

## Bài 5: KHAI THÁC MẪU KẾT HỢP (TT)

### I. Mục tiêu:

Sau khi thực hành xong, sinh viên nắm được:

- Thuật toán Vertical Apriori.

### II. Tóm tắt lý thuyết:

Thuật toán Apriori áp dụng cho một tập dữ liệu dạng nằm ngang (horizontal) còn thuật toán vertical Apriori áp dụng cho một tập dữ liệu dạng thẳng đứng (vertical).

Thuật toán Vertical Apriori được phát biểu như sau:

```
Algorithm VerticalApriori(Transactions:  $\mathcal{T}$ , Minimum Support: minsup)  
begin  
   $k = 1$ ;  
   $\mathcal{F}_1 = \{ \text{All Frequent 1-itemsets} \}$ ;  
  Construct vertical tid lists of each frequent item;  
  while  $\mathcal{F}_k$  is not empty do begin  
    Generate  $\mathcal{C}_{k+1}$  by joining itemset-pairs in  $\mathcal{F}_k$ ;  
    Prune itemsets from  $\mathcal{C}_{k+1}$  that violate downward closure;  
    Generate tid list of each candidate itemset in  $\mathcal{C}_{k+1}$  by intersecting  
      tid lists of the itemset-pair in  $\mathcal{F}_k$  that was used to create it;  
    Determine supports of itemsets in  $\mathcal{C}_{k+1}$  using lengths of their tid lists;  
     $\mathcal{F}_{k+1} =$  Frequent itemsets of  $\mathcal{C}_{k+1}$  together with their tid lists;  
     $k = k + 1$ ;  
  end;  
  return  $(\cup_{i=1}^k \mathcal{F}_i)$ ;  
end
```

**Ví dụ:** cho tập dữ liệu giao dịch được lưu trữ dưới dạng nằm ngang với  $\text{minsup} = 2$

TID	Itemset
100	A, B, E
200	B, D
300	B,C
400	A,B,D
500	A,C
600	A,B,C

Chuyển dữ liệu từ dạng nằm ngang thành dạng thẳng đứng

Items	TID list
A	100, 400, 500, 600
B	100, 200, 300, 400, 600
C	300, 500, 600
D	200, 400
E	100

Do  $minsup = 2$  nên tất cả các item xuất hiện chỉ 1 lần trong giao dịch sẽ được loại trừ khỏi tập dữ liệu

Items	TID list
A	100, 400, 500, 600
B	100, 200, 300, 400, 600
C	300, 500, 600
D	200, 400

Tiếp theo ta tạo 1 danh sách chứa 2-item

Items
A, B
A, C
A, D
B, C
B, D
C, D

Ta kết hợp tất cả item với TID tương ứng

Itemset	TID list
A, B	100, 400, 600
A, C	500, 600
A, D	400
B, C	300, 600
B, D	200, 400
C, D	—

Ta bỏ các 2-item có giá trị support nhỏ hơn  $minsup = 2$ :

Itemset	TID list
A, B	100, 400, 600
A, C	500, 600
B, C	300, 600
B, D	200, 400

Ta lặp lại các bước này nhiều lần, tạo 3- item giao dịch và nó có giá trị support nhỏ hơn giá trị của  $minsup = 2$

Items	TID list
A, B, C	600
A, B, D	400
B, C, D	–

nên ta sẽ khởi tạo luật kết hợp dựa vào bước trước đó

Itemset	TID list
A, B	100, 400, 600
A, C	500, 600
B, C	300, 600
B, D	200, 400

Phát sinh các luật:

