Research Article

Effect of Collaborative Recommender System Parameters: Common Set Cardinality and the Similarity Measure

Mohammad Yahya H. Al-Shamri^{1,2}

¹Computer Engineering Department, College of Computer Science, King Khalid University, P.O. Box 394, Abha, Saudi Arabia

²Electrical Engineering Department, Faculty of Engineering and Architecture, Ibb University, Ibb, Yemen

Correspondence should be addressed to Mohammad Yahya H. Al-Shamri; mohamad.alshamri@gmail.com

Received 25 March 2016; Revised 13 May 2016; Accepted 22 May 2016

Academic Editor: Theo Van Der Weide

Copyright © 2016 Mohammad Yahya H. Al-Shamri. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recommender systems are widespread due to their ability to help Web users surf the Internet in a personalized way. For example, collaborative recommender system is a powerful Web personalization tool for suggesting many useful items to a given user based on opinions collected from his neighbors. Among many, similarity measure is an important factor affecting the performance of the collaborative recommender system. However, the similarity measure itself largely depends on the overlapping between the user profiles. Most of the previous systems are tested on a predefined number of common items and neighbors. However, the system performance may vary if we changed these parameters. The main aim of this paper is to examine the performance of the collaborative recommender system under many similarity measures, common set cardinalities, rating mean groups, and neighborhood set sizes. For this purpose, we propose a modified version for the mean difference weight similarity measure and a new evaluation metric called users' coverage for measuring the recommender system ability for helping users. The experimental results show that the modified mean difference weight similarity measure outperforms other similarity measures and the collaborative recommender system performance varies by varying its parameters; hence we must specify the system parameters in advance.

1. Introduction

Today, Web users face an abundance of choices when they surf the Web. Hence, recommender systems (RSs) as many Web personalization tools become necessary to offer Web users personalized items they may like. These systems become available in many Web sites that cover social networks, e-commerce, e-business, e-tourist, and many others [1, 2]. A recent survey for the applications of the recommender systems and their classifications is given in [2].

Basically, RS compares users based on a suitable similarity measure which plays an important rule for the success of the whole system. However, different similarity measures often lead to different sets of neighbors for a given active user. A good similarity measure will produce a close set of neighbors for a given active user [3]. Actually, many of the existing similarity measures for collaborative recommender systems rely on the overlapping between users. Nevertheless, the size of this overlapping is not explored in detail where most of

the previous work studied similarity measures based on a predefined number of common items [3–8].

This motivates us to study the cardinality effect of the common set on the performance of different similarity measures for collaborative recommender systems. The proximity between two users based upon a single commonly rated item is surely weaker than that of 20 common rated items. Moreover, the second case is more reliable because close sets of neighbors are guaranteed. This paper studies the effect of three parameters, namely, cardinality of the common rated items, the rating mean group, and the number of neighbors on the performance of the collaborative recommender system. The contributions of this paper are threefold:

- (1) The notion of users' coverage is introduced as opposed to items' coverage.
- (2) We proposed a modified version for mean difference weights similarity measure.

- (3) Our experiments are implemented on both synthetic and real data sets.

The rest of this paper is organized as follows: a literature review is given in Section 2 and an introduction to collaborative recommender systems is given in Section 3. The effect of the cardinality of the common set on the performance of different similarity measures is introduced in Section 4. Section 5 presents the experimental methodology used for examining many similarity measures while Section 6 discusses the results of the conducted experiments. Finally, we conclude our work in Section 7.

2. Literature Review

Many papers have discussed and proposed many similarity measures, but they fixed the lowest number of the common items in advance and examined their proposals based on that predefined number [3–9]. For example, Al-Shamri [3] examined traditional approaches and proposed a power coefficient as a similarity measure but he assumed the common set size is greater than or equal to five. Breese et al. [9] did empirical analysis for many similarity measures used for collaborative filtering and realized the effect of a low size of common set. Therefore, they suggested default voting for unrated items to enhance the system performance. The same approach is used by many authors to overcome the low number of common items [10]. This is in agreement with the findings of [11] where they showed that participants of their system were more confident in their choices when the recommender had a high rating overlap with the decision maker. However, default voting approach may not reflect the actual user taste for unrated items.

Usually, active users correlate highly with neighbors having very small number of corated items. These neighbors are terrible predictors because they were based on tiny samples of common items. Authors of [12, 13] devalue similarity weights that were based on a small number of corated items by adding a correlation significance-weighting factor of $C_x/50$ to the original weight, where C_x is the number of corated items. Thus, a full contribution will be only for those users having greater than 50 common items with the active user. However, for many cases it is impossible to find this much of common items.

Vozalis and Margaritis [14] tested Pearson correlation coefficient (PCC) with a fixed number of common items of 25. Later, they tested the same system under many threshold values of common items, namely, 1, 10, 20, 30, and 40. They assumed that the best common item threshold is 20. However, they fixed the number of neighbors and did not consider the user mean rating group in their work. Moreover, all experiments they carried out tested only one similarity measure, PCC. Our paper studies the system performance under varying similarity measure and three other parameters, namely, cardinality of the common set, user mean rating group, and the number of neighbors for the active user. These parameters are tested using synthetic and real datasets. Five choices of common items are tested; the first choice assumes

at least one common item and then at least 5, 10, 15, and 20 common items for the remaining choices.

