

Article

Augmenting Black Sheep Neighbour Importance for Enhancing Rating Prediction Accuracy in Collaborative Filtering

Dionisis Margaris ^{1,*}, Dimitris Spiliotopoulos ² and Costas Vassilakis ³

¹ Department of Digital Systems, University of the Peloponnese, Valiotti's Building, Kladas, 231 00 Sparti, Greece

² Department of Management Science and Technology, University of the Peloponnese, Akadimaikou G. K. Vlachou, 221 31 Tripoli, Greece; dspilot@uop.gr

³ Department of Informatics and Telecommunications, University of the Peloponnese, Akadimaikou G. K. Vlachou, 221 31 Tripoli, Greece; costas@uop.gr

* Correspondence: margaris@uop.gr

Abstract: In this work, an algorithm for enhancing the rating prediction accuracy in collaborative filtering, which does not need any supplementary information, utilising only the users' ratings on items, is presented. This accuracy enhancement is achieved by augmenting the importance of the opinions of 'black sheep near neighbours', which are pairs of near neighbours with opinion agreement on items that deviates from the dominant community opinion on the same item. The presented work substantiates that the weights of near neighbours can be adjusted, based on the degree to which the target user and the near neighbour deviate from the dominant ratings for each item. This concept can be utilized in various other CF algorithms. The experimental evaluation was conducted on six datasets broadly used in CF research, using two user similarity metrics and two rating prediction error metrics. The results show that the proposed technique increases rating prediction accuracy both when used independently and when combined with other CF algorithms. The proposed algorithm is designed to work without the requirements to utilise any supplementary sources of information, such as user relations in social networks and detailed item descriptions. The aforesaid point out both the efficacy and the applicability of the proposed work.

Keywords: collaborative filtering; black sheep users; rating prediction accuracy; evaluation



Citation: Margaris, D.; Spiliotopoulos, D.; Vassilakis, C. Augmenting Black Sheep Neighbour Importance for Enhancing Rating Prediction Accuracy in Collaborative Filtering. *Appl. Sci.* **2021**, *11*, 8369. <https://doi.org/10.3390/app11188369>

Academic Editor:
Ángel González-Prieto

Received: 2 August 2021
Accepted: 7 September 2021
Published: 9 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Collaborative filtering (CF) is a dominant recommender systems technique that consider users' likes and tastes, expressed as item ratings, to create personalised recommendations [1,2]. It ensembles likes from users having similar tastes/preferences, termed as 'near neighbours' to compute rating predictions for items, which then lead to recommendations. Since rating prediction accuracy is directly related to recommendation usefulness and reliability, a major challenge that collaborative filtering systems confront is the enhancement of rating prediction accuracy.

The two main categories of CF algorithms are the memory-based and the model-based ones [3,4]. The algorithms that belong in the first category exploit user rating data to compute the similarity between users (or items). The ones that belong in the second category develop models using various techniques (derived from data mining, machine learning, etc.) [5–7]. Furthermore, there are many hybrid approaches, combining the features of the aforementioned two main categories.

Each of the three approaches has its own advantages: (a) the memory-based approach is characterized by ease of creation and use, high explainability of results, easy incorporation of new data, content-independence of the items being recommended and good scaling with co-rated items; (b) the model-based approach is characterized by matrix sparsity

handling, scalability and high accuracy and (c) the hybrid approach exhibits elevated performance and overcome sparsity and loss of information [3,8–10].

Correspondingly, each of the three approaches exhibit a number of disadvantages: (a) the memory-based approach cannot adequately handle sparsity, does not scale well with the volume of data, and number of entities and falls behind in accuracy; (b) the model-based approach is time-consuming to build and update, cannot be directly applied on a diverse user range, and has low explainability, and (c) the hybrid approach exhibits increased complexity and high implementation cost.

Explainability has been recently recognized as a key aspect in recommender systems. Explainable recommender systems are able to show to their users comprehensible descriptions on why an item is recommended, thus increasing transparency and the likelihood that recommendations are accepted [11–13]. In this area, model-based approaches exhibit considerably lower performance than memory-based approaches, since it is very hard to explain how deep learning networks (which are predominantly used to implement model-based approaches) have amassed their knowledge, on which the recommendations are based [11]. Under this view, an enterprise or organization may opt to employ a memory-based recommender system, aiming to improve the probability that recommendations are accepted while in parallel lowering the recommender system creation and maintenance cost, albeit the accuracy of recommendations may not be optimal.

Moreover, a number of techniques have been developed for memory-based systems, aiming to tackle the deficiencies presented above; these include distributed techniques [14] and optimization methods [15–18] to improve scalability, density enrichment [19] and coverage increase methods [20] to tackle sparsity, and a multitude of methods to improve rating prediction and recommendation quality [21–25]. Considering all the above, memory-based techniques are a viable approach for building contemporary and efficient recommender systems.

Memory-based CF algorithms, as the one presented in this work, typically, include three main steps: (i) find users having similar or at least close tastes, by examining the similarity of already submitted ratings, (ii) predict the rating value that a user would give to an item that he or she has not evaluated yet, and (iii) recommend the items having the highest prediction values to the (target) user [26].

During step (i), for each user U , users computed to have high similarity with U , based on their likings, represented by the ratings they have entered in the rating database (rDB), are labelled as U 's near neighbours (NNs) [27]. In order to identify users' NNs, a similarity function is used, such as the Cosine Similarity and the Pearson Correlation Coefficient, which are the dominant ones in CF research [28,29].

Having found U 's NNs and having computed their numeric similarities with U , during step (ii) a prediction formula is used that computes the numeric rating value that U would give to items that he or she has not evaluated, based on the existed ratings in the rDB [30,31]. The weight/importance of each NN to the prediction value is based only on this similarity with U , calculated during step (i).

Finally, having computed the prediction values of the items that user U has not evaluated yet, during step (iii) the CF system recommends to U the items that obtained the highest prediction values and hence have the highest probability of U actually liking them [32,33].

This work focuses on the second step that a CF system encompasses, i.e., the rating prediction computation; in this step, the proposed approach modifies the weight/importance of each NN to the prediction value, complementing the similarity of U to his/her NNs with a “black sheep factor”. More specifically, for each item, we classify users into two categories of interest, those who seem to like the item and those who do not, based on their ratings on this item. When an item is generally accepted, it is more probable to find pairs of users that have entered high ratings for this item. Conversely, it is less probable to find pairs of users that have entered low ratings for the same item. As a result, the novelty of this work is that, after the average rating value (given by all users) of each item is computed, which

is a simple and straightforward procedure, it modifies the weight/importance of each NN, based on the aforementioned concept, i.e., based on the relative (un-) acceptance of their commonly rated items, in combination with their ratings on these items. The rationale behind the usage of the aforementioned concept derives from the fact that when in real life a human likes a product (e.g., a TV series, a car model, a videogame) that the majority of others do not (and hence this product obtains a relatively low average rating value), and this human finds another one that also likes the exact same product, the probability of valuing his opinion with greater prominence than the opinions of other users, for a future recommendation, is relatively high. This rationale is in line with the use of the inverse document frequency (IDF) metric in information retrieval, where terms occurring less frequently in the document corpus are assigned higher weights [34].

To validate our approach, an extensive evaluation is presented, using (i) two user similarity metrics, (ii) two rating prediction error metrics, and (iii) six datasets that are widely used in CF research.

It is worth mentioning, that the proposed approach (i) does not need any kind of supplementary information, apart from users' ratings on items, and hence can be applied in any CF dataset and (ii) can be fused with other CF approaches, aiming to enhance rating prediction accuracy or efficiency, either using supplementary sources of information, such as users' relations in social networks and detailed characteristics of items [35–37] or not [38–40].

The rest of the paper is structured as follows: in Section 2 the related work is overviewed, while in Section 3 the proposed algorithm is introduced. In Section 4 the methodology for tuning the algorithm's operation is reported, as well as the presented algorithm is evaluated. Finally, Section 5 concludes the paper and outlines future work.

