big data and cognitive computing

MDPI

Article

Exploiting Rating Prediction Certainty for Recommendation Formulation in Collaborative Filtering

Dionisis Margaris [1,*], Kiriakos Sgardelis [1], Dimitris Spiliotopoulos [2] and Costas Vassilakis [3]

[1] Department of Digital Systems, University of the Peloponnese, Kladas, 231 00 Sparta, Greece; k.sgardelis@go.uop.gr
[2] Department of Management Science and Technology, University of the Peloponnese, Thesi Sechi, 221 31 Tripoli, Greece; dspiliot@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: Collaborative filtering is a popular recommender system (RecSys) method that produces rating prediction values for products by combining the ratings that close users have already given to the same products. Afterwards, the products that achieve the highest prediction values are recommended to the user. However, as expected, prediction estimation may contain errors, which, in the case of RecSys, will lead to either not recommending a product that the user would actually like (i.e., purchase, watch, or listen) or to recommending a product that the user would not like, with both cases leading to degraded recommendation quality. Especially in the latter case, the RecSys would be deemed unreliable. In this work, we design and develop a recommendation algorithm that considers both the rating prediction values and the prediction confidence, derived from features associated with rating prediction accuracy in collaborative filtering. The presented algorithm is based on the rationale that it is preferable to recommend an item with a slightly lower prediction value, if that prediction seems to be certain and safe, over another that has a higher value but of lower certainty. The proposed algorithm prevents low-confidence rating predictions from being included in recommendations, ensuring the recommendation quality and reliability of the RecSys.

Keywords: recommender systems; collaborative filtering; algorithm; rating predictions; uncertainty; confidence; evaluation

check for updates

Citation: Margaris, D.; Sgardelis, K.; Spiliotopoulos, D.; Vassilakis, C. Exploiting Rating Prediction Certainty for Recommendation Formulation in Collaborative Filtering. *Big Data Cogn. Comput.* **2024**, *8*, 53. https://doi.org/10.3390/bdcc8060053

Academic Editors: George Stalidis and Dimitrios Kardaras

Received: 25 March 2024
Revised: 17 May 2024
Accepted: 23 May 2024
Published: 27 May 2024

Copyright: © 2024 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 RecSys method that produces rating prediction values for products that a user U has not yet evaluated. This is accomplished by combining the rating values that users that have demonstrated similar likings to U, which are called U's near neighbors (NNs), have already given to these products. The CF method is based on the rationale that since a product is liked (or accepted, purchased, etc.) by user V and this user is deemed to have similar tastes and likings to user U ("is close to U"), then, transitively, user U will probably like this product as well. Afterwards, the products that achieved the highest rating prediction values are recommended to user U, since these products have the highest probability of being actually liked by U [1–4].

For example, if two products, p1 and p2, are candidates for recommendation to user U, and the rating prediction values for these products have been calculated to be respectively $rpv_{p1} = 5/5$ and $rpv_{p2} = 4.5/5$, the RecSys will obviously recommend the first one. While the aforementioned approach seems totally rational, let us consider the practice of humans when assessing reviews. Human users, besides the average review values, also consider the certainty of these prediction values, at least to some extent. In our example, if the value of $rpv_{p1} = 5/5$ is deemed to be of low certainty (e.g., it is based on only one individual's opinion), while $rpv_{p2} = 4.5/5$ has been calculated with high certainty (e.g., it is based

on 100 individuals' opinions), most people in this situation would trust rpv_{p2} more than rpv_{p1} and probably prefer product p2 over product p1, since $rpv_{p2} = 4.5/5$ is a very high prediction value, which also seems to be certain and safe. On the other hand, although $rpv_{p1} = 5/5$ is clearly a higher value, it may be subject to bias, superficial assessment, subjectivity issues, differences in the reason behind the reviews, or other factors. Hence, the selection of p1 entails significantly higher risks.

From the example presented, we can conclude that the certainty (or *confidence)* of rating prediction values, when it can be calculated, can be of great importance to the RecSys.

Recent works have explored prediction features associated with rating prediction accuracy in CF [5,6]. More specifically, these works showed that (1) the number of NNs taking part in the rating prediction formulation, (2) the average rating value of the item taking part in the rating prediction formulation (by all users who have rated it), and (3) the average rating value of the user for whom the rating prediction is being formulated (for all items he has rated) are positively associated with CF rating prediction accuracy. Although these findings provide valuable insights for rating predictions, they have not yet been utilized for rating prediction formulation, since an algorithm that can be directly used in CF datasets was not provided.

In this work, we take into account the observations of previous works [5,6] on the characteristics of rating predictions that can be considered to be of high certainty, and exploit them as follows:

(1) We introduce a novel rating-based recommendation formulation algorithm that exploits the certainty aspect of rating predictions to generate more accurate and useful recommendations.

(2) We validate the effectiveness of the proposed algorithm in terms of recommendation success through an extensive evaluation experiment. The evaluation process (a) includes six CF datasets from multiple sources, widely used in CF RecSys research; (b) considers both the correlation threshold technique and the top-k technique for the NN selection; and (c) employs both the Pearson Correlation Coefficient (PCC) and the Cosine (or Vector) Similarity (CS) for user similarity computation, both widely used in CF RecSys research.
Since the presented algorithm targets the recommendation formulation stage, for the assessment of the algorithm performance, we utilize pertinent metrics, and more specifically (i), the average rating value of the top-N recommended items, (ii) precision, and (iii) the normalized discounted cumulative gain (NDCG) [7–11].

(3) We analyze the additional computational and storage costs incurred in order to compute, store, and utilize the additional data needed by the proposed algorithm, demonstrating its feasibility and applicability.

In order for (i) this algorithm to have universal application and (ii) the results to be independent of the selection of other algorithms and methods (e.g., user review analysis, user social relations, item categories, etc.), only the tuple <user_id, item_id, numeric_rating> is used as input, without any complementary data. We note, however, that—due to its generic nature—the proposed algorithm can be combined with other algorithms aiming to increase rating prediction accuracy, coverage (i.e., the percentage of cases for which a personalized recommendation can be computed), or recommendation accuracy.

The remainder of the paper is organized as follows: in Section 2, we review related work, while in Section 3, we present the prerequisites, the CF methodology, and the proposed algorithm. In Section 4, we present the results of the evaluation of the proposed algorithm, while in Section 5, we analyze the results and discuss the algorithm's limitations. Lastly, in Section 6, we conclude the paper and summarize our future work.

2. Related Work

CF recommendation quality is a research area that has attracted considerable research attention over the last 10 years. This research area consists of two main categories. The first includes research works that incorporate complementary input data, from the char-

acteristics and categories/taxonomy of the products to user social relations and review data. The second includes research works that are solely based on the data stored in the rating matrix.

Regarding the first category, Yang et al. (2017) propose a model-based technique that integrates sparse rating social information with trust network information between the same users in order to upgrade the CF recommendation performance [12]. More specifically, this technique adopts matrix factorization methods to accurately reflect the influence of the users on the formulation of their opinions. Wang et al. (2017) introduced a hybrid technique that incorporates the rating preference of users, the asymmetry between users, the non-linear relationship between variables, and the influence of all possible rated items [13]. Furthermore, they use a comprehensive non-linear formula as well as asymmetric and user preference factors. Xin et al. (2019) introduced an algorithm that employs multiple item relations in RecSys, namely Relational CF [14]. This algorithm infers user preferences by considering both the relation value and the relation type of the items. Furthermore, they design and develop a hierarchical attention method that can model user preferences by considering the relation values of an item contribution and distinguishing the most important types of relations.

Jiang et al. (2019) presented a technique that can be applied to a plethora of RecSys [15]. This slope-one technique is based on user–user similarities and the fusion of trusted data. It contains the following three steps to produce the final recommendation function: trusted data selection, user similarity calculation, and insertion of the aforementioned similarity into the weight factor of the slope one technique. Zhang et al. (2021) presented a deep learning service recommendation CF algorithm [16]. This algorithm embeds the similarity adaptive corrector in the output layer in order to correct the predictive quality of service. The nonlinear and high-dimensional characteristics are captured in a multilayer-perceptron, and high-dimensional vectors are embedded with location features.

Alhijawi and Kilani (2020) introduced BLIGA, a genetic-based RecSys that is based on historical ratings and semantic data [17]. This RecSys evaluates the candidate recommendation lists before producing the recommendation, instead of evaluating the items. BLIGA uses a genetic algorithm to find the optimal item list. It evaluates the users by utilizing fitness functions. These functions estimate the satisfaction level similarity between users and utilize item semantic data and semantic information to estimate item semantic similarity strength. Nilashi et al. (2018) introduced a hybrid CF RecSys algorithm that utilizes dimensionality reduction and ontology methods to overcome the drawbacks of scalability and sparsity [18]. Afterwards, an ontology is used to enhance the CF recommendation accuracy. Clustering methods with singular value decomposition techniques are utilized to discover the closest items and users within a cluster.

All the aforementioned works achieved improvements concerning CF recommendation quality; however, they cannot have universal application, since complementary input data, which they all incorporate, cannot be provided in every case (i.e., in every CF dataset).

In this direction, Liao and Lee (2016) presented a method that reduces the number of items' dimensionality by using a self-constructing clustering technique [19]. More specifically, this method sends dissimilar items to different clusters and similar ones to the same cluster. Then, each user is presented with a ranked set of recommended items. Chen et al. (2021) introduced a CF-based recommendation algorithm that incorporates the dynamic decay function [20]. This function can capture the users' preference variations and interest evolution.

Margaris and Vassilakis (2017) presented a CF algorithm that removes old user ratings from the rating database under the assumption that these do not reflect the users' tastes and likings in the present [21]. This algorithm can be executed in an unsupervised fashion and is found to be able to both enhance rating prediction accuracy and minimize the CF rating database size. Diaz et al. (2019) introduced a Bayesian algorithm that allows the justification of the produced recommendations [22]. This algorithm defines methods based on item-based and user-based CF. It combines these methods into a single user-item method.

Neysiani et al. (2019) aimed to provide a solution to the problem of multi-objective rule mining by using a weighted quality metric [23]. This metric utilizes the confidence and support of association rules, which are found using a genetic algorithm. Furthermore, the use of the genetic algorithm results in fast association rule discovery.

Ren and Wang (2018) introduced a CF-based service recommendation algorithm, which is based on a support vector machine [24]. With the use of this machine, the services that are of no interest to users, utilizing user historical rating data, are filtered out. Furthermore, the distance between the separating hyperplane and the point representing the service is used to compute the preference degree of a user. Thakkar et al. (2019) introduced an algorithm that synthesizes user-based and item-based predictions to produce CF rating predictions with low prediction deviation [25]. The proposed algorithm supports vector regression and also utilizes multiple linear regression.

He et al. [26] introduced a new Graph Convolution Network recommendation model, namely LightGCN, which comprises two components, a light graph convolution and a layer combination. The first component is a simplified form of the fully fledged GCN method, omitting the operations of feature transformation and non-linear activations, which complicate the training phase. The second component computes the final embeddings of the nodes, as the weighted sum of their embeddings on all layers. LightCGN is shown to outperform NGCF. Liu et al. [27] introduced an interest-aware messaging-passing graph convolution network recommendation model that follows the LightGCN simplified network structure. This model can partially alleviate the over-smoothing issue by grouping items and users into subgraphs and then applying to these subgraphs high-order graph convolutions. Li et al. [28] presented the Disentangled Graph Neural Network, a technique that combines the consideration of factor-level ratings on each item with the session purpose. This technique initially utilizes the disentangled learning method to cast embeddings of items into multiple factor embeddings, and afterwards, in order to learn the embedding factor effectively, it employs the gated graph neural network. Wei et al. [29] introduced a Light Graph Transformer model that learns user preference representation from the rated item features by establishing correlations between items. It develops a layer-wise position encoder and a modal-specific embedding for the vicinity calculation, as well as improves the self-attention scoring efficiency by presenting a light self-attention block.

