Vietnam Journal of Computer Science
Vol. 6, No. 3 (2019) 363–376

DOI: 10.1142/S2196888819500192

Improved Movie Recommendations Based on a Hybrid Feature Combination Method

Gharbi Alshammari, Stelios Kapetanakis, Abdullah Alshammari and Nikolaos Polatidis*

School of Computing, Engineering and Mathematics
University of Brighton, Lewes Road
Brighton BN2 4GJ, UK
**n.polatidis@brighton.ac.uk*

Miltos Petridis

Department of Computer Science
Middlesex University, The Burroughs
Hendon, London NW4 4BT, UK

Received 4 December 2018
Accepted 13 June 2019
Published 8 July 2019

Recommender systems help users find relevant items efficiently based on their interests and historical interactions with other users. They are beneficial to businesses by promoting the sale of products and to user by reducing the search burden. Recommender systems can be developed by employing different approaches, including collaborative filtering (CF), demographic filtering (DF), content-based filtering (CBF) and knowledge-based filtering (KBF). However, large amounts of data can produce recommendations that are limited in accuracy because of diversity and sparsity issues. In this paper, we propose a novel hybrid method that combines user–user CF with the attributes of DF to indicate the nearest users, and compare four classifiers against each other. This method has been developed through an investigation of ways to reduce the errors in rating predictions based on users' past interactions, which leads to improved prediction accuracy in all four classification algorithms. We applied a feature combination method that improves the prediction accuracy and to test our approach, we ran an offline evaluation using the 1M MovieLens dataset, well-known evaluation metrics and comparisons between methods with the results validating our proposed method.

Keywords: Recommender systems; collaborative filtering; demographic filtering; hybrid recommendation.

*Corresponding author.

1. Introduction

This paper extends our previous conference paper[1] by utilizing four different recommendation methods whereas in the conference two were used. Furthermore, the evaluation section (Sec. 4) has been expanded by using two additional metrics and more experiments which further enhance the understanding of the results. Finally, a discussion section (Sec. 5) has been added which explains the results in more depth.

The amount of available information on the Internet has increased exponentially in the last decade and this has led to the problem of information overload. More specifically, the E-commerce industry is presenting a wider range of options, which makes it more difficult for users to shop and find products. Hence, to help customers find new items by means of suggestions, companies need to develop a recommender system. Such systems help their sales to grow by providing relevant options that meet users' requirements. For example, in regard to movie recommendation, the Netflix Prize raised the importance of the recommender system in attracting more users, and the competition to produce highly developed algorithms led to more accurate results in recommendations.[2] A recommendation system is an information filtering method that is used to predict the items a certain user will like (the prediction problem) or to recommend a set of top items that meet the user's preferences.[3] Users have trouble handling large volumes of information, and problems with cognitive and data sparsity when attempting to find appropriate information at the right time.[4] Based on profile data, a recommender system can be categorized into four main stages: similarity computation, neighborhood selection, prediction and recommendation. The profile can be modeled by content-based filtering (CBF), collaborative filtering (CF) or demographic filtering (DF). If the user profile contains a set of attributes obtained from the item descriptions that the user has liked, this is CBF. DF is represented by a set of features in a user's profile. CF can be described as when the profile contains a list of items that have been rated. The CF technique is widely used as a recommender system-based method due to its capability and efficiency for predicting similar neighbor users. However, extensive growth in the numbers of users and items may cause a sparsity issue in CF techniques when used on their own. Thus, we have made the following contributions:

(1) We have developed a method that employs both ratings and demographic information, by combining demographic attributes with user–item rating CF to solve the problem of data sparsity.
(2) This method allows us to efficiently calculate similarities in a large dataset with no pre-calculation or pre-processing.
(3) We have evaluated our method using a real dataset and well-known evaluation metrics and shown that it is both practical and effective.

The rest of the paper is organized as follows: Section 2 presents the related works, Sec. 3 delivers the proposed method, Sec. 4 explains the experimental evaluation, Sec. 5 contains the discussion and Sec. 6 contains the conclusions and future work.

2. Related Works

The idea of recommender systems first appeared in the mid-1990s, relying on the idea that users share similar items or opinions, thereby helping to make recommendations to others.[5] The researchers established a collaborative filtering technique based on ratings structure. Hence, the most common formulation is to calculate ratings for items that have not been seen by a particular user. Recommender systems can be defined as adaptable tools that help users search for, filter and classify information, and then recommend relevant items.[6] Recommendation systems use a number of different techniques. These methods can be implemented based on the domain requirement and are able to identify and predict items that meet the users' interests. They also utilize different recommendation algorithms to make suggestions and recommendations.

