A Machine Learning Recommender System Based on Collaborative Filtering Using Gaussian Mixture Model Clustering

Delshad Mohammad Shakoor[1], Vafa Maihami[1], Reza Maihami[2]

[1] Department of Computer Engineering, Islamic Azad University, Sanandaj Branch, Sanandaj, Iran

[2] School of Business and Leadership, Our Lady of the Lake University, San Antonio TX 78207, USA
rmaihami@ollusa.edu

Abstract

Changing and moving toward online shopping has made it necessary to customize customers' needs and provide them more selective options. The buyers search the products' features before deciding to purchase items. The recommender systems facilitate the searching task for customers via narrowing down the search space within the specific products that align the customer needs. Clustering, as a typical machine learning approach, is applied in recommender systems. As an information filtering method, a recommender system clusters user's data to indicate the required factors for more accurate predictions by calculating the similarity between members of a cluster. In this study, using the Gaussian mixture model clustering and considering the scores distance and the value of scores in the Pearson correlation coefficient, a new method is introduced for predicting scores in machine learning recommender systems. To study the proposed method's performance, a Movie Lens data set is evaluated, and the results are compared to some other recommender systems, including the Pearson correlation coefficients similarity criteria, K-means, and fuzzy C-means algorithms. The simulation results indicate that our method has less error than others by increasing the number of neighbors. The results also illustrate that when the number of users increases, the proposed method's accuracy will increase. The reason is that the Gaussian mixture clustering chooses similar users and considers the scores distance in choosing similar neighbors to the active user.

Keywords: Machine Learning; Collaborative Filtering Method; Gaussian Mixture Model Clustering; Pearson Correlation Coefficient.

1. Introduction

Personalization is an inevitable component of electronic commerce. This term means that the provider filters each particular individual's information to provide customers a customized or personalized interaction with the goods, website, services, or company employees. Personal

contact and in-person consultation are approved strategies in customer relationship management. However, this can not happen in online shops since there is no communication between the vendor and the buyer. Therefore, the personalization concept became a crucial requirement for online shopping.

Several mechanisms have been developed for personalization. Among them, many researchers have paid attention to recommender systems. According to the user's available information, these systems recommend similar or potentially related interesting items for a given customer. The enormous volume and variety of digital information make it harder to find the required information. For decades, recommender systems (RS) have been known as a leading approach to dealing with information overflow issues. These systems help to extract information, filter and predict relevant information for users [1, 2]. One of the most robust RS design methods is the Collaborative Filtering Recommender System (CFRS), which analyzes user preferential information for predicting proposals based on their similarity to other users. CFRSs are used in various fields, from commercial to financial services, to recommend users' items of interest [3-5]. Important commercial examples and social media include Twitter, Facebook, Amazon, and eBay. RS has recently been employed in some innovative areas. For example, RS is utilized in trust-based relationships between users in social networks to understand user interests better and improve recommendations [6].

Collaborative filtering (CF) is used to offer personalized information in the majority of recommender systems. CF starts with the rating matrix (an i-by-j-matrix) where i is the customers (rows), and j implies the items (columns). The similarities between the customers (customer-based method) or between the items (product-based method) are computed through different techniques. The required information to complete the rating matrix is collected either implicitly or explicitly. The customer enters the explicit information to the system directly, such as the user's product rating. In contrast, the customer's interaction with the store will indicate implicit information such as the clickstream analysis information and customer's orders.

The objective of CF is making filtering decisions for an individual user based on the judgments of other users. CF selects a subset of the users (neighbors) as predictors, normalizes ratings, and computes a prediction from a weighted combination of the selected neighbors' ratings. Finally, items with the highest predicted ratings are suggested as recommendations.

The CF technique does not need the product's semantic information or manually linking customer and products. Therefore, personalization through the CF is very convenient and

efficient. CF works with interaction between the store and the customer, which is the only required information.

Despite its merits, the CF method has two weak points: 1) Search for similar users in a large data set is challenging. 2) CF works weakly in recognizing the interest of users with little available information.

To address these issues, a Gaussian mixture model (GMM) clustering is developed in this research to improve performance and higher accuracy in user clustering. GMM clusters the users according to user similarity and social trust-based relationships. Estimation of the parameters of GMM is a crucial component of this approach. Heretofore, different methods have been presented to estimate the parameters of the GMM. One of the most popular methods in recent years is the Expectation Maximization (EM) algorithm that speeds up the improvement of clustering parameters [7].

In this study, we introduce a new recommender system that works based on CF and clustering principles. The recommender system will use neighbor users' information to the active user to provide a better recommendation. It is essential to calculate the similarity between all items in the system for finding the nearest neighbors. We propose a new and practical technique according to the GMM clustering to reduce the search space by clustering the users based on individuals' personal information. Our objective is to find the nearest neighbors quickly with high accuracy that empowers the recommender system to offer a better prediction.

The rest of this paper is organized as follows: the literature review is discussed in Section 2. Section 3 defines the problem and introduces the basic concepts of the study. The proposed method of CF based on GMM is presented in Section 4. Section 5 analyzes a simulation case and discusses the results. Finally, the conclusions and future works are shown in Section 6.

