Improving Collaborative Filtering's Rating Prediction Accuracy by Introducing the Common Item Rating Past Criterion

Dionisis Margaris
Department of Informatics and Telecommunications
University of Athens
Athens, Greece
margaris@di.uoa.gr

Dionysios Vasilopoulos, Costas Vassilakis and Dimitris Spiliotopoulos
Department of Informatics and Telecommunications
University of the Peloponnese
Tripoli, Greece
dvasilop@uop.gr, costas@uop.gr, dspiliot@uop.gr

***Abstract*—Collaborative filtering formulates personalized recommendations by considering ratings submitted by users. Collaborative filtering algorithms initially find people having similar likings, by inspecting the similarity of ratings already present in the ratings database. Users exhibiting high similarity regarding their likings are classified as "near neighbors" (NNs) and the ratings entered by each user's near neighbors drive the formulation of recommendations for that user. To quantify the similarity between users, in order to determine a user's NNs, a similarity metric is used. Insofar, similarity metrics proposed in the literature either consider all user ratings equally or take into account temporal variations within the users' or items' ratings history. However users' ratings are co-shaped according to the experiences that they had in the past; therefore if two users enter similar (or dissimilar) ratings for an item while having experienced –to a large extent- the same items in the past, this constitutes stronger evidence about user similarity (or dissimilarity). Insofar however, no similarity metric takes into account this aspect. In this work, we propose and evaluate an algorithm that considers the common item rating past when computing rating predictions, in order to increase rating prediction accuracy.**

Keywords— Collaborative Filtering, Items' Rating Sequence, Common Item Rating Past Criterion, Pearson Correlation Coefficient, Cosine Similarity, Evaluation.

I. Introduction

Collaborative filtering (CF) computes personalized recommendations, by considering users' tastes and likings which have been expressed as item ratings stored in a ratings database. CF algorithms are commonly used for building recommender systems (RSs), since they have been proven to be the very successful in this context [1]. CF algorithms initially find people having similar tastes, by examining the likeness of ratings that have already been submitted; for each user *u*, users found to bear a high degree of likeness with *u* regarding their tastes are labeled as *u*'s nearest neighbors (NNs). Afterwards, when a prediction is made regarding the rating that *u* would enter for an item *i* that *u* has not reviewed yet, the ratings submitted by *u*'s NNs for item *i* are are extracted and combined [1], under the rationale that users who have demonstrated to have similar likings in the past, are bound to exhibit the same similarity patterns in the future too [2,3].

Similarity between users is quantified using a correlation coefficient. A correlation coefficient maps pairs of entities (users or items) to a similarity metric, typically falling in the range of [0, 1] or in the range [-1, 1]; in both cases, the highest value in the range denotes highly similar entities, while the lowest value denotes entirely dissimilar ones.

In the context of CF, the most widely used in CF are the Pearson Correlation Coefficient (PCC) and the Cosine Similarity (CS). The PCC subtracts from each rating of a user *u* the average of all ratings submitted by *u*, in order to address the issue that some users are stricter/more lenient than others; the Cosine Similarity (CS) does not adopt this practice [1,2,4].

However, a user's rating behavior is co-shaped by the items he has interacted with and rated (e.g. a movie viewer's rating behavior is defined by the movies that he has watched) at any specific time and therefore the future ratings are effectively biased by the formerly watched content; however this information is not taken into account by either of these metrics, while this also holds for all similarity metrics that have been proposed in the literature.

In this work, we introduce the *Common Item Rating Past Criterion (CIRPaC)*, which focuses on the sequence that each user's ratings have been entered in the database. More specifically, in the process of formulating a prediction for the rating that user *u* would assign to item *i*, for each of *u*'s NNs $NN_{u,k}$ the algorithm checks the degree of similarity between (a) the set of items that *u* has rated and (b) the set of items that $NN_{u,k}$ has rated *up to the point that* $NN_{u,k}$ had entered his rating for item *i*. The higher the similarity of these two sets, the more the similarity between the two aforementioned users will be "rewarded" (increased), under the rationale that their ratings for item *i* (factual for $NN_{u,k}$, predicted for *u*) will have been influenced by the same set of experiences.

To illustrate the concept of the *CIRPaC criterion*, let us consider the case where we want to predict the rating that user u_1 would give to the item i_1, which corresponds to the historic period drama series "Tudors" (http://www.imdb.com/title/tt0758790/), in order to decide whether it should be recommended to him or not, and only two users, namely u_2 and u_3, have watched and rated this series (in order to be used as NNs). All of our three users, have watched and rated the also historic

978-1-7281-4959-2/19/$31.00 ©2019 IEEE

period drama series "Game of Thrones" (http://www.imdb.com/title/tt0944947/), however u_2 has watched and rated "Tudors" *before* "Game of Thrones", while u_3 has followed the inverse order, i.e. he has watched and rated "Tudors" *after* "Game of Thrones" (resembling the case of our active user, u_1, who is asking a recommendation for it). Taking into account that the "Game of Thrones" series is considered by many people as the best historic period drama series of all times, achieving an IMDB score of 9.5/10, we expect that a user that has already seen "Game of Thrones" will rate "Tudors" by different standards than another user that has not. This example indicates that the rating that a user sets to an item is not rating history-independent; on the contrary it clearly depends on his experience on the items that the user has interacted with and rated up to that time. Therefore, in this paper we (1) introduce the concept of *CIRPaC*, which aims precisely at quantifying and exploiting the degree of similarity between the sets of items that two users had experienced up to the time of rating registration and prediction and (2) investigate how we can incorporate the *CIRPaC* into the rating prediction computation process, so as to leverage the prediction accuracy of CF recommender systems.

