The 50/50 Recommender: A Method Incorporating Personality into Movie Recommender Systems

Orestis Nalmpantis and Christos Tjortjis(✉)

School of Science and Technology, International Hellenic University,
14th km Thessaloniki - Moudania, 57001 Thermi, Greece
{o.nalmpantis,c.tjortjis}@ihu.edu.gr

Abstract. Recommendation systems offer valuable assistance with selecting products and services. This work checks the hypothesis that taking personality into account can improve recommendation quality. Our main goal is to examine the role of personality in Movie Recommender systems. We introduce the concept of combining collaborative techniques with a personality test to provide more personalized movie recommendations. Previous research attempted to incorporate personality in Recommender systems, but no actual implementation appears to have been achieved. We propose a method and developed the 50/50 recommender system, which combines the Big Five personality test with an existing movie recommender, and used it on a renowned movie dataset. Evaluation results showed that users preferred the 50/50 system 3.6% more than the state of the art method. Our findings show that personalization provides better recommendations, even though some extra user input is required upfront.

Keywords: Personalization · Collaborative filtering · Recommendation systems · Data mining

1 Introduction

The massive growth and impact of the World Wide Web had as a result the handling and distribution of huge amounts of data and information. Although this may seem as an improvement for computer technology, it certainly came with some drawbacks.

This is where a recommender system comes in place, an assistive device that directs and guides the user in their search for useful information. Personalization in recommender systems tends to be a new trend, and it is mostly based on the theory of human-computer interaction, which states that computers will and should always work with and for humans.

The main motivation behind this work is the lack of personalization in current recommender systems. We hypothesize that a recommender system should factor in the basic personality traits of the active user and that this will help in producing better recommendations.

Our aim is to use the Big Five Personality test [1], to find the user's personality traits and then connect these traits with his/her movie genre preferences. Then, based

G. Boracchi et al. (Eds.): EANN 2017, CCIS 744, pp. 498–507, 2017.
DOI: 10.1007/978-3-319-65172-9_42

on the movie reviews the user gave to the system and the personality test results, we recommend a list of movies by applying our own formula. This list takes into account 50% his personality and 50% his movie. We also experimented with how users react to a list that is filled with movies that are 80% based on their personality and 20% on their movie reviews to see how far we can go with the addition of Personality in producing final recommendations, and compared these to just using standard k-NN based recommendations.

The remaining of the paper is organized as follows: in Sect. 2, we provide the background to our research, which includes information on CF Recommender Systems, k-NN and Personalization. In Sect. 3, we detail the proposed 50/50 Movie Recommender System. In Sect. 4, we present results and discuss the pros and cons of our method, along with threats to validity. Finally, Sect. 5 concludes with ideas for future improvements.

2 Background

A Recommender system is “any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options” [2]. This is the main concept behind CF Recommender systems.

2.1 Collaborative Filtering Recommender Systems

The most common way of getting a recommendation in real life is by asking a friend for their opinion, preferably someone who usually likes the same things as you. This is exactly the idea behind CF [13]. The basic premise of CF is that if two users have the same opinion about a bunch of products, then they are likely to have similar opinions about other products too. The objective of the algorithm is to predict the active user’s ratings for products they have not yet rated. With these predicted ratings, you can sort the products and recommend the top picks for the active user. The following sub-section shows the Basic Steps in CF as discussed in [9, 10], which are also the basic steps we follow initially for our recommender.

CF Basic Steps

1. The set of ratings for the active user is identified.
2. k-NN is used to select k users who are most similar to the active user, according to a similarity function called Pearson Correlation shown in formula (1).
3. Identify the products that these similar users liked.
4. A prediction is generated with a Prediction Rating Formula, meaning ratings that would hypothetically be given by the active user for each of these products.
5. A set of top N products is recommended based on the top highest predicted ratings of the products in the previous step.

