

1 Introduction [10 points]

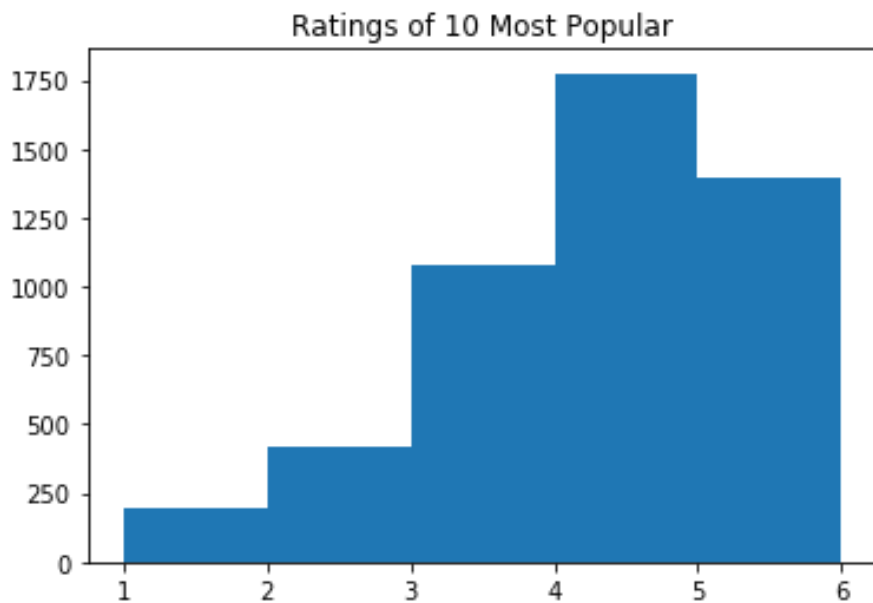
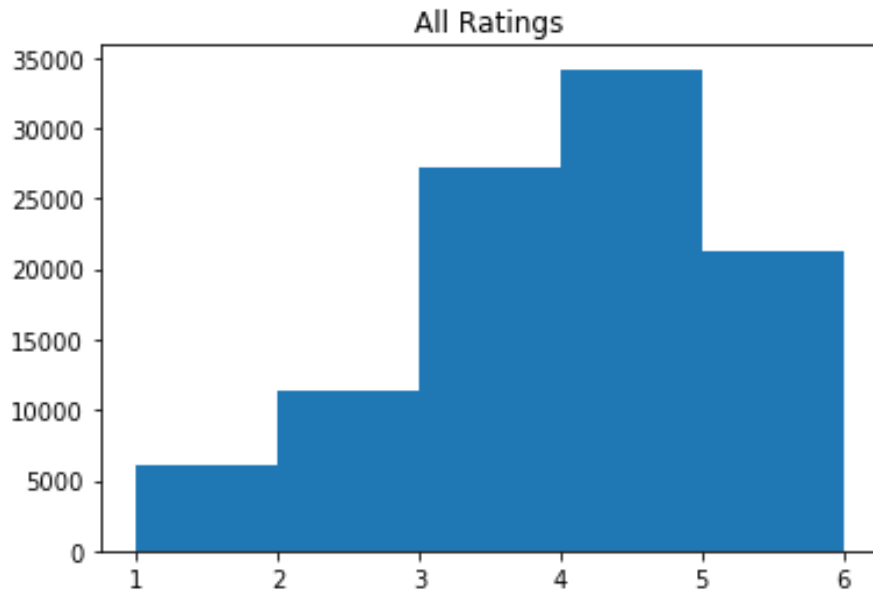
Group members: Frank Kou, James Wei

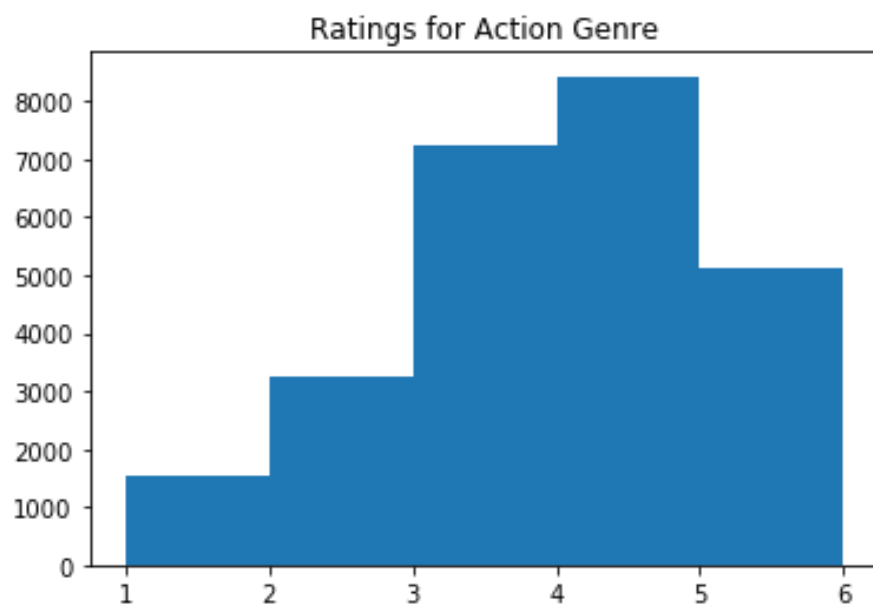
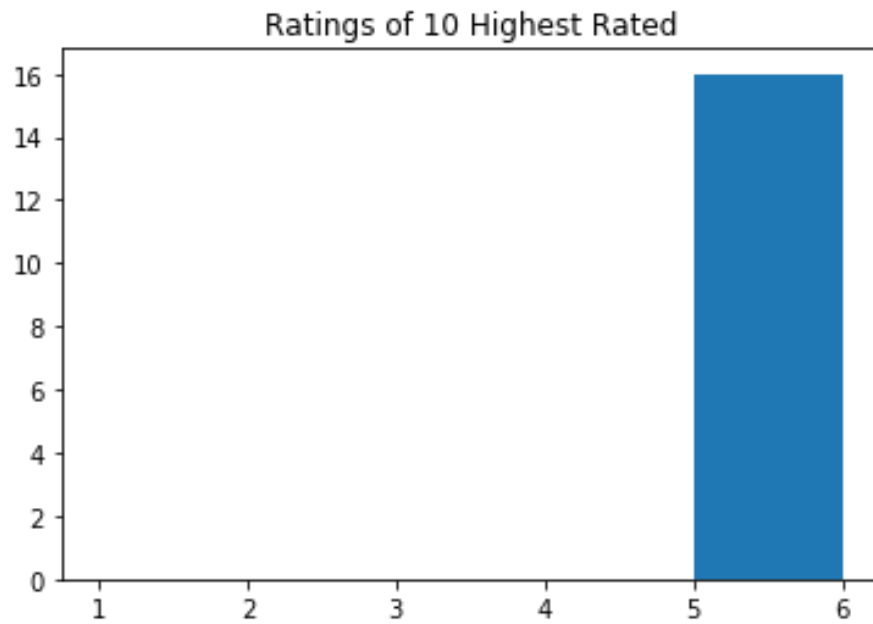
Group name: 343