2.1. *Collaborative filtering*

Collaborative filtering is considered to be the most popular technique for recommender systems. It has been widely implemented in different domains to make recommendations. It is a method of information filtering that seeks to predict the rating that a user would give to a particular item based on a similarity matrix. Collaborative filtering provided the foundation for the first recommender systems, which were used to help people make choices based on the opinions of other people.[7] It helps users to find relevant items and makes suggestions based on similar users' tastes. It has been applied in a variety of areas, such as in movies, books and research articles. In this approach the similarity calculation is based on the user's peers. User-based CF: This method looks for similarity between users based on the same rating pattern.[8] It makes a recommendation based on the similarity between the target user and other users. The idea is that, for a given user, the preferences of similar users (neighbors) can serve as recommendations. A user–user approach was proposed as an appropriate method for recommending items based on expert opinions.[9] In addition, another example is provided by Ref. 10, where mobile activities were recommended to users based on their locations. Item-based CF: This method recommends items based on similarities between items shared with similar users.[8] In Ref. 11, an item-to-item collaborative filtering approach was designed that matched items rated or purchased by the user with other similar items.

2.2. *Demographic filtering*

It is possible to identify the type of person who likes a particular item by referencing their demographic details.[12] User attributes are incorporated into demographic recommender systems and this demographic data is used as a basis for arriving at suitable recommendations, sometimes relying on pregenerated demographic clusters.[13] This information is gathered either explicitly through user registrations or implicitly via navigation of the system they use.[14] Subsequently, demographically similar users are identified by means of the recommendation algorithm. Recommendations are based on how similar people (in terms of their demographics) rated a particular item.[15] In Ref. 13, a hybrid algorithm was presented that keeps the core ideas of two existing recommender systems and enhances them with relevant information extracted from demographic data. The authors in Ref. 12 presented an approach that considers user profiles as vectors constructed from demographic attributes such as age, gender or postcode to find the relationships with other users and calculate similarities between users, in order to generate the final prediction. Demographic-based filters are similar to collaborative filters in the sense that both are able to identify similarities between users. In this case, demographic features are used to determine similarity rather than the users' previous ratings of items.[15] Demographic attributes are added as meta-data to help the neighborhood algorithm find similar users. The author in Ref. 16, presented the importance of these meta-data in producing significant results and providing better recommendations.

In Ref. 17, the author stated that demographic information helps to address the cold-start problem. This is because this approach does not require a detailed history of user ratings before making recommendations, unlike the content-based and collaborative approaches.[18] The importance of demographic information (age, gender) was studied in a research paper on recommender system.[19] The authors showed that demographic information had a significant impact on recommendations. The combination of collaborative filtering and the demographic base can enrich user preferences and more accurately identify their interests.

2.3. *Hybrid recommendation approaches*

More recently, the hybrid recommendation approach has become a widely debated issue. A possible way to combine the recommendation methods was introduced in Ref. 20. Authors in Ref. 21 also introduced a hybrid approach for solving the problem of finding the rating of unrated items in a user–item matrix through a weighted combination of user-based CF and item-based CF. These methods addressed the two major challenges of recommender systems, the accuracy of recommendations and sparsity of data, by simultaneously incorporating the correlation of users and items.

Because of data sparsity, finding the nearest neighbors is becoming more of a challenge, with the fast growth in users and items. In Ref. 22, a switching hybrid approach was proposed to solve the long tail problem in recommendations. A hybrid approach was applied that utilized clustering and genetic algorithms to reduce data sparsity in movie recommendations. The results showed that this approach improves recommendation accuracy.[23] In Ref. 24, a hybrid framework was proposed that utilized collaborative filtering relying on user/item meta-data and demographic data. The framework benefits from the similarity between users via correlation in terms of demographic attributes. This improves prediction and is able to solve the cold-start problem compared with the baseline. The authors point out the importance of item meta-data in overcoming the challenge in which user and item have little information. In Ref. 13, the discussion explored the usefulness of demographic data as an enhancing factor, by employing a hybrid algorithm to improve collaborative filtering in terms of both algorithms, user-based and item-based. In Ref. 25, a combination algorithm was proposed using demographic attributes based on a clustering approach in a weighted scheme. It solved the cold-start problem by assigning a new user to the nearest cluster based on demographic similarity. In contrast to the aforementioned methods, our proposed method is beneficial for exploring the effect of demographic data.

