

國立雲林科技大學 資訊管理所

以交易模式進行資料分析之研究

吳杰峰

M11123015

國立雲林科技大學資訊管理所

梁滄愷

M11123036

國立雲林科技大學資訊管理所

王敬翔

M11123051

國立雲林科技大學資訊管理所

劉昱辰

M11123057

國立雲林科技大學資訊管理所

中華民國 112 年 1 月 4 日

摘要

隨著金融交易日益興盛，許多投資人也會使用交易預測來作為投資衡量的手段，本組利用 Apriori 演算法以及 FP-Growth 演算法以及資料及訓練預測模型，發現執行時間就越短。兩個演算法都具有一樣的趨勢，藉由此發現，期盼往後研究者能更踴躍往此領域做研究。

關鍵字: Apriori、FP-Growth、交易模式、資料預測

Research on data analysis based on transaction mode

WU, JIE-FENG

M11123015

Department of Information Management
National Yunlin University of Science and Technology

LIANG, YU-KAI

M11123036

Department of Information Management
National Yunlin University of Science and Technology

WANG, CHING-HSIANG

M11123051

Department of Information Management
National Yunlin University of Science and Technology

LIU, YU-CHEN

M11123057

Department of Information Management
National Yunlin University of Science and Technology

Abstract

With the increasing prosperity of financial transactions, many investors will also use transaction forecasting as a means of investment measurement. This group uses the Apriori algorithm and FP-Growth algorithm, as well as data and training forecasting models, and finds that the execution time is shorter. Both algorithms have the same trend. With this discovery, it is hoped that researchers will be more active in this field in the future.

Keywords: Apriori, FP-Growth, transaction model, data forecast

一、動機

現今交易方式多變，包含金融、股市、期貨等金融交易皆日益成熟，然而資訊科技的進步更使得股票等金融交易模式獲得更好的預測效果，透過資料分析預測讓投資者有效的收益。

二、目的

本組利用 Apriori 演算法以及 FP-Growth 演算法以及師長提供之數據及進行預測與分析，我們期盼從中獲得預測結果，藉此來判斷何種演算法之效果，可以獲得最有效的預測結果。

三、方法

3.1 操作說明（說明如何操作繳交的程式）

如圖六、圖七，把上面 FP-Growth 的模型套入資料集中，給使用者輸入產品 ID 後，並顯示推薦產品名單。

四、實驗

4.1 前置處理（說明對資料集做了甚麼前置處理）

此研究對資料集先把購買數量小於等於零的商品去除掉，並針對有相同發票號碼的產品做成一個串列。

4.2 實驗設計（實驗如何進行、參數如何設定以及使用的評估指標等）

Apriori 的演算法，在最先開始做關聯分析時，如圖一，先用

`apriori(items,min_support=0.001,min_confidence=0.2,min_lift=3,max_length=7)` 設定好最小支持度、最小信心水準設成 0.2、最小提升度和 list 中出現最大的資料集，在用 `time.perf_counter()` 找出模型訓練時間，第一次的測試執行時間為 0.32989 秒。如圖二，第二次把關聯分析做成

`apriori(items,min_support=0.001,min_confidence=0.02,min_lift=3,max_length=7)` 的

最小信心水準改成 0.02，而第二次的執行時間為 0.56616 秒。

FP-Growth 演算法，在設定關聯分析時，如圖三，用 `fpgrowth(df, min_support=0.001, use_colnames=True)` 把最小支持度設定為 0.001，也就是資料集丟進去並把返回至少有 0.1% 支持的資料集集合的名列，而測試執行時間為 1.37804 秒，如圖五，用 `association_rules(fp, metric="lift", min_threshold=1)` 評估規則是否感興趣的指標為 lift，`min_threshold` 參數設定為 1 來決定候選規則是否感興趣，並顯示出來。

在第二次設定關聯分析時，如圖四，`fpgrowth(df, min_support=0.005, use_colnames=True)` 把最小支持度改為 0.005，而測試時間變為 1.18 秒。

4.3 實驗結果 (以圖表呈現結果、分析結果及解讀)(應分析績效好壞之原因、比較不同分群方法之執行時間、實驗結果以表格或圖形統整)

```
1 from apyori import apriori
2 import time
3 start = time.perf_counter()
4 association_rules = apriori(items, min_support=0.001, min_confidence=0.2, min_lift=3, max_length=7)
5 association_results = list(association_rules)
6 end = time.perf_counter()
7 # 模型訓練的時間
8 exetime = end - start
9 print("執行時間:", round(exetime, 5))
10 print(len(association_results))
11 lift=[]
12 support=[]
13 rule=[]
14 confidence=[]
15 for item1 in association_results:
16     pair = item1[0]
17     item2 = [x for x in pair]
18     rule.append(str(item2[0]) + " -> " + str(item2[1]))
19     support.append(str(item1[1]))
20     confidence.append(str(item1[2][0][2]))
21     lift.append(str(item1[2][0][3]))
22     # print("=====")
23 data = {
24     "Rule": rule,
25     "Support": support,
26     "Confidence": confidence,
27     "Lift": lift
28 }
29 data = pd.DataFrame(data)
```