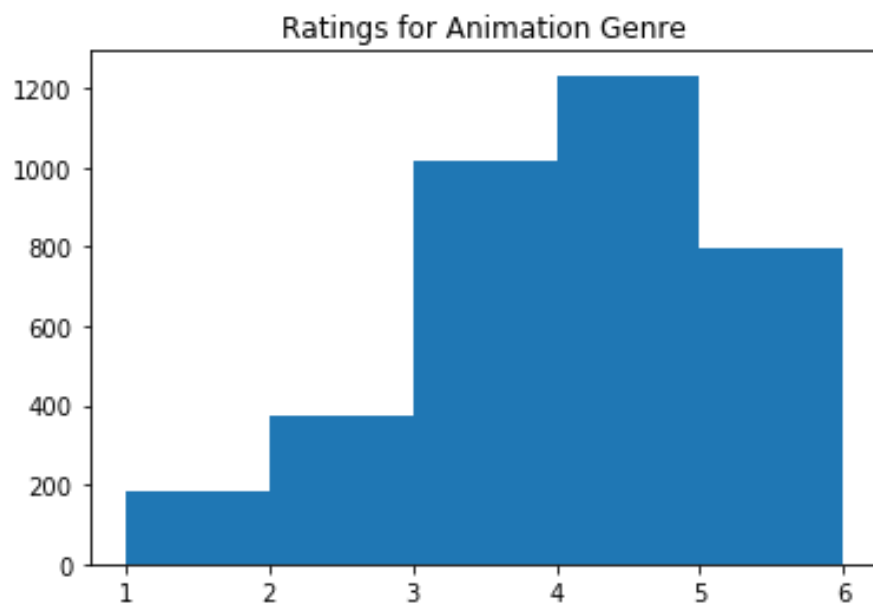
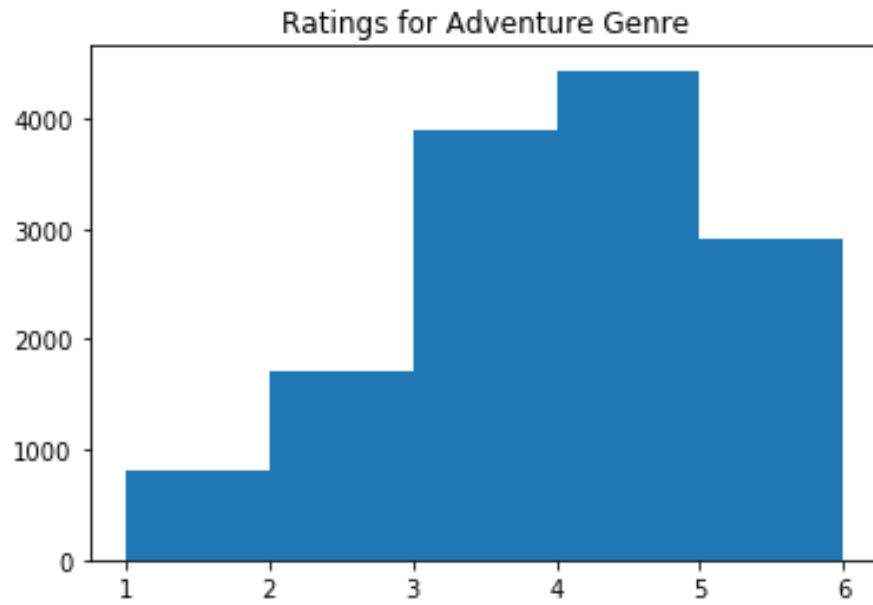
Division of labor: James did basic visualization. Frank implemented first and second method of matrix factorization as well as their corresponding visualizations. James did third method of matrix factorization as well as its corresponding visualization. James and Frank both thought of ideas for 'interesting' visualizations for blog post. James wrote blog post. Frank and James wrote their corresponding sections for report.

