Building a Predictor for Movie Ratings
Final Report

Haowen Cao
caohw@stanford.edu
Daniel Holstein
holstein@stanford.edu
Casatrina Lee
cylee1@stanford.edu

Abstract

We investigate the degree to which movie popularity is a function of cast and genre. We mined the open-source IMDb dataset, and restructured it into two bipartite graphs (a graph of movies to actors, and a graph of movies to genres). Using the HITS algorithm, we assign each actor and director a hub score and an authority score based on the popularity of future projects with other collaborators. Using a similar technique, we also use the ratings of the movie to predict the popularity of a particular genre. We also used PageRank on 2 bipartite graphs (movies to actors, and movies to directors). While PageRank did not achieve another rating prediction method, it did yield interesting insights into the structure of the bipartite graphs.

1 Introduction

IMDb is often used as a platform by which users can obtain relevant information about specific movies and artistes. Such information includes objective characteristics such as the cast and production crew, awards information, genre and production dates of the movies, as well as the biographies, accolades and other projects of a particular artiste in question. The database also provides an insight into the popularity of the movie as it allows users to rate movies, as well as leave comments on message boards. In this project, we focus mainly using ratings as a quantitative measure of the popularity of a movie, relying on the fact that the vast number of users who rate the movies, combined with IMDb's weighted-mean algorithm, allow for the rating information to provide a somewhat accurate representation of the movie's popularity.

Our project hopes to be able to build a predictor of a movie's success based on the actors and directors who are involved in it. We modeled the database as a directed bipartite graph with actors, directors and movies as nodes. We chose to focus primarily on directors and actors based on the assumption that those two roles would be the most visible to audiences and therefore most influential in determining the popularity of the movie. Edges are drawn from actors and directors to movies when the particular person has worked on that particular movie.

Using the HITS algorithm, we would be able to assign each actor and director a hub score and an authority score based on the popularity of his past projects, and therefore train a model to be able to predict the popularity of future projects with other collaborators. Using a similar technique, we also use the ratings of the movie to predict the popularity of a particular genre.

PageRank, HITS close cousin, was also run on two derived IMDb graphs: (1) the bipartite graph from movies to actors, and then the bipartite graph from movies to directors. Naturally, we expected results similar to those obtained via the HITS algorithm, and to obtain yet another rating prediction method. While this was not the case, applying PageRank did yield some interesting insight into the structure of the bipartite graphs.

2 Review of Prior Work

We reviewed other papers who had conducted various analyses on the IMDb dataset and focused mainly on papers that strove to utilize network features to predict movie popularity.

One paper (Oghina et al., 2012) utilized several social media platforms to predict IMDb movie ratings. Using a cross-channel prediction model, the authors focused mainly on data from Twitter and Youtube, extracting the number of tweets, the number of reviews and the number of upvotes and downvotes to draw conclusions on the popularity of the movie. The main statistical method used were

mean absolute error (MAE) and root mean squared error (RMSE). To distinguish between negative and positive reviews, the authors parsed the reviews and utilized sentiment analysis and the Spearman coefficient to classify reviews such that they would become qualitative indicators. The success of this model gives us a basis that social networks can act as accurate predictors of a movie's popularity. We hope to build on this finding by further incorporating objective features of movies (eg. cast, crew, genre, etc) and zoom in on the IMDb dataset instead of cross referencing other social media networks.

Another particularly relevant paper by Grujic, 2008, had also modeled the dataset as a bipartite graph to create a movie recommendation network. However, she focused mainly on comments by users on movies rather than on ratings. She used the movies that a user had commented on as a predictor of the kind of movies that the user would be interested in and constructed three distinct graphs to investigate clustering within the network. Her first and most basic model was to utilize the presence of a comment as an indication of interest by a user on a movie. Her second graph generated recommendation lists for movies based on the number of users each pair of movies had in common. Her final network was a user-preferential random network in which movies were randomly connected to 10 other movies chosen from a probability distribution proportional to the number of users who had commented on those movies.

