Overview

The goal of this project was to come up with an system that could potentially be used as a recommendation scheme for specific users based on their known ratings for specific movies and for all users in general based on inter-movies correlations.

We started out with a data set of 1682 movies and 943 users and exactly 100,000 ratings on the movies in the set from the users in the data set. This meant that we only had 6.3 % of a the maximal data set that we could have had in the hypothetical situation where every user had rated every single movie (1,586,126 ratings in total). We needed to fill in the gaps by coming up with 2 latent factor matrixes that summarized the 1682 by 943 data matrix using the 100,000 data points we had.

Yamei generated the latent factors matrixes using python, while Jan and Obi worked on the plots and visualization in python and MATLAB respectively.

Data Manipulation

The data sets we were given were:

- A matrix of movies and genre tags
- A table of ratings from users, on specific movies

What we needed was a matrix that represented all possible ratings from users to movies (with the actual ratings from the table when applicable, and a zero when that user hadn't rated a specific movie).

We generated this by iterating through the rows of the given ratings table, and extracting the value in the ratings column and storing it in user_movie_matrix(i, j), where i and j correspond to the values in the user id column and the movie id column from the same row.

Learning Algorithm Selection

In order to get a factorization that was similar in comparison to the provided ratings, we implemented both Stochastic Gradient Descent (SGD) and Alternating Least Squares to generate the latent factors. Using ALS resulted in negative values when we tried to recover the equivalent of the user_movie_matrix from the dot product of u and v. This is because of the inverse operation of the sparse user_movie_matrix. This prompted us to finally adopt SVD for as our algorithm for this exercise.

In order to get reasonable result and make the iteration efficient, we update our algorithm several times by making use of matrix manipulation.

We first constructed the user_movie_matrix using the technique mentioned earlier.

For each iteration, we update every row in the u matrix and every column in the v matrix step by step using gradient descent. In our algorithm, we set 1000 iterations, learning rate is 0.0002 and regularization lambda_ is 0.02. In our algorithm, it takes 6-7 seconds for each iteration.

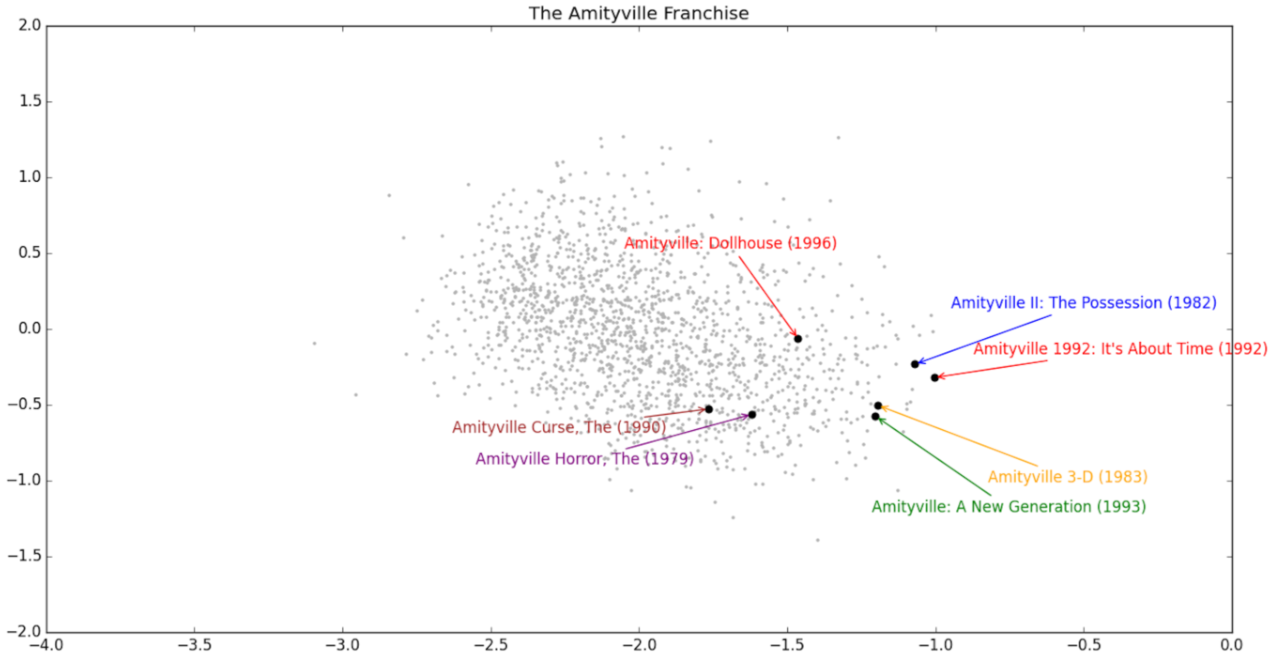
Now, we get u matrix with 943*20 dimensions and v matrix with 20*1682 dimensions. Then, we apply singular value decomposition for u and v, project u and v to 2 features representations.