2 Basic Visualizations [20 points]

Plots





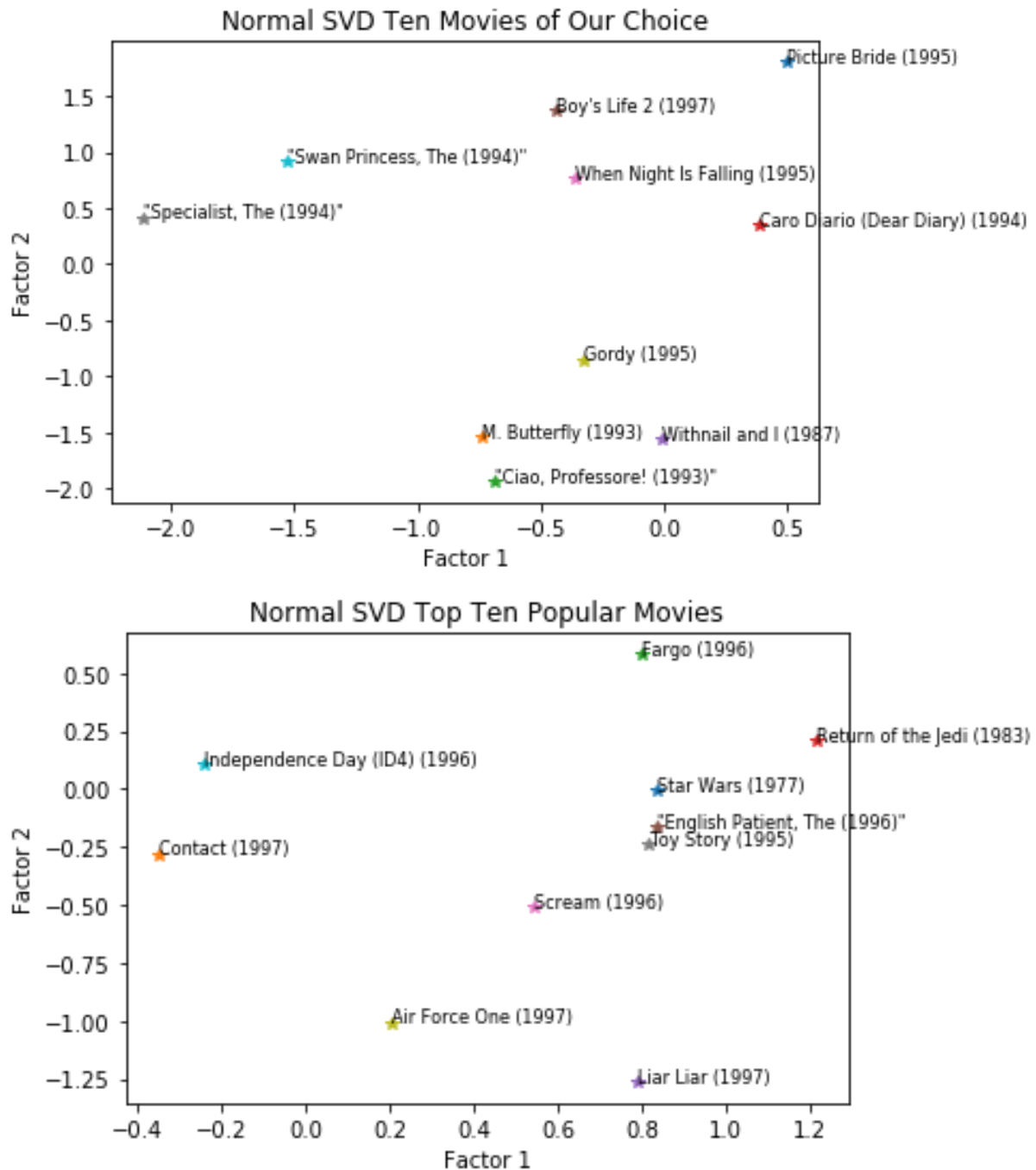


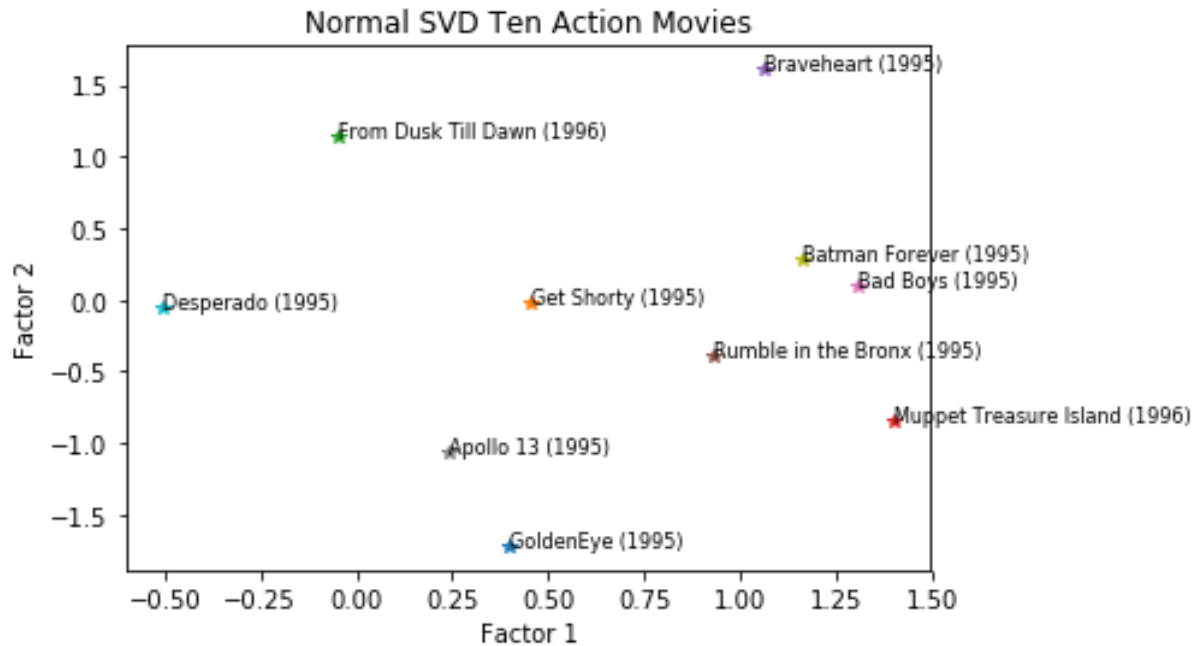
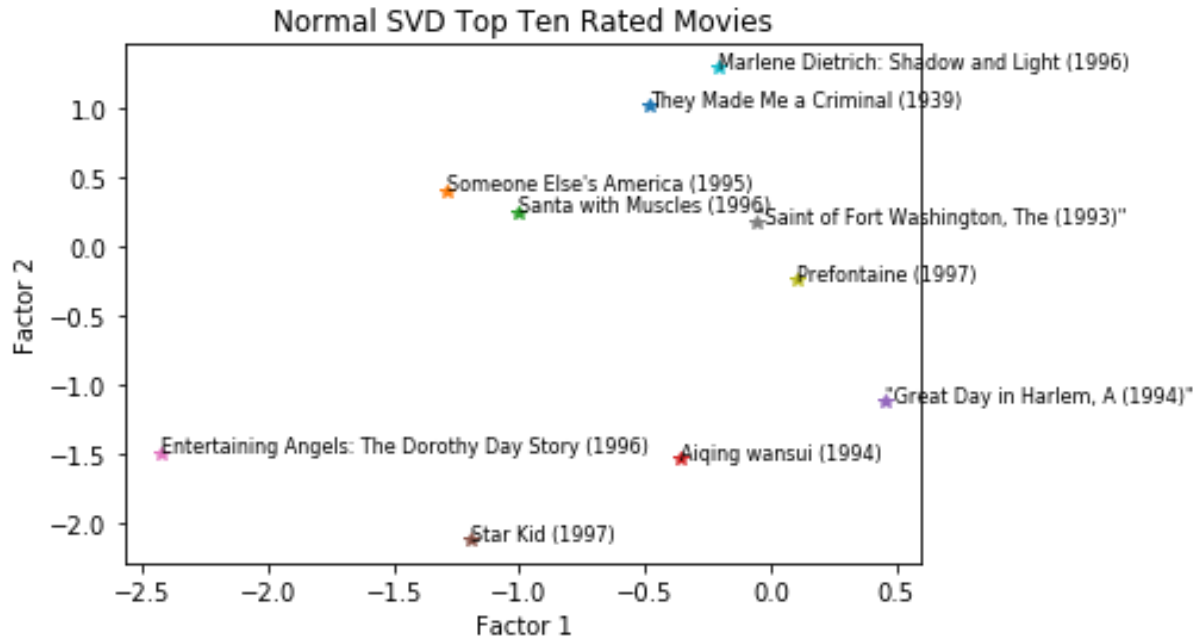
Analysis

