Bilad Alrafidain Journal for Engineering Science and Technology
https://dx.doi.org/10.56990/bajest/2024.030104

ISSN: 2957-9651
Pages: 47-58

Increasing the Effectiveness of Prediction in Recommendation Engines Based on Collaborative Filtering

Roaa Faleh Mahdi[*]
Directorate of Education
Email: Roaafalih78@gmail.com

Abstract

In the era of information abundance, the demand for personalized content recommendations has become paramount. Recommendation engines, particularly those employing collaborative filtering, play a pivotal role in delivering tailored suggestions based on user preferences. As technology evolves, the need to enhance the effectiveness of prediction algorithms within these engines becomes increasingly crucial. This research endeavors to contribute to this evolving landscape by delving into collaborative filtering methodologies, identifying challenges, and proposing novel strategies to elevate the accuracy and relevance of predictions in recommendation systems. Through this exploration, we aim to not only refine existing models but also pave the way for more sophisticated and reliable personalized content recommendations.

This research aims to enhance prediction accuracy in recommendation engines utilizing collaborative filtering. Through an in-depth exploration of collaborative filtering techniques, we propose innovative approaches to improve the effectiveness of predictions. Our study addresses key challenges in collaborative filtering models, offering insights into refined algorithms and methodologies. By fine-tuning the collaborative filtering process, we anticipate a substantial boost in the overall performance of recommendation engines, ultimately advancing the field of personalized content suggestion. The simulation is performed using Java language and using two datasets Movie Lens 1M and Movie Lens 100K.The proposed model was evaluated using the Mean Absolute Error, Precision, and Recall.

The proposed model achieved a mean absolute error value ranging between 0.78 and 0.84 using the Movie Lens 100K dataset, and a mean absolute error value ranging between 0.72 and 0.74 using the Movie Lens 1M dataset for different values of the number of user groups. As for precision and recall, the precision of the proposed model ranged between 0.97 and 0.985 using the Movie Lens 100K data set, and a precision value ranging between 0.944 and 0.954 using the Movie Lens 1M data set, also for different values of the number of user groups.

As for the recall results, the proposed model achieved a recall value ranging between 0.755 and 0.85 using the Movie Lens 100K dataset, and a recall value ranging between 0.72 and 0.75 using the Movie Lens 100K dataset, also for different values of the number of user groups. These results were compared with the PMF, HPF, and NMF algorithms, where the proposed model proved its clear superiority over these algorithms. Using this analysis of the matrix allows us to obtain a good prediction accuracy of users' preferences and to find common groups of people with similar preferences.

Keywords: recommendation systems, collaborative filtering, accuracy, recall, probability distribution

Article history: Received*:* 17/12/2023 Accepted: 30/1/2024

1. Introduction

Recommendation systems are systems capable of providing personalized recommendations to users [1]. There are many fields in which recommendation systems are used, such as television, e-commerce, books, music, or e-learning [2]. There are two main types of recommendation systems that we can distinguish between them depending on the systems' inputs. The first is content-based recommendation systems which describe items through features and verbal depiction required for this type [3], where information and details about items preferred by users can be obtained by observing and monitoring the items consumed by them, as this information is considered a very important requirement for this type of system. Whereas the second is based on collaborative filtering, this type relies on the ratings made by users to discover the taste of users, these systems use an M rating matrix where the information

[*] Corresponding author: Roaafalih78@gmail.com

provided by each user shows how much they like specific items, and is not based on information about the users features or items.

The basic classifications of recommendation systems based on collaborative filtering are as follows. 1) Memory-based recommendation systems: it uses the K-Nearest Neighbor (KNN) Algorithm to predict the users' preferences. The basic idea of this algorithm is that to suggest items of interest to the user, the algorithm is going to find users with similar preferences for a feature of an item and names the users as neighbors of a user, Then the system suggests this element to user dependence on high evaluation of this element by the neighbors. In other words, in the case where the recommendation system expects that the item will be liked by the user, there will be some users who rate the item very positively, whose have the same preferences as user. In this algorithm, we use the similarity function to measure the similarity of users based on their preferences makes predictions more accurate. Not using a scalable and flexible algorithm to predict and recommend items One of the main problems of this kind is that they are not suitable when the number of items and users is too large because the complexity of the algorithm becomes very large, which causes delays in the execution time of the algorithm as well as a decrease in in prediction accuracy. 2) Model-based recommendation systems: It predicts the ratings given by users by relying on a prediction model. Since these recommendation systems include flexible, scalable algorithms, it is recommended to use them if the number of users or items is too large.

This kind requires a learning step to discover user and element matrices. This matrix is called the Classical Matrix. This kind has used for predicting users' ratings because it is very quickly if the learning step is completed. Upon completion of learning step, each user and each item are connected to each other through tow vectors. It uses these two vectors for predicting the rating that user will give to item. The main disadvantage of this kind is that they are not flexible where it gives false predictions if the item is new, so we will focus in this paper on model-based recommendation systems because these recommendation systems include flexible algorithms and it is suitable for using if the number of users and item is too large.

Recommender systems are one of the recent inventions to deal with information overload problem and provide users with personalized recommendations that may be of their interests [4]. Collaborative filtering is the most popular and widely used technique to build recommender systems and has been successfully employed in many applications [5]. However, collaborative filtering suffers from several inherent issues that affect the recommendation accuracy such as: data sparsity and cold start problems caused by the lack of user ratings, so the recommendation results are often unsatisfactory [4]. To address these problems, we propose a recommendation methodology that enhances the recommendation accuracy of collaborative filtering method by leveraging different user situations in these networks to model user preferences.