In step 2, the k-NN Algorithm and Pearson Correlation are used to find similar users to the active user. Generally, the k-NN algorithm represents all the users as Data Points in an X- Y plot and Pearson Correlation is used as a similarity distance metric to find which of these Data Points are closer to the active user. The closer they are, the more similar they are to the active user.

$$Corr(x, y) = \frac{\sum_i (\boldsymbol{xi} - \bar{\boldsymbol{x}})(\boldsymbol{yi} - \bar{\boldsymbol{y}})}{\sqrt{\sum (\boldsymbol{xi} - \bar{\boldsymbol{x}})^2} \sqrt{\sum (\boldsymbol{yi} - \bar{\boldsymbol{y}})^2}} \tag{1}$$

Then in Step 4, once we find the k-nearest neighbors of the active user, we use those neighbors to find the rating that the active user will give to any product, using the following formula (2):

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u).w_{a,u}}{\sum_{u \in U} |w_{a,u}|} \tag{2}$$

This is the foundation of our Recommender, and later this will be enhanced by the addition of the personality feature of the active user.

2.2 Personalization

Computer scientists try to model the human psychological aspects and include them in recommender systems, so to make the recommender more personalized. As described in [3], Recommender systems are not always capable of generating good recommendations for the user based only on raw data. The authors provide their own framework, which is called Human-Recommender Interaction (HRI). It's their way to connect the user needs with the recommender algorithms.

Their two main assumptions are first, that recommenders cannot understand the reason a user asks for recommendations and second, that recommenders should be trained to have personalities and interact with the users in a conversational way. This paper appreciates the use of recommender systems and suggests that people should establish a relationship of trust with the recommendation engine, but also the engine must be able to adapt to the user needs. The second assumption made earlier, is the main reason we use the Big Five personality test so that the recommender "knows" and operates accordingly to the user's personality type.

2.3 The Big Five Personality Test

The Big Five personality test is a collection of 50 questions that follows the Big Five model theory, and when answered, it helps understand how you think and operate as also as how your personality is structured.

In [4], the authors present a study among Facebook users, where they find correlations between their Big Five Scoring and their preferences towards movies, TV shows, books and music. They come up with the table in Fig. 2, which is critical for our 50/50 Recommender System. Each row on the table is a vector, and the values for each cell are in the range of 1–5, and they represent the average score of the Big Five personality traits of the users who liked the corresponding genres.

3 The Proposed 50/50 Movie Recommender Methodology

In this section, we analyze the proposed methodology to incorporate personality into the Movie Recommender System. First, we present the data we used and an analysis on the basic operation of the 50/50 Recommender. Then we give an example with some results where we can notice the differences between the k-NN movie recommendations and our own 50/50 and 80/20 recommendations.

3.1 Dataset

The dataset we used is the MovieLens 100k database [5, 6]. It includes 100.000 ratings from 1000 users on 1700 movies. We also used the *Most rated movies table* which is a table that includes the most rated movies from the Movielens Dataset.

3.2 Main Flow Chart

Here, we present visually the operation of own formula.

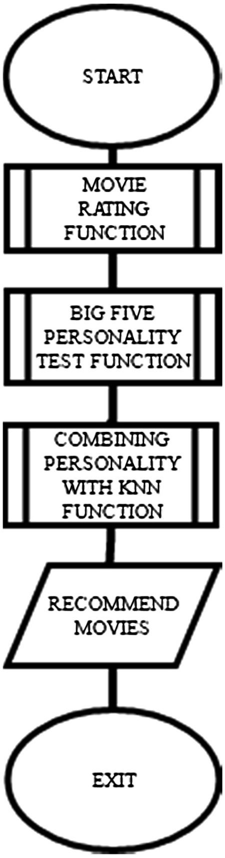

Fig. 1. Main flow chart

Movie Rating Function

This function is used to solve the cold start problem [7] meaning that the system knows nothing about the active user and there needs to be a way for the user to provide information to the system. It also includes the basic CF procedure explained in Sect. 2.1. So, first a list of 10 movies is presented to the user. Here we use the *Most rated movies* table, and we present one movie from this table and one random from the whole dataset alternately. That way, we avoid bias, because each time the user enters the system, the system provides a list with unique movies for him to rate. This goes on until the user has rated 20 movies, and we use these 20 reviews and CF to create a table with predicted ratings for each movie. We call this table *Predicted Ratings* (Table 1). The flow chart is shown in Fig. 1.

Table 1. *Predicted ratings* table head

Movie id	Rating
1	7.29552
2	2.87429
3	9.87429
4	2.87429
5	2.87429
6	2.87429
7	−2.34085

Big Five Personality Test Function

This function presents to the user all the questions from the Personality Test. The active user answers all the questions, and then the system calculates his big five traits scores as, following the procedure described in [1].