To validate our approach, we present an extensive comparative evaluation among:

1. the proposed algorithm,
2. the rating abstention interval-based algorithm presented in [26], which (i) is a state-of-the-art algorithm exploiting temporal, within user history information, to achieve prediction error reduction in the context of CF-based rating predictions and (ii) has been shown to surpass the performance of other state-of-the-art algorithms. It is noted here that the algorithm presented in [26] necessitates the existence of data regarding the interaction among users within social networks and also exhibits a drop regarding coverage (i.e. loses some potential to compute personalized recommendations for users),
3. the dynamic average-based algorithm presented in [5], which (i) is a state-of-the-art algorithm targeting improvement of prediction accuracy in the context of CF, (ii) does not need extra information, regarding users or items (e.g. item categories or user social relationships) and (iii) does not deteriorate the prediction coverage,
4. the plain CF algorithm,

employing both the PCC and CS metrics. To measure rating prediction accuracy, we employ two accuracy metrics, namely the Mean Absolute Error (MAE) and the Root-Mean-Square Error (RMSE). The MAE computes the absolute error between each predicted and actual rating and then calculates the mean, while the RMSE squares the distance between each predicted and actual rating before summing them up, penalizing thus larger errors more severely [6].

The proposed approach can be fused with other algorithms in the domain of CF which aim to improve rating prediction accuracy, efficiency and recommendation quality, including application of clustering [7,8], use of information sourced from social network (SN) [9-11] or pruning of ratings databases [12-14].

The rest of the paper is structured as follows: section 2 overviews related work, while section 3 presents the proposed algorithm. Section 4 presents the algorithm tuning and evaluation procedure and the results obtained and, finally, section 5 concludes the paper and outlines future work.

II. Related Work

Many researchers have insofar proposed algorithms aiming to increase the accuracy of CF-based systems. These works exploit characteristics of the ratings databases or information sourced from external linked data sources [3,15,16]. Whitby et al. [17] postulate that ratings contributed by different raters on a specific agent will roughly follow the same probability distribution and, under this assumption, they propose a filtering technique to tackle both unfairly positive and unfairly negative ratings in a Bayesian reputation system.

A new neighbourhood-based model is proposed by Koren [18]. This model uses a global cost function which is formally optimized, leading to prediction accuracy improvement, while in parallel the advantages of the neighbourhood approach (handling of new users/ratings without the need to retrain the model; prediction explainability; etc.) are maintained. Additionally, in [18] Koren proposes a factorization-based adaptation of the neighbourhood model, which achieves the same levels of prediction accuracy with reduced computational complexity. A new model for quantifying user similarity is proposed by Liu et al. [19], aiming to leverage recommendation quality when user similarity calculation is based only on few ratings. This model takes into account both the global user preferences derived from his/her behaviour and the local context information of each user rating.

User interests and likings may change with the passage of time [20,21] as consequence of the concept drift phenomenon [20]; to this end, timestamps available in rating databases can be exploited to detect concept drifts and adjust predictions accordingly. Koenigstein et al. [22] compute, extract and exploit different temporal dynamics of music ratings, which are combined with taxonomical information of music-related items, creating a rich bias model that offers improved accuracy. Minku et al. [23] use different criteria to partition drifts into non-heterogeneous categories. They further demonstrate that when ensembles exhibit lower diversity, prediction accuracy increases. Under the presence of concept drift, old-aged ratings may not reflect current likings of users; taking this into account, pruning techniques have been proposed [7,8], which remove from the database the old-aged ratings, increasing thus rating prediction accuracy, reducing however the coverage.

Improvement of rating prediction accuracy has been also pursued by Knowledge-based RSs. Margaris et al [24] present a knowledge-based leisure time recommendation algorithm for users of social media. This algorithm considers the user's profile and habits, qualitative attributes of venues such as price and service level, the physical distance between venue locations and ratings entered by the user's influencers. AKNOBAS [25] is a Knowledge-based Segmentation Recommender System, which applies Intelligent Clustering Techniques for Information Systems in order to follow trends, which are then used in the recommendation formulation process.

As SNs are increasingly used, specialized algorithms for recommendation within SNs have been proposed. Konstas et al. [27] examine how SN relationships can be exploited within a

CF-based track recommendation system. Various aspects of the social graph established among users, items and tags, such as user friendships and social annotation are considered in this work. Arazy et al. [28] present a conceptual RS blueprint, encompassing the dimensions of tie strength between users, trust propagation and source's reputation; each of these dimensions is affected by the structure and dynamics of a SN and is considered in the recommendation generation process, which is executed by the system's prediction component. Quijano-Sanchez et al. [29] consider the dimensions of trust between users, users' interaction and aspects of each user's personality, which are integrated into the recommendation algorithm of a content-based RS. Margaris et al. [10] refine the concept of influencer, calculating distinct sets of influencers per interest category, and demonstrate that this approach leads to more useful recommendations. Improvement of venue recommendation accuracy and quality in the context of SNs is pursued in [30] through the consideration of data sourced from web services [37,38] provided by the Internet of Things.

Recently, the variability of user ratings was included in the rating prediction computation process, through which rating prediction quality is improved [5]. Additionally, the work in [26] introduces the concept of rating abstention intervals, i.e. periods of rating inactivity on behalf of the users, which indicate a shift of interest. The computation of rating abstention intervals entails the exploitation of temporal, within-user history information. The algorithm presented in [26] additionally computes and exploits metrics regarding influence levels among users, on the basis of data regarding interaction between users in social networks. Combining these two features, the work in [26] achieves considerable rating prediction accuracy improvements, surpassing other state-of-the-art algorithms. In the same line, [42] detects and exploits shifts in user rating practices to increase prediction accuracy.

However, none of the works mentioned above considers the aspect of shared experiences prior to the rating of each item. The present paper fills this gap by presenting an algorithm that leverages the similarity score of the users who have this content in common and evaluates its performance using different user similarity metrics and datasets.

