Link-Based Character Classification and Genre Prediction in Movie Networks

Colin Kincaid and Salvador Valdes

1 Introduction

In popular media, the ubiquity of character archetypes and the predictability of movie plots have become points of ridicule for the film industry. As an example, see Mindy Kaling's analysis of romantic comedy archetypes [1]. Some scholars have gone so far as to formalize the the archetypal structure of movies in specific genres (e.g., Basinger's book on the "anatomy" of a World War II film [2]). To the human movie critic, movies of the same categories can seem similar to the point of unoriginality.

In this paper, we investigate the merit of claims about the prevalence and predictability of movie archetypes. Using only the networks formed by character interactions, we train classifiers to predict a movie's genre and identify the social roles that its characters perform. First, we use an iterative Dirichlet process Gaussian mixture model (iDPGMM) to assign a tie strength to each link between two characters. We can then use tie strengths to assign each character to one of a few discrete subgroups within a movie, hoping to identify noteworthy characters who conform to genre archetypes. We also use aggregate information about tie strengths—as well as network characteristics such as characteristic path length and transitivity—to form a feature vector for each movie. Finally, we use these vectors along with ground truth genre labels to train a genre-predicting classifier. These results allow us to answer the question of which movie genres are the most predictable.

2 Prior Work

2.1 Genre Prediction

We decided to focus on the problem of movie genre prediction after reading Zhou, Hermans, Karandikar, and Rehg's paper on the topic [3]. They come at the problem with the motivation of predicting movies' genres for use in recommender systems, which lie at the core of products such as Netflix and YouTube. The paper presents the results of predicting a movie's genre based on its trailer. The authors' rationale for using movie trailers is that they require fewer resources to process and store than entire movies, but should capture most of the movies's salient characteristics. We agree with Zhou et al.'s assessment of the importance of genre prediction and are similarly concerned with the tractability of classification. We are optimistic about a network-based approach to movie genre classification because it requires no processing or storage of video files; if successful, our algorithm will only require a movie's interaction graph (easily produced from the script) in order to predict its genre.

2.2 Tie Strengths

In "Discovering Multiple Social Ties for Characterization of Individuals in Online Social Networks," Chung et al. use an iDPGMM to define an arbitrary number of relationship types for a social graph and classify each edge into one of the different types. The model's incremental operation allows nodes to be added to and removed from the network without needing to recompute tie strengths for any nodes other than those related to the modified node's neighbors [4]. Chung et al. use the resultant edge type categories to identify types of social leaders within small networks such as "social brokers" and "regional leaders." We follow this procedure closely, anticipating that the presence of these character types will be a significant feature of movie genres.

3 Dataset

This project uses the MovieGalaxies database of social interaction graphs in movies [5]. Nodes in the networks are movie characters, and two nodes are connected by an edge if the characters appear in the same scene at least once. Though the scope of this project is limited to use of network structure with unweighted edges, the data set comes with edge weights and some basic node properties such as degree and closeness centrality. Movie graphs are also labeled with tags for popular actors and, importantly for our project, genres.

4 Classification Features

The biggest impediment to our classifier's attainment of good results was the difficulty of selecting useful features for training. In addition to some manipulations of tie strength data, we included several graph properties that we hoped would be useful, on their own or in concert with each other, for separating genres.

4.1 Node count

Because some movie genres tend to have fewer named characters than others, we decided to use node count as a feature. Importantly, it is also possible that some features are only significant when combined with node count (e.g., clustering coefficient might not separate genres on its own, but perhaps there is a specific genre characterized by low node count and high clustering coefficient while another is characterized by high node count and high clustering coefficient).

4.2 Maximum degree ratio

We define the maximum degree ratio for a network as the maximum degree of any node divided by the maximum degree that any node in the network could have (number of nodes - 1). We expect this statistic to vary inversely with a movie's complexity, as a low maximum

degree ratio means that there is no central protagonist tied to every other character. Consequently, there must be supporting-character relationships that do not directly fit into the protagonist's story arc.

