Anchoring Effect Mitigation for Complex Recommender System Design

Dionisis Margaris[1], Dimitris Spiliotopoulos[2](✉), and Costas Vassilakis[3]

[1] Department of Digital Systems, University of the Peloponnese, Sparta, Greece
margaris@uop.gr

[2] Department of Management Science and Technology, University of the Peloponnese, Tripoli, Greece
dspiliot@uop.gr

[3] Department of Informatics and Telecommunications, University of the Peloponnese, Tripoli, Greece
costas@uop.gr

Abstract. The adoption of a recommender system depends mostly on the accuracy of the computation of the rating predictions from the users. The user rating prediction accuracy has been a main topic in recent research. Collaborative filtering is one of the most prevalent algorithms for rating prediction, where ratings from close users, Near Neighbours, are used for the prediction of item ratings. For the collaborating filtering algorithm implementation, researchers utilise a large set of parameters, such as the number of close users and user proximity calculation metrics, to automatically determine these users, resulting in time-consuming decision conflicts over the most appropriate selections. This paper explores an approach that enables scientists to make informed selections for the metrics, parameters, and setups, using a prototype user interface with preloaded tools for the scientist to explore close users and attributes. The end users may create simulations to gain insight into the rating prediction process. The user study verifies that the designer anchoring effect of conflict of decision may be mitigated using the proposed approach, enabling transparency for the recommendations and informed decisions from the simulations.

Keywords: User interface · Recommender systems · Collaborative filtering · Rating prediction · Conflict of decision · Anchoring effects in designing complexity · User evaluation · Usability evaluation

1 Introduction

The persistent growth of the content available in web applications, nowadays, has created an abundant amount of information for the users to consume. Recommender systems (RSs) can be used to overcome this issue, by limiting and/or prioritising the information displayed to the users, based on their perceived value [1–3]. One of the most widely used RS techniques is Collaborative Filtering (CF), whose overall goal is to produce accurate rating predictions for items unrated by the users [4–6]. Then, the items scoring

M. Kurosu et al. (Eds.): HCII 2022, LNCS 13516, pp. 424–436, 2022.
https://doi.org/10.1007/978-3-031-17615-9_29

the higher rating prediction values, and hence the highest probability that those items will be desirable, are typically recommended to the users [7–9]. As expected, the closer these rating predictions are to the real rating values, the more accurate the RS is.

CF RS designers use existing datasets to simulate output using RS models and algorithms. This allows them to select the most appropriate approach and finetune it, using low and high cut-off parameters [10–12]. In the absence of appropriate user interfaces (UIs) and visual feedback, this work would traditionally be done programmatically using custom code. With the increase of the available datasets, research approaches and finetuning methods, in the recent years, the task of manual parameterisation for multiple datasets and subsets has been shown to yield suboptimal results in several situations [13–15]. The designer is required to try out approaches and multitudes of parameters per dataset to determine the optimal setup for the recommendation engine [16–18].

Apart from the fatigue and the human errors, it is hypothesised that anchoring effect manifests itself on the designer decision process. Based on the above, the first research hypothesis (RH1) is that designers experience conflict of decision between possibly optimal setups more often than expected. A second hypothesis (RH2) is that conflict resolution (due to task complexity) may be mitigated through visual decluttering of the model/parameter settings and no-code comparative overview.

This paper reports on the user study of 30 participants on the design and evaluation of the model/parameter decision making process for item rating prediction, using (implementing) CF, which is one of the most prevalent algorithm categories for rating prediction in RSs.

The rest of this paper is structured as follows: Sect. 2 summarizes related work, while Sect. 3 overviews the necessary foundations from the CF research area. Section 4 introduces the user interface (UI) and analyses the design and functionalities, while Sect. 5 presents the user evaluation results. Finally, Sect. 6 concludes the paper and outlines the future work.

