Journal of Computer Sciences

Original Research Paper

Ensemble Divide and Conquer Approach to Solve the Rating Scores' Deviation in Recommendation System

Ismail Ahmed Al-Qasem Al-Hadi, Nurfadhlina Mohd Sharef, Md Nasir Sulaiman and Norwati Mustapha

Department of Intelligent Computing, Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

Article history
Received: 15-09-2015
Revised: 23-06-2016
Accepted: 24-06-2016

Corresponding Author:
Ismail Ahmed Al-Qasem Al-Hadi
Department of Intelligent Computing, Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
Email: esmail.hadi2009@gmail.com

Abstract: The rating matrix of a personalized recommendation system contains a high percentage of unknown rating scores which lowers the quality of the prediction. Besides, during data streaming into memory, some rating scores are misplaced from its appropriate cell in the rating matrix which also decrease the quality of the prediction. The singular value decomposition algorithm predicts the unknown rating scores based on the relation between the implicit feedback of both users and items, but exploiting neither the user similarity nor item similarity which leads to low accuracy predictions. There are several factorization methods used in improving the prediction performance of the collaborative filtering technique such as baseline, matrix factorization, neighbour-base. However, the prediction performance of the collaborative filtering using factorization methods is still low while baseline and neighbours-base have limitations in terms of over fitting. Therefore, this paper proposes Ensemble Divide and Conquer (EDC) approach for solving 2 main problems which are the data sparsity and the rating scores' deviation (misplace). The EDC approach is founded by the Singular Value Decomposition (SVD) algorithm which extracts the relationship between the latent feedback of users and the latent feedback of the items. Furthermore, this paper addresses the scale of rating scores as a sub problem which effect on the rank approximation among the users' features. The latent feedback of the users and items are also SVD factors. The results using the EDC approach are more accurate than collaborative filtering and existing methods of matrix factorization namely SVD, baseline, matrix factorization and neighbours-base. This indicates the significance of the latent feedback of both users and items against the different factorization features in improving the prediction accuracy of the collaborative filtering technique.

Keywords: Collaborative Filtering, Matrix Factorization, K-means, Divide and Conquer

Introduction

Recommendation System (RS) is one of the solutions for information overloading to improve the quality of social networks. The personalized recommendation system utilizes the rating scores of the common users to predict the suitable item to be recommended to the target user. The scores in the rating matrix represent the significant features for the users and items, but the rating matrix commonly consists of unknown rating scores (data sparsity) which lower the quality of the predicted scores' accuracy. However, during the streaming of rating scores into the rating matrix, some rating scores deviate from its accurate places (Cui *et al*., 2014). Usually, the deviation is caused by the streaming of the huge amount of rating scores in the rating matrix without care for sorting and managing these scores to extract the accurate latent feedback. In fact, the position of any rating score after streaming of these scores into the rating matrix effect on the values of the latent feedback. The position of the rating scores is a significant factor for predicting the unknown rating scores. Furthermore, the Collaborative Filtering (CF) solves the limitation of RS.

CF (Armentano *et al*., 2012) is used for RS to explore the similarity of users based on the explicit

features (rating scores). Other latent feedbackof users and items can be explored from the rating matrix based on a prediction method such as Singular Value Decomposition (SVD) which utilizes the decomposition vectors of the known rating scores (Koren, 2009). When the rating score deviates from its appropriate cell in the rating matrix, the results of SVD prediction will give low accuracy. Therefore, once the streaming is completed, there is a need to rearrange all rating scores based on the similarities between users or items.

Few methods have been introduced for rating scores rearrangement such as the Divide and Conquer algorithm (DC) (Gu and Eisenstat, 1995). DC is used to solve a problem of the misplaced object or outlier as a result of data streaming. Mackey *et al*. (2011) have used the Divide-Factor-Combine (DFC) algorithm to deal with the base matrix factorization. The DFC algorithm randomly divides the large-scale matrix factorization task into smaller sub-problems and solve those sub-problems in parallel and then combine them using ensemble methods based on low-rank approximations (Mackey *et al*., 2011). Cui *et al*. (2014) have proposed the state-of-the-art divide and conquer k-means clustering algorithm to reduce the imprecision in rearranging the streaming data.

Mackey *et al*. (2011) have rearranged the matrix factorization based on the ensemble method and Cui *et al*. (2014) have identified the data places based on the clustering method and its relations. However, none of these methods have focused on the similarity of users (sim_u) and the similarity of items (sim_i). Hence, the Ensemble Divide and Conquer (EDC) approach is proposed to solve the data sparsity and also the rating scores' deviation (misplace). The EDC approach is instituted by the SVD algorithm which extracts the relationship between the latent feedback of users and the latent feedback of the items. Besides, this work exploits the ranges scale of rating scores as a sub problem which effected on the approximation among the rating scores. Therefore, the normalization method of the rating scores will provide the accurate approximation among users' features.

The EDC approach exploits the relation of latent feedback between sim_u and sim_i and the combination of sim_u and sim_i for improving the accuracy of personalized RS based on three methods. The first method is called Divide and Conquer based on sim_u (DCU). The second method is the Divide and Conquer based on sim_i (DCI). The third method is the Divide and Conquer based on sim_u and sim_i (DCUI) which combine the methods of DCU and DCI. The sim_u or the sim_i will be measured based on the squared Euclidean distance which is used in k-means algorithm. Furthermore, EDC combines four methods which are SVD, DCU, DCI and DCUI for selecting the lowest error and the highest accuracy of prediction. In addition, the CF provides the personalized recommendations of the set of users. Therefore, the average prediction accuracy of the set of the users is computed to benchmark the experimental methods. The EDC method provides a specific value of the Root Mean Squared Error (RMSE) for each user. To evaluate the proposed methods of EDC correctly, the total beneficiaries of the users are computed for each method separately. The beneficiaries of each method are the users who provide the lower RMSE by this method. The total beneficiaries of each method will be compared to the total number of the users to extract the ratio of this method. The proposed EDC methods are differentiated from the previous divide and conquer methods (Mackey *et al*., 2011; Cui *et al*., 2014; Mirbakhsh and Ling, 2013) which used k-means for generating random clusters.

