Visualization of Latent Factors from Movies

Hongjian Lan
hlan@caltech.edu

Xiang Ni
xni@caltech.edu

Taokun Zheng
tzheng@caltech.edu

March 13, 2015

1 Overview

The goal is to visualize and interpret a 2-dimensional latent features for movies of the given data-set. We are given the categorizations of about 1600 movies into 19 genres, and ratings of some users to a specific movie. We applied matrix factorization to the sparse ratings matrix (since not all users are going to rate all movies) to look for latent factor matrices of the movies and users. Then we used the principal component analysis (PCA) to analyze the latent factor matrix of movies and projected each movie to the two strongest latent factors. Finally, we did visualization and interpretation on each category and compared the category average of the two major latent factors.

All team members have contributed equally to this project. Also note that we are submitting a well-documented Python file (project2.py), which is mostly the implementation of matrix factorization, along with this report.

2 Matrix Factorization

2.1 Learning Model Overview

Basically, we want to find latent factors of movies and users from the sparse rating matrix. Formally, the problem set-up is as follows:

There are M users and N movies in total, and each user has rated some movies from 1 to 5. Denote Y be the M by N rating matrix, where Y_{ij} is rating if user i has rated movie j and 0 otherwise.

Let K be the dimension of latent factors, which is a parameter we are free to choose. Denote the latent factor matrix of users by U and latent factor matrix of movies by V, where U is a K by M matrix and V is a K by N matrix. (Note each column of U is the latent factor vector for a specific user, and similarly for V.) And our goal is to find U and V such that $Y \approx U \cdot V^T$, i.e. $U \cdot V^T$ will fit the non-zero entries of Y. Also denote u_i be the i^{th} column vector of U, and same for v_j for V.

Finally, define the observable indicator matrix w_{ij}, such that $w_{ij} = 1$ if Y_{ij} a valid rating (i.e. $Y_{ij} > 0$) and $w_{ij} = 0$ if Y_{ij} is not a rating (i.e. $Y_{ij} = 0$).

Hence for fitting Y with U and V, our objective function becomes

$$\arg\min_{U,V} \left[\frac{\lambda}{2} \left(||U||_{Fro} + ||V||_{Fro} \right) + \sum_{\substack{i=1..M \\ j=1..N}} \omega_{ij} \left(Y_{ij} - u_i^T v_j \right)^2 \right]$$

Finally, since U and V will only model the interaction between users and movies well, but there may be some individual offset for each user and each movie (some idiosyncratic factor), then for the standard matrix decomposition algorithm, we also introduced the offset vectors for movies and users to take this individual factor into account. See detailed discussion and implementation in Section (2.3).

2.2 Stochastic Gradient Descent

Two popular methods to solve the above optimization problem are stochastic gradient descent and alternating minimization.

However, the problem with alternating minimization approach is that it require the computation of matrix inverses. Since our data-set is of a very large dimension and sparse, the resulting inverse matrix obtained by using numerical Python packages would be just approximate and very inaccurate in the sense that its product with the original matrix is far way from the identity matrix. This numerical limitation will definitely cause problem in solving this optimization problem. Hence, we used stochastic gradient decent to train our model.

Before running the stochastic gradient descent algorithm, we first compute the partial derivatives of the objective function with respect to columns of U and V:

$$\partial u_i = \lambda u_i - 2 \sum_j \omega_{ij} (y_{ij} - u_i^T v_j) v_j,$$

$$\partial v_j = \lambda v_j - 2\sum_i \omega_{ij}(y_{ij} - u_i^T v_j)u_i.$$

The idea of stochastic gradient decent is such that, instead of evaluating the above sum of derivatives (which is expensive), at each iteration, it estimates this gradient on the basis of a randomly picked pair of i and j. That way, stochastic gradient descent can drastically reduce training time.

Moreover, we set the stopping criterion of the stochastic gradient decent algorithm as follows: it stops when one of the following two conditions is satisfied:

1. The algorithm has been iterated 1000 times.
2. The norms of $U - newU$ and $V - newV$ are both less than threshold $= 0.001$, where $newU$ and $newV$ are the updated latent motive factor matrices after one iteration of the stochastic gradient decent algorithm.

