A Novel Collaborative Filtering Approach Based on the Opposite Neighbors' Preferences

Abdellah El Fazziki, Ouafae El Aissaoui, Yasser El Madani El Alami, Mohammed Benbrahim and Youssouf El Allioui

Abstract Collaborative filtering (CF) has become an effective way to predict useful items. It is the most widespread recommendation technique. It relies on users who share similar tastes and preferences to suggest the items that they might be interested in. Despite its simplicity and justifiability, the collaborative filtering approach experiences many problems, including sparsity, gray sheep and scalability. These problems lead to deteriorating the accuracy of the obtained results. In this work, we present a novel collaborative filtering approach based on the opposite preferences of users. We focus on enhancing the accuracy of predictions and dealing with gray sheep problem by inferring new similar neighbors based on users who have dissimilar tastes and preferences. For instance, if a user **X** is dissimilar to a user **Y** then the user $\neg\mathbf{X}$ is similar to the user **Y**. The Experimental results performed on two datasets including MovieLens and FilmTrust show that our approach outperforms several baseline recommendation techniques.

Keywords Recommender system · Collaborative filtering · Opposite neighbors · Similarity measure

A. El Fazziki (✉) · O. El Aissaoui · M. Benbrahim
University of Sidi Mohammed Ben Abdellah, Fez, Morocco
e-mail: abdellah.elfazziki@usmba.ac.ma

O. El Aissaoui
e-mail: ouafae.elaissaouil@usmba.ac.ma

M. Benbrahim
e-mail: mohammed.benbrahim@usmba.ac.ma

Y. El Madani El Alami
ENSIAS, University of Mohammed V, Rabat, Morocco
e-mail: y.alami@um5s.net.ma

Y. El Allioui
LS3M, FPK, USMS University, B.P.: 145 25000 Khouribga, Morocco
e-mail: yelallioui@gmail.com

V. Bhateja et al. (eds.), *Embedded Systems and Artificial Intelligence*,
Advances in Intelligent Systems and Computing 1076,
https://doi.org/10.1007/978-981-15-0947-6_77

1 Introduction

Recommender systems (RS) are decision support systems used on the web in order to help users to choose useful items [1]. They aim to deal with the information overload problem by predicting useful items based on users' preferences. RS act as a filter that allows passing the relevant item to the user and blocks the irrelevant one [2]. Recommender systems are largely used in various domains such as movies [3], music [4] libraries [5] and e-commerce [6, 7].

In the last decades, various approaches have been proposed for building robust recommender systems. According to [8], recommender systems can be classified into three main categories: collaborative filtering (CF) [9], content-based [10] and hybrid recommender systems [11]. Thanks to its simplicity and justifiability. Collaborative filtering remains the most commonly implemented approach on the web [12]. CF consists in recommending items based on users who share similar tastes and preferences. Despite its strengths, CF encounters many problems which usually lead to deteriorating the accuracy of the recommendations. For instance, Gray sheep is related to users who have unusual tastes and don't share similar preferences with other users [13]. Hence, finding a reliable neighborhood is a hard task. Scalability is another recurrent problem that occurs when computing similarities among all pairs of users. This task is time-consuming, especially in huge datasets. Conjointly, the sparsity of data is caused when users do not provide explicit feedback. Actually, in most cases, users do not rate items in even though they feel an extreme emotion, either satisfaction or discontent [14].

In this paper, we focus to mitigate gray sheep problems and to improve the accuracy of recommendations based on the opposite preferences of users. In other words, we deal with gray sheep problems by generating new users based on dissimilar neighbors. The underlying assumption of our approach is that if a user X has an opposite opinion of a user Y, then, the user $\neg X$ has the same opinion as the user Y. Our approach will increase the number of similar neighbors and then allow building good recommendations.

The rest of this paper is organized as follows:

In Sect. 2 we present an overview of the collaborative filtering baseline approach. Section 3 introduces the proposed approach and the original contribution of this work. In Sect. 4, we investigate the effectiveness of our proposal using an experimental evaluation on several datasets. The conclusions and some perspectives are outlined in Sect. 5.