2. Related Work

The accuracy of CF-based recommender systems is a research field that has attracted numerous research works over the last years, which are divided into two main categories. The first category includes research works which exploit supplementary sources of information, such as user relations in social networks and item characteristics, while the second category includes research works that are based solely on the information contained in the user-item rating matrix.

In regard to the first category, [41] examines the impact of incorporating social ties in the prediction formulation, targeting prediction accuracy and presents a social network CF algorithm which tunes the contribution of the social information by using a learning method as a weight parameter in the proposed similarity measure. The work in [42] extracts information from distant social relations and captures opinions from users while modelling their interactions with items, introducing a deep social CF algorithm that exploits social network information for recommendation production. Ref. [43] proposes a method that overcomes the cold start problem and the data sparsity in CF, by designing a Matrix Factorization (MF) Linked Open Data model, which uses a knowledge base to find information concerning the new entities. Ref. [44] states that both group affiliation information and social network information may significantly enhance the accuracy of popularity-based voting recommender systems as well as introduce a set of NN-based and MF-based recommender systems for online social voting. The work in [45] combines tie strength with social network information to create a local random walk-based friend recommendation method. Initially, the basis for friend recommendation is constructed, by using a weighted friend network and then this network is used to compute user similarity by a local random walk-based similarity measure. Ref. [46] firstly introduces a sparsity alleviation approach, based on implicit and explicit satisfaction and uses objective and subjective trust, to establish enhanced trust relationships among users. Then, for each target user, it selects the user's trusted neighbours, which are screened using emotional consistency. Finally, it predicts item ratings to obtain the final recommendations lists. Ref. [36] combines time decay factor for rated items, cognition relationships between users,

and personal cognition behaviour into a unified probabilistic MF model and presents a social MF method for personalised recommendation using social interaction factors.

Although all the previous works achieve relatively high rating prediction accuracy improvement, the supplementary source of information required may not always be available. As a result, an algorithm which can work using only the information located in the user-item-rating matrix may prove to be more appropriate, since it can be applied to every CF dataset.

To this extent, ref. [47] introduces an approach that realises an item-variance weighting in item-based CF. More specifically, it applies a time-related correlation degree to form time-aware similarity computation, which estimates the relationship between two items and reduces the importance of an item that has not recently been rated. Ref. [48] presents a CF optimization method that initially incorporates multiple interests to optimize neighbour selection and then it utilises a ranking strategy that rearranges both the top-N item list and the area the threshold controls, maximising the popularity while maintaining a relatively low prediction accuracy reduction. Ref. [49] clusters items and users by using a Gaussian mixture model and builds a new interaction matrix by extracting new item features that manages to solve the impact of rating data sparsity on CF algorithms. Furthermore, by combining the Jaccard and triangle similarities, it proposes a new similarity calculation algorithm. Ref. [50] presents a slope one algorithm based on user similarity and trusted data fusion that can be applied in various CF systems. For the creation of the final recommendation formula, the proposed algorithm includes the procedures of trusted data selection, user similarity calculation and inclusion of this similarity to the weight factor of the improved slope one algorithm. Ref. [51] presents a local similarity algorithm that can use multiple correlation structures between CF users. Firstly, it uses a clustering method to discover groups of similar items and then, for each cluster, it creates a user-based similarity model, namely Cluster-based Local Similarity. Ref. [52] introduces a CF algorithm that exploits repetitive purchased products and symmetric purchasing order, to tackle user big data. The presented algorithm combines a word2vec mechanism with a gradient boosting machine learning architecture to explore the purchased products based on users' click patterns. In [53], a product recommendation method for CF based on the triangle similarity is presented. The similarity metric considers the ratings of both the non-commonly rated items from pairs of users as well as the common rated ones. It is further complemented with the user rating preference behaviour in giving rating preferences. In [54], a CF rating prediction algorithm is introduced that modulates the rating prediction numeric value, based on the relation between the period the rating to be predicted belongs to, in a certain product category and the users' experienced wait period in the same product category, targeting at enhancing the prediction accuracy of CF systems.

Still, none of the aforementioned research allow for the aspect of users that share a positive or negative opinion about an item, but are outliers when compared to the majority of the users in the dataset. The present work fills this gap by introducing an algorithm that modifies the weight/importance of each NN to the prediction value, based on the relative (un-)acceptance of their commonly rated items, in combination with their ratings on these items, and assessing its performance both used independently and combined with another CF algorithm also aiming at enhancing rating prediction accuracy.

3. The Proposed Algorithm

The procedure that a CF algorithm typically follows, when predicting a rating for user U includes three main steps:

1. Find users having close/similar tastes with U , by examining the similarity of already submitted ratings in the rDB, to identify U 's near neighbour (NN) users; these users will operate as recommenders to U . Typically, in CF systems, the metrics used to quantify user similarity, is the Pearson correlation coefficient (PCC) and the Co-

sine Similarity (CS) [55,56], which are expressed as shown in Equations (1) and (2), respectively:

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

$$sim_CS(U, V) = \frac{\sum_k r_{U,k} \cdot r_{V,k}}{\sqrt{\sum_k (r_{U,k})^2} \cdot \sqrt{\sum_k (r_{V,k})^2}}. \quad (2)$$

Generally, for a user V to be considered as U 's NN, their quantified rating similarity value has to exceed a specific threshold, e.g., the value 0.0 for the PCC metric [55].

2. Predict the rating value that U would give to an item i ; in order to compute the rating prediction $p_{U,i}$, the standard CF rating prediction formula [26,57] is typically applied:

$$p_{U,i} = \bar{r}_u + \frac{\sum_{V \in NN_u} sim(U, V) \cdot (r_{V,i} - \bar{r}_V)}{\sum_{V \in NN_u} sim(U, V)}. \quad (3)$$

The weight/importance of each NN to the prediction value is based only on its numeric similarity with U , calculated during the previous step.

3. Recommend to U the items having the highest prediction values; the number of recommended items is determined by the administrator of the recommender system [58,59].

The proposed algorithm targets at augmenting the importance of each NN V , when V and U (a) mutually agree on their opinion on some items and (b) their opinion on the same items deviates from that of the majority of users; to this end, the proposed algorithm adjusts the rating prediction formula given in the second step above (Equation (3)).

More specifically, the proposed algorithm modifies Equation (3) by considering a *black sheep factor* $bsf(U, V)$ between user U , for whom the rating prediction is computed, and each of his NNs, V , as shown in Equation (4):

$$p_{U,i} = \bar{r}_u + \frac{\sum_{V \in NN_u} sim(U, V) \cdot bsb(U, V) \cdot (r_{V,i} - \bar{r}_V)}{\sum_{V \in NN_u} sim(U, V) \cdot bsb(U, V)}, \quad (4)$$

Effectively, the $bsf(U, V)$ factor is an adjustment assigned to each NN's contribution to the prediction computation, based on the degree to which users U and V mutually agree on the rating of items, while at the same time disagreeing with the majority of other users on the same items.

For the application of this algorithm, the $bsf(U, V)$ factor needs to be determined; the setting of the bsf factor to its optimal value is experimentally explored in the following section, along with the prediction accuracy gains of the proposed algorithm.

4. Algorithm Tuning and Experimental Evaluation

In this section, we report on our experiments aiming to:

1. Determine the optimal value of the bsf factor, to tune the proposed algorithm and;
2. Evaluate the accuracy of the rating prediction of the proposed algorithm, both when used independently and when combined with a state-of-the-art CF algorithm also aiming at rating prediction accuracy improvement.

For the evaluation of the rating prediction quality, both the RMSE and MAE error metrics have been employed. Their quantification was accomplished using the standard "hide one" technique, where a rating from each user in the database is hidden and its value is tried to be predicted [60–62]. In our work, this experiment was executed twice, the first time a random rating was hidden for each user, while in the second experiment, his last rating was hidden (considering the ratings' timestamps in the rDB). These

two experiments produced very close results (less than 1% difference observed), hence, we report on the results from the first experiment, for conciseness. The practice described above is the typical one when evaluating a rating prediction CF algorithm [31,63,64].