2 Related Work

Over the recent years RSs has been of major interest by many research works [19–22]. Luo et al. [23] perform unconstrained non-negative latent factor analysis on high-dimensional and sparse matrices by transferring the non-negativity constraints from the decision parameters to the output latent factors and connect them using a dependent mapping function. Afterwards, they theoretically prove that the resulting model precisely represents the original one, by making a mapping function fulfil specific conditions. Lastly, they design efficient unconstrained non-negative latent factor analysis RSs algorithms. Qin et al. [24] generalise the classic position bias model to an attribute-based propensity framework. Their methods allow propensity estimation across a wide range of implicit feedback scenarios and estimate propensity scores based on offline data. These are demonstrated by applying their framework to a Google Drive RS with millions of users. Shin [25] examines how users perceive news recommendations issues and the way they engage and interact with algorithm-recommended news. Furthermore, he introduces an underlying algorithm experience model of news recommendation, integrating the heuristic process of affective, behavioural, and cognitive factors. The proposed algorithm affects the user's perception and system trust, in different ways. The heuristic aspects transpire when the users' subjective beliefs about accuracy and transparency act as a mental shortcut. The mediating role of trust is an indication that the algorithmic performance could be enhanced by establishing algorithmic trust between news recommendation systems and users. The model illustrates the motivation behind user behaviours, as well as the users' cognitive processes of perceptual judgment. Shambour [26] introduces a deep learning-based algorithm for multi-criteria RS that exploits the nonlinear, non-trivial and hidden relations between users, regarding multi-criteria preferences, by the employment of deep auto-encoders, which generates more accurate recommendations. The algorithm produces accurate rating predictions, when compared with state-of-the-art rating prediction algorithms, based on experiments on the TripAdvisor and Yahoo! Movies multi-criteria datasets. Tian et al. [27] introduce a RS that automatically selects a best-suited metaheuristic method without trial and error on a given problem. This algorithm explores the intricacies of optimisation problems, by developing a generic tree-like data structure. It trains a deep recurrent neural network which makes automated algorithm recommendation, by learning to choose the best metaheuristic algorithm. This algorithm makes metaheuristic optimisation techniques accessible to policy makers, industrial practitioners, as well as other stakeholders having no prior knowledge of metaheuristic algorithms. Alhijawi and Kilani [28] introduce a novel genetic-based RS, that operates using historical rating data and semantic information. This genetic algorithm finds the best list of items to the active user, by hierarchically evaluating the individuals using three fitness functions. The first one estimates the strength of the semantic similarity between items, utilising the semantic information on items. The second one estimates the satisfaction level similarity between users, while the last one selects the best recommendation list, based on the predicted ratings.

Accuracy is an important aspect of CF RSs [10, 29, 30]. Singh et al. [31] seek to overcome the issue of CF recommendation inaccuracy, due to the fact that the predicted rating may tend towards the average rating value of the user. This is achieved by mitigating the issues of (i) the consideration of different value of k nearest neighbour per user and (ii) dataset sparsity. To predict the target item, for each item, they select the nearest neighbour, due to the high computational cost of finding k for each/different item. Yan and Tang [32] present a model that uses Gaussian mixture to cluster items and users. Furthermore, their model builds a new interaction matrix, by extracting new features, which solves the rating data sparsity impact on CF algorithms. Lastly, they present a new similarity calculation method that combines the Jaccard and the triangle similarities. Jain et al. [33] present an Enhanced Multistage User-based CF algorithm that predicts the unknown user ratings in two stages, using the active learning concept. This algorithm predicts the anonymous ratings for each stage using the traditional User_CF algorithm. However, it uses the Bhattacharyya Coefficient based nonlinear similarity model for the similarity computations among users. The presented algorithm uses an extension of the simple Enhanced multistage user-based CF algorithm, which achieves to increase the prediction accuracy, by progressively increasing the density of the original rating matrix. Margaris et al. [34] introduce the Experiencing Period Criterion rating prediction algorithm which enhances the prediction accuracy of CF RSs, based on the combination of the time period the rating to be predicted belongs to, in a certain product category, and the users' experiencing wait period in the same product category. The rationale behind this algorithm is that a user may apply different experience practices on different product categories, that is, being reluctant to experience new products in certain categories, while being keen to experience new products in others. Chae et al. [35] present a Rating Augmentation framework with GAN, targeting at alleviating the data sparsity problem in CF, which achieves to improve recommendation accuracy. Also, they identified that the naive Rating Augmentation GAN tends to generate values biased towards high ratings when applying GAN to CF for rating improvement. This issue is addressed by introducing a refined version of RAGAN.

HCI and RS research has started to grow over the last years [36–38]. Braham et al. [39] present a RS for selecting the most relevant design patterns in the HCI domain, by combining ontology-based and text-based techniques and by using ontology models and semantic similarity to retrieve appropriate HCI design patterns. They also validate the presented RS regarding its acceptance, by evaluating the perceived accuracy and perceived experience by users. Dominguez et al. [40] provide a perspective of the many variables involved in the user perception of several aspects of a RS, such as relevance, explainability, domain knowledge and trust. They study several aspects of the user experience with a RS of artistic images, from both HCI and algorithmic perspectives, and then they conduct user studies to evaluate the levels of explainability. Margaris et al. [41] introduce a UI for WS-BPEL designers that supports personalised recommendation and selection of business process functionalities, based on user generated criteria. More specifically, the UI gives the WS-BPEL designers the opportunity, based on either particular or total scores, to ask for web service recommendations and select specific web services out of ordered lists. This allows the users to have the final selection choice, thus overcome the automatic system selection issue that leads to adaptation failure. Locatelli Cezar et al. [42] introduce a RS for the HCI community, recommending papers from the Brazilian Symposium on Human Factors in Computing Systems. This RS applies a post-processing strategy, focused on fairness to balance the users' interests, in order to recommend papers related to each user's profile. Margaris et al. [43] present a specialised UI for WS-BPEL designers, which allows personalised recommendation and business process functionalities selection based on user generated criteria. The UI supports user-specified restrictions, based on non-qualitative criteria, WS preselection, as well as tuning of the number of retrieved candidate web services presented to the WS-BPEL designer.