3. Proposed Method

In this section, the proposed method is defined that combines CF and DF. The main idea of the method, which is not found in other works in the literature, is to have a hybrid recommendation approach that can be easily used for the evaluation of different classifiers in order to identify which classifier performs better when demographic data are integrated into the recommendation process.

The sparsity issue is a major challenge for recommender systems in terms of producing the right recommendations for the right users. This issue has been further expanded due to the growth of items available and of users with few ratings and little user information. This leads to difficulty in finding similarity between two users. In this section, we propose how a feature combination hybrid approach solves the sparsity problem and reduces the error rate through using two classifiers. It combines matching user demographic attributes with the user rating CF method as shown in Fig. 1.

In order to evaluate the proposed method we conducted an experiment on a real dataset that is publicly available from MovieLens.[26] In this paper we used the 1M dataset which contains 1,000,209 ratings that were assigned by 6,040 users on around 3,900 movies. We utilized demographic information that includes age, gender and occupation. We combined the demographic information for each user with the user–item ratings. Hence, each user was defined as a vector with those features.

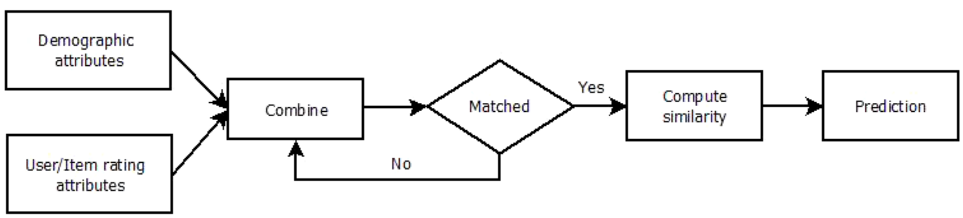

Fig. 1. The architecture of the proposed method.

Algorithm 3.1. Proposed algorithm method

```
Input: User–item rating attributes file (f1) and demographic attributes file (f2).
Output: User demographic attributes with item rating file (f3).
for <row in f2> do
    for <line in f1> do
        if row[0] == line[0] then
            f3 = row[0, 1, ..., N] + line[1, 2, ..., N];
        end if
    end for
end for
```

The attributes used in this filtering system are age, gender and occupation. Those attributes are defined as categorical, and represent each user in a group. They can help in finding similar users, in order to improve the rating prediction accuracy. The profile vector is represented at the attribute level to compute the similarity.

Then, we calculated the similarity between the active user and the nearest one. Next is the final step of calculating the predicted rating. We ran this experiment using Orange 3.7.0, which is a data mining and machine learning tool. We conducted a cross-validation with number of folds of 10. In summary, the steps of the proposed method are:

(1) CF is combined with demographic attributes such as age, gender and occupation to find more similar users. The combination was made through matching the user ID from user–item ratings data with user demographics data as detailed in Algorithm 3.1, where row[0] and line[0] represent the user ID, and row$[0, 1, \ldots, N]$ represent the attributes in CF and line$[1, 2, \ldots, N]$ represent the attributes in DF.
(2) After matching each user with the demographic attributes, the similarity is computed.
(3) In the final step, the predicted rating is calculated and compared with the actual rating to calculate the differences.

4. Experimental Evaluation

Evaluation metrics play an important role in measuring the quality and the performance of a recommender system. Since 1994,[5] the accuracy of the recommender system has been evaluated in the literature in different ways. Furthermore, as there is no standard for evaluation, it is hard to compare the results with other published articles. However, there are main evaluation metrics that are widely applied to benchmark the results and compare them with the proposed algorithms. Most of the empirical studies examining recommender systems have focused on appraising the accuracy of these systems.[27] This insight is useful for evaluating the quality of the system and its ability to forecast the rating for a particular item.