4.3 Clustering Coefficient

A network node's clustering coefficient measures the extent to which its neighbors are connected to each other. We hypothesize that movies of the same genre could have similar global clustering coefficients because characters would have similar propensities to form social clusters.

4.4 Transitivity

A network's transitivity T is given by

$$T = \frac{\sum_{i \in N} 2t_i}{\sum_{i \in N} k_i(k_i - 1)}$$

Transitivity is similar to clustering coefficient in that they both measure the number of triads and the number of edges present in a graph. The two metrics are different in that global clustering coefficient averages all nodes' *individual* clustering coefficients, whereas transitivity uses the *aggregate* numbers of triads and degrees without caring how much any individual node contributes[6]. Both metrics are potentially useful for network analysis: Clustering coefficient tells us about individual characters' properties, whereas transitivity describes properties of the ensemble.

4.5 Maximum Betweenness Centrality

The presence of individual characters with important social roles might not be reported by the clustering coefficient metric, as less central characters can outweigh an individual character's contribution to the global coefficient. Since the presence of a central protagonist might be a feature of some genres (e.g., superhero movies) and not others, we decided to include the maximum betweenness centrality across all nodes as a network feature. We define the maximum normalized betweenness centrality for a network as

$$\max_i g(i) = \max_i \left(\sum_{i \neq j \neq k} \frac{2\sigma_{jk}(i)}{(N-1)(N-2)\sigma_{jk}} \right)$$

where σ_{jk} is the number of shortest paths from node j to node k and $\sigma_{jk}(i)$ is the number of those paths that pass through node i. Note that the betweenness centrality is normalized by the number of edges that can exist in the graph without i being one endpoint, so that the maximum betweenness centrality is not influenced by the number of pairs of nodes in the graph. This metric essentially says how important the movie's most important character is, in terms of brokering social connections.

4.6 Characteristic Path Length

We use a movie graph's characteristic path length as a heuristic for capturing the movie's complexity. A movie typically has a fairly small number of named characters, each of whom is unlikely to appear in a scene without one of the main characters. If a movie's characteristic path length is more than just a couple hops, we hypothesize that there is a good chance that there are multiple connected components in the network (possibly indicating multiple distinct subplots, or at least interpersonal relationships that do not involve one central protagonist). In other words, characteristic path length could correlate directly with the complexity of a movie's plot, which is likely to be a genre-predicting property.

4.7 Tie Strengths

As described in Chung, et al., the two endpoints of a network edge have a cosine similarity to each other, hereafter referred to as tie strength. The equation for tie strength is shown below:

$$\sigma(v, w) = \frac{|\tau(v) \cap \tau(w)|}{\sqrt{|\tau(v)||\tau(w)|}}$$

where $\tau(v)$ denotes v's neighbors. The cosine similarity of two nodes is essentially the ratio of the number for neighbors they have in common to the number of neighbors they could have in common. These values fall between 0 and 1. Figure 1 shows the distribution of tie strengths. The vertical lines are the means produced by DPGMM, discussed below.

We use tie strength values to generate three features for our classifiers. The first was average tie strength: the mean of all tie strengths in a film. In an individual film, low average tie strength would indicate a film with many leaf-nodes, characters that interact with few other characters or who have many acquaintances with whom they do not share a social group. For example, the Breakfast Club has a high average tie strength, while Forrest Gump has a much lower average tie strength.

The second feature was a ratio of tie strength types. Understanding this feature requires some background. We implemented the process outlined in Chung et al.'s paper. The first step is to group edges according to categories delineated by tie strength. Because we seek to identify relationship types that are common across all of our films, groupings must not be per-film, but rather universal. To accomplish this, we combine all of the network into one such that each individual film is a component of a supergraph.