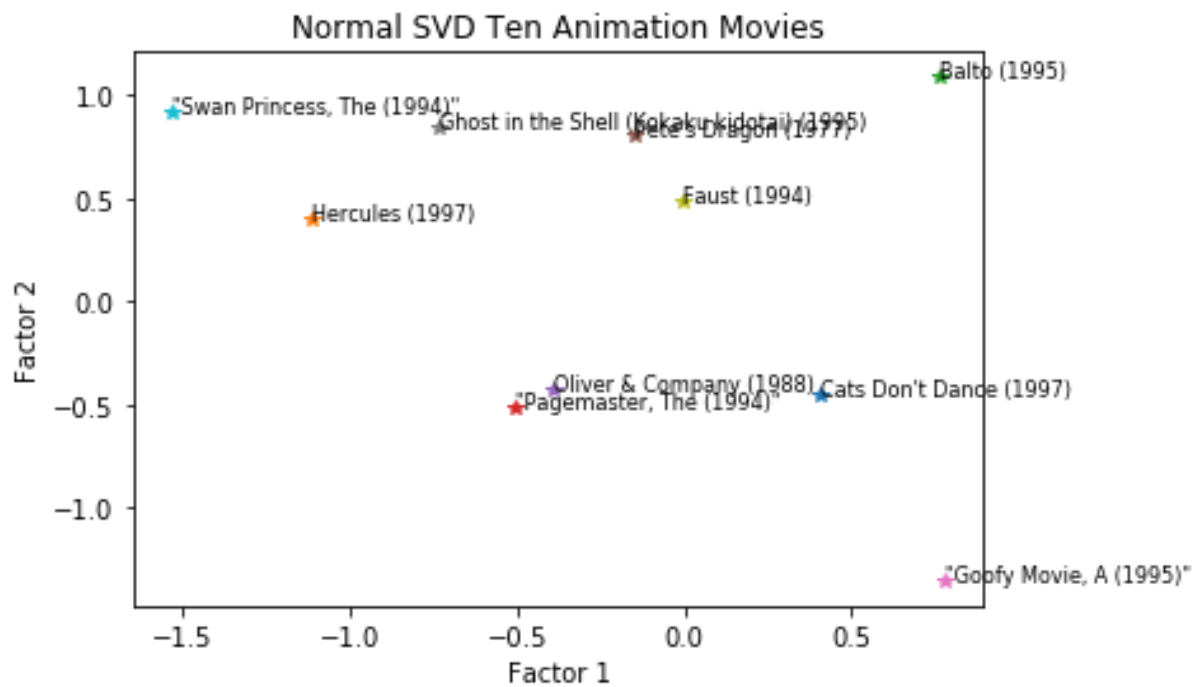
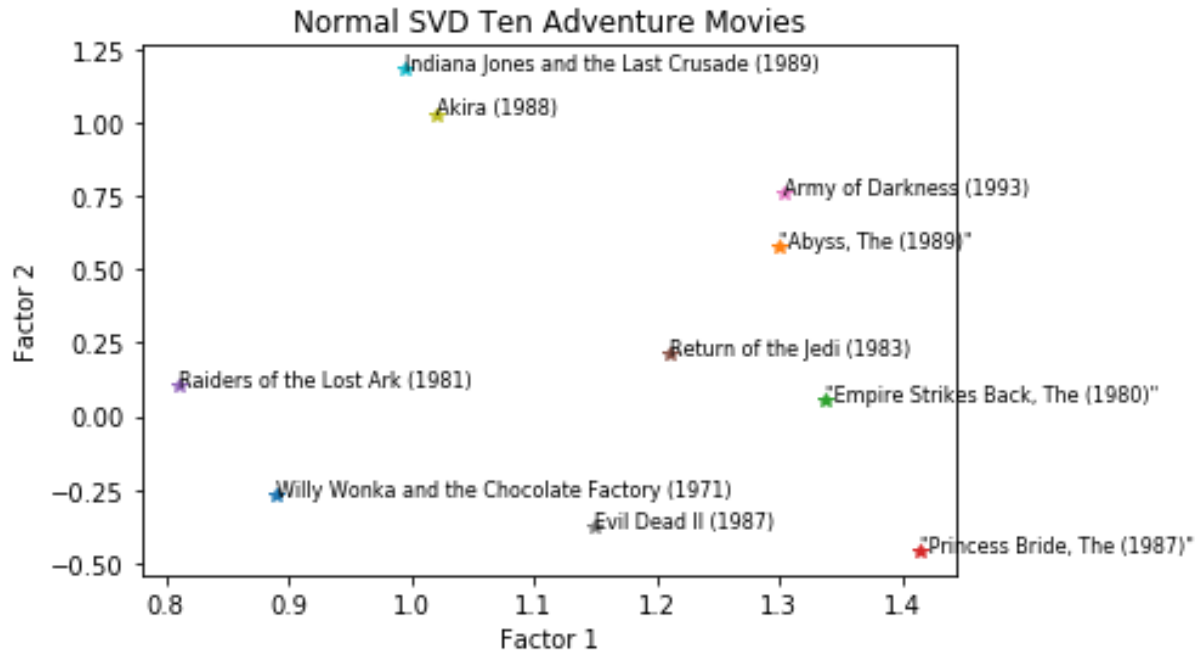
In general, we see that across most of the charts (except "Ratings of 10 Highest Rated"), 4 was the highest frequency rating. The general trend for number of ratings is that 4 got the highest, followed by 3, then 5, 2, 1. The one exception is the 10 most popular movies which has number of ratings as 4, 5, 3, 2, 1. This may be attributed to a bias that the movie is popular and thus it must be good, which explains why in this case, there are more 5 ratings than 3 ratings. Another trend we noticed was that the graphs are mostly left skewed. They all have very similar shapes. For the most part, the results are what we expected to see. We expected the average and mode to be either 3 or 4 because we believe that people generally give more optimistic ratings. Their scales are set higher. That is, an average movie would get somewhere between a 3 and 4, instead of being a 3. The shapes are also what we expected to see because we felt that people generally dislike giving 1 or 2 ratings. After all, it is a movie, which is entertainment. The movie must be extremely horrible or the critic must be very harsh for 1 ratings to be given. However, we did not expect that the ratings of the 10 highest rated movies to be all 5. We thought there would be a majority of 4 and 5 ratings but not all 5. Upon closer inspection, we saw that with most of these movies, they were only rated by either 1 or 2 people. Therefore, if a movie only got a single rating and it was a 5, then it would have a 5 rating overall and thus have the highest average rating possible. Seeing this circumstance makes sense. We see that the ratings of the most popular movies are different from the ratings of the 10 best movies. First of all, the sample size of the best movies are too small. They are outliers, movies that only got 1 or 2 ratings. We see that the most popular movies follow the rating trend of all the rest of the graphs while the ratings for the best movies were all 5. We see that the ratings for the three genres we chose, action, adventure, and animation, were all very similar. They had similar plot shapes as each other as well as the plot of all the ratings.

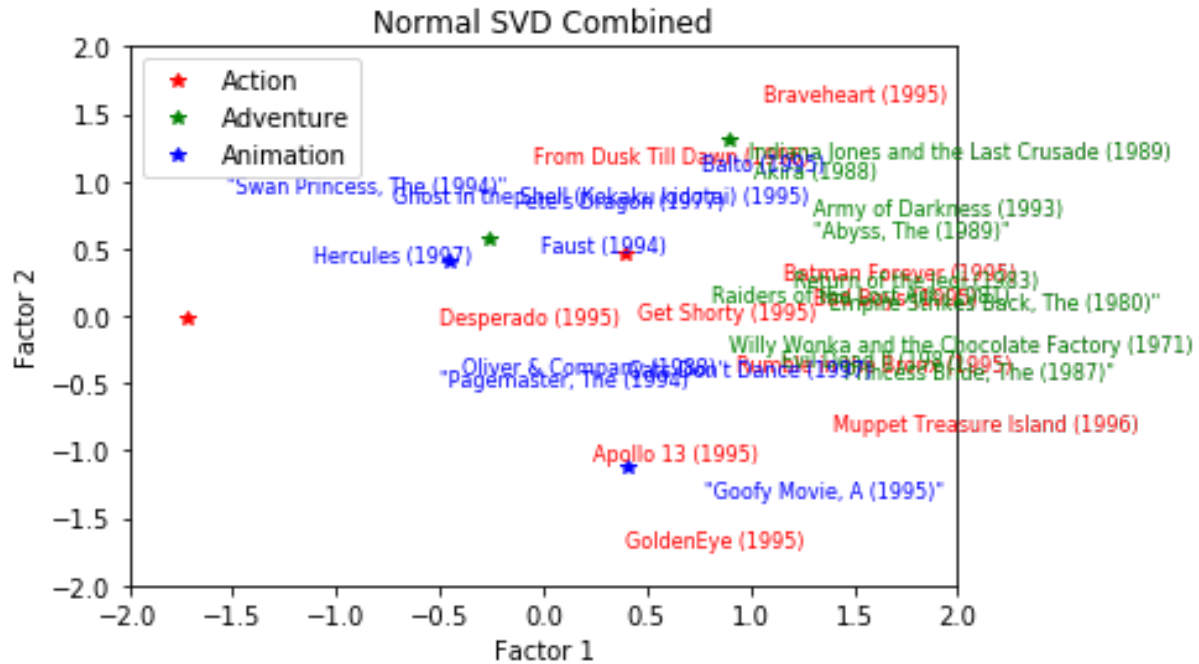
3 Matrix Factorization Methods [40 points]

