Movie Recommendation System using MovieLens Dataset

Timothy Chu and Elliot Frost
Department of Computer Science
University of Rochester
Rochester, NY 14627
tchu5@u.rochester.edu
efrost3@u.rochester.edu

Abstract

Thousands of films exist, and more are made every year. To find a movie that one may like, they may need to spend hours upon hours watching various movies. Our solution to this problem is to utilize collaborative filtering, an algorithm utilized for recommender systems, to help users identify movies they may like based on their past preferences and ratings.

1. Introduction

For our project, we decided to tackle the issue of finding out how to connect people with movies they would enjoy based on their preferences. The reason for doing this is that thousands of films exist, and more are made every year. Although many movie watchers enjoy this new supply of films, it often means that they have to go through many movies, reading reviews and ratings until they find a movie that they enjoy. Thus our model looks to fix that problem by using collaborative filtering techniques to cut down on the time and effort needed to find a good movie.

2. Related Works

2.1. MovieLens unplugged: experiences with an occasionally connected recommender system

In this paper, the researchers over the course of nine months looked at recommender systems on mobile devices on an application called AvantGo. On this app, users could go on to MovieLens and rate movies, and get recommendations based on their ratings while going to a physical movie store or a movie theatre.

Much of the focus of this study centered around usability of this application, as at this time, the development of mobile applications were very much at their infancy. Over the course of the study which lasted nine months researchers soon realized that usage was steadily and consistently falling. This was largely due to the website being far more in depth in regards to information about the movies they wished to see or learn about. Another reason was that information gained from their usage of the app was not synchronized until the next time they used the app, thus reducing the quality of the user experience. User experience was further reduced by lack of recommendations. From this study, researchers learned that rather than having a simplified version of the website, users wanted the full system with constant updates and full access to all interactions regardless of whether they were using a desktop or mobile device.[4]

2.2. Exploring the Value of Personality in Predicting Rating Behaviors: A Study of Category Preferences on MovieLens

In this paper, the two facets of recommender systems, one being a social one and the other being algorithmic in nature is examined in relation to recommender systems. In past research, only minor (but statistically significant) effects have been shown when trying to introduce social/personality based estimations of preference. Meanwhile, mathematical algorithms have had a far greater success when it comes to predicting a persons preferences. The goal of this particular study was to have these two aspects compliment each other when it came to making movie recommendations.

The way this was done was through examination of two basic questions.

1. Does the magnitude of ratings across categories vary by personality type? If so, how?
2. Do the proportions of items consumed across categories vary by personality type? If so, how?

It was found that consumption of movies did vary by personality, but when it came to rating movies, only the trait of

agreeableness (in particular the magnitude of this trait) led to differences in in ratings.

It was found that personality played a big part in many other aspects of movie consumption and ratings. Personality was shown to be a major part of preferences in type of movie a user wanted, with people with different levels of personalities from highly neurotic to non-open minded having different tastes in movies ranging from comedy to drama. Additionally, the magnitudes of ratings changed, with some personalities having either very high or low scores, while others moderated their ratings. This research shows that personality is a critical part of the process of making a effective recommender system, and algorithms alone are not enough.[3]

3. Methods

3.1. Dataset

The repository by MovieLens provided the dataset for our experiments. MovieLens offered two different datasets, one "small" and one "full". The small dataset held 100,000 ratings and 3,600 tag applications applied to 9,000 movies by 600 users. The full dataset held 27,000,000 ratings and 1,100,000 tag applications applied to 58,000 movies by 280,000 users. This dataset includes four different files containing information pertaining to movies and ratings. Each movie gets a unique title and movielens id, and is assigned its categorical variable based on its related genre such as horror or comedy. Movies are also linked to its corresponding IMDB and TMDB id's. Each movie has a five star rating that users give each movie based upon a five star system as well as a timestamp for when the movie was rated. Users can also add tags to the movie they feel relevant such as sub-genres or general comments.

Given the fact that this experiment would be run locally, we chose to ignore the links file unless requested by the user through command line arguments. We did this because the imdbId and tmdbId variables only exist within the links file. The default Ids would change accordingly with where the project is being run.

Our team saw an opportunity to utilize natural language processing to generalize movie tags so that our recommender system could also predict tags a user might assign to a movie. However, this is outside the scope of this project. This would be an area of focus for further research. [1]

3.2. Experiments

With the code for the recommender system written, it was juat a matter of choosing a distance formula that yielded the "best" results.

We tested the following distance formulas with our user-based collaborative filtering recommender system:

- Pearson Correlation
- Cosine Similarity
- Euclidean Distance
- Square Euclidean Distance
- Bray-Curtis Distance
- Canberra distance

Once we've identified the best performing distance formulas, we had our system predict a user's expected ratings for movies using the "top three" formulas. The movies we chose to predict are movies that our user has already rated, giving us a concrete number to compare our performance to.

3.3. Testing Phase

For initial testing to decide which distance formulas should be used, we employed a cross-validation test routine to calculate the Root Mean Square Error (RMSE) for each distance measure. For this experiment, a lower RMSE meant a better result. We ran this experiment with the "full dataset" and it took several hours. The results are below. We also ran this experiment with the smaller dataset and it took approximately 15 minutes to finish running. Results were the about the same with slight variance in the hundredth and thousandth places.