III. The Proposed Algorithm

In CF-based systems, predictions for ratings that a user *U* would give to items are based on the ratings already entered by the "near neighbors of *U*" (NNs), i.e. a set of users who have rated items similarly with *U*. The most widely used metric to quantify similarity between users is the Pearson correlation coefficient [5], where the similarity between two users *U* and *V* is computed as shown in equation (1):

$$\mathrm{simP}(U,\, V)=\frac{\Sigma_k\,(r_{U,k}-\overline{r_U})*(r_{V,k}-\overline{r_V})}{\sqrt{\Sigma_k\big(r_{U,k}-\overline{r_u}\big)^2*\Sigma_k\big(r_{V,k}-\overline{r_V}\big)^2}} \tag{1}$$

where the domain of *k* is the set of items having been rated by both *U* and *V*, while $\overline{r_U}$ and $\overline{r_V}$ are the mean value or ratings entered by users *U* and *V*, respectively. Subsequently, the users having the highest similarity values with *U* are designated as *U's* NNs and denoted as NN_U. Only users *W* for which *simP(U, W)>0* are included in NN_U. Analogously, the Cosine Similarity metric [5] is computed as shown in equation (2):

$$\mathrm{simC}(U,V)=\frac{\Sigma_k(r_{U,k}*r_{V,k})}{\sqrt{\Sigma_k\big(r_{U,k}\big)^2*\Sigma_k\big(r_{V,k}\big)^2}} \tag{2}$$

As can be seen in formulas (1) and (2), the PCC subtracts from each rating $r_{U,k}$ entered by user *U* by the average value of all *U*'s ratings, in order to address the issue that some users may be stricter/more lenient than others when entering ratings; in contrast, the CS metric does not include such a provision.

Afterwards, the prediction $p_{U,i}$ for the rating that user *U* would give on item *i* is computed as shown in formula (3):

$$p_{U,i}=\overline{r_u}+\frac{\Sigma_{V\in NN_u}\mathrm{sim}(U,V)*(r_{V,i}-\overline{r_V})}{\Sigma_{V\in NN_u}\mathrm{sim}(U,V)} \tag{3}$$

The proposed algorithm modifies the prediction computation formula, by including a factor that considers the items' rating sequence within the users' rating sets. More specifically, formula (3) is adapted as follows:

$$p_{U,i}=\overline{r_u}+\frac{\Sigma_{V\in NN_u}\mathrm{sim}(U,V)*\mathrm{CIRPaC\_bonus}(V,U,i)*(r_{V,i}-\overline{r_V})}{\Sigma_{V\in NN_u}\mathrm{sim}(U,V)*\mathrm{CIRPaC\_bonus}(V)} \tag{4}$$

where the *CIRPaC_bonus(V,U,i)* parameter is the weight-bonus assigned to the each NN user *V* of user *U*, depending on the similarity of the between (a) the set of items that *U* has rated and (b) the set of items that *V* has rated *up to the point that V* had entered $r_{V,i}$ in the ratings database.

In more detail, the computation of the *CIRPaC_bonus(V,U,i)* is performed as follows: let *hist(U)* and *hist(V)* be, respectively, the sets of items that users *U* and *V* have rated, and *comRat(U, V)= hist(U)* $\cap$ *hist(V)* = {x_1, x_2, ..., x_l} be the set of items that have been commonly rated by these users. *comRat(U, V)* can be partitioned into two disjoint subsets with respect to item *i* for which the rating prediction is computed: the first subset *pre(V, U, i)* contains all items that had been rated by user *V* before he rated item *i*, while correspondingly *post(V, U, i)* contains all items that had been rated by user *V* after he rated item *i*. Formally, sets *pre(V, U, i)* and *post(V, U, i)* are defined as follows:

$$\begin{aligned}\mathrm{pre}(V,U,i)&=\big\{x\in\mathrm{comRat}(U,V)\colon\ \mathrm{tstamp}(r_{v,x})<\mathrm{tstamp}(r_{v,i})\big\}\\ \mathrm{post}(V,U,i)&=\big\{x\in\mathrm{comRat}(U,V)\colon\ \mathrm{tstamp}(r_{v,x})>\mathrm{tstamp}(r_{v,i})\big\}\end{aligned} \tag{5}$$

Effectively, *pre(V, U, i)* corresponds to the shared experiences between users *U* and *V* that user *V* had perceived before his (factual) rating for item *i*, while *comRat(U, V)* reflects the shared experiences between users *U* and *V* that user *U* has perceived before his (predicted) rating of item *i*. We can measure the similarity of these sets using the Jaccard index [35], where the similarity of two sets *A* and *B* is calculated as follows:

$$J(A,\, B)=\frac{|A\cap B|}{|A\cup B|} \tag{6}$$

Considering that *comRat(U, V)* $\supseteq$ *pre(V, U, i)*, equation (6) can be rewritten as:

$$\mathrm{histSim}(V,U,i)=J(\mathrm{comRat}(U,V),\ \mathrm{pre}(V,U,i))=\frac{|\ \mathrm{pre}(V,U,i)|}{|\mathrm{comRat}(U,V)|} \tag{7}$$

We have also experimented with other similarity measures, such as the Sorensen Similarity Index [11], and the results were in close agreement with those obtained while using the Jaccard index (differences were capped by 1.6% in all cases); hence, we only report on the results obtained while using the Jaccard index.

Finally, *CIRPaC_bonus(V,U,i)* is computed by multiplying the *histSim(V, U, i)* metric by a constant termed *CIRPaC_base*; the optimal value for this constant is determined experimentally, and the relevant experiments are reported in section IV. Since however *CIRPaC_bonus(V,U,i)* is designed to be a weight amplification parameter for cases when users *V* and *U* have considerable common experiences before the rating of item *i*, the *CIRPaC_bonus(V,U,i)* is bounded from below by the value of 1.0. Thus, formally, *CIRPaC_bonus(V,U,i)* is computed as follows:

$$\text{CIRPaC\_bonus(V,U,i)=max(histSim(V,U,i)*CIRPaC\_base,1)} \quad (8)$$

In the next section, we investigate candidate *CIRPaC_base* values, aiming to identify the optimal setting for this parameter and assess the performance of the proposed algorithm.

IV. Algorithm Tuning and Performance Evaluation

In this section, we report on the experiments that were designed to:

1. determine the optimal *CIRPaC_base* parameter value, in order to tune the algorithm and
2. compute the prediction improvement achieved due to the consideration of the *CIRPaC criterion*.

In order to determine the optimal parameter value, we experimentally explored the parameter value solution space, by iteratively selecting parameter value assignments and examining the effect that each parameter value assignment has on rating prediction quality. Rating prediction quality was quantified using two widely used error metrics, namely the Mean Absolute Error (MAE), and the Root Mean Squared Error (RMSE). The use of two different metrics allowed us to gain more extensive insight on the accuracy achieved by each parameter setting, since the MAE metric handles all error scales in a uniform fashion, whereas the RMSE metric penalizes more severely larger errors.

