Improving Recommendation Quality in Collaborative Filtering by Including Prediction Confidence Factors

Kiriakos Sgardelis[1][a], Dionisis Margaris[1][b], Dimitris Spiliotopoulos[2][c], Costas Vassilakis[3][d] and Stefanos Ougiaroglou[4][e]

[1]Department of Digital Systems, University of the Peloponnese, Sparta, Greece
[2]Department of Management Science and Technology, University of the Peloponnese, Tripoli, Greece
[3]Department of Informatics and Telecommunications, University of the Peloponnese, Tripoli, Greece
[4]Department of Information and Electronic Engineering, International Hellenic University, Thessaloniki, Greece

Keywords: Collaborative Filtering, Rating Predictions, Recommender Systems, Certainty Factors, Algorithm.

Abstract: Collaborative filtering is a prevalent recommender system technique which generates rating predictions based on the rating values given by the users' near neighbours. Consequently, for each user, the items scoring the highest prediction values are recommended to them. Unfortunately, predictions inherently entail errors, which, in the case of recommender systems, manifest as unsuccessful recommendations. However, along with each rating prediction value, prediction confidence factors can be computed. As a result, items having low prediction confidence factor values, can be either declined for recommendation or have their recommendation priority demoted. In the former case, some users may receive fewer recommended items or even none, especially when using a sparse dataset. In this paper, we present an algorithm that determines the items to be recommended by considering both the rating prediction values and confidence factors of predictions, allowing for predictions with higher confidence factors to outrank predictions with higher value, but lower confidence. The presented algorithm achieves to enhance the recommendation quality, while at the same time retaining the number of recommendations for each user.

1 INTRODUCTION

Collaborative filtering (CF) is a prevalent technique of predicting rating values in recommender systems (RecSys). It is based on the numeric rating values that users close to the active user (i.e. his near neighbours - NNs) have given to the item (e.g. service, product, etc.) for which the prediction is being computed. Consequently, the items achieving the highest prediction values are suggested to the active user, since their acceptance is of very high probability. The nearer these numeric predictions are to the real numeric rating values, the more successful the RecSys is (Jain et al., 2023; Nguyen et al., 2023).

Let as assume that for a user U, there are two items candidate for recommendation, i_1 and i_2, where their respective CF rating prediction values have been computed at 4.8/5 and 4.6/5. Typically, a RecSys will recommend primarily item i_1 to U, under the premise that higher rating prediction value denotes higher probability that the user will like the item. Let us also assume that the prediction for item i_1 is deemed of low confidence (e.g., it has been computed based on a very low number of ratings, or by ratings contributed by users which have a relatively low degree of similarity to the user for which the recommendation is generated). On the other hand, the prediction for item i_2 is deemed of high confidence (e.g., it is based on 20 "close" NNs' ratings). In such a situation, it appears sensible to opt for recommending i_2 instead of i_1 since, while i_1 has a marginal advantage with regards to its rating prediction value, there is a high

[a] https://orcid.org/0009-0008-7113-8855
[b] https://orcid.org/0000-0002-7487-374X
[c] https://orcid.org/0000-0003-3646-1362
[d] https://orcid.org/0000-0001-9940-1821
[e] https://orcid.org/0000-0003-1094-2520

Sgardelis, K., Margaris, D., Spiliotopoulos, D., Vassilakis, C. and Ougiaroglou, S.
Improving Recommendation Quality in Collaborative Filtering by Including Prediction Confidence Factors.
DOI: 10.5220/0013052200003825

In *Proceedings of the 20th International Conference on Web Information Systems and Technologies (WEBIST 2024)*, pages 372-379
ISBN: 978-989-758-718-4; ISSN: 2184-3252

risk that this value is inaccurate, and hence the user may not actually like the recommended item. On the contrary, the recommendation of i_2 can be deemed to be “safe”. However, a typical RecSys recommends items by considering only rating prediction values.

Rating prediction confidence factors, associated with CF prediction accuracy, have been explored recently (Margaris et al., 2022; Spiliotopoulos et al., 2022). These research works demonstrated that (1) the number of NNs participating in the prediction computation, (2) the item’s mean ratings value and (3) the user’s mean ratings value, are related to rating prediction accuracy. Based on these findings, items having low prediction confidence factor values can be either declined for recommendation or have their recommendation priority demoted. In the former case, some users may receive fewer recommended items or even none, notably when using a sparse dataset.

