被覆に基づくラフ集合を用いた推薦システムに関する研究

MURORAN INSTITUTE OF TECHNOLOGY

DOCTORAL THESIS

A study on recommender systems based on covering-based rough sets

Author:
Zhipeng ZHANG

Supervisor:
Dr. Yasuo KUDO

Submitted to Muroran Institute of Technology
Division of Engineering
In partial fulfillment of the requirements for the degree of

Doctor of Engineering

in

Advanced Information and Electronic Engineering

September 20, 2017

Declaration of Authorship

I, Zhipeng ZHANG, declare that this thesis titled, "A study on recommender systems based on covering-based rough sets" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Acknowledgements

First of all, I would like to express my sincere and heartfelt gratitude to Professor Yasuo Kudo, for being my supervisor in my doctoral period. The current study would not be accomplished without his support, untiring guidance, long hours of discussions. Whenever I have problems in my research, Mr. Kudo always patiently tutor and teach me how to analyze and solve problems. He also helps me revise the content over and over again in the process of thesis writing. I have learned a lot of skills in writing from his kind guidance. His lofty personality will alway inspire me to do positively in my future pursuits.

Likewise, my deepest thanks to the members of Inference System Laboratory, including past members for all their supports, guidances, and assistances in my research work. Good atmosphere of laboratory helped me promote my research smoothly. My sincere thanks to Satoshi Takahashi for teaching me Japanese language and common sense of life in my initial stage of life in Muroran. Whenever I have problems in my life, Takahashi always comes first and helps me. Especially, he patiently corrected many mistakes in my Japanese learning. I wish him a nice life and future.

Also, I would like to give my grateful appreciation to NANJO Auto Interior Co., Ltd for giving me the opportunity to have one month internship. I came into contact with many new technologies and approaches, and also understood the meaning of my area of research. My heartfelt gratitude to Okamoto, Yoshida, Kikuchi for helping me to be familiar with the company's business processes, and taking me to understand and appreciate the history and culture of Hiroshima city. I wish everything goes well with their work and the company will get better and better.

Finally, to my dear wife Yao Zhang and beloved parents, my profound appreciation for the continuous support, understanding, guidance, encouragement and love. Thank you for giving me much-needed confidence to face the future and my endeavors at my own feet.

Abstract

With the rapid development of Internet, the human race has entered the information society and the network era. Internet could provide people with more and more information and services; however, people have to face enormous data and useless information when they enjoy the convenience brought by Internet. Recommender system (RS) has emerged in response to this challenge, which can advise users when making decisions and help users discover items they might not find by themselves.

Collaborative filtering (CF) approach is popularly used in RSs owing to its satisfactory performance. Generally speaking, user-based collaborative filtering (UBCF) and item-based collaborative filtering (IBCF) are two significant approaches in CF, they have been successfully applied to many commercial RSs. However, with various kinds of data and complicated application environment, CF approaches are facing many challenges. For instance, UBCF cannot provide recommendations for an active user with satisfactory accuracy and diversity simultaneously. Personalized recommendations cannot be provided by UBCF for a new user which often has insufficient information. In addition, items that make a more significant contribution cannot have high weighting in IBCF. In view of the above key issues, this dissertation launched a study of the following aspects:

(1) Aiming to provide personalized recommendations for an active user, we apply covering-based rough sets to improve UBCF, and propose a new covering-based collaborative filtering (CBCF) approach. CBCF inserts a user reduction procedure into UBCF, covering reduction in covering-based rough sets is utilized to remove redundant users from all users. Then, k-nearest neighbors are selected from candidate neighbors comprised by the reduct-users. Our experiment results suggest that, for the sparse datasets that often occur in real RSs, CBCF outperforms than the UBCF, and can provide satisfactory accuracy and coverage for an active user at the same time.

(2) In order to provide personalized recommendations for a new user, through a detailed analysis of the characteristic of new users, we reconstruct a decision class to improve the previous CBCF. Unlike the previous CBCF, the decision class in improved CBCF can be extracted easily from the user-item rating matrix. Furthermore, the improved CBCF could provide personalized recommendations without needing special additional information. Our experiment results suggest that the improved CBCF significantly outperforms those of existing work and can provide personalized recommendations for a new user with satisfactory accuracy and diversity simultaneously.

(3) The traditional IBCF approach treats all items as the same weighting; however, because some items may have more important impact when computing the similarity and predictions, item-variance weighting should also be considered. In this paper, we present the time-based correlation degree and covering degree, and apply them to the traditional IBCF approach to rearrange the item weighting. Our experimental results suggest that, our proposed approach can produce recommendations superior to the traditional IBCF and other existing work.

Keywords: Recommender systems; Collaborative filtering; Covering-based rough sets; Covering reduction; Personalized recommendations; Item-variance weighting.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Rapid internet, economic, and technological developments have led to the dramatical growth of the data and information. Internet provides people with more and more information and services and it broke the limit of space and time of traditional life and learning. People can shop on the internet conveniently and study via internet whenever and wherever. However, people have to face enormous data and useless information when they enjoy the convenience brought by internet. This is the famous "information overload" problem (Hiltz and Turoff, 1985; Malone et al., 1987; Edmunds and Morris, 2000). Therefore, more customers are facing the problem of discovering the demanded contents from overwhelmingly massive data. As the result, this problem becomes a popular research topic and attracts attention from lots of scientists.

Generally, there are many stages for users to maintain information from internet. For instance, various portal sites are established, such as yahoo and so on. They help users filter and organize a variety of popular resource and information to discover and browse. However, the organized information is not always able to meet users' need, as well as overwhelming data will make the website overstaffed with the explosive growth of data, which results in the incompletion of information retrieval (Chang and Wang, 2011). In addition, search engines start to emerge so that users are able to retrieve their desired contents, such as google. But the accuracy of search results quite depends on the description towards questions, which is usually not quite precise, thus the caused bias will make it difficult for users to identify exactly their required results (Höchstötter and Lewandowski, 2009).

Recommender system (RS) analyzes the personal behavior of customers to learn their preferences and then recommends products (e.g., books, CDs, movies, and news) that may interest them (Adomavicius and Tuzhilin, 2005; Bobadilla et al., 2013). Currently, RSs are widely used applications in daily life. When users are lack of experience in the related field or can not deal with the huge amount of information, RSs could provide an intelligent information filtering for these users. For example, Amazon uses RSs to provide users with on-line shopping, because users' interests are usually different, so recommendation results need to be personalized, that is to say different users will receive different recommendations. In most of RSs, recommendation results are often presented in a sorted list, the sort number of item is determined by the target user's interest. In order to predict the target user's preference, RSs match the user's personal information (e.g., age, gender, education) and item's characteristics, or collect the user's historical information to make predictions. Generally, active users and new users are two types of users in RSs. An active user

has rated a lot of items, so RSs can utilize these sufficient information to provide reliable recommendations; however, a new user often has very few ratings, it will increase the recommendation difficulty. In recent years, their great commercial value and research potential have rendered RSs increasingly significant in recent years (Kotkov, Wang, and Veijalainen, 2016; Lu et al., 2015).

Currently, collaborative filtering (CF) is one of the most important approaches in RSs, that uses a customer's history as the basis of decision making (Symeonidis et al., 2008; Hameed, Al Jadaan, and Ramachandram, 2012). CF approach was first presented by Goldberg (Goldberg et al., 1992), which assumes that users who have similar preferences in the past will tend to have similar tastes in the future, or an item will be preferred by a user if this item is similar with the preference of this user in the past (Herlocker, Konstan, and Riedl, 2002), and many well-known online service providers are adopting this approach, including Amazon, YouTube, and Google News. CF approach does not need to analyze items' content information, which allows users to rate items according to their own experience and preference, then utilizes users' rating records to provide personalized recommendations for the target user (Herlocker et al., 1999). CF uses not only the target user's information, but also information of other users to filter out clutter and irrelevant information. Therefore, CF could provide recommendations with high quality, and help users find their potential interests. Nowadays, CF is proved to be one of the most successful personalized recommendation approaches, and has been widely utilized in the field of modern RSs.

To introduce CF approach, we first present some CF-related notations and terminologies used in this thesis. Given an RS, let U and I be finite sets of users and items, respectively, $R \cup \{\star\}$ the set of possible item rating scores, and RM the user-item rating matrix, AM the item attribute matrix. Absence of a rating is indicated by an asterisk ($\star$). The rating score of user u for item i is denoted by $r_{u,i} \in R \cup \{\star\}$, and the average of the valid ratings of user u is denoted by $\bar{ru}_u$, the average ratings of the i-th item is denoted by $\bar{ri}_u$. θ is set as the threshold for rating scores, and items with $r_{u,i} \geq \theta$ are defined as items that are relevant to user u. $I_u = \{i \in I | r_{u,i} \neq \star\}$ is the set of all items rated by user u, and I_u^c is the complementary set of I_u, indicating items that have not yet been rated by user u.

1.2 Research objectives

CF has many forms, user-based collaborative filtering (UBCF) and item-based collaborative filtering (IBCF) are two significant forms in CF. In this section, we present the detailed information about UBCF and IBCF.

1.2.1 User-based collaborative filtering

UBCF approach was first proposed by Herlocker (Herlocker et al., 1999), which assumes that users who have similar preferences in the past will tend to have similar tastes in the future. UBCF can provide satisfactory recommendations utilizing only the user's historical ratings, without requiring any other special information, and it has demonstrated remarkable success in RSs. The traditional UBCF can be separated into four steps:

$Step$ 1: Similarity computation. A target user tu's set of candidate neighbors CN_{tu} includes all users. Based on historical rating information, compute the similarity between each user $u \in CN_{tu}$ and the target user tu. Here the Pearson correlation

coefficient approach (1.1) is popularly used as a similarity measure:

$$sim(tu,u)=\frac{\sum_{i\in I_{tu}\cap I_u}\left(r_{tu,i}-\bar{ru}_{tu}\right)\left(r_{u,i}-\bar{ru}_u\right)}{\sqrt{\sum_{i\in I_{tu}\cap I_u}\left(r_{tu,i}-\bar{ru}_{tu}\right)^2}\sqrt{\sum_{i\in I_{tu}\cap I_u}\left(r_{u,i}-\bar{ru}_u\right)^2}}, \tag{1.1}$$

where $sim(tu,u)$ indicates the similarity between the target user tu and a user $u\in CN_{tu}$. $I_{tu}=\{i\in I|r_{tu,i}\neq\star\}$ is the set of all items rated by target user tu, and $\bar{ru}_{tu}$ is the average rating of target user tu:

$$\bar{ru}_{tu}=\frac{\sum_{i\in I_{tu}}r_{tu,i}}{|I_{tu}|}, \tag{1.2}$$

where $|I_{tu}|$ is the cardinality of the set I_{tu}.

Step 2: Neighborhood selection. Select the k most similar (nearest) users from CN_{tu} to comprise the neighborhood $N_{tu}(k)$ for the target user tu.

Step 3: Rating prediction. Normalize ratings and according to the rating information of neighborhoods, predict a rating score $p_{tu,i}$ for each item i in unrated item set I_{tu}^c of the target user tu. The weighted sum approach (1.3) is often used in RSs to predict ratings:

$$p_{tu,i}=\lambda\sum_{u\in N_{tu}(k)\cap U_i}sim(tu,u)*r_{u,i}, \tag{1.3}$$

where $p_{tu,i}$ is the prediction of item i for target user tu, $U_i=\{u\in U|r_{u,i}\neq\star\}$ is the set of users who have rated item i, and multiplier λ is a normalizing factor selected as

$$\lambda=\frac{1}{\sum_{u\in N_{tu}(k)\cap U_i}sim(tu,u)}. \tag{1.4}$$

Step 4: Item recommendation. According to the predicted rating scores, select the top N items that have the highest $p_{tu,i}$ from the candidate items as the recommendations to be provided for target user tu.

Note that, within specific systems, these steps may overlap or the order may be slightly different. Algorithm 1.1 summarizes the traditional UBCF approach.

Algorithm 1.1 Traditional UBCF approach

Input: User-item rating matrix RM and a target user tu.
Output: Recommended items set of size N for the target user tu.
k : Number of users in the neighborhood $N_{tu}(k)$ of the target user tu.
N : Number of items recommended to the target user tu.
I_{tu}^c : Items that have not yet been rated by the target user tu.
CN_{tu} : Candidate neighbors of the target user tu.
$p_{tu,i}$: Rating prediction of item i for the target user tu.
1: $CN_{tu}=U$, then compute the similarity between target user tu and each user $u\in CN_{tu}$;
2: **for** each item $i\in I_{tu}^c$ **do**
3: Find the k most similar users in CN_{tu} to comprise neighborhood $N_{tu}(k)$;
4: Predict rating score $p_{tu,i}$ for item i by neighborhood $N_{tu}(k)$;
5: **end for**
6: Recommend to the target user tu the top N items having the highest $p_{tu,i}$.

1.2.2 Item-based collaborative filtering

IBCF approach was first proposed by Sarwar (Sarwar et al., 2001) and is now a significant approach widely used in RSs, which assumes that items which are similar with a user's preferred item will also be preferred by this user. IBCF has good scalability and can be applied to the huge numbers of items and users that are typical of modern RSs. IBCF can easily handle large data sets and produce better predictions than UBCF. Also in contrast to UBCF, IBCF is able to compute item-item similarity off-line, both saving on-line time and making more effective recommendations. The detailed procedures of IBCF are as follows:

$Step$ 1: Item-item similarity computation. Based on the information of user-item rating matrix RM, IBCF computes the similarity between every item. Several algorithms can be used to make the similarity computation, but the Pearson correlation coefficient is one of the most widely used measure in IBCF:

$$\begin{aligned} & sim(x,y) \\ & = \frac{\sum_{u \in U_x \cap U_y} (r_{u,x} - \bar{ri}_x) * (r_{u,y} - \bar{ri}_y)}{\sqrt{\sum_{u \in U_x \cap U_y} (r_{u,x} - \bar{ri}_x)^2} \sqrt{\sum_{u \in U_x \cap U_y} (r_{u,y} - \bar{ri}_y)^2}}. \end{aligned} \tag{1.5}$$

Here, $U_x = \{u \in U | r_{u,x} \neq \star\}$ is the set of all users who have rated item x and $\bar{ri}_x$ is the average rating of the x-th item.

$Step$ 2: Neighborhood selection. After computing the item-item similarity, select the k most similar (nearest) items from similarity list to comprise the neighborhood $N_i(k)$ for the item $i \in I$.

$Step$ 3: Rating prediction. Normalize ratings and according to the rating information of $N_i(k)$ from the target user tu, predict a rating score $p_{tu,i}$ for item $i \in I$. The weighted sum is a very useful measure used in the IBCF:

$$p_{tu,i} = \frac{\sum_{j \in S_i \cap I_{tu}} sim(i,j) * r_{tu,j}}{\sum_{j \in S_i \cap I_{tu}} |sim(i,j)|}, \tag{1.6}$$

here $I_{tu} = \{x \in I | r_{tu,x} \neq \star\}$ is the set of items that target user tu has rated, S_i is the set of items that are similar to item i, and $p_{tu,i}$ is the prediction made about item i for target user tu.

$Step$ 4: Item recommendations. According to the predicted rating scores, select top N items which have the highest predicted rating score $p_{tu,i}$ from the candidate items as the recommendations to be provided for target user tu.

Note that, within specific systems, these steps may overlap or the order may be slightly different. Algorithm 1.2 presents concise steps of the traditional IBCF approach.

1.3 Research problems

In this thesis, we make research about the following three main problems about CF:

1. For an active user who often has sufficient rating information, the traditional UBCF cannot provide personalized recommendations with good accuracy and diversity simultaneously;

2. For a new user who often has fewer ratings, the traditional UBCF cannot provide personalized recommendations with good diversity while maintaining adequate accuracy;

Algorithm 1.2 Traditional IBCF approach

Input: User-Item rating matrix RM and an target user tu.
Output: Recommended items set of size N for the target user tu.
$N_i(k)$: Neighborhood of the item i.
k : Number of items in the neighborhood $N_i(k)$ of the item i.
N : Number of items recommended to the target user tu.
I_{tu}^c : Items which have not yet rated by the target user tu.
$p_{tu,i}$: Rating prediction of item i for the target user tu.
1: Compute similarity between each item in I;
2: **for** each item $i \in I_{tu}^c$ **do**
3: Find the k most similar items of item i to comprise neighborhood $N_i(k)$;
4: Predict rating score $p_{tu,i}$ for item i by the ratings of neighborhood $N_i(k)$ from the target user tu;
5: **end for**
6: Recommend to the target user tu the top N items having the highest predicted rating scores $p_{tu,i}$.

3. For items which has different contributions in computing similarity and predictions, the traditional IBCF approach cannot make item-variance weighting with respect to their contributions.

For the first and second problems, that is because some studies have shown that the traditional UBCF approach cannot provide satisfactory recommendations with good accuracy and diversity values simultaneously. In addition, resent research has concluded that gains diversity in RS can frequently be accompanied by losses in accuracy, making it difficult to select a reasonable trade-off between accuracy and diversity. Moreover, a new user in RSs usually has fewer ratings, information obtained from a new user is insufficient, it increases the difficulty of recommendations.

For the third problem, that is because in traditional IBCF approach, all items carry the same weight when computing the similarity and predictions. However, it is widely recognized that some items are more important than others and should be given relatively higher weighting.

1.4 Research contributions

In this thesis, there are the following three contributions with respect to three problems listed in Section 1.3:

1. In order to provide personalized recommendations for an active user, we apply covering-based rough sets to the traditional UBCF, and propose a new covering-based collaborative filtering (CBCF) approach. CBCF inserts a user reduction procedure into the traditional UBCF. Covering reduction in covering-based rough sets is used to remove redundant users from all users. Then, k-nearest neighbors are selected from candidate neighbors comprised by the reduct-users. CBCF can select more efficient neighborhood for an active user, and provide recommendations with satisfactory accuracy and diversity simultaneously.

2. In order to provide personalized recommendations for a new user, through a detailed analysis of the characteristic of new users, we reconstruct a decision

class to improve the previous CBCF. In this way, improved CBCF can remove redundant candidate neighbors for a new user as many as possible. Furthermore, unlike the previous CBCF, improved CBCF could provide personalized recommendations without needing special additional information.

3. In order to make item-variance weighting for the traditional IBCF, we present time-related correlation degree and covering degree, and apply them to the traditional IBCF to rearrange the item weight. The proposed novel approach can produces recommendations results superior to those of existing work.

1.5 Outline of the thesis

This thesis is organized as follows:

Chapter 1 presents the introduction, including the research background, research objectives, research problems and research contributions.

Chapter 2 presents the basic knowledge of covering-based rough sets which we used as our research method in this thesis, that covers the definitions of covering and covering approximations space, and six types of covering-based rough sets with respect to the different upper and lower approximation operations. In addition, as a significant concept of covering-based rough sets, covering reduction is also discussed in this chapter.

Chapter 3 describes a CBCF approach we have proposed to provide personalized recommendations for an active user. First, we analyze the deficiency of traditional UBCF and give the problem setting. Then, we apply covering-based rough sets to traditional UBCF, and propose a new CBCF approach. The motivation, detailed procedures and an example of CBCF approach are also presented. Finally, experiments using popular datasets are made to prove that CBCF outperforms than the traditional UBCF, and can provide satisfactory accuracy and diversity simultaneously.

Chapter 4 introduces an improved CBCF approach which could provide personalized recommendations for a new user. First, we summarize the related work about personalized recommendations for a new user. Then, through two popular datasets, we analyze the characteristic of a new user in RSs and present the problem setting. Next, we introduce our improve CBCF, including motivation, reconstruction of decision class and detailed procedures. Moreover, comparisons between the previous CBCF and improved CBCF are also made. Finally, experiments are made to demonstrate that the improved CBCF significantly outperforms those of existing work, and can provide personalized recommendations for a new user with satisfactory accuracy and diversity simultaneously.

Chapter 5 explains a TCIBCF approach we proposed to improve the traditional IBCF by using time factor and covering degree. First, we analyze basic procedures of traditional IBCF and present the problem setting. Then, we present time-related correlation degree and covering degree, and apply them to the traditional IBCF to propose a TCIBCF approach, the motivation and detailed procedures are also discussed. In the end, we design the experiments, and utilize some popular evaluation metrics to indicate that TCIBCF approach produces recommendation results superior to those of existing work.

Finally, the conclusions and directions for future research are presented in Chapter 6.

Chapter 2

Covering-based rough sets

2.1 Introduction

Rough set theory was first presented by Pawlak in the early 1980s (Pawlak, 1982), it is a very useful tool of data analysis for processing vague and uncertain data (Pawlak and Skowron, 2007). Rough set has attracted many researcher's interests all over the world. The core concepts of the rough sets are lower and upper approximations based on equivalence relations, and the knowledge which hidden in the information system can be expressed in the form of decision rules by means of these two concepts. However, the classical rough set based on equivalence relations is not suitable to be applied to complex information system. Hence people proposed different extensions of rough set theory. Covering-based rough sets extend the partition in rough sets to the covering (Yao and Yao, 2012), and use the covering of the domain to construct the lower and upper approximations. In this way, rough sets are enriched from both theory and application in terms of more complicated data.

This chapter presents an overview of the basic knowledge of covering-based rough sets, which covers the definitions of covering and covering approximations space. In addition, we focus on covering reduction which is a significant concept in CBCF, and analyze main types of reduction algorithms.

2.2 Basic definitions and concepts

Pawlak's rough sets are based on equivalence relations. Covering-based rough sets extend a partition to a covering. In this section, we introduce some basic concepts about covering-based rough sets. More information of covering-based rough sets can be found in (Tsang, Degang, and Yeung, 2008; Yao and Yao, 2012; Zhu, 2009). First we list some definitions about coverings used in this thesis.

Definition 2.1. Let T be the domain of discourse, and C a family of subsets of T. If none of the subsets in C is empty and $\cup C = T$, then C is called a covering of T.

Definition 2.2. Let T be a non-empty set, and C a covering of T. We refer to the ordered pair $\langle T, C\rangle$ as the covering approximation space.

Definition 2.3. Let $\langle T, C\rangle$ be a covering approximation space, $x \in T$.

$$Md(x) = \{K \in C | x \in K \wedge (\forall S \in C \wedge x \in S \wedge S \subseteq K \Rightarrow K = S)\}$$

is called the minimal description of x.

Definition 2.4. Let C be a covering of T, $N(x) = \cap\{K \in C | x \in K\}$ is called the neighborhood of x.

Different lower and upper approximation operations would generate different types of covering-based rough sets. The covering-based rough sets was first presented by Zakowski (Zakowski, 1983), who extended Pawlak's rough set theory from a partition to a covering. Pomykala gave the notion of the second type of covering-based rough sets (Pomykala, 1987), while Tsang presented the third type (Tsang et al., 2004), Zhu defined the fourth and fifth types of covering-based rough sets (Zhu and Wang, 2012; Zhu, 2007), and Wang studied the sixth type of covering-based approximations (Wang, Dai, and Zhou, 2004).

Definition 2.5. Let C be a covering of T, $P(T)$ is the power set of T. The operations $CL : P(T) \to P(T)$ and $CL^* : P(T) \to P(T)$ are defined as follows: $\exists X \in P(T)$

$CL(X) = \cup\{K \in C | K \subseteq X\} = \cup\{K | \exists x, s.t.(K \in Md(x)) \wedge (K \subseteq X)\}$

$CL^*(X) = \{x | N(x) \wedge X\} = \cup\{N(x) | N(x) \wedge X\}$

Here $CL(X)$ is the first, the second, the third, the fourth and the fifth covering lower approximation operations and $CL^*(X)$ is the sixth covering lower approximation operations with respect to the covering C. The operations $FH, SH, TH, RH, IH, XH : P(T) \to P(T)$ are defined as follows: $\exists X \in P(T)$

$FH(X) = CL(X) \cup (\cup\{Md(x) | x \in X - CL(X)\})$

$SH(X) = \cup\{K | K \in C, K \cap X \neq \emptyset\}$

$TH(X) = \cup\{Md(x) | x \in X\}$

$RH(X) = CL(X) \cup (\cup\{K | K \cap (X - CL(X)) \neq \emptyset\})$

$IH(X) = CL(X) \cup (\cup\{N(x) | x \in X - CL(X)\}) = \cup\{N(x) | x \in X\}$

$XH(X) = \{x | N(x) \cap X \neq \emptyset\}$

FH, SH, TH, RH, IH, XH are called the first, the second, the third, the fourth, the fifth and the sixth covering upper approximation operations with respect to C, respectively.

2.3 Covering reduction algorithms

Covering reduction is a significant concept in covering-based rough set theory (Yang and Li, 2010). The concept of covering reduction was originally presented by Zhu et al. (Zhu and Wang, 2003). In this thesis, we refer to the algorithm proposed by Zhu et al. (Zhu and Wang, 2007) as the first type of reduction algorithm, which corresponds to the definition of $reduct(C)$ in (Zhu and Wang, 2007). Definition 2.6 defines this algorithm.

Definition 2.6. Let C be a covering of domain T, and $K \in C$. If K is a union of some sets in $C - \{K\}$, K is reducible in C; otherwise, K is irreducible. When all reducible elements are removed from C, the new irreducible covering is called the first-type reduct of C.

Zhu et al. presented two other covering reduction algorithms (Zhu and Wang, 2007; Zhu and Wang, 2012), which we refer to as the second and third types of reduction algorithms, respectively. Definition 2.7 defines the second-type algorithm, which corresponds to the definition of $exclusion(C)$ provided by Zhu et al. (Zhu and Wang, 2007). Definition 2.8 defines the third-type algorithm, which corresponds to the definition of $exact - reduct(C)$ (Zhu and Wang, 2012).

