A Book Recommender System Using Collaborative Filtering Method

Sewar Khalifeh
Computer Engineering Dept., PSUT, Amman, Jordan
Sewarkhalifeh@gmail.com

Amjed Al-Mousa*
Computer Engineering Dept., PSUT, Amman, Jordan,
a.almousa@psut.edu.jo* Corresponding Author

ABSTRACT

Recommender systems are used to generate meaningful recommendations to users based on their preferences, which will be determined following several approaches. This work targets Arab readers by providing accurate and reliable results that match their needs and desirability. Eventually, it will enhance the reading experience for any Arab readers. The main approach is to filter the recommendations, and this can be achieved either by Content-Based filtering or by Collaborative Filtering. The collaborative filtering techniques presented in this paper compute the similarity matrix between items and users' ratings, and then evaluate the recommendations for users. The techniques cover User-Based and Item-Based Collaborative Filtering, as well as Matrix Factorization through an SVD algorithm. A comparison between these techniques is presented in terms of the fitting and testing time, and accuracy. The KNN-based algorithms showed better performance than the matrix factorization method with respect to fitting and testing time. However, the matrix factorization (SVD) algorithm had the best results in terms of accuracy.

CCS CONCEPTS

• **Applied computing** → Computers in other domains; Publishing; • **Human-centered computing** → Collaborative and social computing; Collaborative and social computing systems and tools; • **Computing methodologies** → Machine learning; Machine learning approaches; Factorization methods.

KEYWORDS

User-Based Collaborative Filtering, Item-Based Collaborative Filtering, Matrix Factorization, Recommender Systems

ACM Reference Format:
Sewar Khalifeh and Amjed Al-Mousa*. 2021. A Book Recommender System Using Collaborative Filtering Method. In *International Conference on Data Science, E-learning and Information Systems 2021 (DATA'21), April 05–07, 2021, Ma'an, Jordan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3460620.3460744

DATA'21, April 05–07, 2021, Ma'an, Jordan

ACM ISBN 978-1-4503-8838-2/21/04…$15.00
https://doi.org/10.1145/3460620.3460744

1 INTRODUCTION

The Internet offers many items and products for users to purchase; this has raised a need for a system that filters options to meet users' preferences. The system delivers personalized recommendations to overcome information overload. Recommender Systems helped to provide the best decision-making experience a user can have on the Internet. They guide the user to find what they may like and purchase. Since Recommender systems are a decision-making strategy [1], They predict users' preferences following several methods. The methods might rely on the features of an item, similarities between users and items, or a hybrid approach that combines both to generate suitable suggestions.

The content-based method analyzes users' history of purchasing items and makes recommendations for items with similar features. This method suffers from limited data analysis and relies on the history of users purchases only [1]. It requires more item description to get higher accuracy in recommendations. In collaborative filtering, recommendations are based on similarities between the active user and other users, or the similarities between items ratings. Collaborative filtering has become the most commonly used method to recommend items for users [2]. It can be classified into neighborhood-based (also known as memory based) and model-based.

The neighborhood-based collaborative filtering works by finding the nearest neighbors to the target user, then construct recommendations according to neighbors. Neighborhood-based collaborative filtering can be done following two techniques; user-based and item based. In the user-based technique, the algorithm focuses on similarities between users using their items rankings. Then combines the nearest neighbors' preferences, and generate the top N recommendations for the target user. In the item-based technique, it analyzes the user-item matrix to identify relationships between different items. Then use these relationships to compute recommendations for users indirectly [3]. Item based provides less online computation since the relation between items is static, while in the user-based, it is affected by behavioral changes. In the model-based collaborative filtering, it builds a model without involving the whole dataset. Then the system uses this model to make recommendations faster and with higher scalability. One of the most famous model-based techniques is Matrix Factorization. It works by reducing the dimensionality of the user-item matrix and computes approximate predictions.

This paper will cover testing the neighborhood-based collaborative filtering techniques and Matrix Factorization model-based technique on Arabic Books Dataset. Eventually, evaluate errors, analyze results, and compare between algorithms.

2 LITERATURE REVIEW

This section will discuss similar work on recommender systems and the use of collaborative filtering techniques. The interest in recommender systems has increased after the launch of the popular Netflix prize competition in 2006 [4]. The challenge was to improve the recommendation process for almost 100 million movie ratings in the Netflix dataset.

