

# Merging user and item based collaborative filtering to alleviate data sparsity

Surya Kant<sup>1</sup>  · Tripti Mahara<sup>1</sup>

Received: 29 February 2016/Revised: 17 May 2016/Published online: 23 June 2016

© The Society for Reliability Engineering, Quality and Operations Management (SREQOM), India and The Division of Operation and Maintenance, Lulea University of Technology, Sweden 2016

**Abstract** Memory based algorithms, generally referred as similarity based Collaborative Filtering (CF) algorithm, is one of the most widely accepted approaches to provide service recommendations. It provides personalized and automated suggestions to customers to select variety of products. Memory based algorithms mainly have two kinds of algorithms: User-based and Item-based algorithms. The User-based CF algorithm recommends items by finding similar users. Contrary to User-based CF, an Item-based CF algorithm recommends items by finding similar items. The core of memory based CF technologies is to calculate similarity among users or items. However, due to inherent sparsity, a large number of entries (ratings) in user-item rating matrix are missing. This results in only few available ratings to make prediction for the unknown ratings. This results in poor prediction quality of the CF algorithm. In this paper a hybrid approach is presented that combines user-based CF and item-based CF. It also leverage the biclustering technique to reduce the dimensionality. The biclustering helps to cluster all users/items into several groups. These clusters are then used to measure users/items similarities based on their respective parent groups. To obtain individual prediction, it adopts the user-based and item-based CF schemes based on the computed similarity respectively. Finally it combines the resultant predictions of each model to make final predictions. Interestingly, experiments demonstrated that the proposed approach outperforms the traditional user-based, item-based and

some state of the art recommendation approaches in terms of accuracy of prediction and quality of recommendations.

**Keywords** Collaborative filtering · Recommendation system · Information filtering · Clustering · Bi-clustering

## 1 Introduction

The World Wide Web users, unsurprisingly experience information overload. Recommender system (RS) is a tool to filter data according to interest of the active user. Till date, Collaborative filtering (CF) is the most successful approach employed in a recommendation system. In CF, opinions are gathered from likeminded users (neighbor users) of active user and used in the decision making process. The Memory based and Model based algorithms are the two categories of CF algorithms. Model-based algorithms are based on predictive model. The general idea is to extract some data from the user-item ratings database and use that as a “model” to make recommendations without using entire database each time. The user-item ratings are modeled with a small set of latent factors that represents characteristics of the items and users (i.e. category of items and preference of users etc.) in the system. This model is then trained and later used to make recommendations. The Memory based approach works on the phenomenon that user who has similar preferences in the past will share similar preferences in the future. Hence, the memory based approach utilizes the entire rating matrix to find the most similar user among the set of users for active user. This set of most similar user is called neighborhood and therefore it is named as neighborhood-based method (Töscher et al. 2008). Memory based algorithms can be

---

✉ Surya Kant  
suryak111@gmail.com

<sup>1</sup> Department of Polymer and Process Engineering, IIT Roorkee, Roorkee 247667, India

classified into two broad categories: User-based CF and Item-based CF algorithms.

Although, CF is a very popular and successful approach, sparsity is a major problem that limits its usefulness. In a RS, the quantity of users and items are unsurprisingly ever increasing. However, due to this many users that are very active may have rated or purchased very limited items from the total items available. Even very popular items might have been purchased or rated by a few users. Consequently there exists high sparsity in user-item ratings matrix. According to Sarwar et al. (2001) the density of matrix is lower than 1 %. The core of CF algorithms is to find similar users or items, but due to extremely sparse user ratings matrix available it is hard to find similar users or items rendering poor performance of CF. In case the system manages to evaluate similarity, it may be possible that this similarity may be not trustworthy. The reason being the amount of information processed was insufficient.

