Movie recommendation system

André Clerc, Aurélien Kinet, Jules Afresne, Jelena Simeunovic
A Network Tour of Data Science (EE-558), École polytechnique fédérale de Lausanne

Nowadays, an increasing number of streaming platforms and TV providers offer movie recommendations to users based on his or her preferences. These recommendations perform well, but, in this project, we attempt to achieve better accuracy by building a recommendation system using graph-based approaches. We will investigate its feasibility on the Movielens 100k Dataset. We will actually try to do it by clustering users in groups based on genres and ratings they give to previously watched movies.

I. INTRODUCTION

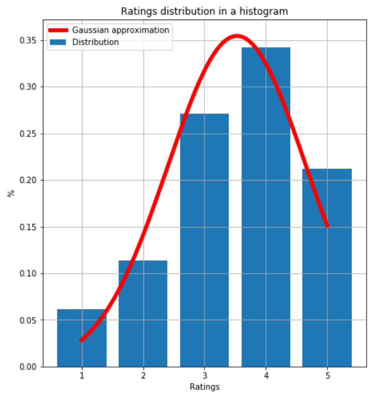
Recommender systems are increasingly important as the number of online streaming platforms grows. For our approach, we will focus on a movie recommendation system. More movies are being produced than ever before, so it is much more difficult for users to decide what to watch; however, it is similarly important for TV providers and advertisers to have accurate recommendation systems. On Netflix alone, people spend 9.45 billion hours looking for what to watch, which amounts to 20.75 minutes per user [2]. Since Netflix is an online platform which already has its own recommendation system, this leads us to the conclusion that this amount of time is much higher on platforms without any recommendations. In order to reduce this time, the recommender system should be improved so that users are more satisfied with their experience.

There are two main branches of recommender algorithms: content-based recommender systems and collaborative filtering models [1]. A content-based recommendation system will use metadata from users (e.g. occupation, gender, age) and from movies (e.g. genre). In contrast, collaborative filtering is based on matrix completion task, and it has been widely used. This approach uses bipartite graphs, where users and items (i.e. movies) are connected. The edge values in this graph are ratings.

II. DATA ACQUISITION

Our implementation will use Kaggle's Movielens 100k Dataset [3], and we will tackle the recommendation problem through a graph-based approach. As a result, we will initially explore both user data and movie data graphs to gain some insights. For each graph, the nodes will be the users and movies respectively, and the edges are derived based on the similarity of features. Afterwards, we will analyze a bipartite user-movie graph, in which the edges of the graph are user ratings, and eventually create our recommendation system. Finally, collaborative filtering will be performed and a Neural Network approach will be used as a benchmark for recommendation. The distribution of Average ratings, as shown in fig. 1, suggests that most users give a mixture of high and low marks, increasing the difficulty of this task.

Submitted for review on 10th of January 2020.



```
--> Average ratings :  3.52986 / 5
--> Standard deviation of the ratings :  1.125673599144316
--> Number of ratings :  100000
```

Fig. 1: Ratings distribution in a histogram

The Movielens Dataset consists of 100,000 ratings from 943 users on 1682 movies. Within the dataset, the user data includes several attributes (i.e. features) such as gender, occupation, age and zipcode, and these will later be used to create a user feature graph. However, to first gain a clearer understanding of the graph we will construct, we first create a plot of the user age distribution as shown in fig. 2.

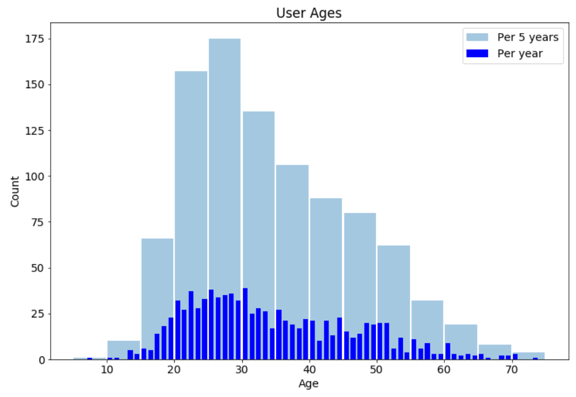


Fig. 2: Distribution of user ages

As well, it is also important to understand the distribution of user occupations in our dataset as seen in fig. 3. In this plot, it is clear to see that students are the most represented occupation in the dataset. Comparing this with fig. 2, this potentially explains the larger concentration of users between the ages of 20-35.

