Movie Mining

Xueyuan Mei[1] and Ziyi Yang[2]

[1]Department of Computer Science, Stanford University
[2]Department of Electrical Engineering, Stanford University

December 12, 2016

Abstract

In this project, we conducted social network analysis of movies. Firstly, We constructed graph database of more than 650 movies. Furthermore, we extracted signature vector of each movie and develop an algorithm to decide similarity between movies. Thirdly, for each type of movies, we generated a representative average-graph. Finally, we classify movie graphs into distinct clusters, via Principal Component Analysis (PCA).

1 Introduction

One interesting view of movies is to address them as social networks. Usually, movies are appreciated in aspect of pictures, lines and characters, and here we'd like to explore its social properties and it might provide us an unique insight to explore formula and features of movies. In a movie graph, each node is character and an edge between two nodes means that these character appear in the same scene. Obviously, the most convenient way to construct a graph is by parsing the movie script. Moviegalaxies[1] is a fantastic project of mapping movies into social network in such way of more than 650 movies.

2 Related Work

For graphs of various movies, one of the most important aspects that we are interested in is how similar they are. The similarity of two graphs can be determined by many features such as average number of degree, size of weakly connected component, clustering coefficient of each node and so on. [2] and [3] proposed a solution to the problem of finding similarity between two networks (graphs). The algorithm is that instead of computing the similarity of each factor separately, we can extract all the features of each network as a "signature" vector and compute the similarity score based on the "signature" vector. In [4], researcher used maximum common edge sub-graph to measure similarity.

We also want to produce average graphs of different movie styles. One challenge in networks domain is that how to generate a graph with prescribed properties? In [5], author proposed an rejection-sample algorithm to achieve this: generated a graph at random and rewired the graph until it met parameters requirements. Besides from this method, another common way to generate graph is calculating the probabilities of edges between two nodes [6].

Classification of graphs is another interesting topic in graph data mining domain. [7] used topological and label attributes to classify graphs. [8] employed pattern recognition algorithm for match networks.

3 Movie Database Establishment

3.1 Fetch Raw Data From Web

The graph data produced by Movieglaxies is in EXtensible Markup Language (XML). And Movieglaxies doesn't provide ensemble download of graph data. So that the first step to collect graph data

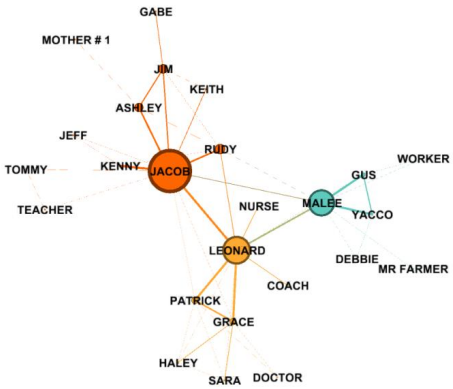

Figure 1: The graph of movie "Twelve and Holding".

is to fetch XML from Movieglaxies website. We use python and BeautifulSoup and urllib library to do this.

3.2 Parse the XML

After fetching XML document from Movieglaxies server, the next step is to extract nodes and edges information from XML files and save them as undirected graphs. The XML provided by Movieglaxies contain plenty information like the node's name, etc. The most fundamental aspect of graph: nodes and edges is saved into txt document as the same as SNAP convention, and the rest of information, like computed yet properties by Moviegalaxies in XML is left out for now.

Now we have the graph of one movie. Fig.1 show the graph of characters in the movie "Twelve and Holding". (Visualized by Gephi [9]) Each node is labeled with the name of characters. We can see that the leading actor Jacob is in the center of the graph and has the highest degree. (The visualization is achieved by Gephi). So far, we demonstrate that we have successfully built the database of movie graph.

4 Signature Vector

In this section, we will produce the signature vector of a graph. Also, these feature vectors reveal some basic properties of movie networks. To explore the distinctions between real-world graphs and movie graphs Compare them with those of the benchmark social networks, Erdös-Renyi, small world graph and two real world network: Real-World Collaboration Network and IMDB actor network (from homework 1). It is demonstrated that movie network share some common features with each other and distinct from other classics types of graphs.

4.1 The Size of Graph

Unlike other real world graphs (notice that movie graph is also artificial and virtual graph, but we still consider the network "real-world"), the size of movie graphs tends to be much smaller.