This paper proposes a new CF approach to address the issue of sparsity. Firstly, the biclustering is adapted for dimension reduction, as well as to simultaneously group (clusters) the users and items of the rating data matrix. The similarity between users or items is computed on the basis of the group they belong to. The prediction for an unknown target rating is made by adapting user-based and item-based approaches. The user-based and the item-based approaches utilize the computed users/items similarity respectively. Finally the approach fuses these resultant predictions to estimate the final rating.

## 2 Background work

In last few years, many researchers have integrated Clustering (Kant and Ansari 2015) with several CF based RS in order to address the sparsity problem. The pioneer work in this field (Sarwar et al. 2002) focuses on partitioning the ratings database into smaller sets by creating clusters of users who have similar preferences. O'Connor and Herlocker (2001) experimented by applying various clustering algorithms to partition items. Xue et al. (2005) used clustering in order to smooth data and for neighborhood selection. Jiang et al. (2006) proposed to combine the iterative clustering algorithm with k-Nearest Neighbor. Here the iterative clustering approach was used to exploit the voting information and also used as a smoothing method to the k-Nearest Neighbor approach.

Huang and Yin (2010) showed that Clustering CF provide more accurate predictions if sparsity of data is too high. Birtolo et al. (2011) confirmed it with their experiments that takes advantage of clustering-based CF. Birtolo and Ronca (2013) proposed two clustering-based algorithms for CF. The first algorithm was based on Fuzzy

C-mean clustering and the other one combined trust and similarity. The proposed framework improves quality and coverage of recommendation. Zhang et al. (2014a) adapted user clustering regularization term in their model to optimize standard matrix factorization. This approach integrated user information along with the user-item ratings information to enhance the performance of the recommendation system.

One-dimension clustering is a very popular approach among researchers to cluster users/items. However, it might be possible to miss some useful information by ignoring the opposite dimension. On the other hand, Biclustering technique clusters both dimension i.e. user dimension and item dimension simultaneously. To deal with sparsity problem of RS, the Biclustering technique can be a better approach than one-way cluster technique as demonstrated in literature. George and Merugu (2005) employed a weighted co-clustering algorithm that simultaneously obtains item and user neighborhoods.

Zhang et al. (2014b) proposed cloud-based CF approach using biclustering and Fusion (BiFu). Alqadah et al. (2014) proposed a collaborative filtering method using biclustering neighborhood approach. The system used the local similarity of biclusters and combined it with the global similarity. Vizine et al. (2015) approach combined CF recommendations with demographic information and adapted SCOAL (Simultaneous Co-Clustering and Learning) algorithm. Symeonidis et al. (2008) have used biclustering to reveal the duality between users and items. To match users' preferences a new similarity measure was proposed. Liang and Leng (2014) proposed a CF approach based on information-theoretic co-clustering.

## 3 Preliminaries: simultaneous clustering

Simultaneous clustering also known as biclustering or co-clustering is an important technique to perform simultaneous clustering (grouping) of a matrix by using both rows and columns. The aim of biclustering is to find sub-matrices (subgroups of rows/columns) with highest correlation. Literature has shown rich evidence of successful use of biclustering algorithms in diverse application areas including bioinformatics, social network analysis, web mining etc.

In this paper, we have used xMotif biclustering algorithm (Murali and Kasif 2003).

The Algorithm 1 presents the xMotif algorithm, which is a key component of the proposed approach. Let

$U = \{u_1, u_2, \dots, u_M\}$ , Set of Users

$S = \{s_1, s_2, \dots, s_N\}$ , Set of Items. Each user  $u_i, i = 1, 2, \dots, M$  has rated a subset of items  $s_j, j = 1, 2, \dots, N$ . We

also have predefined specification  $0 < \alpha, \beta < 1$ .  $\alpha$  is defined as a fraction of total users ( $U$ ). Then a bicluster  $b$  ( $U_b, S_b$ ) consists of a subset of user  $U_b \subseteq U$  that together has coherent preference for a subset of items  $S_b \subseteq S$ . The bicluster  $b$  ( $U_b, S_b$ ) have the following conditions:

- *Size* the cardinality of users in  $U_b$  is at least an  $\alpha$  - fraction of  $U$  (total users),
- *Conservation*  $s_b \in S_b$  have the same state for  $\forall u_b \in U_b$  and
- *Maximality* for each item  $s'_b \notin S_b$ , the item is conserved in at most  $\lfloor \frac{\beta}{U_b} \rfloor$  users.

