Acta Scientiarum

http://periodicos.uem.br/ojs
ISSN on-line: 1807-8664
Doi: 10.4025/actascitechnol.v44i1.58925

COMPUTER SCIENCE

A hybrid approach towards movie recommendation system with collaborative filtering and association rule mining

Wisam Alnadem Mahmood [1*], LaythKamil Almajmaie [2], Ahmed Raad Raheem[3] and Saad Albawi[4]

[1]Department of Computer Science, University of Technology-Iraq, Al-Wehda neighborhood, 19006, Baghdad, Iraq. [2]Department of Computer Engineering, University of Technology-Iraq, Baghdad, Iraq. [3]Department of Computer Science, Directorate General of Education Diyala, Ba'qubah, Iraq. [4]Department of Computer Engineering, University of Diyala, Ba'qubah, Iraq. *Author for correspondence. E-mail: 10860@uotechnology.edu.iq

ABSTRACT. There is a huge information stockpile available on the internet. But the available information still throws a stiff challenge to users while selecting the needed information. Such an issue can be solved by applying information filtering for locating the required information through a Recommender System. While using a RS, the users find it easy to curate and collect relevant information out of massive databanks. Though various types of RS are currently available, yet the RS developed by Collaborative Filtering techniques has proven to be the most suitable for many problems. Among the various Recommended Systems available, movie recommendation system is the most widely used one. In this system, the recommendations will be made based on the similarities in the characteristics as exhibited by users / items. The movie recommendation system contains a huge list of user objects and item objects. This paper combines Collaborative Filtering Technique with association rules mining for better compatibility and assurance while delivering better recommendations. Hence, in the process, the produced recommendations can be considered as strong recommendations. The hybridization involving both collaborative filtering and association rules mining can provide strong, high-quality recommendations, even when enough data is unavailable. This article combines various recommendations for creating a movie recommendation system by using common filtering techniques and data mining techniques.

Keywords: Collaborative filtering; association rule mining; recommendation systems; movies.

Received on April 27, 2021.
Accepted on October 26, 2021.

Introduction

Recommendation systems represent information filtering systems having the inherent capabilities in making effective recommendation of entities/items to users by creating highly smart agent systems (Shridhar & Parmar, 2017). Additionally, the user may receive a wide array of recommended areas like a movie, book, and news (Shridhar & Parmar, 2017; Linden, Smith, & York, 2003). Assuming a set of users as $U = \{u_1, u_2, ..., u_n\}$, while $M = \{m_1, m_2, ..., m_m\}$ represent a movie list, it entitles each user to receive a unique set of recommendation in the form of n-dimensional vector having ordered pairs as follows:

$$u^{(n)} = \langle (m_1, s_u(m_1)), (m_2, s_u(m_2)), ..., (m_n, s_n(m_n)) \rangle$$

Wherein, for a single user i.e. u, S_u (m_j) is considered as an estimate rating function for movie m_j, while $m_j \in M$ stands for the relative value of a movie in the movie set i.e. M in case of that specific user (Ting, Liaw, Wang, & Hong, 2017; Xu, Karaleise, & Li, 2014). In a typical RS with a p x q matrix $UP = [s_{u_k}(m_j)]_{p \times q}$ includes s_{u_k} (m_j) as representing user u_k's interest level in relation to a specific movie (m_j) (Wang, Ma, Liao, & Du, 2017; Hence, in a recommender engine, the $REC: P(UP)xU \rightarrow P(M)$ is mapped as per a specific user profile, after which the respective users manage the mapping task on the movie list (Wang et al., 2017). In general, the recommendation system can be defined as follows:

$$REC(u_p, u_k) = \{\arg\max_{i_j \in I} s_{u_k}(m_j)\}$$

Wherein, user profiles set (UP) represents its subset as u_p that predicts the preference value of movie m_j is represented by s_{u_k} (Zuo, Gong, Zeng, Ma, & Jiao, 2015; Koupaei & Khayyambashi, 2015). A typical system usually represents TOP-N recommendations (Lee, Kim, & Rhee, 2013). Figure 1 provides a specimen of a User-Item rating matrix represented on a 1 – 5 scale.

$$
\begin{array}{c|ccccc|}
 & m_1 & m_2 & m_3 & m_4 & m_5 \\
u_1 & 3 & 0 & 2 & 5 & 0 \\
u_2 & 3 & 0 & 4 & 1 & 0 \\
u_3 & 3 & 2 & 3 & 0 & 5 \\
u_4 & 0 & 3 & 0 & 2 & 1 \\
u_5 & 3 & 0 & 2 & 0 & 2
\end{array}
$$

Figure 1. An example user-movie matrix.

