Steve Letourneau

Peter Cheung

Jeff Hofer

Alex Neilson

Feb. 1, 2017

CMPT 350

Project Proposal

Our proposed group project is to create a data visualization tool that will allow a user to find a movie that they want to watch. The tool will be a web app which will help solve the issue of users trying to determine which movie to watch.

Domain Situation:

It can be difficult deciding which movie to watch based on the multitude of choices. Some user stories for an app that would address this issue are:

* As a movie watcher, I want to discover movies that I find interesting.
* As a user, I want to find movies with similar subject matter that I am interested in.
* As a movie watcher, I want relevant suggestions of movies so I can find new movies to watch.
* As a movie watcher, I want to easily and quickly find movies to watch so that I can save time looking for movies.

The app will solve these problems and create many benefits for users. The app will help users:

* Find a movie or movies that the they will enjoy.
* Save users time by reducing the amount of time searching for movies.
* Broaden users’ knowledge in movies by introducing them to films that they may not have known or heard about.
* Potentially expand or reinforce a user’s interest in a genre, actor or director.

The vis tool will utilize a dataset of movie data which includes data such as: title, directory, year, duration, actors, Facebook likes, gross revenue, genre, plot keywords, budget, and IMDB score. The dataset is in a tabular format and contains approximately 5000 movies. The data source url is: <https://data.world/data-society/imdb-5000-movie-datase>. In addition, we will use the IMDB open API called OMDB: <https://www.omdbapi.com> . This has additional features such as Movie posters, Oscar awards and plot synopsis.

Task and Data Abstraction:

The task that the user is doing is searching the data. Once a target is found, they can query the data to see additional attributes. For example, the movies will be filtered based on similar movies to the movie the user selects. The app will also allow the user to query the movies selected by clicking over a movie and see additional data from the OMDB database such as a plot synopsis, Oscar wins and nominations and extra cast and crew information.

The data abstraction plan is to abstract data from the domain of all data into a more usable form. We will group the data by relating similar attributes between table values to create sets.

Visual Encoding/Integration Idiom:

The two main vis formats we are considering are:

1. Heat Map- all the data will be displayed in a Heat Map graph. The entire dataset will be shown based on the IMDB score. The user can either select a movie from the Heat Map or use a Combo box to find a movie. The Heat Map would then update based on the web application sorting algorithm which would find related movies.
2. Force Directed Graph – this vis idiom shows in a network format. The size of the nodes would increase or decrease based on user inputs. The linkages between nodes (movies) would be based on common traits.

Algorithm:

The algorithm uses the attributes of a film such as actors, director, genre, keywords to find other films with similar attributes. When two films have similar attributes, the score of that attribute is added to a total called the “relevancy score”. The films with the highest relevancy scores are deemed to be the most like the film selected by the user. Broader attributes, such as language, have lower weights and therefore have less effect on the relevancy score. Conversely, precise attributes like keywords yield higher scores and add more greatly to the relevancy score.