Definition 2.7. Let C be a covering of domain T, and $K \in C$. If there exists another element K' of C such that $K \subset K'$, K is an immured element of covering C. When we remove all immured elements from C, the set of all remaining elements is still a covering of T, and this new covering has no immured element. We refer to this new covering as the second-type reduct of C.

Definition 2.8. Let C be a covering of domain T, and $K \in C$. If there exists $K_1, K_2...K_m \in C - K$ such that $K = K_1 \cup, ..., \cup K_m$, and $\forall x \in K$ and $\{x\}$ is not a singleton element of C, $K \subseteq \cup\{K' \mid x \in K' \in C - \{K\}\}$, K is called an exact-reducible element of C. When all exact-reducible elements are removed from C, the new irreducible covering is called the third-type reduct of C.

Comparing the three types of covering reduction algorithms, we find that, the first type removes redundant elements more efficiently than the third type because the third type has an additional restriction condition. For example, we assume that $K \in C$ is a reducible element in the first type, but if there exists $x \in K$ that $\{x\}$ is a singleton element of C, K is not an exact-reducible element in the third type. However, if $K \in C$ is an exact-reducible element in the third type, it must be a reducible element in the first type.

Here, we consider the first and second types. If we assume that $K \in C$ is a reducible element in the first-type algorithm, then there must be other elements whose union is K. For example, for $K = K_1 \cup K_2$, only K should be removed; however, under the same conditions, in the second-type algorithm, K_1 and K_2 would both be considered as immured elements, which should be removed.

Typically, an RS has a vast number of items and each user has different preferences. Therefore it is difficult to represent one user's preferred item set as a union of other users' preferred item sets accurately. In this situation, for the first-type algorithm, few reducible elements can be removed; however, for the second type, there can be a large number of reducible elements, because RSs have a large number of users, it is easy to find one user's preferred item set that includes another user's set. Thus, the second type of covering reduction algorithm can be used to remove more reducible elements in RSs. The second-type of covering reduction algorithm (STCRA) is given in Algorithm 2.1.

Algorithm 2.1 STCRA: The second-type of covering reduction algorithm

Input: A covering of a domain: C.
Output: An irreducible covering of a domain: $reduct(C)$.
K_i,K_j: Elements in the covering C.

```
1: set reduct(C)=C;
2: for i = 1 to card(C) do
3:    for j = 1 to card(C) do
4:       if K_j ⊂ K_i then
5:          if K_j ∈ reduct(C) then
6:             reduct(C) = reduct(C) − {K_j};
7:          end if
8:       end if
9:    end for
10: end for
11: return reduct(C);
```

2.4 Summary

In this chapter, we present the basic definitions and notions of covering-based rough sets. Including covering and covering approximation, and different types of covering-based rough sets according to different upper and lower approximation operations.

Besides that, covering reduction which could remove redundant elements is also demonstrated. We discuss different types of reduction algorithm, and analyze their efficiency of removing redundant users, and get the conclusion that, the second-type of covering reduction algorithm we defined in this chapter can be used to remove more reducible elements.

Chapter 3

CBCF for active users' personalized recommendations

3.1 Introduction

Currently, although most studies focus on developing new approaches to improve RS accuracy, it has been argued that using only an accuracy metric to evaluate RSs is not sufficient and that the diversity of recommendations must also be considered as an important evaluation measure (Kunaver and Požrl, 2017; Clarke et al., 2008; Hu and Pu, 2011). Because in a real business environment, a company can use RSs to obtain more benefits by providing recommendations with higher diversity (Vargas, 2011; Boim, Milo, and Novgorodov, 2011; Di Noia et al., 2014). For example, as there are many movies in the statistical long tail that have only a few ratings, it would be profitable for Netflix if RSs would encourage users to rent movies in the long tail, because these are less costly to license and acquire from distributors than new releases or highly popular movies. However, recent studies have shown that it is very difficult to obtain a reasonable trade-off between the accuracy and diversity of an RS (Liu, Shi, and Guo, 2012; Zhou et al., 2010), because increasing the diversity of recommendations is usually accompanied by a loss in accuracy (Javari and Jalili, 2015).

CF is a significant component of the recommendation process (Hameed, Al Jadaan, and Ramachandram, 2012). UBCF is one of the most useful approaches in CF. Without requiring any other special information, UBCF can only utilize user's historical ratings to provide satisfactory recommendations. However, the traditional UBCF is difficult to achieve good values for accuracy and diversity simultaneously (Herlocker et al., 1999; Herlocker, Konstan, and Riedl, 2002; Zhu et al., 2014). Aiming at improving the traditional UBCF approach to obtain good values of accuracy and coverage at the same time, in this chapter, covering-based rough set theory is applied to RSs. We propose CBCF, a new approach that uses covering reduction to remove redundant users, then neighborhood is selected from candidate neighbors comprised by the reduct-users. Experimental results reveal that our proposed CBCF approach provides better recommendation results than the traditional UBCF approach.

The remainder of this chapter is organized as follows. In Section 3.2, some related works are provided. We review basic concepts involved in the traditional UBCF approach and other work. In Section 3.3, we analyze neighborhood selection problems. In Section 3.4, we give the detailed motivation and procedures of the CBCF approach. In Section 3.5, we describe our experiments and compare CBCF results with the results obtained using the traditional UBCF approach. The summary of this chapter is presented in Section 3.6.

3.2 Related works

UBCF approach was first proposed by Herlocker (Herlocker et al., 1999), which relies on target user neighborhood information to make predictions and recommendations. Neighborhood selection is one crucial procedure of UBCF approach, which selects a set of users from candidate neighbors to comprise neighborhood for an active user. Whether appropriate neighborhood can be selected will have a direct bearing on the rating prediction and item recommendation. In general UBCF approach, k-nearest neighbors (k-NN) approach is proved to be the best method to generate a neighborhood, which picks the k most similar (nearest) users from candidate neighbors to comprise the neighborhood for an active user (Herlocker, Konstan, and Riedl, 2002). So we consider the k-NN UBCF approach as the traditional UBCF approach in the rest of this chapter. More detailed information and procedures of UBCF could be found in Subsection 1.2.1.

Currently, commercial RSs have a large number of users, neighborhood must be composed of a subset of users rather than all users if RSs want to guarantee acceptable response time (Herlocker, Konstan, and Riedl, 2002). Accuracy measures how closed RSs predictions reflect actual user preferences, and coverage interprets the extent to which recommendations cover the set of available items. Both metrics are important in RSs. In the neighborhood of traditional UBCF approach, neighbors tend to have similar tastes, so high predicted scores from them concentrate in few types of items, even just popular items. Due to the popular items often have high ratings from users, so recommendations from the traditional UBCF approach often have high accuracy. However, types of recommendations are very limited, it leads to an unsatisfactory coverage value (Gan and Jiang, 2013). Therefore using the traditional UBCF is difficult to achieve good values for both metrics simultaneously.

Niemann and Wolpers proposed a usage context-based collaborative filtering approach to achieve a reasonable balance between the accuracy and diversity (Niemann and Wolpers, 2013). In the procedure of similarity computation, they described an item by the items it is significantly often used with rather than by its users or content attributes, items were similar if they often occurred in similar usage contexts. This approach could compensate the weaknesses of traditional UBCF approach, and increased the diversity with only one case of accuracy loss. However, this approach needed additional information (e.g. user profiles) which was often not available or incomplete.

Gan and Jiang proposed a network-based collaborative filtering approach to improve the diversity without lowing the accuracy of recommendations (Gan and Jiang, 2013). Before making the rating prediction, they constructed a user similarity network from user's historical data by using a nearest neighbor approach. Due to this network could filter out weak relationships between users, this approach not only enhanced the recommendation accuracy but also improved the diversity; however, the performance of this approach depending on the selected parameter, and the optimal value of parameters was still unknown. Besides, it was difficult to select a suitable mathematical model for the user similarity network.

Adomavicius and Kwon developed a sophisticated graph-theoretic approach to maximize the diversity of recommendations based on maximum flow or maximum bipartite matching computations (Adomavicius and Kwon, 2011). After selecting the neighborhood, this approach predicted rating scores for un-rated items with traditional filtering approach to generate candidate items. Then, let users and items be represented as nodes, and translated the top-N candidate items setting into a graph-theoretic framework to re-rank candidate items. Good values of accuracy and

diversity from this approach depends on the selection criteria for items included in the graph. However, the more items are selected as candidate items for each user, the more diverse and less accurate are the recommendations and vice versa.

Adomavicius and Kwon improved the recommendation diversity by re-ranking the candidate items using a new re-rank technology (Adomavicius and Kwon, 2012). This approach could only utilize traditional UBCF to compute the similarity and provide candidate items rather than some specific algorithm. Then they utilized additional features, such as item absolute likability, item relative likability, item rating variance, and neighbors' rating node to re-rank the candidate items. Finally selected top N items for recommendations. This approach could provide recommendations with good diversity, but it comes at the expense of accuracy.

3.3 Analysis and problem setting

Neighborhood selection is to determine which users' rating information will be utilized to compute the prediction for an active user, in other words, it decides who will be selected as neighborhood of the active user. In theory, every user could be selected as a neighbor. However, modern commercial RSs have vast customers, e.g., Amazon has billions of users, it is impractical to consider every user as a neighbor when trying to maintain real-time performance. A subset of users must be selected as neighborhood if RSs want to guarantee acceptable response time. Herlocker et al. (Herlocker, Konstan, and Riedl, 2002) discussed the size of neighborhood in detail, and drew a conclusion that the size of neighborhood affects the performance of RSs in a reasonably consistent manner. It suggests that, in the real-world situations, a neighborhood of 20 to 60 neighbors is reasonable to be used to make predictions.

Currently, k-NN method is often used in the traditional UBCF approach to make neighborhood selection, neighborhood is comprised by the top k users with highest similarity in candidate neighbors. However, in RSs, some items, especially the popular items, have high rating scores from most of users, and the active user usually also prefer these items. In this case, when using the traditional UBCF approach, users who prefer the popular items are likely to have high similarity with the active user, so they will easily appear in the neighborhood. Other users, who prefer niche items, are difficult to be selected as the neighborhood, but these niche items may also be preferred by the active user. For example, the relevant items of user 1 and user 2 are popular items, the relevant items of user 3 are niche items. Similarity between active user and them are 0.9, 0.8, and 0.7, respectively, besides that, user 2's relevant item set is included in user 1's relevant item set. In traditional UBCF approach, if we select two most similar users as neighborhood, user 1 and user 2 will be selected, in this case, only popular items will be recommended to the active user. However, user 3 also have high similarity with the active user, relevant items of user 3 may also be preferred by the active user. In order to obtain neighborhood with diverse tastes, we can remove user 2 and select user 1 and user 3 as the neighborhood. Because the relevant item set of user 2 is included in user 1's relevant item set, so we can only utilize user 1 to make predictions for popular items rather than both of them. Here, we consider users like user 2, whose relevant item set is included in other user's relevant item set, as the redundant users. In traditional UBCF approach, the k-nearest neighbors have similar taste, so they tend to have similar relevant items, therefore neighborhood usually contains many redundant users. When making prediction, they tend to give high predicted scores for few types of items, even just the popular

items. It causes the traditional UBCF approach cannot provide recommendations with good values of accuracy and diversity simultaneously.

3.4 CBCF for an active user's personalized recommendations

3.4.1 Motivation of CBCF approach

The proposed CBCF approach aims to improve the traditional UBCF approach by reducing redundant users, and constructs neighborhood by users who have high similarity and diverse relevant items. As we discussed above, redundant user's relevant item set is included in other user's relevant item set. According to discussions in section 2.3, reducible element in the second type of covering reduction algorithm is also included in other elements, so we can remove redundant users by using the second type of covering reduction algorithm. Removing all reducible elements means we remove all redundant users.

In general RSs, there are vast items, it means the item domain I is too large. However, different users have rated different items, it may cause not so many users could be considered as redundant users. In order to remove redundant users as many as possible, item domain should be reduced as much as possible. In CBCF approach, we reduce the domain from item set I to active user's decision class D. Items fit the active user's relevant attributes comprise the decision class D. However, in practical application, users usually do not enter their relevant attributes into RSs. Here, in order to obtain relevant attributes of an active user, we sum each attribute value in the active user's relevant item set. Due to the more high rating scores indicate that the more the active user likes the attribute, l number attributes with the largest sums are selected as the relevant attributes.

Relevant attributes of the active user in the following form:

$$[at_1 = av_1] \wedge [at_2 = av_2] \wedge ... \wedge [at_m = av_m],$$

where m means the number of all attributes, at_m is an attribute and av_m is the value of at_m.

3.4.2 Procedures of CBCF approach

In CBCF approach, we insert user reduction step into the traditional UBCF approach. Algorithm 3.1 presents concise steps of the CBCF approach. The detailed procedure is as follows:

$Step$ 1: User reduction. First set I as the domain, relevant items of each user comprise a set in domain I. We construct decision class D for the active user au. The decision class D consists of all items that fit the active user's relevant attributes, defined by (3.1).

$$D = \{i \in I | at_1(i) = av_1, at_2(i) = av_2, ..., at_m(i) = av_m\}, \tag{3.1}$$

where $at_m(i) = av_m$ means that the value of the attribute at_m on item i is av_m.

Then to remove as many redundant users as possible, we reduce the domain from item set I to decision class D, and for each user $u \in U$, the relevant items of user u in domain D comprise the relevant set C_u, where

$$C_u = \{i \in D | r_{u,i} \geq \theta\}. \tag{3.2}$$

Let $C^* = D - \cup C_u$; then, $C = \{C_1, C_2...C_n, C^*\}$ is a covering for the active user in domain D.

Next, based on the second type of covering reduction algorithm in the covering-based rough sets, redundant elements are removed from covering C to obtain reduct(C), we can obtain the active user's reduct-users U^r, where

$$U^r = \{u \in U | C_u \in reduct(C)\}. \tag{3.3}$$

Step 2: Similarity computation. Users in U^r comprise candidate neighbors CN^r_{au} of the active user au. According to the rating information, compute the similarity $sim(au, u)$ between the active user au and each user $u \in CN^r_{au}$ by the similarity measure.

Step 3: Neighborhood selection. The active user au's neighborhood $N^r_{au}(k)$ is composed by k most similar (nearest) users in CN^r_{au}.

Step 4: Rating prediction. Based on rating information of neighborhood $N^r_{au}(k)$, we predict rating score $p_{au,i}$ for each item i in unrated item set I^c_{au} of the active user au.

Algorithm 3.1 CBCF approach

Input: User-item rating matrix RM, item attribute matrix AM, and an active user au.

Output: Recommended items set of size N for the active user au.

k : Number of users in the neighborhood $N^r_{au}(k)$ of the active user au.
N : Number of items recommended to the active user au.
D : Decision class of the active user au.
U^r : Users after making user reduction, reduct-users.
I^c_{au} : Items which have not yet rated by the active user au.
CN^r_{au} : Candidate neighbors of the active user au after making user reduction.
$p_{au,i}$: Rating prediction of item i for the active user au.

1: **for** each user $u \in U$ **do**
2: $C_u = \{i \in D | r_{u,i} \geq \theta\}$.
3: **end for**
4: Let $C^* = D - \cup C_u$; then, $C = \{C_1, C_2...C_n, C^*\}$ is a covering for an active user au in domain D.
5: $reduct(C) = STCRA(C)$
6: Reduct-user $U^r = \{u \in U | C_u \in reduct(C)\}$.
7: $CN^r_{au} = U^r$, compute similarity between the active user au and each user $u \in CN^r_{au}$
8: **for** each item $i \in I^c_{au}$ **do**
9: Find the k most similar users in CN^r_{au} to comprise neighborhood $N^r_{au}(k)$;
10: Predict rating score $p_{au,i}$ for item i by neighborhood $N^r_{au}(k)$;
11: **end for**
12: Recommend to the active user au the top N items having the highest $p_{au,i}$.

3.4.3 Example of CBCF approach in RSs

Here, we present an example to explain the CBCF approach more clearly. Table 3.1 illustrates an User-Item rating matrix RM about rating scores by six users for eight items, U_{au} represents the active user. The rating value is from 1 to 5, where a higher value indicates that the user likes the given item more. Table 3.2 shows the item

attribute matrix AM about eight items, and each item has the following attributes: Horror, Comedy, Drama, Action, and Musical, where a value of 1 indicates that the item is of that genre and a value of 0 indicates it is not. Note that items can be in several attributes simultaneously. The detailed steps are as follow:

TABLE 3.1: Example of user-item rating matrix RM

User	Co-rated items						Target items	
	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8
U_1	2	4	1	3	3	4	5	5
U_2	1	3	2	3	2	5	2	3
U_3	1	2	3	5	3	4	2	1
U_4	2	2	5	1	4	5	1	4
U_5	2	4	5	2	1	3	5	3
U_{au}	1	4	2	5	2	3	$\star$	$\star$

TABLE 3.2: Example of item attribute matrix AM

Item	Attribute				
	Horror	Comedy	Drama	Action	Musical
Item 1	1	0	1	0	0
Item 2	0	1	0	1	1
Item 3	1	1	0	1	1
Item 4	1	1	1	1	0
Item 5	0	1	1	1	0
Item 6	0	1	0	1	0
Item 7	0	1	1	0	0
Item 8	1	0	0	1	1

TABLE 3.3: Example of similarity and rank depending on different approaches

User-User	Traditional UBCF		Proposed CBCF	
	Similarity	Rank	Similarity	Rank
$U_{au} - U_1$	0.501	3	0.501	2
$U_{au} - U_2$	0.563	2	-	-
$U_{au} - U_3$	0.646	1	0.646	1
$U_{au} - U_4$	-0.458	5	-	-
$U_{au} - U_5$	0.075	4	0.075	3

Step 1: User reduction. Here, we treat the rating threshold θ as 3; thus, from the rating matrix RM we can obtain the active user's relevant items set {Item 2, Item 4, Item 6}. We sum each attribute value in the relevant item set according the item attribute matrix AM (Horror=1, Comedy=3, Drama=1, Action=3, Musical=1). Then, two attributes with the largest sums (Comedy and Action) are selected as relevant attributes of the active user. Then, all items that fit the relevant attributes comprise the decision class $D =$ {Item 2, Item 3, Item 4, Item 5, Item 6}.

Reduce the domain from all items set to decision class D. Relevant items of the user u in domain D will be a set C_u :
$C_1 =$ {Item 2, Item 4, Item 5, Item 6}, $C_2 =$ {Item 2, Item 4, Item 6},
$C_3 =$ {Item 3, Item 4, Item 5, Item 6}, $C_4 =$ {Item 3, Item 5, Item 6},

C_5 = {Item 2, Item 3, Item 6}.
Then, $C = \{C_1, C_2, C_3, C_4, C_5\}$ is a covering for the active user in domain D. Based on the definition of the second-type covering reduction algorithm, $C_2 \subset C_1, C_4 \subset C_3$; thus, C_2 and C_4 can be regarded as redundant elements to be removed. Then, we can obtain the reduct(C) = $\{C_1, C_3, C_5\}$, so the reduct-users $U^r = \{U_1, U_3, U_5\}$.

Step 2: Similarity computation. Candidate neighbors CN^r_{au} for the active user are composed by users in U^r, $CN^r_{au} = U^r = \{U_1, U_3, U_5\}$. Then utilize the Pearson correlation coefficient similarity measure to compute the similarity between the active user and each user in CN^r_{au}. Table 3.3 shows results of similarity and user rank for the traditional UBCF and proposed CBCF approaches.

Step 3: Neighborhood selection. If we consider only three nearest users in candidate neighbors as neighborhood of the active user, U_1, U_2, and U_3 will comprise the neighborhood $N_{au}(3)$ for the traditional UBCF; however, for our proposed CBCF approach, U_1, U_3, and U_5 will be considered as the neighborhood $N^r_{au}(3)$.

Step 4: Rating prediction. From the rating scores of $N^r_{au}(3)$, we use the adjusted weighted sum approach to predict the rating scores for item 7 and item 8. Here
$P_{au,7} = 3.284$, $P_{au,8} = 2.588$
Because $P_{au,7} > P_{au,8}$, if we select the top one movie as recommendation, item 7 will be recommended to the active user.

3.4.4 Discussion

To provide recommendations with good values of accuracy and diversity for an active user au, the biggest innovation of the proposed CBCF approach is that, we insert the user reduction procedure into the traditional UBCF approach. For an active user au, before computing the similarity, we remove redundant users from all users to obtain reduct-users U^r and which comprise candidate neighbors CN^r_{au} with diverse tastes, k most similar (nearest) users selected from CN^r_{au} comprise neighborhood $N^r_{au}(k)$. Although comparing with input conditions of the traditional UBCF, our proposed CBCF needs an additional condition: item attribute matrix AM; however, in general RSs, item attribute matrix is very common and easy to obtain.

User reduction is a core component of CBCF approach, which applies the notion of covering reduction to reduct redundant users from all users. First, we set all items I as the domain, and relevant items of each user comprise a set in domain I. However, in this case, there are only a few sets can be removed as redundant elements. To remove as many redundant users as possible, when obtaining the decision class D, we reduce the domain from I to D such that the domain can be sufficiently small. Then, the relevant items of each user in decision class D will be a element of a covering C. Based on the definition of the second type of covering reduction algorithm, for set C_1, if there exists another set C_2 for which $C_1 \subset C_2$, C_1 is considered reducible and therefore removable. In this approach, C_1 denotes the relevant items of user 1 in domain D and $C_1 \subset C_2$ indicates that user 1 and user 2 are likely to prefer same type of items, so we can just utilize user 2 to make prediction for this type of items, thus user 1 can be considered as redundant user to be removed. Removing all reducible elements means that all redundant users are removed from all users, so that this approach can only use the reduct-users U^r to comprise CN^r_{au}. Users in CN^r_{au} have diverse relevant of items, and high similarity users are selected from CN^r_{au} to comprise neighborhood $N^r_{au}(k)$. So in proposed CBCF approach, neighbors in the $N^r_{au}(k)$ have both high similarity and diverse preference, they can make accurate predictions for more types of items and present recommendations with high accuracy and diversity at the same time.

TABLE 3.4: Average size of decision class versus l with the MovieLens dataset

l(number of relevant attributes)	1	2	3	4
Average size (decision class)	583.257	70.363	7.068	0.303

3.5 Experiments and evaluations

In this section, we introduce the evaluation dataset and metrics, examine the effects of the approach components, and compare the CBCF approach's performance with the traditional UBCF approach with different datasets.

3.5.1 Experimental setup and evaluation metrics

In our experiments, we utilized the MovieLens (Herlocker et al., 1999) and Jester (Goldberg et al., 2001) datasets because they are often used to evaluate RSs. The MovieLens 100K dataset consists of 1,682 movies, 943 users, and 100,000 ratings on a scale of 1 to 5. Each user has rated at least 20 movies, and in our study, movies rated above 3 were treated as a user's relevant movies. The Jester 3.9M dataset contains ratings of 100 jokes from 24,983 users. Each user has rated 36 or more jokes. The value range of rating scores is -10 to 10. A value of "99" represents an absent rating. In our experiment, jokes rated above 5 were treated as a user's relevant jokes.

We also used the conventional leave-one-out procedure to evaluate the performance of the proposed approach. For each test user, we only considered items that the user had rated as test items. First, we supposed that the test items had no rating scores from the test user. Then, our approach predicted a rating score for each test item using the information obtained from the remaining users. Finally, comparisons were made between the original and predicted rating scores. For the MovieLens dataset, we summed each attribute value in the test user's set of relevant movies, and l number attributes with the largest sums were selected as the relevant attributes of the test user. As there were 18 attributes for each movie, we computed the average size of decision class in terms of different number of l. Table 3.4 shows the result. If the size of test user's decision class is too big, there will be just fewer redundant users could be removed; however, if the size of test user's decision class is too small, other users' relevant item set will include this decision class easily, in this case, it will lose the meaning of reduction. Overall consideration, we select two attributes to construct the decision class.

For the Jester dataset, no information was presented about item attributes. There were 100 jokes in this dataset, we considered the top 50 jokes sorted by the test user's rating scores as the decision class. If the number of rated jokes from the test user was less than 50, we treated all rated jokes as the decision class. However if the neighbor's set of relevant jokes was too large, it would include the decision class, in this case, covering reduction will lose effectiveness. To avoid this, we selected the top 10% users who had rated the fewest jokes from all users, and utilized these 2,498 users for our experiment.

To measure the performance of the proposed approach, we used the mean absolute error (MAE), root mean square error (RMSE) to represent the accuracy, and coverage to evaluate the diversity of recommendations, all of which are popular metrics for evaluating RSs.

The MAE and RMSE metrics demonstrate the average error between predictions and real values; therefore, the lower these values, the better the accuracy of RSs.

$$MAE = \frac{1}{card(U)} \sum_{u \in U} (\frac{1}{card(O_u)} \sum_{i \in O_u} |p_{u,i} - r_{u,i}|), \tag{3.4}$$

$$RMSE = \frac{1}{card(U)} \sum_{u \in U} \sqrt{\frac{1}{card(O_u)} \sum_{i \in O_u} (p_{u,i} - r_{u,i})^2}, \tag{3.5}$$

where $O_u = \{i \in I | p_{u,i} \neq \star \wedge r_{u,i} \neq \star\}$ indicates set of items rated by user u having prediction values.