To compute the algorithm's prediction error, in terms of MAE and RMSE, we exercised the standard "hide one" technique [2,5,6]: each user's last rating in the database was hidden and then its value was predicted on the basis of the values of other, non-hidden ratings. We also performed a second experiment where, for each user u, a random rating $r_{u,x}$ was selected and hidden, while in parallel all ratings of user u that had been entered after $r_{u,x}$ were dropped; subsequently, the hidden rating was again predicted its value on the basis of the values of other, non-hidden ratings. The results obtained from the two experiments were in close agreement (the differences observed were less than 2% in all cases), therefore we only present the results of the first experiment. All our experiments were run on seven datasets. Five of these datasets are sourced from Amazon [31,32], while the remaining two from MovieLens [33,34]. The Amazon datasets are relatively sparse (a dataset D is considered very sparse if $\text{density(D)}=\frac{\text{\#ratings}}{\text{\#users*\#items}} \ll 1$ [4]), while the MovieLens datasets are relatively dense. We tested both dense and sparse datasets to affirm that the proposed algorithm can be used in sparse and dense datasets alike.

The seven datasets used in our experiments are summarized in Table I and have the following features:

- they are up to date (published between 1996 and 2016),
- they are widely used for benchmarking in CF research,
- they contain each rating's timestamp, which is necessary for the operation of the proposed algorithm and
- they differ in regards to the type of item domain of the dataset (videogames, movies, music and books) and size (ranging from 2 MB to 486 MB in plain text format).

Each dataset was initially preprocessed, and users found to have less than 10 ratings were dropped, since predictions formulated for users with few ratings are known to demonstrate high error levels [2,3]. This procedure did not have any effect on the MovieLens dataset, since it includes only users that have submitted at least 20 ratings. It is worth noting that we repeated the same experiments with datasets where users having between 5 and 10 ratings were retained, in order to gain insight on the proposed algorithm's behavior under contexts more akin to a "cold start" situation. In these contexts, the absolute average prediction error expectedly increased, however the error reduction levels were found to be in close agreement with those reported in the following subsections (±0.5% of the gains reported for the respective cases where users had at least 10 ratings each). These findings indicate that the proposed algorithm can be useful in cold start contexts, however further investigation on this aspect is needed; in our future work we will elaborate on this aspect.

Our experiments were executed on a computer equipped with six Intel Xeon E7 - 4830 @ 2.13 GHz processors with 8 cores each, 256 GB of shared RAM and one 900 GB SATA HDD with a transfer rate of 200 MBps. This computer both hosted the datasets and ran the rating prediction algorithms.

In the remainder of this section, we present and discuss the results obtained from applying the algorithm presented above on these seven datasets.

A. The Amazon "Videogames" dataset

Fig. 1 illustrates the effect of the value of the *CIRPaC_base* parameter on the accuracy of rating predictions computed by the

TABLE I. Datasets Summary

Dataset name	#Users	#Items	#Ratings	Avg. #Ratings / User	Density	DB size (in text format)
Amazon "Videogames" [31,32]	8.1K	50K	157K	19.4	0.039%	4 MB
Amazon "CDs and Vinyl" [31,32]	41.2K	486K	1.3M	31.6	0.006%	32 MB
Amazon "Movies and TV" [31,32]	46.4K	134K	1.3M	28.0	0.021%	31 MB
Amazon "Books" [31,32]	295K	2.33M	8.7M	29.5	0.001%	227 MB
Amazon "Digital Music" [31,32]	6.2K	35K	86K	13.9	0.040%	2 MB
MovieLens "Latest 100K – Recommended for education and development" [33,34]	700	9K	100K	142.8	1.587%	2 MB
MovieLens "Latest 20M – Recommended for new research" dataset [33,34]	138K	27K	20M	144.9	0.537%	486 MB

proposed algorithm, under both similarity metrics; rating accuracy is measured by both the MAE and the RMSE error metrics, while the plain CF algorithm's performance is used as a yardstick. Considering the accuracy of the rating predictions, the best performance for both similarity metrics is achieved when the *CIRPaC_base* parameter is set to 130% (giving thus a maximum of 30% bonus). Under this setting, when using the PCC the MAE drops by 4.35% and the RMSE by 4.6%. As far as the CS metric is concerned, the respective reductions are 3.99% and 3.64%.

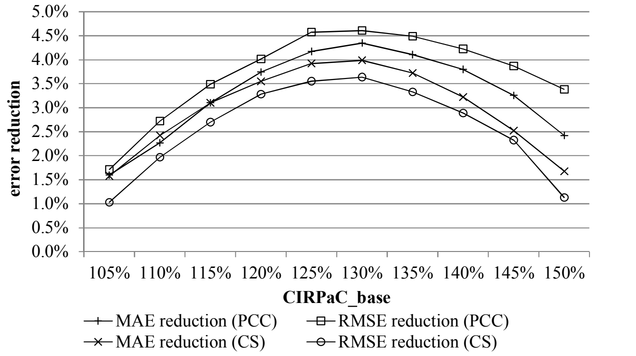

Fig. 1. MAE and RMSE reduction achieved by the proposed algorithm, under different *CIRPaC_base* values and under both similarity metrics for the Amazon "Videogames" dataset.

B. The Amazon "CDs and Vinyl" dataset

Fig. 2 depicts the effect of the *CIRPaC_base* parameter value on the accuracy of rating predictions produced by the proposed algorithm, under both similarity metrics; rating accuracy is measured by both the MAE and the RMSE error metrics. Again, the plain CF algorithm's performance is used as a baseline. Regarding rating prediction quality, the best performance for both similarity metrics is achieved when the *CIRPaC_base* parameter is set to 125%; under this setting, when using the PCC the MAE drops by 5.54% and the RMSE by 4.84%. The respective reductions are for the CS metric are 4.16% and 3.76%.