In this paper, we present an algorithm that determines the items to be recommended by considering both the rating prediction values and confidence factors of predictions, allowing for predictions with higher confidence factors to outrank predictions with higher value, but lower confidence. The presented algorithm enhances recommendation quality, while at the same time retaining the number of recommendations for each user. The proposed algorithm is evaluated against 5 widely used CF datasets (including both dense and sparse, in order to cover all cases). As far as the NN selection is considered, both the top-k and the correlation threshold techniques are considered in the evaluation.

The remainder of the paper is structured as follows: in Section 2 we present the related work. In Sections 3 and 4 we summarize the foundations of confidence factors in CF rating prediction and present the proposed algorithm, respectively. In Section 5 we present the evaluation results and in Section 6 we conclude the paper and outline future work.

2 RELATED WORK

The quality of CF recommendations is a field of major research interest over the last years. The work in (Alhijawi et al., 2021) employs a genetic algorithm, in order to customize the prediction process, based on the active user’s set of NNs. This genetic algorithm enables the optimal solution discovery, without exhaustive analysis, since each person is represented with a vector. Furthermore, with the use of this algorithm, the active user’s search parallelism is a fast process, since the fitness function evaluation is totally independent for each user.

The work in (Y.-C. Chen et al., 2021) introduces a CF-based RecSys dynamic decay CF, which, based on the preference of users’ variations, it incorporates a time decay function. This work extends the human brain memory concept, to discover the users’ levels of interest. As a result, the dynamic decay CF algorithm adjusts the decay function based on different user interest levels. The work in (Z. Wang, 2023) presents a CF algorithm that targets to enhance the recommendation accuracy of tourist activities. This algorithm overcomes sparse data issues, by taking into account user preferences, as well as by using the Jeffries-Matusita vicinity metric (Sen et al., 2019). The work in (Bobadilla et al., 2023) introduces a Generative Adversarial Network-based algorithm, which parametrically produces CF datasets. More specifically, it allows the selection of users, items and samples number, as well as the dataset’s stochastic variability. Furthermore, the presented architecture incorporates a clustering method which transforms the dense produced samples into sparse and discrete ones, as well as a Deep Matrix Factorization model which exports the dense item and user embeddings.

The work in (Fareed et al., 2023) presents a CF RecSys framework which, in order to produce more pertinent and precise recommendations, it incorporates social network information. The presented framework is based on a user-based CF algorithm that estimates user vicinity values based on both their social relations and their item ratings. Furthermore, this vicinity metric is determined by synthesizing the two aforementioned factors, while the respective weights-importance are determined through an optimization step. The work in (Vuong Nguyen et al., 2021) introduces a hybrid RecSys algorithm which overcomes the issues of cold-start and data sparsity of the user ratings, by combining word embedding-based content analysis with CF methods. Been applied on the film domain, this work focuses on perceiving the gist of the movie plot, using word embedding techniques of the films’ features, such as genres, titles, actors, directors, etc.

The work in (R. Wang et al., 2022) introduces a time-aware CF algorithm with two phases, a dynamic user preference phase and a deep learning matching score prediction phase. During the first phase the time-aware attention mechanism models the short-term user preferences. In the second phase the user-item interactions are discovered by deep learning models. The results of the two aforementioned phases are combined for predicting the final score.

Still, the exploitation of the concept of rating prediction confidence factors for enhancing the recommendation quality in CF has not received considerable research attention. Recent works have explored rating prediction factors, based only on the basic CF information (the user-item-rating tuple), related with CF rating prediction accuracy. The works in (Margaris et al., 2022) and in (Spiliotopoulos et al., 2022) show that the NN number, the item's mean ratings value and the user's mean ratings value are related with rating prediction accuracy in CF, in sparse and dense datasets, respectively. The work in (Margaris et al., 2024) exploits these results to propose an algorithm that utilizes confidence factors in CF rating prediction, by eliminating rating predictions having low values of confidence factors from becoming recommendations. Although this algorithm results in a considerable recommendation quality upgrade, the recommendation coverage (i.e., the percentage of users that at least *N* recommendations can be formulated to them, where N is a given algorithm parameter) is significantly decreased, while in the cases of the algorithm's application on (very) sparse CF datasets (e.g., Amazon-sourced datasets), the algorithm becomes almost inapplicable.

The algorithm presented in this work is based only on the very basic CF information, while it also takes into account both the rating prediction and the confidence factors values in CF. However, instead of following the approach undertaken in (Margaris et al., 2024), i.e., pruning the recommendation candidate item list retaining only rating predictions with (very) high confidence, it determines the items to be recommended by considering both the rating prediction values and confidence factors of predictions, allowing for predictions with higher confidence factors to outrank predictions with higher value, but lower confidence. Hence, the presented algorithm achieves to both enhance the recommendation quality, while at the same time retain the number of recommendations for each user, and as a result it can be applied in every CF dataset, including both sparse and dense ones.