In different research fields, the coverage metric can be interpreted and defined differently. We define coverage metric as calculating the percentage of situation in which at least one k-nearest neighbors of the active user can rate an item which has not been rated by that active user. Here, let $S_{u,i}$ as the set of user u's neighbors which have rated the item i, and define $Z_u = \{i \in I | S_{u,i} \neq \emptyset\}$.

$$Coverage = \frac{1}{card(U)} \sum_{u \in U} (100 \times \frac{card(I_u^c \cap Z_u)}{card(I_u^c)}). \tag{3.6}$$

In addition, the reduction rate is defined as an evaluation metric, that measures the effectiveness of removing redundant users from all users. Reduction rate is given as follows:

$$ReductionRate = \frac{1}{card(U)} \sum_{u \in U} \frac{card(CN_u - CN_u^r)}{card(CN_u)}, \tag{3.7}$$

where CN_u means candidate neighbors of user u, CN_u^r represents user u's candidate neighbors after user reduction.

3.5.2 Experimental results and comparisons

We conducted experiments to demonstrate the performance of the proposed CBCF approach. In addition, using different datasets, comparisons of the CBCF and traditional UBCF approaches were performed to verify if the proposed CBCF approach could provide better recommendations or not than traditional UBCF approach. In both experiments, the Pearson correlation coefficient approach was used as the similarity measure, k-NN approach was utilized to select the neighborhood, and the adjusted weighted sum approach was used as the aggregation function. To obtain MAE, RMSR, and coverage values, according to (Herlocker, Konstan, and Riedl, 2002), we selected different size k neighborhood from candidate neighbors, $k \in \{20, 25, 30, ..., 60\}$.

Currently, researches have gotten the conclusion that there is a trade-off relationship between accuracy and coverage in traditional UBCF approach. As increasing the size of neighborhood, coverage metric increases constantly; however, for accuracy metric, it first increases and then decreases (Herlocker et al., 1999; Herlocker, Konstan, and Riedl, 2002). In our experiments, due to the size of neighborhood is in a small range, experimental results may appear that both accuracy and coverage increase as the size of neighborhood increases. However, it does not negate the trade-off relationship between accuracy and coverage in traditional UBCF approach.

Table 3.5 shows results about number of candidate neighbors for traditional UBCF and CBCF approaches in MovieLens and Jester datasets respectively. As can be seen, in MovieLens dataset, there are 943 users, so in traditional UBCF approach, all 943

TABLE 3.5: Number of candidate neighbors for traditional UBCF and CBCF approaches

	UBCF	CBCF	Reduction Rate
MovieLens	943	193	0.795
Jester	2,498	580	0.768

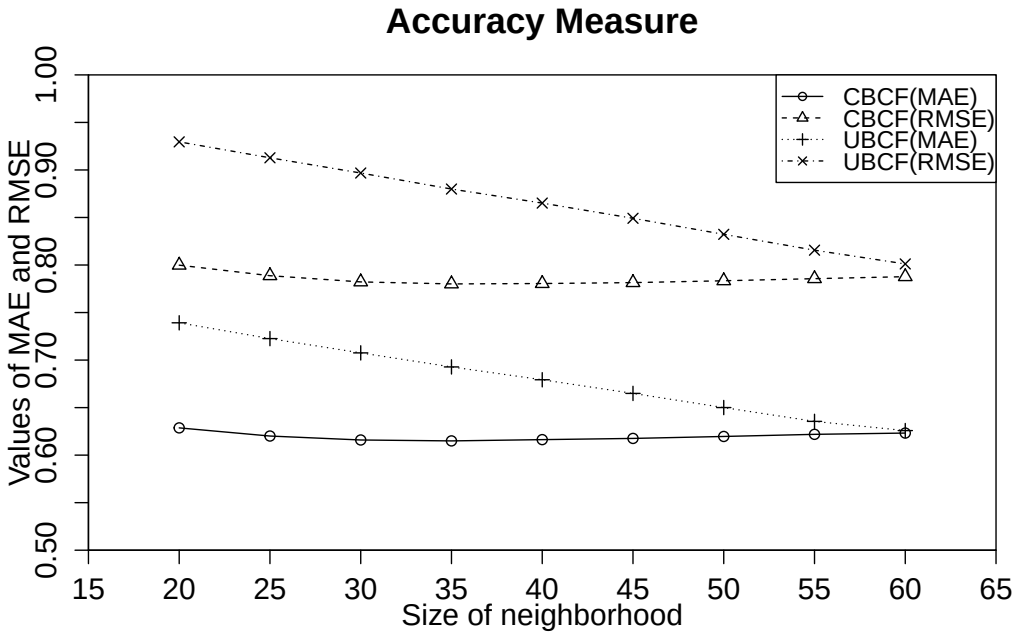

FIGURE 3.1: Accuracy results (MAE and RMSE) versus the size of neighborhood with MovieLens dataset

users will be considered as candidate neighbors. After user reduction, on average, approximately 79.5% of users are removed as redundant users, so in CBCF approach, remaining 193 users will comprise the candidate neighbors. In Jester dataset, recall that there are 2,498 users, so the number of candidate neighbors for traditional UBCF approach is 2,498. The reduction rate is 76.8%, which means approximately 76.8% of users are removed as redundant users on average, so in CBCF approach, the average number of candidate neighbors is 580.

First, we introduce comparisons between the CBCF and traditional UBCF approaches with the MovieLens dataset. Figure 3.1 shows accuracy results (MAE and RMSE) versus the size of neighborhood. As can be seen, for traditional UBCF approach, both MAE and RMSE values decrease as the size of neighborhood increases, when the size of neighborhood is 60, they obtain the least values 0.626 and 0.801 respectively. On the other hand, for CBCF approach, the MAE and RMSE values are stable, and values of two metrics are 0.623 and 0.788 when the size of neighborhood is 60. Overall, for MAE and RMSE metrics, all values of CBCF approach are lower than traditional UBCF approach, which means that the predicted scores by CBCF approach are closer to the original scores. So the proposed CBCF approach outperforms traditional UBCF in terms of MAE and RMSE. Figure 3.2 illustrates the

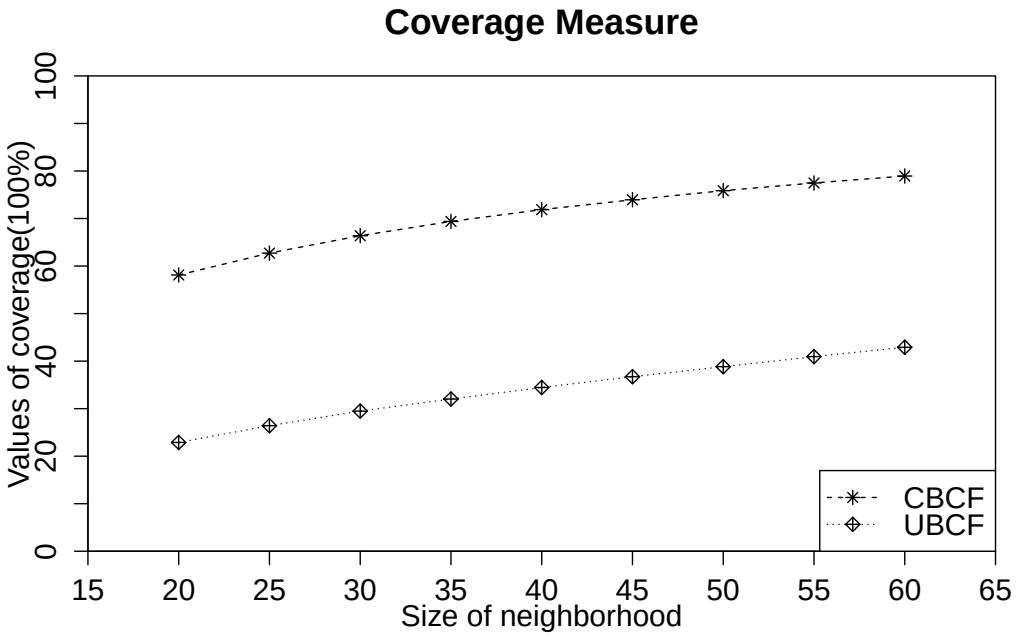

FIGURE 3.2: Coverage results versus the size of neighborhood with MovieLens dataset

coverage metric versus the size of neighborhood. As shown in figure, the coverage of both CBCF and traditional UBCF approaches increases obviously as the size of neighborhood increases. However, the coverage of proposed CBCF approach is higher than traditional UBCF in terms of different size of neighborhood, it means, CBCF approach can recommend more types of movies that the active user has not yet rated. Thus, the comparative results for CBCF and traditional UBCF obtained with MovieLens dataset indicate that, our proposed CBCF approach can select more appropriate neighborhood, and outperform the traditional UBCF approach in terms of accuracy and coverage.

Next, we illustrate comparisons between the CBCF and traditional UBCF approaches with the Jester dataset. Figure 3.3 explains accuracy results (MAE and RMSE) versus the size of neighborhood. As shown in the figure, for both CBCF and traditional UBCF approach, values of MAE and RMSE increase slightly as the size of neighborhood increases, it means the accuracy becomes lower when the neighborhood increases. And for MAE and RMSE metrics, all values of the proposed CBCF approach are higher than traditional UBCF, it indicates that CBCF approach does not outperform in terms of MAE and RMSE. Figure 3.4 shows the coverage metric versus the size of neighborhood. As can be seen, for both CBCF and traditional UBCF approaches, coverage increases slightly as the size of neighborhood increases; however, traditional UBCF is lightly higher than the CBCF approach, which means the CBCF approach cannot recommend more types of jokes for the active user. In conclusion, the comparative results between CBCF and UBCF with Jester dataset reveal that, the proposed CBCF approach is inferior to the traditional UBCF approach in terms of accuracy and coverage.

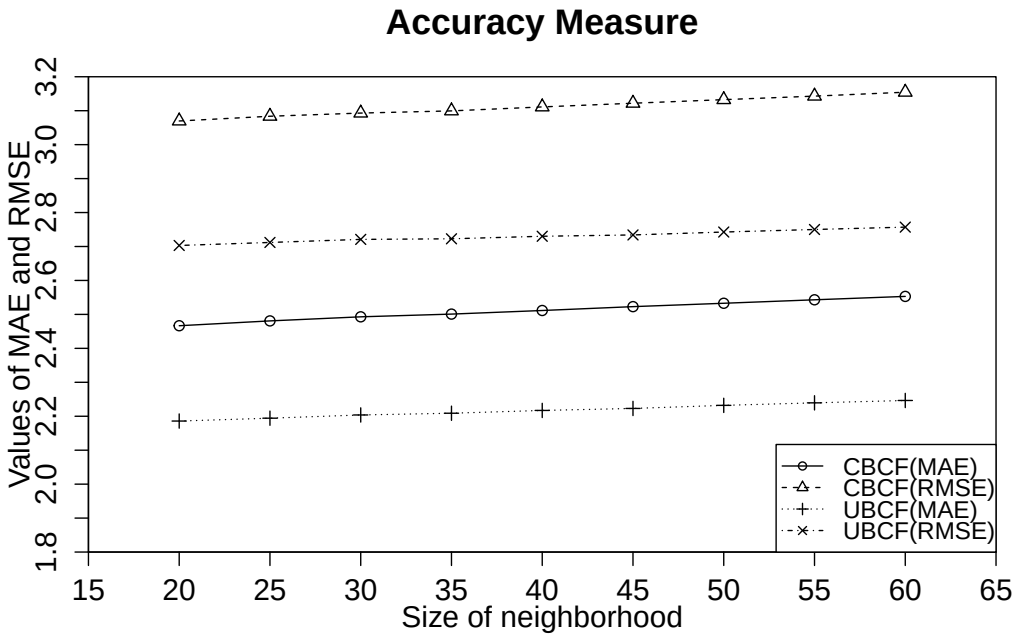

FIGURE 3.3: Accuracy results (MAE and RMSE) versus the size of neighborhood with Jester dataset

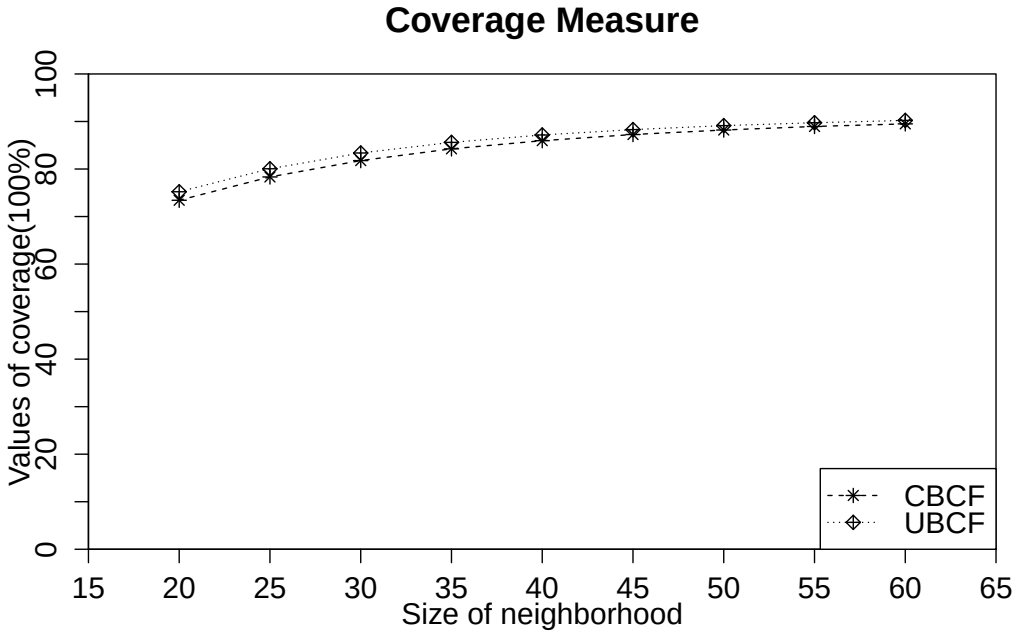

FIGURE 3.4: Coverage results versus the size of neighborhood with Jester dataset

3.5.3 Analysis and discussion

The experimental results indicate that the proposed CBCF approach demonstrates different performance with different datasets. In the MovieLens dataset experiment, there were 1,682 movies and 943 users. For each user, the number of rated items was quite smaller than the number of unrated items; therefore, this dataset is very sparse. In the proposed CBCF approach, user reduction procedure can remove redundant users which may have high similarity but can only make predictions for few types of items, reduct-users with diverse tastes comprise the candidate neighborhood, neighborhood selected from candidate neighbors can predict rating scores for more types of items, so the coverage metric has improved greatly comparing with the traditional UBCF approach. Furthermore, in RSs, although some users have higher similarity with the active user, they cannot provide predictions with high accuracy. For example, some users have rated few items, they often also have few co-rated items with the active user, even only one; however their rating scores for co-rated items are similar. In this case, they will have high similarity, but they may not have similar preferences with the active user, so they cannot provide predictions with high accuracy. As these users have fewer rated items, in CBCF approach, they are easy to be considered as redundant users to be removed, so accuracy metric of CBCF approach has a great improvement than traditional UBCF approach.

In the Jester dataset experiment, we utilized 2,498 users; however, this dataset has only 100 jokes. Thus, for each user, there are fewer unrated jokes than rated jokes. Each joke may be rated many times by different users; thus, this dataset is not sparse. Under these circumstances, all jokes can be considered as popular jokes, and each user can predict rating scores for sufficient types of jokes relative to all 100 jokes. Due to co-rated items are sufficient between each two users, so users having higher similarity with the active user can also provide predictions with higher accuracy. In CBCF approach, user reduction procedure removes some redundant users with higher similarity; however, these users can make predictions with higher accuracy, so the accuracy metric decreases comparing with UBCF approach. Besides, as there are only 100 jokes, and each user has rated sufficient jokes, it means each user can make predictions for almost same types of jokes. So after user reduction, reduct-users, which comprise candidate neighbors, may not have improvements to make predictions for more types of jokes. Therefore, comparing with traditional UBCF approach, the coverage metric of CBCF approach does not have improvements.

Generally, in practical applications, RSs must handle big data that include huge numbers of users and items. Thus, for each user, only small number of items have been rated compared to the huge number of unrated items. Thus, most RSs have sparse datasets, such as the MovieLens dataset. However, for a sparse dataset, the proposed CBCF approach can select more appropriate neighborhood than the UBCF approach and can make recommendations for the active user with satisfactory accuracy and coverage values simultaneously. Thus, the proposed CBCF approach has important significance for RSs.

3.6 Summary

UBCF approach is the most commonly used and studied technology for making recommendations in RSs. Generally, we use accuracy and diversity to evaluate an RS; however, although neighborhood selected by the traditional UBCF approach has high similarity with the active user, neighborhood tends to have similar tastes, so they are like to give high rating scores for few types of items, even only the popular

items. Therefore it is difficult for the traditional UBCF approach to provide satisfactory accuracy and coverage simultaneously.

In this chapter, we have presented the CBCF approach based on covering-based rough sets to improve the traditional UBCF approach. In the proposed CBCF approach, we add the user reduction procedure into the traditional UBCF, covering reduction in covering-based rough set is utilized to remove redundant users from all users, users having diverse preferences comprise reduct-users. Neighborhood is composed by k most similar users in candidate neighbors which consist of reduct-users, so that neighbors in the neighborhood not only have high similarity but also have diverse tastes. Our experimental results indicate that, for sparse datasets (which often appears in practical RSs), unlike traditional UBCF, the proposed CBCF approach can provide recommendations with good values of accuracy and diversity simultaneously. Thus, the proposed CBCF approach can recommend satisfactory recommendations and obtain high confidence from the active user.

Chapter 4

Improved CBCF for new users' personalized recommendations

4.1 Introduction

CF approaches are popularly used in RSs owing to their satisfactory performance. On the assumption that users who have similar preferences in the past will tend to have similar tastes in the future, UBCF has been proposed and applied in practice (Herlocker et al., 1999). UBCF can provide satisfactory recommendations utilizing only the user's historical ratings, without requiring any other special information, and it has demonstrated remarkable success in RSs (Park et al., 2015; Kaleli, 2014; Koohi and Kiani, 2016); however, for traditional UBCF, in candidate neighbors of a target user, because there exists some redundant users who have higher similarity but can make predictions only for a few types of items, the traditional UBCF usually cannot provide recommendations with satisfactory accuracy and diversity at the same time (Herlocker, Konstan, and Riedl, 2002). Many studies have been conducted to increase the diversity of recommendations based on UBCF. Among these studies, some approaches can improve diversity significantly, but accompanied by losses in accuracy (Adomavicius and Kwon, 2011; Adomavicius and Kwon, 2012). Although some methods can improve accuracy and diversity simultaneously, they require additional information that is often not available or incomplete (Gan and Jiang, 2013; Niemann and Wolpers, 2013). CBCF is a useful approach, falling in the latter research line mentioned above, that we proposed in our previous chapter to improve UBCF by removing redundant candidate neighbors. However, all of these studies focus on providing satisfactory recommendations for an active user which often has sufficient rating information; but a new user in RSs differs in some respects (e.g., number of ratings or rating score proportion), recommendation difficulty is increased (Lika, Kolomvatsos, and Hadjiefthymiades, 2014; Son, 2016; Bobadilla et al., 2012; Liu et al., 2014a). Therefore, researchers face the difficult problem of how to utilize only easily obtained information to provide recommendations for a new user with satisfactory accuracy and diversity simultaneously (Ahn, 2008; Tyagi and Bharadwaj, 2012; Chen et al., 2013).

In this chapter, we aim to improve the previous CBCF to provide satisfactory accuracy and diversity of recommendations simultaneously for a new user in RSs. Because a new user often has few ratings, the previous CBCF cannot utilize the insufficient information to remove redundant candidate neighbors for a new user effectively. In our improved CBCF, in order to remove as many redundant candidate neighbors as possible for a new user, by analyzing the proportion and characteristic of new users' rating scores, we reconstruct the decision class by the niche items which have fewer ratings from users. In this way, different from the previous CBCF,

our improved CBCF could remove redundant candidate neighbors efficiently without requiring any special additional information. Experimental results indicate that our improved CBCF can not only improve the accuracy metric, but also increase the diversity of recommendations for a new user for the sort of sparse datasets that often occur in connection with real RSs.

The remainder of this chapter is organized as follows. In Section 4.2, we introduce the traditional UBCF approach and review some studies that attempt to improve UBCF to obtain better performance. In Section 4.3, we present our problem setting through an analysis of real-world datasets. In Section 4.4, we explain the motivation and detailed procedures of our improved CBCF, then make comparisons between the previous CBCF and improved CBCF. In Section 4.5, we describe our experiments and compare the results of our improved CBCF with other existing work. Finally, in Section 4.6, we draw the summary of this chapter.

4.2 Related works

UBCF is an important approach popularly used in RSs that utilizes only a user's historical ratings, without any special additional information; however, the traditional UBCF cannot provide recommendations with satisfactory accuracy and diversity simultaneously (Herlocker, Konstan, and Riedl, 2002). In addition, recent research has concluded that gains in RS diversity can frequently be accompanied by losses in accuracy, making it difficult to select a reasonable trade-off between accuracy and diversity (Liu, Shi, and Guo, 2012; Zhou et al., 2010). More detailed information and procedures of UBCF could be found in Subsection 1.2.1.

To increase the diversity of recommendations while maintaining comparable levels of recommendation accuracy, we developed in our previous work a CBCF approach to improve the traditional UBCF. In previous CBCF, combining with the characteristics of redundant users in UBCF and redundant elements in covering-based rough sets, we inserted a neighbor selection procedure into the traditional UBCF that could remove redundant candidate neighbors by covering reduction algorithm. To remove as many redundant users as possible, according to the sufficient information from an active user, we first extracted relevant attributes of the active user, then constructed decision class by all items that fit the active user's relevant attributes, and reduced the domain from all items to decision class. Because the CBCF approach could select more appropriate users to comprise the neighborhood of an active user, CBCF was able to provide recommendations with satisfactory accuracy and diversity simultaneously for an active user. More detailed information could be found in Chapter 3.

Said, Jain, and Albayrak (Said, Jain, and Albayrak, 2012) investigated the effects of weighting factors on different types of users. To improve the diversity of recommendations for a new user, when computing the similarity, they decreased the impact of items rated by many users, in other words, popular items. In addition, they increased the impact of items rated by few users. This approach can improve the diversity of recommendations for a new user with acceptable accuracy; however, it is unstable and performs differently for different selected datasets.

4.3 Analysis and problem setting

In this section, first, we analyze two popular datasets that are often used to evaluate RS approaches. Then, in accordance with the analysis result, we discuss the problem

TABLE 4.1: Proportion of items and ratings in the MovieLens dataset

	Item number	Item rate	Rating number	Rating rate
Ratings $\geq$ 10K	174	1.63%	2,757,120	27.57%
5K $\leq$ Ratings $<$ 10K	296	2.77%	2,112,854	21.12%
1K $\leq$ Ratings $<$ 5K	1,564	14.64%	3,562,589	35.62%
Ratings $\leq$ 1K	8,647	80.96%	1,569,491	15.69%

TABLE 4.2: Proportion of items and ratings in the Netflix dataset

	Item number	Item rate	Rating number	Rating rate
Ratings $\geq$ 50K	501	2.82%	45,020,066	45.63%
10K $\leq$ Ratings $<$ 50K	1,541	8.67%	34,889,199	35.36%
1K $\leq$ Ratings $<$ 10K	5,084	28.61%	17,193,080	17.42%
Ratings $\leq$ 1K	10,644	59.90%	1,569,491	1.59%

setting of this chapter.

4.3.1 Data analysis

Here, we analyze two popular datasets that were collected from the real world and are often used to evaluate RSs. One is the MovieLens 10M dataset (Herlocker et al., 1999), obtained from the website of the GroupLens lab. This dataset contains 71,567 users, 10,681 movies, and a total of 10,002,054 ratings on a scale of {0.5, 1, 1.5, ..., 5}. Each user has rated at least 20 movies, resulting in a sparsity of 95.81%. The other is the Netflix dataset, obtained from the Netflix Prize website (http://www.netflixprize.com). This dataset contains a total of 100 million ratings from 480,189 users over 17,770 movies (98.81% sparsity). The ratings are on a {1, 2, 3, 4, 5} scale, and each user has rated a different number of movies. In this chapter, considering the size of datasets and the number of experiments we conducted, we used the full datasets for analysis and smaller subsets for some of the experiments in Section 4.5.

First, we perform statistical analyses for these two datasets. Tables 4.1 and 4.2 show the number of items and the corresponding ratings, as well as their proportions according to the different number of ratings. As shown in the tables, for the MovieLens dataset, items that have more than 5K ratings account for only 4.40% of all items, but their corresponding ratings comprise 48.69% of all the ratings. In the Netflix dataset, this performance is more obvious, even though items that have more than 50K ratings comprise only a 2.82% proportion of all items, with ratings corresponding entirely to them accounting for 45.63% of all ratings. From the data analysis above, we can conclude that in real-world database, after sorting all items by descending order according to the number of ratings, the top fewer items usually correspond to a large proportion of the ratings. Therefore, in this chapter, we call them popular items.

