# **Project 1**

**Name: Caleb Thian Jia Le 田家樂**

**Student ID: NN6114035**

1. Experiment on dataset 2022-DM-release-testdata-2.txt ,let

low min\_sup = 0.05

low min\_conf = 0.1

high min\_sup = 0.1

high min\_conf = 0.5

Record the time used by apriori and FP-tree under different combination of parameters mentioned, with the number of rules found respectively.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Time(s) | | Number of Rules Found | |
| min\_sup | min\_conf | Apriori | FP-tree | Apriori | FP-tree |
| Low | Low | 18.36 | 59.21 | 10287 | 1799 |
| Low | High | 18.31 | 59.35 | 3295 | 611 |
| High | Low | 0.14 | 58.90 | 2 | 2 |
| High | High | 0.14 | 59.11 | 2 | 2 |
| (Extra) 0.05 | 0.1 | 147.06 | 59.42 | 123246 | 7942 |
| (Extra) 0.05 | 0.5 | 150.49 | 59.70 | 24960 | 2662 |

Observation:

* Increasing min\_sup reduces the time used of Apriori, because more frequent patterns can be pruned in process, thus increase the speed for pattern generation.
* Time used of FP-tree does not be reduces because min\_sup and min\_conf does not reduces the candidate patterns generated, unlike Apriori
* But at the same time, number of rules found decreased in both algorithm as the threshold of support for rules is increased.
* Increasing min\_conf, with low min\_sup, will reduces the number of rules found
* Extra data will be discussed with the next dataset.

1. Experiment on dataset data.txt, which is generated by IBM Generator, let

low min\_sup = 0.0025

low min\_conf = 0.0025

high min\_sup = 0.003

high min\_conf = 0.7

Record the time used by apriori and FP-tree under different combination of parameters mentioned, with the number of rules found respectively

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Time(s) | | Number of Rules Found | |
| min\_sup | min\_conf | Apriori | FP-tree | Apriori | FP-tree |
| Low | Low | 51.69 | 5.32 | 232 | 192 |
| Low | High | 51.85 | 5.37 | 217 | 175 |
| High | Low | 9.77 | 5.34 | 22 | 2 |
| High | High | 9.8 | 5.36 | 22 | 2 |

Observation:

* Similar as observations mentioned.
* The only different is the time-used for FP-tree is less than the time-used for Apriori, this should because to the difference of datasets, and suitable min\_sup.
* min\_sup directly affect the number of calculation need to be done of these two algorithms. If min\_sup is unsuitable, Apriori may faster than FP-tree

1. Experiment on Kaggle dataset kaggle.txt, which is preprocess by utils.init\_kaggle() on the dataset basket\_analysis.csv[[1]](#footnote-1), let

low min\_sup = 0.1

low min\_conf = 0.1

high min\_sup = 0.2

high min\_conf = 0.5

Record the time used by apriori and FP-tree under different combination of parameters mentioned, with the number of rules found respectively

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Time(s) | | Number of Rules Found | |
| min\_sup | min\_conf | Apriori | FP-tree | Apriori | FP-tree |
| Low | Low | 0.74 | 1.79 | 438 | 252 |
| Low | High | 0.75 | 1.81 | 97 | 68 |
| High | Low | 0.13 | 1.79 | 12 | 54 |
| High | High | 0.13 | 1.77 | 3 | 47 |

Observation:

* Similar as above observation
* The only things different from others is that FP-tree found more rules than Apriori when high min\_sup. It maybe because of the pruning strategies is different between these two algorithms, making the counting is slightly difference.
* Different dataset may have different range for min\_sup and min\_conf, i.e., the maximum for these two parameters that making these algorithms cannot discover any rules from the dataset, can be different, thus low and high definition can also be varying sharply.

1. https://www.kaggle.com/datasets/ahmtcnbs/datasets-for-appiori [↑](#footnote-ref-1)