The first paper on collaborative filtering techniques was Tapestry in 1992 [5]. The system did record users' reactions and contributed by providing the records to other users as feedback. Many collaborative techniques were explicitly discussed and analyzed in research papers revealed a high performance presented by the collaborative filtering method [6]. It concludes that using this technique would be more reliable and justified if the number of users was higher than the number of items, which overcome the data sparsity problem. Another work that analyzed the similarity measures used in collaborative filtering is a study by Agarwal, Ajay, and Minakshi Chauhan (2017) [7]. It represents every similarity measure used in recommender systems, especially the most important ones, such as Pearson correlation coefficient, cosine similarity, and mean squared difference. Given that these similarity measures are not be enough to evaluate the appropriateness of the recommendations. The most effective factor in finding the best recommendations is the data, to get the real output you should provide real input data. This paper considered applying collaborative filtering techniques on an Arabic book dataset to compare methods and use different evaluation metrics to decide which has the best performance in terms of accuracy and speed. The methods include user-based and item-based collaborative filtering, as well as matrix factorization, which will use the mean squared differences to evaluate similarities. Afterward, the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) were used to evaluate each algorithm and finally generate the top N predictions for users using each method's recommendations.

3 EXPERIMENTAL SETUP

The dataset used to implement the book recommender system is the Books Reviews in Arabic Dataset (BRAD) [8]. It contains 510,600 Arabic language book reviews from 76,530 users and 4,993 different titles. The reviews were collected from GoodReads.com website in 2016.

The data distribution is in the following format, rating: review_id: book_id: user_id: review: where the scale of ratings ranges between one and five stars per user.After applying collaborative filtering methods, the ratings, user_id and the item_id columns will only be needed. The reviews and review_id data columns are dropped. The minimum number of reviews per book is one; this gives a better insight through the data and working with the ratings. On average, a user reviewed seven books, and each book received around 102 reviews. Using this clean dataset – that avoids null data - raises the impact of testing the filtering methods. See Table 1

Fig. 1 shows the book rating distribution. A left-skewed plot can be observed which indicates a higher number of ratings between three and five. Fig. 2 shows the number of reviews for each category, it depicts the total number of reviews per category plotted logarithmically [8].

Table 1: BRAD Statistics

Title	Number
Number of Reviews	510,598
Number of Users	76,530
Avg. reviews per user	7
Max reviews per user	396
Min reviews per user	1
Number of books	4,993
Avg. Reviews per book	102
Max Reviews per book	5,522
Min reviews per book	1

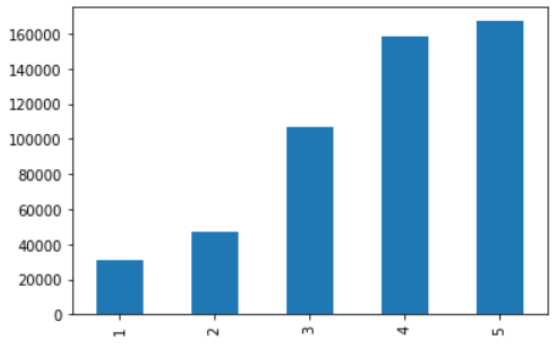

Figure 1: Rating Distribution in BRAD Dataset

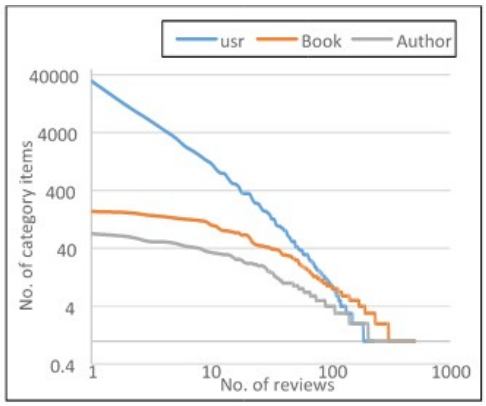

Figure 2: Classification of books reviews' ratings

After importing the dataset, all the algorithms needed can be found in the Surprise Library [9]. Surprise library is a recommender system specialized library, it provides built-in algorithms and datasets that are very useful and precise. The algorithms were implemented using python and within Jupyter notebook environment to provide the programming modularity needed and to be able to test different models and evaluate them quickly.

4 ALGORITHM

This section provides a theoretical analysis behind every method applied to the Arabic book reviews dataset. The ultimate goal of building any recommendation system is generating reliable recommendations that satisfy users' real needs. The first step is to find similarities between users and items in the neighborhood-based class, or by reducing the user-item matrix dimensionality in the model-based class.