2. literature review

Many solutions have been provided for making recommendations in recommender systems [8-10]. Nevertheless, two main approaches play a significant part in developing recommender systems: Content-Based Filtering (CBF) and CF. In the CBF method, items similar to the users' prior choices are recommended. While in CF based recommender systems, items are recommended to the active user according to the prior choices which other users (neighboring users) have made. CF-based methods can make recommendations using only users' priorities regarding only a group of items, and because of their simplicity, they have grown quite popular recently. Hybrid recommendation systems have also been developed to combine the

main methods' strength and increase potency. In the following, we review the notable papers related to our work.

Singh and Solanki [11] presented different evaluation criteria to analyze the recommender systems performance using clustering. The research included clustering algorithms such as fuzzy C-means, K-means, and various nature-inspired algorithms to find the most similar items and users in each cluster. The results showed that fuzzy C-means clustering effectively improves the dispersion issues and the scalability in collaborative filtering systems. Katarya and Verma [12] presented a recommender system based on collaborative filtering that uses nature-inspired grey wolf optimization algorithm and fuzzy C-means clustering technique to predict a movie's rating for a particular user based on his/her historical data and similarity of users. The grey wolf optimization algorithm has applied to a dataset to obtain the first clusters, and also the early positions of the clusters were obtained. FCM has used to categorize users in the dataset with the similarity of user rating. The collaborative recommender system had an outstanding performance according to its accuracy and precision.

A recommender system based on collaborative filtering has developed in Katarya and Verma [13] that uses the K-NN and K-means algorithms. The proposed combinatorial model employs a typical division method and categorizes products according to users. K-means provides initial parameters for Particle Swarm Optimization (PSO) to improve its performance. Then, PSO provides the initial seed and, instead of the precise clustering in the K-means, improves the data items (users) of the fuzzy C-means (FCM) for soft clustering. The model first applies a typical division method to reduce the multi-dimensional accumulated data space.

Zahra, Ghazanfar [14] introduced a K-means clustering based recommender algorithm that examines the scalability issues related with traditional recommender systems. The problem of the traditional K-means clustering algorithms is that they randomly choose the k primary centroid, which leads to incorrect recommendations and increase the cost of clusters offline training. This study shows how the choice of centroid in K-means based recommender systems can improve performance and cost savings. The centroid selection method can utilize the base data's correlation structures, which have higher accuracy and performance than the traditional centroid selection strategies, which randomly select the centroids. The approach is confirmed and proven by an extensive collection of experiments based on five different data sets (from the film, book, and music). These experiments' results showed that the proposed

approach provides a higher quality cluster and faster convergence than other approaches, which in turn improves the accuracy of the proposed offer. Honda, Sugiura [15] Provided a new collaborative filtering approach using local main components. The method is based on a synchronic principal component analysis and fuzzy clustering with an incomplete data set, including non-existing data. The local main components are extracted using the low-rank approximation in the data matrix in the synchronic approach. Non-existing data are predicted using the data matrix approximation.

Gohari, Haghighi [16] analyzed recommendation framework using ant colony optimization to classify the user's neighbors. The method used the semantic data of the object to determine the semantic similarity between the objects. Alhijawi and Kilani [17] investigated a new recommendation method based on a genetic algorithm. The genetic algorithm considers a semantic score for each user based on semantic similarity between the two data relating to the user. Then, the genetic algorithm indicates the degree of active user satisfaction for an individual user. In recent years, deep learning has been used in recommender system studies and has shown reliable results [18, 19]. Zhang, Yao [18] explored the application of deep learning in various aspects of recommender systems.

Clustering algorithms have been used as a similarity criterion in Dhanalakshmi, Anitha [20]. The clustering algorithms find the users in the neighboring of the active user. Three clustering algorithms, namely, K-means, Fuzzy C-means, self-organizing maps, have been compared. Lika, Kolomvatsos [21] addressed the problem of predicting the scores and the cold start in the recommender systems using the clustering algorithm. The clustering algorithm considers the number and distance of the scores in calculating the similarity between individuals, but do not consider the value of the scores and has a low velocity. A few papers have used the GMMs as a probabilistic model in data clustering [22-24]. The advantages of clustering algorithms are considering the number of scores and considering the distance points. Also, the disadvantage of clustering algorithms is low speed disregarding the value of scores. In this paper, we present a new CF recommender system based on the GMM clustering method. Our model reduces the search space by clustering the users and speed up the searching process and filtering. This study contributes to the existing literature in several directions. First, a new machine learning model is developed for CFRS that use GMM to cluster the users and improve the RS's performance. Second, a comparison between this study with current models has been made through multiple performance metrics that provide a

framework for deciding on the RSs. Third, we show the applicability of this model by performing the proposed model on a large dataset.

3. Problem Definition

In this section, we define the problem of the study. We begin with the definition of RS and continue with the concepts of GMM, EM, and Pearson correlation.

Recommending and personalization are essential approaches to combating information overload. RSs are presented to offer the most suitable items to users. One of the most critical issues in the recommender systems is making accurate predictions. CFRSs compute predictions through the history of active user's neighbors, and the correct choice of nearest neighbors who have the most similarity with the active user will significantly increase the accuracy of predictions. To find the nearest neighbors, the similarity between all the items in the system must be calculated. Clustering algorithms can reduce the search space by clustering users; therefore, the system's performance will be increased.

