

Nội dung

KHAI THÁC TẬP PHỒ BIẾN (Frequent Itemset Mining)

- Các khái niệm
- Khai thác tập phồ biến
- Khai thác tập phồ biến đóng
- Khai thác tập phồ biến tối đai
- Nhận xét

DATA MINING

HCMUS - 2024



1

2

BO MON KHOA HOC MAY TINH

3/18/2024

1. Các khái niệm

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

❖ **Tập hàng 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**).

4

BO MON KHOA HOC MAY TINH

3/18/2024

BO MON KHOA HOC MAY TINH

3

BO MON KHOA HOC MAY TINH

3/18/2024

BO MON KHOA HOC MAY TINH

4

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 hàng 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 phồ biến

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

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

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

- ❖ Ví dụ: $minsup = 70\%$
- ❖ $sup(A) = 66.67\% < minsup$
- ❖ $sup(C) = 100\% > minsup$
- A **Không** là tập phồ biến.
- C là tập phồ biến.

1.2 Tập phồ biến đóng

Choi = $\{i_1, i_2, \dots, i_m\}$ - là tập các items

Cho $T = \{t_1, t_2, \dots, t_m\}$ - 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) $i: P(T) \rightarrow P(I), i(Y) = \{x \in I | \forall y \in Y, x \delta y\}$

3/18/2024

BO MON KHOA HOC MAY TINH

6

1.2 Tập phỗ biến đóng

- Ánh xạ (1): $t(X)$ lấy tất cả tử 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 phỗ biến đóng: là tập hạng mục đóng thỏa ngưỡng minsup cho trước.**

3/18/2024

BO MON KHOA HOC MAY TINH

7

BO MON KHOA HOC MAY TINH 8

1.2 Tập phỗ biến đóng

- Ví dụ: Cho cơ sở dữ liệu D với minsup = 30%. Kiểm tra AW, CD có phải là tập phỗ biến đóng?

Sử dụng toán tử 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

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

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

Vậy CD là tập phỗ biến đóng, AW không là tập phỗ biến đóng.

1.2 Tập phỗ biến đóng

- Tóm tắt định nghĩa: Tập phỗ biến đóng là tập phỗ biến mà không có tập nào bao nó là phỗ biến.

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

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

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

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

3/18/2024

BO MON KHOA HOC MAY TINH

BO MON KHOA HOC MAY TINH 9

BO MON KHOA HOC MAY TINH 10

9

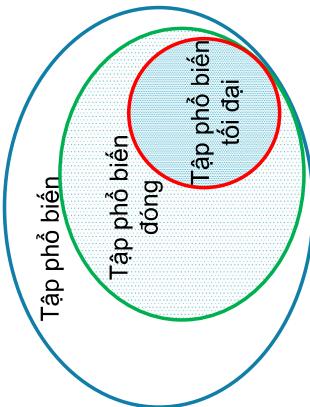
3/18/2024

BO MON KHOA HOC MAY TINH 10

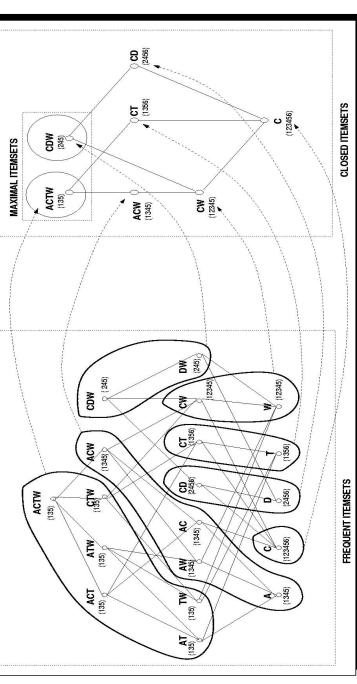
10

1.4 So sánh tập phỗ biến

- Số lượng tập phỗ biến phát sinh trong quá trình khai thác.



Tập phỗ biến, đóng và tối đại



3/18/2024

BO MON KHOA HOC MAY TINH

11

BO MON KHOA HOC MAY TINH 12

2. Khai thác tập phô biến

- **Input:** Tập các giao dịch T , với tập itemsets I
- **Output:** Tất cả các itemsets chứa trong I thỏa:
 - $\text{support} \geq \text{minsup}$

- Tham số:
 - $N = |T|$: số lượng giao dịch
 - $d = |I|$: số lượng itemsets 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 giao hàng.
 - The Web có hàng tỉ từ và hàng tỉ trang

3/18/2024

BỘ MÔN KHOA HỌC MÁY TINH

13

BO MÔN KHOA HỌC MÁY TINH

14

2. Khai thác tập phô biến

Quy tắc Apriori :

- Nếu một tập là phô biến, thì tất cả tập con của nó phải phô biến.
 - Nếu 1 tập không phô biến thì tất cả tập chứa nó không phô 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.