2. Related works

Previous studies by various researchers have extensively explored collaborative filtering techniques for recommendation engines. Early works demonstrated the effectiveness of these approaches, while recent research has delved into areas like matrix factorization and hybrid recommendation systems. Our study builds on these foundations, aiming to contribute innovative strategies to enhance prediction accuracy in collaborative filtering-based recommendation engines.

In [3], a technique was proposed dependent on decomposing the matrix into two matrices. One of these matrices is associated with words and the other matrix is associated with describing items as in [6, 7]. Since the number of words that must be considered is very large. This leads to memory consumption and a delay in execution time for operations based on the previous two arrays.

In [6], a probability model dependent on the Gaussian distribution is introduced for solving the optimization problem, where vectors (α_u and b_i) and classifications $r_{un,\,i}$ are considered to follow Gaussian distribution, however it is still difficult to understand the components of each of the two mentioned vectors since their components can take random values (even negative), it has no direct probabilistic explanation. Versions have been proposed to solve problems with this model [7].

However, an important drawback of all these models is that they cannot excuse the predictions made by system, due to the difficulty in understanding the components of two mentioned vectors being randomly generated.

In [8, 9], extensive survey of collaborative filtering techniques was presented. It included memory-based and model-based methods. It covers topics such as neighborhood selection, similarity measures, and data scarcity issues. Moreover, in [10] matrix factorization techniques for collaborative filtering in recommender systems was introduced. It discusses the underlying concepts, optimization algorithms, and the application of matrix factorization to large-scale datasets. On other hand, in [11] factorization machines, a powerful model for collaborative filtering, was introduced. It discusses the advantages of factorization machines over traditional matrix factorization methods and provides insights into the model's mathematical foundations and optimization techniques. While, the collaborative filtering is a traditional approach, a survey explored the integration of deep learning techniques into recommender systems in [12]. It discusses deep learning models, including neural networks, convolutional neural networks, and recurrent neural networks, and their applications in recommendation tasks.

Recently, a probability model dependence on Poisson distribution [13] was proposed in [14], which is concerned with other types of inputs, where instead of using a classification matrix as input. Some items are explicitly ordered by each user, and creating a matrix that contains the number of times whose user has used each item (without indicating whether the user liked the item). Because the values of the input matrix are number of times of expendables, the prediction was bad when calculating mean absolut error (MAE).

Specifically, we propose a methodology that uses matrix factorization technique. As well as, it exploits both local social contexts represented by modeling explicit user interactions and implicit user interactions with other users. Also, the global social context represented by the user reputation in the whole social network for making recommendations.

3.Research Methods

In probability and statistics, a probability distribution is defined as giving a certain probability for each measurable subset of the outcome set of a random experiment. Most of the important information about the variable is found in a probability distribution generated by each random variable. If there is a random variable X, then the corresponding probability distribution is attributed to the range [a,b], meaning that the probability that X takes a value within the range is P(a≤X≤b).

The Dirichlet Dir ($\vec{\varphi}$) distribution is a part of a group of multivariate continuous probability distributions. Its parameters are defined by a vector ($\vec{\varphi}$) of positive facts [15]. The probability density function (p(x)) is given according to (1):

$$p(x) = Dir(x|\alpha_k) = \prod_{k=1}^{k} x_k^{\alpha_k - 1} \quad : \quad x \in [0,1],\ \sum_{k=1}^{k} x_k = 1 \qquad (1)$$

While, Beta distribution is a continuous probability distribution defined within the range [0,1] by two positive parameters (α, β). These parameters represent the bases of the random variable and control the shape of the distribution [16]. The probability density function is given according to (2).

$$p(x) = Beta\ (x|\alpha,\beta) = x^{\alpha-1}(1-x)^{\beta-1} \quad (2)$$

Concerning binominal distribution [16], it is a distribution for a random experiment that has only two outcomes, one of which is the success of the experiment and the other its failure. The basic condition for this distribution is that the success of the experiment is not affected by its repetition. The probability density function is given according to (3).

$$p(x) \ = Bin(x, a) = a^x(1-a)^{n-x} \quad : \quad x \in [0, n] \quad (3)$$

Finally, categorical distribution is the distribution of a random experiment with limited sets of discrete values [14]. The special case of the polynomial distribution is where the number of times the experiment is performed is n = 1 and the size of the sample space is greater or equal to 2. The probability density function is given according to equation (4).

$$p(x) = \ Cat(x|\alpha) = \prod_{i=1}^{k} \alpha_i^{x_i} \quad : \quad \sum_{i=1}^{k} \alpha_i = 1, \sum_{i=1}^{k} x_i = 1\,, n = 1, k \geq 2 \qquad (4)$$

In the proposed methodology, we suggest a new technique for decomposing the classification matrix into factors [17], while maintaining the advantages found in traditional technologies. As in matrix factor analysis, there are unknown factors (K) that explain the ratings made by users. So, we will have two vectors of dimension K, one associated with the user's a_u and the other associated with the elements b_i. The values of the vectors α_u and b_i can take random real values in the classical matrix factor, but in the proposed methodology they must fall within the range [1, 0], which makes these values known, controlled and easy to understand. In contrast to the classical matrix operator, the values of α_u and b_i have an understandable probabilistic interpretation where, the unknown factors in the proposed methodology appear set of users who have the same preferences and K is the number of sets. Whereas, the value of α_{ump} indicates the percentage of user u belonging to group k, therefore:

$$\sum_{i=1}^{k} a_{u,k} = 1 \qquad (5)$$

The value of $b_{k,i}$ indicates the probability that item i will be liked by users of k group. The proposed methodology gives more accurate predictions for small values of K. Thus, less length of vectors α_u and b_i, which ensures more memory efficiency and less execution time for operations that use these vectors.

