

國立雲林科技大學資訊管理系

機器學習-作業四

Department of Information Management

National Yunlin University of Science & Technology

Assignment

交易資料集關聯分析

Transaction dataset Association rule analysis

楊欣蓓、陳怡君、鄭皓名、陳郁云

指導老師：許中川 博士

Advisor: Chung-Chian Hsu, Ph.D.

中華民國113年1月

January 2024

摘要

眾多商家欲提高商場銷售業績，需找出商品之間的關聯規則，並調整商家銷售商品方式。因此本研究選擇了此交易資料集，欲從中發掘更多潛在關聯規則。此外 Apriori、FP-Growth 演算法的關聯規則建立方式都不同，會因為輸入的物品種類多寡，而影響其分析效率，故本研究欲分析兩種演算法對此交易資料集處理效率。本研究目的為透過交易資料集進行 Apriori、FP-Growth 演算法的關聯規則分析，透過調整支持度 (support) 和信心度 (confidence) 可分別過濾掉非高頻和關聯規則較弱的商品組合，藉此來比較參數影響關聯規則的效率高低、關聯規則的數量變化。在實驗環節，需先進行資料前處理，再將資料型態轉換成模型所需，輸入至 Apriori、FP-Growth 演算法觀察關聯規則變化和效率結果，從實驗結果發現當 support 為0.004時，confidence 的變化也不再影響關聯規則數量，表示關聯規則已收斂、冗餘規則皆已剔除。最後將上述商品推薦結果輸出成 csv 檔，設計出一套商品推薦功能。

關鍵字：數據分析、關聯分析、Apriori、FP-Growth

一、緒論

1.1 研究動機

於現今世代，大多商家希望透過給予顧客更好的購物體驗，獲得更多的銷售業績，為此需去了解顧客消費習慣，從大量歷史消費紀錄中，找出不同商品之間可能存在的關係，此稱為關聯規則，再調整商家銷售商品方式。因此本研究為探討關聯規則，選擇了此交易資料集，此外從資料內容可猜測其來自特定產業領域，對於分析目標具有實際意義的價值，例如：在電子元件等領域，進行關聯規則分析，可以了解元件之間的購買模式或相互關係。上述可助於該領域商場制定出更有效的營銷策略、庫存管理或供應鏈優化。

此次實驗主要是用 Apriori、FP-Growth 演算法進行交易資料關聯規則分析，其中 Apriori 演算法在處理大型資料集時可能效率較低，而 FP-Growth 演算法在這方面通常表現較佳，研究欲比較兩者之間所需花費的時間，來分析出哪種方法更為適用於分析目標，在進行演算法之前需要仔細評估這些因素，以確保分析的有效性和實用性。

1.2 研究目的

本研究目的想以 Apriori、FP-Growth 演算法對交易資料做關聯規則分析，以了解顧客消費習慣，進而分析顧客消費行為，並透過關聯規則分析推薦熱門商品組合。本次實驗使用 Apriori 及 FP-Growth 演算法進行關聯分析，並比較兩種演算法所需花費時間。在執行實驗過程中，進行了資料前處理、對模型設定了不同支持度和信心度，發現不易察覺的交易模式，此模式可能有助於了解顧客的購買行為及偏好，藉由關聯規則分析也能提高顧客的滿意程度並增加再次購買此商品的可能性，並間接了解商品之間的關聯性。

二、實驗方法

2.1 實作說明

本次實驗使用 Apriori 演算法和 FP-Growth 演算法對交易資料進行關聯分析。由於交易數量為零和負值代表該項商品退貨或註銷，因此在資料前處理時，將其交易紀錄刪除，並把重複與空值的交易資料一同剔除、將相同'INVOICE_NO'內容合併在一起，視為同一筆交易紀錄、把交易紀錄轉換型態成模型所需。進行 Apriori、FP-Growth 演算法分析時，調整支持度(support)過濾規則數量，並進行關聯規則分析調整信心度(confidence)觀察規則的前後關係，此外撰寫了關聯規則的輸出、讀入功能以便觀察，最後藉由前述的關聯規則設計出一套商品推薦功能。

2.2 操作說明

本研究執行環境皆採用 Python3.10.10，以 Visual Studio Code 作為開發工具，利用 Apriori 演算法和 FP-Growth 演算法進行關聯分析，並使用 Pandas、Numpy、Mlxtend 等函式庫來讀取資料、分析關聯規則。於資料前處理，利用 Pandas 套件功能，刪除交易數量為零或負的資料、刪除重複交易資料、合併相同'INVOICE_NO'成交易紀錄，利用 TransactionEncoder 轉換交易紀錄內容成模型所需的型態。