3/18/2024

BỘ MÔN KHOA HỌC MÁY TINH

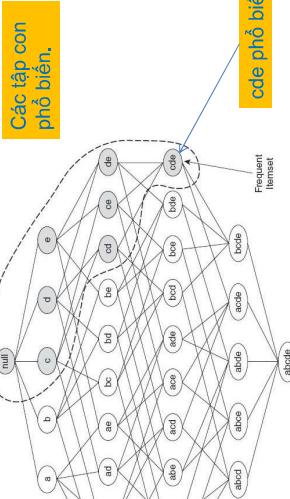
15

BO MÔN KHOA HỌC MÁY TINH

16

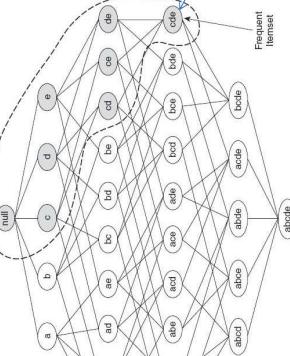
2. Khai thác tập phô biến

2. Khai thác tập phô biến



2. Khai thác tập phô biến

Các tập con
phô biến.



Quy tắc Apriori :

- Nếu một tập là phô biến, thì tất cả tập con của nó phải phô biến.
 - Nếu 1 tập không phô biến thì tất cả tập chứa nó không phô 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.

3/18/2024

BỘ MÔN KHOA HỌC MÁY TINH

15

BO MÔN KHOA HỌC MÁY TINH

16

2. Khai thác tập phô biến

2. Khai thác tập phô biến

- ❖ Thuật toán **Apriori** (*state-of-the art*) được đề xuất bởi R. Agrawal và R. Srikant vào năm 1994 để khai thác tập phô biến.
- ❖ Gọi C_k là các tập có k hàng mục. Thuật toán thực hiện như sau: $k = I$. F là tập hợp các tập phô 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} chưa tập con C_k không phô biến.
 - Bước 4: Thêm các tập C_k thỏa ngưỡng minsup vào F .

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

3/18/2024

BỘ MÔN KHOA HỌC MÁY TINH

17

BO MÔN KHOA HỌC MÁY TINH

18

2.1 Thuật toán Apriori

Đầu vào: D , cơ sở dữ liệu các giao tác.

Kết quả: L , tập các itemset phổ biến.

```
1:  $L_1$  = lầy tất cả các 1-itemset thỏa  $minsup$  trong  $D$ ;
2: for ( $k = 2$ ;  $L_{k-1} \neq \emptyset$ ;  $k++$ ) {
    $C_k$  = apriori_gen( $L_{k-1}$ );
   for each giao tác  $t \in D$  { // duyệt  $D$ 
       $C_t$  = subset( $C_k t$ ); //lấy tất cả các ứng viên của  $C_k$  có trong  $t$ 
      for each  $c \in C_t$ 
          $c.count++$ ;
      }
       $L_k$  =  $\{c \in C_k | c.count \geq minsup\}$ 
   }
   return  $L = \cup_k L_k$ ;
}
```

31/8/2024 BO MÔN KHOA HỌC MÁY TINH 19

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

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

BO MÔN KHOA HỌC MÁY TINH 20

2.1 Thuật toán Apriori

Ví dụ: (tiếp theo) $minsup = 50\%$

C ₂	Support	F ₂	Support	C ₃	Support	F ₃	Support	F ₄	Support
A,C	4	A,C	4	A,C,W	4	A,C	4	A,C,T,W	3
A,D	2	A,T	3	A,C,D	2	A,C,T	3	C,D,T,W	1
A,T	3	A,W	4	A,C,T	3	A,T,W	3		
A,W	4	C,D	4	A,T,W	3	C,D,W	3		
C,D	4	C,T	4	C,D,T	2	C,T,W	3		
C,T	4	C,W	5	C,D,W	3				
C,W	5	D,W	3	C,T,W	3				
D,T	2	T,W	3						
D,W	3								
T,W	3								

Tập con không phổ biến

BO MÔN KHOA HỌC MÁY TINH 21

BO MÔN KHOA HỌC MÁY TINH 22

2.1 Thuật toán Apriori

Ví dụ: (tiếp theo) $minsup = 50\%$

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

⇒ Các tập hàng 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 hàng mục phổ biến thỏa $minsup = 50\%$

2.2 Thuật toán Eclat

❖ Thuật toán Eclat (Equivalence Class Transformation) 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

Cấu trúc IT-tree và các lớp tương đương:
❖ Cho $X \subseteq I$ 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:

$X \times t(X)(Itemset \times Tidset)$ 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)

2.1 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	C ₁	Support	F ₁	Support
1	A, C, T, W	A	4	A	4
2	C, D, W	C	6	C	6
3	A, C, T, W	D	4	D	4
4	A, C, D, W	T	4	T	4
5	A, C, D, T, W	W	5	W	5
6	C, D, T				