An algorithm will be described, which will be helpful in finding users' groups who share similar preferences. The following notation will be used for the classification matrix. N: the number of users in the rating matrix, M: the number of items in the classification matrix and $r_{u,i}$: the rating given by user to item. In many recommendation systems, user can give a rate to items on a scale of 1 to 5. In the proposed methodology, we will depend on the ratings {0.00,0.25,0.50,0.75,1.00} as the values for $r_{u,i}$, and the value $r_{u,i}$=• indicates that user u does not give a rate to item i.

In addition to classification matrix, there are several parameters that must be considered. Namely, K∈N: indicates the number of user groups. α∈(0,1): this parameter relates to the existence of a set of users who have the same preferences and the possibility of obtaining overlapping sets from them, and β>1: this parameter relates to the number of indicators needed to conclude that a group of users share a love for an item. The number of indicators needed for the model to conclude that a set of users like or dislike an item increases by increasing the β. The output of the proposed methodology is two matrices: matrix of size (N×K) that represents a matrix $\alpha_{u,k}$ as in (5) and matrix (K×M) is a matrix $b_{k,i}$.

Dependence on the output of the proposed methodology we can immediately calculate the basic parameters related to users and objects. Where, $p_{u,i}$ which is number of items characteristic the user has rated, is calculated according to (6), $q_{u,i}$ which is the expected rating of user u on element i. It can be obtained as in (7).

$$p_{u,i} = \sum_{K=1}^{K} \alpha_{u,k} . b_{i,k} \qquad (6)$$

$$q_{u,i} = \begin{cases} 1\ if\ 0.0 \le p_{u,i} < 0.2 \\ 2\ if\ 0.2 \le p_{u,i} < 0.4 \\ 3\ if\ 0.4 \le p_{u,i} < 0.6 \\ 4\ if\ 0.6 \le p_{u,i} < 0.8 \\ 5\ if\ 0.8 \le p_{u,i} < 1.0 \end{cases} \quad (7)$$

Each user u is represented by a random variable in the form of a vector with dimension K showing the probability of user u belonging to group k and using the Dirichlet distribution as in (8). Where, $\gamma_{u,k}$ are randomly selected variables representing coefficients to control the value of the Dirichlet distribution. The Dirichlet distribution is chosen here because it is a multivariate distribution whose parameters are determined by a vector φ of positive facts [18].

$$\overrightarrow{\varphi_u} \sim Dir(\gamma_{u,1}, \dots, \gamma_{u,k}) \quad (8)$$

Each item i is represented by a random variable that indicates the probability that users in set k will like the item i as in (9). The beta distribution depends on two positive parameters $(\varepsilon_{i,k}^{+}, \varepsilon_{i,k}^{-})$. These parameters represent the random variable base and have used to control the form of the distribution.

$$K_{i,k} \sim Beta(\varepsilon_{i,k}^{+}, \varepsilon_{i,k}^{-}) \quad (9)$$

The random variable $z_{u,i}$ that follows the categorical distribution and takes values (1,,k). The value of k represents that user u classifies element i as belonging to group k. On the other hand, the random variable $\rho_{u,i}$, which follows the binomial distribution and takes values between (0,R), where the value of R=4 represents the number of the element's characteristic rated by the user. The user rates the R characteristic of items and the rating indicates how many of these characteristic the user likes (the user will like the feature or not like the feature). We take normative classification from $\rho_{u,i}$ according to (10).

$$r_{u,i} = \frac{\rho_{u,i}}{R} \quad (10)$$

Now we know the classification matrix M which contains $\rho_{u,i}$ values. So, the distribution $p(\overrightarrow{\varphi_u}, K_{i,k}, z_{u,i} | \rho_{u,i})$ can be obtained through $q(\overrightarrow{\varphi_u}, K_{i,k}, z_{u,i})$ distribution according to (11) [16]. The distributions $q_{\overrightarrow{\varphi_u}}(\overrightarrow{\varphi_u})$, $q_{K_{i,k}}(K_{i,k})$ and $q_{z_{u,i}}(z_{u,i})$ follow the methodology proposed in [14, 16, 19], and the random variables are considered independent and each of them follows its own distribution.

$$q(\overrightarrow{\varphi_u}, K_{i,k}, z_{u,i})$$

$$\prod_{u=1}^{N} q_{\overrightarrow{\varphi_u}}(\overrightarrow{\varphi_u}) \prod_{i=1}^{M} \begin{matrix} \prod_{K=1}^{K} q_{K_{i,k}}(K_{i,k}) \\ \\ \prod_{r_{u,i\neq\bullet}} q_{z_{u,i}}(z_{u,i}) \end{matrix}$$

(11)

Where, φ_u is a conditional probability variable that follows a Dirichlet distribution. The Dirichlet distribution has the form shown in (12). Where, $\gamma_{u,1}$, ..., $\gamma_{u,k}$ are random variables that need to be learned in the training phase.

$$q_{\overrightarrow{\varphi_u}}(\varphi_u) \sim Dir(\varphi | \gamma_{u,1}, \dots, \gamma_{u,k}) = \prod_{k=1}^{k} (\varphi_u)^{\gamma_{u,k}-1} \quad (12)$$