Furthermore, for this algorithm, we have four parameters: K is the dimension of the latent factors; λ is the regularization parameter; α is the initial learning; $decay$ is the decay of learning rate after each iteration respectively. Since for movies, we have 19 genres, then in order to greatly keep this feature space, we choose K to be 20. For the other three factors, see Section2.4 for further discussion of choices of those parameters.

In pseudocode, our stochastic gradient descent algorithm is as follows:

```
• Step 1: Initialize U and V such that each entry is uniformly distributed on $(0, 1)$.
• Step 2: Repeat until the stopping criterion has reached
  run:
      • Step 2.1: Randomly shuffle the set of pairs of users and movies whose ratings are observed in the training set
      • Step 2.2: For each $(i, j)$ in the shuffled set, do:
          • $\partial u_i = \lambda \cdot u_i - 2(Y_{ij} - u_i^T \cdot v_j) \cdot v_j$
          • $\partial v_j = \lambda \cdot v_j - 2(Y_{ij} - u_i^T \cdot v_j) \cdot u_i$
          • $u_i = u_i - \alpha \cdot \partial u_i$
          • $v_j = v_i - \alpha \cdot \partial v_j$
      • Step 2.3: Update our learning rate, such that
          • $\alpha = decay \cdot \alpha$
```

2.3 Adding Offset Vectors for Movies and Users

We use latent factor matrix to model the interaction between users and movies. However, the rates may depend greatly on the user or movie themselves, rather than their interaction. For example, some users tend to give high rates to movies while others tend to give low rates. In this case the rates different users give to the same movie may vary a lot even though they have similar latent factors. Similarly, two movies with same latent factors may have different reputations: *Titanic* will definitely get higher rates than average romantic movies. So it is wise to add an offset term to every vector of user and movie.

Now our objective function becomes

$$\arg\min_{U,V} \left[\frac{\lambda}{2} \left(||U||_{Fro} + ||V||_{Fro} \right) + \sum_{\substack{i=1..M \\ j=1..N}} \omega_{ij} \left(Y_{ij} - u_i^T v_j - a_i - b_j \right)^2 \right]$$

in which a_i is the offset for the ith user, and b_j is the offset for the jth movie. For each iteration, the step 2.2 of the algorithm in the previous section should be modified as

- $\partial u_i = \lambda \cdot u_i - 2 \cdot (Y_{ij} - u_i^T * v_j - a_i - b_j) * v_j$
- $\partial v_j = \lambda \cdot v_j - 2 \cdot (Y_{ij} - u_i^T * v_j - a_i - b_j) * u_i$
- $\partial a_i = 2 \cdot Y_{ij} - 2 \cdot u_i^T \cdot v_j - 2 \cdot a_i - 2 \cdot b_j$
- $\partial b_j = 2 \cdot Y_{ij} - 2 \cdot u_i^T \cdot v_j - 2 \cdot a_i - 2 \cdot b_j$
- $u_i = u_i - \alpha \cdot \partial u_i$
- $v_j = v_i - \alpha \cdot \partial v_j$
- $a_j = a_i - \alpha \cdot \partial a_j$
- $b_j = b_i - \alpha \cdot \partial b_j$

2.4 Choice of Parameters

There are mainly three parameters we can change: the regularization coefficient λ, the initial learning rate α and the learning decay. We try different combinations of them. Throughout our experiment, the number of latent factors is 20 and the stopping criteriion is when the norms of the partial derivatives are smaller than a threshould, which is fixed to be 0.001. The metric of accuracy is chosen to be the average absolute value of difference between the rate and our estimate for every non-zero entry in the rating matrix Y.

