The Association Between Film Industry Success and Prior Career History: A Machine Learning Approach

The Association Between Film Industry Success and Prior Career History:

A Machine Learning Approach

Michael E. Tashman

A Thesis in the Field of Information Technology

for the Degree of Master of Liberal Arts in Extension Studies

Harvard University

November 2015

Abstract

My thesis project is a means of understanding the conditions associated with success and failure in the American film industry. This is carried out by tracking the careers of several thousand actors and actresses, and the number of votes that their movies have received on IMDb. A fundamental characteristic of film career success is that of influence from prior success or failure—consider that an established "star" will almost certainly receive opportunities denied to an unknown actor, or that a successful actor with a string of poorly received films may stop receiving offers for desirable roles. The goal for this project is to to develop an understanding of how these past events are linked with future success.

The results of this project show a significant difference in career development between actors and actresses—actors' career trajectories are significantly influenced by a small number of "make or break" films, while actresses' careers are based on overall lifetime performance, particularly in an ability to avoid poorly-received films. Indeed, negatively received films are shown to have a distinctly greater influence on actresses' careers than those that were positively received.

These results were obtained from a model using machine learning to find which movies from actors' and actresses' pasts tend to have the most predictive information. The parameters for which movies should be included in this set was optimized using a genetic learning algorithm, considering factors such as: film age, whether it was well-

received or poorly-received, and if so, to what magnitude, and whether the film fits with the natural periodicity that many actors' and actresses' careers exhibit. Results were obtained following an extensive optimization, consisting of approximately 5000 evolutionary steps and 200,000 fitness evaluations, done over 125 hours.

Acknowledgements

Thank you first and foremost to my thesis director, Dr. Albert-László Barabási, for giving me the opportunity to do research in this unique field, with some of the most talented people with whom I've ever had the pleasure of working. Thank you to my research advisor, Dr. Jeff Parker, for his guidance throughout this process, and for introducing me to much of the methodology that underlies this work.

Thank you to Harvard University for creating this incredible opportunity. I would be remiss without mentioning some professors who have had a particular influence on my work, and on my appreciation of mathematics: Jameel al-Aidroos, Paul Bamberg, Sukhada Fadnavis, Cliff Taubes.

Thank you to my parents, Marsha and David, to my brother Craig, and to Juin, Matt, Suzanne, and Theresa—all of whom provided the encouragement, support and inspiration that made this project possible.

Table of Contents

List of Figures

Chapter 1

Introduction

Success in the American film industry is that of of legends and lore. Underlying the romance is a profoundly rational industry: actors build careers on an ability to bring patrons into the theater and generate revenue, producers hire cast members based in part on past performance, and moviegoers choose films to see based on heuristic information such as plot synopsis, media buzz, whether the movie is a sequel/adaptation from an earlier work with which they are familiar, and knowledge of the cast. The success or failure of a movie is therefore linked to the audience's expectation that its cast will deliver a good performance or choose a good project—an expectation that is necessarily derived from knowledge of those actors' past performances and choices. By this process, actors' career trajectories are heavily influenced by their history of success or failure. The focus of my thesis project is to develop an understanding of this mechanism of success: by using artificial intelligence to learn how to identify the prior career movies that will best predict future success, we can learn how careers and built and destroyed.

The work for this project builds upon research done by myself and others at Professor Albert-László Barabási's research group, the Center for Complex Network Research at Northeastern University.

Section 1.1

Prior Work

A good deal of prior work exists, which has attempted to identify leading indicators of film popularity. These indicators are generally based on factors that are measurable either shortly before the release of a given movie, such as pre-release buzz measured by number of tweets published about the film, or shortly after release, such as day-to-day box office receipts. Notably, the prior art does not consider popularity of earlier career work—the central approach of this thesis.

Section 1.1.1: Film Research

Despite the somewhat different approaches taken by the existing research, their results are relevant to this work. One such study is Mestyán, Yesseri, Kertész (2013), which uses the activity level of the editors and viewers of the film's Wikipedia entry, prior to the film's release, to predict its eventual revenue. The authors used a multivariate regression, between the number of page views, the number of users, the number of edits, and the collaborative rigor of the editors – and the eventual financial success of each movie. They found a strong coefficient of determination for these variables, leading to the conclusion that these factors can indeed be seen as predictive of the financial success of the movie. This is relevant, because a good deal of the buzz for a movie before its release is based on the cast and crew—this suggests a basis for explaining why prior work of the

cast and crew can be predictive for their future success: if people are familiar with actors' past work, this will influence attendance of their future films.

The idea that the cast of a film is notably linked to its box office success is supported by Sharda and Delen (2006). In this paper, the authors used a neural network algorithm to predict the financial success of 834 movies, released from 1998 to 2002. The authors used numerous categories of data, including genre, rating, presence of stars, and use of special effects; when their machine learning algorithm was trained they were able to learn which categories were most predictive. Other than number of screens—which is ambiguous as to whether it can be considered a cause of success or simply an effect—the two most significant factors for success were high-end special effects, and presence of "A list" actors and actresses. In this way this study supports the idea that film cast has a strong impact on success or failure.

The importance of film cast is further supported by Levin, Levin and Heath (1997). In this study researchers investigated whether moviegoers would consider the presence of a prominent star in their decision as to whether to see a given film. To that end they conducted a survey with 62 undergraduate students. Participants were divided into groups, such that each group received synopses of a set of films. Some participants were given critics' reviews (positive or negative); some participants were informed of fictitious stars being in the film, as a control, while others were told that real-world stars were in the film. The results showed that in the presence of positive reviews, stars had a

minimal impact on success, as most people were likely to see the movie anyway; in the absence of reviews or when the reviews were negative, however, star presence was seen to significantly increase the likelihood of test subjects seeing the movie.

A key aspect of this thesis is that of the relative impact of positive and negative impressions. A relevant study was conducted by Hennig-Thurau (2014). Researchers used a support vector machine to perform sentiment analysis on tweets about movies, and used the sentiments to predict eventual revenue. However, the researchers separately considered the effects of positive and negative tweets, under the hypothesis that negative buzz may have more impact than positive buzz. Their results confirmed this: they found that although positive tweets had little correlation with eventual revenue, negative tweets were strongly associated with decreased revenue. In this way negative word of mouth can have a more profound impact than positive word of mouth—this will inform some of the analysis in this thesis. The researchers then conducted a poll of film viewers, which found that there was indeed a causation, not just correlation—people would deliberately avoid movies with negative buzz.

Section 1.1.2: Beyond Films

The idea that negative impressions have more influence than positive ones is strongly supported by Baumeister (2001), in a review paper, published in the Review of General Psychology, entitled *"Bad is Stronger than Good".* The researchers surveyed a

wide range of papers in the field of psychology; among their conclusions were that "bad impressions and bad stereotypes are quicker to form and more resistant to disconfirmation than good ones," and that "hardly any exceptions (indicating greater power of good) can be found." This supports the conclusion, in Hennig-Thurau (2014), that bad word of mouth has significantly more impact on moviegoing decisions than good word of mouth.

Relevant research in prediction extends beyond that of predicting movie popularity. In Gruhl, et al., a 2005 paper analyzing the then-nascent blogosphere, researchers attempted to correlate blog mentions of books with each book's sales rank on Amazon.com. Their results held that there tended to be very strong correlations, and that volume of blog mentions could be predictive of future sales rank over a window of several days. Furthermore, the researchers proposed that sales rank may be a Markov process, and suggested using a fixed-length window of sales history to predict the following day's sales. As one question to be investigated in this project is whether film career success is also a Markov process, this is a relevant consideration.

Moving beyond popular media, we also have relevant research in the medical field. In Shipp, et al., researchers at Harvard Medical School and the MIT Center for Genome Research attempted to use machine learning to predict the outcome of diffuse large B-cell lymphoma cases. This form of lymphoma is the most common lymphoid malignancy in adults, and could be cured in less than 50% of patients at the time of

writing in 2002. Using machine learning through a hierarchical clustering algorithm, researchers were able to effectively delineate patients based on survival probabilities, and found two distinct classes – a group with survival probabilities around 70%, and another around 12%. This clustering was conducted through gene-expression profiling, by obtaining the gene-expression pattern of malignant and non-cancerous lymphocytes. Researchers then used the hierarchical clustering algorithm to split patients into two groups, such that one group's expression patterns were more similar to non-cancerous samples, and the other's were more similar to the malignant samples. In addition to separating the groups by survival probabilities, the clustering provided by the model was also able to help doctors identify proper therapeutic targets in patients, where earlier models were not able to provide useful information on this. In this way, machine learning was useful for finding relationships that were not previously apparent – an capability of machine learning that is key to this project.

Section 1.2

Methods for Identifying Leading Indicators of Success and Failure

The central concept for this project is that of learning how to find the films throughout the careers of actors and actresses which best indicate future success or failure. At this point there is reason to believe that past success or failure strongly affects an individual's future prospects; now it is necessary to find which films within one's career establish or prevent future success.

