

國立雲林科技大學資訊管理系

資料探勘-作業四

Department of Information Management

National Yunlin University of Science & Technology

Assignment

交易資料集分析

Transaction Data Set Analysis

張博勝、巫宇哲、陳冠融、翁振洋

指導老師：許中川 博士

Advisor: Chung-Chian Hsu, Ph.D.

中華民國 112 年 1 月

October 2022

## 摘要

在電腦科學以及資料探勘領域中，Apriori 演算法是「關聯規則學習」或是「關聯分析 (Associative Analysis)」的經典演算法之一，目的是在一個資料集當中，找出不同項與項之間可能存在的關係。而在行銷資料科學領域，它有個很特別的名字，被稱為「購物籃分析 (Market Basket analysis)」，也跟課堂上提過的啤酒與尿布的故事有關。關聯分析主要透過「支持度」(Support) 與「信心度」(Confidence) 來對商品項目之間的關聯性，進行篩選。其中，支持度 (Support) 意指即某項目集在資料庫中出現的次數比例。例如：某資料庫中有 100 筆交易紀錄，其中有 20 筆交易有購買啤酒，則啤酒的支持度為 20%。信賴度 (Confidence) 意指兩個項目集之間的條件機率，也就是在 A 出現的情況下，B 出現的機率值。FP-growth 演算法通過構建 FP-tree 來壓縮資料庫中的資訊，從而更加有效地產生頻繁項目集。FP-tree 其實是一棵字首樹，按支援度降序排列，支援度越高的頻繁項離根節點越近，從而使得更多的頻繁項可以共享字首。FP-growth 也是一種經典的頻繁項集和關聯規則的挖掘算法，在較大數據集上 Apriori 需要花費大量的運算開銷，而 FP-growth 卻不會有這個問題。因為 FP-growth 只掃描整個數據庫兩次。

關鍵字：關聯分析(Apriori)、支持度 (Support)、信心度 (Confidence)、FP-growth

# 一、緒論

## 1.1 動機

關聯分析(Apriori)應用的範圍相當廣，也是非常受到歡迎的分析方法，這邊就舉幾個社會上的使用案例：1. 淘寶推薦相關書籍 2. 百度文庫推薦相關文件 3. Walmart 尿布與啤酒 4. 推薦醫療器具組合。

關聯分析的優勢是數據中只需要有關聯數據即可，其它屬性資料是用不到的，關聯分析相對的也比較容易編碼。本研究主要使用的評估指標有支持度(Support)與信心度(Confidence)，藉由這兩評估指標來判斷關聯分析結果的好壞。

## 1.2 目的

本研究主要藉由交易資料集來探討顧客的購物項目，利用關聯分析(Apriori)的技術來進行研究，使用關聯分析的技術可以找尋資料間彼此的關聯，它是透過兩種主要的方式來進行分析：頻繁項集、關聯規則，分析過後可能得到意想不到的結果，就像是經典的 Walmart 尿布與啤酒的故事。這兩樣八桿子打不著關係的商品放在一起，竟然可以增加營業額。

## 二、 資料集

### 2.1 交易資料集

名稱: 交易資料集

原始資料筆數: 157396

前處理後的資料筆數: 128837

表 1 交易資料集欄位介紹

欄位	屬性	內容
0	ITEM_ID	number
1	ITEM_NO	str
2	PRODUCT_TYPE	'MEMORY_EMBEDDED', 'CPU / MPU', 'DISCRETE', 'LINEAR IC', 'OPTICAL AND SENSOR', 'CHIPSET / ASP', 'LOGIC IC', 'MEMORY SYSTEM', 'PEMCO', 'OTHERS'
3	CUST_ID	number
4	TRX_DATE	date
5	INVOICE_NO	str
6	QUANTITY	number

