IERG 4330/ESTR 4316/IEMS 5730 Spring 2022
Homework 2

Release date: Feb 20, 2022

Due date: Mar 8, 2022 (Tuesday) 11:59:00 pm

We will discuss the solution soon after the deadline. No late homework will be accepted!

Every Student MUST include the following statement, together with his/her signature in the submitted homework.

I declare that the assignment submitted on Elearning system is original except for source material explicitly acknowledged, and that the same or related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website http://www.cuhk.edu.hk/policy/academichonesty/.

Signed (Student_____) Date:_____

Name_____ SID_____

Submission notice:

- Submit your homework via the elearning system.
- All students are required to submit this assignment.

General homework policies:

A student may discuss the problems with others. However, the work a student turns in must be created COMPLETELY by oneself ALONE. A student may not share ANY written work or pictures, nor may one copy answers from any source other than one's own brain.

Each student **MUST LIST** on the homework paper the **name of every person he/she has discussed or worked with**. If the answer includes content from any other source, the student **MUST STATE THE SOURCE**. Failure to do so is cheating and will result in sanctions. Copying answers from someone else is cheating even if one lists their name(s) on the homework.

If there is information you need to solve a problem but the information is not stated in the problem, try to find the data somewhere. If you cannot find it, state what data you need, make a reasonable estimate of its value, and justify any assumptions you make. You will be graded not only on whether your answer is correct, but also on whether you have done an intelligent analysis.

Q1 [30 marks + 5 Bonus marks]: Basic Operations of Pig

You are required to perform some simple analysis using Pig on the n-grams dataset of Google books. An 'n-gram' is a phrase with n words. The dataset lists all n-grams present in books from books.google.com along with some statistics.

In this question, you only use the Google books bigram (1-grams). Please go to Reference [1] and [2] to download the two datasets. Each line in these two files has the following format (TAB separated):

bigram	**year**	**match_count**	**volume_count**

An example for 1-grams would be:

circumvallate	1978	335	91
circumvallate	1979	261	95

This means that in 1978(1979), the word "circumvallate" occurred 335(261) times overall, from 91(95) distinct books.

(a) **[Bonus 5 marks]** Install Pig in your Hadoop cluster. You can reuse your Hadoop cluster in IERG 4300/ IEMS 5730 HW#0 and refer to the following link to install Pig 0.17.0 over the master node of your Hadoop cluster :

http://pig.apache.org/docs/r0.17.0/start.html#Pig+Setup

Submit the screenshot(s) of your installation process.

If you choose not to do the bonus question in (a), you can use any well-installed Hadoop cluster, e.g., the IE DIC, or the Hadoop cluster provided by the Google Cloud/AWS [5, 6, 7] to complete the following parts of the question:

(b) **[5 marks]** Upload these two files to HDFS and **join** them into one table.

(c) **[10 marks]** For each unique bigram, compute its average number of occurrences per year. In the above example, the result is:

circumvallate (335 + 261) / 2 = 298

Notes: The denominator is the number of years in which that word has appeared. Assume the data set contains all the 1-grams in the last 100 years, and the above records are the only records for the word 'circumvallate'. Then the average value is:

(335 + 261) / 2 = 298,

instead of

(335 + 261) / 100 = 5.96

(d) **[15 marks]** Output the **20** bigrams with the highest average number of occurrences per year along with their corresponding average values sorted in descending order. If multiple bigrams have the same average value, write down anyone you like (that is, break ties as you wish).

You need to write a Pig script to perform this task and save the output into HDFS.

Submit your output together with the Pig script in one SINGLE pdf file.

Hints:

- This problem is very similar to the word counting example shown in the lecture notes of Pig. You can use the code there and just make some minor changes to perform this task.

Q2 [30 marks + 5 bonus marks]: Basic Operations of Hive

In this question, you are asked to repeat Q1 using Hive and then compare the performance between Hive and Pig.

(a) **[Bonus 5 marks]** Install Hive on top of your own Hadoop cluster. You can reuse your Hadoop cluster in IERG 4300/ IEMS 5730 HW#0 and refer to the following link to install Hive 2.3.8 over the master node of your Hadoop cluster.
https://cwiki.apache.org/confluence/display/Hive/GettingStarted

Submit the screenshot(s) of your installation process.

If you choose not to do the bonus question in (a), you can use any well-installed Hadoop cluster, e.g., the IE DIC, or the Hadoop cluster provided by the Google Cloud/AWS [5, 6, 7].

(b) **[30 marks]** Write a Hive script to perform exactly the same task as that of Q1 with the same datasets stored in the HDFS. Rerun the Pig script in this cluster and compare the performance between Pig and Hive in terms of overall run-time and explain your observation.

Submit your output, explanation, and your Hive commands/ scripts in one SINGLE pdf file.

Hints:

- Hive will store its tables on HDFS and those locations needs to be bootstrapped:
  ```
  $ hdfs dfs -mkdir /tmp
  $ hdfs dfs -mkdir /user/hive/warehouse
  $ hdfs dfs -chmod g+w /tmp
  $ hdfs dfs -chmod g+w /user/hive/warehouse
  ```
- While working with the interactive shell (or otherwise), you should first test on a small subset of the data instead of the whole data set. Once your Hive commands/ scripts work as desired, you can then run them up on the complete data set.