The table 1 list the size information of movie graphs. It shows that the on average there are 30 nodes in a movie and 100 edges (approximately in the same degree as the number of scenes in a movie). It also demonstrates that the movie graph is actually quite density graphs, and average density is around 0.2, which is much larger than other graphs. this indicates that nodes (or character) appeared in the graph (movie) is more likely to have connection than in real world. This can be explained as that in movie, characters are very likely to interact with other and an inactive character won't even exists in a movie, since there will not be any stories or scenes about it.

4.2 Cluster Coefficient

Cluster coefficient measures the degree the nodes tend to cluster together. The table 2 list cluster coefficient (average for movie graph) of different graphs.The movie graph has significantly bigger cluster coefficient than other network. Again, it points out that nodes actively interact with each other in movie graph.

4.3 Degree Distribution

In Fig.2, we plot the average degree of different types of movies. And different movies are very different: the adventure movie have highest average degree, while the mystery movie have the lowest one. In class, we have seen that many real world obeys power-law degree distribution. Let's see if these real-world graph also have such property. We plot the degree distribution of movie "The Crow" (Fig.3):

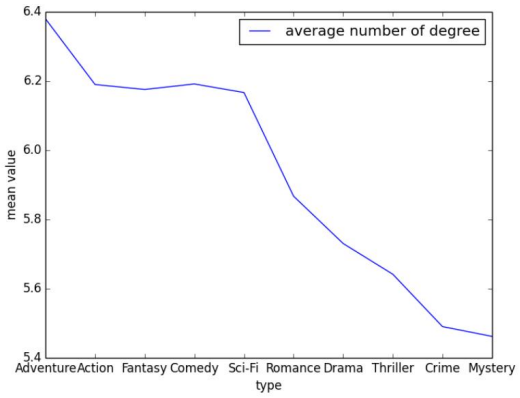

Figure 2: Average degree of different types of movies.

From the curve, we can tell that the degree probability scatters along a straight line, and from CCDF fitting, we obtain the α is about 1.5. Although movies are artificial art, it still maintains the power-law property like other real-world graphs. This is not surprising, since the movie is a mimics of life. And several leading roles are nodes with high degree in the tail of degree curves.

4.4 Diameter

Diameter describes the connectivity and size of graph. Since movie graphs are relatively smaller size, here we use the measure "diameter/number of nodes" to represent normalized diameter. Table 3 shows normalized diameter of 5 different graphs. Surprisingly, the movie have much larger diameter than other benchmark graphs, even though it has higher cluster coefficients. One explanation we come up is that the path linking supporting roles in a movie are very long.

Size	mean	max	min	Standard Deviation
nodes	36.511	109	5	15.1
edges	113.48	629	10	70.619

Table 1: Size Statistics of Movie Graphs.

Cluster Coefficient	Movie Network	Erdös-Renyi	Small World	Collaboration Network	IMDB actor
nodes	0.7082	0.0018	0.2855	0.5304	0.3390

Table 2: Cluster Coefficient of different Graphs.

Diameter	Movie Network	Erdös-Renyi	Small World	Collaboration Network	IMDB actor
hops	0.0503	0.00108	0.00141	0.000338	0.00267

Table 3: Diameter (normalized) of different Graphs.

4.5 Community Structure

Community is another important property of graph. In different network, the community detected can be interpreted in many different ways:for example, the community of citation network can be viewed as academic fields, since researchers from similar fields are more likely to cite each other's work.

Here, we believe that the community detected here is the scene. Since the the edge in movie network means that tow nodes (actors) co-appear in the same scene. The Fig.4 is produced by plotting the average normalized number of community for each movie type. The value is normalized by the number of nodes in each graph. As we can see from the Fig. 4, The number of community varies very much for different movie type. For Adventure and Fantasy movies, there tend to be more communities, while for Sci-Fi and Action, communities are few.