However, none of the above-mentioned works explores how RS scientists can be enabled to make informed selections for the metrics, parameters, and setups of a CF RS, for example, the number of Near Neighbours (NNs) to consider and user similarity calculation metrics. This work introduces a UI with preloaded tools that enables the users to create simulations, to gain insight into the rating prediction process and, ultimately, increase the rating prediction accuracy, to essentially lead to higher recommendation accuracy.

3 Prerequisites

The major concepts from the areas of CF RSs, which are used in our work, are summarized in the following paragraphs.

First and foremost, the ultimate goal of a CF RS is to produce accurate predictions for the products that users have not yet evaluated [6]. Then, the RS typically recommends the products achieving the higher prediction values, to each user, assuming that higher prediction values usually drive to higher probability that the user will actually be interested in these products [44, 45]. Obviously, the more accurate these rating predictions are, the more reliable the recommendations will be, and hence the more successful the RS will be.

The first step of a typical CF RS is to find users y, who share similar tastes with the active user x (that is, the user for whom the rating prediction is being formulated), by comparing the real ratings that both x and y have already entered to common products p. In order to quantify the closeness of similar tastes between users, a user similarity function must be used, such as the Pearson Correlation Coefficient and the Cosine Similarity, the selection of which is made by the RS scientist [46, 47].

Afterwards, all users y, who are found to have (to a large extent) similar ratings with active user x, are considered x's candidate NNs. At this moment, the RS scientist must again select the exact number of NNs that each user x will use for prediction formulation [18, 48].

Lastly, in order to formulate the rating prediction of user x to product p, the existed ratings of x's NNs to the same product are used. This happens under the rationale that, in the real world, humans trust people close to them (close friends, family, etc.) when suggestions for a new experience/commodity are demanded (Fig. 1).

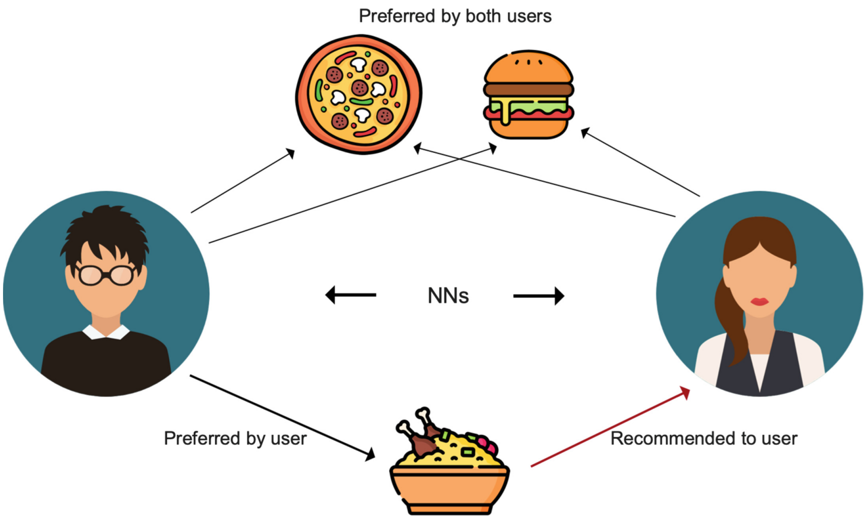

Fig. 1. Collaborative filtering recommender systems use near neighbours to recommend items to users.

Based on the aforementioned steps, both the user similarity function and the exact number of NNs directly affect the rating prediction value and, hence, accuracy, which ultimately affects the recommendation success. As a result, the selections that the RS scientist must take are of critical importance. Scientists spend time researching the optimal setups and are asked to make decisions between multiple seemingly adequate options. Decision conflicts are time consuming and hinder productivity. This necessitated the implementation of a UI with preloaded tools for the RS scientist to support these selections and resolve decision conflict.