The algorithm initially takes  $u_s$  users (randomly selected) as input which acts as seeds for the biclusters. Every seed consists of a set of users  $u_d$  with  $i_d$  number of items. Algorithm 1 summarizes the main steps of xMotif algorithm.

**Algorithm 1.** FindMotif(): (adapted from Murali and Kasif (2003))

```

1. for  $i=1$  to  $u_s$  do
2.    $c \leftarrow$  select a random sample
3.   for  $j=1$  to  $u_d$  do
4.      $D \leftarrow$  select a random sample of size  $i_d$ 
5.     for each user  $u$ , if  $u$  is in the state  $s$  in  $c$  and all the sample in  $D$ , include the pair  $(u, s)$  in the set  $U_b$ 
6.      $S_b \leftarrow$  set of sample that agree with  $c$  in all the user states in  $U_b$ 
7.     Discard  $(U_b, S_b)$  if  $U_b$  contains less than  $\alpha U$  samples
8.   return the cluster  $(U_b^*, S_b^*)$  that maximizes  $|S_b|$ 
    
```

### 3.1 Item based collaborative filtering

The item based CF is based on the phenomena that the similar items to the item the active user have already preferred in the past may be preferred by the same user. Firstly it finds out neighborhood (similar items) for each item based on item similarity comparison and then prediction are made for active user on that item, based on rating history of active user on the neighborhood of the target item. Equation 1 illustrates how the active user  $u$  rates the similar items,  $i$ .

$$Pred_{u,i}^I = \bar{r} + \frac{\sum_{i' \in N(i)} sim(i, i') \times (r_{u,i'} - \bar{r}_{i'})}{\sum_{i' \in N(i)} sim(i, i')} \tag{1}$$

where  $sim(i, i')$  indicates the similarity between two items,  $\bar{r}_i$  is the average rating of item  $i$ ,  $r_{u,i}$  is the rating of given by user  $u$  to item  $i$  and  $N(i)$  is the neighbors of item  $i$ .

Conventionally, literature suggest various measures to compute the similarity between items, including Pearson Correlation coefficients (PCC), Constrained Pearson Correlation coefficients (CPCC), Cosine(COS) similarity measure, Spearman’s Rank Correlation similarity(SRCC), Jaccard (Koutrika et al. 2009) and mean squared difference (MSD) (Cacheda et al. 2011). One of the widely used

similarity measure among these is cosine similarity measure. It is obtained by computing the cosine of the angle between two items.

The COS formulas are defined as follows (Eq. 2):

$$sim(i, i')^{COS} = \frac{\vec{r}_i \cdot \vec{r}_{i'}}{\|\vec{r}_i\| \cdot \|\vec{r}_{i'}\|} \tag{2}$$

where  $\vec{r}_i$  and  $\vec{r}_{i'}$  is the average rating of item  $i$  and  $i'$  respectively.