Normal SVD Plots

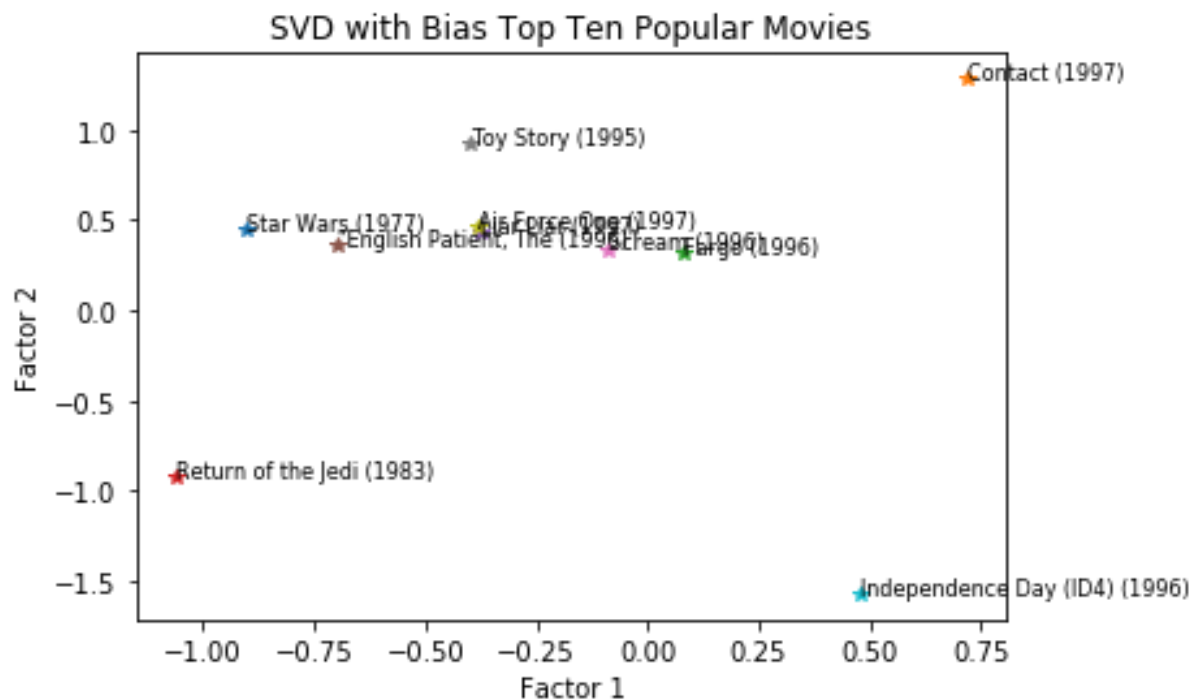
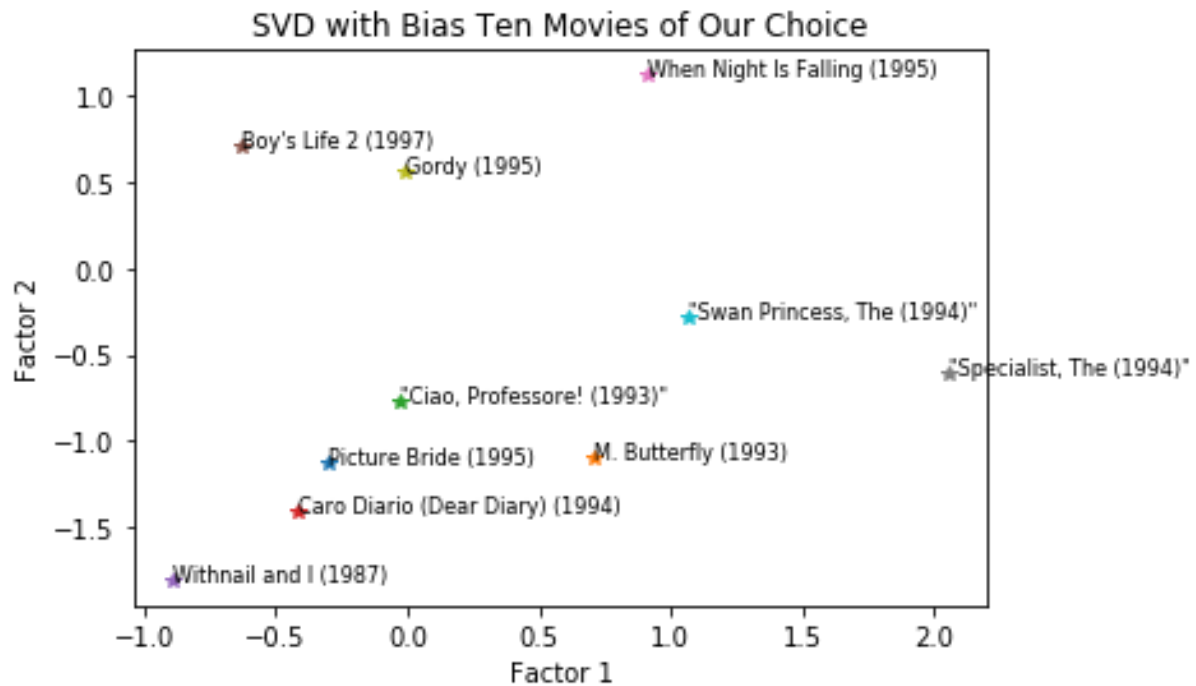