Whereas, $K_{i,k}$ is a conditional probability variable that follows a Beta distribution. The Beta distribution has the form shown in (13). Where, $(\varepsilon_{i\ ,k}^{+}, \varepsilon_{i\ ,k}^{-})$ are random variables that need to be learned in the training phase.

$$q_{K_{i,k}}(K_{i,k}) \sim Beta(K_{i,k} | \varepsilon_{i,k}^{+}, \varepsilon_{i,k}^{-}) = (K_{i,k})^{\varepsilon_{i\ ,k}^{+} - 1} (1 - K_{i,k})^{\varepsilon_{\ i\ ,k}^{-} - 1} \quad (13)$$

Finally, $Z_{u,i}$ is a conditional probability variable that follows a categorical distribution as in (14). Where, $\lambda_{u,i,1}$, ..., $\lambda_{u,i,k}$ are random variables. These variables need to be learned in the training phase.

$$q_{z_{u,i}}(z_{u,i}) \sim Cat(\lambda_{u\ ,i\ ,1}, \dots\dots, \lambda_{u\ ,i\ ,k}) : \lambda_{u\ ,i\ ,1}, \dots\dots, \lambda_{u\ ,i\ ,k} = 1 \quad (14)$$

The algorithm will calculate the value of $\mu_{u,i,k}$. Where, $\mu_{u,i,k}$ is a random variable representing the probability that a user φ_u will belong to group k and the number of features $\rho_{u,i}$ that users belonging to group k. It will like according to (15) as proposed in [16].

$$ln\, q_1(z_1) = E_{z2}(ln\, p(z))$$

$$\Rightarrow q_1(z_1) = exp[E_{z2}\,(ln\,p(z))] \tag{15}$$

Thus, to find the value of $\mu_{u,i,k}$ we use the following form as in (16) [14, 19].

$$\mu_{u,i,k} = exp\{E_{q_{\vec{\varphi},q_k}}\,[ln(p(k|\,\vec{\varphi}_u) \times p(\rho_{u,i}|k_{i,k}))]\}$$

$$= exp\{E_{q_{\vec{\varphi}},q_k}\,[ln\,p(k|\,\vec{\varphi}_u) + ln\,p(\rho_{u,i}|k_{i,k})]\}$$

$$= exp\{E_{q_{\vec{\varphi}}}\,[ln\,p(k|\,\overrightarrow{\varphi_u})] + E_{q_k}[ln\,p(\rho_{u,i}|k_{i,k})]\} \tag{16}$$

Since z follows a categorical distribution. Then as in [16, 19]: $p(k|\,\overrightarrow{\varphi_u}) = \varphi_{u,k}$. While, $\rho_{u,i}$ follows the binominal distribution as in (17). Where, R is the number of item characteristic that can be rated by the user. So $\mu_{u,i,k}$ can be written as in (18) and (19).

$$p(\rho_{u,i}|k_{i,k}) = (k_{i,k})^{\rho_{u,i}}(1 - k_{i,k})^{R-\rho_{u,i}} \tag{17}$$

$$\mu_{u,i,k} = exp\{\,E_{q_{\overrightarrow{\varphi u}}}[ln\,\varphi_{u,k}] + E_{q_k}[ln\,p(\rho_{u,i}|k_{i,k})] \tag{18}$$

$$\mu_{u,i,k} = exp\{\,E_{q_{\overrightarrow{\varphi u}}}[ln\,\varphi_{u,k}] + \rho_{u,i}E_{q_k}[ln\,k_{i,k}] + (R - \rho_{u,i})E_{q_k}[ln(1 - k_{i,k})]\} \tag{19}$$

From characteristics of both Beta distribution and Dirichlet distribution [16, 19]:

$$E_{q_{\overrightarrow{\varphi u}}}[\ln \varphi_{u,k}] = \psi(\gamma_{u,k}) - \psi(\textstyle\sum_{k=1}^{k}\gamma_{u,k}) \tag{20}$$

$$E_{q_k}[ln\,k_{i,k}] = \psi(\varepsilon_{i,k}^{+}) - \psi(\varepsilon_{i,k}^{+} + \varepsilon_{i,k}^{-}) \tag{21}$$

$$E_{q_k}[\ln(1 - k_{i,k})] = \psi(\varepsilon_{i,k}^{-}) - \psi(\varepsilon_{i,k}^{+} + \varepsilon_{i,k}^{-}) \tag{22}$$

So, the equation of $\mu_{u,i,k}$ becomes as in (23). Where, ψ is digamma function. It defined as the logarithmic derivative of the gamma function as in (24), (25) and (26).

$$\mu_{u,i,k} = exp(\,\psi(\gamma_{u,k}) - \psi(\textstyle\sum_{k=1}^{k}\gamma_{u,k}) + \rho_{u,i}.\,\psi(\varepsilon_{i,k}^{+}) + (R - \rho_{u,i}).\,\psi(\varepsilon_{i,k}^{-}) - R.\,\psi(\varepsilon_{i,k}^{+} + \varepsilon_{i,k}^{-})) \tag{23}$$

$$\Psi(x) = (ln\,\Gamma(x))' = \frac{\Gamma'(x)}{\Gamma(x)} \tag{24}$$

$$\Gamma(x) = (x - 1)! \tag{25}$$

$$\Gamma'(x) = (x - 2)! \tag{26}$$

Since $Z_{u,i}$ follows a categorical distribution, then:

$$\lambda_{u\,,i\,,1}, \dots\dots, \lambda_{u\,,i\,,k} = 1$$

$$\lambda_{u,i,k} = \frac{\mu_{u,i,k}}{\mu_{u,i,1} + \dots + \mu_{u,i,k}} \tag{27}$$