3. Collaborative Recommender System

Many types of recommender systems are proposed based on the way they build user models and their work principle [1]. Systematically, any recommender system passes through five phases to perform its job, namely, data collection, profile formulation, similarity computation, neighborhood set selection, and predictions and recommendations. The effect of each phase on the RS performance depends on its position on the RS stack. The early phases affect the RS more because the performance of the next stages depends on them. The most popular RS is the collaborative recommender system (CRS) which relies on the opinions of possible similar users and hence allows the system to recommend out-of-the-box items [4, 5].

Formally, CRS has M users, $U = \{u_1, \dots, u_M\}$, rating explicitly or implicitly K items, and $S = \{s_1, \dots, s_K\}$, such as news, Web pages, books, movies, or CDs. Each user u_i has rated a subset of items S_i . The declared rating of user u_c for an item s_k is denoted by $r_{c,k}$ and the user's average rating is denoted by m_c [4].

During similarity computation phase, the RS matches the active user to the available database of the training users according to a suitable similarity measure. This value is a measure of how closely two users resemble each other. Once similarity values are computed, the system ranks users according to their similarity values with the active user to extract a set of neighbors for him. After that the CRS assigns a predicted rating to all the items seen by the neighborhood set and not by the active user. The predicted rating, $pr_{x,k}$, indicates the expected interestingness of the item s_k to the user u_x [3, 8].

4. Common Set Cardinality and the Similarity Measure

Similarity computation is the third phase for building a recommender system. Obviously, the accuracy and the reliability of this phase rely largely on the two phases below it. This paper concentrates on the similarity computation phase and assumes that all remaining phases are fixed and accurate except changing the number of neighbors for some experiments. For similarity computation, many similarity measures are used in literature and this paper will examine only three of them. The first similarity measure is Pearson correlation coefficient (PCC) [3–5, 14] which is the most popular similarity function for memory-based CRS:

$$\text{corr}(\mathbf{u}_x, \mathbf{u}_y) = \frac{\sum_{s_k \in S_{xy}} (r_{x,k} - m_x)(r_{y,k} - m_y)}{\sqrt{\sum_{s_k \in S_{xy}} (r_{x,k} - m_x)^2 \sum_{s_k \in S_{xy}} (r_{y,k} - m_y)^2}}. \quad (1)$$

PCC computes the similarity between two users based on the common ratings, S_{xy} , both users have declared:

$$S_{xy} = \{s_1, s_2, s_3, \dots, s_C\}, \quad (2)$$

where C is the cardinality of the common set, $C = |S_{xy}|$. Logically, as the common set becomes large, we expect that the computed similarity reflects the true value of similarity between any two users.

The second similarity measure we examined is called cosine similarity measure [1, 3]. This measure treats each user as a vector in the items' space and finds the cosine of the angle between the two vectors as a measure for the similarity between them:

$$\cos(\mathbf{u}_x, \mathbf{u}_y) = \frac{\sum_{s_k \in S_{xy}} r_{x,k} \times r_{y,k}}{\sqrt{\sum_{s_k \in S_{xy}} r_{x,k}^2} \sqrt{\sum_{s_k \in S_{xy}} r_{y,k}^2}}. \quad (3)$$

Again, the common set is the core for this calculation. A third similarity measure is called mean difference weights (MDW) similarity measure proposed by Bobadilla et al. [7]. They used genetic algorithm (GA) learning for evolving the weights for the rating differences between users. However, these weights can be assumed to be fixed to the mean of each difference weight interval that has been proposed in [7]. For our experiments, we set $w(i) = \langle 1, 0.5, 0, -0.5, -1 \rangle$ as done by Al-Shamri and Al-Ashwal [8]:

$$\text{sim}(\mathbf{u}_x, \mathbf{u}_y) = \frac{\sum_{s_k \in S_{xy}} w(|r_{x,k} - r_{y,k}|)}{|S_{xy}|}. \quad (4)$$

We took into account the point raised by [8] about dividing formula (4) by the difference between the maximum and minimum values of the rating scale. They argued that this factor is not necessary because formula (4) already divides the weights by their number. The numerator cannot exceed $|S_{xy}|$ in any way since $w \in [-1, 1]$. The only effect this factor has is to reduce the similarity values, which in turn will reduce the contribution of each neighbor's rating in the aggregation process.

4.1. Modified Mean Difference Weights Similarity Measure. The mean difference weights similarity measure does not take the user's mean into consideration because it was proposed for learning algorithms like GA. However, a correction factor based on the rating means of the two users in consideration can be added when we fix the weights and rely on the direct calculations without a learning algorithm. In this paper, we propose the following correction factor:

$$\alpha(x, y) = \frac{\min(m_x, m_y)}{\max(m_x, m_y)}. \quad (5)$$

The modified mean difference weights similarity measure is simply formula (4) multiplied by the correction factor that takes the mean of each user into consideration:

$$\text{sim}(\mathbf{u}_x, \mathbf{u}_y) = \alpha(x, y) \frac{\sum_{s_k \in S_{xy}} w(|r_{x,k} - r_{y,k}|)}{|S_{xy}|}. \quad (6)$$

Because of the correction factor, this measure does not give high similarity value for users with one common item if their rating means are different. This similarity measure is called modified mean difference weights (MMD).