Grujic found that there was a strong case of clustering within the IMDb network, and our project can build on this presence of clustering to build our predictor. While both papers mentioned above focused on comments, we choose to focus on a more quantitative measure - ratings. This allows us to have an idea of the how popular the movie is since a movie with a rating of 9/10 is clearly more popular than one with 2/10. This should be an improvement over the binary variable of comments where Grujic did not make a distinction between positive comments and negative comments. A universally unpopular film could receive numerous negative comments but would appear as popular' as a box office hit in this model.

3 Data Collection Process

We obtained the raw data using IMDb API. Based on what we need to use, we parsed part of it to obtain the data relevant to our investigation. For movies, we extracted the title of the movie, the year of release, the number of ratings it has received from users, as well as its rating as shown on the IMDb website. For actors, actresses and directors, we curated a list of movies that each actor or director has worked on previously. Finally, we grouped movies according to genre in order to investigate the correlation of genre and the popularity of the respective movies.

Table 1 shows the basic relevant features of our network.

feature	value
movie nodes	20828
actor nodes	62863
actress nodes	34204
director nodes	5671
cast $\rightarrow$ movie edges	866702
movie $\rightarrow$ genre edges	50741
Average labels of movies	2.43
Average rating of movies	6.39

Table 1: the relevant features of the network

4 The Algorithm

4.1 HITS

We used the HITS algorithm as mentioned in class in order to analyze the links between the nodes of our graphs. However, we did make some modifications so as to model our dataset more accurately. Figure 1 shows the modified HITS model we use for our network. In this network it contains three different types of nodes:

- The people nodes p_i: people including actors, actresses and directors.
- The movie nodes m_i: movies including the training set and the test set.
- The genre nodes g_i: genres for all the movies.

First, we create a network with movies, people to use the HITS algorithm. This would be a bipartite network in which all edges can be drawn from an person to a movie that they have worked on. In our model, the movie nodes are hubs and the people nodes are authorities. We evaluate the movie's ratings and model it as the authority score representing the quality as a content and the total sum of votes by experts. Correspondingly, the people nodes represent the quality as an expert since

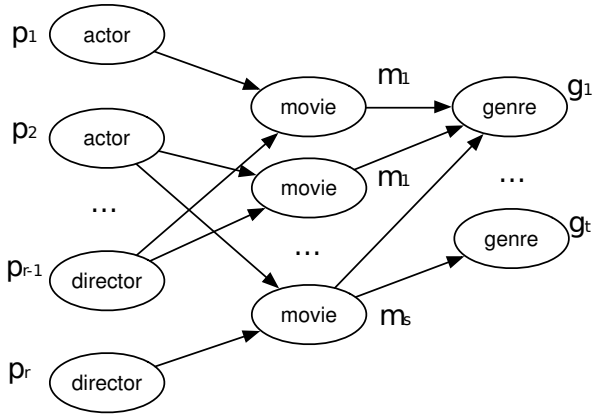

Figure 1: The HITS graph model for people, movies and genres

they are crucial factors in making up the popularity of the film.

Second, we create a bipartite network with movies and genres to use the HITS algorithm. Edges are drawn from a movie to a genre if a movie is part of that genre. This time, movie nodes are hubs and genre nodes are authorities. The genre score is useful to adjust the score of movies since different genres have significant differences in ratings. We can see the top 3 and the bottom 3 ratings of genres from Table 2 and we can see the the genre matters. The top 3: Documentary, Short and News have the scores much higher than the average score 6.39. On the contrary, the bottom 3: Reality-TV, Adult and Horror have the scores much lower than the average score.

Finally, when the movie gets two hub scores from people and genres, we get the average of these two scores to make the prediction.

genre	score
Documentary	7.43
Short	7.37
News	7.21
Reality-TV	5.99
Adult	5.43
Horror	5.31

Table 2: the top and bottom ratings for genres

Besides, to customize the algorithm to our experiments, we made several modifications in order to let the model fit our problem better:

1. We do not do the iteration and normalization on the score. Since we do not have a fixed sum for all the ratings of people, movies or genres, it makes no sense to make the sum as 1 or some other fixed number.

