Ann Oper Res (2017) 256:359–374
DOI 10.1007/s10479-016-2367-1

IUKM2015

Neighbor selection for user-based collaborative filtering using covering-based rough sets

Zhipeng Zhang[1] · **Yasuo Kudo**[2] · **Tetsuya Murai**[3]

Published online: 8 November 2016

Abstract Recommender systems (RSs) provide personalized information by learning user preferences. User-based collaborative filtering (UBCF) is a significant technique widely utilized in RSs. The traditional UBCF approach selects k-nearest neighbors from candidate neighbors comprised by all users; however, this approach cannot achieve good accuracy and coverage values simultaneously. We present a new approach using covering-based rough set theory to improve traditional UBCF in RSs. In this approach, we insert a user reduction procedure into the traditional UBCF approach. 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. Our experimental results suggest that, for the sparse datasets that often occur in real RSs, the proposed approach outperforms than the traditional UBCF, and can provide satisfactory accuracy and coverage simultaneously.

Keywords Covering-based rough sets · User-based collaborative filtering · Covering reduction · Active user · Recommender systems

1 Introduction

With rapid development in technology and improved economic conditions, consumers tend to demand more personalized services. Recently, recommender systems (RSs), which can

✉ Yasuo Kudo
kudo@csse.muroran-it.ac.jp

Zhipeng Zhang
14096507@mmm.muroran-it.ac.jp

Tetsuya Murai
t-murai@photon.chitose.ac.jp

1 Course of Advanced Information and Electronic Engineering, Graduate School of Engineering, Muroran Institute of Technology, Mizumoto 27-1, Muroran 050-8585, Japan

2 College of Information and Systems, Muroran Institute of Technology, Mizumoto 27-1, Muroran 050-8585, Japan

3 Faculty of Science and Technology, Chitose Institute of Science and Technology, 758-65 Bibi, Chitose 066-8655, Japan

provide item recommendations based on personal preferences that help users make purchase decisions, have become very popular (Adomavicius and Tuzhilin 2005; Bobadilla and Ortega 2013; Lu et al. 2015).

Collaborative filtering (CF) is a significant component of the recommendation process (Symeonidis et al. 2008; Hameed et al. 2012). User-based collaborative filtering (UBCF) approach relies on active user neighborhood information to make predictions and recommendations (Herlocker and Konstan 2002). 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 and Konstan 2002). So we consider the k-NN UBCF approach as the traditional UBCF approach in the rest of this paper.

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 and Konstan 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.

Aiming at improving the traditional UBCF approach to obtain good values of accuracy and coverage at the same time, in this study, covering-based rough set theory is applied to RSs. We propose covering-based collaborative filtering (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 paper is organized as follows. In Sect. 2, some background information is provided. We review basic concepts involved in the traditional UBCF approach and covering-based rough sets, describe covering reduction algorithm. In Sect. 3, we analyze neighborhood selection problems, then give the detailed motivation and construction of the CBCF approach. In Sect. 4, we provide an example to demonstrate the CBCF approach. In Sect. 5, we describe our experiments and compare CBCF results with the results obtained using the traditional UBCF approach. Conclusions and suggestions for future work are presented in Sect. 6. Note that this paper is a revised and extended version of a previous paper (Zhang et al. 2015).

2 Background

Here, we introduce the basic knowledge of UBCF approach, then briefly describe covering, covering approximation, and covering reduction. In addition, we analyze and compare three types of reduction algorithms.

2.1 Overview of the traditional UBCF approach

Given an RS, let U and I be finite sets of users and items. Suppose each item has the same attributes. The item attribute matrix is denoted as AM. Let $R \cup \{\star\}$ be the set of possible item rating scores, and RM be the User-Item rating matrix. Note that the absence of a rating is represented by an asterisk ($\star$). The rating score of user u for item i is denoted $r_{u,i} \in R \cup \{\star\}$, and the average of the valid ratings of user u is denoted $\bar{r}_u$. 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, I_u^c is the complementary set of I_u, indicates items which have not yet rated by user u.

