CSE427 – Final Project #1: Collaborative Filtering using the NETFLIX Data

M. Neumann

Due 05/05/2016 1pm (**no extension!**)

Project Goal

In this project your group will predict 100,000 movie ratings for users in a subset of the original NETFLIX data issued for the NETFLIX Prize. This challenge aimed at substantially improving the accuracy of predictions about how much someone is going to enjoy a movie based on their movie preferences. It was issued by the Netflix company and on September 21, 2009 a $1mio Grand Prize was awarded to the winning team.[1]

Goal: Analyze the NETFLIX data using PIG (or MAPREDUCE) and, based on the outcomes of this analysis, develop a feasible and efficient implementation of the collaborative filtering algorithm in MAPREDUCE. After computing the predicted ratings, evaluate those ratings by comparing them to the true ratings (gold standard). Note that MAPREDUCE is only required to find the k-most similar users or items. You do not need to use MAPREDUCE for the predictions and the evaluation. **This is a competition!** Part of the grades for the results (10%) will be assigned according to a ranking of the number and quality of all teams' predictions!

Getting Started

Update your SVN repository, you will find additional materials for the *final project* in the folder `final_project/netflix`. You will have to agree to the following usage conditions to be able to do this project:

Usage Agreement

By using the dataset from the Netflix Prize for you agree as follows:

- I agree to the terms specified in Netflix Prize Rules (cf. README file provided in your SVN repo).
- I agree to **delete** this dataset once the project has been completed.
- I will not redistribute this data in any form.

[1] You can read more about it here: `http://www.netflixprize.com/`.

If you agree to the usage agreement above, download the data from:

```
https://classes.cec.wustl.edu/cse427/netflix_subset.zip
```

Indicating Group Work

Use the file `partners.txt` to indicate group work. Follow these instructions exactly, to ensure to get credit!

- `partners.txt` needs to include up to 3 wustlkeys in the first three lines (one line per wustlkey)
- **first line/wustlkey is the repository, where the solution is located**. We will **only** consider the submission in this repository!
- Every student in a group needs to have **the same** `partners.txt` in the `final_project/spark` folder in their repository (indicating that the partnerships are **mutually accepted**)!
- If you do not have a partner, try to find one! If you want to submit on your own, indicate your wusltkey in the first line of `partners.txt` and leave the second line blank.

Problem 1: Collaborative Filtering Approach

This problem is part of **milestone 1**.

Revise the collaborative filtering approach discussed in the lecture. Make sure you **understand the basic approach** and get a **firm grasp of its variable components**. In your approach you are free to take any combination of the following options/ values for the following parameters:

- similarity measure (JACCARD, COSINE, CORRELATION, other?)
- number of similar users k
- prediction method (weighted or un-weighted average)
- you may threshold the ratings
- you may normalize the ratings or not (cf. Alg 1 vs. Alg 2 discussed in the lecture)
- user-user model or item-item model

Read up on the descriptions of collaborative filtering in the literature (MMDS Chapter 9 and maybe this blog is helpful `http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/`; feel free to search for more sources online)
Describe the approach you are intending to use. Explain/Justify your choices.

Problem 2: Analyzing the Netflix Data

This problem defines **milestone 2**.

In this problem you will **analyze the input data to plan an efficient implementation** of your approach. The format of the dataset is described in the `description.txt` file. You can choose the analysis tool (MapReduce or pig (or even hive or impala)) to use for this problem. Remember to develop and test your implementations on a subsample of the data!

(a) Retrieve the number of items and users in the training set (`TrainingRatings.txt`) and the test set (`TestingRatings.txt`) respectively.

(b) Try to estimate memory needs by computing the expected (i.e., average) overlap of users in test and train and items in test and train respectively.

(c) Use those statistics to find out which way of implementing the collaborative filtering approach is best for this dataset/evaluation task.

(d) If you are using normalized ratings you should implement and run the pre-processing job. Use the following file names: `perp_job_mapper.java`, `perp_job_reducer.java` , `perp_job_driver.java`.

Write a brief report (`report.pdf`) describing your findings and the approach you want to implement. Especially, discuss (i.e., justify) any deviations from your initial plans established in milestone 1.
Submit your implementations by adding a folder called `milestone2` to the `final_project/netflix` folder in your svn repository. Add all scripts or java classes to this folder. **Do NOT add any data!**

Add the new files/folders to your SVN repo before committing:

```
$ svn add milestone2
$ svn add milestone2/report.pdf
$ svn add milestone2/your_files
$ svn commit -m 'milestone 2 submission' .
```

Problem 3: Collaborative Filtering Implementation

Now, you are ready to implement and run your approach. The goal of this final project is to predict as many of the ratings for the 100,000 user-movie pairs in the `TestingRatings.txt` file as possible. The true ratings are given in that file, however, you are only allowed to use them for evaluation purpose!
Detailed implementation requirements/specifications:
Using the 3.25 million ratings provided in the `TrainingRatings.txt` file as given utility matrix implement a collaborative filtering algorithm in MapReduce . You do not need to use MapReduce for the predictions, only to find the k-most similar users (or items). **This is a competition!** Part of the grades for the results (10%) will be assigned according to a ranking of the number and quality of your predictions!

(a) Implement your approach.

(b) Use it to predict the ratings for the (user, item) pairs in `TestingRatings.txt`.

(c) Compute the **Mean Absolute Error** (as defined here: `https://en.wikipedia.org/wiki/Mean_absolute_error`) and the **Root Mean Squared Error** (as defined here: `https://en.wikipedia.org/wiki/Root-mean-square_deviation`) for your predictions.

(d) Add yourself as a new user to the data set. To do this, you will need to create a new, unique user ID for yourself. Select some movies that you have seen among those in the training set, and add your ratings for those. Use your approach to output predictions for the movies you haven't rated, and rank those in decreasing order of the rankings. Do you agree with the predictions of your system? Include a description of this experiment in the project report (cf. next part). Then, check out some of the top ranked movies that you haven't seen (but only **after** you have finished your work on the project!).

(e) Write the project report documenting and discussing your collaborative filtering approach, your implementation, and the obtained evaluation results (number of predictions computed and average error measures). This report should be readable for an informed outsider and it should not require the reader to look at or run any code.

Final Submission Instructions

Submit your report including **documentation**, as well as, **results** as `project_report.pdf` by adding it to the `final_project/netflix` folder in your SVN repository. Submit your implementation by adding your implementations to the `final_project/netflix/src` folder in your SVN repository. **Do NOT add any data!**

Add the new files/folders to your SVN repo before committing:

```
$ svn add src/*
$ svn add project_report.pdf
$ svn commit -m 'final project submission' .
```

Copyrights

Problems are adapted and data is taken from Pedro Domingos' class on Data Mining/Machine Learning at University of Washington, 2012.