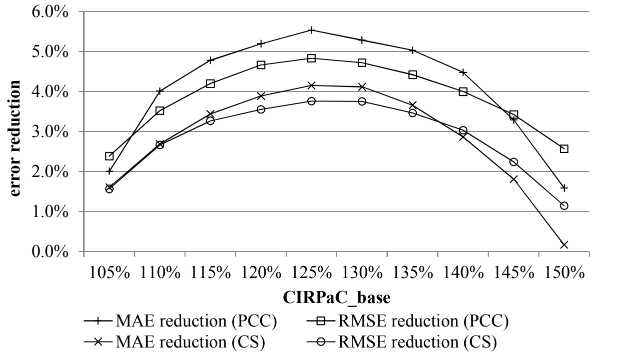

Fig. 2. MAE and RMSE reduction achieved by the proposed algorithm, under different *CIRPaC_base* values and under both similarity metrics for the Amazon "CDs and Vinyl" dataset.

C. The Amazon "Movies and TV" dataset

Fig. 3 displays the effect of the value of the *CIRPaC_base* parameter on the accuracy of rating predictions computed by the proposed algorithm, for both similarity metrics; rating accuracy is measured by both the MAE and the RMSE error metrics, while the plain CF algorithm's performance is used as a baseline. The highest rating prediction accuracy improvement for both metrics is achieved when the *CIRPaC_base* parameter is set to 125%. Under this configuration, when using the PCC the MAE drops by 2.38% and the RMSE by 2.23%. As far as the CS metric is concerned, the respective reductions are 2% and 2.04%.

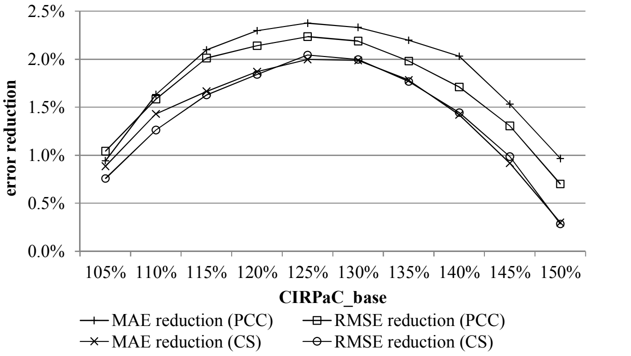

Fig. 3. MAE and RMSE reduction achieved by the proposed algorithm, under different *CIRPaC_base* values and under both similarity metrics for the Amazon "Movies and TV" dataset.

D. The Amazon "Books" dataset

Fig. 4 illustrates the effect of the *CIRPaC_base* parameter value on the accuracy of rating predictions formulated by the proposed algorithm, for both similarity metrics; rating accuracy is measured by both the MAE and the RMSE error metrics, while the plain CF algorithm's performance is used as a baseline. The highest rating prediction accuracy improvement for both similarity metrics is achieved when the *CIRPaC_base* parameter is set 125%; under this setting, when using the PCC the MAE drops by 3.88% and the RMSE by 3.01%. Regarding the CS metric, the respective reductions are 2.88% and 2.0%.

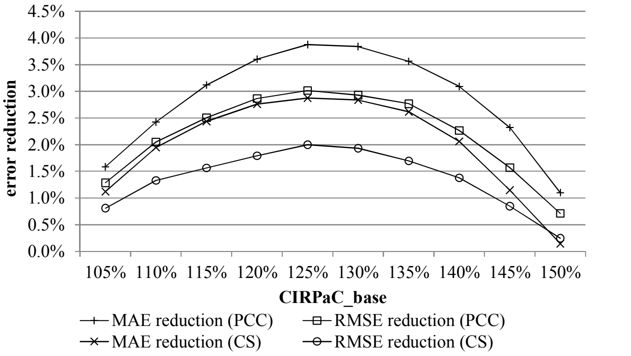

Fig. 4. MAE and RMSE reduction achieved by the proposed algorithm, under different *CIRPaC_base* values and under both similarity metrics for the Amazon "Books" dataset.

E. The Amazon "Digital Music" dataset

Fig. 5 depicts the effect of the *CIRPaC_base* parameter value on the accuracy of rating predictions produced by the proposed algorithm, for both similarity metrics, as this is manifested by the MAE and the RMSE error metrics. The performance of the plain CF algorithm is used as a yardstick.

Regarding rating prediction accuracy, the best performance for both similarity metrics is achieved when the *CIRPaC_base* parameter is set to 125%; when this setting is applied, under the

PCC the MAE drops by 4.3% and the RMSE by 4.44%; under the CS metric, the respective reductions are 3.59% and 3.78 %.

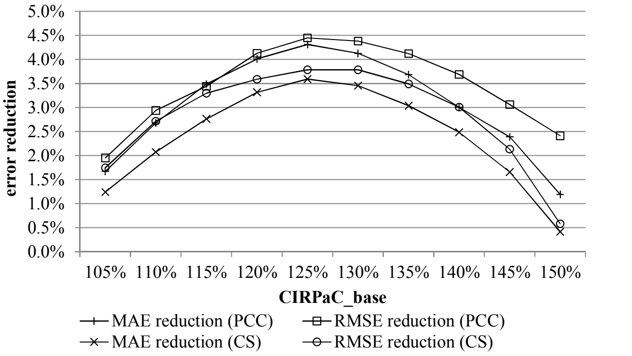

Fig. 5. MAE and RMSE reduction achieved by the proposed algorithm, under different *CIRPaC_base* values and under both similarity metrics for the Amazon "Digital Music" dataset.

F. The MovieLens "Latest 100K – Recommended for education and development" dataset

Fig. 6 shows the effect of the value of the *CIRPaC_base* parameter on the accuracy of rating predictions computed by the proposed algorithm, for both similarity metrics; rating accuracy is measured by both the MAE and the RMSE error metrics, while the plain CF algorithm's performance is used as a baseline.