From Table 1 we can see the choice of λ does not make much difference. While different combinations of the initial learning rate and the learning deccay can lead to very different accuracies. Since what we want to minimize

Table 1: Accuracy (%) of Estimate with Different Parameters

λ	$InitialLearningRate$	$LearningDecay$	$Accuracy$
0.001	0.0001	0.95	1.31879464387
0.001	0.0001	0.8	1.73030876242
0.001	0.0001	0.75	1.72909538936
0.001	0.0001	0.5	2.07256391148
0.001	0.001	0.8	2.38177960544
0.001	0.001	0.5	1.18320588213
0.001	0.001	0.2	1.18426200357
0.001	0.01	0.5	2.10579411164
0.01	0.001	0.5	1.18567987242
0.0001	0.001	0.5	1.19260177415

is non-convex and may have many local minimal points, the accuracy is sensitive to the learning rate of gradient descent. For instance, if the initial learning rate is too small or the rate decays too fast, we may get trapped in a well and can never jump out of it. For each of the combination we run the code for 2 times and it turns out the results of same parameters are nearly the same. We choose λ, the initial learning rate α and the learning decay to be 0.001, 0.001, and 0.5.

3 Visualization and Interpretation

3.1 Principal Component Analysis

We use principal component analysis to identify the most meaningful basis to re-express U and V. We hope that the new basis will filter out the noise and reveal the hidden structures. We illustrate the case of V as follows: First we obtain the singular value decomposition (SVD) of V as $V = A\Sigma B$. Let $A_{1:2}$ be the first two columns of A, which corresponds to the best two-dimensional projection of the movies V. Define $\tilde{V} = A_{1:2}V$, which presents movies as points on a two-dimensional space.

Note for our implementation of SVD, we directly used the APIs provided by Scikit-Learn.

3.2 Visualization

To better understand what the two strongest latent factors we computed from PCA are, we projected our V on those two factors, and selected several movies to plot on a graph. The goal is to see whether those two latent factors relate to the genres of the movies, which are the only given features.

Hence, for each genre, we selected the top 50 (or the number of movies of this genre, whichever is less) most-rated movies, and plotted them on a 2D space. We also computed the average 2D projection of the top 2/3 most-rated movies of every genre for better inter-genre comparison.

Finally, to see whether we can get some insights on U, the latent factors of the users, and its correlation to the factors of movies, we used the same projection on V to project on U. We plotted all users and then the top 50 most-active users (ordered by number of movie ratings) to see their distributions of 2D projection.

Note, on the graphs, we are using abbreviated documentary as ”doc” and film-noir as ”noir”.

3.2.1 Each Genre of Movies

As stated above, for each genre, we plotted the top-50 most-rated movies. All the graphs are demonstrated in Section 4 *Appendix*.

For the first latent factor (i.e. the horizontal axis), we can conclude the following from those graphs:

1) Documentary movies have a rather concentrated first latent factor and they have the largest values in the first latent factor.

2) Animation, children's, fantasy, film-noir, horror, musical, and western have similar concentrated values in the first latent factor, and they are neither small nor large compared to other genres.

3) Crime, mystery, Sci-Fi, war movies also have similar shapes and location in the distribution. Their values in the first latent factor are also neither small nor large, but for each of those genres, their distributions are not so concentrated (i.e. they have more small-value outliers) compared to the group above.

4) Action, adventure, comedy, drama, romance, thriller movies have the smallest values in the first latent factors, and their distributions are rather concentrated.

We can see indeed that each movie does have different first latent factor if it belongs to some different groups of genres.

For the second latent factor (the y axis), on the other hand, we do not see it is very sensible to the change of genres. For each genre, the distribution of the second factor genre are very similar to one another, i.e. they seem to be equally distributed in some similar range. We then conclude that the second latent factor may not depend on genres.

3.2.2 Distribution of Genre Averages

Figure 1 demonstrated the distribution of the average projection of the top $\frac{2}{3}$ movies for each genre.

