The Journal of Supercomputing (2021) 77:244–262
https://doi.org/10.1007/s11227-020-03266-2

A collaborative filtering recommendation system with dynamic time decay

Yi-Cheng Chen[1] · Lin Hui[2] · Tipajin Thaipisutikul[3]

Published online: 1 April 2020

Abstract

The collaborative filtering (CF) technique has been widely utilized in recommendation systems due to the precise prediction of users' interests. Most prior CF methods adapted overall ratings to make predictions by collecting preference information from other users. However, in real applications, people's preferences usually vary with time; the traditional CF could not properly reveal the change in users' interests. In this paper, we propose a novel CF-based recommendation, dynamic decay collaborative filtering (DDCF), which captures the preference variations of users and includes the concept of dynamic time decay. We extend the idea of human brain memory to specify the level of a user's interests (i.e., instantaneous, short-term, or long-term). According to different interest levels, DDCF dynamically tunes the decay function based on users' behaviors. The experimental results show that DDCF with the integration of the dynamic decay concept performs better than traditional CF. In addition, we conduct experiments on real-world datasets to demonstrate the practicability of the proposed DDCF.

Keywords Collaborative filtering · Decay function · Human brain memory · Recommendation system

✉ Lin Hui
amar0627@gmail.com

Yi-Cheng Chen
ycchen@mgt.ncu.edu.tw

Tipajin Thaipisutikul
106582601@cc.ncu.edu.tw

[1] Department of Information Management, National Central University, Taoyüan City 32001, Taiwan

[2] Department of Innovative Information and Technology, Tamkang University, Yilan County 26247, Taiwan

[3] Computer Science and Information Engineering, National Central University, Taoyüan City 32001, Taiwan

1 Introduction

Without any doubt, recommendation systems play an essential role in this era of information explosion, since it extracts and predicts what users want or need from the huge amount of published information or data. Recommendation systems have been applied in a variety of areas, such as e-commerce, travel recommendations, online video platforms, and social tagging predictions, to name a few. Collaborative filtering (CF) is one of the most successful recommendation techniques and has been widely utilized due to the precise prediction of users' interests. Currently, CF can mainly be categorized into user-based and item-based CF according to the similarity calculation and preference prediction. User-based CF and item-based CF consider and derive the similarity between users and items, respectively, and then predict the rating of the target item based on the exploited similarities.

User-based CF first finds similar users with common patterns and then recommends items that these similar users are interested in. As shown in Fig. 1a, we can see that Tim and John are similar; hence, we recommend the sundae and donut to John since Tim likes both of them. In contrast, item-based CF uses similarity among items to determine whether a user would like them or not. For example, in Fig. 1b, since ice cream and sundaes are similar, when John chooses ice cream, we will also recommend the sundae to him.

Obviously, both the user- and item-based CF methods adopt overall ratings to make predictions by collecting preference information from other users. However, in real applications, people's interests usually vary with time; traditional CF could not properly capture the change in users' preferences. For example, many little girls love Barbie dolls, but most of them are no longer interested in them when they grow up.

In this paper, we propose a novel CF-based recommendation, dynamic decay collaborative filtering (abbreviated as *DDCF*), which includes the concept of

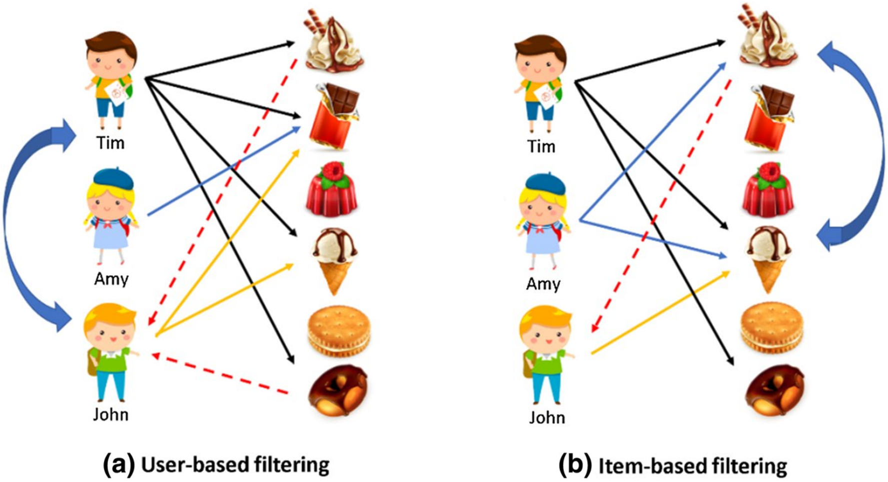


Fig. 1 Concepts of user-based and item-based filtering

dynamic decay function. DDCF can capture users' preference variations and depict the evolution of their interests. Actually, several prior studies [1, 3, 29] pointed out that interest and memory retention are very similar. People's preferences usually decay and vary with time. Our motivation behind this work comes from the concept of the Ebbinghaus forgetting curve [29] of human memory to directly describe the change in a user's interest, as shown in Fig. 2. We have developed the idea of human brain memory to specify the level of a user's preferences (i.e., instantaneous, short-term, or long-term) in our recommendations. For example, as shown in Fig. 2, the number of user reviews affects the preference decay status directly. In short, DDCF has the capability to determine the appropriate decay function to describe the evolution of users' preferences based on their behaviors.

The contributions of our work are described as follows,

- We point out the significance of the time factor in the recommendation system, i.e., the interest of users may vary with time. An elegant recommendation algorithm should gradually attenuate the impact of old data and accurately predict users' future preferences. In this study, the preference decay concept is discussed and included in the CF-based recommendations. We also extend the idea of the human brain memory model to describe preference evolution.
- With the decay function consideration, a novel algorithm, DDCF, has been proposed to effectively recommend items based on users' preferences. To tackle the cold start and sparsity issues of recommendation systems, DDCF utilizes item clustering to group similar items together without any predefined parameters.
- Differing from previous related studies, we propose a dynamic decay method in this study. DDCF specifies the preference level of items, i.e., instantaneous, short-term, or long-term level, and dynamically determines the decay function based on users' rating behaviors.
- When predicting the rating of items, DDCF combines the baseline estimation and decay item-based CF. It can also control the portion of baseline and decay CF to contribute more accurate recommendations.
- To show the practicability of the proposed algorithms, we apply DDCF to real datasets. The experimental studies indicate that the proposed methods are both effective and scalable and outperform the state-of-the-art CF-based algorithms.