三、實驗設計

3.1 資料集

名稱: 交易資料集

原始資料筆數: 157396 筆

資料前處理後資料筆數: 111961 筆

表1

交易資料集欄位介紹

欄位	屬性	內容
0	INVOICE_NO	nominal
1	CUST_ID	nominal
2	ITEM_ID	nominal
3	ITEM_NO	nominal
4	PRODUCT_TYPE	nominal
5	TRX_DATE	interval
6	QUANTITY	ratio

3.2 資料前處理

3.2.1 交易資料集

- 資料前處理
 - 交易數量為零和負值代表商品退貨或註銷，因此將其交易紀錄刪除。
 - 刪除資料內容空白的項目。
 - 刪除資料內容重複的項目。
 - 以交易資料集的 'INVOICE_NO' 為基準，將同一筆交易資料的 'PRODUCT_TYPE' 分於同一列，其內容統整於 OrderList 資料表中。
 - 將交易資料轉換成模型所需的資料型態。

表2

部分資料處理後的交易資料集

特徵 資料	INVOICE_NO	CUST_ID	ITEM_ID	ITEM_NO	PRODUCT_TYPE	TRX_DATE	QUANTITY
No.0	CX47348203	3218	3217532	M25P40-VMN6TPB	MEMORY_EMBEDDED	2016/7/26	2500
No.1	CX47346522	2470	3326781	AU80610006237AASLBX9	CPU / MPU	2016/7/11	50
No.2	CX47348534	16135	740487	MMBD2837LT1G	DISCRETE	2016/7/27	3000
⋮							
No. 135171	216072965	2717205	14427725	NCP45540IMNTWG-H	OTHERS	2016/7/13	3000
No. 136157	216072965	2717205	1433827	IHLP4040DZERR36M01	PEMCO	2016/7/13	500
No. 140702	216072965	2717205	14563956	W25Q16CLSNIGT	MEMORY_EMBEDDED	2016/7/13	2500

表3

部分 OrderList 資料內容

INVOICE	PRODUCTTYPE
CX47348203	MEMORY_EMBEDDED
CX47346522	CPU / MPU
CX47348534	DISCRETE
216072965	OTHERS、PEMCO、MEMORY_EMBEDDED

表4

部分型態轉換後的交易紀錄

INVOICE _NO	CHIPSE T / ASP	CPU / MPU	DISCRE TE	LINEAR IC	LOGIC IC	MEMOR Y_EMBE DED	MEMOR Y_SYST EM	OPTICA LAND SENSOR	OTHERS	PEMCO
CX47348203	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
CX47346522	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CX47348534	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
216072965	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE

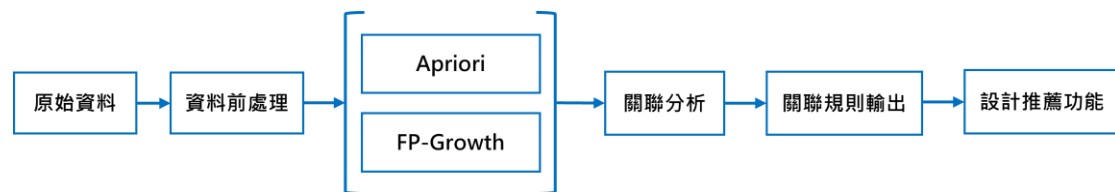
3.3 實驗設計

3.3.1 交易資料集

本研究將交易資料集做資料前處理，包括刪除重複資料、刪除交易數量為零和負值的資料、將相同'INVOICE_NO'的'PRODUCT_TYPE'內容合併在一起視為一筆交易紀錄、把整理好的交易紀錄利用 TransactionEncoder 轉換成演算法所需的型態，進行關聯分析前，計算出關聯規則數量為57002筆，因規則數量龐大會導致分析效率不佳，而使用 Apriori 演算法和 FP-Growth 演算法去調整支持度門檻，剔除掉非高頻的規則，以提升關聯分析效率，接著進行關聯分析，調整信心度去觀察關聯規則之間的差異，透過觀察可發現信心度調整至一定程度時，規則數量則不再被影響，就可將整理好的關聯規則輸出成csv檔以便觀察，藉由前述分析結果，設計出一套推薦商品功能，可供使用者輸入欲購買的商品名稱，並推薦可能會感興趣的商品給使用者，詳細實驗流程說明，如圖1所示。

