Normalization-based Neighborhood Model for Cold Start Problem in Recommendation System

Aafaq Zahid[1], Nurfadhlina Mohd Sharef[1], and Aida Mustapha[2]
[1]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia
[2]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Malaysia

Abstract: *Existing approaches for Recommendation Systems (RS) are mainly based on users' past knowledge and the more popular techniques such as the neighborhood models focus on finding similar users in making recommendations. The cold start problem is due to inaccurate recommendations given to new users because of lack of past data related to those users. To deal with such cases where prior information on the new user is not available, this paper proposes a normalization technique to model user involvement for cold start problem or user likings based on the details of items used in the neighborhood models. The proposed normalization technique was evaluated using two datasets namely MovieLens and GroupLens. The results showed that the proposed technique is able to improve the accuracy of the neighborhood model, which in turn increases the accuracy of an RS.*

Keywords: *Recommender system, cold start, collaborative filtering, normalization.*

Received May 3, 2017; accepted December 17, 2017
https://doi.org/10.34028/iajit/17/3/1

1. Introduction

The advent of Recommendation Systems (RS) across numerous applications such as the news recommendation [3], e-commerce and online shopping [4], and restaurant recommendation [6] led to significant research on RS since the 1990s. RS developed subsequently over time using different techniques for their core recommendation engine which comprised of the Content-Based (CB) technique, Collaborative Filtering (CF) technique, Matrix Factorization (MF) technique, demographic and semantic technique [7]. These techniques provided various ways to perform the recommendation where most of the RS worked on building user profile by updating the information over time as the user used different items and provided feedback to the system. Profile updating helped the system to update the user's preferences tailoring the results to specific users. Although user profiling is fairly accurate, the problem arises when the user first joins the system and his/her profile is empty. This is known as cold start problem whereby the RS are not capable of recommending items to the user due to the absence of his/her prior information [12].

The cold start problems are evaluated under two scenarios. The first scenario is the partial cold start problem in which a limited amount of ratings are available for the new users. The second scenario is the pure cold start problem whereby no prior ratings are available for the new user. In order to deal with the cold start problem, the literature can be categorized into three groups:

1. Using additional user data such as demographic information, social information or questionnaire to place user in appropriate cluster before recommendation is mapped using fuzzy algorithms [12].
2. Putting the users in the most prominent group based on their global preferences and combining them into new clusters [15].
3. Combining methods to perform prediction by using different methods to calculate the ratings such as infusing the fuzzy sets into association rule mining [26].

The process of adding new demographic data or ratings to deal with cold start problem often leads to another limitation as large noisy data piles up after the user information is transformed into new clusters [17]. Limited research in RS uses the normalization technique to reduce the huge data into the specific range that helps to control the gray-sheep problem by rescaling the values [13, 15]. Existing studies combine normalization with neighborhood models (CF and MF) by performing data normalization before constructing the neighborhood models. However, the process is very general and does not intend to solve any problem in the RS. These methods use the normalization to avoid the big difference in the final rating of RS. They also overlook the effect of the ratings of the clusters in which the user lies.

In this study, normalization technique is used in combination with CF and MF methods to solve the cold start problem in RS by considering variables that are not directly related with user profile but have enough significance towards the final recommendation to users. The variables are:

1. Average ratings of the items.
2. Average ratings of the global user for a particular item.
3. Average rating of the cluster user after initial few ratings. Hence, the contribution of this paper is three-fold. First, this paper addresses the effect of normalization on MF. Next, it reports the effect of normalization on both the partial cold start problem and the pure cold start problem. While the average ratings of items and users are important in normalization, the research also considers the cluster averaging and item group averages in order to further normalize the ratings to target the cold start problem.

The remainder of this paper is organized as follows. Section 2 reviews literature relating to recommendation systems. Section 3 presents the proposed normalization technique in combination with the neighborhood model. Section 4 presents the experimental setup and results and section 5 focus on the conclusion of the research and direction of future work.

2. Related Work

Normalization is a standard technique employed in data mining tasks on averaging users' ratings. It is used in the proposed algorithm to avoid the big difference in the final rating of RS and addresses cold start problem through combining CF and MF techniques.

2.1. Collaborative Filtering

CF is a technique that relies on "word of mouth" [18] and requires interaction between users. The opinions of other users are taken into account in providing the personalized recommendation to the users [23, 24]. CF helps in dividing the huge rating matrices identifying similar users and similar items [1]. The state of items and users evolves over time [11]. In other words, CF builds a user's history which helps in providing the recommendations. Prediction approach in CF is divided into item-based CF and user-based CF. The item-based CF predicts by finding similar items to the items rated by the user and calculates a weighted mean from the ratings to predict recommendation for the new item. Figure 1 shows an example of how an item-based CF finds similar books to 'Book A' which User A has already rated and calculates a weighted mean of those ratings to predict the recommendation of 'Book A'. From Figure 1, 'Book A', 'Book B' and 'Book C' are already in the same cluster, while 'Book D' and 'Book E' are in the other cluster. As User A likes 'Book B' and 'Book C', the system also recommends 'Book A' to him. At the same time, given that User A dislikes 'Book E', 'Book D' will not be recommended to the user.

