GTMedoids: A New Grey Sheep Users Detection Approach

Bouchra Boualaoui[1], Ahmed Zellou[2], Lahbib Ajallouda[3]
SPM Research Team, ENSIAS, Mohammed V University in Rabat, Rabat, Morocco[1,2]
Poly-Disciplinary Faculty of Es-Semara, Ibn Zohr University Agadir, Morocco[3]

***Abstract*—Recommender systems have been developed to serve users and provide them with the best suggestions. Despite their success, offering fully identical recommendations to users' preferences remains a difficult task, where the complexity of human taste results in different challenges. Grey sheep user phenomena continues to be one of the most common, where the user is defined by his unique interactions with the system, making it difficult to associate with similar users, as he rarely agrees with them. In this study, we presented a new approach for identifying grey sheep users. It is based on the taste context and nature of user interaction with the system. We grouped similar users using an enhanced Kmedoids clustering method with a new dissimilarity metric and introduced a novel process to distinguish between users. The differentiation is achieved by assigning weights to each cluster based on how much it reflects the grey sheep user characteristics. We evaluated the efficiency of Grey Threshold Medoids (GTMedoids) using the FilmTrust and MovieLens 100k datasets. The results show the superior performance of our approach in detecting grey sheep users.**

***Keywords*—Recommender system; grey sheep users detection; clustering; Kmeans; Kmedoids; dissimilarity metric**

I. Introduction

Recommender systems (RS) have demonstrated their ability to retrieve what could be interesting to users and help improve the user experience in a digital world overwhelmed by the explosion of data and the speed with which data is growing. However, this efficiency is exposed to high degradation due to multiple challenges, related to sparsity [1], [2], scalability [3], or fairness and user satisfaction, of which we can cite grey sheep users (GSUs) [4], where the first step to accurately analyze their preferences is to detect them. Although grey sheep user detection is important, it has been neglected in the literature. In this regard, our motivation behind this work is the importance of fairness and equality treatment that should be attributed to recommender systems to always provide all users with accurate suggestions and not only serve part of them and discriminate against the rest. To accurately detect grey sheep users in a system, the following research questions have been formulated:

- RQ1: How can user similarity be redefined to fully capture user taste behavior?
- RQ2: Can grey sheep users be detected dynamically and effectively from the entire user population without predefined groups?
- RQ3: How does the presence of grey sheep users in a system affect recommendation accuracy?

In this study, we present GTMedoids as a new proposed solution for detecting grey sheep users in a recommender system. We clustered candidate grey sheep users based on their interactions with the system and considered the bi-polarity of human taste. Then, attribute to each cluster a weight, where only clusters with power weights less than a predefined threshold are finally detected as grey sheep clusters as they meet the criteria of this category of users. To prove the validity of our model, we ran a set of experiments on two different benchmarks and demonstrated the efficiency of our model in detecting grey sheep users.

The study is structured into several sections: Section II provides background information and reviews existing literature on grey sheep user detection. Section III details the proposed approach, GTMedoids, and examines its various components. Section IV analyzes the results obtained from the study. Finally, Section V concludes the work and outlines potential future research directions.

II. Related Work

The evolution of recommender systems can be marked with three generations, where the first generation was initiated to focus on content and collaborative techniques [5], [6], the second enhanced quality and precision by involving more contextual information, such as location and time [7], [8], while the third generation is known for leveraging the hidden semantics in supplementary sources of knowledge, including social networks and knowledge graphs [9]. However, during the course of this enhancement, major challenges were addressed. Yet, some of these challenges were less treated than others, regardless of their importance, especially grey sheep users.

Grey sheep users are defined as users with significant deviations from recognized preference patterns [10]. Due to their unique taste, they end up receiving inaccurate suggestions that do not match their interaction with the system [4]. Although the importance of the topic, as it has a direct impact on the system's accuracy and the quality of recommendations for all users, it is considered one of the least treated RS challenges in the literature.

To address the challenge of GSUs, detecting these users remains a crucial step. The detection techniques in the literature can be grouped into two main categories. Outlier detection and clustering techniques. The first group treats GSUs as anomalies and outliers due to their odd choices. These techniques rely on ratings distribution and user profiles [11], [12], [14]. However, other outlier techniques can be explored, as shared in [10], such as angle-based outlier detection (ABOD) [15]. Regardless of the popularity of outlier detection methods as a way to identify grey sheep users in a system, their nature and the way that they are designed highly put the results under investigation, as the final detected set of users can also include those suffering from sparsity and sparsity issues due to their low density, while a grey sheep user is a user that is committed to the system but has different preferences [18], [11]. Adding to that, any data points that are close to a grey sheep user run the risk of being flagged as a grey sheep user [20]. And it is important to also mention that these methods need to treat the whole datasets, which can make the distance or density scores attached to all type of users even less meaningful [29].

On the other hand, the second category uses clustering approaches that are based on computing user similarities [16], [17], [19], [21], [22]. The primary limitation of these approaches is the fact that they cluster all the users of the system and view them all as candidates, which can lead to high computational costs and group all the final grey sheep users in one $(K+1)^{th}$ cluster, leading to a lack of clarity about the degree of grayness of the users, especially in the case where the system aims to remove them [14]. Moreover, all these works, regardless of their category, have neglected the polarity of human taste. However, it deserves attention that [4] highlighted the possibility of identifying GSUs using personality parameters but without real application, as they have used only the cosine similarity with KNN [23]. Therefore, we present GTMedoids, a clustering method that takes into account both sides of user feedback and also weights the degree of users' grayness.

III. GTMedoids Approach

In this section, we present our proposed GTMedoids approach and its different components. Fig. 1 provides a high-level overview of the approach and the interaction between its parts.