4 User Interface Design and Functionalities

The proposed UI was designed and implemented with the following functionalities:

- Selection of the similarity function between the CF users.
- Selection of the exact number of NNs that each user x will use for prediction formulation.
- Real time simulation for selected setups and metrics and graphical representation of accuracy

In the following subsections the above functionalities will be analysed. The UI is depicted in Fig. 2.

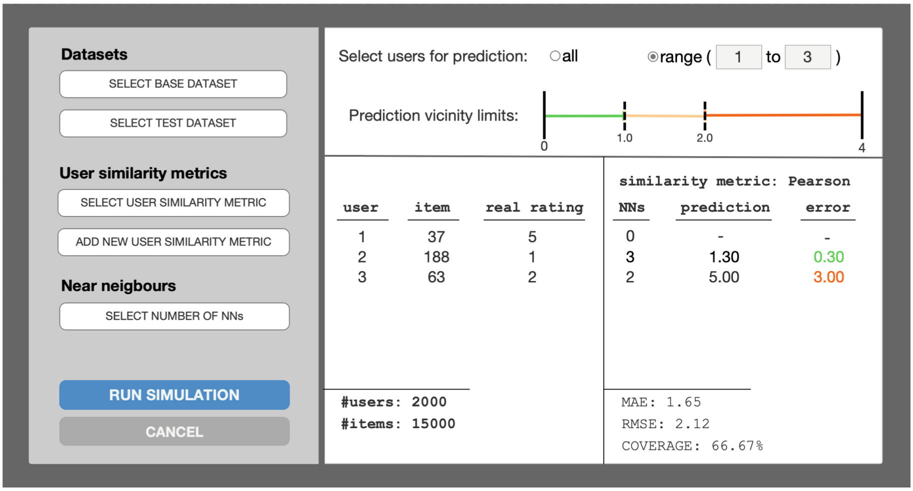

Fig. 2. The UI for rating prediction analysis for collaborating filtering. (Color figure online)

On the left-hand-side, the RS scientist selects all the necessary information a CF algorithm needs, that is, the base dataset, the test dataset, the user similarity metrics, and the number of NNs used. On the upper part of the screen, the scientist selects the users (all or a range of users) which the rating prediction procedure will formulate predictions for. The UI allows the RS scientist to select the exact number of NNs that will take part, for each user, to the rating prediction formulation. If a user has less NNs than the selected number, all of his NNs will be used for the rating prediction formulation.

According to the vicinity between the rating prediction and the real user rating, the user can select the limits for which the UI displays the prediction error in colours (the red colour indicates large prediction errors, the yellow colour indicates medium prediction errors, and the green colour indicates small prediction errors) so that the user can easily locate the cases to focus on.

Lastly, after the simulation is run, the rating predictions are displayed on the right-hand side. The predictions also incorporate statistics that may help the user interpret the results of the rating prediction procedure. These include the number of users and items the dataset contains, the Mean Average Error and the Root Mean Square Error, as well as the prediction coverage of the prediction process.

The UI allows the user to select from the preloaded user similarity functions, such as the Pearson Correlation Coefficient and the Cosine Similarity. Apart from the already implemented user similarity metrics, the user may add, edit, or delete custom metrics (Fig. 3). These add to the flexibility of the UI, since many users prefer to utilise their own code implementations.

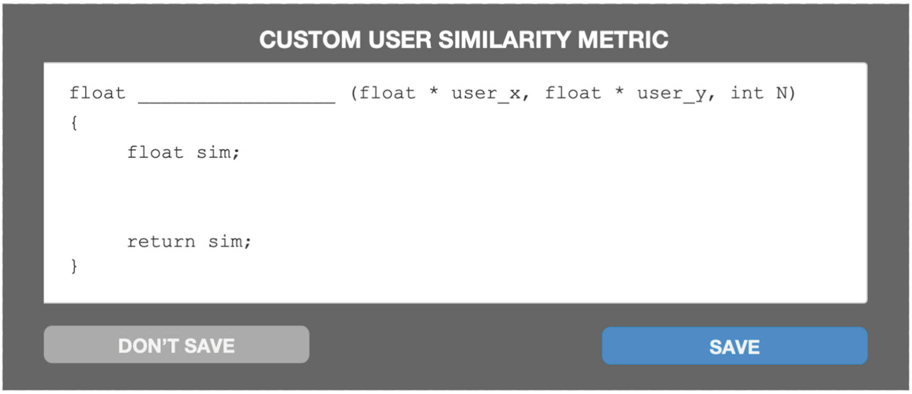

Fig. 3. User-created custom similarity metrics for recommendation simulations.

At the moment, the UI supports the C programming language. However, in our future work we are planning to extend it so that it also supports JAVA and Python programming languages, which many RS scientists use, as well.

