Improving Collaborative Filtering Approach by Leveraging Opposite Users

Abdellah El Fazziki[1(✉)], Yasser El Madani El Alami[2], Ouafae El Aissaoui[1], Youssouf El Allioui[3], and Mohammed Benbrahim[1]

[1] University of Sidi Mohammed Ben Abdellah, Fez, Morocco
{abdellah.elfazziki, ouafae.elaissaouil, mohammed.benbrahim}@usmba.ac.ma
[2] ENSIAS, University of Mohammed V, Rabat, Morocco
y.alami@um5s.net.ma
[3] FPK, University of Hassan Premier, Khouribga, Morocco
youssouf.elallioui@uhp.ac.ma

Abstract. Collaborative filtering is a widely used recommendation approach that aims to predict for a target user the most appropriate items. This approach uses the ratings given by users who share similar tastes and preferences to predict ratings for items that haven't been rated yet. Despite its simplicity and justifiability, CF approach stills suffering from several drawbacks and problems, including sparsity, gray sheep and scalability. These problems affect 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 ¬ 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 · Gray sheep · Opposite neighbors · Similarity measure

1 Introduction

Recommender systems (RS) are web decision support tools which help users to find useful items [1]. They aim to overcome the information overload problem by recommending suitable items based on users' preferences. RS behave as filters that enable passing the appropriate items to users and forbid the inappropriate ones [2]. Recommender systems are widely used in various domains including movies [3], music [4] libraries [5] and e-commerce [6, 7].

Since Tapersty [8] the first commercial recommender system, various approaches have been proposed to design and implement robust recommender systems. According

M. Ezziyyani (Ed.): AI2SD 2019, AISC 1102, pp. 136–145, 2020.
https://doi.org/10.1007/978-3-030-36653-7_14

to [9], recommender systems can be grouped into three main classes: collaborative filtering (CF) [10], content-based [11] and hybrid recommender systems [12, 13]. As results of its simplicity and justifiability, collaborative filtering remains the most commonly used approach on the web [14]. CF aims to recommend items based on users who share as similar tastes and preferences as the active user. Despite its strengths, CF experiences many problems which usually lead to deteriorating the accuracy of the predictions. For instance, Gray sheep is related to users who have unusual tastes and don't share similar preferences with other users [15]. Hence, finding a reliable neighborhood is a hard task. Scalability is another problem which 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 their explicit feedback. Actually, in most cases, users do not rate items in even though they feel an extreme emotion, either satisfaction or discontent [16].

In this paper, we focus to deal with the gray sheep problem and to improve the accuracy of recommendations based on opposite users. In other words, we generate new closer neighbors based on users who have shown different tastes and preferences to the active user. 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 providing accurate recommendations.

The rest of this paper is organized as follows:

In Sect. 2 we present an overview of the collaborative filtering 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 performed on several datasets. The conclusion and some perspectives are presented in Sect. 5.

2 Background

Thanks to its easiness and efficiency, collaborative filtering techniques are the most used approach in recommender systems [10]. CF assumes that useful suggestions can be provided by users who share similar tastes and preferences in the past. These preferences can be collected explicitly or implicitly. In the explicit case, users are asked to provide their opinions about an item using a rating scale [17]. For the implicit preferences, it can be inferred by observing users' actions such as the purchase history and the time spent on web contents [18]. Users, items, and ratings remain the basic input in collaborative filtering. The set of these triplets constitutes a matrix called rating matrix.

CF can be grouped in the two general categories of model-based and neighborhood-based. The former build predictive models from gathered ratings based on machine learning techniques like clustering [19], dimensionality reduction [20], support vector machines and neural networks [21]. The latter [22] is considered as the earliest CF algorithm. It consists in using ratings directly to produce recommendations. This is done using a similarity-based neighborhood for either users or items. User-based relies on building a neighborhood for active users in order to make predictions for unseen items. By the same reasoning, item-based predict ratings based on similar items. Both

use the K nearest neighbors' method to generate predictions. In the following, we present the user-based method.

2.1 Neighborhood-Based Recommendation Tasks

As reported by [23], the Neighborhood-based method consists of three steps presented in the figure (Fig. 1).

Fig. 1. Neighborhood-based CF process

2.1.1 Data Representation

Building the rating matrix is the first task in neighborhood CF. it relies on representing users, items and their ratings in a single matrix. In most cases, the rating matrix is usually sparse as users rate only a small part of the available items [20]. Depending on the used similarity metric, filling missing values can be done by using the average user's ratings or a null value.