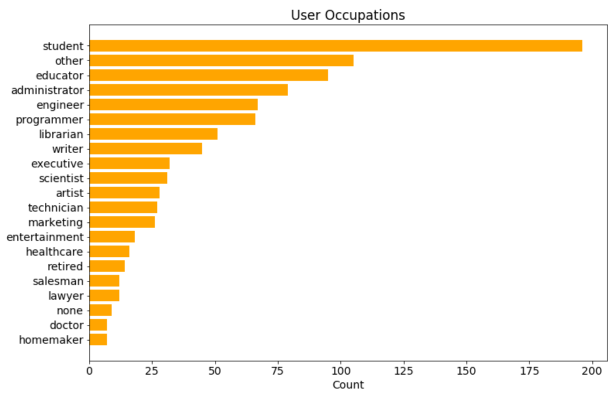


Fig. 3: Distribution of user occupations

III. Data Exploration

Our dataset provides readily usable data; however, a small amount of pre-processing is still required before creating the graphs. For example, the user attributes must first be converted from a textual representation to a numerical representation filled with 1s and 0s. Next, the user feature-graph can be constructed by creating a Euclidean distance matrix between all of the user features and using the resulting mean distance as a threshold to create an adjacency matrix for the graph. The resulting graph contains 943 nodes and a total of 153468 edges.

In analyzing the degree distribution of the users feature-graph seen in fig. 4, we find that it has a Poisson-like distribution, with the exception of some outliers on the right. The presence of these outliers typically indicates a random network, and in this type of network, we expect a low clustering coefficient. However, in our user graph, we computed an average clustering coefficient of $\langle C \rangle = 0.68$. In other words, $\langle C \rangle$ tells us that there is a 68 percent chance that two neighbors of a randomly selected user will also link to each other. Although this value is very high, a high clustering coefficient is more typical for real networks. Therefore, this network could be modeled with Watts-Strogatz model (i.e. small world phenomena).

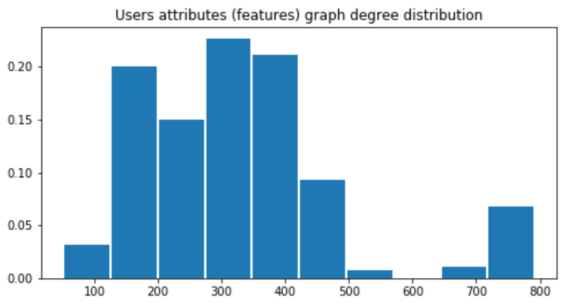


Fig. 4: User degree distribution

Secondly, a movies graph was created by using genres as our features. Other associated features such as the movie title, video release date and IMDb URL were removed for simplicity. Furthermore, we added an average rating feature for each movie based on the mean of users ratings for each particular movie. As we can see, the most popular and rated movies are from genres Drama and Comedy. Furthermore, the degree distribution of features for the movie graph indicates that most movies (i.e. 65 percent) have evenly spread attributes, while there are only a few with connections to almost all other features. This feature degree distribution is right skewed with long tail, known as power low distribution. Therefore, it is a clear indication of a scale-free network. Taking that into account, a Barabási-Albert network represents a good model, as it generates scale free networks.

