International Journal of Industrial Engineering & Management Science, Vol. 8, Issue 1, (2021) 16-23

University of Hormozgan

International Journal of Industrial Engineering & Management Science

journal homepage: www.ijiems.com

Designing a Memory-Based Collaborative Filtering Group Recommender System to Confront the Cold Start Phenomenon

Ehram Safari [a,*], Mozhgan kamari [b]

[a] *Assistant Professor, ICT Research Institute (Iran Telecommunication Research Center), Tehran, Iran*
[b] *Department of Information Technology Science and Research branch, Islamic Azad University Tehran, Iran*

A R T I C L E   I N F O

Article history:
Received: 2021-07-25
Received in revised form: 2022-02-17
Accepted:
2022-02-27

Keywords:
Cold Start Phenomenon
Sparsity Problem
Collaborative Filtering Technique
Group Recommendation System
Memory-Based Approach

A B S T R A C T

Today, people devote more of their time on social networks. In these fields, users need to make sure of activity together and ride them as a group called group recommendation systems. The primary objective of this approach is to propose one or more entities to a group of individuals to maximize the requests and benefits of that group of individuals. Collaborative filtering approaches are widely employed in these procedures and are based on a complete initial ranking in the user-item matrix. However, in the real system, this matrix is still sparse, and the priority of users is unknown. This problem can make memory-based collaborative filtering unsuitable for group recommendation systems. Many types of research have been done to solve these systems' cold start and sparsity problems. However, unlike the developed approaches that emphasize the problem of the sparse item-user matrix in individual recommendation systems, the approach of this research is solving this problem is the group recommendation systems and tries to provide an optimal solution for the sparse matrix of the user-item. The central part of the proposed method is based on a multilayer perceptron that computes the similarity between items. It is indicated that the proposed method gives group members more satisfaction with the other five existing algorithms.

1. Introduction

Recommender systems are used in many areas and are most commonly recognized to offer personalized information. Much research has been done on these individual recommendation systems (IRSs). In these systems, the likelihood of each person's interest in a particular product or item is obtained from the preferences and interests of his/her previous activities. These days, Recommender systems for groups are becoming increasingly popular since people have much activity on social media.

Overall, they are no longer IRSs, and they are group recommender systems (GRSs) that endorse items to groups of users. Their information and data require created from the group and social activities, such as traveling, listening to music, watching movies, etc.[1] Some famous GRSs, have been devised and utilized in the last few years. GRec OC (Group Recommender for Online Communities) [2]is a book recommender system for online communities (i.e., people with similar interests that share information). The system seeks to enhance the pleasure of individual users. Jukola [3] and PartyVote [4] are two systems capable of delivering music to a regular social group of people attending a party/social event. INTRIGUE [5] suggests locations for tourist groups, considering the features of subgroups within that group (such as children and the disabled). The Travel Decision Forum [6] enables a

* Corresponding author.
E-mail address: e.safari@itrc.ac.ir

group to arrange the preferred characteristics of a scheduled joint holiday. Users indicate their choice on a set of attributes.

The approach combines the individual preferences for each feature, and users cooperate with embodied conversational agents depicting other group members to get an agreement to take group choice. In GRS, each member's intentions may be alike or distinct. Obtaining the most typical selection of group members and making the recommendation items likely to satisfy all group members' requirements is an essential topic to be managed. There are two primary methods for group recommendation: (1) collection of each person's ratings and (2) collection of individual recommendation lists.[7] In the first strategy, all individual choices are combined by distinct collection approaches like average or minor misery for each candidate item to calculate the item's group rating. After that, items with the highest group ratings are suggested. Although in the second strategy, mentioned lists for each member are joined into a single list to endorse to a group. Also, to fulfill this goal, it is essential to have choices of the groups to items. Consequently, it is essential to forecast anonymous ratings in each member's choice list. As one of the most famous methods to recommender systems, collaborative filtering has been shown to be effective for solving vast data problems. This method can notice resemblances between users and items, but it weakly meets a sparse selection matrix.

Some investigators attempt to solve this problem by defaulting to the rating value strategy. This method satisfies all unfamiliar choices with a default value like neutral or average ones[8]. Other approaches to solve this problem by means of classification algorithms or imputation techniques like mean imputation to satisfy in anonymous ratings[9]. The primary problem with these researches is that they satisfy all missing ratings with constant values far from fact and do not believe the natural variances and other parameters. The existent approaches to solving the sparse user-item matrix do not have the essential effectiveness. Hence, in this article, we attempt to give a new path to explain the problem of the sparse user-item matrix so that it has the highest accuracy. Opposing the methods that have been technologically advanced so far and emphasizing the problem of the sparsity of the user-product matrix in single recommender systems, this study aims to solve the problem of the sparsity of the matrix in group recommender systems.