The contributions of this paper are threefold. The first is introducing the EDC approach to improve the accuracy of prediction for Movie Lens dataset compare to SVD algorithm. This achievement also indicates EDC novelty in solving the deviation of scores during post-streaming by utilizing the sim_u and the sim_i and their relations. The second is that, the EDC approach gives the lowest RMSE compared to CF, SVD, baseline, MF and neighbours-base where the lowest RMSE is the best. Lastly, the performance of all benchmark methods are improved by different percentages based on normalizing the rating scores of users in the rating matrix from a range of [0-5] to [0-1]. The normalization of rating scores was performed based on standard data mining step to improve the accuracy of the RS. The first part of the paper gives the introduction to the problem of rating score deviation and a brief on the proposed EDC method. The second part focuses on the related works and the steps of EDC, while the third part shows the experimental results and discussion and conclusion.

Related Works

The clustering techniques help to divide the huge sparse rating matrix to k matrices by identifying the similar users and similar items which reducing the dimensionality of the rating matrix. The technique of k-means clustering is one of the widely used iterative optimization algorithm (Han *et al*., 2011). It is observed as a popular clustering approach, due to its integrity of execution (Xu and Wunsch, 2008). Therefore, this algorithm will be used as the main tool for the EDC approach to divide the rating matrix into k clusters. There are five proximity measures which are squared Euclidean distance, city block, hamming, cosine and correlation of coefficient. These measures are used in the k-means algorithm for computing and optimizing the summation of the proximity between the members and the centroid point of the clusters. The main convergence distance measures are squared Euclidean distance, city

block and hamming distance. While the main similarity measures are the cosine and correlation of coefficient (Khalil *et al*., 2009). Divide and Conquer algorithm (DC) is used to reduce the noise in Matrix Factorization (MF) by dividing the large scale MF task into sub-problems. Mackey *et al*. (2011) proposed Divide-Factor-Combine (DFC) approach for reducing the noise of MF with missing entries or outliers. DFC contains 2 algorithms which are DFC-PROJ and DFC-NYS based on the approximation technique. DFC-PROJ divides the orthogonal original MF randomly into sub matrix factorizations while DFC-NYS selects sub matrix and uniformly at random. Clearly, DFC deals with random columns and random rows for rearrangement the MF. The combination among sub matrices based on the approximation factor improves the scalability of matrix factorization (Mackey *et al*., 2011).

During the streaming of data into memory, k-means would face a big challenge of reusing the large data, where each object in each iteration would be fetched from disk into memory, which means the data in memory cannot be recycled and causing poor temporal locality. The collaborative DC algorithm has been proposed to improve the state-of-the-art k-means algorithm and to identify the clusters based on reducing the misplaced objects. The collaborative seeding among different partitions have accelerated the convergence inside each partition and the convergence factor of each cluster, which improve the quality of existing clusters (Cui *et al*., 2014). However, neither Mackey nor Cui have exploited the sim_u and the sim_i features for RS. Besides, the relation factors between users and items are not exploited, which have not made personalized RS possible.

Matrix Factorization

Currently, Matrix Factorization (MF) has become a common approach for CF (Mirbakhsh and Ling, 2013), where MF is one of the most effective prediction approaches which are utilized to address the sparse data (Zhou *et al*., 2011). SVD is a traditional MF technique which is used to predict the sparse rating scores for Movie Lens and E-Commerce datasets in RS based on CF (Sarwar *et al*., 2000). SVD has the ability to extract the latent feedback of users and the latent feedback of items based on the relation between users and items and reducing the dimensionality of a rating matrix. Moreover, this approach is able to calculate low-rank approximations, which can be used to calculate the sim_i (Koren, 2008).The factors of the latent feedback can be extracted by the SVD algorithm as shown in Equation 1:

$$\left[PBV\right] = svd\left(Rating\ Matrix\right) \tag{1}$$

This equation is available in several programming languages such as Matlab. Figure 1 is an example showing the input to the SVD algorithm and the output factors using this algorithm. SVD produces three matrices which are the matrix of the latent feedback of users P, the diagonal matrix B and the matrix of the latent feedback of items V.

Equation 2 is used in several matrix factorization methods for predicting the sparse rating scores. Equation 2 uses the latent matrices of P and V for predicting the sparse rating scores in the rating matrix:

$$\hat{r}_{ui} = P_u V_i^T \tag{2}$$

where, $\hat{r}_{ui}$ is the predicted value of the sparse rating score and P_u is the latent feedback of user u and V^T is transpose the matrix V. This method uses the stochastic gradient descent algorithm (Koren, 2010) to reduce the error prediction. Further features of both users and items can be extracted using the baseline method. Baseline method illustrates the effects of users and items separately. There are two factors based on baseline, which are the users' base b_u and the items' base b_i which are extracted using standard deviation. Equation 3 (Koren, 2008) shows the predicted value using the factors of baseline:

$$\hat{r}_{ui} \leftarrow b_{ui} = \mu + b_u + b_i \tag{3}$$

where, μ represents the mean of the rating scores of users. Figure 2 shows an example of predicting the sparse rating scores using the baseline method. Some of the prediction values by baseline are more than the rating range [0-5] which show the over fitting problem.