2 Background

Collaborative filtering techniques are the most used approach in recommender systems thanks to its easiness and efficiency [9]. CF assumes that useful pieces of advice can be predicted from users who share similar tastes and preferences. These preferences can be expressed explicitly by users using ratings regarding their interests in items [15]. They can also be inferred by monitoring users' behavior such as the history of purchases and the time spent on web content called implicit feedback [16]. The set of these triplets forms a matrix called the rating matrix. It is the basic input in collaborative filtering used to build effective prediction models and users' profiles [17].

In CF, model-based and memory-based techniques remain as the main identified categories. The former build or learn models from collected ratings based on machine learning techniques like clustering techniques [18], dimensionality reduction methods [19], support vector machines, neural networks [20]. The latter referred to as neighborhood-based collaborative filtering. Memory-based CF [21] is considered as the earliest CF algorithm. It relies on building recommendations using a similarity-based neighborhood for either users or items. In fact, user-based CF focuses on building a neighborhood for active users in order to make predictions for unseen items. The same reasoning is used for item-based CF. Both use *K*-nearest neighbor classifiers to generate predictions. In what follows, we present the user-based approach.

2.1 Memory-Based Recommendation Tasks

As reported by [22], the memory-based approach relies on three steps presented in Fig. 1.

2.1.1 Data Representation

The first task in neighborhood CF consists of building the rating matrix and to fill missing values. In fact, in most cases, the rating matrix is usually sparse since users do not rate items in a regular manner [19]. The most used technique in CF relies on filling missing values with the average user's ratings.

Fig. 1 Memory-based CF process

2.1.2 Neighborhood Formation

In this step, a neighborhood of the most similar users is built based on a similarity metric. The most commonly used formula is the Pearson correlation coefficient. It has values between -1 and $+1$ where $+1$ means total positive correlation, -1 a total negative correlation and 0 no association between the two users. It is considered as a standard way of measuring correlation [23]. Thus, the similarity between two users a and b is calculated with the following formula:

$$\mathrm{sim}_{a,b} = \frac{\sum_{j=1}^{n}\left(r_{aj} - \overline{r_a}\right)\left(r_{bj} - \overline{r_b}\right)}{\sqrt{\sum_{j=1}^{n}\left(r_{aj} - \overline{r_a}\right)^2 \sum_{j=1}^{n}\left(r_{bj} - \overline{r_b}\right)^2}} \tag{1}$$

2.1.3 Predictions Generation

The final task in the CF process consists in generating predictions for unseen items. It is computed as an aggregation of similarities between the active user and his neighborhood in addition to their ratings:

$$p_{s,i} = \bar{r}_s + \frac{\sum_{p=1}^{k}\left(r_{p,i} - \overline{r_p}\right) * \mathrm{sim}_{s,p}}{\sum_{p=1}^{k}\left|\mathrm{sim}_{s,p}\right|} \tag{2}$$

K represents the number of closest neighbors. This prediction function uses the KNN technique to estimate the rating of an unseen item i.

Therefore, based on computed predictions, recommender systems can provide top N recommendations as a list of items that the active user has never before shown any interest.

2.2 Evaluation Metrics

Many measures have been used in the literature in order to measure the accuracy of a proposed method. MAE (mean absolute error) and RMSE (root mean squared error) remain as the well-known performance metrics which are broadly used in recommender systems. MAE computes the average absolute differences between predicted ratings and real values as presented in the following formula

$$\mathrm{MAE} = \frac{\sum_{(s,i)}\left|p_{s,i} - r_{s,i}\right|}{N} \tag{3}$$

where N is the number of predicted ratings computed during the test phase. $p_{s,i}$ is the predicted rating of user s to item i. $r_{s,i}$ is the real rating value.

RMSE is a quadratic error metric which measures the square root of the average of the squared differences between predicted and actual ratings:

$$\text{RMSE} = \sqrt{\frac{\sum_{(s,i)}(p_{s,i} - r_{s,i})2}{N}} \tag{4}$$

Even though memory-based techniques are easy to implement and provide good recommendations, they encounter many issues such as sparsity, scalability and gray sheep problem which deteriorate the accuracy of predictions. In gray sheep cases, it is hard for a recommender system to find a dense neighborhood with a high number of similar users. In fact, in most cases computed similarities show a low degree of correlation, even negative correlation for some similarity measures like Pearson correlation coefficient.

