TRƯỜNG ĐH KHTN – TP HCM
KHOA CÔNG NGHỆ THÔNG TIN

KHAI THÁC TẬP PHỔ BIẾN (Frequent Itemset Mining)

DATA MINING — HCMUS - 2024

Nội dung

1. Các khái niệm
2. Khai thác tập phổ biến
3. Khai thác tập phổ biến đóng
4. Khai thác tập phổ biến tối đại
5. Nhận xét

1. Các khái niệm

❖ Hạng mục (***item***): Cho I là một tập các thuộc tính nhị phân. Cho $I = \{I_1, I_2, \ldots, I_m\}$, mỗi I_m là một item.

❖ Tập hạng mục (***itemset***): Một tập $X \subseteq I$ là một tập các hạng mục.

❖ Một CSDL giao tác là một tập gồm nhiều ***itemset***, mỗi ***itemset*** là một giao tác được định danh bởi một giá trị duy nhất là mã giao tác (***tid***).

1. Các khái niệm

Cho CSDL giao tác D như sau.

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

❖ Độ hỗ trợ (***support***) của tập hạng mục X trong cơ sở dữ liệu D, $sup(X)$, là phần trăm số giao tác trong D có chứa X.

❖ Ví dụ:

- $sup(A)$ = 4/6*100 = 66.67%
- $sup(ACD)$ = 2/6*100 = 33.3%

1.1 Tập phổ biến

Cho một tập hạng mục X và cơ sở dữ liệu D.

❖ Tập X là phổ biến trong D nếu $sup(X) \geq minsup$, với $minsup$ là ngưỡng hỗ trợ tối thiểu do người dùng đặt.

❖ Ví dụ: $minsup$ = 70%

$sup(A)$ = 66.67% < $minsup$

$sup(C)$ = 100% > $minsup$

- A **không** là **tập phổ biến.**
- C là **tập phổ biến.**

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

1.2 Tập phổ biến đóng

Cho$I = \{i_1, i_2, \ldots, im\}$ - là tập các items
Cho $T = \{t_1, t_2, \ldots tm\}$ - là tập các giao tác.

❖ Kết nối Galois

Cho quan hệ hai ngôi $\delta \subseteq I \times T$ chứa CSDL cần khai thác.

Với: $X \subseteq I$ và $Y \subseteq T$, ta định nghĩa hai ánh xạ giữa $P(I)$ và $P(T)$ như sau:

a) $t: P(I) \rightarrow P(T), t(X) = \{y \in T | \forall\, x \in X,\ x\delta y\}$

b) $i: P(T) \rightarrow P(I), i(Y) = \{x \in I | \forall y \in Y, x\delta y\}$

1.2 Tập phổ biến đóng

Ánh xạ (1): $t(X)$ lấy tất cả tid của giao tác có chứa tập hạng mục X.

Ánh xạ (2): $i(Y)$ lấy tất cả item tồn tại trong tất cả giao tác Y.

❖ Toán tử đóng: $c = i \circ t$

❖ Tập hạng mục X là tập đóng nếu $c(X) = X$.

⇒ Tập phổ biến đóng: là tập hạng mục đóng thỏa ngưỡng *minsup* cho trước.

1.2 Tập phổ biến đóng

❖ Ví dụ: Cho cơ sở dữ liệu D với minsup = 30%.

Kiểm tra AW, CD có phải là tập phổ biến đóng?

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

Sử dụng toán tử đóng:

c(AW) = i(t(AW)) = i(1345) = ACW

c(CD) = i(t(CD)) = i(2456) = CD

Vậy CD là tập phổ biến đóng, AW **không** là tập phổ biến đóng.

1.2 Tập phổ biến đóng

❖ Tóm tắt định nghĩa: Tập phổ biến đóng là tập phổ biến mà không có tập nào bao nó có cùng độ phổ biến.

- Với F là tập hợp gồm tất cả tập phổ biến.

$$F = \{X \mid X \subseteq I \text{ và } sup(X) \geq minsup\}$$

- Gọi C là tập hợp gồm tất cả tập phổ biến đóng.

$$\Rightarrow \boxed{C = \{X \mid X \in F \text{ và } \nexists Y \supset X \text{ mà } sup(X) = sup(Y)\}}$$

1.3 Tập phổ biến tối đại

❖ Định nghĩa: Tập phổ biến tối đại là tập phổ biến mà không có tập nào bao nó là phổ biến.

$$\boxed{M = \{X \mid X \in F \text{ và } \nexists Y \supset X \text{ mà } Y \in F\}}$$

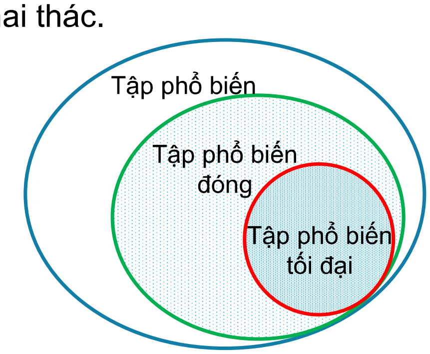
❖ Ví dụ: Cho 3 tập phổ biến $\{A,B\}$, $\{A,C\}$, $\{A,B,D\}$

- $\{A,C\}$ và $\{A,B,D\}$ là ***tập phổ biến tối đại***.