$$\begin{aligned}
 A \rightarrow B \text{ có } conf(A \rightarrow B) &= \frac{support(A, B)}{support(A)} = \frac{3}{4} = 75\%. \\
 B \rightarrow A \text{ có } conf(B \rightarrow A) &= \frac{support(A, B)}{support(B)} = \frac{3}{5} = 60\%. \\
 A \rightarrow C \text{ có } conf(A \rightarrow C) &= \frac{support(A, C)}{support(A)} = \frac{2}{4} = 50\%. \\
 C \rightarrow A \text{ có } conf(C \rightarrow A) &= \frac{support(A, C)}{support(C)} = \frac{2}{3} = 67\%. \\
 B \rightarrow C \text{ có } conf(B \rightarrow C) &= \frac{support(B, C)}{support(B)} = \frac{2}{5} = 40\%. \\
 C \rightarrow B \text{ có } conf(C \rightarrow B) &= \frac{support(B, C)}{support(C)} = \frac{2}{3} = 67\%. \\
 B \rightarrow D \text{ có } conf(B \rightarrow D) &= \frac{support(B, D)}{support(B)} = \frac{2}{5} = 40\%. \\
 D \rightarrow B \text{ có } conf(D \rightarrow B) &= \frac{support(B, D)}{support(D)} = \frac{2}{2} = 100\%.
 \end{aligned}$$

### III. Nội dung thực hành:

Cho CSDL với các giao dịch sau:

TID	Itemset
1	Wine, Chips, Bread, Butter, Milk, Apple
2	Wine, Bread, Butter, Milk
3	Bread, Butter, Milk
4	Chips, Apple
5	Wine, Chips, Bread, Butter, Milk, Apple
6	Wine, Chips, Milk
7	Wine, Chips, Bread, Butter, Apple
8	Wine, Chips, Milk
9	Wine, Bread, Apple
10	Wine, Bread, Butter, Milk
11	Chips, Bread, Butter, Apple
12	Wine, Butter, Milk, Apple
13	Wine, Chips, Bread, Butter, Milk
14	Wine, Bread, Milk, Apple
15	Wine, Bread, Butter, Milk, Apple
16	Wine, Chips, Bread, Butter, Milk, Apple
17	Chips, Bread, Butter, Milk, Apple
18	Chips, Butter, Milk, Apple
19	Wine, Chips, Bread, Butter, Milk, Apple
20	Wine, Bread, Butter, Milk, Apple
21	Wine, Chips, Bread, Milk, Apple
22	Chips

với  $minsup = 3$ .

#### 1. Sử dụng thư viện

- Cài đặt pyECLAT (pip install pyECLAT):

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Huynh>pip install pyECLAT
Collecting pyECLAT
  Downloading pyECLAT-1.0.2-py3-none-any.whl (6.3 kB)
Collecting numpy>=1.17.4
  Downloading numpy-1.21.6-cp37-cp37m-win_amd64.whl (14.0 MB)
----- 14.0/14.0 MB 5.8 MB/s
Collecting tqdm>=4.41.1
  Downloading tqdm-4.65.0-py3-none-any.whl (77 kB)
----- 77.1/77.1 kB ? eta 0
Requirement already satisfied: pandas>=0.25.3 in c:\users\huynh\appdata\local\programs\python\python37\lib\site-packages (from pyECLAT) (0.25.3)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\huynh\appdata\local\programs\python\python37\lib\site-packages (from pandas>=0.25.3->pyECLAT) (2.8.2)
Requirement already satisfied: pytz>=2017.2 in c:\users\huynh\appdata\local\programs\python\python37\lib\site-packages (from pandas>=0.25.3->pyECLAT) (2019.3)
Requirement already satisfied: colorama in c:\users\huynh\appdata\local\programs\python\python37\lib\site-packages (from tqdm>=4.41.1->pyECLAT) (0.4.3)
Requirement already satisfied: six>=1.5 in c:\users\huynh\appdata\local\programs\python\python37\lib\site-packages (from python-dateutil>=2.6.1->pandas>=0.25.3->pyECLAT) (1.12.0)
Installing collected packages: tqdm, numpy, pyECLAT
Attempting uninstall: numpy
  Found existing installation: numpy 1.17.2
  Uninstalling numpy-1.17.2:
    Successfully uninstalled numpy-1.17.2
Successfully installed numpy-1.21.6 pyECLAT-1.0.2 tqdm-4.65.0
```

- Tạo file data.csv (sử dụng lại data.csv của Bài 4) và đọc dữ liệu từ file.