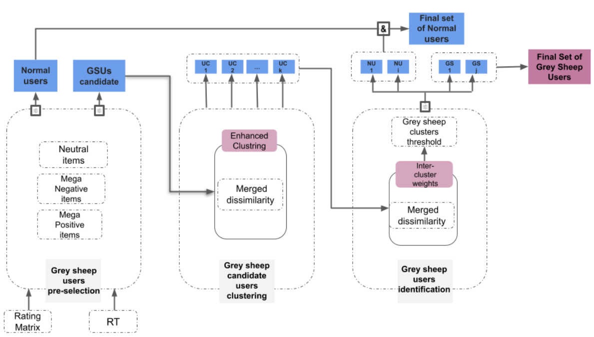

Fig. 1. Grey threshold medoids approach.

A. Grey Sheep Users Pre-Selection

Grey sheep users have unique and distinguished preferences, making the recommendation process more difficult and not as accurate as for normal users, who already have close users in terms of taste and similarities. Consequently, identifying these users should be based mainly on user preferences shared in the rating matrix [24]. Most studies measure preference based on the interaction with the item, where the existence of the interaction is interpreted as an interest and preference, ignoring the fact that an interaction with bad feedback should not be treated as a good one, as human taste is not only reflected in what he likes l but also in what he does not like l, which can be denoted as follows [see Eq. (1)]:

$$taste = \{d/d \in Detest\} \cup \{l/l \in Like\} \quad (1)$$

Taking this into consideration, grey sheep user identification should be based on both types of feedback; as a result, a GSU is then a user who highly disagrees with his community.

Clustering all users with prior knowledge that not all of them are GSUs is not efficient, as only concentrating on a set of users that can meet GSU characteristics; in particular that Kmeans variations are sensitive to outliers because of the mean computations [25]. In our case, outliers are normal users who more often share the same taste as the majority of the community. However, before deciding to exclude these users, we need to quantify the community's taste for comparison with individual preferences. We consider the global classification of items as collaborative signals contributed by all users in the system. An item can then be:

- Mega Positive: is an item that the majority has appreciated by giving it individual positive feedback Pos.
- Mega Negative: is an item that the majority has unappreciated by giving it an individual negative feedback Neg.
- Neutral: is an item having the same amount of individual positive and negative feedback.

The differentiation between positive and negative individual feedback is done based on a rating threshold RT as our approach treats the user-item rating matrix R [see Eq. (2)]:

$$feedback_{u,i} = \begin{cases} Pos & \text{if } r_{u,i} \geq RT \\ Neg & \text{otherwise} \end{cases} \quad (2)$$

where, $r_{u,i}$ is the rating given by the user u to the item i.

After computing the collaborative classification of each item, we refine our initial set to cluster by excluding normal users.

- Normal user: Frequently aligns with global preferences, showing a level of agreement that meets or exceeds the average.
- Grey sheep candidate user: Rarely aligned with the collaborative signals, meaning having an amount of less than the average of agreements.

The filtered users show potential to be grey sheep users, requiring further refinement as the actual feedback and ratings were not used to determine their greyness level or similarities with others. The subsequent step focuses on calculating user similarities and identifying their nearest neighbors.

B. Grey Sheep Candidate Users Clustering

To enhance the accuracy of user clustering, it is essential to include all rated items, even neutral ones. The proposed clustering approach modifies the Kmeans algorithm by applying a new dissimilarity metric and using actual user profiles as the centers of clusters instead of means and leads to more significant associations among user groups [31].

1) The merged dissimilarity metric: We have developed our own measurement metric for two reasons:

a) The multi-dimensional nature of human taste: The dissimilarity function must consider both aspects of taste, with the weight assigned to each depending on user preference. Some users prefer to share positive feedback, while others are more inclined to express negative feedback.

b) Reduce the unexplored items effect: One major issue with distance functions is their inability to differentiate items rated by only one user, leading to the same distance despite differing preferences. To address this, a contextual approach was adopted, where items are considered closer only if they share a collaborative taste. This involves incorporating feedback nature as context with weights, so items can be treated similarly for users if their ratings are closely aligned and they share similar feedback contexts. Fig. 2 illustrates how adding multi-dimensional context improves accuracy in understanding user preferences, as the Euclidean distance can misleadingly suggest similarity between items that do not truly reflect preferences, impacting the mining of user data, where this distance may treat items i_0 and i_1 as equivalent, while they do not truly reflect aligned user preferences. Consequently, this miscalibration leads to false similarities among users, such as u_0 and u_1.

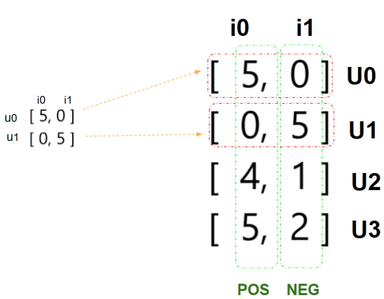

Fig. 2. The impact of the unexplored items on the dissimilarity measurement.