Q3 [40 marks + 20 Bonus marks]: Similar Users Detection in the MovieLens Dataset using Pig

Similar user detection has drawn lots of attention in the machine learning field which is aimed at grouping users with similar interests, behaviours, actions, or general patterns. In this homework, you will implement a similar-users-detection algorithm for the online movie rating system. Basically, users who rate similar scores for the same movies may have common tastes or interests and be grouped as similar users.

To detect similar users, we need to calculate the similarity between each user pair. In this homework, the similarity between a given pair of users (e.g. A and B) is measured as **the total number of movies both A and B have watched** divided by **the total number of movies watched by either A or B**. The following is the formal definition of similarity: Let M(A) be the set of all the movies user A has watched. Then the similarity between user A and user B is defined as:

$$Similarity(A, B) = \frac{|M(A) \cap M(B)|}{|M(A) \cup M(B)|} \quad \text{.....(**)}$$

where |S| means the cardinality of set S.
(Note: if $|M(A) \cup M(B)| = 0$, we set the similarity to be 0.)

The following figure illustrates the idea:

UserId	MovieId
A	a
A	b
A	c
B	b
B	c
B	d
C	b
C	d
D	b
D	c

Fig(a): The format of the data file.

	A	B	C	D
A		2	1	2
B	2		2	2
C	1	2		1
D	2	2	1	

Fig(b): The total number of movies both users in the pair have watched.

	A	B	C	D
A		4	4	3
B	4		3	3
C	4	3		3
D	3	3	3	

Fig(c): The total number of movies watched by either one of the users in the pair.

	A	B	C	D
A		0.5	0.25	0.66
B	0.5		0.66	0.66
C	0.25	0.66		0.33
D	0.66	0.66	0.33	

Fig(d): Pairwise similarity

Two datasets [3][4] with different sizes are provided by MovieLens. Each user is represented by its unique userID and each movie is represented by its unique movieID. The format of the data set is as follows:

<userID>, <movieID>

Write a program in Pig to detect the TOP K similar users for each user. You can use the cluster you built for Q1 and Q2 or you can use the IE DIC or one provided by the Google Cloud/AWS [5, 6, 7].

(a) **[15 marks]** For each pair of users in the dataset [3] and [4], output the number of movies they have both watched.

For your homework submission, you need to submit i) the Pig script and ii) the list of the 10 pairs of users having the largest number of movies watched by both users in the pair within the corresponding dataset. The format of your answer should be as follows:

<userID A>, <userID B>, <the number of movie both A and B have watched> //top 1
…
<userID X>, <userID Y>, <the number of movie both X and Y have watched> //top 10

(b) **[25 marks]** By modifying/ extending part of your codes in part (a), find the Top-K (K=3) most similar users (as defined by Equation (**)) for every user in the datasets [3], [4]. If multiple users have the same similarity, you can just pick any three of them.

Hint:

1. **In part (b), to facilitate the computation of the similarity measure as defined in (**), you can use the inclusion-exclusion principle, i.e.** $|S \cup T| = |S| + |T| - |S \cap T|$

In your submission, you only need to submit your Pig script together with the following output:

i) the user in [3] whose ID shares the same last 2 digits of your CUHK student ID number. For example, if your CUHK student ID number is ****13, you need to output the Top-3 similar user list for User 013, 113, 213, 313, …, 513 (if such users exist in the small dataset [3]), i.e.

013, <similar userID 1>, …, <similar userID K>
113, <similar userID 1>, …, <similar userID K>
...
513, <similar userID 1>, …, <similar userID K>

ii) Similarly, for the dataset in [4], only output the Top-3 similar user list for the users whose ID shares the same last 4 digits of your CUHK student ID number.

(c) **[Mandatory for ESTR4316] [20 Bonus marks for others]** Repeat part (b) using Hive.

Reference:

[1] Google Books 1:
http://storage.googleapis.com/books/ngrams/books/googlebooks-eng-all-1gram-20120701-a.gz
[2] Google Books 2:
http://storage.googleapis.com/books/ngrams/books/googlebooks-eng-all-1gram-20120701-b.gz
[3] Small scale dataset
https://mobitec.ie.cuhk.edu.hk/ierg4330/static_files/assignments/movielens_small.csv
[4] Large scale dataset
https://mobitec.ie.cuhk.edu.hk/ierg4330/static_files/assignments/movielens_large_updated.csv
[5] Cloud Dataproc API
https://cloud.google.com/dataproc/docs/reference/rest
[6] Amazon EMR
https://aws.amazon.com/emr/
[7] One-click Hadoop-Cluster deployment using Amazon EMR (Zoom Recording)
https://cuhk.zoom.us/rec/share/__ozUsXuqyHy3n5BA-MUm01bJedFK6IBfCqD6Qi2oCfcleGJ11rGTeqHi1_45hCO.xGoU3s3813Xn-_Q