表 2 顯示部分交易資料集

ITEM_ID	ITEM_NO	PRODUCT_TYPE	CUST_ID	TRX_DATE	INVOICE_NO	QUANTITY
3217532	M25P40-VMN6TPB	MEMORY_EMBEDDED	3218	2016/7/26	CX47348203	2500
3326781	AU80610006237AASLBX9	CPU / MPU	2470	2016/7/11	CX47346522	50
740487	MMBD2837LT1G	DISCRETE	16135	2016/7/27	CX47348534	3000
3434776	IHLP1616ABER2R2M11	PEMCO	999999999	2016/7/29	A20160700174	0
3217532	M25P40-VMN6TPB	MEMORY_EMBEDDED	3218	2016/7/26	CX47348203	2500

## 三、 方法

### 3.1 實作說明

在關聯分析模型的實驗中，本研究首先將交易資料集做前處理，將數量為零或負值的交易從資料集中剔除，使用兩個不同的關聯分析模型進行執行時間的比較，也嘗試設定不同的支持度及信心度，記錄規則數量及執行時間之變化，過程中並剔除冗餘規則，以減少規則數量。最後撰寫允許輸入數項產品，利用關聯規則推薦於規則右手邊的產品之功能。

### 3.2 操作說明

本研究執行環境皆為 Python3.9.13，並使用 Visual Studio Code 作為使用工具，利用 Pandas、Preprocessing 以及 Numpy 來讀取資料以及進行資料前處理，關聯分析則利用 Apriori 演算法和 FP-Growth 演算法進行運算，比較兩者間的執行時間。

## 四、實驗

### 4.1.前處理

#### 4.1.1. 交易資料集

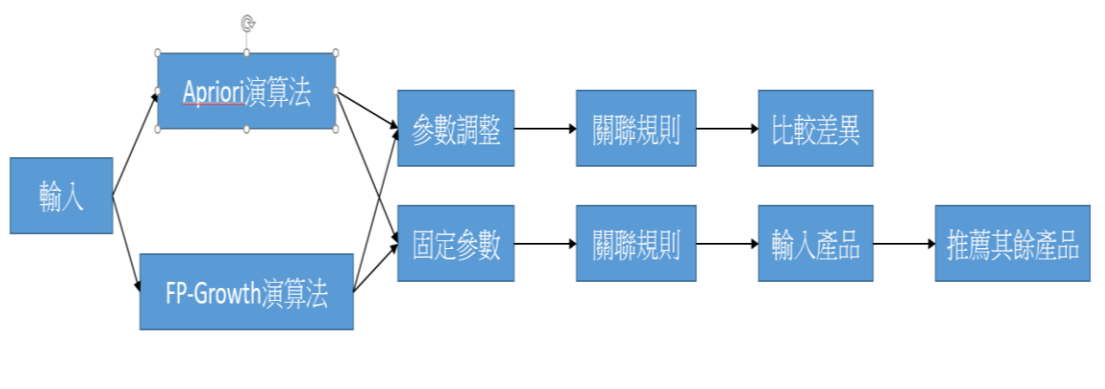
- 資料前處理：將資料欄位數量(QUANTITY)數值為小於或等於零的值進行筆數刪除，因為它代表其交易為退貨或註銷。
- 資料萃取：由於這次要做的是關聯規則分析，因此不需要用到所有資料，本研究只取出發票編號(INVOICE\_NO)和產品類別(PRODUCT\_TYPE)兩個欄位。
- 資料整理：本研究將同一個發票編號所產生相關的產品類別都歸類為同一筆交易紀錄，由於有好幾筆不同的發票編號，因此最後結果將會產生一個二維陣列，如表 3 為部分資料處理後的交易資料集。

表 3.顯示部分資料處理後交易資料集

發票編號	產品類別
CX47348203	'MEMORY EMBEDDED'
CX47346522	'CPU / MPU'
CX47348534	'DISCRETE'
CX47346184	'DISCRETE'
CX47347899	'LINEAR IC', 'LOGIC IC', 'OTHERS'
CX47346191	'OPTICAL AND SENSOR'
.	.
.	.
.	.
CX47348656	'LOGIC IC', 'LINEAR IC', 'DISCRETE', 'OPTICAL AND SENSOR'

