A New Collaborative Filtering Recommendation Algorithm Based on Dimensionality Reduction and Clustering Techniques

Hafed Zarzour[1], Ziad Al-Sharif [2] , Mahmoud Al-Ayyoub[2], Yaser Jararweh[2]
[1]University of Souk Ahras, 41000, Souk Ahras, Algeria
[2]Jordan University of Science and Technology, Irbid, Jordan

***Abstract*— With the advent and explosive growth of the Web over the past decade, recommender systems have become at the heart of the business strategies of e-commerce and Internet-based companies such as Google, YouTube, Facebook, Netflix, LinkedIn, Amazon, etc. Hence, the collaborative filtering recommendation algorithms are highly valuable and play a vital role at the success of such businesses in reaching out to new users and promoting their services and products. With the aim of improving the recommendation performance of such an algorithm, this paper proposes a new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques. The *k*-means algorithm and Singular Value Decomposition (SVD) are both used to cluster similar users and reduce the dimensionality. It proposes and evaluates an effective two stage recommender system that can generate accurate and highly efficient recommendations. The experimental results show that this new method significantly improves the performance of the recommendation systems .**

Keywords*— Collaborative filtering recommendation algorithm; recommender systems; dimension reduction; clustering; SVD

I. INTRODUCTION

Nowadays, recommender systems have become one of the most promising techniques for online companies specializing in Internet-related services and products. Google, YouTube, Facebook, Netflix, LinkedIn, and Amazon are typical examples, in which these recommender systems play a vital role into the core of their business model. These systems predict users' preferences based on their behaviors and help improving the satisfaction of users towards the promoted items. Recommender systems are now considered an essential part of many e-business corporations and various domains ranging from movies, books, and news to research articles. These systems use sentiments that are acquired about products and services from communities of users and promote these products for other users with similar interests. For instance, recommender systems can explore the existing connections between users and their friends and automatically recommend new friends for an active user in a social network context.

Recommender systems are divided into three categories: content-based filtering, collaborative filtering or hybrid methods [1–4]. Content-based filtering methods focus on both the profile of the user's preferences and the item description in order to recommend items that are most similar to the items that are highly rated in the past. Collaborative filtering methods take into consideration a variety of criteria such as users' preferences, activities, and behaviors and recommend items based on the similarities to other users. Hybrid methods combine content-based filtering and collaborative filtering and build on their advantages in order to recommend more items that are suitable.

Collaborative filtering recommendation is popular and most commonly used in practice because of its simplicity and ease of implementation. The collaborative filtering recommendation systems can be classified into two groups: memory-based and model-based [5, 6]. In comparison to memory-based approaches that use the entire similarities between users or items to make predictions, model-based approaches use only a set of ratings to train the model, which is then employed to make a prediction for users' rating of an unrated item or set of items.

This paper introduces a new collaborative filtering recommendation algorithm based on the dimensionality reduction and clustering techniques. The aim is to improve the performance of recommender systems and to overcome their problems of sparsity and cold-start as well as their scalability issues. The *k*-means algorithm and Singular Value Decomposition (SVD) are both used to cluster similar users and reduce the dimensionality, respectively. SVD is one of the dimensionality reduction techniques that are recognized for their capacity to improve the scalability of recommender systems [7, 8]. The experimental results show that our method significantly improves the performance of the recommendation systems.

The rest of this paper is structured as follows: Section II discusses some related studies. Section III explains in details the proposed approach. Section IV describes the experimental results. Finally, Section V concludes this study and proposes the plans for future work.