Furthermore, The MF method uses the factors of baseline and the factors of SVD to learn the factorization features within Equation 4 (Koren, 2009):

$$\hat{r}_{ui} = \mu + b_u + b_i + P_u V_i^T \tag{4}$$

The matrix factorization methods are incorporated with the base features of the neighbours. For example, the model of neighbours-base (Koren, 2010) integrates the factors of baseline with the distance between the rating scores and the base features of the neighbours who provide the rating scores for each item as shown in Equation 5 (Bell and Koren, 2007):

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{x \in N_i} sim_x \left(r_{xi} - b_{xi}\right)}{\sum_{x \in N_i} sim_x} \tag{5}$$

where, N is a set of neighbours that provide item i by rating scores and x is a neighbour which rated item i. The vector of sim_x is the similarity between neighbour x and the target user.r_{xi}, b_{xi} are the rating score of neighbour x and its baseline prediction value, respectively.

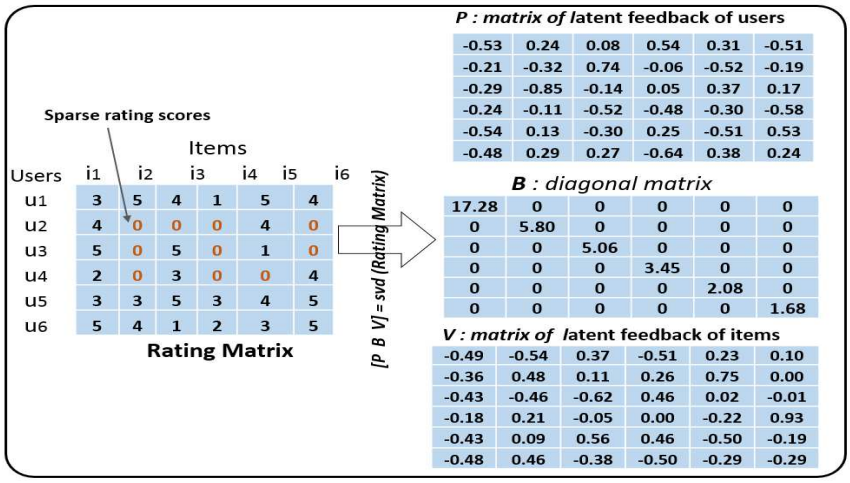

Fig. 1. An example of the SVD factors

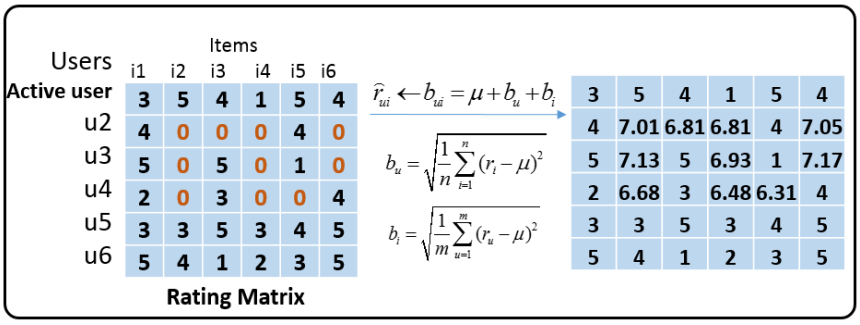

Fig. 2. An example of the baseline predictor

The prediction methods of SVD, baseline, MF and neighbours-base will be used to evaluate the proposed model.

Collaborative Filtering

CF is one of the filtering techniques which RS uses for personalized recommendations (Zhang *et al*., 2011). The members of RS give rating scores about a set of items based on their interests and for personal recommendations based on CF, RS recommends its members based on these rating scores (Bobadilla *et al*., 2013). CF is classified into two types which are memory-based CF and model-based CF. Memory-based CF is used to produce recommendations based on the rating scores of all common users which stores in memory. The rating scores are arranged in the rating matrix and then similarity between the common users and the target user is calculated for predicting the users' interest on items (Ren *et al*., 2013). Therefore, for each target user, a group of common users who have rated the common items more similarly can be recognized as neighbours of the target user (Adibi and Ladani, 2013). Furthermore, top k of users that have high similarities is taken as the nearest neighbours of the target user. Among the limitations of the memory based CF techniques is that the similarity values are determined based on common items and consequently these values of similarity are unreliable because data are sparse when the common items are few (Su and Khoshgoftaar, 2009). On the contrary, the Model-based CF build a model from the specified rating matrix and use prediction method such as Singular Value Decomposition (SVD) to predict the unknown rating scores.

There are three stages in CF process. First, computing the similarity between the common users with the target user, where the cosine function (Zheng and Li, 2011) as shown in Equation 6 is commonly used in this stage (Ahn, 2008):

$$sim(u_a, u_b) = \frac{\sum_{h=1}^{K} r_{u_a,i_h} r_{u_b,i_h}}{\sqrt{\sum_{h=1}^{K} r_{u_a,i_h}^2} \sqrt{\sum_{h=1}^{K} r_{u_b,i_h}^2}} \quad (6)$$

where, $r_{u,i}$ is the rating score which a user u gave to an item i and K is the number of all common items which

rated by both users. Second, computing the predicted rating score value for the item *i*. This is obtained by the deviation from the mean as an aggregation method as shown in Equation 7 (Ahn, 2008):

$$\lambda_i = v_{ua} + \frac{\sum_{h=1}^{M} sim(u_a, u_h)(r_{u_h,i_x} - v_{uh})}{\sqrt{\sum_{h=1}^{M} |sim(u_a, u_h)|}} \quad (7)$$

where, λ_i is the predicted rating score value for the item *i*, v_{ua} *and* v_{uh} are the average rating of the target user u_a and the common user u_h respectively, *M* is the number of common users who have rated item i_x. Last, the Root Mean Squared Error function (RMSE) (Bobadilla *et al*., 2013; Patra *et al*., 2015) is used to benchmark the prediction accuracy of RS as shown in Equation 8:

$$RMSE = \frac{1}{U}\sum_{u=1}^{U}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \lambda_i)^2} \quad (8)$$

where, *U* is the set of the target users, *n* is the number of items that rated by the target user *u*, r_i is the rating score by user *u* for the item *i* and λ_i is the predicted rating score value for the item *i*. Equation 8 provides the average RMSE for evaluating the accuracy prediction of the whole set of target users *U*.

