



Improving collaborative filtering's rating prediction accuracy by introducing the experiencing period criterion

Dionisis Margaris¹ · Dimitris Spiliotopoulos² · Costas Vassilakis² · Dionysios Vasilopoulos²

Received: 27 January 2020 / Accepted: 26 October 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

Collaborative filtering algorithms take into account users' tastes and interests, expressed as ratings, in order to formulate personalized recommendations. These algorithms initially identify each user's "near neighbors," i.e., users having highly similar tastes and likings. Then, their already entered ratings are used, in order to formulate rating predictions, and predictions are typically used thereafter to drive the recommendation formulation process, e.g., by selecting the items with the top-K rating predictions; henceforth, the quality of the rating predictions significantly affects the quality of the generated recommendations. However, certain types of users prefer to *experience* (purchase, listen to, watch, play) items the moment they become available in the stores, or even preorder, while other types of users prefer to *wait for a period of time before experiencing*, until a satisfactory amount of feedback (reviews and/or evaluations) becomes available for the item of interest. Notably, a user may apply varying practices on different item categories, i.e., be keen to experience new items in some categories while being uneager in other categories. To formulate successful recommendations, a recommender system should align with users' patterns of practice and avoid recommending a newly released item to users that delay to experience new items in the particular category, and vice versa. Insofar, however, no algorithm that takes into account this aspect has been proposed. In this work, we (1) present the *Experiencing Period Criterion* rating prediction algorithm (CF_{EPC}) which modifies the rating prediction value based on the combination of the users' *experiencing wait period* in a certain item category and the period the rating to be predicted belongs to, so as to enhance the prediction accuracy of recommender systems and (2) evaluate the accuracy of the proposed algorithm using seven widely used datasets, considering two widely employed user similarity metrics, as well as four accuracy metrics. The results show that the CF_{EPC} algorithm, presented in this paper, achieves a considerable rating prediction quality improvement, in all the datasets tested, indicating that the CF_{EPC} algorithm can provide a basis for formulating more successful recommendations.

Keywords Recommender systems · Rating prediction accuracy · Collaborative filtering · Experiencing period criterion · Ratings' timestamps · Pearson correlation coefficient · Cosine similarity · Evaluation

1 Introduction

Collaborative filtering (CF) targets at formulating personalized recommendations by taking into account user opinions on items expressed as ratings. CF algorithms are distinguished into two main categories, the user-based (or

user–user) and item-based (or item–item). User-based CF algorithms initially examine the resemblance of already entered user ratings on items, in order to identify people who share similar likings, tastes, and opinions—these users are termed as “near neighbors” (NNs)—who will act as potentially recommenders to a user U , since U 's recommendations formulation will be based on their ratings [1]. Finally, the recommendation of an item I is the result of the prediction the user-based CF algorithm will produce for this item, which is based on: (1) the similarity between user U and his NNs and (2) the ratings that U 's NNs have entered for item I [2].

✉ Costas Vassilakis
costas@uop.gr

¹ Department of Informatics and Telecommunications,
University of Athens, Athens, Greece

² Department of Informatics and Telecommunications,
University of the Peloponnese, Tripoli, Greece

However, certain types of users prefer to *experience* items (we will henceforth use the word “experience” to substitute the cases of “purchasing,” “listening,” “watching,” “playing,” “tasting,” etc.) the moment they become available in the stores, or even preorder (e.g., an iPhone, the Microsoft Flight Simulator, a pair of shoes, or an expensive red wine). Other types of buyers prefer to wait for a period of time before ordering; the period may vary depending on the item category—e.g., a smartphone buyer may wait for a couple of months, while a car buyer may wait for a whole year, or even more—until a satisfactory amount of feedback (reviews and/or evaluations) becomes available for the item of interest [3]. Marketing research has recognized this consumer behavior at least three decades ago: in [4] adopters of new products are classified in five categories depending on their experiencing wait period (EWP): innovators; early adopters; early majority; late majority; and laggards (in increasing order of EWP). Since the introduction of this classification, many researchers have studied different related attributes of customer’s behavior and/or factors specific to particular customer categories or market segments [5–8]. Taking the above information into account, recommender systems (RSs) should try to align their recommendations with the users’ patterns of practice, arranging so that users with small EWPs receive recommendations for newly released products, while refraining from presenting such recommendations to users with long EWPs.

In this paper, we address the aforementioned problem by (1) introducing the concept of the *Experiencing Period Criterion (EPC)*, as well as (2) presenting a related algorithm, namely CF_{EPC} , which modifies the rating prediction value based on the combination of the users’ usual experiencing time period in a certain item category and the item’s period the rating to be predicted belong to, in order to enhance the rating prediction accuracy of CF-based RSs.

In order to validate the rating prediction accuracy of the CF_{EPC} algorithm proposed in this work, we conduct an extensive evaluation, using seven contemporary and widely used datasets from various product categories (books, music, movies, videogames, etc.), considering the two most widely used similarity metrics in CF research, namely the Pearson Correlation Coefficient (PCC) and the Cosine Similarity (CS) [1, 2]. To measure the algorithm’s prediction accuracy, we employ four rating prediction error metrics, including the two most widely used ones in CF research, the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) [1, 2]. We also compare the performance of the CF_{EPC} algorithm against the performance of:

- (a) The Common Item Rating Past Criterion-based algorithm (denoted as CF_{CIRPaC}), proposed in [9],

which exploits the ratings’ timestamps to adjust the significance of rating (di)similarity for an item among two users according to the resemblance between the set of experiences that each of the users had before rating the item.

- (b) The *Early Adopters* algorithm (denoted as CF_{EA}), proposed in [10], which also adjusts the significance of rating (di)similarity for an item among two users, however on a different basis. More specifically, the CF_{EA} algorithm introduces an adoption eagerness likeness factor between two ratings for the same item, boosting the importance of (di)similarities between ratings that belong to the same experiencing period of the item (early or late) and attenuating this importance when the ratings belong to different experiencing periods.

Furthermore, the CF_{CIRPaC} and the CF_{EA} algorithms have been published recently, being thus within the state-of-the-art in the field of collaborative filtering, have been shown to surpass the performance of other state-of-the-art CF algorithms.

All three algorithms (the proposed CF_{EPC} , the CF_{CIRPaC} , and the CF_{EA}) are based only on the very basic CF information (no item categories and characteristics, no social relations between users, no demographic data, etc.), which is comprised only of user ratings on items, including the time of the rating, and consequently can be used by every CF-based recommender system.

The proposed methodology achieves considerable improvements considering rating prediction accuracy and recommendation quality and outperforms recently published algorithms. This is achieved because, unlike the CF_{CIRPaC} [9] and the CF_{EA} [10] algorithms that undertake a fine-grained approach to adjust the importance between already entered rating (di)similarities, the CF_{EPC} algorithm proposed in this paper takes a different approach, adjusting the value of the prediction by a factor corresponding to the perceived utility of the item to the user, according to the item “freshness” and the typical EWP of the user, as the latter is derived by the ratings s/he has already entered in the user-item ratings database. This shift of focus has been shown experimentally to lead to improved rating prediction accuracy.

Finally, it is worth noting that the CF_{EPC} algorithm presented in this work, can be combined with other works in the domain of CF, targeting either to: (a) increase rating prediction computation efficiency, (b) enhance rating prediction accuracy, or (c) improve recommendation quality in CF-based RSs. Such works include concept drift detection techniques [11–14], clustering techniques [15–17], algorithms which exploit social network data [5, 6], or hybrid filtering algorithms [18–20].

The rest of the paper is structured as follows: Sect. 2 overviews related work, while Sect. 3 introduces the CF_{EPC} algorithm. Section 4 reports on the methodology for tuning the algorithm operation, as well as evaluates the presented algorithm using seven contemporary and widely used CF datasets, and, finally, Sect. 5 concludes the paper and outlines future work.

2 Related work

CF-based systems' accuracy is a research field targeted by numerous research works [21, 22] over the last years, and many features of the ratings database or external linked data sources, wherever available, have been taken into consideration for improving rating prediction accuracy [18, 23].

[24] recognizes three types of similar users (super dissimilar, average similar, and super similar), by combining traditional similarity metrics. It also presents a new similarity metric that is used in the case of average similar user pairs and evaluates the presented recommendation method by experimenting with data derived from Movielens and Epinions datasets. [25] proposes a neural network-based recommendation framework, which targets to issue of the RSs efficiency by investigating sampling strategies in the stochastic gradient descent training for the framework. It also establishes a connection between the user-item interaction bipartite graph and the loss functions, where the latter are defined on links, while the graph nodes include major computation burdens. [26] proposes an RS algorithm which achieves to improve rating prediction quality by taking into account the users' ratings variability when computing rating predictions. Matrix factorization (MF) techniques [27] constitute an alternative approach to computing rating predictions for users. As noted in [28], MF-based RS algorithms always produce predictions for users ratings on items; however, when these techniques are applied to sparse datasets, the predictions involving items or users which do not have a sufficient number of ratings actually degenerate to a dataset-dependent constant value, and hence are not considered as personalized predictions [29] a phenomenon reflected into the rating prediction accuracy of the algorithm.

A different approach targeting the rating prediction accuracy improvement is the knowledge-based CF systems. [30] presents a context-aware knowledge-based mobile RS, namely RecomMetz, which targets the domain of movie showtimes based on multiple features such as time, location, and crowd information. RecomMetz considers the items to be recommended as composite entities, with the most important aspects of them being the movie, the theater, as well as the showtime. [31] proposes a knowledge-based CF system which produces leisure time

suggestions to Social Network (SN) users. The presented algorithm considers the profile and habits of the users, qualitative attributes of the items-places, such as price, service, and atmosphere, places similarity, the geographical distance between the places' locations, as well as the users' influencers' opinions.