Our experiments were executed on six datasets; four of these were obtained from Amazon [65,66], the fifth was sourced from MovieLens [67,68], while the last was sourced from Netflix [69]. Regarding the four Amazon datasets, we used the 5-core ones, where each user and item have at least 5 ratings, to ensure that, unlike in the simple Amazon datasets where for some users and items only one rating exists in the rating database, at least 4 other ratings exist and hence the application of any CF algorithm can produce valid results. The four Amazon datasets are considered relatively sparse (their density is less than 0.1%), while the MovieLens and the Netflix ones are considered relatively dense (their density is greater than 1%). We opted to use both sparse and dense datasets in order to confirm the applicability of the presented algorithm in every CF dataset, regardless of its density. Table 1 is a synopsis of the datasets utilised in this work.

Table 1. Dataset information.

Dataset Name	#Users	#Items	#Ratings	Density
Amazon "Videogames"	24 K	11 K	232 K	0.09%
Amazon "CDs and Vinyl"	75 K	64 K	1.1 M	0.02%
Amazon "Movies and TV"	124 K	50 K	1.7 M	0.03%
Amazon "Books"	604 K	368 K	8.9 M	0.004%
MovieLens "Latest 100K—Recommended for education and development"	670	9 K	100 K	1.7%
NetFlix competition	480 K	18 K	96 M	1.1%

The aforementioned datasets are widely used in CF research [70–73] and they contain each rating's timestamp (essential information for hiding each user's last rating), while at the same time they vary considering their item domain category (music, videogames and TV series, books and movies).

4.1. Determining the Algorithm Parameters

The goal of first experiment is to determine the optimal value for the *bsf* factor used for the rating prediction formula. To do so, we examined more than 40 candidate setting alternatives, however, we report only on the most indicative ones for conciseness. More specifically, Figure 1 illustrates the average prediction accuracy improvement for each alternative under different *bsf* factor settings, pertaining to the MAE and the RMSE scores.

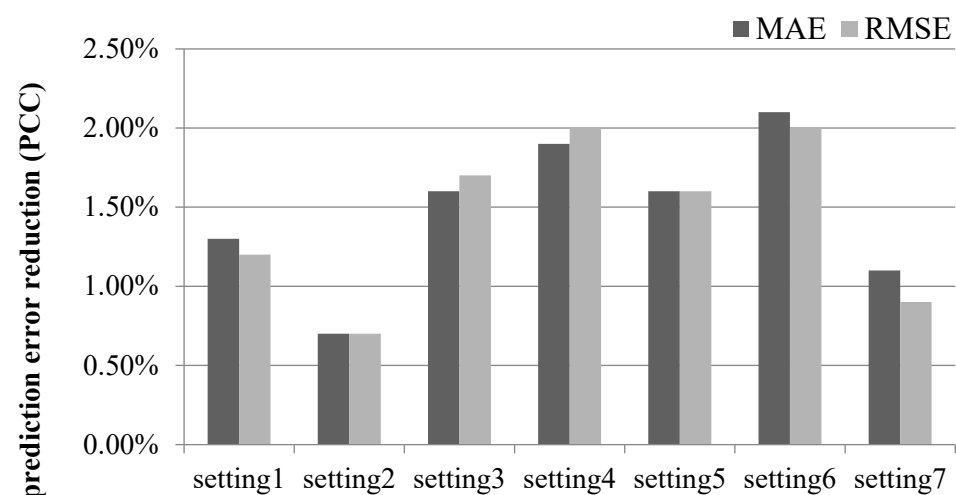


Figure 1. Average prediction error reduction under different *bsf* factor settings.

where each setting_i corresponds to a different setting for the computation of the *bsf* factor as follows:

$$\begin{aligned}
 \text{Setting 1: } b_{sf}(U, V) &= \begin{cases} 1.2, & \text{if } (blackSheepRatings(U, V) \geq 1) \wedge (low_thr = 2.5) \wedge (high_thr = 3.5) \\ 1, & \text{otherwise} \end{cases} \\
 \text{Setting 2: } b_{sb}(U, V) &= \begin{cases} 1.2, & \text{if } (blackSheepRatings(U, V) \geq 1) \wedge (low_thr = 1.5) \wedge (high_thr = 4.5) \\ 1, & \text{otherwise} \end{cases} \\
 \text{Setting 3: } b_{sb}(U, V) &= \begin{cases} 1.2, & \text{if } (blackSheepRatings(U, V) \geq 1) \wedge (low_thr = 2.0) \wedge (high_thr = 4.0) \\ 0.9, & \text{otherwise} \end{cases} \\
 \text{Setting 4: } b_{sb}(U, V) &= \begin{cases} 1.2, & \text{if } (blackSheepRatings(U, V) \geq 1) \wedge (low_thr = 2.5) \wedge (high_thr = 3.5) \\ 0.9, & \text{otherwise} \end{cases} \\
 \text{Setting 5: } b_{sb}(U, V) &= \begin{cases} 1.2, & \text{if } (blackSheepRatings(U, V) \geq 1) \wedge (low_thr = 2.0) \wedge (high_thr = 4.0) \\ 0.8, & \text{otherwise} \end{cases} \\
 \text{Setting 6: } b_{sb}(U, V) &= \begin{cases} 1.2, & \text{if } (blackSheepRatings(U, V) \geq 5\% \cdot numCommonlyRated(U, V)) \\ & \wedge (low_thr = 2.5) \wedge (high_thr = 3.5) \\ 0.9, & \text{otherwise} \end{cases} \\
 \text{Setting 7: } b_{sb}(U, V) &= \begin{cases} 1.2, & \text{if } (blackSheepRatings(U, V) \geq 20\% \cdot numCommonlyRated(U, V)) \\ & \wedge (low_thr = 2.0) \wedge (high_thr = 4.0) \\ 0.8, & \text{otherwise} \end{cases}
 \end{aligned}$$

In the equations presented above, the following notations are used:

- *low_thr* denotes the value below which a rating is considered to be negative; formally, $is_negative(r_{U,i}) \Leftrightarrow r_{U,i} \leq low_thr$
- *high_thr*, correspondingly, represents the value above which a rating is considered to be positive. Formally, $is_positive(r_{U,i}) \Leftrightarrow r_{U,i} \geq high_thr$
- *blackSheepRatings(U, V)* is the number of ratings where users *U* and *V* both have a positive (or negative) rating, while the user community has a negative (or positive), respectively, rating on the same item. Formally:
 - $is_communityPositive(i) \Leftrightarrow \text{average}_{W \in UC}(r_{W,i}) \geq high_thr$, where UC is the user community, i.e., the set of users in the dataset
 - $is_communityNegative(i) \Leftrightarrow \text{average}_{W \in UC}(r_{W,i}) \leq low_thr$
 - $is_BlackSheepRating(U, V, i) \Leftrightarrow (is_Positive(U, i) \wedge is_Positive(V, i) \wedge is_communityNegative(i)) \vee (is_Negative(U, i) \wedge is_Negative(V, i) \wedge is_communityPositive(i))$
 - $blackSheepRatings(U, V) = |\{i \in I : is_BlackSheepRating(U, V, i)\}|$
- *numCommonlyRated(U, V)* is the number of items that have been rated by both *U* and *V*; formally, $numCommonlyRated(U, V) = |\{i \in I : r_{U,i} \neq NULL \wedge r_{V,i} \neq NULL\}|$

In Figure 1 we can notice that the setting where the black sheep factor equals to 1.2 where two users *U* and *V* have at least 5% of their commonly rated items are considered *black sheep ratings*, where the low threshold (that a rating is considered relatively negative) equals to 2.5/5 and the high threshold (that a rating is considered relatively positive) equals to 3.5/5, and 0.9 otherwise, is the optimal one, since it achieves the largest rating prediction gains, for both error quantification metrics.

