國立雲林科技大學

資訊管理系碩士班

資料探勘

關聯規則分析交易資料集

M11123028 林永沂

M11123017 洪國書

指 導 教 師 ：許 中 川

中 華 民 國 112 年 1 月

摘要

隨著不斷進步的資訊科技發展，交易資料也時時刻刻的變動著。而對於企業來說，交易資料的重要性是不容忽視的，企業能夠透過分析交易資料來獲得豐富的資訊，例如顧客的消費習慣、商品銷售狀況、市場趨勢等。本研究使用貴實驗室提供之「交易資料集」，樣本屬性包括產品類型、客戶 ID、日期以及發票號碼等等。本研究透過對此交易資料集進行分析，比較了兩種關聯規則 Apriori 與 FP-Growth 的分析時間，利用關聯規則分析發現資料集中的頻繁項與樣本物品之間的關聯規則，利用這些規則能夠幫助我們了解顧客的購買行為，或是找出商品之間的替代關係，並為市場行銷與產品規劃提供有用資訊。

Keyword：交易資料、關聯規則、頻繁項集、Apriori、FP-Growth

壹、緒論

1.1 研究背景

交易，也就是買賣，是人們日常生活中不可或缺的一部分。從古時人們以「物品交換」方式進行交易，隨著時間的推移，人們發明了「貨幣交易」以達到更便利的交易模式，而後又隨著科技的進步，人們不斷的發明新的交易方式，例如電子商務。總結而言，交易方式正在不斷演進。

隨著不斷進步的資訊科技發展，交易資料也時時刻刻的變動著。而對於企業來說，交易資料的重要性是不容忽視的，企業能夠透過分析交易資料來獲得豐富的資訊，例如顧客的消費習慣、商品銷售狀況、市場趨勢等。這些資訊可以幫助企業制定更有效的營銷策略、提升產品設計、協助預估銷售量。

關聯規則，最早是由 Agrawal et al. (1993)所提出，是一種在大型資料庫中尋找關聯性的統計方法，通常被用於挖掘交易資料集中產品之關聯性，協助零售業者了解顧客間的消費行為，從資料集中得知的關聯性能夠用於支援銷售決策上。

本研究使用貴實驗室提供之「交易資料集」，樣本屬性包括產品類型、客戶ID、日期以及發票號碼等等。透過對交易資料集進行分析，將對於顧客經驗分析、產品推薦以及產品銷售預測能有優秀的效果。例如，企業能夠透過分析交易資料集來了解消費者最常購買的產品，並依此制定訂單與行銷策略。

1.2 研究目的

本研究針對貴實驗室提供之「交易資料集」蒐集資料，使用 Python 來分析該數據，使用了兩種經典的關聯規則分析之探勘方法，Apriori 以及 FP-Growth，並比較兩種方法分析所花費之時間。

貳、資料集

2.1 真實資料集

本次採用許中川教授提供的「交易資料集」。

	ITEM_ID	ITEM_NO	PRODUCT_TYPE	CUST_ID	TRX_DATE	INVOICE_NO	QUANTITY
0	3217532	M25P40-VMN6TPB	MEMORY_EMBEDED	3218	2016-07-26	CX47348203	2500
1	3326781	AU80610006237AASLBX9	CPU / MPU	2470	2016-07-11	CX47346522	50
2	740487	MMBD2837LT1G	DISCRETE	16135	2016-07-27	CX47348534	3000
3	3434776	IHLP1616ABER2R2M11	PEMCO	999999999	2016-07-29	A20160700174	0
4	70072	MMBT3906LT1G	DISCRETE	2356	2016-07-06	CX47346184	12000
5	3204503	PCA9555DWR	LOGIC IC	2506	2016-07-21	A10085337	0
6	3420352	TMP103AYFFR	LINEAR IC	10228	2016-07-25	CX47347899	3000
7	3311565	OV6922-V09N	OPTICAL AND SENSOR	38381	2016-07-06	CX47346191	1152
8	140887	SN74AHC1G126DCKR	LOGIC IC	999999999	2016-07-31	5119	0
9	3216410	SI2303CDS-T1-GE3	DISCRETE	27495	2016-07-11	CX47346636	3000
10	123164	TLZ24B-GS08	DISCRETE	30377	2016-07-28	CX47348680	2500

