ENGG5108/CSCI5510 Big Data Analytics

Assignment 1

Due Date: 23:59 Oct.8, 2016

Submission Instruction:

For this assignment, please submit **electronic version only**. We don't accept hard copy. For the programming questions, you need to submit BOTH **your codes and your results**. Submit codes as zipped tar file and the output of your program in plain-text file. For other questions, answer them in a word document. You should place the relevant files in their separate directory (preferable A, B, C... for each part).

Then compress all your files as one zip named using your student id, e.g., 10xxxxxxxx.zip, and submit to our course account at engg5108@cse.cuhk.edu.hk with email title "**ENGG5108 Asg#1, Your name, Your student ID**".

Part A. Hadoop Programming

The following problem is based on lecture 2.

In this problem, you need to write a MapReduce program to find the top-N pairs of similar users from a repository called Movielens. Provided is the dataset of **Movielens 20M**, which can be found in http://grouplens.org/datasets/movielens/20m/. The original range of the rating scores is in [0.5,5.0]. For simplicity, **we ignore the missing values in the ratings**. You need to calculate the top-N pairs of similar users based on the ratings. In order to calculate the similarity, we redefine the rating score in [0.5,2.5] as `unlike', and the rating score in (2.5,5.0] as `like'. Thus you can preprocess the dataset as (user, movie, `like') or (user, movie, `unlike'). Given a user i, you can construct a set of movies with `like' (denoted by $\mathrm{L_i}$), and also a set of movies with `unlike' (denoted by $\mathrm{U_i}$). Then, for a pair of users of (i,j), you can calculate the similarity between them via the following metric as

$$Jaccard = \frac{(L_i \cap \mathrm{L_j}) \cup (U_i \cap U_j)}{(L_i \cup \mathrm{L_j}) \cup (U_i \cup U_j)}.$$

You output the top-N pairs of similar users based on the values of Jaccard. (Hint: You can leave the final **sorting step** to a UNIX utility called "sort". Your MapReduce program needs only to calculate the metric of Jaccard.)

Please write a map reduce program (one mapper and one reducer) to list the **top-100 pairs of similar users with similarity (from most similarity to least similarity)**. Each line should have a pair of users' ID and the similarity. A valid example is as follows:

"2" "3" 0.75

For this problem, please submit all your codes as a zipped tar file and name the result files as **A.txt** under the directory **A**.

Part B. Frequent Itemsets

The following problem is based on lecture 2.

Suppose there are 9999 baskets, numbered 2 to 10000. Basket b contains all the prime factors of b. In this case, prime numbers are considered as items. Thus, basket 2 has item 2, basket 6 has item 2 and item 3, and so on. Basket 60 consists of items {2, 3, 5}.

Write the A-Priori algorithm with support threshold 50 to list all the frequent itemsets whose size is at least 3. Each line has a frequent itemset in ascending order, separated by a space. Itemsets with larger size should always be listed after itemsets with smaller size. If two itemsets are of the same size, they should be listed in ascending order as well. For example, the following outputs are in valid format:

```
2 3 5

2 3 7

2 3 5 7

2 3 7 11

2 3 5 7 13
```

For this problem, please submit all your codes as a zipped tar file and name the result files as **B.txt** under the directory **B**.

Part C. Locality Sensitive Hashing

The following problem is based on lecture 3.

Please answer this question **in a word file**.

This problem is related to the concept about shingling, minhashing, and locality sensitive hashing. Suppose you are given the following five sentences:

I. acbcbd
II. adbcad
III. cdbdad
IV. bcacbd
V. bdadca

a. Calculate the set of 2-shingles for each sentence and use matrix to represent the sentences, where the element is enumerated from 0.

b. Compute the minhash signature for each column if we use the following three hash functions: $h_1(x) = 7x + 1$ mod 10; $h_2(x) = 8x + 2$ mod 10; $h_3(x) = 6x + 2$ mod 10.

c. Which of these hash functions are true permutations?

d. How close are the estimated Jaccard similarities for the ten pairs of columns to the true Jaccard similarities?