Next, we consider the proportion of rating scores on popular items. Figures 4.1 and 4.2 show the results. As found in the figures, in the MovieLens dataset, the rating range is from 0.5 to 5 with half-star increments, but most of the rating scores are concentrated on {3, 4, 5}. In the Netflix dataset, although the rating range is just from 1 to 5, most of the rating scores are also included in {3, 4, 5}. These results indicate that users' rating scores on popular items are relatively concentrated, with the difference between these rating scores not being very large. Because the

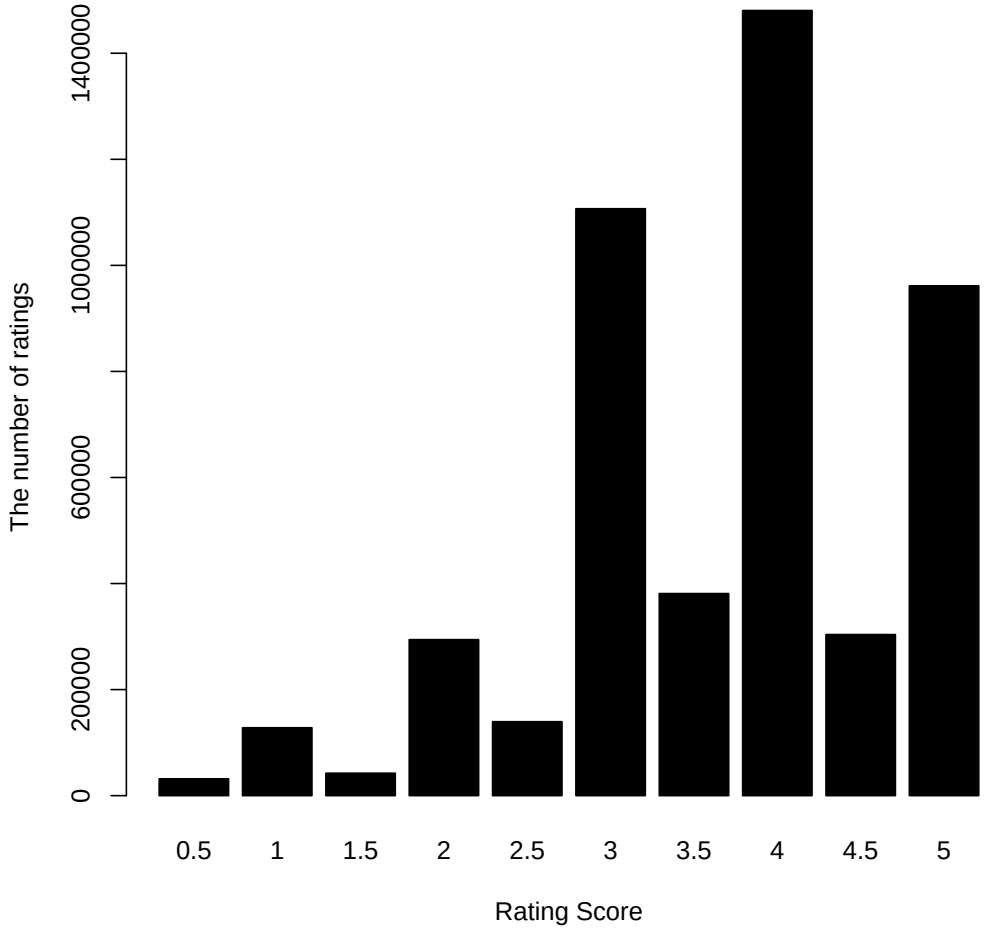

FIGURE 4.1: Proportion of rating scores on popular items in the MovieLens dataset

similarity computation is based on the rating scores of co-rated items, the closer the rating scores, the more similar the two users. Therefore, if most of the co-rated items between two users are popular items, then the rating scores between them will be similar, and they will achieve higher similarity. From the above, we can conclude that if co-rated items between two users concentrate on popular items, the similarity between them will be higher.

Finally, we discuss the percentage of ratings on popular items by users with ratings no more than {20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200} in the two datasets. Figures 4.3 and 4.4 show the results. As shown in the figures, the two datasets have the same performance in that the percentage of ratings on popular items decreases as the number of ratings by users increases. Users with having no more than 20 ratings have the highest percentage, almost 74.72% in the MovieLens dataset and 72.74% in the Netflix dataset. In general, new users often have fewer ratings (i.e., no more than 20 ratings). Therefore, we can conclude that most ratings of new users concentrate on popular items.

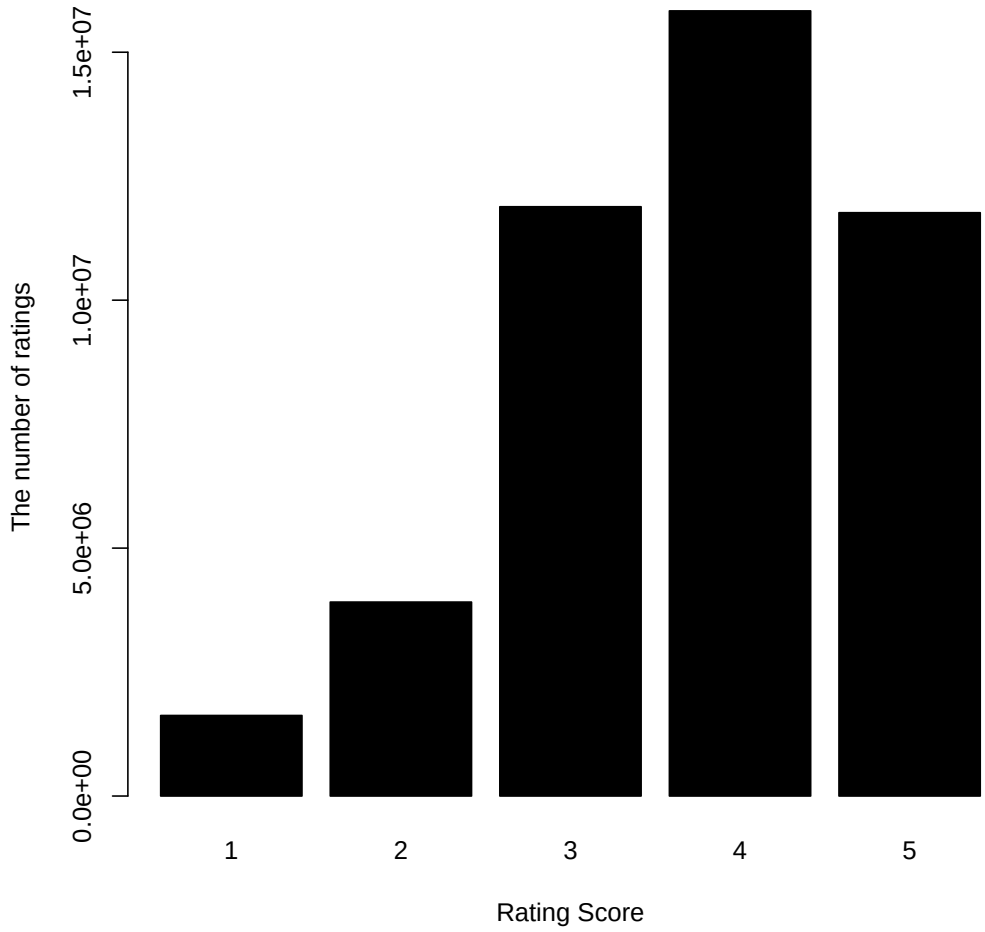

FIGURE 4.2: Proportion of rating scores on popular items in the Netflix dataset

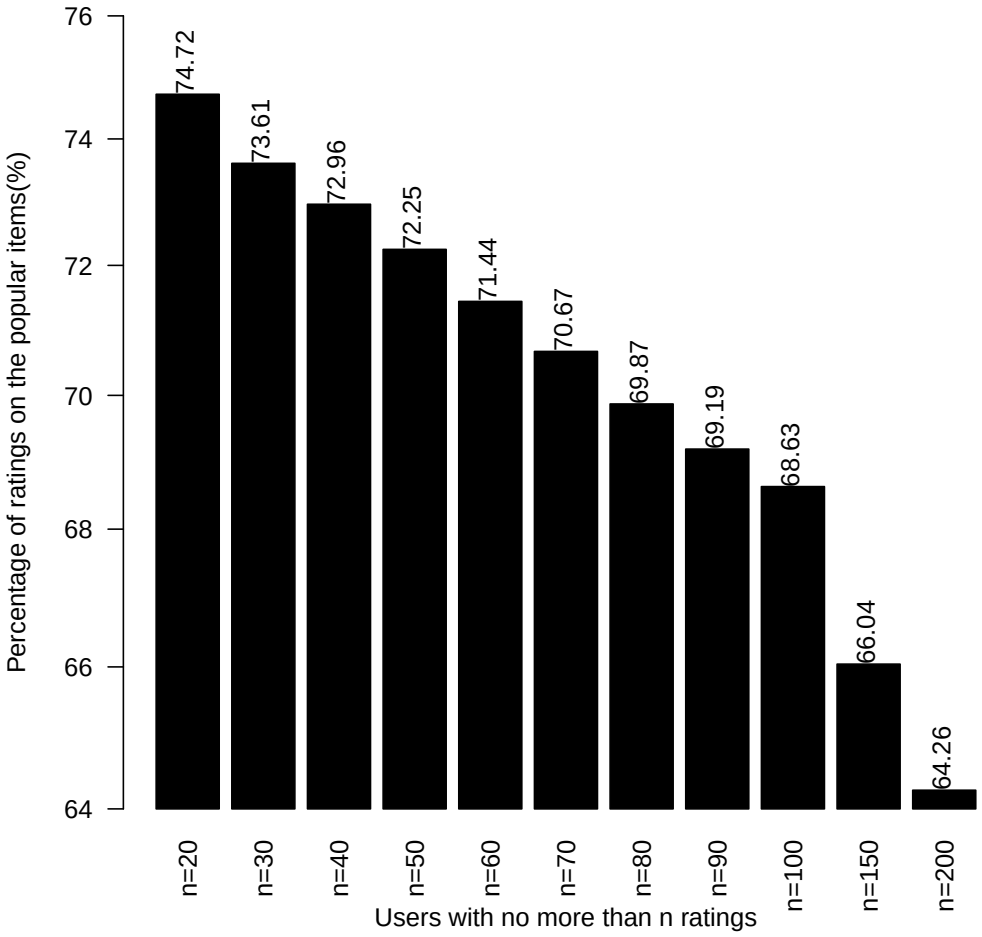

FIGURE 4.3: Percentage of ratings on popular items by users with no more than n ratings in the MovieLens dataset

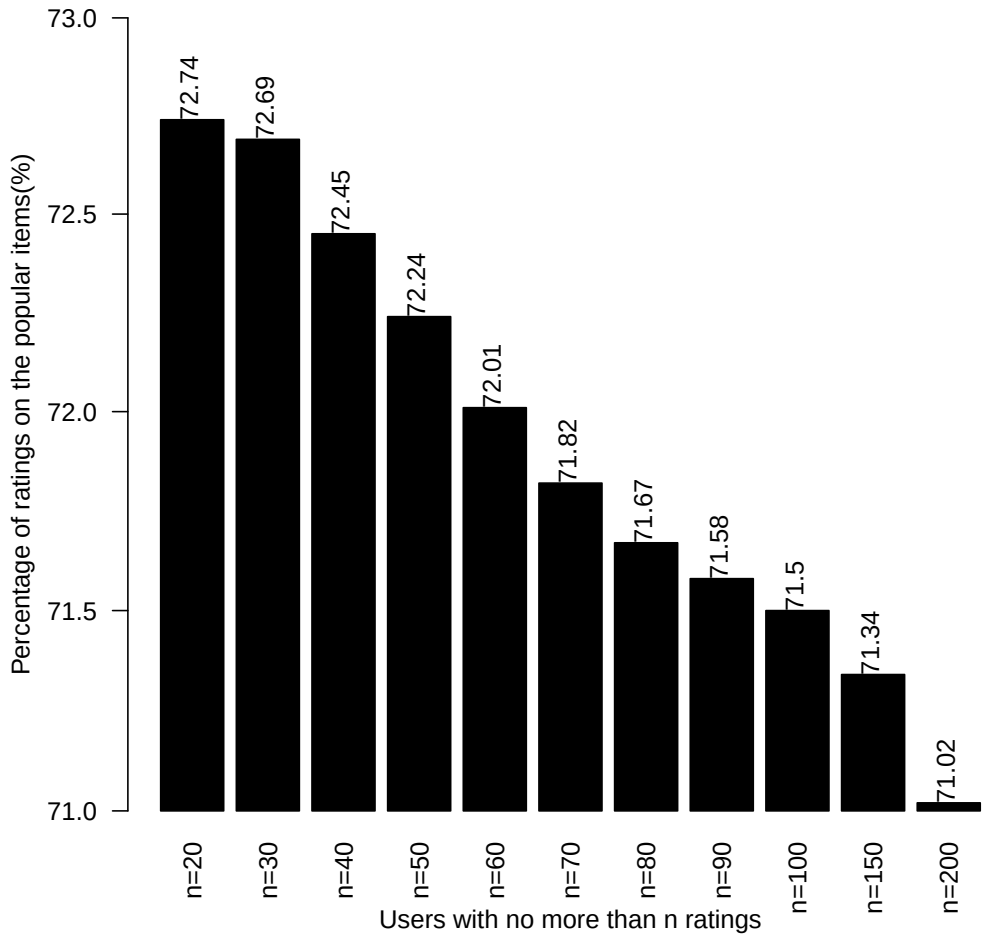

FIGURE 4.4: Percentage of ratings on popular items by users with no more than n ratings in the Netflix dataset

4.3.2 Problem setting

According to our conclusions obtained above, if co-rated items between two users concentrate on popular items, then they will have higher similarity. Because most ratings of a new user are on popular items, if there are other users whose ratings also concentrate on popular items, the similarity between them will be very high, and these users can easily be selected into the neighborhood of the new user. Therefore, in traditional UBCF, a new user's neighborhood is usually comprised of users whose ratings concentrate on popular items.

However, neighborhoods comprised of users whose ratings concentrate on popular items can make predictions only for fewer types of items, perhaps even only popular items. Hence, in traditional UBCF, candidate items with high predicted scores are the most popular items, resulting in a low diversity of recommendations for a new user. In addition, these users can predict accurate rating scores only for popular items rather than for all types of items. Thus, the accuracy of recommendations for a new user will also be unsatisfactory. Here, we define users whose ratings concentrate on popular items as redundant users of a new user. In the traditional UBCF approach, as the neighborhood of a new user often contains many redundant users, recommendations they produce might concentrate on popular items. A new user's acceptance of these recommendations will substantially increase the percentage of ratings on popular items and further improve the similarity between the new user and redundant users. Under these circumstances, redundant users are easier to select into a neighborhood. Finally, as a consequence, a vicious circle is established, and a new user might be able to obtain only recommendations determined by popular items.

4.4 Improved CBCF for a new user's personalized recommendations

Here, we first discuss the motivation of our improved CBCF. Then, we introduce the information about reconstruction of the decision class for a new user. Next, we describe detailed process of improved CBCF, and make comparisons between it with the previous CBCF.

4.4.1 Motivation of improved CBCF approach

In order to provide recommendations with satisfactory accuracy and diversity simultaneously for a new user, our improved CBCF aims to remove as many redundant users as possible, and utilizes the remaining more appropriate users to comprise the neighborhood of a new user.

The target of previous CBCF is to provide satisfactory recommendations for an active user. Because an active user has rated many items, there is sufficient information that could be utilized. Therefore, in the previous CBCF, the decision class consists of items that fit the active user's relevant attributes, and relevant attributes can be obtained from sufficient rating information. However, for a new user, ratings are usually very few, and it is unreliable to extract relevant attributes according to a new user's rating information. Moreover, in the previous CBCF, the item attribute matrix had to be inputted as the indispensable condition, even though some datasets do not have this information. Therefore, for a new user's personalized recommendations, in our improved CBCF approach, we must make full use of the characteristic

of a new user (e.g., fewer ratings or ratings concentrating on popular items), and reconstruct the decision class while ensuring as far as possible that the new approach requires no special additional information.

4.4.2 Reconstruction of decision class for a new user

In accordance with the discussion in Section 4.4.1, we reconstruct the decision class for the new user as the set of niche items in the dataset used for recommendation. As we discussed in Section 4.3.1, in real-world database, after sorting all items by descending order according to the number of ratings, the top fewer items correspond to a large population of the ratings, we called them popular items. We then define niche items as items that are not popular in the dataset.

There are the following two reasons why we reconstruct the decision class for the new user as the set of niche items:

1. Redundant candidate neighbors for a new user are able to be removed as many as possible;
2. The decision class as the set of niche items is easily constructed from the user-item matrix.

The first reason is that we can remove redundant candidate neighbors for a new user as many as possible. It is because that in candidate neighbors of a new user, items rated by a redundant user U_1 are the most popular items, with the result that the set of niche items rated by U_1 is very small, perhaps even empty. Under these circumstances, it is very easy to find another user U_2 whose rated niche items' set includes U_1's. In other words, in the entire set of niche items, U_2 can not only make predictions for items as U_1 does, but also predict ratings for other types of niche items. Therefore, U_2 might be more appropriate for being selected into the neighborhood than U_1, even though the similarity of U_1 might be a little higher. When reducing the domain from item set I to the decision class comprised by the set of niche items, some redundant users who have not rated niche items will be removed first. Then because of a redundant element in a covering is also included in other elements, which has the same characteristics as the set of items rated by redundant users, we can utilize covering reduction to remove redundant users. Even using the covering reduction algorithm cannot remove all the redundant users, but it can remove most of them, and our experiments in Section 4.5 confirm this.

The second reason is that, by utilizing the set of niche items, decision class can be constructed easily without requiring any other special additional information. It is because that the niche items could be extracted easily from the user-item rating matrix which can be obtained from almost all types of RSs.

Algorithm 4.1 constructs a decision class D_{nu} for new user from the user-item rating matrix RM and the ratio threshold rt $(0 < rt < 1)$. In this algorithm, the set of popular items is regarded as the top $(1 - rt) \times 100\%$ items which have the largest number of ratings in I and the decision class D_{nu} for new user is constructed by removing popular items from the set of all items I.

4.4.3 Procedures of improved CBCF approach

In this subsection, we describe the detailed steps of the improved CBCF approach and provide algorithm information in Algorithm 4.2.

Algorithm 4.1 Decision class construction algorithm for new user

Input: User-item rating matrix RM, ratio threshold rt
Output: Decision class D_{nu} for new user
1: **for all** $i \in I$ **do**
2: $n_i \leftarrow$ Count the number of users $u \in U$ such that $r_{u,i} \neq \star$
3: **end for**
4: $D_{nu} = I$
5: **while** $\frac{|D_{nu}|}{|I|} \geq rt$ **do**
6: $i \leftarrow$ Select an item with highest value n_i in D_{nu}
7: $D_{nu} \leftarrow D_{nu} \setminus \{i\}$
8: **end while**
9: **return** D_{nu}

Step 1: User reduction. First set I as the domain, with rated items of each user comprising a set in domain I. We construct a decision class D_{nu} for a new user nu, consisting of all niche items. Then to remove as many redundant users as possible, we reduce the domain from item set I to decision class D_{nu}, and for each user $u \in U$, user u's rated items I_u in domain D_{nu} comprise the set C_u, where

$$C_u = I_u \cap D_{nu}. \tag{4.1}$$

Let $C^* = D_{nu} - \cup C_u$. Then, $C = \{C_1, C_2 ... C_{|U|}, C^*\} - \{\emptyset\}$ is a covering for the new user in domain D_{nu}. Next, based on the covering reduction algorithm for covering-based rough sets, redundant elements are removed from covering C to obtain reduct(C). We can obtain a new user's reduct-users U^r, where

$$U^r = \{u \in U | C_u \in reduct(C)\}. \tag{4.2}$$

Step 2: Similarity computation. Users in U^r comprise candidate neighbors CN^r_{nu} of the new user nu. According to the historical rating information, compute the similarity $sim(nu, u)$ between the new user nu and each user $u \in CN^r_{nu}$ according to the similarity measure.

Step 3: Neighborhood selection. The new user nu's neighborhood $N^r_{nu}(k)$ consists of the k most similar (nearest) users in CN^r_{nu}.

Step 4: Rating prediction. Based on the rating information of neighborhood $N^r_{nu}(k)$, we predict the rating score $p_{nu,i}$ for each item i in unrated item set I^c_{nu} of the new user nu.

Step 5: Item recommendation. According to the predicted rating scores, select the top N items that have the highest $p_{nu,i}$ from the candidate items as the recommendations for the new user nu.

4.4.4 Comparisons between the previous CBCF and improved CBCF

As the same to the previous CBCF, our improved CBCF also has inserted a neighbor selection procedure into the traditional UBCF. However, because the target user is different, neighbor selection methods for two approaches are also different. For the previous CBCF, it aims to provide satisfactory recommendations for an active user which often has many ratings. In order to improve the quality of recommendations, in previous CBCF, we construct the decision class through the sufficient rating information obtained from the active user, and the input information needs both user-item rating matrix and item attribute matrix. Different from the previous

Algorithm 4.2 Improved CBCF approach

Input: User-item rating matrix RM and a new user nu.
Output: Recommended items set of size N for the new user nu.
k : Number of users in the neighborhood $N^r_{nu}(k)$ of the new user nu.
N : Number of items recommended to the new user nu.
D_{nu} : Decision class of the new user nu.
U^r : Users after user reduction, reduct-users.
I^c_{nu} : Items that have not yet been rated by the new user nu.
CN^r_{nu} : Candidate neighbors of the new user nu after user reduction.
$p_{nu,i}$: Rating prediction of item i for the new user nu.
1: **for** each user $u \in U$ **do**
2: $C_u = I_u \cap D_{nu}$.
3: **end for**
4: Let C$^* = D_{nu} - \cup C_u$. Then, $C = \{C_1, C_2 ... C_{|U|}, C^*\} - \{\emptyset\}$ is a covering for the new user nu in domain D_{nu}.
5: $reduct(C) = STCRA(C)$
6: Reduct-user $U^r = \{u \in U | C_u \in reduct(C)\}$.
7: $CN^r_{nu} = U^r$, compute the similarity between the new user nu and each user $u \in CN^r_{nu}$
8: **for** each item $i \in I^c_{nu}$ **do**
9: Find the k most similar users in CN^r_{nu} to comprise neighborhood $N^r_{nu}(k)$;
10: Predict rating score $p_{nu,i}$ for item i by neighborhood $N^r_{nu}(k)$;
11: **end for**
12: Recommend to the new user nu the top N items having the highest $p_{nu,i}$.

CBCF, for our improved CBCF, the target user is a new user, we construct the decision class by niche items which could be extracted easily from the user-item rating matrix. Moreover, our improved CBCF only needs to input the user-item rating matrix rather than any other special information from a new user. Through the above comparisons, we can find that, to provide personalized recommendations, our improved CBCF needs less information (e.g., input information and the target user's ratings information) than previous CBCF. Because a new user usually has insufficient information which could be utilized, so our improved CBCF is more suitable for the new user's personalized recommendations.

4.5 Experiments and evaluations

In this section, we introduce the evaluation dataset and metrics, and compare the performance of the improved CBCF approach with other work using different datasets.

4.5.1 Experimental setup and evaluation metrics

In our experiments, we used the MovieLens and Netflix datasets to evaluate our improved CBCF approach. We also used the Jester dataset (Goldberg et al., 2001), because it has characteristics different from the former two datasets. This dataset contains ratings of 100 jokes from 24,983 users (27.53% sparsity). From the information of these three datasets, we find that MovieLens and Netflix are the same type of dataset, each containing a huge number of items; however, each user has rated fewer items, with the number of rated items being substantially smaller than the

number of unrated items. Therefore, the two datasets are very sparse, and popular items can be found easily. In contrast, the Jester dataset contains only 100 items, each user has rated a sufficient number of items relative to all items, and unrated items are fewer than rated ones; hence, this dataset is not sparse. In addition, in the Jester dataset, every item has been rated by many users, with the result that it is difficult to distinguish whether an item is popular. In fact, we can even say that each item is popular.

To obtain a dataset of manageable size in our experiment, for the MovieLens and Netflix datasets, we select 1,000 users and 1,000 items from each of the original datasets. First, we select 1,000 items based on the item and rating structure of the original dataset by stratified sampling (detailed information is given in Tables 4.3 and 4.4), and we set the ratio threshold $rt = 0.95$, it means the top 5% of items that have the most ratings as popular items, with the remaining 95% items being considered as niche items in our experiments. Next, because we only select 1,000 experimental users from the whole users (71,567 users in the MovieLens dataset, 480,189 users in the Netflix dataset), the sample size is very small relative to the original data. In order to ensure experimental users as similar with the original datasets as possible, we extract users that satisfy the following two conditions as candidate users:

1. The user has at least one rated item in the selected 1,000 items;
2. Percentage of popular items in the user's rated items is no less than the minimum value showed in Figures 4.3 and 4.4.

For example, in the case of the MovieLens dataset, a user is extracted as a candidate user if the user has at least one item in the selected 1,000 items and the percentage of popular items in this user's rated items is no less than 64.26%. Similarly, for the Netflix dataset, the percentage of popular items in a candidate user's rated items is no less than 71.02%. In this way, users who have quite different rating proportion with original datasets will not be extracted, so that every experimental user selected from candidate users could have a rating proportion as closely with the original datasets as possible. Then, we select 200 test users and 800 training users from the candidate users. First, we randomly select 200 users who have ratings numbering no less than five and no more than 25 as the test users, and randomly mask 20% of the ratings in each test user. We regard every test user as a new user, and each new user has at most 20 ratings as training ratings by the masking of ratings and at most 5 ratings as test ratings in our experiments. Finally, 800 users are randomly selected from candidate users as the training users.