Dataset Description

Several experimental studies have used the Movie Lens dataset (Bobadilla *et al*., 2012; Lisboa *et al*., 2013) to evaluate the performance of RS. This dataset recorded the user rating about movies (1-5 scales) for the purpose of building RS. The data were assembled through the website of Movie Lens (movielens.umn.edu) during the seven-month period from September 1997 to April 1998. This data collected 100,000 ratings from 943 users on 1682 movies (each user has rated at least 20 movies) where 95.4% from rates are missing and each user on average rates 5% of the whole items. This data will be used by the EDC method to provide personalized recommendations.

Normalization

The normalization is a method of data transformation for reprocessing the data for the purpose of improving the accuracy and efficiency of mining algorithms involving distance measurements. In RS the rating scores of users for items contain the distance between the range of [0-5] and based on our experiments this distance measurement gives low accuracy especially for Movie Lens dataset.

Table 1. Normalizing the rating scores

Type	Range	Rating scores					
Original	[0-5]	0	1.0	2.0	3.0	4.0	5
Normalized	[0-1]	0	0.2	0.4	0.6	0.8	1

Therefore, the rating scores will be transferred to the scale of 0 to 1 based on Equation 9 (Han *et al*., 2011):

$$r_{ui} = \frac{r_{ui} - x}{y - x}(m - n) + n \quad (9)$$

where, *x* is the minimum value in the whole matrix and *y* is the maximum value in the whole matrix. In addition, *m* is the maximum of target distance and *n* is the minimum target distance. Table 1 shows the rating score values before and after the normalization.

Methodology

The Ensemble Divide and Conquer algorithm (EDC) solves the problem of the deviation of some rating scores by returning predicted rating scores that have the lowest RMSE. The factors of SVD can be used to predict the sparse rating scores using Equation 2. Barragáns-Martínez *et al*. (2010) have used Equation 10 to predict the sparse rating scores:

$$\hat{r}_{ui} = (PBV^T)_{ui} \quad (10)$$

However, Equation 2 and 10 are not convenient for predicting the sparse rating scores, where the predicted values are very small which lower the prediction accuracy of the CF technique. Therefore, the EDC approach uses Equation 11 to predict the sparse rating scores:

$$\hat{r}_{ui} = (PBV)_{ui} \quad (11)$$

This method integrates the latent feedback of users, the diagonal matrix and the latent feedback of items to predict the spare rating scores. Figure 3 shows an example for justifying the Equation 11 in EDC approach.

In Fig. 3, the first row u_1 in the rating matrix represents the target user and u_2 to u_6 are the common users and the zero values are the spares rating scores which act the data sparsity problem. The rating matrix and the factors of *P*, *B* and *V* contain the same values in Fig. 1. The prediction by Equation 2 and 10 provide inaccurate predicted values which have high RMSE, while the predicted values by Equation 11 are more accurate with the lowest RMSE.

EDC approach combines three methods for learning the accurate latent feedback. First, sorting the common users using divide and conquer based on sim_u. Second, sorting the common items using divide and conquer based on sim_i. Third, sorting the rating matrix using the divide and conquer based on both sim_u and sim_i.