Still, none of the aforementioned research works utilize rating prediction features to reduce the uncertain rating predictions from becoming recommendations in CF and, at the same time, retain universal application, which is only based on very basic CF information.

Recent works have explored simple prediction features associated with rating prediction accuracy in CF. However, they did not provide an algorithm that can be directly used in CF datasets [5,6].

This work advances the state-of-the-art research on recommendation accuracy in CF by introducing an algorithm that is based only on very basic CF information. It considers not only the rating prediction value but also the rating prediction (un-)certainty that is derived from features that have been proven to be associated with rating prediction accuracy in CF to provide more successful recommendations to its users.

3. Prerequisites and the Proposed Algorithm

In the following subsections, the procedure of recommendation formulation in CF is briefly presented for self-containment purposes (Section 3.1). The features related to rating prediction accuracy in CF are analyzed in Section 3.2, and finally, the proposed algorithm is introduced in Section 3.3.

3.1. CF Recommendation Formulation Procedure

The recommendation formulation procedure of a typical CF RecSys, for a user U, consists of four steps.

In step 1, the RecSys calculates the similarity between U and the other users in the rating database using a similarity metric. The similarity value, in most of the CF similarity

metrics, is based on the ratings that users have already given to commonly rated items. Popular CF user similarity metrics include the PCC, the CS, the Jaccard Index, the Spearman Correlation, etc. [30–32]. For most of the metrics, the similarity range is either [0, 1] or [−1, 1], where higher values indicate higher similarity (also referred to as *vicinity*) between the users.

In step 2, the users that will play the role of U's NNs in the recommendation procedure are selected. More specifically, these NN users' ratings will be used to formulate the rating prediction values for user U. Popular methods for selecting the NNs in CF are the correlation threshold method and the top-k method. The first one designates as U's NNs those users V whose similarities with U exceed some specified threshold value. The second one retains as U's NNs the k users with the highest similarity with U, where k is a parameter of the algorithm [33–35].

In step 3, the rating prediction numeric values for the items that user U has not yet rated are computed. This is accomplished by combining the rating values of U's NNs, selected in the previous step, for the exact same items. To this end, a rating prediction formula is used. In this work, we use the most popular method for rating prediction formulation in CF, which is the mean-centered formula [36–38].

The last step (step 4) of a RecSys is to formulate the recommendation to the user. It typically recommends the item(s) that achieved the highest rating prediction numeric value(s) in the previous step to user U in the most effective way [39–41].

3.2. CF Rating Prediction Accuracy Features

Recent research works have investigated the relationship between rating prediction features and prediction accuracy in CF [5,6]. More specifically, these works examined CF prediction features and their association with rating prediction accuracy in CF systems. Both works included experiments, which proved the following:

(1) The number of NNs taking part in the rating prediction formulation (NN_no) is associated with the accuracy of the CF rating prediction.
(2) The average rating value of the item taking part in the rating prediction formulation (by all users who have rated it) is associated with the accuracy of the CF rating prediction.
(3) The mean rating numeric value of the user concerning the formulated rating prediction (for all items he has rated) is associated with the accuracy of the CF rating prediction.

In regard to the first feature (NN_no), it was found that a higher number of NNs that take part in the rating prediction formulation is positively associated with increased rating prediction accuracy in CF. Regarding the sparse datasets (with density << 1%), if NN_no $\geq$ 2, then the rating prediction is considered to have relatively high accuracy, while if NN_no $\geq$ 4, then the rating prediction is considered to have very high accuracy. Regarding the dense datasets, if the percentage of NNs taking part in the CF rating prediction is $\geq$6%, then the rating prediction is considered to have relatively high accuracy. If this percentage is $\geq$15%, then the rating prediction is considered to have very high accuracy.

Regarding both the second and third features, when the average rating value is near the limits of the rating range, the rating prediction is considered to have high accuracy. More specifically, when considering a five-star rating scale (as for most of the datasets used in CF research), when either the user's mean rating value or the item's mean rating value are $\leq$2.0 or $\geq$4.0, then the rating prediction is considered to have relatively high accuracy. When the same averages are $\leq$1.5 or $\geq$4.5, then the rating prediction is considered to have very high accuracy.

3.3. The Proposed Algorithm

The proposed algorithm introduces three additional steps in the CF recommendation process:

1. *A preprocessing step*, where, for each item, the mean rating value given by all users who have rated it is computed. More formally, the mean rating value for each item is computed as follows:

$$\overline{r_i} = \frac{\sum_{U \in Raters(i)} r_{U,i}}{|Raters(i)|}$$

where $Raters(i)$ is the set of users that have given a rating for item i, and $r_{U,i}$ is the rating of each user U to this item.

2. During step 3 of a typical RecSys (described in Section 3.1), along with the prediction computation, the NNs number that took part in the prediction is also stored. The rating prediction of user U to item i will be defined as $rp_{U,i}$, while the NNs number that took part in the prediction will be defined as $NN\_no(rp_{U,i})$. Typically, $NN\_no(rp_{U,i})$ is calculated as follows:

$$NN\_no(rp_{U,i}) = |\{V \in NN(U) : r_{U,i} \neq NULL\}|$$

where $NN(U)$ is the set of user U's NNs, which is determined according to the strategy employed by the RS (i.e., either (a) their similarity to U exceeds the specified threshold THR or (b) they are the k users having the highest similarity with U).

3. Lastly, in the fourth step of a typical RecSys, predictions (i) either concerning items without an average rating value given by all users, near the limits of the rating scale, or (ii) produced with a low number of NNs are deemed as uncertain and hence are excluded from being recommended to the user. Formally, in order to compute a recommendation for user U, a *candidate recommendation item set* for user U $CRIS_U$ is computed as follows:

$$CRIS_U = \{i \in I : r_{U,i} = NULL \bigwedge (NN\_no(rp_{U,i}) \geq RECOM\_THR) \bigwedge ((\overline{r_i} \leq LOW\_ITEM\_AVG\_THR) \bigvee (\overline{r_i} \geq HIGH\_ITEM\_AVG\_THR))\}$$

where I is the set of all items, $RECOM\_THR$ is the minimum number of recommenders for a rating prediction to be considered reliable, and $LOW\_ITEM\_AVG\_THR$ (resp. $HIGH\_ITEM\_AVG\_THR$) is the threshold for the average item ratings by all users below which (resp. above which) the rating prediction is considered reliable. After $CRIS_U$ has been computed, the elements of this set with the highest rating predictions are selected for recommendation to the user.

Figure 1 illustrates the workflow of the proposed algorithm. The processing steps in the upper lane correspond to the workflow of the typical CF recommendation algorithm, while the processing steps in the lower lane correspond to the additional steps introduced in the proposed algorithm.

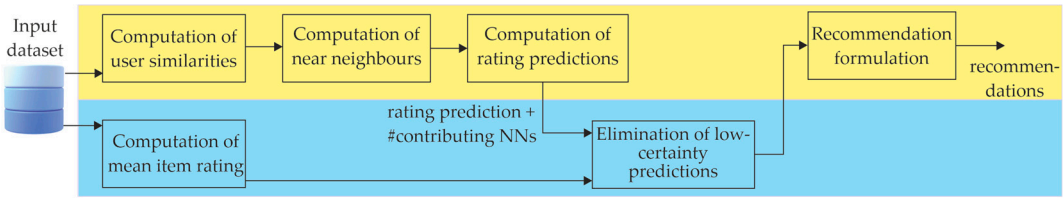

Figure 1. Workflow of the proposed algorithm.

In the following section, we evaluate the proposed algorithm in terms of recommendation accuracy.

4. Experimental Evaluation

In this section, we report on the experiments aiming to evaluate the proposed algorithm in terms of recommendation accuracy.

In order to warrant the generalizability of the experiment output, the experiments in our evaluation have been designed and executed to:

1. Include six CF datasets from multiple sources that are widely used in CF RecSys research.
2. Use both the correlation threshold technique and the top-k technique for NN selection.
3. Include both the PCC and the CS for user similarity, which are the most widely used similarity metrics in CF RecSys research.

The six datasets, along with their attributes, are summarized in Table 1. We can observe that datasets from multiple sources are used. Also, the selected datasets include several product fields, such as videogames, movies, and digital music.

Table 1. The datasets used in the experiments.

Dataset Name	#Ratings	Ratings Range	#Users	#Items	Density Characterization
MovieLens 1 M (https://grouplens.org/datasets/movielens/, accessed on 5 May 2024)	1 M	1–5	6 K	3.7 K	dense
MovieLens 100 K (https://grouplens.org/datasets/movielens/, accessed on 5 May 2024)	100 K	0.5–5.0	600	10 K	dense
Amazon Videogames (https://nijianmo.github.io/amazon/index.html, accessed on 5 May 2024)	473 K	1–5	17.5 K	55 K	sparse
Amazon Digital Music (https://nijianmo.github.io/amazon/index.html, accessed on 5 May 2024)	145 K	1–5	12 K	17 K	sparse
Epinions (https://www.kaggle.com/datasets/masoud3/epinions-trust-network, accessed on 5 May 2024)	922 K	1–5	22 K	296 K	sparse
CiaoDVD (https://guoguibing.github.io/librec/datasets.html, accessed on 5 May 2024)	73 K	1–5	17.6 K	16 K	sparse

Regarding NN set selection, the KNN and the similarity threshold are used since these are the two most widely used methods in CF research. In the first method, K users having the highest similarity values with the active user are used as NNs. In the second method, the users whose similarity values with the active user exceed a specific threshold are used as NNs. For the KNN method, we run experiments with K = 200 and K = 500 based on the selection of K in the experimental sections of related works [3,36,42,43]. For the similarity threshold method, we run experiments with THR = 0.0 and THR = 0.5 since the two similarity metrics used do not share a common value range. More specifically, the PCC has a range of $[-1, 1]$, while the CS has a range of $[0, 1]$ when applied to datasets with positive rating values, as are the six datasets used in this work.

Regarding the evaluation metrics used in our experiments, the standard rating prediction error metrics, such as the Mean Average Error and the Root Mean Square Root (MAE and RMSE, respectively), cannot be used to evaluate the presented algorithm. This is because the presented algorithm targets the recommendation formulation phase rather than the rating prediction computation phase. As a result, the metrics used for evaluating the presented algorithm are as follows: (i) the average real numeric rating value of the top-N recommended items; (ii) their precision; and (iii) their NDCG. Regarding the value of the N number in the top-N selection, we use the numbers three and five, since these numbers are used in many recent RecSys research works [44–48]. Lastly, regarding the precision metric, we follow the approach adopted by numerous works, including [49–51], where the recommendation includes all items for which the prediction falls in the top 30% of the rating scale (in our six datasets, this threshold is set to 3.5/5).