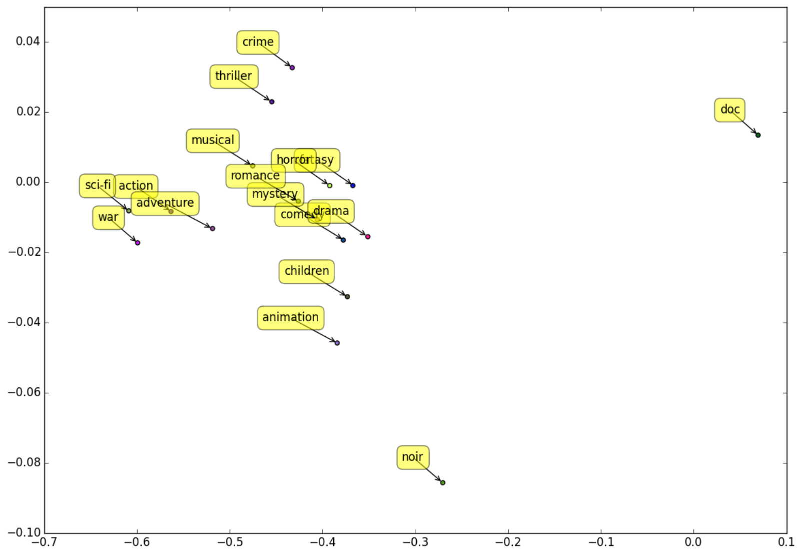


Figure 1: Distribution of the Average of Top $\frac{2}{3}$ Movies for each Genre

We can indeed see that the average Documentary does have the largest first latent factors, as we have seen in the top-50 movie distributions above. On the other hand, action, adventure, war, and Sci-Fi have the smallest first latent factors. All the rest genres have medium first latent factors. The average analysis actually mostly corresponds with the results from the top-50 analysis.

For the second latent factor, we can see that film-noir have the smallest value which really stands out from the rest, and animation and children's have relatively small second latent factor value compared to others. On the other hand, crime, thriller, and documentary have the largest values of second latent factor.

3.2.3 All Users and Top-50 Users

We also projected the latent factors for all users, and the top 50 most-active users, in Figure2.

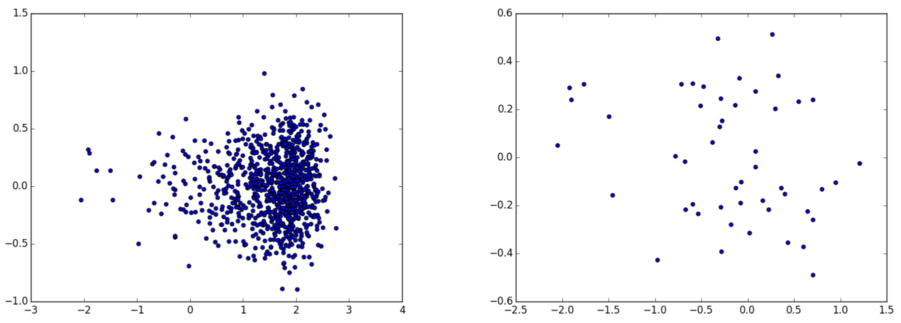


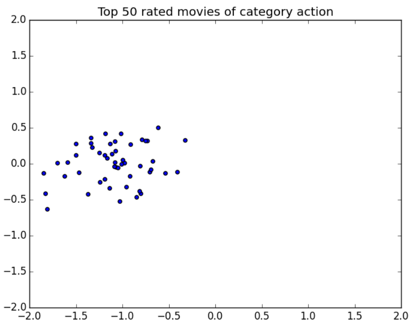
Figure 2: Distribution of All Users (left) and Top 50 Active Users (right)

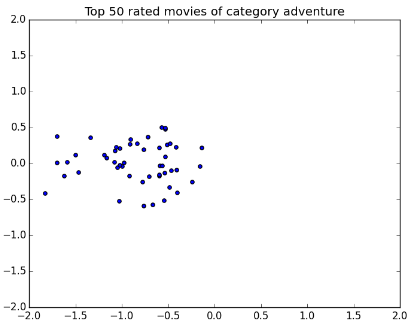
For the distribution of all users, we can see that the second latent factor is again rather equally distributed, while for the first latent factor, the distribution is rather skewed: most of the users are concentrated at the large values, while there are a large number of small-value users.