BO MÔN KHOA HỌC MÁY TINH 20

2.2 Thuật toán Eclat

2.2 Thuật toán Eclat

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

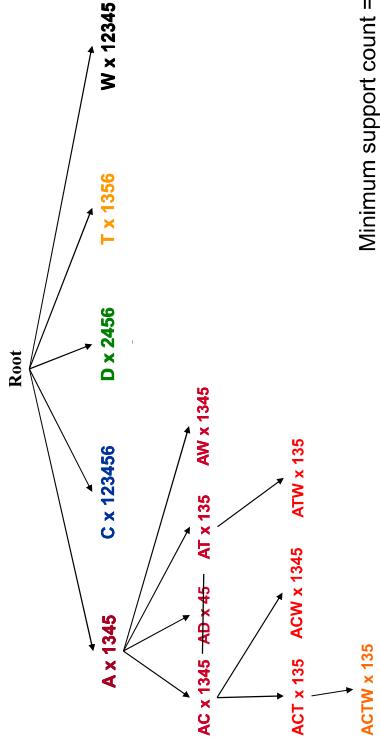
Kết quả: F , tập các itemset phổ biến.

0. Eclat($\{P\}$):

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

31/8/2024 BO MÔN KHOA HỌC MÁY TINH

25



Minimum support count = 3

26

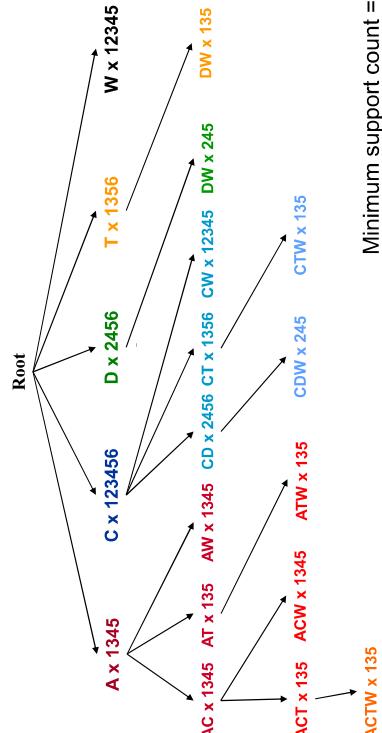
2.2 Thuật toán Eclat

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 hàng mục không phổ biến và tìm tập đóng bằng phương pháp dựa trên mối quan hệ của các tập hàng mục.

31/8/2024 BO MÔN KHOA HỌC MÁY TINH

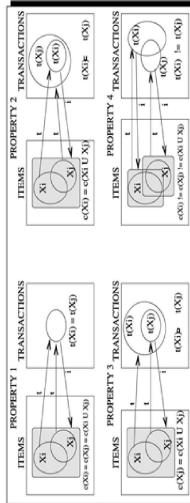
27



Minimum support count = 3

28

3. Thuật toán Charm



- Định lý 1: Đặt $X_i \times t(X_i)$ và $X_i \times t(X_j)$ là hai thành viên bất kỳ của một lớp $[P]$. Bốn thuộc tính sau là:
1. Nếu $t(X_i) = t(X_j)$, thì $c(X_i) = c(X_j)$.
 2. Nếu $t(X_i) \subset t(X_j)$, thì $c(X_i) \neq c(X_j)$, nhưng $c(X_i) = c(X_i \cup X_j)$.
 3. Nếu $t(X_i) \supset t(X_j)$, thì $c(X_i) \neq c(X_j)$, nhưng $c(X_i) = c(X_i \cup X_j)$.
 4. Nếu $t(X_i) \not\subset t(X_j)$ và $t(X_i) \not\supset t(X_j)$, thì $c(X_i) \neq c(X_j)$.

31/8/2024 BO MÔN KHOA HỌC MÁY TINH

29

3. Thuật toán Charm

Đầu vào: CSDL D , $minsup$

Kết quả: tập FC 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:      $P_i = P \cup l_i$  and  $[P_i] = O$   
6:     for each  $l_j \times t(l_j)$  in  $[P]$ , with  $j > i$   
7:          $X = l_i \cup l_j$  and  $Y = t(l_i) \cap t(l_j)$   
8:         CHARM-PROPERTY ( $X \times Y$ ,  $l_i$ ,  $l_j$ ,  $P_i$ ,  $[P_i]$ )  
9:         SUBSUMPTION-CHECK ( $C$ ,  $P_i$ )  
10:         CHARM-EXTEND ( $[P_i]$ ,  $C$ )  
11: delete [ $P_i$ ]
```

31/8/2024 BO MÔN KHOA HỌC MÁY TINH

30

3. Thuật toán Charm

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

```
12: if  $\sigma(X) \geq minsup$  then
13:   if  $t(X_j) = t(X)$  then (1)
14:     remove  $X_j$  from  $[P]$ 
15:      $P_i = P_i \cup X_j$ 
16:   else if  $t(X_j) \subset t(X)$  then (2)
17:      $P_i = P_i \cup X_j$ 
18:   else if  $t(X_j) \supset t(X)$  then (3)
19:     remove  $X_j$  from  $[P]$ 
20:     Add  $X \times Y$  to  $[P_i]$ 
21:   else if  $t(X_j) \neq t(X)$  then (4)
22:     Add  $X \times Y$  to  $[P_i]$ 
```

31/8/2024

BO MON KHOA HOC MAY TINH

3. Thuật toán Charm