3 Our Approach

The baseline collaborative filtering approach uses K-nearest neighbors to make new predictions. It relies on selecting useful users who have shown a high positive correlation to the active user. In most cases, computed similarities can be positive or negative that range from −1 to +1. Thus, users who have shown a negative correlation are not used in the prediction phase. In addition, in gray sheep cases, the active user seems to be lacking the reliable neighbors since most users have low or negative correlations. Figure 2 below presents an example of a gray sheep situation which occurs in memory-based CF process.

The basic idea behind our approach focuses on dealing with gray sheep problem and then enhancing the accuracy of predictions by increasing the size of the reliable neighborhood.

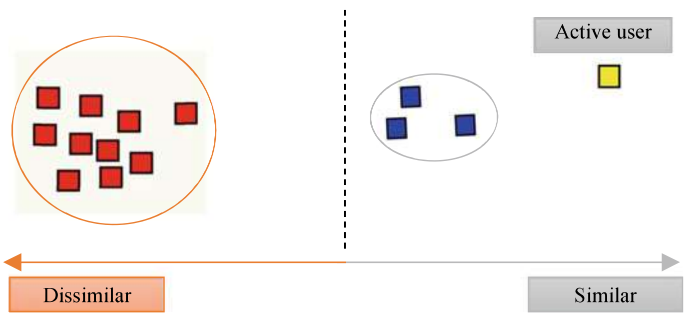

Fig. 2 Example of gray sheep situation in memory-based techniques

Fig. 3 Proposed memory-based CF process

This can be done by exploiting users who have shown a negative correlation or a dissimilarity to the active user in a smart way. To do so, we propose to infer new similar neighbors based on users who have different tastes and preferences for the active user. The underlying assumption of our approach is that if a user X has an opposite interest to a user Y, then, the user $\neg X$ will have the same interest as the user Y. Indeed, new fictive neighbors will be similar to the active user as their similarity values will be close to 1. Therefore, inferred users will enhance the density of the active user neighborhood. Consequently, additional insight will be provided to the recommender engine to make useful recommendations.

The new process includes an additional step (Fig. 3) before forming the active user neighborhood called Rating Matrix augmentation.

Rating matrix augmentation step consists in adding new lines in the rating matrix. Each line will represent an inferred user. He is the opposite user of a real one. This is achieved by deducing the opposite opinion of each rated item using the following formula:

$$\neg r_{\mathrm{aj}} = \mathrm{Max} - r_{\mathrm{aj}} + \mathrm{Min} \tag{5}$$

We denote R the $m \times n$ rating matrix where m is the number of users and n represents the number of items. The entry r_{aj} designates the rating of a user for an item j.

Max and Min represent, respectively, the high and the low value in a given numeric scale.

For instance, in a five-scale rating which ranges from 1 to 5, if a user a provided $r_{\mathrm{aj}} = 5$ as a rating for an item j, then, the inferred rating of user $\neg a$ for the item j will be $\neg r_{\mathrm{ai}} = 1$.

Figure 4 shows an example of opposite ratings on a 5-point scale using the previous formula. As presented, the number 3 has the same value after the opposite transformation. In fact, it represents a neutral opinion.

It lies in inferring ratings of opposite users by providing the opposite opinion on a given user.

Fig. 4 Example of an opposite rating matrix in a 5 point scale

Users \ Items	I_1	I_2	I_3	I_4	I_5	I_6	I_7
a	5		3		2		4
$\neg a$	**1**		**3**		**4**		**2**

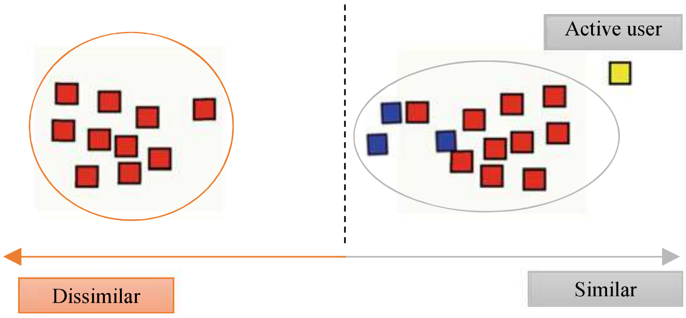

Fig. 5 Example of an active user neighborhood after users inference phase

Figure 5 shows an example of the expected output of the neighborhood formation step. As we can see, blue squares represent the new neighbors based on dissimilar users. The inferred users are likely to be very similar to the active user.