圖 1 交易資料表

資料集中總共有 7 個欄位、157,396 筆數、dtype 為 object，欄位名稱和形態請參考表 1。

欄位名稱	屬性
ITEM_ID	Int64
ITEM_NO	Object
PRODUCT_TYPE	Object
CUST_ID	Int64
TRX_DATE	Datetime64[ns]
INVOICE_NO	Object
QUANTITY	Int64

表 1 交易資料表屬性

參、方法

2.1 研究架構

本研究的研究架構如圖 2，將數據集的資料依序步驟進行統整、分析、決策樹建置、驗證、結論。並試著將交易資料集內的資料利用 Apriori 與 FP-Growth 將資料進行關聯規則分析，得出兩種方法花費時間之比較長條圖，以及支持度與信心度之繪製圖，得出的結果有助於市場行銷與產品規劃相關研究之日後研究參考。

。

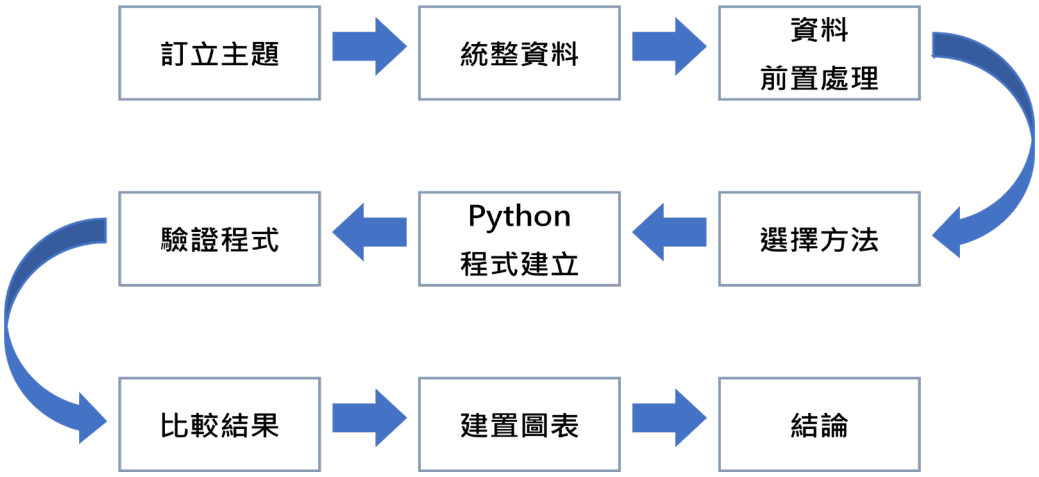

圖 2 研究架構

2.2 執行程式的方法

(1) 選擇資料集

選用貴實驗室提供之交易資料集，並將挑選出來的變數簡化為目標資料集。

(2) 前置處理

針對資料的完整性去除不符合規則或不一致的資料，例如「空值」與「負值」。

(3) 變換

將資料的輸入格式轉換成符合後續資料探勘步驟的格式。

(4) 群聚分析

資料經過轉換後，藉由 Apriori 以及 FP-Growth 演算法，來分析出資料集中的資料關聯，最後比較兩種方法所花費時間，並將其圖像化。

(5) 方法比較

透過以上方法分析後，經由統計及其他技術來確認其結果，設立衡量指標以利於比較兩種關聯規則分析之品質。

肆、實驗

3.1 前置處理

首先本實驗從 collections 導入 Counter。可用於計算列表或其它可迭代對象中項目的出現次數。

接下來使用 lambda 函數和 filter 函數創建一個新列表，該列表僅包含交易資料集中 QUANTITY 欄位大於 0。lambda 函數將元素 t 作為輸入，如果 t[‘數量'] > 0，否則為 False。filter 函數將此 lambda 函數應用於交易資料集中的每個元素，並返回一個新列表，該列表僅包含 lambda 函數返回 True 的元素。