```
Minimum support count = 3
Root
  ↗ A x 1345 - ACW x 1345
  ↗ A x 1345 - ACTW x 135
  ↗ C x 123456 - CD x 2456
  ↗ C x 123456 - CT x 1356
  ↗ D x 2456 - CW x 12345
  ↗ D x 2456 - CDT x 1356
```

32

3. Thuật toán Charm

```
Minimum support count = 3
Root
  ↗ A x 1345 - ACW x 1345
  ↗ A x 1345 - ACTW x 135
  ↗ C x 123456 - CD x 2456
  ↗ C x 123456 - CT x 1356
  ↗ D x 2456 - CW x 12345
  ↗ D x 2456 - CDT x 1356
```

$C \times 123456 \supset D \times 2456$ (3)

⇒ Thêm CD , xóa D

Tương tự các trường hợp còn lại.

33

3. Thuật toán Charm

```
Minimum support count = 3
Root
  ↗ A x 1345 - ACW x 1345
  ↗ A x 1345 - ACTW x 135
  ↗ C x 123456 - CD x 2456
  ↗ C x 123456 - CT x 1356
  ↗ D x 2456 - CW x 12345
  ↗ D x 2456 - CDT x 1356
```

34

3. Thuật toán Charm

```
Minimum support count = 3
Root
  ↗ A x 1345 - ACW x 1345
  ↗ A x 1345 - ACTW x 135
  ↗ C x 123456 - CD x 2456
  ↗ C x 123456 - CT x 1356
  ↗ D x 2456 - CW x 12345
  ↗ D x 2456 - CDT x 1356
```

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

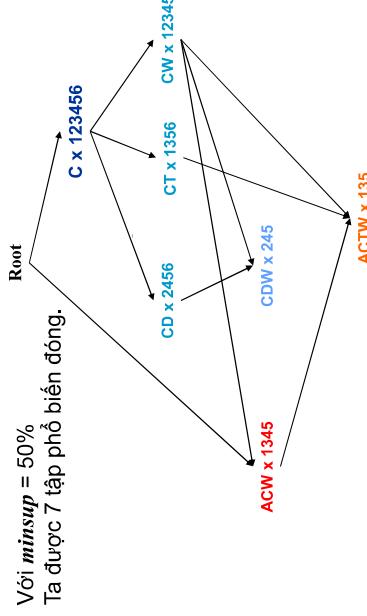
35

3. Thuật toán Charm

```
Minimum support count = 3
Root
  ↗ A x 1345 - ACW x 1345
  ↗ A x 1345 - ACTW x 135
  ↗ C x 123456 - CD x 2456
  ↗ C x 123456 - CT x 1356
  ↗ D x 2456 - CW x 12345
  ↗ D x 2456 - CDT x 1356
```

32

3. Thuật toán Charm



4. Khai thác tập phô biến tối đa

- ❖ K. Gouda và M. J.Zaki đề xuất Thuật toán GenMax để tìm tập phô biến tối đa 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 phô biến tối đa 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).

3/18/2024 BO MON KHOA HOC MAY TINH 39

3/18/2024 BO MON KHOA HOC MAY TINH 40

3/18/2024 BO MON KHOA HOC MAY TINH 41

3/18/2024 BO MON KHOA HOC MAY TINH 42

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 phô biến được rút trích.

4. Thuật toán GenMax

Thuật toán FI-backtrack

Đầu vào:

- I_t tập các itemssets có độ dài t .
- C_t tập những items có thể kết hợp với I_t .
- I là độ dài của itemset.

Kết quả: itemset phô biến

```
FI-backtrack(It, Ct, I)
1: for each x ∈ Ct
2:   It+1 = It ∪ {x} //đồng thời thêm It+1 vào FI
3:   Pt+1 = {y: y ∈ Ct and y > x}
4:   Ct+1 = FI-combine (It+1, Pt+1)
5:   FI-backtrack (It+1, Ct+1, It+1)
```

4. Thuật toán GenMax

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

FI-combine (I_{t+1}, P_{t+1})

