School of Information Technologies
Dr. Ying Zhou & A/P Uwe Roehm

COMP5349: Cloud Computing **1.Sem./2016**

Assignment 2: Spark Programming

Group Work: 20% **05.05.2016**

Introduction

This assignment requires you to write Spark programs to analyze the large movie rating data set from Grouplens `http://grouplens.org/datasets/movielens/`. You can use basic Spark programming API or machine learning API. You will need to demo your solution and submit a report.

Analysis Workloads Description

The analysis consists of two workloads.

In the **first** workload, you are asked to find out a few simple statistics from the large movie data set stored under `/share/movie`. We are interested to find out the top users in each genre. We rank users in each genre by the number of movies they rated in that genre. For instance, the top one user of a particular genre is the user who has rated most movies in that genre.

You are asked to find out the top 5 users of each genre and their rating statistics with respect to the genre and to the whole data set. Suppose user u is one of the top 5 users of genre G, you need to find out:

- Total number of movies u rated in G
- Total number of movies u rated in the data set
- Average rating of u in G
- Average rating of u in the dataset

In the **second** workload, you are asked to write a simple neighborhood based collaborative filtering algorithm for personalized recommendation. The following are the steps involved in the simple algorithm:

- **Personal Rating:** Pick a set of 15 movies. You can choose the movies based on your own interest, or use the top movies from different genre. Give a rate to those movies.

- **Neighborhood Formation:** For all other movies in the data set, compute the weighted cosine similarity (as described below and also in [1])between those movie and the 15 movies you have rated.

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R_u})(R_{u,j} - \bar{R_u})}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R_u})^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R_u})^2}} \tag{1}$$

  Here U denotes the set of users that have co-rated movie i and j; $R_{u,i}$ denotes the rating a user u gives to movie i; $\bar{R_u}$ denotes the average rating of user u in the whole data set.

- **Prediction Generation:** For all other movies in the data set, find the 10 most similar movies out of the 15 movies you have rated. Compute the predicted rate using the weighted sum method as described below and also in [1].

$$P_{u,i} = \frac{\sum_{n \in N}(s_{i,n} * R_{u,n})}{\sum_{n \in N}(\mid s_{i,n} \mid)} \tag{2}$$

  Here $s_{i,n}$ refers to the similarity value between two movies i and n; N refers to the set of movie i's 10 most similar movies out the 15 that you have rated.

- **Recommendation Generation** Rank the predicted rate and recommend the top 50 movies.

Performance Report Requirements

The report should have a section for each workload to describe the design, result and performance of your program. For each workload, please include the following subsections:

- **Application Design**
  In this section, describe the overall design of your application. You should draw a or a few lineage graph and briefly describe the operations you have used.

- **Result**
  In this section, put your result here in a easy to read and understand way (not direct program output). For workload 2, you need to include the set of 15 movies and their ratings

- **Performance Analysis**
  In this section, describe the performance of your application. You should include numbers such as the overall execution time, number of jobs, stages and tasks generated and number of executors you have used to run your program. If you have re-partitioned your data for performance boost, describe that as well.

Deliverable

There are two deliverables: **source code** and brief **report**. Both are due on Week 12 Thursday 26th of May. Please submit the source code and a soft copy of performance report as a zip or tar file in ELearning. You need to demo your implementation on week 12. Please also submit a hard copy of your performance report together with signed cover sheet during week 12's demo.

References

[1] SARWAR, B., KARYPIS, G., KONSTAN, J., AND RIEDL, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (2001), ACM, pp. 285–295.