Identification of Grey Sheep Users by Histogram Intersection in Recommender Systems

Yong Zheng(✉), Mayur Agnani, and Mili Singh

School of Applied Technology, Illinois Institute of Technology,
Chicago, IL 60616, USA
yzheng66@iit.edu, {magnani,msingh32}@hawk.iit.edu

Abstract. Collaborative filtering, as one of the most popular recommendation algorithms, has been well developed in the area of recommender systems. However, one of the classical challenges in collaborative filtering, the problem of "Grey Sheep" user, is still under investigation. "Grey Sheep" users is a group of the users who may neither agree nor disagree with the majority of the users. They may introduce difficulties to produce accurate collaborative recommendations. In this paper, discuss the drawbacks in the approach that can identify the Grey Sheep users by reusing the outlier detection techniques based on the distribution of user-user similarities. We propose to alleviate these drawbacks and improve the identification of Grey Sheep users by using histogram intersection to better produce the user-user similarities. Our experimental results based on the MovieLens 100 K rating data demonstrate the ease and effectiveness of our proposed approach in comparison with existing approaches to identify grey sheep users.

Keywords: Recommender system · Collaborative filtering · Grey sheep

1 Introduction

Recommender system is well-known to assist user's decision making by recommending a list of appropriate items to the end users tailored to their preferences. Several recommender systems have been developed to provide accurate item recommendations. There are three types of these algorithms: collaborative filtering approaches, content-based recommendation algorithms and the hybrid recommendation models [4]. Collaborative filtering (CF) is one of the most popular algorithms since it is effective and it does not rely on any content information.

Most of the efforts on the development of CF algorithms focus on the effectiveness of the recommendations, while far too little attention has been paid to the problem of "Grey Sheep" users which is one of the classical challenges in collaborative filtering. "Grey Sheep" (GS) users [6,11] is a group of the users who may neither agree nor disagree with the majority of the users. They may introduce difficulties to produce accurate collaborative recommendations. Therefore, it has been pointed out that GS users must be identified from the data and treated individually for these reasons:

© Springer International Publishing AG 2017
G. Cong et al. (Eds.): ADMA 2017, LNAI 10604, pp. 148–161, 2017.
https://doi.org/10.1007/978-3-319-69179-4_11

- They may leave negative impact on the quality of recommendations for the users [6–9,11,13,14] in the collaborative filtering algorithms.
- Collaborative filtering approaches do not work well for GS users [6–9,13]. GS users should be treated separately with another type of the recommendation models, such as content-based approaches.
- Due to the presence of GS users, the poor recommendations may result in critical consequences [6,9,11]: unsatisfied users, user defection, failure among learners, inaccurate marketing or advertising strategies, etc.

Most recently, we propose a novel approach to identify the GS users by the distribution of user-user similarities in the collaborative filtering approach [15]. However, one of the drawbacks in this approach is that the user-user similarity cannot be measured if two users did not rate the same items. Also, the user-user similarities may not be reliable if the number of co-rated items by two users is limited. In this paper, we propose an improved approach to alleviate this problem. More specifically, we propose to utilize histogram intersection to re-produce the distribution of user-user similarities.

Our contributions in this paper can be listed as follows:

- Our proposed approach improves the identification of GS users by the distribution of user-user similarities.
- It is the first time to compare different methods of identifying the GS users. The proposed approach in this paper was demonstrated as the best performing one based on the MovieLens 100 K rating data set.

2 Related Work

In this section, we introduce collaborative filtering first, discuss the characteristics of GS users, and finally introduce the corresponding progress of identifying the GS users.

2.1 Collaborative Filtering

Rating prediction is a common task in the recommender systems. Take the movie rating data shown in Table 1 for example, there are four users and four movies. The values in the data matrix represent users' rating on corresponding movies. We have the knowledge about how the four users rate these movies. And we'd like to learn from the knowledge and predict how the user U_4 will rate the movie "Harry Potter 7".