In contrast, for the Jester dataset, because there are only 100 items, we treat the top 50 items that have the most ratings as popular items and the remaining items as niche items in the experiments. Since each user has rated 36 or more jokes in Jester dataset, here we randomly select 200 test users and remove some of their ratings to make them as new users, and 800 users are selected randomly as training users. To avoid the impact of accidental phenomena, we repeat the experiments 20 times for each dataset and compute the average values as our results. After selecting our experimental items and users from original datasets, the average sparsities of selected datasets from MovieLens, Netflix, and Jester are 98.90%, 88.83%, and 36.42%, respectively. Although the sparsities are a little different with original datasets, we can also call selected datasets from the MovieLens and Netflix are sparse, and selected dataset from the Jester is not sparse.

TABLE 4.3: Experimental items versus original data in the MovieLens dataset

	Experimental items	Original items	Item rate
Ratings $\geq$ 10K	16	174	1.63%
5K $\leq$ Ratings $<$ 10K	28	296	2.77%
1K $\leq$ Ratings $<$ 5K	146	1,564	14.64%
Ratings $\leq$ 1K	810	8,647	80.96%
	1,000	10,681	100%

TABLE 4.4: Experimental items versus original data in the Netflix dataset

	Experimental items	Original items	Item rate
Ratings $\geq$ 50K	28	501	2.82%
10K $\leq$ Ratings $<$ 50K	87	1,541	8.67%
1K $\leq$ Ratings $<$ 10K	286	5,084	28.61%
Ratings $\leq$ 1K	599	10,644	59.90%
	1,000	17,770	100%

To measure the performance of the improved CBCF approach, we used the mean absolute error (MAE) and root mean square error (RMSE) to represent the accuracy of recommendations. In addition, we used coverage, mean personality (MP), and mean novelty (MN) to evaluate the diversity of recommendations. In accordance with Herlocker's research (Herlocker, Konstan, and Riedl, 2002), to maintain real-time performance, we selected different sized k neighborhoods from candidate neighbors, $k \in \{20, 25, 30, ..., 60\}$. The detailed information of MAE, RMSE and coverage metrics can be found in Subsection 3.5.1.

MP indicates the average degree of overlap between every two users' recommendations. For example, for two users u_i and u_j, we count the number of recommendations of the corresponding top N items, $Rec_i(N)$ and $Rec_j(N)$, and further normalize this number by the threshold value N to obtain the degree of overlap between two sets of recommendations. It is clear that an approach of higher recommendation diversity will have a larger MP. As discussed by Gan and Jiang (Gan and Jiang, 2013), we use $N = 20$ in our calculation of this metric.

$$MP(N) = 1 - \frac{1}{N} \frac{2}{|U|(|U|-1)} \sum_{1 \leq i < j \leq |U|} |Rec_i(N) \cap Rec_j(N)|. \tag{4.3}$$

MN indicates the novelty of recommendations provided to users. First, it calculates the fraction of users who have ever rated each recommendation, and then computes the sum over all recommendations in $Rec_m(N)$ to obtain the novelty for user u_m. Finally, we calculate the average novelty over all users.

$$MN(N) = -\frac{1}{|U|} \sum_{1 \leq m \leq |U|} \sum_{n \in Rec_m(N)} log_2 f_n, \tag{4.4}$$

where f_n indicates the fraction of users who rated the n^{th} item. We also set $N = 20$ in the calculation of this metric, and an approach will have a larger MN if it can make newer recommendations.

TABLE 4.5: Number of candidate neighbors for the traditional UBCF and CBCF approaches

	UBCF	CBCF	Reduction rate
MovieLens	800	331	0.586
Netflix	800	369	0.539
Jester	800	588	0.265

In addition, the reduction rate is also used to measure the effectiveness of removing redundant users from among all users. The definition of reduction rate can be found in Equation 3.7.

4.5.2 Experimental results and comparisons

To show the performance of the improved CBCF, we compared it with traditional UBCF. Comparisons were also made with the linear collaborative filtering (LINCF) and inverse user frequency collaborative filtering (IUFCF) presented by Said et al. (Said, Jain, and Albayrak, 2012). For convenience, we refer to the improved CBCF approach as CBCF in the rest of this section. In all of the experiments, we used the Pearson correlation coefficient as the similarity measure, and the weighted sum approach to predict the rating scores. Finally, we selected the top N candidate objects with the highest predicted rating scores as the recommendations for a new user.

Table 4.5 shows the results for the number of candidate neighbors for the traditional UBCF and CBCF approaches on the MovieLens, Netflix, and Jester datasets. As can be seen, in the MovieLens and Netflix datasets, after user reduction, on average, more than half of the users are removed as redundant users, with the result that in the CBCF approach, fewer than half of the users will remain to comprise the candidate neighbors. In the Jester dataset, the reduction rate is slightly lower, which means that approximately 26.5% of the users are removed as redundant users on average, with the result that in the CBCF approach, the average number of candidate neighbors is 588.

Figures 4.5, 4.6, 4.7, and 4.8 show the results for MAE and RMSE on the MovieLens and Netflix datasets. As shown in the figures, with increasing neighborhood size, both the MAE and RMSE values decrease in the two datasets. In the MovieLens dataset, both the MAE and RMSE values of the UBCF approach are higher than in the other three approaches, because the lower these values, the better the accuracy, indicating that the other three approaches have improved the accuracy of the traditional UBCF approach. Furthermore, although the MAE and RMSE values of CBCF are higher than those of IUFCF and LINCF in the beginning, CBCF decreases faster than the other approaches as the neighborhood size increases. This indicates that the CBCF approach can provide recommendations with higher accuracy than the other approaches as the neighborhoods grow. In the Netflix dataset, the MAE and RMSE values of UBCF are lower than those of LINCF and IUFCF, indicating that the accuracy of UBCF outperforms that of LINCF and IUFCF; however, the values of UBCF are also higher than those of CBCF, demonstrating that CBCF has improved the accuracy of traditional UBCF, and can provide recommendations with better accuracy than the other approaches.

In contrast, these approaches have different performances on the Jester dataset. As can be seen in Figures 4.9 and 4.10. MAE and RMSE of UBCF have the lowest values, indicating that the accuracy of UBCF is highest among these approaches. This indicates that the CBCF approach and other related work cannot improve the

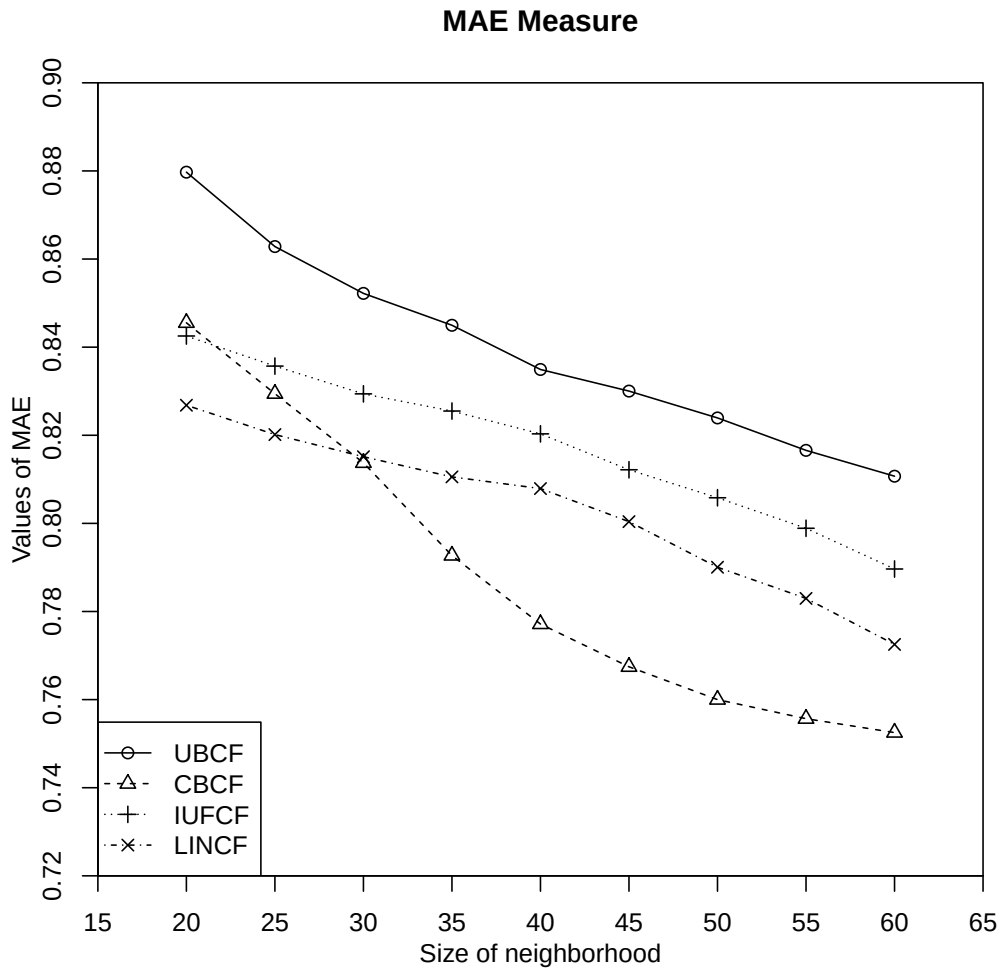

FIGURE 4.5: Result of MAE measure on the MovieLens dataset

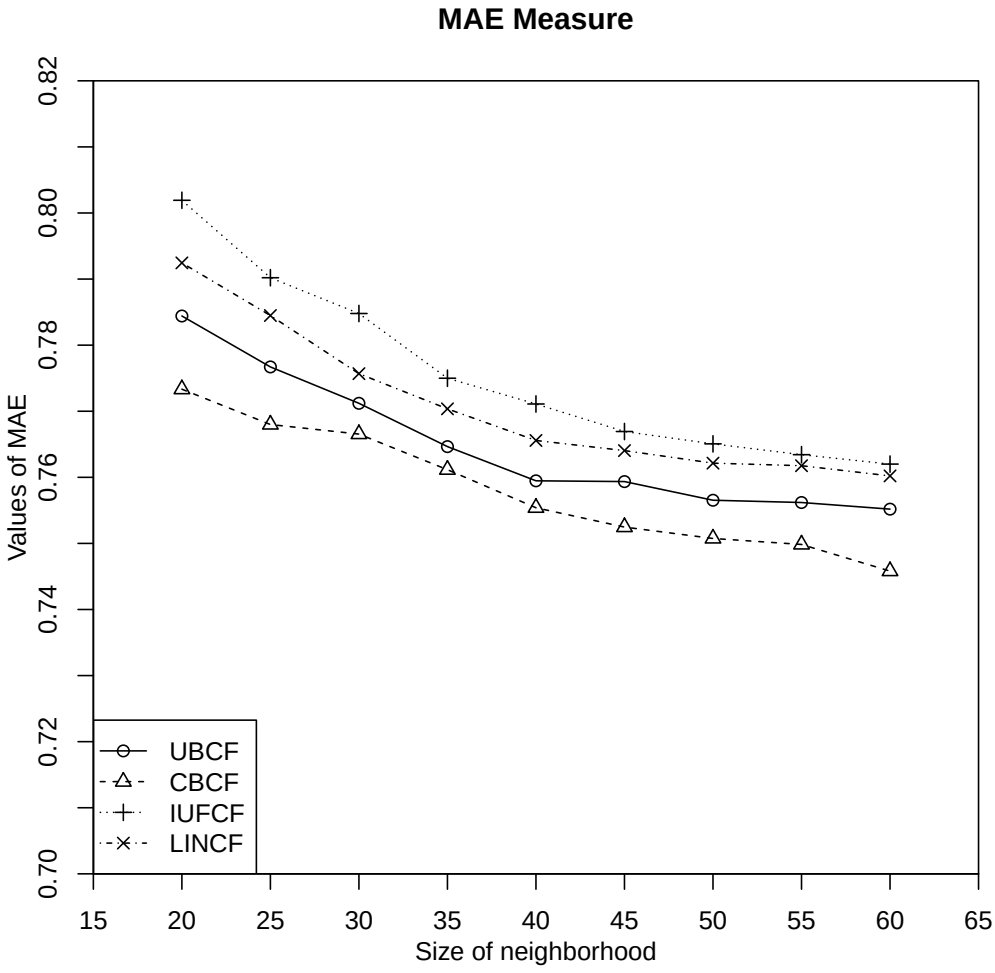

FIGURE 4.6: Result of MAE measure on the Netflix dataset

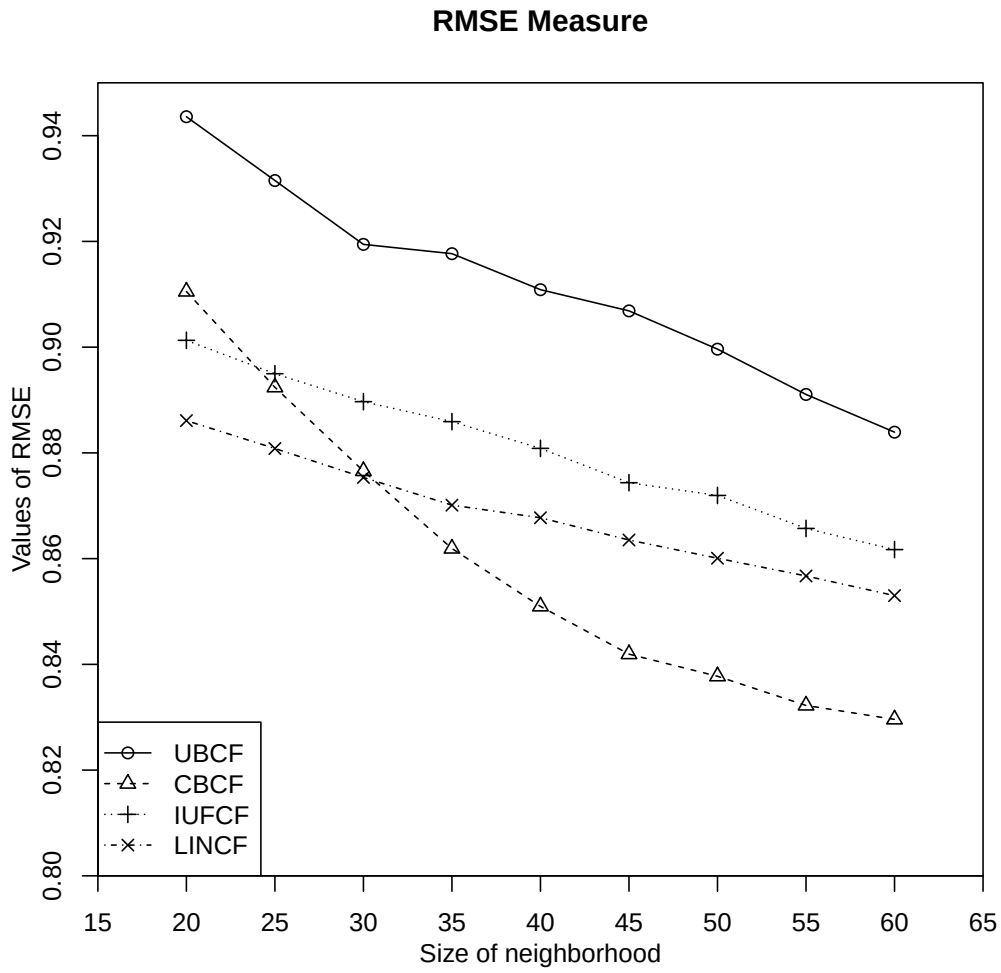

FIGURE 4.7: Result of RMSE measure on the MovieLens dataset

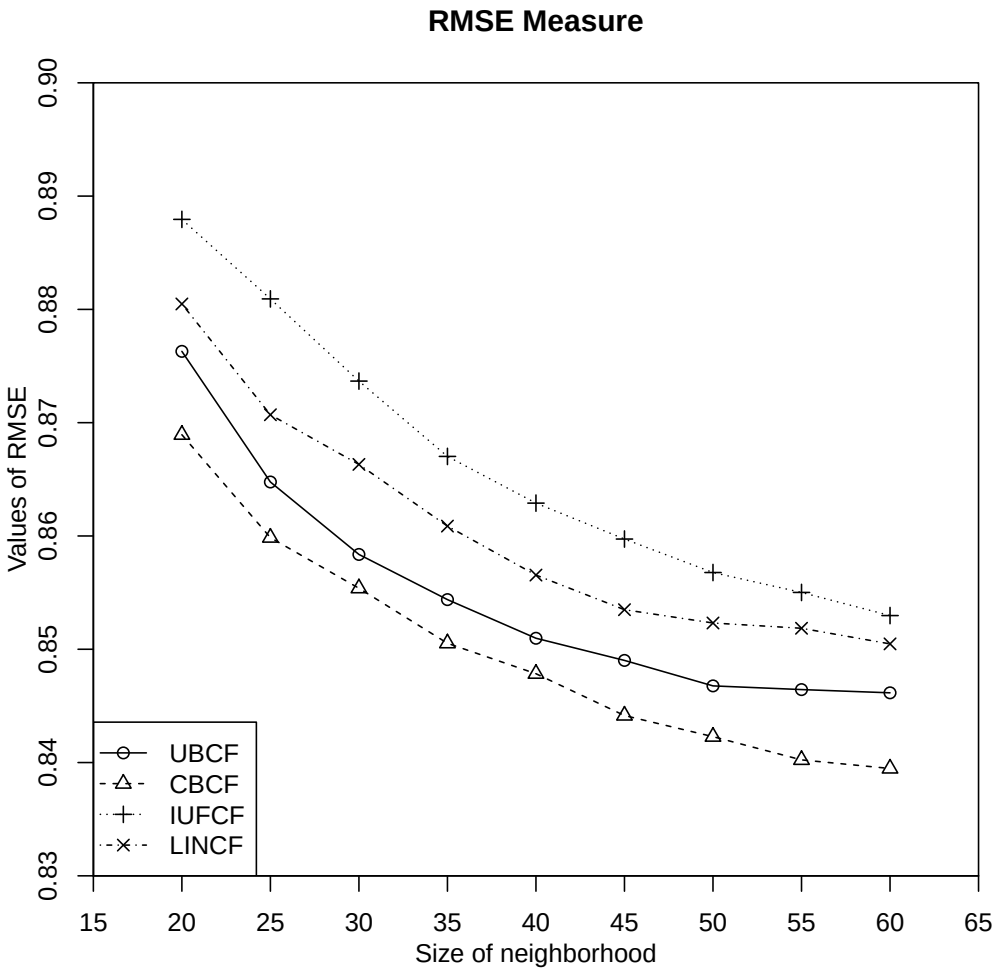

FIGURE 4.8: Result of RMSE measure on the Netflix dataset

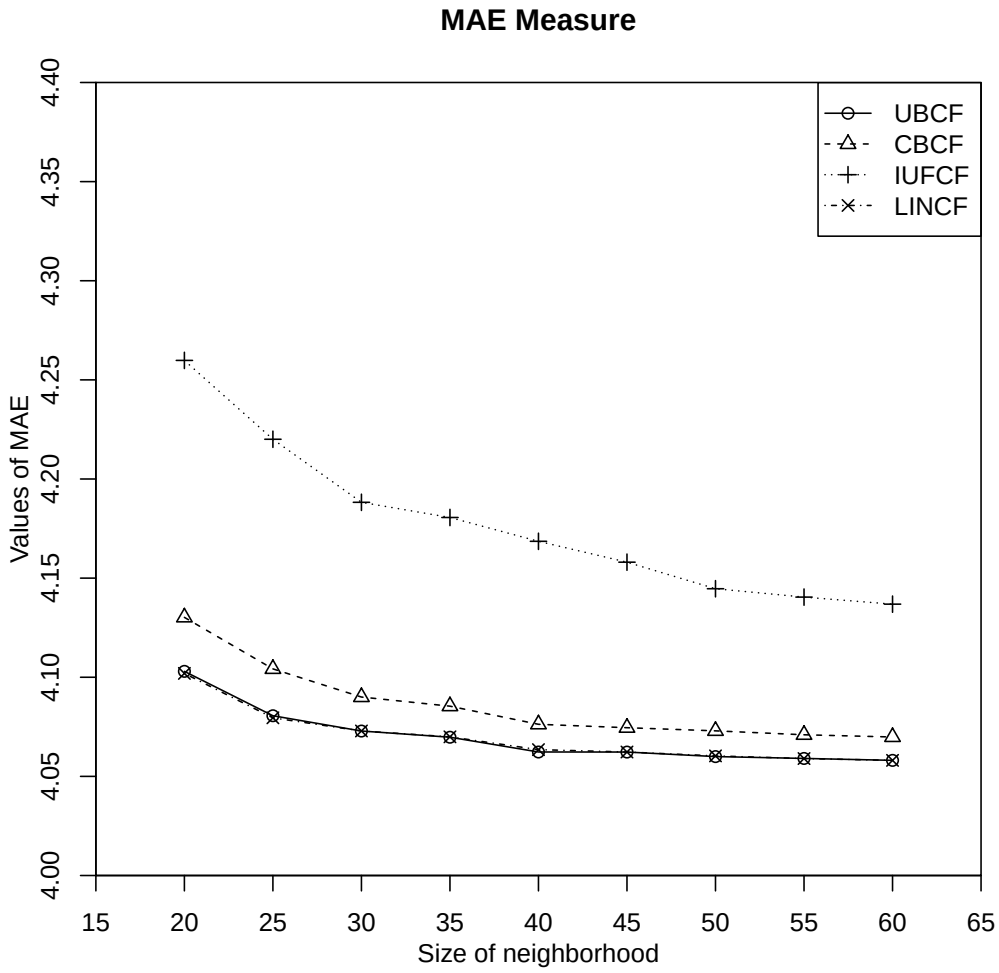

FIGURE 4.9: Result of MAE measure on the Jester dataset

accuracy of the UBCF approach. The MovieLens and Netflix datasets are sparse, but the Jester dataset is not sparse, indicating that in sparse datasets, the accuracy of the CBCF approach outperforms that of the other approaches. In non-sparse datasets, the CBCF cannot improve the accuracy of the traditional UBCF approach.

Figures 4.11, 4.12, and 4.13 show the results for coverage on the MovieLens, Netflix, and Jester datasets, respectively. As shown in the figures, the values of the coverage metrics for all approaches increase as the neighborhood size increases. Furthermore, the coverage of CBCF is significantly higher than that of the other approaches, especially for the MovieLens and Netflix datasets. This shows that CBCF can recommend more types of items that a new user has not rated yet. Therefore, we can conclude that CBCF improves the coverage of traditional UBCF and outperforms the other approaches for both sparse and non-sparse datasets.