u_2dim is 2*943 for users, v_2dim is 2*1682 for movies. We finally use these matrixes to visualize the data.

Visualization Interpretation & Conclusion

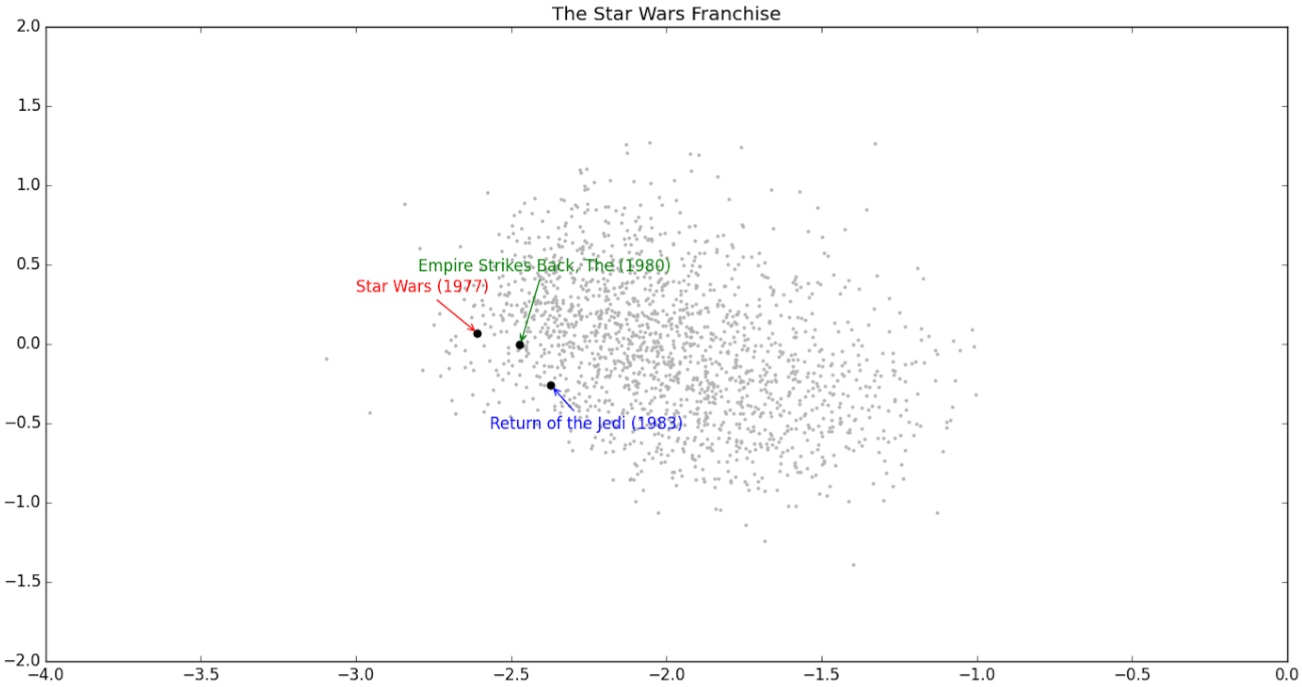
After generating the 2 dimensional projections of the latent factors U and V, we were able to use them to make scatter plots of movies and users. The users scatter plot has little meaning on its own. But when you take the dot product of a specific user and a movie, you get a sense of how highly they would rate the movie. What we are more interested in however is studying the clustering of movies.