User-based collaborative filtering uses similarities between users that are listed in a matrix of ratings between users and items. Those similarities are being computed using some similarity measure such as Cosine similarity, Mean Squared Difference (MSD), or Pearson correlation. All of the mentioned measures are offered in the Surprise library used [9].

4.1 Computation of Similarity

This section evaluates the Mean Squared Difference (MSD) between pairs of users in the User-Based collaborative filtering, and between pairs of items in the item-based collaborative filtering. MSD only considers the absolute ratings, but not consider the number of common ratings [7].

$$msd(u,v) = \frac{1}{I_{uv}} \bullet \sum_{i \in I_u v} (r_u i - r_v i)^2 \quad (1)$$

$$\text{msd}_{sim(u,v)} = \frac{1}{msd(u,v)+1} \quad (2)$$

It starts by finding the MSD between two vectors u and v (can be users or items) using the ratings of the first vector rui and the second vector rvj and Iuv are the set of items rated by users u and v, shown in equation 1). In the second equation, the similarity score is calculated according to the average difference between two vectors (ratings).

To provide a good prediction analysis on the testing stage, the same similarity measurement in item-based and user-based techniques is used. The mean of these squared differences would then provide a measure of similarity: the lower the mean squared difference, the greater the similarity [10].

4.2 Top N Recommendations

In the neighborhood-based collaborative filtering, computing the nearest k neighbors for the active user is a key factor to evaluate the best-predicted recommendation. Choosing the constant k is what controls the performance of finding the target rating. Constant k is the number of users or items near the unknown value; it calculates the similarities between them and selects the nearest n neighbors with the highest correlation.

4.3 Matrix Factorization

Matrix Factorization, shown in Fig. 3, is used over the user-item matrix of ratings. The Singular Value Decomposition (SVD) function reduces the matrix dimensionality and makes it easier to evaluate similarities in the complex matrices.

Assuming a matrix of ratings (U × I), reducing dimensionality will produce two matrices (U × K) and (I ×K) such that their product produces the original (U × I) rating matrix. The prediction $\hat{r_u i}$ is

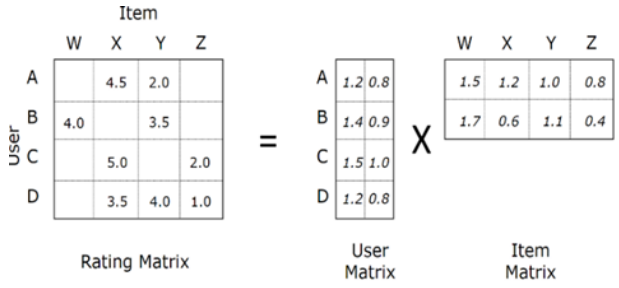

Figure 3: Matrix Factorization

set as:

$$\hat{r_u i} = \mu + b_u + b_i + q_i^T p_u \quad (3)$$

The SVD function provided by Surprise library is used [5]. Documentation of the function provided the details of the SVD evaluation process and its application over the rating dataset used. The terms in equation 3) represent the following:

- μ: the overall average of the ratings
- bu: user bias
- bi: item bias
- pu: vectors for users
- qi: vector for items

Both qi and pu represent the vectors for items and users, such that the dot product in the last term of the equation denotes a certain interaction between the user and the item vectors that finally leads to the user's overall predicted interest in a certain item [11]. The predicted rating can be calculated using the SVD equation, for anonymous user u, the bias bu and the factors pu are considered to be with value zero. The same applies to anonymous items with the bias bi and the factors qi, shown in equation 3).

The biggest problem of using SVD in collaborative filtering algorithms is its high computational cost [12]. An implementation of a Stochastic SVD algorithm was found to overcome this issue.

5 RESULTS AND ANALYSIS

The user-based, item-based and matrix factorization techniques were validated using accuracy measures such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and the Precision and Recall. The accuracy measures give an indication of how relevant and reliable the prediction is. Then, the models mentioned above are applied on a test set. After calculating the error measures, it starts a comparison between them. A cross-validation with five folds is used the resulting average error for each metric is computed.