5 Evaluation

This section reports on the user study of 30 computer science literate participants on the design and evaluation of the model/parameter decision making process for the rating prediction, implementing the CF algorithm. The mean age of users was 20.7 years, 70% male and 30% female.

For RH1, think aloud testing was applied to monitor and register decision conflicts. Time to task completion and number of individual trials per dataset (including back-tracking to former datasets and setups when optimal parameters were found) were also logged. This was studied for 5 datasets, randomly selected out of the 10 trial datasets, widely used in CF research, for the study, namely the Amazon datasets [49, 50] and the MovieLens datasets [51].

The RH2 investigation was contacted as follows. For the remaining half of the datasets, the users explored the selection set of model and parameters for the RS and simulated the applied setups using a prototype UI that implemented the following functionalities:

- Selection of input/simulation base and test datasets.
- Selection of user vicinity metric.
- Capacity to add new, custom vicinity metrics by the user.
- Selection of number of NNs for output prediction.
- Selection of range of target users (i.e., from the datasets) which the prediction output is computed for. This allows for insight into the algorithm and better explainability.
- Colour coded vicinity prediction to real values.

All the aforementioned metrics were also applied and actions were recorded. In addition, a usability evaluation was performed on the latter approach using the moderated feedback, namely the think aloud method and user behaviour and perception metrics, as in similar works [43, 52]. The sessions were run for up to 30 m per task for the total of two tasks per user.

RH1 was found true. The users reported between six and twenty conflicts of decision per session (Fig. 4). The main findings were:

- Time consuming actions: the users attempted to run multiple tries on a dataset, make deductions on the optimal setup and proceed to try the setup for the next dataset (termed as "traditional developer" approach). This resulted in several re-tries of similar parameters and lack of insight on the datasets that followed the ones tested, since they would be examined anew.
- Conflict of decision: the users experienced most conflicts when similar positive results were evaluated, since no insight on the performance of the setups for the other datasets (sparse and dense) could be known.

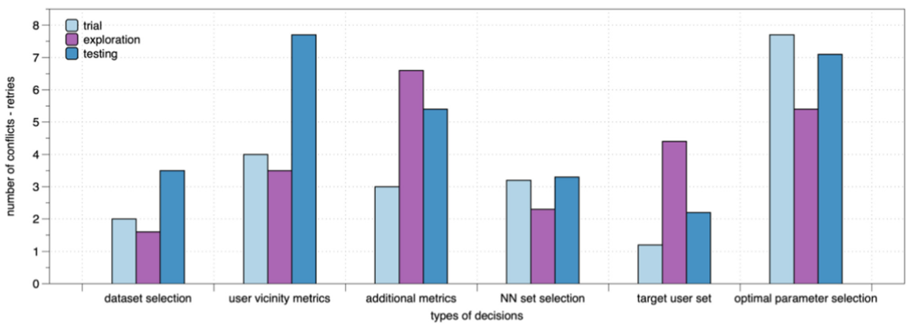

Fig. 4. User-reported average decision conflict breakdown. The results clearly depict the activities and types of decisions that trigger conflicts. Those conflicts are mitigated through the proposed approach.

RH2 was also found true. The users reported that the colour-coded vicinity prediction to real values in a comparative manner was helpful for making informed decisions regarding setups that worked well for groups of datasets (two or more). Since dataset performance varies due to dataset content differences, the target user range selection was helpful in grouping setup performance even better for groups of datasets. In addition, the users reported very high acceptance (behaviour metrics: task completions and confidence; perception metrics: task completion and task difficulty).

The approach that utilised the UI prototype reduced the conflicts by 4.71 times on average, while the maximum reduction was over 6 times on average for the five most problematic types of decisions.

6 Conclusion

This work presented a UI with preloaded tools for RSs scientists, targeting at mitigating the anchoring effect of conflict of decision that the scientists face. More specifically, the scientists can make informed selections for the metrics, parameters and setups, as well as create simulations to gain insight into the rating prediction process. A user study verified that the RS designers face the anchoring effect of conflict of decision. This may be mitigated using the proposed approach, enabling transparency for the recommendations and informed decisions from the graphical results from the recommendation process simulations.

In our future work, we are planning to extend the set of languages that the UI supports, including JAVA and Python programming languages, which many RS scientists use [53, 54]. Furthermore, we are planning the UI to include graphical representation of the rating prediction results. Last, we are planning to extend the rating prediction error metrics, including the most used ones in RS research [10, 12, 55].