4.1. *Predictive accuracy metrics*

This measures the similarity between true user ratings and the predicted ratings. This research applies accuracy metrics to measure the performance of the proposed methods. Both the mean absolute error (MAE) in Eq. (1) and the root-mean-squared error (RMSE) in Eq. (2) are used to evaluate the prediction accuracy of the different recommendation techniques, where p_i is the predicted rating and r_i is the actual rating. Prediction accuracy is enhanced when MAE and RMSE are lower. Here we detail those similarity measures

4.1.1. *Mean absolute error*

It takes the mean of absolute difference between the predicted rating and the actual rating for all the ratings as defined in Eq. (1). In MAE, p_i is the absolute value of the predicted rating, and r_i is the absolute value of the actual rating. It should be considered that lower values provide better results:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |p_i - r_i|. \tag{1}$$

4.1.2. *Root-mean-squared error*

It represents the sample standard deviation of the differences between predicted values and the actual values and is defined in Eq. (2). In RMSE, p_i is the predicted rating and r_i is the actual rating. It should be considered that lower values provide better results:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (p_i - r_i)^2}. \tag{2}$$

4.1.3. *Improvement rate*

Additionally, we have used the improvement rates (IRs) based on MAE and RMSE, respectively. Equation (3) defines the improvement rate for MAE and Eq. (4) defines the improvement rate for RMSE. The values derived from the use of these two equations show the improvement in terms of percentages:

$$\mathrm{IR}_{\mathrm{MAE}} = \frac{\mathrm{MAE}_{\mathrm{base}} - \mathrm{MAE}_{\mathrm{proposed}}}{\mathrm{MAE}_{\mathrm{base}}}, \tag{3}$$

$$\mathrm{IR}_{\mathrm{RMSE}} = \frac{\mathrm{RMSE}_{\mathrm{base}} - \mathrm{RMSE}_{\mathrm{proposed}}}{\mathrm{RMSE}_{\mathrm{base}}}. \tag{4}$$

4.2. *Classification algorithms*

In order to find out which classifier is the most appropriate one to use for this dataset and to make a good prediction for the movie domain, we ran an experiment on those two that are widely applied in movie recommendation for evaluating the results. Next, we describe in detail each classifier.

4.2.1. *k-Nearest neighbor*

The k-nearest neighbor (kNN) is a classifier that finds the k-nearest neighbors. The given user is assigned a number based on similar users who share the most common features of its k nearest neighbor users. Certain factors need to be considered, such as the similarity measurement, which calculates the distance between two vectors p_i and r_i for $i = 1, 2, \ldots, k$ representing the neighborhood, which needs to be positive number. For the purpose of experiments, we have used Euclidean distance which is defined in Eq. (5). In this equation, p and q are two points (users) and the Euclidean distance between them is to be measured based on the sum of the predicted, p, and actual, r, ratings:

$$d(p, q) = \sqrt{\sum_{i=1}^{k} (p_i - r_i)^2}. \tag{5}$$

4.2.2. *Random forest*

This is an ensemble learning classifier that builds a set of decision trees. Each tree is developed from a bootstrap sample from training data. It is more robust with respect to noise. This method has been successfully approved as an accurate machine learning classifier.[28–30] We set the numbers of trees to be 10, 20, 50 and 100, which are the most likely changes in this range.

4.2.3. *Neural network*

This is a feed-forward neural network (NN). This method has been successfully approved as an accurate machine learning classifier and for the experiments 100 layers have been used.

4.2.4. *AdaBoost*

This is another well-known classifier that can be used in machine learning applications. It is a well-known boosting classifier that improves the accuracy through adapting multiple weak classifiers into a more satisfied classifier with better accuracy.[31] For the experiments, a learning rate of 1 has been used with 10 estimators.

4.3. *Results*

In Fig. 2, it is shown that the MAE accuracy metrics are made through applying different kNN classifiers with $k = 3$, 10, 30, 50 and 100. We then performed the experiment with the random forest classifier using a different set of trees and the results are shown in Fig. 3. It is clear that performance is improved when we combined the demographic attributes (CF + DF). However, it is noticeable that the

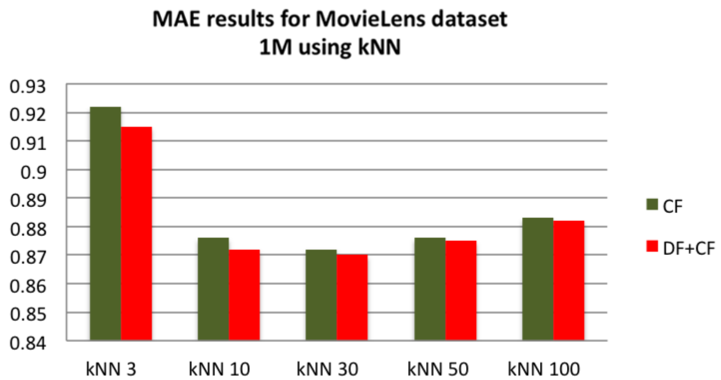

Fig. 2. MAE results for the MovieLens dataset using kNN.

