MovieLens

Maxime L.

8/11/2019

Overview

Our goal is to generate a movie rating prediction algorithm with the lowest RMSE. We use the MovieLens 10M dataset provided by GroupLens in order to train then test the algorithm. You can find more info on the source file at the following address: https://grouplens.org/datasets/movielens/10m/.

The steps are as follows:

1. Download & Import the Data

2. Prepare the trainings & validation datasets
3. Prepare a regularization & RMSE calculation function
4. Try out different ranges of lambda
5. Train the algorithm

6. Get RMSE for the validation dataset

Method

1. Download & Import the Data
   First we initialize with the required packages, installing them if necessary. Then we retrieve the zip from http://files.grouplens.org/datasets/movielens/ml-10m.zip
   We extract the ratings by userId and movieId from the ratings.dat file, giving names to the columns. We extract the movies id, title and genre from the movies.dat file giving names to the columns. We transform movies in a dataframe with the data structure matching ratings.

2. Prepare the trainings & validation datasets We create a train set with 90% of the data and validation set with 10% of the data, using seed of 1 for validation purpose. We only keep movies & users in the validation set which are also in the train set. We add the removed movies & users to the train set. We further create train set and test set with 90% / 10% of the train set data for cross-validation. We only keep movies & users in the test set which are also in the train set. We add the removed movies & users to the train set. We remove now unecessary variables.

3. Prepare a regularization & RMSE calculation function We prepare a regularization & RMSE calculation function that takes 3 input: lambdas, train set & test set, defaulting to train and test sets. The function regularize by movie, as we acknoledge the movie effect, then by user, as we acknowledge the user effect.

i. Average ratings are calculated for the whole training data set and stored in mu.
ii. Regularized ratings per movies are calculated per movies and stored in b_i.
iii. Regularized ratings per user are calculated per user, substracting the movie effect, and stored in b_u.
iv. Predicted ratings are generated using the test data set, with the formula mu + b_i + b_u.
v. RMSE is calculated and returned by the function

4. Try out different ranges of lambda Different ranges of lambdas were tried for the function. Granted that lower-hinge rating number per user is 32 & by movies is 30 (on the training dataset), we started with the range 0-100 by 10 increment. Then we reduced the scale by 1/10 until we got the a RMSE below 0.8649 for lambda = 4.6

5. Train the algorithm
   With the best lambda, we trained the algorithm using the whole training dataset, and calculated the RMSE using the validation data set.

Results

We were able to get the RMSE under 0.8649, at 0.8648224 by regularizing by movie & by user, using this optimal lambda. While it took some time to iterate to find the best lambda, the training is quite fast, making potential use more realistic.

```
## [1] 0.8648224
```

Conclusion

Using regularization per movie and per user, we managed to predict movies ratings to an acceptable level. The trained algorithm is able to work on large dataset, and can be easily replicated with different lambdas. However, it only works for movies with existing ratings, all the more so it works best when you have a certain number of ratings, and will fail to evaluate properly new movies. We would need to add more parameters to predict the ratings of new movies, such as the genre, the cast etc. Adding the genre allow us to lower the RMSE to 0.8629981, but it requires changing the validation data so we did not pursue it here. Thank you for your attention.