Collaborative filtering [11,14] is one of the most popular and classical recommendation algorithms. There are memory-based collaborative filtering, such as the user-based collaborative filtering (UBCF) [12], and model-based collaborative filtering, such as matrix factorization. In this paper, we focus on the UBCF since it suffers from the problem of GS users seriously.

The assumption in UBCF is that a user's rating on one movie is similar to the preferences on the same movie by a group of K users. This group of the

Table 1. Example of a movie rating data

	Pirates of the Caribbean 4	Kung Fu Panda 2	Harry Potter 6	Harry Potter 7
U_1	4	4	1	2
U_2	3	4	2	1
U_3	2	2	4	4
U_4	4	4	1	2

users is well known as K nearest neighbors (KNN). Namely, they are the top-K users who have similar tastes with a given user. Take Table 1 for example, to find the KNN for user U_4, we observe the ratings given by the four users on the given movies except "Harry Potter 7". We can see that U_1 and U_2 actually give similar ratings as U_4 – high ratings (3 or 4-star) on the first two movies and low rating on the movie "Harry Potter 6". Therefore, we infer that U_4 may rate the movie "Harry Potter 7" similarly as how the U_1 and U_2 rate the same movie.

To identify the KNN, we can use similarity measures to calculate user-user similarities or correlations, such as the cosine similarity shown by Eq. 1.

$$sim(U_i, U_j) = \frac{\overrightarrow{R_{U_i}} \bullet \overrightarrow{R_{U_j}}}{\|\overrightarrow{R_{U_i}}\|_2 \times \|\overrightarrow{R_{U_j}}\|_2} \tag{1}$$

We use a rating matrix similar to Table 1 to represent our data. $\overrightarrow{R_{U_i}}$ and $\overrightarrow{R_{U_j}}$ are the row vectors for user U_i and U_j respectively, where the rating is set as zero if a user did not rate the item. The size of these rating vectors is the same as the number of movies. In Eq. 1, the numerator represents the dot product of the two user vectors, while the denominator is the multiplication of two Euclidean norms (i.e., L2 norms). The value of K in KNN refers to the number of the top similar neighbors we need in the rating prediction functions. We need to tune up the performance by varying different numbers for K.

Once the KNN are identified, we can predict how a user rates one item by the rating function described by Eq. 2.

$$P_{a,t} = \bar{r}_a + \frac{\sum\limits_{u \in N} (r_{u,t} - \bar{r}_u) \times sim(a,u)}{\sum\limits_{u \in N} sim(a,u)} \tag{2}$$

where $P_{a,t}$ represents the predicted rating for user a on the item t. N is the top-K nearest neighborhood of users a, and u is one of the users in this neighborhood. The sim function is a similarity measure to calculate user-user similarities or correlations, while we use cosine similarity in our experiments. Accordingly, $r_{u,t}$ is neighbor u's rating on item t, $\bar{r}_a$ is user a's average rating over all items, and $\bar{r}_u$ is u's average rating.

This prediction function tries to aggregate KNN's ratings on the item t to estimate how user a rates t. However, the predicted ratings may be not accurate if user a is a GS user, since the user similarities or correlations between a and his or her neighbors may be very low. From another perspective, if a GS user is selected as one of the neighbors for a common user, it may result in odd recommendations or predictions since GS users may have unusual tastes on the items.

2.2 Grey Sheep Users

Due to the fact that UBCF takes advantage of the user-user similarities to produce the recommendations, the user characteristics in the collaborative filtering techniques become one of the key factors that can affect the quality of recommendations. McCrae et al. categorize the users in the recommender systems into three classes [11]: "the majority of the users fall into the class of *White Sheep* users, where these users have high rating correlations with several other users. The *Black Sheep* users usually have very few or even no correlating users, and the case of black sheep users is an acceptable failure[1]. The bigger problem exists in the group of *Grey Sheep* users, where these users have different opinions or unusual tastes which result in low correlations with many users; and they also cause odd recommendations for their correlated users". Therefore, Grey Sheep (GS) user usually refers to "a small number of individuals who would not benefit from pure collaborative filtering systems because their opinions do not consistently agree or disagree with any group of people [6]".