2. We have general weights for actors, actresses and directors. This is based on the assumption that they each have different extents of influence on the popularity of the movie. Specifically, the director was given the highest weight since we believe that the role of the director is most important in producing a good movie.

3. For the authority score of people, instead of only storing the average score, we also stored the total number of movies that the person has participated in. Therefore we are able to use the weighted score in the algorithm.

4.2 PageRank

We explored the structure of the IMDb graph using the PageRank algorithm. In this approach, we ran PageRank on two distinct, undirected, bipartite graphs: one from movies to actors, the other from movies to directors. With this approach, we did not seed the node ranks in any way with the existing IMDb ratings. PageRank was performed solely using the network structure of the bipartite graphs. Table 3 shows the prameters we use for our PageRank algorithm.

parameter	value
Damping factor	0.85
Convergence difference	10^{-10}
Max number of iterations	1000

Table 3: the parameters for PageRank

With PageRank, all node "types" (movies, actors, and directors) are assigned a single score. Thus, it is possible to directly contrast, in aggregate, the PageRank distribution among node types (ie, contrast rank distribution of actor node group against movie node group). We did this by combining the nodes, calculating PageRank, separating the node populations, and then ordering/graphing the PageRank of each population individually.

In addition to performing PageRank on the graph in its entirety, we also ran PageRank on subgraphs. We ran PageRank specifically on both talk shows and dramas. We also toggled membership in the subgraph by original IMDb ranking as well as number of original IMDb users that rated a particular

movie. Within these subdivisions, we compared relative PageRank scores to their respective, original IMDb movie ratings.

5 Results and Findings

5.1 HITS

Without loss of generality, we randomly hold 75% of the movies as the training set to get the scores for the people and the genres and use the other 25% as the test set. The true average rating in the test set was 6.41 while our predicted average rating was 6.57. This suggests that our predictor worked pretty well, albeit with a slight positive bias. The average absolute difference came out to be 0.8668. This corroborates with our initial findings that our algorithm was an accurate predictor for genre popularity. Compared to the initial result we reported in the milestone, this result shows significant improvement - we managed to decrease the absolute difference of prediction by more than 12%.

In order to better analyze how good our prediction is, we set the baseline prediction to compare to our model. Our baseline is: We let the predicted IMDb rating for each node be the median rating for all the movies in the dataset. And we use the same measuement as the model. From Table 6 we could see HITS can give a better prediction than the baseline, which just simply giving the median score of the movies.

In Figure 2, we illustrate the proportion of differences between the predicted results and the true results. The percentage of movies with a score difference of less than 0.5 was 37.83%, while 67.52% of movies had a score difference of less than 1, and almost all the movies (91.94%) had a score difference of less than 2 from the true rating. We can thus conclude that most of our predictions lie within a very narrow error margin from the true rating as taken from IMDb. This again gives us a good degree of confidence that our model is a good predictor for the popularity of the movie.

Our results still showed a slight positive bias in our results. Examining Figure 3, we illustrate the histogram of the differences distribution vs the true ratings. Like the result from the milestone, the tail is very thin for high ratings, with the majority of the ratings clustering around 6.5. There are very few low ratings across all movies. Given that our prediction tends to be higher than the true average, we are more likely to give a higher score than the original rating if the original rating is very low. We think