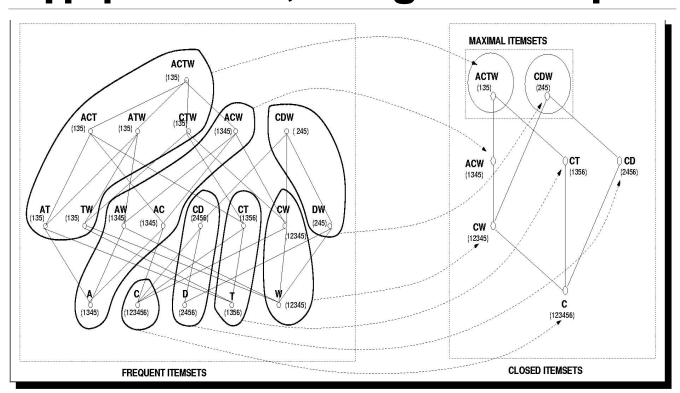
- $\{A,B\}$ **không** phải là tập phổ biến tối đại. Do $\{A,B\}$ là tập con của $\{A,B,D\}$.

1.4 So sánh tập phổ biến

❖ Số lượng tập phổ biến phát sinh trong quá trình khai thác.



Tập phổ biến, đóng và tối đại



2. Khai thác tập phổ biến

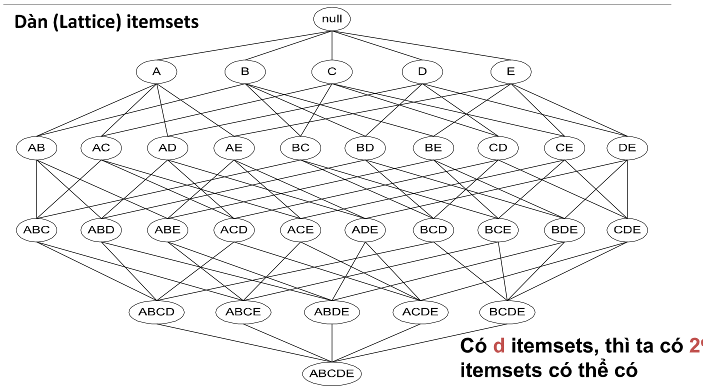
- Input: Tập các giao dịch T, với tập itemstes I
- Output: Tất cả các itemsets chứa trong I thỏa:
  - support ≥ *minsup*

- Tham số:
  - N = |T|: số lượng giao dịch
  - d = |I|: số lượng iemsets riêng biệt.
  - w: số lượng tối đa items của 1 giao dịch.
  - Có bao nhiêu itemsets có thể có ?

- Quy mô của vấn đề:
  - WalMart bán 100,000 mặt hàng và có thể lưu trữ hàng tỉ giỏ hàng.
  - The Web có hàng tỉ từ và hàng tỉ trang

2. Khai thác tập phổ biến

Dàn (Lattice) itemsets



Có d itemsets, thì ta có 2^d itemsets có thể có

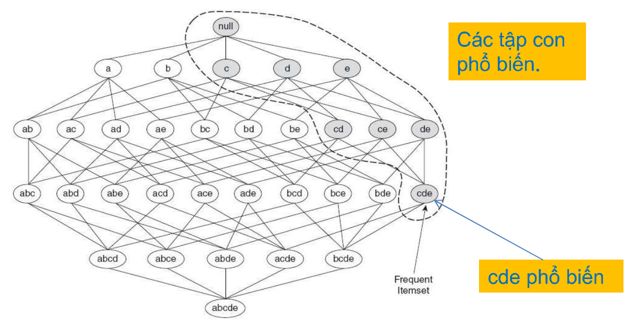
2. Khai thác tập phổ biến

- Quy tắc Apriori :
  - Nếu một tập là phổ biến, thì tất cả tập con của nó phải phổ biến.
  - Nếu 1 tập không phổ biến thì tất cả tập chứa nó không phổ biến.

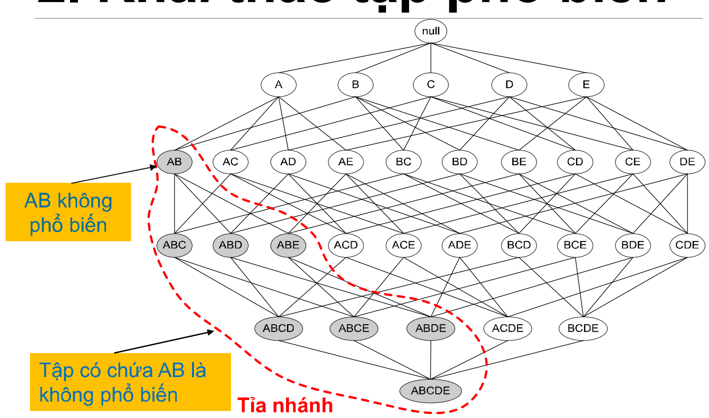
$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

  - Độ hỗ trợ của 1 tập không bao giờ vượt quá độ hỗ trợ các tập con của nó.

2. Khai thác tập phổ biến



2. Khai thác tập phổ biến



2. Khai thác tập phổ biến

❖ Thuật toán **Apriori** (*state-of-the art*) được đề xuất bởi R. Agrawal và R. Srikant vào năm 1994 để khai thác tập phổ biến.