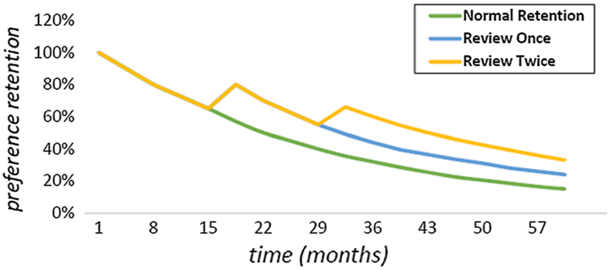


Fig. 2 Evolution of preference retention using the brain memory curve

The remainder of this paper is organized as follows. Sections 2 and 3 provide the related work and some preliminaries, respectively. Section 4 describes the DDCF algorithm. Section 5 presents the experiments and performance study. Finally, we conclude in Sect. 6.

2 Related work

2.1 Collaborative filtering-based recommendation

Sarwar et al. [27] analyzed different item-based recommendation algorithms. They utilized several techniques for computing item–item similarities and proposed recommendation methods. SCF [28] combines item- and user-based collaborative filtering techniques together for recommendation. The authors also mentioned that user-based CF is only suitable for popular item recommendation; by observation, for unpopular items, we should use the item-based CF instead. Zhou et al. [33] utilized a bi-clustering method to group items with an order-preserving matrix and then integrated the similarity calculation into the user-based CF recommendation system. Cai et al. [4] borrowed the idea of object typicality from cognitive psychology and proposed a typicality-based collaborative filtering recommendation system, TyCo. Instead of deriving the similarity according to neighboring users, TyCo has more accurate prediction based on object typicality calculation.

Niemann et al. [22] proposed a collaborative filtering approach based on the items' usage contexts. This approach increases the rating predictions for niche items with fewer usage data available and improves the aggregate diversity of the recommendations. Ma et al. [19] proposed a CF-based method combining k-means clustering and improved the result with SOM. SOM could do a rough cluster preprocessing as input, since k-means clustering needs a proper k setting to get better results. Zhang et al. [31] used a two-layer selection scheme to improve the quality of the selected neighbor for CF recommendation. Two-layer neighbor selection consists of two parts: the availability evaluation module and the trust evaluation module. These two modules are used to calculate user influence and improve recommendations. Gupta et al. [9] combined CF with demographics-based user clusters in a weighted scheme to predict the item rating. The proposed solution is scalable while successfully addressing user cold start and has higher accuracy and coverage. Melville et al. [20] presented an effective framework for combining content and collaboration. They used a content-based predictor to enhance existing user data and then provided personalized suggestions through collaborative filtering. Zhao et al. [32] used a pipeline concept to implement item-based CF on a MapReduce environment for solving the information explosion problem. With this method, CF can be easily applied to a huge dataset.

Some prior studies utilized matrix factorization for improving CF-based recommendation. Nie et al. [21] developed a third-order tensor factorization integrating CF-based technique for recommendation. They also used some latent characteristics to improve accuracy. Chen et al. [5] proposed a tri-factorization method based on orthogonal nonnegative matrix decomposition. After combining with the CF method, the proposed methods could handle the data sparsity issue effectively. Koren et al. [15] proposed a multifaceted CF model, which combines baseline estimates, the neighborhood model, and the latent factor model [14], to significantly improve the accuracy of similarity calculation and output prediction. Ba et al. [2] proposed an approach combining clustering and SVD for collaborative filtering recommendation. They decomposed the rating matrix with the SVD algorithm and calculated the similarity between users, and then found the nearest neighbors in the CF recommendation and predicted the ratings of the items. Using global preference and interest-specific latent factors, Kabbur et al. [13] proposed a nonlinear matrix factorization method to recommend the top-*n* items that users may be interested in. Pirasteh et al. [23] enhanced the recommendation system by exploiting matrix factorization with asymmetric user similarities. Intuitively, two users should be similar when they have common neighbors, even though they do not have any co-rated items.

Renaud-Deputter et al. [26] proposed a novel approach in the implicit feedback recommender system that combines clustering and matrix factorization to yield good results while using only implicit feedback on users' purchasing history without requiring any parameters. There are also some studies which discuss model-based recommendations. Hofmann [10] describes a new model-based algorithm based on a generalization of probabilistic latent semantic analysis to continuous-valued response variables. Hofmann assumes that the observed user ratings can be modeled as a mixture of interest groups which could be characterized by a Gaussian distribution on the normalized ratings. Jiang et al. [12] developed an author topic model-based CF method to facilitate comprehensive points of interest (POIs) recommendations for social users. Many social attributes are adopted for making recommendations or predictions.

2.2 Decay collaborative filtering-based recommendation

As already mentioned, user preferences usually change with time. Some previous works on recommender systems have investigated how to incorporate temporal information into CF-based approaches. Ding [7] mentions the importance of time weight in CF-based methods for recommendation. The accuracy of prediction of collaborative filtering may gradually stop being influenced by the old data. Actually, this concept is intuitive, since the users' preferences usually vary with time. Wu et al. [30] used the power decay function combining user- and item-based collaborative filtering for social tagging label prediction in a digital library. Lee et al. [17] constructed a pseudo-rating CF method using the implicit feedback data. They considered the user's purchase time and the item's rating time for weight decay to improve the recommendation accuracy. Gong et al. [8] proposed a method to evaluate the user's interest change and combined it with the CF model. They used a fixed weight to decay all users' ratings based on item rating time. Richards et al. [25] discussed the advantages and disadvantages of each decay function using for CF. The experimental result also indicates the post-processing time of each decay combined with CF-based recommendation.

To the best of our knowledge, most prior studies utilized one decay function to evaluate and describe the user preference change. Obviously, only one decay function may not properly describe users' complex preference variations. In this paper, due to the similarity of preference and memory, we utilize the memory principle of the human brain to build a model with multiple decay function considerations based on the number and time of the item rating.