There are two significant characteristics of GS users indicated by the related research: On one hand, *GS users do not agree or disagree with other users* [7,11]. Researchers believe GS users may fall on the boundary of the user groups. Ghazanfar et al. [7,8] introduces a clustering technique to identify the GS users, while Gras et al. [9] reuses the outlier detection based on the user's rating distributions. On the other hand, *GS users may have low correlations with many other users, and they have very few highly correlated neighbors* [6].

2.3 Identification of Grey Sheep Users

There are several research [6,11,13,14] that point out the problem of GS user, define or summarize the characteristics of GS users, but very few of the existing work were made to figure out the solutions to identify GS users.

By paying attention to the first characteristics of GS users mentioned in Sect. 2.2, researchers believe GS users may fall on the boundary of the user groups. Ghazanfar et al. [7,8] proposes a clustering technique to identify the GS users, while they define improved centroid selection methods and isolates the GS users from the user community by setting different user similarity thresholds. The main drawback in their approach is the difficulty to find the optimal number

[1] The problem of black sheep users is caused by the situation that we do not have rich or even no rating profiles for these users. It is acceptable failure since the problem can be alleviated or solved if these users will continue to leave more ratings on the items.

of clusters, as well as the high computation cost to end up convergence in the clustering process, not to mention the unpredictable varieties by initial settings and other parameters in the technique. In their experiments, they demonstrate that content-based recommendation algorithms can be applied to improve the recommendation performance for the GS users. By contrast, Gras et al. [9] reuses the outlier detection based on the distribution of user ratings. They additionally take the imprecision of ratings (i.e., prediction errors) into account. However, the rating prediction error can only be used to evaluate whether a user is a GS user, it may not be appropriate to utilize it to identify GS users. It is because GS user is not the only reason that leads to large prediction errors. In other words, a user associated with large prediction errors is not necessary to be a GS user.

Another characteristics is that *GS users may have low correlations with many other users, and they have very few highly correlated neighbors* [6]. Most recently, we made the first attempt to take advantage of this characteristics to identify the GS users by the distribution of user-user similarities in the collaborative filtering approach [15]. More specifically, we statistically analyze a user's correlations with all of the other users, figure out bad and good examples, and reuse the outlier detections to identify potential GS users. Note that our work is different from the Gras et al. [9]'s work, since they stay to work on the distribution of user ratings, while we exploit the distribution of user similarities.

However, one of the drawbacks in the approach [15] is that the user-user similarity cannot be measured if two users did not rate the same items. Also, the user-user similarities may not be reliable if the number of co-rated items by two users is limited. In this paper, we propose an improved approach to alleviate this problem.

3 Methodologies

We first briefly introduce the basic solution proposed in [15]. Afterwards, we introduce and discuss the proposed approach to improve the basic solution in this section.

3.1 Basic Solution by the Distribution of User Similarities

As mentioned in [6], White Sheep users are the common users that have high correlations with other users. Namely, we can find a set of good KNN for White Sheep users. By contrast, GS users have correlations with other users but most of the correlations are relatively low. The basic solution in [15] relies on the following assumptions: A White Sheep user usually has higher correlations with other users, therefore its distribution of user similarities is expected to be left-skewed and the frequency at higher similarities should be significantly larger. In terms of the GS users, we do not have many high correlations with other users, and most of the user similarities are low. In short, the distribution of user similarities for GS users may have the following characteristics:

- It is usually a right-skewed distribution.
- The descriptive statistics of the user similarities, such as the first, second and third quartiles (q1, q2, q3), as well as the mean of the correlations, may be relatively smaller, since GS users have low correlations with other users.

Therefore, the basic solution in [15] can be summarized by the following four steps: distribution representations, example selection, outlier detection and examination of GS users.

Distribution Representations. The first step is to obtain user-user similarities and represent the distribution of user similarities for each user in the data set. We use the cosine similarity described by Eq. 1 to calculate the user-user similarity between every pair of the users. Note that the similarity of two users may be zero if there are no co-rated items by them. We remove the zero similarities from the distribution, since we only focus on the known user-user similarities in our data.

