NOTE: We noticed some silly mistakes in some of our figures and they have been corrected. For that reason figures may vary slightly from what we submitted on Friday.

Miniproject 2: MovieLens Visualization

Introduction

Group Members:

Eli Pinkus
Akshay Vegesna

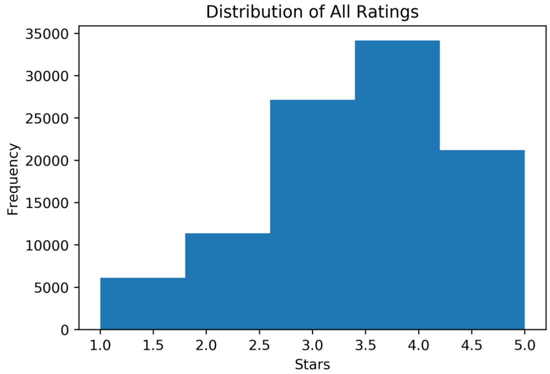
Group Name

Yasser's Disciples

Basic Visualizations

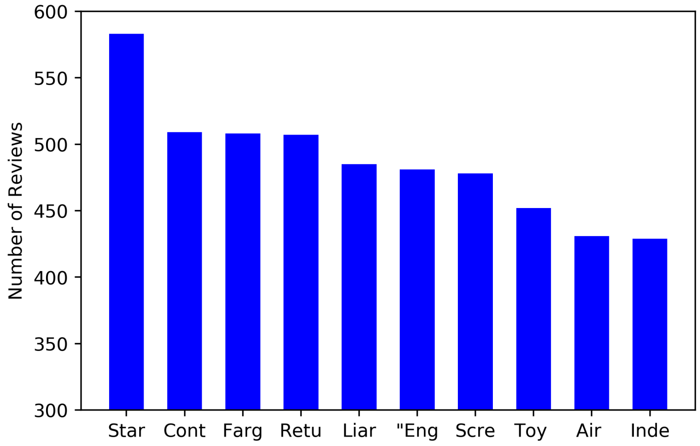
We began by looking to obtain an insightful visualization for the entire set of reviews in the training data. We opted to plot a histogram of the reviews in the training set according to their star ratings. The result is shown in figure 1. We hoped that this visualization might reveal ratings biases within the data set and give us an idea of most common star ratings. We observed that the most common rating is 4 stars. Additionally, 3, 4 and 5 star ratings as a bin represent a disproportionately high number of ratings relative to if the reviews were distributed evenly. This could be a result of a few effects. It is possible that users gravitate, either naturally or by other recommender systems, to movies that they are more inclined to enjoy, thus resulting is a positive bias in reviews overall.

FIGURE 1



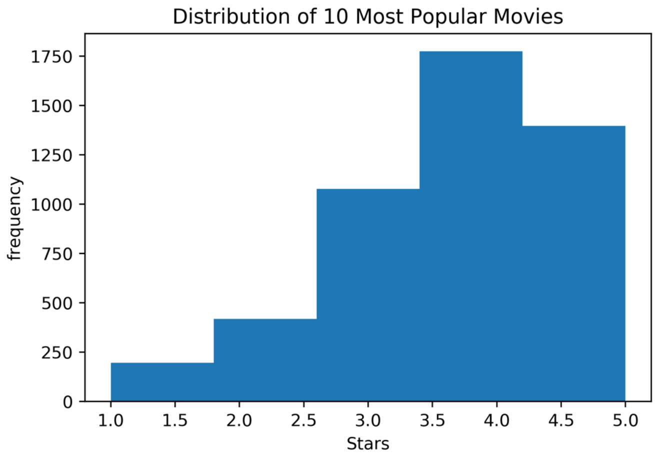
Next, we looked to visualize the 10 most frequently rated movies. We created a bar graph of the quantity of reviews for each of the top 10 movies. The results are shown in figure 2. The results show us that Star Wars was far and away the most frequently rated movie in the group with nearly 600 ratings in the training set. The next most frequently rated movies have barely over 500 reviews and the 10th most frequently rated movie, Independence Day, had fewer than 450 reviews. We quite simply attribute this to Star Wars being among the most popular movies of all time by most metrics. We were also curious how the distribution of ratings varied between the top 10 most frequently rated movies and the dataset as a whole. We plotted a histogram and the results are shown in figure 3. Surprisingly, the distribution of most popular movies match that of the entire data set pretty well. The most popular movies ratings had relatively fewer 5 star ratings and slightly more 1,2, and 3 ratings relative to the overall set distribution. We think it possible that this is the case because people may be more inclined to indulge in a film generally outside of their preferences if it is a big-time blockbuster. Thus, the distribution of ratings may be skewed downward.