The previous parameters are calculated within the training phase according to the methodology proposed I n [14, 19], as in (28), (29) and (30). Where, α is a parameter determines whether the user can belong to more than one K group. β is several indices used to identify users who share similar preferences and $r_{u,i}$ is the rating that user u gives to item i. So, we can calculate $a_{u,k}$ and $b_{k,i}$ according to (31) and (32).

$$\gamma_{u,k} = \alpha + \textstyle\sum_{i|r_{u,i}\neq\bullet}\lambda_{u,i,k} \tag{28}$$

$$\varepsilon_{i,k}^{+} = \beta + \textstyle\sum_{u|r_{u,i}\neq\bullet}\lambda_{u,i,k}.\,Rr_{u,i} \tag{29}$$

$$\varepsilon_{i,k}^{-} = \beta + \textstyle\sum_{u|r_{u,i}\neq\bullet}\lambda_{u,i,k}.\,R\big(1 - r_{u,i}\big) \tag{30}$$

$$a_{u,k} = \frac{\gamma_{u,k}}{\sum_{j=1}^{k}\gamma_{u,j}} \tag{31}$$

$$b_{k,i} = \frac{\varepsilon_{i,k}^{+}}{\varepsilon_{i,k}^{+} + \varepsilon_{i,k}^{-}} \tag{32}$$

Thus, the probability of user u liking item i is as shown in (33).

$$p_{u,i} = E(\frac{\rho_{u,i}|M}{R}) = \frac{1}{R}E(\rho_{u,i}|M) = \textstyle\sum_{k=1}^{K}a_{u,k}b_{k,i} \tag{33}$$

The accuracy of predictions of the proposed model was analyzed using MAE [20]. The MAE measures how far the true value can differ from its experimental value and is equal to the sum of the differences between the experimental value and the true value divided by the sample size. The MAE can be expressed according to (34). Where, y_i represents the value resulting from experimentation, x_i represents the true value (the correct value) and n is number of samples.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \quad (34)$$

The accuracy of recommendations of the proposed model was analyzed using the Precision and the Recall [21]. Where, the Precision is the ratio of correctly recommended items to the total number of recommended items. Whereas, the Recall represents the ratio of correctly recommended items to the number of items that should be recommended.

Simulations were performed using NetBeans using the Java language. The following datasets namely: MovieLens100K and MovieLens1M are used for evaluation because each dataset contains a different number of users (943 and 6040, respectively), items (1682 and 3706, respectively) and ratings (92026 and 911031, respectively). Therefore, is an unbalanced set and provides a more realistic and useful criterion for the algorithms of recommendation systems. We divided the ratings database into two groups: 80% of the ratings in the training set and 20% in the test set are used. We repeated the process 50 times and averaged the performance measures.

