Collaborative Approach to minimize the negative impact between Gray Sheep users and non-Gray Sheep users

Tatsunori Nagashima
George Washington University

***Abstract*—Recommender systems are systems used by many platforms to enhance users' decision abilities. One of the algorithm which are widely used for these systems is Collaborative Filtering(CF) algorithm. CF algorithm(user-based) uses similarity of users who have the common preferences to compute recommendations. There are some drawbacks to this algorithm including Gray Sheep(GS) problem, scalability, and sparsity. The aim of this paper is to propose a new approach which will resolve GS problem. This approach's main idea is to reduce noise created by GS users by placing some weights to reduce influence from GS users to non-GS users and influence from non-GS users to GS users. Since this approach contains several hyperparameters that can be adjusted, controlled experiments will be done to evaluate how each hyperparameters affect the result. The experimental results show that this approach will outperform the traditional CF with certain hyperparameter combinations but perform worse on some hyperparameter combinations.**

I. Introduction

In modern days, people regularly choose what to consume based on online recommendations. This includes choosing videos recommended from an online platform such as YouTube and Netflix, choosing products to buy from Amazon and eBay, and viewing recommended social media pages from Instagram and Twitter. The major goal to these recommendation systems is to provide the most relevant information to users without overloading users with unnecessary information. Some major algorithms used in implementing recommendation systems include content-based filtering, collaborative filtering, and hybrid systems. Among these algorithms, collaborative filtering(CF) algorithm is the most popular algorithm used[3].

There are two types of CF algorithm and these are called memory-based and model-based. Memory-based is further categorized into two categories: user-based and item-based. User-based CF recommendation technique is based on a user's past behavior and other users' preferences. In a regular CF technique, there will be a list of n users u_1, u_2, ... , u_n and a list of m items i_1, i_2, ... , i_m, and each user, u_i has a list of items, I_{ui} where $I < m$, which has rating information which can be collected either explicitly or implicitly[7]. This user and item relationship will be represented in a matrix and used to recommend users an appropriate item based on collaborating the user-item matrix with other users. Because CF technique does not need to have item information such as genres, CF technique will not require space for item information and CF technique can provide recommendations with genres that are different from what a user will typically choose.

However, this technique also has several disadvantages, and one of them is gray sheep(GS) problem. GS users are defined as users who have unusual tastes and do not share similar preferences with other users. This is problematic because collaborating the matrix with these GS users can cause a noise affecting negatively on both GS users and non-GS users. Another problem is scalability. Because CF algorithms has a minimum computation cost of $O((u * i)^2)$ where u is a user and i is an item, computation of the recommendation can cause serious burden to a system when there are more than millions of users and more than tens of millions of items. Having to use the user-item matrix also creates a problem of sparsity. This problem appears when there are many items which are not rated by users leading for a system to have empty data, and this can cause the recommendation system to give inaccurate recommendations.

In this paper, I will propose a new CF approach which addresses the GS problem by placing some weights which will reduce influence from GS users to non-GS users and influence from non-GS users to GS users. This approach's purpose is to not completely neglect influence from different kind of users but reduce influence to the point it is positively affecting both kind of users. This proposal should increase the effectiveness of the utilization of user similarities and accuracy of the recommendation system.

This paper will be followed with 4 major sections. In Section 2, the paper will go over GS problem in more detail and show other approaches done to GS problem by different scholars. In Section 3, the paper will give further explanation to the proposed algorithm. In Section 4, the paper will provide a showcase result of this algorithm and compare them with the traditional CF algorithm. Lastly, in Section 5, the paper will give conclusions and possible plans on future works.

II. Background

Gray Sheep(GS) problem is one of the recent problems raised to improve the CF recommendation algorithm performance. GS users can be identified as a user who

has different tastes from most of the users and have a low correlation coefficient with those users. Note that this is different from Black Sheep users or White Sheep users where Black Sheep users are the group of users who have the opposite tastes from most users and White Sheep users are the group of users who have similar tastes with most users and have high correlation coefficient with those users. Various researches[1,4,6] have stated that GS users must be identified to avoid problems that include:

- Possible negative impact on the quality of recommendation system caused by unusual tastes which GS users have. Regular users might get odd recommendations which they would not be seeing without GS users.