In the following subsections, we analyze the settings of the experiment as well as present and discuss the evaluation results.

4.1. Experimental Settings and Procedure

In this work, we follow the five-fold cross validation process, which is a very common validation procedure in CF [52–56]. In this process, we split the rating dataset into five equal folds. For each iteration, one fold is used as the test set, while the union of the remaining four folds is used as the train set. Lastly, the union of the five iterations (on the test sets) is used for the result.

In order for (i) the results to be applicable on every CF-based RecSys and (ii) the results to be independent of the selection of other algorithms and methods exploiting additional data, which may not be available in all cases (e.g., user review analysis, user social relations, item categories, etc.), only the tuple <user_id, item_id, numeric_rating> is used as input, without any complementary data.

All of the reported experiments were executed on a laptop equipped with 8 GB of RAM and an AMD Ryzen 5 4600 H CPU. Since the experiment code was written as a single-threaded application, only one of the CPU cores was utilized. The code realizing both the baseline and the proposed algorithm was written in C, with custom indexes to improve execution efficiency. All input files were formatted as text files, with each line accommodating a <user_id, item_id, rating> record.

4.2. Evaluation

In this subsection, we report on the experiments, aiming to assess the presented algorithm in terms of recommendation success. The presentation of the results is organized into the following two subsections: Section 4.2.1 presents results under the PCC similarity metric, while Section 4.2.2 discusses the results under the CS similarity metric.

4.2.1. Evaluation Using the PCC Similarity Metric

Figure 2 illustrates the average (real) rating value of the top-3 items recommended, before and after the application of the proposed algorithm in all six datasets, using the top-k technique, with K = 200 for the NN selection, and using the PCC similarity metric; the average across all datasets is also depicted.

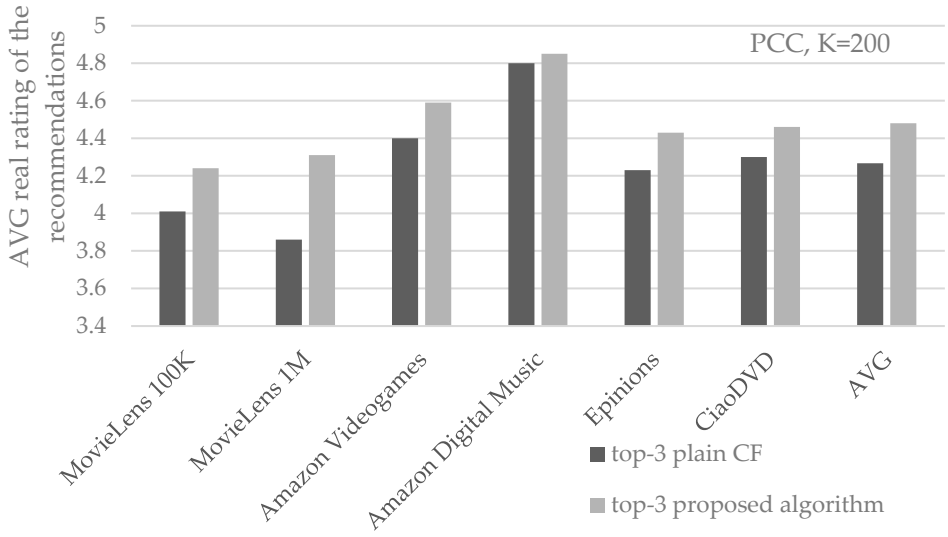

Figure 2. Average rating value of the top-3 recommended items using the PCC user similarity metric and setting K = 200.

The proposed algorithm achieves an average increase in the real rating recommendation value of 5% (from 4.27/5 to 4.48/5), considering the average across all datasets. Examining each dataset individually, it can be seen that when the proposed algorithm is applied to the MovieLens 1 M dataset, where the lowest initial average rating of the recommendations is noticed (due to the highest rating prediction error—both in terms of

MAE and RMSE), at 3.86/5, this average is enhanced to 4.31/5. At the other end, when the proposed algorithm is applied to the Amazon Digital Music dataset (which is the dataset with the lowest rating prediction error), with a 4.8/5 initial average rating of the recommendations, it achieves a smaller enhancement, reaching a value of 4.85/5. The decrease in the performance gains achieved is expected since the improvement margin in the case of the Amazon Digital Music dataset is very small.

Figure 3 illustrates the average recommendation precision value of the top-3 items recommended before and after the application of the proposed algorithm in all six datasets, using the top-k technique with K = 200 for the NN selection and using the PCC similarity metric. The proposed algorithm achieves an average increase in the recommendation precision value of 8.6% (from 82% to 89%). Examining each dataset individually, we can observe that when the proposed algorithm is applied to the MovieLens 1 M dataset, where the lowest initial recommendation precision is noticed, at 67.9% (which practically means that, on average, the users would dislike almost one out of three recommendations made to them), this average is increased to 83.9%. At the other end, we can notice that when the proposed algorithm is applied to the Amazon Digital Music dataset, with 96% recommendation precision, which is very close to the optimum performance, it achieves further improvement, reaching 97.4%.

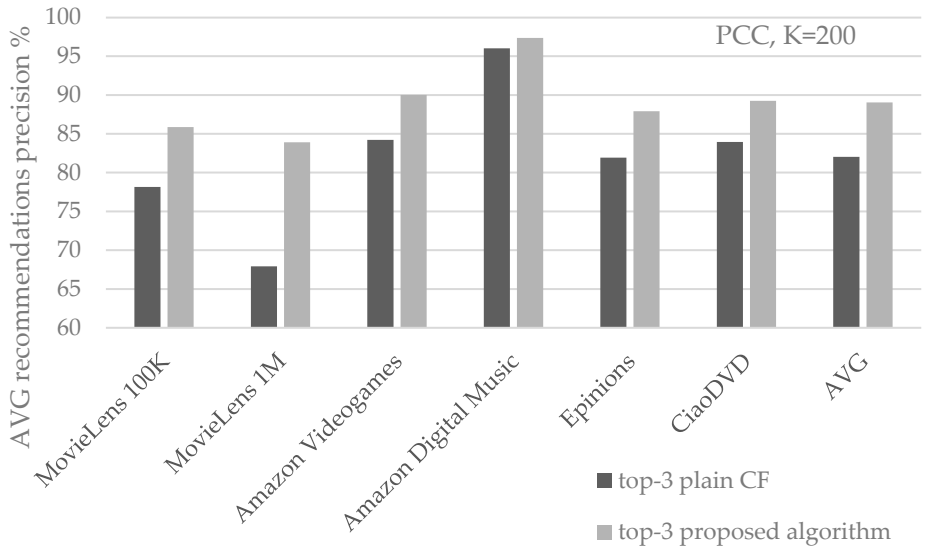

Figure 3. Average precision value of the top-3 recommended items using the PCC user similarity metric and setting K = 200.

Figure 4 illustrates the average NDCG value of the top-3 items recommended, before and after the application of the proposed algorithm in all six datasets, using the top-k technique with K = 200 for the NN selection and the PCC similarity metric. The proposed algorithm achieves an increase in the NDCG in all datasets tested, while its average value is increased from 0.971 to 0.98.

When we increase the number of recommended items to five (top-5 recommended items), similar results are observed. The average real rating recommendation value is enhanced by 4.7% (from 4.28 to 4.48). Regarding the average recommendation precision value, this is also enhanced by 8.2% (from 82.5% to 89.3%), while the average NDCG value is increased from 0.966 to 0.978.

When we increase the number of NNs taking part in the rating prediction formulation to K = 500, similar results are also observed for both the top-3 recommendations and the top-5 recommendations per user. More specifically, in the first case (top-3 recommendations), the proposed algorithm achieves an average enhancement in the real rating recommendation value of 5% (from 4.29/5 to 4.5/5), an average enhancement in the recommendation precision value of 8.3% (from 82.5% to 89.4%), and an average NDCG value

increase from 0.971 to 0.981. In the second case (top-5 recommendations), the proposed algorithm achieves an average enhancement in the real rating recommendation value of 4.5% (from 4.31/5 to 4.5/5), an average enhancement in the recommendation precision value of 7.8% (from 83.2% to 89.7%), and an average NDCG value increase from 0.966 to 0.979.

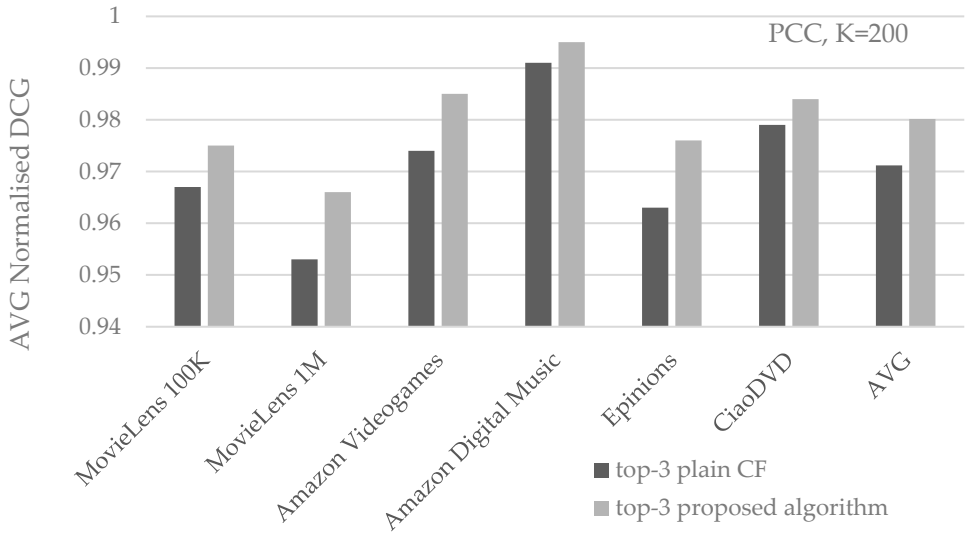

Figure 4. Average NDCG value of the top-3 recommended items using the PCC user similarity metric and setting K = 200.

Figure 5 illustrates the average (real) rating value of the top-3 items recommended, before and after the application of the proposed algorithm in all six datasets, using the correlation threshold technique with THR = 0.5.

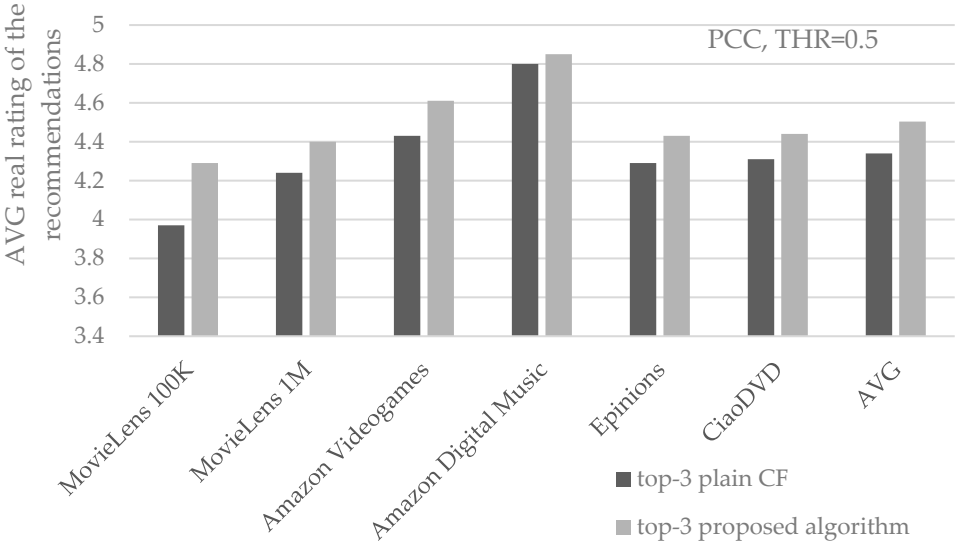

Figure 5. Average rating value of the top-3 recommended items using the PCC user similarity metric and setting THR = 0.5.