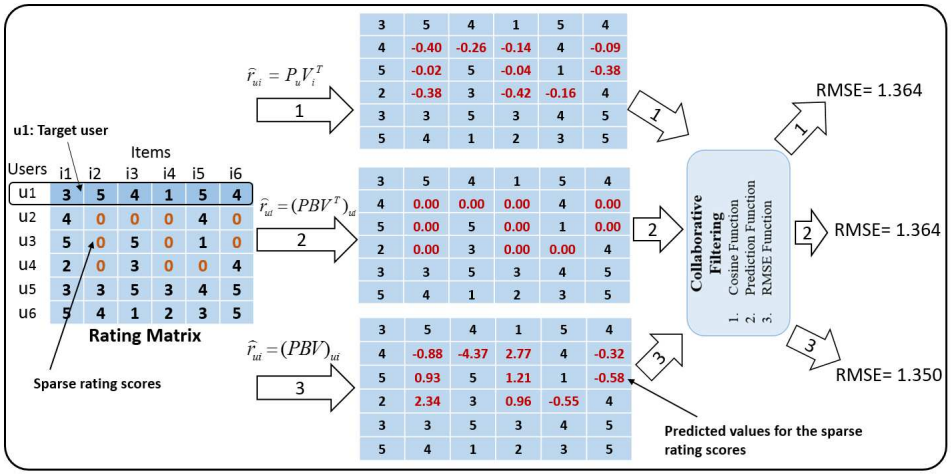

Fig. 3. SVD methods used to predict the spare scores

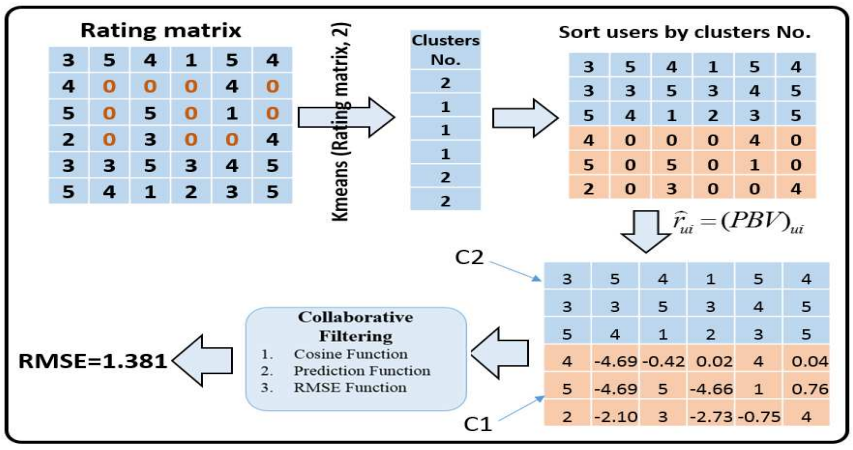

Fig. 4. An example of DCU method

EDC uses the kmeans algorithm to divide the rating matrix and k sets by two clusters. These methods can be described as follows.

Divide and Conquer Based on the Similarity of Users

The algorithm of k-means is used to divide the rating matrix to k clusters based on the sim_u. After dividing the rating matrix into k clusters, Divide and Conquer based on the sim_u (DCU) resorts the clusters based on the best relation between the latent feedback of users. DCU uses k-means to divide his members into k clusters and merge these clusters based on the lowest RMSE. Figure 4 shows an example of the process of DCU with one probability of arrangement. This figure shows how DCU method divides the rating matrix by kmeans algorithm and the users' rating scores arranged based on the cluster's number. The sorted matrix is evaluated by the CF method for getting the value of RMSE.

In Fig. 4, the predicted rating scores are affected by the places of rating scores compared to the original rating matrix in Fig. 3. The arrangement of the rating matrix by cluster number has a positive effect on the predicted rating scores. The method of DCU learns the accutare probability of merging the clusters based on the lowest RMSE. Procedure DCU shows the whole steps and an example of the rating matrix probabilities in Fig. 4. The proposed method uses three clusters (k) which are 2, 3 and 4 and these three k's have 4, 8 and 24 probabilities for merging the clusters respectively:

Procedure DCU
Input: *Rating_Matrix*
Output: *RMSE and Rating Matrix with the accurate predictions for the spare rating scores*
Stages:
1: *Set k clusters and applying k-means to divide the matrix based on* sim_u

k = 2
kmeans(Rating matrix,k)→Cluster1,Cluster2

2: *Arrange 2 matrices based on 2 probabilities of the clusters sorting.*
Cluster1+Cluster2→Matrix1
Cluster2+Cluster1→Matrix2

3: *Predicting the sparse rating scores*
$Matrix1 \rightarrow \hat{r}_{ui} = (PBV)_{ui} \rightarrow RMatrix1$
$Matrix2 \rightarrow \hat{r}_{ui} = (PBV)_{ui} \rightarrow Rmatrix2$

4: *Evaluating both Rmatrix1 and Rmatrix2 according to CF system*
Rmatrix1→CF→RMSE1
Rmatrix2→CF→RMSE2

5: *Choose the rating matrix based on the lowest RMSE*
if RMSE1<= RMSE2 then return Rmatrix1
else return Rmatrix2
End if

Divide and Conquer Based on the Similarity of Items

The similarity features of items represent an important factor where the items that are arranged based on sim_i gives the accurate prediction more than the different items. Divide and Conquer based on the sim_i (DCI) is proposed to learn the accurate relation between latent feedback of the items.

Figure 5 shows an example of the process in DCI where the items in the rating matrix are divided by the kmeans algorithm into two clusters and the columns are sorted based on the cluster number. The prediction accuracy in this example is more accurate than the prediction accuracy in Figure 4 where the DCI method has a lower RMSE compared to the RMSE of the DCU method.

Procedure DCI shows the whole steps and also the probabilities of clusters merging in Fig. 5.

Procedure DCI
Input: *Rating_Matrix*
Output: *RMSE and Rating Matrix with the accurate predictions for the spare rating scores*
Stages:
1: *Set k clusters and applying k-means to divide the matrix*
based on sim_i
k = 2
kmeans(Rating matrix, k)→Cluster1; Cluster2

2: *Arranged 2 matrices based on 2 probabilities of the clusters sorting*
Cluster1+Cluster2→Matrix1
Cluster2; Cluster1→Matrix2

3: *Predicting the sparse rating scores*
$Matrix1 \rightarrow \hat{r}_{ui} = (PBV)_{ui} \rightarrow RMatrix1$
$Matrix2 \rightarrow \hat{r}_{ui} = (PBV)_{ui} \rightarrow Rmatrix2$

4: *Evaluate Rmatrix1 and Rmatrix2 according to CF system*
Rmatrix1→CF→RMSE1
Rmatrix2→CF→RMSE2

5: *Choose the rating matrix based on the lowest RMSE*
if RMSE1<= RMSE2 then return Rmatrix1
else return Rmatrix2
End if

Divide and Conquer Based on Users Similarity and Items Similarity

Some of the target users get the accurate predictions based on the DUC method or DCI method. The method of divide and conquer based on sim_u and sim_i (DCUI) is proposed to combine between the accurate arrangement of DCU method and the accurate arrangement of DCI method. Figure 6 shows an example of this the DCUI process.

