International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 6, December 2021, pp. 5541~5548
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i6.pp5541-5548

Recommendation system using the k-nearest neighbors and singular value decomposition algorithms

Badr Hssina[1], Abdelakder Grota[2], Mohammed Erritali[3]
[1]Faculty of Sciences and Technics, LIM Laboratory, Advanced Smart Systems (ASS) Hassan II University of Casablanca, Morocco
[2,3]Faculty of Sciences and Technics, TIAD Laboratory, Computer Sciences Department, University of Sultan My Slimane, Beni-Mellal, Morocco

Article Info

Article history:

Received Nov 6, 2020
Revised May 26, 2021
Accepted Jun 12, 2021

Keywords:

Collaborative filtering
KNN
Matrix factorization items
Recommendation system
SVD

ABSTRACT

Nowadays, recommendation systems are used successfully to provide items (example: movies, music, books, news, images) tailored to user preferences. Amongst the approaches existing to recommend adequate content, we use the collaborative filtering approach of finding the information that satisfies the user by using the reviews of other users. These reviews are stored in matrices that their sizes increase exponentially to predict whether an item is relevant or not. The evaluation shows that these systems provide unsatisfactory recommendations because of what we call the cold start factor. Our objective is to apply a hybrid approach to improve the quality of our recommendation system. The benefit of this approach is the fact that it does not require a new algorithm for calculating the predictions. We are going to apply two algorithms: k-nearest neighbours (KNN) and the matrix factorization algorithm of collaborative filtering which are based on the method of (singular-value-decomposition). Our combined model has a very high precision and the experiments show that our method can achieve better results.

Corresponding Author:

Badr Hssina
Department of Computer Science
LIM Laboratory, Advanced Smart Systems (ASS)
Faculty of Sciences and Technics
Hassan II University of Casablanca
B.P. 146 Mohammedia-Morocco
Email: badr.hssina@fstm.ac.ma

1. INTRODUCTION

Today, the reason people might be interested in using a recommender system is because they have so many items to choose from in a limited period of time and they cannot rate all possible items. In general, to achieve this goal, users must provide their own profile and they will receive a reduced and personalized image of this information. At first glance, they look like news search engines. However, the difference is that search engines allow you to return all items that match the query, ordered by degree of relevance. Whereas, the goal of the recommendation is to return personalized, interesting and useful content to users.

Recommendation systems are a specific form of information filtering aimed at presenting items of information (movies, music, books, news, images, and web pages) that are likely to be of interest to the user. Generally, a recommendation system makes it possible to compare a user's profile with certain benchmark characteristics, and seeks to predict his need [1]. Indeed, with the increase in the number of users on the

Internet and the volume of data produced each day, it has become necessary to design techniques allowing users to access what interests them as quickly as possible. Among the main problems of recommendation systems is the problem of stability with respect to the dynamic profile of the user. This limitation comes from the fact that if the user is interested in several different elements at the same time, as he can alternate his preferences over time, and if his profile is created in the system, it becomes complex to change his preferences and to take into account their different choices [2].

The capacity of recommendation systems remains limited to adapt to the different choices and preferences of users and to follow the evolution of their profiles, as well as to recommend elements that do not correspond to their different choices and interests, which brings us back to a lack of diversity in the list of recommendations [2]. Despite this, recently good recommendation systems with new qualities have been presented in the literature [3]. The quality of a recommendation system can be seen in its effectiveness in providing users with new and diverse articles, which meet their different interests and preferences.

To answer the problem of the stability of recommendation systems and to offer diversified recommendations in relation to the dynamic profile of users and the scalability of the data, we present in this work a recommendation system capable of generating diversified recommendations [2], [3]. These types of systems respond to different user requests and interests, by developing recommendation algorithms allowing users to belong to different groups, with similar interests, nearest neighbors are selected using a new metric of similarity based on the difference in presence between the membership degrees of the user asset and similar members and a matrix factorization method [4].