The corresponding experiment, using the CS user similarity metric produced similar results, where the optimal setting achieved a rating prediction error reduction of 2% for both the MAE and RMSE metrics.

4.2. Rating Prediction Accuracy Improvement Achieved by the Proposed Algorithm

After the proposed algorithm's optimal setting for the *bsf* factor has been experimentally determined, we present our findings regarding the performance gains in terms of rating prediction accuracy, stemming from the application of the proposed algorithm on the four datasets used in our evaluation (c.f. Table 1). Figure 2 presents the accuracy gains that

the proposed algorithm achieves, in terms of the MAE and RMSE metrics, when using the PCC similarity metric and taking the performance of the plain CF algorithm as a yardstick.

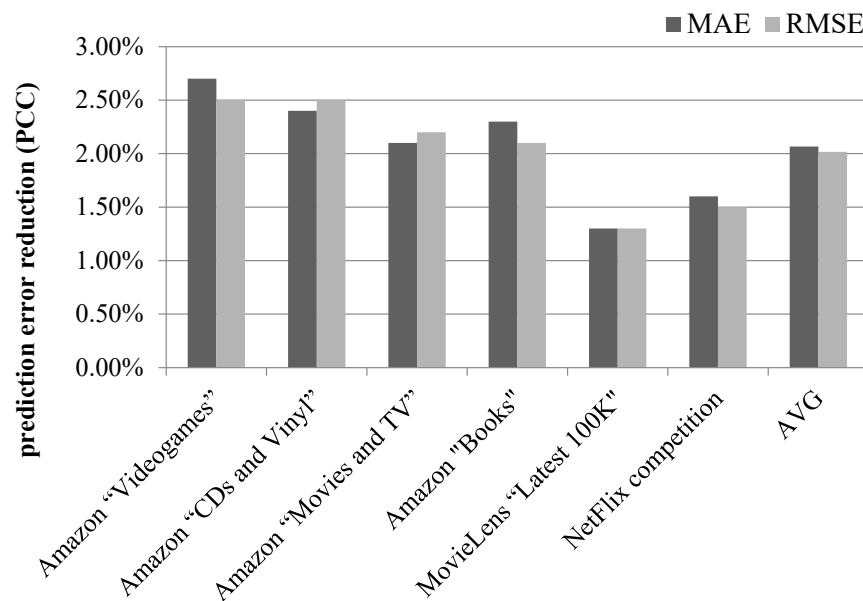


Figure 2. MAE and RMSE reduction achieved by the proposed algorithm, when using the PCC user similarity metric.

The proposed algorithm achieves an average prediction MAE reduction of 2.1% and an average prediction RMSE reduction of 2.0%, when using the PCC user similarity metric. Examining each dataset individually, the performance edge of the proposed algorithm against the plain CF algorithm ranges from 1.3% and 1.3% (for the MovieLens 100K dataset) to 2.7% and 2.5% (for the Amazon "Videogames" dataset), for the MAE and the RMSE metrics, respectively.

Figure 3 presents the accuracy gains achieved by the proposed algorithm in terms of the MAE and RMSE metrics, when using the CS similarity metric and again taking the performance of the plain CF algorithm as a yardstick.

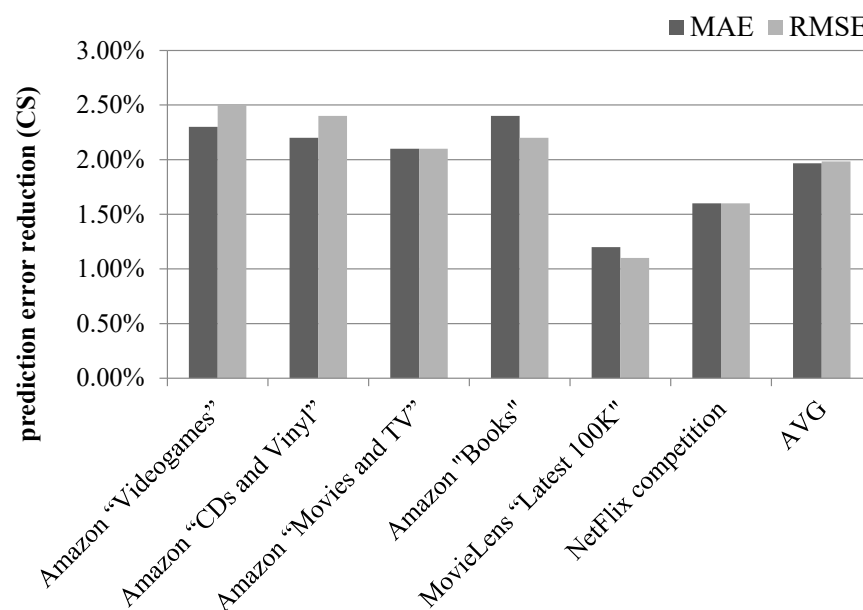


Figure 3. MAE and RMSE reduction achieved by the proposed algorithm, when using the CS user similarity metric.

The proposed algorithm achieves an average prediction MAE reduction of 2% and an average prediction RMSE reduction of 2%, as well, when using the CS user similarity metric. At the individual dataset level, the performance edge of the proposed algorithm against the plain CF algorithm, ranges from 1.2% and 1.1% (for the MovieLens 100K dataset) to 2.3% and 2.5% (for the Amazon “Videogames” dataset), for the MAE and the RMSE metrics, respectively.

4.3. Combining the Proposed Algorithm with a Second Algorithm Targeting Rating Prediction Accuracy Improvement

As stated in the introduction, the proposed algorithm can be easily fused with other CF approaches, aiming to enhance rating prediction accuracy.

The rationale behind the evaluation of the combination of the proposed algorithm with another algorithm is that, currently, many recommender systems have been implemented and use diverse algorithms that aim to achieve increased accuracy. A recommender system administrator may wonder whether the algorithm employed in their system needs to be replaced by the proposed one, and which would be the resulting benefits, or whether the proposed algorithm may be combined with the one already employed and if so, what would the benefits be. As a result, the following experiment offers useful insight regarding the additional accuracy gains that may be reaped for existing recommender systems, if the proposed algorithm is incorporated to complement any existing algorithm(s).

Towards this direction, the third experiment aims at assessing the rating prediction accuracy improvement when combining the proposed algorithm with another CF rating prediction accuracy approach. In particular, we report on our experiments where the proposed algorithm is combined with the CF_{EPC} algorithm [54]. The CF_{EPC} algorithm is a state-of-the-art algorithm (published towards the end of 2020), also targeting at improving the CF rating prediction accuracy, and not needing any additional information on the items or the users (e.g., user social relationships or item categories). Hence, it can be also applied in all CF datasets. Figure 4 illustrates the improvement in the MAE achieved by the inclusion/combination of the presented algorithm to the CF_{EPC} algorithm, when using the PCC as the similarity metric and again taking the performance of the plain CF algorithm as a yardstick.

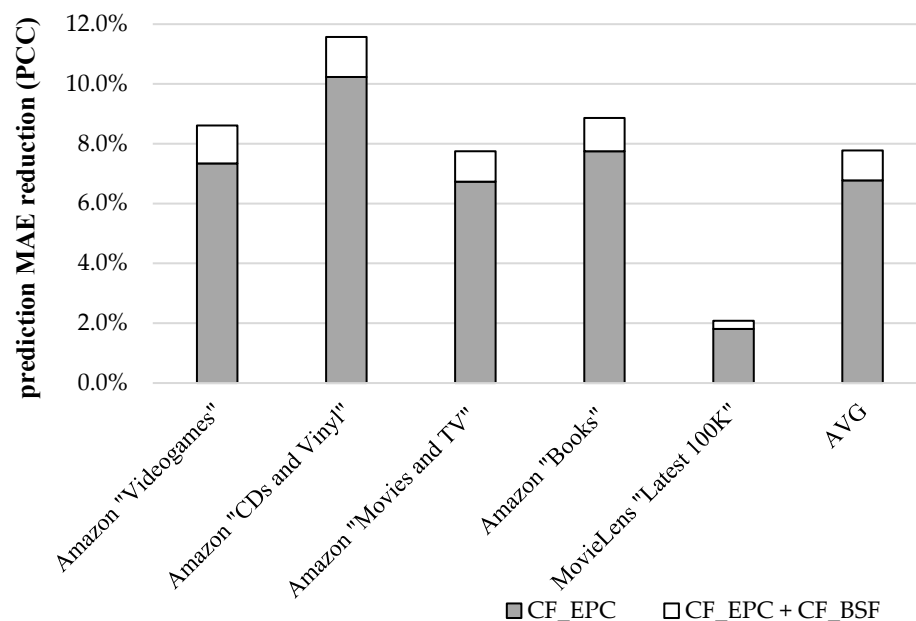


Figure 4. MAE reduction achieved by the inclusion of proposed algorithm to the CF_{EPC} algorithm, when using the PCC user similarity metric.