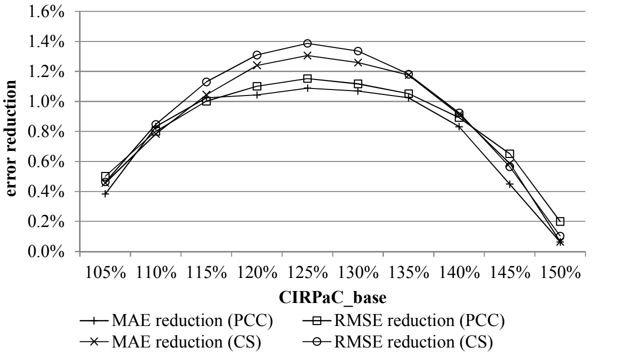

Fig. 6. MAE and RMSE reduction achieved by the proposed algorithm, under different *CIRPaC_base* values and under both similarity metrics for the MovieLens "Latest 100K – Recom. for education and development" dataset.

Considering the accuracy of the rating predictions, the best performance for both similarity metrics is achieved when the *CIRPaC_base* parameter is set to 125%. Under this configuration, when using the PCC the MAE drops by 1.09% and the RMSE by 1.15%. The respective reductions for the CS metric are 1.31% and 1.39%.

G. The MovieLens "Latest 20M – Recommended for new research" dataset

Fig. 7 demonstrates the effect of the value of the *CIRPaC_base* parameter on the accuracy of rating predictions computed by the proposed algorithm, for both similarity metrics, as this is expressed by the MAE and the RMSE metrics. The performance of the plain CF algorithm is used as a baseline.

Regarding the quality of the rating predictions, the best performance for both similarity metrics is achieved when the *CIRPaC_base* parameter is set 125%; under this setting, when using the PCC the MAE drops by 0.48% and the RMSE by 0.44%. The corresponding reduction under the CS similarity metric are 0.46% and 0.39%.

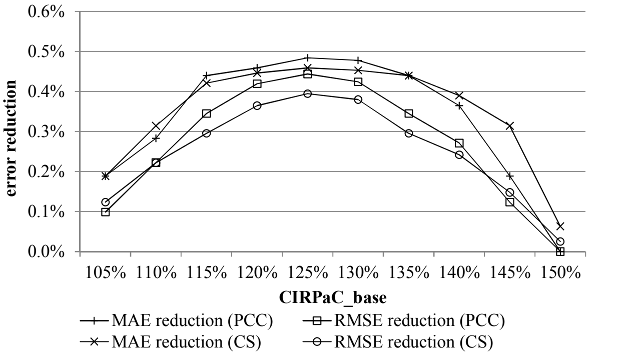

Fig. 7. MAE and RMSE reduction achieved by the proposed algorithm, under different *CIRPaC_base* values and under both similarity metrics for the MovieLens "Latest 20M – Recommended for new research" dataset.

H. Results overview and comparison with previous work

In this section, we overview the results presented in the previous paragraphs and we compare these results with the ones produced by the CF variability algorithm, proposed in [5]. This algorithm was chosen for the comparison since it (i) is a state-of-the-art algorithm targeting improvement of prediction accuracy in the context of CF, (ii) does not need extra information, regarding the users or the items (e.g. item categories or user social relationships) and (iii) does not deteriorate the prediction coverage.

Considering the optimal value of the *CIRPaC_base* parameter, based on the results presented in the previous subsections, we can clearly see that it lays around 125%-130%. More specifically the setting 125% proved to be the optimal one in 6 out of the 7 datasets tested, while in the remaining one (the Amazon "Videogames" dataset), the 130% setting proved to be the best. On a global average, the setting of 125% has a performance edge over the setting of 130%, ranging from 1.5% (MAE reduction under the CS similarity metric) to 2.3% (RMSE reduction under the PCC metric); hence in the next experiments the *CIRPaC_base* parameter will be set to 125%.

From the result analysis in subsections IV.A to IV.G, we can observe that the proposed algorithm achieves higher reductions in sparse datasets rather than in dense ones. An initial analysis revealed that in the two dense datasets considered, there is a smaller probability that experience sequences among users coincide; however deeper analysis on this aspect is required; this aspect will be investigated in the context of our future work.

Fig. 8 depicts the improvement in the MAE achieved by the proposed algorithm, when compared to the CF variability algorithm, proposed in [5], taking the performance of the plain CF algorithm as a baseline and using the PCC as the similarity metric, since this is the one tested in [5]. Clearly, the proposed algorithm achieves the best results, in all the datasets tested, with its MAE reduction being 39.4% higher than that achieved by the CF variability algorithm (3.15% against 2.26% in absolute figures). At individual dataset level, the performance edge of the proposed algorithm against the CF variability algorithm ranges from 13% to 128%.

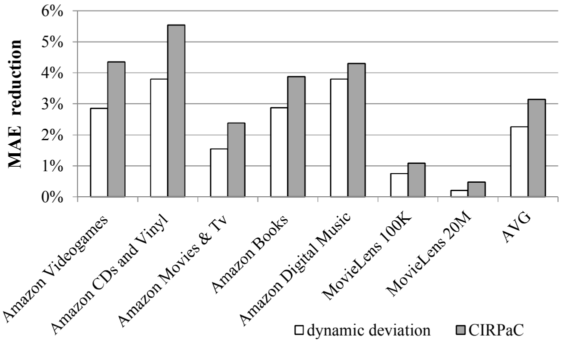

Fig. 8. MAE reduction achieved by the proposed algorithm, in comparison to the Dynamic Deviation Algorithm, proposed in [5].

Fig. 9 depicts the respective improvement in the RMSE achieved by the proposed algorithm, when compared to the CF variability algorithm, proposed in [5], again taking the performance of the plain CF algorithm as a baseline, again using the PCC as the similarity metric. Again, the proposed algorithm clearly achieves the best results, in all the datasets tested, with its MAE reduction being 100% higher than that achieved by the CF variability algorithm (2.96% against 1.48% in absolute figures). At individual dataset level, the performance edge of the proposed algorithm against the CF variability algorithm ranges from 39% to 175%.

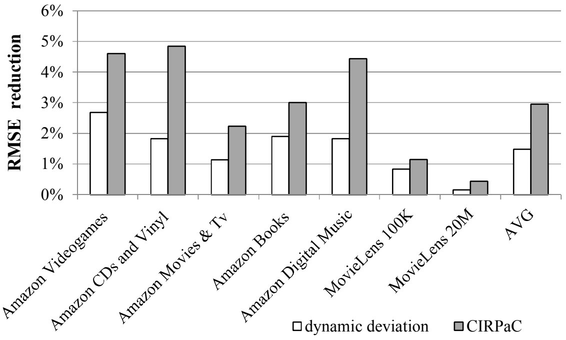

Fig. 9. RMSE reduction achieved by the proposed algorithm, in comparison to the Dynamic Deviation Algorithm, proposed in [5].