Table 2. Example of distribution representations

User	q1	q2	q3	Mean	STD	Skewness
40459	0.051	0.089	0.133	0.098	0.060	0.964
7266	0.028	0.056	0.091	0.064	0.045	1.245
34975	0.128	0.181	0.243	0.193	0.093	0.671
34974	0.093	0.149	0.209	0.156	0.084	0.568
34977	0.047	0.077	0.121	0.112	0.115	2.516
...	...	...	...	...	...	...

As a result, we are able to obtain a list of non-zero user-user similarities for each user. We further represent each user by the descriptive statistics of his or her distribution of the user similarities, including, q1, q2, q3, mean, standard deviation (STD) and skewness, as shown by Table 2.

Example Selection. Outlier detection [5,10] refers to the process of the identification of observations which do not conform to an expected pattern or other items in a data set. Thus it has been selected to distinguish GS users from other users in our approach. Gras et al. [9]'s work also points out that the identification of GS users is closely related to the outlier detection problem in data mining.

To apply the outlier detection, we need to select *good* (i.e., White Sheep users) and *bad* (i.e., potential GS users) examples in order to construct a user matrix similar to Table 2. This step is necessary especially when there are large scale of the users in the matrix. We suggest to filter the users by the descriptive statistics of their similarity distributions, such as the first quartile (q1), the second quartile (q2), the third quartile (q3), as well as mean of the similarity values, etc. More specifically, the bad examples could be selected by the following constraints:

- **Low similarity statistics**: In this case, q1, q2, q3 and mean may be much smaller than other users. We can select a lower-bound as the threshold. For example, if a user's mean similarity is smaller than *the first quartile* of mean similarities (i.e., the list of mean values over all of the users), this user is selected as one of the bad examples. The constraints could be flexible. They can be applied to the mean similarity only, or they could be applied to any subsets of {q1, q2, q3, mean} at the same time.
- **The degree of skewness**: This time, we apply a constraint on the skewness. For example, if a user's skewness value in his or her similarity distribution is larger than *the third quartile* of skewness values over all of the users, this user may be selected as one of the bad examples. It is because GS users may have very few highly correlated neighbors, and most of their user correlations are pretty low, which results in a heavily right-skewed similarity distribution.

Note that the constraints could be flexible or strict. The best choice may vary from data to data.

Outlier Detection. There are several outlier detection [5,10] techniques, such as the probabilistic likelihood approach, the clustering based or the density based methods, etc. We adopt a density based method which relies on the local outlier factor (LOF) [3]. LOF is based on the notion of local density, where locality is given by the $\Bbbk$ nearest neighbors[2] whose distance is used to estimate the density. The nearest neighbor, in our case, can be produced by using distance metrics on the feature matrix, while the feature matrix is the distribution representation matrix as shown in Table 2. By comparing the local density of a user to the local densities of his or her neighbors, one can identify regions of similar density, and the users that have a substantially lower density than their neighbors can be viewed as the outliers (i.e., the GS users) finally. Due to that the distances among the users are required to be calculated, we apply a normalization to the matrix in Table 2 in order to make sure all of the columns are in the same scale.

A user will be viewed as a common user if his or her LOF score is close to the value of 1.0. By contrast, it can be an outlier (i.e., potential GS user) if the LOF score is significantly larger or smaller than 1.0. We set a threshold for the LOF score, and tune up the results by varying the values of $\Bbbk$ and the LOF threshold in our experiments in order to find qualified GS users as many as possible. Note that, not all of the identified outliers are GS users, since it is possible to discover the outliers from the good examples too. We only consider the outliers from the bad examples as GS users in our experiments.

[2] We use $\Bbbk$ to distinguish it from the K in KNN based UBCF algorithm.