Figures 4.14, 4.15, and 4.16 show the results for MP on the MovieLens, Netflix, and Jester datasets, respectively. As shown in the figures, in both the MovieLens and Netflix datasets, the MP values increase as the neighborhood size increases. The IUFCF and LINCF approaches have improved the MP of the traditional UBCF slightly; on the other hand, CBCF greatly increases the MP. In contrast, for the Jester

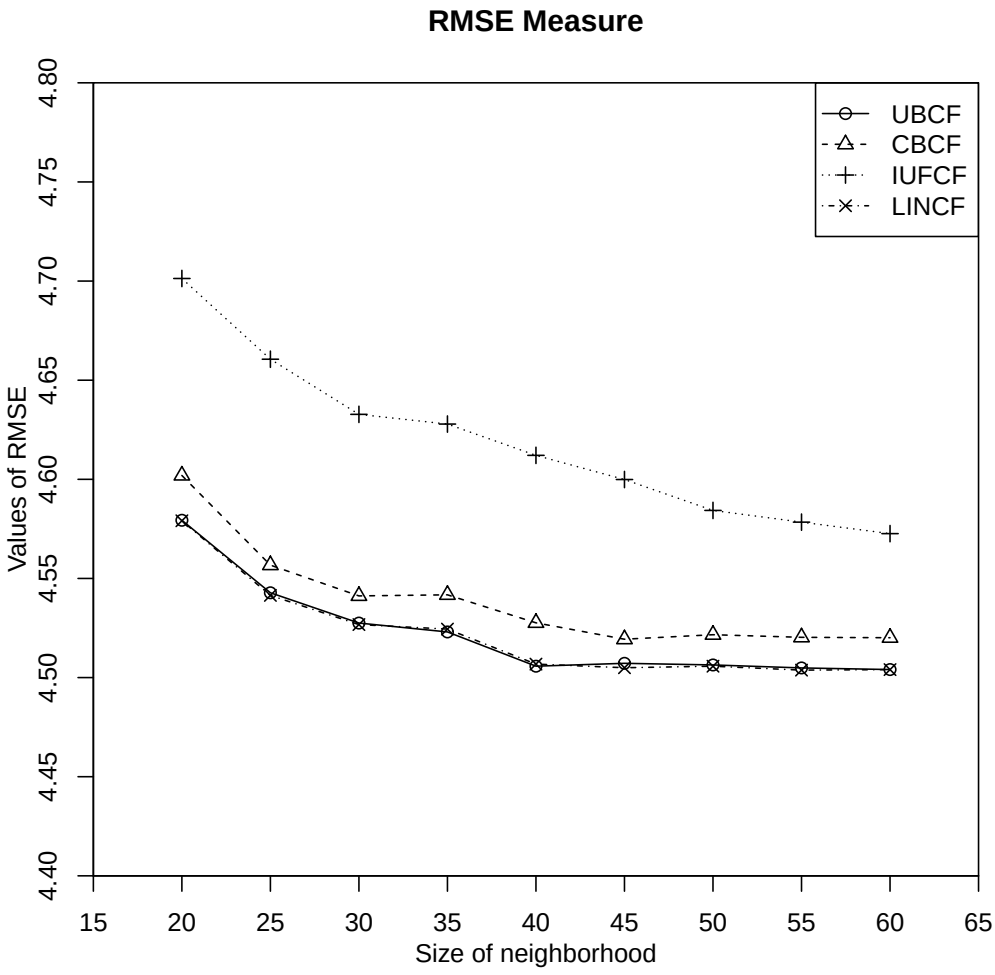

FIGURE 4.10: Result of RMSE measure on the Jester dataset

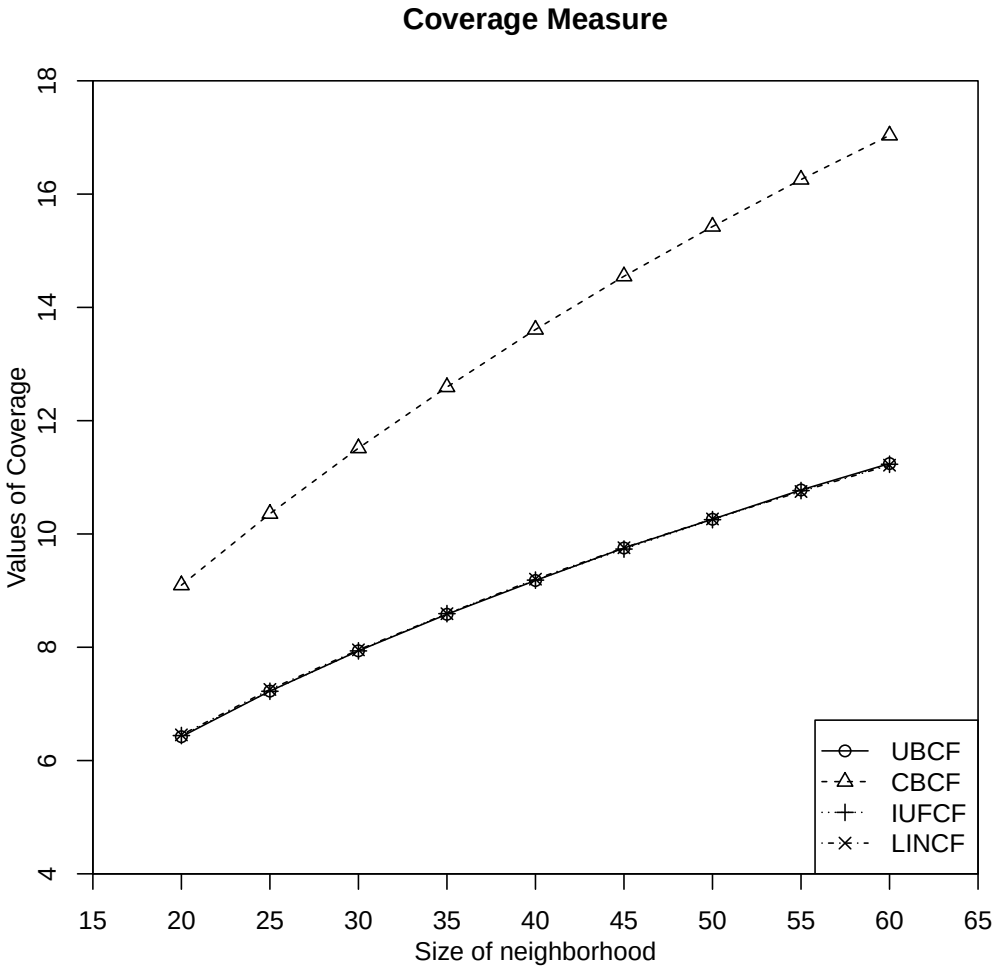

FIGURE 4.11: Result of coverage measure on the MovieLens dataset

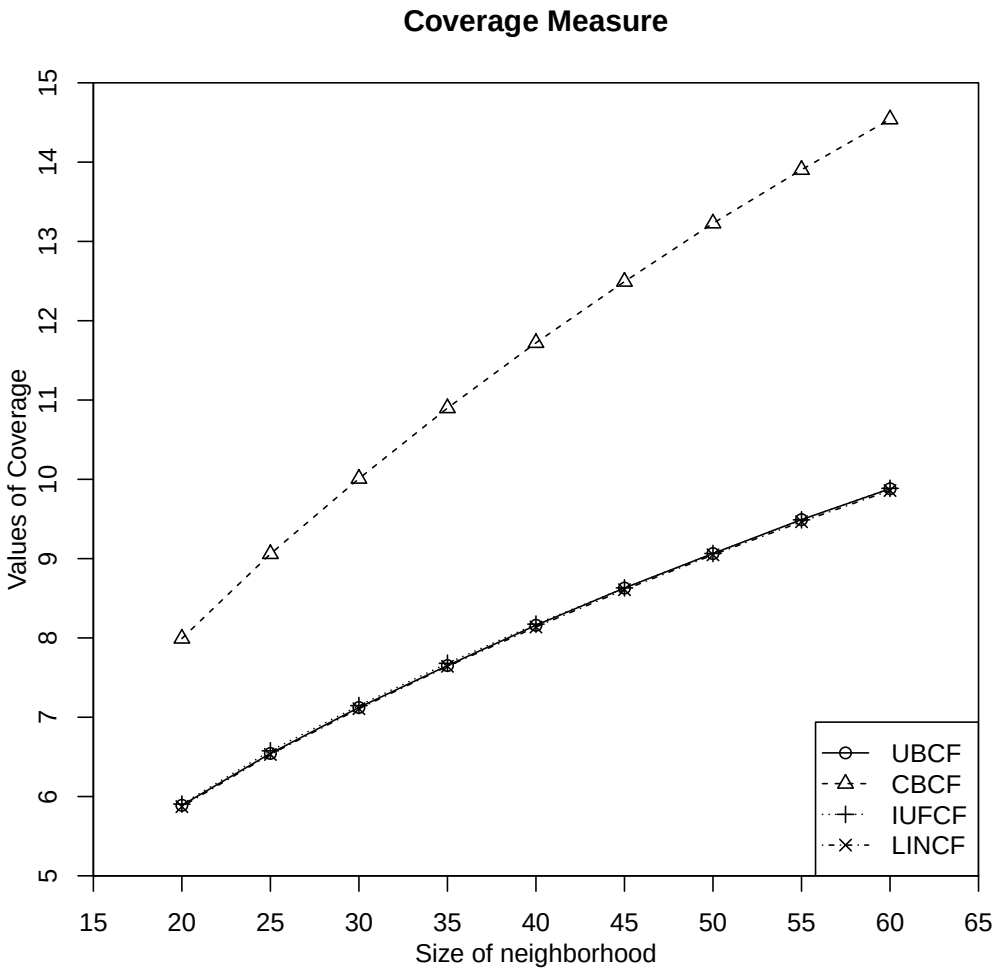

FIGURE 4.12: Result of coverage measure on the Netflix dataset

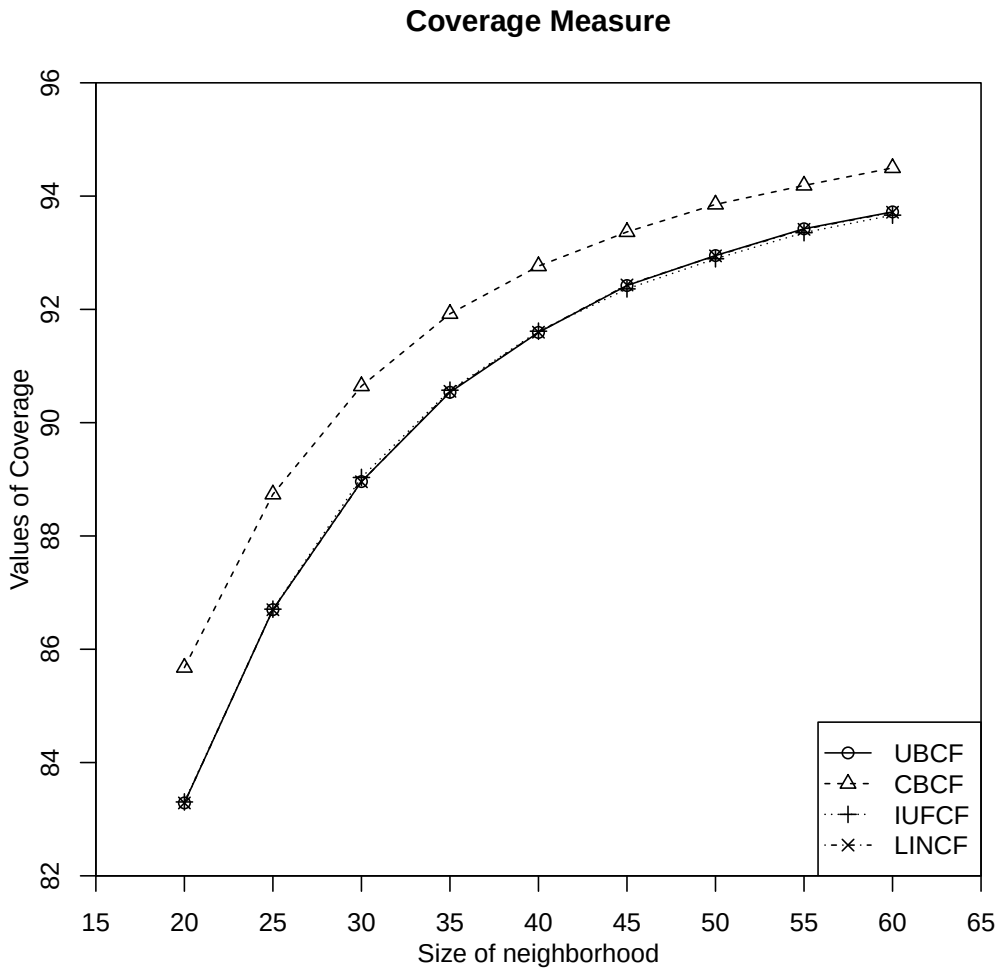

FIGURE 4.13: Result of coverage measure on the Jester dataset

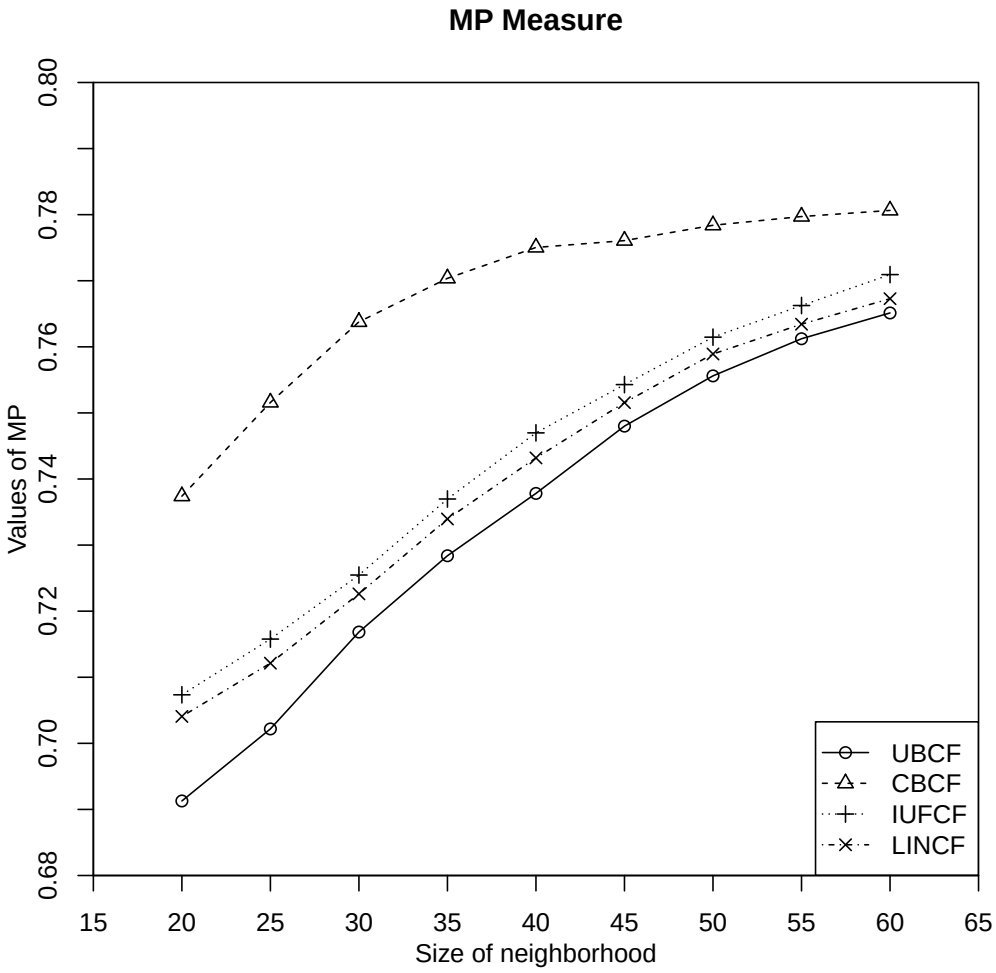

FIGURE 4.14: Result of MP measure on the MovieLens dataset

dataset, the MP decreases as the neighborhood size increases, and the MP of traditional UBCF has the highest values. This indicates that CBCF cannot provide more diverse recommendations than UBCF. From the above results, we can conclude that CBCF outperforms the other approaches on sparse datasets; nevertheless, it cannot improve the diversity of the traditional UBCF approach on non-sparse datasets.

Figures 4.17, 4.18, and 4.19 show the results for MN on the MovieLens, Netflix, and Jester datasets respectively. As shown in the figures, for both the MovieLens and Netflix datasets, the MN values for UBCF, IUFCF, and LINCF are nearly the same; however, CBCF obviously has higher MN values than the other approaches, showing that CBCF can improve MN significantly. On the other hand, for the Jester dataset, although the MN of CBCF is higher than that of UBCF at first, it decreases faster as the neighborhood size increases, indicating that CBCF cannot provide recommendations with higher diversity than traditional UBCF. Therefore, we can conclude that in sparse datasets, CBCF can improve diversity more efficiently than the other approaches; however, it has poor performance for diversity as the neighborhood size increases in non-sparse datasets.