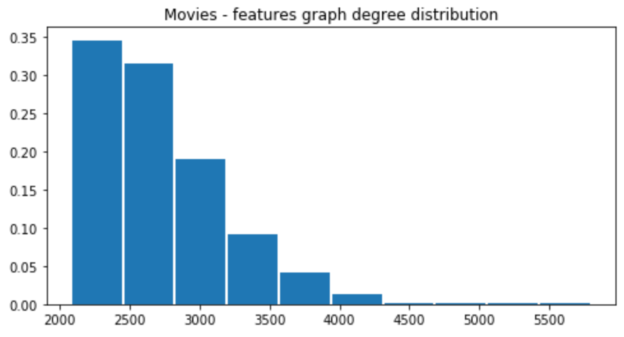


Fig. 5: Movies feature graph degree distribution

Next, a movie graph will be created based on similarity among genres of movies and average ratings, since each movie has been categorized with several genres. This is done by creating an epsilon similarity graph, using Euclidean distance among features. This graph is connected (i.e. number of connected components is 1). The adjacency matrix is not too sparse, as making it more sparse would make the number of unconnected components quite high. This graph has a power low distribution with small outliers which indicates the presence of hubs. If we try to make this matrix more sparse or remove outliers, we will remove nodes with most of connections making this graph disconnected.

IV. Data Exploitation

Before attempting our clustering approach, we created a movie-user graph and explored its properties. Namely, we classified users according to movies that they have rated in order to find what they would like to watch. After several attempts using a K-means algorithm, we determined that this algorithm was not very efficient for this kind of graph. Therefore, we decided to create another, more concise, graph, as we thought that clustering would be easier to achieve in this case. For the creation of this graph, we will do some additional feature processing.

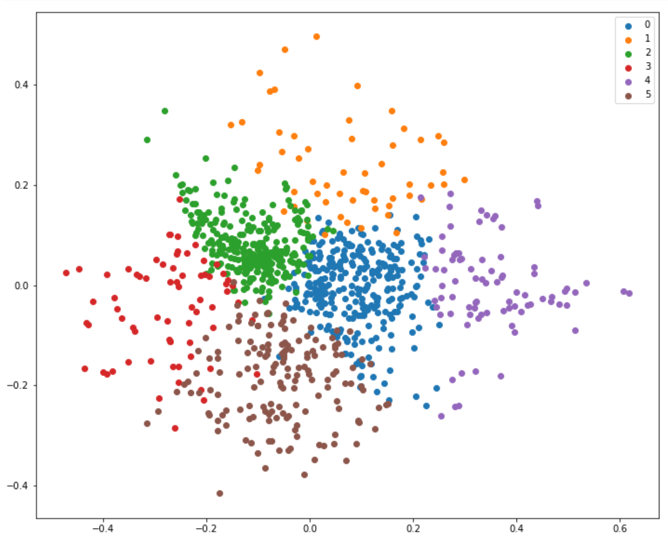


Fig. 6: K-mean clustering, visualization made with Isomap

For every genre, we found a total rating given per user according to the given movie ratings and genres of those movies. Thanks to this processing, we created another graph called category-user. We then performed K-means clustering once again, and by using the elbow method, we decided to focus on six clusters. Afterwards, we performed dimensionality reduction on the graph through Isometric mapping in order to visualize the clustering (represented through Isomaps). The resulting clusters are shown in fig. 6. Furthermore, due to the visualization in fig. 6 and the distribution of users among the clusters shown in fig. 7, we can observe that the clusters are well-defined and that most users are grouped in the fourth and fifth clusters. In addition, we can visualize these users preferences towards certain genres in fig. 8. The figure represents the ratio of ratings per category in each cluster.

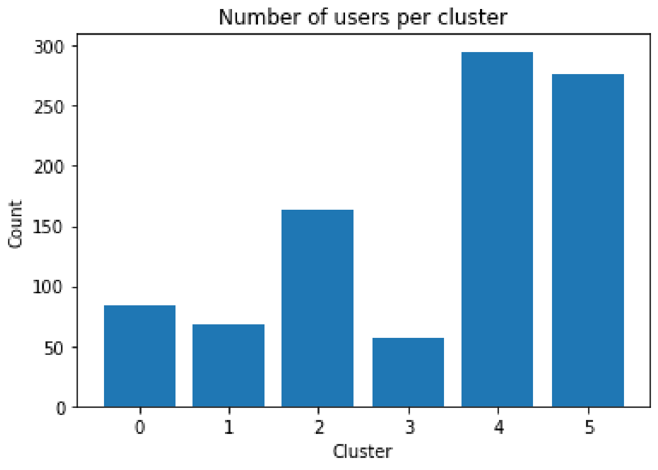


Fig. 7: Size of clusters.