This paper desires to suggest music to a group of randomly given users who have ordered a few items. Moreover, utilize the memory-based CF technique in the sparse user-item matrix. One of the significant parts of our model is the multilayer perceptron (MLP) to foresee resemblances between items and employ it for forecasting unclear values in the user-item matrix. Unlike the designed methods that stress the problem of the sparse item-user matrix in IRSs, this research's method to solving this problem is the matrix in the GRSs and attempts to deliver an optimal solution for the sparse matrix of the user-item. It furthermore solves the problem of adding a novel item to this system, termed the cold start phenomenon.

The remainder of this article in Section 2 shows corresponding efforts in solving sparse user-item matrix. Section 3 defines the primary approaches that have been utilized in general Section 4 describes the suggested technique in detail. Section 5 offers experiments on the offered approach and compares its accuracy with main methods. Eventually, Section 6 completes the paper.

2. Related Work

Collaborative filtering (CF) is the most successful and used technique in preparing recommender systems where items to an active user will be suggested based on the past rating records from like-minded users. Unfortunately, CF may conduct poor recommendations when user ratings on items are very sparse compared to the massive number of users and items in the user-item matrix. [10] apply the users' implicit interaction records with items to efficiently process enormous data by mining association procedures to overcome this problem. It includes the several buying per transaction in association procedures rather than counting total purchases made. [11] uses matrix factorization as a prevalent method for collaborative filtering in recommendation systems that calculates the latent factors for users and items by decomposing a user-item rating matrix. [12] provides a novel genetic algorithm-based recommender system, SimGen, that calculates user resemblance values short of spending well-known resemblance metric calculation algorithms like Pearson correlation and vector cosine-based similarity. The outcomes improve forecast quality and performance, respectively, compared with other methods. The paper [13] suggests a recommender system for travel scheduling built on personalized regard for single and group users by means of the flexible user interface and feedback mechanism. It offers an suitable and exciting travel timetable to the user. Results: First, the flexible user interface is utlized to adjust or eliminate the displeased travel timetable of the user with the detailed timetable. Next, the feedback mechanism makes available a better accuracy rate for the following timetable of the first-hand user. The group recommendation system's suggested hybrid collaborative filtering method resolves the data sparsity problem. Along with this, the K-Meansclustering algorithm is utilized to cluster the users and to group them according to their benefits proficiently. The consequence of collaborative filtering is frequently exposed by sparsity, cold start, and grey sheep problems. To solve these issues in [14], a new collaborative filtering algorithm entitled Altered Client-based Collaborative Filtering (ACCF) for group recommendation is planned. ACCF uses Dragonfly Algorithm to handle sparsity and neighbor selection. The restaurant recommendation system is used as a testbed for the validation of ACCF. [15] devises a novel hybrid recommendation technique is dependent upon CF methods. Consequently, this study deals with two primary disadvantages of recommender systems, sparsity, and scalability, utilizing dimensionality decrease and ontology techniques.

The [16] proposal offers a methodology to enhance Group Recommender Systems. The framework configuration comprises a procedure where an influential group is detected among the target groups to suggest. In order to assist the group members to agree and come to a decision, the visualization of the alternative selected by the powerful group and the reasons why they assumed that recommendation is presented for the target exposed group. [17] devised a novel formulation for recommender systems based on projective nonnegative matrix factorization but diminishes the non-negativity constraint. Conducted by instructive yet direct insight, the offered formulation offers favorable and regular consequences that depend on a minimal number of parameters. This investigation [18] studied the influence that aggregation functions and likeness actions had on the accuracy of memory-based collaborative filtering algorithms, precisely when the actual rating that a user would deliver differs meaningfully from the rating that the average user would supply. [19], offered a group recommendation structure utilizing users' profiles and collaborative filtering over the social networks. The offered structure brings up-to-date profile information by gathering recent user activities. It is revealed through the performance assessment that the offered structure delivers several group recommendations in which users' altering preferences are considered adequately compared to the existing structures. This article [20] suggests a hybrid recommendation algorithm based on CF and music genes and design a personalized music recommendation. In this paper[21], a novel influential collaborative filtering algorithm based on user preference clustering is proposed to lessen the impact of the data sparsity. First, user groups are introduced to determine users with different preferences. Then, regarding the selection of the active user, we get the nearest neighbor set from the corresponding user group/user groups. Similarly, a new similarity measure approach is proposed to preferably compute the similarity between users, which assumes user selection in local and global perspectives. Finally, experimental results demonstrate that the proposed algorithm effectively enhances the performance of recommender systems. [22] enhances memory-based techniques for group recommendation systems by resolving the data sparsity problem. The essence of the suggested approach is based on a support vector machine learning model that calculates resemblances between items. This process utilizes calculated resemblances and improves basic memory-based methods. Experiments reveal that the proposed approach confounds the memory-based methods. It furthermore exhibits that the offered work exceeds the latent factor method, which is very efficient in light conditions. Ultimately, it is indicated that the proposed approach offers better performance than existing approaches on generating group recommendations. This paper wants to improve performance and prediction accuracy for unknown ratings in sparse user-item matrices based on [22].