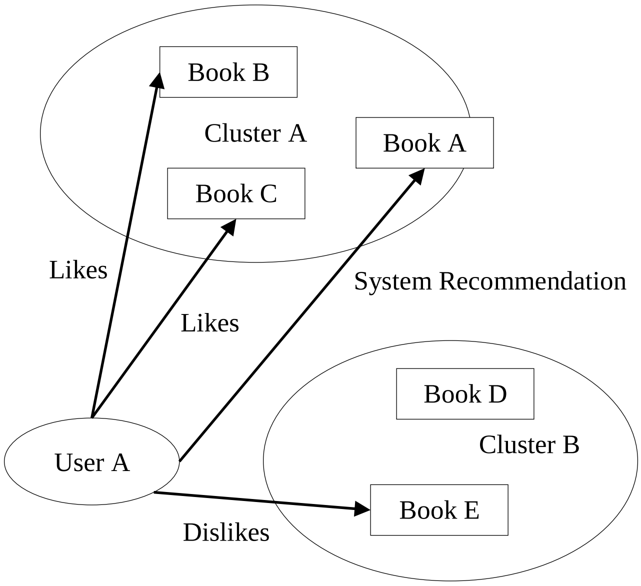

Figure 1. Example of item-based Collaborative Filtering (CF).

User-based approach finds similar users who have rated items that the current user has rated and suggests items that similar users like. Figure 2 shows an example of recommendations made by user-based CF technique. User A and User B are considered similar because of both users like 'Book B' and 'Book C'. Therefore, if User A likes 'Book A', the system will automatically recommend 'Book A' to User B on the basis of similar preferences.

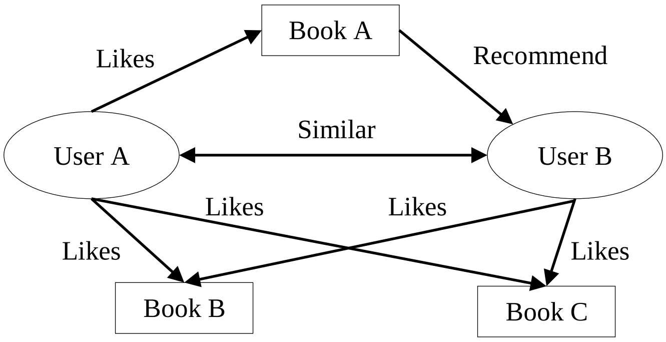

Figure 2. Example of user-based Collaborative Filtering (CF).

An important step in CF is to divide or to cluster the items into groups. The most common clustering technique used is the K-means algorithm, which creates k clusters from a randomly chosen value at the center, called centroids, of these clusters. The algorithm calculates the distance between the users or items, depending on which method is being followed, with the centroids for each cluster. Then, the user is grouped or assigned into the cluster where the distance is the smallest. The shortest distance between two points implies that the items within the cluster are similar to the current user. Nonetheless, CF performance highly depends on the user profile, which will lead to poor performance in the case of cold start scenario.

2.2. Matrix Factorization

MF is a relatively more accurate recommendation technique [20]. It takes the most global approach followed by CF. For example, if User A likes the movie "The Ugly Truth", which has Romantic and Comedy elements, the system will imply that User A also likes "No Strings Attached" which also has Comedy and Romantic elements. MF is usually performed using Single Value Decomposition (SVD) and is used slightly differently from its use in standard algebra. In order to illustrate the working of SVD, consider Table 1.

Table 1. Original user ratings for Movies in the form of a matrix.

	Movie-1	**Movie-2**	**Movie-3**	**Movie-4**
U1	5	3	-	1
U2	4	-	-	1
U3	1	1	-	5
U4	1	-	-	4
U5	-	1	5	4

Table 1 shows original user ratings for a list of users (U1 to U5) with respect to four types of movies (Movie-1 to Movie-4) represented in the form of factor vectors. The cells in the matrix with value '-'indicates that the movie has not been rated by the user, and the recommendation process will predict the possible value in order to determine whether the movie is suitable to be recommended to the user or not. To compute the SVD in the matrix shown in Table 1, first, the empty slots are replaced with zeroes to make it a valid matrix.

This is the matrix of *m x n*, so factor vectors can be described as 2 matrices of *m* x *k* and *n* x *k* whose multiplication produces the original matrix of *m* x *n*. Note that *k* is an arbitrary number known as the latent feature. MF starts by making two random matrices of *m x k* and *n x k* and then multiplies the matrices in order to get *m x n* matrices. Table 2 shows the matrix after multiplication of *k* and *n* for the first iteration.

Table 2. User ratings for Movies after the first iteration.

	Movie-1	**Movie-2**	**Movie-3**	**Movie-4**
U1	2	2.1	*1.9*	4.9
U2	3	*1.2*	*4.2*	3
U3	1.1	0.8	*1.4*	4.5
U4	3.2	*0.3*	*3.9*	1.2
U5	*1.6*	1.9	1.9	2.1

The non-zero values in Table 2 show the user rating for the movies, whereby the difference is calculated between the current matrix and original matrix. This difference is called the error. For example, given that the user U1 is named Tom and the movie, Movie-1 is Mr. Bean. In the original matrix of Table 1, the value for this cell is 5 but in the current matrix of Table 2, the value is 2. This gives an error of 3. The errors are then adjusted in a resultant matrix, which is then multiplied again to get another matrix until all the errors are gone or minimized. For this particular example, the final matrix is shown in Table 3. The values underlined and marked italics are the ratings assigned by MF. Finally, the dot product of Tom's and Mr. Bean's rating vector can be predicted.