Such a matrix shows an entry specifically relevant to user u_1, while designating item m_1 as 3, effectively meaning a rating of 3 given by user u_1 to item m_1 (Liangxing & Aihua, 2010). Similarly, when an entry related to u_1 to m_2 is assigned 0, which, in other words, indicates lack of interest by user u_1 towards item m_2.

The goal of the recommender system's early phases is to identify the most suitable ratings for the things yet to be seen (Banati & Mehta, 2010). The best approach to do this is to anticipate the rating values for new goods that have yet to be viewed by the target users. The prediction algorithm can generate a list of likely items sorted by predicted ratings in descending order. The referral system's next phase is to give better-executed suggestions to users (Salter & Antonopoulos, 2006). Figure 2 depicts the movie recommendation engine at a high-level approach, representing couple of methods for collecting User-Item rating matrix entries (Shaw, Xu, & Geva, 2009; Desrosiers & Karypis, 2011).

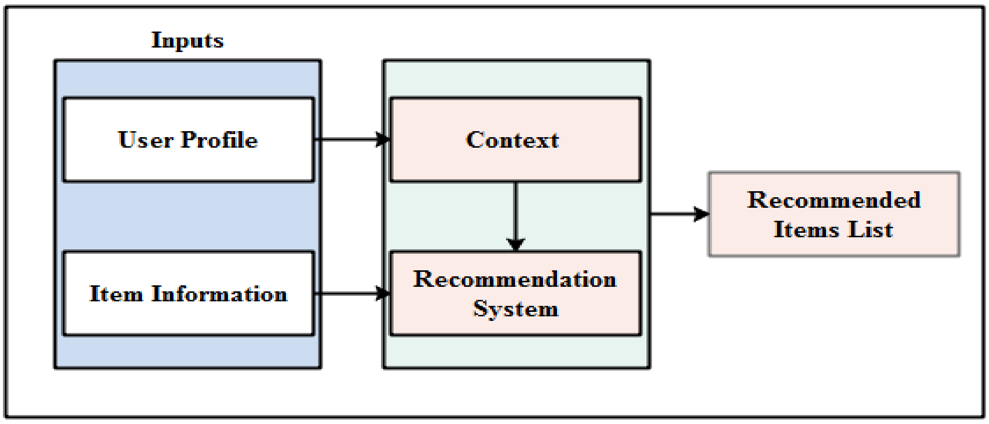

Figure 2. A simple recommendation system.

The explicit method considers the user's ratings over any item. However, it isn't really a useful methodology since not all users usually are inclined to rate the items, whether they see it or miss. In case the user provides a rating, still, the users utilizing such a facility could be limited. As a result, the User-Item rating matrix becomes a sparse matrix, making decision-making harder. The implicit method of considering items in the User-Item rating matrix is the second method. The ratings are typically based on the behaviour of the users. However, several of the suggestions, in general, use data explicitly as well as in implicit manners (Felfemig, Friedrich, Jannach, & Zanker, 2011; Salter & Antonopoulos, 2006). A user's profile is a collection of items that he or she frequently needs or procures for use. The suggestions to target users in collaborative filtering are dependent on real time feedback that any users is seen sharing similar to the taste of the targeted user (Covington, Adams, & Sargin, 2016).

Collaborative filtering

Methods of collaborative filtering make suggestions depending on user evaluations, wherein typical feedback on items is provided by users. The single-user classifications featuring attribute-centric definitions have application in techniques having their basis in content. Approaches based on knowledge do not require a historical evaluation and just need a clear explanation of the needs of the users to create listing of suggestions (Desrosiers & Karypis, 2011; Koren & Bell, 2011). Because knowledge-based systems do not require credentials, they may be more successful at tackling cold-start difficulties than content-based or collaborative systems. In the light of the issue of permanent personalisation, historical data, on the other hand, has weak strength than the systems based on content and collaboration. The model may achieve the same output if another user inserts the similar requirements and data into a dynamic interface based on knowledge. Such models seem to have rather limited scope, particularly given the fact that data from several sources appear in public at the same time (Ting et al., 2017; Xu et al., 2014). Predicated on ratings, collaborative filtering links identical community members to a target user. If two individuals' profiles contain the same or nearly identical things, they will be considered to have similar tastes. These users can be grouped together and handled as a unit. In essence, collaborative filtering algorithms may be implemented in two ways: (1) Methods based on user input (2) Methods based on items.