References

1. Alslaity, A., Tran, T.: Goal modeling-based evaluation for personalized recommendation systems. In: Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, pp. 276–283. ACM, Utrecht Netherlands (2021). https://doi.org/10.1145/3450614.3464619
2. Berkovsky, S., Freyne, J.: Web personalization and recommender systems. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2307–2308. ACM, Sydney, NSW, Australia (2015). https://doi.org/10.1145/2783258.2789995
3. Aivazoglou, M., et al.: A fine-grained social network recommender system. Soc. Netw. Anal. Min. **10**(1), 1–18 (2019). https://doi.org/10.1007/s13278-019-0621-7
4. Margaris, D., Spiliotopoulos, D., Vassilakis, C.: Social relations versus near neighbours: reliable recommenders in limited information social network collaborative filtering for online advertising. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1160–1167. ACM, Vancouver, British Columbia, Canada (2019). https://doi.org/10.1145/3341161.3345620
5. Veras De Sena Rosa, R.E., Guimaraes, F.A.S., da SilvaMendonca, R., de Lucena, V.F.: Improving prediction accuracy in neighborhood-based collaborative filtering by using local similarity. IEEE Access **8**, 142795–142809 (2020). https://doi.org/10.1109/ACCESS.2020.3013733

6. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**, 5–53 (2004). https://doi.org/10.1145/963770.963772
7. Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM. **40**, 66–72 (1997). https://doi.org/10.1145/245108.245124
8. Cechinel, C., Sicilia, M.-Á., Sánchez-Alonso, S., García-Barriocanal, E.: Evaluating collaborative filtering recommendations inside large learning object repositories. Inf. Process. Manage. **49**, 34–50 (2013). https://doi.org/10.1016/j.ipm.2012.07.004
9. Margaris, D., Kobusinska, A., Spiliotopoulos, D., Vassilakis, C.: An adaptive social network-aware collaborative filtering algorithm for improved rating prediction accuracy. IEEE Access **8**, 68301–68310 (2020). https://doi.org/10.1109/ACCESS.2020.2981567
10. Kluver, D., Ekstrand, M.D., Konstan, J.A.: Rating-based collaborative filtering: algorithms and evaluation. In: Brusilovsky, P., He, D. (eds.) Social Information Access. LNCS, vol. 10100, pp. 344–390. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-90092-6_10
11. Cunha, T., Soares, C., de Carvalho, A.C.P.L.F.: Selecting collaborative filtering algorithms using metalearning. In: Frasconi, P., Landwehr, N., Manco, G., Vreeken, J. (eds.) ECML PKDD 2016. LNCS (LNAI), vol. 9852, pp. 393–409. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46227-1_25
12. Jalili, M., Ahmadian, S., Izadi, M., Moradi, P., Salehi, M.: Evaluating collaborative filtering recommender algorithms: a survey. IEEE Access **6**, 74003–74024 (2018). https://doi.org/10.1109/ACCESS.2018.2883742
13. McNee, S.M., Riedl, J., Konstan, J.A.: Making recommendations better: an analytic model for human-recommender interaction. In: Proceedings of the CHI 2006 Extended Abstracts on Human Factors in Computing Systems, pp. 1103–1108. ACM, Montréal, Québec, Canada (2006). https://doi.org/10.1145/1125451.1125660
14. Miller, B.N., Albert, I., Lam, S.K., Konstan, J.A., Riedl, J.: MovieLens unplugged: experiences with an occasionally connected recommender system. In: Proceedings of the 8th International Conference on Intelligent User Interfaces - IUI 2003, p. 263. ACM Press, Miami, Florida, USA (2003). https://doi.org/10.1145/604045.604094
15. Kunkel, J., Loepp, B., Ziegler, J.: A 3D item space visualization for presenting and manipulating user preferences in collaborative filtering. In: Proceedings of the 22nd International Conference on Intelligent User Interfaces, pp. 3–15. ACM, Limassol, Cyprus (2017). https://doi.org/10.1145/3025171.3025189
16. Chae, D.-K., Lee, S.-C., Lee, S.-Y., Kim, S.-W.: On identifying k -nearest neighbors in neighborhood models for efficient and effective collaborative filtering. Neurocomputing **278**, 134–143 (2018). https://doi.org/10.1016/j.neucom.2017.06.081
17. Li, S., Karatzoglou, A., Gentile, C.: Collaborative filtering bandits. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 539–548. ACM, Pisa, Italy (2016). https://doi.org/10.1145/2911451.2911548
18. Verstrepen, K., Goethals, B.: Unifying nearest neighbors collaborative filtering. In: Proceedings of the 8th ACM Conference on Recommender Systems - RecSys 2014, pp. 177–184. ACM Press, Foster City, Silicon Valley, California, USA (2014). https://doi.org/10.1145/2645710.2645731
19. Jannach, D., Manzoor, A., Cai, W., Chen, L.: A survey on conversational recommender systems. ACM Comput. Surv. **54**, 1–36 (2022). https://doi.org/10.1145/3453154
20. Nikolakopoulos, A.N., Ning, X., Desrosiers, C., Karypis, G.: Trust your neighbors: a comprehensive survey of neighborhood-based methods for recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 39–89. Springer, New York, NY (2022). https://doi.org/10.1007/978-1-0716-2197-4_2
21. Karimi, M., Jannach, D., Jugovac, M.: News recommender systems – survey and roads ahead. Inf. Process. Manage. **54**, 1203–1227 (2018). https://doi.org/10.1016/j.ipm.2018.04.008