The presented algorithm achieves an average enhancement in the real rating recommendation value of 3.8% (from 4.34/5 to 4.5/5). Again, we can observe that, even when the proposed algorithm is applied to a dataset with a 4.8/5 initial average rating of the recommendations (the Amazon Digital Music), and, hence, a limited improvement margin, it achieves an increase to 4.85/5.

Figure 6 illustrates the average recommendation precision value of the top-3 items recommended, before and after the application of the proposed algorithm in all six datasets,

using the correlation threshold technique with THR = 0.5 for the NN selection. The proposed algorithm achieves an average enhancement in the recommendation precision value of 6% (from 84.6% to 89.7%). We can again see that even when the proposed algorithm is applied to a dataset that has 96.1% initial average recommendation precision (the Amazon Digital Music), it achieves an increase to 97.4%.

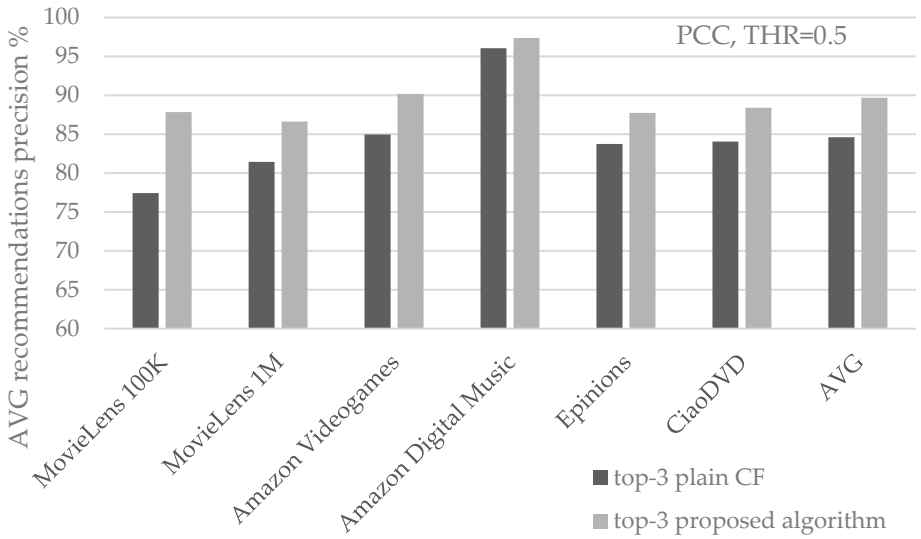

Figure 6. Average precision value of the top-3 recommended items using the PCC user similarity metric and setting THR = 0.5.

Figure 7 illustrates the average NDCG value of the top-3 items recommended before and after the application of the proposed algorithm in all six datasets, using the correlation threshold technique with THR = 0.5 for the NN selection and using the PCC similarity metric. The presented algorithm achieves an increase in the NDCG in all datasets tested, while its average value increases from 0.973 to 0.98.

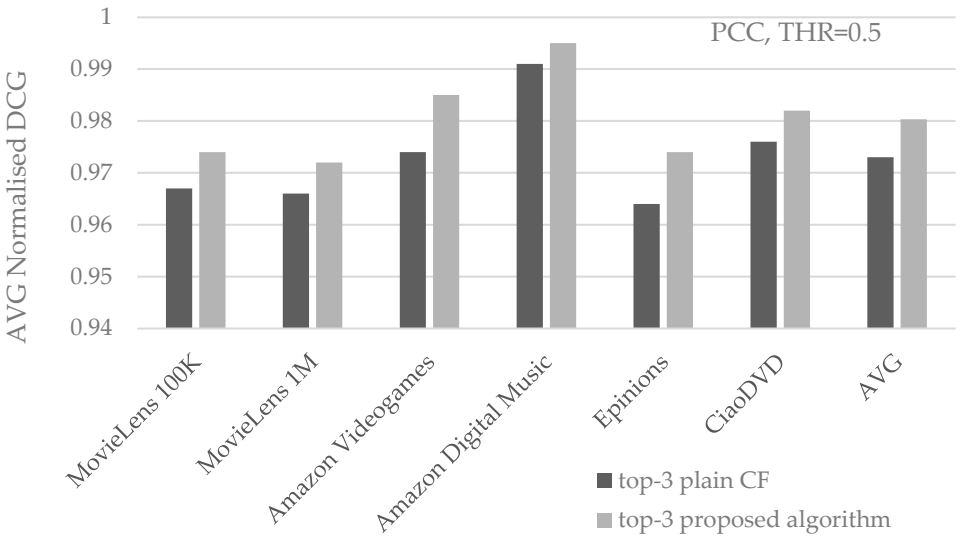

Figure 7. Average NDCG value of the top-3 recommended items when using the PCC user similarity metric and setting THR = 0.5.

When we increase the number of recommended items to five (the top-5 recommended items), similar results are observed, where the average real rating recommendation value is enhanced by 3.5% (from 4.35 to 4.5). Regarding the average recommendation precision

value, this is also enhanced by 5.7% (from 84.8% to 89.7%), while the average NDCG value increases from 0.968 to 0.98.

When we reduce the similarity threshold of the NNs taking part in the rating prediction formulation to THR = 0.0, similar results are also observed for both the top-3 recommendations and the top-5 recommendations per user. More specifically, in the first case (top-3 recommendations) the presented algorithm achieves an average enhancement in the real rating recommendation value of 2.7% (from 4.39/5 to 4.51/5), an average enhancement in the recommendation precision value of 3.6% (from 86% to 89.1%), and an average NDCG value increase from 0.975 to 0.981. In the second case (top-5 recommendations), the proposed algorithm achieves an average enhancement in the real rating recommendation value of 2.7% (from 4.39/5 to 4.51/5), an average enhancement in the recommendation precision value of 4.3% (from 86.1% to 89.8%), and an average NDCG value increase from 0.970 to 0.978.

4.2.2. Evaluation Using the CS Similarity Metric

Figure 8 illustrates the average (real) rating value of the top-3 items recommended before and after the application of the proposed algorithm in all six datasets, using the top-k technique with K = 200, for the NN selection and using the CS similarity metric.

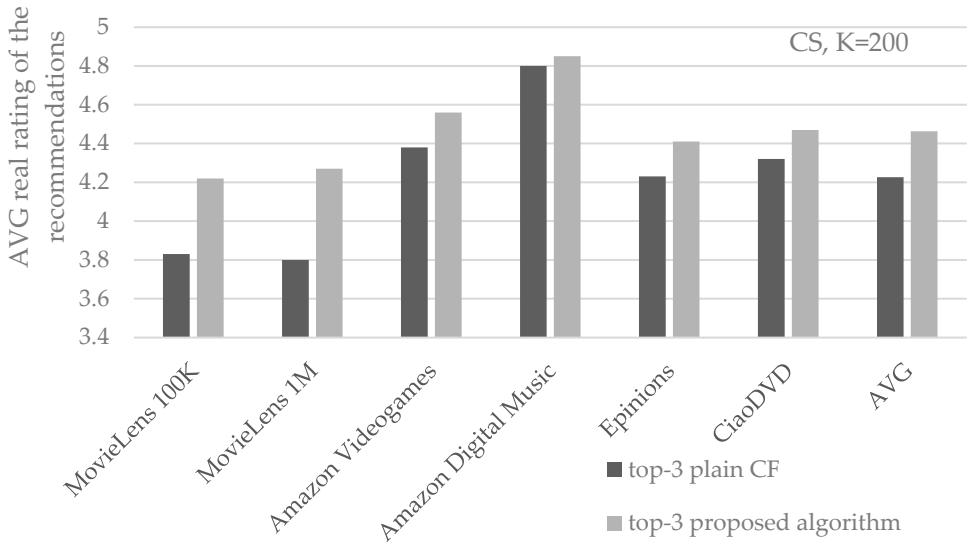

Figure 8. Average rating value of the top-3 recommended items using the CS user similarity metric and setting K = 200.

The proposed algorithm achieves an average enhancement in the real rating recommendation value of 5.6% (from 4.23/5 to 4.46/5). Examining each dataset individually, we can see that, when the presented algorithm is applied to the MovieLens 1 M dataset, where the lowest initial average rating of the recommendations is noticed (due to the highest rating prediction error—both in terms of MAE and RMSE) at 3.8/5, this average is enhanced to 4.27/5. At the other end, when the presented algorithm is applied to the Amazon Digital Music dataset (which is the dataset with the lowest rating prediction error), with a 4.8/5 initial average rating of the recommendations, it is also enhanced to 4.85/5.

Figure 9 illustrates the average recommendation precision value of the top-3 items recommended before and after the application of the proposed algorithm in all six datasets, using the top-k technique with K = 200 for the NN selection and using the CS similarity metric. The proposed algorithm achieves an average enhancement in the recommendation precision value of 9.8% (from 80.6% to 88.5%). Examining each dataset individually, we can see that when the proposed algorithm is applied to the MovieLens 1 M dataset, where the lowest initial recommendation precision is noticed, at 66.1% (which practically means that, on average, the users would dislike one out of three recommendations made to them), this

average is enhanced to 82.9%. At the other end, when the proposed algorithm is applied to the Amazon Digital Music dataset, with 96% recommendation precision, which, again, is almost perfect, it also improves, achieving a result of 97.4%.

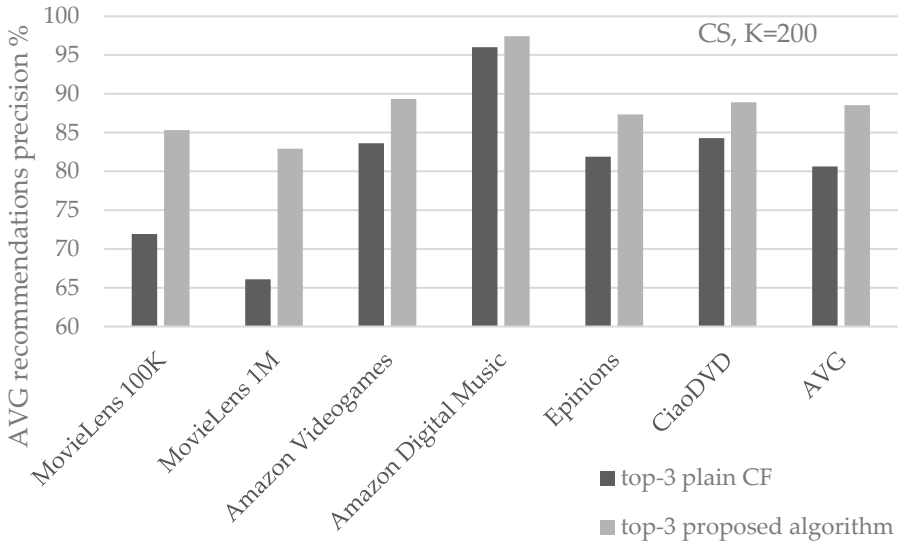

Figure 9. Average precision value of the top-3 recommended items using the CS user similarity metric and setting K = 200.