As a result, predictions will be generated by calculating the similarity between members of a cluster. On the other hand, since a user may belong to several clusters, the Gaussian mixture clustering algorithm can better perform clustering. An important feature of this model is its flexibility to continuous distributions in a variety of forms. Since the essential part of the fitting of this model is the estimation of its parameters, the minimum value of the component is considered for a mixed distribution, and then, by adding a new component to it in a few steps, the model is suitable for describing the data and, therefore, the optimal number of components for primary mixed distribution is determined. The EM algorithm is applied to estimate the parameters by using initial estimations and considering the repeated cycle's hidden variables. This algorithm starts with the initial considering value for the model parameters, and in the next step, which is called the repetition phase, these parameters are updated, and the cycle repeats until the algorithm converges. The repetition phase consists of two steps; calculation of expected value and maximization. In the below sections, we will go through the main steps of the proposed model.

3.1 Gaussian Mixture Model

The Gaussian distribution known as normal distribution is one of the most widely used models used to express continuous variables' distributions. In the situation of one-dimensional variable x, the Gaussian distribution of the variable can be represented as follows.

$$N(x|\mu,\sigma^2)=\frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}}\exp\left\{\frac{-1}{2\sigma^2}(x-\mu)^2\right\} \quad (1)$$

In the Equation (1), μ represents the mean and σ^2 is variance.

The zoning consists of K zone ($K\geq 2$) of $P_1,P_2,\ldots,P_k$. The goal is to examine the features of variable x_i in order to belong to one of these regions. If $N_1,N_2,\ldots,N_k$ be the normal distribution function of the random variable x_i in the regions $P_1,P_2,\ldots,P_k$ then the distribution function of the random variable x_i in region P presented as follows:

$$N(x_i)=\pi_1 N_1(x_i)+\pi_2 N_2(x_i)+\ldots+\pi_k N_k(x_i) \quad (2)$$

π_i is the probability of belonging x_i to the region P_i, and it has a normal distribution N_i .

For $i=1,2,\ldots,k$ and $0<\pi_i<1$, $\sum_{i=1}^{k}\pi_i=1$. The $\pi_i$values are called mixing ratio (Equation 3).

It is a finite mixed distribution with k components in terms of the normal distribution function. If $N_1,N_2,\ldots,N_k$ be the density functions of the random variable x_i in the regions $P_1,P_2,\ldots,P_k$ respectively, then the mixed model in terms of density functions is:

$$f(x)=\pi_1 N_1(x)+\pi_2 N_2(x)+\ldots+\pi_k N_k(x) \quad (3)$$

$$f(x)=\sum_{i=1}^{k}\pi_i N_i(x) \quad (4)$$

The normal density functions N_i for $i=1,2,\ldots,k$ can include the vector of parameter $\theta_i=(\theta_{i1},\theta_{i2},\ldots,\theta_{ih})^T$. Therefore, a parametric finite mixed density with k component is presented as follows:

$$f(x\vee\theta)=\sum_{i=1}^{K}\pi_i N(x|\theta_i) \quad (5)$$

In the Gaussian mixture model, $f(x\vee\theta)$ with components of normal T including mean and variance of Equation (5), which presented as Equation (6).

$$p(x)=\sum_{i=1}^{K}\pi_i N\text{¿} \quad (6)$$

In the above Equation, k represents the number of classes, π_i, the probability of initial occurrence of each class, N, Gaussian density function (normal distribution) of each class, μ_i, the mean vector, and σ_k^2 is the variance of each class [25].

3.2 Expectation Maximization Algorithm

In this method, the lowest value of the component is considered for a Gaussian mixture distribution. Next, a new component adds to the distribution in a few steps, which leads to an appropriate model for describing the data. Repeating this process will provide the optimal number of components for the initial mixed distribution. The EM algorithm is applied to estimate the parameters, using the initial estimates and considering the hidden variables of the repeated cycle. This algorithm starts with considering the initial value for the model parameters, and in the next step, which is called the repetition phase, these parameters are updated, and the cycle repeats until the algorithm converges. The repetition phase consists of two steps: calculation of expected value and maximization.

The summary of the implementation process of the EM algorithm can be expressed as follows:

- **Step 1.** Determine the K (number of clusters).
- **Step 2.** Generate the initial values of Gaussian mixture model parameters. These parameters include mixing coefficient, mean vector, and standard deviation for each mixed component.
- **Step 3.** Calculate the expected value. The probability of each observation is required to determine the expected value. The mixed density function will provide the probability value for each observation.
- **Step 4.** Maximize the expected value. This step updates the model's parameters according to the fitness function.
- **Step 5.** Repeat steps 3 and 4 to achieve convergence in estimating model parameters.

EM algorithm usually converges to a local optimum point because choosing different initial values for θ led to different results. To avoid getting stuck at the local optimum point, we perform the algorithm several times from different starting points and keep the best result that is time-consuming [26].