Here, we present some related studies about the human memory principle. Memory is the ability to reproduce information stored in the brain. Usually, researchers divide memory into three phases, instantaneous memory, short-term memory and long-term memory [1, 3, 29]. Instantaneous memory storage time is very short, and information could be forgotten very fast. On the contrary, the information storage in short-term memory could stay longer in the human brain than instantaneous memory, but will still be forgotten after a while. The information stored in the long-term memory phase is able to stay for a long time and is not easily forgotten.

3 Preliminary

Suppose that there are a set of users $U=\{u_1, \ldots, u_n\}$ and a set of items $O=\{o_1, \ldots, o_m\}$ in a recommendation system. A rating record is a pair $\left(r_{ij}, t_{ij}\right)$ where r_{ij} and t_{ij} are the rate and time of user u_i rating item o_j, respectively. The rating set e_i is the collection of all rating records of user u_i. A user rating vector is defined as $\vec{u_i} = \left(r_{i1}, t_{i1}\right), \left(r_{i2}, t_{i2}\right), \ldots, \left(r_{im}, t_{im}\right)$, i.e., rating records in e_i with respect to all items in O. Note that if user u_i does not rate item o_j, the values of r_{ij}, t_{ij} in $\vec{u_i}$ are both zero. A rating matrix in a recommendation system is defined as,

$$M = \begin{bmatrix} \vec{u_1} \\ \vec{u_2} \\ \vdots \\ \vec{u_n} \end{bmatrix} = \begin{bmatrix} \left(r_{11}, t_{11}\right) & \cdots & \left(r_{1m}, t_{1m}\right) \\ \vdots & \ddots & \vdots \\ \left(r_{n1}, t_{n1}\right) & \cdots & \left(r_{nm}, t_{nm}\right) \end{bmatrix},$$

where n and m are the number of users and items, respectively.

Definition 1 (*Decayed rate*) Assume that the current time is t. The decayed rate of a rating record is

$$D\left(r_{ij}, t_{ij}\right) = \text{decay}_{\mathcal{L}}(\Delta t) \times r_{ij}, \tag{1}$$

where $\Delta t = t - t_{ij}$. The decay function $\text{decay}_{\mathcal{L}}(.)$ could be linear, logistic, power or exponential decay, to name a few. The concept and examples of the decay function are shown in Fig. 3.

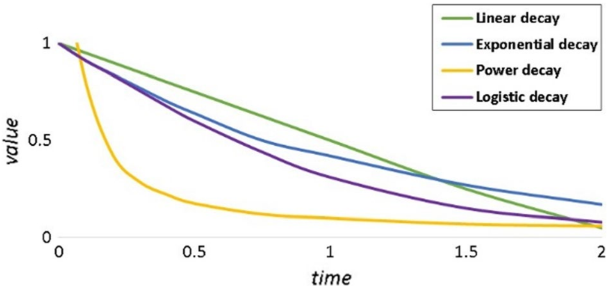
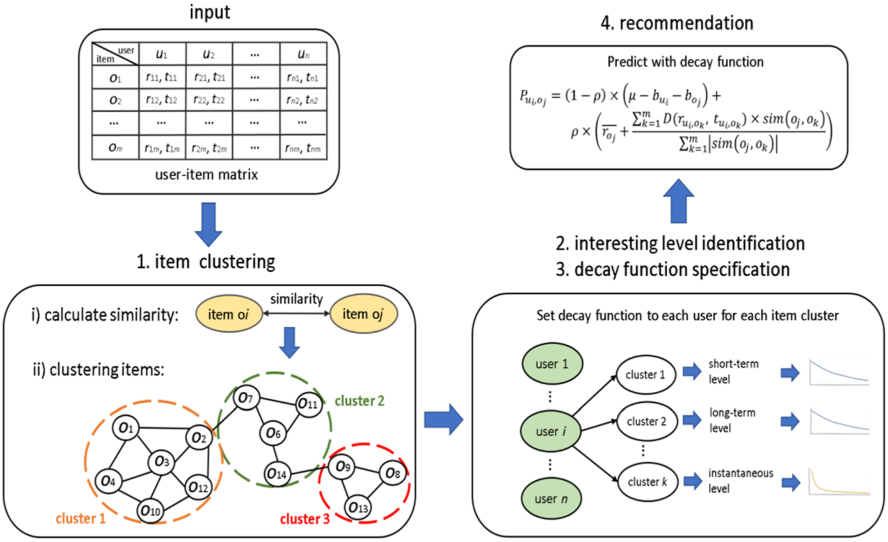


Fig. 3 An example of linear, exponential, power, and logistic decay

4 Proposed method: DDCF

According to the aforementioned discussion, the single decay function may not properly describe the user's complex preference variation. In this study, we propose a novel approach, dynamic decay collaborative filtering (abbreviated as *DDCF*), to effectively predict users' preferences. DDCF has four steps: (1) item clustering, (2) interest level identification, (3) decay function specification, and (4) preference prediction, as shown in Fig. 4.

To tackle the cold start and sparsity issues of recommendation systems, DDCF utilizes item clustering [6] to group similar items together without any predefined parameters. Then, for each user, we identify each cluster's interest level according to the time and number of rating records in the cluster. For each level, DDCF utilizes different decay functions to describe the preference evolution. Finally, we calculate



item \ user	u_1	u_2	···	u_n
o_1	r_{11}, t_{11}	r_{21}, t_{21}	···	r_{n1}, t_{n1}
o_2	r_{12}, t_{12}	r_{22}, t_{22}	···	r_{n2}, t_{n2}
···	···	···	···	···
o_m	r_{1m}, t_{1m}	r_{2m}, t_{2m}	···	r_{nm}, t_{nm}

$$P_{u_i,o_j} = (1-\rho) \times \left(\mu - b_{u_i} - b_{o_j}\right) + \rho \times \left(\overline{r_{o_j}} + \frac{\sum_{k=1}^{m} D(r_{u_i,o_k}, t_{u_i,o_k}) \times sim(o_j, o_k)}{\sum_{k=1}^{m} |sim(o_j, o_k)|}\right)$$

Fig. 4 Concept of proposed DDCF

the similarities among users based on the derived decayed rates and predict the future preferences.

