# A new collaborative approach to solve the gray-sheep users problem in recommender systems

Abdellah EL FAZZIKI
*University of Sidi Mohammed Ben Abdellah,* Fez, Morocco
abdellah.elfazziki@usmba.ac.ma

Ouafae EL AISSAOUI
*University of Sidi Mohammed Ben Abdellah,* Fez, Morocco
ouafae.elaissaouil@usmba.ac.ma

Yasser EL MADANI EL ALAMI
ENSIAS, University of Mohammed V
Rabat, Morocco
y.alami@um5s.net.ma

Youssouf EL ALLIOUI
LS3M, FPK, USMS University, B.P.: 145, 25000, Khouribga, Morocco
yelallioui@gmail.com

Mohammed BENBRAHIM
*University of Sidi Mohammed Ben Abdellah,* Fez, Morocco
mohammed.benbrahim@usmba.ac.ma

***Abstract*—Recommender systems aim to help users to find items that fit their requirements and preferences. In that field, the collaborative filtering (CF) approach is considered as a widely used one. There are two main approaches for CF: memory-based and model-based. Both of the two approaches are based on the use of users' ratings to predict the top-N recommendation for the active user. Despite its simplicity and efficiency, The CF approach stills suffer from many drawbacks including sparsity, gray sheep and scalability. The aim of this work is to deal with the gray sheep problem, by proposing a novel collaborative filtering approach. This novel approach aims to enhance the accuracy of prediction by turning the users whose preferences disagree with the target user, into new similar neighbors. For instance, if a user X is dissimilar to a user Y then the user ¬ X is similar to the user Y.**

**To evaluate the performance of the proposed approach, we have used two datasets including MovieLens and FilmTrust. The Experimental results show that our approach outperforms many traditional recommendation techniques.**

***Keywords—Recommender system, gray-sheep problem, Collaborative filtering, Opposite neighbors, Similarity Measure***

## I. INTRODUCTION

Recommender system are intelligent software that have the ability to recommend items to users by considering their preferences [1]. The main goal of recommender systems is to help users in such situations of information overload by providing them items that suit their requirements. Such systems aim to filter incoming streams of information by enabling passing the relevant ones to the user and blocking the irrelevant ones [2]. Recommender systems have been widely used in different domains such as movies [3], music [4] libraries [5] and e-commerce [6].

Due to the increasing use of recommender systems, many technical approaches to build such systems have been proposed. According to [7], there are three main types of recommender systems: collaborative filtering (CF) [8], content-based [9] and hybrid recommender systems [10]. According to [11], collaborative filtering is the most widely applied approach on the web, while the other approaches are less frequently employed.

CF aims to predict for an unrated item the rating than might be given by the target user, based on the users' ratings. There are two main approaches for CF: memory-based and model-based, where the memory-based approaches are often classified into two main categories, the user-based and the Item-based, where the first one is based on the users who share the same preferences, while the second one is based on the items that are most similar to the target item.

Despite its advantages, CF stills face many challenges and problems that affect the accuracy of recommendations. Among those challenges, there is the Gray sheep which refers to users who have unusual tastes and don't share similar preferences with other users [12]. Therefore, it is so hard to find neighbors. Scalability is another major issue that appears when the size of the data set being enormously huge. It becomes difficult to compute similarities when more and more users and items are added to the database. Also, the sparsity problem occurs when users are very active, but they don't rate the available items. Thus, the user-item matrix is extremely sparse, which can decrease the accuracy, since the similarity measures is computed based on the ratings given by users [13].

In this paper, we address the problem of grey sheep associated with CF by proposing a novel approach that aims to find fictive neighbors for a given user, in order to ameliorate the prediction accuracy. The principle consists in converting the users whose tastes differ from the target user, into neighbors. The underlying assumption of our approach is that if a user X has an opposite opinion of a user Y, then, the user ¬X has the same opinion as the user Y. Our approach will increase the number of similar neighbors, which makes the similarity measures results more significant and then improve the prediction efficiency.

The rest of this paper is organized as follows: Section 2 gives an overview of the collaborative filtering baseline approach. Section 3 describes our proposed approach and the original contribution of this work. The experiments and results are presented in section 4; Finally, Section 5 presents our conclusions and future works.