4.2. User Coverage. By increasing the cardinality of the common set, we expect that the recommender system could not help all the active users. Therefore, we have to measure the system ability to help the intended users through a measure we call it users' coverage or penetration, which is different from that of items' coverage (this will be discussed later). This metric is defined by the following definition.

Definition 1 (users coverage). The users' coverage of a given recommender system with a minimum predefined cardinality of the common set is the number of users benefitting from the recommender system (who can get neighbors and hence predictions) over all the active users of the system:

$$U_{\text{cov}}^{\text{CS}} = \frac{U_P^{\text{CS}}}{M_A}, \quad (7)$$

where U_P^{CS} is the number of active users getting predictions from the system with a given cardinality of the common set (CS) and M_A is the total number of the active users.

This measure helps us to study the effect of increasing the cardinality of the common set on the usability of the system. Low value of the users' coverage means that the system could not help many users because they have low overlapping between them.

4.3. Sample Dataset for Empirical Analysis. We construct a sample dataset in Table 1 for 19 users with 12 items. The first user is the active user and the remaining 18 users are training users. The zero value for an item indicates that this item has not been rated yet by the user and therefore can be suggested for him. The sample data set has the following specifications:

- (i) It should cover many cardinalities of the common set. Therefore, we take many values 1, 2, 5, 8, and 10.
- (ii) It should cover three rating mean groups (small, medium, and high).
- (iii) The sample data is arranged such that one opposite-minded user and three users with different rating means (low, medium, and high) are available for each cardinality of the common set.
- (iv) The last two users represent opposite-minded users to the active user with two different common items, 8 and 10.
- (v) For each rating mean group, the user with the bigger cardinality of the common set inherits the same items of the user with the lower cardinality to see the effect of increasing the cardinality of the common set without changing the previous set of items.

The similarity values between the active user and the training users of the sample dataset are listed in Table 1

TABLE 1: Sample dataset for 19 users with 12 items.

U_x	C	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	Rating mean	Similarity value			
															PCC	COS	MWD	MMD
u_a	—	3	1	5	4	0	4	5	3	0	2	2	1	3.0	—	—	—	—
u_1	1	0	0	5	0	4	0	0	0	3	0	0	0	4.00	1.00	1.00	1.00	0.75
u_2	1	0	1	0	0	1	0	0	0	2	0	0	0	1.33	1.00	1.00	1.00	0.444
u_3	1	0	0	0	0	3	0	0	2	3	0	0	0	2.67	0.00	1.00	0.50	0.444
u_4	1	0	0	1	0	4	0	0	0	3	0	0	0	2.67	-1.00	1.00	-1.00	-0.89
u_5	2	2	0	5	0	4	0	0	0	3	0	0	0	3.50	0.707	0.987	0.75	0.640
u_6	2	0	1	0	0	1	0	0	0	2	0	0	2	1.50	0.000	0.949	0.75	0.375
u_7	2	0	0	0	0	3	0	0	2	4	0	2	0	2.75	0.707	0.981	0.75	0.688
u_8	2	3	0	1	0	4	0	0	0	3	0	0	0	2.75	-0.99	0.759	0.00	0.00
u_9	5	2	0	5	0	5	0	4	2	5	0	3	0	3.71	0.451	0.975	0.60	0.485
u_{10}	5	2	1	0	0	1	0	0	3	2	1	0	2	1.71	0.311	0.937	0.70	0.400
u_{11}	5	4	0	0	3	3	0	0	2	4	0	2	3	3.00	0.236	0.914	0.50	0.500
u_{12}	5	3	0	1	0	4	0	1	4	3	0	3	0	2.71	-0.86	0.727	0.00	0.00
u_{13}	8	2	2	5	4	5	3	4	2	5	0	3	0	3.50	0.612	0.973	0.625	0.536
u_{14}	8	2	1	0	0	1	3	2	3	2	1	1	2	1.80	0.512	0.922	0.50	0.300
u_{15}	8	4	0	0	3	2	2	4	2	3	2	2	3	2.70	0.369	0.927	0.50	0.450
u_{16}	8	3	5	1	2	4	2	1	4	3	0	3	0	2.80	-0.92	0.681	-0.13	-0.12
u_{17}	10	3	5	1	2	0	2	1	3	0	4	4	5	3.00	-1.00	0.636	-0.20	-0.20
u_{18}	8	0	5	1	2	0	2	1	0	0	4	4	5	3.00	-1.00	0.565	-0.50	-0.50

for the above four similarity measures. The results show the following observations.