User-based method

These methods identify those users whose profiles are like the profile of the target user and then do the recommendations. The user-based filtering is like the nearest neighbour method. This technique finds the future preferences of the target users based on the nearest neighbours. It locates users with a taste much like the target user (Adomavicius & Tuzhilin, 2005). In this method, the first step is to collect User-Item rating matrix from user profiles, which is shown in Figure 1. Next, the nearest neighbours for the target user are identified, for which the similarity between the target users and all other users is calculated and the users with higher similarity values are chosen as the closest neighbours. The most widely used similarity measure is the Pearson correlation coefficient method. User-based methods use the entire database to find suggestions (Ting et al., 2017). This approach has been very popular in the past, but now there are other alternatives that offer better recommendations. However, it mainly suffers from a drawback while dealing with a sparse User-Item rating matrix and mainly results in poor quality of recommendations.

Item based methods

In an item-based method, a collaborative filtering technique, a set of user-rated items in the target audience are analysed, before the level of similarity is calculated (Zuo et al., 2015). After calculating the similarity of all items evaluated by the target user, *k* similar items are selected for prediction.

Item similarity

This is the most important step which is followed in the item-based collaborative technique. When similarity is calculated, it strongly influences the quality of the recommendation. There are different methods for calculating item similarity.

-Cosine-based similarity
-Co-relation based similarity
-Adjusted cosine-based similarity
-Adaptive Similarity Measure Model

Item prediction

After calculating the similarity between the two items, while using Equation (1), the most suitable prediction for an item I by user u is represented as given below;

$$P(u,i)=\frac{\sum_{j\in N_i} sim(i,j)*R_{u,j}}{\sum_{j\in N_i} |\, sim(i,j)\, |} \tag{1}$$

Wherein $sim(i,j)$ denotes measures defining similarities that *i* and *j* share. $R_{u,j}$ indicates movie rating *j* that user *u* provides. P(u,i) gives target user a movie prediction and N_i is a set of target user who rate a movie i.

Literature review

A user profile comprises the user's item preferences in a content-based recommendation system. The content of all rated products may be used to create a user profile [5]. This profile is created specifically from the content (keyword) that has been analysed using the procedures described in the item profile section. The relevance of keyword Ki to the user is shown by the weight of each item in the user's profile (Good et al., 1999). The Rocchio algorithm, Bayesian classifier, Winnow algorithm, and cosine similarity measure may all be used to calculate this weight using the average approach (Good et al., 1999). Goldberg, Nichols, Oki, and Terry (1992) coined the phrase "[...] collaborative filtering techniques," claiming that "[...] information filtering can be more successful when people are involved in the filtering process". Resnick, Iacovou, Suchak, Bergstrom, and Riedl (1994). proposed the notion of collaborative filtering two years later. People enjoy what like-minded users like, according to their idea, and two users were deemed like-minded if they rated products similarly. Items that were evaluated positively by one user were suggested to the other, and vice versa (Resnick et al., 1994).

Proposed recommender system

Figure 3 illustrates the framework of our proposed reference model.

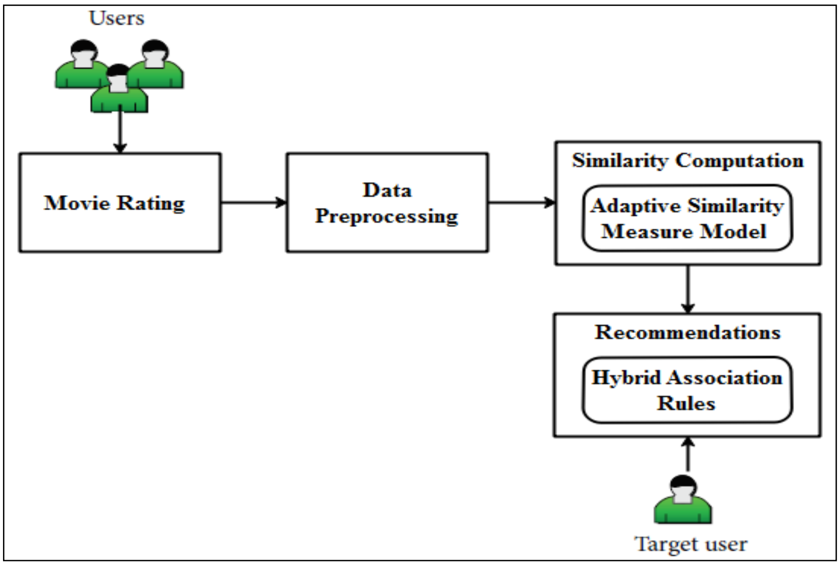

Figure 3. Framework of recommendation model based on collaborative filtering.