Section 1.2.1: Last Film and Whole Career

A straightforward approach to considering past careers is to simply average the popularity of every film that every actor and actress has done, and to find the correlation between that and his or her next film. A second approach is to only correlate every film in a given person's career with its immediate predecessor. These have been done in my prior research at this lab. As we can see in Figure 1.2.1, for both actors and actresses, the last film alone yields stronger correlations than the whole career average. This suggests that the impact to one's career of a single success or failure tends to diminish over time. This implies two possible mechanisms for film industry success. One possibility is that more recent films tend to be more influential, and that we can learn how to identify the films throughout an individual's career which are likely to have the greatest impact on future success. Alternatively, we may find that the very last film, specifically, is the best

predictor of future success—in this case, film industry success would be a Markov process.

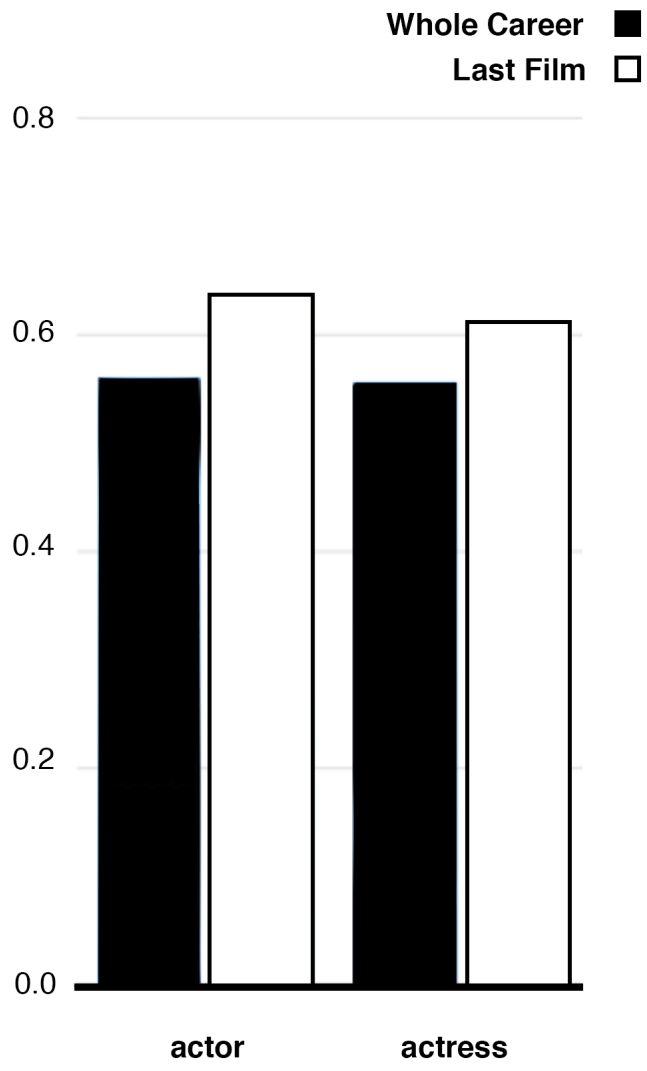


Figure 1.2.1 – Correlation Between Film Votes and Previous Film Votes

Section 1.2.2: Isolating Career Portions

The objective of finding leading indicators for future career success or failure can be carried out through machine learning, by finding specific portions of actors' and actresses' careers which have a tendency of being predictive. Consider films that are either positive or negative outliers, with respect to their popularity relative to that person's career mean. For actors and actresses whose reputation among the general public contributes to the success or failure of their work, we may suspect that especially good and bad films would be better remembered than average ones. Therefore, films that are particularly good or bad may do more to affect the likelihood that people will see or avoid, respectively, their next release. It should also be considered that as films become increasingly old, their popularity may need to be increasingly far from the mean to be remembered. Thus, we can make a model that includes only films that are increasingly far from the mean, as a function of time.

For an example, consider Figure 1.2.2, for a hypothetical director who is presently active creating films. The blue areas in the figure correspond to "memorable" outliers that would be included in the model, and the white area corresponds to "forgettable" films that would be excluded.

In addition to reflecting whether films are either memorable or forgettable, this model would also take into account the notion that more recent films are likely to better represent the current state of a person's work than older films. For example, a director who recently made a series of unpopular movies might be expected to make more unpopular ones, but someone who released a series of unpopular films thirty years ago may have since increased his ability to attract people to the theaters (or perhaps further decreased it). However, an extremely good or extremely poor film from decades past might still be valuable to include in a predictive model.

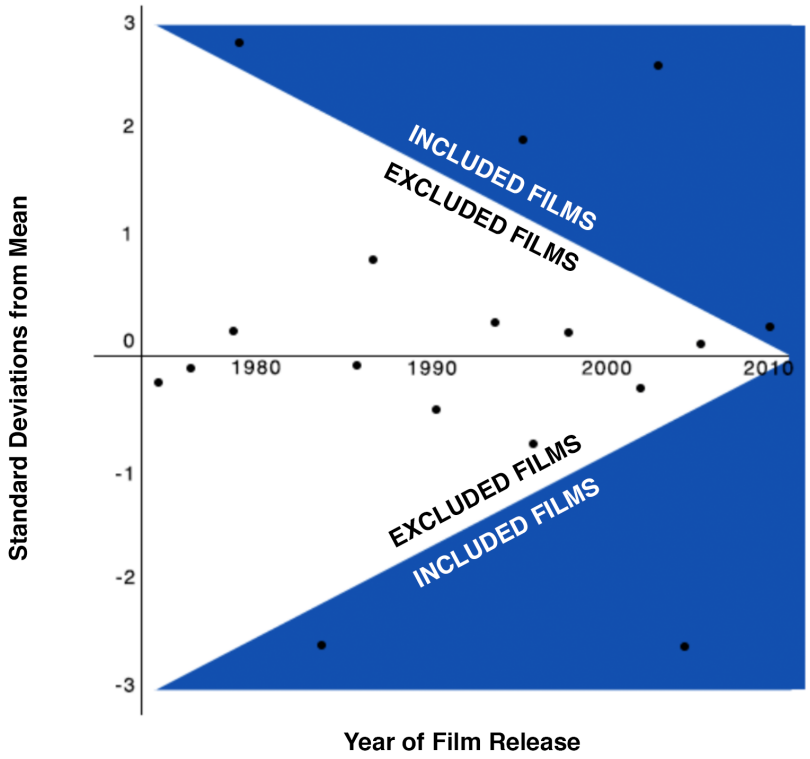


Figure 1.2.2 – Hypothetical Career Slice Method

The calculations required for this model can be easily be carried out algorithmically on a large data set. Furthermore, machine learning can be used to optimize the area in which films are included, allowing this to be a reasonable approach for this project.

Section 1.2.3: Rhythms of Success

Another means for identifying leading indicators for success is to examine patterns of popular and unpopular movies throughout an individual's career. For example, consider an A-list actor who likes to mix high-profile films with "passion projects" – small budget films, intended for a narrow audience, mainly done on the side for fun. If we can establish that this actor tends to follow up every high-budget blockbuster with one or two passion projects, we can assume that if his or her last film was extremely popular, the next one will likely not be.

These "rhythms of success" can endure for long throughout some careers. Other than passion projects, periods of predictable success and failure could result from those who tend to get burned out or uninspired, then have a period of productivity, in repeated cycles.

Identifying periods of popularity and unpopularity could be done on a large data set in a straightforward way.

A Naïve Look

The identification of periodicity could be done in several ways. An intuitive but naïve way would be to simply compute the percentage change in popularity from every

movie in a person's career to their next, then from every movie to the movie two later, then from every movie to the movie three later, and so on, and then comparing the average change for the one movie advancement against the average for two movie advancement, and for three movie advancement, etc.. For example, consider Ron Howard's career as a director – Figure 1.2.3 shows the normalized IMDb popularity of each of his films. There is an apparent periodicity in this graph. We can then compare the average percentage change for one movie ahead, two ahead, and so on (Figure 1.2.4). We can see that the average percentage change is least when comparing every film with the film three movies ahead (the standard deviation of each average is lowest for three movies ahead as well). This would suggest that the best single reference for predicting the popularity of a given Ron Howard movie, would be the movie he directed three films ago.