Using the cross-validation function provides fitting and testing time in the final output. The User-Based and Item-Based models reached the minimum fitting time of one second on average. Here, the testing time was the highest between the different models; average of 4.4 seconds. The maximum fitting time recorded by the matrix factorization model can be observed, around 9.7 seconds, as shown in Fig. 4

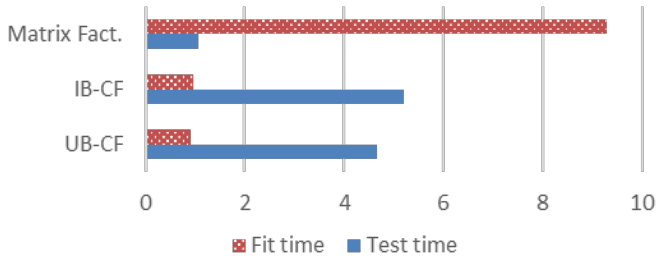

Figure 4: Test and Fit time for each model

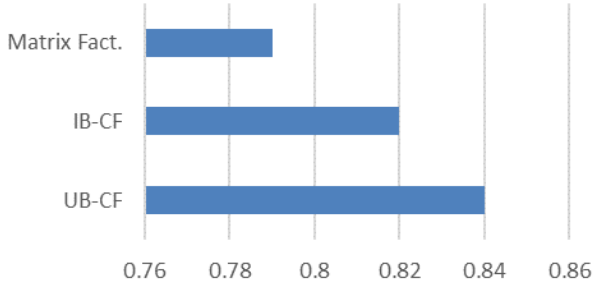

Figure 5: Average MAE score for each model

5.1 Mean Absolute Error (MAE)

MAE is less sensitive to outlier instances that make it a good indicator for models that puts a huge weight on the big relevant changes in the data. In average, MAE treats the individual differences weights equally.

$$MAE = \frac{1}{n}\sum_{(j=1)}^{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{4}$$

The MAE from the previous formula, shown in Fig. 5, is calculated on the test set for each model. For the user-based model, a MAE score of 0.8463 was computed. In item based, the score was 0.8260 and finally Matrix Factorization scored 0.7907 error, which is the lowest MAE value, this gives an indication about which model will be better for the system. Here it is the matrix factorization (SVD) function.

The MAE scores show on the average case how much the model would diverge from the target predicted value for the item, which is in this case, the books. The goal is to apply the model with the minimum MAE score to assure better results that are reasonable enough for the user. See equation 4).

5.2 Root Mean Squared Error (RMSE)

RMSE measures the root of the average squared error of the predictions. It calculates the square difference between the prediction and the real target value, then calculate average those values and produce the root value of the calculation. RMSE is known to be highly susceptible to outliers (unlike MAE). RMSE is sensitive to any unexpected changes in the instances. For data full of noise, it

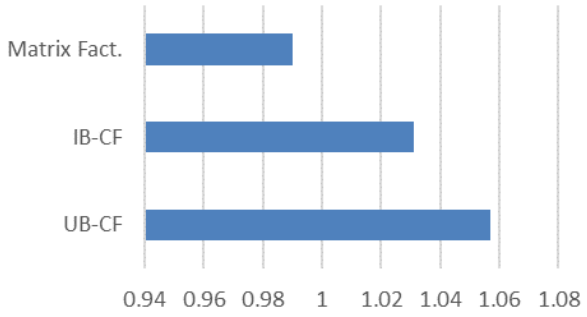

Figure 6: Average RMSE score for each model

is recommended to evaluate errors using MAE, not RMSE. RMSE, shown in equation 5), provides the scale of the errors the same as the scale of target data, which is, in this case, the ratings between one and five.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\Pr edicted_i - Actual_i)^2}{N}} \tag{5}$$

RMSE scores in the test were higher than the MAE scores. This shows how RMSE skewed due to outlier values. In user-based the RMSE score was 1.0578, in item based the RMSE was 1.0315 while in matrix factorization it scored the lowest RMSE value 0.9928. Matrix factorization performance can be generalized as the best model used in the book recommender system which is observed in Fig. 6

Fig. 6 shows the average RMSE scores for each model, the matrix factorization model proved that the dimensionality reduction applications in the recommender system lead to a better, more accurate results. Matrix factorization reduces the noise in the training process and discards unnecessary data and that speeds up the testing while providing better performance.

5.3 Precision and Recall

The precision metric evaluates the models' ability to find all target predictions. The recall metric evaluates how much target predictions were actually retrieved. In the recommender system case, the collaborative filtering techniques that rely on the Top N recommendations for the target user is used. For this definition, the precision and recall values are calculated at a certain k value, where k is the value that matches the N recommendations specifications. By assumption, the user examines the top-K results. The value K has a huge impact on the result. It is important to choose the best k value that evaluates the best Precision and recall score. The K values chosen are the range between 1 and 10.