3. Overview

This section reviews the CF definition and its approaches. CF's fundamental assumption is that if users X and Y rate N items have identical manners (e.g., buying, watching, listening) and therefore will rate or function on other items similarly [23]. CF approaches utilize a database of selections for items by users to forecast extra cases or products a new user might like, called the user-item matrix. There is a list of M users and N items in a standard CF method, and each user has a rate on item I that is rui [24]. There are many challenges for collaborative filtering tasks. CF algorithms are needed to have the ability to deal with highly sparse data, to climb with the increasing numbers of users and items, to make suitable recommendations in a short time, and to deal with adding a new item.

3.1. User-Based Collaborative Filtering

This approach concentrates on the "nearest neighbor" method for recommendations, examining the user's rating patterns and finding the nearest neighbors, i.e., users with ratings identical to yours. The algorithm then moves to provide you with recommendations based on the ratings of these neighbors [25]. The weakness of this method emerges when the user-item matrix is sparse, and therefore the set of standard items in both users' selection lists becomes small or sometimes null. Most of the time, for computing resemblance between two users, one of the most standard resemblance measures, Pearson's correlation (eq. 1), is used as follows[26]:

$$s(a,u) = \frac{\sum_{i=1}^{n}(r_{a,i}-\overline{r_a})(r_{u,i}-\overline{r_u})}{\sqrt{\sum_{i=1}^{n}(r_{a,i}-\overline{r_a})^2}\sqrt{\sum_{i=1}^{n}(r_{u,i}-\overline{r_u})^2}} \tag{1}$$

Anticipating user u's rating on item i (eq. 2) is calculated as down [27]:

$$p(a,i) = \overline{r_a} + \frac{\sum_{i=1}^{n}(r_{u,i}-\overline{r_u}) \times s(a,u)}{\sum_{i=1}^{n} s(a,u)} \tag{2}$$

3.2. Item-based Collaborative Filtering

Item-based filtering inspects the resemblance between different items. It does this by bringing information of how many users rated item X furthermore rated item Y. If the correlation is high sufficiently, a similarity can be assumed to exist between the two items, and they can be supposed to be similar to one another. Item Y will subsequently be suggested to users who rated item X and vice versa [28]. For calculating the similarity between two items, we can utilize various approaches like Pearson's correlation or cosine-based similarity.

4. Proposed Approach

This section describes the suggested approach. In this paper, we aim to handle these problems that original CF naturally sorrows from:

Data sparsity derives from the phenomenon that users commonly rate only a limited number of items.

Cold start directs to the hardship of bootstrapping the RSs for new users or items. That we concentrate on new items.

The focus of CF is to aggregate the ratings of like-minded users. Nevertheless, the declared matrix of user-item ratings is usually very sparse (up to 99%) due to users' lack of understanding or motivation to rate items. In addition, they typically report or obtain only a few or no ratings for the new users or new items.

In public, the core of CF is to enhance the capability for discovering correct and trustworthy neighbors of active users. Nevertheless, gathered data is excessively sparse in the user-item rating matrix. Meanwhile, numerous existent similarity measure techniques utilized in collaborative filtering are not much influential, which leads to poor performance, so in In this paper, we desire to enhance the performance of the CF approach.

Our approach focuses on the forecast process, solving the sparsity Problem, and the cold start of adding new items. It includes the subsequent subsections:

- Forecast process
- Item similarity
- User similarity
- Anticipating user's unknown ratings on items
- Developing group recommendations

Both memory-based CF methods solve the sparsity problem and anticipate unknown ratings by self-prediction, with some restrictions described in the overview section.