```
import numpy as np
import pandas as pd
from pyECLAT import ECLAT

dataframe = pd.read_csv('D:\\Huynh\\DataMining_Lab\\data\\tuan5\\data.csv', header=None)
print(dataframe)
```

	0	1	2	3	4	5
0	Wine	Chips	Bread	Butter	Milk	Apple
1	Wine	NaN	Bread	Butter	Milk	NaN
2	NaN	NaN	Bread	Butter	Milk	NaN
3	NaN	Chips	NaN	NaN	NaN	Apple
4	Wine	Chips	Bread	Butter	Milk	Apple
5	Wine	Chips	NaN	NaN	Milk	NaN
6	Wine	Chips	Bread	Butter	NaN	Apple
7	Wine	Chips	NaN	NaN	Milk	NaN
8	Wine	NaN	Bread	NaN	NaN	Apple
9	Wine	NaN	Bread	Butter	Milk	NaN
10	NaN	Chips	Bread	Butter	NaN	Apple
11	Wine	NaN	NaN	Butter	Milk	Apple
12	Wine	Chips	Bread	Butter	Milk	NaN
13	Wine	NaN	Bread	NaN	Milk	Apple
14	Wine	NaN	Bread	Butter	Milk	Apple
15	Wine	Chips	Bread	Butter	Milk	Apple
16	NaN	Chips	Bread	Butter	Milk	Apple
17	NaN	Chips	NaN	Butter	Milk	Apple
18	Wine	Chips	Bread	Butter	Milk	Apple
19	Wine	NaN	Bread	Butter	Milk	Apple
20	Wine	Chips	Bread	NaN	Milk	Apple
21	NaN	Chips	NaN	NaN	NaN	NaN

- Chuyển dữ liệu thành lớp ECLAT và khởi tạo DataFrame nhị phân

```
eclat_instance = ECLAT(data=dataframe, verbose=True)
print(eclat_instance.df_bin)
```

	Butter	Chips	Wine	Bread	Apple	Milk
0	1	1	1	1	1	1
1	1	0	1	1	0	1
2	1	0	0	1	0	1
3	0	1	0	0	1	0
4	1	1	1	1	1	1
5	0	1	1	0	0	1
6	1	1	1	1	1	0
7	0	1	1	0	0	1
8	0	0	1	1	1	0
9	1	0	1	1	0	1
10	1	1	0	1	1	0
11	1	0	1	0	1	1
12	1	1	1	1	0	1
13	0	0	1	1	1	1
14	1	0	1	1	1	1
15	1	1	1	1	1	1
16	1	1	0	1	1	1
17	1	1	0	0	1	1
18	1	1	1	1	1	1
19	1	0	1	1	1	1
20	0	1	1	1	1	1
21	0	1	0	0	0	0

- Khởi tạo các luật kết hợp

```
# count items in each row
items_per_transaction = eclat_instance.df_bin.astype(int).sum(axis=1)
# the item should appear at least at 5% of transactions
min_support = 0.5
# start from transactions containing at least 2 items
min_combination = 2
# up to maximum items per transaction
max_combination = max(items_per_transaction)
rule_indices, rule_supports = eclat_instance.fit(min_support=min_support,
                                                min_combination=min_combination,
                                                max_combination=max_combination,
                                                separator=' & ',
                                                verbose=True)

result = pd.DataFrame(rule_supports.items(), columns=['Item', 'Support'])
result.sort_values(by=['Support'], ascending=False)
print(result)
```

	Item	Support
0	Butter & Wine	0.500000
1	Butter & Bread	0.590909
2	Butter & Apple	0.500000
3	Butter & Milk	0.590909
4	Wine & Bread	0.590909
5	Wine & Apple	0.500000
6	Wine & Milk	0.636364
7	Bread & Apple	0.545455
8	Bread & Milk	0.590909
9	Apple & Milk	0.500000
10	Butter & Bread & Milk	0.500000
11	Wine & Bread & Milk	0.500000

## 2. Yêu cầu:

- Cài đặt lại thuật toán Vertical Apriori.
- Viết file báo cáo trình bày tóm tắt lại phần code do em tự viết và so sánh kết quả với hàm có sẵn trong thư viện.