Finally, we compare the performance of the proposed algorithm against the algorithm presented in [26], which is a state-of-the-art algorithm exploiting temporal, within user history information, to achieve prediction error reduction in the context of CF-based rating predictions, and has also been shown to surpass the performance of other state-of-the art algorithms. The proposed algorithm achieves an average MAE improvement of 3.15% over all tested datasets, while the respective gains of the algorithm presented in [26] are 2.99%. While the relative difference is limited to 5.4%, it is stressed here that the algorithm presented in [26] requires and exploits additionally information from social networks regarding the influence levels among users, which are not always available. Additionally, the algorithm presented in [26] exhibits a coverage drop which is considerable in the context of sparse datasets; on the other hand, the proposed algorithm fully maintains the coverage levels. It is worth also noting that the algorithm presented in [26] has been shown to surpass the performance of other state-of-the-art algorithms, such as the ones in [36] and [13].

V. Conclusion And Future Work

In this paper we introduced the *Common Item Rating Past Criterion (CIRPaC)*, which considers the effect that items already experienced by a user have on the ratings that he assigns to other items. We have also proposed an algorithm which includes this criterion in the rating prediction computation process, in order to improve prediction accuracy.

The proposed algorithm has been validated through a set of experiments, using two user similarity metrics and seven datasets, both sparse and dense. These experiments showed that the inclusion of items' rating sequence introduces considerable prediction accuracy gains. More specifically, the experiment results have shown that the proposed algorithm delivers a significant MAE reduction, ranging from 0.48% to 5.54%, with an average of 3.15%, and a RMSE reduction, ranging from 0.44% to 4.84% with an average of 2.96%, as far as the PCC metric is concerned. The respective average error reductions, when the CS metric is used, are 2.62% and 2.43% (in all the above percentages, the plain CF algorithm is used as a baseline).

We have also compared the performance of the proposed algorithm against two other state-of-the-art algorithms targeting prediction error reduction, and the proposed algorithm has exhibited superior performance against both of them. More specifically in the comparison against the user rating variability algorithm [5], the proposed algorithm has proved to consistently outperform the user rating variability algorithm across all datasets tested, by margins ranging from 13% to 175%. In the comparison against the algorithm presented in [26], which exploits temporal, within-user history information, the proposed algorithm again proved to achieve better results (by 5.4% on average), even though the algorithm presented in [26] exploits additional information from social networks regarding the influence levels among users, which are not always available.

The algorithm proposed in this paper can be directly integrated in CF-based RSs, since (1) no extra information about the users or the items is required, (2) the additional dataset pre-processing is minimal, being confined to the computation of the items' rating sequence within each user rating set, (3) no additional storage space needs are imposed (4) it can be directly implemented, as a modification of already implemented CF-based systems. Moreover, it can be fused with other algorithms targeting the improvement of rating prediction accuracy or coverage.

In our future work, we will explore additional techniques for improving prediction accuracy in CF. Adaptation of the proposed approach for use with matrix factorization techniques [18] is also considered. Both can be utilized in broader applications of prediction methods [39-41]. Finally, we will explore the potential to combine the proposed technique with other algorithms which aim to improve rating prediction accuracy, recommendation quality or prediction coverage in the CF-based RSs domain.

References

[1] M Balabanovic and Y. Shoham,“Fab: content-based, collaborative recommendation,” Communications of the ACM, vol 40(3), pp. 66-72, 1997.

[2] M. Ekstrand,. R. Riedl and J. Konstan, “Collaborative Filtering Recommender Systems,” Foundations and Trends in Human-Computer Interaction, vol. 4(2), pp. 81-173, 2011.

[3] K. Yu, A. Schwaighofer, V. Tresp, X. Xu and H.P. Kriegel, "Probabilistic Memory-Based Collaborative Filtering," IEEE Transactions on Knowledge Data Engineering, vol. 16(1), 56-69, 2004.

[4] J.B. Schafer, D. Frankowski, J. Herlocker and S. Sen, "Collaborative Filtering Recommender Systems," The Adaptive Web, Lecture Notes in Computer Science, vol. 4321, pp. 291-324, 2007.

[5] D. Margaris and C. Vassilakis, "Improving Collaborative Filtering's Rating Prediction Accuracy by Considering Users' Rating Variability," Proceedings of the 4th IEEE International Conference on Big Data Intelligence and Computing, pp. 1022-1027, 2018.

[6] J. Herlocker, J. Konstan, L. Terveen and J. Riedl, "Evaluating collaborative filtering recommender systems," ACM Transactions in Information Systems, vol. 22(1), pp. 5-53, 2004.

[7] S. Gong, "A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering," Journal of Software, vol. 5(7), pp. 745-752, 2010.

[8] D. Margaris, P. Georgiadis and C. Vassilakis, "A Collaborative Filtering Algorithm with Clustering for Personalized Web Service Selection in Business Processes," Procs. of the 9th IEEE International Conference on Research Challenges in Information Science, pp. 169-180, 2015.

[9] E. Bakshy, D. Eckles, R. Yan and I. Rosenn, "Social Influence in Social Advertising: Evidence from Field Experiments," Proceedings of the 13th ACM Conference on Electronic Commerce, pp. 146-161, 2012.

[10] D. Margaris, C. Vassilakis and P. Georgiadis, "Recommendation information diffusion in social networks considering user influence and semantics," Social Network Analysis and Mining, vol. 6(1), 108, pp. 1-22, 2016.

[11] T.A. Sorensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content, and its application to analyses of the vegetation on Danish commons," K dan Vidensk Selsk Biol Skr 5, pp. 1-34, 1948.

[12] D. Margaris and C. Vassilakis, "Pruning and aging for user histories in collaborative filtering," Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence, pp. 1-8, 2016.