最後使用 Boolean 創建一個新 Data frame，只包含交易資料集中 QUANTITY 字段大於 0 的行。這和上一段程式碼類似，但是語法不同，直接操作在數據框上而不是在列表上。總的來說，這三行程式碼都已經將數量為 0 或負值的欄位所移除。

3.2 實驗設計

首先本實驗導入了幾個函數，分別是 pandas、numpy、matplotlib、seaborn 和 mlxtend。pandas 是用來處理資料的模組，numpy 是用來處理數學運算的模組，matplotlib 和 seaborn 是用來繪圖的模組，mlxtend 是用來進行關聯分析的模組。其中，%matplotlib inline 是用來設定 matplotlib 圖表在 notebook 中顯示的指令。TransactionEncoder 是 mlxtend 模組中用來將資料轉換成頻繁項集分析所需要的格式的類別。本次實驗我們將利用 apriori、association_rules 和 fpgrowth 則是用來進行關聯分析。pd.set_option('display.max_columns', None)和 pd.set_option('display.float_format', lambda x: '%.3f' % x)則是設定 pandas 在顯示資料時的顯示格式。warnings.filterwarnings("ignore")和 warnings.simplefilter(action='ignore', category=FutureWarning)和 warnings.simplefilter(action='ignore', category=DeprecationWarning)則是用來忽略警告訊息的指令。

首先，使用 list(df.groupby(['CUST_ID', 'TRX_DATE'])) 將交易資料集按照 CUST_ID 和 TRX_DATE 分組，並以列表的形式存在 all_transactions 變數中。接下來，使用 transaction[1]['PRODUCT_TYPE'].tolist()取出每組的 PRODUCT_TYPE 欄位的值，並以列表的形式存在 all_transactions 變數中。最終，all_transactions 變數中存有每筆交易中購買的商品的列表。

接下來，使用 TransactionEncoder()建立一個 TransactionEncoder 的實例，並將 all_transactions 變數傳入 fit()函數，將資料轉換成頻繁項集分析所需要的格式。最

後，使用 transform()函數將轉換後的資料轉換成 pandas 的 DataFrame 並存在 trans_encoder_matrix 變數中。

再來本實驗定義了一個函數 perform_rule_calculation()，用於計算頻繁項集和關聯規則。perform_rule_calculation()函數有兩個參數：transact_items_matrix 和 rule_type，其中 transact_items_matrix 是轉換後的資料，rule_type 是要使用的頻繁項集分析的演算法，預設為"fpgrowth"。min_support 則是設定最小支持度，預設為 0.001。首先，使用 if 判斷是否使用 apriori 演算法，如果是則使用 apriori 函數計算頻繁項集，如果不是則使用 fpgrowth 函數計算頻繁項集。計算過程中，使用 min_support 參數設定最小支持度，使用 use_colnames=True 參數將項目名稱顯示在結果中。接下來，使用'number_of_items'欄位計算頻繁項集中項目的數量。最後，回傳頻繁項集和計算所用的時間。compute_association_rule()函數有兩個參數：rule_matrix 和 metric，其中 rule_matrix 是頻繁項集，metric 是要使用的評估指標，預設為"lift"。min_thresh 則是設定最小門檻值，預設為 1。使用 association_rules 函數計算關聯規則，並使用 metric 參數設定評估指標，使用 min_threshold 參數設定最小門檻值。最後，回傳計算出的關聯規則。接下來本實驗使用 numpy 的 polyfit()函數計算兩個變數之間的關係，並使用 numpy 的 poly1d()函數將計算結果轉換成函數。接下來，使用 matplotlib 的 plot()函數繪製圖表，並使用 xlabel()、ylabel()和 title()函數設定圖表的標題、x 軸和 y 軸的名稱。最後我們計算出在使用 Apriori、fpgrowth 演算法中所需的時間並使用 plot_metrics_relationship()函數繪製 Apriori、fp_growth 變數中的 lift 和 confidence 變數之間的關係圖。其中，使用 col1 參數設定 x 軸為 lift，col2 參數設定 y 軸為 confidence。