```
1: C = Ø
2: for each y ∈ Pt+1
3:   if It+1 ∪ {y} là phô biến
4:     C = C ∪ {y} //sắp xếp lại C nếu cần
5: return C;
```

3/18/2024 BO MON KHOA HOC MAY TINH 43

3/18/2024 BO MON KHOA HOC MAY TINH 44

3/18/2024 BO MON KHOA HOC MAY TINH 45

4. Thuật toán GenMax

- ❖ Để tìm tập phô biến tối đa, chỉ cần áp dụng điều kiện loại bỏ đi những tập phô biến không tối đa.

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

3/18/2024

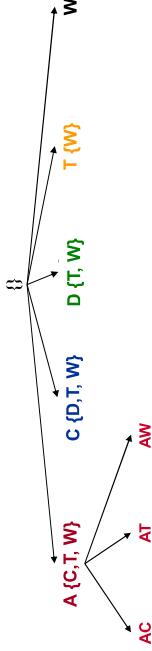
BỘ MÔN KHOA HỌC MÁY TINH

43

44

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 $C_1 = \{AC, AT, AW\} \neq \emptyset$
- Gọi đệ quy hàm backtrack.

45

4. Thuật toán GenMax

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

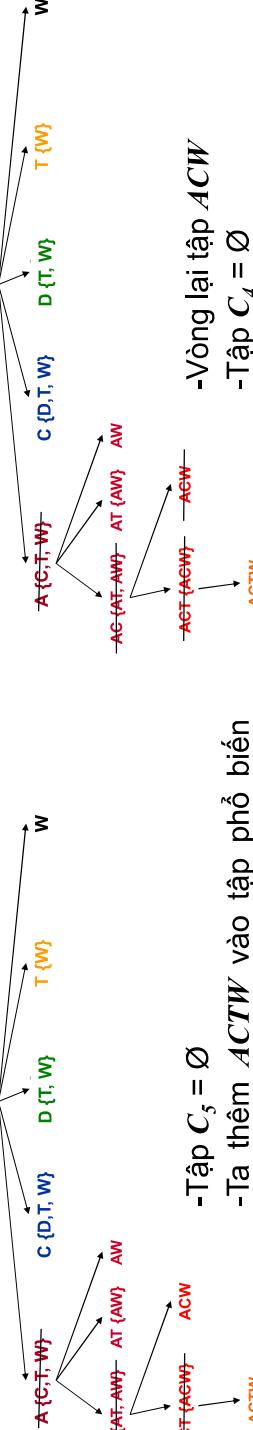


- Tiếp tục kết hợp AC với AT, AW.
- Tập $C_2 = \{ACT, ACW\} \neq \emptyset$
- Gọi đệ quy hàm backtrack.

46

4. Thuật toán GenMax

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



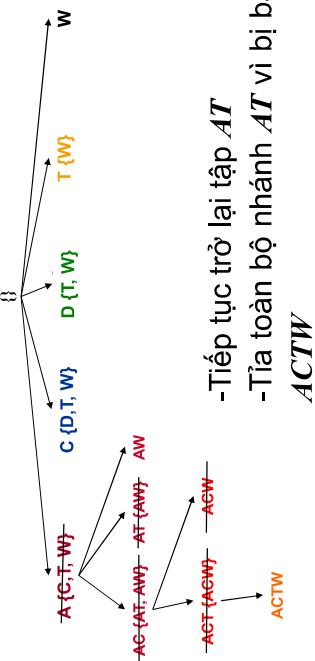
- Vòng lại tập ACW
- Tập $C_3 = \emptyset$
- Ta thêm ACTW vào tập phô biến tối đa.
- Loại ACW vì bị bao bởi ACTW

47

48

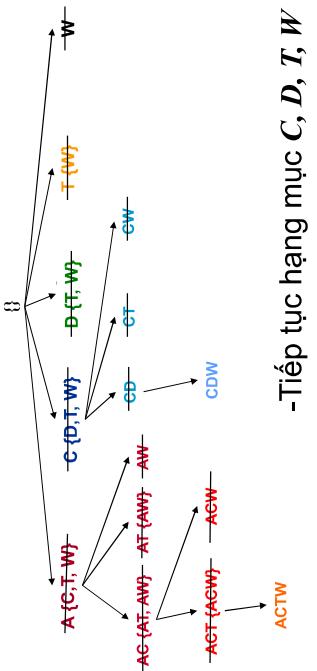
4. Thuật toán GenMax

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



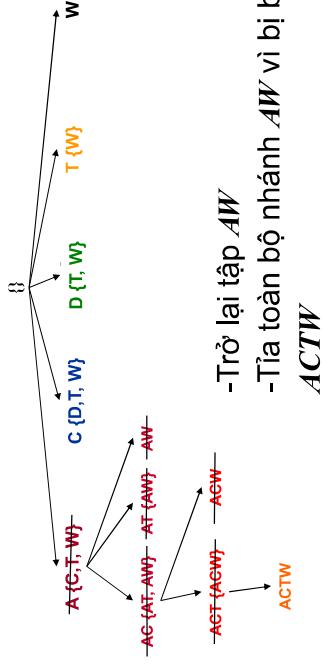
4. Thuật toán GenMax

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



4. Thuật toán GenMax

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



5. Nhận xét

$minsup = 50\%$

19 Tập phô biến

BO MON KHOA HOC MAY TINH

31/8/2024 50

5. Nhận xét

$minsup = 50\%$

7 Tập phô biến đóng

BO MON KHOA HOC MAY TINH

31/8/2024 51

5. Nhận xét

$minsup = 50\%$

2 Tập phô biến tối đại

BO MON KHOA HOC MAY TINH

31/8/2024 52

5. Nhận xét

- ❖ Mỗi quan hệ giữa các tập phô biến như sau:
 $M \subseteq C \subseteq F$.
- ❖ Tập phô biến đóng thê hiện đầy đủ thông tin của tất cả các tập phô biến cùng với độ hỗ trợ chính xác của nó.
- ❖ Luật kết hợp rút trích từ tập phô biến đóng sẽ nhỏ gọn hơn, dễ quản lý, phân tích.
- ❖ Khai thác tập phô 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

Nội dung

KHAI THÁC ITEMSET PHỎ BIÊN SỬ DỤNG DIFFSET

DATA MINING

HCMUS - 2023



6/6/2023

BO MÔN KHOA HỌC MÁY TINH