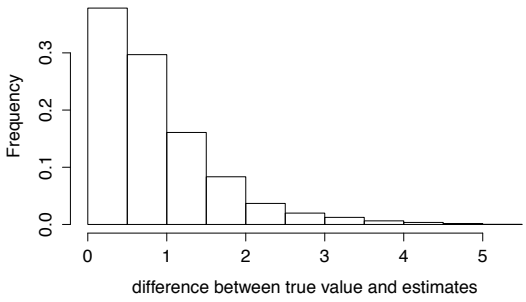

Figure 2: Testset proportion of differences between prediction and true value

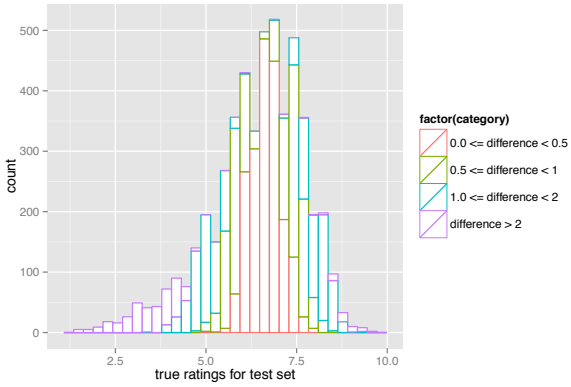

Figure 3: Testset result for relationship between differences and ratings

that the possible reason is that the lowest score for genres are 5.3. Therefore our lower boundary for prediction is $5.3/2 = 2.65$. So we cannot give such an extreme low prediction. Generally speaking, the genre helps adjust the ratings and decrease the difference, but still cannot competely solve the problem of over-rating for low rating movies.

5.2 PageRank

5.2.1 Introductory Findings

1. **Graph of actors and movies:**

   Surprisingly, PageRank gave a different view of this bipartite graph than that produced by HITS. Perhaps Table 4 is best evidenced by considering the nodes with the top PageRank in the actors to movies bipartite graph.

   Of the 18 nodes above, all exactly one third (6 nodes) are labeled as ”Talk-Shows” by IMDb. Another 10 nodes are labeled as ”Dramas.” 2 nodes (”E! True Hollywood Story” and ”Cold Case” fall into neither category, according to

IMDb.) Also, note that 0 of the top 20 nodes are actors (even though actor nodes also received PageRank scores). Only 75 of the 20,821 movies (a mere 0.36%) in the considered graph are talk shows, according to IMDb. Furthermore, 75 of the 117,715 (0.06%) nodes overall (movies and actors) are talk shows. In the top 18 nodes, 33.3% are talk shows.

There is certainly not a one-to-one correspondence between nodes with the highest PageRank and nodes with the highest IMDb rating. Indeed, with approximately 3400 user votes, The View stands with an IMDb rating of 3.2.

2. **Graph with directors and movies:**

Table 5 shows that it relatively even split of prolific directors (Miike, Allen, etc.) and long-running series with many different directors (Doctors).

5.2.2 Influence Split - Film vs. People

Figure 4 is Log-Scale Graph of the n-th largest PageRank node, in movie-actor graph.

As a conglomerate, the PageRank scores of the movie nodes are more variable than that of the more-constant rank-distributed actor nodes. At the high and low ends of the PageRank scale, movies dominate. The top PageRank score for a movie node is greater than an order of magnitude larger than the top actor score.

Figure 5 is Log-scale plot of the n-th largest PageRank node, in movie-director graph.

On the high end of the PageRank spectrum, directors and movies are both present. On the low end, directors dominate. Throughout the bulk of the distribution, the PageRank of the director nodes definitely dominates the PageRank of the movie nodes.

5.2.3 Connecting PageRank to IMDb Rating System

In order to translate PageRank score into IMDb score: We ordered the movie nodes in order of their PageRank score. Then, we ranked the ordered nodes in order of their IMDb rating. We assigned the node with the n-th largest PageRank score the n-th largest rating. This serves as the PageRank-predicted IMDb rating of the node. The difference

Node Type	Title/Name	PageRank Score
Movie	”Holby City”	0.0008331
Person	Miike, Takashi	0.0007477
Movie	”Doctors”	0.0006928
Person	Boll, Uwe	0.0006616
Movie	”Hollyoaks”	0.0006510
Person	Dhawan, David	0.0006163
Person	Allen, Woody	0.0005936