The combination of the CF_{EPC} algorithm with the proposed algorithm resulted in a relative improvement of 15%, on average in relation to the gains obtained when using the plain version of the CF_{EPC} (from 6.8% to 7.8%, in absolute figures), considering the MAE error metric. Similarly, the relative improvement, considering the RMSE error metric has been found to be equal to 19%, on average (from 5.8% to 6.9%, in absolute figures). The experiment demonstrates that the performance gains of the CF_{EPC} algorithm is further enhanced by approximately the 50% of the performance gains achieved when the proposed algorithm is independently applied on sparse datasets (i.e., the Amazon datasets), while for the dense dataset (Movielens Latest 100K dataset) the performance enhancement of the CF_{EPC} algorithm is approximately equal to the 25% of the gains achieved by the proposed algorithm on the same dataset.

Figure 5 illustrates the improvement in the MAE achieved by the inclusion/combination of the presented algorithm to the CF_{EPC} algorithm, when using the CS as the similarity metric and again taking the performance of the plain CF algorithm as a yardstick.

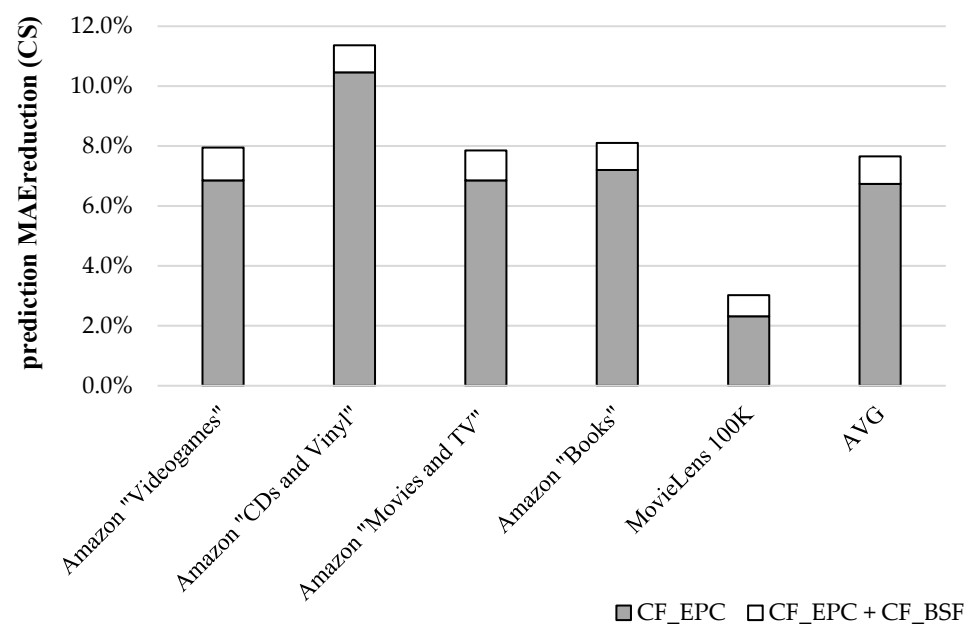


Figure 5. MAE reduction achieved by the inclusion of proposed algorithm to the CF_{EPC} algorithm, when using the CS user similarity metric.

The combination of the CF_{EPC} algorithm with the presented algorithm resulted in a relative improvement of 14%, on average in relation to the gains obtained when using the plain version of the CF_{EPC} (from 6.7% to 7.6%, in absolute figures), considering the MAE error metric. Similarly, the relative improvement, considering the RMSE error metric has been found to be equal to 15%, on average (from 6.1% to 6.9%, in absolute figures). The experiment demonstrates that the performance gains of the CF_{EPC} algorithm is further enhanced by approximately 50% of the performance gains achieved when the proposed algorithm is independently applied on sparse datasets (i.e., the Amazon datasets), while for the dense dataset (Movielens Latest 100K dataset) the performance enhancement of the CF_{EPC} algorithm is approximately equal to the 30% of the gains achieved by the proposed algorithm on the same dataset.

4.4. Complexity Analysis of the Proposed Algorithm

The procedure of computing the average rating value for each item, given by all users, is a procedure that can be easily executed offline (while loading the ratings from the rDB). In case the aforementioned procedure is selected to be executed online, its complexity is $O(r)$, where r is the number of all user ratings in the rating database. When a new rating is added to the database, the complexity to update the average is $O(1)$, since the new average

can be directly computed on the basis of the current one and the number of ratings for the item, as shown in Equation (5):

$$newMean(i) = \frac{currentMean(i).currentNumRatings(i) + newRating(i)}{currentNumRatings(i) + 1}, \quad (5)$$

Regarding space complexity, the overhead introduced by the procedure is 1 real number per item (its average rating) and hence negligible.

The procedure of finding the number of black sheep ratings for each pair of NNs (to compute the *bsf* factor for this pair of users) has a complexity of $O(\#NNs * \#commonRatings)$. According to [74,75], the top-K NNs are retained and the maximum number of NNs typically considered ranges from 20 to 60. The average number of common ratings for NNs pairs varies with the dataset, and the settings used to determine NNs; for instance, in [76] it is suggested that CF system implementors may opt to consider only NNs with at least 10 common ratings, to increase accuracy. In all cases, the computation of the *bsf* factor considers the common ratings for each pair of NNs, and therefore its complexity is identical to the computation of the similarity of the same pair of users, which is an integral step of the CF procedure; therefore, the introduction of the computation of the *bsf* factor does not affect the overall complexity of the algorithm. Notably, the computation of the *bsf* factor need only be performed between a user and his/her NNs (whose number is typically bounded by the K parameter of the top-K NN selection step), yielding significantly lower execution time than the computation of pairwise user similarities, which must be performed for all user pairs. The complexity of the rating prediction phase is not altered, as compared to the typical CF algorithm listed in Equation (3), since only one additional multiplication per considered rating is introduced.

Regarding space complexity, the overhead introduced by the need to maintain the *bsf* factor values is 1 number per each NN pair (the value of their black sheep factor) and can be easily accommodated in contemporary hardware.

5. Conclusion and Future Work

In this work, we have presented a novel CF algorithm that considers the information of the black sheep ratings between NNs in the CF rating prediction procedure for the improvement of the rating prediction accuracy. More specifically, a set of black sheep ratings between two NNs appears when they both like a generally unaccepted item (they both give a relatively high rating when compared to the relatively low average rating given by all database users for this item) or vice versa. The rationale behind the use of the aforementioned concept is derived from the fact that if a human likes an item (e.g., a TV series, a car model, a videogame) in the real world, that the majority of others do not (and hence this product obtains a relatively low average rating value), and this human finds another one that also likes the exact same product (which is a quite rare case), the probability of valuing his opinion with greater prominence than the opinions of other users, for a future recommendation, is relatively high.

