Bài 4: KHAI THÁC MẪU KẾT HỢP

I. Mục tiêu:

Sau khi thực hành xong, sinh viên nắm được:

- Thuật toán Apriori.

II. Tóm tắt lý thuyết:

Thuật toán Apriori là thuật toán được đưa ra cho việc khai phá hạng mục thường xuyên, nghĩa là tìm tất cả tập các hạng mục (itemset) S có độ phổ biến (support) thỏa mãn độ phổ biến tối thiểu *minsupp*: $supp(S) \geqslant minsupp$.

1. Các khái niệm cơ bản:

Cho các item $I = i_1, \ldots, i_m$ và cơ sở dữ liệu giao dịch $D = t_1, \ldots, t_n$

TID	Itemset
t_1	i_1, i_2, i_m
t_2	i_1
$\ldots$	$\ldots$
t_n	i_2, i_m

hoặc

	i_1	i_2	$\ldots$	i_m
t_1	1	1	$\ldots$	1
t_2	1	0	$\ldots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_n	0	1	$\ldots$	1

- Hạng mục (item): các mặt hàng $i_1, i_2, \ldots, i_m$.
- Tập các hạng mục (itemset): danh sách các hạng mục trong giỏ hàng như $\{i_1, i_2, i_m\}, \{i_1\}, \ldots, \{i_2,$
- Giao dịch (transaction): tập các hạng mục được mua trong một giỏ hàng, kèm với mã giao dịch (TID).
- Mẫu phổ biến (frequent item): là mẫu xuất hiện thường xuyên trong tập dữ liệu như $\{i_2, i_m\}$ xuất hiện khá nhiều trong các giao dịch.
- Tập k-hạng mục (k-itemset): ví dụ danh sách sản phẩm (1-itemset) như $\{i_1, i_2, \ldots, i_m\}$, danh sách cặp sản phẩm đi kèm (2-itemset) như $\{\{i_2, i_m\}, \{i_1, i_4\}, \ldots\}$, danh sách 3 sản phẩm đi kèm (3-itemset) như $\{\{i_1, i_2, i_m\}, \{i_2, i_4, i_m\}, \ldots\}$.

- Độ phổ biến (support) và độ tin cậy (confidence) cho Itemset A và B được tính bằng công thức sau:

$$support(A) = \frac{\textit{số giao dịch mà A xuất hiện}}{\textit{tổng số giao dịch}}$$
$$conf(A \to B) = \frac{support(A, B)}{support(A)}$$

- Tập phổ biến (frequent itemset): là tập các hạng mục S (itemset) thỏa mãn độ phổ biến tối thiểu ($minsupp$ – do người dùng xác định như 40% hoặc xuất hiện 5 lần). Nếu $supp(S) \geqslant minsupp$ thì S là tập phổ biến.

- Luật kết hợp (association rule): kí hiệu $A \to B$, nghĩa là khi A có mặt thì B cũng có mặt (với xác suất nào đó).

2. Thuật toán Apriori:

Thuật toán Apriori bắt đầu bằng việc đếm các support của các item riêng biệt để khởi tạo 1-itemsets phổ biến. Tập 1-itemsets được kết hợp để tạo ra 2-itemset ứng cử viên mà support của nó được đếm. Tập 2-itemsets được tiếp tục dùng. Tổng quát, các itemset chiều dài k được sử dụng để khởi tạo các ứng cử viên chiều dài $(k+1)$ cho việc tăng giá trị của k. Cho $\mathcal{F}_k$ ký hiệu tập hợp k-itemsets phổ biến, và C_k ký hiệu tập hợp k-itemsets ứng cử. Cốt lõi của xấp xỉ là để khởi tạo lặp lại $(k+1)$-ứng cử viên $\mathcal{C}_{k+1}$ từ k-itemsets trong $\mathcal{F}_k$ đã được tìm thấy bởi thuật toán. Các mẫu thường xuyên của $(k+1)$-ứng cử này được tính đối với cơ sở dữ liệu giao dịch (transaction). Trong khi việc khởi tạo $(k+1)$-ứng cử viên, không gian tìm kiếm có thể được xén bớt bởi việc kiểm tra tất cả k-subset của $\mathcal{C}_{k+1}$ hay không được bao gồm trong $\mathcal{F}_k$. Nếu 1 cặp itemset X và Y trong $\mathcal{F}_k$ có $(k-1)$ item chung thì sự kết nối giữa chúng sử dụng $(k-1)$ item chung sẽ khởi tạo một itemset ứng cử kích thước $(k+1)$. Ví dụ, 2 tập 3-itemset $\{a, b, c\}$ (hoặc abc cho ngắn gọn) và $\{a, b, d\}$ (hoặc abd cho ngắn gọn), khi chúng kết nối với nhau trong 2 item chung a và b, sẽ sinh ra 4-itemset ứng cử $abcd$.
Thuật toán Apriori được phát biểu như sau:

Algorithm *Apriori*(Transactions: $\mathcal{T}$, Minimum Support: *minsup*)
begin
  $k = 1$;
  $\mathcal{F}_1 = \{$ All Frequent 1-itemsets $\}$;
  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;
    Determine $\mathcal{F}_{k+1}$ by support counting on $(\mathcal{C}_{k+1}, \mathcal{T})$ and retaining
      itemsets from $\mathcal{C}_{k+1}$ with support at least *minsup*;
    $k = k + 1$;
  end;
  return$(\cup_{i=1}^{k} \mathcal{F}_i)$;
end

Ví dụ: : Cho tập dữ liệu gồm 6 giao dịch với 0 biểu diễn sự vắng mặt của một item và 1 biểu diễn sự có mặt của nó

TID	Wine	Chips	Bread	Milk
1	1	1	1	1
2	1	0	1	1
3	0	0	1	1
4	0	1	0	0
5	1	1	1	1
6	1	1	0	1

với $minsupp = 60\%$, $min\ confidence\ = 80\%$.

- Với $k = 1$, ta khởi tạo $\mathcal{F}_1$

Item	frequency	support
Wine	4	67%
Chips	4	67%
Bread	4	67%
Milk	5	83%

- Khởi tạo $\mathcal{C}_2$ bằng việc kết hợp các cặp item của $\mathcal{F}_1$

$$\{\{Wine, Chips\}, \{Wine, Bread\}, \{Wine, Milk\}, \{Chips, Bread\}, \{Chips, Milk\}, \{Bread, Milk\}\}$$

- Tạo $\mathcal{F}_2$

Item	frequency	support
Wine, Chips	3	50%
Wine, Bread	3	50%
Wine, Milk	4	67%
Chips, Bread	2	33%
Chips, Milk	3	50%
Bread, Milk	4	67%

- Tìm các hạng mục quan trọng dựa vào $minsup = 60\%$ nên ta lấy các 2-item sau:

$$\{Wine, Milk\}, \{Bread, Milk\}$$

- Phát sinh các luật

$$Wine \rightarrow Milk \text{ có } conf(Wine \rightarrow Milk) = \frac{support(Wine, Milk)}{support(Wine)} = 4/4 = 100\%$$
$$Milk \rightarrow Wine \text{ có } conf(Milk \rightarrow Wine) = \frac{support(Wine, Milk)}{support(Milk)} = 4/5 = 80\%$$
$$Bread \rightarrow Milk \text{ có } conf(Bread \rightarrow Milk) = \frac{support(Bread, Milk)}{support(Bread)} = 4/4 = 100\%$$
$$Milk \rightarrow Bread \text{ có } conf(Milk \rightarrow Bread) = \frac{support(Milk, Bread)}{support(Milk)} = 4/5 = 80\%$$