### 3.2 User based collaborative filtering

The User-based CF predicts the active user’s interest for an unrated item based on rating information from ‘neighbors’ (similar users). Hence, it is necessary to find out the nearest ‘neighbors’ for each user. The ‘neighbors’ of an active user are those users who shares similar interests. Equation 3

shows how prediction is achieved by user-based collaborative filtering, where  $sim(u, u')$  is the similarity between a pair of users i.e. users  $u$  and  $u'$ .

$$Pred_{u,i}^U = \bar{r}_u + \frac{\sum_{u' \in N(u)} sim(u, u') \times (r_{u',i} - \bar{r}_{u'})}{\sum_{u' \in N(u)} sim(u, u')} \tag{3}$$

The PCC and COS are the most popular used similarity measures. The COS formulas are defined as follows (Eq. 4):

$$sim(u, u')^{COS} = \frac{\vec{r}_U \cdot \vec{r}_{U'}}{\|\vec{r}_U\| \cdot \|\vec{r}_{U'}\|} \tag{4}$$

where  $\vec{r}_U$  and  $\vec{r}_{U'}$  is the mean rating value of user  $u$  and  $u'$  respectively.

## 4 Proposed system: BiUCF

In practice, there exists high sparsity in user-item ratings matrix. In some cases, the density of matrix is lower than 1 % (Sarwar et al. 2001). This is due to the fact that users’ rate only some of the items that they have viewed from a large pool of items in user-item ratings matrix. Therefore, the user-based or item-based model needs to predict the

unknown ratings from very few available ratings. The intuition behind the weighted sum approach is that both models can complement each other and possibly can produce more accurate predictions as compared to individual model.

For example, there may be an instance when the target item has been brought and rated only by few users. This results in item based CF algorithm making prediction by using only few ratings. Simultaneously, if we have enough information about target user like his general rating habits or behavior, it can be used instead of using only few ratings. Similarly if the target user has not rated many items but we have enough information about target item based on the item-based regression model then this information can be used instead of using few items data. Therefore based on the key idea of (Jannach et al. 2012), we propose to

combine user- based and item-based CF algorithm in a weighted sum manner. The weighted sum of user-based and item-based models can be computed as (Eq. 5):

$$Pred_{u,i} = w_u \times Pred_{u,i}^U + w_i \times Pred_{u,i}^I \quad (5)$$

where  $Pred_{u,i}^U$  and  $Pred_{u,i}^I$  are the prediction results of user-based CF and item-based CF algorithms respectively, and  $w_u$  and  $w_i$  are the weights which control the influence of the corresponding models. To minimize the prediction error (difference between the predicted and the actual rating) individual weights for each single user and item need to be optimized. Hence, a fast, heuristic gradient descent approach is adopted based on an existing work by Koren (2010) and Gedikli et al. (2011). It efficiently estimates weight for every user and item.

**Algorithm 2:** BiUCF algorithm

```

9. Split dataset  $R_{m \times n}$  into disjoint training  $\{Tr_{m \times n}\}$  and testing  $\{Te_{m \times n}\}$  sets
10. for  $Tr_{m \times n}$  do
11.    $\{C^u, C^i\} \leftarrow \text{FindMotif}()$ 
12.   return  $\{C^u, C^i\}$ 
      //  $C^u$  and  $C^i$  represents user and item clusters respectively
13. end for
14. for all users  $u$  in  $Tr_{m \times n}$  do
15.    $sim(u, u') \leftarrow$  Calculate the similarities between the target user  $u$  and  $u'$  rest of
      the user in this cluster to which active user belongs i.e  $u \wedge u' \in C^u$ 
16.    $Pred_{u,i}^U \leftarrow$  Compute UserBased Prediction
17. end for
18. for all item  $i$  in  $Tr_{m \times n}$  do
19.    $sim(i, i') \leftarrow$  Calculate the similarities between the target item  $i$  and  $i'$  rest of
      the item in this cluster to which active item belongs i.e.  $i \wedge i' \in C^i$ 
20.    $Pred_{u,i}^I \leftarrow$  Compute ItemBased Prediction
21. end for
22. Initialize: # iterations,  $w_u, w_i, \gamma$  and  $\alpha$ 
23. for # iterations do
24.   for each user  $u$  do
25.     for each rated item  $i$  of user  $u$  do
26.        $Pred_{u,i} = w_u \times Pred_{u,i}^U + w_i \times Pred_{u,i}^I$ 
27.        $E_{u,i} = R_{u,i} - Pred_{u,i}$ 
28.        $w_u = w_u + \gamma(E_{u,i} - \alpha w_u)$ 
29.        $w_i = w_i + \gamma(E_{u,i} - \alpha w_i)$ 
30.     end for
31.   end for
32. end for
33. Evaluate RMSE and MAE of predicted ratings for  $Te$ .