The final metric is a user-personalized metric that merges both sides of human tastes and can be presented as follows [see Eq. (3)]:

$$MergedD(u_a, u_b) = weight_{dislike} \times DisLikedD(u_a, u_b) + weight_{like} \times LikedD(u_a, u_b) \quad (3)$$

As the proposed metric is a composition of the two components, the liked and disliked dissimilarities are calculated using Eq. (4):

$$LikedD(u_a, u_b) = \sqrt{\sum_{i=1}^{n}((\frac{itemlikes_i}{maxLikes})^2 * (r_{u_a,i} - r_{u_b,i})^2)} \quad (4)$$

The liked dissimilarity function is characterized by its ability to add the global taste context to the dissimilarity computation between users. It measures how close the feedback of the users u_a and u_b is to the global positive taste, where it attributes more weight and importance to positive items and assesses how users u_a and u_b interact with these items. Similarly, the disliked dissimilarity function focuses more on less globally admired items [see Eq. (5)]:

$$DislikedD(u_a, u_b) = \sqrt{\sum_{i=1}^{n}((\frac{itemDislikes_i}{maxDisLikes})^2 * (r_{u_a,i} - r_{u_b,i})^2)} \quad (5)$$

where,

- $r_{u',i}$ are the ratings of user u' for item i.
- $maxLikes$ the maximum number of users who liked any single item.
- $itemLikes_i$ calculates the number of users who rated item i greater than or equal to the threshold RT.
- $maxDisLikes$ the maximum number of users who disliked any single item.
- $itemDislikes_i$ calculates the number of users rated item i less than or equal to the threshold RT.

To reflect the fact that humans differ in which side of taste they tend to express the most, we added the like and dislike weights [see Eq. (6) and Eq. (7)]:

$$weight_{like} = \frac{(\#likedItems_{u_a} + \#likedItems_{u_b})}{(\#ratedItems_{u_a} + \#ratedItemsu_b)} \quad (6)$$

$$weight_{dislike} = \frac{(\#dislikedItems_{u_a} + \#dislikedItems_{u_b})}{(\#ratedItems_{u_a} + \#ratedItemsu_b)} \quad (7)$$

where,

- $\#ratedItems_{u'}$: how many items were rated by the user u'.
- $\#likedItems_{u'}$: how many items were liked by the user u'.
- $\#dislikedItems_{u'}$: how many items were disliked by the user u'.

Therefore, the merged dissimilarity function, Eq. (3), is used to cluster users based on their preferences, Algorithm 1 defines the steps followed to cluster the users.

The final returned User Clusters, UC, group the grey sheep users candidate into clusters by similarity. In the next section, we explore how the final set of grey sheep users is identified.

Algorithm 1 Grey Kmedoids Clustering Algorithm

Require: Number of clusters K, Grey sheep candidate users $GC = \{gc_1, gc_2, \ldots, gc_n\}$
Ensure: UserCluster $UC = \{UC_1, UC_2, ..., UC_k\}$ and medoids $M = \{m_1, \ldots, m_K\}$
select K objects randomly from GC as initial medoids
$M \leftarrow \{m_1, m_2, \ldots, m_K\}$
repeat
  assignment:
  for each $gc \in GC$ **do**
    assign gc to the nearest medoid using

$$i \Leftarrow \arg\min_{j \in \{1,\ldots,K\}} d_{\text{merged}}(gc, m_j)$$

    add gc to cluster UC_i
  end for
  compute total cost:

$$S(M) = \sum_{i=1}^{K} \sum_{gc \in UC_i} d_{\text{merged}}(gc, m_i)$$

  swap:
  for each medoid $m \in M$ **do**
    for each non-medoid $o \in GC \setminus M$ **do**
      $M' \leftarrow (M \setminus \{m\}) \cup \{o\}$
      reassign all points using M'
      compute $S(M')$
      if $S(M') < S(M)$ **then**
        $M \leftarrow M'$
      end if
    end for
  end for
until medoids do not change
return M and clusters $UC_1, \ldots, UC_K$

C. Convergence Analysis of GTMedoids

As GTMedoids is a clustering method built based on Kmedoids, proving its convergence depends on proving that our algorithm can to reach a fixed point where neither assignments nor medoids change. In other words, the objective function should be monotonically non-increasing and cannot decrease indefinitely over a finite set [30].

Let $d_{merged}(\cdot, \cdot)$ denote the dissimilarity function used in the algorithm. The objective is the within-cluster merged-dissimilarity sum [see Eq. (8)]:

$$S'(\mathcal{UC}, \mathbf{m}) = \sum_{j=1}^{K} \sum_{i \in UC_j} d_{\text{merged}}(i, m_j), \tag{8}$$

where, $\mathcal{UC} = \{UC_1, \ldots, UC_K\}$ denotes the current user clusters, and $\mathbf{m} = (m_1, \ldots, m_K)$ denotes the set of cluster prototypes or medoids.

GTMedoids proceeds by alternating between two steps:

- **User assignment:** Given current medoid $\mathbf{m}^{(t)}$, each user i is assigned to the closest cluster UC [see Eq. (9)]:

$$uc^{(t+1)}(i) = \arg\min_{j \in \{1,\ldots,K\}} d_{\text{merged}}(i, m_j^{(t)}). \tag{9}$$

- **Medoid update:** Given the updated clusters $\mathcal{UC}^{(t+1)}$, each medoid is updated as Eq. (10):

$$m_j^{(t+1)} = \arg\min_{u \in UC_j^{(t+1)}} \sum_{i \in UC_j^{(t+1)}} d_{\text{merged}}(i, u). \tag{10}$$

The algorithm converges when neither the medoids nor the assignments change.

1) Proof of convergence:

a) Lemma a (User Assignment Step Monotonicity): Given fixed medoids $\mathbf{m}^{(t)}$. The assignment step does not increase the objective function S'.

Demonstration. Assigning user i to the cluster whose medoid minimizes the objective results in the smallest possible contribution for that user [see Eq. (11)]:

$$d_{\text{merged}}(i, m_{uc^{(t+1)}(i)}) \leq d_{\text{merged}}(i, m_{uc^{(t)}(i)}) \tag{11}$$

Hence, summing over all users, the total objective cannot increase [see Eq. (12)]:

$$S'(\mathcal{UC}^{(t+1)}, \mathbf{m}^{(t)}) \leq S'(\mathcal{UC}^{(t)}, \mathbf{m}^{(t)}). \tag{12}$$

b) Lemma b (Medoids Update Monotonicity): Given fixed cluster assignments $\mathcal{UC}^{(t+1)}$, the medoid update step does not increase S'.