## II. *BACKGROUND*

CF techniques aim to predict items for a user based on a set of preferred items that were previously rated by other users. There are two types of ratings. Either the users are explicitly asked to rate items using a rating scale [14], or the ratings can be inferred implicitly by analyzing the user's behaviors while using a website such as the history of purchases [15]. The users' ratings are gathered in a matrix called the rating matrix which consists of a table where each row represents a user, each column represents a specific item, and the number at the intersection of a row and a column represents the user's rating value. This matrix is the basic input in collaborative filtering used to build effective prediction models and users' profiles [16].

978-1-7281-0003-6/19/$31.00 ©2019 IEEE

CF algorithms are categorized into two main approaches: model-based and memory-based techniques. The first one aims to build a model based on a training dataset of ratings, and then use that model to predict unobserved ratings and make recommendations. There are many machine learning algorithms that can be used to build a model such as, clustering techniques [17], dimensionality reduction methods [18], support vector machines, neural networks [19]. Memory-based CF is considered as the earliest CF approach [20]. This approach utilizes the entire rating matrix to generate a prediction. In Memory-based CF there are two main algorithms: the item-based CF and the user-based CF. Both of them aim to select the k-nearest neighbors using a similarity measure, and based on the selected neighbors the prediction can be computed. In what follows, we will focus our study on the user-based approach.

### A. *User-based approqch Recommendation tasks*

The user based approach requires three mains steps as presented in the below figure:


Fig. 1. Memory-Based CF process

#### 1) *Data representation*

To build a recommender system using a user based approach, it is required firstly to create a user-item rating matrix, where rows represent the users and columns the items and the intersections between them represent the ratings. In many cases, the users don't rate items regularly, which causes the sparsity problem. One way to solve this problem is by filling the missing values with the average user's ratings.

#### 2) *Neighborhood formation*

In user-based approaches, the prediction of a user rating on an item is based on the ratings given to that item by the nearest neighbors. So a set of nearest neighbors has to be selected using a similarity measure between users [21]. One commonly used similarity metric is the Pearson correlation coefficient that can be computed using the following formula :

$$sim_{a,b} = \frac{\sum_{j=1}^{n}(r_{aj}-\overline{r_a})(r_{bj}-\overline{r_b})}{\sqrt{\sum_{j=1}^{n}(r_{aj}-\overline{r_a})^2 \sum_{j=1}^{n}(r_{bj}-\overline{r_b})^2}} \quad (1)$$

#### 3) *Predictions generation*

After selecting the k-nearest neighbors who have already rated the target item, their ratings can be used by a prediction function to compute the rating value that can be given by the target user. The prediction can be generated using the following formula:

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

The above function can be used to compute the predicted ratings for all items that have not yet been seen by a user, and based on the prediction values the Top-N recommended items can be determined. As can be noticed, the prediction function uses the KNN technique where K represents the number of closest neighbors.

### B. *Evaluation metrics*

After building any prediction model, it is necessary to evaluate its performance in making predictions. In recommender systems, there are two commonly used evaluation metrics: the MAE (mean absolute error) and the RMSE (root mean squared error). The MAE is a measure of average absolute differences between observed and predicted ratings. The lower the MAE, the better the model is.

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

The Root Mean Squared Error (RMSE) measures the average magnitude error made by the prediction function while predicting a user's rating. It's the square root of the average of squared residuals. Residuals are the difference between the actual ratings and the predicted ratings. The lower the RMSE, the better the model.

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

From the above-described steps, we can notice that the user-based approach is so easy to implement and give good recommendations, but despite these advantages, there are many drawbacks within this approach that affect its efficiency such as sparsity, scalability and the gray sheep problem. In the gray sheep problem, it is a hard task to find neighbors for a user whose taste is unique. In this case, the results obtained from the similarity measure function show a very low correlation. Thus the prediction can't be done. In the next section, we will present a proposed approach to solve this problem.

## III. *Our Approach*

As mentioned in the previous section, the user-based approach uses a similarity measure to compute similarities between users and then select the nearest neighbors for a given user. The computed similarities can be positive or negative. The users whose similarities are positive are well correlated with the target user, therefore, their ratings can be used to make a prediction. In the gray sheep case, it is difficult to select neighbors and make a prediction, since most users have low or negative correlations. The figure 2 below presents an example of a gray sheep situation which occurs in user-based approach

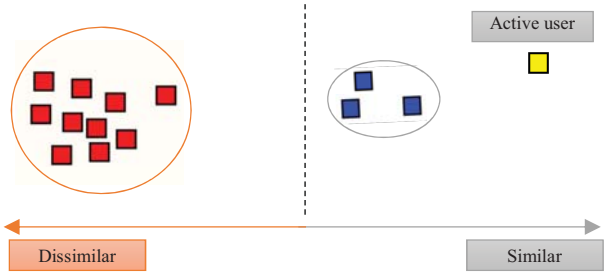


Fig. 2. Example of gray sheep situation in user-based techniques

The purpose of this work is to deal with the gray sheep problem by proposing an approach that aims to increase the size of the neighbors to use their ratings while computing the prediction function and then improving the recommendations.