```

The proposed approach called as BiUCF depicted in Algorithm 2 uses a biclustering algorithm and fuses the User and Item based CF. It computes  $w_u$  and  $w_i$  for all users and items which are used to infer a user preference to an item.

## 5 Experimental setup

### 5.1 Data sets

A large number of benchmark datasets are publically available to evaluate the RS approach. The proposed model has been evaluated with three well known real-life datasets (1) MovieLens 100K (ML-100K), consists of 100,000 ratings on 1682 items (movies) made by 943 users. This dataset is 93.69 % sparse, which means that 4.25 % of the movie has been rated by users. (2) MovieLens 1 M (ML-1 M), consists of approximately one million ratings from 6040 users who reviewed 3952 movies. The sparsity rating of ML-1 M is about 95.75 %. In both the MovieLens dataset, each movie is rated on a scale from 1 to 5 and (3) the EachMovie dataset where we have extracted a subset of 1004 users who reviewed 1091 movies with rating scale from 1 to 6.

### 5.2 Evaluation measures

After developing a new algorithm for RS, the next step comprises of evaluating the algorithm. The evaluation checks the performance of algorithm under given circumstances. To evaluate the accuracy of prediction, the most widely used metrics are: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) (Bobadilla et al. 2013).

The MAE is defined as in Eq. 6:

$$MAE = \frac{1}{|S|} \sum_{i=1}^{|S|} |Pred_i - r_i| \quad (6)$$

where  $|S|$  is the cardinality of the test ratings,  $Pred_i$  the vote predicted for a movie, and  $r_i$  the true rating.

The RMSE is presented in Eq. 7:

$$RMSE = \sqrt{\frac{1}{|S|} \sum_{i=1}^{|S|} (Pred_i - r_i)^2}. \quad (7)$$

### 5.3 Experimental result and analysis

For the purpose of evaluating and validating the effectiveness of the proposed framework, its performance is compared with some existing techniques at different sparsity levels. The dataset is partitioned into two disjoint subsets i.e. training set and the test set. The training set is treated as known information and used to train the algorithm and the test set is used to measure the MAE and RMSE of predictions. To test the sensitivity of proposed work on different scales of the training set and the test set on predicted results, we construct nine pair's of training sets and the test set. A split 10–90 means that 10 % of ratings are selected as the training set and the other 90 % ratings for testing. A fivefold cross validation has been conducted and the average MAE and RMSE was taken. In order to show the efficiency of BiUCF, we compare it with the RSVD (Funk 2006), ItemRank (Gori and Pucci 2002), iExpand (Liu et al. 2012), LDA (Liu et al. 2012), artificial immune systems based collaborative filtering (AIS + CF) (Chen et al. 2014) and user-based collaborative filtering (UCF) (Resnick et al. 1994) for the rating prediction accuracy.

Table 1 and 2 shows the MAEs and RMSEs of state-of-the-art methods on MovieLens 100 k dataset.

The results show that BiUCF outperforms all the other methods in terms of the obtained MAE/RMSE. A smaller value of MAE/RMSE (represented in bold) means a better performance. The MAE of AIS + CF are more stable than BiUCF, when the sparsity level of data is average (splits between 40–60 and 80–20). However, when the sparsity is very low in between 90–10 or very high 10–90, 20–80 splits and 30–70, BiUCF outperforms the AIS + CF.