- CF algorithms do not work well with GS users. Because GS users have different tastes from most users, the level of satisfaction will not be as high as most of the users. Content-based algorithms are often recommended to increase the level of satisfaction of GS users.

- Low level of recommendation quality created by not identifying GS users may lead to harmful consequences to services including unsatisfied users, inaccurate marketing, and failure of addressing possible problems. Unlike Black Sheep users, the number of GS users are high enough to not be ignored when designing the CF algorithm. If left ignored, recommendations for both non-GS users and GS users will be highly affected, and this may lead to a significant decrease in performance of recommendations.

Some major works on improving the CF algorithm that handles GS users can be seen in the form of hybrid recommender system, which is a system that uses both the CF and content-based algorithm. Balabonovic and Shoham[2] have proposed the recommender system, Fab, which uses CF techniques to identify similar preferences between users using user profiles of preferences obtained in web-pages with content-based techniques. Tennakoon[8] has taken different approach. Datasets will be first categorized and tagged based on categories. The data will then be assigned a weight. Once data is passed into a recommendation algorithm, the algorithm will run the CF algorithm, and when CF identifies a user as a GS user, it will move a user to content-based algorithm.

There are also approaches which are not necessarily using hybrid recommender system. One example of this will be the CF matrix augmentation by Fazziki and others[2]. This approach tackles the GS problem by resolving the lack of GS neighbors, and this is done by increasing the size of GS neighbors where the fictive GS neighbors are created by inverting the rating of non-GS users. Another example will be the outlier detection done by Yong[9]. Yong proposed to use a distribution matrix which contains statistical information such as q1, q2, q3, mean, and standard deviation with the local outlier factor algorithm to identify GS users.

III. Approach Taken

As mentioned above, GS users have low correlation with other users. To obtain these correlation, I will be using cosine similarity.

$$similarity = \frac{A * B}{||A|| * ||B||} \tag{1}$$

Users who have higher value of cosine similarity will indicate higher correlation with other users and users who have lower value of cosine similarity will indicate lower correlation with other users. Using this property, we can identify GS users. If a user has many other users who share high value of cosine similarity, the user will be non-GS users and if a user has low amount of other users who share high value of cosine similarity, the user will be a GS user.

The following figure(Figure 1) shows steps to execute proposed algorithm.

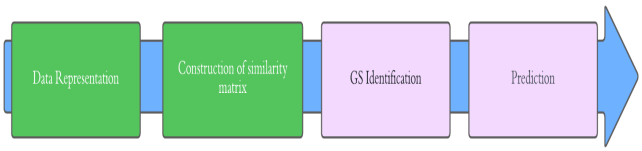

Fig. 1. Proposed CF approach

The first step is formatting data in a proper manner. This will usually be in user-item matrix format where each user or item is represented in the appropriate row or column and holds a value of evaluation.

The second step will be constructing a similarity matrix, and this is done using cosine similarity equation shown in equation 1 and respective values of evaluation from the first step.

The third step will be identifying GS users, and this step involves two hyperparameters: GS-ratio threshold and similarity threshold. GS-ratio threshold is a parameter which indicates proportion of GS users to be assigned. For example, if GS-ratio is set as 0.1, 10% of users will be assigned as GS users. Similarity threshold is a parameter which indicates the minimum expected value of “high” similarity. In other words, if a user shares similarity value to another user that is higher than the threshold, that similarity will be considered as “high” similarity. GS users are identified by taking the number of users specified by GS-ratio who has the lowest count of “high” similarities.

The last step will be making prediction using the information from previous steps, and the following equation will be used.

$$P_{ms} = \sum_{n \in N} \frac{r_n * constraint(sim_{sn} - abs(sign_s - sign_n) * i)}{constraint(sim_{sn} - abs(sign_s - sign_n) * i)} \quad (2)$$

where P_{ms} represents the prediction made for the specific movie m, and the specific individual s. N is the top k-nearest neighbor of user s and n is one of the neighbors from N. sim_{sn} denotes similarity of user s and user n. $sign_s$ and $sign_n$ contains label value to the respective user where -0.5 represents GS user and 0.5 represents non-GS user. The subtraction of these label values will determine how the influence, i will affect the weight of particular combinations of users. $constraint$ is a function to prevent similarity weight to go over 1 or go below 0.