Examinations. With different values of $\mathbb{k}$ and the LOF threshold, we are able to collect different sets of the users as the GS users. We use the following approaches to examine the quality of the GS users:

- The recommendation performance for the group of GS users by collaborative filtering must be significantly worse than the performance for the White Sheep users. More specifically, the average rating prediction errors (see Sect. 4.1) based on the rating profiles associated with these GS users must be significantly higher than the errors that are associated with non-GS users. If the prediction errors for GS users and the remaining group of the users are close, we will perform two-independent sample statistical test to examine the degree of significance.
- We additionally visualize the distribution of similarities for GS users, in comparison with the one by non-GS users. The distribution of user similarities for GS and White sheep users are right and left-skewed respectively.

We tune up the values of $\mathbb{k}$ and the LOF threshold to find GS users as many as possible. But note that GS users are always a small proportion of the users in the data.

3.2 Improved Approach by Histogram Intersection

The basic solution by the distribution of user similarities is highly dependent with the user-user similarities and the distribution of these similarity values. However, there is a well-known drawback in the similarity calculations (such as the cosine similarity or the Pearson correlations) – the similarity between two users can be obtained only when they have co-rated items. Also, the similarity value may be not that reliable if the number of co-rated items is limited.

Therefore, we seek solutions to improve the quality of user-user similarities. One of the approaches is to generate user-user similarities by the histogram intersections [2] based on the distribution of cosine similarities. More specifically, we use cosine similarity to calculate the user-user similarity values first. As a result, each user can be represented by the distribution of similarities between other users and him or her. We can represent this distribution by a histogram which is constructed by N bins. Each bin can be viewed as a bar in the histogram. In our experiment, we use 40 bins with distance of 0.025 (i.e., the range is [0, 1] which represents the similarity). Furthermore, the similarity between two users can be re-calculate by the similarity between two histograms. Histogram intersection becomes one of the ways to measure the similarity between two histograms.

An example can be shown by Fig. 1. The blue and orange regions represent two histograms, while the pink areas stand for the histogram intersections. Larger the pink area is, more similar two histograms will be.

Assume there are two users u_1 and u_2. We use I and M to represent the histogram representation of u_1 and u_2's distribution of user-user similarities. These similarity values are obtained by the cosine similarity in UBCF.

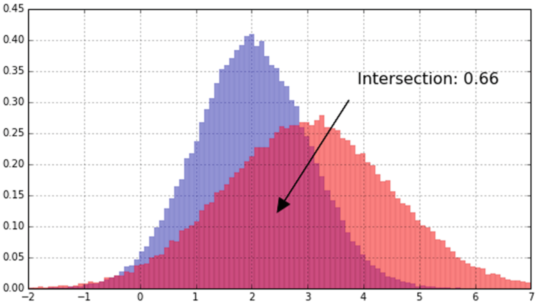

Fig. 1. Example of histogram intersections

The similarity between u_1 and u_2 by the histogram intersection can be simply re-calculated by:

$$sim(u_1, u_2) = \frac{\sum_{j=1}^{N} Min(I_j, M_j)}{\sum_{j=1}^{N} M_j} \tag{3}$$

where N represents the number of bars or bins in the histogram. I_j and M_j indicate the frequency value in the j^{th} bar or bin in histogram I and M respectively. The function Min is used to get the minimal value between I_j and M_j. By this way, we can still calculate the similarity between two users, even if they do not have co-rated items.

4 Experiments and Results

4.1 Experimental Settings

We use the MovieLens 100 K rating data set[3] which is a movie rating data available for research. In this data, we have around 100,000 ratings given by 1,000 users on 1,700 movies. We simply split the data into training and testing set, where the training set is 80% of the whole data. Each user has rated at least 20 movies. We believe these users have rich rating profiles, and black sheep users are not included in this data.

We apply our proposed methodologies on the training set to identify GS users, and examine them by the recommendation performance over the test set. To obtain the prediction errors, we apply UBCF described by Eq. 2 as the collaborative filtering recommendation algorithm. In UBCF, we adopt the cosine similarity to measure the user-user similarities, and vary different value of K ($K = 100$ is the besting setting in our experiments) in order to find the best KNN. The recommendation performance is measured by mean absolute error

[3] https://grouplens.org/datasets/movielens/100k/.