3 CONFIDENCE FACTORS IN CF RATING PREDICTION

Contemporary works have studied rating prediction factors related with CF prediction accuracy. More specifically, the works in (Margaris et al., 2022) and in (Spiliotopoulos et al., 2022) showed a positive association between CF rating prediction accuracy and the following factors:

(a) $F_{\#NN}$, which considers the number of NNs participating in the prediction computation,

(b) F_{Uavg}, which relates to average value of the user's ratings for whom the prediction is being computed, and

(c) F_{Iavg}, which considers the average value of the item's ratings for which the prediction is being computed.

Table 1 summarizes the thresholds of the aforementioned factors, that a prediction is classified (i) as a high accuracy one and (ii) as a very high accuracy one, both in sparse and dense datasets. Regarding the F_{Uavg} and the F_{Iavg} factors, we consider a 5-star rating scale evaluation. These criteria are exploited by the proposed algorithm to formulate recommendations. In the next section we present and analyze the proposed RecSys algorithm in detail.

Table 1: Thresholds of the CF prediction accuracy factors for classifying predictions.

Factor	High Accuracy	Very High Accuracy
$F_{\#NN}$	≥ 2 (sparse) / ≥ 6% (dense)	≥ 4 (sparse) ≥ 15% (dense)
F_{Uavg}	[1.0, 2.0] or [4.0, 5.0]	[1.0, 1.5] or [4.5, 5.0]
F_{Iavg}	[1.0, 2.0] or [4.0, 5.0]	[1.0, 1.5] or [4.5, 5.0]

4 THE PROPOSED ALGORITHM

As noted above, the algorithm proposed in this paper determines the items to be recommended by considering both the rating prediction values and confidence factors of predictions, allowing for predictions with higher confidence factors to outrank predictions with higher value, but lower confidence.

Considering the formulation of the initial recommendation candidate list (ICRL), in this paper, we adopt the approach followed by many works (Felfernig et al., 2018; Margaris et al., 2020; Trattner et al., 2024), where the items achieving a rating prediction value in the top 30% of the rating range (i.e., 3.5/5 for the 5-star rating scale) are considered eligible for recommendation to the users.

The proposed algorithm essentially redefines the step of ranking the items to be recommended in CF RecSys. More specifically, instead of simply ranking the items that pass the recommendation threshold (the top 30% of the rating range, as mentioned above) in descending order of their rating prediction value, the algorithm considers both the rating prediction value and the confidence estimation associated with the computation of this value. This is realized through the following steps (for simplicity, we assume a rating scale [1-5], as in the majority of the CF datasets, however the algorithm can be easily adapted to accommodate different rating scales):

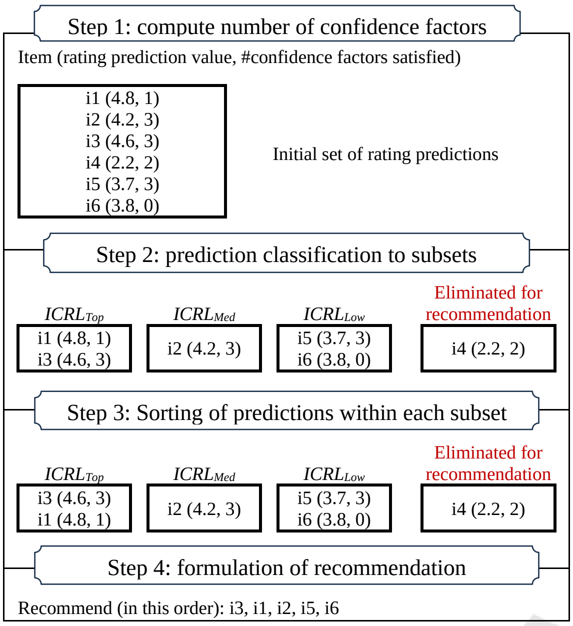

Figure 1: Example execution of the proposed algorithm.

