Homework 1: Recommender systems

Published Date:
March 7, 2019

Due Date:
March 22, 2019 @11:59pm

Description:

****
This is an individual assignment.
****

Overview and Assignment Goals:

The objectives of this assignment are the following:

- Use Apache Spark to build a Recommender system and predict
- Experiment with various similarity measures
- Explore hybrid CF systems
- RMSE will be used to test your submission

Detailed Description:
Develop a Collaborative Filtering system to predict as accurately as possible the user item ratings.

Collaborative Filtering (CF) systems measure similarity of users by their item preferences and/or measure similarity of items by the users who like them. For this CF systems extract Item profiles and user profiles and then compute similarity of rows and columns in the Utility Matrix. (In this assignment you are given a number of ratings, from which it is possible to build a utility matrix.) In addition to using various similarity measures for finding the most similar items or users, one can use latent factor models (matrix decomposition) and other hybrid approaches to improve on the training and test data RMSE scores. We encourage you use functions available in spark libraries for similarity computation, SVD decomposition etc. Performing these tasks in parallel on multiple cores is required as the dataset is quite large.

The goal of this assignment is to allow you to develop collaborative filtering models that can **predict the rating** of a specific item from a specific user given a history of other ratings.

To evaluate the performance of your results we will use the Root-Mean-Squared-Error (RMSE).

Caveats:

+ Use the data mining knowledge you have gained until now, wisely, to optimize your results.

+ The default memory assigned to the Spark runtime may not be enough to process this data file, depending on how you write your algorithm. If your program fails with
`java.lang.OutOfMemoryError: Java heap space`
then you'll need to increase the memory assigned to the Spark runtime. In general spark uses --driver-memory to set the runtime memory. On the HPC we will be running spark on top of Slurm, thus spark can only get as much memory as slurm allocates.

Data Description:

The training dataset consists of 85724 ratings and the test dataset consists of 2154 ratings. We provide you with the training data ratings and the test ratings are held out. The data are provided as text in train.dat and test.dat, which should be processed appropriately.

train.dat: Training set (UserID <tab separator> ItemID <tab separator> Rating (Integers 1 to 5) <tab separator> Timestamp (Unix time stamp).
test.dat: Testing set (UserID <tab separator> ItemID, no rating provided).
format.dat: A sample submission with 2154 ratings being all 1 (The one values shall be replaced with your predicted ratings in the order of the test.dat file).

Rules:

- This is an individual assignment. Discussion of broad level strategies are allowed but any copying of prediction files and source codes will result in an honor code violation.
- The recommender system should be implemented in Spark. Feel free to use the programming language of your choice for this assignment (Python, Scala, or Java).
- Some of your classmates may choose not to see the leaderboard status prior to the submission deadline. Please do not share leaderboard status information with others.
- The public leaderboard shows results for 50% of randomly chosen test instances only. This is a standard practice in data mining challenges to avoid gaming of the system. The private leaderboard will be released after the submission deadline, based on all the entries in the test set.
- In a given day (00:00:00 to 23:59:59), you are allowed to submit a prediction file only 5 times.
- The final ranking will always be based on the last submission, not your best submission. Carefully decide what your last submission should be.
- Each time you submit a prediction file, you will also need to include the code that generated that prediction. Acceptable formats are: py, tar.gz, zip. *Your submission will not be valid unless it produces the output in the prediction file.*

Deliverables:

- Valid submissions to the Leader Board website: https://coe-cmp.sjsu.edu/clp/ (username is your MySJSU email and your password is your MySJSU password). ***Submission system will open on 3/22/2019.***
- **Canvas submission of Problems solutions and report:**
  - In addition to the problem solution, provide a 2-page, single-spaced report describing details regarding the steps you followed for text processing and classifier model development. The report should be in PDF format and the file should be called **<SJSU_ID>.pdf**. Be sure to include the following in the report:
  - Name and SJSU ID.
  - Rank & RMSE-score for your submission (at the time of writing the report). If you chose not to see the leaderboard, state so.
  - Your approach.
  - Your methodology of choosing the approach and associated parameters.
  - Any special instructions for running your code.

Grading:

Grading for the Programming Assignment will be split on your implementation (80%) and report (20%). Extra credit (1% of final grade) will be awarded to the top-3 performing algorithms. Note that extra credit throughout the semester will be tallied outside of Canvas and will be added to the final grade at the end of the semester.

Files: In Canvas, you can find

- *Training Data:* train.dat
- *Test Data:* test.dat
- *Format File:* format.dat