❖ Gọi C_k là các tập có k hạng mục. Thuật toán thực hiện như sau: $k = 1$. F là tập hợp các tập phổ biến.

- Bước 1: Đếm độ hỗ trợ của từng tập trong C_k.
- Bước 2: Phát sinh ứng viên C_{k+1} dựa trên C_k.
- Bước 3: Loại bỏ các ứng viên C_{k+1} chứa tập con C_k không phổ biến.
- Bước 4: Thêm các tập C_k thỏa ngưỡng minsup vào F.

Thuật toán lặp lại đến khi tất cả tập phổ biến được phát sinh.

2.1 Thuật toán Apriori

Đầu vào:
- ***D***, cơ sở dữ liệu các giao tác.
- ***minsup***, ngưỡng support tối thiểu.

Kết quả:
- ***L***, tập các itemset phổ biến.

```
L_1 = lấy tất cả các 1-itemset thỏa minsup trong D;
for (k = 2; L_{k-1} ≠ Ø; k++) {
      C_k = apriori_gen(L_{k-1});
      for each giao tác t ∈ D { // duyệt D
            C_t = subset(C_k,t); //lấy tất cả các ứng viên của C_k có trong t
            for each c ∈ C_t
                  c.count++;
      }
      L_k = {c ∈ C_k|c.count ≥ minsup}
}
return L = ∪_k L_k;
```


2.1 Thuật toán Apriori

Ví dụ: Cho CSDL giao tác như sau. Tìm tất cả tập phổ biến thỏa ngưỡng ***minsup*** = 50% (sup.count ≥ 3).

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

C_1	Support
A	4
C	6
D	4
T	4
W	5

F_1	Support
A	4
C	6
D	4
T	4
W	5

2.1 Thuật toán Apriori

Ví dụ: (tiếp theo) ***minsup*** = 50%

C_2	Support
A,C	4
A,D	2
A,T	3
A,W	4
C,D	4
C,T	4
C,W	5
D,T	2
D,W	3
T,W	3

F_2	Support
A,C	4
A,T	3
A,W	4
C,D	4
C,T	4
C,W	5
D,W	3
T,W	3

C_3	Support
A,C,W	4
A,C,D	2
A,C,T	3
A,T,W	3
C,D,T	2
C,D,W	3
C,T,W	3

Tập con không phổ biến

2.1 Thuật toán Apriori

Ví dụ: (tiếp theo) ***minsup*** = 50%

F_3	Support
A,C,W	4
A,C,T	3
A,T,W	3
C,D,W	3
C,T,W	3

C_4	Support
A,C,T,W	3
C,D,T,W	1

F_4	Support
A,C,T,W	4

⇒ Các tập hạng mục có tập con không phổ biến bị loại bỏ trong quá trình phát sinh ứng viên. Nên cần phải đếm độ hỗ trợ sau đó mới loại bỏ.

Như vậy có tất cả 19 tập hạng mục phổ biến thỏa *minsup* = 50%

2.2 Thuật toán Eclat

❖ Thuật toán **Eclat** (**E**quivalence **Cla**ss **T**ransformation) của M. J. Zaki và đồng sự đề xuất sử dụng mã giao tác (Tidset) để tính nhanh độ hỗ trợ thay vì lưu độ hỗ trợ như Apriori.

TID	Items
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

Items	Tidset
A	1 3 4 5
C	1 2 3 4 5 6
D	2 4 5 6
T	1 3 5 6
W	1 2 3 4 5

$t(A) = 1345;\ t(AD) = t(A) \cap t(D) = 1345 \cap 2456 = 45.$

2.2 Thuật toán Eclat

Cấu trúc IT-tree và các lớp tương đương:

❖Cho X⊆***I***, ta định nghĩa hàm ***p***(**X**,k) = **X**[1:k] gồm k phần tử đầu của **X** và quan hệ tương đương dựa vào tiền tố như sau:

$$\forall X, Y \subseteq I, X \equiv_{\theta_k} Y \Leftrightarrow p(X,k) = p(Y,k)$$

❖Mỗi nút trên IT-tree gồm 2 thành phần:

X×*t*(X)(Itemset×Tidset) được gọi là **IT-pair**, thực chất là một lớp tiền tố. Các nút con của X thuộc về lớp tương đương của X vì chúng chia sẻ chung tiền tố X (*t*(X) là tập các giao dịch có chứa X)

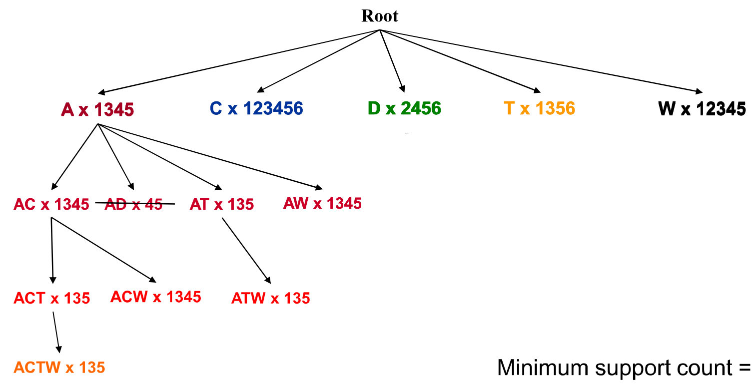
2.2 Thuật toán Eclat

Đầu vào:
- ***P***, các tập 1-hạng mục cùng tidset.
- ***minsup***, ngưỡng support tối thiểu.