2. RELATED WORK

Recommendation systems are a specific form of information filtering aimed at presenting items of information (movies, music, books, news, images, and web pages) that are likely to be of interest to the user. Generally, a recommendation system makes it possible to compare a user's profile with certain benchmark characteristics, and seeks to predict his need. The most popular definition of recommendation systems is that of Robin Burke: A system capable of providing personalized recommendations or guiding the user to relevant or useful resources within a large data space [4]. The repository of a recommendation system usually consists of a list of users who have expressed their preferences for various items. As mentioned before, a choice expressed by a user for an item is called a view, and is often represented by a triple (user, rating, and item). These views can take different forms. Moreover, the majority of recommendation systems use binary ratings (like/dislike) or scores in the form of a scale of 1 to 5. The triplets (user, element and rating) form this called the score matrix. The pairs (element, user) for which the user did not give an element score are values ignored in the matrix.

The objective of a referral system can be summed up in two parts. The first part is prediction: given a user and an item, what would be the user's preference for that item, the system must predict the value of the marked notes. The second part is the recommendation; what ordered list (n elements) of recommendations can the system suggest? This is called a Top-n list. It should be mentioned that the list of n recommendations is not necessarily the list of n elements with the most relevant prediction values. A recommendation algorithm can use other criteria, such as context [3], [4] because score prediction is not the only criterion used to produce a list of recommendations [5].

3. PHASES OF RECOMMANDATION PROCESS OF OUR SYSTEM

3.1. Information collection

The two major concepts to take into account in the data collection phase are: data sources and data producers. On the one hand, the concept of data source poses problems linked to the multiplicity and reliability of data sources. On the other hand, the concept of data producer raises several issues, the most important of which is the actual choice of persons from whom the user profile can be derived [4]. Furthermore, for the data preprocessing phase, existing conventional techniques can be implemented: data reduction, data transformation, data transmoding and data transcoding. The data that will be used by the data mining algorithms in the data preparation phase can be ideally structured according to 4 concepts [5]:

- Explicit data (provided explicitly by the user and which can be reused by the mechanisms for using the profile);
- Implicit data (implicitly collected from user interactions) which will make it possible to define indicators allowing user preferences to be deduced;
- Context data for better use according to the contexts of the profiles constructed;
- Semantic data which is based on semantic resources to remove semantic ambiguities on the implicit data used to build the profiles.

3.2. Explicit feedback

In this type of feedback, users are forced to give their opinion on products, objects or items in general. Users can do this via a rating system (for example with 5 stars to validate, a satisfaction questionnaire), or by posting their opinion on a given element (for example the "Like" function on social networks) [6].

3.3. Implicit feedback

The implicit collection, also called passive, concerns the interactions of users on the system. This collection includes the number of views on a video, the tracking of the number of visits to a page, the history of purchases on an e-commerce platform or the time spent on a given section [7].

4. RECOMMANDATION SYSTEM APPROCHES

4.1. Content-based filtering

Content filtering is based on the content of documents (subjects) to compare them to a profile, itself made up of subjects. This type of filtering is a general evolution of studies on information filtering. System users then have a profile that describes their areas of interest. Each user's profile can contain a list of topics or preferences [8], [9]. The system compares the description of a new arrived document with the user's profile to predict the usefulness of that document for that user [10]. Associating documents with a user profile is an advantage of content-based filtering systems.

4.2. Collaborative filtering

The objective of collaborative filtering [11] is to use the evaluations made by users on certain documents (content), in order to recommend these same documents to other users, and without analyzing the content of the documents. Users of a collaborative filtering system can benefit from the results of others by receiving recommendations for which closest users have given a favorable value judgment, without the system extracting the content of the documents. This independence of the system in relation to the representation of the data [11], [12] can be applied in contexts where the content is difficult to analyze, and in particular it can be used for any type of data: text, audio, image and video. Thus, the user is able to discover different interesting areas, because the principle of collaborative filtering is absolutely not based on the thematic dimension of profiles, and is not subject to the "funnel" effect [13]. Collaborative filtering constitutes an important advantage which is that the value judgments of the users integrate not only the thematic dimension but also other factors related to the quality of the documents such as novelty and diversity. Among the problems with this type of filtering is cold start [14]: it is the fact that a user must vote on a large number of items before getting recommendations.