IV. Experimentation and Results

To prove the credibility of proposed algorithm, MovieLens dataset[5] will be used. The prediction of proposed algorithm will be compared against traditional user-based CF algorithm.

A. Dataset collection

MovieLens dataset is widely used in academic research and projects for collaborative filtering topic. This dataset contains 943 users and 1682 Movies with 100,000 ratings where ratings are ranged from 1 to 5 where 1 indicates the worst and 5 indicates the best. This dataset also have a constraint where each user has at least rated 20 movies.

B. Evaluation metrics

When evaluating a prediction model, choosing evaluation metrics is one of the important step to validate the model. In this experiment, I will be using the Mean Absolute Error(MAE) which is commonly used in papers of similar topic.

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

Lower MAE value will indicate more effective model and higher MAE value will indicate less effective model.

C. Experiments

To reduce as much deviation as possible, experiments were done using n-fold cross-validation for $round(150/n)$ iterations and averaged out [1]. Since this algorithm contains numerous hyperparameters, base parameters were first determined and one of hyperparameters were modified to do controlled experimentation. Base parameters will consist of the following:

- k-nearest neighbor(neighborhood size): 25
- influence: 0.2
- GS-ratio threshold: 0.2
- Similarity threshold: 0.5
- n: 25

Note that n in n-fold cross-validation is also included as a parameter to be controlled. This is to examine how training-test set ratio will affect accuracy of the recommendation system. The following figures show experiment results obtained using different controlled variables.

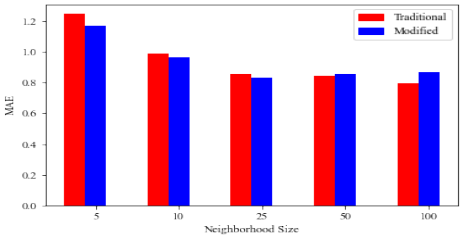

Fig. 2. Modified CF vs CF on change in neighborhood size

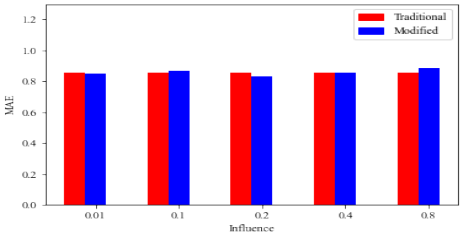

Fig. 3. Modified CF vs CF on change in influence

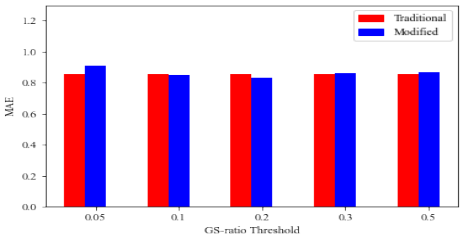

Fig. 4. Modified CF vs CF on change in GS-ratio threshold

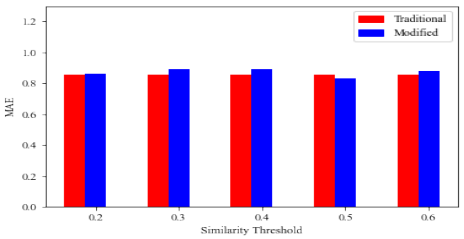

Fig. 5. Modified CF vs CF on change in similarity threshold

[1]Training set and testing set will be randomly shuffled after each iteration

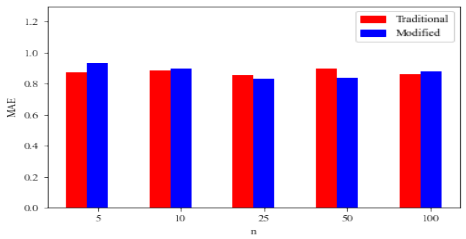

Fig. 6. Modified CF vs CF on change in n-cross validation

D. Analysis

There are several key features of these figures that should be discussed. The first feature is that the proposed algorithm is outperformed by the traditional CF algorithm for most cases. This is especially apparent in hyperparameters including influence, GS-ratio threshold and similarity threshold. However, there is also at least one set of hyperparameters which outperforms the traditional CF algorithm. This indicates that tuning would be essential to apply the proposed algorithm.