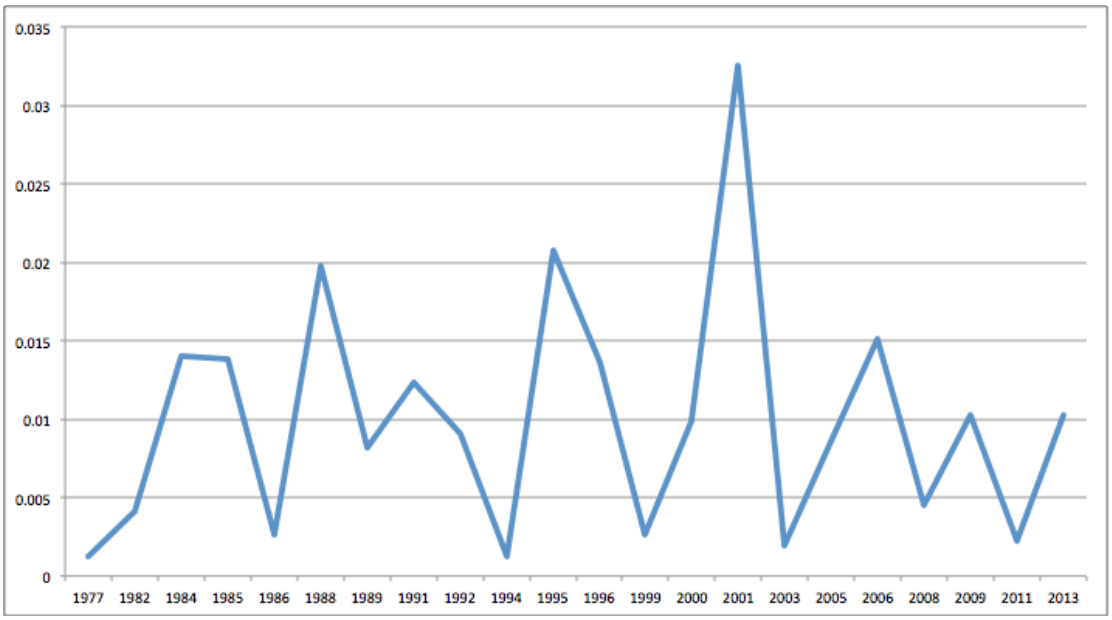


Figure 1.2.3 – Popularity of Each Ron Howard Film

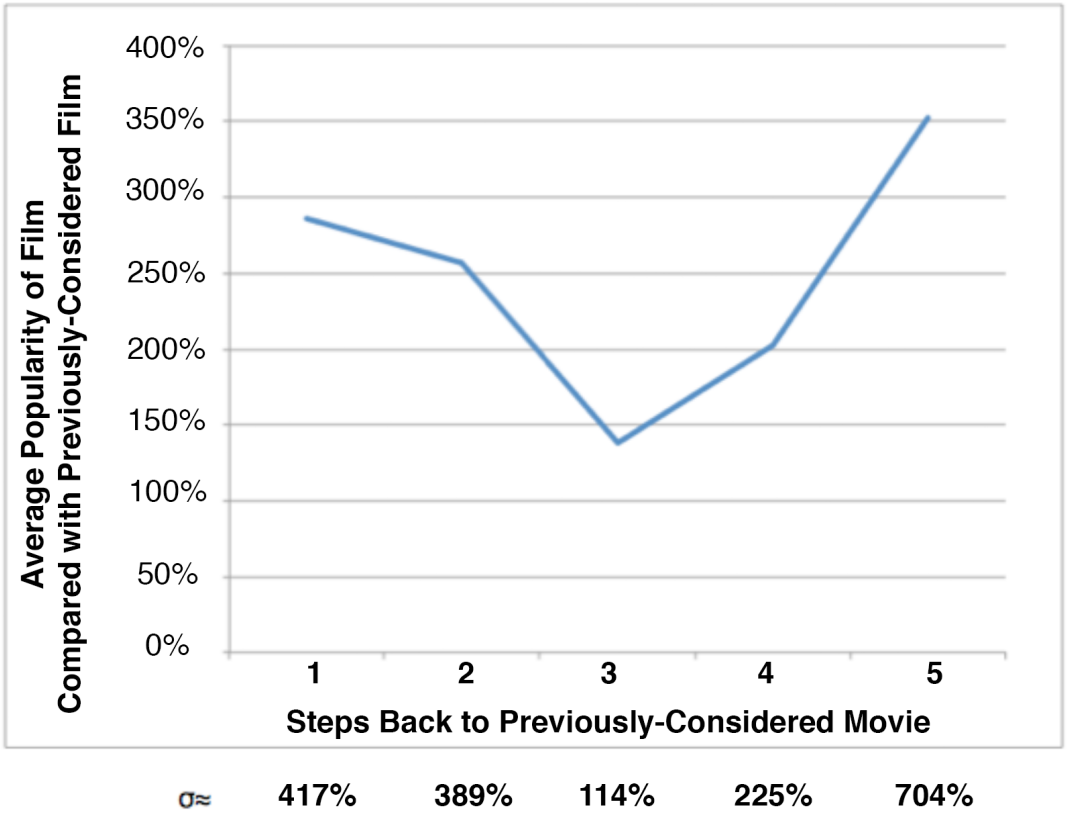


Figure 1.2.4 – Ron Howard Career Periodicity

Using Autocorrelation

Although the above method provides an intuitive look at the basis for rhythms of success, a more rigorous mathematical approach would be to use autocorrelation. For an autocorrelation equation between times *s* and *t*, we can use:

$$R(s,t) = \frac{E[(X_t - \mu_t)(X_s - \mu_s)]}{\sigma_t \sigma_s}$$

where X_i is the value of the series at time i, and E is the expected value. If R is well-defined, it will return a value in the range of [-1, 1]. (Dunn) Thus we can apply the autocorrelation function for the series of movies in every individual's career, and ascertain the strongest period length for each individual.

Section 1.3

Survey of Machine Learning Approaches

In order to optimize the sets of most predictive movies, and find a general method for identifying predictive films in an individual's career, we need an effective machine learning approach. This section discusses different machine learning approaches and algorithms, and their potential for use in this project.

Genetic Learning

Fundamentally, any genetic learning algorithm requires two elements:

- A representation of possible solutions in a format that can be easily modified and evolved.
- A fitness function to evaluate a set of potential solutions, and decide which are most effective.

Both of these elements can be devised to fit this problem. For the set of possible solutions, we have the set of all of the reasonable values of the parameters to separate films to be included from those to be excluded. We consider the set of reasonable values to be the set of all possible values, excluding any ranges about which we are comfortable assuming beforehand that they will not yield any useful results. These parameters will

include relative popularity compared to the mean for that individual's career, and age of the film. The fitness function will take the parameter values as an argument, find which movies to include from every career based on those parameters, then find the average popularity of those films, and finally correlate those averages for every given person with their next film. Higher correlations will always reflect a better result, so the algorithm will have a clear way to determine fitness.

The genetic algorithm aims to model genetic evolution. It begins with an initial naïve value for each parameter, such as 0, then creates a "population" of randomly mutated parameter values. Once it evaluates the fitness of each element in the population, the algorithm maintains a pre-set cohort of the strongest elements as the basis for the next "generation." It continues creating new generations of values randomly mutated from the preserved cohort, with the results regularly improving, and stops either when a predetermined fitness level is reached, a pre-set amount of time has elapsed, or when a set maximum number of evolutionary generations has been produced.

Genetic learning fits this thesis very well. Its largest drawbacks – the requirements for easily evolvable parameters and an objective fitness function – are straightforward to develop in this project. Therefore this class of algorithm is well-suited to my thesis, and represents the core of my methodology.

Association Rule Learning

Association rule learning is a form of machine learning, based on the premise that in a sufficiently large database, combinations between items can be used for predictions. For example, in retail sales, an association rule algorithm might be used to analyze a database of every sale over a significant period of time. It could then find a rule, perhaps {pens, paper} -> {highlighters}, suggesting that many people who buy pens and paper tend to also buy highlighters in the same transaction. This would then suggest that the retail establishment might consider selling highlighters near their pens and paper, since this is a common combination.

An association rule algorithm analyzes a given database and finds rules, typically through two distinct steps: First, the algorithm finds all frequent sets of items in the database, by an exhaustive search through all possible item combinations. This makes such a search potentially infeasible in practice, because the set of potential sets of items is the power set over the entire set of items in the database; the power set has a size equal to 2^n-1, where n is the number of items in the database. However, this is mitigated by the *downward closure property of support*, which holds that when we have a frequent set of items, all of its subsets are also frequent; therefore, if we have an infrequent set of items, all of its supersets must also be infrequent. Due to this property, it may be possible to efficiently find all frequent sets of items.

This form of data mining does have some relevance for this project, such as to determine if there are commonalities for movies that have specific combinations of popularity and unpopularity among their cast and crew. One salient example: do films with relatively unestablished directors and established actors perform better than films with established directors and unestablished actors? This question and others like it may help provide insight into the nature of human career success as pertains to the film industry. However, this class of machine learning was ruled out for this project, as association classification does not fit in closely enough with the scope of the question being investigated.

Artificial Neural Networks

Artificial neural networks are computational models based on the general functionality of the neural mechanisms of the brain and central nervous system. These algorithms are capable of learning and pattern recognition, adapting to data as they progress through an analysis. This approach is useful for tasks that are difficult to approach using typical rule-based programming, such as handwriting recognition and computer vision.

