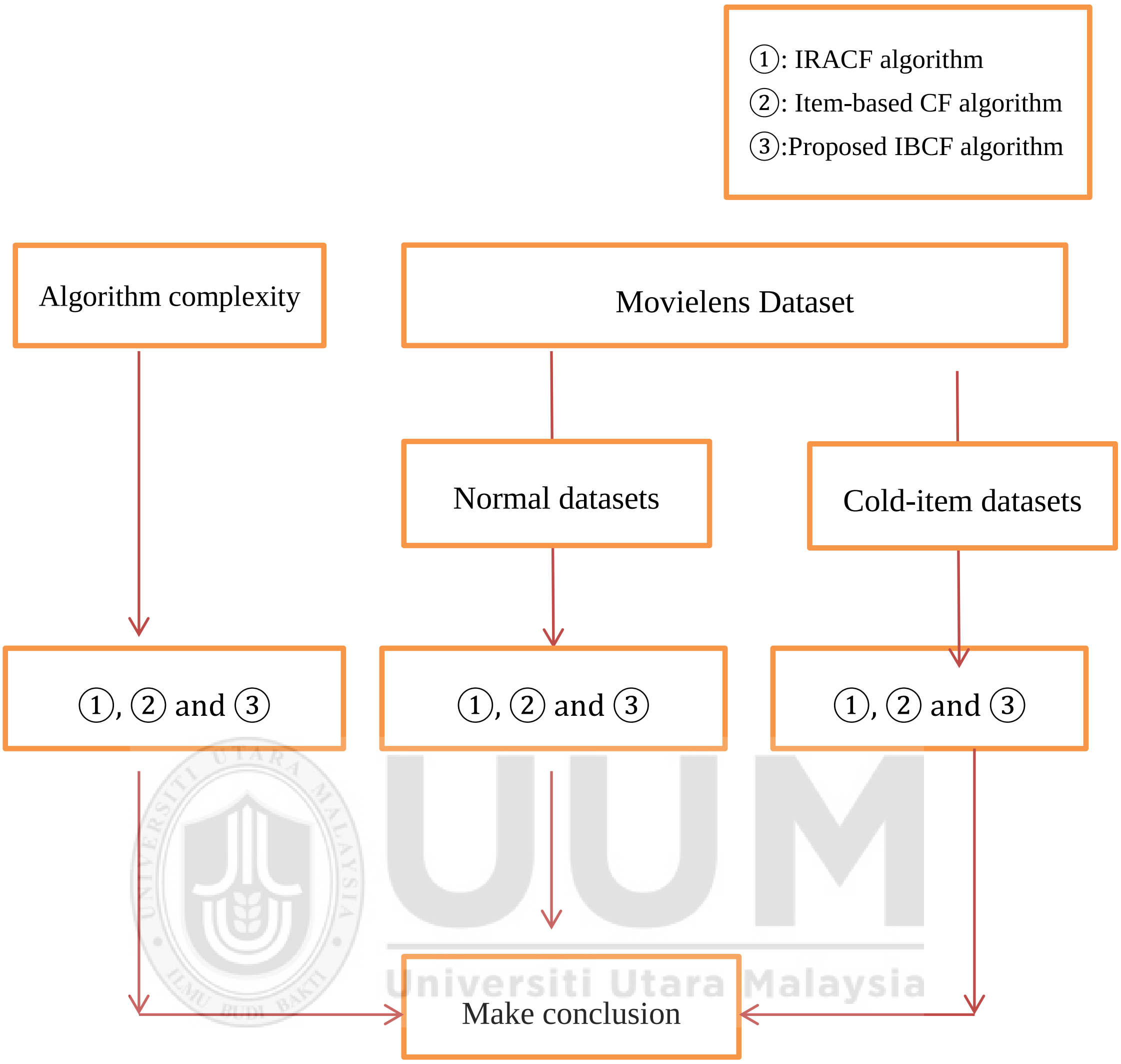


Figure 4.1 The Flowchart of the Experiment

1. The left branch shows that the proposed IBCF algorithm was compared the algorithm complexity to other CF algorithms in space complexity and time complexity, demonstrating the complexity between the proposed CF and other CF algorithms.