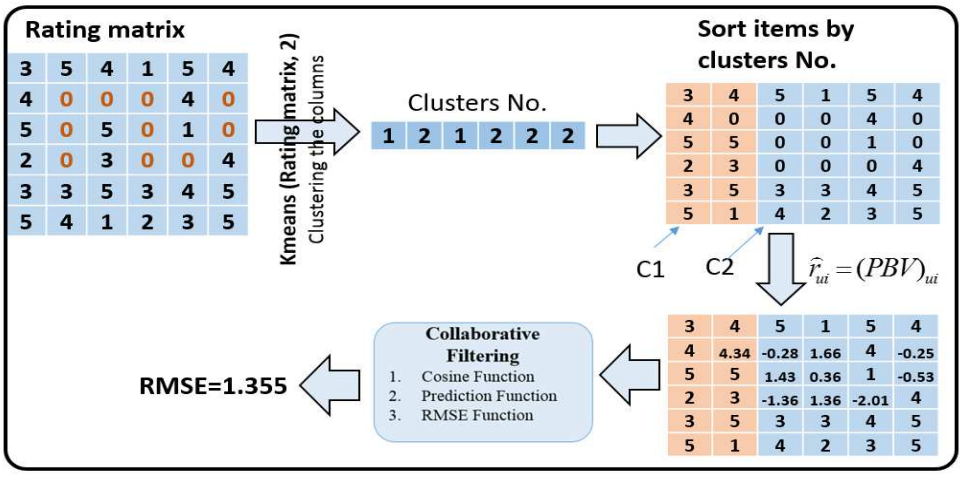

Fig. 5. An example of DCI method

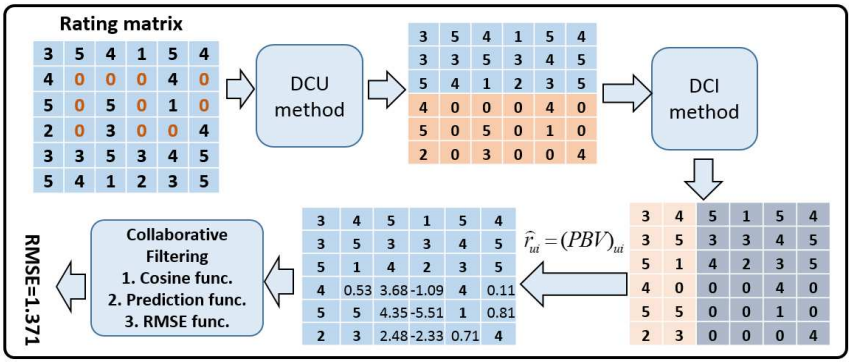

Fig. 6. An example of DCUI method

The prediction performance by DCUI in this example is higher than the prediction performance in Fig. 4 and less than the prediction performance in Fig. 5. The probabilities of merging the clusters are used in the DCU method and in the DCI method. Procedure DCUI shows its learning process in four stages:

Procedure DCUI
Input: *Rating_Matrix*
Output: *Rating Matrix with the accurate predictions for the spare rating scores*
Stages:
1: *Rating matrix→DCU→Rmatrix1*
2: *Rmatrix1→DCI→Rmatrix2*
3: *Predicting the sparse rating scores*
Rmatrix2→ $\hat{r}_{ui} = (PBV)_{ui}$ *→Rmatrix3*
4: *Evaluate Rmatrix3*
Rmatrix3→CF→ RMSE1

The approach of EDC combines the three methods of DCU, DCI and DCUI to learn the accurate places of rating scores compared to the original rating matrix.

Ensemble Divide and Conquer Algorithm

The EDC Algorithm shows the main process of this approach as follows:

EDC Algorithm
Input: *Rating Matrix of the target user after removing the new items*
Output: *Rating Matrix with the accurate prediction of the sparse rating scores*
Stages:
1: *SVD: Extracting the matrices of latent feedback*
[*PBV*] = SVD(*Rating Matrix*)
2: *Rating Matrix→* $\hat{r}_{ui} = (PBV)_{ui}$ *→CF →RMSE1;Matrix1*
3: *DCU procedure:*
Rating Matrix→DCU→RMSE2; Matrix2
4: *DCI procedure:*
Rating Matrix→DCI→RMSE3; Matrix3
5: *DCUI procedure:*
Rating Matrix→DCU→Matrix→DCI→ RMSE4; Matrix4
6: *Choose the accurate Rating Matrix based on the lowest RMSE*
If RMSE1< = RMSE2&& RMSE1<=RMSE3&& RMSE1< = RMSE4 then
RMatrix= Matrix1;
Elseif RMSE2< = RMSE1&& RMSE2<=RMSE3 && RMSE2<=RMSE4 then
RMatrix = Matrix2;
Elseif RMSE3<= RMSE1&& RMSE3<=RMSE2 && RMSE3<= RMSE4 then
RMatrix =Matrix3;
Elseif RMSE4<= RMSE1 && RMSE4<=RMSE2 && RMSE4<= RMSE3 then
RMatrix = Matrix4;
End If
7: *Testing the accurate Rating Matrix*
Rmatrix→CF→RMSE

This algorithm rearranges the users and the items in the rating matrix based on the accurate places of the users' rating scores which reduce the deviation of the rating scores during the streaming process into the memory.

Experimental Results

The Movie Lens data set is used to test the EDC approach and benchmark its performance compared to CF and four methods of MF. The average results are taken to avoid the fluctuation of the RMSE for the whole users and to get the real benchmark. In order to evaluate the prediction accuracy for the sparse rating scores by EDC approach, the following observations are performed:

- Finding the suitable *k* of the clustering and the merging process through a comparison among RMSE and the time complexity for each *k*

- Benchmark the coverage of the users' beneficiaries from DCU, DCI and DCUI, where the methods of EDC are testing each target user separately
- The comparison between the prediction methods of unknown rating scores based on the original range [0-5] and the normalized range [0-1] to benchmark the percentage of improving the prediction quality using EDC and other benchmark method such as SVD, baseline and the neighbours based on baseline

Best k Cluster, RMSE and Time Complexity

The EDC methods use three clusters (k) which are 2, 3 and 4 and these three k's have 2, 8 and 24 probabilities for merging the clusters respectively. These three k's are used to investigate the clustering effect on the latent feedback of members in the rating matrix based on the range scores [0-5]. From our feasibility studies, Table 2 shows the performance of 4 clusters is more accurate than 2 clusters and 3 clusters. Furthermore, the performance of EDC is more accurate than the SVD, DCU, DCI and DCUI methods. The time complexity of EDC methods increases in parallel because the number of probabilities for merging these clusters are increased also. However, the time complixity of 4 clusters is a small (less than 10 sec.) and the accuracy prediction of EDC has improved. Therefore, we use 4 clusters for the validations in the next sections because 4 clusters give the accurate predictions during the suitable time of processing.