The idea of our approach is to exploit the users whose preferences are different from the target user, by turning them into neighbors. This idea will help to infer new fictive

neighbors whose similarity values with the active user are 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 following figure (figure 3) presents our proposed approach that includes an additional step called Rating Matrix augmentation that comes before determining the active user's neighbors.

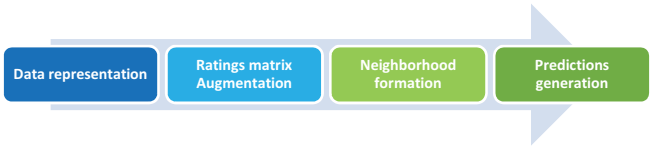


Fig. 3. Proposed Memory Based CF Process

Rating matrix augmentation step aims to add new rows in the rating matrix. Each row represents a new inferred user whose ratings are opposite to the real one. The opposite ratings can be deducted using the following formula:

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 $raj$ refers to the rating given by user a for an item j.

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

For example, in a 5-scale rating which ranges from 1 to 5, if a user a provided $raj = 5$ as a rating for an item j, then, the inferred rating of user $\neg a$ for the item j will be $\neg rai = 1$.

TABLE I. EXAMPLE OF AN OPPOSITE RATING MATRIX IN A 5 POINT SCALE

| Users | Items | | | | | | |
|---|---|---|---|---|---|---|---|
| | *i1* | *i2* | *i3* | *i4* | *i5* | *i6* | *i7* |
| *a* | 5 | | 3 | | 2 | | 4 |
| ¬*a* | 1 | | 3 | | 4 | | 2 |

The above table shows an example of opposite ratings on a 5-point scale using the previous formula. As can be noticed, the number 3 doesn't change after applying the formula. In fact, it represents a neutral rating.

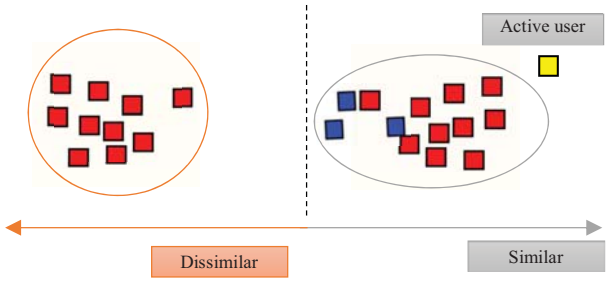


Fig. 4. Example of an active user neighborhood after users inference phase

The above figure shows an example of the fictive neighbors that can be resulted after applying the neighborhood formation step on the fictive users. As we can ntice, red squares represent the new fictive neighbors. The inferred neighbors are likely to be positively correlated with the target user.

## IV. EXPERIMENTATION AND RESULTS

To evaluate the performance of our proposed approach we used the MovieLens and FilmTrust datasets. The main objective is to compare the performance of the proposed approach with the User Based Collaborative Filtering approach using 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 the results of the comparison study are presented to determine the most performant approach.

### A. Datasets collection

The experiments were performed on 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 consists of 1682 movies, 943 users and 100,000 ratings.

FilmTrust dataset consists of 1856 users, 2092 movies and 759922 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.

### B. Experiments

To evaluate our approach, we did several experiments using MovieLens and FilmTrust datasets. To compute the validation metrics we used a 10-fold cross-validation technic. We launched these experiments on a laptop computer with an Intel i5 at 2.4GHz and 8 GB RAM.

