Movie recommender system using signal diffusion

Deniz Ira, Jonathan Labhard, Daniil Dmitriev, Paul Griesser
Department of Data Science, EPFL, Switzerland

Story

Recommendation systems are of a great popularity nowadays with the spread of online shops, advertising and streaming platform. Classical methods of movie recommendation rely on matrix completion methods, such as matrix factorization. While they exhibit a strong performance, usually there is no simple way to incorporate explicitly the relations between the users or the movies. Amount of such structured data is enormous nowadays and this drives the need of exploring novel ways to build a graph based recommendation system which is the goal of our project.

Acquisition

The data can be downloaded from the official website in the form of three tables: the first one contains the user information, the second one contains the features of the movies and the third table provides the ratings which links the users with movies. From this table data we have multiple ways to create a graph structure.

1) **Bipartite graph.** We can create a bipartite graph directly out of the ratings, where the weighted edge between the user and the film is equal to the given rating.
2) **Disjoint graphs.** Another way is only to use the user-specific data to create a user graph (e.g. connect by the same job/age) and the movie-specific data to create a movie graph (e.g. connect by genre of the film)
3) **Joint graphs.** The most general way is to use both the specific data and the ranking that connect users with films. We can deduct that two users have similar preferences not only by their age and occupation, but also by the rating that they gave to different movies.

Some recent work use the bipartite graph structure as the base for their methods. However, since bipartite graphs do not model user-user or movie-movie connections, we decided to focus on the joint graphs approach where we can better explore the inner structures in the data.
For the moment, we focus only on the tables containing movie features and user movie ratings tables in order to create some graphs to analyse. We have two sorts of graphs, one where the nodes are the users and one where the nodes are the movies. For the user graph, the edges connecting two nodes is the combination of:

1) The Euclidean distance of all the ratings of movies that a given pair of users have in common.
2) The Jaccard distance of all the movies a pair of users have watched.

We chose to combine these two metrics because it is important to consider the similarity in movies that two users watch as well as how the ratings change for a pair of users.
For the movie graph, the edges connecting two movies are computed with the same metrics: the euclidean distance in ratings for all the users that rated the movie as well as the Jaccard similarity of the users that rated the movie.

Exploration

We threshold our graphs weights in order to have a single connected component, this necessity is explained in more details in the next section. The graphs are quite dense, since a lot of people have watched similar movies, and a single connected component combined with that generates a dense network.

Exploitation

To recommend movies to a user, we will learn how this user would rate all movies and then output the movies that he didn't see that have highest learnt rating. We will use a signal on our movie graph. For each movie node, the signal value will be the rating of that user for that movie, and zero if the movie hasn't been rated. So for each different user we will have a different signal. Diffusing the signal to the nodes that have a zero value in a smooth manner correspond then to infer the ratings of all non-rated movie for a user. We will also do it on the user graph, and the signals will correspond to the movies. In order to learn the ratings for all movies we need a connected component. Otherwise if we have an isolate component where the signal value is zero for each node, no diffusion can take place.
The underlying assumption is that the final signal should be globally smooth on the graph, that is, movies connected by strong edge weights tend to have similar ratings. Respectively, users strongly connected tend to give similar rating for a particular movie.

Fig. 1. Movies graph using gephi