執行時間: 0.32989

59

圖一、Apriori 演算法第一次測試結果執行時間

```

1 start = time.perf_counter()
2 association_rules = apriori(items,min_support=0.001,min_confidence=0.02,min_lift=3,max_length=7)
3 association_results = list(association_rules)
4 end = time.perf_counter()
5 #模型訓練的時間
6 exetime = end - start
7 print("執行時間:", round(exetime,5))
8 print(len(association_results))
9 lift=[]
10 support=[]
11 rule=[]
12 confidence=[]
13 for item1 in association_results:
14     pair = item1[0]
15     item2 = [x for x in pair]
16     rule.append(str(item2[0]) + " -> " + str(item2[1]))
17     support.append(str(item1[1]))
18     confidence.append(str(item1[2][0][2]))
19     lift.append(str(item1[2][0][3]))
20     # print("=====")
21 data1 = {
22     "Rule":rule,
23     "Support":support,
24     "Confidence":confidence,
25     "Lift":lift
26 }
27 data1 = pd.DataFrame(data1)
28 data1

```

執行時間: 0.56616

59

圖二、Apriori 演算法第二次測試結果執行時間

```

1 start = time.perf_counter()
2 # fpgrowth(df, min_support=0.6,use_colnames=True)
3 fp =fpgrowth(df, min_support=0.001, use_colnames=True)
4 # fpgrowth(df, min_support=0.6, use_colnames=True)
5 end = time.perf_counter()
6 #模型訓練的時間
7 exetime = end - start
8 print("執行時間:", round(exetime,5))
9 fprules =association_rules(fp, metric="lift", min_threshold=1)
10 fprules

```

執行時間: 1.37804

圖三、FP-Growth 演算法第一次測試結果執行時間

```

1 start = time.perf_counter()
2 fp = fpgrowth(df, min_support=0.005, use_colnames=True)
3 end = time.perf_counter()
4 #模型訓練的時間
5 exetime = end - start
6 print("執行時間:", round(exetime,5))
7 fprules = association_rules(fp, metric="lift", min_threshold=1)

```

執行時間: 1.18883

圖四、FP-Growth 演算法第二次測試結果執行時間

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	{70448}	{70509}	0.004576	0.008137	0.001206	0.263473	32.377911	0.001168	1.346675
1	{70509}	{70448}	0.008137	0.004576	0.001206	0.148148	32.377911	0.001168	1.168542
2	{1708}	{70509}	0.004822	0.008137	0.001699	0.352273	43.290404	0.001659	1.531297
3	{70509}	{1708}	0.008137	0.004822	0.001699	0.208754	43.290404	0.001659	1.257735
4	{70448}	{1708}	0.005206	0.004822	0.001260	0.242105	50.206579	0.001235	1.313082
...									
137	{15280990}	{15286872}	0.007891	0.004302	0.001534	0.194444	45.202760	0.001500	1.236039
138	{14790248}	{14782215}	0.005315	0.003123	0.001425	0.268041	85.815518	0.001408	1.361930
139	{14782215}	{14790248}	0.003123	0.005315	0.001425	0.456140	85.815518	0.001408	1.828936
140	{14790248}	{14929876}	0.005315	0.002603	0.001096	0.206186	79.214324	0.001082	1.256461
141	{14929876}	{14790248}	0.002603	0.005315	0.001096	0.421053	79.214324	0.001082	1.718092

142 rows × 9 columns

圖五、FP-Growth 演算法顯示 conviction 大於 1 的資料集

```

1 # 寫入csv
2 data[['Rule', 'Support', 'Confidence', 'Lift']].to_csv("apridata.csv")
3 data1[['Rule', 'Support', 'Confidence', 'Lift']].to_csv("apridata1.csv")
4 fprules[['antecedents', "consequents"]].to_csv("fprules.csv")

```

```

1 # 讀取剛剛寫入的檔案
2 aprdf=pd.read_csv("/content/apridata.csv").drop(['Unnamed: 0'],axis=1)
3 aprdf1=pd.read_csv("/content/apridata1.csv").drop(['Unnamed: 0'],axis=1)
4 fpdf=pd.read_csv("/content/fprules.csv").drop(['Unnamed: 0'],axis=1)

```

圖六、把模型寫入資料集

```

1 print('請輸入想購買之商品：')
2 productlist=list(map(int, input().split(",")))
3 antecedents =fprules["antecedents"]
4 consequents =fprules["consequents"]
5 recommend_list = []
6 for i in range(len(fprules)):
7     if(productlist==list(antecedents[i])):
8         recommend_list.append(list(consequents[i]))
9 recommend_list.sort()
10 print("推薦產品名單",recommend_list)

```

請輸入想購買之商品：

1708

推薦產品名單 [[1698], [23004], [70449], [70509], [88411], [88444], [88444, 70509]]

圖七、FP-Growth 輸入產品 ID

FP-Growth		min_support
1	1.37	0.001
2	1.18	0.005

圖八、FP-Growth 2 次執行時間

Apriori		min_confidence
1	0.32	0.2
2	0.56	0.02

圖九、Apriori 2 次執行時間

五、結論

在圖八可以看到在 FP-Growth 的兩次最小支持度的改變為，最小支持度越大，執行時間就越短。而在圖九可以看到在 Apriori 的兩次最小信度的改變為，最小信度越大，執行時間就越短。兩個演算法都是一樣的趨勢。

參考資料

[關聯分析] **Apriori** 演算法; 2018-3-11

Retrieved from: https://www.maxlist.xyz/2018/11/03/python_apriori/