4.6 Motifs

Another topological property we are interested in is motif. We take 5 kinds of motifs into consideration (Fig. 5). 2 are of three nodes, in (a) one pair of nodes are disconnected, and in (b) 3 nodes are fully connected. 3 are of four nodes, (c) have no diagonals, (d) with only 1 diagonals and (e) fully connected. Here we count these 5 types of motifs in graph of Pulp Fiction and a randomly

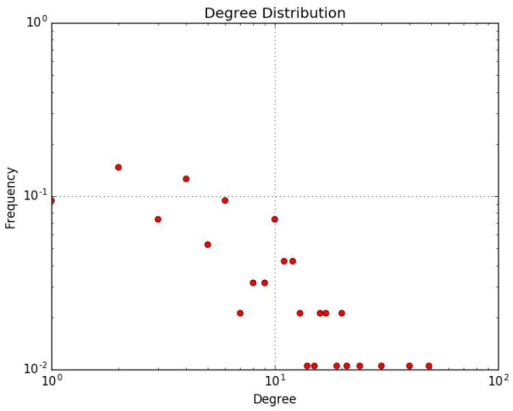

Figure 3: The power law distribution of movie "The Crow".

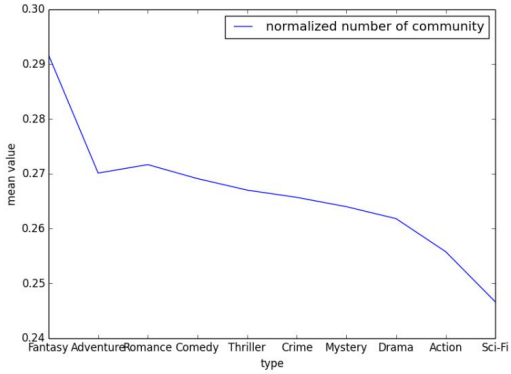

Figure 4: normalized number of community by movie types.

Motifs	type(a)	type(b)	type(c)	type(d)	type(e)
Pulp Fiction	262	119	229	375	243
Random Graph	221	25	189	51	3

Table 4: Count of motifs in pulp Fiction and a random graph

generated graph with the same number of nodes and edges. The results in table in table 4 indicate that the topological structure in movie graph is quite different from random graph. The small number of motifs with complicated structure in random graph indicates the naive geometry of random graph. And it also reflects the complicated regional sub-network in movies.

5 Movie Similarity

We want to decide which movies are similar based on signature vectors. The detailed algorithm is described in 1. The input is the the signature vector of a movie and the output is the most similar movie.

Algorithm 1 similarity algorithm

1: **for** each movie graph's signature vector s_1 **do**
2: Normalize by the maximum of all movies
3: **for** each other movie graph's signature vector s_w **do**
4: Compute euclidean distance between s_1 and s_2
return the movie with smallest euclidean distance

Generally speaking, we pick the movie with the smallest euclidean distance to the target movie as the most similar one. Some of matching pair are shown in table 5. One interesting finding is that the most similar movie of "Lord of Rings (LoR): The Fellowship of Rings" is "Mission Impossible II", while the most similar movie to "LoR: The Return of the King" is "Mission Impossible I". In Fig.6, we present the graph of Mission Impossible 2 (Fig.6(a)) and The Fellowship of Rings (Fig.6(b)). Those graphs look pretty similar to each other: high cluster coefficients, high densities, similar graph size and multiple major modes.

The Pulp Fiction is most similar to Go. In Fig.7, we also present their graphs. It is obvious that these two graphs are of similar size and closed cluster coefficients. And they are both crime and thriller movies.

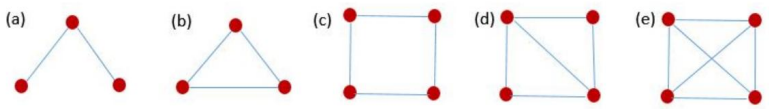

Figure 5: 5 types of motifs

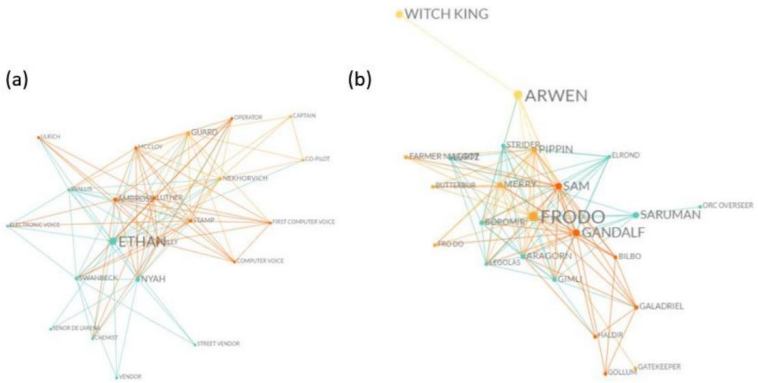

Figure 6: (a): Graph of Mission Impossible II. (b): Graph of LoR I

6 Average Graph

In this section, we introduced an algorithm to produce representative graph for each genre of movies. The idea is calculating the mean degree distribution, mean nodes and mean edges, mean μ_2 (the second eigenvalue of the Laplacian matrix) of certain types of movies. And then with these prescribed parameters, we can generate an average graph using an algorithm we developed.