Pearson RMSE:	Cosine RMSE:	Euclidean RMSE:
0.506	0.687	0.891

The table above shows the top three performing distance formulas with their associated Root Mean Square Error value. These three distance formulas are available for the user to use in code through command line arguments, however the code defaults to Pearson Correlation since it gave us the lowest Root Mean Square Error.[2]

We utilized the following formula to incorporate Pearson Correlation into our recommender system.

$$\mathrm{sim}(u,v) = \frac{\sum_{i \in I_u \cap I_v} (r_{ui}\mu_u)(r_{vi}\mu_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui}\mu_u)^2}\sqrt{\sum_{i \in I_u \cap I_v} (r_{vi}\mu_v)^2}}$$

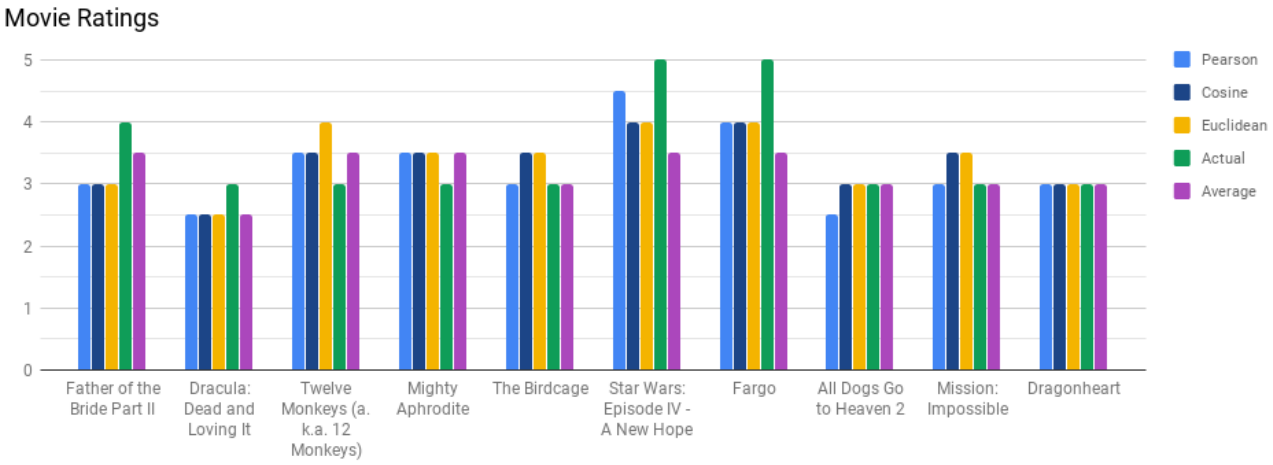

Figure 1:

Pearson Rating Accuracy:	Cosine Rating Accuracy:	Euclidean Rating Accuracy:
87.8%	85.2%	83.4%

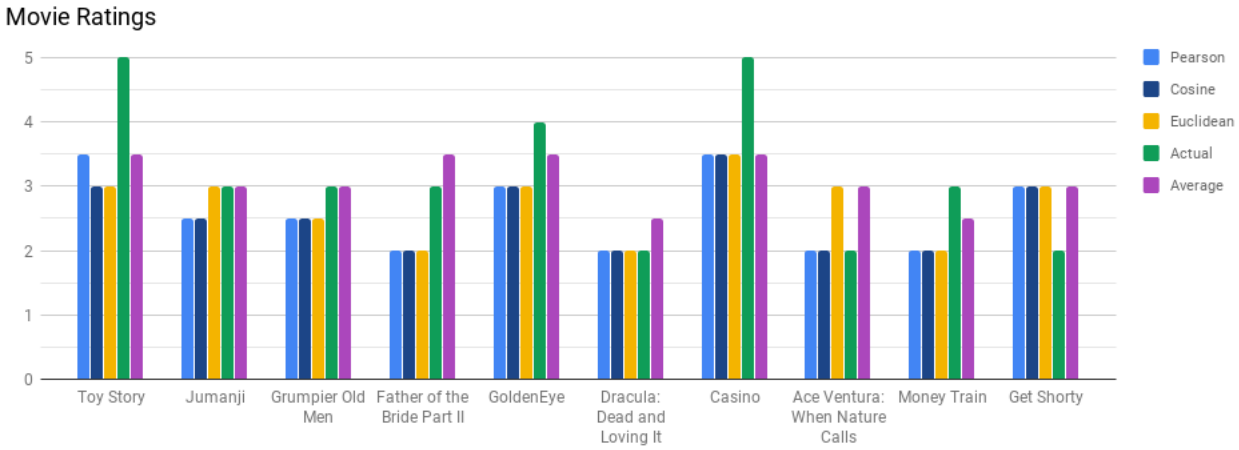

Figure 2:

Pearson Rating Accuracy:	Cosine Rating Accuracy:	Euclidean Rating Accuracy:
76.5%	75.5%	72.2%

4. Results

Figures 1 and 2 display the ratings generated by each of the three distance formulas, as well as the actual rating and the movie's average rating. In addition, the caption shows the approximate average. In both cases, using Pearson Correlation to calculate distance for our recommender system gave us the most accurate movie rating predictions when compared to the "actual" rating the user gave. In addition, Cosine Similarity gave us better results than Euclidean Distance, as expected based on the data from the Root Mean Square Error calculations.

5. Conclusion

Overall, we can say that our collaborative filtering based recommender system works to some degree with any type of distance formula with varying levels of success, with Pearson Correlation giving us the best results.

References

[1] Movielens, Oct 2016.

[2] HAN, J. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2012.

[3] KARUMUR, R. P., NGUYEN, T. T., AND KONSTAN, J. A. Exploring the value of personality in predicting rating behaviors: A study of category preferences on movielens. In *Proceedings of the 10th ACM Conference on Recommender Systems* (New York, NY, USA, 2016), RecSys '16, ACM, pp. 139–142.

[4] MILLER, B. N., ALBERT, I., LAM, S. K., KONSTAN, J. A., AND RIEDL, J. Movielens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of the 8th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2003), IUI '03, ACM, pp. 263–266.