Artificial neural networks are designed around a series of simple nodes – artificial "neurons" – which each perform a simple function on the input, and act in parallel,

similarly to a biological neural network. Furthermore, every node has an individual weight to balance its impact on the final output. These weights are adaptive and are tuned by a learning algorithm, thus allowing the algorithm to change and optimize itself as it processes data.

Due to the ability of the algorithm to optimize itself, it is well suited to situations in which a set of rules cannot be predetermined by researchers. Therefore, it could be a useful way to approach the problem in this project. However, optimizing with a neural network is a computationally intensive process—due to the time constraints of this project, this approach was ruled out. It may, however, be a valuable method for future research.

Bayesian Networks

A Bayesian network functions through a probabilistic, directed graph model. This is based on a set of random variables that are weighted according to conditional dependencies from the initial directed graph, where the probabilistic model is a Bayesian probability function. For example, this could be used to solve the following problem: *Given the probability of rain on a certain day, and given the probability that the sprinkler would be going on a rainy day, what is the probability that the grass is wet?*

With the nature of this kind of algorithm, it is well-suited to investigating the presence of hidden states in data, such as that of a Hidden Markov Model. While this approach could be relevant to developing an understanding of the mechanics of the film industry, it is not well-suited for the set optimization problem central to this specific project, so it was ruled out for my thesis.

Support Vector Machine

Another powerful machine learning approach is that of a support vector machine. This method is useful if we have a set of data about which we know there is a hidden classification, but for which there is no simple way to determine the correct way of classifying individual items. For example, consider spam filtering: we know that every item is either spam or not spam; the objective is to classify individual messages accurately.[1] A support vector machine is a good way to approach these kinds of problems. An SVM can take a training set, for which the classifications are already known, and use the various dimensions of the data to predict the classifications for the remaining items. As pertains to this project, the most appropriate use for an SVM would be to investigate "hidden" traits, such as through the application of a hidden Markov model. However, it would be impractically difficult to create a training set, such that the SVM could find

---

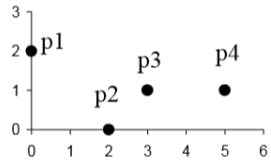
[1] Spam filtering is, of course, an especially difficult problem, because the proper classification varies from person to person. For one person, an promotional email from Amazon may be a useful way to learn about the week's deals; someone else may consider it junk.

relevant traits. For this reason, using an SVM in this project was determined not be an optimal approach.

Clustering

Clustering is a process which organizes a data set such that the items that are closest to one another – or are explicitly in the same group – are the most similar. It is common in tasks such as image analysis, bioinformatics and pattern recognition. Clustering can be carried out in different ways depending on the task to be done, so various algorithms and approaches exist.

A common approach is *connectivity based clustering* – also known as hierarchical clustering. In this, data points are organized such that the distance between points approximates their relative similarity; humans must then look at a plot and arbitrarily decide where one cluster ends and the next begins. Consider the following figure.[2]



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0.000	2.828	3.162	5.099
p2	2.828	0.000	1.414	3.162
p3	3.162	1.414	0.000	2.000
p4	5.099	3.162	2.000	0.000

Figure 1.3.1 – Sample Proximity Matrix for Connectivity-Based Clustering

[2] *Image adapted from http://www.cs.umn.edu/~han/dmclass/cluster_survey_10_02_00.pdf., pp. 7, Figure 2*

In this example, we have a set of four points: A clustering algorithm has determined the proximity of each point from every other point as a function of their respective original characteristics. For example, we can observe that the algorithm has determined that *p1* and *p4* are less similar than *p1* and *p2*. The graph simply shows these proximities visually. In this type of graph, the absolute location of each point does not matter – the information lies in the distance between points. With more points we might have visible clusters; the proximity of any point to the "center" of a cluster reflects the degree to which that point exhibits the characteristics that, in the algorithm's determination, most strongly define that cluster.

Unfortunately, this approach is not robust in the context of outliers, which will often either show up as more clusters or cause other clusters to merge together. For these reasons, in some circumstances hierarchical clustering is not reliable. However, this approach has valuable applications: for example, hierarchical clustering was used to predict lymphoma patient prognosis in Shipp, et al.

Another common approach to clustering is *k-means clustering*, which differs from hierarchical clustering in that it separates data into explicitly separate groups. However, the number of clusters, *k*, must be specified in advance; this is potentially a major drawback. (*Cluster analysis*) Another approach is *distribution-based clustering*, which defines clusters as being groups of objects which most likely all belong to the same statistical distribution. This method separates data points effectively, while allowing for

visually distinct but overlapping clusters. This method does not require a pre-determined number of clusters, and can be effective at capturing correlation and dependence between attributes.

While clustering would be useful in the context of general research in this area, it would not be optimal for this project. The best use for clustering would be for after we know which parts of a person's career are the most predictive—we could then generate clusters based on the predictiveness of different combinations of movie traits (age, popularity, and so on). While this could be useful, clustering would not be appropriate for the optimization process itself – that is, ascertaining the predictiveness of the various factors – which is the central aspect of this thesis.

Section 1.4

Research Goals

The current state of the research suggests that correlations between each movie in a person's career are strongest with that of the immediately previous film, rather than the sum of the popularity of every previous career film, or a randomly assigned movie from that person's career. This result implies that film success may have Markovian properties – where the best way to predict the popularity of a person's next film is simply to know the popularity of their last film, and to apply it to some prediction matrix which remains constant. In other words, it would imply that in Hollywood, everyone is "only as good as his or her last film." The chief purpose of the research in this proposal, then, is to understand where the predictive information lies throughout a person's career.

If the results of this research are such that predictions can be significantly improved by using prior career data, this would indicate that film success is not Markovian; furthermore, the specific films to include would suggest the ways by which success in the film industry is created and maintained. For example, the predictive importance of a single positive outlier from early in someone's career, may provide insight into the degree of long-lasting popularity a person may enjoy from one major hit movie.

Conversely, if earlier career data has a negligible impact on predictions compared with the last movie alone, this would suggest that film industry success may truly be a Markov process, and that a person's last film is in fact the major determinant of their future popularity and success.

Chapter 2

Data Sets, Software Design & Methodology

The software used in this project has two main components: the first aggregates film statistics from several databases and finds the correlations between the set of an individual's earlier work and their next film; the second component is built around a machine learning framework and optimizes the set of films. This section will explore those components in detail.

Section 2.1

Data Sets

The movie data used in this project is derived from several databases, all stored in a comma-separated value (*.csv*) format:

- films.csv [95217 rows, 5 columns]: A compilation of every American-made film, including movie ID, title, year of release, real number of votes, and normalized vote count.
- wikipediaMovies.txt [4,009 rows, 4 columns]: A database of every American-made film released from 2008 to 2014 which has an entry on Wikipedia. This was useful for

filtering the overall number of films considered, which will be discussed later in this section.

- relevantFilms.csv [35,232 rows, 9 columns]: A database of every role in every movie in *wikipediaMovies.txt*. Includes a reference to the next movie that the given person released, as well as type (actor or actress), normalized votes, person ID, and career average and size of a standard deviation up to the point of making that film. To improve computation time, this also includes references to where the person's entire career data, up to that point, is located in AllVotes.csv.
- allVotes.csv [419,943 rows, 7 columns]: A database of the career history of every actor and actress in *wikipediaMovies.txt*.
- films_actor.csv [64,565 rows, 3 columns], films_actress.csv [35,030 rows, 3 columns]: A database of every actor and actress by name, person ID, and movie ID, for cross-referencing purposes.

When the program is run, the data is loaded into memory, and duplicate or malformed entries are removed. Then all movies are cross-referenced with wikipediaMovies.txt, and any movies and cast not in the database—that is, not involved with movies with entries on Wikipedia, and released between 2008 and 2014—are removed. Note that this truncation only applies to the final group of movies the popularity of which we are trying to predict: when looking into the past career of every actor and actress in these movies, we will use the data for every film they were in, regardless of release year or Wikipedia entry.

This removal is done for several reasons. First, extremely low-budget films may not represent the behavior of the mainstream film industry, and exist in large enough numbers to risk a significant distortion of the results; by restricting our results to only films with Wikipedia entries, we limit our consideration to only mass-market and mainstream "indie" films. Second, the project is focused on understanding the film industry as it is today, so careers that ended several decades past may not represent current film industry behavior. Finally, there is a particular anomaly that we need to avoid: consider an actor who is active long before becoming famous, but upon reaching "stardom," his fans start to watch his earlier work, and rate those films more on IMDb. In this way the popularity of his earlier films is influenced by that of his *later* work, an anomaly which would heavily distort this analysis. We can significantly minimize this by restricting our consideration to only a slice of recently released films.

In relevantFilms, we have a row representing film popularity, as well as a row for a reference to the film that was released next in that person's career. By default, for each actor, the software finds the Pearson correlation between the popularity of every given film and that of the film that is listed to have been released next. When we want to find the correlations using the whole career average, instead of that person's immediately prior film, we simply replace the last film popularity in relevantFilms with an average of every film from the beginning of that person's career up to the film in question. It is easy to find this set to average, because relevantFilms includes a reference to where, in

allVotes, we may find the person's entire career history up to the point of that film. We can then find the average of the values in that set and find the correlation with next film popularity.

