Building Recommendation Systems Using the Algorithms KNN and SVD

https://doi.org/10.3991/ijes.v9i1.20569

Badr Hssina (✉)
Advanced Smart Systems (ASS) Hassan II University of Casablanca,
Casablanca Morocco
badr.hssina@fstm.ac.ma

Abdelakder Grota, Mohammed Erritali
University of Sultan My Slimane, Beni-Mellal, Morocco

Abstract—Recommendation systems are used successfully to provide items (example: Movies, music, books, news, images) tailored to user preferences.

Among the approaches proposed, we use the collaborative filtering approach of finding the information that satisfies the user by using the reviews of other users. These ratings are stored in matrices that their sizes increase exponentially to predict whether an item is interesting or not. The problem is that these systems overlook that an assessment may have been influenced by other factors which we call the cold start factor.

Our objective is to apply a hybrid approach of recommendation systems to improve the quality of the recommendation. The advantage of this approach is the fact that it does not require a new algorithm for calculating the predictions. We are going to apply the two K-closest neighbor algorithms and the matrix factorization algorithm of collaborative filtering which are based on the method of (singular value decomposition).

Keywords—Collaborative filtering, Matrix factorization items, Recommendation system, SVD

1 Introduction

With the increasing amount of information and the number of users on the Internet today, it has become essential to design mechanisms that allow users to access to what interests them as quickly as possible. From there, recommendation systems were born including their goals: process a large amount of feedback, minimize user time spent to research and suggest relevant resources [1].

One of the major problems with recommender systems is the problem of the stability of these systems against the dynamic profile of the user [2]. This problem comes down to 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 difficult to change his preferences and take into account their different choices and preferences.

This limits the capacity recommendation systems to follow the evolution of the user profile and adapt to their different choices and preferences, and then recommend items that do not correspond not to his different choices and interests, which leads to a lack of diversity in the lists of recommendation [1] [2]. However, in recent years, new qualities of a good recommender system have been presented in the literature. An effective referral system must offer new and diverse items to users, which meet their different interests and preferences [3].

In order to find a solution to the problem of the stability of recommendation systems by report to the dynamic user profile and to offer diversified recommendations to users, while considering the effects of data scalability, we present in this thesis a recommendation system capable of generating diversified recommendations that meet the various choices and interests of the user, by developing recommendation algorithms allows users to belong to different groups, with similar interests, neighbors the closest are selected using a new measure of similarity based on the difference presence between the membership degrees of the active user and members of similar groups and a matrix factorization method[2][3].

2 Related Work

The goal of a recommendation system is to provide the user with relevant objects according to his preferences. It drastically reduces the time spent by the user to find the objects that are most interesting to him, and also to find objects that he is likely to like but that he would not necessarily have paid attention to.

Recommendation systems have been defined in several ways. The most popular and general definition that we quote here is that of Robin Burke [4] that we have translated as: Recommendation system: System capable of providing recommendations personalized or to guide the user to interesting or useful resources within a large data space. The information domain for a recommendation system in general consists of a list of users who have expressed their preferences for various items. As we have seen previously, a preference expressed by a user for an item is called a note, and is often represented by a triple (user, item, and note). These notes can take different forms. However, the majority of systems use scores in the form of a scale of 1 to 5, or binary notes (I like / I don't like). The set of triples (user, item, and note) forms what is called the score matrix. The pairs (user, item) where the user does not have given item scores are unknown values in the matrix.

A recommendation system focuses on two tasks. The first task is the prediction: given a user and an item, what would be the user's preference for that item? In In other words, the system must predict the value of the marked notes? The second task is the recommendation: given a user, which ordered list what n recommendations can we suggest? This is called a Top-n list. Note that the list of the Top-n recommendations is not necessarily the list of the n items with the highest values prediction. The prediction of scores is not the only criterion used to produce a list of recommenda-

tions. Indeed, a recommendation algorithm can use other criteria, such as the context [3] [4] [5].