Many works identified that SN information can be incorporated in a CF-based system, in order to upgrade rating prediction accuracy. [32] proposes an SN memory-based CF algorithm and examines the impact of incorporating social ties in the rating prediction formulation, targeting rating prediction accuracy. Furthermore, in order to tune the contribution of the SN information, it uses a learning method in the presented similarity measure, as a weight parameter. [17] presents a clustering approach to CF recommendation that, instead of using rating data, exploits relationships between SN users in order to identify their neighborhoods. Furthermore, the algorithm applies a complex network clustering technique on the SN users, in order to formulate similar user groups and then, uses traditional CF algorithms in order to generate the recommendations. [33] proposes an algorithm that targets SN RSs which operate using sparse user-item rating matrixes and sparse SN graphs. For each user, this algorithm computes partial predictions for item ratings items from both the user's CF neighborhood and SN neighborhood, which are then combined using a weighted average technique. The weight is adjusted to fit the characteristics of the dataset, while for users having no CF (resp. SN) near neighbors, item predictions are based only on the respective user's social neighborhood (resp. CF neighborhood). Although all the aforementioned approaches successfully improve CF prediction accuracy, the improvement is based on information found in complementary information sources (e.g., an SN), which is not always available.

Recent research has shown that the exploitation of the time parameter in the rating prediction formulation process can significantly improve CF systems rating prediction accuracy, due to the concept drift phenomenon (when the relation between the input data and the target variable changes over time) [34]. A fuzzy user-interest drift detection recommender algorithm, which adapts to user-interest drift in order to improve rating prediction accuracy, is introduced in [35], while [36] builds a Bayesian personalized ranking algorithm which takes into account the user behavior temporal dynamics and price sensitivity. The work in [37] presents a concept drift MF algorithm for tracking concept drift, by developing a modified stochastic gradient descent method. The work in [38] introduces an algorithm which by taking into account users rating abstention intervals improves the rating prediction accuracy, proving that the periods of users rating inactivity may often indicate a shift of user interest.

Pruning techniques, based on the time the ratings have assigned to the items, have also been proposed [28, 39], in order to eliminate aged ratings from the ratings database, in the sense that they fail to reflect users' current trends, and hence they contribute to the formulation of high error predictions.

Recently, the work in [9] has introduced the Common Item Rating Past Criterion (CIRPaC), which increases the similarity between user U , for whom a prediction is being formulated, and his NN user V , when they have experienced and rated common content before assigning a rating to the item i for which a prediction is being formulated. Furthermore, the work in [10] proposes the "early adopters" algorithm which improves rating prediction quality, by taking into account the eagerness shown by consumers to purchase and rate items. The "early adopters" algorithm regulates the weight that each NN's V rating for an item i affects the rating prediction for that item formulated for a user U , depending on whether V and U exhibit similar levels of eagerness to adopt item i or not.

However, none of the above-mentioned works takes into account the information of the experiencing *period* (Early or Late) in which the users' usual experiencing time and the time of their rating to be predicted belong. The present paper fills this gap by presenting an algorithm that amplifies or diminishes the prediction value produced by a CF system, based on whether the aforementioned rating times belong to the item's *same (common) period* or not, respectively, and assesses its performance using (a) two similarity metrics, (b) two error metrics, as well as (c) seven widely used CF datasets.

3 The proposed algorithm

CF algorithms consist of two main steps for the formulation of a prediction for a user U . The first step is to identify U 's NNs (the set of users who will act as recommenders to user U), while the second step is to compute the personalized predictions for U , based on the ratings of these NNs.

The set of U 's NNs, denoted as NN_U , consists of users that have rated items similarly to U ; typically, the quantified rating similarity is required to exceed some threshold, e.g., the value 0 for the PCC. The similarity metric between two users U and V is typically quantified using the PCC [1, 2], which is expressed as follows:

$$sim_p(U, V) = \frac{\sum_k (r_{U,k} - \bar{r}_U) * (r_{V,k} - \bar{r}_V)}{\sqrt{\sum_k (r_{U,k} - \bar{r}_U)^2 * \sum_k (r_{V,k} - \bar{r}_V)^2}} \quad (1)$$

where k is the set of items both being rated by users U and V , while \bar{r}_U and \bar{r}_V are the mean values of ratings entered by users U and V in the rating database, respectively.

Similarly, the CS metric [1, 2] is expressed as follows:

$$sim_{cs}(U, V) = \frac{\sum_k r_{U,k} * r_{V,k}}{\sqrt{\sum_k (r_{U,k})^2} * \sqrt{\sum_k (r_{V,k})^2}} \quad (2)$$

Subsequently, in order to compute a rating prediction $p_{U,i}$ for the rating that user U would assign to item i , the following formula [1, 2] is applied:

$$p_{U,i} = \bar{r}_U + \frac{\sum_{V \in NN_U} sim(U, V) * (r_{V,i} - \bar{r}_V)}{\sum_{V \in NN_U} sim(U, V)} \quad (3)$$

where $sim(U, V)$ is either $sim_p(U, V)$ (equation 1) or $sim_{cs}(U, V)$ (equation 2), depending on the similarity metric utilised. The proposed algorithm modifies the prediction computation formula (3), by accommodating an adjustment that is based on the relation between (a) the experiencing period for item i that the computed prediction to be made belongs to and (b) the user-specific EWP for items. More specifically, the formula (3) is modified as follows:

$$p_{U,i}^t = \bar{r}_U + \frac{\sum_{V \in NN_U} sim(U, V) * (r_{V,i} - \bar{r}_V)}{\sum_{V \in NN_U} sim(U, V)} + EWP_adj(U, i, t) \quad (4)$$

where factor $EWP_adj(U, i, t)$ is an adjustment (positive/bonus or negative/malus) assigned to each prediction, based on (a) the time t at which the prediction is computed, (b) the lifespan of item i , and (c) user U 's usual EWP (short or prolonged) for item i 's category. The $EWP_adj(U, i, t)$ factor aims to try to align their recommendations with the users' patterns of practice, arranging so that users with small EWPs receive recommendations for newly released products, while refraining from presenting such recommendations to users with long EWPs; and conversely for products which have been released before a considerable amount of time.

The rationale behind the inclusion of the $EWP_adj(U, i, t)$ factor in the rating prediction formula is that when a user U usually waits for a period of time before experiencing items (cars, mobile phones or an innovative product, in general), in order to get useful product reviews from other users, an RS which recommends to him a product that it has just been released, will *probably* lead to an unsuccessful recommendation and vice versa. Note that the notation for the prediction in formula (4) is extended to include the time t at which the prediction is computed: this is in-line with the rationale presented above, since the perceived utility of the item to the user may increase or decrease along the time axis, since—for instance—a user categorized as an "innovator" is not expected to be interested in products that are already present in the market for a long time.

In order to compute the value of the $EWP_adj(U, i, t)$ factor, the following information is required:

- Each item’s effective lifespan EL_i , i.e., the period between the item’s release date rel_i , and the item’s end-of-life date eol_i . If the item’s release date or end-of-life dates are unavailable, they are estimated following the method introduced in [10], where the product’s lifespan is assumed to begin when the first rating assigned for this item enters the ratings database. Respectively, the end of the item’s effective lifespan is set to the timestamp of the last rating present in the database: for products that are actively consumed and rated, this will be close to the present time point, while for retired products this corresponds to the time point beyond which consumers have shown no interest for the product. Value normalization is performed according to the standard minmax formula.
- The *early experiencing period for each item i* , denoted as EEP_i , is a period that is deemed to be close to the beginning of the item’s effective lifespan. EEP_i starts at the item’s release date rel_i , whereas the end of EEP_i is set to a percentage of the duration of the item’s lifespan; this percentage is the *early experiencing period threshold (EEPT)*, and its optimal value is determined experimentally in Sect. 4. Formally,

$$EEP_i = [rel_i, rel_i + (eol_i - rel_i) * EEPT] \tag{5}$$

where $EEPT \in [0, 1]$.

- The *late experiencing period for each item i* , denoted as LEP_i , is a period that is deemed to be close to the end of the item’s effective lifespan. LEP_i ends at the item’s end-of-life date eol_i , while its beginning is set to a percentage of the duration of the item’s lifespan; this percentage is the *late experiencing period threshold (LEPT)*, and its optimal value is determined experimentally in Sect. 4. Formally

$$LEP_i = (eol_i - (eol_i - rel_i) * LEPT, eol_i] \tag{6}$$

If $(EEPT + LEPT = 1)$, then the item’s lifespan is partitioned into two experiencing periods, the early and the late one, with no time gap between them. If, however, $(EEPT + LEPT < 1)$, then a time gap exists between EEP_i and LEP_i , which is deemed to be a “neutral” period, in the sense that it is not classified either as early, or as late.

- The *normalized timestamp of rating $r_{U,i}$* is denoted as $NT(r_{U,i})$; this quantity refers to the amount of time that has elapsed until user U has rated the item i , normalized to the item i ’s effective lifespan EL_i . Formally, $NT(r_{U,i})$ is expressed as follows:

$$NT(r_{U,i}) = \frac{timestamp(r_{U,i}) - rel_i}{eol_i - rel_i} \tag{8}$$

Similarly, we can also compute the *normalized time for a prediction $p_{U,i}$* , by considering its timestamp to be equal to the time at which its computation took place.