In order to reduce computation time, I used the *snowfall* package in R, which enables parallel processing. I implemented it to use parallel processing for the task of finding the films in each career which fit the given parameters, and allowed the machine learning framework to divide up its work in parallel as well. On my computer, which has 8 addressable cores, this reduced computation time by roughly an order of magnitude.

Section 2.2

Finding Correlations

The next objective is to create a function which can consider just portions of a person's career, to correlate the average popularity of only included films with that of the individual's next film. Recall the visualization in Figure 1.2.1. The chief concern for optimization is that of balancing granularity with computation time. On my personal computer, evaluating the correlation results for any individual set of career slice parameters takes about 5 seconds per run. We then have the objective of achieving a rigorous result while minimizing the necessary number of runs. For example, suppose we were to just focus on high granularity. Consider a basic brute force approach: we could remove the dividing lines in Figure 1.2.1 and simply divide the graph into 100 sections, where each section is either included or excluded. For every possible combination of included and excluded sections, we could find the average popularity of all of the films in the included sections. With this, we could then find which combination gives us the highest correlation, and thus have the best possible "map" for inclusion and exclusion. This would yield a very clear result. However, if we consider that every section can either be included or excluded, that gives us 2^{100} possible combinations, each of which must be evaluated. At 5 seconds per run, this would take about $2.0 \cdot 10^{23}$ years before achieving a result.

Clearly, then, computation time is an important consideration. However, the example given in Figure 1.2.2 is perhaps too simple, as it only focuses on outliers. There is also the possibility that the most predictive films would tend to be the ones that stay closer to the center of the graph, closer to the career average.[3] To address this, I applied some modifications to the basic concept from Figure 1.2.2.

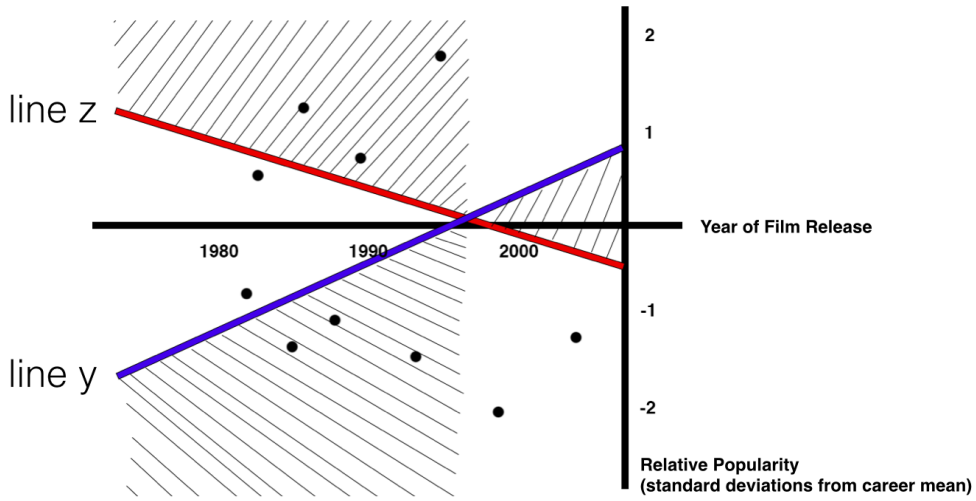


Figure 2.2.1 – General Career Slice Model

Similarly to Figure 1.2.2, we have the year of film release on the horizontal axis, and relative popularity on the vertical axis. Every dot represents a single film, and the plot as a whole reflects one individual's career. For the optimization, we simply have two independent lines, Line Y and Line Z. For any year on the plot where Line Y is greater than Line Z, the only movies to be included are those that fall between both lines. For any year where Line Y is less than Line Z, the only movies to be included are those that are

[3] *This could occur if people disregard memories of especially good or bad movies, such that they assume that an unusual performance may not be representative of that person's usual output.*

above Line Z or below Line Y. Finally, if Line Y ≈ Line Z, no films will be included, other than the most recent film.

The machine learning algorithm will move these lines, in order to optimize the set of included movies. This approach gives the algorithm the flexibility to either include just outliers, just movies near the mean, or (as in Figure 2.2.1) some combination of both. This has the benefit of requiring the algorithm to optimize few variables. Since Line Y and Line Z are simple lines, the algorithm just needs to optimize a slope and intercept for each; representing just four variables. This will keep the computation time to a minimum while allowing a good deal of flexibility for the algorithm.

In the current setup, every optimized set of movies contains that person's last movie—if the set would otherwise be empty (if Y ~ Z), then that would be the only movie used for correlations; otherwise, it would be kept alongside the movies in the included areas.[4] However, using the autocorrelation method described in Section 1.2.3, I found the optimal period length for every career. As the machine learning algorithm optimizes the career slices, it will also be checking to see whether the last movie should be replaced with the movie one period length before the next movie (the movie with which we are finding a correlation). Furthermore, some preliminary testing has shown that correlations may be stronger using the movie several period lengths back, so the algorithm will be given the freedom to test the correlations for any arbitrary number of

[4] *It would be removed, however, if the same movie is also in an included area, to keep from counting that movie twice.*

period lengths back, and find which provides the overall strongest result. This will be reflected in a new argument in *actor* and *actress* for number of period lengths to look back.

In the final code, the entire selection and optimization portion was written as a single function, *actor(a, b, c, d, P)* and *actress(w, x, y, z, p).* Prior research at my lab suggests that actors' and actresses' careers evolve differently, so it was prudent to optimize separately. For actors, *a* and *b* are the slope and intercept, respectively, of Line Y; *c* and *d* for Line Z. For actresses, *w, x, y, z* serve the same respective roles. *P* and *p* represent the number of period lengths to look back, where a value of 0 would instruct the algorithm to just use the Last Movie. Therefore we have a simple function to find the correlation results given any possible parameters. This function was then used as the fitness function for the machine learning algorithm.

Section 2.3

Machine Learning

After a considerable review of possible machine learning approaches (Section 1.3), I decided to do the optimization using a genetic learning algorithm. Recall that there are two necessary components of any genetic learning (GL) approach:

- A representation of possible solutions in a format that can be easily modified and evolved.
- A fitness function to evaluate a set of potential solutions, and objectively decide which are most effective.

It was apparent that my project would work well with this approach. The parameters in *actor(a, b, c, d)* and *actress(w, x, y, z)* can be easily modified in small, incremental steps, and are therefore well-suited for an evolutionary approach. Secondly, the results of *actor* and *actress* from each attempted set of parameters provide an effective fitness function for objectively evaluating each attempt.

The programming for this project was done entirely in R, and so finding an effective GL framework in R was a necessity. I decided to use a package called *rgp,* because it provides a good deal of flexibility and is well-documented. The following is an overview of how I chose to configure *rgp*, with explanations of each option and a brief justification.

geneticProgramming vs. multiNicheGeneticProgramming

The default option in this choice is *geneticProgramming*, in which which *rgp* will approach optimizations in the way as previously described: a set of possible parameters is randomly generated, then each possibility is evaluated, where the best one is stored for the next generation. In the next generation, random variations are made against the best of the earlier parameters, and a new set to evaluate is generated. After a sufficient number of generations, a highly evolved solution is returned.

multiNicheGeneticProgramming works in much the same way, except that separate, isolated groups are allowed to evolve concurrently. After a sufficient number of generations, we may have several effective but highly different solutions. Conversely, if the same solution is eventually found in each niche, this provides more confidence that that solution may be the only viable approach.

I chose *multiNicheGeneticProgramming*—if several vastly different but equally effective results are discovered, that can provide valuable information about the nature of film industry success.

stopCondition

Genetic programming typically provides several different options for *stopCondition*—the point at which the algorithm stop running and returns a result. Consider that, like the biological evolution process upon which it is modeled, the genetic learning process can continue indefinitely: there is no point when it is simply "done." Therefore, we must decide beforehand the point at which we will be satisfied.

The most straightforward stop condition is time: we can simply specify the number of seconds before finishing the algorithm and getting the results. Another option is number of evolutionary steps – the algorithm will stop after a set number of generations. Finally, we can have the algorithm run indefinitely until it reaches a pre-set fitness level, which in this case would be a certain correlation value.

Because time is a pertinent consideration in the context of a thesis project, I chose to set time as the stop condition. I ran separate evaluations for *actor* and *actress*, and in each case specified 172,000 seconds (about two days) for each run.

numberOfNiches

If using *multiNicheGeneticProgramming*, we can pre-set the number of niches in which to concurrently run the genetic evolution. For this project, I chose 8 niches, to match the number of addressable CPU cores on my computer.

populationSize

Specifies the number of individuals in each population – that is, the size of each genetic evolution generation. For this, I chose [final number].