Figure 10 illustrates the average NDCG value of the top-3 items recommended before and after the application of the proposed algorithm in all six datasets, using the top-k technique with K = 200 for the NN selection and using the CS similarity metric. The proposed algorithm achieves an increase in the NDCG in all datasets tested, while its average value is increased from 0.969 to 0.979.

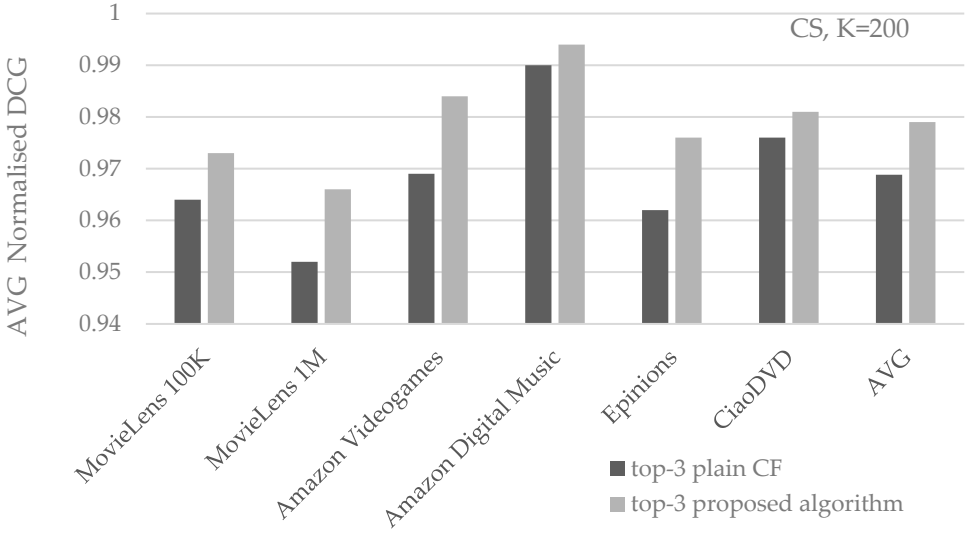

Figure 10. Average NDCG value of the top-3 recommended items using the CS user similarity metric and setting K = 200.

When we increase the number of recommended items to five (the top-5 recommended items), similar results are observed, where the average real rating recommendation value is enhanced by 5.4% (from 4.24 to 4.47). Regarding the average recommendation precision value, it is also enhanced by 9.4% (from 81.2% to 88.9%), while the average NDCG value is increased from 0.962 to 0.979.

When we increase the number of NNs taking part in the rating prediction formulation to K = 500, similar results are also observed for both the top-3 recommendations and the top-5 recommendations per user. More specifically, in the first case (top-3 recommendations), the proposed algorithm achieves an average enhancement in the real rating recommendation value of 4.8% (from 4.27/5 to 4.47/5), an average enhancement in the recommendation precision value of 8.2% (from 81.8% to 88.5%), and an average NDCG value increase from 0.969 to 0.978. In the second case (top-5 recommendations), the proposed algorithm achieves an average enhancement in the real rating recommendation value of 4.3% (from 4.29/5 to 4.48/5), an average enhancement in the recommendation precision value of 7.5% (from 82.7% to 88.9%), and an average NDCG value increase from 0.963 to 0.975.

Figure 11 illustrates the average (real) rating value of the top-3 items recommended, before and after the application of the proposed algorithm in all six datasets, using the correlation threshold technique with THR = 0.5.

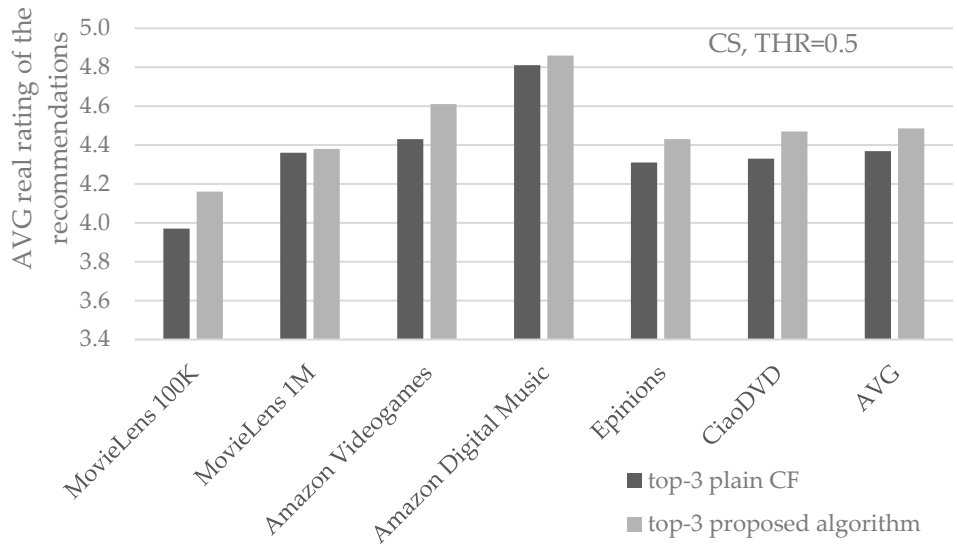

Figure 11. Average rating value of the top-3 recommended items using the CS user similarity metric and setting THR = 0.5.

The presented algorithm achieves an average enhancement in the real rating recommendation value of 2.7% (from 4.37/5 to 4.49/5). We can see again that, even when the proposed algorithm is applied to a dataset with a 4.81/5 initial average rating of the recommendations (the Amazon Digital Music), it achieves an increase to 4.86/5.

Figure 12 illustrates the average recommendation precision value of the top-3 items recommended before and after the application of the proposed algorithm in all six datasets, using the correlation threshold technique with THR = 0.5 for the NN selection. The presented algorithm achieves an average enhancement in the recommendation precision value of 4.1% (from 85.4% to 88.8%). Examining each dataset individually, we can see that, when the proposed algorithm is applied to the MovieLens 100 K dataset, where the lowest initial recommendation precision is noticed at 76.4% (which practically means that, on average, the users would dislike almost one out of four recommendations made to them), this average is enhanced to 82.5%. At the other end, when the presented algorithm is applied to the Amazon Digital Music dataset, with 96.1% recommendation precision, which, again, is almost perfect, it also achieves an increase to 97.5%.

Figure 13 illustrates the average NDCG value of the top-3 items recommended before and after the application of the proposed algorithm in all six datasets, using the correlation threshold technique with THR = 0.5 for the NN selection and using the CS similarity metric.

The proposed algorithm achieves an increase in the normalized DCG in all datasets tested, while its average value increases from 0.974 to 0.98.

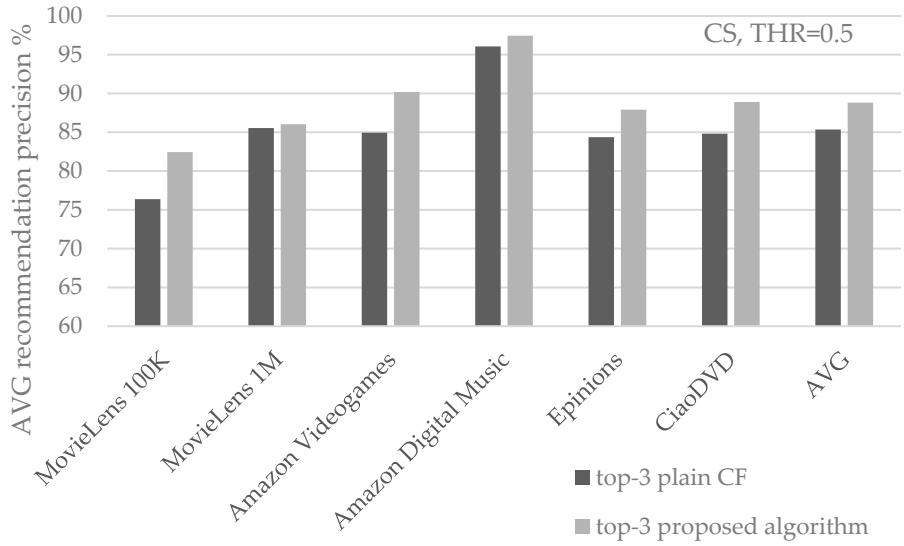

Figure 12. Average precision value of the top-3 recommended items using the CS user similarity metric and setting THR = 0.5.

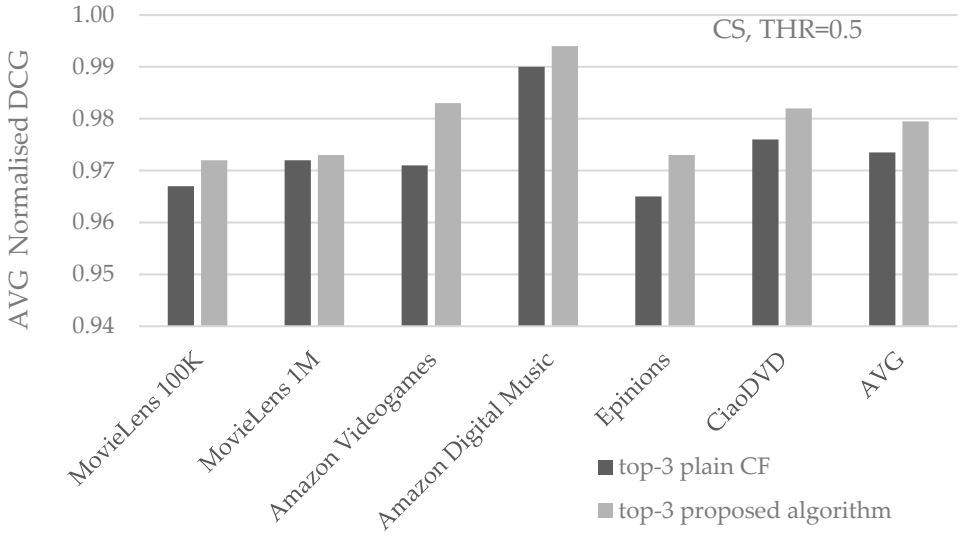

Figure 13. The average NDCG value of the top-3 recommended items when THR = 0.5 and using the CS.

When we increase the number of recommended items to five (the top-5 recommended items), similar results are observed, where the average real rating recommendation value is enhanced by 2.5%, the average recommendation precision value is enhanced by 4.2%, and the average NDCG value increases from 0.968 to 0.976.

Lastly, when we lower the similarity threshold of the NNs taking part in the rating prediction formulation to THR = 0.0, similar results are observed for both the top-3 recommendations and the top-5 recommendations per user. More specifically, the proposed algorithm achieves an average enhancement in the real rating recommendation value of 2.7% and an average enhancement in the recommendation precision value of 4.1%. Regarding the average NDCG value, it increases from 0.974 to 0.980 and from 0.968 to 0.976 for the top-3 and the top-5 recommendations per user, respectively.