First, we analyze movies that form a sequel. Take the Amityville sequel for instance:



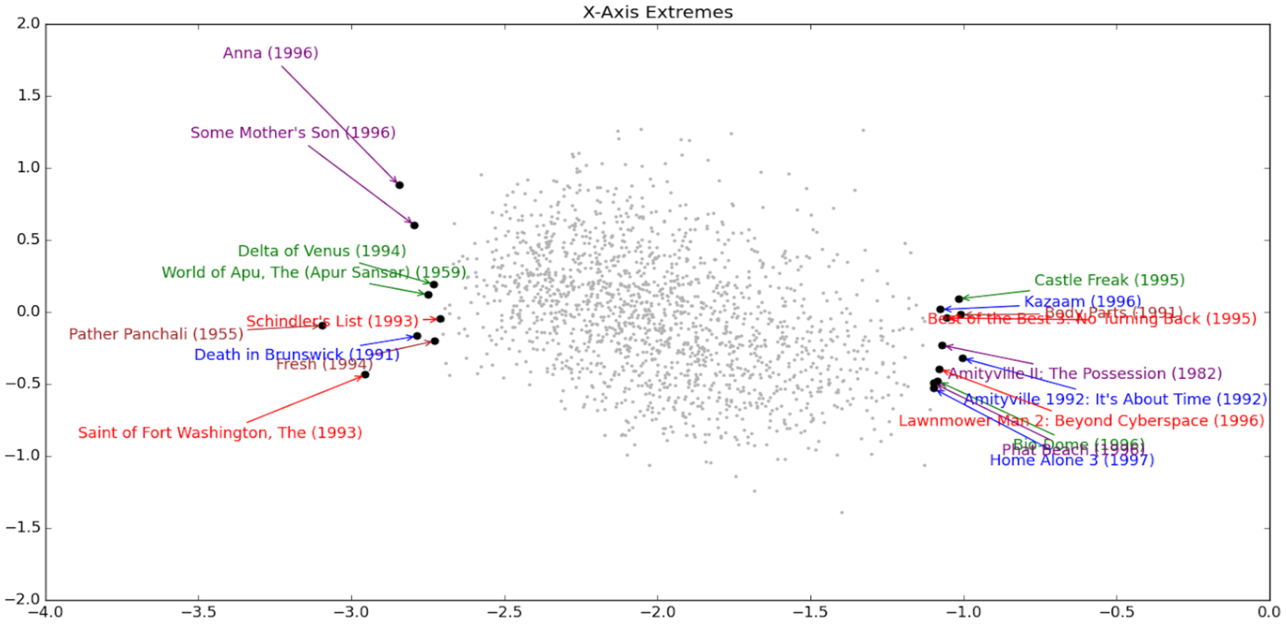
The blue points represent the entire set of movies and where they lie in the 2 dimensional projection of the v (dimension, movie_id) latent factor. You can see that 4 out of the 6 points that are the Amityville sequel fall within a radius of less than 0.1 for a cluster consisting of just these 4 points. When taking all 6 points into consideration, a cluster radius of less than 0.4 arises. This is representative of very strong similarity between 4 of the movies and fair amount of similarity between the other 2. The reduction in similarity could be attributed to over approximation in the final v-matrix, or due to qualitative properties of the movie that changed across the sequel such as a change in cast members, or a deviation from the usual theme of the sequel.