22. Deldjoo, Y., Schedl, M., Hidasi, B., Wei, Y., He, X.: Multimedia recommender systems: algorithms and challenges. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 973–1014. Springer US, New York, NY (2022). https://doi.org/10.1007/978-1-0716-2197-4_25
23. Luo, X., Zhou, M., Li, S., Wu, D., Liu, Z., Shang, M.: Algorithms of unconstrained non-negative latent factor analysis for recommender systems. IEEE Trans. Big Data. **7**, 227–240 (2021). https://doi.org/10.1109/TBDATA.2019.2916868
24. Qin, Z., Chen, S.J., Metzler, D., Noh, Y., Qin, J., Wang, X.: Attribute-based propensity for unbiased learning in recommender systems: algorithm and case studies. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2359–2367. ACM, Virtual Event, CA, USA (2020). https://doi.org/10.1145/3394486.3403285
25. Shin, D.: How do users interact with algorithm recommender systems? The interaction of users, algorithms, and performance. Comput. Hum. Behav. **109**, 106344 (2020). https://doi.org/10.1016/j.chb.2020.106344
26. Shambour, Q.: A deep learning based algorithm for multi-criteria recommender systems. Knowl.-Based Syst. **211**, 106545 (2021). https://doi.org/10.1016/j.knosys.2020.106545
27. Tian, Y., Peng, S., Zhang, X., Rodemann, T., Tan, K.C., Jin, Y.: A recommender system for metaheuristic algorithms for continuous optimization based on deep recurrent neural networks. IEEE Trans. Artif. Intell. **1**, 5–18 (2020). https://doi.org/10.1109/TAI.2020.3022339
28. Alhijawi, B., Kilani, Y.: A collaborative filtering recommender system using genetic algorithm. Inf. Process. Manage. **57**, 102310 (2020). https://doi.org/10.1016/j.ipm.2020.102310
29. Rendle, S., Krichene, W., Zhang, L., Anderson, J.: Neural collaborative filtering vs. matrix factorization revisited. In: Proceedings of the Fourteenth ACM Conference on Recommender Systems, pp. 240–248. ACM, Virtual Event, Brazil (2020). https://doi.org/10.1145/3383313.3412488
30. Koren, Y., Rendle, S., Bell, R.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 91–142. Springer, New York, NY (2022). https://doi.org/10.1007/978-1-0716-2197-4_3
31. Singh, P.K., Sinha, M., Das, S., Choudhury, P.: Enhancing recommendation accuracy of item-based collaborative filtering using Bhattacharyya coefficient and most similar item. Appl. Intell. **50**(12), 4708–4731 (2020). https://doi.org/10.1007/s10489-020-01775-4
32. Yan, H., Tang, Y.: Collaborative filtering based on gaussian mixture model and improved jaccard similarity. IEEE Access **7**, 118690–118701 (2019). https://doi.org/10.1109/ACCESS.2019.2936630
33. Jain, A., Nagar, S., Singh, P.K., Dhar, J.: EMUCF: enhanced multistage user-based collaborative filtering through non-linear similarity for recommendation systems. Expert Syst. Appl. **161**, 113724 (2020). https://doi.org/10.1016/j.eswa.2020.113724
34. Margaris, D., Spiliotopoulos, D., Vassilakis, C., Vasilopoulos, D.: Improving collaborative filtering's rating prediction accuracy by introducing the experiencing period criterion. Neural Comput. Appl., 1-18 (2020). https://doi.org/10.1007/s00521-020-05460-y
35. Chae, D.-K., Kang, J.-S., Kim, S.-W., Choi, J.: Rating augmentation with generative adversarial networks towards accurate collaborative filtering. In: Proceedings of The World Wide Web Conference on - WWW 2019, pp. 2616–2622. ACM Press, San Francisco, CA, USA (2019). https://doi.org/10.1145/3308558.3313413
36. Jesse, M., Jannach, D.: Digital nudging with recommender systems: survey and future directions. Comput. Hum. Behav. Rep. **3**, 100052 (2021). https://doi.org/10.1016/j.chbr.2020.100052