4.3. Execution Efficiency

In this subsection, we discuss the execution efficiency of the presented algorithm, focusing on the overheads introduced, compared to the typical RecSys pipelines. In particular, the overheads introduced are as follows:

1. Considering the preprocessing step (where, for each item, the average rating value given by all users who have rated it is computed), in some cases, this procedure has already been executed, since some similarity metrics incorporate this value (e.g., the adjusted cosine measure). Therefore, in these cases, no additional overhead is incurred. Nevertheless, even in cases where this step is actually executed, the additional load introduced due to this preprocessing step is negligible, since it requires only one scan of the rating matrix. In terms of complexity, the maximum additional overhead is O(#ratings).
2. For the second additional step (where the number of the NNs that took part in the prediction is computed), the additional load of this action is practically zero, since the number of contributed NNs is determined in all cases during the rating prediction computation process.
3. Lastly, the overhead introduced by the third additional step (where the certainty of a rating prediction is assessed based on the features listed in Section 3.2) is negligible, since the assessment consists of simple comparisons. Notably, the filtering of items for which the prediction does not satisfy the certainty criteria is performed before the sorting of the candidate item recommendation list; therefore, the item list that is finally sorted to compute the final recommendation typically has a lower cardinality than the full list, resulting in execution time savings.

According to the analysis presented above, the maximum additional overhead introduced by the proposed algorithm is $AC = O(\#ratings)$. Considering the fact that the complexity of optimized similarity computation between a single pair of users u and v is $O(|I_u| + |I_v|)$, where I_u and I_v are the number of ratings entered by users u and v, respectively [57], the overall complexity of the similarity computation phase *is* $CSC = O(\#users^2) \times O(\#ratings\_per\_user) = O(\#users^2 \times \#ratings)$. Since $\#users \times \#ratings\_per\_user = \#ratings$, $CSC = O(\#users \times \#ratings) = O(\#users) \times AC$. Therefore, especially for large datasets where the number of users is high, the maximum additional complexity introduced by the proposed algorithm will be much smaller than the complexity of the similarity computation phase alone. Taking into account that the complete recommendation generation workflow includes, besides similarity computation, (a) rating prediction computation and (b) recommendation formulation, we can conclude that the additional overhead introduced by the proposed algorithm is practically very small to negligible.

To further quantify the overhead incurred due to the application of the proposed algorithm, we measured the execution time for the experiments described in Section 4.2 for all of the selected datasets. These measurements are depicted in Table 2; the datasets selected are MovieLens 1 M (the largest dense dataset), Amazon Digital Music (the smallest sparse dataset), and Epinions (the largest sparse datasets). The measurements concur with the theoretical analysis presented above, with the relative overhead in sparse datasets being reduced as the dataset size increases. The measurements also show that the overhead for dense datasets is also limited (0.3% for the MovieLens 1 M dataset).

Table 2. Experiment execution time for the baseline algorithm vs. the proposed algorithm.

Dataset Name	Baseline Algorithm Execution Time (sec)	Proposed Algorithm Execution Time (sec)	Overhead (sec)	Overhead (%)
Amazon Digital Music (sparse)	20.2	20.7	0.5	2.5%
Epinions (sparse)	3188	3192	4	0.13%
MovieLens 1 M (dense)	1406	1410	4	0.3%

Regarding the space efficiency of the algorithm, the additional data that need to be stored are as follows:

1. The storage of the items' average ratings requires an additional space of one real number per item;
2. The storage of the user's average ratings requires an additional space of one real number per user;
3. For the number of NNs that took part in the prediction, the algorithm needs to allocate one integer per prediction. Considering that the number of NNs that contributed to the formulation is typically small, one byte per prediction suffices. For cases where the number of NNs exceeds 255 (and therefore cannot be stored in one byte) storing the value of 255 suffices, since it corresponds to very high accuracy.

Overall, the space required is deemed small and easily manageable by modern computer systems.

5. Discussion of the Results

The experimental evaluation results presented in the previous section substantiate that the proposed algorithm effectively enhances recommendation quality in CF, as assessed in terms of the real rating values of the recommended items as well as their precision and their NDCG. More specifically, if we apply the plain CF algorithm (without using the proposed enhancements), the average rating value of the recommended items to the users is found in the range of [3.8, 4.8], while with the application of the presented algorithm, the range is enhanced to [4.2, 4.85]. In regard to the recommendation precision, the average of the plain CF algorithm is found in the range of [66%, 96%], while with the application of the presented algorithm, the range is enhanced to [82%, 97.4%].

The two most notable cases in our experiments are as follows:

- The case of the MovieLens 1 M dataset when using the KNN technique (in both similarity metrics): The use of the plain CF algorithm produced mediocre results since the average rating value of the recommended items was measured at 3.8/5 and the average recommendation precision was measured at 66%. This practically means that the users would dislike one out of three recommendations made to them, a case that clearly jeopardizes the credibility of the RecSys. However, when the proposed algorithm was applied, the average rating value of the recommended items was enhanced to 4.3, while the average recommendation precision was also enhanced to 83%, practically restoring the reliability of the RecSys.
- The case of the Amazon Digital_Music dataset: In this dataset, both the RMSE and the MAE of the plain CF algorithm were found to be very low, and hence, in most of the cases, the RecSys recommended the items with the highest numeric values. As a result, both the mean rating value of the recommended items and the mean recommendation precision were measured to be extremely high, at 4.8/5 and 96%, respectively, leaving practically no room for improvement for the proposed algorithm. However, even in this case, the proposed algorithm managed to enhance both of these metrics to 4.85/5 and 97.4%, respectively.

Overall, we can observe that the proposed algorithm not only protects the reliability of the RecSys from low recommendation precision, which occurs in some cases, but also achieves results of very high accuracy for the majority of the cases examined (the mean precision value of all cases was found to be $\geq$82%).

The algorithm proposed in this paper exploits the notion of certainty in rating predictions to achieve better precision in recommendations, with the improvements ranging from 1.4% to 24%. Notably, this is realized while maintaining a high level of coverage, with the losses in coverage due to the exclusion of non-certain predictions ranging from 2.6% to 15%. Similar approaches have been undertaken in the literature for a number of recommendation formulation algorithms, both in the CF domain (e.g., user rating profile (URP) [58] and SVD++ [59]) and using other recommender system technologies, such

as matrix factorization (e.g., Biased MF [60], Bayesian non-negative matrix factorization (BNMF) [61], non-negative matrix factorization (NMF) [62], and probabilistic matrix factorization (PMF) [63]).

In [64] the Bernoulli Matrix Factorization (BeMF) algorithm for rating prediction computation is introduced, along with a method for assessing rating certainty for predictions computed using the BeMF algorithm. Afterwards, in the recommendation generation phase, ratings with low certainty levels are excluded from the recommendation in order to improve rating precision. The performance of the proposed rating precision enhancement algorithm is evaluated against three datasets, one dense (MovieLens) and two sparse ones (FilmTrust and MyAnimeList), and is compared to the performance of rating precision algorithms that are based on the aforementioned rating prediction algorithms (URP, SVD++, Biased MF, NMF, and PMF). According to the results of these evaluations, the CF-based methods exhibit only marginal precision gains when the proposed methods exploiting rating prediction certainty are applied, with the improvement in recommendation precision being less than 0.4% in all cases. Considering the algorithms from the matrix factorization domain, Biased MF, NMF, and PMF exhibit similar behavior to the CF-based algorithms, i.e., marginal recommendation precision improvements are reaped, with the benefits being up to 0.35% in all cases. BeMF and BNMF, on the other hand, achieve higher improvements in precision, albeit with substantial drops in recall. For the cases where recall losses do not exceed 20%, which can be deemed acceptable to tolerable, the corresponding gains in precision range from 1.6 to 7.1% (average improvement: 4.3% for BeMF; 3.8% for BNMF).

Considering the above, the algorithm proposed in this paper clearly surpasses the performance of similar algorithms in the CF domain, while also achieving higher improvements than the respective algorithms in the matrix factorization domain. In our future work, we will also consider the adaptation of the proposed algorithm to work with rating prediction algorithms in the matrix factorization and machine learning domains.

Bias is a major issue in RecSys and needs to be considered and addressed in rating prediction and recommendation formulation algorithms. Considering the core aspects of the algorithm proposed in our paper, these are not directly affected by the bias problem since the criteria used to distinguish certain predictions from non-certain ones are (a) the number of near neighbors contributing to the formulation of a prediction and (b) the average rating of an item. The number of neighbors of a user is not affected by differences in rating practices, since the two similarity measures used in the paper effectively tackle these diversities. More specifically, the CS metric measures the angle between the rating vectors, not their magnitude [65,66], and the PCC measure subtracts the mean of each user's rating from the corresponding user's rating values [30]. Regarding the average rating of an item, this is computed over the ratings entered by all users. Therefore, differences in rating practices are smoothed by the fact that the set of users that rate each item is expected to consist of users with both strict and lenient rating habits. However, there may be cases where the set of users that have rated an item is skewed, containing a significantly higher number of users with strict or lenient rating practices. This may lead to the item being suppressed or overrepresented in recommendations, respectively. This issue will be considered in our future work.

6. Conclusions and Future Work

In this work, we presented a novel rating-based algorithm for recommendation formulation that aims to produce more accurate and useful recommendations. The proposed algorithm exploits the concept of rating prediction certainty, derived from features that have proven to be associated with rating prediction accuracy in CF [5,6], arranging so that rating predictions of low certainty are not utilized in the stage of recommendation formulation.

More specifically, a high-value rating prediction that has either (1) a low number of NNs taking part in the rating prediction formulation or (2) the average value of the item

for which the prediction is being formulated is not close to the limits of the rating range is deemed uncertain, and, hence, it is not forwarded as a recommendation to the user.

The presented algorithm was experimentally validated through a set of experiments using six rating datasets from multiple sources, two user similarity metrics, and two NN selection techniques that are all widely used in CF research to ensure the generalizability, applicability, and reliability of the results. The evaluation results have shown significant improvement in recommendation accuracy, enhancing the quality of the generated recommendations. This is verified by quantifying the quality using three widely used metrics, namely (i) the average rating value of the recommended items, (ii) the precision of the recommended items, and (iii) the normalized discounted cumulative gain of the recommended items. The results demonstrate that the algorithm achieves performance gains in all item domains (the experiments cover a wide range of item domain datasets, from music and videogames to movies) and across a variety of dataset features, such as size, number of users and items, density, etc. Furthermore, the performance gains are achieved under (a) both similarity metrics and (b) both NN selection methods and under all relevant parameter settings (similarity thresholds or number of retained NNs).

The proposed algorithm proved to successfully handle cases where the use of the plain CF algorithm exhibited low recommendation precision (65–70%), which would jeopardize the reliability of the RecSys (since the users would actually dislike one out of three recommendations made to them), while at the same time being able to improve cases with very high initial precision (95–97%), albeit in the latter case improvements were limited, since the performance enhancement margins were very small.