Our approach is also based on self-prediction, but it uses more information and item data, and ratings in the user-item matrix.

Prediction Process

Multilayer Perception(MLP) Model for Computing Items' Similarity

The fundamental part of our proposed approach is a machine learning regression approach named MLP, utilized to compute similarities between items. The items are in the union list of items in each member's preference list.

The MLP is maybe the most widespread network architecture today for classification and regression. MLPs are feed-forward neural networks naturally comprised of several layers of nodes with unidirectional connections, usually trained by backpropagation. The learning method of the MLP network is based on the data samples contained the N-dimensional input vector x, and the M-dimensional expected output vector d, called destination. By processing the input vector x, the MLP delivers the output signal vector y (x, w), where w is the vector of adjusted weights. The error signal created actuates a control mechanism of the learning algorithm. The correctional adjustments are designed to make the output signal y_k (k=1, 2,...,M) to the expected response d_k in a step-by-step manner. The learning algorithm of MLP is based on the minimization of the error function described on the learning set (x_i,d_i)for i=1,2,...,N.

There are various activation functions in MLP like identity, logistic, relu, tanh. The subsequent section will assess various functions and parameters like learning rate and hidden layer. We select the one with the best functionality and performance. Subsequently, this paper displays the output of this trained model by $SI'_{i,j}$, which implies the similarity value of two items. Approach for calculating user's similarity.

Our proposed approach desires to overwhelm the drawback of Pearson's correlation method in sparse situations. The Pearson's correlation technique for calculating the similarity of users is restricted to everyday items in both users' rated item lists. In sparseness matrices, with lots of empty cells in the user-item matrix the collections of standard items are small, so comparing user's preferences are dependent upon a few items, which drives lower accuracy and performance. In order to solve this problem, the concept of the proposed approach is as down:

- Comparing every item rated by one member with all items in another member's preference list, one by one.
- Involved in all possible combinations of items in rated item lists of both members.
- Assumed similarity of users on this is dependent upon their close ratings for similar items.

$$NSU = \frac{\sum_{\forall i \in R_U \cap \forall j \in R_V} \left(1 - \frac{|r_{ui} - r_{vj}|}{r_{\max} - r_{\min}}\right) \times SI'_{ij}}{\sum_{\forall i \in R_U \cap \forall j \in R_V} SI'_{ij}}, \qquad (3)$$

$$R_u = \{i \mid r_{ui} \neq 0\}, R_v = \{j \mid r_{vj} \neq 0\}$$

From Eq. 3, it is evident that if two users have rated two similar items, the related part of the numerator of Eq. 3 becomes one, and thus, the similarity measure of two users becomes bigger.

Predicting User's Unknown Ratings on Items

The prediction method in [22] considered i_0 as the most similar item to item i that user v has rated it and computed similarities between users by Eq. 4. In our method, we check the sparse matrix and realize a similar item is insufficient. In other words, rating on a similar item is unknown. So, our proposed method considered three similar items rated for increasing accuracy and performance of predicting unknown cells in the user-item matrix. As mentioned before, the similarity between items is computed according to the MLP model.

$$p(a,i) = \bar{r}_a + \frac{\sum_{i=1}^{n} (r_{u,i} - \bar{r}_u) \times s(a,u)}{\sum_{i=1}^{n} s(a,u)} \qquad (4)$$

Generating Group Recommendation

For developing group recommendations, we require to utilize collections typically in GRSs as down:

- Unleased misery: assumed the minimum value of item ratings in all users' preference lists (Eq. 5).

$$P_{Gi} = \min(R_{ui}), R_{ui} = r_{ui} \, or \, P_{ui} \qquad (5)$$

Average function: calculates the average of users' rates for items and suggests items with the highest values:

$$P_{Gi} = \frac{\sum_{u=1}^{m} R_{ui}}{M}, R_{ui} = r_{ui} \, or \, P_{ui} \qquad (6)$$

5. Experimental Results

Our suggested approach is examined with Million Song Dataset (MSD[1]) in this section. The dataset includes standard songs' songs, such as artist name, title, and year released. Further, the data includes more developed information, for example, artist familiarity, the length of the song, artist hotness. Furthermore, they utilize user ratings to items table named user-item matrix[2] for recommending music to a group of users.