- (1) The similarity value of u_1 with u_a is 1 while it is only 0.612 with u_{13} which inherits the same ratings of u_1 and adds more common items with the active user. In general, u_{13} and u_a have 8 common items and out of them four are with positive taste similarity and four are with negative taste similarity. That means the two users agree in their mode for positive and negative items. Hence, we expect higher similarity value than that with u_1 . However, the similarity value is only 0.612. That means low cardinality misleads the RS by giving high similarity values for users having a very little history with the active user.
- (2) Three similarity measures (PCC, COS, and MWD) give a full positive similarity value for u_1 and u_2 with the active user even if they have only one common item with u_a . Moreover, u_5 and u_6 inherit the same ratings of u_1 and u_2 , respectively, but they get lower similarity values by all similarity measures. That means increasing the common set cardinality reduces the similarity values.
- (3) PCC with only one common item will give always the maximum similarity value irrespective of the rating mean values of the two users. This is because the numerator of the formula always equals the denominator for this case.
- (4) MWD gives the same similarity value, 0.75, for u_5 , u_6 , and u_7 . These users have two common items with u_a and belong to three different rating mean groups. This

case is changed with MMD, which takes the mean of the users into consideration.

- (5) u_6 has zero similarity (neutral similarity value in PCC scale) even if it is more close to u_a than u_5 . This case is corrected when the cardinality of the common set is increased to 5 for the same rating mean group.
- (6) PCC can identify opposite-minded users easily where it gives -1 similarity value for u_{17} and u_{18} . That means the two users are dissimilar with the active user. PCC gives also negative results for all opposite-minded users, u_4 , u_8 , u_{12} , and u_{16} .
- (7) The results show that COS cannot identify opposite-minded users where it gives high similarity values 0.636 and 0.565 for u_{17} and u_{18} , respectively. High results are also given for all opposite-minded users, u_4 , u_8 , u_{12} , and u_{16} . These values are very misleading to the RS.

To get an overall view for each rating mean group, Table 2 summarizes the results for each rating mean group. We can see the following observations from this table.

- (i) Usually, similarity values decrease as we increase the cardinality of the common set. This is because of the increased number of items in the user profile.
- (ii) PCC is very sensitive to the cardinality of the common set because it calculates the rating deviation from the rating mean not the rating itself. This deviation spans negative and positive values and becomes more variable as we increase the common items. Hence, it gives different similarity values for different common sets

TABLE 2: Results classification according to the rating mean groups.

Sequence of users	Similarity measure	Sequence of similarity values	Maximum difference
$u_1 \rightarrow u_5 \rightarrow u_9 \rightarrow u_{13}$ (high mean group)	PCC	$1 \rightarrow 0.707 \rightarrow 0.451 \rightarrow 0.612$	0.549
	COS	$1 \rightarrow 0.987 \rightarrow 0.975 \rightarrow 0.973$	0.027
	MWD	$1 \rightarrow 0.75 \rightarrow 0.6 \rightarrow 0.625$	0.4
	MMD	$0.75 \rightarrow 0.64 \rightarrow 0.485 \rightarrow 0.536$	0.265
$u_2 \rightarrow u_6 \rightarrow u_{10} \rightarrow u_{14}$ (low mean group)	PCC	$1 \rightarrow 0 \rightarrow 0.311 \rightarrow 0.512$	1
	COS	$1 \rightarrow 0.949 \rightarrow 0.937 \rightarrow 0.922$	0.078
	MWD	$1 \rightarrow 0.75 \rightarrow 0.7 \rightarrow 0.5$	0.5
	MMD	$0.444 \rightarrow 0.375 \rightarrow 0.4 \rightarrow 0.3$	0.144
$u_3 \rightarrow u_7 \rightarrow u_{11} \rightarrow u_{15}$ (medium mean group)	PCC	$0 \rightarrow 0.707 \rightarrow 0.236 \rightarrow 0.369$	0.707
	COS	$1 \rightarrow 0.981 \rightarrow 0.914 \rightarrow 0.927$	0.086
	MWD	$0.5 \rightarrow 0.75 \rightarrow 0.5 \rightarrow 0.5$	0.25
	MMD	$0.444 \rightarrow 0.688 \rightarrow 0.5 \rightarrow 0.45$	0.244
$u_4 \rightarrow u_8 \rightarrow u_{12} \rightarrow u_{16}$ (opposite-minded group)	PCC	$-1 \rightarrow -0.99 \rightarrow -0.86 \rightarrow -0.92$	-0.14
	COS	$1 \rightarrow 0.759 \rightarrow 0.727 \rightarrow 0.681$	0.319
	MWD	$-1 \rightarrow 0 \rightarrow 0 \rightarrow -0.13$	-1
	MMD	$-0.89 \rightarrow 0 \rightarrow 0 \rightarrow -0.12$	-0.89

and these values decrease by increasing the cardinality of the common set. Accordingly, many users may get unfair similarity values, hence increasing or reducing their contribution to the active user based on these values.

- (iii) PCC performance with opposite-minded users is reasonable as it always gives negative values with a low deviation value. That means PCC has good capabilities for capturing opposite-minded users and hence can easily prevent opposite-minded users from being neighbors for a given active user.
- (iv) COS is less sensitive to the common set cardinality. It always gives high values even for opposite-minded users like group four in Table 2. These values decrease a little bit by increasing the cardinality of the common set. That means this similarity measure is not good for capturing neither like-minded users nor opposite-minded users.
- (v) MWD and MMD give similarity values that have less deviation than that of PCC. The weakness of these two measures lies in their ability for capturing opposite-minded users. They give zero similarity values for u_8 and u_{12} and this value represents a neutral case. This weakness can be alleviated by counting only those users having greater than zero similarity value.