Kết quả:
- ***F***, tập các itemset phổ biến.

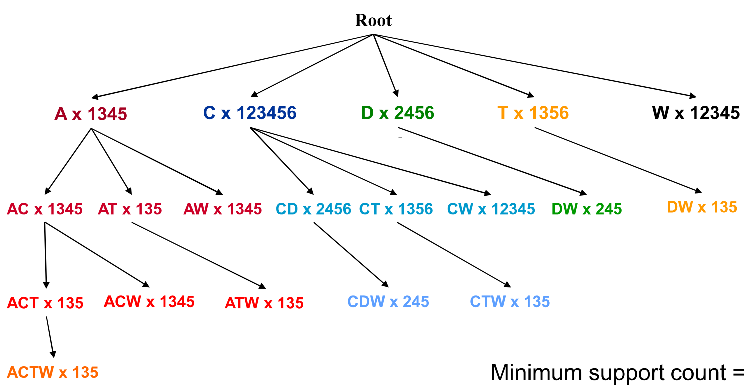
0. **Eclat**($[P]$):
1. for all $X_i \in [P]$ do
2. $\quad T_i = \emptyset$
3. $\quad$ for all $X_j \in [P]$, with $j > i$ do
4. $\quad\quad R = X_i \cup X_j$;
5. $\quad\quad t(R) = t(X_j) \cap t(X_i)$;
6. $\quad\quad$ if $\sigma(R) \geq$ ***minsup*** then
7. $\quad\quad\quad T_i = T_i \cup \{R\}$; $F_{|R|} = F_{|R|} \cup \{R\}$;
8. for all $T_i \neq \emptyset$ do **Eclat**(T_i);

2.2 Thuật toán Eclat



Minimum support count = 3

2.2 Thuật toán Eclat

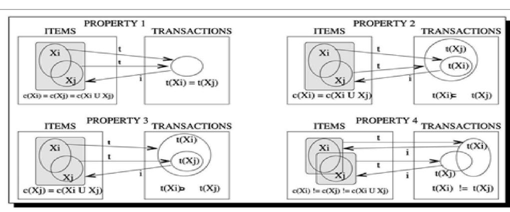


Minimum support count = 3

3. Khai thác tập phổ biến đóng

- ❖M. J. Zaki cùng đồng sự đề xuất Thuật toán CHARM để khai thác những mẫu phổ biến đóng.
- ❖ Thuật toán sử dụng ***tidset*** và ***duyệt theo chiều sâu trước*** tương tự như thuật toán Eclat.
- ❖ Thuật toán áp dụng một số cải tiến để cắt tỉa bớt các tập hạng mục không phổ biến và tìm tập đóng bằng phương pháp dự trên mối quan hệ của các tập hạng mục.

3. Thuật toán Charm



Định lý 1: Đặt $X_i \times t(X_i)$ và $X_j \times t(X_j)$ là hai thành viên bất kỳ của một lớp [P]. Bốn thuộc tính sau là:

1. Nếu t(Xi) = t(Xj), thì c(Xi)= c(Xj) = c(Xi ∪ Xj).
2. Nếu t(Xi) ⊂ t(Xj), thì c(Xi)≠ c(Xj), nhưng c(Xi)= c(Xi ∪ Xj)
3. Nếu t(Xi) ⊃ t(Xj), thì c(Xi)≠ c(Xj), nhưng c(Xj)= c(Xi ∪ Xj)
4. Nếu t(Xi) ⊄ t(Xj)và t(Xj) ⊄ t(Xi), thì c(Xi)≠c(Xj)≠ c(Xi ∪ Xj

3. Thuật toán Charm

Đầu vào: CSDL *D*, *minsup*

Kết quả: tập *FCI* gồm tất cả các tập phổ biến đóng của CSDL

CHARM (*D*, *minsup*):
1: $[\emptyset] = \{l_i \times t(l_i): l_i \in I \wedge \sigma(l_i) \geq minsup\}$
2: CHARM-EXTEND ($[\emptyset]$, $C = \emptyset$)
3: return *C* //tất cả itemset đóng
CHARM-EXTEND ($[P]$, C):
4: **for each** $l_i \times t(l_i)$ in $[P]$
5: $\quad P_i = P \cup l_i$ and $[P_i] = \emptyset$
6: $\quad$ **for each** $l_j \times t(l_j)$ in $[P]$, with j > i
7: $\quad\quad X = l_j$ and $Y = t(l_i) \cap t(l_j)$
8: $\quad\quad$ CHARM-PROPERTY ($X \times Y, l_i, l_j, P_i, [P_i], [P]$)
9: $\quad$ SUBSUMPTION-CHECK (C, P_i)
10: $\quad$ CHARM-EXTEND ($[P_i], C$)
11: $\quad$ **delete** $[P_i]$