978-1-5386-4366-2/18/$31.00 ©2018 IEEE

II. Related Work

In the past decades, many studies have been conducted to develop methods for recommender systems and improve their accuracy. In [9], the authors proposed a fuzzy c-means approach for a collaborative user-based filtering system. They used the MovieLens datasets to compare the different techniques of clustering. In [10], the authors investigated the applicability of the cluster ensemble approaches for recommender systems. They utilized *k*-means and Self-Organizing Maps (SOM) as baseline clustering techniques, and the multiple clustering ensemble technique to combine the results of clusters. In [11], the authors proposed a keyword-aware service recommendation method, named KASR, to indicate users' preferences and generate appropriate recommendations on MapReduce [12] for big data applications. In [13], Lee et al. proposed an adaptive recommendation algorithm, ACFSC, that is focused on scalable clustering. They addressed the problem of scalability by composing neighborhood based on reducing time complexity. They also addressed the problem of sparsity by making items' and users' feature vectors incrementally learning. In [14], the authors proposed a typical model that integrates collaborative filtering, clustering techniques, and social network analysis (SNA) in order to enhance the perdition accuracy results in recommender systems. Their model uses SNA to identify the people who are most influential on social networks and then uses these people to conduct clustering analysis. Following that, the model focuses on cluster-index collaborative filtering to make accurate recommendations. Additionally, Tian et al. [15] developed a new method for improving recommendation quality by formalizing trust relationships in online social networks. In [16], authors proposed two varieties of algorithms for developing an effective recommender system. The first one uses the improved *k*-means clustering technique while the second one uses the improved *k*-means clustering technique coupled with principal component analysis to enhance the recommendation accuracy for big data.

III. Proposed Appraoch

A key contribution of our work is to construct, in two main phases, an effective recommender system that can generate accurate recommendations regardless of the dataset size. The first stage is called *offline model creation*. In this stage, the model of recommendation is created by clustering the users' ratings following their preferences, reducing the dimensions of data and then calculating the similarities. The *k*-means algorithm and SVD technique are both used in this stage to cluster similar users and reduce the dimensionality, respectively. The second stage is the *online model utilization* wherein the created model is used in producing accurate recommendations for a given active user.

A. Users Clustering

K-means method is one of the most popular clustering algorithms that has been used extensively in data mining and in most recommender system industry. In our context, we use *k*-means to create *k* clusters, each of which contains users having similar preferences in terms of ratings. Therefore, the user clustering process helps in improving the recommendation performance because the considered cluster contains much fewer users in comparison with the general population that consists of all users.

To adapt the traditional *k*-means method to be used as a user clustering technique for recommender system, some of its steps need to be modified. First, random *k* users are selected as the initial center of the *k*-clusters. Second, the rest of users are assigned to nearest clusters in terms of the distance between them and the center of each cluster. A similarity measure is used to calculate the distance value. Third, a new mean of the user's cluster is calculated to define the new center for each cluster. Fourth, for each user, the distance is recalculated in order to define to which cluster the user should be added. Finally, the re-assignment of users according to their distances is repeated until the termination criteria are met. The steps of the user clustering are as follows:

Step 1: Input user-item rating matrix, k cluster;

Step 2: Randomly select initial k users clustering centers;

Step 3: Calculate the distances between centers and users, then assign users to the most nearest cluster;

Step 4: For each users' cluster, calculate the average as new partition centers;

Step 5: Use the new partition centers to redistribute users into new clusters;

Step 6: Repeat Steps 4 and 5 until the algorithm converge to a stable partition;

Step 7: Output k cluster represented as a center-items rating matrix.