4.1 Item clustering

Cold start and sparsity are two fatal issues in CF-based recommendation. Cold start is related to recommendations for new users or items. Since the system does not have information about new users or items, it is very difficult to make precise recommendations. The sparsity problem is caused by the insufficient number of transactions and feedback data. It is difficult for the recommendation system to distinguish the similar interests among users, which will downgrade the usability of the collaborative filtering. DDCF uses a parameter-free clustering algorithm to solve the cold start and sparsity issues in CF-based recommendation. As in Definition 2, we derive the relation strength by Jaccard coefficient between two users by filtering out the insignificant relations (i.e., when the relation value is lower than a user-specified threshold α).

Definition 2 (*Relation strength*) Given an item o, the profile $p=\{p_1, p_2, \ldots, p_k\}$ consists of k features of item o. The relation between two items can be derived by.

$$R\left(o_i, o_j\right) = \frac{\left|p_i \cap p_j\right|}{\sqrt{\left|p_i\right| \times \left|p_j\right|}}. \tag{2}$$

With the user-specified threshold α, the relation strength is defined as,

$$\mathrm{RS}\left(o_i, o_j\right) = \begin{cases} R\left(o_i, o_j\right), & \text{if } R\left(o_i, o_j\right) \geq \alpha \\ 0, & \text{if } R\left(o_i, o_j\right) < \alpha \end{cases}. \tag{3}$$

Obviously, α could control how dense the relations among items are when clustering and then this would affect the efficiency of the process.

Algorithm 1: *Item_clustering* (O)

```
$C \leftarrow \varnothing$;
$C = \{c_1, \ldots, c_m\}$ ← set each $o \in O$ as a community in $C$
while true do      // community detection
    for each $c_i \in C$ do
        $N(c_i)$ ← collect all neighbors have $RS$ value to $c_i$;
        for each $c_j \in N(c_i)$ do
            if $max\_RS(c_i) = max\_RS(c_j)$
                $c_i$ ← merge $c_i$ and $c_j$;
                $C \leftarrow C - c_j$;
    claculate $S(C)$;
    if $\Delta S < 0$      // strength gain in Definition 3
        break;
output $C$
```

After deriving the relation strength, we use a parameter-free algorithm to cluster the items in the system. The pseudocode is given in Algorithm 1. DDCF proposes a modularity-like evaluation, as shown in Definition 3, to be the terminated criteria of the hierarchical clustering. At each iteration, based on the clustering result from the last iteration, we merge all pairs of items with the strongest relation strength among their neighbors to form larger clusters. Suppose the clustering results in the last iteration and in the current iteration are C and C', respectively. If the strength gained from C to C' is negative, DDCF will stop clustering, since the previous clustering result is good enough. Obviously, we can significantly decrease the time consumed in the clustering due to reducing the computation iteration.

Definition 3 (*Strength gain*) Given an item set $O=\{o_1, \ldots, o_m\}$ in a recommendation system and the clustering result $C=\{c_1, c_2, \ldots, c_p\}$, the strength function is defined as,

$$S(C) = \sum_{k=1}^{p} \left[\frac{\mathrm{IS}_k}{\mathrm{TS}} - \left(\frac{\mathrm{OS}_k}{\mathrm{TS}} \right)^2 \right], \tag{4}$$

where $\mathrm{IS}_k = \sum_{o_i,o_j \in c_k} \mathrm{RS}\left(o_i, o_j\right)$ is the summation of total relation strengths among items inside cluster c_k, $\mathrm{DS}_k = \sum_{o_i \in c_k, o_j \in O} \mathrm{RS}\left(o_i, o_j\right)$ is the summation of relation strengths of items in cluster c_k and other items not in c_k, and $\mathrm{TS} = \sum_{o_i,o_j \in O} \mathrm{RS}\left(o_i, o_j\right)$ is the summation of all relation strengths between any two items in the recommendation system. With two different clustering results C and C', the strength gain from C to C' is defined as,

$$\Delta S_{C \to C'} = S(C) - S\left(C'\right). \tag{5}$$

4.2 Interest level and decay function identification

After clustering items, for each user, DDCF identifies the interest level of each cluster based on his/her rating behavior. We borrow the concept of human brain memory [1, 3] to describe the preference variation. DDCF categorizes users' preferences into instantaneous, short-term, and long-term interest level extending from the idea of the Ebbinghaus forgetting curve [29]. The preference at the instantaneous level is usually very short and may decay fast. On the contrary, the preference at the short-term level may stay longer in the brain than it does at the instantaneous level, but will still decay after a while. The preference at the long-term level is able to stay for a long time and is not easily forgotten.

Suppose the clustering result of item set O in a recommendation system is $C=\{c_1, c_2, \ldots, c_p\}$. For a user u_i and his/her rating set e_i, we could collect all rating records of items clustering in c_k and derive a rating sequence $\left(r_{i1}, t_{i1}\right), \left(r_{i2}, t_{i2}\right), \ldots, \left(r_{i\ell}, t_{i\ell}\right)$ by sorting the rating record with t_{ij} in nondecreasing order. Given a user-specified time size w, the significant set $se_{ik} = \{(r_{ij}, t_{ij}) | t_{ij+1} - t_{ij} \le w, t_{i0} = t_{i1}, 0 \le j \le \ell\}$. According to the Ebbinghaus memory curve [29], we usually will not forget one thing easily after reviewing or mentioning it a significant number of times. We borrow this idea and extend it to describe preference variation. Hence, the interest level $\mathcal{L}_{ik}$ of c_k for u_i is defined as,

$$\mathcal{L}_{ik} = \begin{cases} \text{instantaneous\_level}, & \text{if } 0 < \left|se_{ik}\right| \le \delta \\ \text{short-term\_level}, & \text{if } \delta < \left|se_{ik}\right| \le \theta \\ \text{long-term\_level}, & \text{if } \theta < \left|se_{ik}\right| \end{cases}. \tag{6}$$

Notice that δ and θ are two thresholds to identify the minimum number of ratings of interest level.