In this model, we use a new similarity calculation method that takes more factors into account while calculating the similarities that the targeted user share with others, followed by selected movies collected by identical users (Xu et al., 2014). Finally, we compile a list of suggestions of such movies that is sent to the target user as recommendation. Such frameworks follow many stages.

Data pre-processing

Data pre-processing is an important step in the recommendation system model. Here, the User-Item rating matrix is constructed as per a user profile. But most of the entries are left empty as most of the users may have presumably viewed very few items. So, all the missing values are filled with 0s that indicates that users have not yet experienced the item (Covington et al., 2016; Lika, Kolomvatsos, & Hadjiefthymiades, 2014).

Similarity computation

This step is the basic procedure. Suppose a recommendation model comprising *m* users as well as *n* movies with some movies have already been rated by every user. A bipartite network may describe the relation that users have with movies. When users are defined as $U=\{u_1,u_2,....,u_m\}$ and set of movies as, then the recommendation model is represented as $m \times n$ adjacency matrix $A=\{\delta_{ij}\}$, wherein, if a user *i* selects movie *j*, or in other cases, $\delta_{ij} = 0$. In the next step, collaborative filtering is used to compute the similarities between two users (Sene, Kamsu-Foguem, & Rumeau, 2018). The degree of similarity to the α freely adjustable parameter influences the calculation of $[(1-|\lambda_{li}-\lambda_{lj}|/M)/\kappa(m_l)]^{\alpha}$ similarity in two-way networks of a user movie. As a result, the movie m_l to the similarity s_{ij} may have a negative correlation with its degree $k(m_l)$ and a positive correlation with its preference and trust ratio (Roth et al., 2010). The rating score range of the movie is expressed as *M*; while $k(m_l)$ denotes how many users rate this movie; and λ_{li} - λ_{lj} represents difference between the maximum and minimum rating scores.

This means that it makes no sense if both users choose a popular movie. While two users choose a very unpopular movie, such users must share common tastes. Suppose, the contribution of m_l to s_{ij} has inverse proportionality in relation to $k(m_l)$ and direct proportionality to $|\lambda_{li} - \lambda_{lj}|$, then s_{ij} can be expressed as [21]:

$$s_{ij} = \frac{1}{\sqrt{\kappa(u_i).\kappa(u_j)}} \sum_{l=1}^{n} \delta_{li}\delta_{lj} \left[\frac{(1-(|\lambda_{li}-\lambda_{lj}|/M))}{\kappa(m_l)} \right]^{\alpha}$$

The movie rating score range i.e. M equalled the difference between the maximum and minimum movie rating scores. $\lambda_{1i}/\lambda_{1j}$ denotes the preference level that movie rating m_l received from user u_i. While u_i/u_j .$k(m_l)$ shows how many users have selected this movie. Similarly, $k(u_i)$ represents the number of movies that a specific user chooses. Notwithstanding the calculation of the user similarities, we may find movies that are not chosen by the target user but selected by users whose preferences are much similar to the target user (Shaw, Xu, & Geva, 2010). Then, we can predict the overall preference of the target users for these unseen items as;

$$\Phi_{ij} = \sum_{l=1, l \neq i}^{m} s_{il} a_{jl}$$

In the process of recommendation, we get the item in a descending order of Φ_{ij}, which is not viewed by the target user(s).

Build the association rules

After finding the similarities, then we must find the hybrid association rules for the given user preferences. As shown in Figure 4.

Hybrid Recommender Algorithm with Association Rules

For users of x movies, create a Matrix M_1 and fill with 0s;
Then, store users x N i.e. for Top N recommendations into Matrix M_2
The similarity between user reviews are then calculated by using Adaptive Similarity Measure Model and the results are moved into Matrix M_3
For every rule in the Rule Set $\Re$
  For every user in the Users Set U
    For every movie in the set M_L in the left side of Rule Set $\Re$
      If user U has rated an movie in the set M_L
        Then for every movie in M_R from right side Rule Set $\Re$
$$M_1[U][M_R] \mathrel{+}= conf\,(\Re)$$
For every user in the user set U
  Select the Top-N rating values from the Matrix $M_1[U]$ into the second matrix $M_2[U]$
For every user in the user set U
  For every movie in the m from the Matrix $M_2[U]$
    For each similar movie M_s from $M_3[M]$ (Similarity Coefficient)
      If the Movie M_s is not rated, then add that M_s to $M_2[U]$
Return M_2

Figure 4. hyberid recommender algorithem with associtation rules.