Chapter 3

Results

The findings presented below are the product of the *rgp* implementation and data sets described in Chapter 2, following 125 hours of computation, encompassing approximately 5,000 evolutionary steps, and 200,000 fitness evaluations.

The expectation at the start of this project was that actors and actresses may exhibit very different career patterns—this held true in the final results.

Section 3.1

Actors

Recall from Section 2.2 that we optimized using two lines, Line Y and Line Z: in any year where Y > Z, the area of included films for correlation is between the two lines; where Z > Y, the included areas are those above Z and below Y; where Y ~ Z, there is no included area.

For actors, we have the following result, which we will refer to as **Actor Result**. For these equations, *x* equals film age:

Actor Result:

$$Y = 34.398x - 34.291$$

$$Z = 14.059x - 6.88$$

We have two plots of this result. In Figure 3.1.1, we have a typical range for both axes, in order to give a visual sense of how small the included area is. Because the relevant area is very small, we then have Figure 3.1.2, in which we see the plot zoomed in for greater detail, with the areas for included films highlighted.

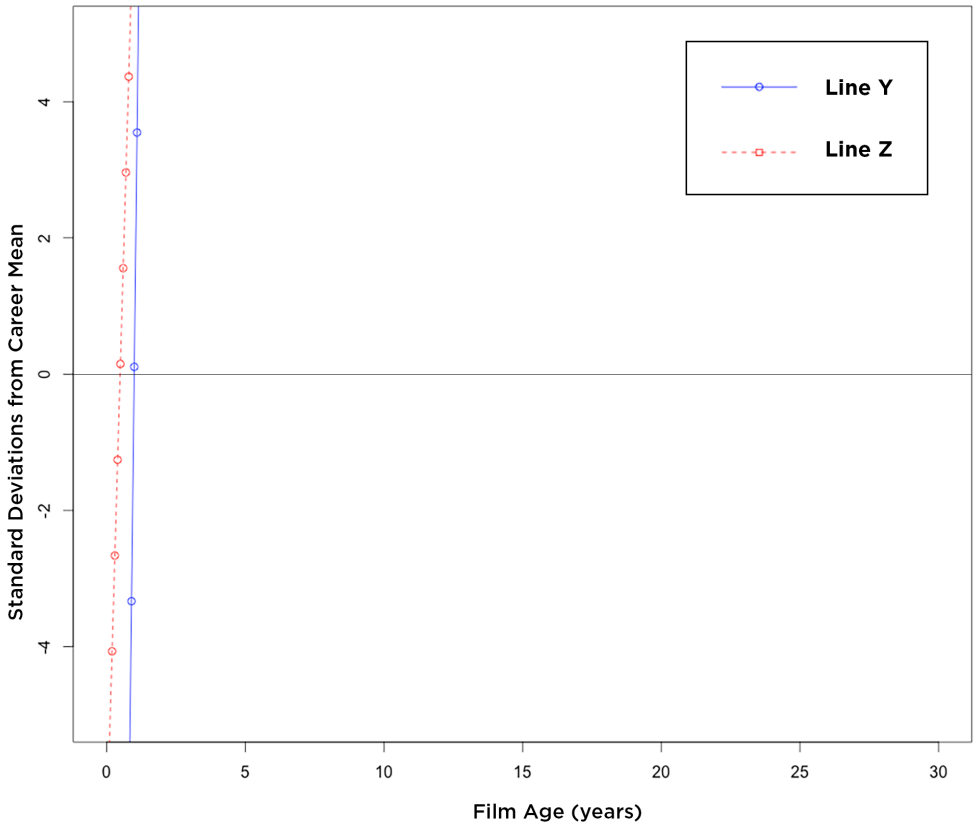


Figure 3.1.1 - Actor Result

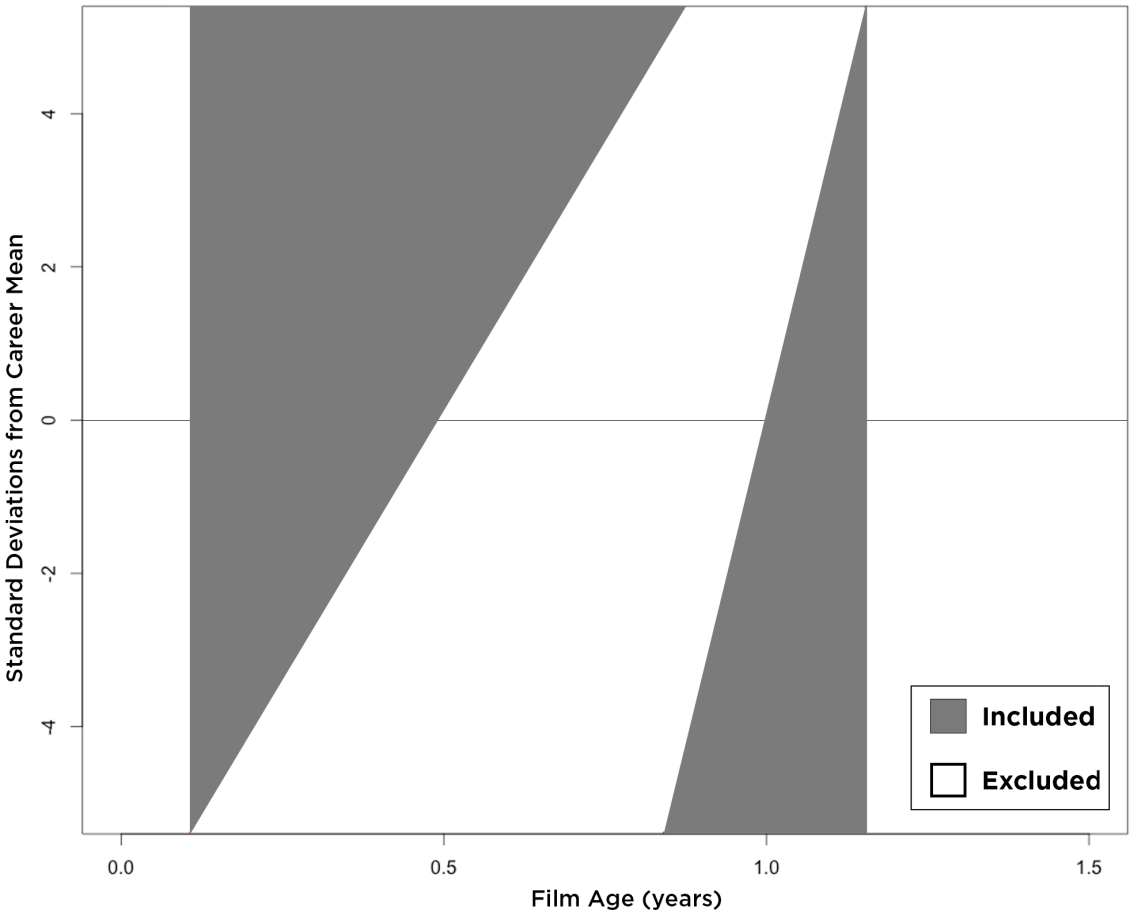


Figure 3.1.2 – Actor Result Zoomed

We can see by inspection that the area of inclusion is very small, confined to only some movies less than 1.5 years old. We see in Figure 3.1.3 that the relative improvement over Last Movie correlation is extremely negligible.

Actor Result provides a nominally better correlation than Last Movie; this is why it was found and returned. However, the improvement is small enough as to be incidental. The result is that the machine learning algorithm could not find a significant improvement to Last Movie through career slices. From this, we find that broad portions of an actor's career have little influence on his overall success or failure, with the very last film having as much predictive power as any career slice.

Recall from Section 2.2 that every set of included films includes the Last Movie, but that the algorithm has the freedom to replace that film with the one that was released one or more period lengths back. Note that in Figure 3.1.3, Actor Result reflects the correlations using the last movie. Here were see the results of looking back one or more period lengths.

With actors, we find that using looking back one or more period lengths, rather than including the last film, gives us a small improvement. The algorithm found that the optimal result is found when looking back 19 period lengths.[5] We have relative correlations in Figure 3.1.4, in which we see that the improvement from looking back using optimal period lengths is very small. Therefore we find that for actors, the very last film has virtually all of the predictive power of any other set of data.

