Comp 135 Project3 Report-Recommendation System
Jialin Zhang

Collaborative Filtering: python code: recommmendation-system.py

The hyperparameter of *SVD* is n_factors which is K and reg_all which is λ. I set K as [50, 75, 100, 125, 150] and λ as [0.01, 0.05, 0.1, 0.5, 1]. Then I use *GridSearchCV* to find **the best parameter, K =100, λ= 0.05. A**fter I have trained the recommendation system, I moved to evaluate it against the test set by two approaches.

First approach: I use the model trained before to make prediction and then compare the prediction against the true rating in the test set only. The *mean absolute error* is 0.7293, which is quite high. The explanation is that it cannot well represent real applications as we don't know which item a user will view in the future. The evaluation does not consider the feedback. For example, if a user doesn't view some items in the test set, it is likely because the user doesn't like these items. This evaluation method doesn't use such information. Thus, it is not a good evaluation method for this problem.

Second approach: let the model to recommend 5 items for each user, then evaluate the quality of the 5 recommended items by their average rating. To recommend 5 items to a user, I use the model to predict the user's rating of all items which are not reviewed in the training set and then recommend top-ranked ones. If the pair of the user and the recommended item is in the test set, then use the true rating. Otherwise, assume the true rating is 2, meaning that the user doesn't like the item. Lastly, calculate the overall average rating of all recommended items. It is easy to recommend top 5 items to a user. The hard part is the transfer of trainset and testset. I have to transfer raw data to the inner data to use the package-*Surprise*. Finally, I solve the hardest part by composing a **3-layer for-loop** to replace the trained rating of recommended items by the true rating and 2. And then I record the total rating of the recommended 5 items by users and calculated the *overall average rating*, 2.113, which is almost 2 and a proper reason is the most ratings of the items do not appear in the testset.

Vector Analysis: python code: feature-analysis.py

I use *Logistic Regression Classifier* to analyze the user vector and obtain the *cross-validation error* as 0.286, the *standard deviation* **is 0.003.** I use *Random Forest Regression* to analyze the item vector and obtain the *mean squared error,* 188.204, and the *mean squared error* obtained from *naïve mod*el is 203.0**44**.

Before I start to train the models, I prepare the data first. At the beginning, I merge the trainset csv file and the testset csv file together and remove the title in the middle and construct a new csv called data.csv containing the new dataset which is used to obtain the user vector and item vector later. With the data.csv file, I transfer the data to surprise trainset. And then I use the model trained before to decompose the data to obtain the user and item vector which now is the inner data. Then I transfer the inner user and item vector to raw ones. And I remove the

titles in original releaser year file and gender file to make them usable. After preparing these data well, I turn to vector analysis.

In the user vector analysis part, I choose to use *logistic regression classifier* and tune the hyperparameter λ. I get a *cross validation error* as 0.286 with *standard deviation* as 0.003. This error is reasonable and means the user features have information about the users but not exactly. Still, we can use the user features.

In the item vector analysis part, I choose to use *random forest regression* model and tune the hyperparameter maximum depth. The *best maximum depth* is 3 and the *mean squared error* is 188.204, which is near to the naïve model result 203.044. My random forest regression model gives me a mean squared error near to the naïve model result, which roughly means the item features don't have information about the movies at least cannot well represent the movies. The expected result should be the mean squared error obtain from my model is much smaller than the naïve model.

Question: How to incorporate user information and movie information into the recommendation model?

After SVD I get two vectors which represent some information of the user and item respectively. Here, I have one more information about user which is the gender information. I can add the gender information to the user vector as the addictive feature. And the same as item, I can add the release year to the item vector as the addictive feature. These two feature can give me additional information about users and items which will be helpful to make a prediction whether the user will like a new item or a new item will be clicked by the user.