3.3 Correction of Pearson Similarity Criterion

One of the most critical parts of the memory-centric CF algorithm is to determine the similarity between users. The correct choice of a similarity function is a critical factor for determining the similarity between users since it significantly affects the recommendations' accuracy. A common criterion for obtaining similarities is the Pearson correlation coefficient. Studies have proven that the Pearson correlation coefficient has better performance than other similarity criteria [11]. This coefficient specifies a linear relationship between two distinct variables, and its value varies from -1 to +1. The value of +1 denotes the complete relationship between the two variables, and the value of -1 denotes no (lack of) relationship between the two variables. In other words, +1 shows that two users have completely related interests, while the number of -1 shows a conflict of interests between two users. The Pearson correlation coefficient has widely used as a similarity criterion in the recommender systems, which has some disadvantages that are given below.

1. Not considering the number of items in the calculation of similarity
2. Not considering the distance scores in the calculation of similarity
3. Not considering the value of items in the calculation of similarity

While the number of items impacts the similarity score, for example, assume that 2 users have the same views in the 5 common items, that according to the Pearson method, the similarity value of these 2 users will be +1. Also, assume that 2 users in the 100 items have the same views; in this case, the similarity value of these 2 users will be +1 too. In other words, the Pearson correlation coefficient does not consider the number of items as well as the number of common items. To this end, we use Equation (7) as a coefficient to consider the number of common items.

$$\frac{|O_{a,b}|}{|I|} \tag{7}$$

Equation (7) is the ratio of the number of common items to the total number of items, $|O_{a,b}|$ is the number of common items and $|I|$ is the total number of items. The other disadvantage of Pearson's method is not taking into account the distance of scores in calculating similarity. In other words, the Pearson correlation coefficient has only focused on the increasing numbers. We use the coefficient (8) to modify the correlation between two users to solve this problem. Therefore, the similarity between *a* and *b* is determined by equation (9).

$$\frac{1}{1+\sqrt{\sum_{i=1}^{m}\left(r_{a,i}-r_{b,i}\right)^2}} \tag{8}$$

$$¿(a,b)=\left(\frac{1}{1+\sqrt{\sum_{i=1}^{m}\left(r_{a,i}-r_{b,i}\right)^2}}\right)\frac{\sum_{p\in P}\left(r_{a,p}-\bar{r}_a\right)\left(r_{b,p}-\bar{r}_b\right)}{\sqrt{\sum_{p\in P}\left(r_{a,p}-\bar{r}_a\right)^2}\sqrt{\sum_{p\in P}\left(r_{b,p}-\bar{r}_b\right)^2}} \tag{9}$$

4. The CF system using GMM

Figure 1 shows the steps for the proposed method.