- *User U ’s mean experiencing time (MET(U))*: this quantity refers to the mean value of the normalized timestamps of U ’s ratings. A value close to 0 indicates that U tends to experience items immediately as they become available (thus U belongs to the *innovators* category identified in [4]), while a value close to 1 indicates the exact opposite practice (classifying thus U to the *laggards* category listed in [4]). Formally, $MET(U)$ is expressed as follows:

$$MET(U) = \frac{\sum_{r \in R(U)} NT(r_{U,i})}{|R_U|} \tag{9}$$

where $R(U)$ denotes the set of ratings that have been entered by user U .

Using the quantities above, the $EWP_adj(U, i, t)$ factor is defined as shown in Eq. (10):

$$EWP_adj(U, i, t) = \begin{cases} CEPB, & \text{if } (t \in EEP_i \wedge MET(U) < EEPT) \vee (t \in LEP_i \wedge MET(U) > 1 - LEPT) \\ -DEPM, & \text{if } (t \in EEP_i \wedge MET(U) > 1 - LEPT) \vee (t \in LEP_i \wedge MET(U) < EEPT) \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $LEPT \in [0, 1]$. Additionally, since EEP_i and LEP_i must not overlap, it follows that

$$EEPT + LEPT \leq 1 \tag{7}$$

Effectively, the first branch of Eq. (10) computes a positive value, equal to $CEPB$ (Common Experiencing Period Bonus) for factor $EWP_adj(U, i, t)$ when either (a) the time at which the prediction is computed falls

Table 1 Algorithm parameters and notations

Parameter/notation	Description
$EWP_adj(U,i,t)$	An adjustment for the prediction of $r_{U,i}$, with the prediction being computed at time t
EL_i	The effective lifespan of item i
EEP_i	The early experiencing period for item i
EEPT	The early experiencing period threshold
LEP_i	The late experiencing period for item i
LEPT	The late experiencing period threshold
$NT(r_{U,i})$	The timestamp of rating $r_{U,i}$ normalized to the item's effective lifespan EL_i
$MET(U)$	The mean experiencing time for user U
CEPB	The Common Experiencing Period Bonus, boosting a predicted rating's value
DEPM	The Diverse Experiencing Period Malus, demoting a predicted rating's value

within the item's early experiencing period *and* the user has exhibited an eagerness to experience items or (b) the time at which the prediction is computed falls within the item's late experiencing period *and* the user has exhibited un-eagerness to experience items. Thus, the predictions for "fresh" items are boosted for "early adopters," and so are the predictions for "mature" items for "late adopters."

Conversely, the second branch computes a negative value, equal to *DEPM* (Dissimilar Experiencing Period Malus) for factor $EWP_adj(U, i, t)$ when either (a) the time at which the prediction is computed falls within the item's early experiencing period *and* the user has exhibited un-eagerness to experience item or (b) the time at which the prediction is computed falls within the item's late experiencing period *and* the user has exhibited an eagerness to

experience item. Thus, the predictions for "fresh" items are lowered for "late adopters," and so are the predictions for "mature" items for "early adopters."

Finally, the third branch concerns predictions that are formulated in item's "neutral" periods (i.e., neither within EEP_i and nor within LEP_i), as well as predictions formulated for users which are neither early nor late adopters; for such cases, the prediction rating computed by Eq. (3) is left intact.

Table 1 summarizes the algorithm parameters and notations introduced above.

Listing 1 presents the CF_{EPC} algorithm introduced in this paper, and more specifically the rating prediction formulation function (realizing formula (4)) as well as the computation of the EWP_adj parameter (formula (10)).

```

FUNCTION computePrediction(User X, Item it, TIME t)
/* Implementation of formula (4), used for the prediction formulation.
   INPUT: X is the user for whom the prediction will be computed; it is the respective item; t is the
   time at which the prediction is computed.
   OUTPUT: The rating prediction, or NULL if no NNs of X has rated it, and hence a prediction
   cannot be computed. */
   predictionNumerator = 0.0
   predictionDenominator = 0.0
   FOREACH Y ∈ NNX
     IF (rY,it ≠ NULL) THEN
       predictionNumerator += sim(X, Y) * (rY,i -  $\bar{r}_Y$ )
       predictionDenominator += sim(X, Y)
     ENDIF
   END /* FOREACH */
   IF (predictionDenominator = 0) THEN /* no NN of X has rated it */
     RETURN NULL
   ELSE /* at least one NN of X has rated item it. Compute the rating prediction, performing
     an adjustment according to the experiencing period. */
     return  $\bar{r}_X$  + (predictionNumerator / predictionDenominator) + EWP_adj(X, it, t)
   END
END /* FUNCTION */

FUNCTION EWP_adj (User U, Item i, Time t)
/* Implementation of formula (10), formulating the EWP adjustment assigned to each prediction.
   INPUT: U is the user for whom the EWP adjustment will be computed; i is the respective item; t is
   the time at which the prediction is computed.
   OUTPUT: the EWP adjustment computed. */

   /* reli is the item's release date (min timestamp concerning item i in the rating database), and eoli
   is the item's end-of-life date (max timestamp concerning item i in the rating database).
   The optimal value of the EEPT and LEPT parameters are determined experimentally in section 4.
   */
   EEPi = [reli, reli + (eoli - reli) * EEPT]
   LEPi = (eoli - (eoli - reli) * LEPT, eoli)

   NT(rU,i) = (timestamp(rU,i) - reli) / (eoli - reli) /* the normalized timestamp of rating rU,i */

   /* compute the mean experience time */
   sum_MET = 0
   ratings_count = 0
   FOREACH r ∈ R(U) /* the set of ratings that have been entered by user U */
     sum_MET = sum_MET + NT(rU,i)
     ratings_count++
   END /* FOREACH */

   MET(U) = sum_MET / ratings_count /* mean experiencing time of user U */
   IF ( (t ∈ EEPi AND MET(U) < EEPT) OR (t ∈ LEPi AND MET(U) > 1 - LEPT) ) THEN
     RETURN (CEPB)
   ELSE IF ( (t ∈ EEPi AND MET(U) > 1 - LEPT) OR (t ∈ LEPi AND MET(U) < EEPT) ) THEN
     RETURN (-DEPM)
   ELSE
     RETURN (0)
   END IF
END /* FUNCTION */

```

Listing 1. Rating prediction computation and computation of the EWP_adj adjustment.

For the application of this algorithm, the parameters *EEPT*, *LEPT*, *CEPB*, and *DEPM*, listed above, need to be determined. The setting of these parameters to their optimal values is explored experimentally in the following section, where the evaluation of the CF_{EPC} algorithm is also presented.

4 Algorithm tuning and experimental evaluation

In the following section, we report on our experiments aiming to:

1. Determine the optimal values for parameters *EEPT*, *LEPT*, *CEPB*, and *DEPM*, in order to tune the CF_{EPC} algorithm and
2. Assess the rating prediction accuracy achieved by the CF_{EPC} algorithm, to quantify the gains obtained due to the consideration of the Experiencing Period Criterion.
3. Compare the performance of the CF_{EPC} algorithm proposed in this paper, against the performance of the CF_{CIRPaC} algorithm presented in [9], and the CF_{EA} algorithm proposed in [10], which are also based only on the very basic CF information, comprising only user ratings on items, including the time of the rating, so that it can be used by every CF-based recommender system. In all comparisons, the performance of the plain CF algorithm is used as a baseline. Both the CF_{CIRPaC} and the CF_{EPC} algorithm are recently (2019) published state-of-the-art CF algorithms which additionally (a) aim at increasing the rating prediction accuracy through the exploitation of experiencing times and (b) necessitate no external information (e.g., consumers' text reviews or user relationships sourced from SNs).

In order to quantify the rating prediction quality, both the MAE and the RMSE [1, 2] error metrics have been employed, used in the majority of the works targeting at rating prediction computation, e.g., [40–44]. The reason behind the use of two error metrics is that while the MAE is a linear score (i.e., all the individual differences are weighted equally in the average), the RMSE is a quadratic scoring metric, which gives a relatively high weight to large errors.

In order to evaluate the algorithms' rating prediction MAE and RMSE scores, we exercised the standard “hide one” technique [1, 2, 9]: each time a particular rating in the database was hidden and then its value was tried to be predicted based on the values of other, non-hidden ratings; this process was repeated for each rating in the database.

Additionally, a second experiment was performed, where, for each user, only his last rating (based on the ratings' timestamps) was hidden, and again its value was predicted based on the other, non-hidden, ratings. The above two experiments gave very similar results (the differences observed were less than 2.5% in all cases); hence, we report only on the results of the first experiment, for conciseness purposes. Furthermore, we believe that the first experiment gives a more comprehensive insight into the algorithm's performance since the latter experiment is biased to computing predictions that lie within the items' LEP.

To validate the significance of the MAE and RMSE reduction results, we conducted statistical significance tests, across all datasets, between the CF_{EPC} algorithm, presented in this paper and each one of the CF_{CIRPaC} and CF_{EA} algorithms, used in the comparison.

Furthermore, in this paper, two additional error metrics have been employed for further quantifying the rating prediction quality: (a) the percentage of the cases an algorithm formulates the prediction closest to the real rating, namely “closest prediction,” and (b) the above-recommender-threshold F1-measure values of the formulated predictions, following the approach utilized in [45, 46].