The primary objective is to study the precision of the recommended approach in anticipating unknown rated items in the user-item matrix and measure group members' satisfaction. The systems presented so far do not support the ability to modify the concept, and if users modify their minds, the training should be repeated.

Evaluation Data

The specified evaluation data arrives from the MSD, which utilized the Echo Nest to emanate data points about one million famous recent songs. The MSD cooperates between the Echo Nest and Lab ROSA, a laboratory working towards intelligent machine listening. The data contains standard information about the songs as below:

Song, title, release, artist, terms, artist_hotness, artist_familiarity, duration, year

We chose this dataset for evaluation because we required a dataset with the song's complete characteristics to train our MLP model in calculating songs similarities. So we require to qualify data for utilizing it in our offered model. As noted earlier, we have employed the MLP model for forecasting unknown ratings in the user-item matrix. At the start phase of training the approach, it was essential to test the accuracy of the trained MLP model. For this purpose, the following pre-processes were done:

Deformation of Data

In order to utilize resemblance data between songs and design an MLP model on them, it was required to qualify relevant data for the training process. First, we require to aggregate information from various datasets in MSD, two songs, their properties, and similarity degree between them evolved as follows:

Song_x, title_x, release_x, artist_x, terms_x, artist_hotness_x, artist_familiarity_x, duration_x, year_x, Song_y, title_y, release_y, artist_y, terms_y, artist_hotness_y, artist_familiarity_y, duration_y, year_y

Feature Selection

Due to the low number of attributes available from the available tracks and the unsuitable nature of some attributes (such as the name of the song provider), in calculating the similarity of the two tracks to each other, using attribute selection in order to create new attributes and release some unsuitable attributes. In this regard, the subsequent attributes have been released from the correlation matrix:

- Track name and codes (title and track_id and song_id)
- The company releases songs
- Name and reader information for each song (artist_name and artist_mbid)
- Year of release of each song (year)

the subsequent attributes are created from existing attributes and added to the collection tracking dataset:

- Distinction in publish year of two songs from each other
- Distinction in song duration of two tracks from each other
- The similarity/dissimilarity artist of two songs
- The similarity/dissimilarity of the publishing company of two tracks
- Distinction in the degree of familiarity of two songs from each other
- Distinction in the degree of hotness of two songs from each other data normalization

The scale of some variables varies, such as the length of the track changes with other variables. So, by using normalization approaches, the values of all variables are transmitted to the range 0 to 1, thus, the great effect of the data on modeling is eliminated.

Experimental Results of Predict Item-Similarity

To evade over-fitting and utilize all patterns in data, we employed k-fold Cross-validation to calculate the quality of a neural network. After forecasting anonymous ratings, we compared the forecasted ratings with actual values to calculate the proposed approach's accuracy. For assessing, we utilized mean absolute error (MAE), one of the widely employed evaluation metrics in CF algorithms [29].

$$MAE = \frac{\sum_{(u,i)\in T} | p_{ui} - r_{ui} |}{|T|} \quad (7)$$

The MSE metric calculates the average of absolute differences between predicted and actual values. So lower value of MAE exhibits higher accuracy of the algorithm. T represents test sets of users and items whose actual ratings r_{ui} and forecasted values p_{ui} are available. |T| is the size of the test set.

To assess different activation functions, we perform an experiment in which the MLP layers have different activation functions, including Logistic, Identity, Tanh, and Relu. According to table 1, the logistic activation function has provided better performance.

Table 1. Different Activation Function sample

Activation Function	MAE
Logistic	0.053

Identity	0.059
Tanh	0.063
Relu	0.07

To calculate the SVM model's accuracy, MAE values of different regression models were compared utilizing the Waikato Environment for Knowledge Analysis (WEKA) software tool. All parameters in different methods were tested. In all cases, the MLP had the minimum and the best MAE value (Fig. 1).