Demonstration. For each cluster $UC_j^{(t+1)}$, the medoid $m_j^{(t+1)}$ is chosen as the minimizer of $\sum_{i \in UC_j^{(t+1)}} d_{\text{merged}}(i, u)$ over all $u \in UC_j^{(t+1)}$, as presented in Eq. (10).
By the characterization of the argmin, the chosen medoid cannot result in a larger value than any other one, including the previous one. Consequently, summing over all clusters yields [see Eq. (13)]:

$$S'(\mathcal{UC}^{(t+1)}, \mathbf{m}^{(t+1)}) \leq S'(\mathcal{UC}^{(t+1)}, \mathbf{m}^{(t)}). \tag{13}$$

c) Lemma c (Monotonic Decrease of the Objective): The objective sequence $\{S'^{(t)}\}$ generated by GTMedoids is monotonically decreasing.

Demonstration. Combining Lemmas a and b yields [see Eq. (14)]:

$$S'(\mathcal{UC}^{(t+1)}, \mathbf{m}^{(t+1)}) \leq S'(\mathcal{UC}^{(t+1)}, \mathbf{m}^{(t)}) \leq S'(\mathcal{UC}^{(t)}, \mathbf{m}^{(t)}). \tag{14}$$

d) Lemma d (Finite descent): GTMedoids terminates after a finite number of iterations.

Demonstration. Each medoid is selected from a finite set of user indices. Additionally, there are finitely possible partitions of n users into K clusters. Therefore, the number of possible combinations $(\mathcal{UC}, \mathbf{M})$ is finite.

Considering that the objective function is monotonically decreasing and cannot decrease indefinitely over a finite set, GTMedoids must eventually reach an optimized pair $(UC^{\star}, M^{\star})$ where neither the medoids nor the assignments change.

e) Theorem (Convergence of GTMedoids): GTMedoids converges in a finite number of updates to a local minimum of the objective function S'.

Demonstration. By Lemmas a–b, the objective function is monotonically decreasing.

By Lemma c, the sequence of solutions must reach an optimum; no additional changes are applied on the clusters, resulting in decreasing S' further.

D. Grey Sheep Users Identification

Clustering groups similar data points but struggles to clearly distinguish between dominant groups and outliers. In this regard, we have provided a new method that can be adopted for this matter in general, and not only for grey sheep user identification or recommender systems. Our technique is able to quantify the importance and weight of each cluster and detect the less impactful ones; in other words, we detect grey sheep users in our case.

1) Inter-Cluster Weights: We have successfully identified final clusters of users by analyzing their feedback and proximity based on multi-dimensional tastes and community preferences. The next step involves selecting only the grey sheep clusters, using the characteristics of GSUs as a baseline for this selection process:

- GSUs represent a minority; otherwise, mining their preferences and providing them with recommendations would have been easier.
- GSUs have distinguished preferences from users, meaning that a grey sheep user should be located far from all users.

We used the softmax function to calculate the weight of each cluster j, where the general equation is Eq. (15):

$$\tilde{\alpha}_j = \frac{exp(\alpha_j)}{\sum_{c \in UC} exp(\alpha_c)} \tag{15}$$

The softmax function heavily depends on α_j, to to obtain meaningful weights, we translated the GSUs characteristics to quantified metrics using the following Eq. (16):

$$\alpha_j = \begin{cases} \text{if } \#UC_j > 1 : \\ (\frac{\#UC_j}{max(\#UC)}) \times (1 - \frac{(\sum_{gc \in UC_j}(MergedD(gc, c_j))}{((\#UC_j \times max(MergedD(UC_j, c_j)))}) \\ \text{otherwise:} \\ 0 \end{cases} \tag{16}$$

The GSUs representing the minority clause are represented by the first part of the α_j multiplication function, and the fact that a grey sheep user should be located far from all users is the second part. We attribute a zero impact to the cluster that we already know are grey sheep clusters as they have only one user. By applying the softmax to the α_j, we obtained inter-cluster powers $\tilde{\alpha_j}$.

2) Grey sheep clusters threshold: Having an ideal system clear of grey sheep users and each user is having enough similar users sharing with him the same preferences means that cluster powers should be equal, meaning all the clusters should have $\frac{1}{K}$ as power weight. However, as there is no such perfect system in reality, we have added a tolerance rate to not be too strict and handled normal users as grey ones just because they have some disagreements with the collaborative signals [see Eq. (17)].

$$\epsilon = \frac{Max(\tilde{\alpha}) - Min(\tilde{\alpha})}{K} \tag{17}$$

By including the clause ϵ to verify if the difference between the maximum and the minimum is equally distributed and provide a tolerance to make sure that only users with high derivation from the reset as identified as GSUs [see Eq. (18)].

$$T = \frac{1}{K} + \epsilon \tag{18}$$

This threshold can detect all the clusters that impact and influence more than the others, and then any cluster with a weight larger than this threshold should be categorized as a normal user cluster.

IV. Results and Discussion

In this section, we demonstrate the validity of our approach and evaluate its performance in identifying grey sheep users through experiments with two distinct datasets.

A. Evaluation Protocol and Metrics

Since there is no public benchmarking to tackle GSUs [13], [14], we assessed our methodology using the most commonly utilized datasets found in related research, MovieLens 100K (ML) [26] and FilmTrust (FM) [27], as shared in Table I.