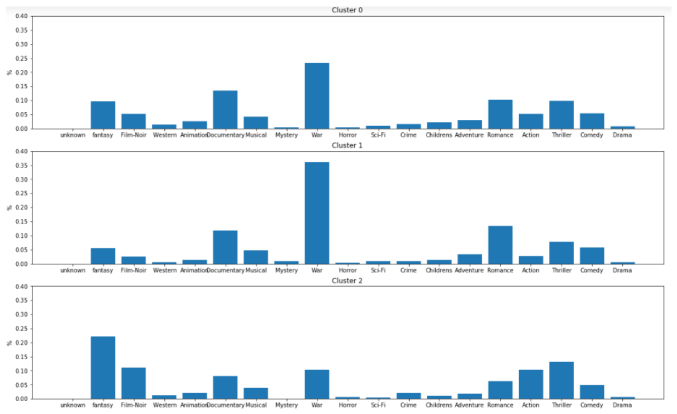


Fig. 8: Example of clusters with genre preferences

We have made visualisations of the number of users in each cluster, as well as movies that they have watched. Furthermore, we have created several different visualisations based on this clustering and found ratings for each cluster as well. One particularly interesting result is the distribution of the ratio of users that have watched a certain movie fig. 9. If we take, for example, the movie with id number 26, the ratio of users that have watched that movie is highest cluster 3. Movie 26 is an action movie, without any other genre combination, and the majority of people in that cluster give the highest ratings to action movies compared to other clusters.

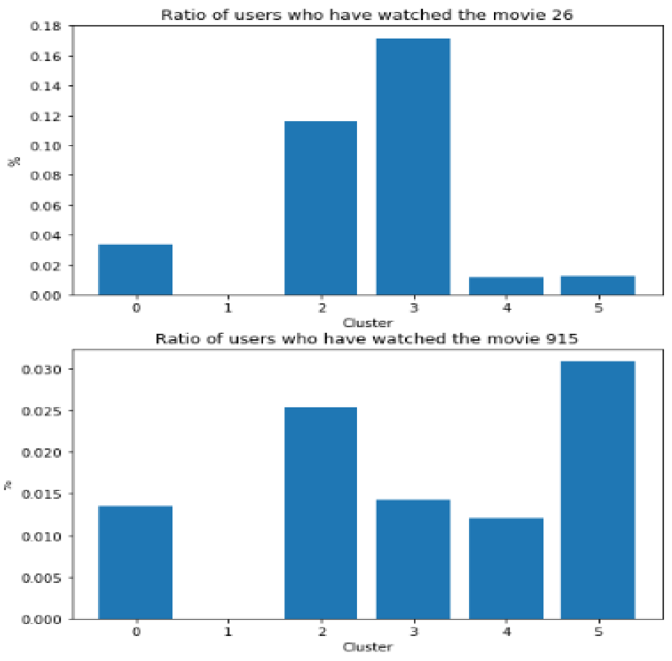
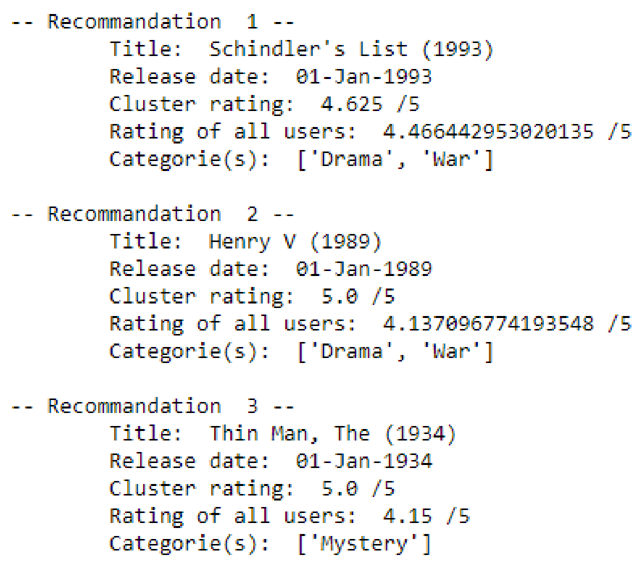