- Step 1: The algorithm computes the number of confidence factors ($F_{\#NN}$, F_{Uavg} and F_{Iavg}) that are fulfilled by each prediction in the *ICRL*. More specifically, for each rating prediction *rp* in these subsets, it computes the associated confidence ranking score CRS_{rp} as follows:
  $CRS_{rp} = CRS_{rp,F\#NN} + CRS_{rp,FUavg} + CRS_{rp,FIavg}$
  where:
  $$CRS_{rp,F\#NN} = \begin{cases} 1 & if\ \#NNs(rp) \geq Thr \\ 0 & otherwise \end{cases}$$
  with *#NNs(rp)* denoting the number of NNs contributing to the computation of *rp*, and *Thr* being the dataset-dependent threshold for classifying *rp* as a high accuracy one, considering the $F_{\#NN}$ criterion (cf. section 3);
  $$CRS_{rp,FUavg} = \begin{cases} 1 & if\ (\overline{\overline{U}} \leq 1.5) \vee (\overline{\overline{U}} \geq 4.5) \\ 0 & otherwise \end{cases}$$
  with $\overline{\overline{U}}$ denoting the average ratings entered by the user for whom *rp* has been computed; and
  $$CRS_{rp,FIavg} = \begin{cases} 1 & if\ (\overline{\overline{I}} \leq 1.5) \vee (\overline{\overline{I}} \geq 4.5) \\ 0 & otherwise \end{cases}$$
  with $\overline{\overline{I}}$ denoting the average ratings entered for the item for which *rp* has been computed.
- Step 2: The algorithm partitions the set of the items to be recommended into subsets, with each subset covering a rating prediction range of 0.5 (or 10% of the rating scale). Effectively, the following subsets will be formulated:
  - $ICRL_{Top}$, which includes the items in the IRCL having rating prediction values in the range [4.5, 5]. This list contains the items for which we can assume that the user will "definitely" like them;
  - $ICRL_{Med}$, which includes the items in the IRCL having rating prediction values in the range [4, 4.5). This list contains the items for which we can assume that the user will "most probably" like them; and
  - $ICRL_{Low}$, which includes the items in the IRCL having rating prediction values in the range [3.5, 4). This list contains the items for which we can assume that the user will "probably" like them.
- Step 3: The algorithm sorts the items contained in each subset, in descending order of their CRS_{rp} score. For rating predictions having equal CRS_{rp} score values, the numeric rating prediction value is used as a tiebreaker.
- Step 4: The recommendation formulation process begins to select items from $ICRL_{Top}$ in descending order of the sorting performed in step 3, until the target number of recommendations is reached. If the elements of $ICRL_{Top}$ do not suffice, then the elements of $ICRL_{Med}$ are used, and -if needed- the elements of $ICRL_{Low}$ are also considered.

An example of the proposed algorithm is illustrated in Figure 1, while in the next section, we assess its recommendation accuracy.

5 EVALUATION

In this section, we detail on the experiments of recommendation accuracy and recommendation coverage of the presented algorithm.

5.1 Experimental Settings

Our experimental evaluation utilises five CF datasets, where the first three are sparse and the last two are dense, covering thus all sparsity levels. These five datasets are broadly used in CF research and are summarized in Table 2.

Considering the user-user vicinity metrics, we employ the Pearson Correlation Coefficient (PCC) (Ajaegbu, 2021; Jain et al., 2023). For the NN selection method, in this work, we employ both the top-k (KNN) and the correlation threshold (THR) techniques (Li et al., 2020; Singh et al., 2020). More specifically, following the approaches of the works in (Fkih, 2022; Margaris et al., 2024; D. Wang et al., 2020) in our experiments we set the K=200 and K=500, regarding the top-k technique, and T=0.0 and T=0.5, regarding the correlation threshold technique.

Table 2: The attributes of the datasets used in our experiments.

Dataset Name	Dataset Attributes
Amazon Videogames (Ni et al., 2019)	#ratings: 473K / ratings range: 1-5 #users: 17,500 / #items: 55,000 density: 0,05% (sparse)
Amazon Digital Music (Ni et al., 2019)	#ratings: 145K / ratings range: 1-5 #users: 12,000 / #items: 17,000 density: 0.07% (sparse)
CiaoDVD (Guo et al., 2014)	#ratings: 73K / ratings range: 1-5 #users: 17,600 / #items: 16,000 density: 0.026% (sparse)
MovieLens 100K (Harper & Konstan, 2016)	#ratings: 100K/ ratings range: 0.5-5 #users: 600 / #items: 9,700 density: 1.72% (dense)
MovieLens 1M (Harper & Konstan, 2016)	#ratings: 1,000K/ ratings range: 1-5 #users: 6,000 / #items: 3,700 density: 4.5% (dense)

Regarding the evaluation metrics, in this work, we employ (i) the precision of the recommendations, (ii) their average real numeric rating values and (iii) their normalized discounted cumulative gain (NDCG), following the works in (Chin et al., 2022; Krichene & Rendle, 2020), while regarding the number of recommended items we use the top-3 and top-5.