5. Experiments

This section discusses the methodology of choosing the dataset for our experiments, the way of dividing the dataset into training and test subsets, and the metrics used for evaluating the system performance. The selected dataset should reflect different mean groups and different cardinalities of the

common set to study both effects. The following subsections analyze the MovieLens dataset and select the experiments dataset for this paper.

5.1. One Million MovieLens Dataset. One million MovieLens dataset consists of 1000209 ratings assigned by 6040 users on 3900 movies [15]. This paper studies three parameters of the collaborative recommender system, namely, common set cardinality, the user rating mean group, and the number of neighbors for a given active user which have a direct effect on the performance of the similarity measure. Therefore, we further subdivide each dataset into three data subsets according to the rating mean group of each user. We consider three rating mean groups, low mean group (between 1 and 2.4), medium mean group (between 2.4 and 3.4), and high mean group (between 3.4 and 5). Accordingly, nine data subsets can be obtained in Table 3.

The results of Table 3 show that only less than 1% of the total users have low rating mean. Most users usually rate items highly where 78.15% of the total users have high rating. Hence, this dataset represents a skewed real dataset.

5.2. Experimental Dataset. We conduct our experiments using an elected dataset from the 1M MovieLens dataset [15]. To test the similarity measures under all possible rating mean groups and all possible rating behaviors, we select from Table 3 the group that has the lowest number of users as the benchmark. Therefore, our dataset takes the subsets of the low rating mean group as the reference for the other rating mean groups. All users of this group belong to the low rating mean group but reflect three different rating behaviors, low number of ratings (13), medium number of ratings (15), and high number of ratings (8). Accordingly, we randomly

TABLE 3: Some statistics of 1 million MovieLens dataset.

Category	Specifications		Number of ratings	Number of users	Percentage	Total percentage
	Ratings range	Mean range				
C11	[1, 100]		565	13	0.2152	0.596
C21	[101, 500]	[0, 2.4[	25256	15	0.2483	
C31	>500		130686	8	0.1325	
C12	[1, 100]		2759	521	8.6258	21.2582
C22	[101, 500]	[2.4, 3.4[	142875	580	9.6026	
C32	>500		409259	183	3.0298	
C13	[1, 100]		5127	2597	42.9967	78.1457
C23	[101, 500]	[3.4, 5]	137292	1918	31.7550	
C33	>500		146390	205	3.3940	

select 36 users from the other two rating mean groups for our dataset. Hence, we have 36 users by 3 rating mean groups, or equivalently 108 users.

For experiments, we use leave-one-out cross-validation, which uses each time one user of the dataset as the test user and the remaining users as the training users. Thus each user is used the same number of times for training and once for testing. Thus the number of total users, training users, and active users are $M = 108$, $M_T = 107$, and $M_A = 108$, respectively.

During the testing phase, the set of active user declared ratings, S_a , is divided randomly into two disjoint sets, namely, training ratings (S_a^{TR}) (20%) and test ratings (S_a^{TE}) (80%) such that $S_a = S_a^{TR} \cup S_a^{TE}$. The RS treats S_a^{TR} as the only declared ratings while S_a^{TE} are treated as unseen ratings that the system would attempt to predict for testing the RS performance [3].

5.3. Conducted Experiments. We conduct four experiments on the 108 users' dataset, the first experiment uses Pearson correlation coefficient (formula (1)) for the similarity computation and we call it Correlation-Based RS (CBRS). The second experiment uses cosine vector similarity measure (formula (3)) for the similarity computation and we call it Cosine Vector RS (CVRs). The third experiment uses mean difference weights similarity measure (formula (4)) for the similarity computation and we call it Difference Weights RS (DWRS). The fourth experiment uses modified mean difference weights similarity measure (formula (6)) for the similarity computation and we call it MDRS.

Each experiment is performed five times with different cardinalities of the common set. First, we assume the cardinality of the common set to be greater than or equal to one, and then we increase it by five each time until we reach 20. Table 4 illustrates different cardinalities of the common set and their codes.

5.4. Evaluation Metrics. The performance of each examined CRS is evaluated using items' coverage, percentage of the correct predictions (PCP), and mean absolute error (MAE) [3, 16, 17]. Items' coverage measures the percentage of items for which a CRS can provide predictions, that is, the number

TABLE 4: Common sets and their cardinalities.

Common set	Code	Size
1	CS1	$C \geq 1$
2	CS2	$C \geq 5$
3	CS3	$C \geq 10$
4	CS4	$C \geq 15$
5	CS5	$C \geq 20$

of items for which the CRS can generate predictions for a given active user over the total number of unseen items for the same user. The total items' coverage over all active users is

$$\text{Coverage} = \frac{\sum_{i=1}^{M_A} N_i^p}{\sum_{i=1}^{M_A} |S_i^{TE}|}. \quad (8)$$

Here, N_i^p is the total number of predicted items for user u_i and M_A is the total number of the active users. Low item coverage value indicates that the CRS will not be able to assist the user with many of the items he has not rated.

