# Stat 542 Third Project-Recommender System

# Team: RSS

## 1. General Idea

(1) Data cleaning steps

First, we load the train and test file and split the strings in movie.dat and user.dat. Then, by merging with MovieID and UserID and converting it with as.data.frame function, we can generate a rating matrix with all the information (UserID, MovieID, ratings, genres, ages, etc.) included.

It's worth noting that some of the functions in this field required certain format in data frame. Be sure to look at the format requirement and transform the dataset into the correct form before applying.

(2) Model selection steps

As a general idea, the rating of a movie should have three effects:

User effect (Whether a user prefers to give higher score),

Item effect (Whether a movie is good and generally receive higher score)

Linear covariates effects (genre effect, age effect, occupation effect, etc)

Obviously, we can build a simple model based on a simple linear regression model based on these three effects and generate a simple model to predict the ratings.

This method is packaged in "rectools" package and can be applied by function trainMM with the rating matrix as the parameter.

We can see the result including three factors is close to a simple linear regression just with user effect and item effect, suggesting that linear covariates effects may not have too much influence in the final recommendation system.

| | User effect, Item effect | All Effects |
|---|---|---|
| RMSE | 0.9328680 | 0.927404 |

By further analysis, we can see that the coefficient of item effect in a linear model is around 1.19 while user effect with 0.86, meaning that item effect is the most influential effect here.

Another model is based on package “recosystem”, a newly published package. The general idea of this model is to approximate the ratings matrix by the product of two matrixes of lower dimension representing each user and each item. The full description is in this website:

https://cran.r-project.org/web/packages/recosystem/vignettes/introduction.html

The packages and functions mentioned above are newly published and still under developing. There are still some bugs in these functions when we applied them in our project. Also, some of the althogrim like SVDplusplus, which is available in python, is not available in R now.

We hope to see a steady and powerful package in R soon.

## 3. Model Performance

| | Performance | | | | |
|---|---|---|---|---|---|
| Running time (i7-6820HQ, 32GB RAM) | Data cleaning time:4.86 seconds | | | | |
| | Modeling time:164.14 seconds | | | | |
| Model1(trainMM) Rmse | 0.9264 | 0.9285 | 0.9273 | 0.9262 | 0.9269 |
| Model2(recosystem) Rmse | 0.8598 | 0.8632 | 0.8603 | 0.8610 | 0.8597 |

## 4. Conclusion

(1) This dataset is an item-based dataset since coefficient of item effect is higher. On the contrary, the prediction accuracy will not change much even we totally remove linear covariates effects in the simple linear regression model. It means that good movies are always easier to be accepted despite the preference might be different.

(2) As a relatively new topic in statistical learning, there is not many mature packages or functions can be directly applied in building a recommendation system right now.

(3) During our training process, the minimal train error is around 0.78, suggesting that the best result might be around 0.80-0.85. SVDplusplus in python might have slightly better outcome.

## 5. Acknowledgement

The second modeling idea is from Yixuan Qiu, a 4th year Phd student of Purdue University, Department of Statistics. More information can be found in his GitHub https://github.com/yixuan/recosystem.