In order to generate predictions for the unrated items in the datasets summarized in Table 2, the five-fold cross validation process was followed (L. Chen et al., 2021; Zhang et al., 2021).

5.2 Evaluation Results

5.2.1 Recommendation Coverage

Figure 2 depicts the recommendation coverage considering the top-3 recommendations, when employing the KNN technique and having set the number of near neighbours to 200. This diagram effectively depicts the percentage of cases where each algorithm could produce a *complete recommendation*, i.e., a recommendation containing three items. We can observe that the algorithm proposed in this paper fully maintains the coverage attained by the plain CF algorithm in all cases, while the algorithm proposed in (Margaris et al., 2024) suffers substantial coverage drops. Especially when considering sparse datasets, coverage drops exhibited by the algorithm proposed in (Margaris et al., 2024) range from 75.2% (CiaoDVD) to 82.2% (Amazon Videogames), rendering this algorithm practically unusable for these datasets, since in 92-98% of the total cases, it would be not capable of offering a complete recommendation.

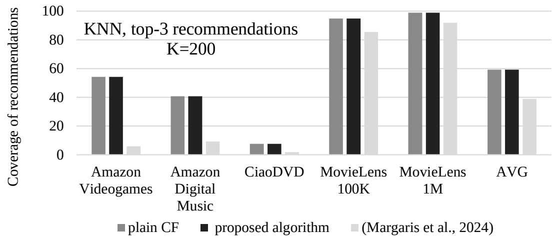

Figure 2: Recommendation coverage considering the top-3 recommendations, using the KNN technique (K=200).

For dense datasets, the coverage drop of the algorithm proposed in (Margaris et al., 2024) is again considerable, ranging from 7.9% to 9.9%.

When the KNN technique is employed with K=500, the increase of near neighbours leads to more candidate items, hence the coverage drop observed for the algorithm proposed in (Margaris et al., 2024) is lower, ranging from 69.7% to 73% in sparse datasets and from 4.9% to 16.3% in dense datasets. However, still the percentage of cases for which a complete recommendation can be offered is very low in sparse datasets (2%-15%), therefore the algorithm proposed in (Margaris et al., 2024) is effectively not applicable for sparse datasets.

The results obtained under the threshold method (THR) are similar: for sparse datasets, the algorithm proposed in (Margaris et al., 2024) exhibits very low coverage, ranging from 1.7% to 15.11%, with the coverage dropping between 74.1% and 83.5%, being thus, again, practically non-applicable.

Comparable results are obtained when the number of items offered per recommendation is increased to 5, in both cases. On the other hand, the algorithm proposed in this paper retains the coverage achieved by the plain CF algorithm in all cases.

5.2.2 Recommendation Accuracy

Considering that the algorithm in (Margaris et al., 2024) has been shown in subsection 5.2.1 to be practically not applicable for sparse datasets, due to the sharp coverage drops, and additionally suffers substantial coverage drops when applied to dense datasets, in the following we will compare the recommendation accuracy of the proposed algorithm against the plain CF algorithm only.

Figure 3 depicts the recommendation precision of the top-3 recommendations, when the KNN technique is employed with K=200. Considering the mean of all five datasets, the presented algorithm increases the recommendation precision by 3% (from 82% to 84.5%). At individual dataset level, two cases are notable: firstly, when the plain CF algorithm is used with the MovieLens 1M dataset, the precision results are mediocre (67.9%). However, when the presented algorithm is employed, the precision is enhanced at 71.7% (a 5.6% increase). Secondly, when considering the Amazon Digital Music dataset, the plain CF algorithm achieves a recommendation precision equal to 96%, leaving very small room for enhancement. Nevertheless, the proposed algorithm achieves to enhance the recommendation precision even by a small amount, to 96.67%.

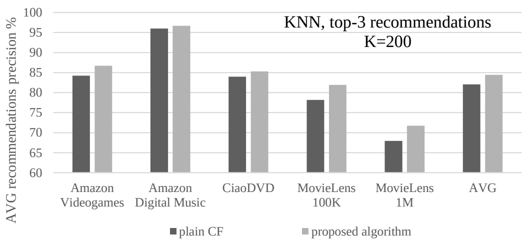

Figure 3: Mean precision value of the top-3 recommendations, under the KNN technique (K=200).

