

## Lab 5 – A Recommender System Using MovieLens

**Assigned: 4/21/16**

**Due: 4/30/16 by 11:59pm**

For this lab, you will be creating your own recommender system using the MovieLens dataset and using it to recommend yourself some movies.

### Setup

Go into your Git directory and do a “git pull” to get the new materials for this assignment. You should see a “lab4” directory if the pull was successful. If you do not see it, contact me.

### Movie Recommender Using Collaborative Filtering

In class we discussed how collaborative filtering can be used to build recommendation systems. In this lab you will be doing just that. Your task is to build a recommendation system using MLLib, collaborative filtering, and the MovieLens dataset.

#### Part 1: Building the recommender

You will be using the MovieLens ml-100k dataset (same one you used for lab2) to build your recommender. As always, split the dataset into a training and validation set (there is no test set for this assignment). The randomSplit method in MLLib will work just fine.

Next, use MLLib’s built-in ALS matrix factorization module to build a recommendation model:

<http://spark.apache.org/docs/latest/mllib-collaborative-filtering.html>

In your report, answer the following questions:

- Fix the rank at 10 plot how the validation set accuracy changes as you vary the number of iterations from 5 to 20.
- Fix the number of iterations at 10 and plot how the validation set accuracy changes as you vary the rank from 5 to 20.

Note that to calculate the accuracy of your recommender, you must compare the recommender's estimated rating with the actual user's rating. **If the recommender is within 1 point of the actual rating, this is considered an accurate recommendation.** Otherwise, it is considered an incorrect recommendation. For example, let's say that the recommender estimates that user A will give movie X a 3 star rating. If user A actually gave movie X either a 2, 3, or a 4 star rating, then this recommendation was correct. Go through every datapoint in the validation set and count the total number of correct and incorrect recommendations. The final accuracy is simply the percentage of correct recommendations.

Keep tweaking the parameters until your validation accuracy is as high as you think it will get. Once you are done tweaking, train your recommender on the entire dataset.

In your report, answer the following questions:

- What were the final parameters you used for your model?
- With these parameters, what was the accuracy on the validation set?

## **Part 2: Testing the recommender**

Now, build a simple command-line based UI around your final recommender. When you run the program, the recommender should ask the user for their ratings on 10 random movies in the MovieLens dataset. After the users enter their ratings (a 0 rating means that the user hasn't seen the movie), the recommender should then recommend 10 new movies that the user should watch.

**Please put a README in your repo explaining how to use your recommendation engine.**

Once you finish this, run the recommendation engine on yourself.

In your report, answer the following questions:

- Which ten movies did the recommender system ask you to rate, and what were your ratings on those movies?
- What ten movies did the system recommend to you after analyzing your ratings?
- How accurate were the recommendations? That is, did the system recommend movies that you liked?