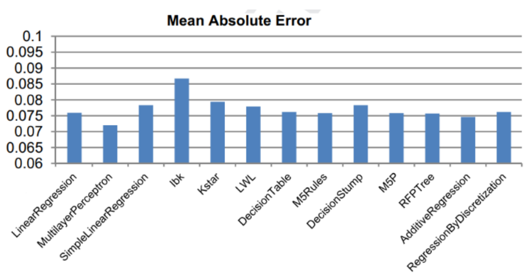

Fig. 1. MAE value for different regression methods

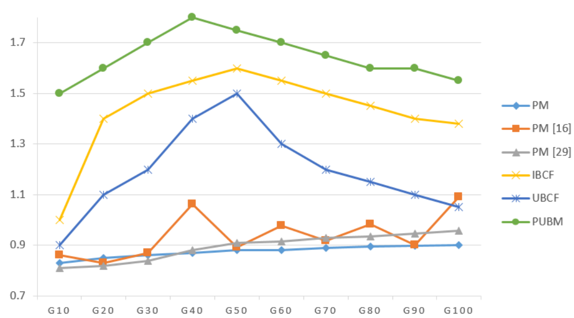

Fig. 2. Comparison of MAE metric for different CF approaches

Fig. 2 outlines the accuracy of the proposed approach (PM), suggested technique in [16] (PM [16]), suggested approach in [16] (PM [29]), basic user-based (UBCF), item-based (IBCF), Pearson user-based method (PUBM). This figure shows that the user-based approach conducts better than the item-based approach, and the suggested approach exceeds the other five techniques due to employing more information and withdrawing the restrictions of the previous approaches. The suggested approach assumes both items and users' similarities and thus has a lower prediction error. By raising the group's size, the error value utilizing our approach grows, and at the same time, it stays in 0.7–1 intervals. This is a regular performance because by increasing the number of users, the user-item matrix evolves enormous, and by employing the available information about users' ratings, the forecast procedure evolves harder, and the projection error increases. Notably, the approach handles the error rate in 0.7–1 intervals, demonstrating its ability to address it.

Experimental Results of Satisfaction

sat(G) and sat(u) offer group and user happiness.
User happiness is calculated as down:

$$sat(u) = \frac{\sum_{i=1}^{k} R_{ui}}{n \max(R_{ui})}, R_{ui} = r_{ui} \; or \; P_{ui} \tag{8}$$

Group happiness is the average of user happiness and is calculated as follow:

$$sat(u) = \frac{\sum_{\forall u \in G} sat(u)}{M} \tag{9}$$

It is time to estimate the proposed method's performance in sparse data states. To conduct experiments and research in sparse matrices, we utilized algorithms PUBM, IBCF, UBCF, PM [16], and PM [26] with the suggested algorithm (PM).

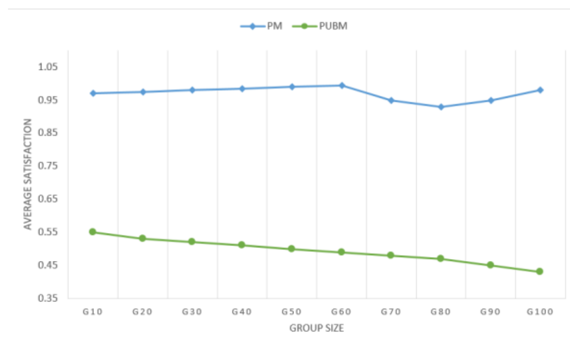

Fig.3.Comparison of group happiness values utilizing least misery function for PM and PUBM

Fig. 3 depicts a comparison between PM and PUBM. It demonstrates that our suggested approach conducts better in all groups than PUBM. Basically, on average, the MAE value of our approach is about 0.75 less than PUBM. Also, this figure represents the results of comparing group happiness values for groups in PM and PUBM cases. In this experiment, we employed our suggested approach and Pearson user-based approach for predicting unknown ratings, and later, the average part was applied for collection aims. We considered five items for suggesting to the group. Similarly, Fig. 4 depicts the effects of experiments when the least misery approach is employed for preference collection.

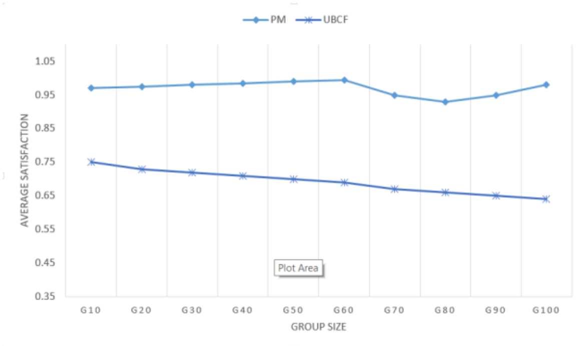

Fig.4.Comparison of group happiness values utilizing most minor misery function for PM and UBCF

As Figs. 5, 6, and 7 demonstrate that our approach has higher group happiness and, consequently, better performance on group recommendations, presently influenced by its good action in the projection phase. Figures demonstrate that by improving the number of users in each group, the average happiness of the group evolves higher. It is expected because when the number of users in a random group increases, the probability of existing users with different tastes will increase, and meeting all users evolves very hard.