4.2.1. Memory based techniques

The data is represented in the form of a "User x Item" matrix [15] for a collaborative memory-based filtering system. The rows represent the users and the columns constitute the items. The type of memory-based approaches use user feedback on items (in the form of reviews), in order to generate recommendations. This type mainly applies statistical techniques in order to identify neighboring users having, on the same set of elements, ratings similar to those of the active user. Once the neighbors are identified, the memory-based approach uses different algorithms to combine the opinions of the neighbors and generate predictions to the active user [16]. This method mainly uses ranks.

The degree of correlation between the users and the user for whom we wish to make the recommendation determines the weight given to the rating of each user. Since systems typically have to handle a large number of users, then making recommendations based on ratings from millions of users can have severe performance consequences [17], [18]. Furthermore, when the number of users reaches a certain threshold, a selection of the "best" neighbors must be made. Pearson's similarity is based on the calculation of correlations, only user currents are taken into account [19].

$$pearson_{sim}(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \mu_u).(r_{vi} - \mu_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \mu_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \mu_v)^2}}$$

Vector space models are widely adopted in the field of information retrieval, so we will talk about numerical similarity [19]. These approaches use a feature vector, in dimensional space, to represent each object and calculate numerical similarity based on the cosine measure or Pearson's correlation. Among the approaches cited in the literature we can cite: This measurement uses the full vector representation, i.e. the frequency of objects (words).

Two objects (documents) are similar if their vectors are confused. If two objects are not similar, their vectors form an angle (U, V) whose cosine represents the value of the similarity. The formula is defined by the ratio of the dot product of the vectors u and v and the product of the norm of u and v.

$$\text{cosine}_{\text{sim}}(u,v) = \frac{\sum_{i \in I_{uv}} r_{ui}.r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2}.\sqrt{\sum_{i \in I_{uv}} r_{vi}^2}}$$

We generally use the k-nearest neighbor (k-NN) algorithm to determine which are the most relevant neighbors to select and generate reliable recommendations, which allows to select only the k best neighbors with the highest correlation value. Can distinguish two methods of collaborative filtering based on memory: the method based on item-centered memory and the user-centered memory-based method [20].

4.2.2. Model-based techniques

Models have been incorporated into recommendation systems for improve and remedy problems with memory-based methods. Algorithms based on the model are also based on previous evaluations (profiles) of users, but this method does not directly calculate predictions, it classifies users according to groups or learn models from their data. For the construction of the model several methods are used. In general, the methods based on the model use machine learning techniques, such as clustering, matrix factorization, Bayesian networks, and decision trees [19], [20].

In our approach, we will focus mainly on matrix factorization as well called matrix de-composition. It consists in breaking down a matrix into several other matrices. To find the original matrix, it will suffice to make the product of these matrices between them. The Matrix factorization has given good results in recommender systems.

5. HYBRID FILTRING

Noting the advantages and disadvantages of each of the two above approaches, it is understood that many systems are based on their combination, which makes them so-called hybrid filtering systems [21]-[23]. In general, hybridization takes place in two phases as shown in Figure 1:

- Separately apply collaborative filtering and other filtering techniques to generate candidate recommendations.
- Combine these sets of preliminary recommendations using some methods such as weighting, mixing, cascading, and switching, to produce the final recommendations for users [9]. More generally, hybrid systems manage content-oriented user profiles, and the comparison between these profiles results in the formation of user communities that allow collaborative filtering [24].

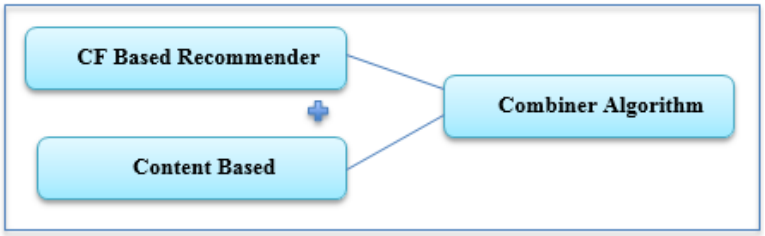


Figure 1. Hybrid filtering techniques

6. EVALUATION METRICS FOR RECOMMENDATION ALGORITHMS

6.1. Dataset

In this work we will analyze the Movielens 100 k Dataset which consists of 100.000 ratings from 1000 users on 1700 movies. All users in this dataset have at least rated 20 movies. Apart from this information, simple demographic information for the users like age, gender, occupation is included. The dataset can be obtained on the following permalink: http://grouplens.org/datasets/movielens/100k/.

