Data Mining 2018

HW1 Frequent Itemset Mining

姓名：吳睿哲

學號 :r06921095

日期 ： 2018/10/21

(1)

Apriori

Apriori 我主要是依照上課所教的 horizontal set 去實行,並沒有用到特殊的 hash table candidate set pruning.主要的流程可以分成三個步驟:

1.讀取資料 ： 將資料整理成 horizontal set 的架構,並用 dictionary 去存,key 是每筆交易編號,value 是商品名稱。

2.創建 Candidate： 創建 candidate 其實就是把上一輪的 frequent itemset 拿來做排列組合。

3.形成 Frequent Itemset： 這一步主要就是把創建處來 candidate 小於 minimum support 的部分 prune 掉,並用之繼續建立 candidate,直到沒有辦法找到更多的 candidate set 為止。

提升 performance 的部分其實就是靠第三個步驟,把小於 minimum support 的 frequent itemset 的 candidate 都 prune 掉,減少排列組合運算。

Eclat

Eclat 的部分主要是把資料的呈現方式改成講義上的 vertical TIDSET data presentation,並且用 bit vector 的方式呈現,程式架構我一樣是用 dictionary 去存,只是 value 的部分改成存 numpy array,numpy array 裡面存的是 binary 的值,也就是此人是否購買商品,此演算法透過求頻繁 k 項集的交集來獲取 k+1 項集,我是用遞迴的方式來實現。

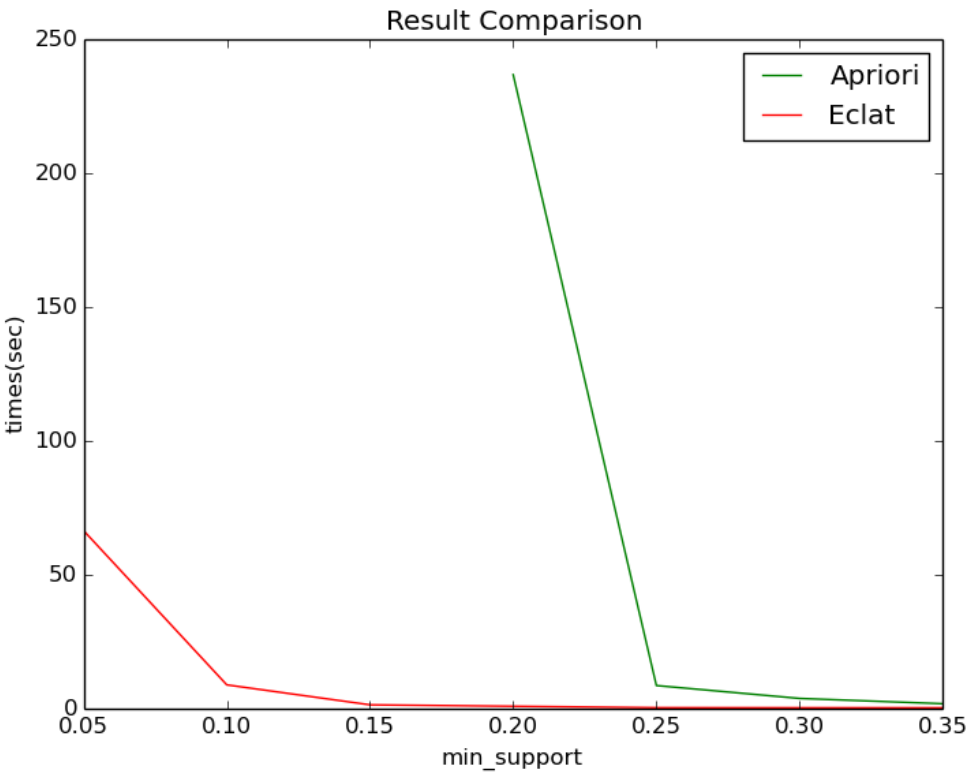
提升 performance 的部分除了 Eclat 演算法本身就比 Apriori 快以外,主要是 bitvector 在做 binary operation 的時候很快速,而且 numpy 也有支援 binary array 的 binary operation,藉此提高速度。

觀察結果

由第二小題的圖可以看出,apriori 的 performance 比 eclat 差很多,因為他需要多次掃描資料庫,所以需要耗費大量時間,elcat 雖然看似在 minimum support 很大的時候運算速度相當快,但是由於遞迴的關係,會耗費大量資源在做交集運算,當數據量很龐大時可能會造成系統負擔。

(2)

Plot


Result Comparison
Apriori
Eclat
times(sec)
min_support
250
200
150
100
50
0
0.05
0.10
0.15
0.20
0.25
0.30
0.35