關聯規則輸出，首先本實驗定義了一個函數 save_rules()，用於將關聯規則儲存到指定的檔案中。save_rules()函數有兩個參數：rules 和 file_path。其中，rules 是要儲存的關聯規則，file_path 是儲存檔案的路徑。使用 with open()函數打開要儲存的檔案，並使用 for 迴圈遍歷關聯規則。然後，使用 try except 結構轉換每個交易的所有產品的類型，如果無法轉換則跳過。最後，將轉換後的交易追加到 transactions 變數中。接下來，使用 apyori 模組中的 apriori()函數計算關聯規則，並使用 min_support 參數設定最小支援度為 0.5，min_confidence 參數設定最小信心水準為 0.75。最後，使用 save_rules()函數將關聯規則儲存到 rules.txt 檔案中。

關聯規則讀取，首先本實驗定義了一個函數 load_rules()，用於從指定的檔案讀取關聯規則。load_rules()函數有一個參數：file_path，其中 file_path 是讀取檔案的路徑。使用 with open()函數打開要讀取的檔案。然後，使用 for 迴圈遍歷檔案中的每一行。接下來，使用 split()函數將每行的前件和後件分割成兩個部分。接下來，使用 split()

函數將前件和後件分割成個別的產品。最後，使用元組建立關聯規則並加入到 rules 變數中。最後，使用 return 傳回 rules 變數。

3.3 實驗結果

本實驗利用 Apriori、FP-Growth 探勘出交易資料集中的 PRODUCT_TYPE 的支持度與信心度，並且將探勘出的關聯規則輸出至指定的資料夾以及讀取已存檔之關聯規則，最後我們比較了 Apriori 與 FP-Growth 演算法所花費的時間，分別是 0.0416 秒與 0.0552 秒。以下將列出本次實驗得出之結果圖：

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(DISCRETE)	(MEMORY_EMBEDED)	0.286	0.101	0.040	0.139	1.370	0.011	1.043
1	(MEMORY_EMBEDED)	(DISCRETE)	0.101	0.286	0.040	0.392	1.370	0.011	1.174
2	(LOGIC IC)	(MEMORY_EMBEDED)	0.160	0.101	0.033	0.205	2.029	0.017	1.131
3	(MEMORY_EMBEDED)	(LOGIC IC)	0.101	0.160	0.033	0.324	2.029	0.017	1.244
4	(MEMORY_EMBEDED)	(LINEAR IC)	0.101	0.310	0.040	0.398	1.284	0.009	1.146
5	(LINEAR IC)	(MEMORY_EMBEDED)	0.310	0.101	0.040	0.130	1.284	0.009	1.033
6	(DISCRETE, MEMORY_EMBEDED)	(LINEAR IC)	0.040	0.310	0.029	0.720	2.322	0.016	2.462
7	(DISCRETE, LINEAR IC)	(MEMORY_EMBEDED)	0.131	0.101	0.029	0.218	2.152	0.015	1.149
8	(MEMORY_EMBEDED, LINEAR IC)	(DISCRETE)	0.040	0.286	0.029	0.709	2.478	0.017	2.454
9	(DISCRETE)	(MEMORY_EMBEDED, LINEAR IC)	0.286	0.040	0.029	0.100	2.478	0.017	1.066

圖 3 FP-Growth 信心度與支持度

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(CHIPSET / ASP)	(CPU / MPU)	0.074	0.184	0.021	0.282	1.530	0.007	1.136
1	(CHIPSET / ASP)	(DISCRETE)	0.074	0.286	0.016	0.218	0.763	-0.005	0.913
2	(CHIPSET / ASP)	(LINEAR IC)	0.074	0.310	0.019	0.254	0.819	-0.004	0.925
3	(DISCRETE)	(LINEAR IC)	0.286	0.310	0.131	0.458	1.479	0.042	1.274
4	(LINEAR IC)	(DISCRETE)	0.310	0.286	0.131	0.423	1.479	0.042	1.237
5	(DISCRETE)	(LOGIC IC)	0.286	0.160	0.091	0.317	1.984	0.045	1.231
6	(LOGIC IC)	(DISCRETE)	0.160	0.286	0.091	0.568	1.984	0.045	1.652
7	(MEMORY_EMBEDED)	(DISCRETE)	0.101	0.286	0.040	0.392	1.370	0.011	1.174
8	(MEMORY_SYSTEM)	(DISCRETE)	0.034	0.286	0.007	0.216	0.753	-0.002	0.910
9	(OPTICAL AND SENSOR)	(DISCRETE)	0.072	0.286	0.020	0.275	0.962	-0.001	0.985