To investigate the efficiency of our model in detecting GSUs, we employed the mean absolute error (MAE) as an evaluation metric. This metric quantifies the system's ability to reflect the user preferences. As a result, a GSU should have a high mean absolute error which demonstrates system failures in mining preferences. Moreover, we evaluatedthe MAE variation according to theuser type. We utilized the metrics introduced in [19], which denote the MAE increase

TABLE I. THE PROPRIETIES OF THE USED DATASETS

Dataset	#Users	#Items	#Ratings	Scale
MovieLens 100K (ML)	943	1682	1000	1-5
FilmTrust(FT)	1508	2071	35497	0.5-4

$Inc(MAE)$ for grey sheep users and MAE improvement $Imp(MAE)$ for normal users.

$$Imp(MAE) = 1 - \frac{MAE_{NU}}{MAE_{AllUsers}} \quad (19)$$

As GSUs preferences can also affect the overall quality of the system, theoretically excluding them from the recommendation should improve the suggestion. Eq. (19) quantifies the possible improvement for the remaining users in terms of MAE variation, where approaching 1 indicates the enhancement of the recommendation.

$$Inc(MAE) = 1 - \frac{MAE_{NU}}{MAE_{GSU}} \quad (20)$$

Eq. (20) was designed to evaluate the performance of the system in detecting grey sheep users, where values approaching 1 reflect the system's ability to identify these users. The MAE is obtained using Eq. (21), where UG is one of the user groups: all users, grey sheep users, or normal users.

$$MAE_{UG} = \frac{\sum_{(u,i)\in UG} |\hat{r}_{u,i} - r_{u,i}|}{\#UG} \quad (21)$$

To calculate the predicted ratings, we implemented User-Based Collaborative Filtering (UBCF) [28], as it highly depends on the nearest users to calculate the rating, which will help in assessing the performance of our model in detecting GSUs, where the ratings are predicted with:

$$\hat{r}_{u,i} = \tilde{r}_u + \frac{\sum_{(u')\in N_u} (r_{u',i} - \tilde{r}_{u'}) \times sim(u,u')}{\sum_{(u')\in N_u} sim(u,u')} \quad (22)$$

where,

- $\tilde{r}_u$ and $\tilde{r}_{u'}$: are the average ratings of the users u and u'.
- N_u: is the set of nearest users to the user u.
- $r_{u',i}$: is the rating r given by the user u' to the item i.

GTMedoids is based on an enhanced variation of the standard Kmeans clustering approach, in which the Euclidean distance is primarily used [see Eq. (23)]:

$$Euclidean(u_a, u_b) = \sqrt{\sum_{i=1}^{n}((r_{u_a,i} - r_{u_b,i})^2)} \quad (23)$$

To demonstrate the need for the merged dissimilarity metric we compared the results obtained by the approach using Euclidean and our metrics. Where the user taste prediction is computed using the same technique in Eq. (22) with the Euclidean distance instead of the merged dissimilarity metric [see Eq. (24)]:

$$\hat{r}_{u,i} = \tilde{r}_u + \frac{\sum_{(u')\in N_u} (r_{u',i} - \tilde{r}_{u'}) \times Euclidean(u,u')}{\sum_{(u')\in N_u} Euclidean(u,u')} \quad (24)$$

In addition to all the metrics cited above, we introduce the average rating variance metric, where the rating variance of an item i is calculated as Eq. (25):

$$Variance_i = avg(r_{GSU,i}) - avg(r_{NU,i}) \quad (25)$$

where,

- $avg(r_{GSU,i})$ is the average rating of all the grey sheep users GSU who interact with item i.
- $avg(r_{NU,i})$ is the average rating of all normal users NU who interact with item i.

The $Variance_i$ metric is able to quantify the deviation of the grey sheep users' preferences from the normal ones, where values approaching 0 indicate a similar taste, while values approaching the maximum rating option represent a high deviation in the taste.

B. Experiments Setup and Configuration

To differentiate between positive and negative feedback, an average rating serves as a threshold favoring positive reviews, as ratings often indicate user interest, according to existing literature. The rating thresholds are set at 2.5 for FilmTrust and 3 for MovieLens. Furthermore, since an improved version of the Kmeans algorithm is part of this approach, it is crucial to determine the number of clusters (K) before executing the clustering algorithm. The elbow method was employed to identify the optimal number of clusters.

Based on the experimental results, the optimal K for $MovieLens100K$ and $FilmTrust$ is 3, as illustrated in Fig. 3. Therefore, for each dataset, we clustered the users into three clusters.

C. Results Analysis

Our model identified 273 of 943 users and 292 of 1508 users as grey sheep candidates on Movielens and FilmTrust, respectively, as shown in Table II, where 121 and 129 of them were detected as grey sheep users grouped in two grey clusters for both datasets.