FIGURE 2



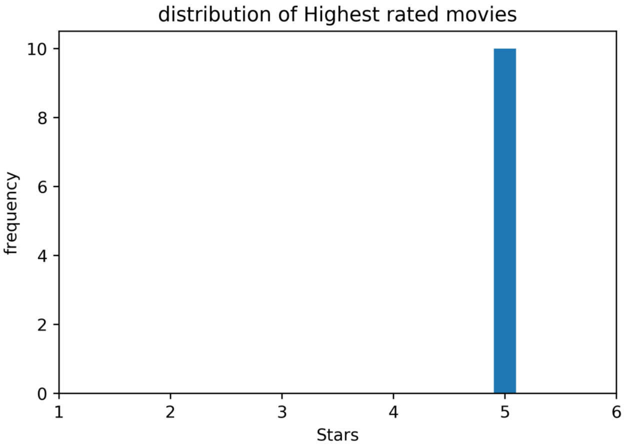
Key:
Star Wars (1977)
Contact (1997)
Fargo (1996)
Return of the Jedi (1983)
Liar Liar (1997)
English Patient, The (1996)
Scream (1996)
Toy Story (1995)
Air Force One (1997)
Independence Day (ID4) (1996)

FIGURE 3



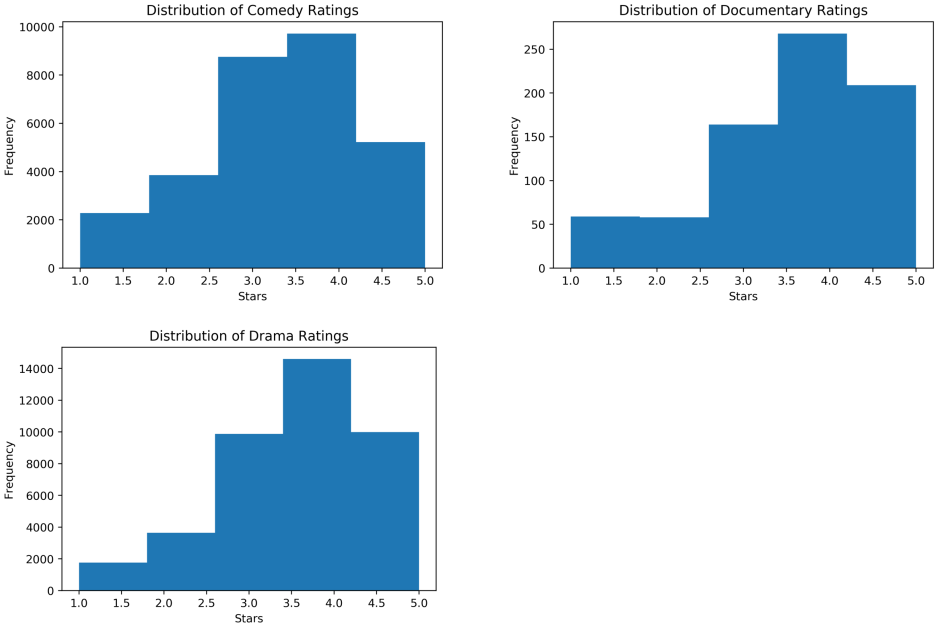
Next, we looked to visualize the top 5 highest average rating movies in the dataset. Inspection revealed that each such film had an average rating of 5 stars with fewer than 4 ratings each. For this reason we don't find this 10 movie subset to be all that interesting as it seems to reveal little about all-time favorite movies due to the lack of representative data. We plotted a histogram of the average reviews which in this case is perfectly analogues to overall reviews of this subset because all reviews are 5 stars. The results are in figure 4.