The traditional UBCF approach is one type of CF, it utilizes the information of an active user's neighborhood to make predictions and recommendations, it can be separated into three steps:

Step 1: Similarity computation An active user au's candidate neighbors CN_{au} are comprised by all users. Based on historical rating information, compute similarity between each user $u \in CN_{au}$ and an active user au. Here, Pearson correlation coefficient approach (1) is popularly used as a similarity measure:

$$sim(au, u) = \frac{\sum_{i \in I_{au} \cap I_u} \left(r_{au,i} - \bar{r}_{au}\right)\left(r_{u,i} - \bar{r}_u\right)}{\sqrt{\sum_{i \in I_{au} \cap I_u} \left(r_{au,i} - \bar{r}_{au}\right)^2}\sqrt{\sum_{i \in I_{au} \cap I_u} \left(r_{u,i} - \bar{r}_u\right)^2}}, \tag{1}$$

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

$$\bar{r}_{au} = \frac{\sum_{i \in I_{au}} r_{au,i}}{card(I_{au})}. \tag{2}$$

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

Step 3: Rating prediction and item recommendation Normalize ratings and according to the rating information of neighborhood, predict a rating score $p_{au,i}$ for each item i in unrated item set I_{au}^c of the active user au. The adjusted weighted sum approach (3) is often utilized to make rating prediction.

$$p_{au,i} = \bar{r}_{au} + \lambda \sum_{u \in N_{au}(k) \cap U_i} sim(au, u) * (r_{u,i} - \bar{r}_u), \tag{3}$$

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

$$\lambda = \frac{1}{\sum_{u \in N_{au}(k) \cap U_i} sim(au, u)}. \tag{4}$$

Note that, within specific systems, these steps may overlap or the order may be slightly different. Algorithm 1 summarizes the traditional UBCF approach.

2.2 Covering-based rough sets and covering reduction theory

Rough set theory was first presented by Pawlak in the early 1980s (Pawlak 1982). Lower and upper approximation operations are key concepts in classical rough set theory, and an equivalence relation, i.e., a partition, is the simplest formulation of the lower and upper approximation operations (Pawlak and Skowron 2007). Covering-based rough sets extend

Algorithm 1 Traditional UBCF approach

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

the classical rough set by utilizing a covering of the domain rather than a partition. Here, we define covering and covering approximation space. More detailed explanations can be found in the literature (Tsang et al. 2008; Zhu 2009a, b; Yao and Yao 2012).

Definition 1 Let T be the domain of discourse and C be a family of subsets of T. If none of the subsets in C is empty and $\cup C = T$, C is called a covering of T.

Definition 2 Let T be a non-empty set and C be a covering of T. Then, we refer to the ordered pair $\langle T, C\rangle$ as the covering approximation space.

Note that a partition of T is a covering of T; thus, the concept of a covering is an extension of a partition. Different lower and upper approximation operations generate different types of covering-based rough sets. The covering-based rough set was first presented by Zakowski (1983). Zakowski extended Pawlak's rough set theory from a partition to a covering, and presented the basic concept of covering. In addition, related studies have been undertaken by Pomykala (1987), Tsang et al. (2004), Wang et al. (2004), Zhu (2007), and Zhu and Wang (2012).

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 and Wang (2003). In this paper, we refer to the algorithm proposed by 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 3 defines this algorithm.

Definition 3 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 and Wang (2007, 2012) presented two other covering reduction algorithms, which we refer to as the second and third types of reduction algorithms, respectively. Definition 4 defines the second-type algorithm, which corresponds to the definition of *exclusion*(C) provided by Zhu and Wang (2007). Definition 5 defines the third-type algorithm, which corresponds to the definition of *exact-reduct*(C) (Zhu and Wang 2012).

Definition 4 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 5 Let C be a covering of domain T, and $K \in C$. If there exists $K_1, K_2, \ldots K_m \in C - K$ such that $K = K_1 \cup, \ldots, \cup K_m$, 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.