The PCP is the percent of the correctly predicted items by the system to the total number of items in the test ratings set of the active user. The set of correctly predicted items for a given active user and the total PCP over all the active users are defined by the following formulae [18]:

$$\begin{aligned} \text{CorrectSet}(u_a) &= \{s_k \mid s_k \in S_a^{TE}, \text{pr}_{a,k} = r_{a,k}\}, \\ \text{PCP} &= \frac{\sum_{i=1}^{M_A} |\text{CorrectSet}(u_i)|}{\sum_{i=1}^{M_A} |S_i^{TE}|} \times 100\%. \end{aligned} \quad (9)$$

The MAE measures the deviation of predictions generated by the CRS from the true ratings specified by the active user [3, 16, 17]. The MAE over all the active users (M_A) is given by formula (10). Lower MAE corresponds to more accurate predictions of a given CRS:

$$\text{MAE} = \frac{1}{M_A} \sum_{i=1}^{M_A} \left(\frac{1}{|S_i^{TE}|} \sum_{k=1}^{|S_i^{TE}|} |\text{pr}_{i,k} - r_{i,k}| \right). \quad (10)$$

TABLE 5: Users' coverage of CBRS, CVRS, DWRS, and MDRS.

C	CBRS		CVRS		DWRS		MDRS	
	Number	%	Number	%	Number	%	Number	%
CS1	106	98.15	108	100	108	100	108	100
CS2	98	90.74	101	93.52	99	91.67	99	91.67
CS3	80	74.07	80	74.07	80	74.07	80	74.07
CS4	71	65.74	71	65.74	70	64.82	70	64.82
CS5	54	50	56	51.85	55	50.93	55	50.93

TABLE 6: PCP and coverage of the examined RS with different common set sizes.

	N_x	PCP					Coverage					MAE				
		CS1	CS2	CS3	CS4	CS5	CS1	CS2	CS3	CS4	CS5	CS1	CS2	CS3	CS4	CS5
CBRS	10	10.15	23.22	27.83	30.18	32.09	32.05	69.62	80.08	84.95	87.99	7.02	1.66	1.09	0.9	0.62
	30	31.7	34.38	35.34	35.71	36.43	89.02	94.46	95.81	95.78	96.61	1.23	0.97	0.77	0.71	0.49
	50	35.53	36.11	36.24	36.25	36.74	96.49	97.19	97.25	96.67	97.12	1	0.93	0.75	0.7	0.48
CVRS	10	6.74	20.24	27.44	30.5	31.98	22.98	60.33	78.13	85.05	87.39	9.02	2.15	1.08	0.85	0.63
	30	26.02	33.92	35.23	35.59	36.05	75.77	93.52	95.77	96.19	96.46	1.84	1.05	0.74	0.68	0.52
	50	34.27	35.49	36.13	35.98	36.6	94.46	97.04	97.77	97.48	97.44	1.08	1	0.71	0.66	0.5
DWRS	10	18	26.53	29.94	31.65	32.95	51.59	74.52	82.79	86.66	88.74	3.62	1.44	0.99	0.84	0.6
	30	34.2	35.45	36.23	36.09	36.54	91.71	94.41	95.49	95.79	96.34	1.12	0.95	0.76	0.69	0.5
	50	36.48	36.53	36.83	36.49	37.15	96.17	96.88	97.1	97.1	97.39	1.01	0.91	0.74	0.68	0.49
MDRS	10	23.08	29.22	31.46	31.78	32.95	63.93	79.67	85.41	87.43	89.16	2.41	1.3	0.95	0.83	0.6
	30	34.84	35.65	36.13	36.01	36.68	93.04	94.74	95.6	95.83	96.27	1.09	0.94	0.76	0.69	0.5
	50	36.47	36.48	36.75	36.64	37.45	96.37	96.81	97.06	97.09	97.39	1.01	0.91	0.74	0.68	0.49

The predicted rating, $pr_{x,k}$, is usually computed as an aggregate of the ratings of u_x 's neighborhood set for the same item s_k . The common prediction formulae are [1, 3]

$$pr_{x,k} = \frac{\sum_{u_y \in N_x} \text{sim}(\mathbf{u}_x, \mathbf{u}_y) \times r_{y,k}}{\sum_{u_y \in N_x} |\text{sim}(\mathbf{u}_x, \mathbf{u}_y)|}, \quad (11a)$$

$$pr_{x,k} = m_x + \frac{\sum_{u_y \in N_x} \text{sim}(\mathbf{u}_x, \mathbf{u}_y) (r_{y,k} - m_y)}{\sum_{u_y \in N_x} |\text{sim}(\mathbf{u}_x, \mathbf{u}_y)|}, \quad (11b)$$

where N_x denotes the set of neighbors for u_x who have rated item s_k and m_y is the average rating of user u_y . We used priority-based prediction formula used in [3, 8] where formula (11b) is used first. If its predicted rating is out of the rating range, then formula (11a) is used. For all experiments, the neighborhood set size N_x is varied from 10 to 50 by a step size of 20 each time, $N_x = \{10, 30, 50\}$.