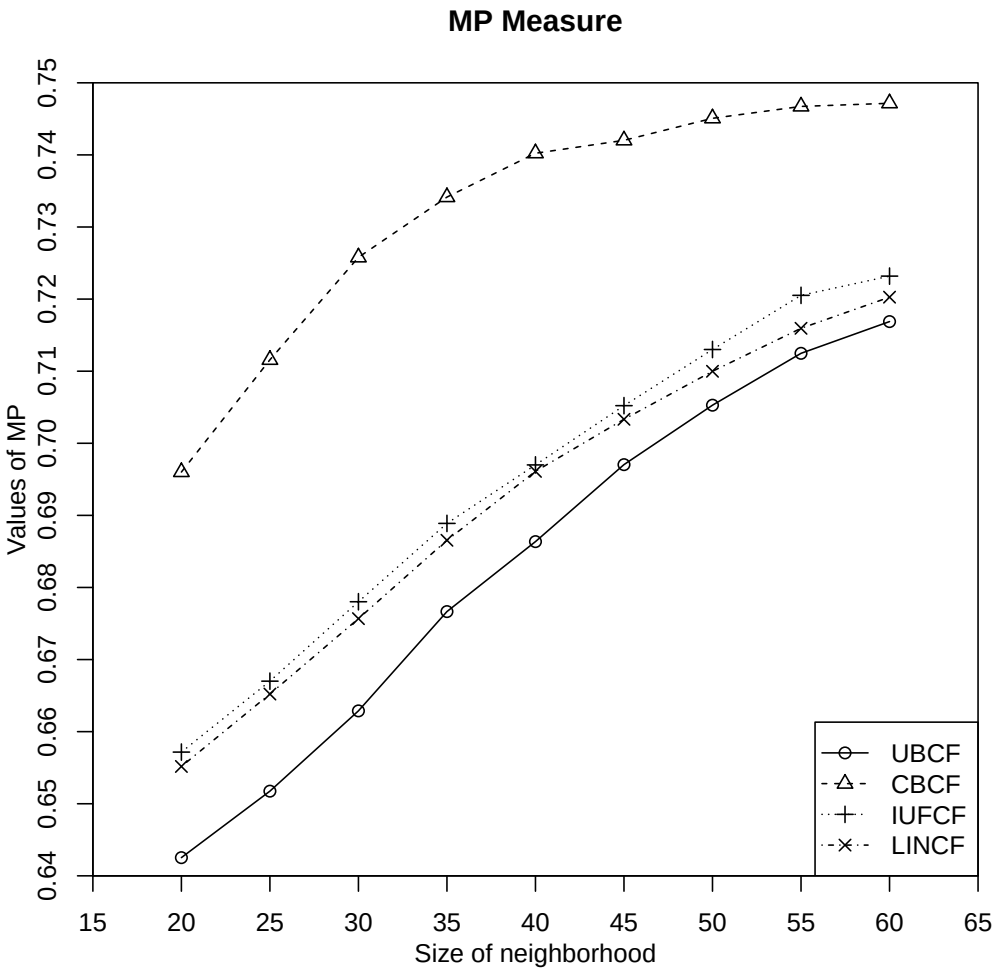

FIGURE 4.15: Result of MP measure on the Netflix dataset

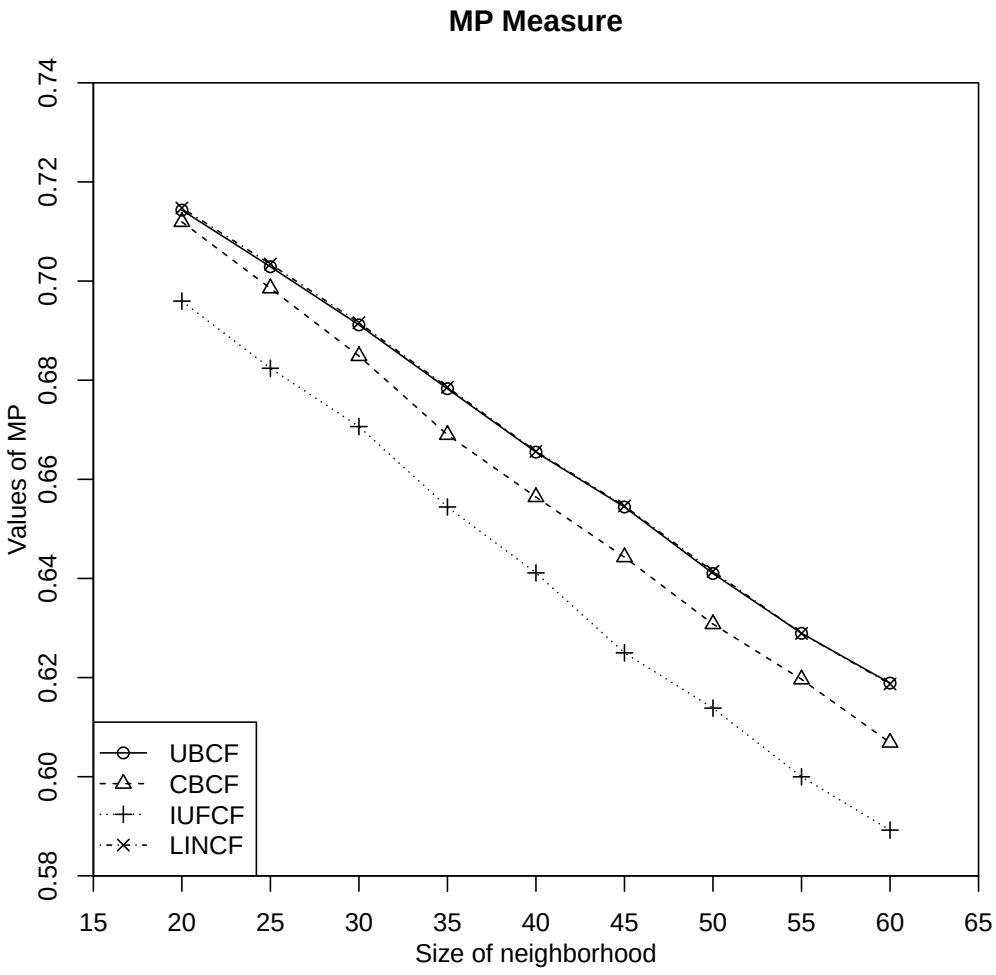

FIGURE 4.16: Result of MP measure on the Jester dataset

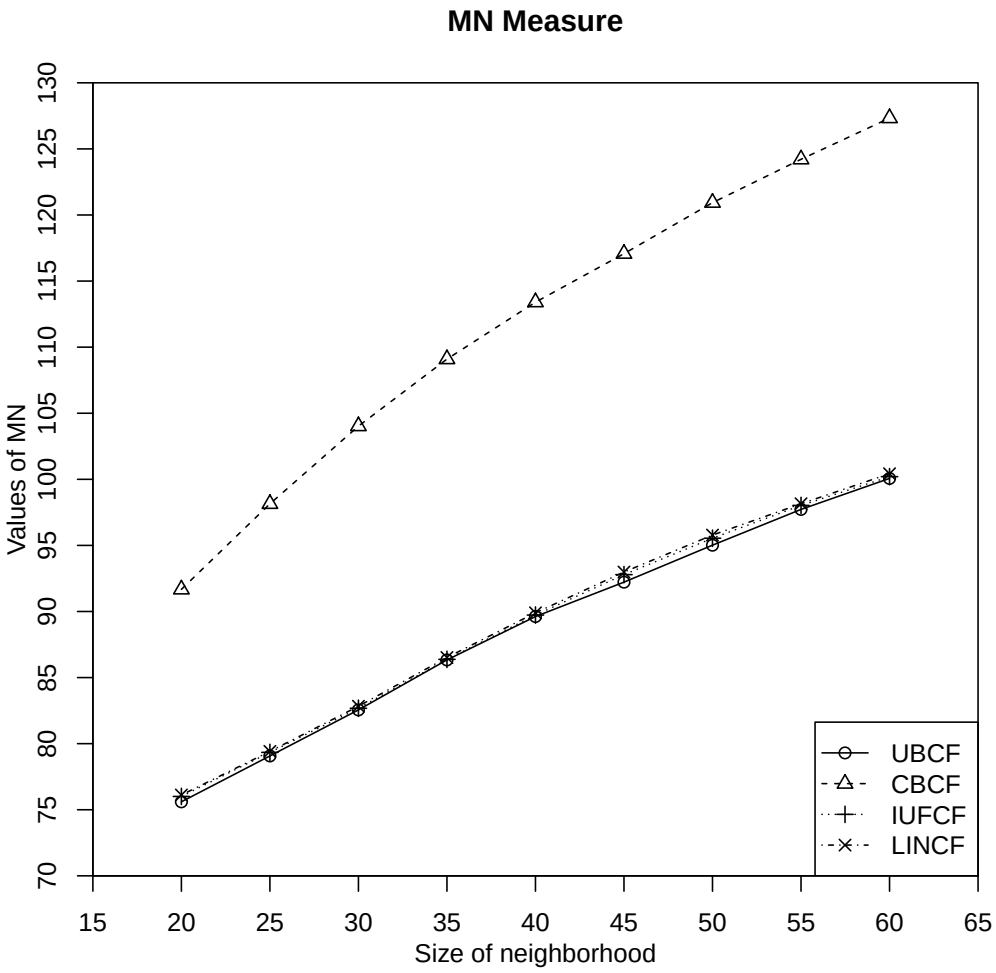

FIGURE 4.17: Result of MN measure on the MovieLens dataset

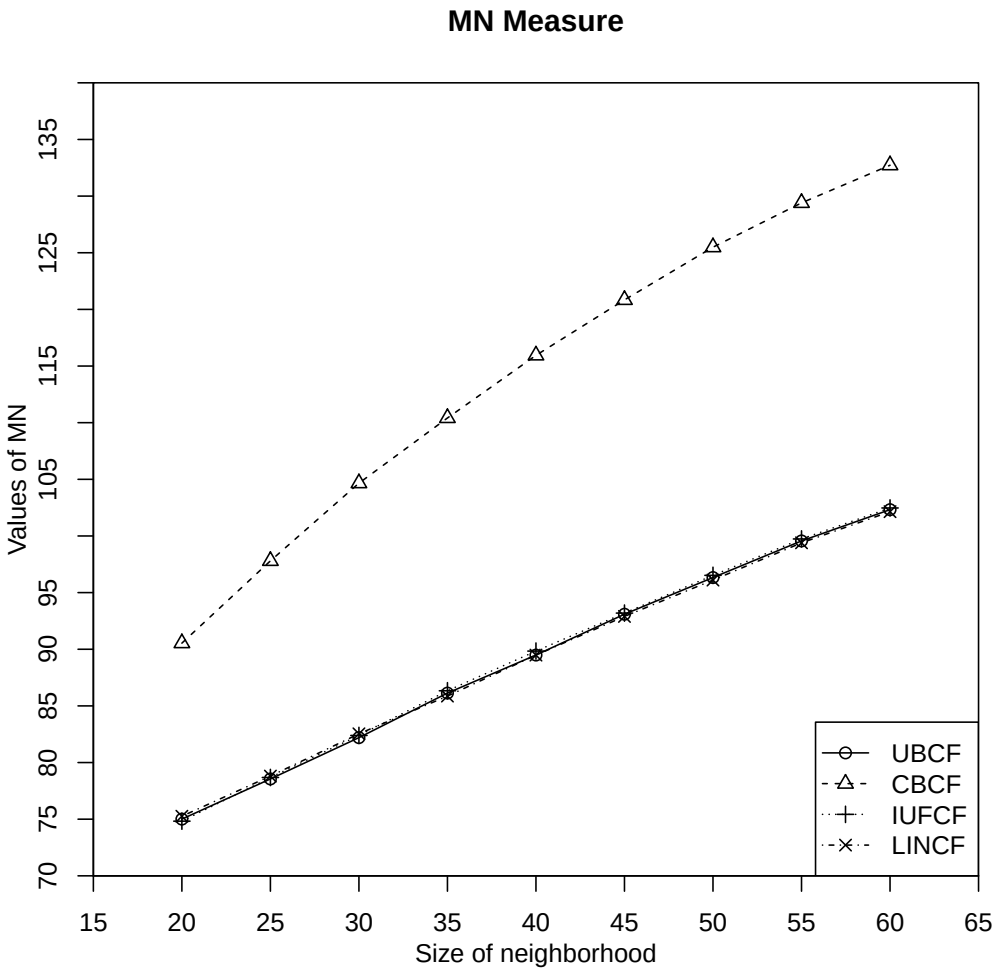

FIGURE 4.18: Result of MN measure on the Netflix dataset

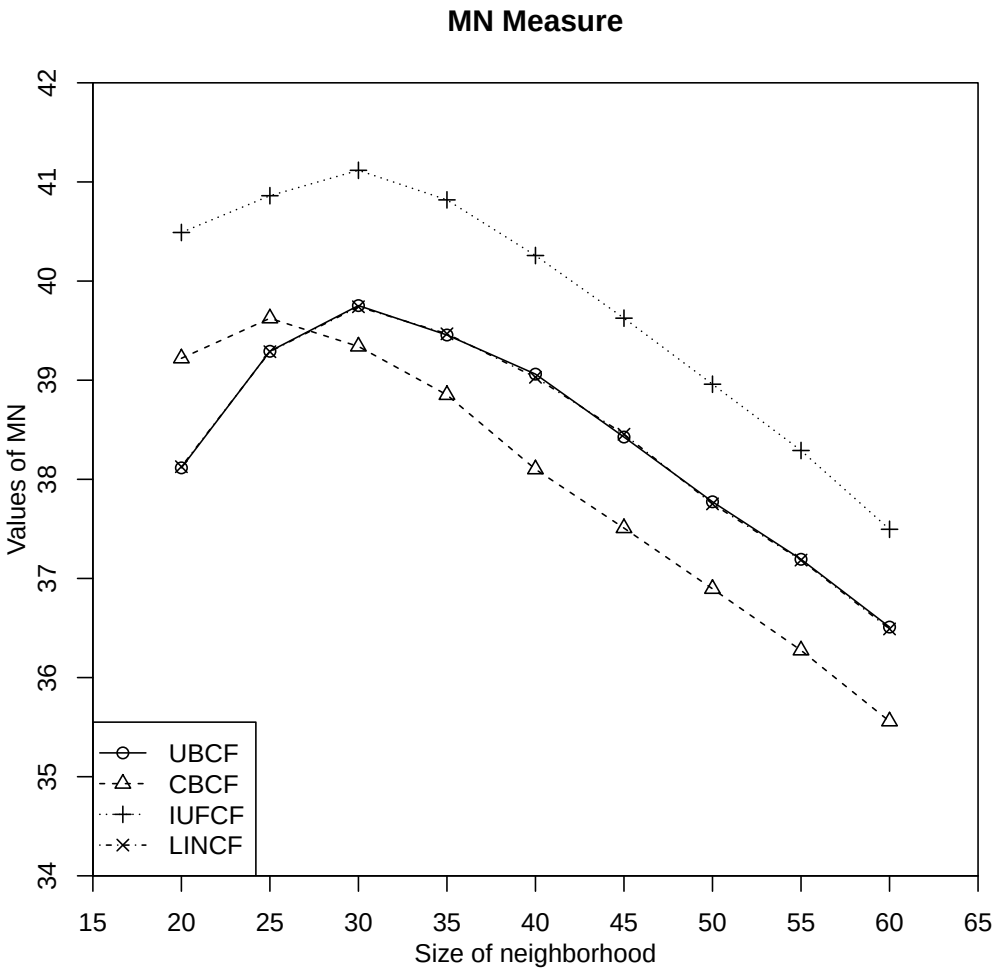

FIGURE 4.19: Result of MN measure on the Jester dataset

4.5.3 Analysis and discussion

Our experimental results indicate that the improved CBCF approach shows different performances with different datasets. For the MovieLens and Netflix datasets, there are huge numbers of items; however, for each user, the number of rated items is substantially smaller than the number of unrated items. Therefore, the two datasets are very sparse. Under these circumstances, because of users' different behaviors, there might exist some users whose ratings concentrate only on popular items (users whom we have defined as redundant users). As we have confirmed that a new user's ratings also concentrate on popular items, the similarity between redundant users and the new user might be very high, and the neighborhood of the new user might consist almost entirely of redundant users, causing recommendations from these redundant users to concentrate on fewer types of items, perhaps even only popular items. The improved CBCF approach used the covering reduction algorithm to remove as many as these redundant users as possible. After user reduction, the neighborhood of a new user in the improved CBCF approach consist of users who have rated diverse items, with the result that the diversity of improved CBCF greatly outperforms that of the other existing approaches. In addition, because ratings of redundant users concentrate on popular items, they have no ability to make accurate predictions for niche items, resulting in lower recommendation accuracy in the traditional UBCF approach. In the improved CBCF approach, redundant users in a neighborhood are mostly removed, with the result that a neighborhood in the improved CBCF approach can make predictions for many types of items rather than only popular items, thereby also improving accuracy.

In contrast, for the Jester dataset, the total number of items is 100, and each user has rated at least 36 items. Hence, each user has rated sufficiently many items relative to all 100 items, and each item can be considered as a popular item. Therefore, this dataset is not sparse. In this case, in the improved CBCF approach, each user can be considered as a redundant user, with the result that reduction loses its significance. Recommendations from improved CBCF might also concentrate on popular items, with the result that the diversity of the improved CBCF approach are inferior to those of traditional UBCF. However, because user reduction can select users who have rated more types of items, the coverage of improved CBCF is still higher than that of the other approaches. In addition, co-rated items between two users are sufficient, so neighbors with higher similarity can ensure the prediction of more accurate ratings; however, some neighbors with higher similarity might be considered as redundant users to be removed, with the result that the accuracy does not improve but decreases.

Generally, in practical applications, RSs must handle large data that include huge numbers of users and items. Thus, for each user, only a small number of items have been rated compared with the huge number of unrated items. Therefore, most RSs have sparse datasets, such as the MovieLens and Netflix datasets. However, for a sparse dataset, the improved CBCF approach can remove redundant users to create more appropriate neighborhoods than the UBCF approach and provide recommendations for a new user with more satisfactory accuracy and diversity values than in existing work. Thus, the improved CBCF approach is significant for RSs.

4.6 Summary

In this chapter, we have improved CBCF to achieve personalized recommendations for a new user. The improved CBCF approach reconstructs the decision class to account for the set of niche items and uses covering reduction in covering-based rough sets to remove redundant users from candidate neighbors. By removing redundant users who have high similarity with the new user but can make predictions for only a few types of items, improved CBCF makes great improvements in both the accuracy and diversity metrics while utilizing only the user-item rating matrix with no other special information. Our experiments also show superiority of our approach by comparing it with traditional UBCF and other existing work. Although the improved CBCF is inferior to the traditional UBCF in non-sparse datasets (e.g., the Jester dataset), it greatly outperforms other relevant work in sparse datasets (e.g., the MovieLens and Netflix datasets), which occur more often in the real world. Therefore, our approach could be applied to provide satisfactory recommendations for a new user in real world RSs.

Chapter 5

Item-variance weighting for IBCF by using time-related correlation degree and covering degree

5.1 Introduction

RSs estimate ratings of items that are not yet to be consumed by users, and provide items with the best predicted ratings for customers, so the accuracy is a key issue, good or bad of accuracy directly affects service quality of RSs and user experience (Bobadilla et al., 2013). CF approaches are popularly used in RSs due to the satisfactory performances (Symeonidis et al., 2008). With the assumption that an item will be preferred by a user if this item is similar with the preference of this user in the past, IBCF has been proposed and has achieved success both in research and practice in recent years, such as Amazon (Sarwar et al., 2001; Linden, Smith, and York, 2003; Li et al., 2016; Li et al., 2014). In the traditional IBCF approach, all items carry the same weight when computing the similarity and prediction; however, it is widely recognized that some items are more important than others and should be given relatively higher weighting (Karypis, 2001). So the traditional IBCF often can not provide recommendations with satisfactory accuracy.

In this chapter, we reconsider the item weight in IBCF approach. We apply the time weight to the item-item similarity computation to improve the predictive accuracy, ratings which rated by the same user in closer time will have higher weight when computing the similarity. On the other hand, we insert the covering degree to the rating prediction to increase the classification accuracy, items which are closer with user's preference will have higher covering degree, and will have high weight when make prediction. Experiments prove that our proposed approach supply better performance than traditional IBCF and other existing work, and provide recommendations with satisfactory accuracy.

The remainder of this chapter is organized as follows. In Section 5.2, we introduce the traditional IBCF approach and related work. In Section 5.3, we analysis the traditional IBCF and give the problem setting. In Section 5.4, we describe the time-related correlation degree and covering degree, and apply them to the traditional IBCF approach. In Section 5.5, we present our experiments and compare our results with the traditional IBCF approach and other work. Finally, in Section 5.6, we show the summary of this work.

5.2 Related works

IBCF was first presented by Sarwar (Sarwar et al., 2001), which computes similarity between two items by comparing users' ratings on them. IBCF can dramatically improve the scalability of CF and can be applied to the huge numbers of items and users that are typical of modern RSs. IBCF can easily handle large data sets and produce better predictions than UBCF. Also in contrast to UBCF, IBCF is able to compute item-item similarity off-line, both saving on-line time and making more effective recommendations (Papagelis and Plexousakis, 2005). Up until now, IBCF has been widely used in many applications in the real world, such as at Amazon.com. The detailed information and procedures could be found in Subsection 1.2.2.

Currently, many researchers have applied item-variance weighting to the traditional IBCF (Kefalas and Manolopoulos, 2017; Frémal and Lecron, 2017; Esparza, OMahony, and Smyth, 2011; Hu and Ester, 2013; Koren, 2010; Lee, Park, and Park, 2008; Liu et al., 2014b). Ding and Li (Ding and Li, 2005) used clustering to discriminate between different kinds of items, and presented a novel algorithm to compute the time weights for different items in a manner that will assign a decreasing weight to old data. To each item cluster, they trace each user's purchase interest change and introduce a personalized decay factor according to the user own purchase behavior. Their new algorithm can substantially improve the precision of IBCF without introducing higher order computational complexity. However, this approach only considered the time weight to predict ratings, moreover, it did not consider other item weight that may make important effects on the performance of RS.

5.3 Analysis and problem setting

IBCF has achieved success both in research and practice. Similarity computation and rating prediction are two main procedures in IBCF. However, in traditional IBCF approaches, all items carry the same weight in both item-item similarity computation and rating prediction. Generally, in our real life, it is widely recognized that some items are more important than others and should be given relatively higher weighting. For example, as shown in Table 5.1, item 1, item 2, and item 3 are rated respectively by users: U_1, U_2, U_3 for the same scores. If computing the similarity by the traditional IBCF, similarity between item 1 and item 2 will be equal to the similarity between item 1 and item 3. However, the rated time is different, because some people's interests change with time quickly, an item that was rated recently by a same user should have a bigger impact than the item that was rated during a long time interval. Therefore, item 1 should be more similar with item 2 than item 3. Through the analysis of the above, we can find that, we should consider the item-variance weighting in the traditional IBCF.

TABLE 5.1: Example of user-item rating matrix RM

	Item 1		Item 2		Item 3	
	Score	Rated time	Score	Rated time	Score	Rated time
U_1	3	2016-7	3	2016-7	3	2015-7
U_2	4	2016-7	4	2016-7	4	2015-7
U_3	5	2016-7	5	2016-7	5	2015-7

5.4 Time-related correlation degree and covering degree for the traditional IBCF

5.4.1 Motivation of proposed approach

Our novel approach increases the weights of items that make a more significant contribution to the process of similarity computation and rating prediction, to extract more precise and satisfactory recommendations from the RS.

In the IBCF approach, computation of the item-item similarity and rating prediction are separate procedures (Goldberg et al., 1992). In current versions of IBCF, however, the two procedures give the same weights to all items. In reality, a person's preferences may change over time, and a different rating score may be given by the same user to the same item. When computing item-item similarity, therefore, a time factor should be included. In addition, items with lower similarity to the target item may nevertheless make a significant contribution to the prediction, so weightings should be given to the items used to make rating prediction.

5.4.2 Time-related correlation degree and covering degree

People are most likely to be interested in an item that they have evaluated recently. So, for the same user, the score given to an item that was rated at approximately the same time as the target item will make greater contribution to the similarity computation. A person's memory can be represented as a linear curve, first changing fast, then more slowly. To express the degree of correlation between two items over time, Ding and Li (Ding and Li, 2005) proposed a time function as follows:

$$f(t) = e^{-\lambda * t}, \tag{5.1}$$

where, $\lambda = \frac{1}{T_0}$ is the decay rate, if $T_0 = 30$ days, the time weight reduces by month. Here, based on the time function above, we proposed the following time-related correlation degree:

$$f(I_u(x, y)) = e^{-\lambda * |t_{u,x} - t_{u,y}|}. \tag{5.2}$$

Here, $f(I_u(x, y))$ is a gradually decreasing function tracking the degree of correlation between item x and item y for a target user u, and $t_{u,x}$ is the time at which item x was rated by user u. From the function $f(I_u(x, y))$, as the times at which two items were rated by the target user become closer and the value of $|t_{u,x} - t_{u,y}|$ becomes smaller, the degree of correlation between the two items increases. The time-related correlation degree function can therefore effectively estimate the relevance of an item and increase the weight of items that were rated closer in time to the target item.

It is well known that a person's current interests are strongly correlated with previous preferences. If an item has characteristics that are close to the target user's known previous interests, this item should be given a greater weight when making predictions. Here, we propose the covering degree function as follows:

Let $\langle T, C \rangle$ be a covering approximation space. For a set $X \subseteq T$ and $K \in C$, we define the covering degree as

$$CD(K, X) = \begin{cases} \frac{card(K \cap X)}{card(K)} & \text{if} \quad K \neq \emptyset, \\ 0, & \text{if} \quad otherwise. \end{cases} \tag{5.3}$$

It is clear that $CD(K, X) \in [0, 1]$, and the value $CD(K, X)$ can be interpreted as the degree to which the element K of covering C is included in the set X. In

RSs, an item's neighborhoods are most relevant to the item itself and could express the common characteristics of the item. If the target user's preference set is treated as X, then the item's neighborhood set can be considered to be K. If an item has a higher covering degree, $CD(K, X)$ has a greater value, and the item should be given a greater weight when predicting the rating score for the target item.

5.4.3 Procedures of proposed approach

In our proposed approach, time-related correlation degree and covering degree are applied to compute item-item similarity and rating prediction respectively. Furthermore, θ is set as the threshold for rating score, and items with $r_{u,x} \geq \theta$ are defined as items relevant to user u. An target user tu does not need to input any special information, then the new approach could output recommended items Rec.

$Step$ 1: Item-item similarity computation with time weight. According to the user-item rating matrix RM, we insert the time-related correlation degree into Pearson correlation coefficient to compute the item-item similarity. Here, we have

$$sim(x,y) = \frac{\sum_{u \in U_x \cap U_y} (r_{u,x} - \bar{ri}_x) * (r_{u,y} - \bar{ri}_y) * f(I_u(x,y))}{\sqrt{\sum_{u \in U_x \cap U_y} (r_{u,x} - \bar{ri}_x)^2} \sqrt{\sum_{u \in U_x \cap U_y} (r_{u,y} - \bar{ri}_y)^2}}. \tag{5.4}$$

$Step$ 2: Neighborhood selection. From the similarity list of item $i \in I_{tu}^c$, we select top k items as item i's neighborhood $N_i(k)$. In items domain I, for each similar item $j \in N_i(k)$ of the item i, we further select top q items from the similarity list of item j, which comprise the item set C_j. Let $C* = I - \cup C_j$. $C = \{C_1, C_2, ..., C_k, C*\}$ is then a covering in domain I.

$Step$ 3: Rating prediction. In the domain I, relevant items of each target user tu will comprise the relevant set R_{tu}, where

$$R_{tu} = \{p \in I | r_{tu,p} \geq \theta\}. \tag{5.5}$$

Then based on the covering degree, in the neighborhood $N_i(k)$ of the item i, for each item $j \in N_i(k)$, we compute the covering degree between each item set C_j and R_{tu}, and apply it to weighted sum approach to predict the rating score of the item $i \in I_{tu}^c$ from the target user tu, here

$$p_{tu,i} = \frac{\sum_{j \in N_i(k) \cap I_{tu}} sim(i,j) * r_{tu,j} * CD(C_j, R_{tu})}{\sum_{j \in N_i(k) \cap I_{tu}} |sim(i,j) * CD(C_j, R_{tu})|}. \tag{5.6}$$

$Step$ 4: Item recommendations. When all predictions are completed, the top N items in the prediction list are selected as the recommended items.

5.5 Experiments and evaluations

In this section, we describe the evaluation dataset and metrics, examine the performance of the new approach, and compare it with the traditional IBCF approach.

5.5.1 Experimental setup and evaluation metrics

In our experiments we used the MovieLens 100K dataset (Herlocker et al., 1999), which is often used to evaluate RSs. In our study, movies rated above 3 were treated as relevant to that user. For every user, 20% of the rated items were treated as test items and the remaining 80% as training items. Our experiment predicted a rating

Algorithm 5.1 Proposed approach

Input: User-Item rating matrix RM with rating time and an target user tu.
Output: Recommended items set of size N for the target user tu.
SL_i^t : Similarity list of item i with time weight.
$N_i(k)$: Neighborhood of the item i.
R_{tu} : Relevant items of the target user tu.
k : Number of items in the neighborhood $N_i(k)$ of the item i.
N : Number of items recommended to the target user tu.
I_{tu}^c : Items which have not yet rated by the target user tu.
$p_{tu,i}^{cd}$: Rating prediction of item i for the target user tu with covering degree.

1: Apply the time weight to similarity measure, and compute similarity between each item in I;
2: **for** each item $i \in I_{tu}^c$ **do**
3: Find the k most similar items in SL_i^t of item i to comprise neighborhood $N_i(k)$;
4: **for** each item $j \in N_i(k)$ **do**
5: The neighborhood of item j comprise the item set C_j. Use the covering degree function to compute $CD(C_j, R_{tu})$.
6: **end for**
7: Apply the covering degree to the prediction function. Predict rating score $p_{tu,i}^{cd}$ for item $i \in I_{tu}^c$ by the ratings of $N_i(k)$ from the target user tu;
8: **end for**
9: Recommend to the target user tu the top N items having the highest predicted rating scores.

score for each test item based on the training items. As the MovieLens dataset was collected over a seven-month period, we were able to use the month of rating to test our time-based correlation degree function.

The mean absolute error (MAE), root mean square error (RMSE), precision, recall, and F1 were used as evaluation metrics, all of which are widely used for evaluating RSs. MAE and RMSE, which compare the numerical prediction values against the original user ratings, are the measures most commonly used for evaluating the accuracy of a recommender method. We have given the information of MAE and RMSE in Subsection 3.5.1, here we present the definition of precision, recall and F1. Let Rec_{tu} as the set of N recommendations to the target user tu. Precision is the proportion of recommended items that the target user actually liked in recommendations. This measure is as high as possible for good performance.

$$Precision = \frac{1}{|U|} \sum_{tu \in U} \frac{\#\{i \in Rec_{tu} | r_{tu,i} \geq \theta\}}{N}. \tag{5.7}$$

Recall indicates the proportion of relevant recommended items from the number of relevant items. This measure should be as high as possible for good performance. Hence, the recall is computed as follows:

$$Recall = \frac{1}{|U|} \sum_{tu \in U} \frac{\#\{i \in Rec_{tu} | r_{tu,i} \geq \theta\}}{R_{tu}}. \tag{5.8}$$

F1 is a combination of precision and recall.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{5.9}$$

Precision, recall, and F1 indicate how well the RS discovers items that are desirable to the target user. In calculating the precision, recall, and F1 values, the number of recommendations was set to 2, 4, 6, 8, 10, and 12.

5.5.2 Experimental results and comparisons

We named the IBCF approach incorporating time-related correlation degree and covering degree TCIBCF. To test the performance of our new approach, we compared its results with those from the standard IBCF. Additional, we also made comparison with popular approach TWIBCF presented by Ding (Ding and Li, 2005). In the experimental testing of the IBCF approach, the Pearson correlation coefficient was used as the similarity measure, and the weighted sum was used to predict the rating score. To evaluate the TCIBCF approach, we then added the time-based correlation degree and covering degree to the similarity computation and rating prediction.

Table 5.2 shows the results of the different metrics applied to TCIBCF, TWIBCF, and IBCF. To set the neighborhood size, we selected different numbers from the similarity list as items similar to the target item. In the TCIBCF approach, the size of the similar items k was equal to the size of each similar item's neighborhood q. It can be observed from Table 5.2 that, using the TCIBCF approach, the MAE and RMSE values became smaller as the neighborhood size increased. As lower values of MAE and RMSE indicate better accuracy, the accuracy of TCIBCF improved as the neighborhood size increased. In contrast, using the IBCF approach the values of MAE and RMSE did not decrease linearly as the neighborhood size increased. The best MAE and RMSE values of 1.192 and 1.390 were recorded when the size of neighborhood was 20, and the worst values of 1.217 and 1.424, when the size of neighborhood was 10 and 40 respectively. For TWIBCF approach, although the values of MAE and RMSE were smaller than IBCF approach, they were also bigger than TCIBCF approach. Beside that, same as IBCF approach, values of MAE and RMSE for TWIBCF approach also decreased nonlinearly as the neighborhood size increased. The best MAE and RMSE values of 1.181 and 1.339 were recorded when the size of neighborhood was 20, and the worst values of 1.206 and 1.396, when the size of neighborhood was 10 and 50 respectively. Overall, the proposed TCIBCF approach outperformed the traditional IBCF and TWIBCF approaches in terms of MAE and RMSE metrics.

Figures 5.1, 5.2, and 5.3 show the precision, recall, and F1 measures for $TCIBCF_{k=50}$, $TWIBCF_{k=50}$, and $IBCF_{k=50}$. As can be seen, in the precision measure, the TCIBCF approach was stable, and the TWIBCF and IBCF had a small fluctuation as the number of recommendations increased; however the proposed TCIBCF approach had higher precision values than TWIBCF and IBCF approaches. For recall and F1 measures, all the TCIBCF, TWIBCF, and IBCF approaches increased significantly as the number of recommendations increased. Furthermore, at first, recall and F1 of TCIBCF were lower than TWIBCF and IBCF when the number of recommendations was 2, however as the number of recommendations increased, TCIBCF had faster improvements, and had much better values than TWIBCF and IBCF approaches. Overall, the proposed TCIBCF approach could make better recommendations than the traditional IBCF and TWIBCF approaches in terms of precision, recall, and F1 metrics.

5.5.3 Analysis and discussion

The most significant innovation in our approach is the ability to weight items that make a greater contribution to the similarity computation and rating prediction. As

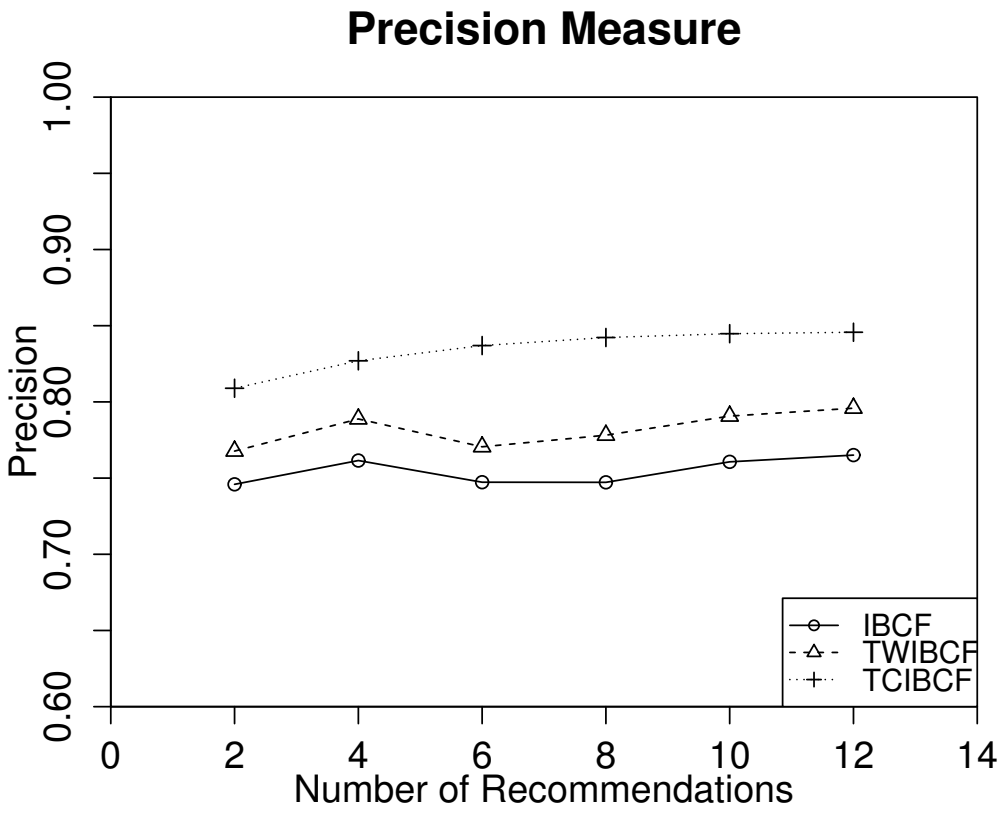

FIGURE 5.1: Precision of the TCIBCF, TWIBCF, and IBCF against the number of recommendations

the key items then play a more significant role in the RSs, the recommendations generated are closer to the real tastes of the target user.