From here DPGMM, is applied. Unlike Chung et al., our process does not require incremental DPGMM because our networks are not dynamic. Edges do not come and go after statistics are computed. One of the primary functions of DPGMM in this process is to extract the number of edge types, or how many groups the tie strength data can be bucketed into. When run on just a few graphs, DPGMM refuses to group edges into more than two buckets, one where the mean tie strength is 0.6 and the other where it is 1.0. When run on all 89,225 edges, DPGMM still found these two groups, but was also able to identify several other relationships whose meanings were less clear. For this project, we have chosen to only classify relationships according to the two dominant groups, as they are easily justified by the histogram in Figure 1. Frequencies of the two types are reported in Table 1.

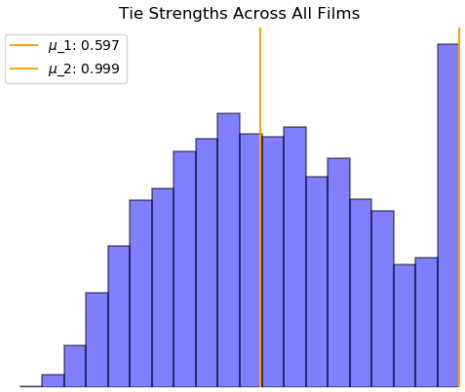

Figure 1: Tie Strengths

Table 1: 2 Groupings

mean	edges
0.5973	81485
0.9999	7770

At this point, we can compute the second feature: strong tie ratio. This is simply the number of strong ties (edges belonging to the group with mean tie strength of 1.0) divided by the number of edges in a film. The strong tie group describes a very specific type of relationship, usually between outlying characters who appear in a single scene together. This ratio is a measure of the prevalence of these types of scenes.

Lastly, we want to characterize nodes according to the types of ties they have. Our data were much sparser than that presented in the paper, so we made the decision to classify any node with a strong tie as a "strongly tied node." These nodes exist in small cliques within graphs. The ratio of strongly tied nodes to total node count is the third and last feature we draw from tie strength data.

Strongly tied nodes usually come in small cliques in a film. In most films, they are minor characters who appear in a single scene together. For example, in the film Forrest Gump, there are ten of these small cliques, groups of people the protagonist appears with once and

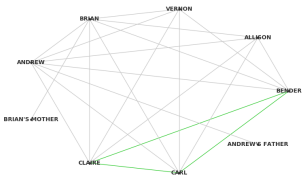

Figure 2:
The Breakfast Club:
Strong Ties

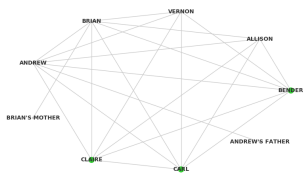

Figure 3:
The Breakfast Club:
Strongly Tied Nodes

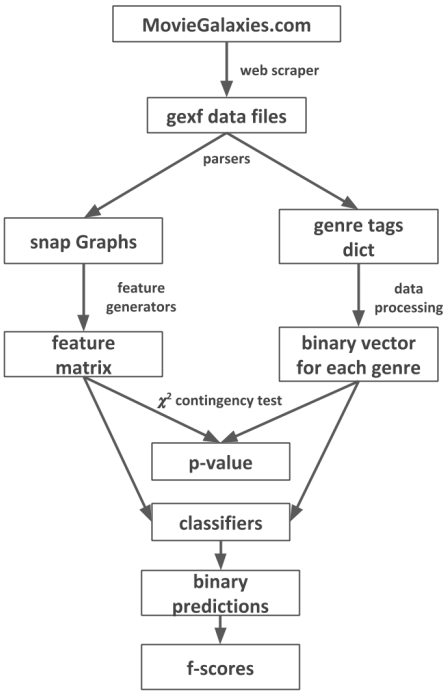

Figure 4: Prediction Pipeline

never revisits. Contrary to their name, strong ties do not imply a strong social relationship. In The Breakfast Club, shown in Figure 3, strongly tied nodes are simply those characters who are mutually not acquainted with outsiders like parents and the janitor.