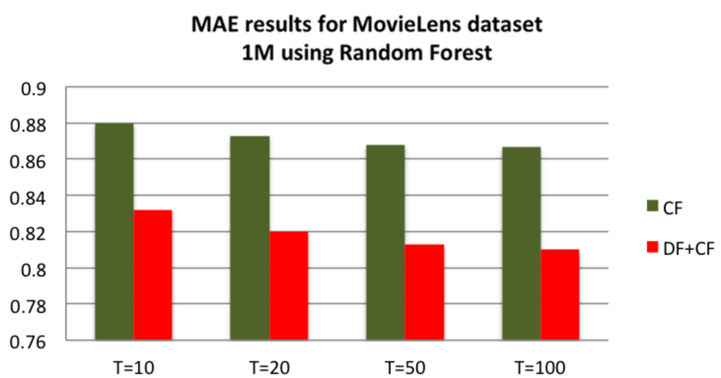

Fig. 3. MAE results for the MovieLens dataset using random forest.

improvement in random forest is much higher than in kNN. For example, in Fig. 3, when $T = 10$ the improvement is 5%. Whereas in Fig. 2, when $k = 3$ the improvement is only 1%. Additionally, Fig. 4 shows the results for the neural network and AdaBoost classifiers. It is shown that a small improvement exists for the neural network but a significant improvement exists for the AdaBoost method.

Figures 5–7 show the results based on the RMSE metrics. It can be seen that there is an improvement in the random forest classifier compared to kNN in the all cases and improvement for the neural network and AdaBoost classifiers as well.

Figures 8 and 9 show the improvement rates based on the results obtained from the MAE and RMSE experiments. It is shown that in both MAE and RMSE cases AdaBoost is the most improved method, followed by random forest, neural network and kNN.

Overall, we ran the experiment with four different classifiers. The experiment demonstrates that demographic attributes have a significant impact on predictive accuracy results. Additionally, we also compare the results between kNN, random

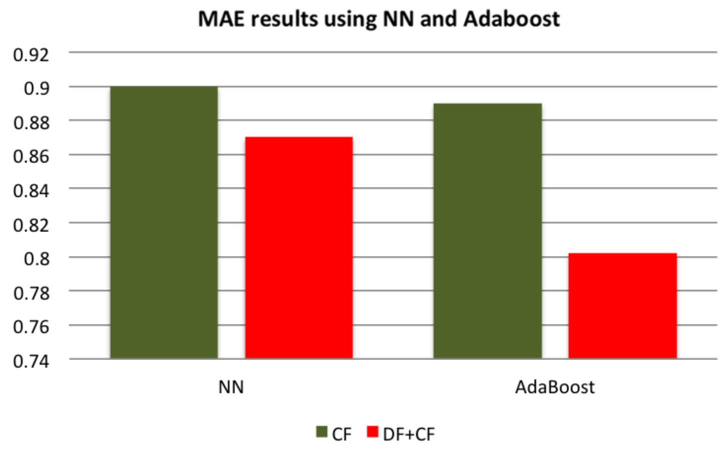

Fig. 4. MAE results for the MovieLens dataset using NN and AdaBoost.

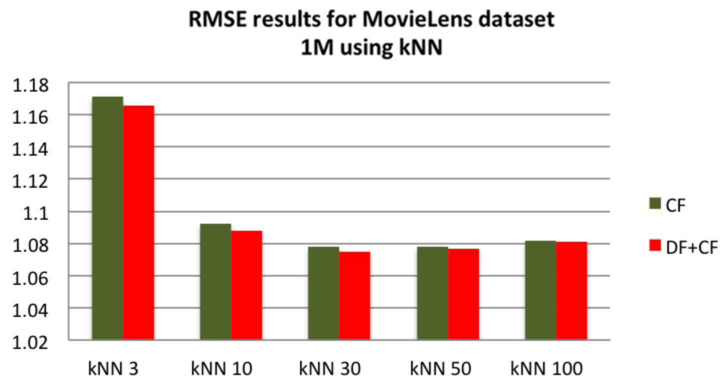

Fig. 5. RMSE results for the MovieLens dataset using kNN.