An identified limitation of this work is that, since the proposed algorithm practically excludes item predictions from becoming item recommendations, the cases where a recommendation can be produced based solely on CF are reduced. This problem is minor in relatively dense datasets (e.g., in the MovieLens 1 M dataset, the initial coverage is 97%, and, after the application of the proposed algorithm, it drops to 94–95%). However, in sparse datasets, this problem is more serious (e.g., in the Epinions dataset, the initial coverage is 76%, and, after the application of the proposed algorithm, it drops to 64–65%). In both cases, mitigation methods for this issue can be used, e.g., recommend an item that is widely accepted (an item with a very high rating average) or try to use complementary sources of information, such as item categories (e.g., if user X seems to like thrillers, suggest a thriller to him), etc. In general, research output, focusing on (1) low density and coverage CF cases [67–69], (2) cold start CF issues [70–72], and (3) hybrid CF techniques [73–75] can be easily used to overcome this limitation. Some preliminary tests have been conducted using the algorithm for density enrichment of sparse datasets introduced in [76], which introduces and exploits the concept of robust ratings to reduce the dataset sparsity: the Epinions dataset (which demonstrated the highest coverage drop) was densified using the algorithm proposed in [76], and then the algorithm proposed in this paper was applied on the densified dataset. Under this setting, the proposed algorithm achieved coverage equal to 75%, almost fully compensating the coverage loss incurred when the proposed algorithm was applied to the original (non-densified) Epinions dataset. On the other hand, the application of the proposed algorithm to the densified dataset limited the accuracy gains initially achieved. For instance, while the average precision value of the top-3 recommended items using the PCC user similarity metric and setting THR = 0.5, the gains for precision dropped from 4.01% to 3.13%, yet the precision gains still remain considerable.

Another method to tackle the reduction in coverage would be to modify the proposed algorithm so as not to exclude predictions of low certainty from becoming recommendations but rather consider them as fallback options in the absence of predictions with high certainty. Our preliminary experiments with the CiaoDVD dataset indicate that this variation of the algorithm, when applied in the scenario of using the top-3 recommended items using the PCC user similarity metric and setting K = 200, fully restored the coverage of the baseline algorithm (as expected), limiting, however, the precision gains from 5.3% to 4.26%; still, the gains in precision remain substantial. The interplay between coverage loss and

accuracy gains under different scenarios and the combination of the proposed algorithm with algorithms on coverage increase that have been proposed in the literature will be further explored in our future work.

The proposed algorithm exploits traits from the CF recommendation process to assess the certainty of predictions and utilizes these traits thereafter in the recommendation formulation stage. This rationale can also be applied to non-CF based RecSys methods. For instance, works concerning the explainability of graph convolutional networks identify traits of recommendations that explain why each item was recommended, correlating recommendations to input/training data. In particular, ref. [77] quantifies the contributions and importance of nodes by means of their gradient square values, while ref. [78] establishes a mapping between the final GCN layer and input nodes, generating an estimate of each node's importance for the result. In our future work, we plan to investigate how these traits can be used to assess the certainty of individual item rating predictions and recommendations and how these assessments can be used to increase recommendation accuracy.

Besides the issues discussed above, our future work will focus on exploring more features and their association with prediction accuracy, as well as using them in the recommendation production phase. Lastly, we are planning to include basic complementary sources of information, which can be available in the majority of the RecSys, such as basic user demographics (i.e., origin and gender) and item categories (such as types of movies, etc.). The use of ranking-based approaches will also be considered.

Author Contributions: Conceptualization, D.M., K.S., D.S. and C.V.; methodology, D.M., K.S., D.S. and C.V.; software, D.M., K.S., D.S. and C.V.; validation, D.M., K.S., D.S. and C.V.; formal analysis, D.M., K.S., D.S. and C.V.; investigation, D.M., K.S., D.S. and C.V.; resources, D.M., K.S., D.S. and C.V.; data curation, D.M., K.S., D.S. and C.V.; writing—original draft preparation, D.M., K.S., D.S. and C.V.; writing—review and editing, D.M., K.S., D.S. and C.V.; visualization, D.M., K.S., D.S. and C.V.; supervision, D.M., K.S., D.S. and C.V.; project administration, D.M., K.S., 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 analyzed in this study. These data can be found here: https://grouplens.org/datasets/movielens/ (accessed on 20 March 2024), https://guoguibing.github.io/librec/datasets.html (accessed on 20 March 2024), https://www.kaggle.com/datasets/masoud3/epinions-trust-network (accessed on 20 March 2024), and https://nijianmo.github.io/amazon/index.html (accessed on 20 March 2024).

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

References

1. Ramakrishnan, G.; Saicharan, V.; Chandrasekaran, K.; Rathnamma, M.V.; Ramana, V.V. Collaborative Filtering for Book Recommendation System. In *Soft Computing for Problem Solving*; Das, K.N., Bansal, J.C., Deep, K., Nagar, A.K., Pathipooranam, P., Naidu, R.C., Eds.; Advances in Intelligent Systems and Computing; Springer: Singapore, 2020; Volume 1057, pp. 325–338. ISBN 9789811501838.
2. Wang, F.; Zhu, H.; Srivastava, G.; Li, S.; Khosravi, M.R.; Qi, L. Robust Collaborative Filtering Recommendation With User-Item-Trust Records. *IEEE Trans. Comput. Soc. Syst.* **2022**, *9*, 986–996. [CrossRef]
3. 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]
4. 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]
5. Margaris, D.; Vassilakis, C.; Spiliotopoulos, D. On Producing Accurate Rating Predictions in Sparse Collaborative Filtering Datasets. *Information* **2022**, *13*, 302. [CrossRef]
6. Spiliotopoulos, D.; Margaris, D.; Vassilakis, C. On Exploiting Rating Prediction Accuracy Features in Dense Collaborative Filtering Datasets. *Information* **2022**, *13*, 428. [CrossRef]

7. Feng, L.; Zhao, Q.; Zhou, C. Improving Performances of Top-N Recommendations with Co-Clustering Method. *Expert Syst. Appl.* **2020**, *143*, 113078. [CrossRef]
8. Singh, P.K.; Setta, S.; Pramanik, P.K.D.; Choudhury, P. Improving the Accuracy of Collaborative Filtering-Based Recommendations by Considering the Temporal Variance of Top-N Neighbors. In *International Conference on Innovative Computing and Communications*; Khanna, A., Gupta, D., Bhattacharyya, S., Snasel, V., Platos, J., Hassanien, A.E., Eds.; Advances in Intelligent Systems and Computing; Springer: Singapore, 2020; Volume 1087, pp. 1–10. ISBN 9789811512858.
9. Chen, S.-H.; Sou, S.-I.; Hsieh, H.-P. Top-N Music Recommendation Framework for Precision and Novelty under Diversity Group Size and Similarity. *J. Intell. Inf. Syst.* **2024**, *62*, 1–26. [CrossRef]
10. Gienapp, L.; Stein, B.; Hagen, M.; Potthast, M. Estimating Topic Difficulty Using Normalized Discounted Cumulated Gain. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, Ireland, 19–23 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 2033–2036.
11. Jayashree, R.; Christy, A. Improving the Enhanced Recommended System Using Bayesian Approximation Method and Normalized Discounted Cumulative Gain. *Procedia Comput. Sci.* **2015**, *50*, 216–222. [CrossRef]
12. Yang, B.; Lei, Y.; Liu, J.; Li, W. Social Collaborative Filtering by Trust. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1633–1647. [CrossRef]
13. Wang, Y.; Deng, J.; Gao, J.; Zhang, P. A Hybrid User Similarity Model for Collaborative Filtering. *Inf. Sci.* **2017**, *418–419*, 102–118. [CrossRef]
14. Xin, X.; He, X.; Zhang, Y.; Zhang, Y.; Jose, J. Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 18 July 2019; ACM: New York, NY, USA, 2019; pp. 125–134.
15. 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]
16. Zhang, Y.; Yin, C.; Wu, Q.; He, Q.; Zhu, H. Location-Aware Deep Collaborative Filtering for Service Recommendation. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 3796–3807. [CrossRef]
17. Alhijawi, B.; Kilani, Y. A Collaborative Filtering Recommender System Using Genetic Algorithm. *Inf. Process. Manag.* **2020**, *57*, 102310. [CrossRef]
18. Nilashi, M.; Ibrahim, O.; Bagherifard, K. A Recommender System Based on Collaborative Filtering Using Ontology and Dimensionality Reduction Techniques. *Expert Syst. Appl.* **2018**, *92*, 507–520. [CrossRef]
19. Liao, C.-L.; Lee, S.-J. A Clustering Based Approach to Improving the Efficiency of Collaborative Filtering Recommendation. *Electron. Commer. Res. Appl.* **2016**, *18*, 1–9. [CrossRef]
20. Chen, Y.-C.; Hui, L.; Thaipisutikul, T. A Collaborative Filtering Recommendation System with Dynamic Time Decay. *J. Supercomput.* **2021**, *77*, 244–262. [CrossRef]
21. Margaris, D.; Vassilakis, C. Improving Collaborative Filtering's Rating Prediction Quality in Dense Datasets, by Pruning Old Ratings. In Proceedings of the Proceedings—IEEE Symposium on Computers and Communications, Heraklion, Greece, 3–6 July 2017; pp. 1168–1174.
22. Valdiviezo-Diaz, P.; Ortega, F.; Cobos, E.; Lara-Cabrera, R. A Collaborative Filtering Approach Based on Naïve Bayes Classifier. *IEEE Access* **2019**, *7*, 108581–108592. [CrossRef]
23. Neysiani, B.S.; Soltani, N.; Mofidi, R.; Nadimi-Shahraki, M.H. Improve Performance of Association Rule-Based Collaborative Filtering Recommendation Systems Using Genetic Algorithm. *Int. J. Inf. Technol. Comput. Sci.* **2019**, *11*, 48–55. [CrossRef]
24. Ren, L.; Wang, W. An SVM-Based Collaborative Filtering Approach for Top-N Web Services Recommendation. *Future Gener. Comput. Syst.* **2018**, *78*, 531–543. [CrossRef]
25. 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*; Satapathy, S.C., Joshi, A., Eds.; Smart Innovation, Systems and Technologies; Springer: Singapore, 2019; Volume 107, pp. 173–180. ISBN 9789811317460.
26. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, 25 July 2020; ACM: New York, NY, USA, 2020; pp. 639–648.
27. Liu, F.; Cheng, Z.; Zhu, L.; Gao, Z.; Nie, L. Interest-Aware Message-Passing GCN for Recommendation. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19 April 2021; ACM: New York, NY, USA, 2021; pp. 1296–1305.
28. Li, A.; Cheng, Z.; Liu, F.; Gao, Z.; Guan, W.; Peng, Y. Disentangled Graph Neural Networks for Session-Based Recommendation. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 7870–7882. [CrossRef]
29. Wei, Y.; Wang, X.; Li, Q.; Nie, L.; Li, Y.; Li, X.; Chua, T.-S. Contrastive Learning for Cold-Start Recommendation. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, China, 17 October 2021; ACM: New York, NY, USA, 2021; pp. 5382–5390.
30. Jain, G.; Mahara, T.; Tripathi, K.N. A Survey of Similarity Measures for Collaborative Filtering-Based Recommender System. In *Soft Computing: Theories and Applications*; Pant, M., Sharma, T.K., Verma, O.P., Singla, R., Sikander, A., Eds.; Advances in Intelligent Systems and Computing; Springer: Singapore, 2020; Volume 1053, pp. 343–352. ISBN 9789811507502.
31. Amer, A.A.; Abdalla, H.I.; Nguyen, L. Enhancing recommendation systems performance using highly-effective similarity measures. *Knowl.-Based Syst.* **2021**, *217*, 106842. [CrossRef]