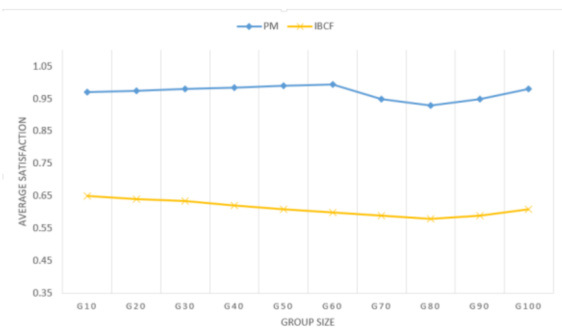

Fig.5.Comparison of group happiness values utilizing the average function for PM and IBCF

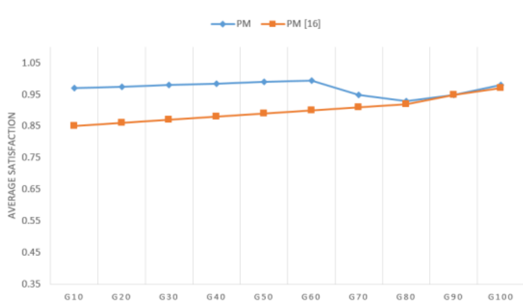

Fig.6.Comparison of group happiness values utilizing the average function for PM and PM[16]

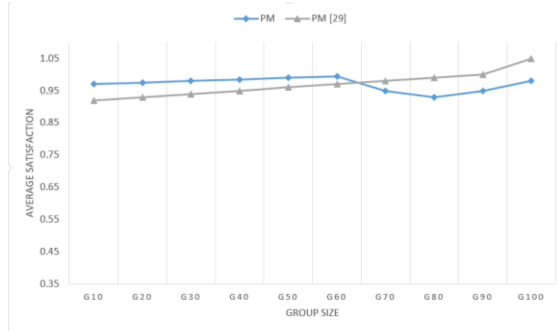

Fig.7. Comparison of group happiness values utilizing the average function for PM and PM[29]

6. Conclusion & Future Works

This research suggested a group recommendation system that presents appropriate items for users. The primary purpose of this study is to offer a group recommendation system that could solve the sparsity and cold start problem by anticipating and filling out empty cells of the user-item matrix and suggesting items that would guide to more user happiness. Unlike many memory-based collaboration filtering methods that cannot solve the sparsity issue, our system can anticipate the empty values of the user-item matrix. We deliver millions of songs to assess the accuracy of our offered approach compared with memory-based collaboration filtering methods. Our offered approach is compared with the presented procedure in [16] (PM [16]), offered technique in [29] (PM [29]), basic user-based (UBCF), item-based (IBCF), Pearson user-based method (PUBM). The outcomes demonstrate that our approach has higher happiness and lower mean absolute error than the two primary methods in the prediction process.

In our future work, we intend to assess the technique's accuracy in groups with similar or different users and analyze their impacts on the error rate in forecasting anonymous rankings. In addition, we can resume our research into higher-dimensional groups and solve the problem of scalability in a group's referential systems. This investigation utilized the correlation between items as a similarity criterion. Describing other similarity criteria can investigate its influence on the created model. As noted beforehand, three of the most similar items were used to forecast unfamiliar rates in the user-item matrix. In the coming framework of a system with more powerful hardware features, we can choose more similar items to anticipate anonymous scores in the user-item matrix and explore the effect of this method on the grade of the model.

References

[1] A. Delic and J. Masthoff, "Group Recommender Systems," in Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization, 2018, pp. 377–378.

[2] J. K. Kim, H. K. Kim, H. Y. Oh, and Y. U. Ryu, "A group recommendation system for online communities," Int. J. Inf. Manage., vol. 30, no. 3, pp. 212–219, Jun. 2010.

[3] P. O'Hara, Kenton and Lipson, Matthew and Jansen, Marcel and Unger, Axel and Jeffries, Huw and Macer, "Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques," no. ACM, pp. 145--154, 2004.

[4] D. Sprague, F. Wu, and M. Tory, "Music Selection Using the PartyVote Democratic Jukebox," in Proceedings of the Working Conference on Advanced Visual Interfaces, 2008, pp. 433–436.

[5] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso, "Tailoring the Recommendation of Tourist Information to Heterogeneous User Groups," in Hypermedia: Openness, Structural Awareness, and Adaptivity, 2002, pp. 280–295.