2. The middle branch shows that the proposed IBCF algorithm was compared to the traditional IBCF algorithm and IRACF algorithm in the different split percentage of

Benchmark dataset (normal dataset). The benchmark dataset is critical for developing, evaluating, and comparing machine learning algorithms, which can present a better reflection of algorithm ability.

3. The right branch shows that the proposed IBCF algorithm was compared to the traditional IBCF algorithm and IRACF algorithm in the cold-item dataset, the dataset contains new items. In the preprocessing phase, the dataset was randomly dropped some different percentage of items in the training dataset in order to simulate the cold-start problem. This branch demonstrates that the traditional IBCF algorithm suffer from the cold-start problem; on the contrary, the proposed IBCF can ease the cold-start problem and have better performance than IRACF.

4.3.1 Evaluation Methodology

Before comparing the Algorithms, the evaluation metrics are precision rate, recall rate and F1 score in the confusion matrix. In CF algorithm, the algorithm cannot generate negative label, because there is no sense that define the rest of items in the dataset are user does not like, so as evaluation metrics the accuracy rate cannot calculated by CF algorithm.

- Precision rate: $\frac{TP}{TP+FP}$
- Recall rate: $\frac{TP}{TP+FN}$
- F1 Score: $2\cdot \frac{Precision*recall}{Precision+ recall}$

4.4 Experiment Result

A. Compare Algorithm Complexity between Item-based CF Algorithm, IRACF Algorithm and Proposed IBCF Algorithm

In IBCF algorithm, which tries to calculate the similarity between each item, the algorithm was run as most as $\frac{1}{2}N^2$ times (shown in Table 4.1), the time complexity of IBCF can be defined as $\mathrm{O}(N^2)$ and the space complexity also can be represented as $\mathrm{O}(N^2)$. While the proposed IBCF algorithm calculates the similarity between each item and attributes, the proposed IBCF algorithm was run as most as N^2, the time complexity of the proposed IBCF algorithm same as IRACF and also follows the $\mathrm{O}(N^2)$.

Table 4.1

The Similarity Matrix

	Item1	Item2	…..	…..	Item N-1	Item N
Item1	1	..	..	..	..	..
Item2	..	1	..	..	..	..
…..	..	..	1	..	..	..
…..	..	..	..	1	..	..
Item N-1	..	..	..	..	1	..
Item N	..	..	..	..	..	1

This experiment demonstrates the algorithm complexity between IBCF and proposed IBCF, it can be seen that the proposed IBCF algorithm has the same complexity as IBCF and IRACF, which time and space complexity also follow $\mathrm{O}(N^2)$, provide a piece of evidence that the proposed improvement does not increase the computational complexity

B. Compare Proposed Algorithm with IRACF and Item-based CF Algorithm in the Stable Dataset

In this experiment, a benchmark dataset provided by Movielens has been used for better recommendation result. The IRACF, Item-based CF and proposed item-based CF were compared in the different split percentage of the dataset (training: testing) ratios, e.g., 50:50. 60:40 and 70:30, testing in multiple rounds are performed using different partitions, aims to validate the experiment results is not caused by chance. The evaluation metrics involve Precision rate, Recall rate, and F1 Score. The experiment results are summarized in Table 4.2.

Table 4.2

The Experiment results for different CF algorithms based on training and testing ratios

	Precision	Recall	F1
Percentage (Training: Testing)		1:9	
IRACF	0.29	0.01	0.02
Item-based CF	0.28	0.02	0.04
Proposed CF	0.22	0.02	0.04
		2:8	
IRACF	0.35	0.02	0.05
Item-based CF	0.46	0.02	0.04

Proposed CF	0.39	0.02	0.04
		3:7	
IRACF	0.50	0.03	0.06
Item-based CF	0.50	0.03	0.06
Proposed CF	0.47	0.03	0.06
		4:6	
IRACF	0.24	0.04	0.06
Item-based CF	0.52	0.04	0.07
Proposed CF	0.46	0.03	0.06
		5:5	
IRACF	0.17	0.04	0.07
Item-based CF	0.48	0.05	0.09
Proposed CF	0.45	0.04	0.07
		6:4	
IRACF	0.14	0.03	0.07
Item-based CF	0.44	0.05	0.09
Proposed CF	0.38	0.04	0.07
		7:3	
IRACF	0.10	0.04	0.07
Item-based CF	0.38	0.06	0.1
Proposed CF	0.34	0.04	0.07
		8:2	
IRACF	0.09	0.05	0.07