**Table 1** Performance comparison with (Chen et al. 2014) based on MAE results on the ML100K dataset

Split./Alg	RSVD	ItemRank	CF	LDA	iExpand	AIS + CF	BiUCF
10–90	0.887	0.845	0.919	0.909	0.844	0.846	<b>0.834</b>
20–80	0.798	0.796	0.822	0.825	0.795	0.84	<b>0.786</b>
30–70	0.77	0.775	0.787	0.792	0.777	0.787	<b>0.766</b>
40–60	0.76	0.749	0.772	0.778	0.769	<b>0.743</b>	0.753
50–50	0.751	0.755	0.756	0.767	0.759	<b>0.714</b>	0.751
60–40	0.748	0.754	0.751	0.765	0.759	<b>0.703</b>	0.739
70–30	0.747	0.748	0.744	0.758	0.753	<b>0.696</b>	0.719
80–20	0.74	0.749	0.741	0.759	0.755	<b>0.682</b>	0.703
90–10	0.739	0.757	0.746	0.764	0.763	0.68	<b>0.659</b>

**Table 2** performance comparison with (Liu et al. 2012) based on RMSE results on the ML100K dataset

Split./Alg	RSVD	ItemRank	CF	LDA	iExpand	BiUCF
10–90	1.157	1.076	1.172	1.155	1.075	<b>1.065</b>
20–80	1.042	1.006	1.048	1.050	1.008	<b>1.001</b>
30–70	0.995	0.987	1.004	1.008	0.988	<b>0.976</b>
40–60	0.975	0.949	0.983	0.992	0.976	<b>0.958</b>
50–50	0.957	0.958	0.961	0.975	0.963	<b>0.954</b>
60–40	0.947	0.958	0.956	0.972	0.962	<b>0.938</b>
70–30	0.937	0.949	0.945	0.961	0.945	<b>0.915</b>
80–20	0.933	0.949	0.942	0.961	0.955	<b>0.896</b>
90–10	0.915	0.957	0.946	0.967	0.965	<b>0.835</b>

**Table 3** MAE/RMSE comparison of Bi-CF algorithms on each movie dataset

Algorithm	MAE	RMSE
BiUCF	<b>0.7700</b>	<b>1.0379</b>
IBCF	0.9725	1.2642
UBCF	1.0588	1.3570

**Table 4** MAE/RMSE comparison of Bi-CF algorithms on ML1 M dataset

Algorithm	MAE	RMSE
BiUCF	<b>0.7379</b>	<b>0.9378</b>
IBCF	0.8258	1.0645
UBCF	0.8567	1.0920

The RSVD, LDA and iExpand algorithms based on the dimension reduction have not performed well. However, BiUCF uses biclustering for dimension reduction and is able to discover the better indirect correlations in data, hence it performs better. Another interesting observation can be drawn from the results that if the training set is smaller and sparser, then more significant improvement has been made by BiUCF when compared with all the other algorithms.

Moreover, the performance of our proposed framework has been further demonstrated and evaluated with -EachMovie and ML-1M dataset. We adopted two traditional collaborative methods as baselines for this comparison. The baselines include user-based CF with Cosine Correlation Coefficient (UBCF), and an item-based CF with Cosine Correlation Coefficient (IBCF).

The size of the original EachMovie dataset is uncontrollable large which may take high computational time; hence we randomly selected 1004 users who reviewed the 1091 movies from the database. We have partitioned this subset into 5 folds, and then mean MAE and RMSE have been taken as the final output which is listed in the Table 3.

The ML-1M dataset has also been partitioned into fivefolds for evaluating the performance of the purposes method. Table 4 illustrates the mean MAE and RMSE. The proposed method provides better MAE/RMSE for this dataset to the one obtained for the previous dataset. This is because the range of voting possibility of MovieLens dataset is lower than the EachMovie dataset.

## 6 Conclusion

From past few decades, CF has proved itself as a powerful tool to handle the problem of information overload. Personalized recommendations are made to users on the basis of their preferences/taste and those of other similar users.