We will use a hybrid collaborative filtering approach where we will combine the results of the k nearest neighbor algorithm and the model based SVD algorithm to predict the movie ratings of the users [25], [26]. The advantage of the collaborative filtering algorithms is that no knowledge about item features is needed. So we can ignore the movie tags and the demographic information and concentrate on the users and

their ratings. We will evaluate the hybrid model to see if a combination between a model based (SVD) and a memory-based (KNN) approach delivers better results than each of the approaches on their own.

6.2. Results and evaluation

For the implementation of this project we have used "surprise" a Python scikit for recommender systems. It has predefined all major recommendation algorithms such as KNN, SVD. We created a new hybrid algorithm by combining the results of KNN and SVD. On http://surpriselib.com/ you have access to the surprise library.

Hence, we first run SVD on the training data and get a model. Then we do the same with KNN. With KNN we implemented a user-based collaborative filtering model. To compute the similarity between the K nearest neighbor in the KNN algorithm we used cosine similarity. For both SVD and KNN we get predictions for the movie ratings of each user. The results are combined by averaging the estimated rating of KNN and SVD.

We used 5 cross-fold validation for splitting our data in train and testing sets. As evaluation metrics we used root mean square error, mean absolute error and precision and recall. The precision and recall results of the 5 cross fold validation was averaged for each algorithm (SVD, KNN, combination of SVD and KNN, random prediction).

$$MAE = \frac{1}{|\hat{R}|}\sum_{\hat{r}_{ui}\in\hat{R}}|r_{ui} - \hat{r}_{ui}|$$

Where r_{ui} is the predicted rating for user u on item i, r_{ui} is the actual rating and N is the total number of ratings on the item set. The lower the MAE, the more accurately the recommendation engine predicts user ratings. Also, the root mean square error (RMSE) is given by Cotter *et al.* [27] as;

$$RMSE = \sqrt{\frac{1}{|\hat{R}|}\sum_{\hat{r}_{ui}\in\hat{R}}(r_{ui} - \hat{r}_{ui})^2}$$

Root mean square error (RMSE) puts more emphasis on larger absolute error and the lower the RMSE is, the better the recommendation accuracy. As we can observe in Figures 2 and 3, the SVD model outperforms KNN and the random predictor in all metrics. It has the smallest RMSE, MAE and recall and the highest precision. The KNN model is nearly as good as SVD. SVD is just 3.95 % better in RMSE, 3.99% better in MAE. Furthermore, SVD has a 3.94% higher precision and a 5.69 % better recall rate. Of course, both, KNN and SVD, are much better than the random prediction model.