Item-based CF	0.29	0.05	0.09
Proposed CF	0.27	0.04	0.07
		9:1	
IRACF	0.04	0.04	0.04
Item-based CF	0.18	0.04	0.07
Proposed CF	0.16	0.03	0.05

Table 4.2 illustrates the result of precision, recall, and F1 score of three algorithms test in benchmark dataset. As shown in the table, the IRACF almost has same performance than proposed IBCF. The Item-based CF always shows good preference in precision rate and gets best F1 score in 7:3 dataset. While proposed CF algorithm reaches best F1 score in 5:5 dataset, all algorithms get low recall, because the quantity of recommendation list reaches in a large scale, which causes the recall tends to be low.

The Figure 4.2 also shows the F1 score between three algorithms in different ratios of the dataset, with the training data ratio increasing, the three algorithms all follow upward trend, and due to the dataset has less users (610 users) than items(9724 items), three algorithms shows good Precision rate in different split dataset. At the same time, IBCF get best performance in this dataset, because large proportion of items makes item similarity matrix works well. As to proposed IBCF and IRACF, the algorithms combine with item attributes and tent to make new items recommendation, but there is no record about the new items in the dataset, which may cause the lower F1 score than traditional IBCF.

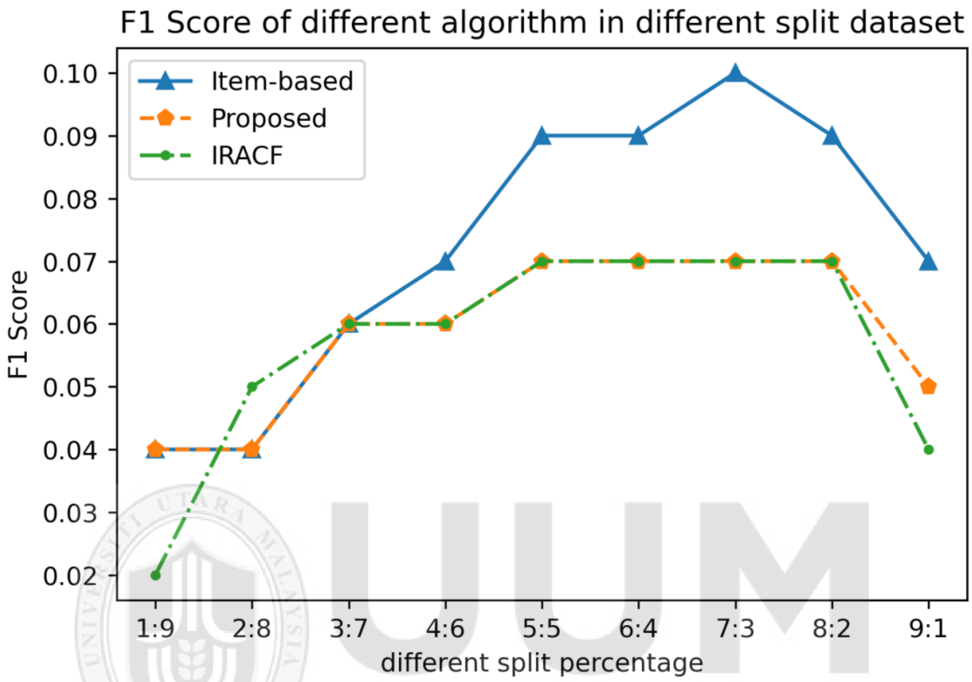


Figure 4.2 The F1 Score between different algorithm and different split percentage of the dataset

In summary, the proposed IBCF algorithm shows same performance than IRACF and IBCF get better recommendation result than other two algorithms. However, IBCF cannot work well on a dataset containing cold items in the next experiment. In contrast, the proposed IBCF algorithm and IRACF can overcome the cold item issue, which achieves a balace between stability and novelty.