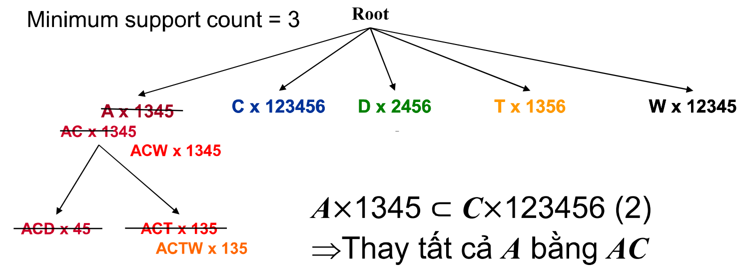
3. Thuật toán Charm

CHARM-PROPERTY $(X \times Y, X_i, X_j, P_i, [P_i], [P])$

if $(\sigma(X) \geq minsup)$ **then**
    if $t(X_i) = t(X_j)$ **then (1)**
        remove X_j from $[P]$
        $P_i = P_i \cup X_j$
    else if $t(X_i) \subset t(X_j)$ **then (2)**
        $P_i = P_i \cup X_j$
    else if $t(X_i) \supset t(X_j)$ **then (3)**
        remove X_j from $[P]$
        Add $X \times Y$ to $[P_i]$
    else if $t(X_i) \neq t(X_j)$ **then (4)**
        Add $X \times Y$ to $[P_i]$

3. Thuật toán Charm

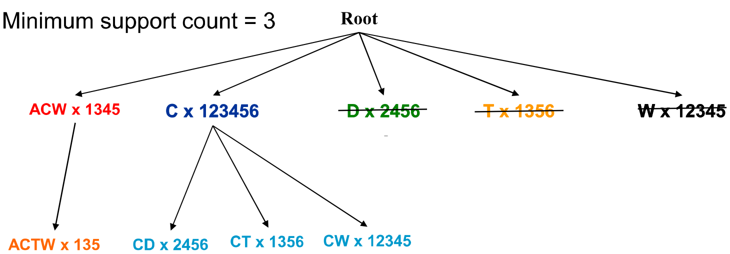
Minimum support count = 3



$A \times 1345 \subset C \times 123456$ (2)
$\Rightarrow$Thay tất cả A bằng AC
$AC \times 1345 \neq D \times 2456$ (4)
$\Rightarrow$Thêm ACD, loại vì < minsup
Tương tự các trường hợp còn lại.

3. Thuật toán Charm

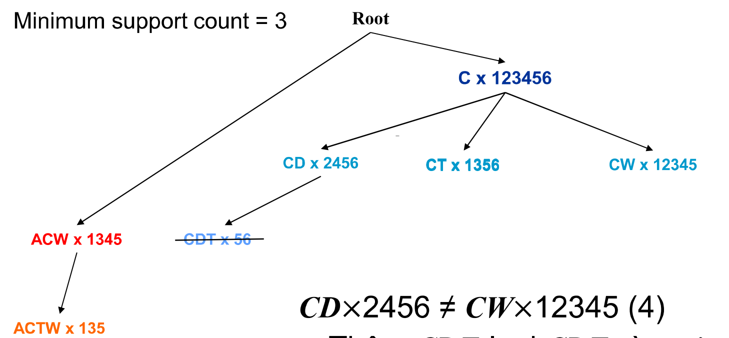
Minimum support count = 3



$C \times 123456 \supset D \times 2456$ (3)
$\Rightarrow$Thêm CD, xóa D
Tương tự các trường hợp còn lại.

3. Thuật toán Charm

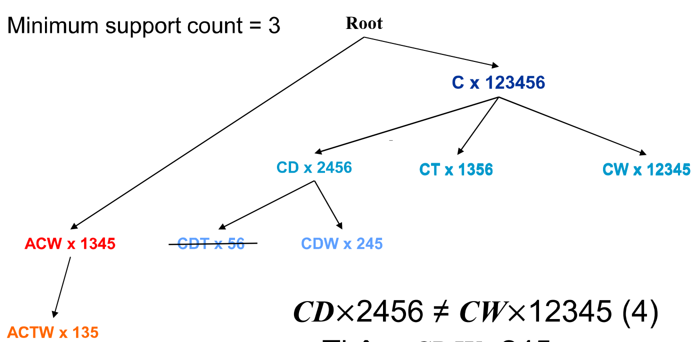
Minimum support count = 3



$CD \times 2456 \neq CW \times 12345$ (4)
$\Rightarrow$Thêm CDT, loại CDT *vì* < $minsup$

