Movie Rating Problem

Jeng, Ya-wen(鄭雅文)
r06922097@ntu.edu.tw

➢ Introduction

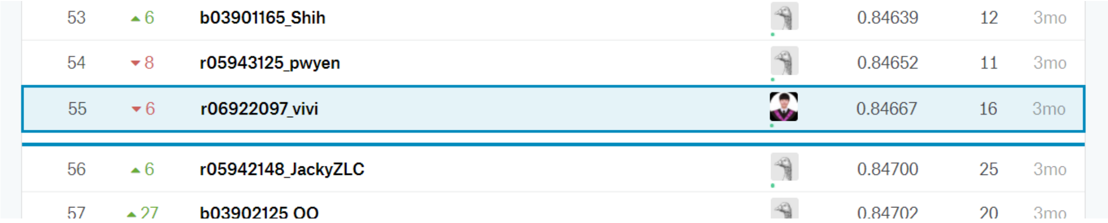
I have been exposed to this issue before, so I will show the experiment I have conducted and go through this topic to do analysis.

The problem I want to solve here is to **predict the rating of unseen user-movie pair given user ID and movie ID**, using matrix-factorization methods.

➢ Related Work

I have joined a Kaggle competition which is part of the homework of “Maching Learning” course. My result ranked 55th out of 335 teams. It shows that I can make a good prediction of rating given user ID and movie ID.

Some specifications	
Training data	UserID, MovieID, Ratings
Testing data	UserID, MovieID
Score evaluation	RMSE(the lower the better)



53	▲ 6	b03901165_Shih	0.84639	12	3mo
54	▼ 8	r05943125_pwyen	0.84652	11	3mo
55	▼ 6	r06922097_vivi	0.84667	16	3mo
56	▲ 6	r05942148_JackyZLC	0.84700	25	3mo
57	▲ 27	b03902125 OO	0.84702	20	3mo

➢ Experiments

Since the dataset is quite different between the one I used and the given dataset(https://grouplens.org/datasets/movielens/20m/). I re-train the model using the given dataset and split the data in `ratings.csv` into training set(90%) and testing set(10%) with random shuffle.

After 10 epochs, the RMSE is about 0.8230 on training set, and 0.7060 on the testing set.

Movie type estimation:

As training the matrix factorization model, we can obtain user and movie embedded vectors, which represents some features about movies/users and relationships between them, with arbitrary dimensions.

I use a 500-dimension vector to represent a movie/user. With TSNE to reduce dimension and K-means to classify the movies (about the original vectors) into 5 categories, I get the following results:

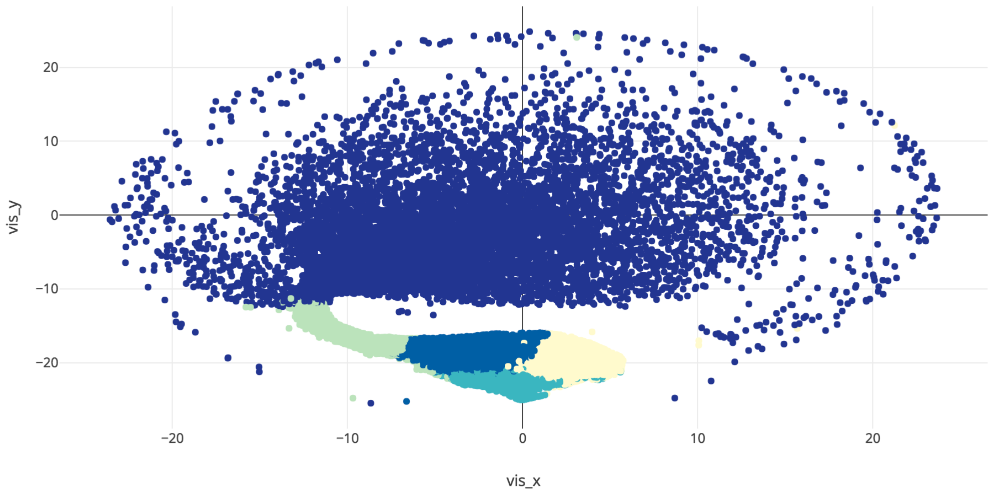


Figure 1. Visualization of Movie Embedding Layer through TSNE and K-means

The plot shows that the K-means categories might be underestimated, so the blue area is too big for us to learn information.