C. Using proposed IBCF compare with IRACF and IBCF to make a recommendation in the dataset contains cold items

In this experiment, the proposed IBCF algorithm, IRACF and IBCF were tested in the dataset which contains cold items. To obtain the cold-items dataset, at first, splitting the dataset into training dataset and testing dataset, then random choice different percentages of the items and drop relevant item records form training dataset, after that, these items in testing dataset was totally become cold-items (because there was no ratings about these items in training dataset) which can be defined these items as cold or new items, and tests three algorithms in cold-item dataset to simulate cold-item recommendation. The experiment result shown as Table 4.3,

Table 4.3

The Experiment results in a cold-item dataset

	Precision	recall	F1
		1% Cold-Items	
IRACF	0.15	0.06	0.09
IBCF	0.30	0.02	0.04
Proposed	0.15	0.06	0.09
		2% Cold-Items	
IRACF	0.14	0.03	0.05
IBCF	0.20	0.01	0.02
Proposed	0.14	0.03	0.05
		3% Cold-Items	

IRACF	0.07	0.03	0.04
IBCF	0.20	0.01	0.02
Proposed	0.08	0.03	0.04
		4% Cold-Items	
IRACF	0.07	0.018	0.03
IBCF	0.18	0.005	0.01
Proposed	0.08	0.02	0.03
		5% Cold-Items	
IRACF	0.06	0.009	0.016
IBCF	0.11	0.004	0.008
Proposed	0.08	0.01	0.02
		6% Cold-Items	
IRACF	0.06	0.009	0.016
IBCF	0.08	0.002	0.004
Proposed	0.06	0.01	0.017
		7% Cold-Items	
IRACF	0.06	0.008	0.014
IBCF	0.08	0.003	0.006
Proposed	0.06	0.01	0.017
		8% Cold-Items	
IRACF	0.06	0.005	0.010
IBCF	0.07	0.002	0.004
Proposed	0.06	0.007	0.013
		9% Cold-Items	
IRACF	0.05	0.004	0.007
IBCF	0.05	0.002	0.004
Proposed	0.05	0.005	0.009
		10% Cold-Items	

IRACF	0.04	0.003	0.005
IBCF	0.04	0.002	0.004
Proposed	0.04	0.004	0.007

Table 4.3 shows the traditional IBCF suffers from cold-start problems and cannot work well on the cold-item dataset. On the contrary, the proposed IBCF and IRACF can overcome the cold-start problem and get almost the same preference on cold-item dataset. Figure 4.3 shows the F1 score for all three algorithms.