Beneficiaries' Coverage

Table 3 shows the percentages of beneficiaries (users) coverage from SVD and other EDC methods based on the range of rating scores [0-5]. As a result of the different behaviours of users, the response to any target user for any method is different to the other target users. Therefore, the total number of the target users (beneficiaries) who have the highest accuracy prediction using each method is investigated using the EDC approach. The percentage of the beneficiaries are represented by dividing the total number of the beneficiaries on the total number of the whole users. For instance, 7 users get high accuracy from the whole users using SVD, then the ratio of coverage is 7 divide on 943 which give 0.74%. EDC has the highest beneficiaries coverage by 99.26% compared to 0.74% by the SVD method (refers to Equation6) where the EDC approach browses the total target users which got the lowest RMSE by using SVD or any method of EDC. Therefore, EDC has improved the performance of SVD by the similarity of users, similarity of items and the combination of them. DCI has beneficiaries more than DCU. The combination of them, DCUI has covered 40% of the beneficiaries which mean DCUI is more accurate than DCU and DCI. EDC collects all beneficiaries based on pair wise comparison.

Table 2. RMSE based on the number of clusters

		Average RMSE of 943 target user				
	SVD	----------				Time
Cluster No.	Eq. 11	DCU	DCI	DCUI	EDC	(s)
k = 2	1.015	1.014	1.014	1.016	1.002	1.09
k = 3	1.015	1.004	0.997	0.996	0.992	3.24
k = 4	1.015	1.000	0.991	0.990	0.987	9.63

Table 3. Percentage of the beneficiary's coverage

Beneficiaries coverage	SVD Eq. 11	DCU	DCI	DCUI	EDC
Total Beneficiaries	7	215	342	379	943
Ratio Beneficiaries to the total users	1%	23%	36%	40%	100%

Table 4. Average RMSE based on the range scales

Method	Range [0-5]	Range [0-1]
CF	1.005	0.201
SVD (Equation 2)	1.038	0.207
Baseline (Equation 3)	1.708	0.344
MF (Equation 4)	1.003	0.201
Neighbours-base (Equation 5)	1.012	0.202
EDC	0.987	0.197

Therefore, EDC solves the problems of data sparsity and the deviation of rating scores and it has improved the relation of latent feedback of the rating matrix perfectly. Furthermore, the results indicate that the latent feedback is more effective and more accurate for accurate prediction based on EDC approach.

Normalization Effect

The small range of the known rating scores [0-1] gives high performance of prediction compared to the big range [1-5]. Therefore, these methods are implemented by using both the range scales for browsing the comparison between them. Table 4 shows the average RMSE of each method and a high percentage of improvement based [0-1] comparing to [0-5] where the performance of all validation methods are increased based on the scale [0-1]. The neighbour base gives more accuracy than Baseline but less than CF. The shortcoming of the neighbour base is the complexity time is high and not suitable for big rating matrix, e.g., in our experiments EDC takes 7 sec for each user compared to the neighbours-base method which take 485 sec for each user. MF also more accurate than CF, SVD, Baseline and neighbours_base. The proposed approach of EDC has returned the lowest RMSE comparing to CF, Baseline, Neighbours-base and MF methods.

Discussion and Conclusion

The main problems of CF are data sparsity, scalability and cold start (Zhang *et al*., 2011). The neighbourhood model is one of the most successful approaches that are used to solve the sparsity problem

and obtained the accurate recommendations, even though there is lower numbers of ratings available in the neighbours of items. A disadvantage of the neighbourhood approach is the low number of neighbours who can provide the accurate predictions. The method of Baseline is used to extract the base features of users and items and SVD is one of the most accurate and scalable algorithms for prediction and solving the challenges of the data sparsity. MF has achieved the accurate prediction performance compared to CF, SVD, Baseline and neighbours_base methods. As a result of stream rating scores of users for items, some rating scores arranged into imprecise place or far from similar rating scores which give imprecise latent feedback. Therefore, the main purpose of EDC approach is to manage the deviation of the rating scores for getting the best interaction between users and items which effect on two important latent factors which founded by the SVD method. The EDC uses k-means algorithm to divide the common users and common items into k clusters. The EDC approach rearranges the misplaced rating scores in the rating matrix by learning the accurate latent feedback of users and items based on the lowest values of RMSE. The experimental results of EDC give high accuracy for the prediction of sparse rating scores compared to CF and four existing methods of MF in this study.

The results of the existing functions of MF are less than CF because these methods give some predicted values bigger than the range scales of rating scores (over fitting). Finally, EDC produces the accurate latent feedback of users and items based on SVD factors which are more important for prediction than the base features, neighbours-base features and MF features. In the future work, the divide and conquer process will be integrated with the latent features of the rating matrix based on the MF methods.

Acknowledgement

This article is part of the research outputs done under the Intelligent Systems group at the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia. The authors would like to thank everyone who has contributed to the progress of the research.

Funding Information

Fundamental research grant funded by the Ministry of Education Malaysia.

Author's Contributions

Ismail Ahmed Al-Qasem Al-Hadi: Main author of the paper and main researcher in the project. Responsible for the writing of the majority of the paper. Developed the solutions proposed and conducted the experiments.