2.1.2 Neighborhood Formation

This step consists in building a neighborhood of the most similar users based on a similarity measure. One of the most commonly used similarity metrics is the cosine similarity. It is computed by dividing the dot product of two users by the product of their magnitudes. The resulting values range from 0 to 1, where 0 means that there is no correlation between the two users, and 1 means that they are the same. Note that the cosine similarity measure deals with the lack of ratings by setting zero value. The similarity between two users a and b is calculated with the following formula:

$$sim_{a,b} = \cos(a,b) = \frac{ab}{\|a\|\|b\|} = \frac{\sum_{j=1}^{n} r_{aj} * r_{bj}}{\sqrt{\sum_{j=1}^{n} r_{aj}^2} * \sqrt{\sum_{j=1}^{n} r_{bj}^2}} \quad (1)$$

2.1.3 Predictions Generation

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

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

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

Therefore, based on the generated predictions, recommender systems provide top N recommendations as a list of items that the active user has never seen before.

2.2 Evaluation Metrics

In the literature, the performance of recommender systems is often measured using MAE (mean absolute error) and RMSE (root mean squared error). They remain as the commonly used performance metrics in the evaluation task. MAE has the advantage of being easy to interpret. As presented in the following formula, MAE calculates the average absolute differences between predicted ratings and actual values:

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

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 actual 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:

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

Even though Neighborhood-based techniques are easy to implement and provide good recommendations, they experience many problems such as sparsity, scalability and gray sheep problem which decrease the quality of recommendations. In gray sheep cases, it is a challenging task to find a reliable neighborhood with a high number of similar users. In fact, in most cases computed similarities show a low similarity, even negative correlation for some similarity measures like Pearson correlation coefficient and cosine similarity.

3 Our Approach

The baseline collaborative filtering approach uses K-nearest neighbors to make new predictions. It relies on using users who have shown a high similarity to the active user. Thus, users who have shown a low similarity 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 of them are distant. Figure 2 below presents an example of a gray sheep case which happens in Neighborhood-based CF process.

The principle idea behind our approach focuses on dealing with gray sheep problem and then enhancing the accuracy of predictions by increasing the number of reliable neighbors before starting the prediction phase.

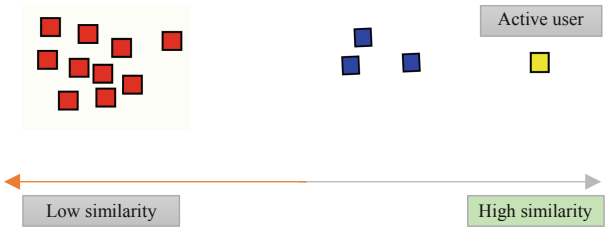

Fig. 2. Example of a gray sheep case in neighborhood-based techniques

This can be done by exploiting users who have shown a low similarity to the active user sagaciously. To do so, we propose to infer new neighbors based on users who have shown different tastes and preferences to 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 accurate recommendations.

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

Fig. 3. Proposed neighborhood based CF process

Matrix augmentation step consists in inferring new fictive users based on real ones. They are the opposite of available users in the rating matrix. This is achieved by inferring the opposite opinion of each rated item using the following formula:

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

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

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

Giving an example of a 5-scale rating which ranges from 1 to 5, if a user a gave $r_{aj} = 5$ as a rating for the item j, then, the inferred rating of user $\neg a$ will be $\neg r_{aj} = 1$.

Items / Users	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. 4. Example of an opposite user in a 5 point scale

Figure 4 shows an example of an opposite user in a 5-point scale. a represent the inferred user after applying the previous formula to the user a. It relies on inferring the ratings of an opposite user by providing the opposite opinion of each rating of a given user. As presented, the number 3 has the same value after the opposite transformation. In fact, it represents a neutral opinion.

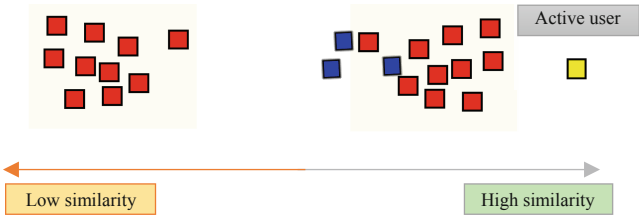

Fig. 5. Example of an active user neighborhood after users inference phase

Figure 5 shows an example of the expected result of the neighborhood formation step. As we can see, blue squares represent the new inferred neighbors formed by the matrix augmentation step. The inferred users are likely to be closer 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 collection of the 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 the User-based CF approach.