Fig. 9: Example of ratio of users per movie

Finally, we have implemented our recommendation system based on the group in which we have clustered a user. The first step was determining a user's specific cluster based on K-mean clustering. Next, we found all movies that the user watched along with the movies that the user did not watch, but that were related to its group. Then, we defined a score for each of the unwatched movies, taking into account the average rating of cluster and the average rating of other users concerning the movie. After all possible movies are rated, we offer the movies with the best score. Our system can choose how many suggestions the recommendation system provides for a certain user group. We have tested our code, and as an example, the first three recommendations for User 189 are shown below in fig. 10.



```
-- Recommandation  1 --
        Title:  Schindler's List (1993)
        Release date:  01-Jan-1993
        Cluster rating:  4.625 /5
        Rating of all users:  4.466442953020135 /5
        Categorie(s):  ['Drama', 'War']

-- Recommandation  2 --
        Title:  Henry V (1989)
        Release date:  01-Jan-1989
        Cluster rating:  5.0 /5
        Rating of all users:  4.137096774193548 /5
        Categorie(s):  ['Drama', 'War']

-- Recommandation  3 --
        Title:  Thin Man, The (1934)
        Release date:  01-Jan-1934
        Cluster rating:  5.0 /5
        Rating of all users:  4.15 /5
        Categorie(s):  ['Mystery']
```

Fig. 10: Recommendation for User 189

We also tried a model Collaborative filtering based recommender[5]. In this approach we have used Low rank Matrix Factorization. First we used Keras Embedding and then Neural Network for recommendation and prediction of rate. We have compared findings from our graph-group-based model against collaborative filtering model. For user 189 and first three movies that our graph-based model has recommended, collaborative filtering give following results: ”Schindler’s List (1993)” predicted rating: 3.91, ”Henry V (1989)” predicted rating: 3.26 and ”Thin Man, The (1934)” predicted rating: 1.74. This is showing that our graph-based recommendation system give comparable results with collaborative filtering model that used Neural Networks. Although, we were satisfied with results, there is a place for improvement.

CONCLUSION

Our findings in regards to users’ preferences towards certain genres leads to interesting conclusions and other potential opportunities. For example, one might imagine that future movies could be produced by targeting a majority of users through a set of strategic genre combinations. In addition, this concept could be extended to countless other applications where a user must make a choice from a number of possibilities.

Our graph-based recommendation system gives comparable results with the baseline collaborative filtering model that has been widely used in recommendation systems today. Possible hindrances in this approach is that if a user is incorrectly classified due to noise or error in a certain cluster, all recommendations for the user will be affected. Thus, the recommendation will be biased towards certain cluster groups. On the other hand, it is possible to explore and improve this (clustering) graph-based method by using Graph Convolution Neural Network.

REFERENCES

[1] Rianne van den Berg, Thomas N. Kipf, Max Welling, “Graph Convolutional Matrix Completion”

[2] How much time is wasted worldwide on people looking for movies to watch on Netflix (annually)?, https://askwonder.com/research/time-wasted-worldwide-people-looking-movies-watch-netflix-annually-9l4sd364g

[3] F. Maxwell Harper and Joseph A. Konstan. 2015. ”The MovieLens Datasets: History and Context.”*ACM Transactions on Interactive Intelligent Systems* (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872

[4] Albert-László Barabási, ”Netork Science”,Northeastern University, Section 3.8. Small Worlds, http://networksciencebook.com/

[5] Collaborative-Filtering-Based-Recommender-Systems, Kaggle, https://www.kaggle.com/rajmehra03/cf-based-recsys-by-low-rank-matrix-factorization