- Ở bước lược bỏ, ta có $\mathcal{F}_2 = \{(Wine, Milk), (Bread, Milk)\}$

- Khởi tạo $\mathcal{C}_3$ bằng việc kết hợp các cặp item của $\mathcal{F}_2$

$$\{\{Wine, Bread, Milk\}\}$$

- Tạo $\mathcal{F}_3$

Item	frequency	support
Wine, Bread , Milk	3	50%

- Tìm các hạng mục quan trọng dựa vào $minsup = 60\%$ nên $\mathcal{F}_3 = \varnothing \Rightarrow$ Thuật toán kết thúc. Vậy ta có 2 luật sau:

$$Wine \rightarrow Milk \text{ có } conf(Wine \rightarrow Milk) = 100\%$$

$$Bread \rightarrow Milk \text{ có } conf(Bread \rightarrow Milk) == 100\%$$

III. Nội dung thực hành:

1. Cài đặt thuật toán Apriori

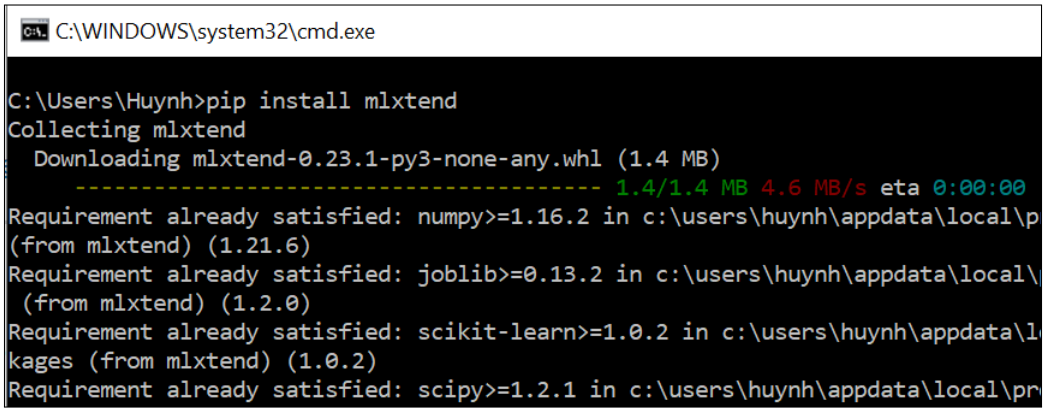
- Cho CSDL với các giao dịch sau:

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

- Sử dụng Excel để tạo file "data.csv' như trong hình

	A	B	C	D	E	F
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				

- Cài đặt mlxtend: pip install mlxtend



- Import các thư viện và load dữ liệu

```
import numpy as np
import pandas as pd
from mlxtend.frequent_patterns import apriori,association_rules
from mlxtend.preprocessing import TransactionEncoder

#Load dữ liệu
df = pd.read_csv('D:\\Huynh\\DataMining_Lab\\data\\tuan4\\data.csv', header=None)
print(df)
```

	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 DataFrame thành dạng danh sách (list)

```
records = []
for i in range(0,data.shape[0]):
    records.append([str(data.values[i,j]) for j in range(0,data.shape[1])])
```

- Chuyển records thành dạng transaction

```
#chuyển records thành transaction
te = TransactionEncoder()
te_ary = te.fit(records).transform(records)
df1 = pd.DataFrame(te_ary, columns=te.columns_)
print(df1)
```

- Xây dựng mô hình Apriori

```
frequent_itemsets = apriori(df1, min_support=0.6, use_colnames=True)
print(frequent_itemsets)
```

	support	itemsets
0	0.681818	(Apple)
1	0.727273	(Bread)
2	0.681818	(Butter)
3	0.636364	(Chips)
4	0.772727	(Milk)
5	0.727273	(Wine)
6	0.818182	(nan)
7	0.636364	(Milk, Wine)

- In kết quả

```python
#bulid association rules using support metric
rules = association_rules(frequent_itemsets, metric="support", support_only=True,
                          min_threshold=0.1)

rules = rules[['antecedents', 'consequents', 'support']]
print(rules)
```

```
  antecedents consequents   support
0      (Milk)      (Wine)  0.636364
1      (Wine)      (Milk)  0.636364
```

2. Yêu cầu:

- Cài đặt lại thuật toán 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.