3 Covering-based collaborative filtering approach

Here, first we discuss neighborhood selection problems in the traditional UBCF approach. To address these problems, we propose CBCF approach and describe its detail process. In addition, we discuss the innovative aspects and significance of the proposed approach.

Algorithm 2 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 $K_j \subset K_i$ then
5:       if $K_j \in reduct(C)$ then
6:         $reduct(C) = reduct(C) - \{K_j\}$;
7:       end if
8:     end if
9:   end for
10: end for
11: return $reduct(C)$;
```

3.1 Neighborhood selection problems in traditional UBCF approach

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 and Konstan (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 20–60 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 coverage simultaneously.

3.2 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 Sect. 2.2, 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:

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

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

3.3 Construction

In CBCF approach, we insert user reduction step into the traditional UBCF approach. Algorithm 3 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 (5).

$$D = \{i \in I | at_1(i) = av_1, at_2(i) = av_2, \ldots, at_m(i) = av_m\}, \tag{5}$$

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

$$C_u = \{i \in D | r_{u,i} \geq \theta\}. \tag{6}$$

Let $C^* = D - \cup C_u$; then, $C = \{C_1, C_2 \ldots C_n, C^*\}$ 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

$$U^r = \{u \in U | C_u \in reduct(C)\}. \tag{7}$$

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.

3.4 Discussion

To provide recommendations with good values of accuracy and coverage 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

Algorithm 3 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, $C = \{C_1, C_2, \ldots C_n, C^*\}$ 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 $p_{au,i}$.
```

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 coverage at the same time.

4 Example of CBCF approach in RSs

Here, we present an example to explain the CBCF approach more clearly. Table 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

Table 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 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 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

likes the given item more. Table 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:

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 $D = \{\text{Item 2, Item 3, Item 4, Item 5, Item 6}\}$.

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 :

$$\begin{aligned}
&C_1 = \{\text{Item 2, Item 4, Item 5, Item 6}\}, \quad C_2 = \{\text{Item 2, Item 4, Item 6}\},\\
&C_3 = \{\text{Item 3, Item 4, Item 5, Item 6}\}, \quad C_4 = \{\text{Item 3, Item 5, Item 6}\},\\
&C_5 = \{\text{Item 2, Item 3, Item 6}\}.
\end{aligned}$$

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 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.

5 Experiments and evaluation

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.

5.1 Experimental setup and evaluation metrics

In our experiments, we utilized the MovieLens (Herlocker et al. 1999) and Jester (Ken et al. 2001) datasets because they are often used to evaluate RSs. The MovieLens 100K dataset consists of 1682 movies, 943 users, and 100,000 ratings on a scale of 1–5. 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

Table 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

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 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 2498 users for our experiment.

To measure the performance of the proposed approach, we used the mean absolute error (MAE), root mean square error (RMSE), and coverage as evaluation metrics, 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.

$$MAE = \frac{1}{card(U)} \sum_{u \in U} \left(\frac{1}{card(O_u)} \sum_{i \in O_u} |p_{u,i} - r_{u,i}| \right), \tag{8}$$

$$RMSE = \frac{1}{card(U)} \sum_{u \in U} \sqrt{\frac{1}{card(O_u)} \sum_{i \in O_u} (p_{u,i} - r_{u,i})^2}, \tag{9}$$

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 $Z_u = \{i \in I | S_{u,i} \neq \emptyset\}$.

$$Coverage = \frac{1}{card(U)} \sum_{u \in U} \left(100 \times \frac{card(I_u^c \cap Z_u)}{card(I_u^c)} \right). \tag{10}$$

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:

$$Reduction Rate = \frac{1}{card(U)} \sum_{u \in U} \frac{card(CN_u - CN_u^r)}{card(CN_u)}, \tag{11}$$

Table 5 Number of candidate neighbors for traditional UBCF and CBCF approaches

	UBCF	CBCF	Reduction rate
MovieLens	943	193	0.795
Jester	2498	580	0.768

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

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 and Konstan (2002), we selected different size k neighborhood from candidate neighbors, $k \in \{20, 25, 30, \ldots, 60\}$. 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 and Konstan 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 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 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 2498 users, so the number of candidate neighbors for traditional UBCF approach is 2498. 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 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 2 illustrates the 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