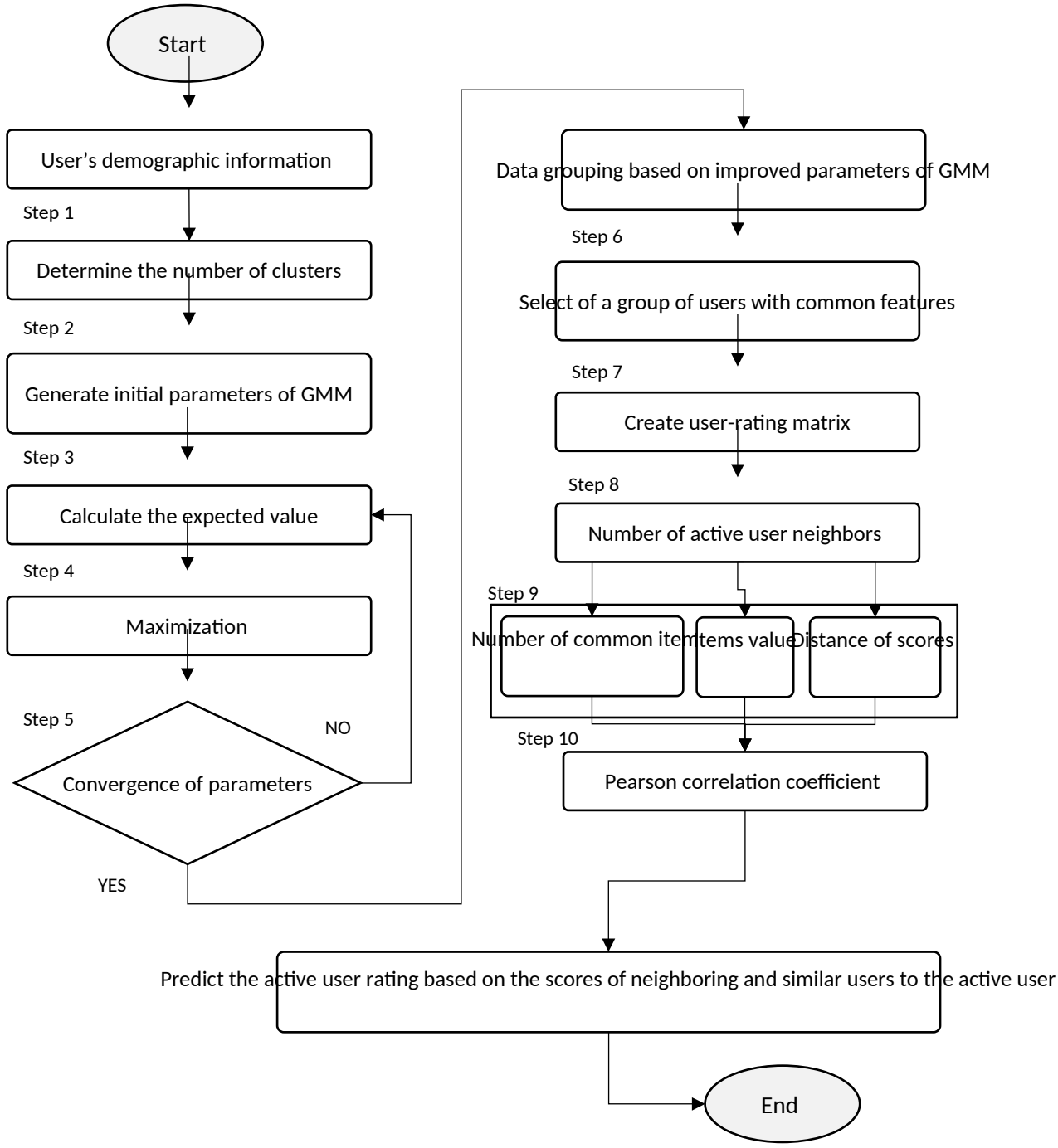

Figure (1): The proposed framework of CF system using GMM

Steps 1 through 5 are described in Section 3.2. Here we explain the rest of the algorithm. Steps before Step 6 are designed to find the data clusters according to GMM. These steps can reduce the search space by clustering users, thereby increases the scalability of the system. Thus, predictions will be generated by calculating the similarity between members in a cluster.

On the other hand, as a user may belong to several clusters, the clustering algorithm of the GMM can better perform clustering. An essential feature of this model is its flexibility to continuous distributions in a variety of forms. Step 6 selects a group of users with common features. The features came from the implicit and explicit data that users interacted with the system. Step 7 creates a user-rating matrix. As we discussed earlier, this matrix ranked the priority of customers regarding different products. Then, Step 8 finds the number of neighbors of the active user. Step 9 computes three measures; the number of items, items' value, and distance of scores between the active user and its neighbors. The information of Step 9 helps to indicate the Pearson correlation coefficient in Step 10. Finally, the algorithm predicts the active user rating based on the score of neighboring and similar users to the active user based on Equation (10).

$$pred(a,p)=\overline{r_a}+\frac{\sum_{b\in N} ¿(a,b)*(r_{b,p}-\overline{r_b})}{\sum_{b\in N} ¿(a,b)} \quad (10)$$

5. Simulation Results

In this section, we analyze the performance of the model using a MovieLen100K dataset. This dataset has been studied in multiple papers. We run our model using the MATLAB package on a system with 8 GB of memory and a 5-core processor 2.7 GHz was. Here, we first introduce the data set and evaluation criteria. Then, the model's result has been discussed.

5.1 Data Set

The MovieLens100K dataset is introduced in Harper and Konstan [27]. Table 1 shows the information items for the dataset.

Table (1): The data set used to evaluate the performance of the proposed method

Data set	Number of user	Number of Items	Type of Items	Number of Scores	Scores	Scale of Scores	Type of Scores
MovieLens100K	943	1682	Movie	100.000	Ordinal	+1 to +5	True

In the MovieLens100K dataset, the Scores are from +1 to +5. +1 score means no interest and the +5 score means the most interest in the movie. Users also rated at least 20 Movies. To

evaluate the proposed method's performance, the data are divided into two sets of training data and testing data. The training set consists of 20% of the data, and the testing set consists of 5% of the data.

5.2 Evaluation Criteria

We need to introduce some evaluation criteria to study the performance of the prosed .model

- **Mean Absolute Error:** The most common criterion for comparison is the mean absolute error used to evaluate a system's ability to predict a user interest in a particular item. This criterion gains the mean absolute value of the difference between the actual value and the predicted value as Equation (11).

$$MAE = \frac{\sum_{i=1}^{n} |r_{a,i} - r_{p,i}|}{n} \tag{11}$$

Where $r_{a,i}$ is the real score of user a to item i. $r_{p,i}$ is the predicted score of user a to item i and N is the number of predictions. The MAE criterion has been widely used in the evaluation of recommender systems.

- **Root Mean Square Error:** Root mean square error is also a common criterion in evaluating recommender systems as defined in Equation (12).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} |r_{a,i} - r_{p,i}|}{n}} \tag{12}$$

Where $r_{a,i}$ is the real score of user a to item i. $r_{p,i}$ is the predicted score of user a to item i and N is the number of predictions.

5.3 Evaluation Result of Proposed Method

Figure 2 shows the mean absolute error values for the MovieLens100K data set. Our model has the lowest error value for all neighboring values compared to other Fuzzy C- Means and K-means methods. In Fig. 3, the root mean squared error is also shown. Compared to the Pearson method, this method has the lowest error rate for the number of neighbors 30 and 70, and for the number of neighbors 10 and 20 has comparable results with the Pearson method.