5 Experiments

Our classification pipeline is described in Figure 4. Beginning with Graph Exchange XML Format (GEXF) files of individual movie networks, we perform parsing, statistical analysis, and machine learning to assess classifier performance.

5.1 Parsing

The MovieGalaxies data set consists of GEXF representations of individual movie networks. The NetworkX Python library allowed us to parse GEXF files into NetworkX graph representations, which we then exported as edge lists (since we only cared about network structure) for conversion into a SNAP graph. By scraping the tags associated with each film from the MovieGalaxies website, we generated a list of tags for each film id. From this list, we generated a binary vector for each genre, using a 1 if a film was in the genre and 0 if it was not.

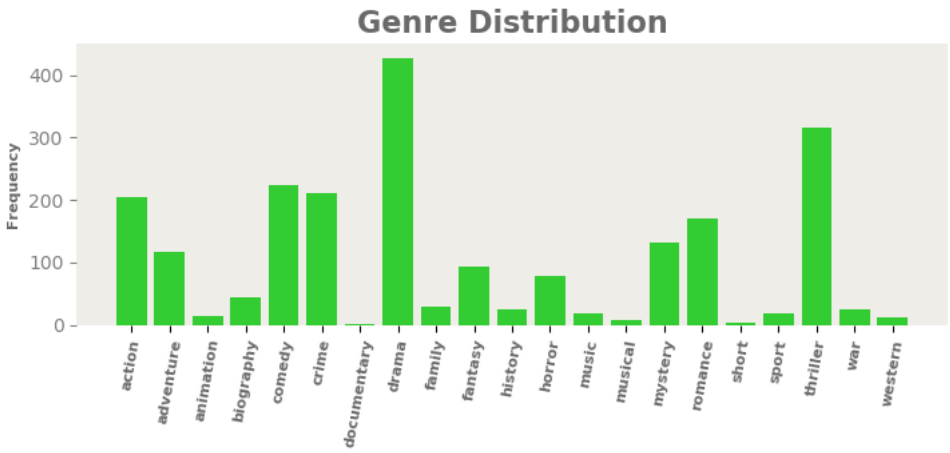

Figure 5: Several genres were not frequent enough to produce f-scores

At this point, we committed to using multiple binary classifications rather than a multi-class approach. We saw that one film could be tagged with multiple genres, and forcing each film into one genre could potentially worsen performance by reducing the number of positive data points in other genres. After producing early results, we returned to the parsing stage to cull genres. Figure 5, below, shows the counts of each film in the dataset.

We saw that several genres were so infrequent that held-out data often contained zero positive data points. Consequently, we could not draw conclusions from the resulting F1 scores. These results will be discussed in depth in following sections, but it is important to point out that while all films were used to represent the population, we only attempted to positively classify those in genres with 100 or more examples. The following genres were classified: action, adventure, comedy, crime, mystery, romance, and thriller.

5.2 Individual Feature Analysis

The result of the individual feature analysis is one of the final products of this paper. Labeled 'p-values' on the pipeline, this step determined which features were important in isolation. For each genre, we divided the feature into two distributions: the distribution within the genre and the distribution outside of the genre. We then used a chi-squared test to determine the probability that the two distributions were drawn from the same population. Figure 6, above, shows the results of these tests. The light blue represents the distribution of the feature across non-action films and the dark blue is the distribution within the action genre.