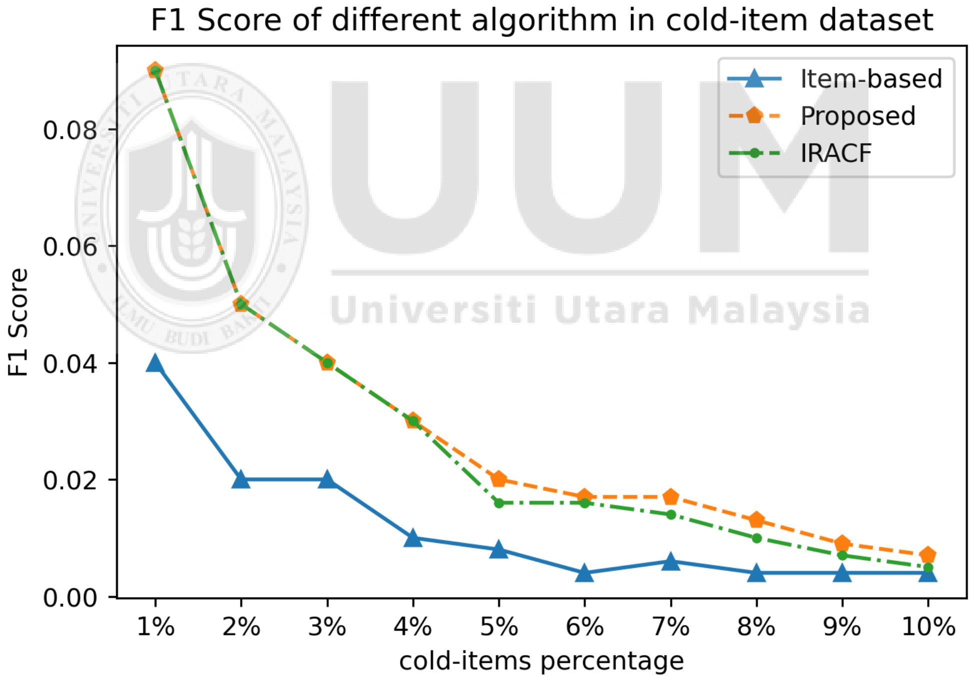


Figure 4.3 The F1 score between three algorithms in the cold-item dataset

Figure 4.3 demonstrate that with the cold-item percentage increasing, the three algorithm shows downtrend in F1 score because there is not enough information or item record provided to algorithms to work well. Nevertheless, comparing to the experiment Two, it can be found that traditional IBCF suffers from a cold-start problem, the performance gets a greater decrease in a cold-item dataset, at the same time, the proposed IBCF algorithm shows better than the IBCF, and have a little bit improvement on F1 score than IRACF, which due to the improvement of the similar calculation phase. This experiment proved that the proposed IBCF algorithm has the ability to overcome the cold-start problem and get better performance than IRACF.

.

To summarize, this section compares algorithm complexity and performances for three algorithms in the different dataset. The results show that the proposed IBCF algorithm can reduce the cold–start problem against traditional IBCF by adding a new phase to provide more content information to the algorithm, at the same time performance well than IRACF, which achieve the objectives of this study.

4.5 Summary

This chapter provides a detail of the experiment result. The evaluation phase followed by the performance in the algorithms based on Precision, Recall and F1 score rates. Based on these results, it shows that the proposed IBCF algorithm can overcome the cold-start problem. In general, this chapter compares the result of validation performance of IRACF algorithm, Item-based CF algorithm and proposed IBCF algorithm.

CHAPTER FIVE
CONCLUSION

5.1 Introduction

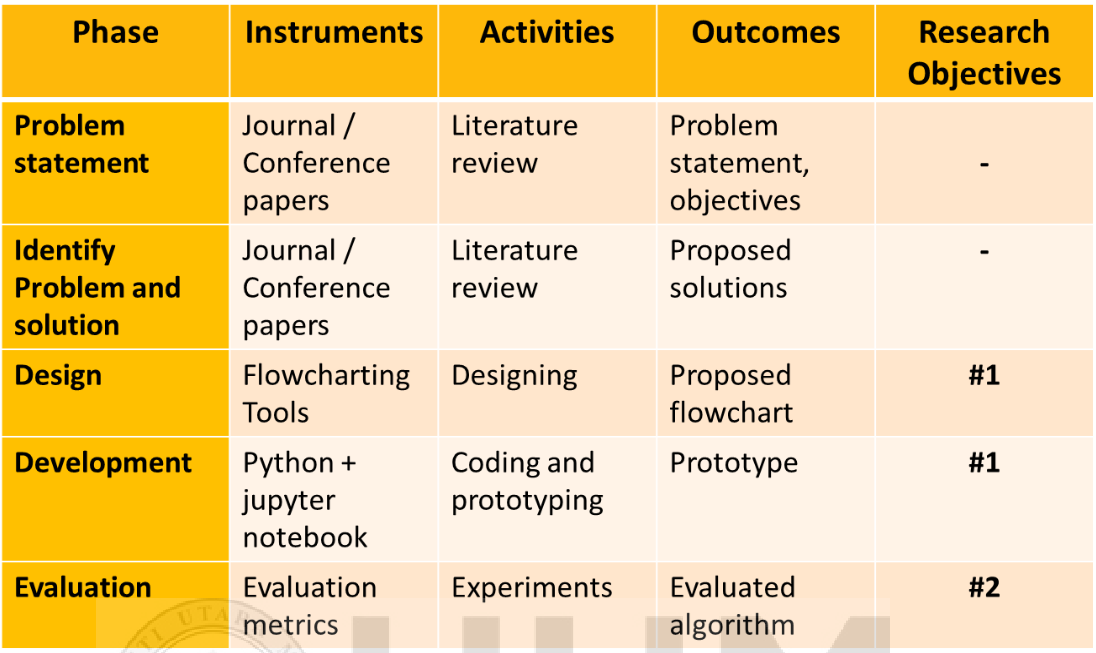
This chapter presents the objectives of this research with the contribution to the CF algorithm in the recommendation system. An improved IBCF algorithm model has been designed and developed as research objectives, and the evaluation phase of the algorithm also discussed in chapter 4. This chapter also includes the major contributions, limitation of the study and recommendation for future work.