37. Jannach, D., Pu, P., Ricci, F., Zanker, M.: Recommender systems: past, present future. AIMag. **42**, 3–6 (2021). https://doi.org/10.1609/aimag.v42i3.18139
38. Vultureanu-Albisi, A., Badica, C.: Recommender systems: an explainable AI perspective. In: Proceedings of the 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), pp. 1–6. IEEE, Kocaeli, Turkey (2021). https://doi.org/10.1109/INISTA52262.2021.9548125
39. Braham, A., Khemaja, M., Buendía, F., Gargouri, F.: A hybrid recommender system for HCI design pattern recommendations. Appl. Sci. **11**, 10776 (2021). https://doi.org/10.3390/app112210776
40. Dominguez, V., Donoso-Guzmán, I., Messina, P., Parra, D.: Algorithmic and HCI aspects for explaining recommendations of artistic images. ACM Trans. Interact. Intell. Syst. **10**, 1–31 (2020). https://doi.org/10.1145/3369396
41. Margaris, D., Spiliotopoulos, D., Vassilakis, C., Karagiorgos, G.: A user interface for personalized web service selection in business processes. In: Stephanidis, C., et al. (eds.) HCII 2020. LNCS, vol. 12427, pp. 560–573. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60152-2_41
42. Cezar, N.L., de Borba, C., Gasparini, I., Lichtnow, D.: Applying a post-processing strategy to consider the multiple interests of users of a paper recommender system. In: Proceedings of the XVII Brazilian Symposium on Information Systems, pp. 1–7. ACM, Uberlândia, Brazil (2021). https://doi.org/10.1145/3466933.3466985
43. Margaris, D., Spiliotopoulos, D., Vasilopoulos, D., Vassilakis, C.: A user interface for personalising WS-BPEL scenarios. In: Nah, F.-H., Siau, K. (eds.) HCII 2021. LNCS, vol. 12783, pp. 399–416. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77750-0_25
44. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the Tenth International Conference on World Wide Web - WWW 2001, pp. 285–295. ACM Press, Hong Kong, Hong Kong (2001). https://doi.org/10.1145/371920.372071
45. Xue, F., He, X., Wang, X., Xu, J., Liu, K., Hong, R.: Deep item-based collaborative filtering for top-N recommendation. ACM Trans. Inf. Syst. **37**, 1–25 (2019). https://doi.org/10.1145/3314578
46. Herlocker, J., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Inf. Retrieval **5**, 287–310 (2002). https://doi.org/10.1023/A:1020443909834
47. Elahi, M., Ricci, F., Rubens, N.: A survey of active learning in collaborative filtering recommender systems. Comput. Sci. Rev. **20**, 29–50 (2016). https://doi.org/10.1016/j.cosrev.2016.05.002
48. Margaris, D., Vasilopoulos, D., Vassilakis, C., Spiliotopoulos, D.: Improving collaborative filtering's rating prediction accuracy by introducing the common item rating past criterion. In: Proceedings of the 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1–8. IEEE, PATRAS, Greece (2019). https://doi.org/10.1109/IISA.2019.8900758
49. He, R., McAuley, J.: Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In: Proceedings of the 25th International Conference on World Wide Web, pp. 507–517. International World Wide Web Conferences Steering Committee, Montréal, Québec, Canada (2016). https://doi.org/10.1145/2872427.2883037
50. McAuley, J., Targett, C., Shi, Q., van den Hengel, A.: Image-based recommendations on styles and substitutes. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval., pp. 43–52. ACM, Santiago, Chile (2015). https://doi.org/10.1145/2766462.2767755
51. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. **5**, 1–19 (2016). https://doi.org/10.1145/2827872

52. Spiliotopoulos, D., Margaris, D., Vassilakis, C.: Data-assisted persona construction using social media data. BDCC. **4**, 21 (2020). https://doi.org/10.3390/bdcc4030021
53. Ortega, F., Mayor, J., López-Fernández, D., Lara-Cabrera, R.: CF4J 2.0: adapting collaborative filtering for java to new challenges of collaborative filtering based recommender systems. Knowl-Based Syst **215**, 106629 (2021). https://doi.org/10.1016/j.knosys.2020.106629
54. Ekstrand, M.D.: LensKit for python: next-generation software for recommender systems experiments. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2999–3006. ACM, Virtual Event, Ireland (2020). https://doi.org/10.1145/3340531.3412778
55. McLaughlin, M.R., Herlocker, J.L.: A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In: Proceedings of the 27th annual International Conference on Research and Development in Information Retrieval - SIGIR 2004, p. 329. ACM Press, Sheffield, United Kingdom (2004). https://doi.org/10.1145/1008992.1009050