1. Giới thiệu

- Diffset là một cách biểu diễn dữ liệu được đưa ra bởi M. J. Zaki và K. Gouda.
- Phương pháp này theo vết sụ thay đổi trong tidset của các ứng viên sau khi kết hợp với các ứng viên dùng phát sinh ra chúng.

- Mục tiêu của hướng tiếp cận là tiết kiệm bộ nhớ sử dụng để lưu các tidset và tăng hiệu xuất thực thi thuật toán.

6/6/2023

BO MÔN KHOA HỌC MÁY TINH

4

BO MÔN KHOA HỌC MÁY TINH

2

2. Đặt vấn đề

Khai thác itemset phỏ biến.

Transaction	Items	Minimum Support = 80%	Minimum Support Count = 5
1	A, C, T, W		
2	C, D, W		
3	A, C, T, W		
4	A, C, D, W	{C}	100%
5	A, C, D, T, W	{W}	83%
6	C, D, T	{CW}	83%

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

Transaction	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

a. Độ support

A	C	D	T	W
1	1	0	1	1
2	0	1	0	1
3	1	1	0	1
4	1	1	0	1
5	1	1	1	1
6	0	1	1	0

c. Bitvector

b. Tidset

6/6/2023

BO MÔN KHOA HỌC MÁY TINH

5

BO MÔN KHOA HỌC MÁY TINH

6

3. Các phương pháp biểu diễn

3. Các phương pháp biểu diễn

a. Lưu giá trị support

- Mỗi itemset sẽ lưu kèm với một giá trị support. Phải đếm support cho từng itemset.

b. Sử dụng Tidset (Mã giao tac)

- Tổn bộ nhớ lưu các mã giao tac cho từng itemset.
- Duyệt CSDL một lần. Tính support bằng phép toán giao (intersection).

c. Sử dụng Bitvector (vector nhị phân)

- Tính toán support nhanh bằng phép toán AND.

4. Thuật toán Eclat

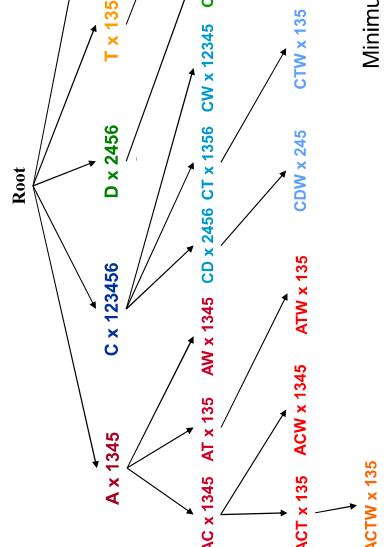
0. Eclat([P]):

- for all $X_i \in [P]$ do
- $T_i = \emptyset$
- for all $X_j \in [P]$, with $j > i$ do
 - $R = X_i \cup X_j$;
 - $t(R) = t(X_i) \cap t(X_j)$;
 - if $\sigma(R) \geq \min_sup$ then
 - $T_i = T_i \cup \{R\}$; $F_{|R|} = F_{|R|} \cup \{R\}$;
- for all $T_i \neq \emptyset$ do Eclat(T_i);

8

BO MON KHOA HOC MAY TINH

4. Thuật toán Eclat



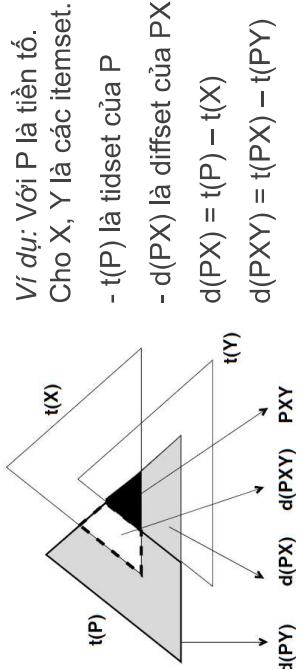
9

6/6/2023

BO MON KHOA HOC MAY TINH

5. Sử dụng Diffset (1)

- Lưu sự thay đổi Tidset của từng đối tượng cùng một lớp hoặc cùng tiền tố (prefix).



10

BO MON KHOA HOC MAY TINH

5. Sử dụng Diffset (2)

- Tính diffset của PXY:

$$d(PXY) = t(PX) - t(PY)$$

Tuy nhiên chúng ta chỉ lưu diffset của PX và PY là $d(PX), d(PY)$.

$$d(PXY) = t(PY) + t(P) - t(PX)$$