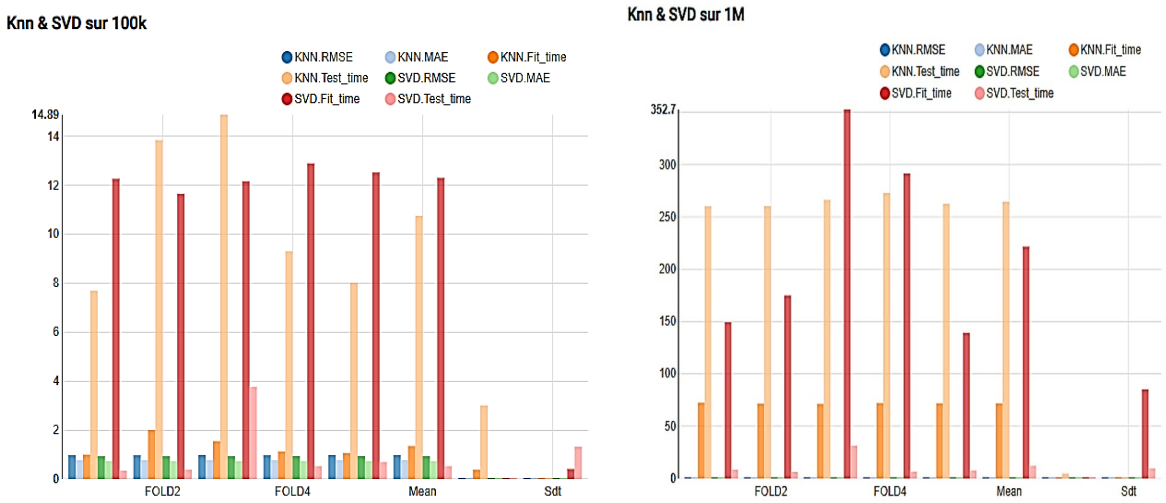


Figure 2. Evaluation of RMSE, MAE on 100k and on 1M of the two algorithms KNN and SVD

KNN for example has a 37.28% smaller MAE and a 36.14% smaller RMSE than the random predictor, which is enormous. It should also be noted, that the difference for MAE and the difference for RMSE between the models is almost the same. For example, SVD is around 4% better in RMSE (the exact value is 3.95% as stated before) as well as in MAE (3.99%) than KNN. And KNN is around 36% better in

RMSE and MAE than the random predictor. This closeness between RMSE and MAE may indicate, that these metrics are very similar and that one does not get any additional information by applying both metrics.

Now let us compare our combined model with the other models. Since SVD is the best of the single models, it is sufficient to just compare SVD with the combined model. In regard to RMSE, the combined approach is only 0.245% better than the SVD model. For MAE however, it's the opposite, here the SVD model is 0.256% better than the combined approach. Regarding precision SVD has a 0.126% higher precision that the combined model. Also the recall rate of the SVD algorithm is 0.7% higher than that of the combined algorithm.

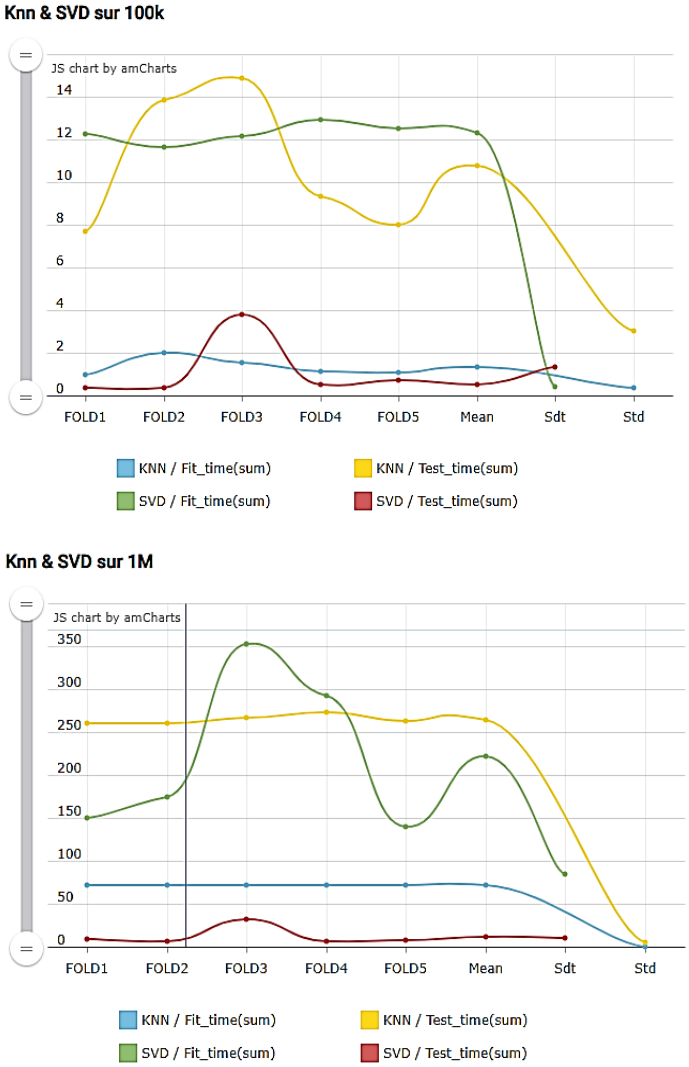


Figure 3. Evaluation of fit-time test-time on 100k and on 1M of the two algorithms KNN and SVD