Combining Personality with k-NN Flow Chart

This is our main contribution. First, we take the Big Five Scoring of the active user and we subtract each of his traits from the corresponding trait of each genre in Fig. 2. Then we take the absolute value of these results and add them. That way, we convert them into a distance metric, so to calculate which genres the user prefers. The lower the value, the more the active user prefers this genre.

Next step is to rearrange the *Predicted Ratings* table, based on these genre preferences. For the 50/50, we take the predicted rating of the movie and divide it by 2, and we also take the number which represents the genre preference of the user and divide it by 2. If we add these numbers, we get a new predicted rating for the movie, where the k-NN scoring of the movie and his genre preferences count 50% each. For the 80/20, we take the predicted rating of the movie and multiply it by 0.2 and the genre preference of the user and multiply it by 0.8. That way, personality is now 80% and k-NN is 20%.

3.3 Example

The user enters the system and reviews 20 movies. CF is applied and the *Predicted Ratings* table is created. This is the most important table in our system. The following table (Table 1) is an example of the first 7 entries of the *Predicted Ratings* table.

The higher the value of the rating the most likely he will like this movie. This table (Table 1) is then rearranged in a descending order and has only the first 10 movies saved in a new table called the k-NN *Movie List*. Then the user is presented with the Big Five Personality test, and after he completes it he gets the results presented in Table 2.

Table 2. User's personality traits

	ope	con	ext	agr	neu
User	3.87	3.56	3.53	4.7	2.89

Based on these results, we check the following table (Fig. 2) from [4] and see which genres the active user prefers based upon his personality.

We always pick the genre that has the closest value to the user's Personality Traits score. For example, the active's user score on Agreeableness is 4.7, which as we can see is high and there is not a corresponding value to the table. That simply means that we will pick the greater value for this trait, which is 3.68 and corresponds to the Adventure genre.

Now, in the following table (Table 3), you can see the scoring for the action and adventure genre. We will use that as an example to show the subtractions we need to do to find the final distance metric between the user and each genre. This will lead us to his final genre preferences.

	All users				
MOVIE GENRE	OPE	CON	EXT	AGR	NEU
action	3.87	3.45	3.57	3.58	2.72
adventure	3.91	3.56	3.54	3.68	2.61
animation	4.04	3.22	3.26	3.35	3.02
cartoon	3.95	3.33	3.49	3.57	2.81
comedy	3.88	3.44	3.58	3.60	2.75
cult	4.27	3.10	3.45	3.40	3.16
drama	3.99	3.43	3.66	3.60	2.86
foreign	4.15	3.46	3.47	3.54	2.81
horror	3.90	3.38	3.52	3.47	2.91
independent	4.31	3.59	3.51	3.55	2.69
neo-noir	4.34	3.35	3.33	3.37	2.97
parody	4.13	3.36	3.35	3.28	2.73
romance	3.84	3.48	3.62	3.62	2.85
science fiction	3.99	3.55	3.33	3.57	2.73
tragedy	4.40	3.34	3.27	3.52	3.11
war	3.82	3.51	3.49	3.50	2.71

Fig. 2. Correlation between Genre and Big Five Traits

In the next table (Table 4), we only have to subtract each trait from Table 2 from the corresponding trait Table 3. We then take that absolute value of each result and add it so that we get a final number which is a distance metric. The closer to 0, the better the user likes this specific genre.

Table 3. Example genre trait scores

	ope	con	ext	agr	neu
Action	3.87	3.45	3.57	3.58	2.72
Adventure	3.91	3.56	3.54	3.68	2.61

Table 4. Calculating 50/50 2

	ope	con	ext	agr	neu	Pers. score
Action	|3.87–3.87|	|3.56–3.45|	|3.53–3.57|	|4.7–3.58|	|2.89–2.72|	1.44
Adventure	|3.87–3.91|	|3.56–3.56|	|3.53–3.54|	|4.7–3.68|	|2.89–2.61|	1.35

We can see that the user has a slight preference towards the adventure genre. We do the same calculations for all the genres and then create a table (Table 5) that includes these results in an ascending order. These are the genre preferences of the active user based upon his personality.