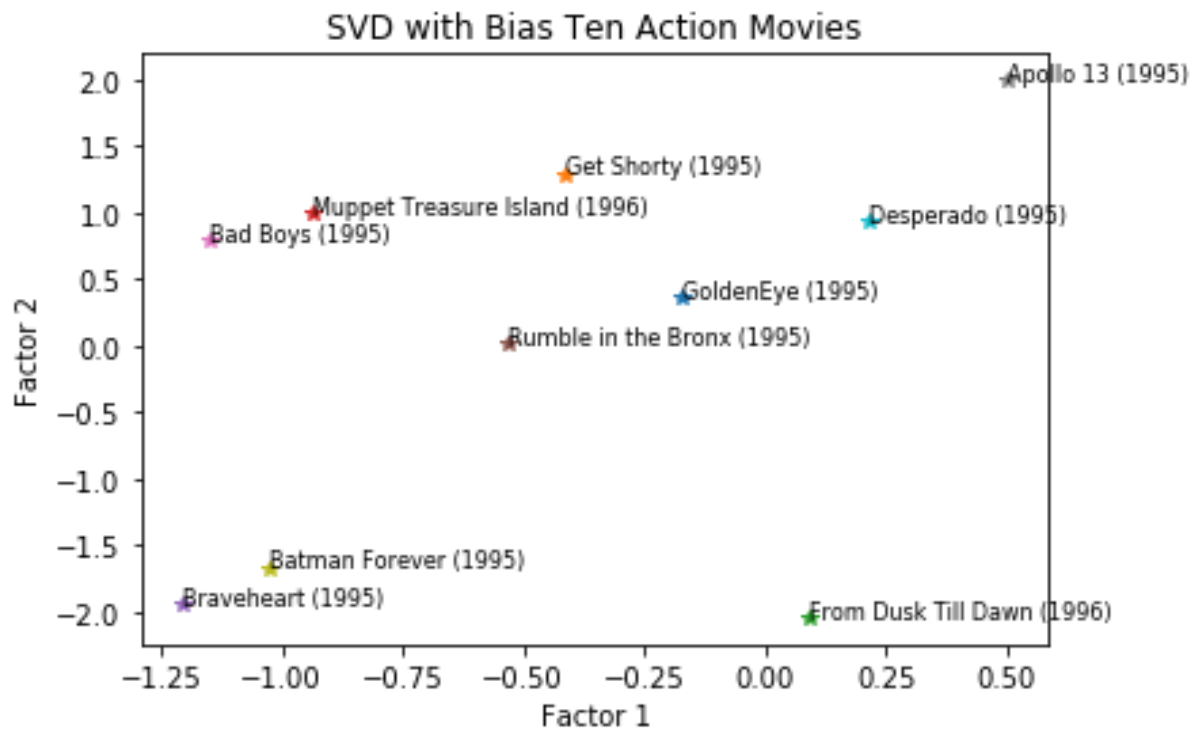
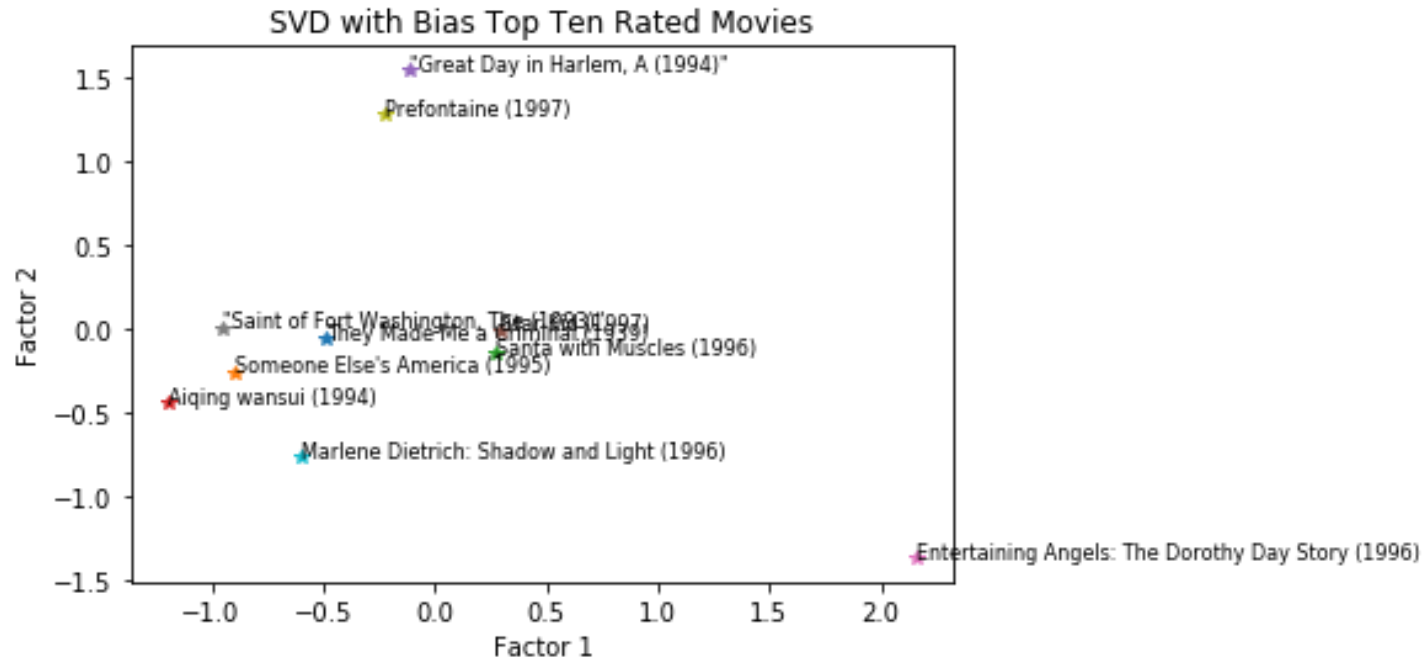


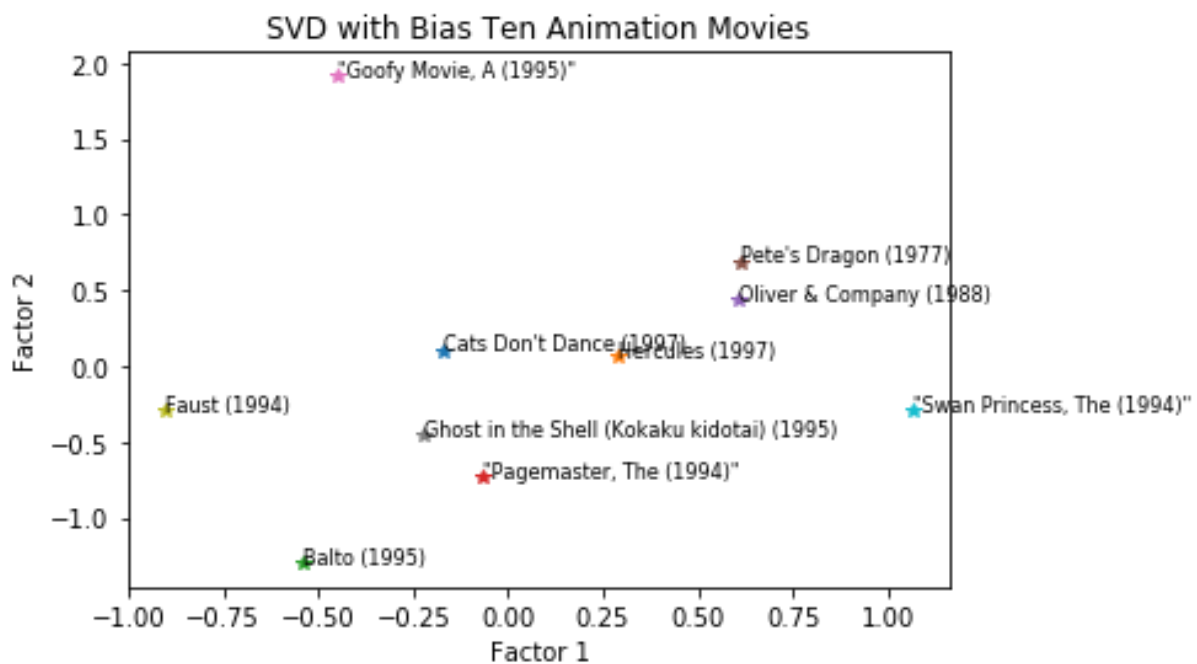
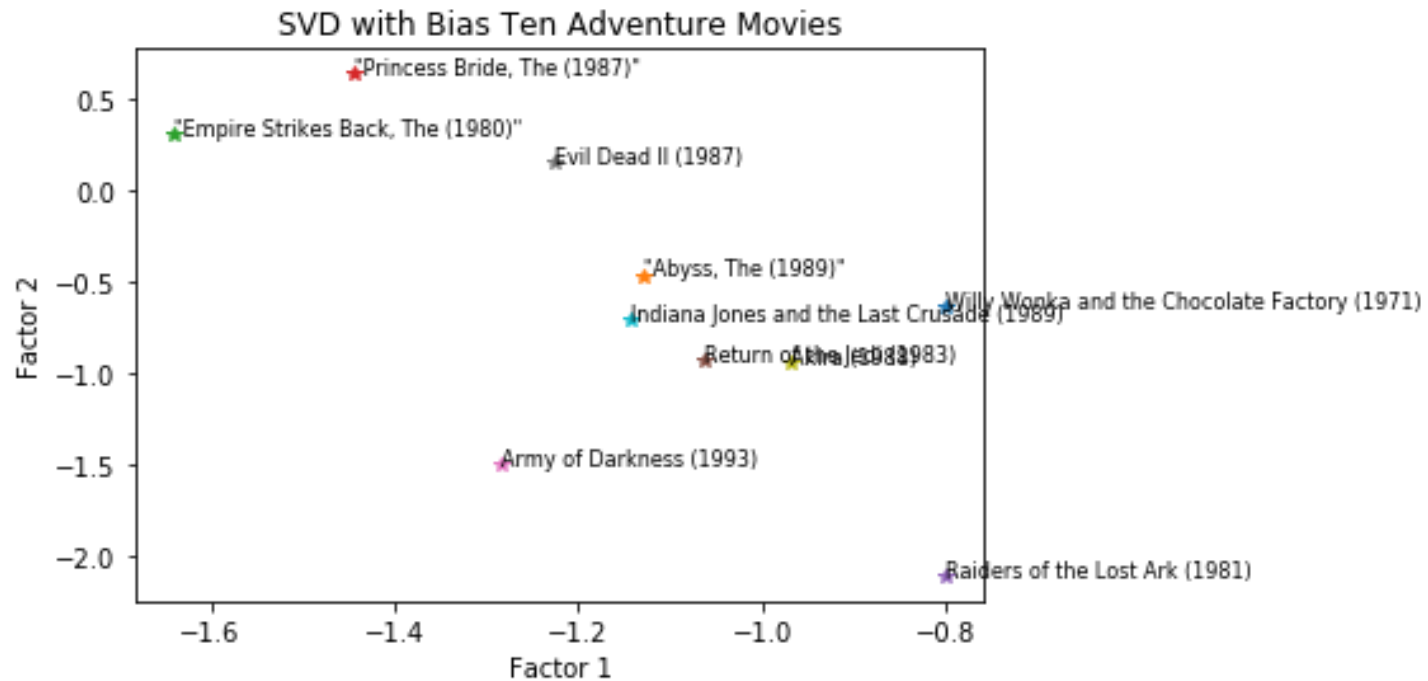