Experimental and comparative results suggest that our new TCIBCF approach can achieve better accuracy than the IBCF and TWIBCF approaches, as the predicted scores which the TCIBCF generated more closely matched the original scores. The TCIBCF also achieved better precision, recall, and F1 values than the IBCF and TWIBCF approaches. This suggests that a target user will agree with more recommendations made by the proposed TCIBCF than with those produced by IBCF and TWIBCF approaches.

In our approach, items are given different weights in the item-item similarity computation. For example, take two items x and y that have the same rating scores from the target user tu, so that $t_{tu,x} = t_{tu,y}$. The time at which target user tu rated item i is $t_{tu,i}$. This is closer than the rating time of item y, so that $|t_{tu,i} - t_{tu,x}| < |t_{tu,i} - t_{tu,y}|$. In a traditional IBCF, items x and y will be given the same weight when computing item-item similarity, because they received the same rating score from the target user. However, the preferences of target user tu may have changed, so that the rating of item x will have a greater influence than that of item y. In our approach, $f(I_{tu}(i,x)) > f(I_{tu}(i,y))$, so item x will carry a higher weight than item y when computing item-item similarity. After the similarity has been computed, items similar to the target item were selected to predict the rating score. Here, we used the covering degree function to compute the weight of each similar item. Item set C_j

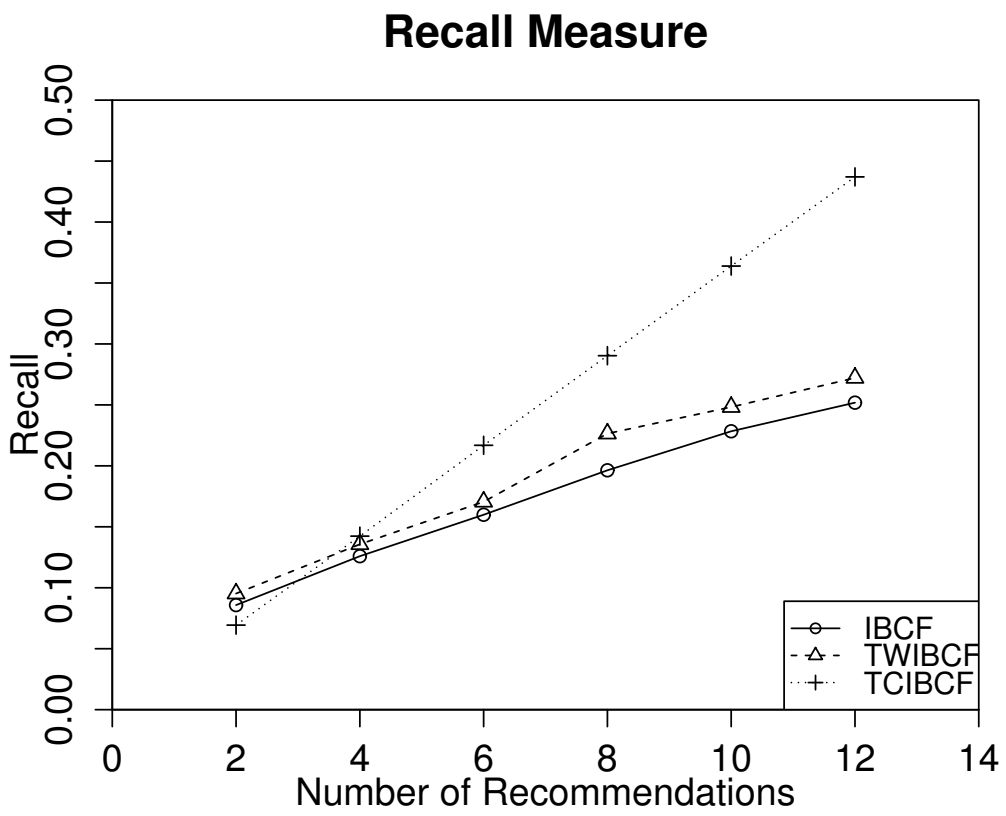

FIGURE 5.2: Recall for the TCIBCF, TWIBCF, and IBCF against the number of recommendations

captures the characteristics of similar item j. If C_j has a greater degree of inclusion in the target user's previous preferences R_{tu}, so that the value of $CD(C_j, R_{tu})$ is high, this suggests that the characteristics of item j are nearer to the target user's previous preferences, and that this item is more likely to be preferred by the target user. Item j will therefore be given a greater weight when making the rating prediction.

5.6 Summary

IBCF is an important technology which is widely used in RSs. This approach builds an item-item similarity matrix as the basis for rating predictions. The IBCF approach has good scalability and can be used with large-scale data information system. However, in IBCF all items are treated as having the same weight. In reality, a customer's preferences may change over time, so that a time factor should be used when computing item-item similarity. In addition, items that appear similar to the target item make different contributions to the rating predictions, so the relative weights of similar items also need to be taken into account.

In this chapter, we introduced a time-related correlation degree function, allowing a time factor to be applied to item-item similarity. For each target user, items that were rated nearer the time of the target item rating will have greater weight when computing item-item similarity. We further added the covering degree function to the rating prediction procedure. Items with a higher covering degree with

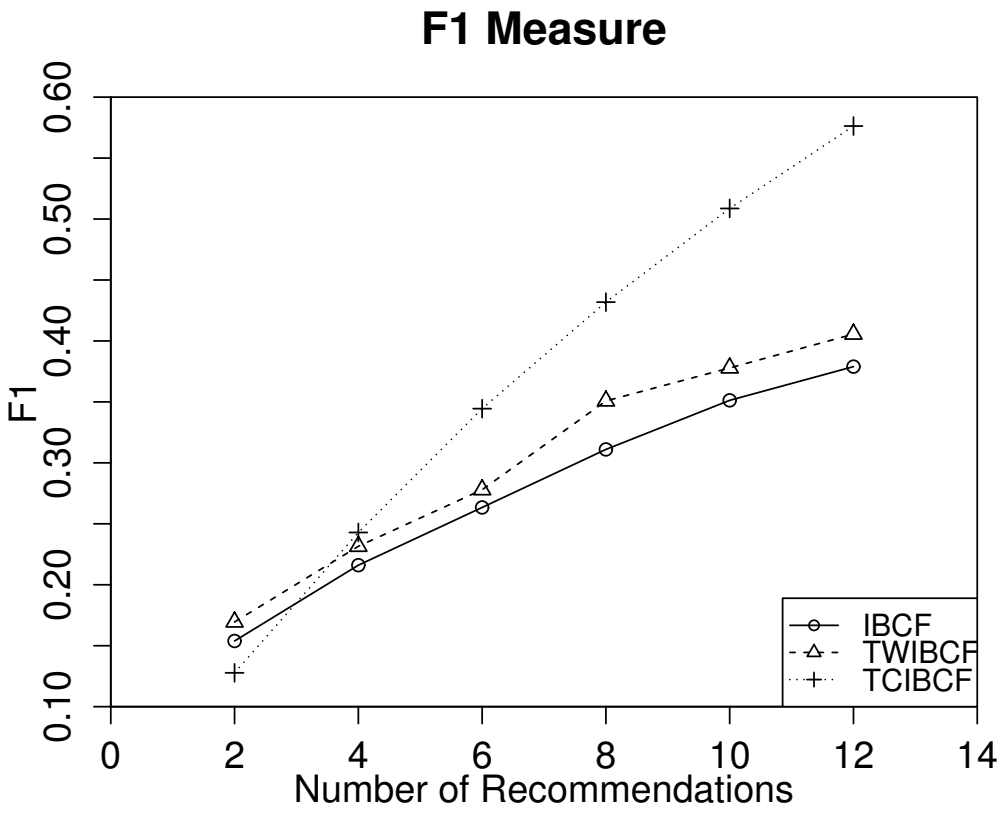

FIGURE 5.3: F1 for the TCIBCF, TWIBCF, and IBCF versus the number of recommendations

the target user's relevant set will have greater power to predict the rating score of the target item. The results of our experiments demonstrated that our novel TCIBCF could achieve better accuracy and higher precision than existing work, making it an effective approach to RS.

TABLE 5.2: Results of MAE and RMSE metrics

Approaches	MAE	RMSE
$TCIBCF_{k=10}$	**1.032**	**1.200**
$TWIBCF_{k=10}$	1.206	1.358
$IBCF_{k=10}$	1.217	1.397
$TCIBCF_{k=20}$	**1.072**	**1.259**
$TWIBCF_{k=20}$	1.181	1.339
$IBCF_{k=20}$	1.192	1.390
$TCIBCF_{k=30}$	**1.132**	**1.345**
$TWIBCF_{k=30}$	1.186	1.387
$IBCF_{k=30}$	1.205	1.403
$TCIBCF_{k=40}$	**1.144**	**1.361**
$TWIBCF_{k=40}$	1.198	1.377
$IBCF_{k=40}$	1.216	1.424
$TCIBCF_{k=50}$	**1.164**	**1.382**
$TWIBCF_{k=50}$	1.189	1.396
$IBCF_{k=50}$	1.206	1.414

Chapter 6

Conclusions and future work

6.1 Thesis summary

With the rapid development of technology, the data and information has been increasing at a dramatical speed. Therefore, more customers are facing the problem of discovering the demanded contents from overwhelmingly massive data. As the result, this problem becomes a popular research topic and attracts attention from lots of scientists. RSs can properly deal with the information overload problem, which are widely welcome by users and thus adopted by great amount of websites and corporations. Many approaches are proposed for RSs by far, CF is the most significant and successful approach among them. Nevertheless, some problems in existing CF approach could prevent the further development of CF in RSs. For example, UBCF cannot provide recommendations for an active user with satisfactory accuracy and diversity simultaneously. Personalized recommendations cannot be provided by UBCF for a new user which often has insufficient information. In addition, items that make a more significant contribution cannot have high weighting in IBCF.

In this dissertation, to address and solve problems mentioned above, we utilize covering-based rough sets as our research method, and improve the traditional UBCF and IBCF respectively. The main work of this dissertation is summarized as follows:

First, a CBCF approach is proposed to improve the traditional UBCF for active users' personalized recommendations. For traditional UBCF, in candidate neighbors of an active user, because there exists some redundant users who have higher similarity but can make predictions only for a few types of items, the traditional UBCF usually can not provide recommendations with satisfactory accuracy and diversity at the same time for an active user. Combining with the characteristics of redundant users in UBCF and redundant elements in covering-based rough sets, CBCF inserts a neighbor selection procedure into the traditional UBCF that could remove redundant candidate neighbors by covering reduction algorithm. To remove as many redundant users as possible, according to the sufficient information from an active user, we first extracted relevant attributes of the active user, then constructed decision class by all items that fit the active user's relevant attributes, and reduced the domain from all items to decision class. Experimental results suggest that CBCF outperforms than the traditional UBCF and can provide recommendations with satisfactory accuracy and diversity simultaneously for an active user.

Second, CBCF approach is improved for new users' personalized recommendations. For the previous CBCF, new users' ratings are usually very few, and it is unreliable to extract relevant attributes according to a new user's rating information. Moreover, in the previous CBCF, the item attribute matrix had to be inputted as the indispensable condition, even though some datasets do not have this information.

Therefore, we improve previous CBCF approach, and reconstruct the decision class for the new user as the set of niche items in the dataset used for recommendation. In this way, redundant candidate neighbors for a new user are able to be removed as many as possible, and the decision class is easily constructed from the user-item matrix without needing special additional information. Experimental results suggest that the improved CBCF significantly outperforms those of existing work and can provide personalized recommendations with satisfactory accuracy and diversity simultaneously for a new user.

Third, a TCIBCF approach is proposed to improve the traditional IBCF for item-variance weighting. For the traditional IBCF, all items are accorded the same weight when computing the similarity and making predictions. However, different items have different contributions to the process of similarity computation and rating prediction, some items that make a more significant contributions should have higher weighting. Therefore, we present time-related correlation degree and covering degree, and apply them to the traditional IBCF to propose TCIBCF. TCIBCF is able to increase the weighting of items that make a greater contribution to the similarity computation and rating prediction. Experimental results suggest that TCIBCF can produce recommendation results superior to those of existing work.

6.2 Future research directions

While the current dissertation can address some problems of CF in RSs, some aspects remain to be investigated in future studies.

First, because the proposed CBCF belongs to CF approaches, so although the CBCF approach is proposed to focus on improving the traditional UBCF, the principles of CBCF approach can also be incorporated into IBCF; however, how to define the redundant items requires further consideration. On the other hand, the CBCF approach aims to improve personalized recommendations for both new and active users, with the result that we can summarize it to propose a CBCF framework for RSs.

Second, the proposed TCIBCF approach can be further extended to address the new item cold-start issue, which is a very difficult problem faced in RSs, because a new item only has been rated by few users, so we cannot obtain sufficient information from the new item.

Bibliography

Adomavicius, Gediminas and YoungOk Kwon (2011). "Maximizing aggregate recommendation diversity: A graph-theoretic approach". In: *Proc. of the 1st International Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011)*, pp. 3–10.

— (2012). "Improving aggregate recommendation diversity using ranking-based techniques". In: *IEEE Transactions on Knowledge and Data Engineering* 24.5, pp. 896–911.

Adomavicius, Gediminas and Alexander Tuzhilin (2005). "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions". In: *IEEE transactions on knowledge and data engineering* 17.6, pp. 734–749.

Ahn, Hyung Jun (2008). "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem". In: *Information Sciences* 178.1, pp. 37–51.

Bobadilla, JesúS et al. (2012). "A collaborative filtering approach to mitigate the new user cold start problem". In: *Knowledge-Based Systems* 26, pp. 225–238.

Bobadilla, Jesús et al. (2013). "Recommender systems survey". In: *Knowledge-based systems* 46, pp. 109–132.

Boim, Rubi, Tova Milo, and Slava Novgorodov (2011). "Diversification and refinement in collaborative filtering recommender". In: *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, pp. 739–744.

Chang, Hsin Hsin and I Chen Wang (2011). "Enterprise Information Portals in support of business process, design teams and collaborative commerce performance". In: *International Journal of Information Management* 31.2, pp. 171–182.

Chen, Chien Chin et al. (2013). "An effective recommendation method for cold start new users using trust and distrust networks". In: *Information Sciences* 224, pp. 19–36.

Clarke, Charles LA et al. (2008). "Novelty and diversity in information retrieval evaluation". In: *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 659–666.

Di Noia, Tommaso et al. (2014). "An analysis of users' propensity toward diversity in recommendations". In: *Proceedings of the 8th ACM Conference on Recommender Systems*. ACM, pp. 285–288.

Ding, Yi and Xue Li (2005). "Time weight collaborative filtering". In: *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, pp. 485–492.

Edmunds, Angela and Anne Morris (2000). "The problem of information overload in business organisations: a review of the literature". In: *International journal of information management* 20.1, pp. 17–28.

Esparza, Sandra Garcia, Michael P OMahony, and Barry Smyth (2011). "Effective product recommendation using the real-time web". In: *Research and Development in Intelligent Systems XXVII*. Springer, pp. 5–18.

Frémal, Sébastien and Fabian Lecron (2017). "Weighting strategies for a recommender system using item clustering based on genres". In: *Expert Systems with Applications* 77, pp. 105–113.

Gan, Mingxin and Rui Jiang (2013). "Constructing a user similarity network to remove adverse influence of popular objects for personalized recommendation". In: *Expert Systems with Applications* 40.10, pp. 4044–4053.

Goldberg, David et al. (1992). "Using collaborative filtering to weave an information tapestry". In: *Communications of the ACM* 35.12, pp. 61–70.

Goldberg, Ken et al. (2001). "Eigentaste: A constant time collaborative filtering algorithm". In: *information retrieval* 4.2, pp. 133–151.

Hameed, Mohd Abdul, Omar Al Jadaan, and S Ramachandram (2012). "Collaborative filtering based recommendation system: A survey". In: *International Journal on Computer Science and Engineering* 4.5, p. 859.

Herlocker, Jon, Joseph A Konstan, and John Riedl (2002). "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms". In: *Information retrieval* 5.4, pp. 287–310.

Herlocker, Jonathan L et al. (1999). "An algorithmic framework for performing collaborative filtering". In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 230–237.

Hiltz, Starr R and Murray Turoff (1985). "Structuring computer-mediated communication systems to avoid information overload". In: *Communications of the ACM* 28.7, pp. 680–689.

Höchstötter, Nadine and Dirk Lewandowski (2009). "What users see–Structures in search engine results pages". In: *Information Sciences* 179.12, pp. 1796–1812.

Hu, Bo and Martin Ester (2013). "Spatial topic modeling in online social media for location recommendation". In: *Proceedings of the 7th ACM conference on Recommender systems*. ACM, pp. 25–32.

Hu, Rong and Pearl Pu (2011). "Helping Users Perceive Recommendation Diversity." In: *DiveRS@ RecSys*, pp. 43–50.

Javari, Amin and Mahdi Jalili (2015). "A probabilistic model to resolve diversity-accuracy challenge of recommendation systems". In: *Knowledge and Information Systems* 44.3, pp. 609–627.

Kaleli, Cihan (2014). "An entropy-based neighbor selection approach for collaborative filtering". In: *Knowledge-Based Systems* 56, pp. 273–280.

Karypis, George (2001). "Evaluation of item-based top-n recommendation algorithms". In: *Proceedings of the tenth international conference on Information and knowledge management*. ACM, pp. 247–254.

Kefalas, Pavlos and Yannis Manolopoulos (2017). "A time-aware spatio-textual recommender system". In: *Expert Systems with Applications* 78, pp. 396–406.

Koohi, Hamidreza and Kourosh Kiani (2016). "User based Collaborative Filtering using fuzzy C-means". In: *Measurement* 91, pp. 134–139.

Koren, Yehuda (2010). "Collaborative filtering with temporal dynamics". In: *Communications of the ACM* 53.4, pp. 89–97.

Kotkov, Denis, Shuaiqiang Wang, and Jari Veijalainen (2016). "A survey of serendipity in recommender systems". In: *Knowledge-Based Systems* 111, pp. 180–192.

Kunaver, Matevž and Tomaž Požrl (2017). "Diversity in recommender systems–A survey". In: *Knowledge-Based Systems*.

Lee, Tong Queue, Young Park, and Yong-Tae Park (2008). "A time-based approach to effective recommender systems using implicit feedback". In: *Expert systems with applications* 34.4, pp. 3055–3062.

Li, Dongsheng et al. (2014). "Item-based top-N recommendation resilient to aggregated information revelation". In: *Knowledge-Based Systems* 67, pp. 290–304.

Li, Dongsheng et al. (2016). "An algorithm for efficient privacy-preserving item-based collaborative filtering". In: *Future Generation Computer Systems* 55, pp. 311–320.

Lika, Blerina, Kostas Kolomvatsos, and Stathes Hadjiefthymiades (2014). "Facing the cold start problem in recommender systems". In: *Expert Systems with Applications* 41.4, pp. 2065–2073.

Linden, Greg, Brent Smith, and Jeremy York (2003). "Amazon. com recommendations: Item-to-item collaborative filtering". In: *IEEE Internet computing* 7.1, pp. 76–80.

Liu, Haifeng et al. (2014a). "A new user similarity model to improve the accuracy of collaborative filtering". In: *Knowledge-Based Systems* 56, pp. 156–166.

Liu, Jian-Guo, Kerui Shi, and Qiang Guo (2012). "Solving the accuracy-diversity dilemma via directed random walks". In: *Physical Review E* 85.1, p. 016118.

Liu, Long et al. (2014b). "A real-time personalized route recommendation system for self-drive tourists based on vehicle to vehicle communication". In: *Expert Systems with Applications* 41.7, pp. 3409–3417.

Lu, Jie et al. (2015). "Recommender system application developments: a survey". In: *Decision Support Systems* 74, pp. 12–32.

Malone, Thomas W et al. (1987). "Intelligent information-sharing systems". In: *Communications of the ACM* 30.5, pp. 390–402.

Niemann, Katja and Martin Wolpers (2013). "A new collaborative filtering approach for increasing the aggregate diversity of recommender systems". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 955–963.

Papagelis, Manos and Dimitris Plexousakis (2005). "Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents". In: *Engineering Applications of Artificial Intelligence* 18.7, pp. 781–789.

Park, Youngki et al. (2015). "Reversed CF: A fast collaborative filtering algorithm using a k-nearest neighbor graph". In: *Expert Systems with Applications* 42.8, pp. 4022–4028.

Pawlak, Zdzisław (1982). "Rough sets". In: *International Journal of Parallel Programming* 11.5, pp. 341–356.

Pawlak, Zdzisław and Andrzej Skowron (2007). "Rudiments of rough sets". In: *Information sciences* 177.1, pp. 3–27.

Pomykala, Janusz A (1987). "Approximation operations in approximation space". In: *Bulletin of the Polish Academy of Sciences* 35.9-10, pp. 653–662.

Said, Alan, Brijnesh J Jain, and Sahin Albayrak (2012). "Analyzing weighting schemes in collaborative filtering: cold start, post cold start and power users". In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM, pp. 2035–2040.

Sarwar, Badrul et al. (2001). "Item-based collaborative filtering recommendation algorithms". In: *Proceedings of the 10th international conference on World Wide Web*. ACM, pp. 285–295.

Son, Le Hoang (2016). "Dealing with the new user cold-start problem in recommender systems: A comparative review". In: *Information Systems* 58, pp. 87–104.

Symeonidis, Panagiotis et al. (2008). "Collaborative recommender systems: Combining effectiveness and efficiency". In: *Expert Systems with Applications* 34.4, pp. 2995–3013.

Tsang, Eric CC, Chen Degang, and Daniel S Yeung (2008). "Approximations and reducts with covering generalized rough sets". In: *Computers & Mathematics with Applications* 56.1, pp. 279–289.

Tsang, Eric CC et al. (2004). "On the upper approximations of covering generalized rough sets". In: *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*. Vol. 7. IEEE, pp. 4200–4203.

Tyagi, Shweta and Kamal K Bharadwaj (2012). "Enhanced new user recommendations based on quantitative association rule mining". In: *Procedia Computer Science* 10, pp. 102–109.

Vargas, Saúl (2011). "New approaches to diversity and novelty in recommender systems". In: *Fourth BCS-IRSG symposium on future directions in information access (FDIA 2011), Koblenz*. Vol. 31.

Wang, Jian-peng, Dai Dai, and Zheng-chun Zhou (2004). "Fuzzy covering generalized rough sets". In: *Journal of Zhoukou Teachers College* 21.2, pp. 20–22.

Yang, Tian and Qingguo Li (2010). "Reduction about approximation spaces of covering generalized rough sets". In: *International Journal of Approximate Reasoning* 51.3, pp. 335–345.

Yao, Yiyu and Bingxue Yao (2012). "Covering based rough set approximations". In: *Information Sciences* 200, pp. 91–107.

Zakowski, Wojciech (1983). "Approximations in the space (u, π)". In: *Demonstratio mathematica* 16.3, pp. 761–769.

Zhou, Tao et al. (2010). "Solving the apparent diversity-accuracy dilemma of recommender systems". In: *Proceedings of the National Academy of Sciences* 107.10, pp. 4511–4515.

Zhu, Tianqing et al. (2014). "An effective privacy preserving algorithm for neighborhood-based collaborative filtering". In: *Future Generation Computer Systems* 36, pp. 142–155.

Zhu, William (2007). "Topological approaches to covering rough sets". In: *Information sciences* 177.6, pp. 1499–1508.

— (2009). "Relationship between generalized rough sets based on binary relation and covering". In: *Information Sciences* 179.3, pp. 210–225.

Zhu, William and Fei-Yue Wang (2003). "Reduction and axiomization of covering generalized rough sets". In: *Information sciences* 152, pp. 217–230.

— (2007). "On three types of covering-based rough sets". In: *IEEE transactions on knowledge and data engineering* 19.8.

— (2012). "The fourth type of covering-based rough sets". In: *Information Sciences* 201, pp. 80–92.

Appendix A

Papers published during the doctoral period

[1] Zhipeng Zhang, Yasuo Kudo, Tetsuya Murai (2016). "Neighbor selection for user-based collaborative filtering using covering-based rough sets". In: Annals of Operations Research. doi:10.1007/s10479-016-2367-1.
[2] Zhipeng Zhang, Yasuo Kudo, Tetsuya Murai (2016). "Improvement of item-based collaborative filtering by adding time factor and covering degree". In: Proc. of 8th Int. Symposium on Soft Computing and Intelligent Systems and 17th Int. Symposium on Advanced Intelligent Systems (SCIS&ISIS2016), SOFT, pp. 543-547.
[3] Zhipeng Zhang, Yasuo Kudo, Tetsuya Murai (2015). "Modification of the covering-based collaborative filtering model to alleviate the new user cold-start problem". In: Proc. of 16th Int. Symposium on Advanced Intelligent Systems (ISIS 2015), KIIS, pp. 1238-1249.
[4] Zhipeng Zhang, Yasuo Kudo, Tetsuya Murai (2015). "Applying covering-based Rough set theory to user-based collaborative filtering to enhance the quality of recommendations". In: Proc. of 4th Int. Symposium on Integrated Uncertainty in Knowledge Modeling and Decision Making (IUKM 2015), LNAI 9376, pp. 279-289.