Figure 4 depicts the mean real rating value of the top-3 recommendations, when the KNN technique is employed (again, with K=200). Considering the mean of all five datasets, the presented algorithm is able to increase the recommendation value by 1.9% (from 4.27/5 to 4.35/5). The improvements observed for the MovieLens 1M and Amazon Digital Music datasets are similar to the ones discussed regarding the recommendation precision: in MovieLens 1M a considerable improvement is achieved (3.86/5 is elevated to 3.98/5), while for the Amazon Digital Music the improvement margins are very slim, leading to modest gains (4.8/5 to 4.83/5).

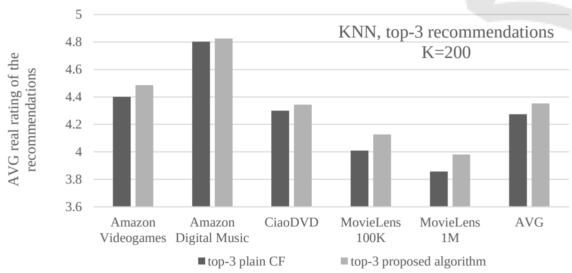

Figure 4: Mean real rating value of the top-3 recommendations, when employing the KNN technique.

Figure 5 depicts the mean NDCG value of the top-3 recommendations, when the KNN technique is employed with K=200. Considering the mean of all five datasets, the algorithm proposed in this paper achieves a NDCG value enhancement from 0.973 to 0.979. The improvements observed for the MovieLens 1M and Amazon Digital Music datasets are similar to the ones discussed regarding the previous two evaluation metrics: in MovieLens 1M a considerable improvement is achieved (0.94 is increased to 0.95), while for the Amazon Digital Music the improvement margins are very slim, leading to modest gains (0.991 to 0.993).

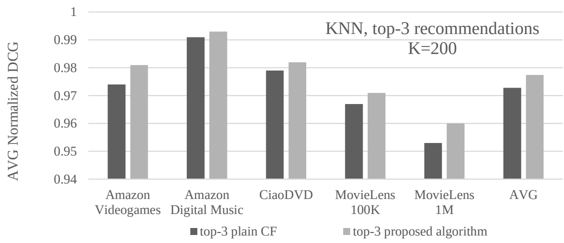

Figure 5: Mean NDCG value of the top-3 recommendations, when employing the KNN technique.

Similar results are observed when the recommended items are increased to 5 (top-5 recommendations). More specifically, the mean precision value is enhanced from 82.6% to 84.5%, the mean rating value from 4.28/5 to 4.35/5 and the mean NDCG from 0.967 to 0.973.

When K is increased to 500 (500 NNs per user), similar results are observed: the average precision value increases from 82.6% to 85.1%, the mean rating value from 4.31/5 to 4.38/5 and the mean NDCG from 0.973 to 0.978, in the top-3 recommendations setting. Regarding the top-5 recommendations setting, the respective numbers are 83.2% and 85.2%, 4.31/5 and 4.38/5, and 0.967 and 0.973.

Regarding the mean recommendation precision of the top-3 recommendations, when the THR technique is employed, with the threshold being set to T=0.0, considering the mean of all five datasets, the presented algorithm increases recommendation precision by 2.3% (from 85.2% to 87.2%). The respective mean rating value of all five datasets, is found increase by 1.6% (from 4.35/5 to 4.42/5), while the mean NDCG value increases from 0.975 to 0.979.

Similar results are observed when the number of items per recommendation are increased from 3 to 5 (top-5 recommendations). More specifically, the mean precision is upgraded from 85.5% to 87%, the average rating value is upgraded from 4.37/5 to 4.42/5, while the mean NDCG from 0.970 to 0.975.

When threshold T is increased to 0.5, similar results are again observed. More specifically, the mean precision value is enhanced from 84.8% to 86.6%, the mean rating value from 4.35/5 to 4.42/5 and, finally, the mean NDCG is enhanced from 0.975 to 0.979, in the top-3 recommendations case. When the recommendations are increased from 3 to 5 (top-5 recommendations), the respective numbers are 85% and 86.4%, 4.36/5 and 4.41/5, and 0.970 and 0.975.

5.2.3 Execution Efficiency

The presented algorithm introduces three distinct overheads, in comparison to the plain CF algorithm.

The first one concerns the rating prediction step where (a) the average rating value of each item and of each user, and (b) the NN number of each rating prediction, have to be calculated. Both of these computations can be performed offline, as well as the PCC metric, used in this work, includes the calculation of the average rating value of each user, and therefore this overhead is considered negligible.

The second overhead concerns the partitioning of each rating prediction to the four subsets ($ICRL_{Top}$, $ICRL_{Med}$, $ICRL_{Low}$ and "eliminated"), which takes place in the step 2 of the proposed algorithm. Since this process requires two minor additional actions (one comparison and a separate store), for each rating prediction, again, this individual overhead is considered negligible.