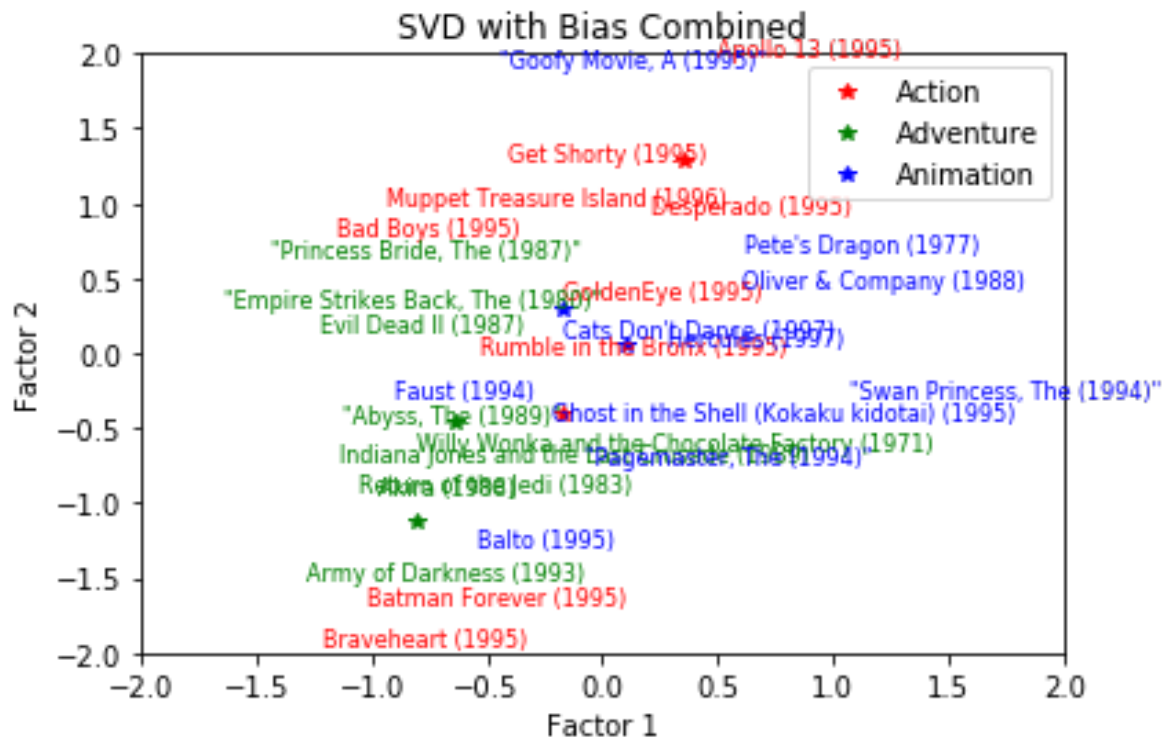


SVD with Bias Plots









Off-the-shelf Implementation Plots

TODO post your visualizations

Description

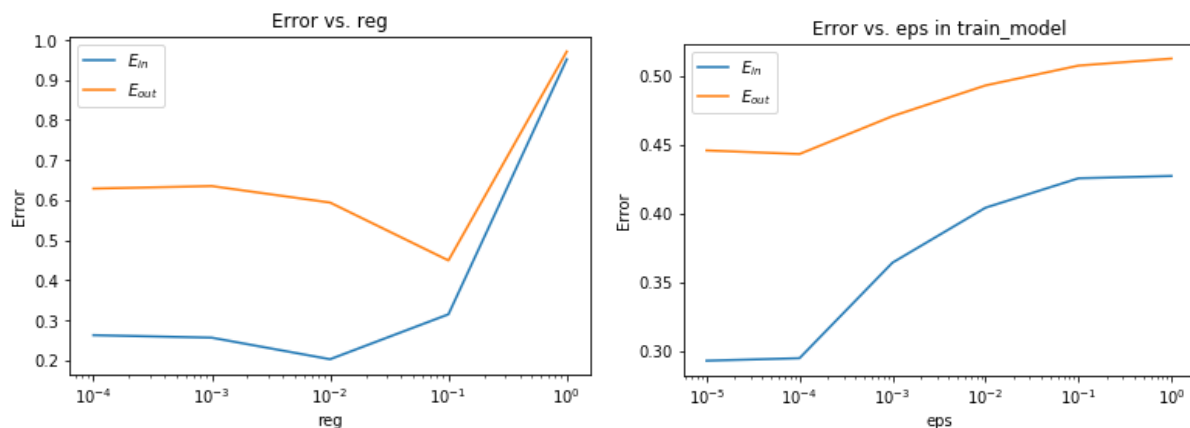
Normal SVD - This is the SVD that we used from Homework 5. This is a matrix decomposition method used for reducing a matrix to its constituent parts in order to predict a full matrix that is missing many values. In our case, we have a $M \times N$ user by movie matrix (943 users \times 1682 movies). We split this into latent factors matrix U^T and V with size $M \times k$ and $k \times N$ respectively, where $k = 20$.

SVD with bias - This is like the normal SVD but we incorporate bias.

