FilmFinder 2.0: One Meaningful Visualization from IMDb's Raw Data Files

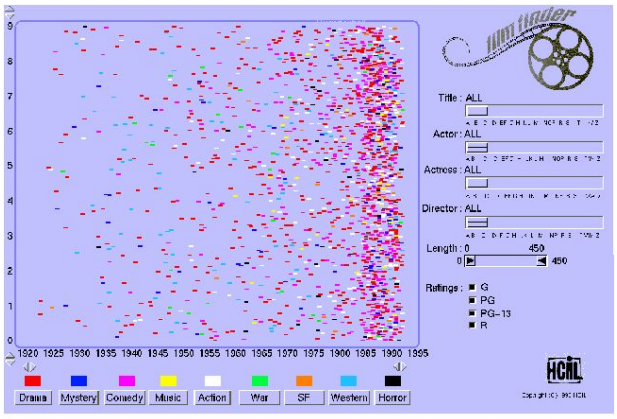
Sarah Wymer
Computer Science, B.S. 2017
swymer@stanford.edu

ABSTRACT
There is currently no way to view all of IMDb's raw text files in a meaningful way. My project is meant to use all of this data to create a new and improved FilmFinder that gives the user more freedom, e.g. letting them search for more than one actor's work at a time. Keeping Tufte's design principles in mind, I parsed IMDb's data into JSON and implemented FilmFinder 2.0 usind d3.

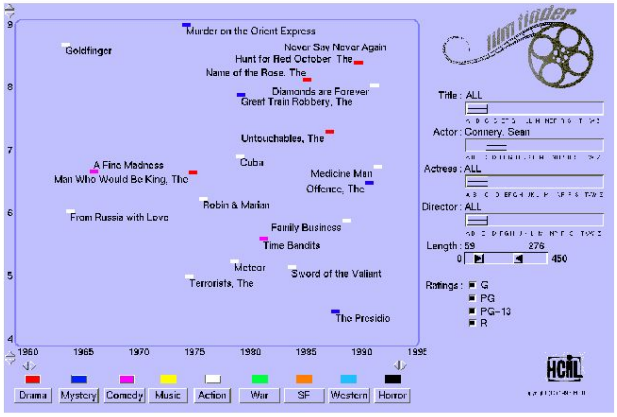
INTRODUCTION
IMDb puts all of its data in raw list files available for download on their website. Currently, there exists no way to visualize all of this data in one place. This could be because IMDb has not provided an up-to-date API for searching this data. The primary motivation behind this visualization project is to take IMDb's raw data dump and put it all together in one meaningful visualization tool. This would make the data files human-readable, allowing users to easily navigate the data and see patterns that aren't visible in list files, or even in IMDb's current interface.

RELATED WORK
There were many directions I could have gone, but I chose to use the data to improve upon the FilmFinder tool. This tool was mentioned in *Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays* (Ahlberg & Shneiderman). This prototype used dynamic queries and filters to allow users to search for films to watch. My project, FilmFinder 2.0, was designed to be more visually appealing and offer more freedom to the user.

For example, the old FilmFinder only allows the user to select one actor, one actress, and one director at a time. This makes sense if the user is only interested in the work of a single actor, but perhaps the user would like a movie with either Meryl Streep or George Clooney, as long as the film is recent. My FilmFinder 2.0 tool would allow the user to select as many actors or directors as they want at a time. On top of offering more user freedom, this tool was also designed to use color less liberally in an attempt to achieve better aesthetics.



Ahlberg & Shneiderman, Color plate 1. The FilmFinder.



Ahlberg & Shneiderman, Color plate 2. Categories have been selected, the displayed is zoomed into 1960-95 and popularity 4-9, and Sean Connery has been selected.

METHODS
The work I did on FilmFinder 2.0 could be divided into three steps: 1) designing the interface, 2) parsing the IMDb data files into one large JSON file, and 3) implementation.

Design
After the progress presentation, a peer asked me, "How would Tufte make FilmFinder?" This motivated the visual design of my project. I wanted to limit the use of colors unless they were meaningful. I also aimed to make a clean, geometric design with liberal use of white space.

I wanted the interaction design to be similar to the SF Crime Data project we completed earlier in the quarter. I wanted to make use of filters to create a dynamic tool. I surveyed my peers to see what they think about when trying to decide on a film to watch, and most of them mentioned the movie's popularity, how recent it was released, and its genre. Thus, movie data points are plotted on a graph of rating vs. release date, and the genre filters take up a large amount of visual weight.