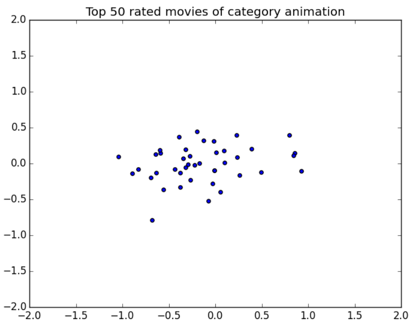
However, for the distribution of the top 50 active users, the distribution seems to be very uniform, unfortunately.

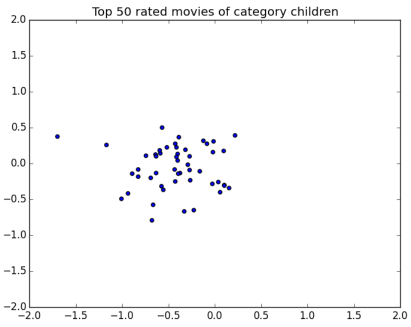
4 Appendix

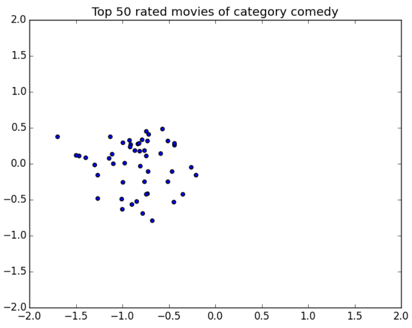
4.1 Plots for Top 50 most-rated Movies of Each Genre