Another feature which can be noticed is small differences in performance between the traditional CF and proposed algorithm. This is also more apparent in figures that have influence, GS-ratio threshold or similarity threshold as controlled variable. A common characteristic among these hyperparameters is that these hyperparameters are used to find how much influence GS users can have on non-GS users or vice-versa. There are two major possible causes, and one of them is the fact that dataset that has being used is relatively small. While MovieLens dataset is widely used in this field, dataset that has been used is usualy the 20M dataset and not the 100K dataset which I have used. This reduction in a number of data might have caused a decrease in a number of GS users to the point that effect GS users have on non-GS users to be negligible. Another major cause would be misclassifying a Black Sheep user, which is a user who have completely opposite tastes to most of the users, as a GS user. Because a Black Sheep user is usually not used in calculation of recommendation score because of its distance from most of the users, influence from a Black Sheep user will not be apparent.

Last feature that can be seen is a clear pattern in figure 2 and figure 6. Figure 1 has a linear relationship on a difference between the traditional CF algorithm and the proposed algorithm, and Figure 6 has a quadratic relation on the difference. Exploiting this pattern can lead to effective hyperparameters or parameters selection.

V. Conclusion

While the traditional CF algorithm is widely used in recommendation systems, they still suffer from drawbacks including GS problem. In this paper, I proposed the new approach which aims to reduce influence from a GS-user to a non-GS user and a GS-user to a non-GS user. Controlled experiments are done to examine how each parameters affect the recommendation score, and I found out that it is difficult to find the most effective parameters for some hyperparameters.

Reflecting on the experimentation results, my future work will consist of the following:

- Employing MovieLens 20M dataset and other similar datasets to check if the proposed algorithm's cause of ineffectiveness is due to the dataset selection.

- Revising GS user identification. This can include changing the identification method from identifying a certain ratio of users with lower "high" similarities as GS-user to identifying a user who has "high" similarities lower than a specific count as GS-user.

- Develop an effective way of tuning hyperparameters. This can include utilization of combinatorial optimization algorithm including simulated annealing.

References

[1] Ansari, A., S. Essegaier, and R. Kohli. 2000. "Internet recommendation systems," *Journal of Marketing Research*, vol. 37, no. 3, pp. 363–375, 2000.

[2] Balabanović, M. and Y. Shoham. 1997. "Content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[3] Fazziki, A. E., O. E. Aissaoui, Y. E. M. E. Alami, et al. 2019. "A new collaborative approach to solve the gray-sheep users problem in recommender systems", *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, 2019, pp.1-4

[4] Gras, B., A. Brun, and A. Boyer. 2016. "Identifying Grey Sheep Users in Collaborative Filtering: a Distribution-Based Technique", *Proceeding of 2016 Conference on User Model Adaptation and Personalization*, pp.17-26. DOI: http://dx.doi.org/10.1145/2930238.2930242

[5] Maxwell H. F. and Konstan A. J. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872

[6] Melville, P., R. J. Mooney, and R. Nagarajan. 2002. "Content- boosted collaborative filtering for improved recommendations," in *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI '02)*, pp. 187–192, Edmonton, Canada, 2002.

[7] Su, X. and T. M. Khoshgoftaar. 2009. "A Survey of Collaborative Filtering Techniques", Hindawi Publishing Corporation, London, UK

[8] Tennakoon, A., N. Gamlath, G. Kirindage, et al. 2020. "Hybrid Recommender for Condensed Sinhala News with Grey Sheep User Identification", *2020 2nd International Conference on Advancements in Computing (ICAC)*, 2020, pp.228-233

[9] Zheng, Y., M. Agnani, and M. Singh. 2017. "Identifying Grey Sheep Users By the Distribution of User Similarities In Collaborative Filtering", *RIIT*, Rochester, NY