5.2 Research Objectives

This study has achieved all objectives that mentioned in Chapter 1 as shown in Figure 5.1

5.2.1 Objectives 1: To develop an improvement of the IBCF algorithm

Based on the previous research by Li et. al. (2018), the method to overcome the cold-start problem is to provide another item attributes to the algorithm for new items recommendation. Based on that, a proposed IBCF algorithm has been developed and explained in Chapter 3. This study adopted a different item attribute similarity formulation than previous research. By representing the item attribute similarity as more reasonable and intuitive, this study design different experiments to evaluate the performance of the proposed algorithm.



Phase	Instruments	Activities	Outcomes	Research Objectives
Problem statement	Journal / Conference papers	Literature review	Problem statement, objectives	-
Identify Problem and solution	Journal / Conference papers	Literature review	Proposed solutions	-
Design	Flowcharting Tools	Designing	Proposed flowchart	**#1**
Development	Python + jupyter notebook	Coding and prototyping	Prototype	**#1**
Evaluation	Evaluation metrics	Experiments	Evaluated algorithm	**#2**

Figure 5.1 Research Objective Review

Compared to traditional IBCF algorithm, the proposed IBCF algorithm has introduced an additional phase to utilize more vectors to calculate the similarity between existing and new items, which provides additional information about the new items in computing phase to reduce issues existed in the cold item recommendation part.

5.2.2 Objectives 2: To validate the proposed CF algorithm with data

In this study, the result of evaluation performance is validated based on experiments. The proposed IBCF algorithm was compared with IBCF and IRACF in algorithm complexity, precision rate, recall rate and F1 score to ensure the better performance of the proposed algorithm. Chapter 4 elaborates detail of the comparison results. Based on the experiment, the proposed IBCF algorithm gets a better performance than IRACF and IBCF, while overcoming the cold-start problem and receives a better recommendation result in datasets which contains cold-item.

5.3 Research Contribution

Improvement of CF algorithm in recommendation system in this study has contributed to both theoretical and practical as described in the following sub sections.

5.3.1 Theoretical Contribution

There are two contributions in this study that have been identified as theoretical contribution such as:

1. Providing a solution to overcome the cold-start problem in the recommendation system by analyzing more data features to extract more information to the algorithm.

2. Providing an evaluation performance based on validation in recommendation algorithm. This can be used as the evaluation measurement in other algorithm validation.

5.3.2 Practical Contribution

There are two contributions in practical aspect to the recommendation system

1. Through improving the quantity of recommendation system, the proposed IBCF can be applied in many filed for cold-items recommendation.

2. Give an insight to deal with new items problem, by combine items rating and items attributes to overcome cold-start problem.

5.4 Limitations and Recommendation

The result obtained in this study is convincing. However, a few factors may have influenced the generalizability. Here are some factors, which may be possibly improved in future.

- The data were collected from Movielens in one hundred thousand data size. Hence, future studies should consider more scale and more quantity of dataset.
- To integrated other algorithm with the CF algorithm to extract more hidden information behind the data in order to enhance the recommendation results.

5.5 Summary

In conclusion, this chapter has discussed and concluded the overall research, the achievement of research objective, the research contribution of theoretical and practical aspects, the research limitations, and the recommendations for further development. Furthermore, it also contributes to a recommendation system in reducing the cold-start problem.

REFERENCES

Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749. https://doi.org/10.1109/TKDE.2005.99

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

Dou, Y., Yang, H., & Deng, X. (2017). A Survey of Collaborative Filtering Algorithms for Social Recommender Systems. *Proceedings - 2016 12th International Conference on Semantics, Knowledge and Grids, SKG 2016*, 40–46. https://doi.org/10.1109/SKG.2016.014

Dubey, A., Gupta, A., Raturi, N., & Saxena, P. (2018). *Item-Based Collaborative Filtering Using Sentiment Analysis of User Reviews*. https://doi.org/10.1007/978-981-13-2035-4_8

Ekstrand, M. D., Riedl, John, & Konstan, J. A. (2011). Collaborative Filtering Recommender Systems. *Foundations and Trends® in Human–Computer Interaction*, *4*(2), 81–173. https://doi.org/10.1561/1100000009