4 Experimentation and Results

We conducted several experiments using MovieLens and FilmTrust datasets to evaluate the effectiveness of our proposed approach. The main objective is to study the performance of the proposed approach over real-world datasets. In this section, we first present a brief description of the used datasets. Second, we present the evaluation procedure and the specification test environment. Then, we summarize our experimental results by comparing the performance of our proposed approach with a well-known baseline CF approach.

4.1 Datasets Collection

The experiments were performed on two commonly used datasets: MovieLens and FilmTrust [31]. Both are academic research projects of web-based movie recommender systems.

MovieLens is a five-point scale rating dataset that ranges from 1 (means bad) to 5 (means excellent). It consists of 1682 movies, 943 users and 100,000 ratings.

FilmTrust dataset consists of 1856 users, 2092 movies and 759,922 ratings. It was collected from a movie recommender systems website based on a social network which includes ratings and reviews. Ratings are numeric values on a 5-point scale between 0.5 and 4 stars.

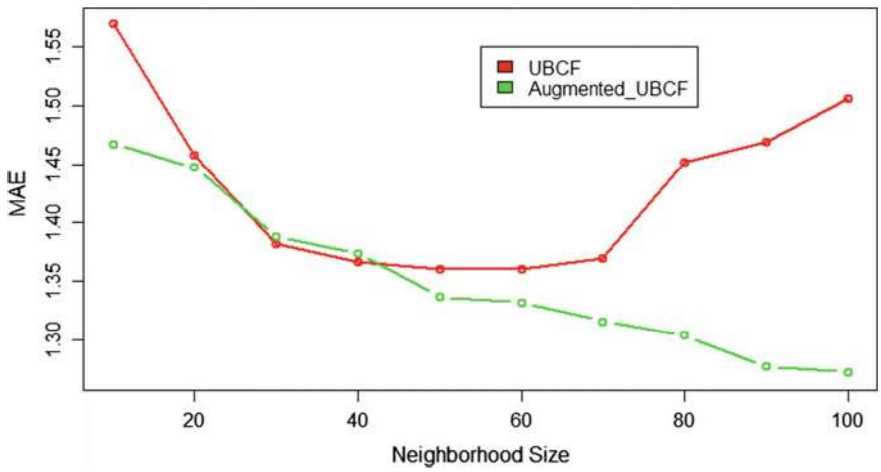

Fig. 6 MAE comparison using FilmTrust dataset

4.2 Experiments

To test our approach, we conducted a set of experiments using MovieLens and FilmTrust datasets. We reported the average results of a 10-fold cross-validation. We ran these experiments on a laptop computer with an Intel i5 at 2.4 GHz and 8 GB RAM.

Figures 6 and 7 show the obtained results of comparing the User-Based Collaborative Filtering approach (UBCF) as a baseline approach, and our proposed approach named Augmented UBCF for each dataset. The figures depict a comparison on MAE where the horizontal axis is the number of users in the neighborhood. It increases from 10 to 100 at an interval of 10. In Fig. 7, we can see the MAE of our approach and the baseline technique, are inversely proportional to the neighborhood size. We can see that our approach has lower MAE than the baseline approach. In Fig. 6 we see that our approach keeps a regular decreasing manner for the MAE while the baseline approach decreases until $N = 30$ then it remains stable until $N = 70$ where MAE starts increasing.

Overall, we can conclude that our approach provides better performance than the baseline approach in both datasets.

5 Conclusion and Perspectives

In this paper, we have proposed a novel collaborative filtering approach based on the opposite preferences of users. We focused on enhancing the accuracy of predictions and dealing with gray sheep problem. Our approach relies on inferring new similar