### 4.2.實驗設計

圖 1.實驗設計流程圖



本研究實驗設計如上圖，由上章節所提及之前處理資料作為輸入，放入不同的演算法進行關聯分析，其中調整演算法中支持度和信心度的最小門檻，比較不同參數的規則數量，以及針對不同演算法的計算時間進行比較。另外，藉由固定參數而產生的關聯規則，來進行推薦關聯產品之功能。

### 4.3.交易資料集實驗結果

#### 4.3.1 Apriori 演算法與 FP-Growth 演算法

將我們整理好的資料丟進 Apriori 演算法，並把支持度(support)和信心度(confidence)做參數的測試，得出表 4 為參數測試結果，包含支持度、信心度，運行時間和透過關聯規則所產生出來的相關數量。

表 4.顯示 Apriori 演算法參數測試之結果

support	confidence	Spend time	count
0.001	0.008	0.059197	216
0.001	0.009	0.051166	207
0.001	0.010	0.046158	201
0.001	0.011	0.050170	197
0.002	0.008	0.047154	144
0.002	0.009	0.042138	137
0.002	0.010	0.041138	135
0.002	0.011	0.040127	131
0.003	0.008	0.028097	72
0.003	0.009	0.029095	72
0.003	0.010	0.037121	72
0.003	0.011	0.035120	72
0.004	0.008	0.037121	36
0.004	0.009	0.028093	36
0.004	0.010	0.033110	36
0.004	0.011	0.033537	36
0.005	0.008	0.039129	30
0.005	0.009	0.037122	30
0.005	0.010	0.044147	30
0.005	0.011	0.040133	30

FP-Growth 演算法也如上述方法一樣，並把支持度(support)和信心度(confidence)所設定的參數與上面設定相同方便做比較，得出表 5 為參數測試結果。

表 5.顯示 FP-Growth 演算法參數測試之結果

support	confidence	Spend time	count
0.001	0.008	0.374762	216
0.001	0.009	0.449518	207
0.001	0.010	0.287305	201
0.001	0.011	0.232322	197
0.002	0.008	0.231698	144
0.002	0.009	0.241309	137

續下表



續上表

0.002	0.011	0.237790	131
0.002	0.010	0.245358	135
0.003	0.008	0.238786	72
0.003	0.009	0.255846	72
0.003	0.010	0.259858	72
0.003	0.011	0.250831	72
0.004	0.008	0.255858	36
0.004	0.009	0.220744	36
0.004	0.010	0.205183	36
0.004	0.011	0.219774	36
0.005	0.008	0.201663	30
0.005	0.009	0.207909	30
0.005	0.010	0.214710	30
0.005	0.011	0.207241	30

本研究發現在同個參數設置之下他所產生出來的關聯規則數量是一模一樣的，唯一差別只有在運行時間，在這資料集下，FP-Growth 演算法所花費的時間要比 Apriori 演算法要花的時間多不少。

#### 4.3.2 關聯規則產生

由上述兩種演算法比較可得知，他所得出來的關聯規則會是一樣的，因此本文只列出其中一個演算法所跑出來之關聯規則結果，如表 6 為 Apriori 演算法跑出之關聯規則，支持度(support)設定為 0.004，信心度(confidence)設定為 0.01。

表 6.顯示 Apriori 演算法之關聯規則結果

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
(CPU / MPU)	(CHIPSET / ASP)	0.110746	0.044112	0.005343	0.048243	1.093658	0.000458	1.004341
(CHIPSET / ASP)	(CPU / MPU)	0.044112	0.110746	0.005343	0.121118	1.093658	0.000458	1.011802
(LINEAR IC)	(CPU / MPU)	0.260398	0.110746	0.007672	0.029461	0.266026	- 0.021166	0.916248
(CPU / MPU)	(LINEAR IC)	0.110746	0.260398	0.007672	0.069273	0.266026	- 0.021166	0.794650
(OTHERS)	(CPU / MPU)	0.113403	0.110746	0.005288	0.046630	0.421051	- 0.007271	0.932748
(CPU / MPU)	(LOGIC IC)	0.110746	0.121541	0.005206	0.047006	0.386754	- 0.008254	0.921789
(LOGIC IC)	(CPU / MPU)	0.121541	0.110746	0.005206	0.042831	0.386754	- 0.008254	0.929047
(CPU / MPU)	(OTHERS)	0.110746	0.113403	0.005288	0.047749	0.421051	- 0.007271	0.931053