圖 1

交易資料集實驗設計流程圖



3.4 實驗結果

本研究實驗分為兩部份，第一部份針對交易資料集使用 Apriori 與 FP-Growth 兩種演算法做關聯分析，並透過不同的超參數組合，觀察兩演算法間關聯分析的結果與執行時間差異；第二部份導出關聯分析的內容，利用其資訊設計出一組功能，供使用者輸入欲購買的商品名稱並推薦可能感興趣的其他商品。

3.4.1 交易資料集關聯分析

本實驗針對交易資料集使用 Apriori 與 FP-Growth 兩種演算法做關聯分析，並透過不同超參數組合來觀察分析結果與執行時間的差異。本實驗將 support 變動範圍設定為 0.001到 0.008，每次增量 0.001；confident 變動範圍設定為 0.010 到 0.015，每次增量0.002，其中 Apriori 演算法的超參數設定及其各組合表現如表5，FP-Growth演算法的超參數設定及其各組合表現如表6。透過上述兩張表可發現當 support 與 confident 提升，關聯分析組合數遞減。

透過比較兩種關聯分析演算法在不同超參數組合，可以發現當 support 值達到0.004時，關聯規則數量已經不會因為 confident 提升而減少，因此本研究認為當 support=0.004時，已將其多數冗餘規則剔除，導致 confident 的調整不影響關聯規則的數量增減，最後也將其組合結果作為推薦功能設計的資料來源。

另外，由於 FP-growth 演算法會先將資料集儲存在 FP 樹，相較 Apriori 減少了對資料集掃描次數，所以遇到大型資料集時，FP-growth 演算法的效率會比 Apriori 演算法更高。針對實驗結果比較兩種關聯分析演算法的執行時間時，本研究發現 FP-growth 演算法搜尋時間確實比 Apriori 稍顯快速，如圖2所示。但部份幾個情況下無明顯有效率，本研究推論可能是因為交易資料集數量較少，因此，此特性在實驗中並沒有特別顯著的效果。

圖 2

FP-growth 演算法與 Apriori 演算法執行時間比較

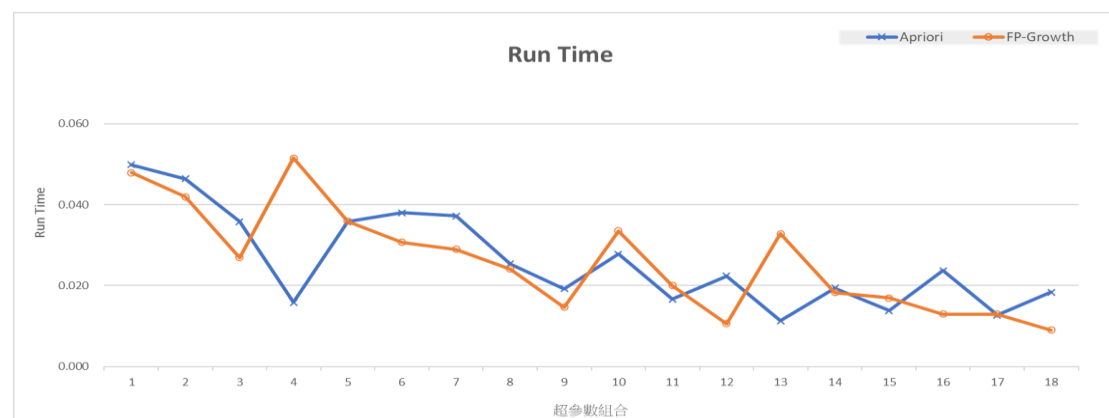


表5

Apriori 各超參數組合分析表現

超參數組合			分析結果	
set	support	confident	run time	association rules
1	0.001	0.010	0.050	201
2		0.012	0.046	190
3		0.014	0.036	182
4	0.002	0.010	0.016	135
5		0.012	0.036	130
6		0.014	0.038	124
7	0.003	0.010	0.037	72
8		0.012	0.025	71
9		0.014	0.019	65
10	0.004	0.010	0.028	36
11		0.012	0.017	36
12		0.014	0.022	36
13	0.005	0.010	0.011	30
14		0.012	0.019	30
15		0.014	0.014	30
16	0.006	0.010	0.024	16
17		0.012	0.013	16
18		0.014	0.018	16
19	0.007	0.010	0.012	14
20		0.012	0.017	14
21		0.014	0.003	14
22	0.008	0.010	0.014	12
23		0.012	0.021	12
24		0.014	0.015	12