Next step is to rearrange the k-NN *Movie List* based on these genre preferences. Let's assume that the k-NN algorithm predicted that the active user will provide a rating of 4 on Star Wars and a rating of a 4 on Psycho. For the 50/50, our recommender takes the predicted rating of the Star Wars movie from *Predicted Ratings* table and divides it by 2. Because Star Wars is a Sci-Fi movie, it also takes the value of the sci-fi genre from Table 5 and divides it with 2. We then add these two numbers and the new pseudo rating for Start wars is (4/2) + (1/2) = 2.5. By applying the same concept to the Psycho movie, we can see that the new pseudo rating for the psycho movie is 2.715. The higher the value, the less you will like this movie.

Table 5. 50/50 user genre preferences

Cartoon	0.8
Sci-fi	1
Drama	1.3
Horror	1.43
Action	1.44
War	1.5

If we only had the k-NN suggestions, Psycho and Star Wars would be equally presented to the user as they both have the same predicted rating. However, with the addition of the personality factor, Star Wars will be suggested first and then Psycho. We follow the same procedure for the 80/20 recommender, with the only difference that now we multiply the k-NN predicted rating with 0.2 and the Personality genre rating with 0.8. On the following table, we can see the final recommendations (Table 6).

The k-NN recommendations list is the one that we get by simply applying CF. On the 50/50 list, we see the Sci-fi movies rising in the rankings and even horror movies like Alien and Psycho appeared in the list, where previously on the k-NN list, they

Table 6. Final Recommandations

k-NN recommendations	50/50 recommendations	80/20 recommendations
Usual suspect, The	Blade runner	Lion King, The
Blade runner	Silence of the lambs, The	Blade runner
Silence of the lambs, The	Empire strikes back, The	Silence of the lambs, The
Empire strikes back, The	Alien	Wallace and Gromit: best animations
Amadeus	Psycho	Wrong trousers, The
Schindler's list	Amadeus	L.A. Confidential
One flew over the cuckoo's	Schindler's list	Close shave, A
Casablanca	One flew over the cuckoo's	Maltese Falcon, The
It's a wonderful life	Casablanca	Beauty and the beast
Rear window	It's a wonderful life	Faust

weren't included. This is because the user has a preference on Sci-fi and Horror movies based on Table 5. We can clearly see that the list is now rearranged based on the user's personality.

On the 80/20 list, we see that a cartoon movie is now suggested first, because cartoon is the favorite genre of the active user, as we see again on Table 5. We also see the appearance of more cartoon movies.

On the next and final Section, we discuss the future work on our movie recommender.

4 Evaluation

For the evaluation part, we sent a modified version of the recommender system to 50 different people. 32 responded, but 2 did not follow the instructions, thus their responses were discarded. Out of these 30 people, 10 were female and 20 were male, aged between 18 and 65. The modified version differs from the final version of the recommender is that it did not reveal which recommendation list was produced by which method. Users were presented with three different lists of movies, and they had to rank those in order of preference, according to how well these lists reflected their actual preferences.

The users awarded their first preference 3 points, the second preference 2 points and the last preference, 1 point. They could award equal points to tie cases, i.e. if lists A and B were equally preferable they were both awarded 3 points each, just as they could both be awarded 2 points each, if they were both second best. There were 15 such ties. We then added the scores for each list to find the total number of points for each method, and the respective percentage of the total number of points.

As it can be seen in Table 7, top preferred method was the 50/50 with 36.92% of the total points, second came k-NN with 34.36%, while the 80/20 method received 28.72% of the points. This means that the 50/50 method outperforms k-NN, whilst the 80/20 method, despite being heavily biased towards personality, did not perform poorly. This confirms our hypothesis that taking personality into account would improve recommendation quality. Table 8 shows results if we remove the 80/20 method, leaving 50/50 with 51.8% and k-NN lagging with 48.2%, 3.6% less than 50/50.