I also count the percentage of genres appearing in each category (the following figure), it shows that the K-means categories can barely identify the genres either. The interesting thing here is that the trend in Category 2 is quite different from the others. I find that this Category is more latest movies than the other categories and that its genres is more acceptable by universals, e.g. Adventure, Action, Comedy, Sci-Fi.

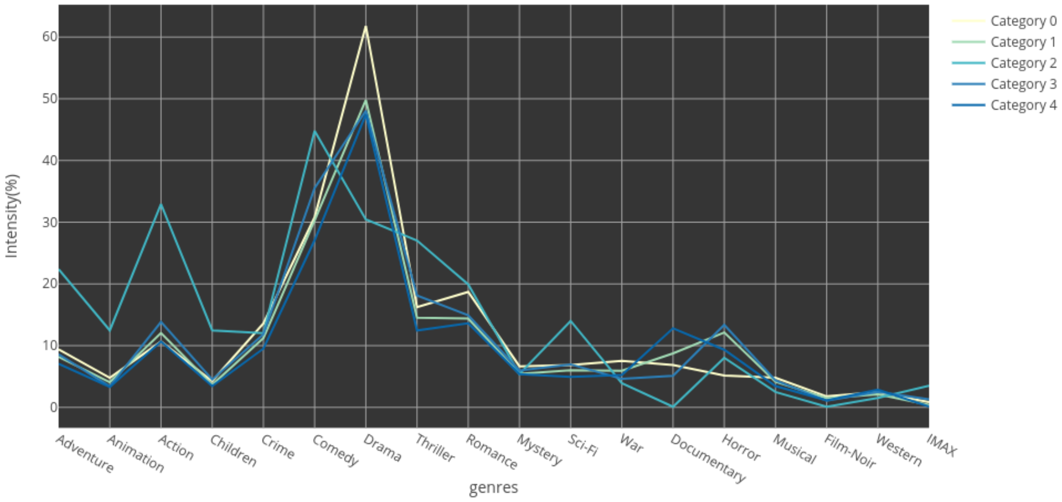


Figure 2. Percentage of genres in each estimated category

Then I plot another scatter plot showing the amount of rating people. I finally find the problem: some movies were rated by few people (the gray area is the rating count lower than the others), so it is hard to learn its behaviors by the machine.

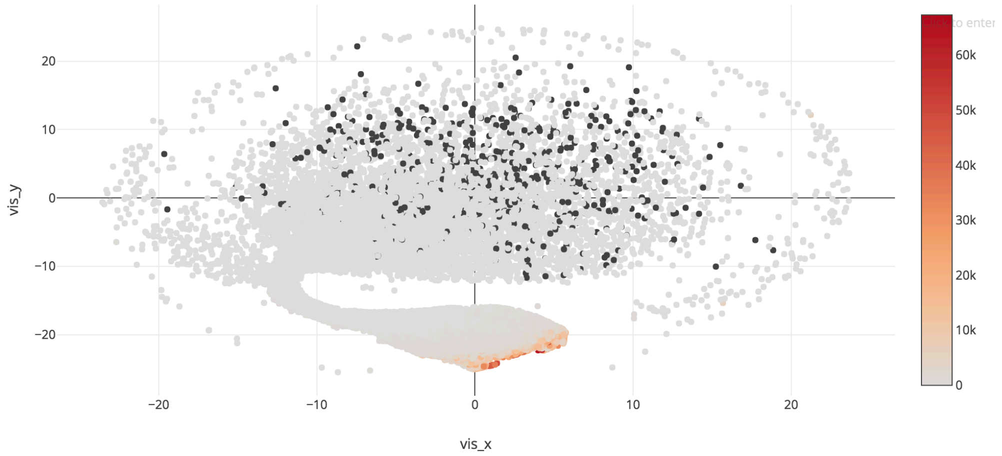


Figure 3. Rating Count of each Movie

After removing the movie which rating count less than 1000, the plot becomes:

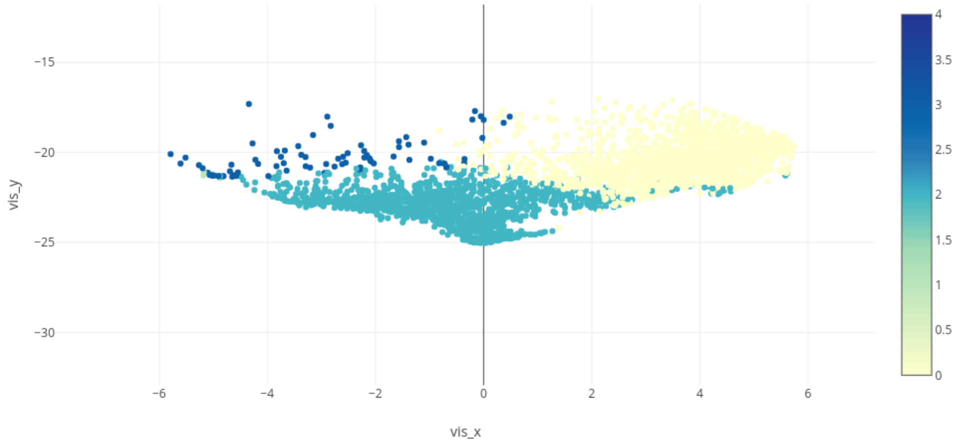


Figure 4. K-means Estimated Categories

This figure extracts the movies with higher rating counts and is labeled by its estimated Category number.

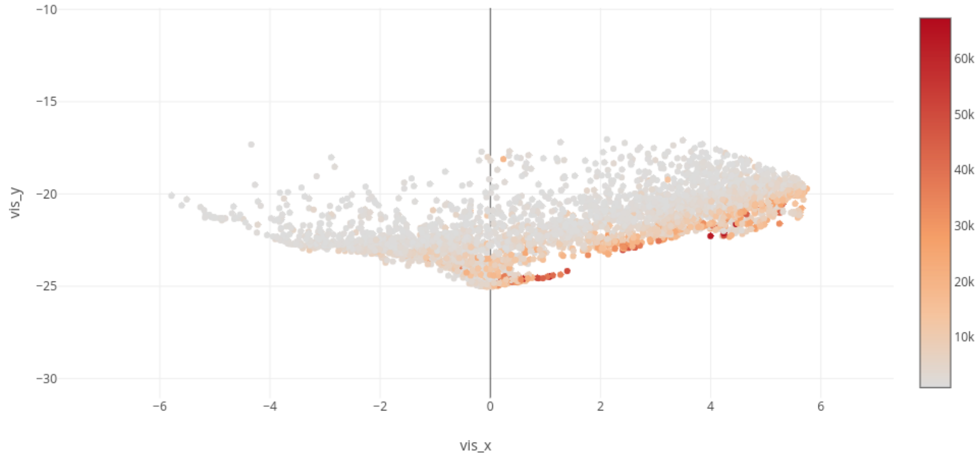


Figure 5. Number of Rating People

This figure is labeled by its rating count. (Ranging > 1000)