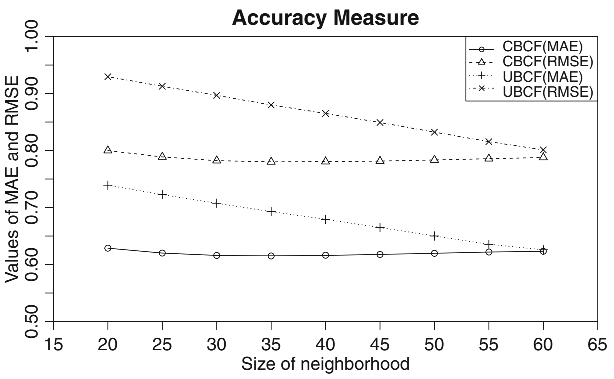

Fig. 1 Accuracy results (MAE and RMSE) versus the size of neighborhood with MovieLens dataset

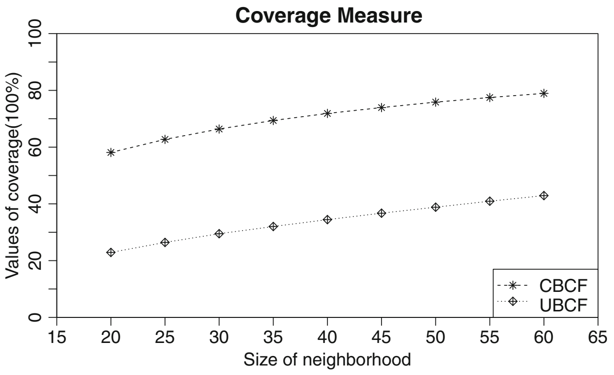

Fig. 2 Coverage results versus the size of neighborhood with MovieLens dataset

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 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 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.

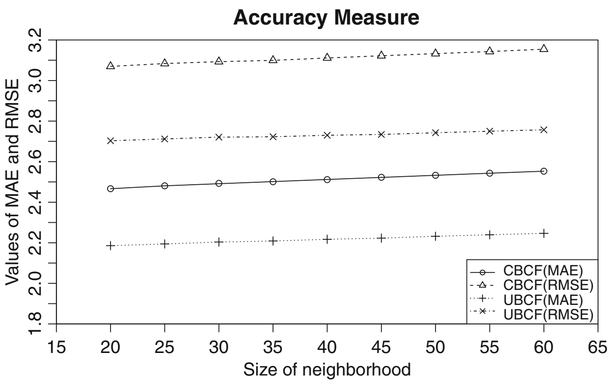

Fig. 3 Accuracy results (MAE and RMSE) versus the size of neighborhood with Jester dataset

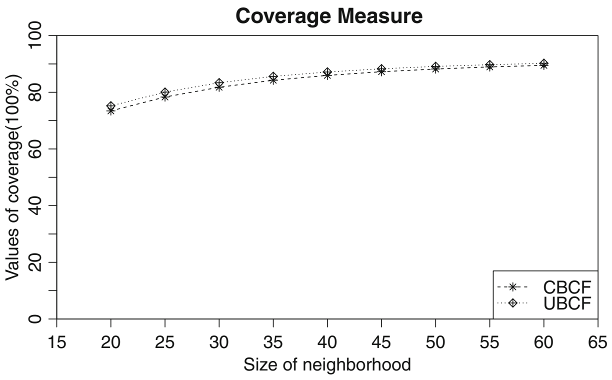

Fig. 4 Coverage results versus the size of neighborhood with Jester dataset

5.3 Discussion

The experimental results indicate that the proposed CBCF approach demonstrates different performance with different datasets. In the MovieLens dataset experiment, there were 1682 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 2498 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.