(MAE) which can be depicted by Eq. 4. T represents the test set, where $|T|$ denotes the total number of ratings in the test set. $R_{a,t}$ is the actual rating given by user a on item t. (a,t) is the <user, item> tuple in the test set. $P_{a,t}$ is the predicted rating by the function in Eq. 2. The "abs" function is able to return the absolute value of the prediction error.

$$MAE = \frac{1}{|T|} \sum_{(a,t) \epsilon T} abs(P_{a,t} - R_{a,t}) \tag{4}$$

4.2 Results and Findings

We follow the four steps in Sect. 3 to identify the GS users from the training set. As mentioned in the Sect. 3.1, it is flexible to set different constraints to select good and bad examples. In our experiments, we tried both strict and loose constraints. The strict constraints can be described as follows: we go through the distribution representation matrix, and select the bad examples (i.e., potential GS users) if his or her q1, q2 and mean similarity value is smaller than the first quartile of the q1, q2 and mean distribution of all the users. According, the loose constraints will seek the bad examples by using the filtering rule that q1, q2 and mean similarity value is smaller than the second or the third quartile of the q1, q2 and mean distribution of all the users. However, there is not clear pattern to say which constraint is better. In our experiments, the loose constraints can help find more GS users if we use the basic solution in Sect. 3.1, while the strict constraint is the better one if we use the improved approach discussed in Sect. 3.2. In the following paragraphs, we only present the optimal results based on the corresponding constraints. The group of bad examples is further filtered by the

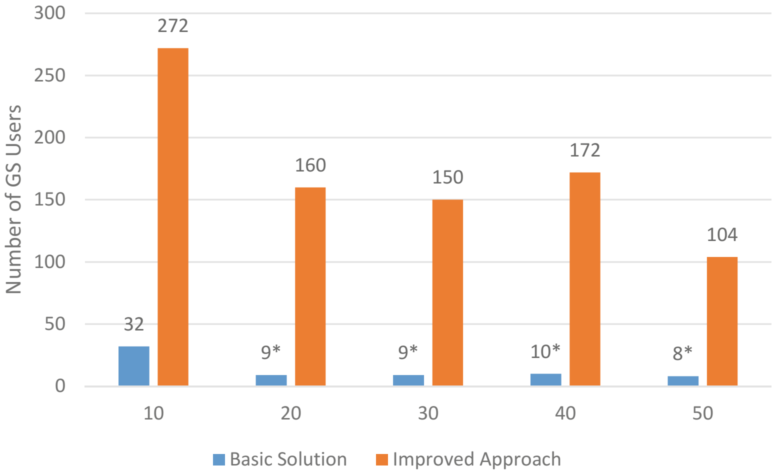

Fig. 2. The number of GS Users identified by different $\mathbb{k}$ values. Note that (*) tells that the two-independent sample statistical test was failed in that setting.

skewness – the users with skewness value smaller than the third quartile of the skewness distribution over all the users will be removed.

Afterwards, we blend the good and bad examples, and apply the LOF technique to identify the GS users. We tried different values of $\Bbbk$ and LOF thresholds in our experiments. The number of GS users identified can be shown by Fig. 2, while the x-axis represents $\Bbbk$ value. Note that the GS users are only the outliers from the bad examples. In addition, the group of GS users can only be considered as effective ones if the MAE of the rating profiles associated with these users is significantly larger than the MAE based on the non-GS users. We use 95% as confidence level, and apply the two-independent sample statistical test to examine whether they meet this requirement.

Based on the Fig. 2, we can observe that more GS users can be identified if we use the improved approach which utilizes the histogram intersection to produce user-user similarities. More specifically, by varying different value of $\Bbbk$ and LOF thresholds, we can only find qualified GS users by setting $\Bbbk$ as 10 in the basic solution. The results based on other $\Bbbk$ values failed the statistical tests, which tells that the MAE value by these GS users in UBCF is not significantly larger than the MAE obtained from non-GS users.