All our experiments were executed on seven datasets. Five of these datasets have been obtained from Amazon [47, 48], while the remaining two have been sourced from MovieLens [49, 50]. Some of the datasets are relatively sparse (such as the Amazon “Books,” whose density is equal to 0.004%), while some of them are relatively dense (such as the Amazon “Office Supplies,” with its density is equal to 0.448%). The reason behind testing both sparse and dense datasets is to establish the applicability of the CF_{EPC} algorithm in every dataset. It has to be mentioned that we have used the exact same datasets against which the CF_{CIRPaC} [9] and CF_{EA} [10] algorithms were tested; however, in this paper, we have used their respective 5-core datasets (in which each of the remaining users and items has at least 5 ratings). The reason for this change is that in many cases (5–15% in each of the Amazon datasets) predictions could not be formulated due to the fact that the rating to be predicted was the only one recorded for that particular item within the database, hence neither the *EEP* nor the *LEP* could be set. Furthermore, if we include the cases that only two ratings of a particular item exist, which constitute an extreme situation, where only the *DEPM* will be given in the prediction formulation (since the rating with the earliest timestamp belongs to the item's *EEP*, while the other rating belongs to its relative *LEP*), the overall problematic cases are calculated at 10–25% in each of the Amazon datasets, making these datasets untrustworthy for evaluation. Hence, in order to eliminate these cases, the 5-core datasets were used (where for each rating to be

predicted, there are at least 4 more ratings on this item, enough for the EEP and LEP to be successfully set). It is worth noting that in the experiments reported for the CF_{CIRPaC} algorithm [9], the datasets were preprocessed to remove users having less than 10 ratings each, while this practice was also used for the experiments performed for the evaluation of the CF_{EA} algorithm [51].

The seven datasets used in our experiments are summarized in Table 2, while they also exhibit the following properties:

1. They contain each rating's timestamp, the existence of which is essential for the operation of the CF_{EPC} algorithm,
2. They are contemporary (published between 1996 and 2019),
3. They are widely used as benchmarking datasets in CF research and
4. They differ in regard to the category of product domain of the dataset (books, music, food, videogames, etc.) and size (ranging from 1.4 MB to 216 in plain text format).

For our experiments, we used a laptop equipped with an Intel N5000 @ 1.1 GHz CPUs with 8 GB of RAM and a 256 GB SSD with a transfer rate of 560MBps, which hosted the seven datasets, used in our work, and ran the rating prediction algorithms.

4.1 Determining the algorithm's parameters

The goal of the first experiment is to determine the optimal values for parameters EEPT, LEPT, CEPB, and DEPM, used in the CF_{EPC} algorithm. In order to find the optimal setting for the aforementioned parameters, we explored different combinations of values for them. In total, more than 50 candidate setting combinations were examined; however, for conciseness purposes, we report only on the

most indicative ones. More specifically, for each setting combination, we present the prediction accuracy improvement achieved, in terms of the MAE and the RMSE scores.

It has to be mentioned that the results were found to be relatively consistent across the seven datasets tested (in the majority of the combinations tested, the ranking of parameter value combinations was the same across all datasets); hence, in this subsection, we only present the average values of the respective error metrics for all datasets.

Figure 1 illustrates the rating prediction error reduction (using both the MAE and the RMSE error metrics) under different parameter value combinations when similarity is measured using the PCC similarity metric.

In Fig. 1, we can observe that the setting where $EEPT = 50\%$, $LEPT = 50\%$ (meaning that no "neutral" period will be set), $CEPB = 0.6$, and $DEPM = 0.6$ is the one achieving the largest prediction error reductions, for both error quantification metrics (MAE and RMSE). In more detail, the MAE drops by 5.81%, on average, and the RMSE is reduced by 5.04%. It has to be mentioned that this setting has proven to be the optimal one, in all datasets tested, among the eight settings illustrated in Fig. 1 (this is also true among settings that were tested but not shown). In the three last settings depicted in Fig. 1, EEPT and LEPT are both set to 1.0 and 0, respectively, and therefore all ratings and predictions are artificially considered as "early," and therefore the CEPB bonus is added to all predictions; however, the value of the bonus depends on the distance between the user's mean normalized experience time and the prediction's normalized time: when this distance is zero (i.e., the rating prediction is computed at exactly the time point that the user experiences items, on average), the bonus takes its maximum value (0.5, 0.7 and 1.0, for each of the three cases, respectively). Conversely, when the distance increases, the value of the bonus is

Table 2 Datasets summary

Dataset name	#Users (K)	#Items (K)	#Ratings	Avg.#Ratings/ User	Density (%)	DB size (in text format) (MB)
Amazon "Video-games" [47, 48]	24	11	232 K	9.7	0.089	5
Amazon "CDs and Vinyl" [47, 48]	75	64	1.1 M	14.7	0.023	25
Amazon "Movies and TV" [47, 48]	124	50	1.7 M	13.7	0.027	40
Amazon "Books" [47, 48]	604	368	8.9 M	14.7	0.004	216
Amazon "Digital Music" [47, 48]	5.5	3.5	65 K	11.8	0.327	1.4
MovieLens "Latest 100 K—Recommended for education and development" [49, 50]	670	9	100 K	166	1.85	2.2
MovieLens "Latest 20 M—Recommended for new research" dataset [49, 50]	138	27	20 M	144.9	0.537	486

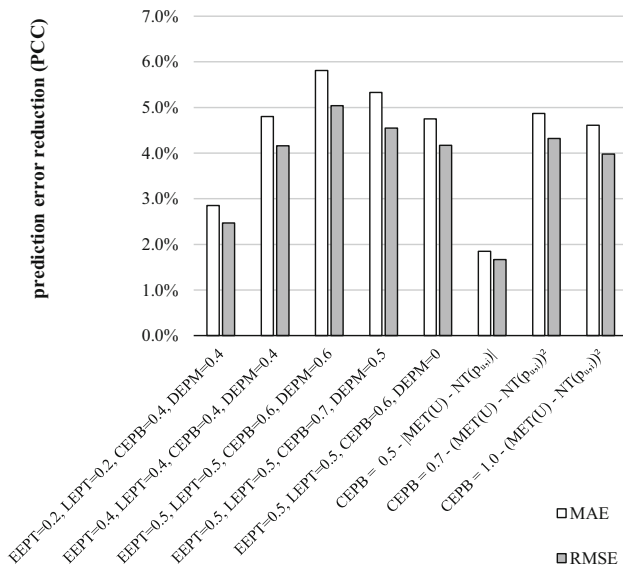


Fig. 1 Prediction error reduction under different parameter value combinations, when using the PCC as the similarity metric

reduced and may even assume negative values (except for the third case, where the value of CEPB is non-negative).

Similarly, when the CS similarity metric is used, the exact same setting is the one that delivers the highest reductions in prediction errors, again, for both MAE and RMSE quantification metrics; using this setting, the respective reductions are 5.64% and 4.98%, as shown in Fig. 2.

When using the CS similarity metric, the setting where the EEPT = 50%, the LEPT = 50%, the CEPB = 0.6, and the DEPM = 0.6 is ranked again first in all seven datasets tested, among the eight settings illustrated in Fig. 2 (this is also true among settings that were tested but not shown).

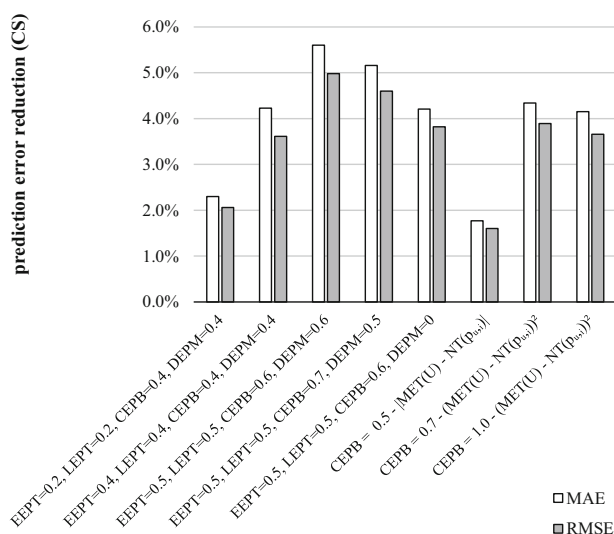


Fig. 2 Prediction error reduction under different parameter value combinations, when using the CS similarity metric

4.2 Comparison with previous work

After having determined the optimal parameter values for the operation of the CF_{EPC} algorithm, we proceed in evaluating the algorithm’s performance in terms of rating prediction accuracy, using the seven datasets listed in Table 2, using the performance of the plain CF algorithm as a baseline. Besides obtaining absolute metrics regarding improvements in the prediction accuracy achieved by the CF_{EPC} algorithm, presented in this paper, we compare its performance against the performance of the CF_{CIRPaC} and the CF_{EA} algorithms, introduced in [9, 10], respectively. As noted above, both CF_{CIRPaC} and CF_{EA} (a) are state-of-the-art algorithms targeting the improvement of rating prediction accuracy in the context of CF, (b) do not need extra information, regarding the users or the items (e.g., item categories or user social relationships), and hence can be applied in every CF-based RS, and (c) do not deteriorate the prediction coverage (i.e., the cases for which a personalized prediction can be formulated).

In the following paragraphs, we report on our findings regarding the aforementioned, using (1) a CF implementation, which employs the PCC user similarity metric, and (2) a CF implementation, which employs the CS user similarity metric.

4.2.1 Comparison using the PCC similarity metric

Figure 3 illustrates the improvement in the MAE achieved by the CF_{EPC} algorithm, when compared to the CF_{CIRPaC} [9] and CF_{EA} [10] algorithms, taking the performance of the plain CF algorithm as a baseline and using the PCC as the similarity metric.