3 Phases of Recommendation Process

3.1 Information collection phase

In the data collection phase, two major concepts must be taken into account: data producers and data sources [4]. The concept of data producer raises several issues, the most important of which is the effective choice of individuals from whom the user profile can be derived. The concept of data source raises issues linked to the multiplicity and reliability of data sources. In the data preprocessing phase, existing conventional techniques can be used: data reduction, data transformation, data transcoding, transmoding data, etc.

In the data preparation phase, the data that will be used by the Data mining algorithms can be ideally structured according to four concepts: explicit data (provided explicitly by the user and which can be reused as is by the profile usage mechanisms); implicit data (collected implicitly at from user interactions) which will make it possible to define indicators allowing infer the interests of the user; the context data which will allow a better use in depending on the contexts of the profiles constructed; the semantic data that will make it possible to raise the semantic ambiguities on the implicit data used to construct the profiles [5] .

3.2 Explicit feedback

Explicit (active) collection: In this case, users are asked to provide their opinions on products / objects. They can do this via a rating system (e.g. a 5 star grid, a satisfaction questionnaire), or even by posting their opinions on a given element (e.g. the “Like” function on the social network Facebook) [6].

3.3 Implicit feedback

Implicit (passive) collection: Implicit collection is concerned with user interactions on the system. Examples of this collection include monitoring the number of visits on a page, the number of views on a video, the time spent on a given section or the history of purchases on an e-commerce platform [7].

4 Recommendation Filtering Techniques

4.1 Content-based filtering

Content-based Filtering, which is a general evolution of studies on information filtering, relies on the content of documents (topics covered) to compare them to a pro-

file itself made up of topics. Each user of the system then has a profile that describes areas of interest. For example, the profile may contain a list of topics or preferences that the user likes or dislikes. When a new document arrives, [8] [9] the system compares the document description with the user's profile to predict the usefulness of this document for that user.

The advantage of cognitive filtering systems, based on content, is that they allow documents to be associated with a user profile [10].

4.2 Collaborative filtering

The principle of collaborative filtering (Collaborative Filtering "CF") is to use the evaluations made by users on certain documents (content), in order to recommend these same documents to other users, and without the need to analyse the contents of the documents [11].

All users of the collaborative filtering system can benefit from the evaluations of others by receiving recommendations for which the closest users have issued a favourable value judgment, without the system having an extraction process content of documents. Thanks to its independence vis-`a-vis the representation of data, [11] [12] this technique can be applied in contexts where the content is either unavailable or difficult to analyse, and in particular it can be used for any type of data: text, image, audio and video. In addition, the user is able to discover various interesting areas, because the principle collaborative filtering is absolutely not based on the thematic dimension of profiles, and is not subject to the "funnel" effect [13].

Another advantage of collaborative filtering is that users' value judgments incorporate not only the thematic dimension but also other factors relating to the quality of documents such as diversity, novelty, etc.

The CF suffers from several big problems. The main problem being the cold start: it is the fact that a user must vote on a lot of items before getting recommendations [14].

Memory based techniques: In a memory-based collaborative filtering system, data is represented as in the form of a "User x Item" matrix where the rows represent the U users and the columns constitute the items I [5].

The memory-based approach exploits user feedback on items (under form of notes for example), in order to generate the predictions. This approach mainly applies statistical techniques in order to identify neighbouring users having, on a same set of items, similar ratings to those of the active user. Once the neighbours are identified, the memory-based approach uses different algorithms to combine the neighbour reviews and generate recommendations to the active user. This method uses then mainly ratings (for example: films) [16].

The weight given to the rating of each user is determined by the degree of correlation between this user and the user for whom we wish to make the recommendation. Systems must usually also handle a large number of users. Make recommendations based on Ratings from millions of users can have serious performance implications [16] [17]. Thus, when the number of users reaches a certain threshold, a selection of the "best" neighbours must be done. Pearson's similarity measure is based on the

correlation calculation. Only users currents (or articles) are taken into account. Pearson's correlation coefficient can be seen as a cantered mean cosine similarity and is defined as [19].