We have experimentally validated the proposed algorithm through a set of experiments, using two user similarity metrics, namely the PCC and the CS (which are the two most used user similarity metrics in CF research [77–79]), two rating prediction error metrics, namely the MAE and the RMSE, and six datasets of diverse product categories (videogames, music, books, movies and TV series) to ensure the reliability and generalisability of the results. Furthermore, the proposed algorithm was tested both as a standalone application and combined with another CF algorithm also aiming at enhancing rating prediction accuracy [54]. The evaluation results have shown that significant prediction accuracy gains were introduced through the inclusion of the proposed algorithm. In the first case (standalone application) an average of 2% rating prediction error reduction was found, considering all cases. In the second case (when combined with another CF algorithm) the inclusion of the proposed algorithm achieved a further average rating prediction error

relative reduction of 16%. Lastly, regarding the time and space complexity of the proposed algorithm, it was shown that both overheads are small.

An identified limitation of this work is that datasets in which the majority of the user ratings are close to the mean value of the rating scale could potentially limit the gains that can be reaped by the algorithm.

Our future work will initially focus on addressing the aforementioned limitation. Furthermore, we plan to explore alternative methods for rating prediction error reduction in CF datasets, in general. Finally, we will examine the extension of the proposed algorithm so that it can include additional data sources, such as IoT data [80–82], social network-sourced information [44,83,84], and demographic features [85–87], targeting at further improving rating prediction accuracy.

Author Contributions: Conceptualization, D.M., D.S. and C.V.; methodology, D.M., D.S. and C.V.; software, D.M., D.S. and C.V.; validation, D.M., D.S. and C.V.; formal analysis, D.M., D.S. and C.V.; investigation, D.M., D.S. and C.V.; resources, D.M., D.S. and C.V.; data curation, D.M., D.S. and C.V.; writing—original draft preparation, D.M., D.S. and C.V.; writing—review and editing, D.M., D.S. and C.V.; visualization, D.M., D.S. and C.V.; supervision, D.M., D.S. and C.V.; project administration, D.M., D.S. and C.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analysed in this study. This data can be found: <https://cseweb.ucsd.edu/~jmcauley/datasets.html> (accessed on 11 June 2021) and <https://grouplens.org/datasets/movielens/> (accessed on 11 June 2021) and <https://www.kaggle.com/netflix-inc/netflix-prize-data> (accessed on 11 June 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Balabanović, M.; Shoham, Y. Fab: Content-based, collaborative recommendation. *Commun. ACM* **1997**, *40*, 66–72. [CrossRef]
- Lara-Cabrera, R.; González-Prieto, Á.; Ortega, F. Deep Matrix Factorization Approach for Collaborative Filtering Recommender Systems. *Appl. Sci.* **2020**, *10*, 4926. [CrossRef]
- Aditya, P.H.; Budi, I.; Munajat, Q. A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for E-commerce in Indonesia: A case study PT X. In Proceedings of the 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Malang, Indonesia, 15–16 October 2016; pp. 303–308.
- Cechinel, C.; Sicilia, M.-Á.; Sánchez-Alonso, S.; García-Barricóanal, E. Evaluating collaborative filtering recommendations inside large learning object repositories. *Inf. Process. Manag.* **2013**, *49*, 34–50. [CrossRef]
- Herlocker, J.L.; Konstan, J.A.; Terveen, L.G.; Riedl, J.T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **2004**, *22*, 5–53. [CrossRef]
- Chen, R.; Hua, Q.; Chang, Y.-S.; Wang, B.; Zhang, L.; Kong, X. A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks. *IEEE Access* **2018**, *6*, 64301–64320. [CrossRef]
- Jalili, M.; Ahmadian, S.; Izadi, M.; Moradi, P.; Salehi, M. Evaluating Collaborative Filtering Recommender Algorithms: A Survey. *IEEE Access* **2018**, *6*, 74003–74024. [CrossRef]
- Gong, S.; Ye, H.; Tan, H. Combining Memory-Based and Model-Based Collaborative Filtering in Recommender System. In Proceedings of the 2009 Pacific-Asia Conference on Circuits, Communications and Systems, Chengdu, China, 16–17 May 2009; pp. 690–693.
- Aramanda, A.; Md Abdul, S.; Vedala, R. A Comparison Analysis of Collaborative Filtering Techniques for Recommender Systems. In *Lecture Notes in Electrical Engineering*; Springer: Singapore, 2021; Volume 698, pp. 87–95.
- Zhang, R.; Liu, Q.; Li, C.G.; Wei, J.-X.; Ma, H. Collaborative Filtering for Recommender Systems. In Proceedings of the 2014 Second International Conference on Advanced Cloud and Big Data, Huangshan, China, 20–22 November 2014; pp. 301–308.
- Dong, M.; Yuan, F.; Yao, L.; Wang, X.; Xu, X.; Zhu, L. Trust in recommender systems: A deep learning perspective. *arXiv* **2020**, arXiv:2004.03774.
- Toma, C.L. Counting on Friends: Cues to Perceived Trustworthiness in Facebook Profiles. In Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media, Ann Arbor, MI, USA, 1–4 June 2014; pp. 495–504.