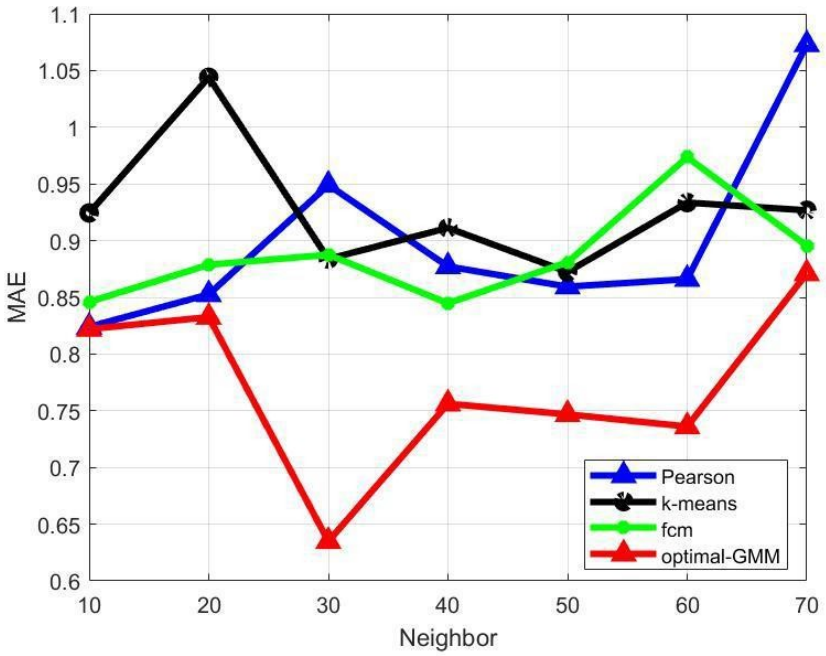

Figure (2): Mean Absolute Error for MovieLens100k Dataset

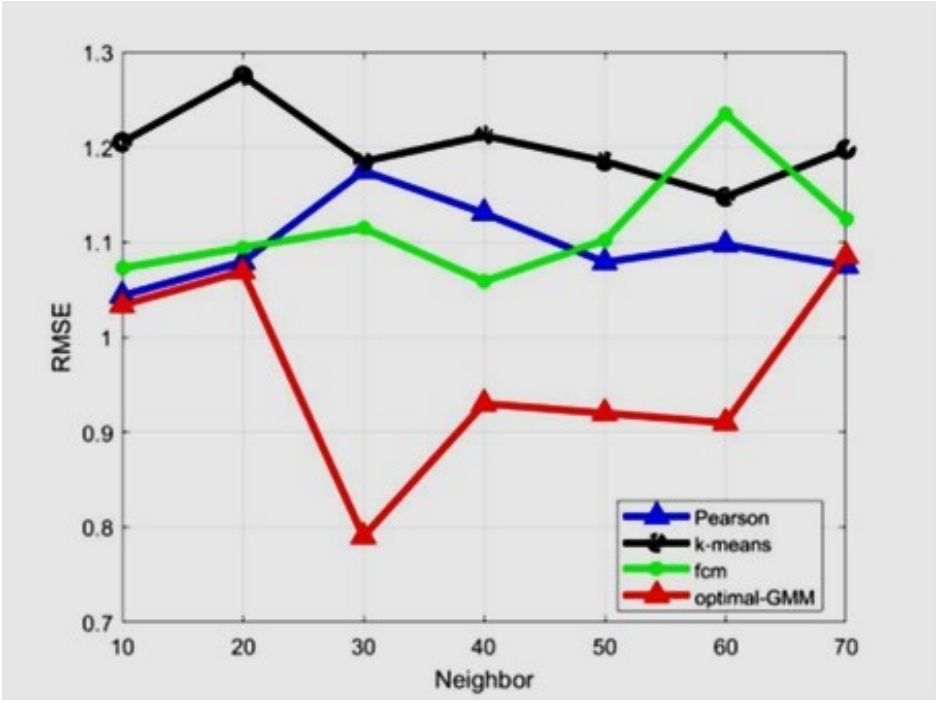

Figure (3): Root Mean Square Error for MovieLens100k Dataset

As shown in Fig. 3, the proposed method has a better performance and result than the other methods in the MovieLens100k dataset. For example, the mean absolute error of the dataset in the neighbor size of 70 in the Pearson method is 1.07, in the K-mean is 0.92, and in the Fuzzy C-Means is 0.89, while in the proposed method, the error rate is equal to 0.87, which means the error value is reduced. The reason for this error reduction in the proposed method

can be deduced from considering the number, value, and distance of the scores in the Pearson correlation coefficient and using the demographic information clustering of users in the Gaussian model, which led to optimally calculation of the individual's similarity value.

In Table 2, the MAE values of the proposed method and other methods are shown in the MovieLens100k dataset. The best result values among the methods are shown with bold fonts. Obviously, the proposed method in the small size of the neighbors has a higher percentage of correct predictions than the other ones, and the correct percentages of predictions in all neighbors are higher than all other methods.