Table 5: top PageRank scores for directors and movies

Title	PageRank Score	Number of Ratings
”Law & Order”	0.002493	21190
”Doctors”	0.002473	572
”The Tonight Show with Jay Leno”	0.002107	7326
”Law & Order: Special Victims Unit”	0.001990	41465
”ER”	0.001935	34592
”Holby City”	0.001720	746
”Jimmy Kimmel Live!”	0.001626	5244
”CSI: Crime Scene Investigation”	0.001429	58261
”Law & Order: Criminal Intent”	0.001365	15103
”Late Show with David Letterman”	0.001381	7873
”Late Night with Conan O’Brien”	0.001341	9680
”Criminal Minds”	0.001181	82584
”CSI: Miami”	0.001115	36950
”NYPD Blue”	0.001010	7340
”E! True Hollywood Story”	0.000981	550
”The Late Late Show with Craig Ferguson”	0.000971	7240
”Cold Case”	0.000961	16116
”The View”	0.000954	3391

Table 4: top pagerank scores in actors to movies graph

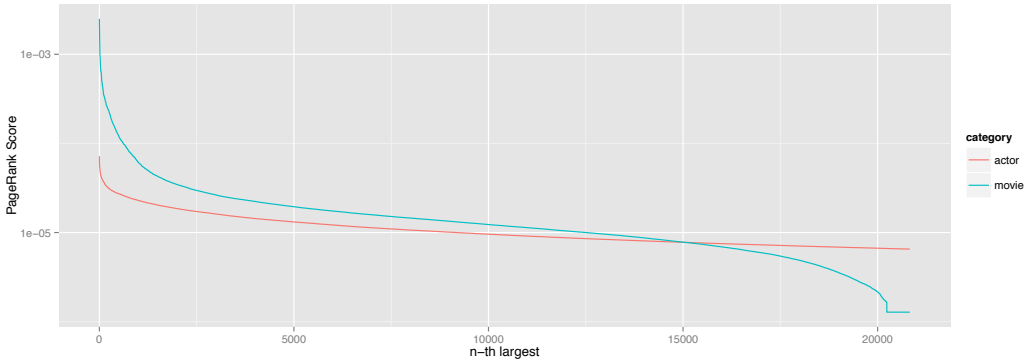

Figure 4: Log scale plot in movie-actor graph

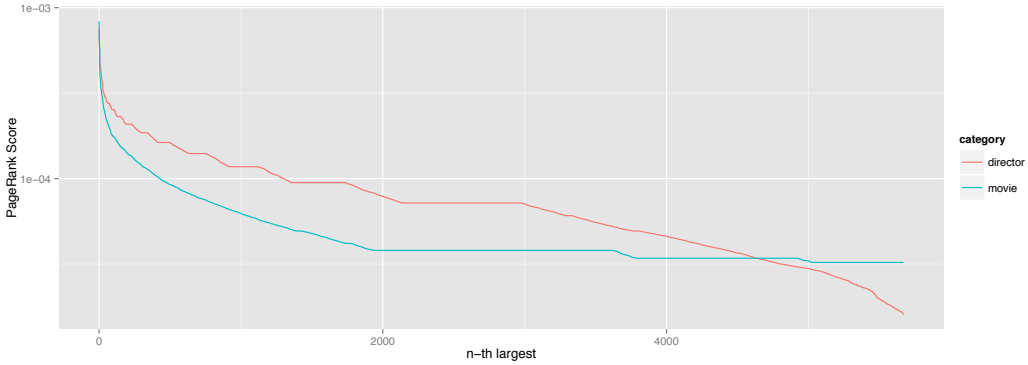

Figure 5: Log scale plot in movie-director graph

between the predicted IMDb rating from the actual IMDb rating is the prediction error for that node. Table 6 shows the relative statistics of the result.