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

In the field of information retrieval, vector space models are widely adopted, so we will speak of numerical similarity. These approaches [19] 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.

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

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

Model-based techniques: Model-based methods 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, decision trees, etc [19][20].

In our work, we will focus mainly on matrix factorization as well called matrix decomposition. 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 Filtering

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.

In general, hybridization takes place in two phases:

- 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, switching, etc. 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 [22]

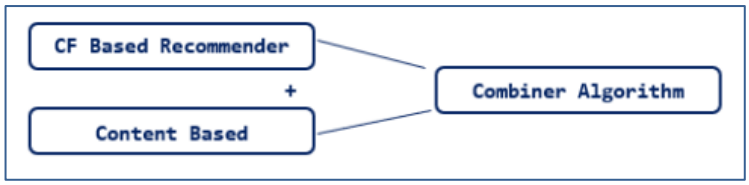

Fig. 1. Hybrid filtering

6 Evaluation Metrics for Recommendation Algorithms

6.1 Dataset

In this project we will analyse the Movielens 100k 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 Neighbour algorithm and the model based SVD algorithm to predict the movie ratings of the users. 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 neighbour 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. [21] 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.

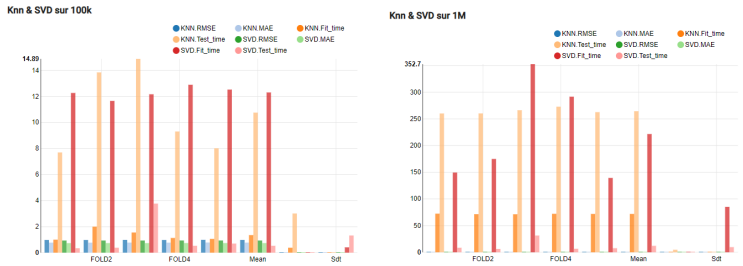

Fig. 2. Evaluation of RMSE, MAE on 100k and on 1M of the two algorithms KNN and SVD

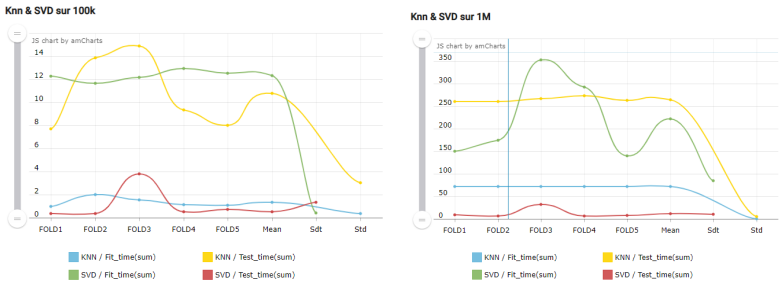

Fig. 3. Evaluation of Fit-tim Test-time on 100k and on 1M of the two algorithms KNN and SVD.

As we can observe, 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.

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.

7 Conclusion

Our combined model has a very high precision. This means that most of the recommended items are relevant. However, 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 combined 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 observe 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 model-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. For KNN however, for every new data point we have to run the whole algorithm again.

8 References

[1] Konstan JA, Riedl J. Recommender systems: from algorithms to user experience. User Model User-Adapt Interact 2012; 22:101–23. https://doi.org/10.1007/s11257-011-9112-x

[2] Pan C, Li W. Research paper recommendation with topic analysis. In Computer Design and Applications IEEE 2010;4, pp. V4-225.

[3] Pu P, Chen L, Hu R. A user-centric evaluation framework for recommender systems. In: Proceedings of the fifth ACM conference on Recommender Systems (RecSys'11), ACM, New York, NY, USA; 2011. p. 57–164. https://doi.org/10.1145/2043932.2043962

[4] Hu R, Pu P. Potential acceptance issues of personality-ASED recommender systems. In: Proceedings of ACM conference on recommender systems (RecSys'09), New York City, NY, USA; October 2009. p. 22–5. https://doi.org/10.1145/1639714.1639753

[5] Pathak B, Garfinkel R, Gopal R, Venkatesan R, Yin F. Empirical analysis of the impact of recommender systems on sales. J Manage Inform Syst2010;27(2):159–88.