Table 3. Final matrix.

	Movie-1	**Movie-2**	**Movie-3**	**Movie-4**
U1	4.97	2.98	*2.18*	0.98
U2	3.97	*2.40*	*1.97*	0.99
U3	1.02	0.93	*5.32*	4.93
U4	1.00	*0.85*	*4.59*	3.93
U5	*1.36*	1.07	4.89	4.12

MF is a step towards the memory-based neighborhood models. In recent years, the MF technique has been gaining popularity due to its compact results and better recommendations. The ability to adapt the learning in MF helps it to predict a better outcome for the particular user [20]. Hence, MF is able to generate recommendations with higher accuracy. But as the descendant of neighborhood models, MF still suffers from the cold start problem. Shardanand and Maes [22] attempted to solve the cold start problem in MF by using additional personality measures such as used in CF [22]. However, in principle, additional data increases the processing time, making MF computationally very expensive despite its good results [23]. Apart from the cold start problem, MF also suffers from data scalability where all the items in the matrix are treated equally without any preference for any specific items [24].

2.3. Normalization

The recommendation system in [12] performed data normalization before the clustering process takes place. The technique divided the prediction into several steps. For example, the user Tom gives the movie, Mr. Bean, a rating of 4.1. The normalization process for this rating can be broken down as follows.

- Baseline Rating: Baseline rating for movie Mr. Bean (can be the average of all the ratings, which for current example is 3.9).
- User-Specific Effect: Tom gave 0.9 less than the average rating.
- Movie-Specific Effect: Mr. Bean performed above average on the box office so it gets a rating of 0.7 higher than other movies.
- Specific Interaction: This is less predictable. For example, Rowan Attkinson, the leading actor in Mr. Bean, is in a comedy movie, and therefore, this combination should get a rating of 0.4 higher than the average ratings.

By adding all these values, the normalization process will give the final rating of 4.1 for Tom (3.9-0.9+0.7+0.4), which contends with the actual model prediction of 4.1 rating. In addition, the recommendation can be improved as more factors are added to the system:

- The number of days since the movie was first released. Continuing with the example, Tom might be waiting for the movie to be released as he might be the biggest fan of the movie and thus, watched it on the first or second day from the release date.
- The number of people who rated this movie.
- Overall rating of the movie.

Consistent with the findings in [13], this example shows that normalization is the simplest process that can be performed on the data while producing good recommendations [13]. Normalization is very effective due to its simplicity and minimum errors. Normalization works on baseline ratings of both users and items, which helps in resolving cold start problems without requiring extensive data. Any missing rating in the normalization technique is considered zero. Hence, it does not affect the final outcome. Normalization also avoids mood changes as it works on the parameters to recommend items on moods. Normalization considers user profile and average items in determining the user ratings, which in turn avoids the stability versus plasticity as well as the gray-sheep problems. Furthermore, normalization takes the average rating, which also avoids difference of opinion to some level.

However, despite all its strengths, normalization has an inherent weakness as a standalone system. It requires too many parameters in order to provide satisfying ratings. The technique also works on generalization as it relies on average ratings and does not consider any other user ratings.

2.4. Hybrid Filtering

The techniques defined above are individual, but some authors have combined more than one techniques to produce better recommendations [8, 21]. The most commonly used techniques combined are CF and CB [21]. Schiaffino and Amandi [21] said that by combining CF and CB, best results possible can be achieved. Burke [8] however has changed views and used CF with Knowledge Based System (KBS). A detailed discussion has been given by Burke [7] of possible and proposed hybrid techniques. We will focus on pros and cons of the hybrid techniques available.

CF and CB are the most common techniques combined in the hybrid [16]. The advantage of merging CF and CB in hybrid is that it overcomes the shortcomings in both techniques by merging them. As CB has stability versus plasticity problem, CF helps it with neighborhood model and that problem does not exist in resulting hybrid. The main problem in CF is gray sheep problem. By considering the user personal preferences in the CB this hybrid technique solves this problem. However, the problem that still exists in this hybrid model is cold start problem as both techniques possess this problem.

The hybrid of CF and CB has its short comings in terms of cold start problem. Chandak, Girase, and Mukhopadhyay [9] came up with another hybrid to solve the cold start problem with CF, CB and Demographic Filtering (DF). In this hybrid technique, they combine the DF with CF and CB to overcome the cold start problem. While the technique somehow minimizes the cold start and stability versus plasticity problems, the gray sheep problem still exists as CF was used as main algorithm while the other algorithms were used to short list the items recommended by CF.

After the introduction of MF, researchers started looking into the hybrid methods with MF [2]. One such technique is introduced by Vizine, Luiz and Hruschka [2, 22] whereby they combine the MF with DF. As discussed, MF also has cold start problem so this technique is combined to fix the cold start problem by the help of DF. Although the combination of MF and DF is adequate, the cold start problem solved by the DF for MF is not as appropriate and gives rise to the problem of difference of opinion for the people from same region or gender.

Researchers have also applied the normalization on CF [13]. This hybrid technique was designed in the way that data was firstly normalized and then given to CF. The hybrid has problems as normalization is not able to solve the gray sheep problem in CF because of the adjustment of the system.