FIGURE 4



We proceed to focus on three genres for our final visualization. We chose comedy, drama, and documentary. We plotted a histogram of the ratings for movies in each genre. The results are in figure 5. We expected comedy ratings to be skewed more toward higher ratings considering the light hearted, "happy" nature of the genre. We also expected documentaries to be more polarizing since the content is more likely to be opinion charged or political which could cause certain users to rate. We didn't have strong presuppositions about the drama genre. We were surprised that each genre was fairly close the distribution of the entire dataset. Documentary surprisingly had more 1 star ratings than 2 star ratings but it also had by far the smallest sample size so the distribution is more volatile and less meaningful. Documentaries also had a slightly larger proportion of 5 star ratings which we believe can be attributed to a selection bias in the topic of documentary that people chose to view.

Figure 5



Matrix Factorization Methods

Set 5 Implementation

For the matrix factorization for set 5 implementation, we directly ran our set 5 implementation on the ‘data.txt’ set given. The only slight complication was that the output matrices U and V were in the dimensions M by K and N by K but this was easily fixed by taking the transpose. This matrix factorization method worked by the following training objective. Note that the norms are the norms of Frobenius. Specifically we minimized these training objectives by doing stochastic gradient descent on both U and V. The loss of this matrix factorization method on the test set was approximately 1.12 after training for 100 epochs.

$$\underset{U,V}{\operatorname{argmin}} \frac{\lambda}{2}\left(\|U\|^2 + \|V\|^2\right) + \sum_{(i,j)\in S}\left(Y_{i,j} - u_i^T v_j\right)^2$$

Set 5 Implementation with bias terms

For the matrix factorization for set 5 with the bias terms a and b, we modified our set 5 implementation by including a gradient descent for the a and b terms. The training objective of this method of matrix factorization is the following:

$$\underset{U,V,a,b}{\operatorname{argmin}} \frac{\lambda}{2}\left(\|U\|^2 + \|V\|^2\right) + \sum_{(i,j)\in S}\left(Y_{i,j} - \left(u_i^T v_j + a_i + b_j\right)\right)^2$$

In order to modify our stochastic gradient descent implementation that was used in our set 5 implementation, we needed to find the gradients of the bias terms a and b, which were found to be the following:

$$\delta_{a_i} = -\sum_{j=1}^{N} Y_{ij} - (u_i^T v_j + a_i + b_j)$$

$$\delta_{b_j} = -\sum_{i=1}^{N} Y_{ij} - (u_i^T v_j + a_i + b_j)$$

After we derived this gradient formula, it was a simple matter to work it in to the set 5 implementation and obtain the two output matrices U and V (and bias terms a and b). Although we did get matrices of the for M by K and N by K for U and V respectively, we again fixed this problem by just taking the transpose. The loss of this matrix factorization method on the test set was approximately 1.11 after training for 100 epochs.

Off-the-shelf Implementation with bias terms

For our off the shelf implementation, we used the simple sk-learn package sklearn.decomposition.NMF. This package didn't end up giving great performance when we checked the error on the test set, so we stuck to our set 5 implementations code.