We can clearly notice that the CF_{EPC} algorithm, presented in this paper, achieves an average prediction MAE

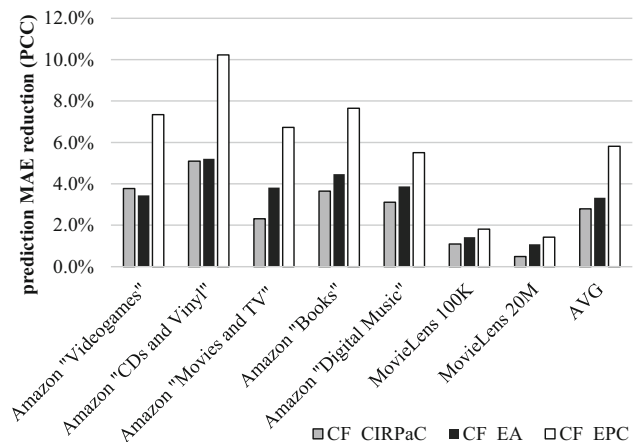


Fig. 3 MAE reduction achieved by the proposed algorithm, in comparison to the CF_{CIRPaC} [9] and CF_{EA} [10] algorithms, when using the PCC similarity metric

reduction equal to 5.81%, exceeding by approximately 110% the mean improvement attained by the CF_{CIRPaC} algorithm [9], while the corresponding improvement against the CF_{EA} algorithm is approximately 75%. At the individual dataset level, the performance edge of the proposed algorithm against the CF_{CIRPaC} algorithm ranges from 66% (for the MovieLens 100 K dataset) to 190% (for both the Amazon “Movies and TV” and the MovieLens 20 M datasets), while CF_{EPC} outperforms the CF_{EA} algorithm by a margin ranging from 30% (for the MovieLens 100 K dataset) to 210% (for the Amazon “Videogames” dataset).

Figure 4 illustrates the respective improvement in the RMSE achieved by the proposed algorithm, when compared to the CF_{CIRPaC} [9] and CF_{EA} [10] algorithms, again taking the performance of the plain CF algorithm as a baseline and using the PCC as the similarity metric.

We can again notice that the CF_{EPC} algorithm, presented in this paper, achieves an average prediction RMSE reduction equal to 5.04%, exceeding by 90% the performance of the CF_{CIRPaC} algorithm [9], on average; the respective improvement against the CF_{EA} algorithm is 70%. At the individual dataset level, the performance edge of the proposed algorithm against the CF_{CIRPaC} algorithm ranges from 47% (for the MovieLens 100 K dataset) to 190% (for the MovieLens 20 M dataset), while additionally CF_{EPC} outperforms CF_{EA} by a margin varying from 16% (for the MovieLens 100 K) to 110% (for the Amazon Videogames dataset).

To validate the significance of the MAE and RMSE reduction results, when using the PCC similarity metric, we conducted a statistical significance test on the results obtained for the three algorithms. First, an Anova analysis was performed to establish statistical significance regarding the performance of the three algorithms, while

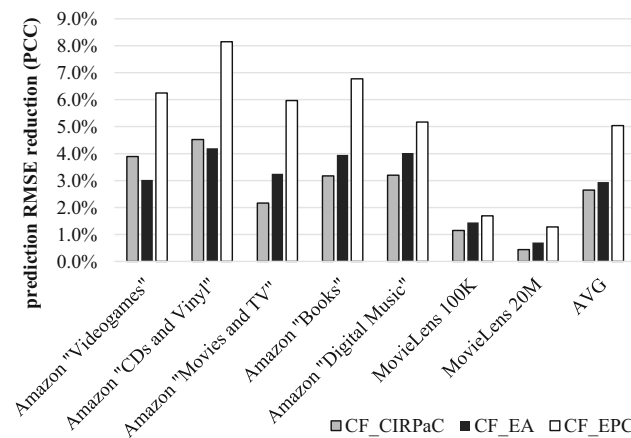


Fig. 4 RMSE reduction achieved by the proposed algorithm, in comparison to the CF_{CIRPaC} [9] and CF_{EA} [10] algorithms, when using the PCC similarity metric

subsequently post hoc Tukey HSD tests were applied to establish statistical significance between the results of CF_{EPC} and the results of each of the other two algorithms. The outcome of the statistical tests regarding the MAE and RMSE metrics are depicted in Tables 3 and 4, respectively. All *p*-values are less than 0.01, establishing statistical significance regarding the observed differences of the algorithms’ performance.

Figure 5 illustrates the percentage of cases where each of the three algorithms considered in the evaluation achieved the prediction closest to the real rating in each dataset tested, using the PCC as the similarity metric. Note that, in the case of a tie, i.e., when two (or all three) algorithms formulate the closest prediction, both (or all, respectively) algorithms are considered as achieving the closest prediction, and hence the sum of the scores of the three algorithms may exceed 100%.

The “Amazon Books” dataset exhibits a considerably high number of ties, approximately equal to 35% of the overall number of the rating predictions computed. This is owing to the fact that the distribution of ratings in this dataset is highly skewed, with 61.7% of the ratings being equal to 5.0 (the maximum rating). Consequently, in many cases two or more algorithms compute a rating prediction equal to 5.0, resulting in a tie. Similar skewing is also observed in all other Amazon-sourced datasets, however to a lesser extent.

We can observe that the CF_{EPC} algorithm, presented in this paper, manages to formulate the closest prediction to the real rating in 99.7% of the cases (ties included) across all the datasets tested. The lowest score for the CF_{EPC} algorithm is 98.92%, observed in the MovieLens 100 K dataset, while the highest score is 99.88%, observed for the MovieLens 20 M dataset. The CF_{CIRPaC} and the CF_{EA} algorithms achieve significantly lower scores, equal to 7.3% and 7.2%, respectively.

Figure 6 depicts the above-recommender-threshold F1-measure values for the three algorithms considered in this experiment, adapting the work in [45, 46]. In this experiment, items for which the rating prediction was above the threshold of 3.5/5 for the Amazon datasets, or 7/10 for the MovieLens datasets were considered to be a recommendation for the user; the information retrieval metrics *precision*, *recall*, and *F1-measure* were computed according to Eqs. (11)–(13) [45]:

$$precision = \frac{|\{relevantItems\} \cap \{recommendedItems\}|}{|\{recommendedItems\}|} \tag{11}$$

$$recall = \frac{|\{relevantItems\} \cap \{recommendedItems\}|}{|\{relevantItems\}|} \tag{12}$$

Table 3 Statistical tests regarding the MAE under the PCC similarity metric

Dataset name	Anova p -value	Post hoc Tukey HSD Test CF_{EPC} versus CF_{CIRPaC} p -value	Post hoc Tukey HSD Test CF_{EPC} versus CF_{EA} p -value
Amazon "Video-games"	4.23×10^{-6}	0.0028	0.0026
Amazon "CDs and Vinyl"	4.56×10^{-6}	0.0029	0.0034
Amazon "Movies and TV"	2.87×10^{-6}	0.0013	0.0043
Amazon "Books"	4.01×10^{-6}	0.0026	0.0045
Amazon "Digital Music"	4.68×10^{-6}	0.0030	0.0053
MovieLens "Latest 100 K—Recommended for education and development"	4.87×10^{-6}	0.0032	0.0059
MovieLens "Latest 20 M—Recommended for new research" dataset	2.96×10^{-6}	0.0014	0.0056

Table 4 Statistical tests regarding the RMSE under the PCC similarity metric

Dataset name	Anova p -value	Post hoc Tukey HSD Test CF_{EPC} versus CF_{CIRPaC} p -value	Post hoc Tukey HSD Test CF_{EPC} versus CF_{EA} p -value
Amazon "Video-games"	4.89×10^{-6}	0.0035	0.0025
Amazon "CDs and Vinyl"	4.76×10^{-6}	0.0045	0.0035
Amazon "Movies and TV"	3.01×10^{-6}	0.0015	0.0046
Amazon "Books"	3.96×10^{-6}	0.0026	0.0047
Amazon "Digital Music"	4.97×10^{-6}	0.0035	0.0059
MovieLens "Latest 100 K—Recommended for education and development"	6.02×10^{-6}	0.0054	0.0072
MovieLens "Latest 20 M—Recommended for new research" dataset	2.43×10^{-6}	0.0012	0.0041

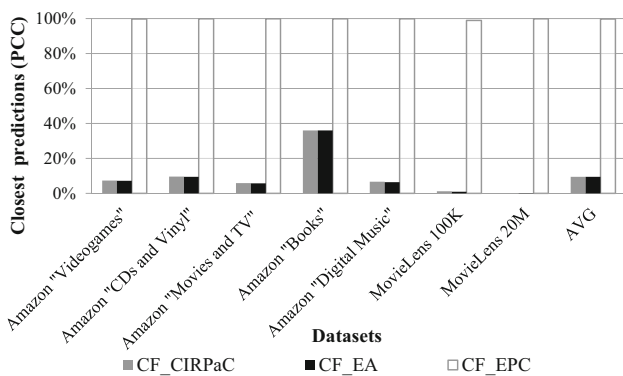


Fig. 5 Closest predictions to the real rating percentage achieved by the proposed algorithm, in comparison to the CF_{CIRPaC} [9] and CF_{EA} [10] algorithms, when using the PCC similarity metric

$$F1-measure = 2 * \frac{precision * recall}{precision + recall} \quad (13)$$

In all cases, relevant items are considered to be those that are known to be rated by the user above the threshold

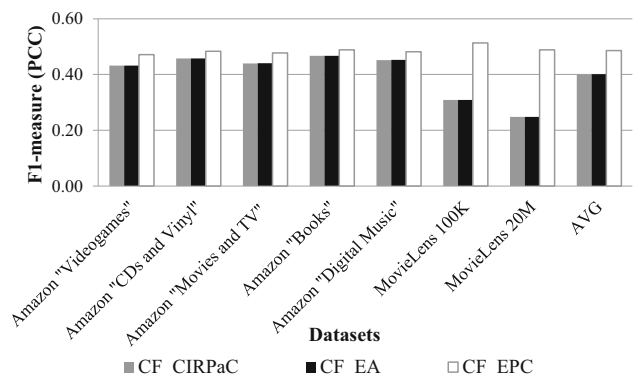


Fig. 6 F1-measure values for the predictions formulated, achieved by the proposed algorithm, in comparison to the CF_{CIRPaC} [9] and CF_{EA} [10] algorithms, when using the PCC similarity metric

(3.5/5 for the Amazon datasets or 7/10 for the MovieLens datasets).