It is easy to generate a graph with given number of nodes, given number of edges obeying prescribed degree distribution. Think of nodes as a point with branches sticking out. And the number of branches of is the same with node degree. And then connect nodes accordance with these branches.The reason why we pick the second eigenvalue of the Laplacian matrix μ_2 as another parameter is that it represents some overall, intrinsic and latent properties.

To generate a graph with prescribed μ_2, we switch two edge e_1 and e_2 (if switchable) in the graph until its μ_2 reach a range of target μ_2. Assume a and b are the source node and destination node of e_1, and c and d are the source node and destination node of e_1. Delete e_1 and e_2. Add edge ac and bd. Define that e_1 and e_2 are switchable if edges ac and bd did not exist in the graph. The switch algorithm is shown in algorithm 3, and the generation algorithm is shown in algorithm 2

Fig. 8(a) shows the estimated graph of science fiction movie. The biggest node indicate that there is a leading node or leading actor. It looks pretty similar with that of graph of Terminator Salvation (TS), with the same level of clustering coefficients (TS: 0.65 and general graph: 0.62), same diameter (3 hops) and density around 0.15. Another cool idea is that we can mix mean vectors of movies like science fiction and action movie and produce sci-fic, romance and action movie, for example, something like James Bond 007.

Algorithm 2 Generation algorithm

```
procedure GENERATION(degree distribution, size, μ2)
    Generate a graph G with prescribed number of nodes, edges and degree distribution.
    while μ2 of G ≠ μ2 do
        SingleSwitch(G)
    return G
```

Movies	Most Similar Movie
LoR: The Fellowship of Rings	Mission Impossible 2
LoR: The Return of the King	Mission Impossible 1
Pulp Fiction	Go

Table 5: Movies Similarity

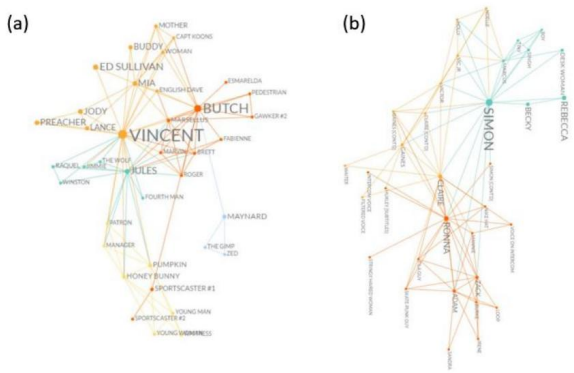

Figure 7: (a): Graph of Pulp Fiction. (b): Graph of Go (tilted)

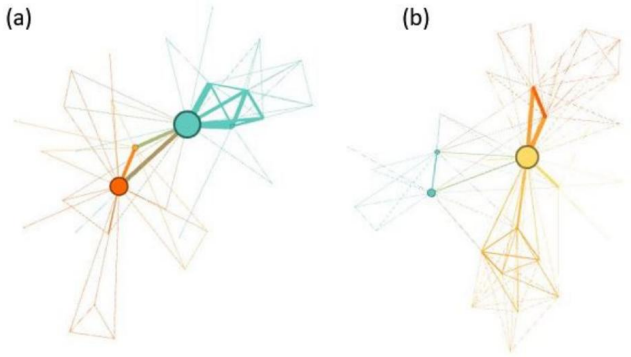

Figure 8: The general graph of science fiction movie and graph of Terminator Salvation.

Algorithm 3 SingleSwitch algorithm

```
procedure SingleSwitch(graph G)
    Pick two edges e1 and e2 at random
    if e1 and e2 are switchable then
        Switch e1 and e2
    return G
```

7 Classification by Principal Component Analysis

When it comes to dealing with high dimensional features, Principal Component Analysis (PCA) is a powerful tool to visualize them on to a 2D environment. Our vectors, either in the terms of standard properties or in the terms of motifs, are in 5 or 6 dimensions, resulting the feature matrix M of all 676 movies has the size of $676 \times 5 or 6$ To bring down the dimension using PCA, we first normalize M by substracting each row of M with its mean. Secondly, we can calculate the sum of norm of each column : $\sigma^2 = \frac{1}{m}\sum(M_i^2)$ Then we can normalize the substracted matrix M by divide each row of M with σ.