續下表

續上表

(LINEAR IC)	(DISCRETE)	0.260398	0.257905	0.034906	0.134049	0.519761	- 0.032252	0.856971
(DISCRETE)	(LINEAR IC)	0.257905	0.260398	0.034906	0.135345	0.519761	- 0.032252	0.855372
(DISCRETE)	(LOGIC IC)	0.257905	0.121541	0.026988	0.104643	0.860965	- 0.004358	0.981127
(LOGIC IC)	(DISCRETE)	0.121541	0.257905	0.026988	0.222047	0.860965	- 0.004358	0.953908
(MEMORY_EMBEDDED)	(DISCRETE)	0.066716	0.257905	0.005562	0.083368	0.323250	- 0.011644	0.809589
(DISCRETE)	(MEMORY_EMBEDDED)	0.257905	0.066716	0.005562	0.021566	0.323250	- 0.011644	0.953855
(DISCRETE)	(OPTICAL AND SENSOR)	0.257905	0.039865	0.005562	0.021566	0.540971	- 0.004719	0.981297
(OPTICAL AND SENSOR)	(DISCRETE)	0.039865	0.257905	0.005562	0.139519	0.540971	- 0.004719	0.862419
(DISCRETE)	(OTHERS)	0.257905	0.113403	0.006658	0.025815	0.227642	- 0.022589	0.910091
(OTHERS)	(DISCRETE)	0.113403	0.257905	0.006658	0.058710	0.227642	- 0.022589	0.788381
(PEMCO)	(DISCRETE)	0.119486	0.257905	0.004877	0.040816	0.158261	- 0.025939	0.773674
(DISCRETE)	(PEMCO)	0.257905	0.119486	0.004877	0.018910	0.158261	- 0.025939	0.897485
(LINEAR IC)	(LOGIC IC)	0.260398	0.121541	0.043454	0.166877	1.373012	0.011805	1.054417
(LOGIC IC)	(LINEAR IC)	0.121541	0.260398	0.043454	0.357529	1.373012	0.011805	1.151184
(LINEAR IC)	(MEMORY_EMBEDDED)	0.260398	0.066716	0.005973	0.022938	0.343811	- 0.011400	0.955194
(MEMORY_EMBEDDED)	(LINEAR IC)	0.066716	0.260398	0.005973	0.089528	0.343811	- 0.011400	0.812328
(LINEAR IC)	(OTHERS)	0.260398	0.113403	0.005233	0.020097	0.177215	- 0.024297	0.904780
(OTHERS)	(LINEAR IC)	0.113403	0.260398	0.005233	0.046146	0.177215	- 0.024297	0.775383
(MEMORY_EMBEDDED)	(LOGIC IC)	0.066716	0.121541	0.004576	0.068583	0.564280	- 0.003533	0.943143
(LOGIC IC)	(MEMORY_EMBEDDED)	0.121541	0.066716	0.004576	0.037647	0.564280	- 0.003533	0.969793
(MEMORY_SYSTEM)	(OTHERS)	0.021234	0.113403	0.004110	0.193548	1.706724	0.001702	1.099380
(OTHERS)	(MEMORY_SYSTEM)	0.113403	0.021234	0.004110	0.036241	1.706724	0.001702	1.015571
(LINEAR IC, DISCRETE)	(LOGIC IC)	0.034906	0.121541	0.014713	0.421507	3.468026	0.010471	1.518530