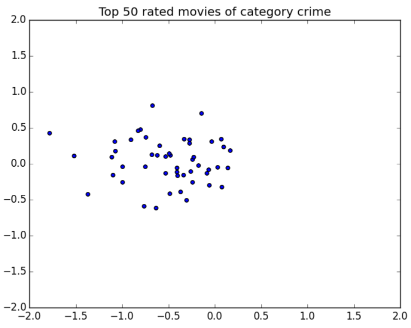


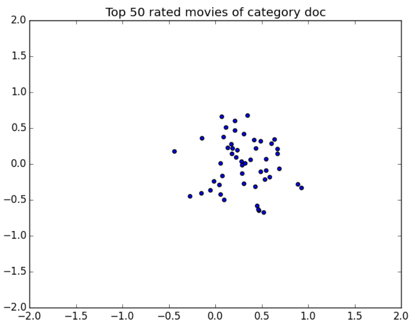


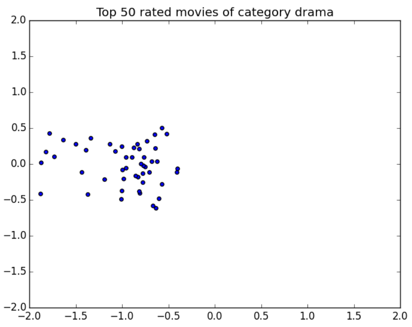


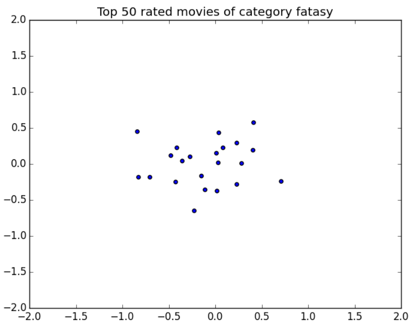


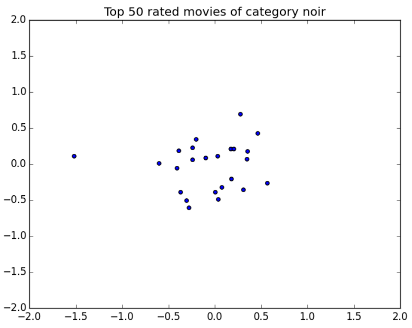


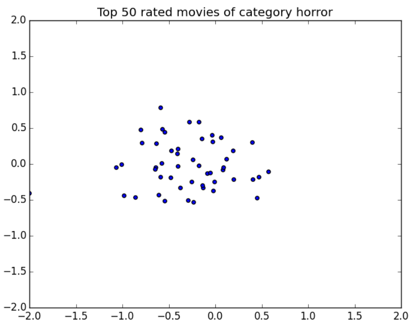


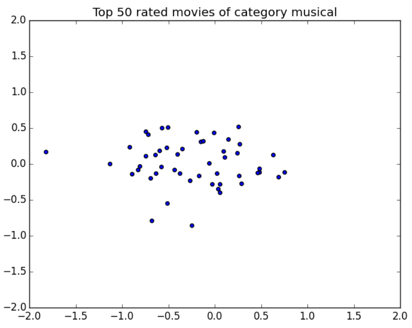

Top 50 rated movies of category doc

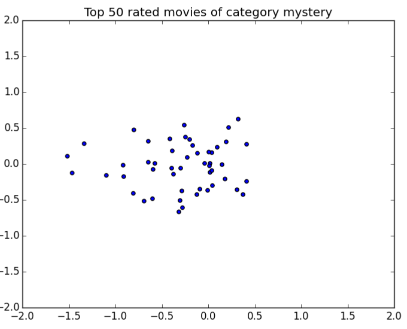

Top 50 rated movies of category drama

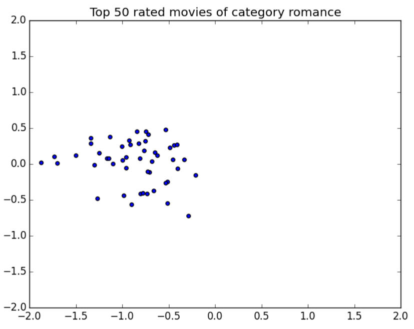

Top 50 rated movies of category fatasy

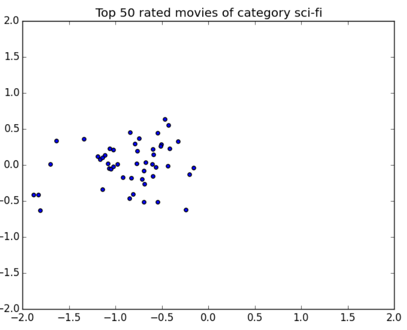

Top 50 rated movies of category noir

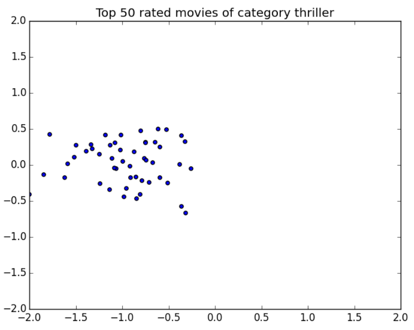

Top 50 rated movies of category horror

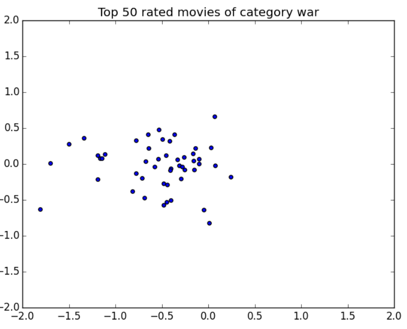

Top 50 rated movies of category musical

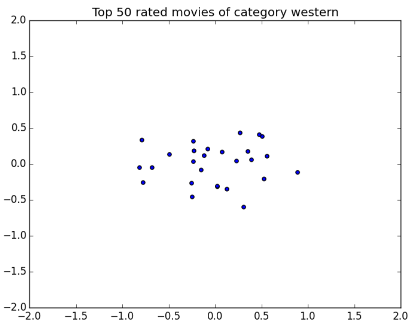

Top 50 rated movies of category mystery


Top 50 rated movies of category romance


Top 50 rated movies of category sci-fi


Top 50 rated movies of category thriller


Top 50 rated movies of category war


Top 50 rated movies of category western