Bell and Koren [5] used normalization in neighborhood based MF. The normalization used in MF was to avoid items that have huge rating difference. Koren and Bell [14] only consider the average rating of the user and average rating of the items in determining the prediction of the rating to the user. The normalization is performed before performing MF to avoid big errors in the data. However lesser variables used for the normalization proved it to be less effective towards the problems in RS.

In this paper DF, CB and KBS methods are ignored because although they pose positive effects, their negative aspects still cannot be avoided after a certain point. For this reason this study chooses Normalization and CF.

3. Normalization-based Neighborhood Model

This paper proposes two combinatorial approaches to normalization with Content-based Filtering (CF) as well as MF techniques. The algorithms are called Content-based Filtering Normalization (CFN) and Matrix Factorization Normalization (MFN) respectively. The design of the proposed algorithms is different from [13] since normalization is used as the base algorithm instead of CF and MF in [13]. Figure 3 shows the design of the first algorithm, CFN, in which the average of all clusters is fed to the normalization process through CF.

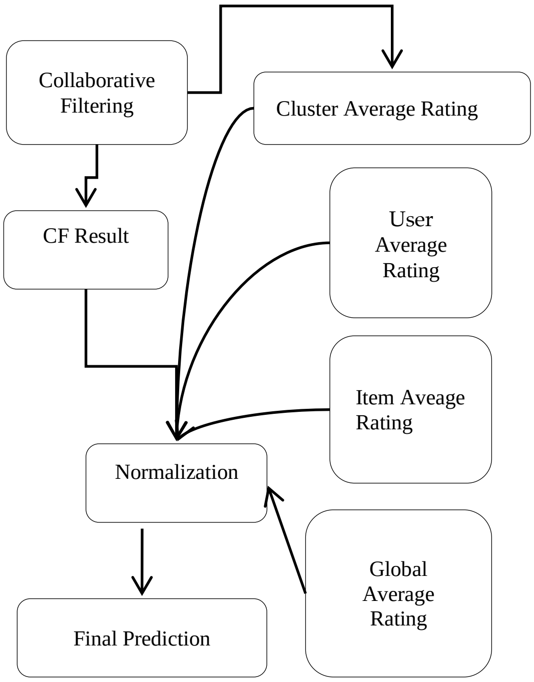

Figure 3. Proposed algorithm CFN.

To further explore the proposed algorithm, it is defined with pseudocode. The process is divided into 2 steps. In the first step, whenever the user gives any rating to an item, clustering is performed (refer Algorithm 1). In the second step, the item needs to be predicted for the user (refer Algorithm 2).

Algorithm 1: CF Clustering

```
Input: Users
Choose random K centroids
Repeat for all users
    Find user closest centroid.
Add to the user to the cluster of closest centroid.
        Recalculate the centroid
```

Algorithm 2: CFN Prediction

```
Input: User Y,Item Z
Set item_average as the average of the rating given to the particular item by all the user who reviewed the item.
Set user_average as the average of the rating given by the particular user to all item reviewed by the user.
Set global_average as the average of rating for a particular genre.

Find cluster of User Y
Set cluster_average as the average rating for the current genre of the item.
Set x as item_averag
Set item_prediction as average rating of the item in the cluster.
Set alpha = Hellinger distance between global_average and item_average
If item_prediction >= item_average
    x = x+alpha
else
    x = x-alpha
Set beta = Hellinger distance between global_average and cluster_average
If cluster_average >= global_average
    x = x+beta
else
    x = x-beta

Set gamma = Hellinger distance between user_average and cluster_average
If user_average >= global_average
    x = x+gamma
else
        x = x-gamma
Set user_average_rating vector as rating by user to all genres
Set global_average_rating vector as rating by all users for all genre.
Set delta to Euclidean distance of user_average_rating and global_average_rating
If x > item_average_rating
        Prediction = item_average_rating + (delta * 1/3)
Else
        Prediction = item_average_rating - (delta * 1/3)
Return Prediction
```

From Figure 3, the average includes the user average rating, item average rating and global average rating. These are calculated separately as they are independent of CF. This process is described in three steps as follows.

- *Step* 1: Distance of Item Average (q) with Predicted Rating (p)

The distance between item average rating (q) and the value predicted by CF is known as α and is represented in Equation (1) as follows:

$$\alpha = \mathrm{H(p, q)} \tag{1}$$

Where a is used to adjusting x according to the rules shown in Equation (2). This adjustment is necessary to make the x closest to the average rating.

$$f(x, p, q) = \begin{cases} x = x + \alpha\, if\ p \geq q \\ x = x - \alpha if\ p < q \end{cases} \tag{2}$$

In Equation (3), x is being replaced by item average rating (q) as the initial average rating because the CFN algorithm takes into account every user in average.

$$f(x, p, q) = \begin{cases} x = q + \alpha\, if\ p \geq q \\ x = q - \alpha if\ p < q \end{cases} \tag{3}$$

- *Step* 2: Distance of Global Average (g) with Cluster Average (c)

This step calculates the distance between global average ratings (g) with a cluster average rating (c) as shown in Equation (4).

$$\beta = H(g, c) \tag{4}$$

To minimize the effect when the user is not very close to the center, the CFN algorithm uses this distance.