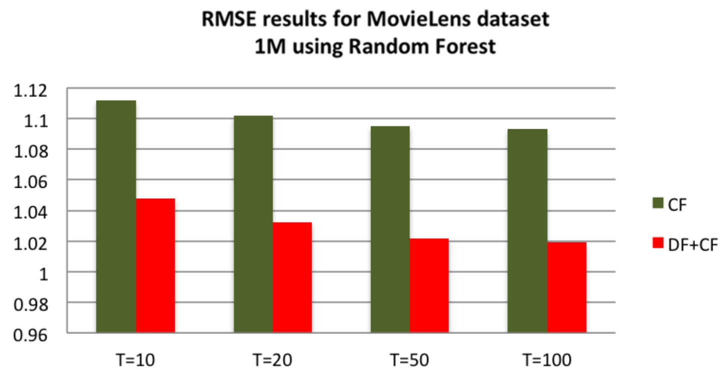

Fig. 6. RMSE results for the MovieLens dataset using random forest.

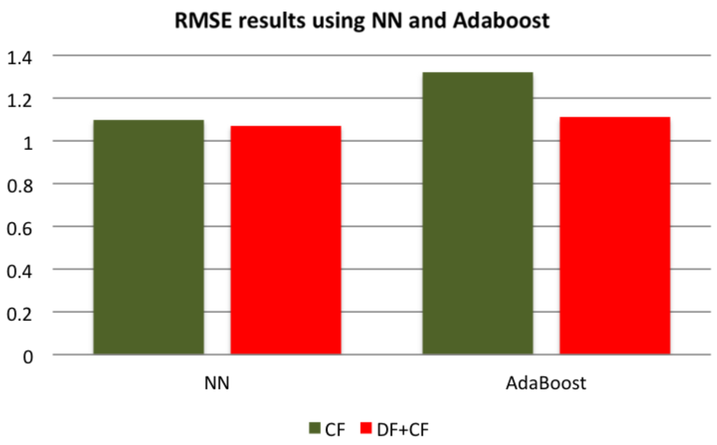

Fig. 7. RMSE results for the MovieLens dataset using NN and AdaBoost

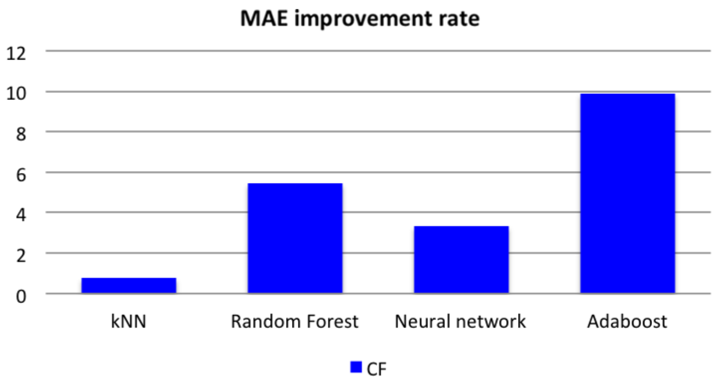

Fig. 8. MAE improvement rate.

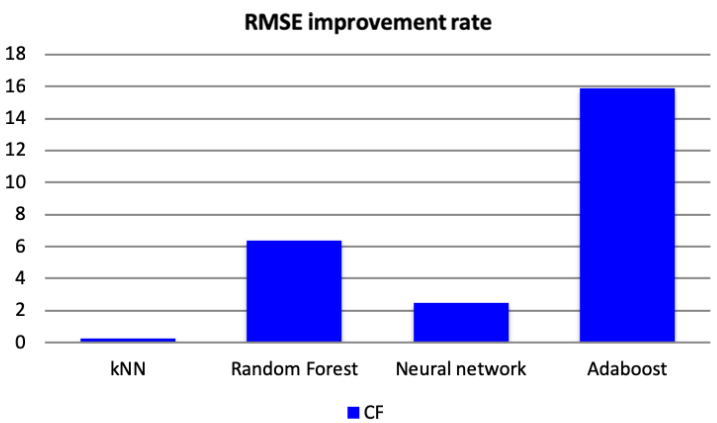

Fig. 9. RMSE improvement rate.

forest, neural network and AdaBoost when recommender systems use demographic characteristics.