7. CONCLUSION

As a conclusion, our combined model has very high precision and experiments show better results. This means that most of the rec-ommended items are relevant. Though, the model has also a relatively low recall, which means that the proportion of relevant items that are recommended is very small. The same applies for SVD and KNN. The results have shown, that the com-bined model, where we averaged the estimated ratings of the KNN and SVD model, is not significantly better than for example the SVD model alone. In fact, we can ob-serve from the results, that the SVD model performs much better than the KNN model on the 100k movielens dataset, such that, if we combine the models, the result for the combined model is in most metrics (MAE, precision and recall) slightly worse than for the SVD algorithm. Hence, the combination of the SVD and KNN model is not worth the effort and we would do better if we just used the

SVD algorithm. As a mod-el-based approach it is much faster than the KNN approach, because we have only to generate the model the first time and then can use this for new data points. This approach potentially offers the benefits of both speed and scalability.

REFERENCES

[1] A. J. Konstan and J. Riedl, "Recommender systems: from algorithms to user experience," *User Modeling and User-Adapted Interaction*, vol. 22, pp. 101-123, 2012.

[2] C. Pan and W. Li, "Research paper recommendation with topic analysis," *2010 International Conference On Computer Design and Applications,* vol. 4, pp. V4-225, 2010, doi: 10.1109/ICCDA.2010.5541170.

[3] P. Pu, L. Chen, and R. Hu, "A user-centric evaluation framework for recommender systems," *Proceedings of the fifth ACM conference on Recommender Systems (RecSys'11),* ACM, New York, NY, USA, 2011, pp. 57-164.

[4] R. Hu and P. Pu "Potential acceptance issues of personality-ASED recommender systems," *Proceedings of ACM conference on recommender systems (RecSys'09),* New York City, NY, USA; Oct. 2009, pp. 22-25.

[5] B. Pathak, R. Garfinkel, R. Gopal, R. Venkatesan, and F. Yin, "Empirical analysis of the impact of recommender systems on sale," *Journal of Management Information Systems*, vol. 27, no. 2, pp. 159-188, 2010, doi: 10.2307/29780174.

[6] M. A. Rashid *et al.,* "Getting to know you: learning new user preferences in recommender systems," *Proceedings of the international conference on intelligent user interfaces,* 2002. pp. 127-134, doi: 10.1145/502716.502737.

[7] B. J. Schafer, J. Konstan, and J. Riedl, "Recommender system in ecommerce," *Proceedings of the 1st ACM conference on electronic commerce,* 1999, pp. 158-166, 10.1145/336992.337035.

[8] P. Resnick, and R. H. Varian, "Recommender system's," *Commun ACM,* vol. 40, no. 3, pp. 56-58, 1997, doi: 10.1145/245108.24512.

[9] M. A. Acilar and A. Arslan, "A collaborative filtering method based on Artificial Immune Network," *Expert Systems with Applications,* vol. 36, no. 4, pp. 3024-3032, 2009, doi: 10.1016/j.eswa.2008.10.029.

[10] S. L. Chen, H. F. Hsu, C. M. Chen and C. Y. Hsu, "Developing recommender systems with the consideration of product profitability for sellers," *Information Sciences*, vol. 178, no. 4, pp. 1032-1048, 2008, doi: 10.1016/j.ins.2007.09.027.

[11] M. Jalali, N. Mustapha, M. Sulaiman and A. Mamay, "WEBPUM: a web-based recommendation system to predict user future movement," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6201-6212, 2010, doi: 10.1016/j.eswa.2010.02.105.

[12] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender system. A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, 2005, doi: 10.1109/TKDE.2005.99.

[13] N. C. Ziegler, M. S. McNee, A. J. Konstan and G. Lausen, "Improving recommendation lists through topic diversification," *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 22-32, doi: 10.1145/1060745.1060754.

[14] H. S. Min and I. Han, "Detection of the customer time-variant pattern for improving recommender system," *Expert Systems with Applications,* vol. 28, no. 2, pp. 189-199, 2005, doi: 10.1016/j.eswa.2004.10.001.