13. Bakshy, E.; Eckles, D.; Yan, R.; Rosenn, I. Social influence in social advertising: Evidence from field experiments. In Proceedings of the ACM Conference on Electronic Commerce, Valencia, Spain, 4–8 June 2012; pp. 146–161.
14. Han, P.; Xie, B.; Yang, F.; Shen, R. A scalable P2P recommender system based on distributed collaborative filtering. *Expert Syst. Appl.* **2004**, *27*, 203–210. [[CrossRef](#)]
15. Karabadjji, N.E.I.; Beldjoudi, S.; Seridi, H.; Aridhi, S.; Dhifli, W. Improving memory-based user collaborative filtering with evolutionary multi-objective optimization. *Expert Syst. Appl.* **2018**, *98*, 153–165. [[CrossRef](#)]
16. Pirasteh, P.; Hwang, D.; Jung, J.E. Weighted Similarity Schemes for High Scalability in User-Based Collaborative Filtering. *Mob. Netw. Appl.* **2015**, *20*, 497–507. [[CrossRef](#)]
17. Bell, R.M.; Koren, Y. Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights. In Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, USA, 28–31 October 2007; pp. 43–52.
18. Das, A.S.; Datar, M.; Garg, A.; Rajaram, S. Google news personalization: Scalable online collaborative filtering. In Proceedings of the 16th International Conference on World Wide Web—WWW'07, Banff, AB, Canada, 8–12 May 2007; ACM Press: New York, NY, USA, 2007; p. 271.
19. Margaritis, D.; Spiliotopoulos, D.; Karagiorgos, G.; Vassilakis, C. An Algorithm for Density Enrichment of Sparse Collaborative Filtering Datasets Using Robust Predictions as Derived Ratings. *Algorithms* **2020**, *13*, 174. [[CrossRef](#)]
20. Margaritis, D.; Spiliotopoulos, D.; Karagiorgos, G.; Vassilakis, C.; Vasilopoulos, D. On Addressing the Low Rating Prediction Coverage in Sparse Datasets Using Virtual Ratings. *SN Comput. Sci.* **2021**, *2*, 255. [[CrossRef](#)]
21. Chen, L.; Yuan, Y.; Yang, J.; Zahir, A. Improving the Prediction Quality in Memory-Based Collaborative Filtering Using Categorical Features. *Electronics* **2021**, *10*, 214. [[CrossRef](#)]
22. Singh, M.; Mehrotra, M. Impact of clustering on quality of recommendation in cluster-based collaborative filtering: An empirical study. *Int. J. Bus. Intell. Data Min.* **2020**, *17*, 206. [[CrossRef](#)]
23. Alhijawi, B.; Al-Naymat, G.; Obeid, N.; Awajan, A. Novel predictive model to improve the accuracy of collaborative filtering recommender systems. *Inf. Syst.* **2021**, *96*, 101670. [[CrossRef](#)]
24. Singh, P.K.; Pramanik, P.K.D.; Choudhury, P. An improved similarity calculation method for collaborative filtering-based recommendation, considering neighbor's liking and disliking of categorical attributes of items. *J. Inf. Optim. Sci.* **2019**, *40*, 397–412. [[CrossRef](#)]
25. Lima, G.R.; Mello, C.E.; Lyra, A.; Zimbrão, G. Applying landmarks to enhance memory-based collaborative filtering. *Inf. Sci.* **2020**, *513*, 412–428. [[CrossRef](#)]
26. Ekstrand, M.D. Collaborative Filtering Recommender Systems. *Found. Trends Hum. Comput. Interact.* **2011**, *4*, 81–173. [[CrossRef](#)]
27. Schafer, J.B.; Frankowski, D.; Herlocker, J.; Sen, S. Collaborative Filtering Recommender Systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 291–324.
28. Choi, K.; Suh, Y. A new similarity function for selecting neighbors for each target item in collaborative filtering. *Knowl. Based Syst.* **2013**, *37*, 146–153. [[CrossRef](#)]
29. Liu, H.; Hu, Z.; Mian, A.; Tian, H.; Zhu, X. A new user similarity model to improve the accuracy of collaborative filtering. *Knowl. Based Syst.* **2014**, *56*, 156–166. [[CrossRef](#)]
30. Zhang, L.; Li, Z.; Sun, X. Iterative rating prediction for neighborhood-based collaborative filtering. *Appl. Intell.* **2021**, *51*, 6810–6822. [[CrossRef](#)]
31. Shi, W.; Wang, L.; Qin, J. User Embedding for Rating Prediction in SVD++-Based Collaborative Filtering. *Symmetry Basel* **2020**, *12*, 121. [[CrossRef](#)]
32. Ren, L.; Wang, W. An SVM-based collaborative filtering approach for Top-N web services recommendation. *Futur. Gener. Comput. Syst.* **2018**, *78*, 531–543. [[CrossRef](#)]
33. Kuang, L.; Yu, L.; Huang, L.; Wang, Y.; Ma, P.; Li, C.; Zhu, Y. A Personalized QoS Prediction Approach for CPS Service Recommendation Based on Reputation and Location-Aware Collaborative Filtering. *Sensors* **2018**, *18*, 1556. [[CrossRef](#)]
34. Robertson, S. Understanding inverse document frequency: On theoretical arguments for IDF. *J. Doc.* **2004**, *60*, 503–520. [[CrossRef](#)]
35. Sánchez-Moreno, D.; López Batista, V.; Vicente, M.D.M.; Sánchez Lázaro, Á.L.; Moreno-García, M.N. Exploiting the User Social Context to Address Neighborhood Bias in Collaborative Filtering Music Recommender Systems. *Information* **2020**, *11*, 439. [[CrossRef](#)]
36. Chen, R.; Chang, Y.-S.; Hua, Q.; Gao, Q.; Ji, X.; Wang, B. An enhanced social matrix factorization model for recommendation based on social networks using social interaction factors. *Multimed. Tools Appl.* **2020**, *79*, 14147–14177. [[CrossRef](#)]
37. Bok, K.; Ko, G.; Lim, J.; Yoo, J. Personalized content recommendation scheme based on trust in online social networks. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5572. [[CrossRef](#)]
38. Ma, T.; Wang, X.; Zhou, F.; Wang, S. Research on diversity and accuracy of the recommendation system based on multi-objective optimization. *Neural Comput. Appl.* **2020**, *1–9*. [[CrossRef](#)]
39. Margaritis, D.; Vasilopoulos, D.; Vassilakis, C.; Spiliotopoulos, D. Improving Collaborative Filtering's Rating Prediction Accuracy by Introducing the Common Item Rating Past Criterion. In Proceedings of the 10th International Conference on Information, Intelligence, Systems and Applications, IISA 2019, Patras, Greece, 15–17 July 2019; pp. 1022–1027.
40. Thakkar, P.; Varma, K.; Ukani, V.; Mankad, S.; Tanwar, S. Combining User-Based and Item-Based Collaborative Filtering Using Machine Learning. In *Information and Communication Technology for Intelligent Systems*; Springer: Singapore, 2019; pp. 173–180.

41. Zarei, M.R.; Moosavi, M.R. A Memory-Based Collaborative Filtering Recommender System Using Social Ties. In Proceedings of the 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA), Tehran, Iran, 6–7 March 2019; pp. 263–267.
42. Fan, W.; Ma, Y.; Yin, D.; Wang, J.; Tang, J.; Li, Q. Deep social collaborative filtering. In Proceedings of the 13th ACM Conference on Recommender Systems, Copenhagen, Denmark, 16–20 September 2019; ACM: New York, NY, USA, 2019; pp. 305–313.
43. Natarajan, S.; Vairavasundaram, S.; Natarajan, S.; Gandomi, A.H. Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data. *Expert Syst. Appl.* **2020**, *149*, 113248. [CrossRef]
44. Yang, X.; Liang, C.; Zhao, M.; Wang, H.; Ding, H.; Liu, Y.; Li, Y.; Zhang, J. Collaborative Filtering-Based Recommendation of Online Social Voting. *IEEE Trans. Comput. Soc. Syst.* **2017**, *4*, 1–13. [CrossRef]
45. Zhang, T. Research on collaborative filtering recommendation algorithm based on social network. *Int. J. Internet Manuf. Serv.* **2019**, *6*, 343. [CrossRef]
46. Guo, L.; Liang, J.; Zhu, Y.; Luo, Y.; Sun, L.; Zheng, X. Collaborative filtering recommendation based on trust and emotion. *J. Intell. Inf. Syst.* **2019**, *53*, 113–135. [CrossRef]
47. Zhang, Z.-P.; Kudo, Y.; Murai, T.; Ren, Y.-G. Enhancing Recommendation Accuracy of Item-Based Collaborative Filtering via Item-Variance Weighting. *Appl. Sci.* **2019**, *9*, 1928. [CrossRef]
48. Zhang, L.; Wei, Q.; Zhang, L.; Wang, B.; Ho, W.-H. Diversity Balancing for Two-Stage Collaborative Filtering in Recommender Systems. *Appl. Sci.* **2020**, *10*, 1257. [CrossRef]
49. Yan, H.; Tang, Y. Collaborative Filtering Based on Gaussian Mixture Model and Improved Jaccard Similarity. *IEEE Access* **2019**, *7*, 118690–118701. [CrossRef]
50. Jiang, L.; Cheng, Y.; Yang, L.; Li, J.; Yan, H.; Wang, X. A trust-based collaborative filtering algorithm for E-commerce recommendation system. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 3023–3034. [CrossRef]
51. Veras De Sena Rosa, R.E.; Guimaraes, F.A.S.; da Silva Mendonça, R.; de Lucena, V.F. Improving Prediction Accuracy in Neighborhood-Based Collaborative Filtering by Using Local Similarity. *IEEE Access* **2020**, *8*, 142795–142809. [CrossRef]
52. Shahbazi, Z.; Hazra, D.; Park, S.; Byun, Y.C. Toward Improving the Prediction Accuracy of Product Recommendation System Using Extreme Gradient Boosting and Encoding Approaches. *Symmetry Basel* **2020**, *12*, 1566. [CrossRef]
53. Iftikhar, A.; Ghazanfar, M.A.; Ayub, M.; Mehmood, Z.; Maqsood, M. An Improved Product Recommendation Method for Collaborative Filtering. *IEEE Access* **2020**, *8*, 123841–123857. [CrossRef]
54. Margaris, D.; Spiliotopoulos, D.; Vassilakis, C.; Vasilopoulos, D. Improving collaborative filtering’s rating prediction accuracy by introducing the experiencing period criterion. *Neural Comput. Appl.* **2020**. [CrossRef]
55. Chen, V.X.; Tang, T.Y. Incorporating Singular Value Decomposition in User-based Collaborative Filtering Technique for a Movie Recommendation System. In Proceedings of the 2019 International Conference on Pattern Recognition and Artificial Intelligence—PRAI ’19, Wenzhou, China, 26–28 August 2019; ACM Press: New York, NY, USA, 2019; pp. 12–15.
56. Liu, X. Improved Collaborative Filtering Algorithm Based on Multi-dimensional Fusion Similarity. In Proceedings of the 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), Tianjin, China, 9–11 August 2019; pp. 440–443.
57. Singh, P.K.; Sinha, M.; Das, S.; Choudhury, P. Enhancing recommendation accuracy of item-based collaborative filtering using Bhattacharyya coefficient and most similar item. *Appl. Intell.* **2020**, *50*, 4708–4731. [CrossRef]
58. Cao, H.; Chen, Z.; Cheng, M.; Zhao, S.; Wang, T.; Li, Y. You Recommend, I Buy. *Proc. ACM Hum. Comput. Interact.* **2021**, *5*, 1–25. [CrossRef]
59. Chen, M.; Beutel, A.; Covington, P.; Jain, S.; Belletti, F.; Chi, E.H. Top-K Off-Policy Correction for a REINFORCE Recommender System. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 February 2019; ACM: New York, NY, USA, 2019; pp. 456–464.
60. Jia, H.; Saule, E. An Analysis of Citation Recommender Systems. In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, Sydney, Australia, 31 July–3 August 2017; ACM: New York, NY, USA, 2017; pp. 216–223.
61. Doan, T.-N.; Lim, E.-P. Modeling Check-In Behavior with Geographical Neighborhood Influence of Venues. In *Advanced Data Mining and Applications*; ADMA 2017 Lecture Notes in Computer Science; Cong, G., Peng, W.C., Zhang, W., Li, C., Sun, A., Eds.; Springer: Cham, Switzerland, 2017; Volume 10604, pp. 429–444.
62. Margaris, D.; Kobusinska, A.; Spiliotopoulos, D.; Vassilakis, C. An Adaptive Social Network-Aware Collaborative Filtering Algorithm for Improved Rating Prediction Accuracy. *IEEE Access* **2020**, *8*, 68301–68310. [CrossRef]
63. Hassanieh, L.A.; Jaoudeh, C.A.; Abdo, J.B.; Demerjian, J. Similarity measures for collaborative filtering recommender systems. In Proceedings of the 2018 IEEE Middle East and North Africa Communications Conference (MENACOMM), Jounieh, Lebanon, 18–20 April 2018; pp. 1–5.
64. Kumar, P.; Kumar, V.; Thakur, R.S. A new approach for rating prediction system using collaborative filtering. *Iran J. Comput. Sci.* **2019**, *2*, 81–87. [CrossRef]
65. McAuley, J.; Pandey, R.; Leskovec, J. Inferring Networks of Substitutable and Complementary Products. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD ’15, Sydney, Australia, 10–13 August 2015; pp. 785–794.
66. Amazon Product Data. Available online: <http://jmcauley.ucsd.edu/data/amazon/links.html> (accessed on 11 June 2021).
67. Movie Lens Datasets. Available online: <http://grouplens.org/datasets/movielens/> (accessed on 11 June 2021).