We can observe that the proposed algorithm achieves a relative improvement of the F1-measure over both baseline algorithms of approximately 24% in relative terms (or by

9.9% in absolute figures). At the individual dataset level, the improvement ranges from 2% (absolute figures) for the Amazon Books dataset to 20% (absolute figures) for the MovieLens 100 K and the MovieLens 20 M datasets.

4.2.2 Comparison using the CS similarity metric

Figure 7 illustrates the improvement in the MAE achieved by the proposed algorithm, when compared to the CF_{CIRPaC} and CF_{EA} algorithms, again taking the performance of the plain CF algorithm as a baseline, however this time using the CS as the similarity metric.

We can clearly notice that the CF_{EPC} algorithm, presented in this paper, achieves an average prediction MAE reduction equal to 5.64%, exceeding by 124% the performance of the CF_{CIRPaC} algorithm, on average. In relation to the CF_{EA} algorithm, the CF_{EPC} algorithm achieves an MAE reduction higher by approximately 100%. At individual dataset level, the performance edge of the proposed algorithm against the CF_{CIRPaC} algorithm ranges from 70% (for the MovieLens 100 K dataset) to 275% (for the MovieLens 20 M dataset); the corresponding performance edge for the CF_{EA} algorithm ranges from 55% (for the MovieLens 100 K dataset) to 155% (for the Amazon “CDs and Vinyl” dataset).

Figure 8 illustrates the respective improvement in the RMSE achieved by the proposed algorithm, when compared to the CF_{CIRPaC} and the CF_{EA} algorithms, again taking the performance of the plain CF algorithm as a baseline and using the CS as the similarity metric.

The CF_{EPC} algorithm, presented in this paper, achieves an average prediction RMSE reduction equal to 4.98%, exceeding the performance of the CF_{CIRPaC} and the CF_{EA} algorithms by 120% and 90%, respectively. At the individual dataset level, the performance edge of the proposed

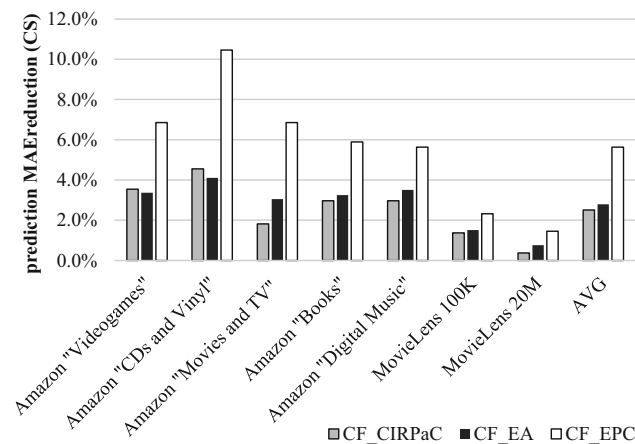


Fig. 7 MAE reduction achieved by the proposed algorithm, in comparison to the CF_{CIRPaC} [9] and CF_{EA} [10] algorithms, when using the CS similarity metric

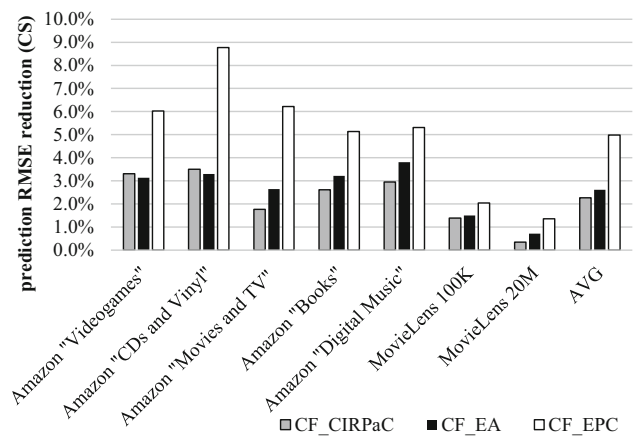


Fig. 8 RMSE reduction achieved by the proposed algorithm, in comparison to the CF_{CIRPaC} [9] and CF_{EA} [10] algorithms, when using the CS similarity metric

algorithm against the CF_{CIRPaC} algorithm ranges from 47% (for the MovieLens 100 K dataset) to 290% (for the MovieLens 20 M dataset); the respective performance edge of the CF_{EPC} algorithm against the CF_{EA} algorithm ranges from 40 (for the Amazon “Digital Music” dataset) to 165% (for the Amazon “CDs and Vinyl” dataset).

To validate the significance of the MAE and RMSE improvement results, when using the CS similarity metric, we conducted a statistical significance test on the results obtained for the three algorithms. Similarly to the case of the PCC (c.f. Sect. 4.2.1), first, an Anova analysis was performed to establish statistical significance regarding the performance of the three algorithms, while subsequently post hoc Tukey HSD tests were applied to establish statistical significance between the results of CF_{EPC} and the results of each of the other two algorithms. The outcome of the statistical tests regarding the MAE and RMSE metrics is depicted in Tables 5 and 6, respectively. All p -values are less than 0.01, establishing statistical significance regarding the observed differences of the algorithms’ performance.

Figure 9 illustrates the percentage of cases where each of the three algorithms considered in the evaluation achieved the prediction closest to the real rating in each dataset tested, using the CS as the similarity metric. Note that, in the case of a tie, i.e., when two (or all three) algorithms formulate the closest prediction, both (or all, respectively) algorithms are considered as achieving the closest prediction and hence the sum of the scores of the three algorithms may exceed 100%.

We can clearly notice that the CF_{EPC} algorithm, presented in this paper, manages to formulate the closest prediction in 99.8% of the cases (ties included) across all the datasets tested, in contrast with both of the CF_{CIRPaC} and the CF_{EA} algorithm, whose closest prediction cases are

Table 5 Statistical tests regarding the MAE under the PCC similarity metric

Dataset name	Anova p -value	Post hoc Tukey HSD Test CF_{EPC} versus CF_{CIRPaC} p -value	Post hoc Tukey HSD Test CF_{EPC} versus CF_{EA} p -value
Amazon "Video-games"	4.58×10^{-6}	0.0029	0.0025
Amazon "CDs and Vinyl"	2.71×10^{-6}	0.0024	0.0018
Amazon "Movies and TV"	1.46×10^{-6}	0.0003	0.0029
Amazon "Books"	4.03×10^{-6}	0.0028	0.0032
Amazon "Digital Music"	3.02×10^{-6}	0.0030	0.0053
MovieLens "Latest 100 K—Recommended for education and development"	4.76×10^{-6}	0.0031	0.0056
MovieLens "Latest 20 M—Recommended for new research" dataset	1.22×10^{-6}	0.0003	0.0028

Table 6 Statistical tests regarding the RMSE under the PCC similarity metric

Dataset name	Anova p -value	Post hoc Tukey HSD Test CF_{EPC} versus CF_{CIRPaC} p -value	Post hoc Tukey HSD Test CF_{EPC} versus CF_{EA} p -value
Amazon "Video-games"	4.08×10^{-6}	0.0029	0.0028
Amazon "CDs and Vinyl"	2.57×10^{-6}	0.0017	0.0016
Amazon "Movies and TV"	1.23×10^{-6}	0.0004	0.0027
Amazon "Books"	2.99×10^{-6}	0.0027	0.0053
Amazon "Digital Music"	4.81×10^{-6}	0.0033	0.0078
MovieLens "Latest 100 K—Recommended for education and development"	6.21×10^{-6}	0.0060	0.0082
MovieLens "Latest 20 M—Recommended for new research" dataset	1.26×10^{-6}	0.0003	0.0028

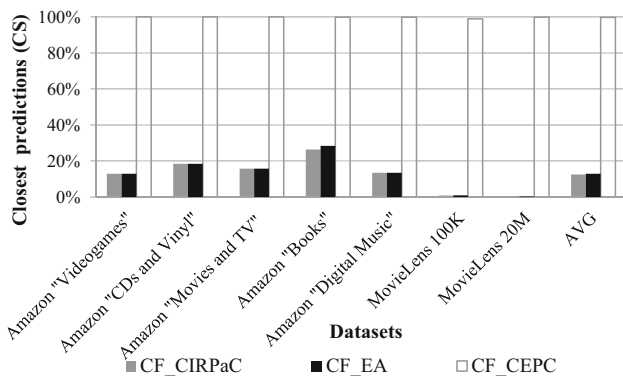


Fig. 9 Closest predictions percentage achieved by the proposed algorithm, in comparison to the CF_{CIRPaC} [9] and CF_{EA} [10] algorithms, when using the CS similarity metric