Nurfadhlina Mohd Sharef: Supervision chairman in the project. Responsible for the editing of the majority of the paper and commenting the research ideas.

Md Nasir Sulaiman: Supervision committee in the project. Responsible for the editing of some parts of the paper and commenting the research ideas.

Norwati Mustapha: Responsible for the editing of some parts of the paper and commenting the research ideas.

Ethics

This article is original and contains unpublished material. The corresponding author approved the manuscript and confirms that no ethical issues involved.

References

Adibi, P. and B.T. Ladani, 2013. A collaborative filtering recommender system based on user's time pattern activity. Proceedings of the 5th Conference on Information and Knowledge Technology, May 28-30, IEEE Xplore Press, Shiraz, pp: 252-257. DOI: 10.1109/IKT.2013.6620074

Ahn, H.J., 2008. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Inform. Sci., 178: 37-51. DOI: 10.1016/j.ins.2007.07.024

Armentano, M.G., D. Godoy and A. Amandi, 2012. Topology-based recommendation of users in micro-blogging communities. J. Comput. Sci. Technol., 27: 624-634. DOI: 10.1007/s11390-012-1249-5

Barragáns-Martínez, A.B., E.C. Montenegro, J.C. Burguillo, M. Rey-López and F.A. Mikic-Fonte *et al*., 2010. A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. Inform. Sci., 180: 4290-4311. DOI: 10.1016/j.ins.2010.07.024

Bell, R.M. and Y. Koren, 2007. Lessons from the Netflix prize challenge. ACM SIGKDD Explorat. Newslett., 9: 75-79.DOI: 10.1145/1345448.1345465

Bobadilla, J., F. Ortega, A. Hernando and A. Arroyo, 2012. A balanced memory-based collaborative filtering similarity measure. Int. J. Intelli. Syst., 27: 939-946. DOI: 10.1002/int.21556

Bobadilla, J., F. Ortega, A. Hernando and A. Guti´errez, 2013. Recommender systems survey. Knowl.-Based Syst., 46: 109-132. DOI: 10.1016/j.knosys.2013.03.012

Cui, H., G. Ruan, J. Xue, R. Xie and L. Wang *et al*., 2014. A collaborative divide-and-conquer K-means clustering algorithm for processing large data. Proceedings of the 11th ACM Conference on Computing Frontiers, May 20-22, Cagliari, Italy. DOI: 10.1145/2597917.2597918

Gu, M. and S.C. Eisenstat, 1995. A divide-and-conquer algorithm for the bidiagonal svd. SIAM J. Matrix Anal. Applic., 16: 79-92. DOI: 10.1137/S0895479892242232

Han, J., M. Kamber and J. Pei, 2011. Data Mining: Concepts and Techniques. 3rd Edn., Elsevier, Burlington, ISBN-10: 0123814804, pp: 744.

Khalil, F., J. Li and H. Wang, 2009. An integrated model for next page access prediction. Int. J. Knowl. Web Intell., 1: 48-80. DOI: 10.1504/IJKWI.2009.027925

Koren, Y., 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 24-27, Las Vegas, NV, USA, pp: 426-434. DOI: 10.1145/1401890.1401944

Koren, Y., 2009. Collaborative filtering with temporal dynamics. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Jun 28-Jul. 01, Paris, France, pp: 447-456. DOI: 10.1145/1557019.1557072

Koren, Y., 2010. Factor in the neighbors: Scalable and accurate collaborative filtering. ACM Trans. Knowl. Discovery Data. DOI: 10.1145/1644873.1644874

Lisboa, P.J., H. Nawaf and W. Bhaya, 2013. Improving recommendation systems by modeling the stability of implicit behaviour. The Post Graduate Network Symposium.

Mackey, L.W., M.I. Jordan and A. Talwalkar, 2011. Divide-and-conquer matrix factorization. Adv. Neural Inform. Process. Syst.

Mirbakhsh, N. and C.X. Ling, 2013. Clustering-based factorized collaborative filtering. Proceedings of the 7th ACM Conference on Recommender Systems, Oct. 12-16, Hong Kong, China, pp: 315-318. DOI: 10.1145/2507157.2507233

Patra, B.K., R. Launonen, V. Ollikainen and S. Nandi, 2015. A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. Knowl. Based Syst., 82: 163-177. DOI: 10.1016/j.knosys.2015.03.001

Ren, Y., G. Li, J. Zhang and W. Zhou, 2013. Lazy collaborative filtering for data sets with missing values. IEEE Trans. Cybernet., 43: 1822-1834. DOI: 10.1109/TSMCB.2012.2231411

Sarwar, B., G. Karypis, J. Konstan and J. Riedl, 2000. Application of dimensionality reduction in recommender system-a case study. Technical report.

Su, X. and T.M. Khoshgoftaar, 2009. A survey of collaborative filtering techniques. Adv. Artificial Intell. DOI: 10.1155/2009/421425

Xu, R. and D. Wunsch, 2008. Clustering. 1st Edn., John Wiley and Sons, Oxford, ISBN-10: 0470382783, pp: 400.

Zhang, Z.K., T. Zhou and Y.C. Zhang, 2011. Tag-aware recommender systems: A state-of-the-art survey. J. Comput. Sci. Technol., 26: 767-777. DOI: 10.1007/s11390-011-0176-1

Zheng, N. and Q. Li, 2011. A recommender system based on tag and time information for social tagging systems. Expert Syst. Applic., 38: 4575-4587. DOI: 10.1016/j.eswa.2010.09.131

Zhou, K., S.H. Yang and H. Zha, 2011. Functional matrix factorizations for cold-start recommendation. Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Jul. 24-28, Beijing, China, pp: 315-324. DOI: 10.1145/2009916.2009961