CS 470 Homework 2

Due by Thursday, February 20, 2020 at 8:00 PM

Submission instructions

Submit your assignment through the QTest system, using course ID: **CS470** and exam ID: **hw2**. Upload a single ZIP archive file named **hw2.zip**, containing all the files of your solution:

1. The program's source files.
2. A `README.txt` file explaining how to compile and run your program.
3. A file named `result500.txt` which is your solution for the dataset `T10I4D100K.txt` with minimum support count 500.
4. The report in PDF format.
5. The LaTeX source files used to typeset the report.

No email submissions are accepted. No late submissions are accepted.

At the top of your solution, include a section named "Collaboration statement" in which you acknowledge any collaboration, help, or resource you used or consulted to complete this assignment.

1 Apriori Algorithm (100 points)

Implement the Apriori algorithm for frequent itemsets mining, either in Java or Python. The algorithm's pseudocode and related procedures are in the textbook. You are encouraged to use existing or your own optimization techniques to speed up the algorithm. Explain and discuss the techniques you used, and provide the appropriate references.

Your program should be executable with 3 parameters: the name of the input dataset file, the threshold of minimum support count, and the name of the output file (in that order). The minimum support count is an integer. An itemset is frequent if its support count is greater than or equal to this threshold. You can assume all items are represented by integers.

Your program should output a file that contains all the frequent itemsets together with their support counts. The output file should have the following format: each line contains a single frequent itemset as a list of items separated by a single space. Its support count is included between a pair of parentheses

at the end of the line. For example: 1 2 3 (5) represents an itemset containing items 1, 2, and 3 with a support count of 5. An example of output is provided in the file `example-output.txt`.

Test your implementation on the provided dataset `T10I4D100K.txt`. Measure its execution time as well as number of frequent itemsets with various minimum support values. The test dataset is a synthetic dataset that contains 100,000 transactions with an average size of 10 items from a set of 1000 distinct items. You can also try your program with other datasets, such as those at http://fimi.uantwerpen.be/data/.

Write a report in LaTeX presenting your results on the test dataset and other datasets that you have tried. Explain and discuss, if any, the algorithmic optimizations you have used in your implementation. Discuss the experiences and lessons you have learned from this assignment.

Please start early and be warned that an implementation without careful planning or efficient data structures could run for days! Excessively slow implementations will receive zero points (see grading criteria section). There are a few online repositories for frequent pattern mining implementations, such as http://fimi.uantwerpen.be/src/. You can study them but you are asked not to copy their implementations for this assignment. Remember that the Honor Code applies, and an automatic plagiarism checker will be used on submissions.

Grading criteria

- 70 points for correct and complete implementation. -10 for minor mistakes; -20 if there is some mistake but the program still works in most cases; -30 for more serious mistakes but the program still works in several cases. Zero points if the program does not compile, or if the program compiles but gives mostly the wrong results or crashes. Zero points if the program takes longer than 15 minutes on dataset T10I4D100K with minimum support count 500, on a CPU equal or faster than Intel Core i7-6700 at 3.40GHz.
- 30 points for a complete, clear, and well organized report. Zero points if the report is not typeset using LaTeX, or if the LaTeX source code is not provided.
- -10 points for insufficient comments in the code.
- -10 points for each deviation from the submission instructions.
- -10 points for missing collaboration statement.