3. Thuật toán Charm

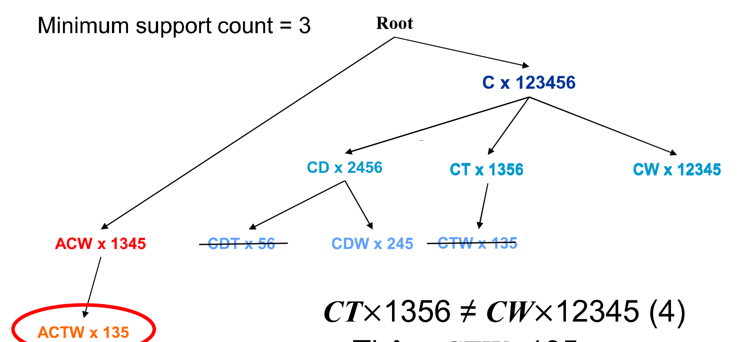
Minimum support count = 3



$CD \times 2456 \neq CW \times 12345$ (4)
$\Rightarrow$Thêm $CDW \times 245$

3. Thuật toán Charm

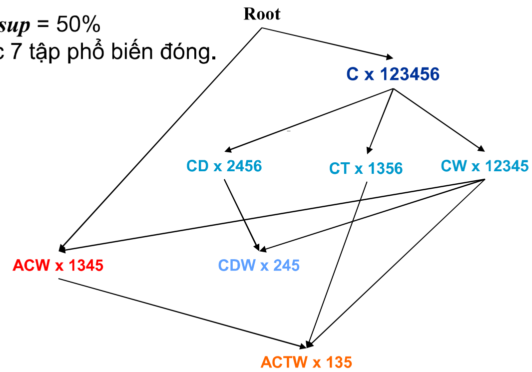
Minimum support count = 3



$CT \times 1356 \neq CW \times 12345$ (4)
$\Rightarrow$Thêm $CTW \times 135$
Loại vì bị bao bởi $ACTW \times 135$

3. Thuật toán Charm

Với ***minsup*** = 50%
Ta được 7 tập phổ biến đóng.



4. Khai thác tập phổ biến tối đại

❖ K. Gouda và M. J.Zaki đề xuất Thuật toán GenMax để tìm tập phổ biến tối đại dựa trên tiến trình backtrack.

❖ Thuật toán cũng sử dụng tidset, và cách duyệt cây tương tự như Eclat.

❖ Từng hạng mục sẽ được lấy ra những hạng mục khả kết hợp với nó (tập kết hợp thỏa *minsup*).

4. Thuật toán GenMax

❖ K. Gouda và M. J.Zaki đề xuất Thuật toán GenMax để tìm tập phổ biến tối đại dựa trên tiến trình backtrack.

❖ Thuật toán cũng sử dụng tidset, và cách duyệt cây tương tự như Eclat.

❖ Từng hạng mục sẽ được lấy ra những hạng mục có thể kết hợp với nó (tập kết hợp thỏa *minsup*).

4. Thuật toán GenMax

Mục tiêu của tiến trình backtrack là:

❖ Lấy ra những tập khả kết hợp với tập hạng mục đang xét.

❖ Kết hợp tập hạng mục với các tập khả kết hợp với nó để tạo tập k+1-hạng mục tiếp theo.

❖ Thực hiện đệ quy đến khi tất cả tập hạng mục phổ biến được rút trích.

4. Thuật toán GenMax

Thuật toán FI-backtrack
Đầu vào:
- I_ℓ tập các itemsets có độ dài l.
- C_ℓ tập những items có thể kết hợp với I.
- l là độ dài của itemset.

Kết quả: itemset phổ biến

FI-backtrack (I_ℓ, C_ℓ, l)
for each $x \in C_\ell$
  $I_{\ell+1} = I_\ell \cup \{x\}$ //đồng thời thêm $I_{\ell+1}$ vào ***FI***
  $P_{\ell+1} = \{y: y \in C_\ell$ and $y > x\}$
  $C_{\ell+1}$ = FI-combine $(I_{\ell+1}, P_{\ell+1})$
  FI-backtrack $(I_{\ell+1}, C_{\ell+1}, l+1)$

4. Thuật toán GenMax

Hàm FI-combine: dùng kết hợp các hạng mục lại với nhau.

FI-combine $(I_{\ell+1}, P_{\ell+1})$
$C = \emptyset$
for each $y \in P_{\ell+1}$
  if $I_{\ell+1} \cup \{y\}$ là phổ biến
    $C = C \cup \{y\}$ //sắp xếp lại C nếu cần
return C;

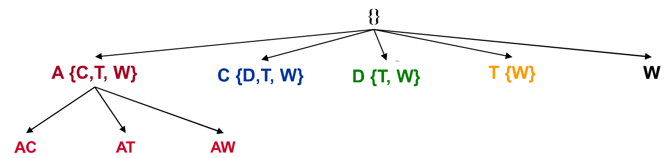
4. Thuật toán GenMax

❖ Để tìm tập phổ biến tối đại, chỉ cần áp dụng điều kiện loại bỏ đi những tập phổ biến không tối đại.

MFI-backtrack (I_ℓ, C_ℓ, l)
1: **for each** x ∈ C_ℓ
2: $I_{\ell+1} = I_\ell \cup \{x\}$
3: $P_{\ell+1} = \{y: y \in C_\ell \text{ and } y > x\}$
4:* **if** $I_{\ell+1} \cup P_{\ell+1}$ có tập bao nó trong MFI
5:* return //tất cả nhánh con bị cắt tỉa
6: $C_{\ell+1}$ = **FI-combine** ($I_{\ell+1}$, $P_{\ell+1}$)
7:* **if** $C_{\ell+1}$ is empty
8:* **if** $I_{\ell+1}$ không có tập nào bao nó trong MFI
9:* MFI = MFI ∪ $I_{\ell+1}$
10: **else** MFI-backtrack ($I_{\ell+1}$, $C_{\ell+1}$, l+1)

