IERG4300/ESTR4300/ENGG5108 Fall 2025
Homework 2

Release date: Oct 15, 2025
Due date: Nov 3, 2025 (Mon) 11:59 pm
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 the 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

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.

Question 0 [20 marks]: Frequent Itemsets

Consider running the **PCY** algorithm to count frequent item pairs on a dataset with **700 million** baskets. Suppose each basket contains ***n*** items and there are ***d*** distinct item pairs amongst all of the baskets. Consider the following setup during the first pass of PCY: after keeping the counters for every singleton itemset observed during the first pass, we can still afford to store in main memory **500 million** integers, each of which will be used as a bucket. Assume further that ***d*** is much larger than the total number of buckets available, *i.e.*, ***d*** >> **500 million**.

(a) [10 marks] What is the minimum support threshold ***s*** (in absolute number) we can allow if the average count for a bucket should be no more than 40% of the threshold ***s***? Please detail the steps to derive it.

(b) [10 marks] Suppose that A, B, C, D, E, and F are all the items under consideration. For a particular support threshold, the maximal frequent itemsets are {A, B, C} and {C, E, F}. What are all the other frequent itemsets?

Question 1 [80 marks + 20 Bonus marks]: Finding Frequent Itemsets

In this problem, we use a subset of the Yelp review dataset. The dataset[1] contains many user reviews, which are extracted from a website (Yelp) that publishes crowd-sourced reviews. The original dataset has been pre-processed as follows:

- Apply a sliding window with a length of 40 words on each review. All the words in a sliding window are collected to construct an individual basket.
- Remove duplicate words in one basket, and then filter out some common words.
- You can download the pre-processed data from the following link: https://mobitec.ie.cuhk.edu.hk/ierg4300Fall2025/static_files/homework/review2025.zip
- Each line of the dataset is a space-separated list of words that corresponds to one basket.

For **Q1(a), (b)** and **(d)**, the threshold for a frequent pair is defined as *s*=0.01. The frequency of a pair (i, j) is defined as: *Occurrence of pair (i, j)* / *Total number of baskets*. If the number of frequent pairs is more than 30, please only submit the **Top 30** pairs (if any) in your report. Your results should include the frequent pairs and their corresponding count.

- You are allowed to use Linux command *sort* to post-process your results in all the following sub-questions.

[1] Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." In Advances in neural information processing systems, pp. 649-657. 2015.

(a) [20 marks] Implement the A-Priori algorithm to find frequent pairs on a single machine

Refer to the lecture slides (pages 30 and 31) for implementing the **A-Priori** algorithm. **You should not use MapReduce Framework for this sub-question.**

Note: You can run this job on one single AWS/ Google Cloud machine or your PC. Note that dicvmc4.ie.cuhk.edu.hk is only a client for our DIC cluster. Please do **NOT** run your program for this question on this machine.

(b) [30 marks] Implement the SON algorithm on MapReduce to find frequent pairs

Implement the **SON** algorithm under the MapReduce framework to find frequent pairs. **Note that your code should be scalable.** In other words, your final results should be consistent if you use different numbers of mappers/ reducers for each job. You need to implement two MapReduce jobs as follows:

- The first MapReduce job should use **A-priori** algorithm to find the candidate pairs, which are frequent in at least one input file.
- The second MapReduce job counts only the candidate frequent pairs.

Tips:

- In the second MapReduce job, each mapper will load all the candidate pairs. You can pass them as a supplementary file.
- The total number of baskets/ records of the dataset can be used as a prior for the second MapReduce job.

Performance comparison requirements:

- (Optional) Wrap the two MapReduce rounds as a single executable by putting those commands you typed in a shell script (*.sh).
- Compare the overall execution time of (a) and (b).
- Report the commands you used to submit Hadoop jobs.

Note: You may use the IE Data-Intensive Cluster (DIC) or any other Hadoop cluster on a cloud platform of your choice (e.g., the AWS/Google Cloud cluster built in HW#0) to complete this problem.

(c) [30 marks] Implement SON on MapReduce to find frequent triplets

The threshold for frequent triplets is defined as *s*=0.005. The frequency of a triplet (i, j, k) is defined as: *Occurrence of triplet (i, j, k)* / *Total number of baskets*. You need to implement the SON algorithm to find all frequent triplets and then sort the frequent triplets by their count in descending order. In your report, you need to show some specific frequent triplets (as well as their counts) that fulfil the following requirements:

1. The count of the frequent triplet ranks **Top 50** among all frequent triplets.

2. The last digit of the rank of the frequent triplet is the same as the last digit of your SID. For example, if your SID is 1155xxxxx2, then you need to find the frequent triplets ranked the 2nd, 12th, 22nd, 32nd, etc.

Tip:

- In case of memory error, you may need to use multiple (*e.g.* 20+) mappers/ reducers.

(d) [20 Bonus marks] Use the PCY algorithm to filter the candidate pairs in the SON algorithm

Implement the **SON** algorithm under the MapReduce framework. Use the **PCY** algorithm to filter the candidate pairs in the first MapReduce job. You can use the following hash function.

HashFunction=hash(word_1 + word_2) mod 100000

For example, the result of the word pair (‘Monday’, ‘Tuesday’) can be implemented as follows in Python:

HashFunction=hash(‘Monday’ + ‘Tuesday’) % 100000

Performance comparison requirements:

- (Optional) Wrap the two MapReduce rounds as a single executable by putting those commands you typed in a shell script (*.sh).
- Compare the overall execution time of (a), (b), and (d).
- Report the commands you used to submit Hadoop jobs.

Part (d) is an optional (bonus) part for IERG4300 but is required for ESTR4300 and ENGG5108.

Submission requirement:

- You need to submit **BOTH** your code and your homework report.
- Please embed the relevant codes (with comments) and the results into your report **PDF** file.
- You should package all your codes into a single zip file individually and submit it together with the PDF file to the Blackboard system. Please do **NOT** package your report PDF file into your code zip file.
- Please include the signed statement (the first page of the homework file) in your homework report. You should also fill in your name and student ID in the statement.