ASSIGNMENT 5

CS5304 - LARGE SCALE RECOMMENDATION SYSTEMS

1. Assignment

In this assignment, you'll be building a recommendation system. You'll be given data that comprises Users, Items, Ratings and Timestamps. For this assignment, we'll not be using the timestamps. We'll focus on the User-Item Utility matrix and build latent-factor based recommendation systems. You'll evaluate these systems in two ways: Objective evaluations via RMSE (root mean square error) metrics on Test data and Subjective evaluations by generating a list of recommendations for all the users on one of the test partitions.

2. Data Set Details

We will be working with the MovieLens data set which includes ratings by a large number of users on a collection of movies. We'll be working with the **10M** data set which includes 10 million ratings applied to 10,000 movies by 72,000 users.

- MovieLens 10M data set

3. Task 1

Taking ratings data from the MovieLens10M data set, build a matrix factorization model in PyTorch with a small number of latent factors, not more than **5** latent factors.

To split the data into training and test sets, please use the **split_ratings.sh** script provided with the dataset, which will split the data into 7 different sets. Use the ones named r1-r5 for your experiments.

Please include L2 regularization in your model to ensure that the weights matrices are not too large. Make sure you try different regularization parameters $[\lambda \in (0.001, 0.01, 0.1)]$ and select the model that gives you the best RMSE under 5-fold cross-validation.

4. Task 2

So far, we used the ratings as they were. We didn't try to remove bias factors. There are several ways of removing bias and building recommendation systems on the deviations. Incorporate bias terms in your factorization model and retrain the recommendation engine. You will repeat 5-fold cross-validation to select the best regularization parameters.

5. Task 3

Take one of your factorized models and use it to build a recommendation engine. Given a user from the closed set, it should come back and give movie recommendations. Run this program on **r5.test** and produce recommendations for each user in r5.test. For each user in this data set, you should produce the top 5 recommendations.

You may find the following links helpful:

- Matrix Factorization in PyTorch
- Spotlight package in PyTorch

6. Assignment Submission

In your submission, we'll need the following

- Your code that documents the experiments you performed including data reading, splitting, cross validation and testing. This should be in a file called **assign5.py**.
- Your experimental results documenting your validation of the 2 models (plain vanilla, bias removed model). Please also report the mean test set RMSE and Standard Error for both these models. This should be in a file called **assign5_report.pdf**.
- Your top-5 recommendations for all users in r5.test with each row in TSV format as follows: $< uid >< rec1 >< rec2 >< rec3 >< rec4 >< rec5 >$. This should be in a file called **assign5_r5results.tsv**.

7. Grading Rubric

- Model implementations (correctness, documentation, etc.) - 7 points
- Writeup and analysis - 5 points
- Results file in proper format - 3 points

8. Regrading Policy

We'll accept regrade requests for 1 week after releasing the assignment grade. Please email Andrew (apd64@cornell.edu) with why you think your assignment should be regraded. We're also happy to clarify the meaning of comments. We will regrade the entire assignment, and it is possible that after the regrade your assignment might have a lower score.