4.1 Datasets Collection

The experiments were conducted using two commonly used datasets: MovieLens and FilmTrust. Both are academic research projects of web-based movie recommender systems.

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

The FilmTrust dataset was collected from an academic movie recommender systems website based. It consists of 1856 users, 2092 movies and 759922 ratings. All ratings are stored as numeric values on a 5-point scale that ranges from 0.5 (means bad) to 5 (means excellent).

4.2 Experiments

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

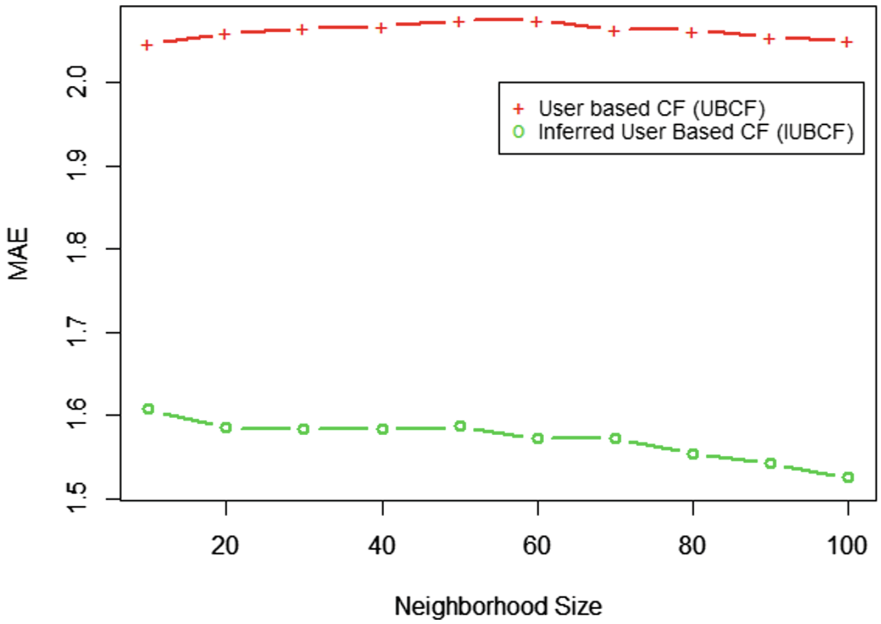

Fig. 6. MAE comparison using Filmtrust dataset

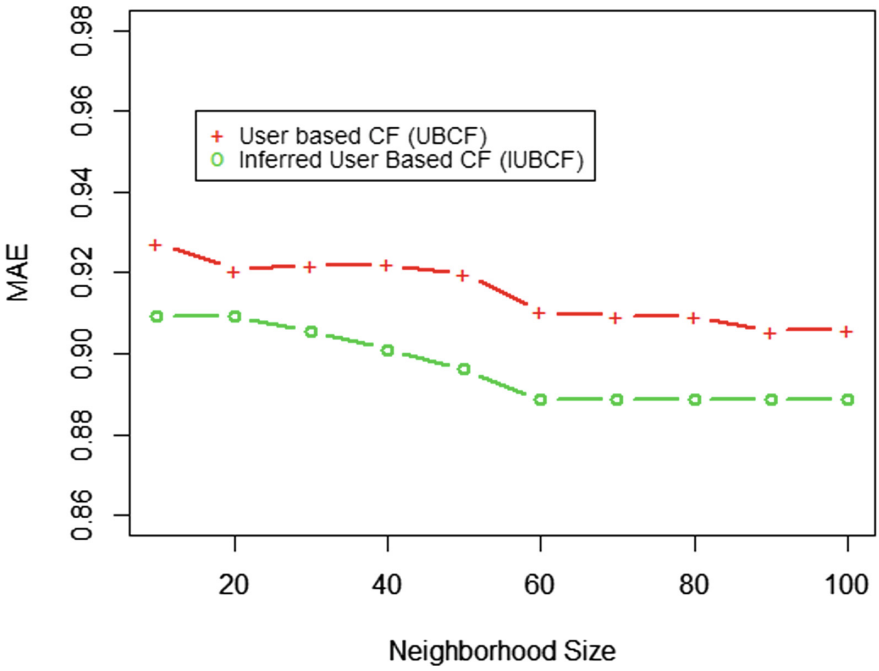

Fig. 7. MAE comparison using Movielens dataset

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 Inferred User Based Collaborative Filtering (IUBCF) for each dataset. The figures depict a comparison on MAE where the horizontal axis represents the neighborhood size in each experiment. It increases from 10 to 100 at the interval of 10. In Fig. 6 we see that our approach keeps a regular decreasing manner for the MAE while the baseline approach increases until N = 50 then it remains stable until N = 60 where the MAE starts decreasing. In Fig. 7, we can see the MAE of our approach and the baseline technique, are inversely proportional to the neighborhood size. They decrease with a regular manner until N = 60. Then they remain stable till N = 100. We can see that our approach has lower MAE than the baseline approach.

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 to improve the quality of recommendations and dealing with gray sheep problem. It relies on inferring new users with high similarities to the active user based on actual ones available in the recommender system. Each new fictive user is characterized by his opposite preferences of a real user. We focused on enhancing the accuracy of predictions and dealing

with gray sheep problem. Our approach relies on inferring new neighbors with high similarities based on users who have shown different tastes and preferences to the active user. In order to evaluate the effectiveness of our algorithm, we compare its performance with UBCF as a baseline approach. Experiments with FilmTrust and MovieLens datasets indicate that our proposed approach improves the performance of the accuracy of predictions while dealing with gray sheep problem. For future work, we attempt to investigate the hybridization our approach with various machine learning techniques, which seems to improve the accuracy of recommendations.