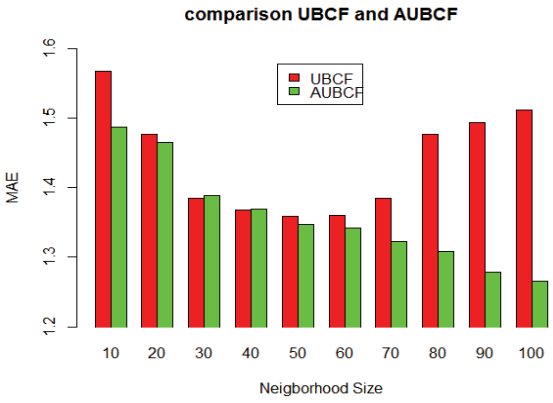


Fig. 5. MAE comparison using FilmTrust dataset

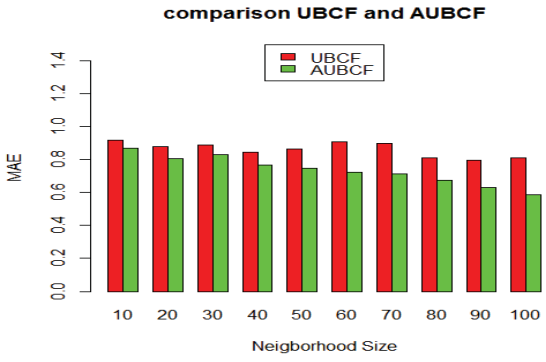


Fig. 6. MAE comparison using MovieLens dataset

Figures 6 and 7 display for each dataset, the results obtained after comparing our proposed approach named Augmented User-Based Collaborative Filtering approach (AUBCF) and the UBCF. 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 the interval of 10. In figure 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 figure 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.

## V. CONCLUSION & PERSPECTIVES

Collaborative filtering is a widely used approach in recommender systems that stills suffer from many drawbacks, including the gray sheep problem. In this paper, we have proposed a novel collaborative filtering approach to deal with that problem. The proposed approach aims to infer new fictive neighbors for the active user based on users who have shown dissimilar tastes and preferences. In order to test our algorithm, we compared it with UBCF as a baseline approach. The comparison was done based on two datasets including FilmTrust and MovieLens. The obtained results showed that our approach outperforms the UBCF and improves the accuracy of predictions in the case of gray sheep problems. In future work, we would like to investigate the hybridization of our approach with various machine learning techniques, to enhance the accuracy of recommendations.

## REFERENCES


[1] N. Polatidis and C. K. Georgiadis, "A dynamic multi-level collaborative filtering method for improved recommendations," Comput. Stand. Interfaces, 2017.

[2] F. Ortega, B. Zhu, J. Bobadilla, and A. Hernando, "CF4J: Collaborative filtering for Java," Knowledge-Based Syst., 2018.

[3] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," ACM Trans. Manag. Inf. Syst., 2015.

[4] Ò.` Celma and P. Lamere, "Music recommendation and discovery revisited," in RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems, 2011.

[5] J. Callan and A. F. Smeaton, "Personalisation and recommender systems in digital libraries," Int. J. Digit. Libr., 2005.

[6] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," IEEE Internet Comput., 2003.

[7] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering. 2005.

[8] Sangeeta and N. Duhan, "Collaborative filtering-based recommender system," in Advances in Intelligent Systems and Computing, 2018.

[9] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2007.

[10] V. Vekariya and G. R. Kulkarni, "Hybrid recommender systems: Survey and experiments," in 2012 2nd International Conference on Digital Information and Communication Technology and its Applications, DICTAP 2012, 2012.

[11] M. Fu, H. Qu, D. Moges, and L. Lu, "Attention based collaborative filtering," Neurocomputing, vol. 311, pp. 88–98, Oct. 2018.

[12] M. K. Najafabadi, A. Mohamed, and C. W. Onn, "An impact of time and item influencer in collaborative filtering recommendations using graph-based model," Inf. Process. Manag., 2019.

[13] E. Vozalis and K. Margaritis, "Analysis of Recommender Systems' Algorithms," Hercma, 2003.

[14] G. Jawaheer, M. Szomszor, and P. Kostkova, "Comparison of implicit and explicit feedback from an online music recommendation service," in Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec 2010, Held at the 4th ACM Conference on Recommender Systems, RecSys 2010, 2010.

[15] Y. Hu, C. Volinsky, and Y. Koren, "Collaborative filtering for implicit feedback datasets," in Proceedings - IEEE International Conference on Data Mining, ICDM, 2008.

[16] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," Egyptian Informatics Journal. 2015.

[17] C. F. Tsai and C. Hung, "Cluster ensembles in collaborative filtering recommendation," Appl. Soft Comput. J., 2012.

[18] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," KDD Cup Work., 2007.

[19] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," in Procedia Computer Science, 2015.

[20] G. F. Cooper, S. Moral, P. P. (Prakash P. . Shenoy, and W. . Conference on Uncertainty in Artificial Intelligence (14th : 1998 : Madison, Empirical analysis of predictive algorithms for collaborative filtering. Morgan Kaufmann Publishers, 1998.

[21] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th International Conference on World Wide Web, WWW 2001, 2001.