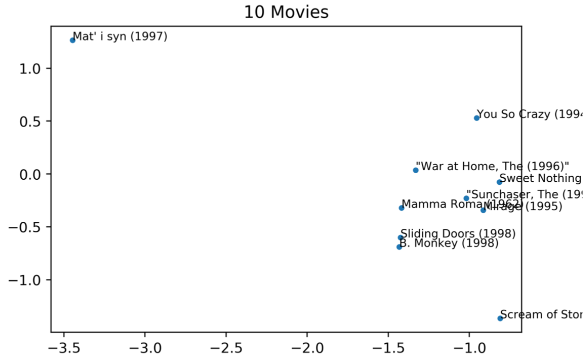
We also ran into a problem because the package sklearn.decompositon.NMF required a matrix representation of data. We were initially using a representation of the form Y_{ij} which had points indexed by i's and j's so missing values were not an issue. We fixed this problem by setting these empty entries in the full matrix to be the mean of the column of the matrix. It was then simple to use the sklearn package to obtain the two U and V matrices. The loss for the matrix factorization on the test set was approximately 2.22.

Visualization of Matrix Factorization projected in 2 Dimensions

The plots going forward came from simple matrix factorization in 2d (our set 5 implementation). The reasons for this choice will become clear after the discussion in the conclusion.

10 Random Movies

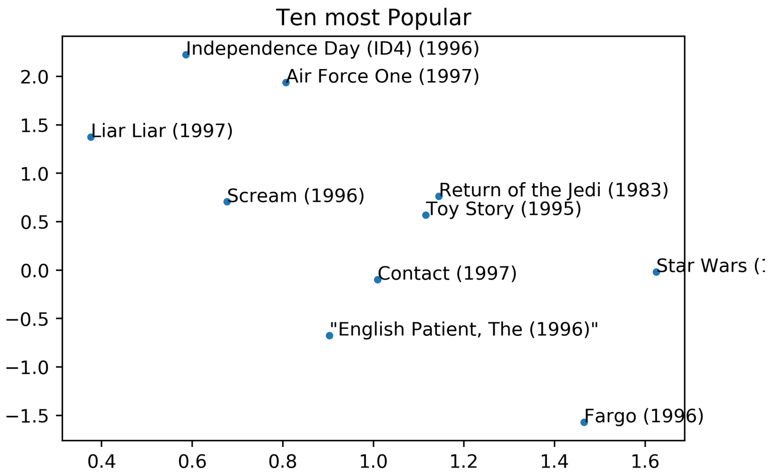
We were curious to see the result of our 2D visualization for 10 random consecutive films from the dataset. The results can be seen below:



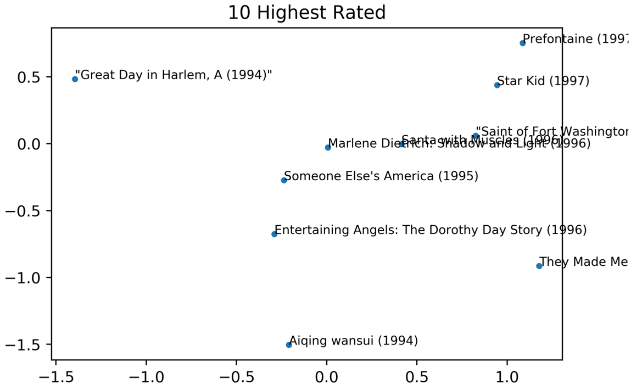
The movies have a fairly strong clustering apart from 'Mat' I syn (1997). Although we were not explicitly given any information about the ordering of films in the data set, we were hoping that this visualization might reveal a trend. In this case we see strong clustering and think it likely that there may be some untold structure to the ordering of the films as given in movies.txt.

It is important to note that the 2D visualization figures have variable x and y axis scales. In theory this could reduce the significance of apparent clustering so when we say there is strong clustering we implicitly mean that the clustering is strong relative to axis we saw in other figures with various clustering behaviours.