4. Thuật toán GenMax

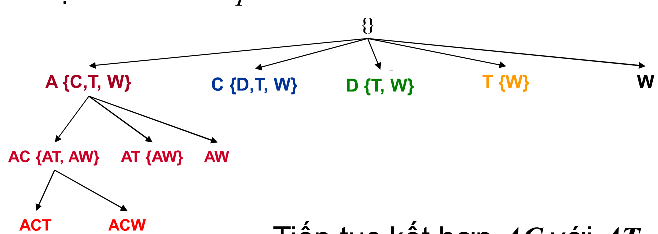
Ví dụ: Đầu tiên ta có các tập khả kết hợp với hạng mục A với *minsup* = 50%



-Đầu tiên kết hợp A lần lượt C, T, W.
-Tập $C_2 = \{AC, AT, AW\} \neq \emptyset$
-Gọi đệ quy hàm backtrack.

4. Thuật toán GenMax

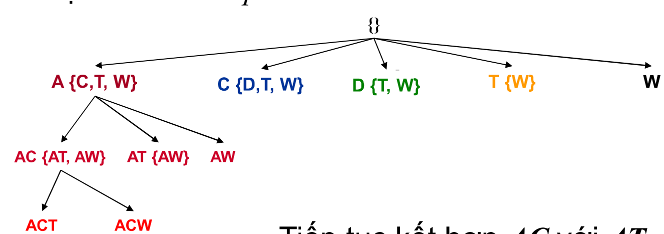
Ví dụ: Đầu tiên ta có các tập khả kết hợp với hạng mục A với *minsup* = 50%



-Tiếp tục kết hợp AC với AT, AW.
-Tập $C_3 = \{ACT, ACW\} \neq \emptyset$
-Gọi đệ quy hàm backtrack.

4. Thuật toán GenMax

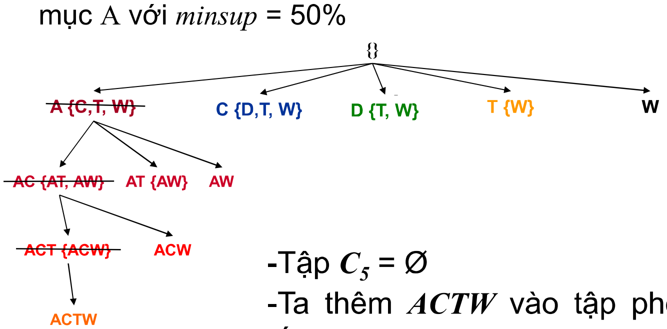
Ví dụ: Đầu tiên ta có các tập khả kết hợp với hạng mục A với *minsup* = 50%



-Tiếp tục kết hợp AC với AT, AW.
-Tập $C_3 = \{ACT, ACW\} \neq \emptyset$
-Gọi đệ quy hàm backtrack.

4. Thuật toán GenMax

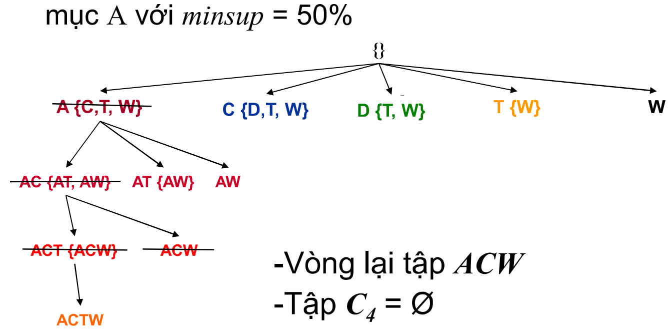
Ví dụ: Đầu tiên ta có các tập khả kết hợp với hạng mục A với *minsup* = 50%



-Tập $C_5 = \emptyset$
-Ta thêm $ACTW$ vào tập phổ biến tối đại.

4. Thuật toán GenMax

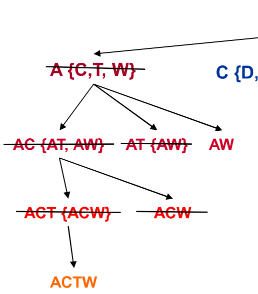
Ví dụ: Đầu tiên ta có các tập khả kết hợp với hạng mục A với *minsup* = 50%