Initially, the first two steps need to be performed, wherein the first is about creating association rules. The second steps calculate the similarity of the movies by their genre and store it in the similarity matrix M_3. The next step is to analyze all the rules and allow each user to create a new matrix M_1 (Koren & Bell, 2011; Felfemig et al., 2011). This matrix contains the confidence level with which the movie can be recommended to the user. Credibility is achieved by adding confident association rules in the recommender systems. This confidence matrix is then used to select the top N recommendations for each user based on the confidence level. They are stored in another matrix M_2. Movies that belong to the same genre as the top N movies stored in the matrix are finally added to the playlist (Koupaei & Khayyambashi, 2015).

Experimental results and analysis

Dataset

This paper uses MovieLens data set provided by Group Lens Research (Xu et al., 2014). It contains 100,000 movies on a scale of 1 to 10 from 5, 943 users for 1682 movies. This dataset is an already preprocessed one. Currently, the chosen threshold is 0.7 for a movie i.e. if a user has a rating of more than 0.7 for a movie, then; the user is likely to like the movie. If you select this similar threshold, the proportion of similar movies from all test users to the total number of movies (called the similar ratio) becomes 0.45.

Results

A few general properties of the data have been presented below for better clarity on the issues. First look at how the ratings change across the year for genres. Figure 5, illustrate Ratings change across the year for genre of animation, sci-fi, war and western.

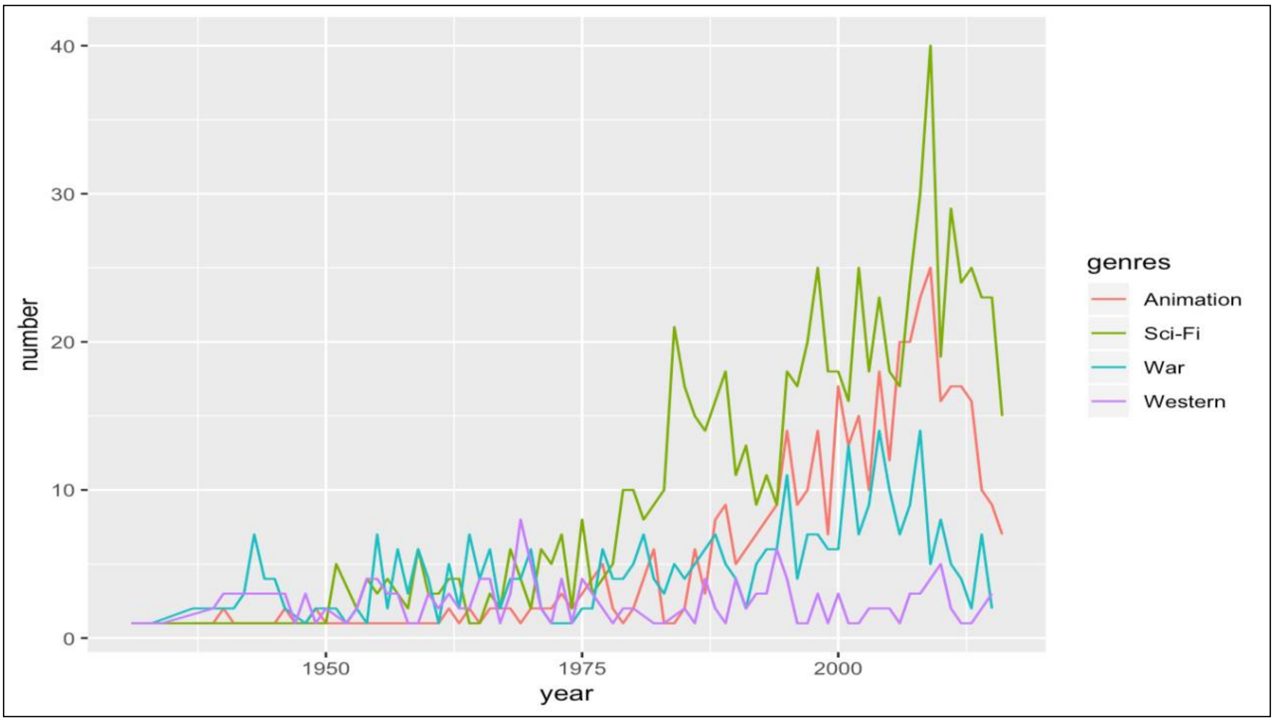

Figure 5. Ratings change across the year for genre of animation, sci-fi , war and western.