However, despite their success and popularity, the problem of data sparsity remains an obstacle. In this paper BiUCF algorithm that takes Biclustering into consideration for clustering user/item simultaneously and then the preference of user for an unrated item has been evaluated using weighted sum of user/item-based models. The detailed numerical analysis on three benchmark datasets indicates that the results of proposed approach are comparable with the traditional approach as well as some state art techniques. The proposed technique provides better accuracy of prediction along with quality of recommendation. However, the proposed BiUCF algorithm can be further improved. We plan to extend the BiUCF algorithm in a real-world recommender system.

## References

- Alqadah F, Reddy CK, Hu J, Alqadah HF (2015) Biclustering neighborhood-based collaborative filtering method for top-n recommender systems. *Knowl Inf Syst* 44(2):475–491. doi:10.1007/s10115-014-0771-x
- Birtolo C, Ronca D (2013) Advances in clustering collaborative filtering by means of fuzzy C-means and trust. *Expert Syst Appl* 40(17):6997–7009. doi:10.1016/j.eswa.2013.06.022
- Birtolo C, Ronca D, Armenise R, Ascione M (2011) Personalized suggestions by means of collaborative filtering: a comparison of two different model-based techniques. In *Nature and biologically inspired computing (NaBIC), 2011 third world Congress on IEEE*, pp 444–450. doi:10.1109/NaBIC.2011.6089628.
- Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. *Knowl-Based Syst* 46:109–132. doi:10.1016/j.knosys.2013.03.012
- Cacheda F, Carneiro V, Fernández D, Formoso V (2011) Comparison of collaborative filtering algorithms. *ACM Trans Web* 5(1):1–33. doi:10.1145/1921591.1921593
- Chen MH, Teng CH, Chang PC (2015) Applying artificial immune systems to collaborative filtering for movie recommendation. *Adv Eng Inform* 29(4):830–839. doi:10.1016/j.aei.2015.04.005
- Funk S (2006) Netflix update: try this at home (December 2006). <http://sifter.org/simon/journal/20061211.html>