The results also indicate that the AdaBoost classifier outperforms all the others which raise high attention to be utilized in the real applications or software using recommendation techniques. More precisely, demographic filtering with the user-based collaborative filtering. This method will be more applicable in a domain where user characteristics play an important role such as Facebook.

5. Discussion

Recommender systems are decision-support systems found on the Web to solve the information overload problem. Users rely on recommender systems to receive good recommendations in Web environments, thus resulting in a reduced search burden for them, and businesses rely on the possibility of better user experience and improved sales by utilizing recommendation technologies.

In this paper, we proposed a method that can complement a business in its recommendation algorithm selection process. This particular method can be applied in domains where demographic information about the user is available such as movie recommendation domains. Collaborative filtering is combined with demographic information using a simple series of steps, thus resulting in improved recommendations for all evaluated classifiers. Such a method is particularly useful for business in scenarios where a new method might be time-consuming or costly to develop and an existing approach needs to be used.

By using the proposed method within a recommendation library and changing only the name of the recommendation algorithm and evaluating each, it can be easily identified which one performs better for the given settings. For example, when the kNN algorithm is applied there is a small but noticeable increase in terms of improvement, neural network is second with a higher improvement rate, random forest is the third with an even higher improvement rate from neural network and AdaBoost is the one with the highest improvement rate. Thus, by noticing the MAE and RMSE improvement rates in Figs. 8 and 9, respectively, it can be easily identified that the preferred choice for a business would be AdaBoost.

6. Conclusions and Future Work

In this paper, we proposed a novel hybrid method for recommender systems based on a simultaneous combination of user-based collaborative filtering and demographic attributes. The results suggest that demographic filtering can effectively improve the overall recommendation. Moreover, the proposed method addresses two common challenges of recommendation systems, namely sparsity of data and improved accuracy of recommender systems, by combining the hidden relations between users and comparing two different classifiers with a large dataset. The proposed method is a comparison between the kNN, the random forest, the neural network and

AdaBoost classifiers, with the results validating the proposed method. In the future, we aim to use specific users who rate only a few items and possibly other attributes related to items.

References

1. G. Alshammari, S. Kapetanakis, A. Alshammari, N. Polatidis and M. Petridis, A hybrid feature combination method that improves recommendations, in *Proc. Int. Conf. Computational Collective Intelligence*, Lecture Notes in Computer Science, Vol. 11055 (Springer, Chem, 2018), pp. 209–218.
2. X. Amatriain, Beyond data: From user information to business value through personalized recommendations and consumer science, in *Proc. 22nd ACM Int. Conf. Information & Knowledge Management* (ACM, 2013), pp. 2201–2208.
3. L. Shah, H. Gaudani and P. Balani, Survey on recommendation system, *Int. J. Comput. Appl.* **137**(7) (2016) 43–49.
4. J. Bobadilla, F. Ortega, A. Hernando and A. Gutiérrez, Recommender systems survey, *Knowl.-Based Syst.* **46** (2013) 109–132.
5. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, GroupLens: An open architecture for collaborative filtering of netnews, in *Proc. 1994 ACM Conf. Computer Supported Cooperative Work* (ACM, 1994), pp. 175–186.
6. P. Resnick and H. R. Varian, Recommender systems, *Commun. ACM* **40**(3) (1997) 56–58.
7. D. Goldberg, D. Nichols, B. M. Oki and D. Terry, Using collaborative filtering to weave an information tapestry, *Commun. ACM* **35**(12) (1992) 61–70.
8. S. Spiegel, A hybrid approach to recommender systems based on matrix factorization, Diploma Thesis, Department for Agent Technologies and Telecommunications, Technical University Berlin (2009).
9. X. Amatriain, N. Lathia, J. M. Pujol, H. Kwak and N. Oliver, The wisdom of the few: A collaborative filtering approach based on expert opinions from the Web, in *Proc. 32nd Int. ACM SIGIR Conf. Research and Development in Information Retrieval* (ACM, 2009), pp. 532–539.
10. V. W. Zheng, B. Cao, Y. Zheng, X. Xie and Q. Yang, Collaborative filtering meets mobile recommendation: A user-centered approach, in *Proc. Twenty-Fourth AAAI Conf. Artificial Intelligence* (AAAI Press, 2010), pp. 236–241.
11. G. Linden, B. Smith and J. York, Amazon.com recommendations: Item-to-item collaborative filtering, *IEEE Internet Comput.* **7**(1) (2003) 76–80.
12. M. J. Pazzani and D. Billsus, Content-based recommendation systems, in *The Adaptive Web*, Lecture Notes in Computer Science, Vol. 4321 (Springer, 2007), pp. 325–341.
13. M. Vozalis and K. G. Margaritis, Collaborative filtering enhanced by demographic correlation, in *Proc. AIAI Symp. Professional Practice in AI, Part of the 18th World Computer Congress* (2004).
14. A.-N. Moldovan and C. H. Muntean, Personalisation of the multimedia content delivered to mobile device users, in *Proc. 2009 IEEE Int. Symp. Broadband Multimedia Systems and Broadcasting* (IEEE, 2009), pp. 1–6.
15. N. Tintarev, Explaining recommendations, in *User Modeling 2007*, Lecture Notes in Computer Science, Vol. 4511 (Springer, 2007), pp. 470–474.
16. P. Mittal, Metadata based recommender systems in *Proc. 2014 Int. Conf. Advances in Computing, Communications and Informatics* (2014), pp. 2659–2664.
17. J. L. Redpath, Improving the performance of recommender algorithms, Ph.D. thesis, University of Ulster (2010).