68. Harper, F.M.; Konstan, J.A. The Movie Lens Datasets. *ACM Trans. Interact. Intell. Syst.* **2016**, *5*, 1–19. [[CrossRef](#)]
69. Bennett, J.; Elkan, C.; Liu, B.; Smyth, P.; Tikk, D. KDD Cup and workshop 2007. *ACM SIGKDD Explor. Newsl.* **2007**, *9*, 51–52. [[CrossRef](#)]
70. Mei, D.; Huang, N.; Li, X. Light Graph Convolutional Collaborative Filtering with Multi-Aspect Information. *IEEE Access* **2021**, *9*, 34433–34441. [[CrossRef](#)]
71. Barkan, O.; Fuchs, Y.; Caciularu, A.; Koenigstein, N. Explainable Recommendations via Attentive Multi-Persona Collaborative Filtering. In Proceedings of the Fourteenth ACM Conference on Recommender Systems, Brasilia, Brazil, 22–26 September 2020; ACM: New York, NY, USA, 2020; pp. 468–473.
72. Zhang, Y.; Lou, J.; Chen, L.; Yuan, X.; Li, J.; Johnsten, T.; Tzeng, N.-F. Towards Poisoning the Neural Collaborative Filtering-Based Recommender Systems. In *Computer Security—ESORICS 2020, Proceedings of the 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, 14–18 September 2020*; Springer: Cham, Switzerland, 2020; pp. 461–479.
73. Fang, J.; Li, B.; Gao, M. Collaborative filtering recommendation algorithm based on deep neural network fusion. *Int. J. Sens. Netw.* **2020**, *34*, 71. [[CrossRef](#)]
74. Zhang, Z.; Kudo, Y.; Murai, T. Neighbor selection for user-based collaborative filtering using covering-based rough sets. *Ann. Oper. Res.* **2017**, *256*, 359–374. [[CrossRef](#)]
75. Herlocker, J.L.; Konstan, J.A.; Borchers, A.; Riedl, J. An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval—SIGIR '99, Berkeley, CA, USA, 15–19 August 1999; ACM Press: New York, NY, USA, 1999; pp. 230–237.
76. Saric, A.; Hadzikadic, M.; Wilson, D. Alternative Formulas for Rating Prediction Using Collaborative Filtering. In *International Symposium on Methodologies for Intelligent Systems*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 301–310.
77. Jain, G.; Mahara, T.; Tripathi, K.N. A Survey of Similarity Measures for Collaborative Filtering-Based Recommender System. In *Soft Computing: Theories and Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 343–352.
78. Bellogín, A.; Sánchez, P. Collaborative filtering based on subsequence matching: A new approach. *Inf. Sci.* **2017**, *418*, 432–446. [[CrossRef](#)]
79. Margaris, D.; Spiliotopoulos, D.; Vassilakis, C. 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), Vancouver, BC, Canada, 27–30 August 2019; ACM: Vancouver, BC, Canada, 2019; pp. 1160–1167.
80. Ortega, F.; González-Prieto, Á.; Bobadilla, J.; Gutiérrez, A. Collaborative Filtering to Predict Sensor Array Values in Large IoT Networks. *Sensors* **2020**, *20*, 4628. [[CrossRef](#)] [[PubMed](#)]
81. Cui, Z.; Xu, X.; Xue, F.; Cai, X.; Cao, Y.; Zhang, W.; Chen, J. Personalized Recommendation System Based on Collaborative Filtering for IoT Scenarios. *IEEE Trans. Serv. Comput.* **2020**, *13*, 685–695. [[CrossRef](#)]
82. Gao, H.; Xu, Y.; Yin, Y.; Zhang, W.; Li, R.; Wang, X. Context-Aware QoS Prediction with Neural Collaborative Filtering for Internet-of-Things Services. *IEEE Internet Things J.* **2020**, *7*, 4532–4542. [[CrossRef](#)]
83. Li, X.; Cheng, X.; Su, S.; Li, S.; Yang, J. A hybrid collaborative filtering model for social influence prediction in event-based social networks. *Neurocomputing* **2017**, *230*, 197–209. [[CrossRef](#)]
84. Zhou, Y.; Liu, L.; Lee, K.; Palanisamy, B.; Zhang, Q. Improving Collaborative Filtering with Social Influence over Heterogeneous Information Networks. *ACM Trans. Internet Technol.* **2020**, *20*, 1–29. [[CrossRef](#)]
85. Bobadilla, J.; González-Prieto, Á.; Ortega, F.; Lara-Cabrera, R. Deep learning feature selection to unhide demographic recommender systems factors. *Neural Comput. Appl.* **2021**, *33*, 7291–7308. [[CrossRef](#)]
86. Yassine, A.; Mohamed, L.; Al Achhab, M. Intelligent recommender system based on unsupervised machine learning and demographic attributes. *Simul. Model. Pract. Theory* **2021**, *107*, 102198. [[CrossRef](#)]
87. Keerthika, K.; Saravanan, T. Enhanced Product Recommendations based on Seasonality and Demography in Ecommerce. In Proceedings of the 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 18–19 December 2020; pp. 721–723.