According to the level characteristic, DDCF assigns different decay functions for three interest levels, instantaneous, short-term, and long-term. As mentioned above, the instantaneous level is usually very short and may decay fast. We choose the power decay to simulate the preference change. However, when a user rates the items in one cluster over several times, it may mean that he/she is quite interested in this type of item. We could utilize the logistic and exponential decay functions to simulate the preference evolutions of the short-term and long-term levels, respectively. The decay function of each level is defined as,

$$\begin{aligned} \text{decay}_{\text{instant}}(\Delta t) &= \Delta t^{-\lambda} \cdot \alpha, \\ \text{decay}_{\text{short}}(\Delta t) &= \frac{2}{1 + e^{\lambda \cdot \Delta t}}, \\ \text{decay}_{\text{long}}(\Delta t) &= e^{-\lambda \cdot \Delta t}. \end{aligned} \tag{7}$$

Notice that the parameters λ and α could tune the decay degree of function and are usually derived by heuristic evaluation.

4.3 Preference prediction

Differing from traditional CF-based recommendation, DDCF uses baseline estimation and similarity calculation with decay consideration to predict the rate of the item. As pointed out in several discussions in prior studies [16, 24, 27], item-based CF methods usually have better accuracy than user-based CF. Hence, we extend the idea of item-based CF for recommendation. There are several methods which can be utilized for deriving the similarity between two items. In this study, DDCF adopts three methods: cosine, adjusted cosine, and Pearson coefficients, for calculating the similarity between two items. In the next section, we will discuss how each derivation method affects the final prediction results.

Definition 4 (*Item similarity*) Suppose that there are a set of users $U=\{u_1, \ldots, u_n\}$ and a set of items $O=\{o_1, \ldots, o_m\}$ in a recommendation system. Given two items $o_x, o_y \in O$, U_{o_x, o_y} is the set of users in U rated o_x and o_y simultaneously. The $\overline{o_x}$ and $\overline{o_y}$ are the average rates of items o_x and o_y in the recommendation system, respectively. Three methods are adopted for deriving similarity between two items o_x and o_y.

1. The cosine similarity is defined as,

$$cos\_sim\left(o_x, o_y\right) = \frac{\sum_{u_k \in U_{o_x,o_y}} r_{u_k,o_x} \times r_{u_k,o_y}}{\sqrt{\sum_{k=1}^{n} (r_{u_k,o_x})^2} \times \sqrt{\sum_{k=1}^{n} (r_{u_k,o_y})^2}}, \tag{8}$$

2. The adjusted cosine similarity is defined as,

$$acos\_sim\left(o_x, o_y\right) = \frac{\sum_{u_k \in U_{o_x,o_y}} (r_{u_k,o_x} - \overline{r}_{o_x}) \times (r_{u_k,o_y} - \overline{r}_{o_y})}{\sqrt{\sum_{k=1}^{n} (r_{u_k,o_x} - \overline{r}_{o_x})^2} \times \sqrt{\sum_{k=1}^{n} (r_{u_k,o_y} - \overline{r}_{o_y})^2}}, \tag{9}$$

3. The Pearson similarity is defined as,

$$pear\_sim\left(o_x, o_y\right) = \frac{\sum_{u_k \in U_{o_x,o_y}} (r_{u_k,o_x} - \overline{r}_{u_k}) \times (r_{u_k,o_y} - \overline{r}_{u_k})}{\sqrt{\sum_{k=1}^{n} (r_{u_k,o_x} - \overline{r}_{u_k})^2} \times \sqrt{\sum_{k=1}^{n} (r_{u_k,o_y} - \overline{r}_{u_k})^2}}, \tag{10}$$

In DDCF, for a user u_i, the rate prediction of a certain item o_j could be derived as follows:

$$P_{u_i,o_j} = (1-\rho) \times \left(\mu - b_{u_i} - b_{o_j}\right) + \rho \times \left(\overline{r_{o_j}} + \frac{\sum_{k=1}^{m} D(r_{u_i,o_k}, t_{u_i,o_k}) \times sim\left(o_j, o_k\right)}{\sum_{k=1}^{m} \left|sim\left(o_j, o_k\right)\right|}\right), \tag{11}$$

where $0 \leq \rho \leq 1$. Actually, Eq. (11) can be decomposed into two parts: baseline estimation and decay CF. The parameter ρ is used to control the portion of baseline estimation and decay CF contributing to the final prediction result. We utilize $\mu - b_{u_i} - b_{o_j}$ as the baseline estimation to predict the rating value; μ is the average rate of all items in the recommendation system, and $b_{u_i} andb_{o_j}$ are the deviations of the rates of user u_i and item o_j, respectively. Then, when calculating the decay CF for prediction, we use Eq. (1) to derive the decayed rate $D(r_{u_i,o_k}, t_{u_i,o_k})$ based on the time t_{u_i,o_k} and the corresponding decay function in Eq. (7). The similarity function $sim(.)$ could be $cos\_sim(.)$, $acos\_sim(.)$, or $pear\_sim(.)$ as defined in Definition 4. Obviously, given a dataset, different similarity calculations may have different prediction results. We will discuss how similarity calculation affects prediction accuracy in more detail in the next section.

5 Experimental results

To evaluate the performance of the proposed DDCF, five CF-based methods: (1) traditional item-based CF (IBCF), (2) fixed exponential decay CF (DCF-exp), (3) fixed power decay CF (DCF-pow), (4) fixed logistic decay CF (DCF-log), and (5) fixed linear decay CF (DCF-lin), are implemented for comparison. All algorithms were coded in C++ language and tested on a workstation with Intel i7-3370 3.4 GHz with 8 GB main memory. A comprehensive performance study has been conducted on two real datasets, MovieLens [11], to show the applicability of DDCF. The description of the MovieLens dataset is as shown in Table 1. The MovieLens-100K dataset contains 100,000 ratings (1–5 scale) from 716 users for 3,952 movies, while the MovieLens-1M dataset contains 1,000,000 ratings (1–5 scale) from 6040 users for 3952 movies.

5.1 Discussion of prediction accuracy

In this section, we discuss the accuracy of prediction by DDCF. To measure the statistical accuracy of prediction, we use the mean absolute error (MAE) and root-mean-square error (RMSE) as the metrics to evaluate the quality of the prediction results. The MAE and RMSE are defined as,

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} \left| \tilde{r}_i - r_i \right|}{n}. \tag{12}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(\tilde{r}_i - r_i \right)^2}{n}}. \tag{13}$$

n is the number of total predicted rating, $\tilde{r}_i$ the predicted rating for the ith item, and r_i is the user's true rating for the ith item. MAE is defined as the average absolute difference between predicted ratings and actual ratings; likewise, RMSE is the average root-square difference between predicted values and actual values. Both measures are frequently used to assess the goodness of the predicted values by a model or an estimator.