The plot clearly shows that genres for Sci-Fi are more in year 2000 than the previous years.

User associations

Maximum rule length

The rule length can be defined as the quantum of items/movies present in the rule precedent. In this paper, different rule lengths are considered (i.e. 2, 4, 6, 8, and 10) by choosing minimum confidence as 100% and the rule number within the range of 5-100. The Table1 shows a rule length of 8 that gives better performance when compared to others rule lengths (i.e. 2, 4, 6, and 10).

Table1. Performance for different maximum rule length.

Rule Length	2	4	6	8	10
Accuracy	0.693512	0.693612	0.695676	0.697897	0.694542
Precision	0.704377	0.724086	0.733488	0.737898	0.736089
Recall	0.572616	0.545435	0.528645	0.528451	0.520823

In general, the larger maximum rule length gets more rules above a certain minimal support and minimal confidence. But, MovieLens data set contains very few rules with only one rule length greater than 8 with a relatively high value support and confidence. Moreover, long rules tend to cause over fitting of data. Hence, the rule with a maximum length of 8 is selected for further analysis. Figure 6, shown Performance for different minimum confidence with a constant score threshold.

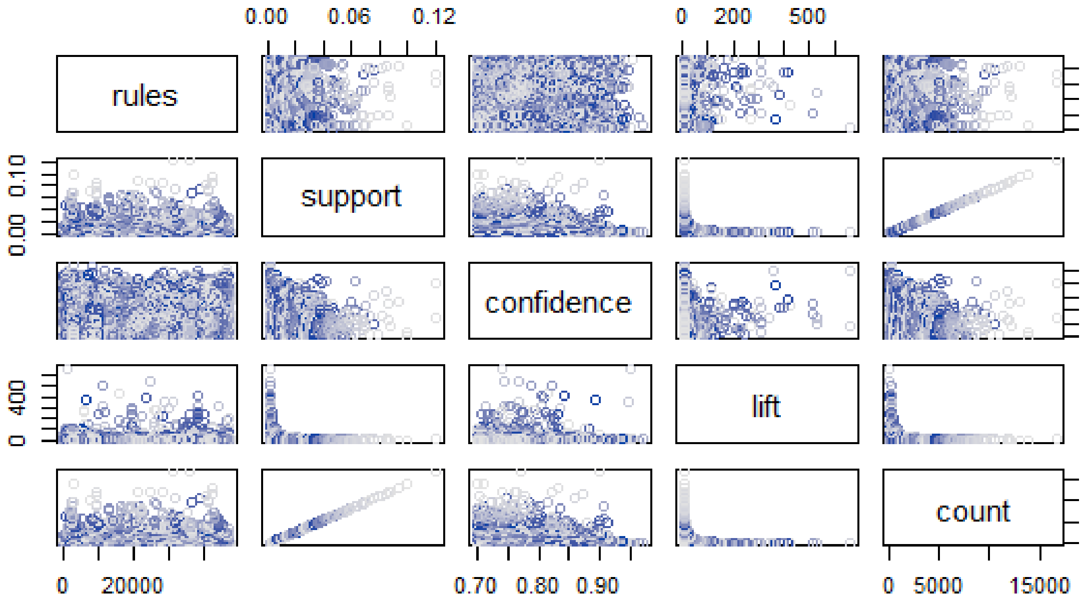

Figure 6. Performance for different minimum confidence with a constant score threshold

Figure 7, plotting different association rules with support confidence and lift. The hybrid association algorithm performance is tested by varying the minimum confidence (i.e. 0.85, 0.90, 0.95, 1) with score threshold = 0.15 and num_rule = 5~100.

