**Reports of Data Mining Project 1**

What do you observe in the below 4 scenarios? What could be the reason? Include the Runtime statistics, # of frequent patterns, and # of results under 4 scenarios.

在這項實驗中，關於 low boundary與high boundary我在sup分別設置成0.05與0.2，而在conf分別設置成0.1與0.7，並為了凸顯出不同設置的差異，選擇以apriori algorithm去進行實驗。以下表格為實驗結果：

|  |  |  |  |
| --- | --- | --- | --- |
| Apriori | Runtime(s) | # of Freq\_patterns | # of results |
| Low sup, Low conf | 56.88 | 5354 | 57193 |
| Low sup, High conf | 57.84 | 5354 | 2589 |
| High sup, Low conf | 1.37 | 117 | 326 |
| High sup, High conf | 1.43 | 117 | 42 |

可以看到在runtime方面High sup遠快於Low sup，因為min\_sup的閥值高所以能篩選掉許多不符合的pattern，使得後續遞迴的情況減少，而min\_conf的調整沒有什麼影響到runtime，這是因為conf的計算需要先有freq\_patterns才能做運算，而真正影響到計算時間的正是找freq\_patterns的過程，而此過程只有與min\_sup相關，可以看到freq\_patterns數目不會因conf不同而不同。

conf所影響的就是後續根據freq\_patterns計算關聯法則時，才會在這裡篩掉低於min\_conf的results，所以相比low conf，high conf的results數目會大幅減少。

|  |  |
| --- | --- |
| FP\_growth | Runtime(s) |
| Low sup, Low conf | 1.92 |
| Low sup, High conf | 1.72 |
| High sup, Low conf | 0.41 |
| High sup, High conf | 0.42 |

FP\_growth明顯減少時間複雜度的問題，但若min\_sup調得更高，可能導致apriori因為簡單架構反而較快，而FP\_growth因基礎架構複雜因而較慢。

Bonus: kaggle-groceries

可以明顯感受到apriori與fp\_growth兩者時間差異性，而關於min\_sup, min\_conf的設置，到了(0.001, 0.2)還只有兩筆results，能了解若要運用於實際場合可能需要設置較小min\_sup, min\_conf或是更大的資料集才能有效截取出結果。