6. Analysis of the Results

The results show that many active users could not gain a benefit from the systems when the cardinality of the common set is increased. The users' coverage values for all systems are listed in Table 5. The values of the users' coverage of CBRS are somehow less than that of CVRS because CBRS similarity values can be positive or negative. Usually, negative similarity values are discarded as they are for dissimilar users with

the active user and hence the number of users with positive similarity values will be less. The users with negative similarity values are always discarded as they are not trusted users.

The results show that the users' coverage is high for low cardinality values and starts decreasing by increasing the cardinality of the common set for all systems. The lowest value is $U_p^5 = 50$ for CBRS, which indicates that only 50% of the active users can get recommendations when the cardinality of the common set is 5. Users' coverage values of MDRS are similar to that of DWRS as they differ only in the correction factor.

Actually, the similarity measure is a crucial part for any CRS. However, our experiments show that this impact depends largely on the cardinality of the common set. The results of all systems for all metrics, three neighborhood set sizes, and five cardinalities of the common set are listed in Table 6. These results show that we cannot compare different similarity measures without taking into account the cardinality of the common set.

Horizontally, the results are improved with increasing the cardinality of the common set and also vertically with increasing the neighborhood set sizes. This improvement gets narrower as we move in both directions. We may conclude that implementing RS with high value of neighborhood set size and low cardinality of common sets hides the effectiveness of the similarity measure. In general, increasing the cardinality of the common set has two contradictory effects on the CRS performance. Negatively, it reduces the users' coverage and hence the system could not help as many users as

TABLE 7: Improvement percentages of the examined RSs for PCP, coverage, and MAE.

N_x	PCP				Coverage				MAE			
	CBRS	CVRS	DWRS	MDRS	CBRS	CVRS	DWRS	MDRS	CBRS	CVRS	DWRS	MDRS
10	216.23	374.31	83.07	42.63	174.49	280.31	72.01	39.47	91.21	92.98	83.34	75.06
30	14.93	38.54	6.83	5.27	8.53	27.3	5.05	3.47	60.35	71.91	55.63	54.34
50	3.4	6.79	1.83	2.67	0.66	3.16	1.27	1.07	51.97	53.32	51.95	51.66

TABLE 8: Best system in terms of PCP, coverage, and MAE.

N_x	PCP					Coverage					MAE				
	CS1	CS2	CS3	CS4	CS5	CS1	CS2	CS3	CS4	CS5	CS1	CS2	CS3	CS4	CS5
10	MDRS					MDRS					MDRS				
30	MDRS		DWRS		MDRS	MDRS			CBRS		MDRS		CVRS	CBRS	
50	DWRS		MDRS			CBRS		CVRS		CBRS		MDRS	CVRS	CBRS	

before. Positively, it enhances the CRS ability for identifying true neighbors and hence increases the system accuracy.

For more clarifications, we computed the improvement percentages between the results of CS1 and CS5 in terms of PCP, coverage, and MAE for all systems. For comparison purposes, we used the following two formulae for measuring increase improvement percentages (for PCP and coverage) and decrease improvement percentages (for MAE) [18]:

$$P_x^+ = \frac{x_{\text{new}} - x_{\text{old}}}{x_{\text{old}}} \times 100\%, \quad (12a)$$

$$P_x^- = \frac{x_{\text{old}} - x_{\text{new}}}{x_{\text{old}}} \times 100\%. \quad (12b)$$

Table 7 summarizes the improvement percentages of PCP, coverage, and MAE for CBRS, CVRS, DWRS, and MDRS. The results show that there are large improvement percentages between systems using low cardinalities of the common set and those using large cardinalities. The improvement percentages are very high for $N_x = 10$ especially for CVRS where they reach 374.31%, 280.31%, and 92.98% for PCP, coverage, and MAE, respectively. That means close neighbors and consequently predictors are found with CS5. Actually, we expect good results for a given active user if he could find a close set of neighbors even if the number of neighbors is low ($N_x = 10$). Low cardinality of the common set does not allow the system to elect representative neighbors and thus the performance is low. The improvement percentages decrease as we increase the neighborhood set sizes. This is a usual result because the set of neighbors gets wider and therefore the chance is more to get close neighbors than before.

Another important point is that MDRS gets the lowest improvement percentages (only 42.63%) even if its overall performance is better than the other ones. This indicates that this similarity measure is able to elect representative neighbors from the very beginning and hence its improvement is slow.

For more clarity, Table 8 lists the best system in terms of both neighborhood set size and the cardinality of the common set while Figures 1–3 depict the results of all

cardinalities of the common set for the two extreme values of neighborhood set size; that is, $N_x = 10$ and $N_x = 50$. The results show the following points.