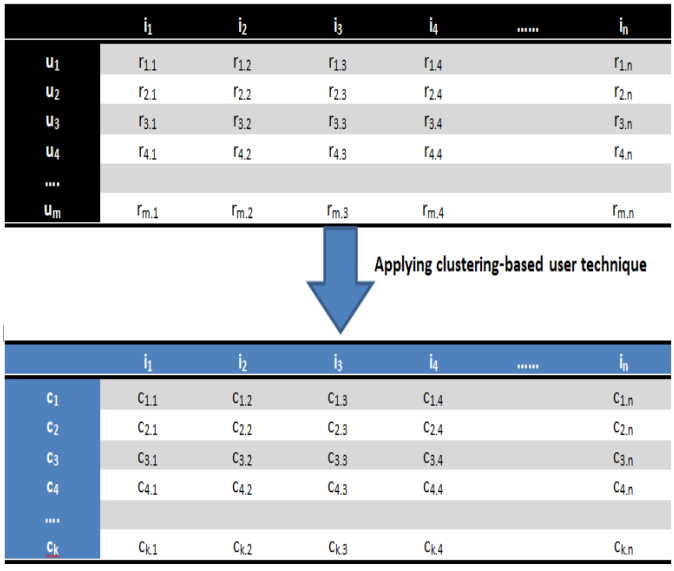

Fig. 1. Creation of cluster center-item rating matrix from user-item rating matrix.

Figure 1 shows how a new cluster center-item rating matrix is obtained from the initial user-item rating matrix after applying the *k*-means algorithm. In the user-item rating matrix, the columns represent items, rows represent users, and $r_{i,j}$ represents the rating of a user i on an item j. Similarly, in the cluster center-item rating matrix, the columns represent items, rows represent cluster centers of users, and $c_{x,y}$ represents the average rating of users cluster x on an item y.

B. SVD Technique

Singular Value Decomposition (SVD) is one of the most important matrix factorization methods usually used for reducing the number of features of a set of data. This implies, for example, the reduction of space dimensions from A to B where B < A. The SVD theorem states that for all rectangular matrix X[n,m] in which the *n* rows represents the clusters centers of users and the *m* columns represents the items, X may be decomposed as follow: ($X=U \cdot S \cdot V^T$), where U is an orthonormal matrix of size m *x* r, in which its r columns are the left singular vectors, S is a diagonal matrix of size r *x* r having singular values of X, and V^T is an orthonormal matrix of size r *x* n having the right singular vectors. More precisely, the matrix X includes m clusters centers of users and r factors. The r in the diagonal matrix S is the rank of the matrix and the V matrix contains n items and r factors.

Figure 2 shows how to reduce the dimension of a given center-items rating matrix by converting it into clusters centers, singular values and items matrices. Cosine Similarity (COS) is used to compute the similarity. COS is one of the most extensively used *similarity* measures in collaborative filtering recommendation systems. The formula is:

$$COS(x,y) = \frac{\sum_{i=1}^{n} p_{x,i} \times p_{y,i}}{\sqrt{\sum_{i=1}^{n} p_{x,i}^2} \sqrt{\sum_{i=1}^{n} p_{y,i}^2}} \qquad (1)$$

where COS(x, y) denotes the cosine similarity between user x and user y, n is the dimension the rating matix, $p_{x,i}$ is the preference of user x to item i, and $p_{y,i}$ is the preference of user y to item i. Therefore, the steps of the offline stage algorithm are as follows:

Step 1: Input user item matrix including users' ratings data;

Step 2: Create users clusters using *k*-means;

Step 3: For each cluster, apply SVD to obtain the decomposition matrices;

Step 4: For each matrix obtained from the decomposition step, calculate the similarity;

Step 5: Output recommendation model.

Once the recommendation model is created and trained in the offline stage, the tasks of prediction and the recommendation can be performed in the online stage. In this stage, we perform the SVD calculation to find the neighbors of active user using users' clusters. Thus, the steps of the online stage algorithm are defined as follows:

Step 1: Input active user u, item i and recommendation model;

Step 2: Use the original matrix X to find clusters containing users who rated the item i;

Step 3: Use formula 2 to predict the rating of the active user u;

Step 4: Output rating recommendation.

The prediction of the rating for a given user is performed by using the following formula:

$$c_{ij} = \bar{c}_{\,i} + (U_k \cdot S_k)_{i,} \cdot V_{,j}^T \qquad (2)$$

where c_{ij} is the rating to be predicted for an active user i on item j, $\bar{c}_{\,i}$ is the average rating in the cluster, $V_{,j}^T$ is the j^{th} column in the matrix V^T, $(U_k \cdot S_k)_{i,}$ is the i^{th} row of the matrix resulting from dot product of $(U_k \cdot S_k)$.