$$d(PXY) = (t(P) - t(PY)) - (t(P) - t(PX))$$

$$\Rightarrow \boxed{d(PXY) = d(PY) - d(PX)}$$

- Tính support PXY: $\sigma(PXY) = \sigma(PX) - |d(PXY)|$

5. Sử dụng Diffset (3)

Item	TIDSET	Item	DIFFSET
A	1 3 4 5	A	2 6
C	1 2 3 4 5 6	C	
D	2 4 5 6	D	1 3
T	1 3 5 6	T	2 4
W	1 2 3 4 5	W	6

11

BO MON KHOA HOC MAY TINH

$$\begin{aligned} d(AC) &= d(C) - d(A) = \{ \emptyset \} - \{ 2, 6 \} = \{ \emptyset \} \\ \Rightarrow \sigma(AC) &= \sigma(A) - |d(AC)| = 4 - 0 = 4 \end{aligned}$$

12

BO MON KHOA HOC MAY TINH

5. Sử dụng Diffset (4)

Item	TIDSET	Item	DIFFSET
A	1 3 4 5	A	2 6
C	1 2 3 4 5 6	C	
D	2 4 5 6	D	1 3
T	1 3 5 6	T	2 4
W	1 2 3 4 5	W	6

Ví dụ: Tính support của itemset AD

$$d(AD) = d(D) - d(A) = \{1, 3\} - \{2, 6\} = \{1, 3\}$$

$$\Rightarrow \sigma(AD) = \sigma(A) - |d(AD)| = 4 - 2 = 2$$

5. Sử dụng Diffset (5)

Item	TIDSET	Item	DIFFSET
A	1 3 4 5	A	2 6
C	1 2 3 4 5 6	C	
D	2 4 5 6	D	1 3
T	1 3 5 6	T	2 4
W	1 2 3 4 5	W	6

Ví dụ: Tính support của itemset ACD

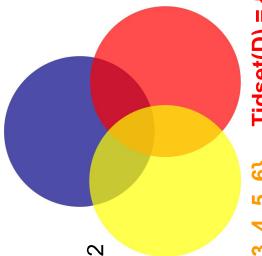
$$d(ACD) = d(AD) - d(AC) = \{1, 3\} - \{\emptyset\} = \{1, 3\}$$

$$\Rightarrow \sigma(ACD) = \sigma(AC) - |d(ACD)| = 4 - 2 = 2$$

5. Sử dụng Diffset (5)

Item	TIDSET	Item	DIFFSET
A	1 3 4 5	A	2 6
C	1 2 3 4 5 6	C	
D	2 4 5 6	D	1 3
T	1 3 5 6	T	2 4
W	1 2 3 4 5	W	6

$$Tidset(A) = \{1, 3, 4, 5\}$$



$$Tidset(D) = \{2, 4, 5, 6\}$$

$$Tidset(C) = \{1, 2, 3, 4, 5, 6\}$$

5. Sử dụng Diffset (5)

Item	TIDSET	Item	DIFFSET
A	1 3 4 5	A	2 6
C	1 2 3 4 5 6	C	
D	2 4 5 6	D	1 3
T	1 3 5 6	T	2 4
W	1 2 3 4 5	W	6

$$d(A) = \{2, 6\} \Rightarrow sup = 4$$

$$d(C) = \emptyset \Rightarrow sup = 6$$

$$d(D) = \{1, 3\} \Rightarrow sup = 4$$

$$d(T) = \{2, 4\} \Rightarrow sup = 4$$

$$d(W) = \{6\} \Rightarrow sup = 5$$

$$*d(AC) = \{\} \Rightarrow sup = 4 - 0 = 4$$

$$d(AD) = \{1, 3\} \Rightarrow sup = 4 - 2 = 2$$

$$d(AT) = \{4\} \Rightarrow sup = 4 - 1 = 3$$

$$d(AW) = \{\} \Rightarrow sup = 4 - 0 = 4$$

$$*d(CD) = \{1, 3\} \Rightarrow sup = 6 - 2 = 4$$

$$d(CT) = \{2, 4\} \Rightarrow sup = 6 - 2 = 4$$

$$d(CW) = \{6\} \Rightarrow sup = 6 - 1 = 5$$

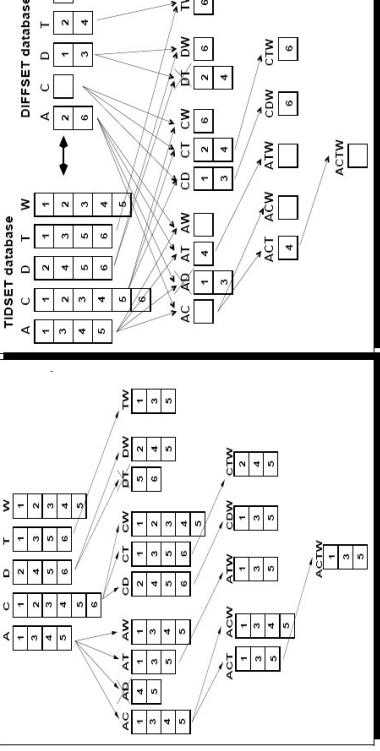
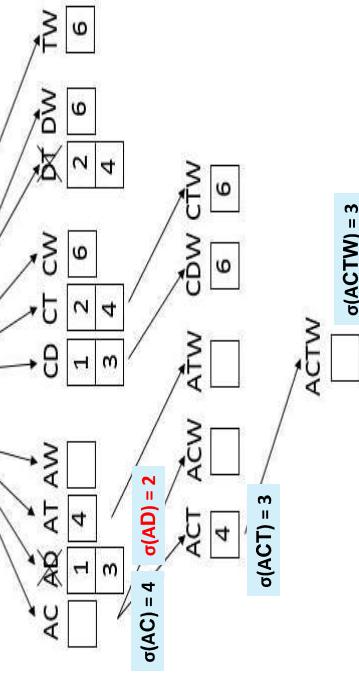
$$d(DT) = \{2, 4\} \Rightarrow sup = 4 - 2 = 2$$