For the optimal general case in precision@K and recall@K, their values must be 0.5 for each. Using K=1 leaves a low value of recall which means that a big amount of good recommendations were not presented to the user and this should not happen. Instead, the value of K is raised to five and again to 10 and by this, the results improved clearly. For example, precision@K and recall@K reached very good values around 0.6 and 0.7 in each model as presented in Fig. 7

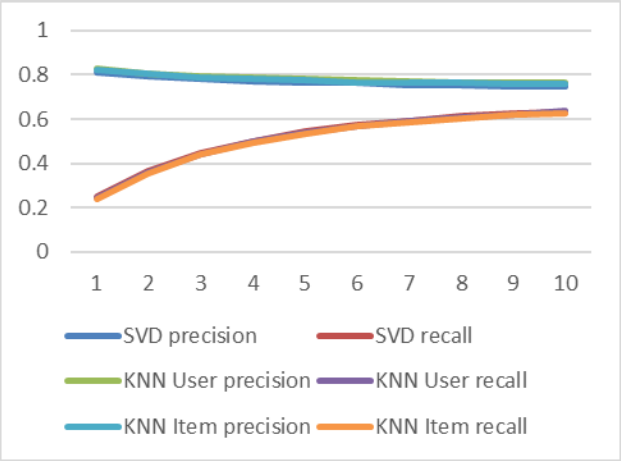

Figure 7: Precision and Recall for different K values

It is observed that the precision@K and recall@K values differed each time the K value changed. The three algorithms showed the same behavior with each increase in K. This happened because the differences between the algorithm results were convergent in terms of the precision and recall measurements.

6 CONCLUSION

Recommender systems encouraged the lifestyle choices people take every day. Books are one of the richest and diverse contents, which make it difficult to select or recommend one. Thus, building a good recommender system requires a good filtering model, the appropriate similarity measure, and the use a good accuracy metric to improve the systems' ability to predict the right items for users.

This paper introduced different collaborative filtering techniques, and used different accuracy metrics (such as the RMSE, MAE, precision, and recall) to analyze and compare these models to select the best fit for the data. Moreover, it compared the performance different models for this problem. In terms of fitting and testing time, the KNN-based algorithms showed better performance than the matrix factorization method represented in the SVD algorithm. However, the matrix factorization (SVD) algorithm had the best results in terms of accuracy.

ACKNOWLEDGMENTS

The authors would like to thank PSUT for supporting the publication of this research, which is an extension of a machine learning class project.

REFERENCES

[1] Isinkaye, F., Folajimi, Y. and Ojokoh, B. (2019). Recommendation systems: Principles, methods and evaluation.
[2] Agarwal, Ajay, and Minakshi Chauhan. "Similarity Measures used in Recommender Systems: A Study." International Journal of Engineering Technology Science and Research IJETSR, ISSN (2017): 2394-3386.
[3] Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Www, 1, 285-295.
[4] Bennett, James, and Stan Lanning. "The netflix prize." Proceedings of KDD cup and workshop. Vol. 2007. 2007.
[5] Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12), 61-70.
[6] Ponnam, L. T., Punyasamudram, S. D., Nallagulla, S. N., & Yellamati, S. (2016, February). Movie recommender system using item based collaborative filtering technique. In 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS) (pp. 1-5). IEEE.
[7] Liu, Haifeng, *et al.* "A new user similarity model to improve the accuracy of collaborative filtering." Knowledge-Based Systems 56 (2014): 156-166.
[8] Elnagar A. and Einea O. 'BRAD 1.0: Book reviews in Arabic dataset'. 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), pp. 1-8, Nov 2016. DOI: 10.1109/AICCSA.2016.7945800.
[9] Hug, Nicolas. "Surprise, a Python library for recommender systems." URL: http://surpriselib.com (2017).
[10] Al Hassanieh, Lamis, *et al.* "Similarity measures for collaborative filtering recommender systems." 2018 IEEE Middle East and North Africa Communications Conference (MENACOMM). IEEE, 2018.
[11] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.
[12] Bokde, Dheeraj & Girase, Sheetal & Mukhopadhyay, Debajyoti. (2015). Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey. Procedia Computer Science. 49. 10.1016/j.procs.2015.04.23.