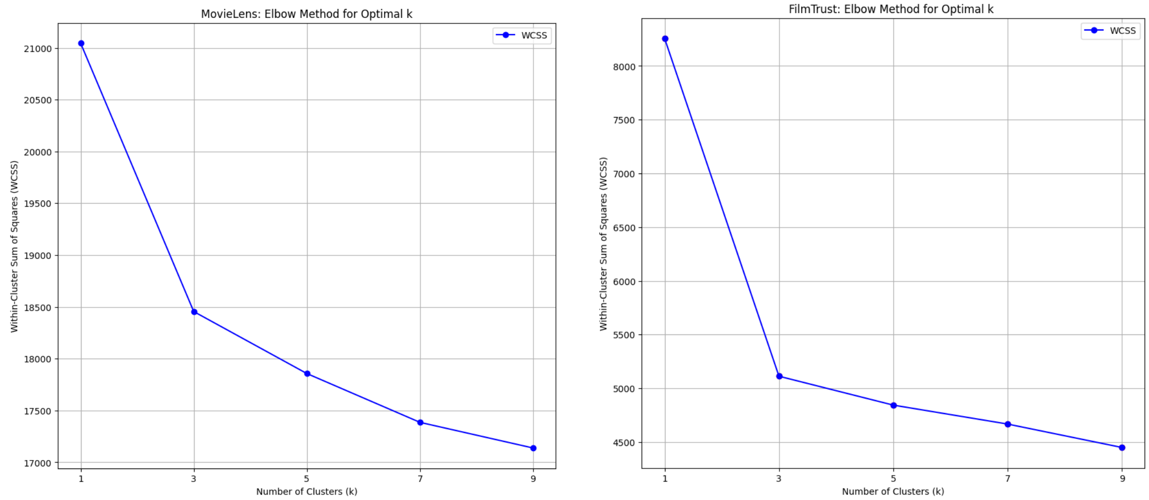

Fig. 3. Elbow method for identifying optimal K.

TABLE II. GREY SHEEP USERS DETECTION PER DATASET

Dataset	#users	#clusters	#grey Clusters	#GS candidate	#GSUs
ML100k	943	3	2	273	121
FM	1508	3	2	292	129

In assessing the validity of results, the impact of closest users on individual recommendations was measured. Grey sheep users exhibit high MAE values due to their closest users being more distantly located compared to the neighbors of normal users. This results in grey sheep consistently receiving inaccurate suggestions, as illustrated in Fig. 4, which compares MAE across user types.

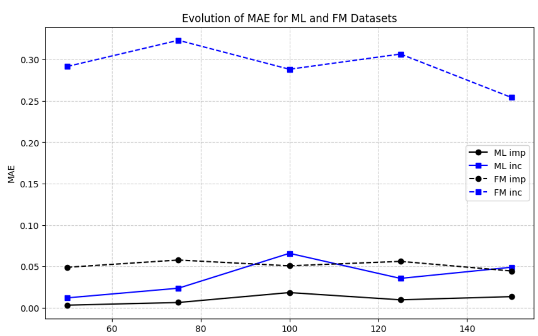

Fig. 4. MAE variations by user type.

The best results were achieved for K=75 and K=100 for FM and ML, respectively, where the mean absolute error almost doubled for the FM grey sheep users and increased by 6.616% for ML, as shown in Table III.

We propose GTMedoids, a clustering approach that categorizes users based on their greyness degree, contrasting with traditional methods that enhance centroid selection. Our model's efficiency is demonstrated through a comparison with existing literature [19], and the results are presented in Table IV. Note that FilmTrust has shown better outcomes, which can be justified by the RT precision as the rating scale is 0.5 and also the more valuable number of user-item interactions provided in this dataset.

Our approach has surpassed existing solutions in detecting grey sheep users while also enhancing recommendations for normal users, despite its primary focus on grey sheep selection within the community. This effectively supports the rationale behind our concept. Additional analysis was conducted by comparing average ratings of items rated by both user types, results of which are illustrated in Fig. 5. In the figure, red data points represent the average ratings given by grey sheep users, while blue data points reflect the overall average ratings from normal users for the same items. The figure clearly highlights the differences in opinion between the user types, evidenced by the distinct data point distributions for each item.

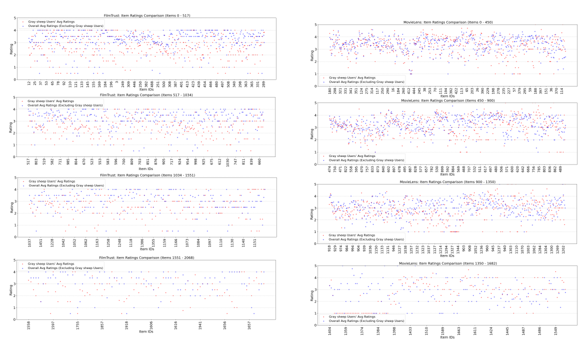

Fig. 5. Grey sheep and normal users average rating comparison.

To assist with the complexity of numerous data points, we calculated the average rating variance between grey and normal sheep users using Eq. 25.

The fluctuation presented in Fig. 6 illustrates the grey sheep user preferences deviation from the global collaborative taste and proves that these users rarely agree with the rest of the users when the difference is zero.

[1]Note: The results shared in this table were taken from the original work [19].

TABLE III. GTMedoids' Performance in Terms of MAE Variation by User Type

	All Users	Normal Users	GSUs	$MAE_{NU}\%$	$MAE_{GS}\%$
MAE (FM) k=75	0.6776	0.6385	0.9434	-5.77%	$+32.31\%$
MAE (ML) k=100	0.8488	0.8332	0.8919	-1.83%	$+5.07\%$

TABLE IV. The Performance of GTMedoids in Terms of Grey Sheep Users Detection

Methods	$Inc(MAE)$ ML	$Imp(MAE)$ ML	$Imp(MAE)$ FM	$Inc(MAE)$ FM
KMeans Standard[1]	0.027044	0.008989	0.014441	0.090519
KMeans++[1]	0.030882	0.007733	0.013397	0.089948
KMeans Pluspower[1]	0.037215	0.009111	0.018492	0.088580
Power Weight + Distance[1]	0.041141	0.011093	0.021348	0.093265
Power Weight Item + Distance[1]	0.029595	0.010316	0.020630	0.098378
GTMedoids	**0.065814**	**0.018378**	**0.057703**	**0.323192**

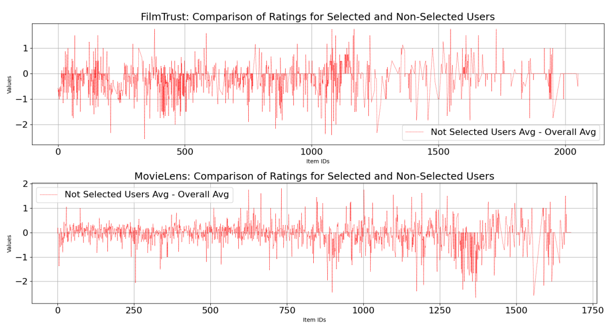

Fig. 6. The variation of ratings between grey sheep and normal users.