- (i) MDRS is the best in terms of all metrics for $N_x = 10$ and all cardinality values of the common set while CVRS is the worst one. That means MDRS generates true neighbors from the very beginning.
- (ii) CBRS results go faster to that of MDRS than CVRS by increasing the cardinality of the common set.
- (iii) In terms of PCP, CVRS is the worst for all N_x and all cardinalities of the common set.
- (iv) As we increase N_x or the cardinality of the common set, the performances of all systems become very close. This indicates that the effect of similarity measure becomes less important compared to other factors.
- (v) Increasing N_x for CS3 to CS5 gives some advantage to CBRS or CVRS. However, this advantage is very small compared to the results of the other systems. In this situation, other factors play more effectively for the recommender system.
- (vi) Increasing N_x hides the ability of the similarity measure to select good neighbors. In this case, the contributions of so many neighbors compensate each other and consequently hide the differences between the similarity measures.
- (vii) Vertically with CS1 and CS2, MDRS and DWRS are the best in terms of PCP and MAE. At this stage, many neighbors who have been ranked at the top for CBRS and CVRS may have only one or two items in common with the active user. These neighbors are not true neighbors as their overlapping with the active user is very small.

7. Conclusions

One important way for enhancing the CRS accuracy is to select a similarity measure that produces a close set of

FIGURE 1: PCP of the examined RSs for different cardinalities of the common set and $N = 10$ and $N = 50$. FIGURE 2: Coverage of the examined RSs for different cardinalities of the common set and $N = 10$ and $N = 50$. FIGURE 3: MAE of the examined RSs for different cardinalities of the common set and $N = 10$ and $N = 50$.

neighbors. The results show that the effect of the similarity measure depends on the cardinality of the common set, as the system accuracy gets better with high values of this set. Moreover, this effect becomes less significant as we increase the number of the neighbors for the active user. In this case, large number of neighbors let them compensate each other and hence the differences between systems become very low.

The modified mean difference weights similarity measure outperforms other systems for many cases as it takes the ratings mean into consideration. In general, increasing the cardinality of the common set has two contradictory effects on the CRS performance. Negatively, it reduces the users' coverage and hence the system could not help as many users as before. Positively, it enhances the CRS ability for

identifying true neighbors and hence increases the system accuracy. The results show that some similarity measures outperform others for a specific cardinality of the common set. However, the role may change with another cardinality. For example, MDRS performs better than both CBRS and CVRS for CS1 and $N_x = 10$. However, the same system lost its advantage with CS4, CS5, and $N_x = 50$ in terms of MAE. This indicates that the performance of similarity measures differ for different cardinalities of the common sets.

Many approaches have been done for alleviating the effect of small sets of common items, some of them try to predict missing items, and others try to devalue the contribution of the corresponding users. However, our view for future work is to propose new techniques that rely directly on the actual user data without any predictions to fill the missing values.

Competing Interests

The author declares that there are no competing interests.

References

- [1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013.
- [2] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.
- [3] M. Y. H. Al-Shamri, "Power coefficient as a similarity measure for collaborative recommender system," *Expert Systems with Applications*, vol. 41, pp. 5680–5688, 2014.
- [4] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communication of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.
- [5] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., vol. 4321 of *Lecture Notes in Computer Science*, pp. 291–324, Springer, Berlin, Germany, 2007.
- [6] M. Y. H. Al-Shamri and K. K. Bharadwaj, "A hybrid preference-based recommender system based on fuzzy concordance/discordance principle," in *Proceedings of the third Indian International Conference on Artificial Intelligence (IICAI '07)*, pp. 301–314, Pune, India, 2007.
- [7] J. Bobadilla, F. Ortega, A. Hernando, and J. Alcalá, "Improving collaborative filtering recommender system results and performance using genetic algorithms," *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1310–1316, 2011.
- [8] M. Y. Al-Shamri and N. H. Al-Ashwal, "Fuzzy-weighted similarity measures for memory-based collaborative recommender systems," *Journal of Intelligent Learning Systems and Applications*, vol. 6, no. 1, pp. 1–10, 2014.
- [9] J. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI '98)*, pp. 43–52, Morgan Kaufmann, Madison, Wis, USA, July 1998.
- [10] A. Bellogín and A. P. de Vries, "Understanding similarity metrics in neighbour-based recommender systems," in *Proceedings of the 4th International Conference on the Theory of Information Retrieval (ICTIR '13)*, pp. 48–55, Copenhagen, Denmark, October 2013.
- [11] P. Bonhard, C. Harries, J. McCarthy, and M. A. Sasse, "Accounting for taste: using profile similarity to improve recommender systems," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '06)*, pp. 1057–1066, Montréal, Canada, April 2006.
- [12] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Information Retrieval*, vol. 5, no. 4, pp. 287–310, 2002.
- [13] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Foundations and Trends in Human-Computer Interaction*, vol. 4, no. 2, pp. 81–173, 2010.
- [14] E. G. Vozalis and K. G. Margaritis, "Recommender systems: an experimental comparison of two filtering algorithms," in *Proceedings of the 9th Panhellenic Conference in Informatics (PCI '03)*, pp. 152–166, Thessaloniki, Greece, 2003.
- [15] <http://www.movielens.umn.edu>.
- [16] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.
- [17] E. Vozalis and K. Margaritis, "Analysis of recommender systems' algorithms," in *Proceedings of the 6th Hellenic-European Conference on Computer Mathematics and Its Applications (HERCMA '03)*, Athens, Greece, 2003.
- [18] M. Y. Al-Shamri, "User profiling approaches for demographic recommender systems," *Knowledge-Based Systems*, vol. 100, pp. 175–187, 2016.