The same applies to the Star Wars sequel:



When we analyze the movies by more qualitative properties, we notice even stronger clustering.

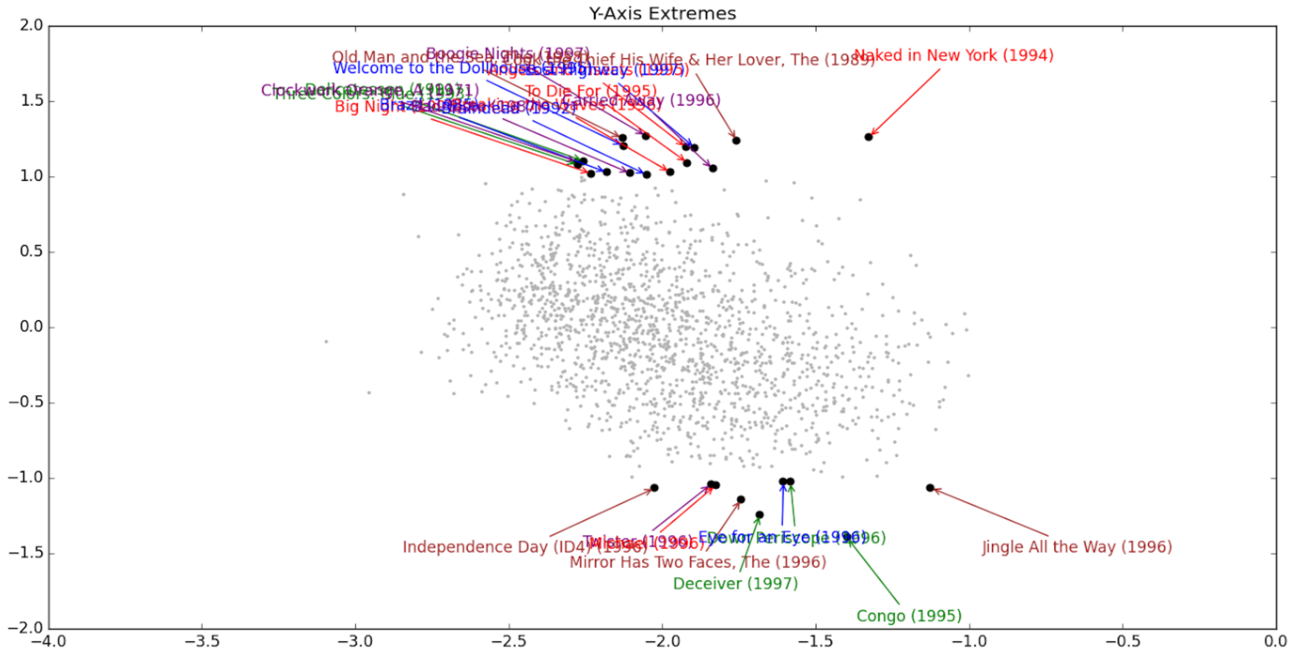
Observe the divergence between movies with intellectual plots and movies with more superficial story-lines:



The x-axis seems to be an indicator of the plot's depth. Films on the left are more intellectual and and contain deeper messages, while films on the right contain more cheap thrills and pandering to a certain mainstream demographic, whether for horror, action, or kids.