In the first experiment, we compare the MAE and RMSE of DDCF (utilized adjusted cosine similarity, i.e., *acos_sim* in Eq. (9)) with another five CF-based methods on two real datasets, as shown in Tables 2 and 3. Obviously, the proposed DDCF has the best accuracy of prediction compared to the other algorithms.

In the second experiment, to show the accuracy of DDCF with different training–testing partitions, we vary the ratio of the training and testing portions of the MovieLens-1M dataset from 50 to 90%. As shown in Figs. 5 and 6, compared to the other CF-based methods, DDCF has better accuracy. Notice that DDCF still has

Table 1 Descriptions of the MovieLens [11] datasets

	MovieLens-100K	MovieLens-1M
Number of users	943	6040
Number of movies	1682	3952
Number of ratings	100,000	1,000,209
Average rated movies/user	125	149.7
Rating range	1–5	1–5

Table 2 MAE and RMSE in the MovieLens-100K dataset

	MAE	RMSE
IBCF	0.810133	1.088575
DCF-exp	0.802042	1.078469
DCF-log	0.802041	1.078469
DCF-pow	0.802221	1.078666
DCF-lin	0.80954	1.04201
DDCF	0.737513	0.946979

Table 3 MAE and RMSE in the MovieLens-1M dataset

	MAE	RMSE
IBCF	0.771146	0.985259
DCF-exp	0.769135	0.960049
DCF-log	0.769136	0.960023
DCF-pow	0.768411	0.95936
DCF-lin	0.770700	0.964040
DDCF	0.728071	0.924965

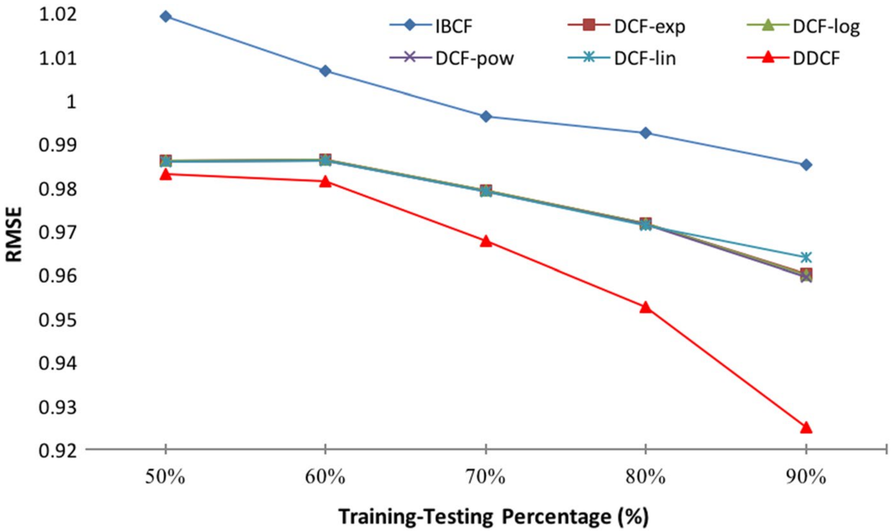


Fig. 5 MAE of the six algorithms on the MovieLens-100K dataset

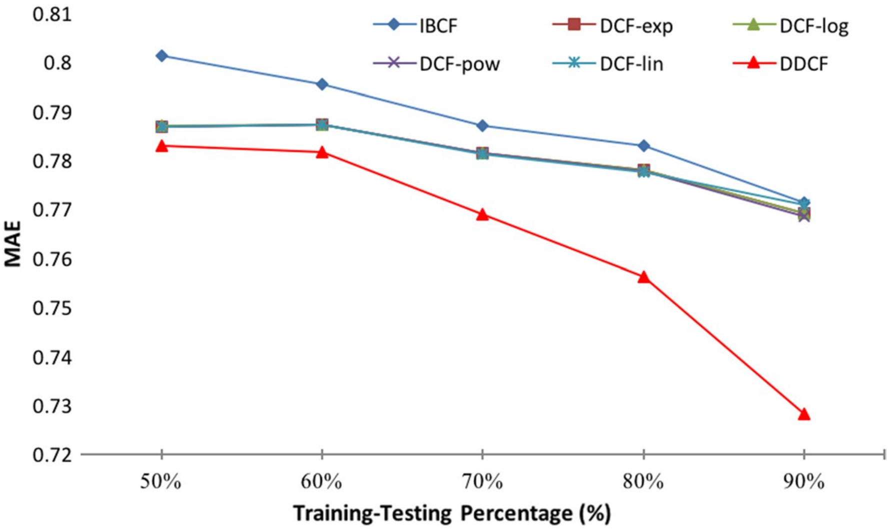


Fig. 6 RMSE of the six algorithms on the MovieLens-1M dataset

more precise prediction than CF using the fixed decay functions (i.e., power, logistic, linear, and exponential). This is partly because dynamically tuning the decay function could more accurately simulate the variance of preference.

5.2 The effect of the similarity function

In this section, we discuss how similarity deriving methods affect the prediction results. We compare the adoption of cosine, adjusted cosine, and Pearson similarity calculation in DDCF to observe the accuracy of the Movielens-1M datasets. With the training–testing ratio 90–10%, as shown in Table 4, adjusted cosine could lead to better prediction results than the other two methods. This is partly because the adjusted cosine performs normalization, i.e., minus the item average rating, before calculating the similarity between two items. We can also observe that the Pearson coefficient does not perform well in terms of depicting the similarity between two items. The normalization with user means in Pearson may not be suitable for describing the preference variations of items in DDCF.