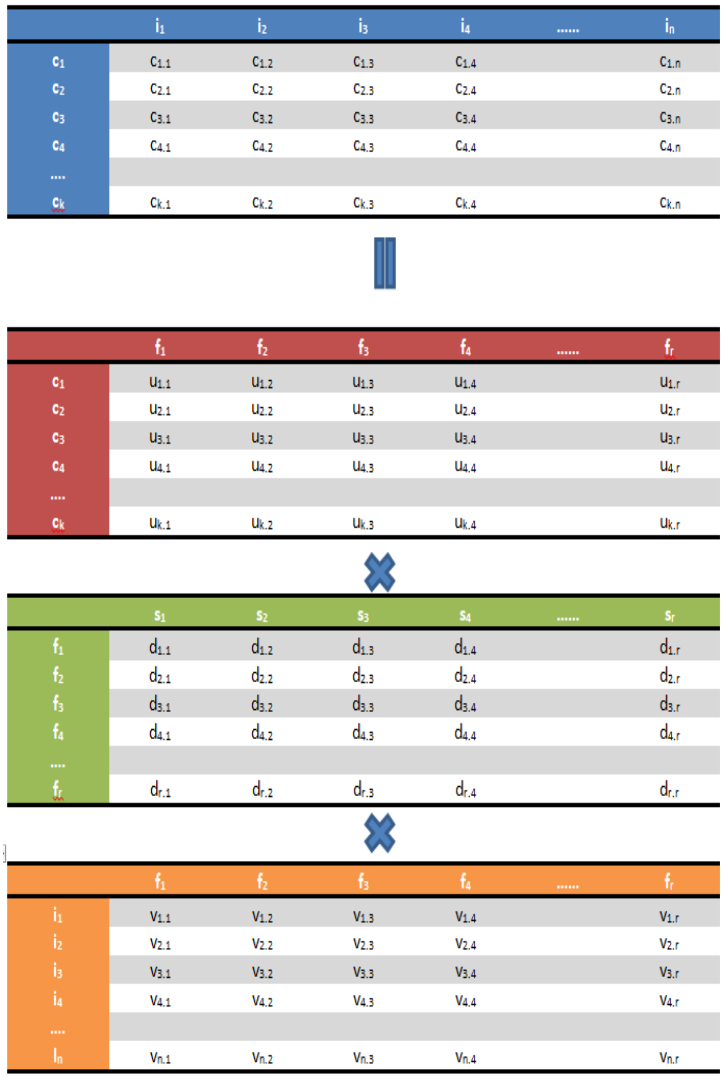

Fig. 2. SVD procedure.

IV. Experimental Results

To evaluate the effectiveness of the proposed approach, two of the most popular datasets are used. The first one consists of MovieLens 1M that includes about one million of ratings made by about six thousand participants rating about four thousand online movies, while the second one consists of MovieLens 10M that includes about ten million of ratings made by about seventy thousand participants rating about ten thousand movies [17]. The ratings used in MovieLens 1M and MovieLens 10M ranging from 1 to 5 stars and both datasets were created by GroupLens Research, which is a research lab in the Department of Computer Science and Engineering at the University of Minnesota. All of these datasets were split in training and testing sets with a percentage of 80% and 20%, respectively.

In order to compare the performance of the methods in the present experiment, the Root Mean Square Error (RMSE) is adopted as a predictive accuracy metric. RMSE was widely in evaluating recommender systems. More specifically, RMSE provides the difference between the true and predicted likelihood about a user selecting an item. A smaller value of RMSE suggests better performance. RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{}(p_{u,i} - r_{u,i})^2} \qquad (3)$$

where $p_{u,i}$ is the predicted rating for user u on item i, $r_{u,i}$ is the actual rating, and N is the total number of ratings on the items set.