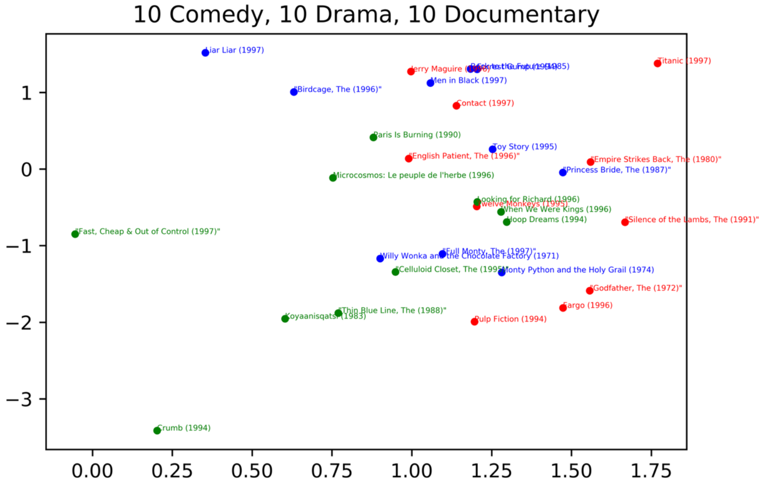
10 Most Popular movies



We see an interesting trend in the 2D projections for the 10 most popular movies in that the movies seem somewhat aligned with the downward diagonal of our view of the space. If we consider each axis to represent some attribute of the film, although we can't really speak intelligently about what those attributes are, it is interesting to note that this group of popular films seems to trend on the diagonal of the attributes implying that movies with low values of both attributes or high values of both attributes (relative to our viewing window) are generally not the most popular movies.



This image contains the 10 highest rated movies. At first glance, most of these titles seem unrecognizable and this is because these movies were just rated 5 stars by one person. Although we considered choosing a threshold for the number of reviews to qualify consideration for the highest rated movies, we eventually settled on simply following the exact given instructions which yielded the above graph. No real conclusions could be made from this graph because each one of the ratings contained data from only one user.



This image shows the highest rated movies from each genre of our choice. Note that the blue dots represent comedy movies, the green dots represent documentaries and the red dots represent dramas. A general trend was that even on the projected 2D space, points that were close together generally corresponded to movies that held some similarities. This is fairly evident in the figure given.

We test an extreme case of these by considering a documentary which is highly removed from the rest of the data as seen in the figure--Fast, Cheap and Out of Control. Fast, Cheap and Out of Control is a movie where the director invents a new camera technique which allows the interview subject to face the interviewer and the audience simultaneously. Clearly, we see here that the outlier in the latent space is also likely an outlier in the actual space.

Another more evident feature of the figure is that there are general trends in genres. For example, the green dots (documentaries) tend to be in the bottom corner, the red dots (comedies) tend to be on the right side, and the blue dots (dramas) tend to be at the top. In general, this seems to point to the fact that comedy movies tended to have a high value on the second principle axis. Similarly, documentaries seem to have a low value and dramas have a high value on the first principle axis.

Conclusion

Our plots in the latent space came from the U and V learned with our set 5 implementation. This was because after a grid search for the optimal regularization parameter for simple factorization and simple factorization with bias, we found that the optimal values had comparable performance on the test set. Our off-the-shelf implementation gave generally poor error on the test set for comparison.

We therefore arbitrarily chose the simple factorization method to generate our visualizations on, though we did explore some simple visualizations on the other factorization methods to get some intuition for the other latent spaces. In particular, the latent space for the simple matrix factorization with bias terms strongly resembled the latent space in our simple matrix factorization without bias terms. Further, the off-the-shelf learned V tended to have high clustering in the latent space and trends in data were much harder to identify. We generally attributed this to this to the fact that taking the mean of the rows and columns for the missing values led to some noise that was hard to avoid.

In general, we were happy with our results. We found a 2-dimensional space that represented our 1682 movies and 943 users well, and we were able to identify trends in the latent space that were consistent with our intuition about the movies themselves. Given more time, we would probably spend more time experimenting with off-the-shelf implementations and find a way to avoid the missing-value problem.