Equation (5) summarizes the entire scenario and updates the value of prediction component (x).

$$f(x,g,c) = \begin{cases} x = x + \beta if\ c \geq g \\ x = x - \beta if\ c < g \end{cases} \tag{5}$$

- *Step* 3: Distance User Average (u) and Cluster Average (c)

In order to add personalization, the distance between the cluster average (c) and a user average (u) is calculated and named as γ as shown in Equation (6). This means that even if the user is inside the cluster, he/she does not necessarily give his/her rating exactly the same as the average rating by the cluster.

$$\gamma = H(c,u) \tag{6}$$

In order to utilize the effect of γ, the CFN algorithm checks whether the user average (u) is greater than the cluster average (c). This means if the user tends to give higher ratings by γ component, then γ is added as part of the prediction component (x) or otherwise it is subtracted. This is shown in Equation (7).

$$f(x,u,c) = \begin{cases} x = x + \gamma if\ u \geq c \\ x = x - \gamma if\ u < c \end{cases} \tag{7}$$

This component determines the final value of the prediction. Now user average ratings for all genres (y) and vector of global average rating (z) are passed to the formula for Euclidean Distance (ED) as shown in Equation (8) in order to calculate δ.

$$\delta = d\,(y, z) \tag{8}$$

The ED in Equation (8) shows the distance between user average rating and global average rating. Next, the CFN algorithm makes a decision of what to do with this distance. Item average rating is used to make the decision to subtract or add δ item average rating with prediction component (x). If x is greater than item average rating (q), this means the system has predicted this user to give higher ratings than the average item rating. Therefore, it adds the δ component, which is predicted to be the difference between item average rating (q) and user rating (u), or otherwise, it subtracts the δ component. In Equation (9), the variable m contains the final prediction for the user by the CFN algorithm. A factor of 1/3 is multiplied with δ to keep the resultant value inbound of the higher value of the rating.

$$\mathrm{f(x,q,m,\delta)} = \begin{cases} \mathrm{m = q + (\delta \times 1/3) If\ x \geq q} \\ \mathrm{m = q - (\delta \times 1/3) if\ x} < q \end{cases} \tag{9}$$

For the second algorithm, MFN, the design is similar to that of CFN as it also accepts the output from the MF and is fed into the normalization process. However, instead of using clusters, MF uses the matrix in terms of result assessment. Figure 4 shows that only the result of MF is passed to normalization in addition to the item average rating.

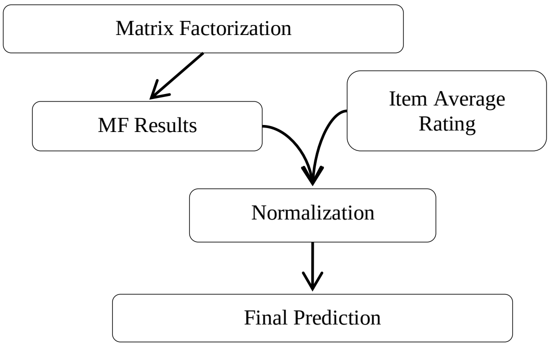

Figure 4. Proposed algorithm MFN.

The pseudocode of MFNis given below to get a better understanding of the algorithm. MFN also operates in 2 steps. Step one is to perform the FM (refer Algorithm 3). Step two is to perform the Normalization of MF (refer Algorithm 4).

Algorithm 3: Matrix Factorization

```
Input: Users and ratings matrix
Transform users rating to matrix of M * N as main_matrix
Create 2 random vectors M * k and N * k where dot product is M * N
Repeat until error greater than the threshold:
    Multiply M * k and N *k
    Compare with main_matrix
    Calculate error
```

Algorithm 4:MF Prediction

```
Input: User Y and Item Z
Set mf_prediction as rating predicted by MF
Set global_average as an average of rating for the particular genre.
Set user_average_rating vector as rating by user to all genres
Set global_average_rating vector as rating by all users for all genre.
Set delta to Euclidean distance of user_average_rating and global_average_rating
If mf_prediction >= global_average
    Prediction = global_average + delta
Else
    Prediction = global_average - delta
```

In MFN, the user average rating and global average rating are not required as the data needs to be normalized in order to address the gray-sheep problem faced by CF. To address the cold start problem, MFN uses the item average rating to help improve the accuracy while at the same time solving the gray-sheep problem. To illustrate the working of MFN, consider two vectors y and z, which are used to calculate a distance for the resultant δ. However, the calculation yields that an equal weight is given to all the genres, even if user seldom watches a particular genre. This is considered unfair to the user. To fix this problem, a new variable w is added to the ED calculation as shown in Equation (10).

$$d(y,z) = \sqrt{\sum_{i=1}^{k} w_i (y_i - z_i)^2} \tag{10}$$

In the proposed MFN algorithm, it is imperative to assess the impact of vectors (*y*) and (*z*). Note that the distance between user average rating and global average rating always depends on the personalization vector. Therefore, instead of taking the prediction of MF, the distance between the users' ratings from the average ratings must be minimized. The weighted distance δ shows that the genre that the user likes to watch or enjoys the most has been given a higher priority. After calculating the δ, MFN proceeds to perform addition or subtraction between δ and the item average rating (*q*). If the value predicted by MF is greater than item average rating, then the MFN algorithm adds or subtracts. As shown in Equation (11), *n* is the final rating predicted by the MFN.