[15] K. Yu, A. Schwaighofer, V. Tresp, X. Xu and P. H. Kriegel, "Probabilistic memory-based collaborative filtering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 1, pp. 56-69, 2004, doi: 10.1109/TKDE.2004.1264822.

[16] H. Lieberman, "Letizia: an agent that assists web browsing," *Proceedings of the 14th international joint conference on Artificial intelligence,* Montreal, Canada, vol. 1, 1995, pp. 924-9.

[17] J. M. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5, pp. 393-408, 1999, doi: 10.1023/A:1006544522159.

[18] A. Jennings and H. Higuchi, "A personal news service based on a user model neural network," *IEICE Trans Inform Syst*, vol. E75, no. D2, pp. 198-209, 1992.

[19] G. Murat and O. G. Sule, "Combination of web page recommender systems," *Expert Systems with Applications*, vol. 37, no. 4, pp. 2911-1922, 2010, doi: 10.1016/j.eswa.2009.09.046.

[20] B. Mobasher, "Recommender systems. Kunstliche Intelligenz," *Special Issue on Web Mining, BottcherIT Verlag*, Bremen, Germany, vol. 3, pp. 41-43, 2007.

[21] G. M. Vozalis and G. K. Margaritis, "Applying SVD on item-based filtering," *5th International Conference on Intelligent Systems Design and Applications (ISDA'05),* 2005, pp. 464-469, doi: 10.1109/ISDA.2005.25.

[22] X. Guan, T. C. Li and Y. Guan, "Matrix factorization with rating completion: An enhanced SVD model for collaborative filtering recommender systems," *IEEE access*, vol. 5, pp. 27668-27678, 2017, doi: 10.1109/ACCESS.2017.2772226.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," Minnesota Univ. Minneapolis Dept of Computer Science, 2000.

[24] D. Mican and N. Tomai, "Association ruled-based recommender system for personalization in adaptive web-based applications," *International Conference on Web Engineering (ICWE 2010),* vol. 6385, 2010, pp. 85-90, doi: 10.1007/978-3-642-16985-4_8.

[25] X. Zhou, J. He, G. Huang, and Y. Zhang, "SVD-based incremental approaches for recommender systems," *Journal of Computer and System Sciences*, vol. 81, no. 4, pp. 717-733, 2015, doi: 10.1016/j.jcss.2014.11.016.

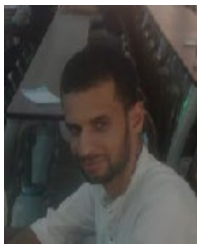
[26] Q. Ba, X. Li, and Z. Bai, "Clustering collaborative filtering recommendation system based on SVD algorithm," *2013 IEEE 4th International Conference on Software Engineering and Service Science*, 2013, pp. 963-967, doi: 10.1109/ICSESS.2013.6615466.

[27] H. Y. M. Al-Shamri, and K. K. Bharadway, "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1386-1399, 2008, doi: 10.1016/j.eswa.2007.08.016.

BIOGRAPHIES OF AUTHORS

Badr Hssina obtained a master's degree in business intelligence from the faculty of science and Techniques, Beni Mellal at Morocco in 2011 and a Ph.D degree in Computer Sciences from the same faculty of Sultan Moulay Slimane University, Morocco in 2017. His current interests include developing specification and design techniques for use within E-learning, data mining, information Retrieval system, semantic web and cryptography. He is currently a professor at the Faculty of Science and Techniques, Mohammedia, University Hassan II of Casablanca, and also a member of the LIM laboratory, and the team Advanced Smart Systems (ASS).

Abdelakder Grota obtained a master's degree in business intelligence from the faculty of science and Techniques, Beni Mellal at Morocco in 2020, and Ph.D student in the TIAD laboratory of sultan moulay slimane University.



Mohammed Erritali obtained a master's degree in business intelligence from the faculty of science and Techniques, Beni Mellal at Morocco in 2010 and a Ph.D. degree in Computer Sciences from the faculty of sciences, Mohamed V Agdal University, Rabat, Morocco in 2013. His current interests include developing specification and design techniques for use within Intelligent Network, data mining, information Retrieval, image processing and cryptography. He is currently a professor at the Faculty of Science and Techniques, University Sultan Moulay Slimane, and also a member of the TIAD laboratory.