As we proposed a new dissimilarity function, we studied the impact of it instead of using Euclidean distance. To run the model using the traditional approach, we must determine the optimal number of clusters. As illustrated in Fig. 7, K was 3 for both ML and FM datasets.

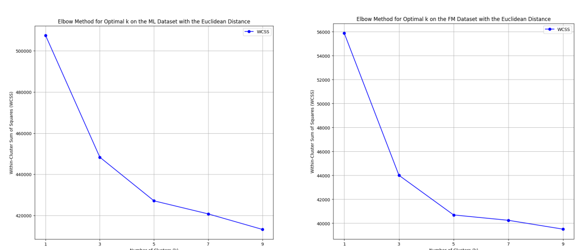

Fig. 7. Elbow method for identifying optimal K with the Euclidean distance.

By implementing the approach using the Euclidean distance, we obtained 155 and 142 users selected as GSUs in FM and ML, respectively. Fig. 8 shows the importance considering the feedback context while detecting GSUs in the system. Where the model was more accurate with our newly presented function such that not only fewer users were handled as GSUs but also the MEA increased. The selected users with the merged dissimilarity metric were also tested with the standard Euclidean function to predict their preferences. However, the prediction was less accurate for these users compared to the users detected and tested using only the Euclidean function. Therefore, the merged dissimilarity metric is more precise in detecting users who represent a minority in terms of taste and preference uniqueness.

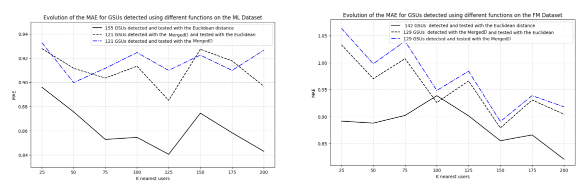

Fig. 8. The merged dissimilarity metric efficiency in detecting GSUs in comparison with the Euclidean distance.

This section experimentally demonstrates the performance of our model and its efficiency in detecting grey sheep users in the system. GTMedoids was able to identify users with unique preferences that the system often failed to recommend to them accurately.

V. Conclusion

In this study, we introduce Grey Threshold Medoids (GTMedoids), a novel method for identifying grey sheep users. The initial phase of our approach involves filtering users based on their frequency of agreement with the community's preferences, specifically isolating those who infrequently agree and frequently disagree with prevailing tastes for subsequent clustering. This selection process acknowledges the complexity of individual tastes, characterized as bi-polar, meaning that interaction with a particular item does not guarantee user appreciation. Clustering is conducted via a novel personalized metric that evaluates the significance of both positive and negative feedback from users while assessing their similarities. However, the clusters generated by this method do not inherently indicate which should be classified as grey sheep clusters. To address this, we have devised a new selection mechanism applicable beyond the realm of recommender systems. We compute the power of each cluster based on grey sheep user characteristics and finally identify the less-influencing clusters as grey sheep clusters that group our grey sheep users. The experiments proved the validity and accuracy of the proposed approach

As GTMedoids is based on the rating matrix in the future, it can be integrated to detect grey sheep users in pre-existing

RSs in the literature so that they can receive special treatment while making recommendations. Another future direction is to improve GTMedoids itself and not to be limited to just GSU detection but also by adding the recommendation module to it to treat each group of users fairly, either by adding KG graphs as a supplementary source of knowledge to strengthen user and item embedding or by leveraging the attention mechanism in such a way that the model learning phase can focus effectively on what is important to these users.

REFERENCES

[1] W. Zhang, X. Zhang, H. Wang, D. Chen, A deep variational matrix factorization method for recommendation on large scale sparse dataset, Neurocomputing 334 (2019) 206–218. doi:10.1016/j.neucom.2019.01.028.

[2] N. Idrissi, A. Zellou, A systematic literature review of sparsity issues in recommender systems, Social Network Analysis and Mining 10 (12 2020). doi:10.1007/s13278-020-0626-2.

[3] T. Trinh, V.-H. Nguyen, N. Nguyen, D.-N. Nguyen, Product collaborative filtering based recommendation systems for large-scale e-commerce, International Journal of Information Management Data Insights 5 (1) (2025) 100322. doi:https://doi.org/10.1016/j.jjimei.2025.100322.

[4] A. Srivastava, P. K. Bala, B. Kumar, New perspectives on gray sheep behavior in e-commerce recommendations, Journal of Retailing and Consumer Services 53 (2020) 101764. doi:https://doi.org/10.1016/j.jretconser.2019.02.018.

[5] A. A. Kardan, M. Ebrahimi, A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups, Information Sciences 219 (2013) 93–110. doi:10.1016/j.ins.2012.07.011.

[6] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, WWW '01, Association for Computing Machinery, New York, NY, USA, 2001, p. 285–295. doi:10.1145/371920.372071.

[7] G. Behera, N. Nain, Collaborative filtering with temporal features for movie recommendation system, Procedia Computer Science 218 (2023) 1366–1373, international Conference on Machine Learning and Data Engineering. doi:https://doi.org/10.1016/j.procs.2023.01.115.

[8] A. Tourani, H. A. Rahmani, M. Naghiaei, Y. Deldjoo, Capri: Context-aware point-of-interest recommendation framework, Software Impacts 19 (2024) 100606. doi:https://doi.org/10.1016/j.simpa.2023.100606.