Our approach uses k-means clustering and SVD techniques, noted k-means-SVD-based recommendation, is compared with k-means-based recommendation and k-nearest neighbor-based recommendation, respectively. The neighbors ranging from 10 to 100 are considered in this experimentation.

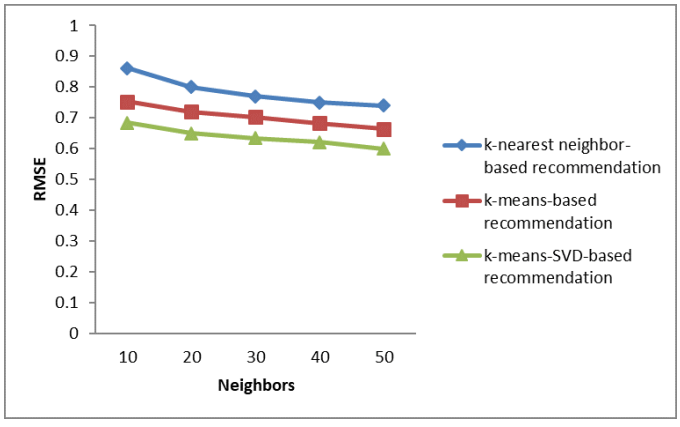

Fig. 3. Comparing different methods in terms of RMSE for MovieLens 1M.

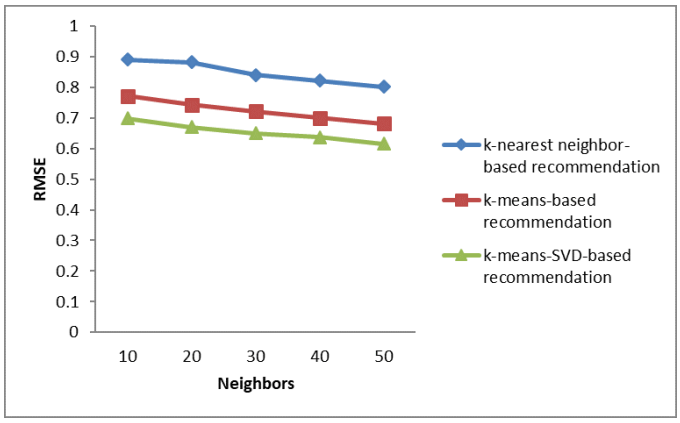

Fig. 4. Comparing different methods in terms of RMSE for MovieLens 10M.

Figure 3 presents the RMSE results of the k-means-SVD-based recommendation, k-means-based recommendation and k-nearest neighbor-based recommendation methods using MovieLens 10M. From this figure, it can be seen that our k-means-SVD-based recommendation method outperforms the two other methods for all neighborhood sizes on MovieLens 1M dataset in terms of the prediction accuracy. It can be also seen that the k-means-based recommendation method is better than the k-nearest neighbor-based recommendation method.

Figure 4 shows the RMSE results for all considered methods using the dataset MovieLens 10M. In this way, it can be found that our method remains the lowest values in the RMSE curve in the whole neighbors range compared to that for the k-means-based recommendation and k-nearest neighbor-based recommendation methods.

From Figures 3 and 4, the better prediction accuracy results can be explained by the fact that in the proposed method we use the potentialities provided by k-means clustering and SVD techniques.

V. Conclusion

In this paper, we proposed a new method for recommender systems that benefits from the potentialities provided by the k-means clustering algorithm and SVD technique. Firstly, the k-means clustering algorithm was adopted to cluster users in the same partition according to their preferences, and then the SVD was used in each cluster not only as a dimensionality reduction technique but also as a powerful mechanism, which could efficiently help in finding the most similar users. To evaluate the performance of the proposed method, we conducted experimentations on two real-world datasets for movies recommendation called MovieLens 1M and MovieLens 10M, which contain about 1 million and 10 million ratings made by anonymous users, respectively. In addition, RMSE metric was adopted to evaluate the predictive accuracy of the proposed method in comparison with well-known k-nearest neighbor-based recommendation and k-means-based recommendation methods.