表6

FP-Growth 各超參數組合分析表現

超參數組合			分析結果	
set	support	confident	run time	association rules
1	0.001	0.010	0.048	201
2		0.012	0.042	190
3		0.014	0.027	182
4	0.002	0.010	0.052	135
5		0.012	0.036	130
6		0.014	0.031	124
7	0.003	0.010	0.029	72
8		0.012	0.024	71
9		0.014	0.015	65
10	0.004	0.010	0.034	36
11		0.012	0.020	36
12		0.014	0.011	36
13	0.005	0.010	0.033	30
14		0.012	0.018	30
15		0.014	0.017	30
16	0.006	0.010	0.013	16
17		0.012	0.013	16
18		0.014	0.009	16
19	0.007	0.010	0.019	14
20		0.012	0.012	14
21		0.014	0.012	14
22	0.008	0.010	0.008	12
23		0.012	0.022	12
24		0.014	0.014	12

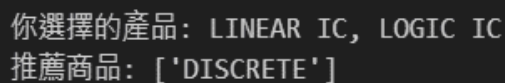
3.4.2 商品推薦功能設計

本實驗使用 $\text{support}=0.004$, $\text{confident}=0.010$ 的關聯分析結果來製作商品推薦功能。首先對導入資料做前處理，將其沒意義的空白欄先刪除後，將 'antecedents' 與 'consequents' 兩欄資料型態轉換成 string，以方便接下來資料搜尋及輸出處理。

接著設計出一個輸入單元，方便使用者寫下想要購買的商品（多項內容則使用”，”隔開）。本研究透過抓取使用者的輸入品項，和資料集中的 'antecedents' 比對並輸出對應的 'consequents' 欄內容作為推薦商品。以 LINEAR IC 與 LOGIC IC 為例，當資料格輸入 LINEAR IC, LOGIC IC 後，下方即會顯示「推薦商品: ['DISCRETE']」，如圖3；若輸入內容沒有於找不到對應的資訊，則會輸出告知找不到相關商品，以 LINEAR IC 與 PEMCO 為例，當資料格輸入 LINEAR IC, PEMCO 後，下方即會顯示「找不到與您選擇有相關的產品！」，如圖4。

圖3

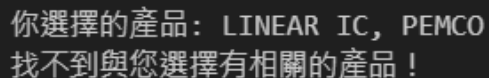
LINEAR IC 與 LOGIC IC 推薦商品的輸出結果



你選擇的產品: LINEAR IC, LOGIC IC
推薦商品: ['DISCRETE']

圖4

LINEAR IC 與 PEMCO 推薦商品的輸出結果



你選擇的產品: LINEAR IC, PEMCO
找不到與您選擇有相關的產品！

四、結論

在交易資料集中，Apriori 與 FP-Growth 兩種演算法做關聯分析並比較不同 support 與 confident 組合下關聯規則的數量多寡。實驗過程發現當 support=0.004 時，調整 confident 也不影響關聯規則數量，分析結果已收斂到一定程度，冗餘關聯規則已經剔除。此外 FP-Growth 演算法雖然比 Apriori 演算法效率高，但並非相當顯著，本研究推論可能是交易資料集數量並非相當龐大所致。

於第二部份的實驗中，本研究也透過關聯分析的結果設計出一組推薦使用者可能感興趣商品的功能。透過使用者輸入欲購買商品資料與關聯規則的 antecedents 比對，輸出相應的 consequents 作為推薦商品；若使用者輸入的多項商品中無對應內容，則發出無推薦內容的通知。

參考文獻

Dario Radečić (2021)。如何使用 TensorFlow 優化學習率——比你想像的要容易。

[How to Optimize Learning Rate with TensorFlow — It's Easier Than You Think | by Dario Radečić | Towards Data Science](#)

Ryan Lu (2018)。Preprocessing Data：類別型特徵_OneHotEncoder &LabelEncoder 介紹與實作。

[https://medium.com/ai%E5%8F%8D%E6%96%97%E5%9F%8E/preprocessing-data-onehotencoder-labelencoder-%E5%AF%A6%E4%BD%9C-968936124d59](#)