In addition, the statistical tests are all passed when we vary the $\Bbbk$ value in the improved approach discussed in Sect. 4. However, the difference between the MAE values by the GS users and non-GS users could be very small. Therefore, we decide to choose the result by using $\Bbbk$ as 50 as the optimal result, while the MAE by GS users is 0.810 and it is 0.760 for the non-GS users.

Table 3. MAE results

	All users	Good examples	Bad examples	Remaining users	GreySheep
Basic solution	0.765	0.766	0.763	0.762	0.844
Improved approach	0.765	0.766	0.777	0.760	0.810

Table 3 describes the MAE evaluated based on the rating profiles in the test set associated with different groups of the users. The "remaining users" refer to users excluding the identified GS users. There are no statistically differences on MAE values for these user groups if we do not take the group of GS uses into account. The MAE by the identified GS users is significantly higher than the one by other group of the users at the 95% confidence level.

Furthermore, we compare our approaches with the two existing methods which are used to identify GS users: the clustering based method [8] and the distribution based method [9]. The number of identified GS users can be described by Fig. 3. We can find that our proposed approaches can beat the clustering-based method, while the distribution-based method is able to find more GS users than the basic solution we propose in our previous research. The best performing solution is still the one that we utilize the histogram intersection

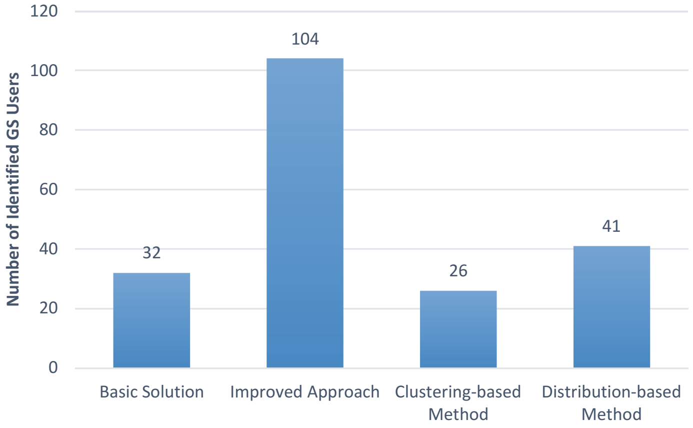

Fig. 3. Comparison between our approaches and existing methods

to calculate the user-user similarities. Keep in mind that the complexity of our proposed approach is much lower than these two existing methods, since we only need to apply the outlier detection techniques after the example selections.

We look into the characteristics of identified GS and White Sheep users. We select two GS users and two White Sheep users as the representatives, visualize the distribution of user similarities, as shown in Fig. 4. The bars in slate blue and coral present the histograms for two users, while the bars in plum capture the overlaps between two histograms. The x-axis is the bins of the similarities, while we put the similarity values (in range [0, 1]) into 40 bins with each bin size as 0.025. The y-axis can tell how many similarity or correlation values that fall in corresponding bins. These distributions of user similarities are produced by the histogram intersection – that's the reason why the similarity values are not that close to 1.0.