Table 7. 50/50 and 80/20 vs. k-NN Evaluation

	50/50	k-NN	80/20
# Points	72	67	56
%	36.92	34.36	28.72

4.1 Discussion

Observing the evaluation results we can conclude that personality plays a significant part in recommender systems, as the 50/50 was the favorite method for most users and

Table 8. 50/50 vs. k-NN Evaluation

	50/50	k-NN
# Points	72	67
%	51.8	48.2

even the 80/20 fared well. It is possible that a different mix could give even better results. However, it requires a significant overhead initially, when users need to take the personality test, even though this only need to take place once and the results could be used in many subsequent recommendations, not necessarily only movie related. In any case, there is a tradeoff between investing time to take the test and improving subsequent recommendation experience.

Threats to Validity
Our approach could be subject to a few threats to validity. First and foremost was the selection bias in the evaluation phase. No matter how many people you target to participate to an evaluation like the one described previously, there are few safeguards that the sample is random and representative of various demographics.

Last, we cannot neglect the Hawthorne Effect [11, 12], where users might complete the personality test with dishonest answers as they know they are being observed. In the next section, we discuss some important improvements that can be done in the future.

5 Future Work

As we discussed in Sect. 4.1, the time needed to answer the personality test was a reason why some users dropped off. An improvement for our recommender would be to reduce the number of questions in the personality test to the ones that are most important. This can be achieved by using decision trees for classification to find questions that have the strongest impact on classification, (possibly the ones with the widest range of answers, which provided the highest information gain).

We also plan to explore different mixes of x% personality and (100 − x)% k-NN recommender, and potentially plot a performance graph for various x/100 − x spreads in order to better understand how personality affects the outcome and find the optimal balance. So far, we experimented with 0/100, 50/50 and 80/20 and noticed that the users' preference raises on the 50/50 and on the 80/20 there is a decrease. It would be interesting to see at which percentage exactly we have the peak of preference. An improvement would also be to include more metadata, which can be gathered from IMDB [8].

The MovieLens database includes also gender, age and occupation information for each user. In the future, these data could be used so to cluster the users and make more accurate predictions.

Acknowledgments. The authors would like to thank the Hellenic Artificial Intelligence Society (EETN) for covering part of their expenses to participate in EANN 2017.

References

1. Goldberg, L.R.: The development of markers for the Big-Five factor structure. Psychol. Assess. **4**(1), 26 (1992)
2. Burke, R.: Hybrid recommender systems: survey and experiments. User Model. User-Adap. Interact. **12**(4), 331–370 (2002)
3. McNee, S.M., Riedl, J., Konstan, J.A.: Making recommendations better: an analytic model for human-recommender interaction. In: CHI 2006 Extended Abstracts on Human Factors in Computing Systems (CHI EA 2006), pp. 1103–1108. ACM, New York (2006)
4. Cantador, I., Fernández-Tobías, I., Bellogín, A.: Relating personality types with user preferences in multiple entertainment domains. In: UMAP Workshops (2013)
5. MovieLens 100K Dataset: Grouplens. https://grouplens.org/datasets/movielens/100k/. Accessed 20 Apr 2017
6. MovieLens: Non-commercial, personalized movie recommendations. https://movielens.org/. Accessed 20 Apr 2017
7. Nadimi-Shahraki, M.-H., Bahadorpour, M.: Cold-start problem in collaborative recommender systems: efficient methods based on ask-to-rate technique. J. Comput. Inf. Technol. CIT **22**(2), 105–113 (2014)
8. IMDB. http://www.imdb.com/. Accessed 20 Apr 2017
9. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. **2009**, Article ID 421425 (2009). 19 p.
10. Melville, P., Sindhwani, V.: Recommender systems. In: Sammut, C., Webb, G.I. (eds.) Encyclopedia of Machine Learning, pp. 829–838. Springer US, New York (2011). doi:10.1007/978-0-387-30164-8_705
11. Monahan, T., Fisher, J.A.: Benefits of "observer effects": lessons from the field. Qual. Res. (QR) **10**(3), 357–376 (2010)
12. Gerogiannis, V.C., Karageorgos, A., Liu, L., Tjortjis, C.: Personalised fuzzy recommendation for high involvement products. In: IEEE International Conference Systems, Man, and Cybernetics (SMC 2013), pp. 4884–4890 (2013)
13. Hill, W., Stead, L., Rosenstein, M., Furnas, G.: Recommending and evaluating choices in a virtual community of use. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1995), pp. 194–201. ACM Press/Addison-Wesley Publishing Co. (1995)