Graph-Based Navigation of a Box Office Prediction System

Mat Kelly*, Michael L. Nelson†, Michele C. Weigle‡
Old Dominion University
Norfolk, Virginia 23529, USA

ABSTRACT

Predicting movies' box office performance is a bit like gambling and predicting the stock market: there is a lot of luck involved but analyzing the system's input frequently leads a better result. There are many inputs to box office prediction but certainly among the most influential in predicting future success are the actors involved and their past performances. Grasping the variables needed in the large data sets to make a box office prediction is an overwhelming task where a visualization would be useful. In this paper we describe a means to visualize these large data sets and distill them down into a simple causal approach that makes understanding the factors that affect box office performance easier.

1 INTRODUCTION

Predicting ticket sales and viewer ratings of movies at the box office is a largely intractable, multivariate problem that requires an extensive ontology and some degree of luck. As the problem is complex, visualizing the factors that contribute to a movie's success helps to understand the weight that each parameter contributes. In this paper we describe a visualization approach currently being developed to accomplish the task of visualizing a subset of data from a large collection where the subset is used to describe a measure (product) representative of the larger set.

2 RELATED WORK

Shneiderman noted that many successful products embody the way of "Overview first, zoom and filter, then details-on-demand" in terms of design and interactivity [5], facets we used as the basis for the visualization's design. Herman and Holten [1, 2] each went into details about graph-based navigation as well as the concerns of perspective and traversal. We utilized this work in the visualization's context switching. Raitner formalized the structures that make up navigational graphs and optimized traversal methods [4]. Panaligan used modern data mining techniques to predict box office success based on research done before the fact by potential viewers [3].

3 APPROACH/DESIGN

As the problem of movie prediction is multivariate, performing dimensionality reduction allows us to take control of the problem. Much like stock market prediction, the number of variables that affect the outcome is largely infinite and stochastic, which allows us to develop a model to exploit the variance caused by randomness while harnessing the causal nature of a movies' attributes to its box office take and rating.

We put forth an intentionally naïve approach of ignoring many parameters for the movie's success with our design and focused on a short list of contributory inputs relating to each movie: the top three actors, the leading writer, the leading director, and the previous success (rating and box office take) of each considered entity (first three in this list) in their respective other movie endeavors.

*e-mail: mkelly@cs.odu.edu
†e-mail: mln@cs.odu.edu
‡e-mail: mweigle@cs.odu.edu

3.1 Strategic Data Collection and Cleaning

We considered a limited corpus of movies as a basis for our algorithm with the intention of the resulting predictive scheme and visualization to be extrapolated to other movies. From our data set (see Section 5) we pivoted on the movies provided and proceeded with the data collection phase. For each movie we obtained the first three actors from the movie's billing. For each actor in the movie, we obtained the sub-corpus of previous movies with which the actor was involved. We trimmed this sub-corpus to only movies in which the focused actor was billed in the first three actors on the movie's list of actors. The purpose of this was to reduce the dimensionality to the degree that would be useful for our heuristic and no further. For each movie that each actor was in (as limited by the above), we repeated the process of selecting the first three actors and performed the operation recursively. As the corpus is progressively built, data about each movie is retained including the critics' rating of the movie (on a scale from 0.0 to 10.0) and the box office take on the opening weekend of the movie (in U.S. dollars, or converted if in another currency).

3.2 Visualization Design

We wished to organize the information that we chose as a focus in a simple, clean way that allows the user to explore each parameter to our prediction as well as how each parameter contributed to the prediction results. Our initial mockup added each of the elements listed above (movie in-focus, movie's actors and each actor's previous movies) in a clean, dynamic form, hiding parameters we believed were beyond the scope or not displaying them in the interface thereby implying their lack of contribution toward our prediction.

With movies having a natural precedence of the people involved in their creation, we considered this through the sub-corpuses created to generate a degree of prominence that an actor's inclusion provided while maintaining the display of billing order. Though it is likely that billing order (the relative position in the credits where an actor's name resides) closely correlates to prominence, we maintained precedence to ensure inclusion of the actor (recall, we only consider the top three actors) in the output regardless of prior work.

The user interface elements are node and link based as shown in Figure 1. The center contains the movie (**Movie Circle**) in focus by the user with its title shown (top half of circle) as well as our prediction results (bottom half of circle), including Viewer Rating and Box Office Take. Along the top half of the circle are attached three nodes (**Actor Nodes**) symbolizing each of the three contributing actors. The actors are arranged in decreasing Actor Prominence from left to right, per our computation. The radius of each node signifies the actor's impact toward the success of the film. The information within the Movie Circle represents the product of the calculation using the values represented by the visual attributes of the connected nodes.

Emanating from each Actor Node toward the closest border of the visualization is a colored area (**Actor Swath**). The size of each Actor Swath is relative to the calculated prominence of the actor from past films within the accumulated corpus. The area of the swath of each actor, relative to the other actors' Actor Swaths at the same distance from the respective Actor Node, is representative of the relative cardinality of the impact of the respective actor's previous films.

Attached to the bottom half of the Actor Node are two categories

of contributory inputs, the writers and the directors. For simplicity, we only consider the lead of each but make the secondary of each available (not shown) in the user interface for navigational and exploratory purposes.