Table (2): Comparison of MAE values in the MovieLens100k dataset

Neighbors	Pearson	K-means	Fuzzy C-Means	GMM
10	0.82402	0.92442	0.84594	**0.82200**
20	0.85261	1.04400	0.87872	**0.83261**
30	0.94914	0.88384	0.88762	**0.63489**
40	0.87722	0.91124	0.84477	**0.75616**
50	0.85944	0.87247	0.88098	**0.74685**
60	0.86609	0.93357	0.97357	**0.73609**
70	1.07300	0.92675	0.89552	**0.87107**

In the first column of the table (2), the size of active user neighbors is shown, starting from 10 up to 70. The similarity measures of the Pearson correlation coefficient, K-mean, and Fuzzy C-Means and the proposed method (GMM) are shown, respectively. For example, in the neighbor size of 30, the proposed method predicts accurately more than 5% higher than other methods. One of the critical components in the recommender systems is finding an active user neighboring section that, if properly selected, can significantly increase the accuracy of the recommendations. The reason for the proposed method's superiority is more accurately finding the active user's neighbors. However, compared to the Pearson method for the number of neighbors 30 has the lowest error rate, and for the number of neighbors 10 and 20, the results are equal with the Pearson method.

In Table 3, the RMSE values of the proposed method and other methods are shown in the MovieLens100k dataset. The best result values among the methods are shown with bold fonts. It is evident that the proposed method in the small size of the neighbors has a higher percentage of correct predictions than the other methods.

Table (3): Comparison of RMSE values in the MovieLens100k dataset

Neighbors	Pearson	K-means	Fuzzy C-Means	GMM
10	1.0438	1.2052	1.0728	**0.03380**
20	1.0793	1.2752	1.0944	**0.06930**
30	1.1749	1.1849	1.1150	**0.79031**
40	1.1306	1.2121	1.0588	**0.93010**
50	1.0789	1.1852	1.1021	**0.92000**
60	1.0976	1.1477	1.2353	**0.91000**
70	1.0751	1.1978	1.1243	**1.08510**

In the first column of the table (3), the size of active user neighbors is shown, starting from 10 up to 70. The similarity measures of the Pearson correlation coefficient, K-mean, and Fuzzy C-Means and the proposed method (GMM) are shown, respectively. For example, in the neighbor size of the 30, the proposed method predicts accurately more than 5% higher than other methods. One of the key components in the recommender systems is finding an active user neighboring section that, if properly selected, can significantly increase the accuracy of the recommendations. The reason for the proposed method's superiority is more accurately finding the active user's neighbors. However, compared to the Pearson method for the number of neighbors 30 has the lowest error rate, and for the number of neighbors 10 and 20, the results are equal with the Pearson method.

6. Conclusions and future works

The RSs are intelligent systems that refine existing information on the Internet by identifying each user's interests and priorities and providing users with appropriate and relevant suggestions. The most commonly used algorithm in the RSs is a CF algorithm with relatively better results than other RSs. CF's main idea is that if two users have the same scoring on common items, they have the same interests. Therefore, in this way, the recommendations are given to the active user based on the neighbors. One of the most critical parts of the RSs is finding neighbors, if properly selected, significantly increasing the recommendations' accuracy.

A concrete way to find neighbors is the use of similarity measurement criteria. Measuring the similarity of common item scores is used to calculate the similarity between the active user and other users. Several similarity measurements have been reported in previous works. The RSs use CF to predict the active user rank to product or service. This method depends on the active user's similarity to its neighbors; the similarity level can be increased by user clustering. Various clustering algorithms such as k-means and fuzzy C-means algorithms

exist for user clustering, but the k-means uses hard clustering methods to grouping users, and a user can only belong to a cluster. In the fuzzy C-means method, despite using soft clustering, a user can belong to several clusters; however, users' statistical distribution does not consider in clustering.

In this study, the clustering method of the GMM is used to increase the accuracy of clustering. The probability percentage of a user belonging to different clusters is evaluated by taking into account the statistical distribution of users and the conditional probability (Bayes). However, the Gaussian mixture model depends on the initial parameters such as the mean and variance of society and, based on these parameters' different values, presents different results from clustering. In order to determine the optimal initial parameters of the Gaussian mixture model in this study, the expected maximization (EM) method is proposed. On the other hand, the Pearson correlation coefficient is widely used for similarity measurements. The Pearson correlation coefficient has disadvantages such as not considering the number of common items, the distance of scores, and the value of scores. In the proposed method, these disadvantages are eliminated, and a new criterion is introduced using the Pearson correlation coefficient. To evaluate the performance of our proposed method, the MovieLens100k dataset was used. In all simulation results, the proposed method has the least error. The experiments performed on the MovieLens dataset showed that the proposed model might offer high performance in terms of precision and more predictability and personalized predictions. Compared with the existing methods with a Mean Absolute Error of 0.78 (MAE), our result is 3.503% better with 0.75 mean absolute error - it was shown that the proposed approach shows better results.

This study can be extended in several directions. There are various correlation coefficients in statistics that can be used as a criterion of similarity or combine in collaborative refinement systems. Clustering users with similar interests and combining with fuzzy methods can also reduce the recommender system's error. Using user information and items can effectively impact the accuracy of the recommendations. It is possible to increase the accuracy of the recommendations by extracting the experts and using their comments.