Fan, H., Wu, K., Parvin, H., Beigi, A., & Pho, K. H. (2021). A Hybrid Recommender System Using KNN and Clustering. *International Journal of Information Technology and Decision Making*, *20*(2), 553–596. https://doi.org/10.1142/S021962202150005X

Fern ández, Di., Formoso, V., Cacheda, F., & Carneiro, V. (2020). A Content-Based Approach to Profile Expansion. *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, *28*(6), 981–1002. https://doi.org/10.1142/S0218488520500385

Gandhi, S., & Gandhi, M. (2018). Hybrid Recommendation System with Collaborative Filtering and Association Rule Mining Using Big Data. *2018 3rd International Conference for Convergence in Technology, I2CT 2018*, 1–5. https://doi.org/10.1109/I2CT.2018.8529683

Guo, Y., Yin, C., Li, M., Ren, X., & Liu, P. (2018). Mobile e-Commerce Recommendation System Based on Multi-Source Information Fusion for Sustainable e-Business. *Sustainability*, *10*(2), 147.

https://doi.org/10.3390/su10010147

Iwendi, C., Ibeke, E., Eggoni, H., Velagala, S., & Srivastava, G. (2021). Pointer-Based Item-to-Item Collaborative Filtering Recommendation System Using a Machine Learning Model. *International Journal of Information Technology & Decision Making*, *20*, 1–22. https://doi.org/10.1142/s0219622021500619

Jiang, L., Cheng, Y., Yang, L., Li, J., Yan, H., & Wang, X. (2019). A trust-based collaborative filtering algorithm for E-commerce recommendation system. *Journal of Ambient Intelligence and Humanized Computing*, *10*(8), 3023–3034. https://doi.org/10.1007/s12652-018-0928-7

Kanakia, A., Eide, D., Shen, Z., & Wang, K. (2019). A scalable hybrid research paper recommender system for Microsoft academic. *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, *7*, 2893–2899. https://doi.org/10.1145/3308558.3313700

Karahodža, B., Donko, D., & Šupić, H. (2017). Modeling Long-Term User Profile in Collaborative Filtering. *International Journal on Artificial Intelligence Tools*, *26*(6), 1–26. https://doi.org/10.1142/S021821301750021X

Li, X., & Li, D. (2019). An Improved Collaborative Filtering Recommendation Algorithm and Recommendation Strategy. *Mobile Information Systems*, *2019*, 431–435. https://doi.org/10.1155/2019/3560968

Li, Z. long, Huang, M., & Zhang, Y. (2018). A collaborative filtering algorithm of calculating similarity based on item rating and attributes. *Proceedings - 2017 14th Web Information Systems and Applications Conference, WISA 2017*, *2018-Janua*, 215–218. https://doi.org/10.1109/WISA.2017.35

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. *Expert Systems with Applications*, *41*(4 PART 2), 2065–2073. https://doi.org/10.1016/j.eswa.2013.09.005

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

Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook* (pp. 73–105). https://doi.org/10.1007/978-0-387-85820-3_3

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems*, *74*, 12–32. https://doi.org/10.1016/j.dss.2015.03.008

Menon, S. P., & Hegde, N. P. (2015). A survey of tools and applications in big data. *Proceedings of 2015 IEEE 9th International Conference on Intelligent Systems and Control, ISCO 2015*, *7*. https://doi.org/10.1109/ISCO.2015.7282364

Nassar, N., Jafar, A., & Rahhal, Y. (2020). A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowledge-Based Systems, 187,* 104811. https://doi.org/10.1016/j.knosys.2019.06.019

Natarajan, S., Vairavasundaram, S., Natarajan, S., & Gandomi, A. H. (2020). Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data. *Expert Systems with Applications, 149*, 113248. https://doi.org/10.1016/j.eswa.2020.113248

Pal, A., Parhi, P., & Aggarwal, M. (2018). An improved content based collaborative filtering algorithm for movie recommendations. *2017 10th International Conference on Contemporary Computing, IC3 2017*, *2018-Janua*(August), 1–3. https://doi.org/10.1109/IC3.2017.8284357

Pazzani, M. J. (1999). A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review, 13*(5), 393–408. https://doi.org/10.1023/A:1006544522159

Pazzani, M. J., & Billsus, D. (2007). Content-Based Recommendation Systems. *In The Adaptive Web*, 325–341.