續下表

續上表

(LINEAR IC, LOGIC IC)	(DISCRETE)	0.043454	0.257905	0.014713	0.338588	1.312841	0.003506	1.121986
(DISCRETE, LOGIC IC)	(LINEAR IC)	0.026988	0.260398	0.014713	0.545178	2.093634	0.007686	1.626134
(LINEAR IC)	(DISCRETE, LOGIC IC)	0.260398	0.026988	0.014713	0.056503	2.093634	0.007686	1.031282
(DISCRETE)	(LINEAR IC, LOGIC IC)	0.257905	0.043454	0.014713	0.057049	1.312841	0.003506	1.014417
(LOGIC IC)	(LINEAR IC, DISCRETE)	0.121541	0.034906	0.014713	0.121055	3.468026	0.010471	1.098014

### 4.3.3 推薦產品

本文利用了上述表 6 所得到之關聯規則結果匯出檔案，利於後面推薦產品使用，當使用者輸入一個或數個產品時系統可以自動的根據使用者輸入的產品透過關聯分析之後，推薦其可能有興趣的產品。下面將講述運行過程。

範例：使用者輸入單個產品

當使用者輸入 LOGIC IC 這項產品時系統會自動推薦其它產品，如圖 2 所示。

圖 2.顯示經由關聯規則後推薦之結果

```
你選擇的產品: LOGIC IC
你可能也有興趣的東西: ['CPU / MPU' 'DISCRETE' 'LINEAR IC' 'MEMORY_EMBEDED']
```

範例：使用者輸入多個產品

當使用者輸入 DISCRETE, LOGIC IC 多項產品時系統也會自動推薦其它產品，如圖 3 所示。

圖 3.顯示經由關聯規則後推薦之結果

```
你選擇的產品: DISCRETE, LOGIC IC
你可能也有興趣的東西: ['LINEAR IC']
```

範例：使用者輸入產品後，無法找到其它相關產品

當使用者輸入 DISCRETE, PEMCO 多個產品時，因為它們沒有在關聯規則裡面，因此會無法找到使用者輸入的相關產品，如圖 4 所示。

圖 4.顯示經由關聯規則後找不到推薦之結果

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

## 五、 結論

在交易資料集裡本研究發現由於每個發票編號所購買出來的產品類別都不盡相同，因此產生出來的高頻項目集就很少，在本實驗中支持度基本上要小於 0.05 才會有關聯產生。在運行時間上面，調升或調降支持度或信心度的值不會影響演算法所需花的時間，然而在 Apriori 演算法和 FP-Growth 演算法所花的時間相比，理論上 FP-Growth 應該要比 Apriori 還要快上很多，但本實驗結果得出 FP-Growth 卻是要比 Apriori 要慢上很多，所以經過上網查詢相關資料，本研究認為是因為交易資料集所產生的高頻資料集太少，也就是說每一筆交易紀錄都很零散重複性不高，導致 FP-Growth 在建造 tree 時所花費的時間太高了，才會讓運行時間輸給 Apriori。最後本研究也選擇了 Apriori 演算法，並做適當的參數調整來作為關聯規則，以便做後續產品推薦的功能。

## 參考文獻

行銷資料科學 (2019)。你怎麼處理顧客交易資訊？Apriori 演算法。

<https://reurl.cc/pZjYna>

阿新 (2019)。關聯分析（一）--FP-Growth 演算法。

<https://www.796t.com/content/1546888205.html>

索羅格 (2019)。從零實現機器學習算法（十四）FP-growth。

<https://zhuanlan.zhihu.com/p/67653006>

Chwang (2020)。Machine Learning-關聯分析-Apriori 演算法-詳細解說啤酒與尿布的背後原理 Python 實作-Scikit Learn 一步一步教學。

<https://reurl.cc/rZjEyN>

Icebns. (2020, April 15). 常用的關聯規則演算法 (Apriori 演算法、FP-Growth 演算法) 的優缺點. CSDN.

<https://blog.csdn.net/icebns/article/details/105535366>