12.5% and 12.8%, respectively. The lowest score for the CF_{EPC} algorithm is 98.93%, observed in the MovieLens 100 K dataset, while the highest score is 99.99%, observed for the Amazon "CDs and Vinyl" and the Amazon "Movies and TV" datasets. The CF_{CIRPaC} and the CF_{EA}

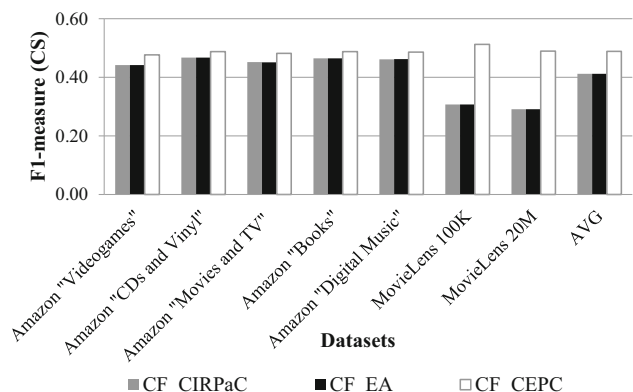


Fig. 10 F1-measure values for the predictions formulated, achieved by the proposed algorithm, in comparison to the CF_{CIRPaC} [9] and CF_{EA} [10] algorithms, when using the CS similarity metric

algorithms achieve significantly lower scores, equal to 12.45% and 12.82%, respectively. Again, we can observe that in all Amazon datasets a high number of ties occurs, owing to the data skew present in the datasets, as detailed in Sect. 4.2.1.

Finally, Fig. 10 depicts the above-recommender-threshold F1-measure values for the three algorithms considered in this experiment, following the same methodology described in Sect. 4.2.1. We can observe that the proposed algorithm achieves a relative improvement of the F1-measure over both baseline algorithms of approximately 18.5% (7.7% in absolute figures). At the individual dataset level, the improvement ranges from 2% (absolute figures) for the Amazon Books and the Amazon “Digital Music” datasets to 20% (absolute figures) for the MovieLens 100 K and the MovieLens 20 M datasets.

Overall, the previous experiments clearly indicate that the presented CF_{EPC} algorithm surpasses the performance of both the CF_{CIRPaC} [9] and CF_{EA} algorithms [10], in all datasets and under both similarity metrics. It is worth also noting that the CF_{CIRPaC} algorithm presented in [9] and the CF_{EA} algorithm [10] have both been shown to surpass the performance of other state-of-the-art algorithms, such as the ones in [26, 28].

5 Conclusions and future work

In this paper, we have presented a novel CF algorithm, namely CF_{EPC} , which considers the information of the users’ *Experiencing Period* in the CF prediction process, in order to improve rating prediction accuracy.

The proposed algorithm exhibits promising results and outperforms recently published algorithms. This is achieved through the amplification of the prediction value when both the user’s *usual experiencing time* and the time of the rating to be predicted belong to the same *experiencing period* (Early or Late) of the item and, conversely, reduces it when they belong to different *experiencing periods*. The rationale behind the usage of the aforementioned information is that when a user U usually prefers to wait for a period of time before experiencing items (until a satisfactory amount of feedback becomes available for the item of interest), it is essential for an RS to align to this practice and not recommend to him brand new products, which have just been released, but rather recommend products that have been in the market for a suitable amount of time. And, similarly, when a recommendation is offered to an “early adopter” user, the RS should include in this recommendation “fresh” products, rather than products that the user will consider as outdated.

The presented algorithm has been experimentally validated through a set of experiments, using two user similarity metrics and seven datasets of multiple domains (books, music, food, etc.) and the evaluation results have shown that the inclusion of the *experiencing period criterion* introduces significant prediction accuracy gains.

More specifically, the experimental results have shown that the CF_{EPC} algorithm achieves a considerable MAE reduction of 5.8%, on average (ranging from 1.4% to 10.2%), when selecting the PCC similarity metric, and an MAE reduction of 5.64% on average (ranging from 1.45% to 10.46%) when selecting the CS similarity metric. The respective average RMSE reductions found to be 5.04% (ranging from 1.28% to 8.15%), when selecting the PCC similarity metric, and 4.98% (ranging from 1.36% to 8.77%), when selecting the CS similarity metric (in all the aforementioned percentages, the plain CF algorithm was used as a baseline).

We have also compared the performance of the CF_{EPC} algorithm against two recently published (2019) state-of-the-art algorithms, which also target at prediction error reduction by considering item experiencing times, namely the CF_{CIRPaC} [9] and the CF_{EA} [10] algorithms. The proposed algorithm has exhibited superior performance against both algorithms included in the comparison, in all cases, managing to formulate the closest prediction in 99.7% of the cases (ties included) across all the datasets tested.

More specifically in the comparison against the CF_{CIRPaC} algorithm [9], the CF_{EPC} algorithm, proposed in this work, has proved to consistently outperform the CF_{CIRPaC} algorithm across all datasets tested, achieving an average MAE reduction across all datasets 110% higher than the reduction attained by the CF_{CIRPaC} algorithm, under the PCC similarity metric; the corresponding performance edge against the CF_{EA} algorithm is 75%. The CF_{EPC} algorithm has been found to perform better than the CF_{CIRPaC} and the CF_{EA} algorithms in each individual test made, i.e., for every combination of test dataset, prediction error measure, or similarity metric.

Furthermore, statistical significance testing, across all datasets, between the CF_{EPC} algorithm, presented in this paper and each one of the two aforementioned algorithms, indicated that the proposed algorithm is shown to be statistically significant with a confidence interval of 95% with both the CF_{CIRPaC} and CF_{EA} algorithms, under both similarity metrics.

Lastly, we evaluated the proposed algorithm using the F1-measure, typically employed to assess recommendation quality, and the CF_{EPC} algorithm was found to achieve an overall F1-measure score of 0.486, when using the PCC similarity metric, and an overall F1-measure score of 0.489, when using the CS similarity metric, surpassing the performance of both baseline algorithms.

It is worth noting that the CF_{EPC} algorithm can be combined with an already implemented CF-based recommender system, since (1) it is easy to be implemented, through the modification of existed CF-based RSs, (2) it needs no extra information about neither the users nor the

items, (3) it needs minimal additional dataset pre-processing time, computing only each items' first and last rating times, as well as each user's usual-average rating time, and minimal extra storage space, storing the aforementioned information, and (4) it can be easily combined with other algorithms that have been proposed for improving rating prediction accuracy, coverage, computational efficiency or recommendation quality (e.g., [52, 11, 12, 5, 6, 18–20]).

Our future work will focus on further studying the users' usual experiencing phases in rating prediction computation, as well as exploring alternative algorithms for reducing prediction error in CF datasets. Furthermore, we are planning to evaluate the algorithms' performance under more user similarity metrics, such as the Spearman coefficient and the Euclidean distance [53–55], where those are proposed by the literature as more suitable for either the item category or the additional information. Adaptation of the proposed algorithm for use with MF approaches [56–58] is also considered.

Finally, the combination of the proposed method with other algorithms, such as algorithms who exploit social network information [59–61] and IoT information [62–64], where available, for further improving the quality of recommendations formulated.

Compliance with ethical standards

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

- Balabanovic M, Shoham Y (1997) Fab: content-based, collaborative recommendation. *Commun ACM* 40(3):66–72
- Ekstrand M, Riedl R, Konstan J (2011) Collaborative filtering recommender systems. *Found Trends Hum-Comput Interact* 4(2):81–173
- Tobbin P, Adjei J (2012) Understanding the characteristics of early and late adopters of technology: the case of mobile money. *Int J E-Serv Mob Appl* 4(2):37–54
- Mahajan V, Muller E, Srivastava RK (1990) Determination of adopter categories by using innovation diffusion models. *J Mark Res* 27(1):37–50
- Kalaï A, Zayani CA, Amous I, Abdelghani W, Sèdes F (2018) Social collaborative service recommendation approach based on user's trust and domain-specific expertise. *Fut Gen Comput Syst* 80:355–367
- Wang X, Zhu W, Liu C (2019) Social recommendation with optimal limited attention. In: 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1518–1527
- Mattila M, Karjaluo H, Pentto T (2003) Internet banking adoption among mature customers: early majority or laggards? *J Serv Mark* 17(5):514–528
- Martinez E, Polo Y, Flavián C (1998) The acceptance and diffusion of new consumer durables: differences between first and last adopters. *J Consumer Mark* 15(4):323–342
- Margaris D, Vasilopoulos D, Vassilakis C, Spiliotopoulos D (2019) Improving collaborative filtering's rating prediction accuracy by introducing the common item rating past criterion. In: 2019 10th IEEE international conference on information, intelligence, systems and applications (IISA), pp 1–8
- Margaris D, Spiliotopoulos D, Vassilakis C (2019) Improving collaborative filtering's rating prediction quality by exploiting the item adoption eagerness information. In: 2019 IEEE/WIC/ACM international conference on web intelligence (WI 2019), pp 342–347
- Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2013) A survey on concept drift adaptation. *ACM Comput Surv* 1(1), Article 1
- Sun Y, Shao H, Wang S (2019) Efficient ensemble classification for multi-label data streams with concept drift. *Information* 10(5):158
- Ahmad HS, Nurjanah D, Rismala R (2017) A combination of individual model on memory-based group recommender system to the books domain. In: 2017 5th international conference on information and communication technology (ICoICT7), pp 1–6
- Naz S, Maqsood M, Durani MY (2019) An efficient algorithm for recommender system using kernel mapping techniques. In: 2019 8th international conference on software and computer applications (ICSCA '19), pp 115–119
- Gong S (2010) A collaborative filtering recommendation algorithm based on user clustering and item clustering. *J Softw* 5(7):745–752
- Logesh R, Subramaniaswamy V, Malathi D, Sivaramakrishnan N, Vijayakumar V (2019) Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method. *Neural Comput Appl* 32:2141–2164
- Pham M, Cao Y, Klamma R, Jarke M (2011) A clustering approach for collaborative filtering recommendation using social network analysis. *J Univ Comput Sci* 17(4):583–604
- Vozalis M, Markos A, Margaritis K (2009) A hybrid approach for improving prediction coverage of collaborative filtering. *Artif Intell Appl Innov* 296:491–498
- Yang X, Zhou S, Cao M (2019) An approach to alleviate the sparsity problem of hybrid collaborative filtering based recommendations: the product-attribute perspective from user reviews. In: *Mobile networks and applications*, pp 1–15
- Zhang S, Yao L, Xu X (2017) Autosvd++: An efficient hybrid collaborative filtering model via contractive auto-encoders. In: *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pp 957–960
- Nilashi M, Ibrahim O, Bagherifard K (2018) A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Syst Appl* 92:507–520
- Zhang F, Gong T, Lee VE, Zhao G, Rong C, Qu G (2016) Fast algorithms to evaluate collaborative filtering recommender systems. *Knowl-Based Syst* 96:96–103
- Zou H, Chen C, Zhao C, Yang B, Kang Z (2019) Hybrid collaborative filtering with semi-stacked denoising autoencoders for recommendation. In: 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech), pp 87–93
- Hasan M, Ahmed S, Malik M, Ahmed S (2016) A comprehensive approach towards user-based collaborative filtering recommender system. In: 2016 international workshop on computational intelligence (IWCI), pp 159–164