[6] Rashid AM, Albert I, Cosley D, Lam SK, McNee SM, Konstan JA et al. Getting to know you: learning new user preferences in recommender systems. In: Proceedings of the international conference on intelligent user interfaces; 2002. p. 127–34. https://doi.org/10.1145/502716.502737

[7] Schafer JB, Konstan J, Riedl J. Recommender system in ecommerce. In: Proceedings of the 1st ACM conference on electronic commerce; 1999. p.158–66.

[8] Resnick P, Varian HR. Recommender system's. Commun ACM 1997;40(3):56–8. http://dx.doi.org/10.1145/245108.24512.

[9] Acilar AM, Arslan A. A collaborative filtering method based on Artificial Immune Network. Exp Syst Appl 2009;36(4):3024–32.

[10] Chen LS, Hsu FH, Chen MC, Hsu YC. Developing recommender systems with the consideration of product profitability for sellers. Int J Inform Sci 2008;178(4):1032–48. https://doi.org/10.1016/j.ins.2007.09.027

[11] Jalali M, Mustapha N, Sulaiman M, Mamay A. WEBPUM: a web-based recommendation system to predict user future movement. Exp Syst Applicat 2010;37(9):6201–12. https://doi.org/10.1016/j.eswa.2010.02.105

[12] Adomavicius G, Tuzhilin A. Toward the next generation of recommender system. A survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 2005;17(6):734–49. https://doi.org/10.1109/tkde.2005.99

[13] Ziegler CN, McNee SM, Konstan JA, Lausen G. Improving recommendation lists through topic diversification. In: Proceedings of the 14th international conference on World Wide Web; 2005. p. 22–32. https://doi.org/10.1145/1060745.1060754

[14] Min SH, Han I. Detection of the customer time-variant pattern for improving recommender system. Exp Syst Applicat 2010;37(4):2911–22.

[15] Celma O, Serra X. FOAFing the Music: bridging the semantic gap in music recommendation. Web Semant: Sci Serv Agents World Wide Web 2008;16(4):250–6. https://doi.org/10.1016/j.websem.2008.09.004

[16] Lieberman H. Letizia: an agent that assists web browsing. In: Proceedings of the 1995 international joint conference on artificial intelligence. Montreal, Canada; 1995. p. 924–9.

[17] Pazzani MJ. A framework for collaborative, content-based and demographic filtering. Artific Intell Rev 1999; 13:393–408, No. 5(6).

[18] Jennings A, Higuchi H. A personal news service based on a user model neural network. IEICE Trans Inform Syst 1992; E75- D (2):198–209.

[19] Murat G, Sule GO. Combination of web page recommender systems. Exp Syst Applicat 2010;37(4):2911–22.

[20] Mobasher B. Recommender systems. Kunstliche Intelligenz. Special Issue on Web Mining, BottcherIT Verlag, Bremen, Germany, vol. 3; 2007. p. 41–3.
[21] Al-Shamri MYH, Bharadway KK. Fuzzy-genetic approach to recommender systems based on a novel hybrid user model. Expert Syst Appl 2008;35(3):1386–99. https://doi.org/10.1016/j.eswa.2007.08.016
[22] Mican D, Tomai N. Association ruled-based recommender system for personalization in adaptive web-based applications.

9 Authors

Badr Hssina works at the Faculty of Sciences and Technics in LIM laboratory Advanced Smart Systems (ASS) Hassan II University of Casablanca in Morocco. badr.hssina@fstm.ac.ma

Abdelakder Grota and **Mohammed Erritali** work as Faculty of Sciences and Technics in TIAD laboratory at the department of Computer Sciences in University of Sultan My Slimane, Beni-Mellal, Morocco.

Article submitted 2020-12-16. Resubmitted 2021-02-06. Final acceptance 2021-02-08. Final version published as submitted by the authors