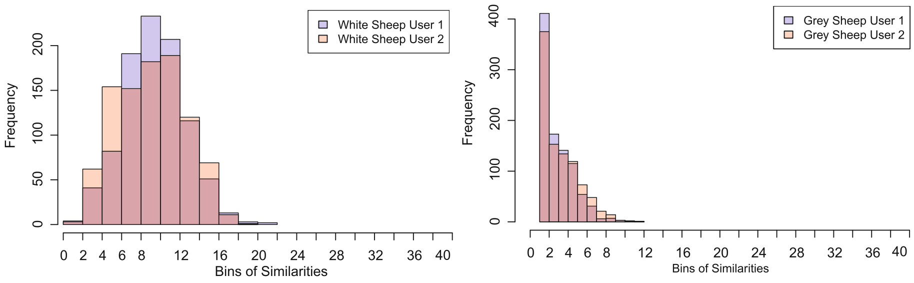

Fig. 4. Visualization of the similarity distributions

We can observe that the distributions for both GS and White Sheep users are right-skewed, if we take all of the 40 bins into consideration. While focusing on the first 20 bins, we can tell that the distribution of user similarities by the GS users is heavily right-skewed, and the one for the White Sheep users is close to normal distribution. In addition, we can clearly notice that the correlations between GS users and other users are pretty low, which presents a heavily right-skewed distribution of the similarities. The situation is much better for the White Sheep users, since they usually have highly correlated neighbors. According to the observations at the bins from 12 and 20, we can discover that we have at least 300 high correlations for the White Sheep users, but almost zero for the GS users. This pattern is consistent with the definition of GS and White Sheep users in [11]. According to previous research [7,11], we need to apply other recommendation algorithms (such as content-based approaches) to reduce the prediction errors for these GS users, where we do not explore further in this paper.

5 Conclusions

In this paper, we improve the approach of identifying Grey Sheep users based on the distribution of user similarities by utilizing the histogram intersection to better produce user-user similarities. The proposed approach in this paper is much easier than the previous methods [7,9,11] in terms of the complexity. The improved approach that utilizes the histogram intersection is demonstrated as the best performing solution in comparison with the existing methods to identify Grey Sheep users in the MovieLens 100 K data.

In our future work, we will apply the proposed approach to other data sets rather than the data in the movie domain. Also, we believe the same approach can also be used to identify *Grey Sheep items* in addition to the Grey Sheep users. The problem of Grey Sheep users may not only happen in the traditional recommender systems, but also exist in other types of the recommender systems. For example, in the context-aware recommender systems [1,16,17], the definition of Grey Sheep users could be the users who have unusual tastes in specific contextual situations. The proposed approach in this paper can be easily extended to these special recommender systems, and we may explore it in the future.

References

1. Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A.: Context-aware recommender systems. AI Magazine **32**, 67–80 (2011)
2. Barla, A., Odone, F., Verri, A.: Histogram intersection kernel for image classification. In: Proceedings 2003 International Conference on Image Processing, p. III-513-16 (2003)
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: ACM Sigmod Record, vol. 29, pp. 93–104. ACM (2000)
4. Burke, R.: Hybrid recommender systems: survey and experiments. User Modeling User-adapted Interact. **12**(4), 331–370 (2002)
5. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. (CSUR) **41**(3), 15 (2009)

6. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining content-based and collaborative filters in an online newspaper. In: Proceedings of ACM SIGIR Workshop on Recommender Systems, vol. 60 (1999)
7. Ghazanfar, M., Prugel-Bennett, A.: Fulfilling the needs of gray-sheep users in recommender systems, a clustering solution. In Proceedings of the 2011 International Conference on Information Systems and Computational Intelligence, pp. 18–20 (2011)
8. Ghazanfar, M.A., Prügel-Bennett, A.: Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. Expert Syst. Appl. **41**(7), 3261–3275 (2014)
9. Gras, B., Brun, A., Boyer, A.: Identifying grey sheep users in collaborative filtering: a distribution-based technique. In: Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, pp. 17–26. ACM (2016)
10. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial Intell. Rev. **22**(2), 85–126 (2004)
11. McCrae, J., Piatek, A., Langley, A.: Collaborative filtering. http://www.imperialviolet.org (2004)
12. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186. ACM (1994)
13. Ruiz-Montiel, M., Aldana-Montes, J.F.: Semantically enhanced recommender systems. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009. LNCS, vol. 5872, pp. 604–609. Springer, Heidelberg (2009). doi:10.1007/978-3-642-05290-3_74
14. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artificial Intell. **2009**, 4 (2009)
15. Zheng, Y., Agnani, M., Singh, M.: Identifying grey sheep users by the distribution of user similarities in collaborative filtering. In: Proceedings of The 6th ACM Conference on Research in Information Technology. ACM (2017)
16. Zheng, Y., Burke, R., Mobasher, B.: Splitting approaches for context-aware recommendation: an empirical study. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing, pp. 274–279. ACM (2014)
17. Zheng, Y., Mobasher, B., Burke, R.: CSLIM: contextual SLIM recommendation algorithms. In: Proceedings of the 8th ACM Conference on Recommender Systems, pp. 301–304. ACM (2014)