The experimental results showed that our method improved significantly the performance of the recommendations and remained the lowest values in the RMSE curve in the whole neighbors range. As part our future work, we intend to experiment the proposed method on other datasets using other metrics such as *precision* and *recall*. We also plan to study the scalability of our method.

REFERENCES

[1] A. Merve Acilar and A. Arslan, "A collaborative filtering method based on artificial immune network," Expert Systems with Applications, vol. 36, no. 4, pp. 8324–8332, May 2009.

[2] L. CHEN, F. HSU, M. CHEN, and Y. HSU, "Developing recommender systems with the consideration of product profitability for sellers," Information Sciences, vol. 178, no. 4, pp. 1032–1048, Feb. 2008.

[3] M. Jalali, N. Mustapha, M. N. Sulaiman, and A. Mamat, "WebPUM: A Web-based recommendation system to predict user future movements," Expert Systems with Applications, vol. 37, no. 9, pp. 6201–6212, Sep. 2010

[4] B. Smith and G. Linden, "Two Decades of Recommender Systems at Amazon.com," IEEE Internet Computing, vol. 21, no. 3, pp. 12–18, May 2017.

[5] T. Li, A. Liu, and C. Huang, "A Similarity Scenario-Based Recommendation Model With Small Disturbances for Unknown Items in Social Networks," IEEE Access, vol. 4, pp. 9251–9272, 2016.

[6] T. K. Paradarami, N. D. Bastian, and J. L. Wightman, "A hybrid recommender system using artificial neural networks," Expert Systems with Applications, vol. 83, pp. 300–313, Oct. 2017.

[7] S. Arora and S. Goel, "Improving the Accuracy of Recommender Systems Through Annealing," Lecture Notes in Networks and Systems, pp. 295–304, Nov. 2017.

[8] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[9] H. Koohi and K. Kiani, "User based Collaborative Filtering using fuzzy C-means," Measurement, vol. 91, pp. 134–139, Sep. 2016.

[10] C.-F. Tsai and C. Hung, "Cluster ensembles in collaborative filtering recommendation," Applied Soft Computing, vol. 12, no. 4, pp. 1417–1425, Apr. 2012.

[11] S. Meng, W. Dou, X. Zhang, and J. Chen, "KASR: A Keyword-Aware Service Recommendation Method on MapReduce for Big Data Applications," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 12, pp. 3221–3231, Dec. 2014.

[12] D. Cheng, J. Rao, Y. Guo, C. Jiang, and X. Zhou, "Improving Performance of Heterogeneous MapReduce Clusters with Adaptive Task Tuning," IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 3, pp. 774–786, Mar. 2017.

[13] O.J. Lee, M.S. Hong, J.J. Jung, J. Shin, and P. Kim, "Adaptive Collaborative Filtering Based on Scalable Clustering for Big Recommender Systems," Acta Polytechnica Hungarica, vol. 13, no. 2, Feb. 2016.

[14] K. Kim and H. Ahn, "Recommender systems using cluster-indexing collaborative filtering and social data analytics," International Journal of Production Research, vol. 55, no. 17, pp. 5037–5049, Feb. 2017.

[15] H. Tian and P. Liang, "Personalized Service Recommendation Based on Trust Relationship," Scientific Programming, vol. 2017, pp. 1–8, 2017.

[16] H. Zarzour, F. Maazouzi, M. Soltani and C. Chemam, "An improved collaborative filtering recommendation algorithm for big data," Proceedings of the 6th IFIP International Conference on Computational Intelligence and Its Applications, CIIA'2018, 2018.

[17] F. M. Harper and J. A. Konstan, "The MovieLens Datasets," ACM Transactions on Interactive Intelligent Systems, vol. 5, no. 4, pp. 1–19, Dec. 2015.