- Gedikli F, Bagdat F, Ge M, Jannach D (2011) RF-REC: fast and accurate computation of recommendations based on rating frequencies. In: Proceedings of 13th IEEE international conference on commerce and enterprise computing, CEC, pp 50–57. doi:[10.1109/CEC.2011.24](https://doi.org/10.1109/CEC.2011.24)
- George T, Merugu S (2005) A scalable collaborative filtering framework based on co-clustering. Proc-IEEE Int Conf Data Min ICDM. doi:[10.1109/ICDM.2005.14](https://doi.org/10.1109/ICDM.2005.14)
- Gori M, Pucci A (2002) ItemRank: a random-walk based scoring algorithm for recommender engines. In: Proceedings of the international joint conference on artificial intelligence, pp 2766–2771
- Huang C, Yin J (2010) Effective association clusters filtering to cold-start recommendations. In: Proceedings—2010 7th international conference on fuzzy systems and knowledge discovery, FSKD 2010 5 (Fskd), pp 2461–2464. doi:[10.1109/FSKD.2010.5569294](https://doi.org/10.1109/FSKD.2010.5569294)
- Jannach D, Karakaya Z, Gedikli F (2012) Accuracy improvements for multi-criteria recommender systems. Ec 1(212):674–689. doi:[10.1145/2229012.2229065](https://doi.org/10.1145/2229012.2229065)
- Jiang XM, Wen GS, Wei GF (2006) Optimizing collaborative filtering by interpolating the individual and group behaviors. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) 3841 LNCS, pp 568–578. doi:[10.1007/11610113\\_50](https://doi.org/10.1007/11610113_50)
- Kant S, Ansari IA (2015) An improved K means clustering with Atkinson index to classify liver patient dataset. Int J Syst Assur Eng Manag. doi:[10.1007/s13198-015-0365-3](https://doi.org/10.1007/s13198-015-0365-3)
- Koren Y (2010) Factor in the neighbors: factor in the neighbors: scalable and accurate collaborative filtering. ACM Trans Knowl Discov Data 4(1):1–24. doi:[10.1145/1644873.1644874](https://doi.org/10.1145/1644873.1644874)
- Koutrika G, Benjamin B, Hector G-M (2009) FlexRecs: expressing and combining flexible recommendations. In: Proceedings of the 35th SIGMOD international conference on management of data, pp 745–58. doi:[10.1145/1559845.1559923](https://doi.org/10.1145/1559845.1559923)
- Liang C, Leng Y (2014) Collaborative filtering based on information-theoretic co-clustering. Int J Syst Sci 45(3):589–597. doi:[10.1080/00207721.2012.724109](https://doi.org/10.1080/00207721.2012.724109)
- Liu Q, Chen E, Xiong H, Ding CHQ, Chen J (2012) Enhancing collaborative filtering by user interest expansion via personalized ranking. IEEE Trans Syst Man Cybern B Cybern 42(1):218–233. doi:[10.1109/TSMCB.2011.2163711](https://doi.org/10.1109/TSMCB.2011.2163711)
- Murali TM, Kasif S (2003) Extracting conserved gene expression motifs from gene expression data. Pac Symp Biocomput 88:77–88. doi:[10.1142/9789812776303\\_0008](https://doi.org/10.1142/9789812776303_0008)
- O'Connor M, Herlocker J (2001) Clustering Items for collaborative filtering. In: SIGIR 2001, workshop on recommender systems
- Resnick P, Neophytos I, Mitesh S, Peter B, John R (1994) “GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on computer supported cooperative work, pp 175–86. doi:[10.1145/192844.192905](https://doi.org/10.1145/192844.192905)
- Sarwar B, George K, Joseph K, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, pp 285–295. ACM. doi:[10.1145/371920.372071](https://doi.org/10.1145/371920.372071)
- Sarwar BM, Karypis G, Konstan J, Riedl J (2002) Recommender systems for large-scale e-commerce: scalable neighborhood formation using clustering. Communications 50(12):158–167
- Symeonidis P, Nanopoulos A, Papadopoulos AN, Manolopoulos Y (2008) Nearest-biclusters collaborative filtering based on constant and coherent values. Inf Retrieval 11(1):51–75. doi:[10.1007/s10791-007-9038-4](https://doi.org/10.1007/s10791-007-9038-4)
- Vizine P, Andre L, Eduardo RH (2015) Simultaneous co-clustering and learning to address the cold start problem in recommender systems. Knowl-Based Syst 82:11–19. doi:[10.1016/j.knosys.2015.02.016](https://doi.org/10.1016/j.knosys.2015.02.016)
- Xue G-R, Chenxi L, Qiang Y, WenSi X, Hua-Jun Z, Yong Y, Zheng C (2005) Scalable collaborative filtering using cluster-based smoothing. In: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval—SIGIR'05, p 114. doi:[10.1145/1076034.1076056](https://doi.org/10.1145/1076034.1076056)
- Töscher A, Graz A, Legenstein R (2008) Improved neighborhood-based algorithms for large-scale recommender systems
- Zhang CX, Zhang ZK, Lu Y, Liu C, Liu H, Yan XY (2014a) Information filtering via collaborative user clustering modeling. Phys A 396:195–203. doi:[10.1016/j.physa.2013.11.024](https://doi.org/10.1016/j.physa.2013.11.024)
- Zhang D, Hsu C-H, Chen M, Chen Q, Xiong N, Lloret J (2014b) Cold-start recommendation using biclustering and fusion for large-scale social recommender systems. IEEE Trans Emerg Topics Comput 2(2):239–250. doi:[10.1109/TETC.2013.2283233](https://doi.org/10.1109/TETC.2013.2283233)