[13] D. Margaris and C. Vassilakis, "Enhancing User Rating Database Consistency through Pruning," Transactions on Large-Scale Data and Knowledge-Centered Systems, vol. XXXIV, pp. 33–64, 2017.

[14] D. Margaris and C. Vassilakis, "Improving Collaborative Filtering's Rating Prediction Quality in Dense Datasets, by Pruning Old Ratings," Proceedings of the 22nd IEEE Symposium on Computers and Communications, pp. 1168-1174, 2017.

[15] R. Dias and M. Fonseca, "Improving Music Recommendation in Session-Based Collaborative Filtering by Using Temporal Context," Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence, pp. 783-788, 2013.

[16] D. Margaris, C. Vassilakis and P. Georgiadis, "Query personalization using social network information and collaborative filtering techniques," Future Generation Computer Systems, vol. 78(1), pp. 440-450, 2018.

[17] A.Whitby, A. Jøsang and J. Indulska, "Filtering Out Unfair Ratings in Bayesian Reputation Systems," Proceedings of the Workshop on Trust in Agent Societies, at the Autonomous Agents and Multi Agent Systems Conference (AAMAS2004), vol. 4, 2004.

[18] Y. Koren, R. Bell and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol. 42(8), pp. 30-37, 2009.

[19] H. Liu, Z. Hu, A. Mian, H. Tian and X. and Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," Knowledge-Based Systems, vol. 56, pp. 156-166, 2014.

[20] J.Gama, I. Zliobaite, A. Bifet, M. Pechenizki and A. Bouchachia, "A Survey on Concept Drift Adaptation," ACM Computing Surveys, vol. 1(1), Article 1, 2013

[21] L. Li, L. Zheng, F. Yang and T. Li, "Modeling and broadening temporal user interest in personalized news recommendation," Expert Systems with Applications, vol. 41 (7), pp. 3168-3177, 2014.

[22] N. Koenigstein, G. Dror and Y. Koren, "Yahoo! Music recommendations: modeling music ratings with temporal dynamics and item taxonomy," Proceedings of the fifth ACM conference on Recommender systems (RecSys '11), pp. 165-172, 2011.

[23] L. Minku, A. White and X Yao, "The Impact of Diversity on Online Ensemble Learning in the Presence of Concept Drift," IEEE Transactions on Knowledge and Data Engineering, vol. 22(5), pp. 730-742, 2010.

[24] D. Margaris, C. Vassilakis and P. Georgiadis, "Knowledge-Based Leisure Time Recommendations in Social Networks," Current Trends on Knowledge-Based Systems: Theory and Applications, pp. 23-48, 2017.

[25] A. Rodríguez, J. Torres, E. Jimenez, M. Gomez and G. Alor-Hernandez, "AKNOBAS: A knowledge-based segmentation recommender system based on intelligent data mining techniques," Computer Science and Information Systems, vol. 9(2), pp. 713-740, 2012.

[26] D. Margaris and C. Vassilakis, "Exploiting Rating Abstention Intervals for Addressing Concept Drift in Social Network Recommender Systems," Informatics, vol. 5(2), 21, 2018.

[27] I. Konstas, V. Stathopoulos and J.M. Jose, "On social networks and collaborative recommendation," Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pp. 195-202, 2009.

[28] O. Arazy, N. Kumar and B. Shapira, "Improving Social Recommender Systems," IT professional, vol. 11(4), 2009.

[29] L. Quijano-Sanchez, J.A. Recio-Garcia and B. Diaz-Agudo, "Group recommendation methods for social network environments," 3rd Workshop on Recommender Systems and the Social Web at the 5th ACM International Conference on Recommender Systems, pp. 1-24, 2011.

[30] D. Margaris and C. Vassilakis, "Exploiting Internet of Things Information to Enhance Venues' Recommendation Accuracy," Service Oriented Computing & Applications, vol. 11(4), pp. 393-409, 2017.

[31] Amazon product data. Available online: http://jmcauley.ucsd.edu/data/amazon/links.html (accessed on August 19, 2019).

[32] J.J. McAuley, C. Targett, Q. Shi and A. Van den Hengel, "Image-Based Recommendations on Styles and Substitutes," Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 43-52, 2015.

[33] MovieLens datasets. Available online: http://grouplens.org/datasets/movielens/ (accessed on August 19, 2019).

[34] F. Harper and J. Konstan, "The MovieLens Datasets: History and Context," ACM Transactions on Interactive Intelligent Systems, vol. 5(4), Article no. 19, 2016.

[35] P.N. Tan, M. Steinbach and V. Kumar, "Introduction to Data Mining," Addison-Wesley, 2005.

[36] Y. Koren, "Collaborative Filtering with Temporal Dynamics," Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 447-456, 2009.

[37] D. Margaris, C. Vassilakis and P. Georgiadis, "Adapting WS-BPEL scenario execution using collaborative filtering techniques", Proceedings of the 7th IEEE International Conference on Research Challenges in Information Science, pp. 174-184, 2013.

[38] D. Margaris, C. Vassilakis and P. Georgiadis, "An integrated framework for QoS-based adaptation and exception resolution in WS-BPEL scenarios", Proceedings of the 28th ACM Symposium on Applied Computing, pp. 1900-1906, 2013.

[39] D. Antonakaki, D. Spiliotopoulos, C.V. Samaras, S. Ioannidis and P. Fragopoulou, "Investigating the Complete Corpus of Referendum and Elections Tweets", Proceedings of the IEEE/ACM Conference on Advances in Social Networks Analysis and Mining, pp. 100-105, 2016.

[40] Schefbeck, G., Spiliotopoulos, D. and T. Risse, The Recent Challenge in Web Archiving: Archiving the Social Web. Proceedings of the International Council on Archives Congress, Brisbane, Australia, 2012.

[41] D. Antonakaki, D. Spiliotopoulos, C.V Samaras, P. Pratikakis, S. Ioannidis and P. Fragopoulou, "Social media analysis during political turbulence". PloS one , Vol. 12, 10 (2017), e0186836

[42] D. Margaris and C. Vassilakis, "Improving Collaborative Filtering's Rating Prediction Quality by Considering Shifts in Rating Practices," Proceedings of the 19th IEEE International Conference on Business Informatics, pp. 158-166, 2017.