-Vòng lại tập ACW
-Tập $C_4 = \emptyset$
-Loại ACW vì bị bao bởi $ACTW$

4. Thuật toán GenMax

Ví dụ: Đầu tiên ta có các tập khả kết hợp với hạng mục A với *minsup* = 50%

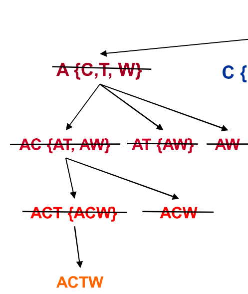


-Tiếp tục trở lại tập ***AT***

-Tỉa toàn bộ nhánh ***AT*** vì bị bao bởi ***ACTW***

4. Thuật toán GenMax

Ví dụ: Đầu tiên ta có các tập khả kết hợp với hạng mục A với *minsup* = 50%

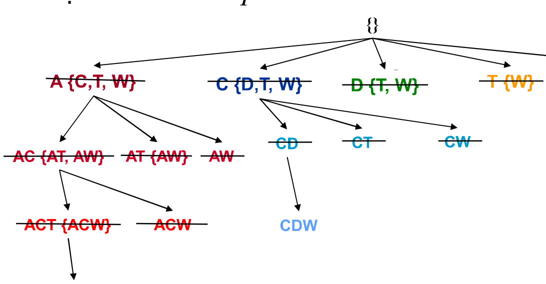


-Trở lại tập ***AW***

-Tỉa toàn bộ nhánh ***AW*** vì bị bao bởi ***ACTW***

4. Thuật toán GenMax

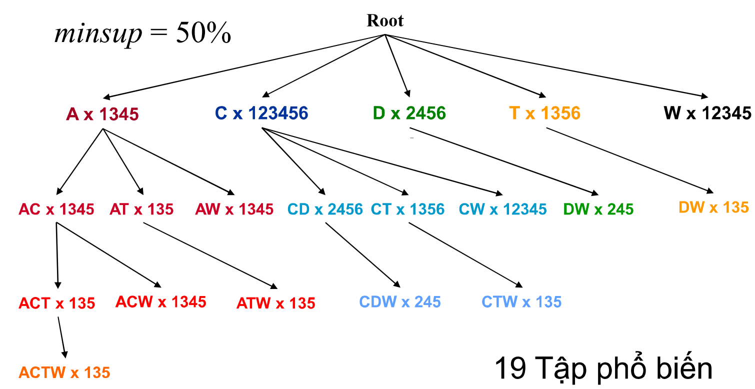
Ví dụ: Đầu tiên ta có các tập khả kết hợp với hạng mục A với *minsup* = 50%



-Tiếp tục hạng mục ***C, D, T, W***

5. Nhận xét

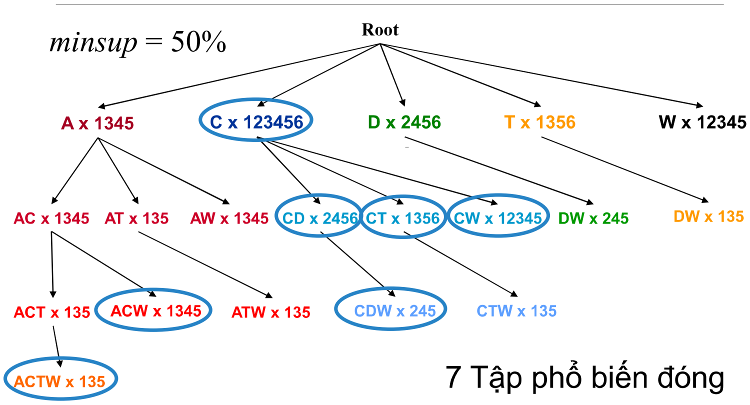
minsup = 50%



19 Tập phổ biến

5. Nhận xét

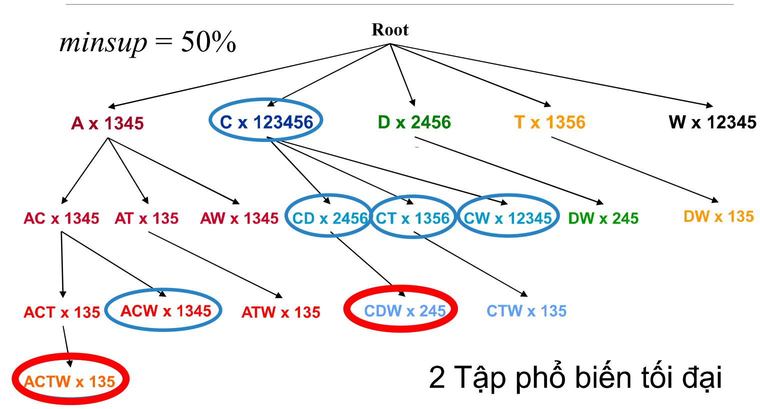
minsup = 50%



7 Tập phổ biến đóng

5. Nhận xét

minsup = 50%



2 Tập phổ biến tối đại

5. Nhận xét

- ❖ Mối quan hệ giữa các tập phổ biến như sau: $M \subseteq C \subseteq F$.
- ❖ Tập phổ biến đóng thể hiện đầy đủ thông tin của tất cả các tập phổ biến cùng với độ hỗ trợ chính xác của nó.
- ❖ Luật kết hợp rút trích từ tập phổ biến đóng sẽ nhỏ gọn hơn, dễ quản lý, phân tích.
- ❖ Khai thác tập phổ biến tối đại thích hợp với CSDL dày đặc, khi mà số lượng tập đóng cũng có thể rất lớn.

Tài liệu tham khảo

[1] M. J. Zaki, ***Closed Itemset Mining And Non-redundant Association Rule Mining,*** Computer Science Department, Rensselaer Polytechnic Institute.

[2] M. J. Zaki, ***Scalable Algorithms for Association Mining,*** IEEE Transactions on Knowledge and Data Engineering, 12(3), May/Jun 2000, pp. 372-390.

[3] M. J. Zaki and K. Gouda, ***GenMax: An Efficient Algorithm for Mining Maximal Frequent Itemsets,*** Data Mining and Knowledge Discovery: An International Journal, 11(3), 2005, pp .223-242.

Thanks for your listening !!
Q & A