References

1. Ricci, F., L. Rokach, and B. Shapira, *Introduction to recommender systems handbook*, in *Recommender systems handbook*. 2011, Springer. p. 1-35.
2. Adomavicius, G. and Y. Kwon, *New recommendation techniques for multicriteria rating systems.* IEEE Intelligent Systems, 2007. **22**(3): p. 48-55.
3. Wang, L., et al., *Intelligent fashion recommender system: Fuzzy logic in personalized garment design.* IEEE Transactions on Human-Machine Systems, 2014. **45**(1): p. 95-109.

4. Hu, Y., et al., *Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering.* ieee transactions on services computing, 2014. **8**(5): p. 782-794.
5. Herlocker, J.L., J.A. Konstan, and J. Riedl. *Explaining collaborative filtering recommendations*. in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. 2000.
6. Asabere, N.Y., et al., *Improving smart conference participation through socially aware recommendation.* IEEE Transactions on Human-Machine Systems, 2014. **44**(5): p. 689-700.
7. Domingues, M.A., et al. *Using contextual information from topic hierarchies to improve context-aware recommender systems*. in *2014 22nd International Conference on Pattern Recognition*. 2014. IEEE.
8. Palomares, I., F. Browne, and P. Davis, *Multi-view fuzzy information fusion in collaborative filtering recommender systems: Application to the urban resilience domain.* Data & Knowledge Engineering, 2018. **113**: p. 64-80.
9. Nilashi, M., O. Ibrahim, and K. Bagherifard, *A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques.* Expert Systems with Applications, 2018. **92**: p. 507-520.
10. Maihami, V., D. Zandi, and K. Naderi, *Proposing a novel method for improving the performance of collaborative filtering systems regarding the priority of similar users.* Physica A: Statistical Mechanics and its Applications, 2019. **536**: p. 121021.
11. Singh, S.P. and S. Solanki. *Recommender System Survey: Clustering to Nature Inspired Algorithm*. in *Proceedings of 2nd International Conference on Communication, Computing and Networking*. 2019. Springer.
12. Katarya, R. and O.P. Verma, *Recommender system with grey wolf optimizer and FCM.* Neural Computing and Applications, 2018. **30**(5): p. 1679-1687.
13. Katarya, R. and O.P. Verma, *A collaborative recommender system enhanced with particle swarm optimization technique.* Multimedia Tools and Applications, 2016. **75**(15): p. 9225-9239.
14. Zahra, S., et al., *Novel centroid selection approaches for KMeans-clustering based recommender systems.* Information sciences, 2015. **320**: p. 156-189.
15. Honda, K., et al. *Collaborative filtering using principal component analysis and fuzzy clustering*. in *Asia-Pacific Conference on Web Intelligence*. 2001. Springer.
16. Gohari, F.S., H. Haghighi, and F.S. Aliee, *A semantic-enhanced trust based recommender system using ant colony optimization.* Applied Intelligence, 2017. **46**(2): p. 328-364.
17. Alhijawi, B. and Y. Kilani, *A collaborative filtering recommender system using genetic algorithm.* Information Processing & Management, 2020. **57**(6): p. 102310.
18. Zhang, S., et al., *Deep learning based recommender system: A survey and new perspectives.* ACM Computing Surveys (CSUR), 2019. **52**(1): p. 1-38.
19. Kiran, R., P. Kumar, and B. Bhasker, *DNNRec: A novel deep learning based hybrid recommender system.* Expert Systems with Applications, 2020. **144**: p. 113054.
20. Dhanalakshmi, K., et al., *Recommendation system based on clustering and collaborative filtering.* International Journal of Innovative Research in Computer and Communication Engineering, 2015: p. 2482-2488.
21. Lika, B., K. Kolomvatsos, and S. Hadjiefthymiades, *Facing the cold start problem in recommender systems.* Expert Systems with Applications, 2014. **41**(4): p. 2065-2073.
22. Yan, H. and Y. Tang, *Collaborative filtering based on gaussian mixture model and improved Jaccard similarity.* IEEE Access, 2019. **7**: p. 118690-118701.
23. Xu, L., X. Li, and Y. Guo, *Gauss-core extension dependent prediction algorithm for collaborative filtering recommendation.* Cluster Computing, 2019. **22**(5): p. 11501-11511.
24. Van, D.N., V.T. Pham, and T.M. Thanh. *Robust Content-Based Recommendation Distribution System with Gaussian Mixture Model*. in *International Conference on Computational Collective Intelligence*. 2020. Springer.
25. Gorgoglione, M. and U. Panniello. *Including context in a transactional recommender system using a pre-filtering approach: two real e-commerce applications*. in *2009 International Conference on Advanced Information Networking and Applications Workshops*. 2009. IEEE.

26. Panniello, U., A. Tuzhilin, and M. Gorgoglione, *Comparing context-aware recommender systems in terms of accuracy and diversity.* User Modeling and User-Adapted Interaction, 2014. **24**(1-2): p. 35-65.
27. Harper, F.M. and J.A. Konstan, *The movielens datasets: History and context.* Acm transactions on interactive intelligent systems (tiis), 2015. **5**(4): p. 1-19.