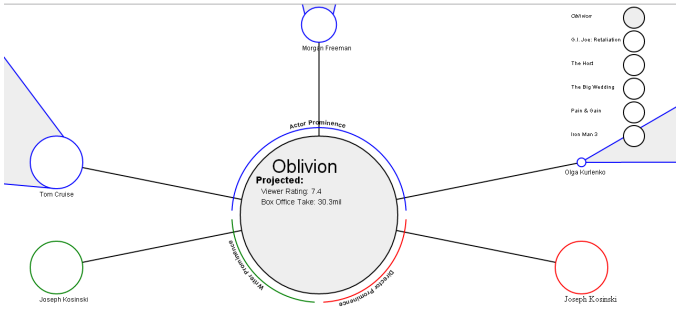

Figure 1: The visualization simplifies box office performance prediction into a simple black box system whose inputs are greatly reduced. By providing an exploratory interface, users are able to gauge the weight of each component (actor, writer, director) toward the end result through a node-and-link based navigational system.

4 Interactivity

As the purpose of the visualization is to provide an exploratory means of navigating the data used to produce a product when aggregated (e.g., Actors, Directory, etc. to Box Office Take), we have provided the facilities to get more information on the entities (represented as nodes) used as parameters and their respective impact on the product. When a user hovers over any node, an actor for instance, more information for the actor is displayed to the user within the node. Clicking on a node puts that entity into focus, e.g., clicking on an actor changes to an actor-centric perspective rather than the movie-centric one in Figure 1. Likewise, clicking on a swath associated with a node puts the visualization in a mode centered around the entity but in the context the entity plays in the current visualization. For example, if an entity has both acted and directed various movies, selecting the swath will change the focus of the visualization on the role the entity had in the movie currently active in the visualization. Likewise, selecting the swath attached to a writer or director will put the entity into focus in the context they played in the active movie rather than the entity's complete works, as is accessible by selecting the entity itself rather than the swath.

Selecting the center node that displays product information gives the user insight into how the product was formulated as well as additional information that is not easily encoded in the surrounding entities. In our use case of Box Office Take and Rating prediction, this information might contain the quantified weights that each entity formulaically contributed as well as any additional buffer or scalar used. Allowing for these values to be customized by the user would provide additional insight into how the parameters (e.g., actors in our example) contribute to the product as well as adjusting the formula to be more accurate when applied to future products (e.g., yet-to-be-released movies).

5 The Data

The seed data was from IMDB via .list database dump files with secondary information obtained through an unofficial IMDB API[1]. With the data obtained, we establish a basis for prediction using the the actors, business, movies and ratings .list files in the IMDB repository. This data set, though unwieldy due to its size, allowed us to establish links between the movie entities and to create a hierarchy between actors, movies, rankings and box office take. After aggregating the data from the two sources, we manually cleaned the resulting data set and transformed the data into JSON via a Python-based scipt. With the original development of the project being for the IEEE VAST challenge[2], opting to use data beyond the restrictions imposed by the contest allowed us to focus on the visualization in lieu of not being able to compete in the contest, which focused more on the prediction.

6 Implementation/Tools

The graphical and interactive portion of the visualization was built using D3.js 3.1.6[3], jQuery 2.0.0[4] and glue JavaScript. We performed data cleaning by scripting in Python[5]. The IMDB API was queried with Python (after encountering the JavaScript Same Origin policy restriction) and the IMDBAPI.org (an unofficial partial wrapper to the IMDB API) API was queried with JavaScript. The IMDbPY[6] library was evaluated and initially used but we found compatibility inconsistent and resorted to manual data cleaning.

7 Caveats

In the interest of computational efficiency, we excluded some key factors that might have contributed to a better prediction system.

- We only considered the 5 most recent movies by the actor to minimize the temporal complexity of the calculation. This likely affected actors who had many contributions where only the recent ones have flopped (e.g., Steve Buscemi).
- We excluded short films, foreign films, and movies whose initial release was not in the United States, as IMDB frequently did not contain the required information necessary to perform the prediction.
- We derived our corpus from the given set of movies and recursed from there, making it likely that new movies without all of the movies being in our corpus would require the corpus to be augmented prior to giving a prediction.

8 Conclusions

We believe our method of visualizing the large quantity of data that made up our sample corpus would suitably scale to a larger dataset. The application of the visualization to show the weight various parameters contribute to a predictive algorithm is only one potential use case for this method to represent the data. With more consideration for edge case movies that might require a different representation of parameters to effectively show each parameter's importance to the movie, the visualization serves as a good starting point for applying visualization principles to the large data set for the sake of box office prediction and data navigation.

References

[1] Herman, I and Melancon, G. and Marshall, M.S. Graph Visualization and Navigation in Information Visualization A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43.

[2] Holten, Danny. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Transactions on Visualization and Computer Graphics*, (5):741–748, 2006.

[3] R. Panaligan and A. Chen. Quantifying Movie Magic with Google Search. Technical report, Google, June 2013.

[4] Raitner, M. *Efficient Visual Navigation of Hierarchically Structured Graphs*. PhD thesis, University of Passau, 2004.

[5] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization. In *Proceedings of the IEEE Conference on Visual Languages*, pages 336–343, September 1996.

[1] http://imdbapi.org

[2] http://ieeevis.org/year/2013/info/call-participation/vast-challenge

[3] http://d3js.org/

[4] http://jquery.com/

[5] http://www.python.org/

[6] http://imdbpy.sourceforge.net/