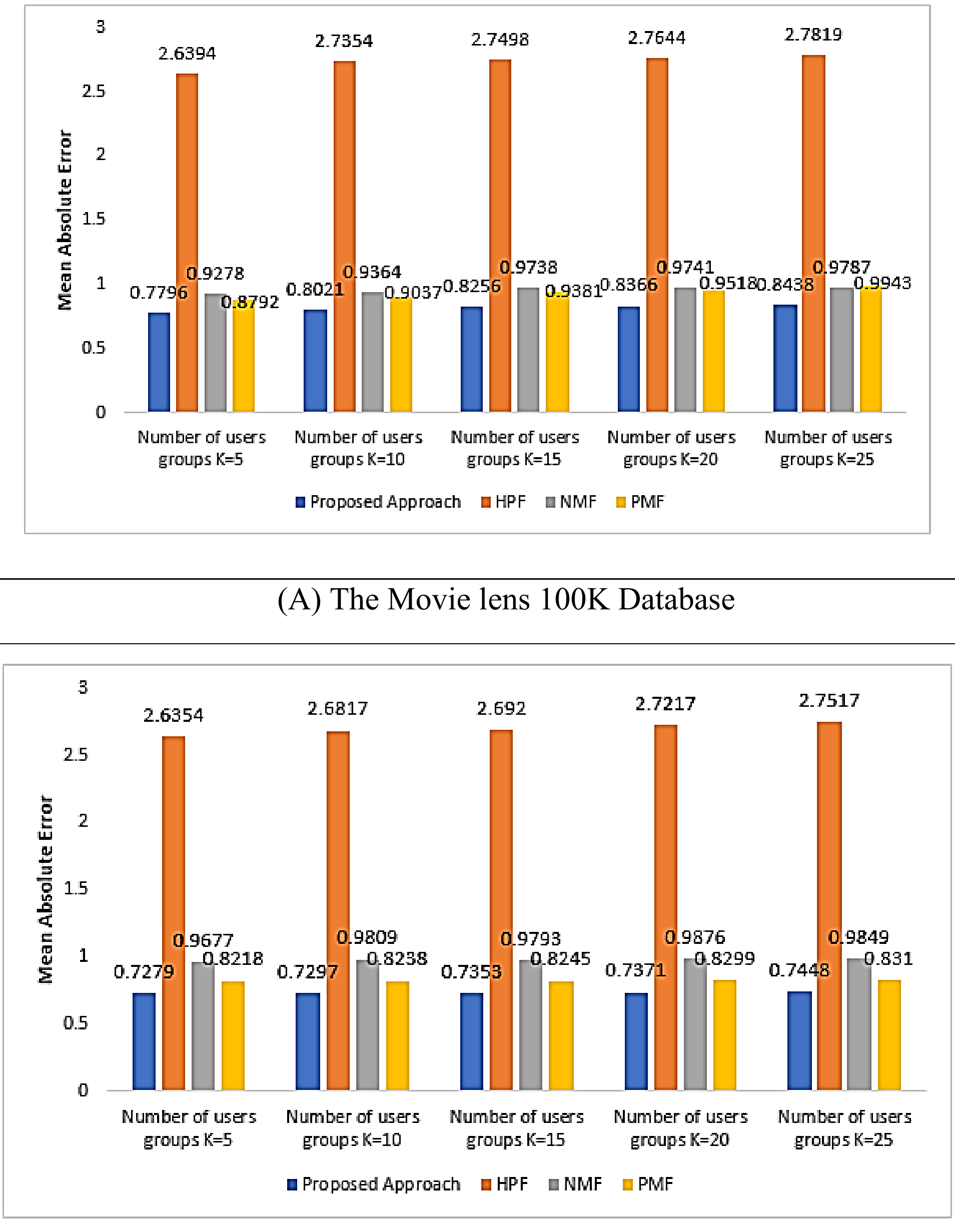
The algorithm can be presented as follow:

```
1- Input: M,a,β;
2- Output: $a_{u,k}^*$, $b_{i,k}^*$;
3- Initialize ⇄ randomly $\gamma_{u,k}$,$\varepsilon_{i,k}^+$,$\varepsilon_{i,k}^-$;
4- $rate_{u,i}^{old}$=1;
5- For each user u
6- For each item i rated by user u
7- For each factor k
8- Calculate $\lambda_{u,i,k}$ according equation (27);
9- Calculate $a_{u,k}$, $b_{i,k}$ according equations (31) (32);
10- $rate_{u,i}^{new} \leftarrow a_{u,k} * b_{i,k}$;
11- result ← min(abs($rate_{u,i}^{new}$ - $r_{u,i}$), abs($rate_{u,i}^{old}$ - $r_{u,i}$));
12- $rate_{u,i}^{old} \leftarrow rate_{u,i}^{new}$;
13- For each user u
14- For each factor k
15- Update $\gamma_{u,k}$ according equation (28);
16- For each item i rated by user u
17- For each factor k
18- Update $\varepsilon_{i,k}^{+}$ according equation (29);
19- Update $\varepsilon_{i,k}^{-}$ according equation (30);
20- Repeat from step 6;
21- $a_{u,k}^* \leftarrow a_{u,k}$ according min result;
22- $b_{i,k}^* \leftarrow b_{i,k}$ according min result;
23- Output: $a_{u,k}^*$ matrix, $b_{i,k}^*$ matrix
```

The methods explain clearly how the author carried out the research. The method must describe the research design clearly, the replicable research procedures, describe how to summarize and analyze the data.

4. Results and Discussion

The accuracy of predictions and recommendations of the proposed model were analyzed and we compared the proposed model with other models namely probabilistic matrix factorization (PMF) [6], non-negative matrix factorization (NMF) [22, 23] and hierarchical poisson factorization (HPF) [24] methodologies. Fig. 1 shows the results of calculating the MAE of the proposed methodology and the PMF, NMF, HPF methodologies using the Movie Lens 100K database (Fig. 1(a)) and using Movie Lens 1M database (Fig. 1(b)).



(A) The Movie lens 100K Database

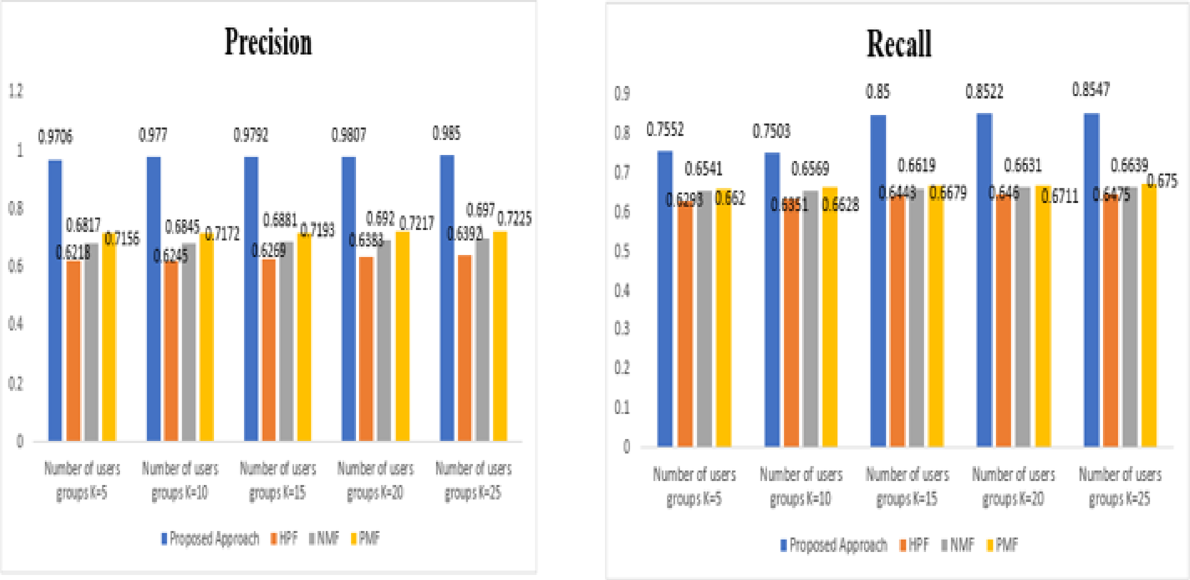
(B) The Movie lens 1M

Fig. 1. Mean Absolut Error (MAE) Of the Proposed Methodology and the PMF, NMF, HPF Methodologies Using (A) The Movie lens 100K Database And (B) The Movie lens 1M