18. R. Burke, Hybrid web recommender systems, in *The Adaptive Web*, Lecture Notes in Computer Science, Vol. 4321 (Springer, 2007), pp. 377–408.
19. J. Beel, S. Langer, A. Nürnberger and M. Genzmehr, The impact of demographics (age and gender) and other user-characteristics on evaluating recommender systems, in *Research and Advanced Technology for Digital Libraries*, Lecture Notes in Computer Science, Vol. 8092 (Springer Science+Business Media, 2013), pp. 396–400.
20. R. Burke, Hybrid recommender systems: Survey and experiments, *User Model. User-Adapt. Interact.* **12**(4) (2002) 331–370.
21. S. Wei, X. Zheng, D. Chen and C. Chen, A hybrid approach for movie recommendation via tags and ratings, *Electron. Comm. Res. Applic.* **18** (2016) 83–94.
22. G. Alshammari, J. L. Jorro-Aragoneses, S. Kapetanakis, M. Petridis, J. A. Recio-García and B. Díaz-Agudo, A hybrid CBR approach for the long tail problem in recommender systems, in *Proc. Int. Conf. Case-Based Reasoning*, Lecture Notes in Computer Science, Vol. 10339 (Springer, 2017), pp. 35–45.
23. Z. Wang, X. Yu, N. Feng and Z. Wang, An improved collaborative movie recommendation system using computational intelligence, *J. Vis. Lang. Comput.* **25** (2014) 667–675.
24. S. Bremer, A. Schelten, E. Lohmann and M. Kleinsteuber, A framework for training hybrid recommender systems, in *Proc. Rec SysKTL Workshop* (2017), pp. 30–37.
25. J. Gupta and J. Gadge, Performance analysis of recommendation system based on collaborative filtering and demographics, in *Proc. 2015 Int. Conf. Communication, Information & Computing Technology* (2015).
26. F. M. Harper and J. A. Konstan, The MovieLens datasets: History and context, *ACM Trans. Interact. Intell. Syst.* **5**(4) (2016) 19.
27. J. L. Herlocker, J. A. Konstan, A. Borchers and J. Riedl, An algorithmic framework for performing collaborative filtering, in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Research and Development in Information Retrieval* (ACM, 1999), pp. 230–237.
28. G. Louppe, Understanding random forests: From theory to practice, arXiv:1407.7502 [stat.ML].
29. L. Breiman, Random forests (2001), https://www.stat.berkele.edu/~breiman/randomforest2001.pdf.
30. H. Zhang, F. Min and S. Wang, A random forest approach to model-based recommendation, *J. Inf. Comput. Sci.* **11**(15) (2014) 5341–5348.
31. X. Jiang, Z. Niu, J. Guo, G. Mustafa, Z. Lin, B. Chen and Q. Zhou, Novel boosting frameworks to improve the performance of collaborative filtering, in *Proc. Asian Conf. Machine Learning* (2013), pp. 87–99.