5.3 Binary Classification

To perform binary classification, we used three types of classifiers: a random guesser, a Support Vector Machine (SVM), and a Random Forest classifier from the SciKit-Learn toolbox. The random guesser was used as a baseline and labeled test data as positive $P\%$ of the time, where P was the percent of positive data points in the training data. We chose this as the baseline classifier because the F1 score metric approaches zero if either precision or recall

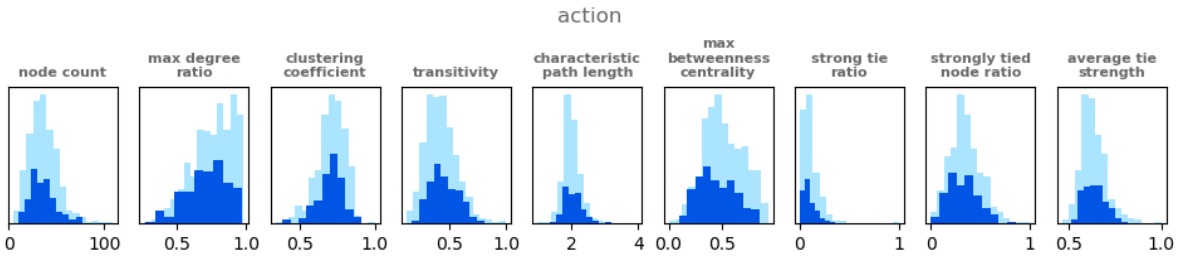

Figure 6: Feature Distribution within Genre

approaches zero, so a simple 'guess X every time' would not be a fair baseline.

The SVM was fine-tuned to achieve the best performance possible in one category. One of the experiments suggested in SciKit-learn documentation was a grid search over SVM parameters C and γ [7]. C is a smoothing parameter. A small value for C smooths the decision boundary while a large value for C attempts to classify all training examples correctly. A large value for γ gives training examples a small radius of influence and a small value for γ gives a large radius of influence. We did not see a local optimum, but rather a gradient: large C values produced lower F1 scores. Large γ values also produced lower F1 scores.

The third classifier was Random Forest. We experimented with several depths and chose to report on the maximum depth and a depth of 1.

To evaluate classifiers, k-fold cross-validation was used with k=5. K-fold is a common cross-validation that allowed us to recycle our small dataset as both test and train data and gave us confidence that our F1 scores were not one-off results.

6 Findings

6.1 Individual Feature Analysis

Figure 7, above, shows p-values computed by comparing the distribution of each feature within a genre to the distribution of that feature outside of the genre. For example, there is a 0.9% probability that the distribution of transitivity values within the action genre was drawn from the same population as the transitivity values from non-action genres. This can be observed in the distributions in Figure 6 and suggests that transitivity is a useful feature for classifying whether a movie is in the action genre.

6.2 Binary Classification

The F1 scores of our classifiers are shown in Figure 8, below. The only genre the classifers could effectively identify was drama, the most common genre. This has positive implications for the performance with larger datasets. Both SVM and Random Forest outperformed our baseline, the random guesser. Our tuned SVM performed best with an F1 score of 0.71 in the drama genre. The high-depth Random Forest performed more similarly to the random guesser, while low-depth Random Forest was comparable to the tuned SVM.