$$f(k,q,n,\delta) = \begin{cases} n = \ q + \delta \; if k \ \geq q \\ n = q - \delta \; if k < q \end{cases} \tag{11}$$

4. Experimental Results and Analysis

In evaluating the performance of the proposed algorithms of CFN and MFN, respectively, a series of comparative experiments were carried out using two public datasets; MovieLens [14] and Group Lens [19]. The datasets consist of 100,000 ratings (1-5) from 943 users for 1682 movies. Users must have rated at least 20 items in order to be included in the dataset. The dataset was then divided into five random sets from the original data, whereby 80% of the data was used as the training sets and the remainder 20% for the test sets. The test sets in all the cases are disjointed. The experiments were performed with the following modifications on the dataset:

- Random selection of 120 users' ratings was reduced to 1-20 items or the maximum number of items if the training set has less than 20 ratings for certain user. This is to replicate the evaluation setting for Cluster with Minimum Error (CME) used in [14].
- Ratings of 120 randomly selected users were deleted from the training sets in order to mitigate the cold start problem. This is to replicate the evaluation setting for Dynamic Classification (DC) used in [14].

Following [27], the performance of the RS was evaluated using three groups of evaluation metrics. The first group is called the Predictive Accuracy metric, which tells the closeness of the recommended results to the actual results. This group of metrics consists of Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Normalized Mean Squared Error (NMAE). The second group is called the Classification Accuracy metric, which decides if the item is beneficiary to the user or not. This group contains F1-measure. The third group is called the Rank Accuracy metric, which measures the proximity of the ordering of recommended items.

The choice of the metric used depends on the selected tasks in the experiments. More importantly, the choice is made to suit the algorithms used in the comparative experiments. In general, many researchers used the de facto standard for benchmarking RS, which is the MAE [13]. A smaller number of works used NMAE (i.e., [14]) in measuring the cold start problem.

4.1. Effect of Normalization on MF

Table 4 shows the comparison of MAE and Root RMSE for MFN and CFN algorithms and compares the results with [13].

Table 4. Results of MFN and CFN against Bell and Koren [4].

	MAE	RMSE
N+MF [13]	0.776	0.977
CFN	0.785	0.988
MFN	0.759*	0.973*

In Table 4, values with an asterisk (*) show the best results. The value shows that implementation of N+MF outperformed CFN, which is consistent with the results from [13]. The difference for RMSE is greater in the comparison with CFN mainly because normalization is used for the solution of larger error and RMSE targets the larger error in the RS. MFN outperformed [13] in both MAE and RMSE. The main reason for this difference is that MFN and algorithms in [13] are based on MF which does not have the gray-sheep problem. MFN also used normalization to target the cold start problem in a better way than [13], and thus, performed better in terms of MAE and RMSE.

4.2. Effects of Normalization on Partial Cold Start Problem

Partial cold start problem is described as a part of cold start problem, where the user rating is less than a threshold to build a better profile for RS and recommends better results. The proposed CFN and MFN algorithms are also compared with the CME method by [25]. Figure 5 shows the comparison of the techniques, CFN, MFN and CME in terms of NMAE that NMAE in solving the cold start problems.

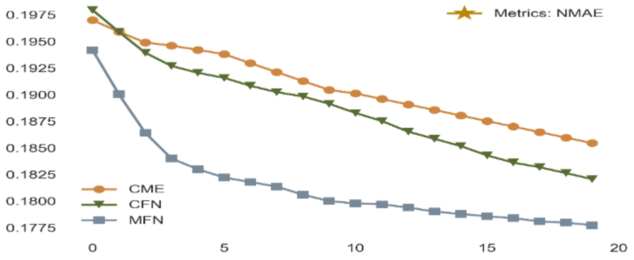

Figure 5. Comparison of partial cold start problem with CME.

The comparison shows that at the very start, CME performed better than CFN. After some ratings, CFN gradually improved and performed better when the user profile grows. Such effect is due to the clustering capability in CF. where clustering forms an important part of the process. MFN results are better than CME and indicate that applying normalization on MF can improve the cold start problem because of the better reliability of MF and tendency of normalization in further improving the results.

4.3. Effect of Normalization on Pure Cold Start Problem

Pure cold start problems are the problems where initial ratings of the users are not available. Most of the algorithms available in the literature are unable to perform under these circumstances as they are designed to perform for user profile where more ratings are available. One notable work that handled pure cold start problem is by using DC [25]. Vizine Pereira and Hruschka [25] compared the performance of CFN and MFN with the performance of DC. The results of DC outperformed CFN because normalization requires more variables or extra parameters to perform at its best. The lack of data produces wrong results for CF, which in turn, provides wrong variables to normalization process leading to inaccurate results. MFN, however, is producing better results because MF is providing more accurate outcomes to the normalization. Dynamic group recommendation has been also explored in music recommendation domain [10].

4.4. Effect of Normalization on Recommender System

As the main focus of this paper is resolving cold start problem, the comparison of the results is done with the algorithms that are based on the similar problem (Figure 6). Since normalization can improve the performance of RS overall, the MFN and CFN results are also compared with recent algorithms such as SVMReg, CCF and MF-Based CF+PrevClass Based [24] as well. It can be seen in Figure 7 that MFN performed better than MF-Based CF+PrevClass Based while CFN failed to perform well against those algorithms.