Table 4 MAE and RMSE with different similarity calculations. (The MovieLens-1M dataset with 90%-10% training–testing ratio)

	MAE	RMSE
Pearson	0.729096	0.926689
Cosine	0.729232	0.927182
Adjusted cosine	0.728071	0.924965

Table 5 Concept of precision and recall rates

	User-watched movies	User-not-watched movies
Recommended movies	True Positive (TP)	False Positive (FP)
Non-recommended movies	False Negative (FN)	True Negative (TN)

5.3 Recommendation quality

To show the quality of the recommendations, we use the precision rate, recall rate, and *f*-measure to evaluate the top-k recommendations of DDCF and the other CF-based methods. We use Table 5 to explain the concept of precision and recall rates. True Positive (TP) means the set of recommended movies that users will watch and give ratings for; False Positive (FP) means the set of recommended movies that users will never watch. On the contrary, False Negative (FN) is the set of nonrecommended movies that users will actually watch and True Negative (TN) is the set of nonrecommended movies that users really will not watch.

The precision rate, recall rate, and *f*-measure are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{15}$$

$$F\text{-Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{16}$$

In the following experiments, the top-k highest prediction score movies are recommended to users. From Figs. 7, 8, 9, 10, 11, and 12, we use the precision rate, recall rate, and *f*-measure to evaluate the quality of recommendations on the MovieLens-100K and MovieLens-1M datasets. As shown in Figs. 7 and 8, the precision

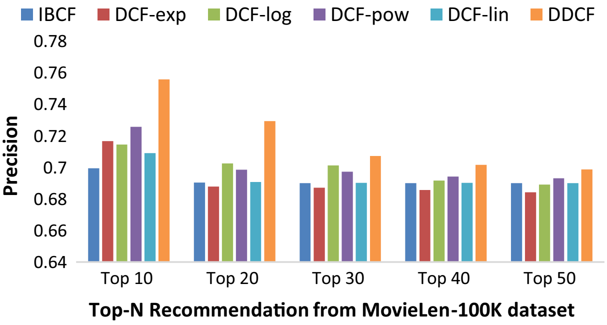


Fig. 7 Precision rate based on the top-N recommendations from the MovieLens-100K dataset

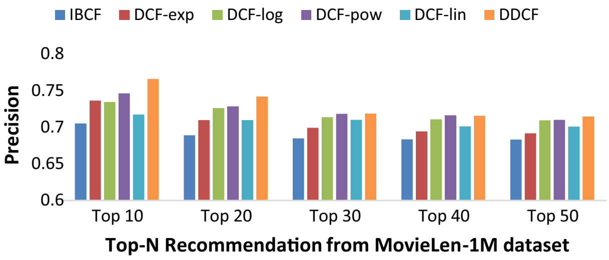


Fig. 8 Precision rate based on top-N recommendations from the MovieLens-1M dataset

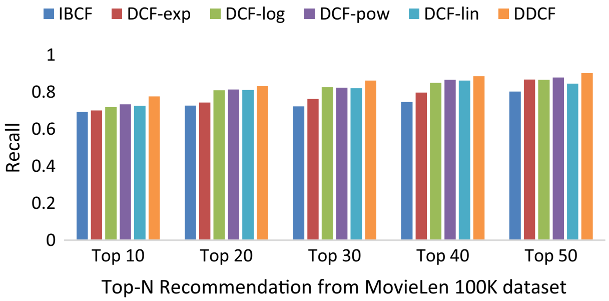


Fig. 9 Recall rate based on the top-N recommendations from the MovieLens-100K dataset

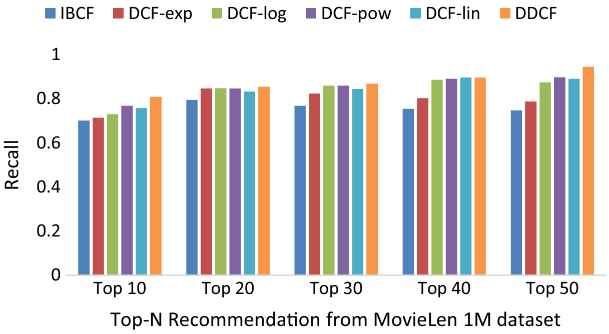


Fig. 10 Recall rate based on the top-N recommendations from the MovieLens-1M dataset

results indicate that the recommendation of DDCF has the best results compared with all different top-k settings. In addition, compared to other CF-based methods, DDCF has shown the most significant highest precision results, especially with the top-10 recommendations.

For recall rate measurement, as shown in Figs. 9 and 10, DDCF still has the best results compared with other CF-based methods in both the MovieLens 100K and 1M datasets. The recall rate of the proposed DDCF outperforms IBCF and other fixed decay CF-based methods when using the top-30 recommendations.

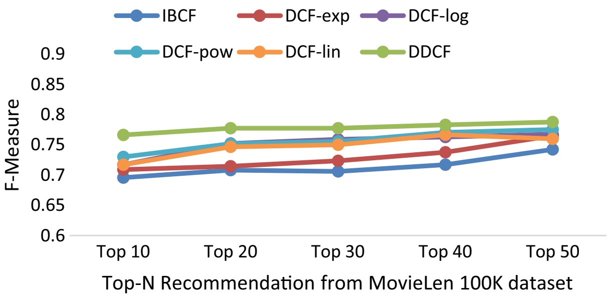


Fig. 11 *F*-measure based on the top-N recommendations from the MovieLens-100K dataset

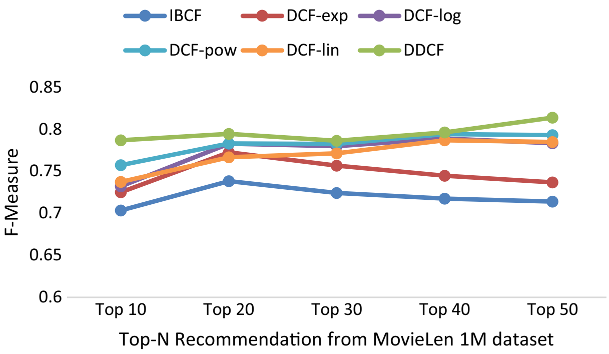


Fig. 12 *F*-measure based on the top-N recommendations from the MovieLens-1M dataset

Regarding the final experiment, we discuss the *F*-measure of all algorithms. Likewise, for the evaluation of *F*-measure, DDCF also performs better than the other recommendation algorithms. From Figs. 11 and 12, we can observe that DDCF has a high *F*-measure value with the top-10 and top-50 recommendations. Notice that when doing top-30 and top-40 recommendations, the *F*-measure of DDCF is almost the same as DCF-pow and DCF-log. This is partly because the recall rates of the three algorithms are very similar when $N = 30$ and 40.