After normalization we have a normalized version of feature matrix D, we can apply PCA technique as follows: First, we take $C = D^T D$ This results in C with size 5×5 or 6×6. Secondly, bringing down the dimension by taking the eigenvectors corresponding to the top 2 largest eigenvalues of matrix C. The two eigenvectors combined results in matrix e which has dimension: (5, 2) or (6, 2). Finally, to extract the principal component of M: $PCA = M^T e$. The result PCA will have size (767, 2), and thus presentable in a 2D plot.

Here we apply PCA to feature vectors of comedy, drama and action movies. The result is shown in Fig.9. The clustering looks quite interesting: action movie tends to cluster together, indicating that actions' social network are very similar and maybe the storytelling and scene schema of actions are even similar. Dramas tend to scatter around, and this is correspondent with style of drama–various and fickle. Comedies concentrate less than action movies but still tend to have similar network schema.

8 Conclusion

Firstly, We constructed the database of 676 movies. Secondly, for each movie, we extracted a representative signature vector including cluster coefficients, diameters and number of different motifs. Thirdly, using feature vectors, we determined the similarity between graphs and give the most similar one of each movie. Fourthly, we developed an algorithm to generate graph with prescribed property. For each genre of movies, we give an average graph. Finally, we apply PCA to classify graphs into different groups and observe interesting clustering of movie graphs.

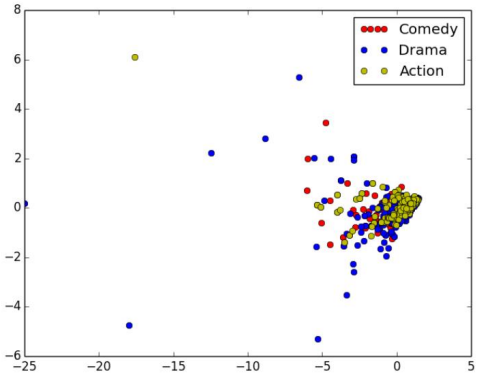

Figure 9: (a): The general graph of science fiction movie . (B): Graph of Terminator Salvation.

9 Future Work

For now, the Moviegalaxies just contain around 700 movies. As the database grow larger, our future work can involve collecting data from more movies. Furthermore, the movie graph is of relatively small size, and to test our algorithm in a family of larger graphs, say graphs of proteins with similar function. Also, the algorithm generating graphs with prescribed features can be improved by designing more intelligent method to modify features.

10 Acknowledgment & Breakdown of Work

We'd like to thank Prof. Leskovec for giving interesting and clear lectures. We also want to thank TAs for their efforts through this quarter.

We contributed equally to the project report.

References

[1] J. Kaminski, M. Schober, R. Albaladejo, O. Zastupailo, and C. Hidalgo, "Moviegalaxies - social networks in movies," Dec 2012.

[2] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos, "Netsimile: a scalable approach to size-independent network similarity," *arXiv preprint arXiv:1209.2684*, 2012.

[3] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos, "Network similarity via multiple social theories," in *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pp. 1439–1440, IEEE, 2013.

[4] J. W. Raymond, E. J. Gardiner, and P. Willett, "Rascal: Calculation of graph similarity using maximum common edge subgraphs," *The Computer Journal*, vol. 45, no. 6, pp. 631–644, 2002.

[5] X. Ying and X. Wu, "Graph generation with prescribed feature constraints.," in *SDM*, vol. 9, pp. 966–977, SIAM, 2009.

[6] F. Y. Bois and G. Gayraud, "Probabilistic generation of random networks taking into account information on motifs occurrence," *Journal of Computational Biology*, vol. 22, no. 1, pp. 25–36, 2015.

[7] G. Li, M. Semerci, B. Yener, and M. J. Zaki, "Effective graph classification based on topological and label attributes," *Statistical Analysis and Data Mining*, vol. 5, no. 4, pp. 265–283, 2012.

[8] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *International journal of pattern recognition and artificial intelligence*, vol. 18, no. 03, pp. 265–298, 2004.

[9] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," 2009.