Parsing IMDb files

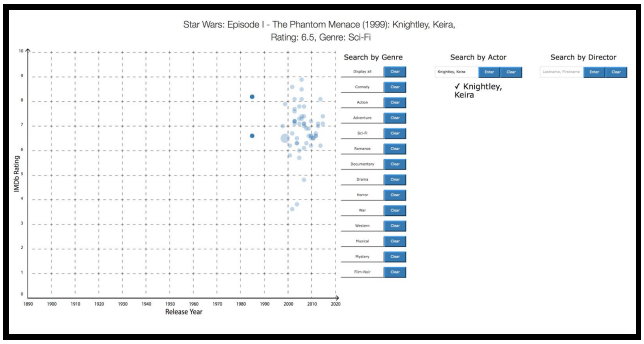
The list files were large, some reaching 19 million lines of text. In order to parse these files, I coded my own programs in C++ to split, parse, and merge code into one JSON file. The final JSON file had the format: {"Title": --, "Year": --, "Genre": --, "Rating": --, "Actors": [], "Director":--}. I used JSONLint to verify the JSON file as I went.

Implementation

I created FilmFinder 2.0 using Javascript, HTML, CSS, Bootstrap, and D3. I used the same methods and algorithms I had used for the SF Crime Data project. For example, when a user selects one genre, the program searches the JSON array for all films with that genre. For each one of those films, a data point is plotted and mouse listeners are added. These listeners are used to show popups of data whenever the user mouses over a data point.

RESULTS

FilmFinder 2.0 currently only uses title, year, cast, director, and rating data. When users want to find a film, they generally search by two or three actors or directors they like, then surveying the results. Most users who were not looking for films to watch were interested in searching their favorite actor or director to see a visualization of their career.

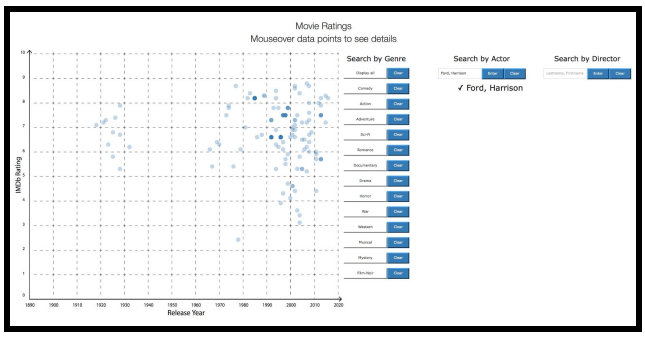


Drawback

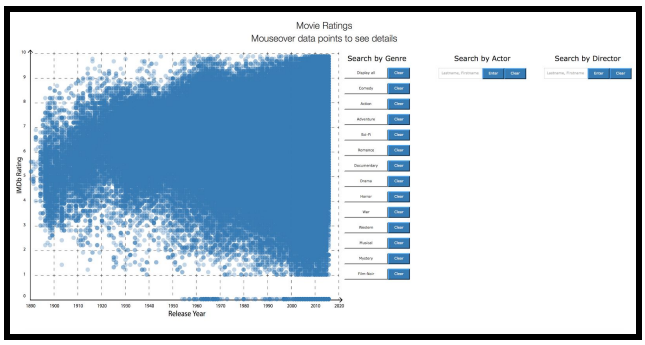
At the moment, d3 is not ideal for large swaths of data. This causes the tool to take around 4.19s to load in Safari.

DISCUSSION

It was interesting to me that when users played around with the tool, they ended up abandoning their original intentions and analyzing trends that they could see in the data.



For example, one user searched Harrison Ford, and was shocked to learn that there were two groups of data points: one around the turn of the century, and one around the 1920s-1930s. They learned that there was once another successful actor named Harrison Ford!



At one point, one user clicked "Display All" in the genres category. This made all of the data appear. The first conclusion someone might draw from this visual is that there is too much data to see anything. However, the user noticed that movies were rated lower the more recently they were made. They believed this to be because no one on IMDb would have the heart to rate an old movie poorly, because films were such a new art in the early 1900s.

Above all, this project has taught me that with interactive visualizations of several dimensions, a user may seek to learn one thing, but when the data is visualized, he can infer far more information than he originally sought.

FUTURE WORK

There were many additions to my current executable that had to be put aside for the sake of time. I intend to continue my work on the FilmFinder in several ways:

- Implementing AND filters (e.g., movies with Meryl Streep AND George Clooney)
- Having different tab views allowing the user to plot data against other metrics beside film rating. For example, running time, setting location, rating (PG-13, R, etc.).
- Color coding each new data point, to differentiate between each actor selected.
- When two data points overlap, mouseover will allow you to see the information for both data points (as two popups).
- Including a screenshot from the movie with the data popups.
- Having a film search functionality, where all the data from one film is displayed, as well as a "films like this" tool that lists films with similar metadata.

As mentioned before, the FilmFinder 2.0 is merely one direction this visualization could have taken. It would be interesting if the work could be extended into an academic tool, used to compare trends in filmmaking over time. Perhaps the data could be supplemented with information like mood, film techniques, or colors (i.e. a visual of the screen colors as a movie progresses). This would allow film scholars to understand trends in filmmaking that may not have been known before.

ACKNOWLEDGMENTS

Thanks to the CS448b teaching staff for an awesome quarter!

REFERENCES

1. Ahlberg, C. & Shneiderman, B. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. *Proceeding: Chi '94 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1994), 313-317.
2. JSONLint, https://github.com/circlecell/jsonlintdotcom.