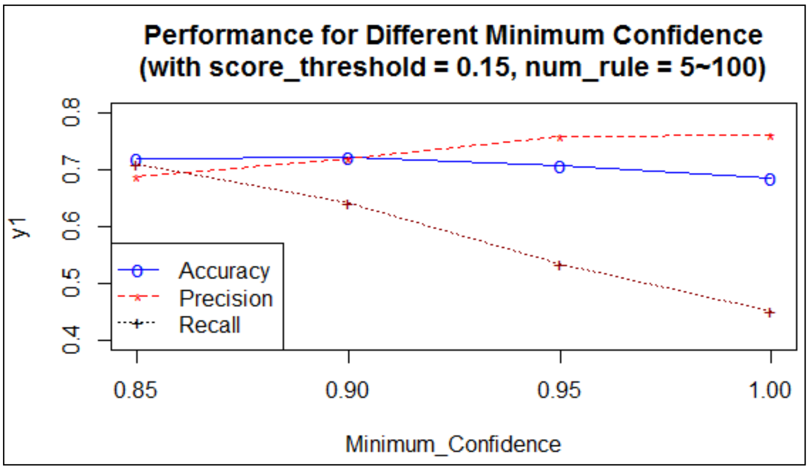

Figure 7. plotting different association rules with support confidence and lift

It is observed that this method produces the most precise result at 0.76, besides recall value at 0.45, satisfying the desired minimum confidence level of 100%, which in turn shows that these rule too can be recommend for training the Recommender Systems. However, when the minimum confidence value varies, this method shows a tradeoff between precision and recall values and hence, the recall linear threshold value is used to improve better combination of the precision as shown in Figure 8.

In fact, higher precision and a higher recall tend to give better recommendations to the target user. Figure 8 shows that by varying the minimum confidence value, we can achieve upto 95 % precision and better recall values.

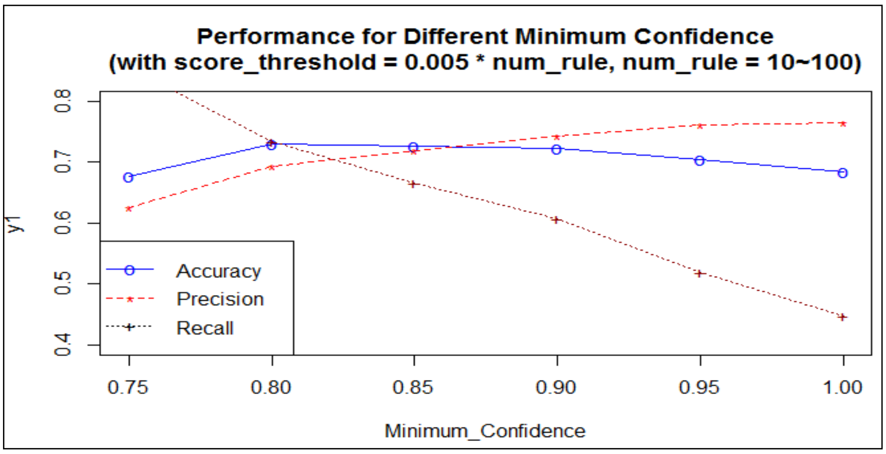

Figure 8. Performance for different minimum confidence with a linear score threshold precision is regarded as the most important thing for recommender systems.

Conclusion

This paper describes two different methods of recommendation. While the primary focus is on achieving improvements in the quality of our recommendations, it is likely to ultimately help the targeted users with strong recommendations. It has been shown that even a moderate improvement in performance may help avoid hindrance to the effective functioning of the collaborative filtering technique. In addition, the results may also help to design new and more a powerful hybrid architecture to provide solid recommendations to the users.

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734–749. DOI: http://doi.org/10.1109/tkde.2005.99

Banati, H., & Mehta, S. (2010). Memetic collaborative filtering based recommender system. In *Proceedings of the 2010 Second Vaagdevi International Conference on Information Technology for Real World Problems*, (p. 102-107). DOI: http://doi.org/10.1109/VCON.2010.28

Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for Youtube recommendations In *Proceedings of the 10th ACM conference on recommender systems* [RecSys '16], (p. 191-198). DOI: http://doi.org/10.1145/2959100.2959190

Desrosiers, C., & Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (p. 107–144). Cham, SW: Springer. DOI: http://doi.org/10.1007/978-0-387-85820-3

Felfemig, A., Friedrich, G., Jannach, D., & Zanker, M. (2011). Developing constraint-based recommenders. In F. Ricci, L. Rokach, B. Shapira & P. Kantor (Eds.), *Recommender Systems Handbook* (p.187–251). Boston, MA: Springer. DOI: http://doi.org/10.1007/978-0-387-85820-3_6

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. DOI: http://doi.org/10.1145/138859.138867

Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999). Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the 16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence* [AAAI'99/IAAI'99], (p. 439-446). DOI: http://doi.org/10.5555/315149.315352

Koren, Y., & Bell, R. (2011). Advances in collaborative filtering. In F. Ricci, L. Rokach, B. Shapira & P. Kantor (Eds.), *Recommender Systems Handbook* (p. 145–186). Boston, MA: Springer.

Koupaei, Z. E., & Khayyambashi, M. (2015). Improving recommendation diversity by classifying users based on demographic information. *IJCCER – International Journal of Computer & Communication Engineering Research, 3*, 68-74.