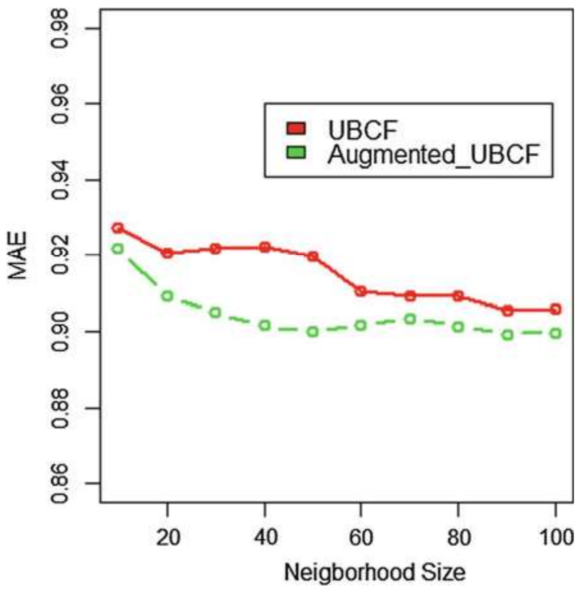

Fig. 7 MAE comparison using MovieLens dataset

neighbors based on users who have shown dissimilar tastes and preferences to the active user. In order to test our algorithm, we compare it with UBCF as a baseline approach. A set of experiments performed on two datasets including FilmTrust and MovieLens datasets show that our proposed approach has achieved good performance while solving gray sheep problem. As future work, we plan to investigate the effectiveness of hybridizing our approach with various machine learning techniques which seem to bring powerful insight to recommender systems.

References

1. Polatidis, N., Georgiadis, C.K.: A dynamic multi-level collaborative filtering method for improved recommendations. Comput. Stand. Interfaces **51**, 14–21 (2017)
2. F. Ortega, B. Zhu, J. Bobadilla, and A. Hernando, "Knowledge-Based Systems CF4J : Collaborative filtering for Java," *Knowledge-Based Syst.*, vol. 0, pp. 1–6, 2018
3. Gomez-uribe, C.A., Hunt, N.: The Netflix recommender system_algorithms. Bus. Value Pdf **6**(4) (2015)
4. Celma, O.: Music recommendation and discovery in the long tail. In: Citeulikeorg, pp. 252 (2008)
5. Callan, J., et al.: Personalisation and recommender systems in digital libraries joint NSF-EU DELOS working group report. Library (Lond) **5**(May), 299–308 (2003)
6. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)
7. Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2017)

8. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
9. Ekstrand, M.D.: Collaborative filtering recommender systems. Found. Trends® Human–Computer Interact. **4**(2), 81–173 (2011)
10. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. Constr. Build. Mater. **171**, 546–557 (2007)
11. Burke, R.: Hybrid recommender systems: survey and experiments. User Model. User Adap. Interact. **12**(4), 331–370 (2002)
12. Fu, M., Qu, H., Moges, D., Lu, L.: Attention based collaborative filtering. Neurocomputing **311**, 88–98 (2018)
13. Najafabadi, M.K., Mohamed, A., Onn, C.W.: An impact of time and item influencer in collaborative filtering recommendations using graph-based model. Inf. Process. Manag. **56**(3), 526–540 (2019)
14. Vozalis, E., Margaritis, K.G.: Analysis of recommender systems algorithms. In: 6th Hellenic European Conference Computer Mathematics & its Applications (HERCMA), vol. 2003, pp. 1–14. Athens, Greece (2003)
15. Jawaheer, G., Szomszor, M., Kostkova, P.: Comparison of implicit and explicit feedback from an online music recommendation service, pp. 47–51 (2006)
16. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. Gastroenterology **1**, S415 (2008)
17. Isinkaye, F.O., Folajimi, Y.O., Ojokoh, B.A.: Recommendation systems: principles, methods and evaluation. Egypt. Informatics J. **16**(3), 261–273 (2015)
18. Tsai, C.F., Hung, C.: Cluster ensembles in collaborative filtering recommendation. Appl. Soft Comput. J. **12**(4), 1417–1425 (2012)
19. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of KDD Cup Workshop, pp. 2–5 (2007)
20. Agrawal, S., Agrawal, J.: Survey on anomaly detection using data mining techniques. Procedia Comput. Sci. **60**(1), 708–713 (2015)
21. Breese, J., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Annual Conference on Uncertainty Artificial Intelligence, pp. 43–52 (1998)
22. Bhaidani, S.: Recommender system algorithms. In: Proceedings of International Conference on Weblogs Social Media ICWSM (2007, 2008)
23. Riedl, J., Sarwar, B., Karypis, G., Konstan, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of 10th International Conference on WorldWideWeb, pp. 285–295 (200)1