The last overhead concerns the sorting of the rating predictions within each subset. Since the plain CF recommendation algorithm performs anyhow a sorting of all rating predictions generated for each user, there is no additional overhead. In fact, since the algorithm needs to sort three smaller sets, rather than one larger one, the proposed algorithm will need less time to perform the sorting, as compared to the plain CF algorithm.

As a result, based on the overhead analysis, the overall additional overhead is considered negligible. Furthermore, to verify the aforementioned theoretical overhead analysis output, we measured the execution times of two datasets, the Amazon Videogames and the MovieLens 100K, between the plain CF and the proposed algorithm. The additional overhead was found to be less than 1.2% in both datasets.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we presented a CF recommendation algorithm which determines the items to be recommended by considering both the rating prediction values and confidence factors of predictions, allowing for predictions with higher confidence factors to outrank predictions with higher value, but lower confidence. The presented algorithm achieves to enhance the recommendation quality, while at the same time retaining the number of recommendations for each user.

More specifically, the presented algorithm partitions the items candidate for recommendation into three subsets/groups, based on their rating prediction values, corresponding to items where the user will (i) "definitely" like them, (b) "most probably" like them and (c) "probably" like them. Afterwards, the algorithm sorts the items contained in each set, in descending order, based not on their rating prediction value (as the plain CF algorithm does), but on the number of confidence factors the items' predictions satisfy. At the end, the recommendation process begins to select items from the first subset, then continues, as needed, to the second and finally to the third one, for each user.

The proposed algorithm was evaluated through a set of experiments, which included five rating datasets, both dense and sparse, as well as two NN selection methods. These experiments have shown that the proposed algorithm maintains recommendation coverage levels, while achieving satisfactory enhancement in recommendation accuracy, as calculated in terms of (i) recommendation precision, (ii) mean real rating value of the recommended items, and (iii) NDCG metrics.

Furthermore, the presented algorithm (i) needs no supplementary information concerning either the users or the items, and (ii) has been shown to induce negligible additional overhead, indicating both its wide applicability and effectiveness.

Regarding future work, we are planning to explore more features related to prediction accuracy and apply them into the recommendation process. Furthermore, we will focus on including basic supplementary RecSys information sources, e.g. user and item attributes, demographics, and types-categories of items.

REFERENCES

Ajaegbu, C. (2021). An optimized item-based collaborative filtering algorithm. Journal of Ambient Intelligence and Humanized Computing, 12(12), 10629–10636. https://doi.org/10.1007/s12652-020-02876-1

Alhijawi, B., Al-Naymat, G., Obeid, N., & Awajan, A. (2021). Novel predictive model to improve the accuracy of collaborative filtering recommender systems. Information Systems, 96, 101670. https://doi.org/10.1016/j.is.2020.101670

Bobadilla, J., Gutiérrez, A., Yera, R., & Martínez, L. (2023). Creating synthetic datasets for collaborative filtering recommender systems using generative adversarial networks. Knowledge-Based Systems, 280, 111016. https://doi.org/10.1016/j.knosys.2023.111016

Chen, L., Yuan, Y., Yang, J., & Zahir, A. (2021). Improving the Prediction Quality in Memory-Based Collaborative Filtering Using Categorical Features. Electronics, 10(2), 214. https://doi.org/10.3390/electronics10020214

Chen, Y.-C., Hui, L., & Thaipisutikul, T. (2021). A collaborative filtering recommendation system with dynamic time decay. The J. of Supercomputing, 77(1), 244–262. https://doi.org/10.1007/s11227-020-03266-2

Chin, J. Y., Chen, Y., & Cong, G. (2022). The Datasets Dilemma: How Much Do We Really Know About Recommendation Datasets? Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 141–149. https://doi.org/10.1145/3488560.3498519

Fareed, A., Hassan, S., Belhaouari, S. B., & Halim, Z. (2023). A collaborative filtering recommendation framework utilizing social networks. Machine Learning with Applications, 14, 100495. https://doi.org/10.1016/j.mlwa.2023.100495

Felfernig, A., Boratto, L., Stettinger, M., & Tkalčič, M. (2018). Evaluating Group Recommender Systems. In A. Felfernig, L. Boratto, M. Stettinger, & M. Tkalčič, Group Recommender Systems (pp. 59–71). Springer. https://doi.org/10.1007/978-3-319-75067-5_3