Method	avg(diff)	median(diff)	std.
Baseline	0.9953	0.8	0.8478
HITS	0.8668	0.69	0.7671
PageRank	1.141	1.1	1.141

Table 6: statistical results for different methods

We could see that: not only is PageRank score not a solid predictor of IMDb rating, but, on average, it performs in excess of 40% worse than an incredibly naive prediction method (predicting every node with the global median).

Given the preponderance of talk shows in the top-PageRanked nodes, one may note that a successful talk show, fundamentally, "links" to a concentration of high-profile talent. Table 7 shows the PageRank scores for the top talk shows ordered by the number of ratings. A natural question is whether or not, among talk shows, the number of IMDb ratings it attracts (the amount of attention it garnered on the web) would at all be associated with PageRank (the viewer "attention" a node has to distribute to its guests/neighbor nodes). Indeed, 40% of the most-rated talk shows are present in the set of twenty nodes with the highest PageRank in the actor-movie graph.

We see from Table 7 that four of the ten most-rated talk shows also appear in the set of eighteen nodes with highest PageRank score in the movie-actor graph. Exactly one of the ten highest-rated talk shows (The Late Night Show with Conan OBrien, rated at 8.5) appears in all three data sets.

6 Conclusion

In using the HITS algorithm to analyze the IMDb network, we found that our predictor for the movie's rating based on the cast and for the genre's popularity based on the movies was fairly accurate. It demonstrated a low error rate and low average absolute difference from the test set. However, the predictor displayed a slight positive bias and tended to assign ratings that were higher than the true rating. This can be attributed to the form of the data, in which people tend to be positively biased in giving ratings, with most ratings clustering around the

Title	number of Ratings	PageRank Score
”Top Gear”	50822	1.766e-05
”The Daily Show”	21069	0.000799
”The Colbert Report”	20993	0.000471
”Conan”	12422	5.657e-05
”Late Night with Conan O’Brien”	9680	0.00134
”Da Ali G Show”	9430	9.787e-06
”Ellen: The Ellen DeGeneres Show”	7966	0.000741
”Late Show with David Letterman”	7873	0.00138
”The Tonight Show with Jay Leno”	7326	0.00210
”The Late Late Show with Craig Ferguson”	7240	0.000971

Table 7: Talk shows, order by the number of ratings

6-8 range. Besides, the genre ratings constrained a lower boundary for the prediction. But there are often very few low ratings; thus, our predictor, when trained on this data, would assign a higher score than the true rating when analyzing a movie with a low true rating.

Unlike HITS, PageRank does not seem to bear much of a relation to IMDb movie rating. (In fact, predicting a node’s IMDb rating from its relative PageRank score leads to larger errors than naively predicting it from the global IMDb median rating.) Instead, nodes with the highest PageRank scores in the movie-actor graph seem to be the ”great connectors” of the television-/film- world. Talk shows are over-represented in nodes with the top PageRank compared to the general node population (by a factor of five hundred, orders of magnitude above chance). Shows like Jay Leno or David Letterman serial-interview high-profile guest after high-profile guest. Dramas like Law & Order run for years and include a myriad of high-profile guest stars. Nodes with high PageRank are movies/shows (not people) who link to a variety of high-profile talent.

The greatest ”connectors” in the movie-actor graph are all movies (or shows), instead of people. On the other hand, in the movie-director graph the nodes with the greatest PageRank are divided more-or-less evenly between movie and director nodes.

In the course of our project, we noted that one main limitation of the dataset was the inability to standardize ratings as a metric of popularity. A reviewer’s rating of 9 might correspond to another reviewer’s rating of 7. The presence of personal bias therefore implies that ratings are inherently subjective. We relied on using a large enough dataset in order to minimize this error.

Moving forward, ratings are by one metric by which one can measure a movie’s popularity or success. Much like the paper by Oghina et al, we could cross reference IMDb’s data with other social networking sites, or take into account the reviews posted on message boards. This would allow us to gain a better insight into what exactly it was about the movie that made it popular. Perhaps if comments mentioned an actor or director specifically, our weights in our HITS algorithm would change correspondingly.