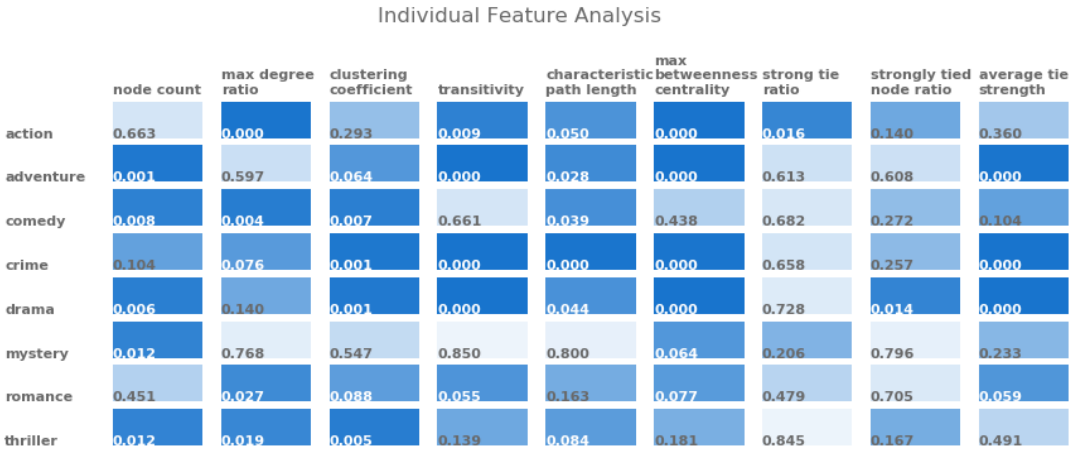

Figure 7: Feature Distribution p-Values

Figure 8: F-Scores for Binary Classifiers

7 Future Work

This project abounds with opportunities for further experimentation. First, we could attempt to get more features out of our data. There is a nearly unlimited number of graph properties we could generate by manipulating the network structure differently, and it is likely that some of them would help our classifiers. We could also add a layer of richness to our data by using the edge weights present in the original graph. It would require some innovation in defining new metrics, but we could use edge weights (corresponding to the number of shared scenes between characters) to augment our existing tie strength statistics and make different relationship types more distinguishable. Next, we can generate more data. This project only used networks available in the MovieGalaxies data set, but as we point out when comparing our network-based approach to the trailer-based approach, generating movie networks in this format is easy. Assuming that movie scripts mostly follow a standard structure, we can quickly run scripts through a parser to identify which characters appear in scenes together. At that point, we have everything we need to analyze the movie. Finally, we can use our data (which we can generate more of and get more out of) in different ways. We chose to predict genre in this project, but there are other interesting properties of movies such as box office revenue and average critic rating, which we could just as easily scrape from sources such as IMDb and try to predict using classifiers. We can even move away from machine learning and use our features for a clustering problem, where feature vectors act as points in space and we attempt to predict genre based on an algorithm such as hierarchical clustering or k-means. Clustering would probably work best on movies labeled with only one genre, but if we have enough of those, we might end up with a visualization that lets us see the which movie genres are most similar to each other in terms of network structure.

8 Conclusion

In this project, we sought to discover how accurately a computer can predict a movie's genre based solely on the structure of the characters' network. We built a pipeline to which we can input movie networks, generate features of those networks, train movie genre classifiers, and evaluate their performance. The pipeline is configurable for easy addition and removal of feature-generating functions and also measures how well those features identify movies of specific genres.

While our classifiers were only able to perform well on one genre, it is promising that it performed well on the genre with the most samples in our data set. We believe that by implementing some or all of the ideas listed in our Future Work section, there is strong potential to classify movie genres accurately based solely on network structure, easing a particularly resource-intensive step in recommender systems and settling the question of which movie genres are the most predictable.

We would like to thank Jermain Kaminski, the MovieGalaxies founder who gave us access to the data set's GEXF files when our attempts to scrape led to some movie ID mismatches, as well as Stanley Jacob, who helped us scrape tags from the Movie Galaxies website. We are also grateful to Poorvi Bhargava and the rest of the CS224W teaching staff.

References

[1] Kaling, M. (2011, October 03). Flick Chicks. Retrieved November 16, 2017, from https://www.newyorker.com/magazine/2011/10/03/flick-chicks

[2] Basinger, J. (2003). The World War II combat film: Anatomy of a genre. Wesleyan University Press.

[3] Zhou, H., Hermans, T., Karandikar, A. V., & Rehg, J. M. (2010, October). Movie genre classification via scene categorization. In Proceedings of the 18th ACM international conference on Multimedia (pp. 747-750). ACM. Chicago

[4] Chung, Ming-Hua, et al. "Discovering Multiple Social Ties for Characterization of Individuals in Online Social Networks." Network Intelligence Conference (ENIC), 2016 Third European. IEEE, 2016. [5] Movie Galaxies. Available: http://moviegalaxies.com/movies.

[6] Rubinov, M., & Sporns, O. (2010). Complex network measures of brain connectivity: uses and interpretations. Neuroimage, 52(3), 1059-1069.

[7] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.