Large-Scale Data Mining: Models and Algorithms
ECE 232E Summer 2018
Prof. Vwani Roychowdhury
UCLA, Department of ECE

Project 5
IMDb Mining

Due on Monday, Sept. 3, 2018 by 11:59 PM PDT

In this project, we will study the various properties of Internet Movie Database (IMDb). In the first part of the project, we will explore the properties of a directed actor/actress network. In the second part of the project, we will explore the properties of an undirected movie network.

1. Actor/Actress network

In this part of the project, we will create the network using the data from the following text files:

- **actor_movies.txt**
- **actress_movies.txt**

The text files can be downloaded from the following link:
https://ucla.box.com/s/z45q3g5zrpay8b8gtbql6ojaecb7kj2u

In order to create the network in a consistent manner, you will need to do some data preprocessing. The preprocessing consists of 2 parts:

1. Merging the two text files into one and then removing the actor/actress who has acted in less than 10 movies
2. Cleaning the merged text file

The cleaning part is necessary to avoid inconsistency in the network creation. If you analyze the merged text file, then you will observe that same movie might be counted multiple times due to the role of the actor/actress in that movie. For example, we might have

- Movie X (voice)
- Movie X (as uncredited)

If you don't clean the merged text file, then Movie X (voice) and Movie X (as uncredited) will be considered as different movies. Therefore, you will need to perform some cleaning operations to remove inconsistencies of various types.

QUESTION 1: Perform the preprocessing on the two text files and report the total number of actors and actresses and total number of unique movies that these actors and actresses have acted in.

1. Directed actor/actress network creation

We will use the processed text file to create the directed actor/actress network. The nodes of the network are the actor/actress and there are weighted edges between the nodes in the network. The weights of the edges are given by equation 1

$$w_{i \to j} = \frac{|S_i \cap S_j|}{|S_i|} \tag{1}$$

where S_i is the set of movies in which actor/actress v_i has acted in and S_j is the set of movies in which actor/actress v_j has acted in.

QUESTION 2: Create a weighted directed actor/actress network using the processed text file and equation 1. Plot the in-degree distribution of the actor/actress network. Briefly comment on the in-degree distribution.

2. Actor pairings

In this section, we will try to find the pairings between actors. We will consider the following 10 actors:

- Tom Cruise
- Emma Watson (II)
- George Clooney
- Tom Hanks
- Dwayne Johnson (I)
- Johnny Depp
- Will Smith (I)
- Meryl Streep
- Leonardo DiCaprio
- Brad Pitt

QUESTION 3: Design a simple algorithm to find the actor pairings. To be specific, your algorithm should take as input one of the actors listed above and should return the name of the actor with whom the input actor prefers to work the most. Run your algorithm for the actors listed above and report the actor names returned by your algorithm. Also for each pair, report the (input actor, output actor) edge weight. Does all the actor pairing make sense?

3. Actor rankings

In this section, we will extract the top 10 actor/actress from the network.

QUESTION 4: Use Google's PageRank algorithm to find the top 10 actor/actress in the network. Report the top 10 actor/actress and also the number of movies and the in-degree of each of the actor/actress in the top 10 list. Does the top 10 list have any actor/actress listed in the previous section? If it does not have any of the actor/actress listed in the previous section, please provide an explanation for this phenomenon.

QUESTION 5: Report the PageRank scores of the actor/actress listed in the previous section. Also, report the number of movies each of these actor/actress have acted in and also their in-degree.

2. Movie network

In this part, we will create an undirected movie network and then explore the various structural properties of the network.

1. Undirected movie network creation

We will use the processed text files from the previous section to create the movie network. The nodes of the network are the movies and there are weighted edges between the nodes in the network. To reduce the size of the network, we will only consider movies that has at least 5 actor/actress in it. The weights of the edges are given by equation 2

$$w_{i\to j} = \frac{|A_i \cap A_j|}{|A_i \cup A_j|} \tag{2}$$

where A_i is the set of actors in movie v_i and A_j is the set of actors in movie v_j. Since,

$$w_{i\to j} = w_{j\to i}$$

so we have an undirected network.