---

[5] *This sounds large, but the median period length for actors is 2 movies, so for at least half of all actors this would entail looking back either 19 or 38 movies. Because many actors release several movies per year, in many cases the pivotal movie would be made in the past decade or so.*

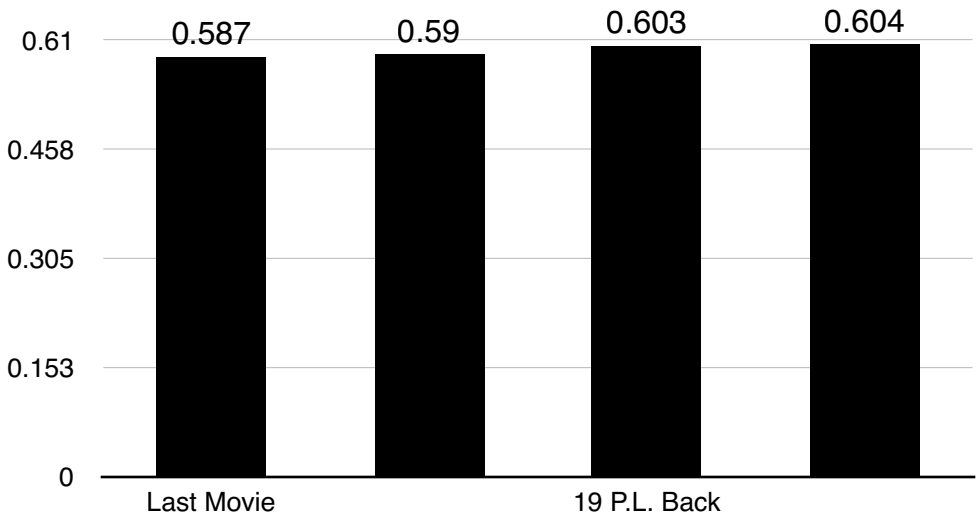


Figure 3.1.4 – Actor Correlations with Rhythms of Success

Section 3.2

Actresses

For actresses, the career progression is very different from that of actors. We have the following equations for **Actress Result.** Figure 3.2.1 is a plot of Line Y and Z, with the areas for included films.

Actress Result

$$Y = 0x + 0$$

$$Z = 2.12x + 0$$

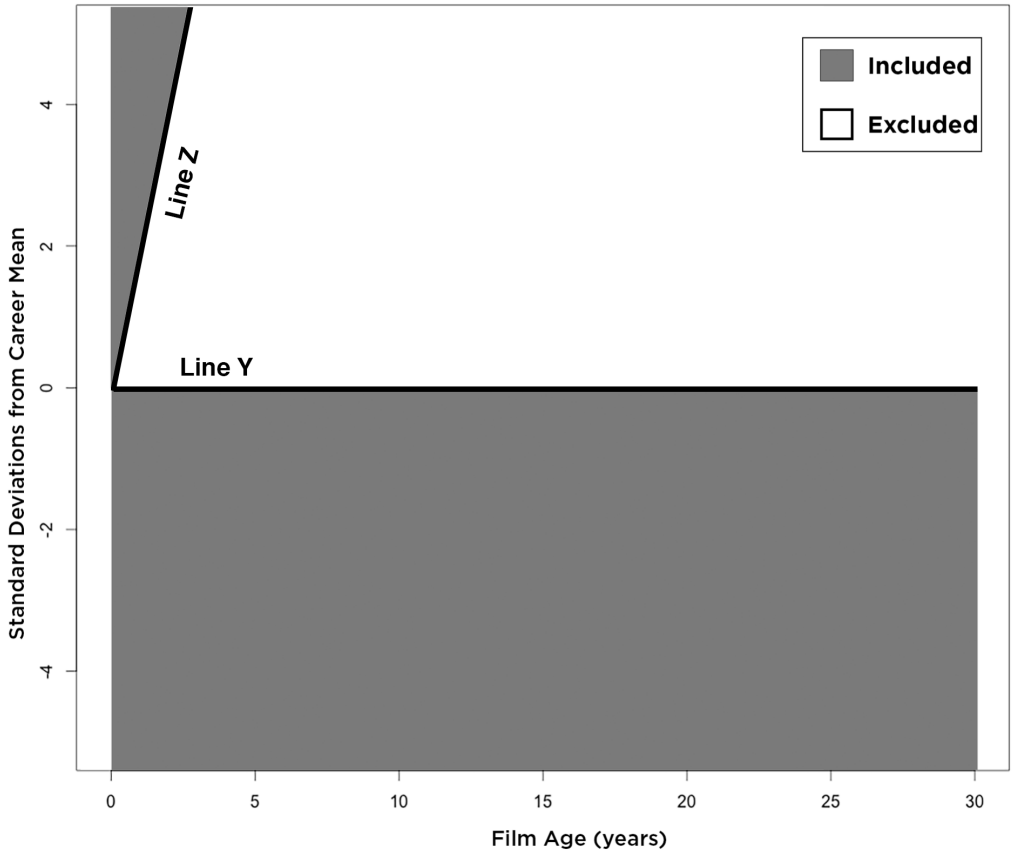


Figure 3.2.1 – Actress Result

Here we clearly see the relative influence of negative and positive impressions. Above-average films only provide strong predictive information if they are within a year or two old, with movies less than one standard deviation above the mean barely having any impact. However, poorly received movies contain strong predictive information regardless of their age or deviation from the mean.

We can see in Figure 3.2.2 the correlations from Last Movie, Whole Career, and Actress Result. For comparison, I have also included the correlation result of only above-average films. We can clearly see the power of negative impressions.

Next, we consider rhythms of success. For actresses, the best results were obtained when we correlated each movie with the movie two period lengths back. As we can see in Figure 3.2.3, when we correlate each movie with only the film two period lengths back, we have a correlation roughly equal to that of Actress Result. Therefore, we find that the same predictive information generally exists in the single movie two period lengths back, as in the entirety of the movies included in Actress Result. Furthermore, when we correlate each movie with the set of the included films in Actress Result, but replace Last Film with the movie from two period lengths back, we get somewhat stronger results.

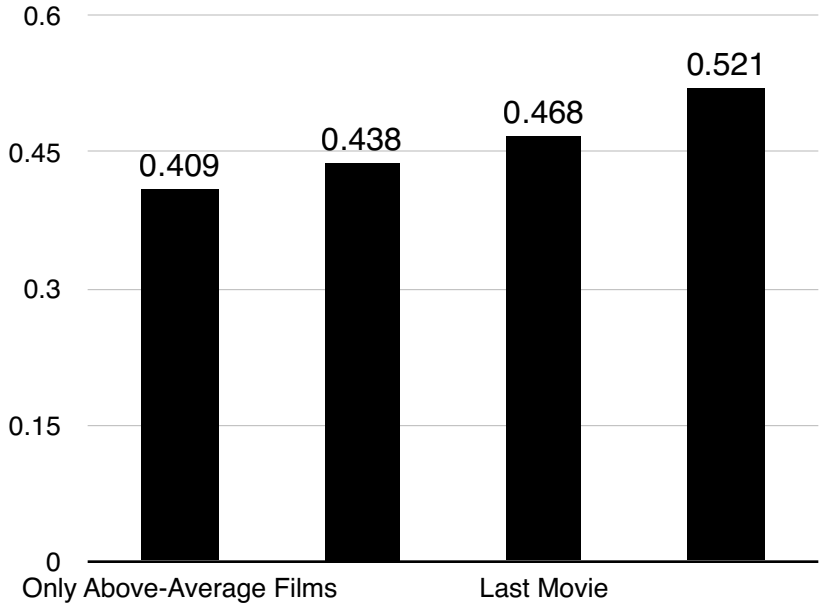


Figure 3.2.2 – Actress Correlations

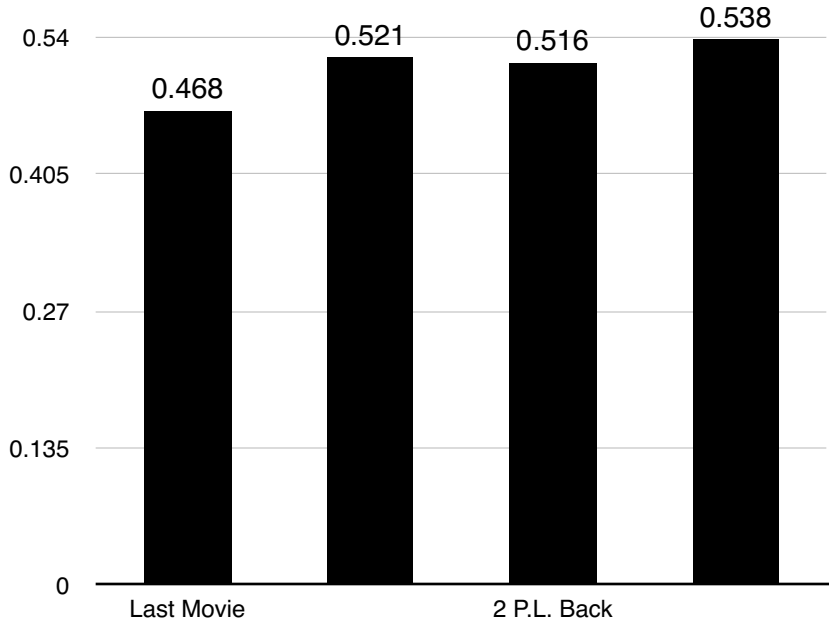


Figure 3.2.3 – Actress Correlations with Rhythms of Success

Chapter 4

Discussion

Section 4.1

Pivotal Career Moments for Actors and Actresses

In Chapter 3 we see that for actors, the very last movie provides correlations nearly as strong as any attempt to use earlier career history. This may provide the impression that for actors in Hollywood, "you're only as good as your last movie"—suggesting that an actor's most recent film will alone determine whether his career rises or falls. However, a closer examination rules this out, because certain earlier work can provide equally strong correlations as that of the last film. If it were true that an actor's success or failure is *determined* by his last film, then that last film should provide far better predictions than any other releases, because the last film would have the power to alter an actor's career trajectory. This is not the case. Instead, the last film should be seen as a *reflection* of a process that has already happened, based on a small number of pivotal successes or failures earlier in the actor's career. In this way, although the last film provides a strong predictive information for actors, it is unlikely to have the power to change the state that it is reporting.