6 Conclusion and future work

UBCF approach is the most commonly used and studied technology for making recommendations in RSs. Generally, we use accuracy and coverage 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 paper, 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 coverage simultaneously. Thus, the proposed CBCF approach can recommend satisfactory recommendations and obtain high confidence from the active user.

In the future, we plan to improve the proposed CBCF approach to address the new user cold-start problem, which is an extremely difficult issue in RSs; however, the proposed CBCF approach cannot select appropriate neighborhood based on insufficient new user data. We must propose a new similarity measure to select neighborhood efficiently and make satisfactory recommendations for new users.

References

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

Bobadilla, J., & Ortega, F. (2013). Recommender system survey. *Knowledge-Based Systems*, *46*, 109–132.

Gan, M. X., & Jiang, R. (2013). Constructing a user similarity network to remove adverse influence of popular objects for personalized recommendation. *Expert Systems with Application*, *40*, 4044–4053.

Hameed, M. A., Jadaan, O. A., & Ramachandram, S. (2012). Collaborative filtering based recommendation system: A survey. *International Journal on Computer Science and Engineering*, *4*(5), 859–876.

Herlocker, J. L., & Konstan, J. A. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, *5*, 287–310.

Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (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*, pp. 230–237.

Ken, G., Theresa, R., Dhruv, G., & Chris, P. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, *4*(2), 133–151.

Lu, J., Wu, D., & Mao, M. (2015). Recommender system application developments: A survey. *Decision Support Systems*, *74*, 12–32.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences*, *11*, 341–356.

Pawlak, Z., & Skowron, A. (2007). Rudiments of rough sets. *Information Sciences*, *177*, 3–27.

Pomykala, J. A. (1987). Approximation operations in approximation space. *Bulletin of the Polish Academy of Sciences Mathematics*, *35*(1), 653–662.

Symeonidis, P., Nanopoulos, A., Papadopoulos, A. N., & Manolopoulos, Y. (2008). Collaborative recommender systems: Combining effectiveness and efficiency. *Expert Systems with Application*, *34*, 2995–3013.

Tsang, E., Cheng, D., Lee, J., & Yeung, D. (2004). On the upper approximations of covering generalized rough sets. In *Proceedings of the 3rd international conference machine learning and cybernetics*, pp. 4200–4203.

Tsang, E., Chen, D., & Yeung, D. S. (2008). Approximations and reducts with covering generalized rough sets. *Computers and Mathematics With Applications*, *56*, 279–289.

Wang, J., Dai, D., & Zhou, Z. (2004). Fuzzy covering generalized rough sets. *Journal of Zhoukou Teachers College*, *21*, 20–22.

Yang, T., & Li, Q. G. (2010). Reduction about approximation spaces of covering generalized rough sets. *International Journal of Approximate Reasoning*, *51*, 335–345.

Yao, Y. Y., & Yao, B. X. (2012). Covering based rough set approximations. *Information Sciences*, *200*, 91–107.

Zakowski, W. (1983). Approximations in the space (u, π). *Demonstration Math*, *16*, 761–769.

Zhang, Z. P., Kudo, Y., & Murai, T. (2015). Applying covering-based rough set theory to user-based collaborative filtering to enhance the quality of recommendations. In *Proceedings of the 4th international symposium on IUKM*, pp. 279–289.

Zhu, W. (2007). Topological approached to covering rough sets. *Information Sciences*, *177*, 1499–1508.

Zhu, W. (2009a). Relationship between generalized rough sets based on binary relation. *Information Sciences*, *179*, 210–225.

Zhu, W. (2009b). Relationship among basic concepts in covering-based rough sets. *Information Sciences*, *179*, 2478–2486.

Zhu, W., & Wang, F. Y. (2003). Reduction and maximization of covering generalized rough sets. *Information Sciences*, *152*, 217–230.

Zhu, W., & Wang, F. Y. (2007). On three types of covering-based rough sets. *IEEE Transactions on Knowledge and Data Engineering*, *19*, 1131–1144.

Zhu, W., & Wang, F. Y. (2012). The fourth type of covering-based rough sets. *Information Sciences*, *201*, 80–92.