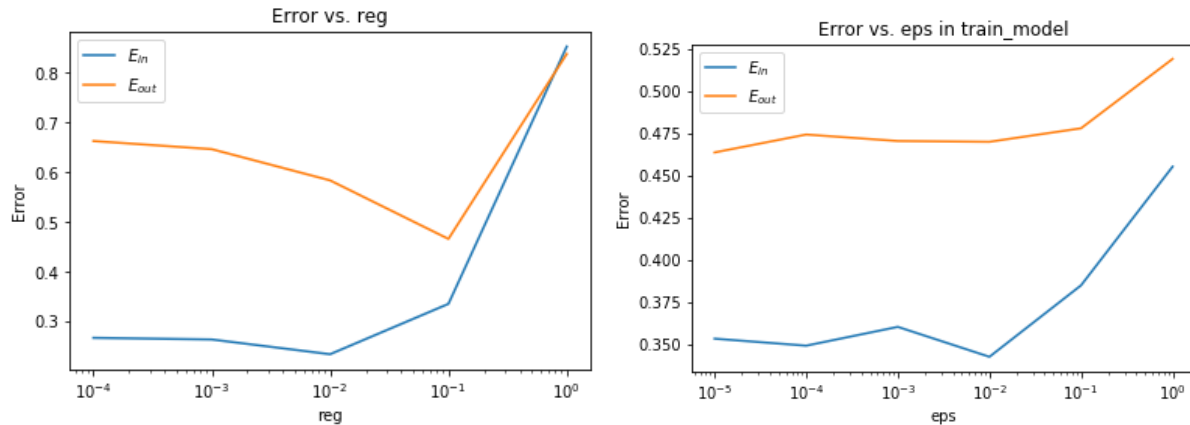
Off-the-shelf - ***** TODO *****

Methods

Normal SVD works by splitting an $M \times N$ matrix and tries to learn a latent factor model U and V . U^T and V have size $M \times k$ and $k \times N$ respectively. This is done by using gradient decent and stopping when we have met our stopping condition. We use Frobenius Norm. When we have successfully found our U and V , we can multiply the matrices together to get the full matrix and predict what rating each user gives to each movie. In our project, we were given that k has to equal 20. To find the other parameters, we created test cases and plots to determine our parameters. We found out that using an regularizer constant of 0.1 and epsilon of 0.0001 minimized our test error. See below.



SVD bias works the same way as normal SVD, except this time we incorporate a bias that we find using gradient decent. This is so we can model the global tendency of a movie's average rating and model the tendency of how a user rates on average. This keeps U and V more focused on variability between users and movies. This is especially important because different users, for example, have different opinions on how ratings are given. One user may be very optimistic and consistently give only 4 and 5s. Another user may be a harsh critics who isn't afraid to give 1s. Similarly, we found that regularizer constant of 0.1 and epsilon of 0.00001 were the best. See below.



Off-the-shelf implementation works by

TODO

Differences

The first two methods differ in that the first one does not have a bias while the second method does incorporate a bias. The first two methods and the off-the-shelf implementation differ by

TODO

Performance

On the test case, the normal SVD got a test error of 0.511003967792. The SVD with bias got a test error of 0.472148622408. The off-the-shelf implementation got a test error of

TODO

Test error explanation

We see that overall, the test errors were pretty high. This is due to the fact that we start of with a 943 users x 1682 movies matrix. This gives us a total of 1,586,126 values in the matrix. In the training dataset, we were only provided with 90,000 values, which means learning U and V to predict the whole matrix was a hefty task. It makes sense that our test errors were pretty high. The SVD with bias did perform better than SVD with no bias by about 4 percent. This makes sense because there is a tendency for how a user rates a movie and how a movie is rated. For example, one user may be very optimistic and give relatively high ratings for all of his/her ratings. Incorporating a bias can help us model this because then it makes sure that U and V are focused on variability between users and movies. That is, in our example, it models how that optimistic user varies between how he/she judges movies. The off-the-shelf implementation differences explain the

test differences by

TODO

4 Matrix Factorization Visualizations [30 points]

General trends

TODO

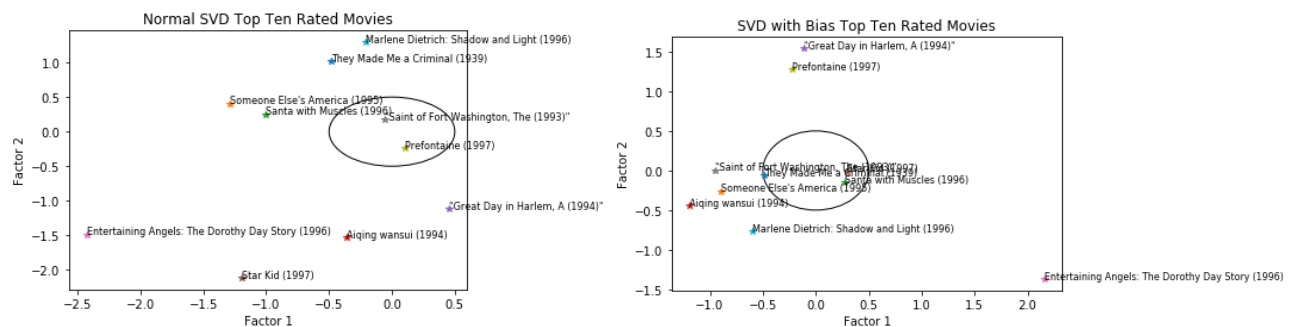
Expectation

The visuals did match what we expected to see. We expected to see some minor trends but no major trends because the test accuracies were high.

Popular vs best movies visualization

We can see that in the popular movie plots, many of the movies are clustered around the origin (0, 0). These movies seem to have latent factor 1 and 2 with values close to 0. For the highest rated movies, in the plots, we can see that these movies have a more widespread distribution with varying factor 1 and 2 values.

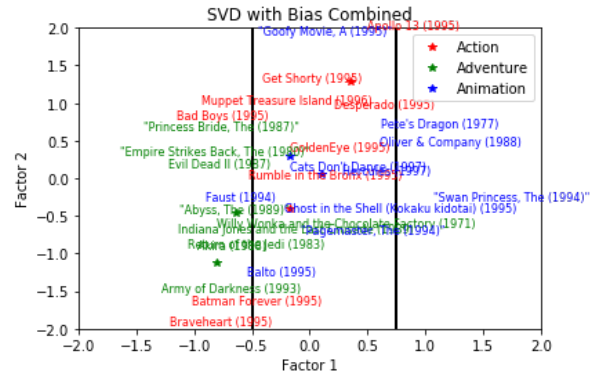
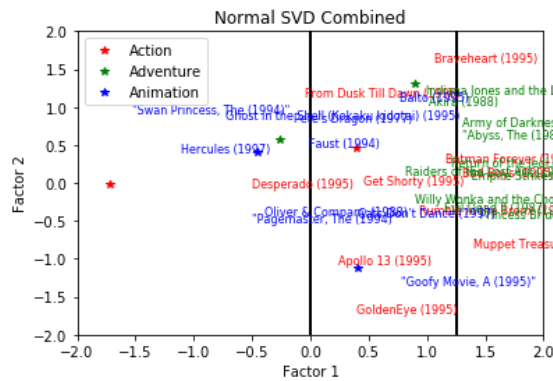
TODO



Three genres visualization

The three genres we chose were action, adventure, and animation. Looking at the plots of the combined genres, we see that in the visualization of normal SVD, the three genres were varying across the Factor 2 axis. For Factor 1, it seems like animation movies had a small factor 1 than action movies, which have a smaller factor 1 than adventure movies. Looking at the visualization for SVD with bias, we see that all three genres similarly are varying across the factor 2 axis, but in this case, adventure movies had a smaller factor 1 than action movies, which have a smaller factor 1 than animation movies. In both cases, the three genres are generally separated vertically (we can use vertical lines to approximately separate the three genres).

TODO



Visualization differences between methods

TODO