[9] B. Boualaoui, A. Zellou, M. Berquedich, Knowledge graph-based recommender systems to mitigate data sparsity: A systematic literature review, International Journal of Interactive Mobile Technologies (iJIM) 19 (03) (2025) pp. 115–140. doi:10.3991/ijim.v19i03.49427.

[10] Y. Zheng, An empirical comparison of outlier detection methods for identifying grey-sheep users in recommender systems, in: L. Barolli (Ed.), Advanced Information Networking and Applications, Springer Nature Switzerland, Cham, 2024, pp. 62–73.

[11] B. Gras, A. Brun, A. Boyer, Identifying grey sheep users in collaborative filtering: A distribution-based technique, in: Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, UMAP '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 17–26. doi:10.1145/2930238.2930242.

[12] Y. Zheng, M. Agnani, M. Singh, Identification of grey sheep users by histogram intersection in recommender systems, in: G. Cong, W.-C. Peng, W. E. Zhang, C. Li, A. Sun (Eds.), Advanced Data Mining and Applications, Springer International Publishing, Cham, 2017, pp. 148–161.

[13] B. Boualaoui, A. Zellou, L. Ajallouda, Towards a new evaluation protocol for grey sheep user detection approaches, in: M. Kodad, O. Moussaoui (Eds.), The 3rd International Conference on Artificial Intelligence and Smart Applications (AISA'25), Volume 1: Artificial Intelligence, IoT, and Smart Applications, Springer Nature Switzerland, Cham, 2026, pp. 13–25. doi:10.1007/978-3-032-18716-1_2.

[14] R. Alabdulrahman, H. Viktor, Catering for unique tastes: Targeting grey-sheep users recommender systems through one-class machine learning, Expert Systems with Applications 166 (2021) 114061. doi:https://doi.org/10.1016/j.eswa.2020.114061.

[15] H.-P. Kriegel, M. Schubert, A. Zimek, Angle-based outlier detection in high-dimensional data, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, Association for Computing Machinery, New York, NY, USA, 2008, p. 444–452. doi:10.1145/1401890.1401946.

[16] M. Ghazanfar, A. Prugel-Bennett, Fulfilling the needs of gray-sheep users in recommender systems, a clustering solution, in: 2011 International Conference on Information Systems and Computational Intelligence (18/01/11 - 20/01/11), 2011, event Dates: 18-20, January.

[17] M. A. Ghazanfar, A. Prügel-Bennett, Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems, Expert Systems with Applications 41 (7) (2014) 3261–3275. doi:https://doi.org/10.1016/j.eswa.2013.11.010.

[18] X. Amatriain, J. M. Pujol, N. Oliver, I Like It... I Like It Not: Evaluating User Ratings Noise in Recommender Systems, in: User Modeling, Adaptation, and Personalization, Springer, Berlin, Heidelberg, 2009, pp. 247–258. doi:10.1007/$978-3-642-02247-0_24$.

[19] R. Rashidi, K. Khamforoosh, A. Sheikhahmadi, An analytic approach to separate users by introducing new combinations of initial centers of clustering, Physica A: Statistical Mechanics and its Applications 551 (2020) 124185. doi:https://doi.org/10.1016/j.physa.2020.124185.

[20] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Computing Surveys 41 (3) (2009) Article 15, 58 pages. doi:10.1145/1541880.1541882.

[21] B. Kaur, S. Rani, Identification of gray sheep using different clustering algorithms, in: D. Goyal, A. K. Gupta, V. Piuri, M. Ganzha, M. Paprzycki (Eds.), Proceedings of the Second International Conference on Information Management and Machine Intelligence, Springer Singapore, Singapore, 2021, pp. 211–217.

[22] H. Jindal, S. Agarwal, N. Sardana, Powkmeans: A hybrid approach for gray sheep users detection and their recommendations, Int. J. Inf. Technol. Web Eng. 13 (2) (2018) 56–69. doi:10.4018/IJITWE.2018040106.

[23] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27. doi:10.1109/TIT.1967.1053964.

[24] B. Gras, A. Brun, A. Boyer, Can matrix factorization improve the accuracy of recommendations provided to grey sheep users?, in: Proceedings of the 13th International Conference on Web Information Systems and Technologies - WEBIST, INSTICC, SciTePress, 2017, pp. 88–96. doi:10.5220/0006302700880096.

[25] A. Nowak-Brzezińska, I. Gaibei, How the outliers influence the quality of clustering?, Entropy 24 (2022) 917. doi:10.3390/e24070917.

[26] F. M. Harper, J. A. Konstan, The movielens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (4) (Dec. 2015). doi:10.1145/2827872.

[27] G. Guo, J. Zhang, N. Yorke-Smith, A novel evidence-based bayesian similarity measure for recommender systems, ACM Trans. Web 10 (2) (May 2016). doi:10.1145/2856037.

[28] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94, Association for Computing Machinery, New York, NY, USA, 1994, p. 175–186. doi:10.1145/192844.192905.

[29] C. C. Aggarwal, P. S. Yu, Outlier detection for high dimensional data, SIGMOD Record 30 (2) (2001) 37–46. doi:10.1145/376284.375668.

[30] Y. Mo, R. Xing, H. Hou, A distributed higher-order k-medoids clustering algorithm for network partition, IEEE Transactions on Network Science and Engineering 11 (5) (2024) 4181–4193. doi:10.1109/TNSE.2024.3402383.

[31] M. Tiwari, M. J. Zhang, J. Mayclin, S. Thrun, C. Piech, I. Shomorony, BanditPAM: almost linear time k-medoids clustering via multi-armed bandits, in: Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS), 2020, Article 856, pp. 1–12. doi:10.5555/3495724.3496580.