圖 4 Apriori 信心度與支持度

圖 5 FP-Growth 之 Lift 與 Confidence 繪製圖

圖 6 Apriori 之 Lift 與 Confidence 繪製圖

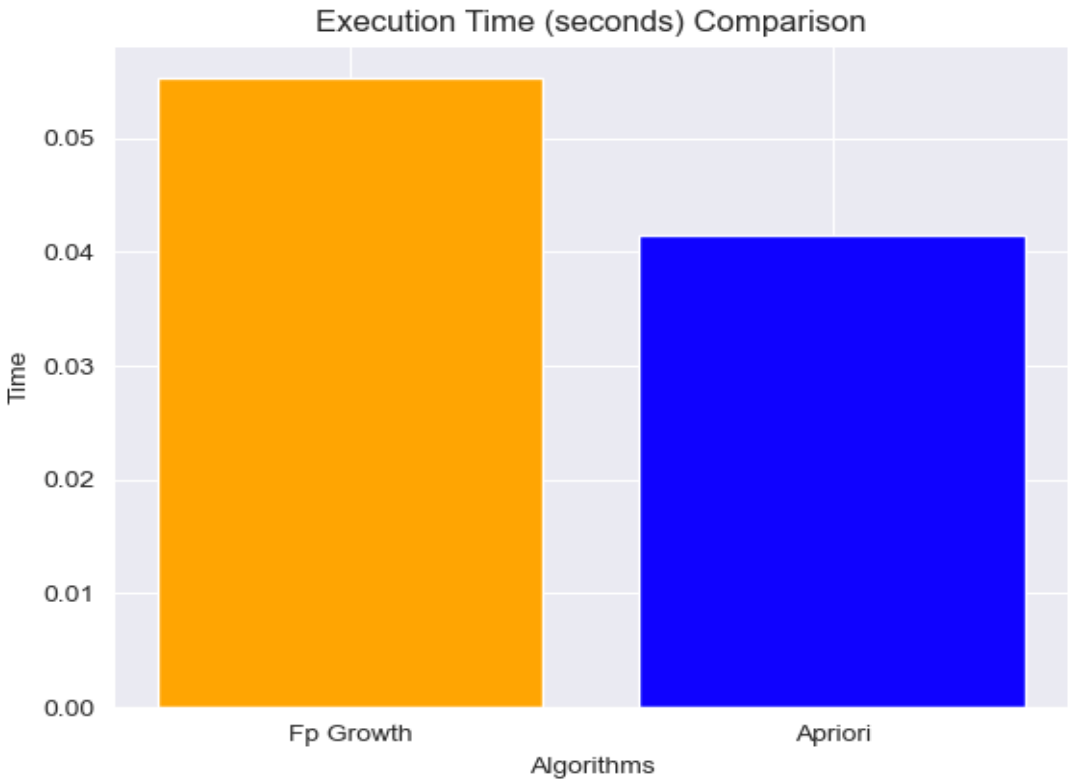

圖 7 FP-Growth 與 Apriori 分析時間比較

伍、結論

本研究對一筆交易資料集進行關聯規則分析，並且比較了兩種關聯規則 Apriori 與 FP-Growth 的分析時間，利用關聯規則分析發現資料集中的頻繁項與樣本物品之間的關聯規則，利用這些規則能夠幫助我們了解顧客的購買行為，或是找出商品之間的替代關係，並為市場行銷與產品規劃提供有用資訊。

參考文獻

Agrawal R., Srikant R., “Mining Sequential Patterns,” Proceedings of the 7th 21. Agrawal R.,Imielinski T., Swami A., “Mining Association Rules Between Sets Items in Large Database,” In proc. Of the ACM SIGMOD Conference on Management of Data, pp.207- 216, 1993.