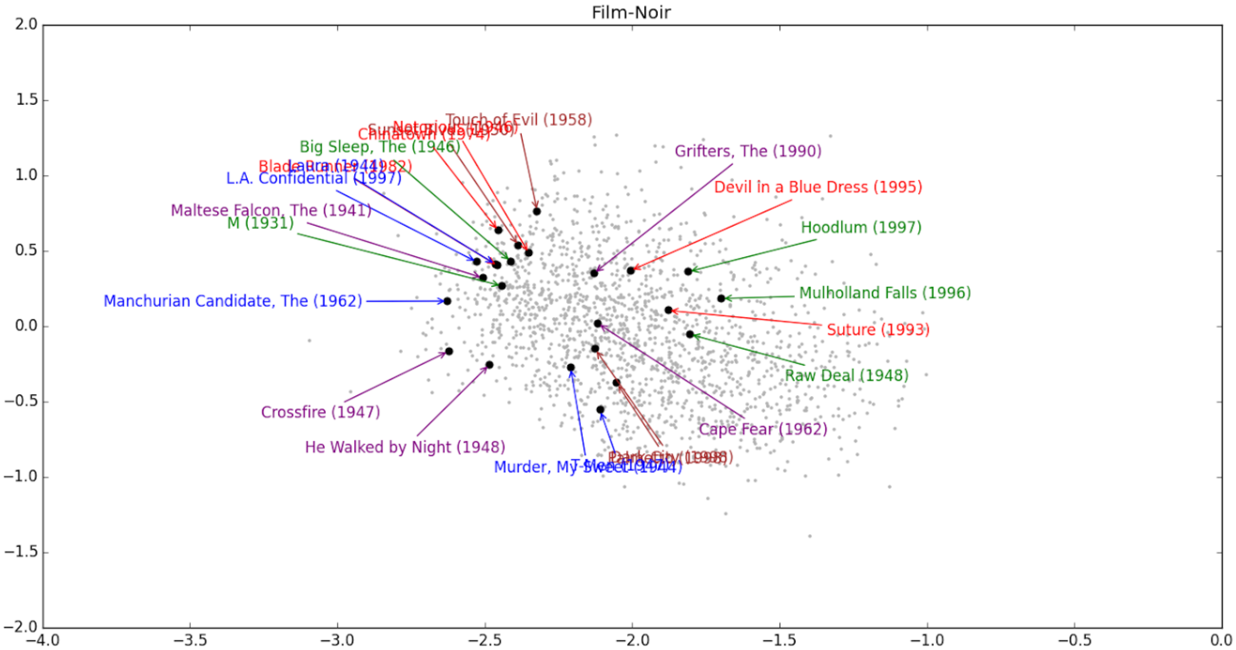
There are two important things that can be observed here. One is the close clustering amongst intellectual movies as well as the close clustering between superficial movies. The other important observation is the high dispersion between both groups. This aligns perfectly with what with expect to see from a movie recommendation system like Netflix's recommendation engine.

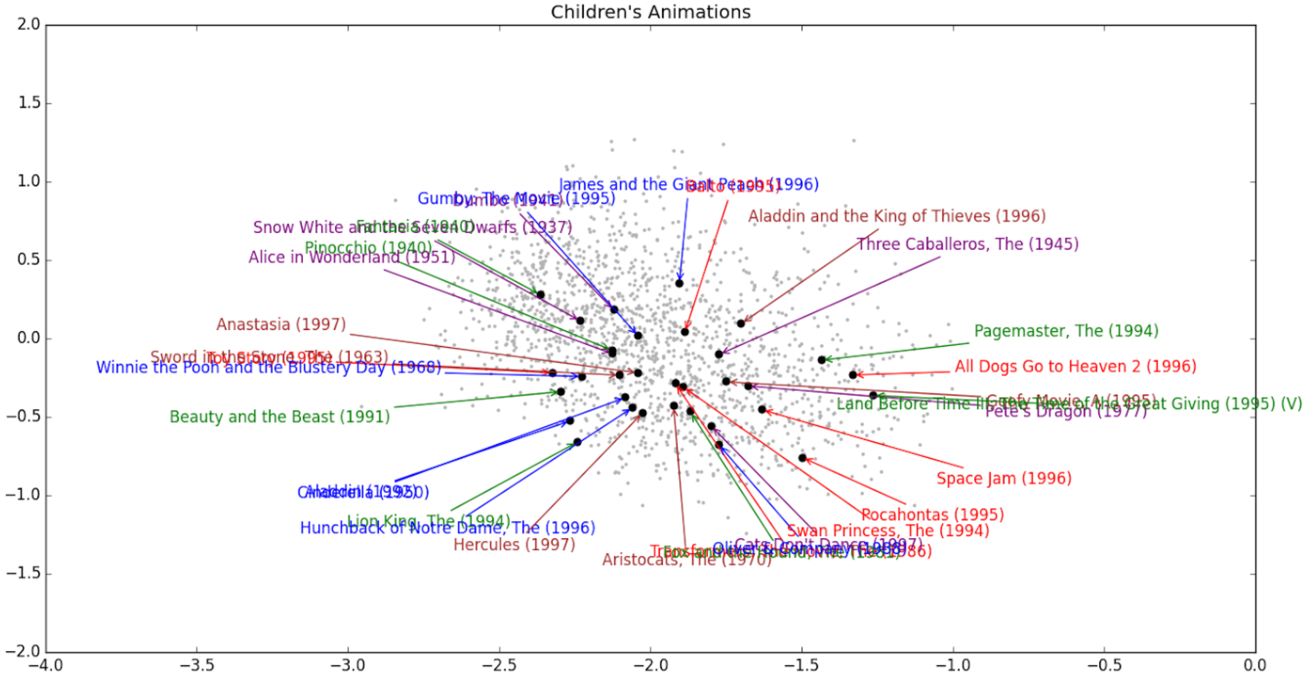
Observe the same properties when we compare action-packed movies with movies that appeal dominantly to women:



The y-axis seems to be an indicator of whether a film generally appeals to women or men
In some cases, looking at one or two genres at a time does not show as much clustering as we expected.

For instance, film noir movies are fairly spread apart, and so are children's animations:


Film-Noir
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−4.0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
Touch of Evil (1958)
Sunset Blvd. (1950)
Notorious (1946)
Chinatown (1974)
Big Sleep, The (1946)
Blade Runner (1982)
Laura (1944)
L.A. Confidential (1997)
Maltese Falcon, The (1941)
M (1931)
Grifters, The (1990)
Devil in a Blue Dress (1995)
Hoodlum (1997)
Mulholland Falls (1996)
Manchurian Candidate, The (1962)
Suture (1993)
Raw Deal (1948)
Crossfire (1947)
Cape Fear (1962)
He Walked by Night (1948)
Murder, My Sweet (1944)
T-Men (1947)
Dark City (1998)
Palmetto (1998)


Children's Animations
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−4.0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
James and the Giant Peach (1996)
Balto (1995)
Gumby: The Movie (1995)
Dumbo (1941)
Aladdin and the King of Thieves (1996)
Snow White and the Seven Dwarfs (1937)
Fantasia (1940)
Three Caballeros, The (1945)
Pinocchio (1940)
Alice in Wonderland (1951)
Anastasia (1997)
Pagemaster, The (1994)
Sword in the Stone, The (1963)
Toy Story (1995)
Winnie the Pooh and the Blustery Day (1968)
All Dogs Go to Heaven 2 (1996)
Goofy Movie, A (1995)
Land Before Time III: The Time of the Great Giving (1995) (V)
Pete's Dragon (1977)
Beauty and the Beast (1991)
Space Jam (1996)
Aladdin (1992)
Cinderella (1950)
Pocahontas (1995)
Lion King, The (1994)
Swan Princess, The (1994)
Hunchback of Notre Dame, The (1996)
Hercules (1997)
Cats Don't Dance (1997)
Transformers: The Movie, The (1986)
Fox and the Hound, The (1981)
Oliver & Company (1988)
Aristocats, The (1970)

In summary, the clustering generated by the 2D projections doesn't depart from expected human behaviour in terms of what movies have a similar rating pattern and what other movies an individual is likely to enjoy if he rated a neighbouring movie highly. This implies that our factorization is indeed a good approximation of the hypothetical matrix where everyone rates all 1682 movies.