Because actors' success is most strongly associated with a very small number of films, these films should be seen as the pivotal events in an actor's career trajectory. For actresses, there is less emphasis on pivotal "make-or-break" career moments, as success is based on broad trends and career history. From this, we can also conclude that an actress is not "only as good as her last movie."

Section 4.2

Negative and Positive Impressions

We see in Figure 3.2.1 that for actresses, well-received movies provide strong predictive information for future success for only a short period of time—although as one might expect, the longevity of this impact is directly correlated with how far the film is are above that person's career average. In contrast, poorly received movies have much greater longevity—this impact persists for the duration of an actress's career, regardless of how far below average the film falls. This affirms the findings in Hennig-Thurau (2014) and Baumeister (2001), that negative impressions have a far more powerful and long-lasting impact on decision-making than that of positive impressions.

Notably, we do not see the same pattern for actors, but because of the inability for *any* career portions, positive or negative, to improve predictions over the very last film. From this we can broadly conclude that moviegoers remember very few movies in an actor's past work, whether positive or negative, so that the ones they do recall become

very influential. With actresses, viewers largely forget their positive roles, but are very likely to remember disappointments for long stretches of time, and then act according to those long-held impressions when an actress's new movie is released.

This represents a significant difference in career development based on gender, and we might consider it valuable to explore its root social causes. This is, of course, well outside of the purview of this project, but could make for a meaningful investigation in sociology or anthropology.

Section 4.3

Markov Process for Career Success

A question formulated at the beginning of this project is whether film industry success could be a Markov process. This would be if success is a "memoryless" process, such that we can make the most accurate predictions for future success based only on the success of a person's most recent film.

We can see that for actresses, success is definitively not a Markov process. This is evident from the results in Figures 3.2.3 and 3.2.4, where we can see that the Actress Result optimization and the rhythms of success method both provide notably better correlations with future success than that of last film alone—ruling out the possibility that actress success is a memoryless process.

For actors, the status as a Markov process is less clear. We know that the last film alone provides correlations that are extremely close to that of the Actor Result optimization and the rhythms of success method. The minor improvement by the latter two methods has little practical benefit, and may be the result of a coincidental arrangement of the data that resulted in a trivially improved correlation. However, we recognize that this could alternatively be an indication that some of an actor's earlier work does have predictive information not contained in the last film; even if this only improves predictions by a slight degree, it would still make actor success decidedly not a Markov process. It is for this reason that while these results raise the prospect that actor success is Markovian, they do not conclusively indicate it as such.

Section 4.4

General Observations

From these results, we find that an actress looking to build a successful career in the film industry would be well-advised to exercise considerable discipline, in minimizing the number and magnitude of her poorly-received films. This suggests a process that would discourage risk-taking, as failures are remembered long after most successes are forgotten. A positive aspect of this is that for an established actress, with a history of many well-liked movies and few poorly-received ones, her career will be

unlikely to be harmed from a small number of bad films—viewers would tend to remember her overall career trends and not just a few projects.

Actors face a different situation. As we've found, actors' career trajectories are the result of a small number of long-remembered films. This encourages risk-taking, relative to that of actresses' careers. Most failures will be forgotten (along with most successes), so actors face a smaller downside in taking on a risky project that could be a hit. The negative aspect is that even a successful actor could have his career damaged or ruined by a small number of poorly-received films, if those become his "make-or-break films," because people will tend to disregard most of his other output.

While these results show distinct, broad trends, we expect that some individual actors and actresses will have careers that are less reflective of these results. We may consider the career trajectory of Parker Posey, whose fame is not derived from a small number of key films, but rather from significant roles in many small productions. A character actor, like Steve Buscemi, may have built his success more on his unique acting style than on a few notable roles. Finally, because films derive their success from various factors outside of cast and crew, many individual films will succeed or fail on merits outside of its associated actors.

These results, then, show distinct trends on a large scale: while some careers and films will not reflect these decision-making patterns, many will. For this reason, future research should focus on developing a greater understanding of how films become make-or-break in an individual's career. We would also benefit from an understanding of the means by which poorly-received pivotal films could be replaced in the public consciousness by well-liked ones. In this way actors could mitigate the impact of risky films to their careers, and even alter their career trajectory after the fact.

References

Asur, S., Huberman, B. (2010). Predicting the Future with Social Media. *2010 IEEE/ WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.* doi:10.1016/j.apenergy.2013.03.027

Baumeister, R., Bratslavsky, E., Finkenauer, C., Vohs, K. (2001) Bad is Stronger Than Good. *Review of General Psychology*, *5*(4) 323-370. doi: 10.1037//1089-2680.5.4.32

Berkhin, P. (2006) A survey of clustering data mining techniques. Grouping multidimensional data: Recent Advances in Clustering. *Springer Berlin Heidelberg*, 25-71. doi: 10.1007/3-540-28349-8_2

Chakrabarti, A., Sinha, S. (2013) Self-organized coordination in collective response of non- interacting agents: Emergence of bimodality in box-office success. arXiv: 1312.1474v1

Dellarocas, C., Xiaoquan, Z., Awad, N. (2007) Exploring The Value of Online Product Reviews In Forecasting Sales: The Case of Motion Pictures. *Journal of Interactive Marketing, 24*(1). doi: 10.1002/dir.20087

Doshi, L., Krauss, J., Nann, S., Gloor, P. (2010) Predicting Movie Prices through Dynamic Social Network Analysis. *Procedia Social and Behavioral Sciences, 2*(4), 6423-6433. doi: 10.1016/j.sbspro.2010.04.052

Dunn, P. (2010) *Measurement and Data Analysis for Engineering and Science*. Edition 2. Boca Raton, FL: CRC Press

Gruhl, D., Guha, R., Kumar, R., Novak, J., Tomkins, A. (2005) The predictive power of online chatter. *Proceedings of KDD '05*. 78-87 doi: 10.1145/1081870.1081883

Hennig-Thurau, T., Wiertz, C., Feldhaus, F. (2014) Does Twitter Matter? The Impact of Microblogging Word of Mouth on Consumers' Adoption of New Movies. *Academy of Marketing Science, 43*(3), 375-394. doi: 10.1007/s11747-014-0388-3

Levin, A., Levin, I., Heath, C.E.. (1997) Movie Stars and Authors As Brand Names: Measuring Brand Equity in Experiential Products. *Advances in Consumer Research, 24*. 175-181.

Lorenz, J. (2009) Universality in movie rating distributions. *European Physical Journal B*, *71*, 251-258. doi: 10.1140/epjb/e2009-00283-3

Mestyán, M., Yasseri, T., Kertész, J. (2013) Early Prediction of Movie Box Office Success Based on Wikipedia Activity Big Data. *PLoS ONE 8*(8). doi:10.1371/journal.pone.0071226

Mitchell, M. (1999) *An Introduction to Genetic Algorithms*, Cambridge, MA: MIT Press

Sharda, R., Delen, D. (2006) Predicting Box Office Success of Motion Pictures with Neural Networks. *Expert Systems with Applications 30*(2), 243-254. doi: 10.1016/j.eswa.2005.07.018

Shipp, M., et al. (2002) Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine, 8*(1), 68-74

Sreenivasan, S. (2013) Quantitative analysis of the evolution of novelty through crowdsourced keywords. *Scientific Reports 3*, Article 2758. doi:10.1038/srep02758

Tan, P., Steinbach, M., Kumar, V. (2005) *Introduction to Data Mining*. Boston, MA: Addison-Wesley, Edition 1

Uzzi, B., Spiro, J. (2005) Collaboration and Creativity: The Small World Problem. *American Journal of Sociology 111*(2), 447-504

Wong, F., Sen, S., Chiang, M. (2012) Why Watching Movie Tweets Won't Tell the Whole Story, *Proceedings of the 2012 ACM Workshop on Online Social Networks*, 61-66. doi: 10.1145/2342549.2342564

Wuchty, S., Jones, B., Uzzi, B. (2007) The Increasing Dominance of Teams in Production of Knowledge. *Science 316*(5827), 1036-1039. doi:10.1126/science.1136099

Yu, X., Liu, Y., Huang, X., An, A. (2012) Mining Online Reviews for Predicting Sale Performance: A Case Study in the Movie Domain. *IEEE Transactions on Knowledge and Data Engineering, 24*(4), 720-734. doi: 10.1109/TKDE.2010.269