25. Chen T, Sun Y, Shi Y, Hong L (2017) On sampling strategies for neural network-based collaborative filtering. In: 23rd ACM SIGKDD international conference on knowledge discovery and data mining (KDD '17), pp 767–776
26. Margaris D, Vassilakis C (2018) Improving collaborative filtering's rating prediction accuracy by considering users' rating variability. In: 2018 IEEE 16th Intl Conf on dependable, autonomous and secure computing, 16th intl conf on pervasive intelligence and computing, 4th intl conf on big data intelligence and computing and cyber science and technology congress (DASC/PiCom/DataCom/CyberSciTech), pp 1022–1027
27. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. *IEEE Comput* 42(8):42–49
28. Margaris D, Vassilakis C (2017) Enhancing user rating database consistency through pruning. *Trans Large-Scale Data- Knowledge-Centered Syst XXXIV*:33–64
29. Wen H, Ding G, Liu C, Wang J (2014) Matrix factorization meets cosine similarity: addressing sparsity problem in collaborative filtering recommender system. *Proc APWeb 2014*:306–317
30. Colombo-Mendoza LO, Valencia-García R, Rodríguez-González A, Alor-Hernández C, Samper-Zapatero J (2015) RecomMetz: a context aware knowledge-based mobile recommender system for movie showtimes. *Expert Syst Appl* 42(3):1202–1222
31. Margaris D, Vassilakis C, Georgiadis P (2017) Knowledge-based leisure time recommendations in social networks. In: *Current trends on knowledge-based systems: theory and applications*, pp 23–48
32. Zarei MR, Moosavi MR (2019) A memory-based collaborative filtering recommender system using social ties. In: 4th international conference on pattern recognition and image analysis (IPRIA), pp 263–267
33. Margaris D, Spiliotopoulos D, Vassilakis C (2019) Social relations versus near neighbours: reliable recommenders in Limited Information Social Network Collaborative Filtering for online advertising. In: *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM 2019)*, pp 1–8
34. Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2013) A survey on concept drift adaptation. *ACM Comput Surv* 46(4):44
35. Zhang Q, Wu D, Zhang G, Lu J (2016) Fuzzy user-interest drift detection based recommender systems. In: 2016 IEEE international conference on fuzzy systems (FUZZ-IEEE), pp 1274–1281
36. Wang C, Chen G, Wei Q, Liu G, Guo X (2019) Personalized promotion recommendation through consumer experience evolution modeling. In: 2019 international fuzzy systems association world congress (IFSAC 2019), pp 692–703
37. Lo Y, Liao W, Chang C, Lee Y (2018) Temporal matrix factorization for tracking concept drift in individual user preferences. *IEEE Trans Comput Soc Syst* 5(1):156–168
38. Margaris D, Vassilakis C (2018) Exploiting rating abstention intervals for addressing concept drift in social network recommender systems. *Informatics* 5(2):21
39. Margaris D, Vassilakis C (2017) Improving collaborative filtering's rating prediction quality in dense datasets, by pruning old ratings. In: 2017 IEEE symposium on computers and communications (ISCC), pp 1168–1174
40. Ma X, Lei X, Zhao G, Qian X (2018) Rating prediction by exploring user's preference and sentiment. *Multimed Tools Appl* 77(6):6425–6444
41. Chen L, Liu Y, Zheng Z, Yu P (2018) Heterogeneous neural attentive factorization machine for rating prediction. In: 27th ACM international conference on information and knowledge management, pp 833–842
42. Sejwal VK, Abulaish M (2019) Trust and context-based rating prediction using collaborative filtering: a hybrid approach. In: *Proceedings of the 9th international conference on web intelligence, mining and semantics*, pp 1–10
43. Li Y, Liu J, Ren J, Chang Y (2020) A novel implicit trust recommendation approach for rating prediction. *IEEE Access* 8:98305–98315
44. Bell RM, Koren Y (2007) Lessons from the Netflix prize challenge. *ACM SIGKDD Explor Newsl* 9(2):75–79
45. Felfernig A, Boratto L, Stettinger M, Tkalčič M (2018) Evaluating group recommender systems. In: *Group recommender systems*, pp 59–71
46. AlEroud A, Karabatis G (2017) Using contextual information to identify cyber-attacks. In: *Information fusion for cyber-security analytics*, pp 1–16
47. Amazon product data. Available online: <http://jmcauley.ucsd.edu/data/amazon/links.html>. Accessed 4 April 2019
48. McAuley JJ, Pandey R, Leskovec J (2015) Inferring networks of substitutable and complementary products. In: *Proceedings of the 21th ACM SIGKDD conference*, pp 785–794
49. MovieLens datasets. Available online: <http://grouplens.org/datasets/movielens/>. Accessed 4 April 2019
50. Harper FM, Konstan JA (2015) The movielens datasets: history and context. *ACM Trans Interact Intell Syst* 5(4):19
51. Margaris D, Spiliotopoulos D, Vassilakis C (2019) Experimental results for considering item adoption eagerness information in collaborative filtering's rating prediction. *Software and database systems lab technical report TR-19003*, <https://soda.dit.uop.gr/?q=TR-19003>. Accessed 4 January 2020
52. Lumauag R, Sison AM, Medina R (2019) An enhanced recommendation algorithm based on modified user-based collaborative filtering. In: 2019 IEEE 4th international conference on computer and communication systems (ICCCS), pp 198–202
53. Cheng W, Zhu X, Chen X, Li M, Lu J, Li P (2019) Manhattan distance based adaptive 3D transform-domain collaborative filtering for laser speckle imaging of blood flow. *IEEE Trans Med Imaging* 38(7):1726–1735
54. Jiang S, Fang S, An Q, Lavery JE (2019) A sub-one quasi-norm-based similarity measure for collaborative filtering in recommender systems. *Inf Sci* 487:142–155
55. Korhonen J (2019) Assessing Personally perceived image quality via image features and collaborative filtering. In: 2019 IEEE conference on computer vision and pattern recognition, pp 8169–8177
56. Cui J, Lu T, Li D, Gu N (2019) Matrix approximation with cumulative penalty for collaborative filtering. In: 2019 IEEE 23rd international conference on computer supported cooperative work in design (CSCWD), pp 458–463
57. Fernández-Tobías I, Cantador I, Tomeo P, Anelli VW, DiNoia T (2019) Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization. *User Model User-Adap Inter* 29(2):443–486
58. Silva J, Varela N, Lezama OBP, Hernández H, Ventura JM, de la Hoz B, Coronel LP (2019) Multi-dimension tensor factorization collaborative filtering recommendation for academic profiles. In: 2019 International symposium on neural networks, pp 200–209
59. Hassan T (2019) Trust and Trustworthiness in social recommender systems. In: 2019 World Wide Web Conference, pp 529–532
60. Madani Y, Erritali M, Bengourram J, Salihan F (2019) Social collaborative filtering approach for recommending courses in an e-learning platform. *Procedia Comput Sci* 151:1164–1169
61. Wu J, Chang J, Cao Q, Liang C (2019) A trust propagation and collaborative filtering based method for incomplete information in social network group decision making with type-2 linguistic trust. *Comput Ind Eng* 127:853–864
62. Subramaniaswamy V, Manogaran G, Logesh R, Vijayakumar V, Chilamkurti N, Malathi D, Senthilselvan N (2019) An ontology-

- driven personalized food recommendation in IoT-based health-care system. *J Supercomput* 75(6):3184–3216
63. Ren C, Chen J, Kuo Y, Wu D, Yang M (2018) Recommender system for mobile users. *Multimed Tools Appl* 77(4):4133–4153
64. Huang Z, Xu X, Ni J, Zhu H, Wang C (2019) Multimodal representation learning for recommendation in Internet of Things. *IEEE Internet of Things J* 6(6):10675–10685

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