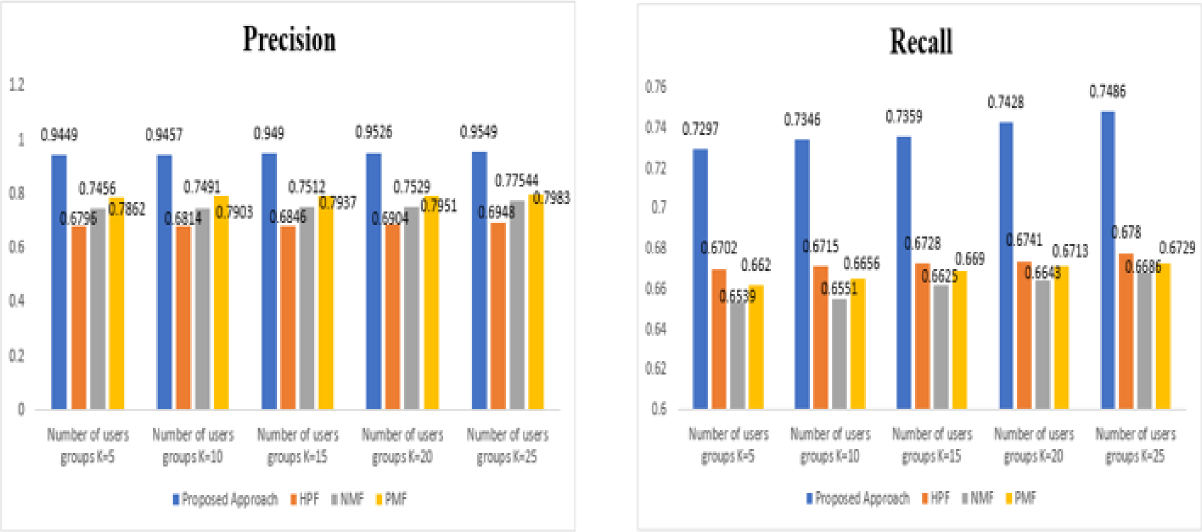
We note from Fig. 1(a) that the proposed approach is superior to other methodologies in terms of the mean absolute error, as the proposed model provides accurate predictions of the likelihood that users will like the items. Examining the Mean Absolute Error (MAE) results, the proposed approach consistently demonstrates superior predictive accuracy compared to HPF, NMF, and PMF across varying numbers of user groups (K). At K=5, the proposed approach achieves a MAE of 0.7796, outperforming HPF (2.6394), NMF (0.9278), and PMF (0.8792). This trend continues as K increases: at K=10 (0.8021 vs. 2.7354, 0.9364, 0.9037), K=15 (0.8256 vs. 2.7498, 0.9738, 0.9381), K=20 (0.8366 vs. 2.7644, 0.9741, 0.9518), and K=25 (0.8438 vs. 2.7819, 0.9787, 0.9943).

These results underline the consistent advantage of the proposed approach, emphasizing its ability to deliver more accurate predictions as the complexity of user segmentation increases. The widening gap in MAE scores between our approach and alternative methods with higher K values reinforces the robustness and scalability of the proposed approach in enhancing collaborative filtering-based recommendation systems. Also, we note from the Fig. 1(b) that the proposed approach remains superior to other methodologies in terms of the mean absolute error, despite the increase in the volume of data within the database, as the proposed model provides accurate predictions of the likelihood that users will like the items. Analyzing the Mean Absolute Error (MAE) results for the Movie Lens 1M dataset, the proposed approach consistently outperforms HPF, NMF, and PMF across various numbers of user groups (K). At K=5, the proposed approach achieves a MAE of 0.7279, surpassing HPF (2.6354), NMF (0.9677), and PMF (0.8218). This trend persists as K increases: at K=10 (0.7297 vs. 2.6817, 0.9809, 0.8238), K=15 (0.7353 vs. 2.692, 0.9793, 0.8245), K=20 (0.7371 vs. 2.7217, 0.9876, 0.8299), and K=25 (0.7448 vs. 2.7517, 0.9849, 0.831).These numerical results emphasize the robustness of the proposed approach in delivering more accurate predictions across different user group configurations within the Movie Lens 1M dataset. The consistently lower MAE scores highlight the efficacy of our approach in enhancing collaborative filtering-based recommendation systems for this larger dataset.

Fig. 2 shows the Precision and Recall results of the proposed methodology and PMF, NMF, HPF methodologies using the Movie Lens 100K database (Fig. 2(a)) and using the Movie Lens 1M database (Fig. 2(b)). We note from Fig. 2(a) that the Precision results of proposed methodology are higher than in other methodologies, and this confirms the results obtained in calculating the MAE. Also, we note from Fig. 2(b) that the Precision results of proposed methodology are higher than in other methodologies despite the increase in the volume of data within the database as the proposed model provides accurate recommendations to users, and this confirms the results obtained in calculation of the MAE. On the other hand, we note from Fig. 2(a) and (b) that the Recall results of the proposed methodology are higher than in other methodologies for the Movie Lens 100K and the Movie Lens 1M databases.