QUESTION 6: Create a weighted undirected movie network using equation 2. Plot the degree distribution of the movie network. Briefly comment on the degree distribution.

2. Communities in the movie network

In this part, we will extract the communities in the movie network and explore their relationship with the movie genre. For this part you will need to load the **movie_genre.txt** file.

QUESTION 7: Use the Fast Greedy community detection algorithm to find the communities in the movie network. Pick 10 communities and for each community plot the distribution of the genres of the movies in the community.

QUESTION 8:

8.1: In each community determine the most dominant genre based simply on frequency counts. Which generes tend to be the most frequent dominant ones across communities and why?

8.2: In each community, for the i^{th} genre assign a score of $\ln(c(i)) \times \frac{p(i)}{q(i)}$ where: $c(i)$ is the number of movies belonging to genre i in the community; $p(i)$ is the fraction of genre i movies in the community, and $q(i)$ is the fraction of genre i movies in the entire data set. Now determine the most dominant genre in each communitiy based on the modified scores. What are your findings and how do they differ from the results in 8.1.

8.3: Find a community of movies that has size between 10 and 20. Determine all the actors who acted in these movies and plot the corresponding bipartite graph (i.e. restricted to these particular movies and actors). Determine three most important actors and explain how they help form the community. Is there a correlation between these actors and the dominant genres you found for this community in 8.1 and 8.2.

3. Neighborhood analysis of movies

In this part of the project, you will need to load the **movie_rating.txt** file and we will explore the neighborhood of the following 3 movies:

- Batman v Superman: Dawn of Justice (2016); Rating: 6.6
- Mission: Impossible - Rogue Nation (2015); Rating: 7.4
- Minions (2015); Rating: 6.4

QUESTION 9: For each of the movies listed above, extract it's neighbors and plot the distribution of the available ratings of the movies in the neighborhood. Is the average rating of the movies in the neighborhood similar to the rating of the movie whose neighbors have been extracted? In this question, you should have 3 plots.

QUESTION 10: Repeat question 10, but now restrict the neighborhood to consist of movies from the same community. Is there a better match between the average rating of the movies in the restricted neighborhood and the rating of the movie whose neighbors have been extracted. In this question, you should have 3 plots.

QUESTION 11: For each of the movies listed above, extract it's top 5 neighbors and also report the community membership of the top 5 neighbors. In this question, the sorting is done based on the edge weights.

4. Predicting ratings of movies

In this part of the project, we will explore various rating prediction techniques to predict the ratings of the following 3 movies:

- Batman v Superman: Dawn of Justice (2016)
- Mission: Impossible - Rogue Nation (2015)
- Minions (2015)

QUESTION 12: Train a regression model to predict the ratings of movies: for the training set you can pick any subset of movies with available ratings as the target variables; you have to specify the exact feature set that you use to train the regression model and report the root mean squared error (RMSE). Now use this trained model to predict the ratings of the 3 movies listed above (which obviously should not be included in your training data).

We will now predict the ratings of the movies using a different approach. To be specific, we will use a bipartite graph approach for rating prediction. In a bipartite graph, $G = (V, E)$, we have a partition of the vertex set such that

$$V_1 \cup V_2 = V$$
$$V_1 \cap V_2 = \varnothing$$

and

$$e_{ij} = (v_i, v_j)$$

where $v_i \in V_1$ and $v_j \in V_2$. In a bipartite graph, vertices belonging to the same partitioned set are non-adjacent.
In this project, we will create a bipartite graph in the following manner:

- V_1 represents the set of actor/actresses
- V_2 represents the set of movies
- There is an edge e_{ij} between a node in V_1 and V_2 if the actor i has acted in movie j

QUESTION 13: Create a bipartite graph following the procedure described above. Determine and justify a metric for assigning a weight to each actor. Then, predict the ratings of the 3 movies using the weights of the actors in the bipartite graph. Report the RMSE. Is this rating mechanism better than the one in Question 12? Justify your answer.