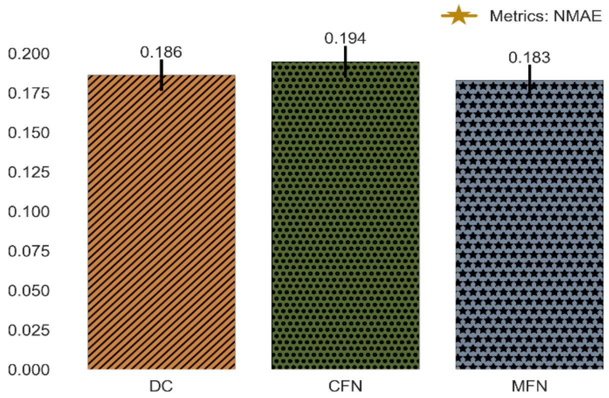

Figure 6. Comparison of pure cold start problem with DC.

5. Conclusions

This paper addressed the effect of normalization on improving the performance of RS using a combinatorial approach to the neighborhood models. Two algorithms were proposed namely MFN and CFN. The results show that MFN improves the NMAE as compared to DC and CME. MFN also improves the MAE of RS as shown in Figure 7 from the hybrid combination of MF and normalization by [13], which is best amongst the algorithms compared in this paper. The main reason for the high performance of MFN lies in the normalization algorithm, which performed very well under the accurate variables provided. While MF avoided the gray-sheep problem, normalization handled the process of cold start, mood changes stability versus plasticity as well as the difference of opinion. Results have shown that their combination has improved the accuracy of the RS.

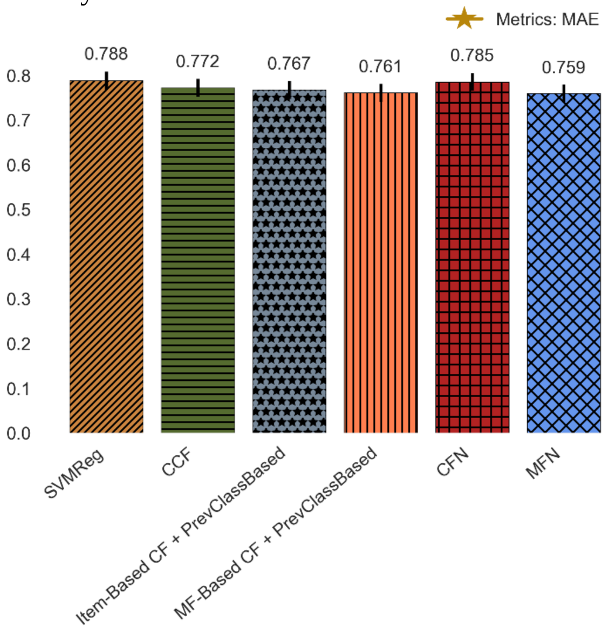

Figure 7. Comparison with other established algorithms.

Contrary to MFN, CFN performed lower than the MFNdue to various problems identified. CFN only performs better in partial cold start problem where the NMAE gradually improved in the comparative experiments. The lower performance of CFN is attributed to two factors. The first is due to the normalization process. RS implementation in [13] normalized the data before the clustering techniques were applied. However, CFN normalized the data using the prediction provided by CF based on raw data. Because the noise in the data can affect the performance of CF-based algorithms [17], the errors produced in CF are being propagated to the normalization stage affecting the accuracy of the CFN algorithm.

Secondly, the clustering variables that are passed to the normalization process in an effort to solve the cold start problem are not accurate altogether because the initial data itself has not been normalized. In pure cold start problem, the user is passed to the group where he/she does not belong leading to the failure of CFN. In partial cold start problems, after failing in the beginning, users are subsequently passed into the correct clusters which improve the accuracy. One possible exploration in the future is to discover the effect of normalization on CFN to make NCFN. In addition, the MFN algorithm can be further optimized by testing it against the different variety of other datasets.

References

[1] Al-Hadi I., Sharef N., Sulaiman M., and Mustapha N., "Ensemble Divide and Conquer Approach to Solve the Rating Scores' Deviation in Recommendation System," *Journal of Computer Science*, vol. 12, no. 6, pp. 265-275, 2016.

[2] Al-Hadi I., Sharef N., Sulaiman M., and Mustapha N., "Review of the Temporal Recommendation System with Matrix Factorization," *International Journal of Innovative Computing, Information and Control*, vol. 13, no. 5, pp. 1579-1594, 2017.

[3] Anand D. and Bharadwaj K., "Utilizing Various Sparsity Measures for Enhancing Accuracy of Collaborative Recommender Systems Based on Local and Global Similarities," *Expert Systems with Applications*, vol. 38, no. 5, pp. 1501-1509, 2011.

[4] Bell R. and Koren Y., "Lessons from the Netflix Prize Challenge," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 75-79, 2007.

[5] Bell R. and Koren Y., "Improved Neighborhood-Based Collaborative Filtering," *in Preseedings of KDD Cup and Workshop at the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, pp. 7-14, 2007.

[6] Bobadilla J., Ortega F., Hernando A., and Gutiérrez A., "Recommender Systems Survey," *Knowledge-Based Systems*, vol. 46, pp. 109-132, 2013.

[7] Burke R., "Hybrid recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331-370, 2002.

[8] Burke R., *The Adaptive Web*, Springer, 2007.