(A) The Movie lens 100K Database



(B) The Movie lens 1M Database

Fig. 1. Precision and Recall of The Proposed Methodology And PMF, NMF, HPF Methodologies Using (A) The Movie lens 100K Database And (B) The Movie lens 1M Database

We note from the previous results that the proposed methodology provides results that are superior to other methodologies, and we also note that the accuracy improves with the increase in the number of user groups, and the reason for this is the dependence on finer details in the recommendation, and the greater the number of groups, the greater the accuracy, and at the same time the processing operations necessary to reach the result increased. Therefore, the proposed methodology has great flexibility and a way of working that is compatible with all systems and available processing resources, where the system administrator can determine the number of groups according to the resources available to the processing center and according to the load on the processing center.

5. Conclusion

In this paper a methodology based on classical matrix factorization for filter-based recommendation systems is presented, in which a vector of elements is assigned to each user and each element. The components of these vectors differ from the classical matrix factorization, as they take values within the range from zero to one, which makes them superior to the second method, as they allow identifying and finding nested groups of users with the same taste, with an accurate prediction. This makes the proposed methodology highly flexible and this allows it to be run on different types of systems with different resources available for each of them.

In future works, this research can be developed by taking advantage of the great flexibility of the proposed methodology. Where, it is possible to create a system based on artificial intelligence that automatically adjusts the number of user groups in real time based on the data of the existing load and other parameters that can be studied.

References

[1] J. Bobadilla, A. Hernando, F. Ortega, and A. Gutiérrez, "Collaborative filtering based on significances," Information Sciences, vol. 185, pp. 1-17, 2012.

[2] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with

singular value decomposition," Information Sciences, vol. 180, pp. 4290-4311, 2010.

[3] E. R. Núñez-Valdéz, J. M. Cueva Lovelle, O. Sanjuán Martínez, V. García-Díaz, P. Ordoñez de Pablos, and C. E. Montenegro Marín, "Implicit feedback techniques on recommender systems applied to electronic books," Computers in Human Behavior, vol. 28, pp. 1186-1193, 2012.

[4] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: Challenges and remedies," *Artificial Intelligence Review,* vol. 52, pp. 1-37, 2019.

[5] N. Thongsri, P. Warintarawej, S. Chotkaew, and W. Saetang, "Implementation of a personalized food recommendation system based on collaborative filtering and knapsack method," International Journal of Electrical and Computer Engineering, vol. 12, pp. 630-638, 2022.

[6] J. Bobadilla, F. Serradilla, and A. Hernando, "Collaborative filtering adapted to recommender systems of e-learning," Knowledge-Based Systems, vol. 22, pp. 261-265, 2009.

[7] J. J. Castro-Schez, R. Miguel, D. Vallejo, and L. M. López-López, "A highly adaptive recommender system based on fuzzy logic for B2C e-commerce portals," Expert Systems with Applications, vol. 38, pp. 2441-2454, 2011.

[8] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," Advances in Artificial Intelligence, vol. 2009, p. 421425, 2009.

[9] [9]Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," ACM Computing Surveys, vol. 47, pp. 1-45, 2014.

[10] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol. 42, pp. 30-37, 2009.

[11] R.-Y. Sun, "Optimization for deep learning: An overview," *Journal of the Operations Research Society of China,* vol. 8, pp. 249-294, 2020.

[12] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning-based recommender system: A survey and new perspectives," ACM Computing Surveys vol. 52, p. Article 5, 2019.

[13] P. Gopalan, J. M. Hofman, and D. M. Blei, "Scalable recommendation with hierarchical Poisson factorization," presented at the Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, Amsterdam, Netherlands, 2015.

[14] P. Gopalan, J. M. Hofman, and D. M. Blei, "Scalable recommendation with poisson factorization," arXiv preprint arXiv:1311.1704, 2013.

[15] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," Knowledge-Based Systems, vol. 46, pp. 109-132, 2013.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," presented at the Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, Vancouver, British Columbia, Canada, 2001.

[17] Y. Yamaguchi, A. Sato, W. M. Boerner, R. Sato, and H. Yamada, "Four-Component scattering power decomposition with rotation of coherency matrix," IEEE Transactions on Geoscience and Remote Sensing, vol. 49, pp. 2251-2258, 2011.

[18] R. Lara-Cabrera, Á. González, F. Ortega, and Á. González-Prieto, "Dirichlet Matrix Factorization: A reliable classification-based recommender system," Applied Sciences, vol. 12, p. 1223, 2022.

[19] S. K. Lee, Y. H. Cho, and S. H. Kim, "Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations," Information Sciences, vol. 180, pp. 2142-2155, 2010.

[20] K. Li, X. Zhou, F. Lin, W. Zeng, and G. Alterovitz, "Deep probabilistic matrix factorization framework for online collaborative filtering," IEEE Access, vol. 7, pp. 56117-56128, 2019.

[21] M. Buckland and F. Gey, "The relationship between Recall and Precision," Journal of

the American Society for Information Science, vol. 45, pp. 12-19, 1994.

[22] T. Hofmann, "Latent semantic models for collaborative filtering," ACM Transactions on Information Systems, vol. 22, pp. 89-115, 2004.

[23] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol. 401, pp. 788-791, 1999.

[24] Y. Li and Y. Xiaoping, "Building complete collaborative filtering method system," in 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering, Hangzhou, China, 2010, pp. 412-417.