Pirasteh, P., Jung, J. J., & Hwang, D. (2014). Item-based collaborative filtering with attribute correlation: A case study on movie recommendation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8398 LNAI*(PART 2), 245–252. https://doi.org/10.1007/978-3-319-05458-2_26

Safoury, L., & Salah, A. (2013). Exploiting User Demographic Attributes for Solving Cold-Start Problem in Recommender System. *Lecture Notes on Software Engineering*, *1*(3), 303–307. https://doi.org/10.7763/LNSE.2013.V1.66

Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the Tenth International Conference on World Wide Web - WWW '01*, *10*, 285–295. https://doi.org/10.1145/371920.372071

Sayyed, F. R., Argiddi, R. V, & Apte, S. S. (2013). Collaborative Filtering Recommender System for Financial Market. *International Journal of Engineering and Advanced Technology (IJEAT)*, *2*(6), 389–391.

Schafer, J. Ben, Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, *5*(1–2), 115–153. https://doi.org/10.1007/978-1-4615-1627-9_6

Shambour, Q., Hourani, M., & Fraihat, S. (2016). An Item-based Multi-Criteria Collaborative Filtering Algorithm for Personalized Recommender Systems. *International Journal of Advanced Computer Science and Applications*, *7*(8), 15–17. https://doi.org/10.14569/ijacsa.2016.070837

Sharma, P., & Yadav, L. (2020). Book Recommendation System using Item based Collaborative Filtering. *International Journal of Innovative Research in Computer Science & Technology*, *8*(4), 5960–5965. https://doi.org/10.21276/ijircst.2020.8.4.2

Sharma, R., Gopalani, D., & Meena, Y. (2017). Collaborative filtering-based recommender system: Approaches and research challenges. *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, 1–6. https://doi.org/10.1109/CIACT.2017.7977363

Singh, V. K., Mukherjee, M., & Mehta, G. K. (2011). Combining collaborative filtering and sentiment classification for improved movie recommendations. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7080 LNAI*, 38–50. https://doi.org/10.1007/978-3-642-25725-4_4

Smith, B., & Linden, G. (2017). Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Computing*, *21*(3), 12–18. https://doi.org/10.1109/MIC.2017.72

Son, J., & Kim, S. B. (2017). Content-based filtering for recommendation systems using multiattribute networks. *Expert Systems with Applications*, *89*, 404–412. https://doi.org/10.1016/j.eswa.2017.08.008

Thakkar, P., Varma, K., Ukani, V., Mankad, S., & Tanwar, S. (2019). *Collaborative Filtering Using Machine Learning*. https://doi.org/10.1007/978-981-13-1747-7

Thorat, P. B., M. Goudar, R., & Barve, S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications*, *110*(4), 31–36.

https://doi.org/10.5120/19308-0760

Wang, C. D., Deng, Z. H., Lai, J. H., & Yu, P. S. (2019). Serendipitous recommendation in e-commerce using innovator-based collaborative filtering. *IEEE Transactions on Cybernetics*, *49*(7), 2678–2692. https://doi.org/10.1109/TCYB.2018.2841924

Wei, J., He, J., Chen, K., Zhou, Y., & Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, *69*, 29–39. https://doi.org/10.1016/j.eswa.2016.09.040

Xiao, Y., Ai, P., Hsu, C. H., Wang, H., & Jiao, X. (2015). Time-ordered collaborative filtering for news recommendation. *China Communications*, *12*(12), 53–62. https://doi.org/10.1109/CC.2015.7385528

Zhao, X. W., Guo, Y., He, Y., Jiang, H., Wu, Y., & Li, X. (2014). We Know What You Want to Buy: A Demographic-based System for Product Recommendation On Microblogs. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 7(2), 1935–1944. https://doi.org/10.1145/2623330.2623351

Zhu, Y., Lin, J., He, S., Wang, B., Guan, Z., Liu, H., & Cai, D. (2020). Addressing the Item Cold-Start Problem by Attribute-Driven Active Learning. IEEE Transactions on Knowledge and Data Engineering, 32(4), 631–644. https://doi.org/10.1109/TKDE.2019.2891530