Fkih, F. (2022). Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. Journal of King Saud University – Comp. and Inf. Sciences, 34(9), 7645–7669. https://doi.org/10.1016/j.jksuci.2021.09.014

Guo, G., Zhang, J., Thalmann, D., & Yorke-Smith, N. (2014). ETAF: An extended trust antecedents framework for trust prediction. 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 540–547. https://doi.org/10.1109/ASONAM.2014.6921639

Harper, F. M., & Konstan, J. A. (2016). The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems, 5(4), 1–19. https://doi.org/10.1145/2827872

Jain, G., Mahara, T., & Sharma, S. C. (2023). Performance Evaluation of Time-based Recommendation System in Collaborative Filtering Technique. Procedia Computer Science, 218, 1834–1844. https://doi.org/10.1016/j.procs.2023.01.161

Krichene, W., & Rendle, S. (2020). On Sampled Metrics for Item Recommendation. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1748–1757. https://doi.org/10.1145/3394486.3403226

Li, D., Jin, R., Gao, J., & Liu, Z. (2020). On Sampling Top-K Recommendation Evaluation. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2114–2124. https://doi.org/10.1145/3394486.3403262

Margaris, D., Sgardelis, K., Spiliotopoulos, D., & Vassilakis, C. (2024). Exploiting Rating Prediction Certainty for Recommendation Formulation in Collaborative Filtering. Big Data and Cognitive Computing, 8(6), 53. https://doi.org/10.3390/bdcc8060053

Margaris, D., Vassilakis, C., & Spiliotopoulos, D. (2020). What makes a review a reliable rating in recommender systems? Inf. Processing & Management, 57(6), 102304. https://doi.org/10.1016/j.ipm.2020.102304

Margaris, D., Vassilakis, C., & Spiliotopoulos, D. (2022). On Producing Accurate Rating Predictions in Sparse Collaborative Filtering Datasets. Information, 13(6), 302. https://doi.org/10.3390/info13060302

Nguyen, L. V., Vo, Q.-T., & Nguyen, T.-H. (2023). Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services. Big Data and Cognitive Computing, 7(2), 106. https://doi.org/10.3390/bdcc7020106

Ni, J., Li, J., & McAuley, J. (2019). Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 188–197. https://doi.org/10.18653/v1/D19-1018

Sen, R., Goswami, S., & Chakraborty, B. (2019). Jeffries-Matusita distance as a tool for feature selection. 2019 International Conference on Data Science and Engineering (ICDSE), 15–20. https://doi.org/10.1109/ICDSE47409.2019.8971800

Singh, P. K., Sinha, M., Das, S., & Choudhury, P. (2020). Enhancing recommendation accuracy of item-based collaborative filtering using Bhattacharyya coefficient and most similar item. Applied Intel., 50(12), 4708–4731. https://doi.org/10.1007/s10489-020-01775-4

Spiliotopoulos, D., Margaris, D., & Vassilakis, C. (2022). On Exploiting Rating Prediction Accuracy Features in Dense Collaborative Filtering Datasets. Information, 13(9), 428. https://doi.org/10.3390/info13090428

Trattner, C., Said, A., Boratto, L., & Felfernig, A. (2024). Evaluating Group Recommender Systems. In A. Felfernig, L. Boratto, M. Stettinger, & M. Tkalčič (Eds.), Group Recommender Systems (pp. 63–75). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-44943-7_3

Vuong Nguyen, L., Nguyen, T., Jung, J. J., & Camacho, D. (2021). Extending collaborative filtering recommendation using word embedding: A hybrid approach. Concurrency and Computation: Practice and Experience, 35(16), e6232. https://doi.org/10.1002/cpe.6232

Wang, D., Yih, Y., & Ventresca, M. (2020). Improving neighbor-based collaborative filtering by using a hybrid similarity measurement. Exp. Syst. with Appl., 160, 113651. https://doi.org/10.1016/j.eswa.2020.113651

Wang, R., Wu, Z., Lou, J., & Jiang, Y. (2022). Attention-based dynamic user modeling and Deep Collaborative filtering recommendation. Expert Systems with Applications, 188, 116036. https://doi.org/10.1016/j.eswa.2021.116036

Wang, Z. (2023). Intelligent recommendation model of tourist places based on collaborative filtering and user preferences. Applied Artificial Intel., 37(1), 2203574. https://doi.org/10.1080/08839514.2023.2203574

Zhang, L., Li, Z., & Sun, X. (2021). Iterative rating prediction for neighborhood-based collaborative filtering. Applied Intelligence, 51(10), 6810–6822. https://doi.org/10.1007/s10489-021-02237-1