

# Hw1 report

Using data : lit -fname MYDATA -ntrans 5 -tlen 15 -randseed -87 -conf 0.3 -nitems 0.05

With min\_sup=0.2 min\_conf=0.2

Spending time:

Apriori: 13.0088s

FP\_growth: 0.8802s

With min\_sup=0.05 min\_conf=0.2

Spending time:

Apriori: 1312.8553s

FP\_growth: 6.1876s

With min\_sup=0.2 min\_conf=0.05

Spending time:

Apriori: 13.0460s

FP\_growth: 0.9663s

With min\_sup=0.05 min\_conf=0.05

Spending time:

Apriori: 1263.2598s

FP\_growth: 6.4607s

```
write_file Done in 0.00 seconds
PS C:\Users\User\OneDrive\桌面\hw1_example_2023> python3 main.py --dataset=myData.data --min_conf=0.2, --min_sup=0.05
2023-10-16 23:29:16,434 [main.py:467] INFO: Arguments: Namespace(dataset='myData.data', min_conf=0.2, min_sup=0.05)
read_file Done in 0.05 seconds
apriori spends:1312.8553s
write_file Done in 0.49 seconds
fp_growth spends:6.1876s
write_file Done in 0.49 seconds
PS C:\Users\User\OneDrive\桌面\hw1_example_2023> python3 main.py --dataset=myData.data --min_conf=0.05, --min_sup=0.2
2023-10-16 23:53:18,699 [main.py:467] INFO: Arguments: Namespace(dataset='myData.data', min_conf=0.05, min_sup=0.2)
read_file Done in 0.05 seconds
apriori spends:13.0460s
write_file Done in 0.00 seconds
fp_growth spends:0.9663s
write_file Done in 0.01 seconds
PS C:\Users\User\OneDrive\桌面\hw1_example_2023> python3 main.py --dataset=myData.data --min_conf=0.05, --min_sup=0.05
2023-10-16 23:54:53,223 [main.py:467] INFO: Arguments: Namespace(dataset='myData.data', min_conf=0.05, min_sup=0.05)
read_file Done in 0.05 seconds
apriori spends:1263.2598s
write_file Done in 0.80 seconds
fp_growth spends:6.4607s
write_file Done in 0.73 seconds
```

經由這 4 組的驗證我發現 min\_sup, min\_conf 越小生成出來的 rule 數就越多，但 min\_sup, min\_conf 下降到同樣機率，min\_sup 變小生出來的 rule 數明顯比 min\_conf 還要多

再來這 4 組的執行時間我發現 min\_sup 越小執行時間越長，但 min\_conf 的大小好像沒有多大的影響，我猜測是因為資料集的資料都比較相關，因此做出來的條件機率都普遍筆較高才沒有什麼影響，另外 min\_conf 只有在 generate rule 時會判斷 confident，但演算法最花時間的地方在於如何生成各個組合的 freq Set 所以 min\_conf 只影響要不要寫入 CSV 裡面而以所以影響不大，但 min\_sup 與單元素 candidate 高度相關，而單元素 candidate 的各數會直接影響後續各個長度 freqSet 的組合生成，所以 min\_sup 越小執行時間越長