資料探勘

專案作業四

組員:黃凱祥、吳俊園、賀鈞嗣、吳鈞達

2023年1月

摘要

此次實驗是針對大量的交易資料進行關聯規則的分析，前處理中預先剔除退貨以及註銷的交易(數量為零或負值)，對於不同的資料分別設定信心度及支持度。實驗中使用了Apriori演算法以及FP-Growth演算法，通過修改不同的參數設定，紀錄規則數量和執行時間並進行比對。

關鍵字:python、Apriori、FP-Growth

1. 緒論

1.1動機

透過對課堂上所教授的Apriori和FP-Growth演算法，使用作業提供的交易資料集，以了解學習的成果和累積實作之經驗。

1.2目的

在交易資料集中進行挖掘資料關聯性和頻繁項，主要觀看使用Apriori和FP-Growth演算法兩個執行的速度，以及支持度、信心度和最小支持度多寡對資料關聯規則分析影響。

1. 資料集

2.1資料集

2.1.1 交易資料集

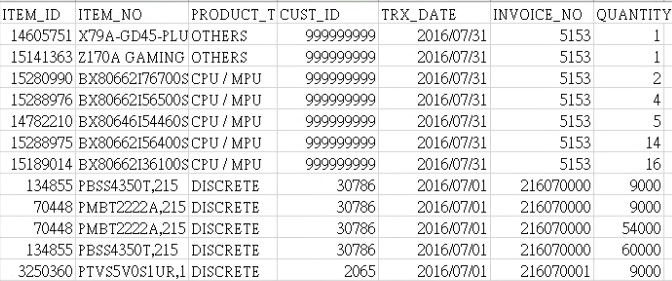


Figure 1交易部分資料

1. 資料筆數：128838。
2. 資料屬性欄位數：7。
3. 資料型態：顧客交易資料。
4. 缺失值：無。

**其屬性的資訊為**

1. ITEM\_ID : 商品編號。
2. ITEM\_NO：商品條碼。
3. PRODUCT\_TYPE：產品類型。
4. CUST\_ID：顧客編號。
5. TRX\_DATE：交易日期。
6. INVOICE\_NO: 發票編號
7. QUANTITY : 購買數量
8. 方法

3.1前置處理

在交易資料集中，將購買數量為零或是負值的交易剔除。

3.2實驗設計

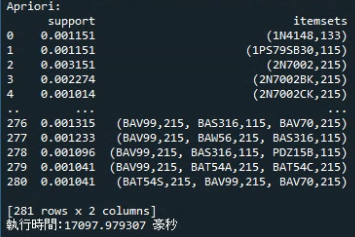
由於scikit-learn沒有包含內建交易資料集，必須讀檔，且設定欄位，將交易資料集以發票順序做排序invoice-sorted，排序完成後再從程式碼中相同發票編號放在同一個陣列作為一比交易，分別使用Apriori、FP-Growth演算法，使用Apriori會得到每個item或多個item組合的支持度找出frequentitemsets，調整min\_support參數，也就是最小支持度，即在所有交易中支持度大於最小支持度，才可

符合frequenset的要求，並且使用time函式個別得出執行速度，其中FP-Growth跟上述Apriori一樣的方法。

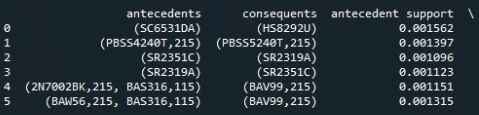
交易資料集中，商品種類太多種，min\_support值要設到低於0.008才會有頻繁項集。

3.3實驗結果

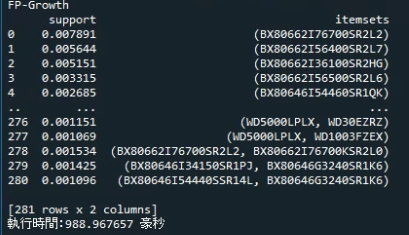
首先將交易資料集中使apriori演算法，調整min\_support至0.001，產生大於min\_support的frequent itemset和支持度和執行速度。



再產生商品的關聯規則



最後使用FP-Growth演算法，調整min\_support至0.001，產生大於min\_support的frequent itemset和支持度和執行速度。



1. 結論

在交易資料集中，需要預處理資料，將數量為零或負值的剔除，且要把發票順序排序，將相同的放在同一個陣列，才能開始關聯規則分析，並且商品的種類太多必須將min\_support值低於0.008才會產生frequent item，最小支持度越大，發現產生的frequent item越少，各item的支持度也會降低，最後在使用time函數，發現FP-Growth比Apriori執行速度更快，因為FP-Growth指掃描兩次。

1. 參考文獻

[1] *Apriori Algorithm(Mlxtend library)* https://artsdatascience.wordpress.com/2019/12/10/apriori-algorithm/