6 Conclusion

In this paper, we propose a novel CF-based recommendation system, *DDCF*, which includes the concept of dynamic decay function. DDCF extends the idea of human brain memory to dynamically adjust the decay functions based on users' behaviors. To tackle the cold start and sparsity issues of recommendation systems, we utilize item clustering to group similar items together without any predefined parameters. Furthermore, DDCF combines baseline estimation and decay item-based recommendation to predict users' ratings. The experimental results indicate that DDCF performs better than traditional collaborative filtering and other fixed decay function consideration. We also applied the proposed DDCF to two real datasets to show its practicability.

Funding This study was supported by Ministry of Science and Technology, Taiwan (Grant numbers MOST 108-2221-E-008-063-MY3, MOST 108-2221-E-032-036-).

References

1. Atkinson R, Shiffrin R (1968) Human memory: a proposed system and its control processes. Psychol Learn Motiv 2:89–195
2. Ba Q, Li X, Bai Z (2013) Clustering collaborative filtering recommendation system based on SVD algorithm. In: 2013 4th IEEE International Conference on Software Engineering and Service Science (ICSESS' 13), pp 963–967
3. Baddeley A (1997) Human memory: theory and practice. Psychology Press, England
4. Cai Y, Leung H, Li Q, Min H, Tang J, Li J (2014) Typicality-based collaborative filtering recommendation. IEEE Trans Knowl Data Eng 26(3):766–779
5. Chen G, Wang F, Zhang C (2009) Collaborative filtering using orthogonal nonnegative matrix tri-factorization. Inf Process Manag 45(3):368–379
6. Chen Y, Zhu W, Peng W, Lee W, Lee S (2014) CIM: community-based influence maximization in social networks. ACM Trans Intell Syst Technol (TIST) 5(2):1–31
7. Ding Y, Li X (2005) Time weight collaborative filtering. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM'05), pp 485–492
8. Gong S, Cheng G (2008) Mining user interest change for improving collaborative filtering. In: Proceeding of Second Intelligent Information Technology Application (IITA'08), pp 24–27
9. Gupta J, Gadge J (2015) Performance analysis of recommendation system based on collaborative filtering and demographics. In: 2015 international conference on Communication, Information & Computing Technology (ICCICT'15)
10. Hofmann T (2003) Collaborative filtering via Gaussian probabilistic latent semantic analysis. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'03), pp 259–266
11. https://movielens.org
12. Jiang S, Qian X, Shen J, Fu Y, Mei T (2015) Author topic model-based collaborative filtering for personalized POI recommendations. IEEE Trans Multimed 17(6):907–918
13. Kabbur S, Karypis G (2014) Nlmf: nonlinear matrix factorization methods for top-n recommender systems. In: 2014 IEEE International Conference on Data Mining Workshop (ICDMW'14), pp 167–174
14. Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'08), pp 426–434
15. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42(8):42–49
16. Kumar A, Gupta S, Singh S, Shukla K (2015) Comparison of various metrics used in collaborative filtering for recommendation system,. In: 2015 Eighth International Conference on Contemporary Computing (IC3 2015)
17. Lee T, Park Y, Park Y (2008) A time-based approach to effective recommender systems using implicit feedback. Expert Syst Appl 34:3055–3062
18. Li D, Cao P, Guo Y, Lei M (2013) Time weight update model based on the memory principle in collaborative filtering. J Comput 8(11):2763–2767
19. Ma Z, Yang Y, Wang F, Li C, Li L (2014) The SOM based improved K-means clustering collaborative filtering algorithm in TV recommendation system. In: 2014 Second International Conference on Advanced Cloud and Big Data (CBD'14), pp 288–295

20. Melville P, Mooney R, Nagarajan R (2002) Content-boosted collaborative filtering for improved recommendations. In: Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)
21. Nie U, Liu Y, Yu X (2014) Weighted aspect-based collaborative filtering. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'14), pp 1071–1074
22. Niemann K, Wolpers M (2013) A new collaborative filtering approach for increasing the aggregate diversity of recommender systems. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD' 13), pp 955–963
23. Pirasteh P, Hwang D, Jung J (2015) Exploiting matrix factorization to asymmetric user similarities in recommendation systems. Knowl Based Syst 83:51–57
24. Resnick P, Iacovou N, Suchak M, Bergstorm P, Riedl J (1994) Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work (CSCW'94), pp 175–186
25. Richards D, Morgan T, Wilson R, Schwebach V, Young G (1993) Good times, bad times, and the diversionary use of force: a tale of some not-so-free agents. J Confl Resolut 37(3):504–535
26. Renaud-Deputter S, Xiong T, Wang S (2013) Combining collaborative filtering and clustering for implicit recommender system. In: 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA'13), pp 748–755
27. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web (WWW10), pp 285–295
28. Wang J, Yin J (2013) Combining user-based and item-based collaborative filtering techniques to improve recommendation diversity. In: The IEEE 6th International Conference on Biomedical Engineering and Informatics (BMEI'13), pp 661–665
29. Wixted J, Ebbesen E (1997) Genuine power curves in forgetting: a quantitative analysis of individual subject forgetting functions. Mem Cognit 25(5):731–739
30. Wu D, Yuan Z, Yu K, Pan H (2012) Temporal social tagging based collaborative filtering recommender for digital library. In: International Conference on Asian Digital Libraries (ICADL'2012), pp 199–208
31. Zhang Z, Liu Y, Jin Z, Zhang R (2018) A dynamic trust based two-layer neighbor selection scheme towards online recommender systems. Neurocomputing 285:94–103
32. Zhao Z, Wang C, Wan Y, Huang Z, Lai J (2015) Pipeline item-based collaborative filtering based on MapReduce. In: The IEEE Fifth International Conference on Big Data and cloud computing (BDCloud'15), pp 9–14
33. Zhou W, Zhang G, Zhao X, Li M, Hu X, Xue Y (2015) A collaborative filtering algorithm based on biclustering. In: Proceedings of the 2015 International Conference on Machine Learning and Cybernetics (ICMLC 2015), pp 533–538

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.