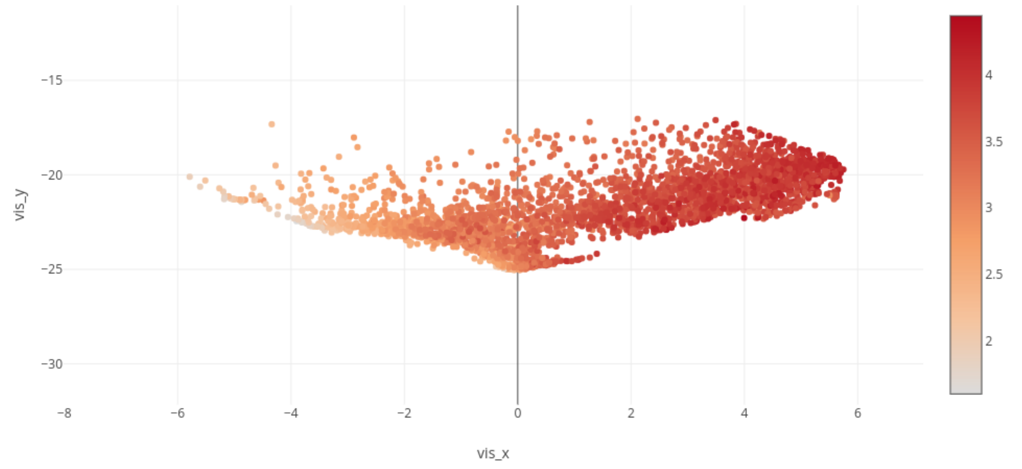


Figure 6. Movie Ratings

This figure is labeled by movies' real ratings. (Ranging 0-5)

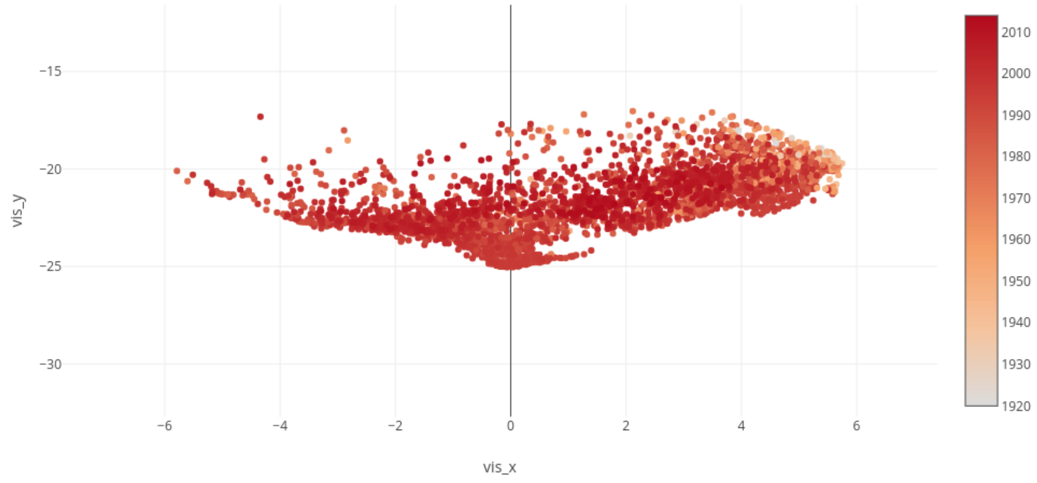


Figure 7. Movie Produced Year

This figure is labeled by movies produced year.

➢ Analysis & Conclusion

We can infer some conclusions through these figures:

- The movie embedding vectors can really be plotted to show some information:

1. It shows the relationship between movies and it is gradient.
2. In figure 5, the closer to bottom, a movie is more popular (more rating count).
3. In figure 6, the closer to right hand side, a movie is more recommend by people (higher rating).
4. In figure 7, the closer to upper right corner, a movie is older, but is it not obviously.

(Note that the training process only uses user ID, movie ID and their ratings).

- The K-means categories can be interpreted as:

1. Category 0: the highest rating score ones, and its main genres is Drama.
2. Category 2: the most popular ones (it may happen because the movies which are latest were rated by users more accessible to rating systems.), and its main

genres are Adventure, Action, Comedy, Sci-Fi.

- If we want to make a recommendation system, we can convert the movie/user/rating records into movie embedding vectors. The more similar between two movie vectors, the more possible that these two movies behaves similar patterns to users. Additionally, we can search recommended movies by number of ratings (In figure 5, the downside direction) and ratings (In figure 6, the right direction).

➢ Future Works

1. Analyzing users embedding layers as doing with movies embedding layers.
2. Using rating timestamp to see that people watch movies in what order.
3. Finding out what kind of movies or what kind of users are more predictable.