[9] Chandak M., Girase S., and Mukhopadhyay D., "Introducing Hybrid Technique for Optimization of Book Recommender System," *Procedia Computer Science*, vol. 45, pp. 23-31, 2015.

[10] Chen J., Liu Y., and Li D., "Dynamic Group Recommendation with Modified Collaborative Filtering and Temporal Factor," *The International Arab Journal of Information Technology*, vol. 13, no. 2, pp. 294-301, 2016.

[11] Chen J., Wang H., and Yan Z., "Evolutionary Heterogeneous Clustering for Rating Prediction Based on User Collaborative Filtering," *Swarm and Evolutionary Computation*, vol. 38, pp. 35-41, 2017.

[12] Frasincar F., IJntema W., Goossen F., and Hogenboom F., *Business Intelligence Applications and the Web*, IGI Global, 2012.

[13] Ghazanfar M. and Prügel-Bennett A., "Leveraging Clustering Approaches to Solve the Gray-Sheep Users Problem in Recommender Systems," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3261-3275, 2014.

[14] Koren Y., Bell R., and Volinsky C., "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30-37, 2009.

[15] Leung C., Chan S., and Chung F., "An Empirical Study of A Cross-Level Association Rule Mining Approach to Cold-Start Recommendations," *Knowledge-Based Systems*, vol. 21, no. 7, pp. 515-529, 2008.

[16] Luo X., Xia Y., and Zhu Q., "Incremental Collaborative Filtering Recommender Based on Regularized Matrix Factorization," *Knowledge-Based Systems*, vol. 27, pp. 271-280, 2012.

[17] Luo X., XiaY., and Zhu Q., "Applying the Learning Rate Adaptation to the Matrix Factorization Based Collaborative Filtering," *Knowledge-Based Systems*, vol. 37, pp. 154-164, 2013.

[18] Montaner M., López B., and De La Rosa J., "A Taxonomy of Recommender Agents on the Internet," *Artificial Intelligence Review*, vol. 19, no. 4, pp. 285-330, 2003.

[19] Resnick P., Iacovou N., Suchak M., Bergstrom P., and Riedl J., "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *in Proceedings of ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, pp. 175-186, 1994.

[20] Schafer J., Konstan J., and Riedl J., "Recommender Systems in E-Commerce," *in Proceedings of the 1st ACM Conference on Electronic Commerce*, Denver, pp. 185-166, 1999.

[21] Schiaffino S. and Amandi A., "Building an Expert Travel Agent as A Software Agent," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1291-1299, 2009.

[22] Shardanand U. and Maes P., "Social Information Filtering: Algorithms for Automating 'Word of Mouth'," *in Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, Denver, pp. 210-217, 1995.

[23] Thong N., "Intuitionistic Fuzzy Recommender Systems: An Effective Tool for Medical

Diagnosis," *Knowledge-Based Syst*, vol. 74, pp. 133-150, 2015.

[24] Toledo R., Mota Y., and Martínez L., "Correcting Noisy Ratings in Collaborative Recommender Systems," *Knowledge-Based Systems*, vol. 76, pp. 96-108, 2015.

[25] Vizine Pereira A. and Hruschka E., "Simultaneous Co-Clustering and Learning to Address the Cold Start Problem in Recommender Systems," *Knowledge-Based Systems*, vol. 82, pp. 11-19, 2015.

[26] Zahid A. and Sharef M., "Normalization and Matrix Factoization-based Methods for Recommender System," *in Proceedings of 3rd International conference on Computer Science and Computational Mathematics*, Langkawi, pp. 378-383, 2014.

[27] Zhuang L., Jing F., and Zhu X., "Movie Review Mining and Summarization," *in Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pp. 43-50, Arlington, 2006.

Aafaq Zahid is a Master Degree holder in Artificial Intelligence, and has been developing complex mission critical applications for the past 8 years. With his ambition in Artificial intelligence he combined AI in different other areas of computer science. His recent works include combining AI with GPS to achieve the best results for car tracking and vehicle malfunctions predictions. He currently holds position of Head of R&D Department in Azure Innovations, Malaysia.

Nurfadhlina Mohd Sharef is an Associate Professor at the Department of Computer Science and currently Head of Intelligent Computing Research Group at the Faculty of Computer Science and Information Technology, University of Putra, Malaysia. Her research interests are in the areas of Intelligent Computing and she has experience in both academic and consultancy projects involving data science and analytics, text mining, semantic web, and recommendations systems. She also has experience in building data mining solutions for various domains such as economic, medical, logistics. Among her recent projects are the development of adaptive method for the translation of natural language question to semantic query language, development of temporal based recommender system using bacterial foraging optimization algorithm, development of an adaptive method for feature selection and multi-objective optimization for breast cancer recurrence prediction, and development of deep learning model for multi-class classification of tweets. She was also engaged in several consultation projects such as the design of the online logistics aggregation web-based and mobile-based service, the design of the fuzzy aggregation based data analytics for security threat profiling and database integration from heterogeneous resources.

Aida Mustapha received the B.Sc. degree in Computer Science from Michigan Technological University and the M.IT degree in Computer Science from UKM, Malaysia in 1998 and 2004, respectively. She received her Ph.D. in Artificial Intelligence focusing on dialogue systems. She is currently an active researcher in the area of Computational Linguistics, Soft Computing, Data Mining, and Agent-based Systems.