32. Nguyen, L.V.; Vo, Q.-T.; Nguyen, T.-H. Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services. *Big Data Cogn. Comput.* **2023**, *7*, 106. [CrossRef]
33. Faruk, K.O.; Rahman, A.; Shusmita, S.A.; Awlad, M.S.I.; Das, P.; Mehedi, M.H.K.; Iqbal, S.; Rasel, A.A. K Nearest Neighbour Collaborative Filtering for Expertise Recommendation Systems. In *Distributed Computing and Artificial Intelligence, 19th International Conference*; Omatu, S., Mehmood, R., Sitek, P., Cicerone, S., Rodríguez, S., Eds.; Lecture Notes in Networks and Systems; Springer International Publishing: Cham, Switzerland, 2023; Volume 583, pp. 187–196. ISBN 978-3-031-20858-4.
34. Houshmand Nanehkaran, F.; Lajevardi, S.M.; Mahlouji Bidgholi, M. Nearest Neighbors Algorithm and Genetic-based Collaborative Filtering. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6538. [CrossRef]
35. Margaris, D.; Vassilakis, C.; Spiliotopoulos, D.; Ougiaroglou, S. Rating Prediction Quality Enhancement in Low-Density Collaborative Filtering Datasets. *Big Data Cogn. Comput.* **2023**, *7*, 59. [CrossRef]
36. Fkih, F. Similarity Measures for Collaborative Filtering-Based Recommender Systems: Review and Experimental Comparison. *J. King Saud Univ.—Comput. Inf. Sci.* **2022**, *34*, 7645–7669. [CrossRef]
37. Veras De Sena Rosa, R.E.; Guimaraes, F.A.S.; da Silva Mendonca, R.; de Lucena, V.F. Improving Prediction Accuracy in Neighborhood-Based Collaborative Filtering by Using Local Similarity. *IEEE Access* **2020**, *8*, 142795–142809. [CrossRef]
38. Fkih, F. Enhancing Item-Based Collaborative Filtering by Users' Similarities Injection and Low-Quality Data Handling. *Data Knowl. Eng.* **2023**, *144*, 102126. [CrossRef]
39. Cremonesi, P.; Elahi, M.; Garzotto, F. User Interface Patterns in Recommendation-Empowered Content Intensive Multimedia Applications. *Multimed. Tools Appl.* **2017**, *76*, 5275–5309. [CrossRef]
40. Du, F.; Plaisant, C.; Spring, N.; Shneiderman, B. Visual Interfaces for Recommendation Systems: Finding Similar and Dissimilar Peers. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 1–23. [CrossRef]
41. Dominguez, V.; Donoso-Guzmán, I.; Messina, P.; Parra, D. Algorithmic and HCI Aspects for Explaining Recommendations of Artistic Images. *ACM Trans. Interact. Intell. Syst.* **2020**, *10*, 1–31. [CrossRef]
42. Yin, F. Sparsity-Tolerated Algorithm with Missing Value Recovering in User-Based Collaborative Filtering Recommendation. *J. Inf. Comput. Sci.* **2013**, *10*, 4939–4948. [CrossRef]
43. Zeng, C.; Xing, C.-X.; Zhou, L.-Z.; Zheng, X.-H. Similarity Measure and Instance Selection for Collaborative Filtering. *Int. J. Electron. Commer.* **2004**, *8*, 115–129. [CrossRef]
44. Afoudi, Y.; Lazaar, M.; Al Achhab, M. Hybrid Recommendation System Combined Content-Based Filtering and Collaborative Prediction Using Artificial Neural Network. *Simul. Model. Pract. Theory* **2021**, *113*, 102375. [CrossRef]
45. Guo, S.; Wang, Y.; Yuan, H.; Huang, Z.; Chen, J.; Wang, X. TAERT: Triple-Attentional Explainable Recommendation with Temporal Convolutional Network. *Inf. Sci.* **2021**, *567*, 185–200. [CrossRef]
46. Zhu, N.; Cao, J.; Lu, X.; Gu, Q. Leveraging Pointwise Prediction with Learning to Rank for Top-N Recommendation. *World Wide Web* **2021**, *24*, 375–396. [CrossRef]
47. Kaya, T.; Kaleli, C. A Novel Top-n Recommendation Method for Multi-Criteria Collaborative Filtering. *Expert Syst. Appl.* **2022**, *198*, 116695. [CrossRef]
48. Safarov, F.; Kutlimuratov, A.; Abdusalomov, A.B.; Nasimov, R.; Cho, Y.-I. Deep Learning Recommendations of E-Education Based on Clustering and Sequence. *Electronics* **2023**, *12*, 809. [CrossRef]
49. AlEroud, A.; Karabatis, G. Using Contextual Information to Identify Cyber-Attacks. In *Information Fusion for Cyber-Security Analytics*; Studies in Computational Intelligence; Alsmadi, I., Karabatis, G., Aleroud, A., Eds.; Springer: Cham, Switzerland, 2017; Volume 691, pp. 1–16.
50. Felfernig, A.; Boratto, L.; Stettinger, M.; Tkalčič, M. Evaluating Group Recommender Systems. In *Group Recommender Systems. SpringerBriefs in Electrical and Computer Engineering*; Springer: Cham, Switzerland, 2018; pp. 59–71.
51. Margaris, D.; Vassilakis, C.; Spiliotopoulos, D. What Makes a Review a Reliable Rating in Recommender Systems? *Inf. Process. Manag.* **2020**, *57*, 102304. [CrossRef]
52. Lee, Y.-C. Application of Support Vector Machines to Corporate Credit Rating Prediction. *Expert Syst. Appl.* **2007**, *33*, 67–74. [CrossRef]
53. Li, X.; Zhao, H.; Wang, Z.; Yu, Z. Research on Movie Rating Prediction Algorithms. In Proceedings of the 2020 5th IEEE International Conference on Big Data Analytics (ICBDA), Xiamen, China, 8–11 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 121–125.
54. Dai, Z.; Yuchen, Z.; Li, A.; Qian, G. The Application of Machine Learning in Bank Credit Rating Prediction and Risk Assessment. In Proceedings of the 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Nanchang, China, 26 March 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 986–989.
55. Arther Sandag, G.; Gara, F. Irfan Android Application Market Prediction Based on User Ratings Using KNN. In Proceedings of the 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS), Manado, Indonesia, 27 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
56. Chowdhury, R.; Zaman, F.U.; Sharker, A.; Rahman, M.; Shah, F.M. Rate Insight: A Comparative Study on Different Machine Learning and Deep Learning Approaches for Product Review Rating Prediction in Bengali Language. In Proceedings of the 2022 25th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 17 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 406–411.

57. Zhang, F.; Gong, T.; Lee, V.E.; Zhao, G.; Rong, C.; Qu, G. Fast Algorithms to Evaluate Collaborative Filtering Recommender Systems. *Knowl.-Based Syst.* **2016**, *96*, 96–103. [CrossRef]
58. Sánchez, P.; Bellogín, A. Building User Profiles Based on Sequences for Content and Collaborative Filtering. *Inf. Process. Manag.* **2019**, *56*, 192–211. [CrossRef]
59. Anwar, T.; Uma, V.; Srivastava, G. Rec-CFSVD++: Implementing Recommendation System Using Collaborative Filtering and Singular Value Decomposition (SVD)++. *Int. J. Inf. Technol. Decis. Mak.* **2021**, *20*, 1075–1093. [CrossRef]
60. Koren, Y.; Bell, R.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
61. Lumbreras, A.; Filstroff, L.; Févotte, C. Bayesian Mean-Parameterized Nonnegative Binary Matrix Factorization. *Data Min. Knowl. Discov.* **2020**, *34*, 1898–1935. [CrossRef]
62. Chen, W.-S.; Zeng, Q.; Pan, B. A Survey of Deep Nonnegative Matrix Factorization. *Neurocomputing* **2022**, *491*, 305–320. [CrossRef]
63. Xu, S.; Zhuang, H.; Sun, F.; Wang, S.; Wu, T.; Dong, J. Recommendation Algorithm of Probabilistic Matrix Factorization Based on Directed Trust. *Comput. Electr. Eng.* **2021**, *93*, 107206. [CrossRef]
64. Ortega, F.; Lara-Cabrera, R.; González-Prieto, Á.; Bobadilla, J. Providing Reliability in Recommender Systems through Bernoulli Matrix Factorization. *Inf. Sci.* **2021**, *553*, 110–128. [CrossRef]
65. Coscrato, V.; Bridge, D. Estimating and Evaluating the Uncertainty of Rating Predictions and Top-n Recommendations in Recommender Systems. *ACM Trans. Recomm. Syst.* **2023**, *1*, 1–34. [CrossRef]
66. Nguyen, L.V.; Hong, M.-S.; Jung, J.J.; Sohn, B.-S. Cognitive Similarity-Based Collaborative Filtering Recommendation System. *Appl. Sci.* **2020**, *10*, 4183. [CrossRef]
67. Feng, C.; Liang, J.; Song, P.; Wang, Z. A Fusion Collaborative Filtering Method for Sparse Data in Recommender Systems. *Inf. Sci.* **2020**, *521*, 365–379. [CrossRef]
68. Wang, Y.; Wang, P.; Liu, Z.; Zhang, L.Y. A New Item Similarity Based on α -Divergence for Collaborative Filtering in Sparse Data. *Expert Syst. Appl.* **2021**, *166*, 114074. [CrossRef]
69. Yusefi Hafshejani, Z.; Kaedi, M.; Fatemi, A. Improving Sparsity and New User Problems in Collaborative Filtering by Clustering the Personality Factors. *Electron. Commer. Res.* **2018**, *18*, 813–836. [CrossRef]
70. Herce-Zelaya, J.; Porcel, C.; Bernabé-Moreno, J.; Tejeda-Lorente, A.; Herrera-Viedma, E. New Technique to Alleviate the Cold Start Problem in Recommender Systems Using Information from Social Media and Random Decision Forests. *Inf. Sci.* **2020**, *536*, 156–170. [CrossRef]
71. Shao, Y.; Xie, Y. Research on Cold-Start Problem of Collaborative Filtering Algorithm. In Proceedings of the 2019 3rd International Conference on Big Data Research, Cergy-Pontoise, France, 20 November 2019; ACM: New York, NY, USA, 2019; pp. 67–71.
72. Zhang, Z.; Zhang, Y.; Ren, Y. Employing Neighborhood Reduction for Alleviating Sparsity and Cold Start Problems in User-Based Collaborative Filtering. *Inf. Retr. J.* **2020**, *23*, 449–472. [CrossRef]
73. Yang, X.; Zhou, S.; Cao, M. An Approach to Alleviate the Sparsity Problem of Hybrid Collaborative Filtering Based Recommendations: The Product-Attribute Perspective from User Reviews. *Mob. Netw. Appl.* **2020**, *25*, 376–390. [CrossRef]
74. 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]
75. Duricic, T.; Lacic, E.; Kowald, D.; Lex, E. Trust-Based Collaborative Filtering: Tackling the Cold Start Problem Using Regular Equivalence. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 27 September 2018; ACM: New York, NY, USA, 2018; pp. 446–450.
76. Margaris, 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]
77. Baldassarre, F.; Azizpour, H. Explainability Techniques for Graph Convolutional Networks. *arXiv* **2019**, arXiv:1905.13686.
78. Pope, P.E.; Kolouri, S.; Rostami, M.; Martin, C.E.; Hoffmann, H. Explainability Methods for Graph Convolutional Neural Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 10764–10773.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.