Lee, C.-H., Kim, Y.-H., & Rhee, P.-K. (2013). Web personalization expert with combining collaborative filtering and association rule mining technique. *Expert Systems and Applications, 21*(3), 131-137. DOI: http://doi.org/10.1016/S0957-4174(01)00034-3

Liangxing, Y., & Aihua, D. (2010). Hybrid product recommender system for apparel retailingcustomers. In *Proceedings of the 2010 WASE International Conference onInformation Engineering* [ICIE '10], (p. 356-360). DOI: http://doi.org/10.1109/ICIE.2010.91

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. *Expert Systems with Applications: An International Journal, 41*(4), 2065-2073, DOI: http://doi.org/10.1016/j.eswa.2013.09.005

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing, 7*(1), 76–80. DOI: http://doi.org/10.1109/mic.2003.1167344

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer supported Cooperative Work* [CSCW'94], (p. 175–186). DOI: http://doi.org/10.1145/192844.192905

Roth, M., Ben-David, A., Deutscher, D., Flysher, G., Horn, I., Leichtberg, ... Meron, R. (2010). Suggesting friends using the implicit social graph. In *Proceedings of 16th International Conference on Knowledge Discovery and Data Mining* [ACM SIGKDD], (p. 233-242). DOI: http://doi.org/10.1145/1835804.1835836

Salter, J., & Antonopoulos, N. (2006). CinemaScreen recommender agent: combining collaborative filtering and content-based filtering. *IEEE Intelligent Systems, 21*(1), 35-41. DOI: http://doi.org/10.1109/MIS.2006.4

Salter, J., & Antonopoulos, N. (2006). CinemaScreen recommender agent: combining collaborative and content-based filtering. *IEEE Intelligent Systems, 21*(1), 35-41. DOI: http://doi.org/10.1109/MIS.2006.4

Sene, A., Kamsu-Foguem, B., & Rumeau, P. (2018). Discovering frequent patterns for in-flight incidents. *Cognitive Systems Research, 49*, 97-113. DOI: http://doi.org/10.1016/j.cogsys.2018.01.002

Shaw, G., Xu, Y., & Geva, S. (2009). Investigating the use of association rules in improving the recommender systems. In J. Kay, P. Thomas & A. Trotman (Eds.), *Proceedings of the Fourteenth Australasian Document Computing Symposium. School of Information Technologies* (p. 1-4). Sydney, AU: University of Sydney.

Shaw, G., Xu, Y., & Geva, S. (2010). Using association rules to solve the cold-start problem in recommender systems. In M. J. Zaki, V. Pudi, Xu Yu, J. & Ravindran, B. (Eds.), *Advances in Knowledge Discovery and Data Mining, Part I: 14th Pacific-Asia Conference, PAKDD 2010 Proceedings [Lecture Notes in Computer Science]* (Vol. 6118, p. 340-347). Germany: Springer.

Shridhar, M., & Parmar, M. (2017). Survey on association rule mining and its approaches. *International Journal of Computer Sciences and Engineering, 5*(3), 129-135.

Ting, C.-K., Liaw, R.-T., Wang, T.-C., & Hong, T.-P. (2017). Mining fuzzy association rules using a memetic algorithm based on structure representation. *Memetic Computing, 10*(5), 15–28. DOI: http://doi.org/10.1007/s12293-016-0220-3

Wang, Q., Ma, J., Liao, X., & Du, W. (2017). A context-aware researcher recommendation system for university-industry collaboration on R&D projects. *Decision Support Systems, 103*(C), 46–57. DOI: http://doi.org/10.1016/j.dss.2017.09.001

Wang, Q., Zeng, C., Zhou, W., Li, T., Shwartz, L., & Grabarnik, G. Y. (2017). Online interactive collaborative filtering using multi-armed bandit with dependent arms. arXiv:1708.03058. DOI: http://doi.org/10.1109/TKDE.2018.2866041

Xu, J., Karaleise, J.-W., & Li, S. (2014). The development, status and trends of recommender systems: a comprehensive and critical literature review. In *Proceedings of Mathematics and Computers in Sciences and Industry* [MCSI 2014], (p. 117-122). Recovered from: https://bitlybr.com/4URQ

Zuo, Y., Gong, M., Zeng, J., Ma, L., & Jiao, L. (2015). Personalized recommendation based on evolutionary multi-objective optimization [Research Frontier]. *IEEE Computational Intelligence Magazine, 10*(1), 52-62. DOI: http://doi.org/10.1109/MCI.2014.2369894