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. Ortega, F., Zhu, B., Bobadilla, J., Hernando, A.: Knowledge-base d systems CF4J : collaborative filtering for Java. Knowledge-Based Syst. **0**, 1–6 (2018)
3. Gomez-uribe, C.A., Hunt, N.: The Netflix Recommender System_Algorithms, Business Value.pdf. **6**(4) (2015)
4. Celma, O.: Music recommendation and discovery in the long tail. Citeulikeorg, p. 252 (2008)
5. Callan, J., et al.: Personalisation and recommender systems in digital libraries joint NSF-EU DELOS working group report. Libr. (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. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Commun. ACM **35**(12), 61–70 (1992)
9. 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)
10. Ekstrand, M.D.: Collaborative filtering recommender systems, found. Trends® Human–Comput Interact. **4**(2), 81–173 (2011)
11. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. Constr. Build. Mater. **171**, 546–557 (2007)
12. Burke, R.: Hybrid recommender systems: survey and user model. User Adapted Interact. **12** (4), 331–370 (2002)
13. El Alami, Y.E.M., Nfaoui, E.H., El Beqqali, O.: Toward an effective hybrid collaborative filtering : a new approach based on matrix factorization and heuristic-based neighborhood (2015)
14. Fu, M., Qu, H., Moges, D., Lu, L.: Attention based collaborative filtering. Neurocomputing **311**, 88–98 (2018)
15. 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)
16. E. Vozalis, K.G. Margaritis, Analysis of recommender systems algorithms. In: 6th Hellenic European Conference on Computer Mathematics and its Applications (HERCMA), vol. 2003, pp. 1–14. Athens, Greece (2003)

17. Jawaheer, G., Szomszor, M., Kostkova, P.: Comparison of Implicit and Explicit Feedback from an Online Music Recommendation Service, pp. 47–51
18. Hu,Y., Koren, Y., Volinsky, C.: Collaborative Filtering for Implicit Feedback Datasets. Gastroenterology. **1**, S415 (2008)
19. Tsai, C.F., Hung, C.: Cluster ensembles in collaborative filtering recommendation. Appl. Soft Comput. J. **12**(4), 1417–1425 (2012)
20. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of KDD Cup and Workshop, pp. 2–5 (2007)
21. Agrawal, S., Agrawal, J.: Survey on anomaly detection using data mining techniques. Procedia Comput. Sci. **60**(1), 708–713 (2015)
22. Breese, J., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial intelligence, pp. 43–52 (1998)
23. Bhaidani, S.: Recommender system algorithms. In: Proceedings of International Conference on weblogs and Social Media ICWSM 2007 (2008)