[6] A. Jameson, "More Than the Sum of Its Members: Challenges for Group Recommender Systems," in Proceedings of the Working Conference on Advanced Visual Interfaces, 2004, pp. 48–54.

[7] I. Christensen and S. Schiaffino, "Entertainment recommender systems for group of users," Expert Syst. Appl., vol. 38, pp. 14127–14135, 2011.

[8] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso, "Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices," Appl. Artif. Intell., vol. 17, no. 8–9, pp. 687–714, 2010.

[9] [X. Su, T. M. Khoshgoftaar, and R. Greiner, "A Mixture Imputation-Boosted Collaborative Filter.," in FLAIRS conference, 2008, pp. 312–316.

[10] M. K. Najafabadi, M. N. Mahrin, S. Chuprat, and H. M. Sarkan, "Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data," Comput. Human Behav., vol. 67, pp. 113–128, Feb. 2017.

[11] T. Xiao and H. Shen, "Neural variational matrix factorization for collaborative filtering in recommendation systems," Appl. Intell., Apr. 2019.

[12] B. Alhijawi and Y. Kilani, "Using genetic algorithms for measuring the similarity values between users in collaborative filtering recommender systems," in 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), 2016, pp. 1–6.

[13] S. Sangeetha and V. Subramaniyaswamy, "Enhanced Travel Planning System for Group of users using Hybrid Collaborative Filtering," Indian J. Sci. Technol., vol. 9, no. 48, Dec. 2016.

[14] A. Roy, S. Banerjee, M. Sarkar, A. Darwish, M. Elhoseny, and A. E. Hassanien, "Exploring New Vista of intelligent collaborative filtering: A restaurant recommendation paradigm," J. Comput. Sci., vol. 27, pp. 168–182, Jul. 2018.

[15] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," Expert Syst. Appl., vol. 92, pp. 507–520, Feb. 2018.

[16] L. Recalde, "A Social Framework for Set Recommendation in Group Recommender Systems," Springer, Cham, 2017, pp. 735–743.

[17] C. G. Bampis, C. Rusu, H. Hajj, and A. C. Bovik, "Robust matrix factorization for collaborative filtering in recommender systems," in 2017 51st Asilomar Conference on Signals, Systems, and Computers, 2017, pp. 415–419.

[18] A. Hassan and A. Syed, "An Assessment of Collaborative Filtering-Based Recommender Systems : And their Ability to take the Individual Preferences of Users into Account," 2017.

[19] K. Bok, J. Lim, H. Yang, and J. Yoo, "Social group recommendation based on dynamic profiles and collaborative filtering," Neurocomputing, vol. 209, pp. 3–13, 2016.

[20] D. Wu, "Music Personalized Recommendation System Based on Hybrid Filtration," 2019 Int. Conf. Intell. Transp. Big Data Smart City, pp. 430–433, 2019.

[21] J. Zhang, Y. Lin, M. Lin, and J. Liu, "An effective collaborative filtering algorithm based on user preference clustering," Appl. Intell., vol. 45, no. 2, pp. 230–240, 2016.

[22] S. Ghazarian and M. A. Nematbakhsh, "Enhancing memory-based collaborative filtering for group recommender systems," Expert Syst. Appl., vol. 42, no. 7, pp. 3801–3812, May 2015.

[23] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," Inf. Retr. Boston., vol. 4, no. 2, pp. 133–151, 2001.

[24] B. N. Miller, J. A. Konstan, and J. Riedl, "PocketLens: Toward a personal recommender system," ACM Trans. Inf. Syst., vol. 22, no. 3, pp. 437–476, 2004.

[25] R. Devooght and H. Bersini, "Collaborative filtering with recurrent neural networks," arXiv Prepr. arXiv1608.07400, 2016.

[26] M. Y. H. Al-Shamri, "Power coefficient as a similarity measure for memory-based collaborative recommender systems," Expert Syst. Appl., vol. 41, no. 13, pp. 5680–5688, 2014.

[27] D. Billsus and M. J. Pazzani, "Learning Collaborative Information Filters.," in Icml, 1998, vol. 98, pp. 46–54.

[28] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," Adv. Artif. Intell., vol. 2009, 2009.

[29] Q. Zhang, J. Lu, D. Wu, G. Zhang, A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities, IEEE Trans. Neural Netw. Learn. Syst. 30 (2019) 1998–2012, http://dx.doi.org/10.1109/ TNNLS.2018.2875144.