$$d(DW) = \{6\} \Rightarrow sup = 4 - 1 = 3$$

$$d(ACD) = \{1, 3\} \Rightarrow sup = 4 - 2 = 2$$

$$\Rightarrow sup = 4 - 2 = 2$$

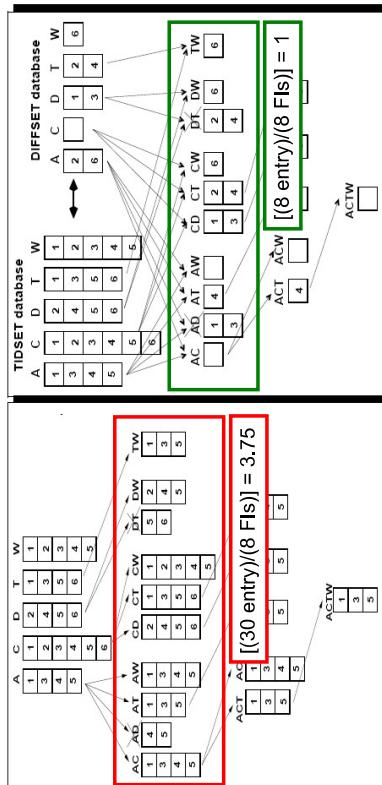
5. Sử dụng Diffset (6) (So sánh Tidsets và Diffsets)



5. Sử dụng Diffset (5)

(So sánh Tidsets và Diffsets)

5. Sử dụng Diffset (7)



19

6/6/2023

BO MON KHOA HOC MAY TINH

20

5. Sử dụng Diffset (7)

Giảm tỷ lệ (Reduction Ratio)

- Cho lớp P

Gọi PX và PY là lớp thành viên với $t(PX)$ và $t(PY)$

Xét Itemset mới PXY trong lớp PX
 PXY có thể được lưu trữ $t(PXY)$ hoặc $d(PXY)$

- Định nghĩa: giảm tỷ lệ $r = t(PXY)/d(PXY)$

Đổi với diffset sẽ có lợi nếu $r \geq 1$
hoặc $t(PXY)/d(PXY) \geq 1$

6/6/2023

BO MON KHOA HOC MAY TINH

21

6/6/2023

BO MON KHOA HOC MAY TINH

22

5. Sử dụng Diffset (7)

$$r = t(PXY) / d(PXY)$$

- Thay $d(PXY) \Rightarrow t(PXY)/(t(PX) - t(PY)) \geq 1$
- Khi $t(PX) - t(PY) = t(PX) - t(PXY)$
Ta có $t(PXY) = (t(PX) - t(PXY))$
- Chia cho $t(PXY)$ được $c \frac{1}{t(PXY)-1} \geq 1$
- Sau khi đơn giản được $t(PX)/t(PXY) \leq 2$

⇒ Điều đó có nghĩa là nếu $d(PXY) \geq t(PX)$
bằng ít nhất $1/2$ của PX thì ta chuyển sang sử dụng diffset.

6/6/2023

BO MON KHOA HOC MAY TINH

22

6. Thuật toán dEclat

- Thuật toán áp dụng diffset vào phương pháp Eclat (state-of-the-art) trước đó sử dụng tidset.
- Thuật toán duyệt theo chiều sâu trước (DFS). Bắt đầu với các diffset của các items phô biến.
- Vòng lặp đệ quy để tìm tất cả các itemset phô biến ở cấp hiện tại. Tiến trình lặp lại cho đến khi tất cả các itemset phô biến được khai thác.

6. Thuật toán dEclat

0. dEclat($[P]$):

- for all $X_i \in [P]$ do
 - for all $X_j \in [P]$, with $j > i$ do
 - $R = X_i \cup X_j$;
 - $d(R) = d(X_j) - d(X_i)$;
 - if $\sigma(R) \geq \min_sup$ then
 - $T_i = T_i \cup \{R\}$; // T_i initially empty
- if $T_i \neq \emptyset$ then $\text{dEclat}(T_i)$;

23

BO MON KHOA HOC MAY TINH

6/6/2023

24

6. Thuật toán dCharm

- ❖ Thuật toán áp dụng diffset vào thuật toán Charm thay vì sử dụng tidsset.
- ❖ Thuật toán duyệt theo chiều sâu trước (DFS). Bắt đầu với các diffset của các items phô biến.
- ❖ Thuật toán sử dụng các bước tia nhánh dựa vào mối quan hệ của các tập con

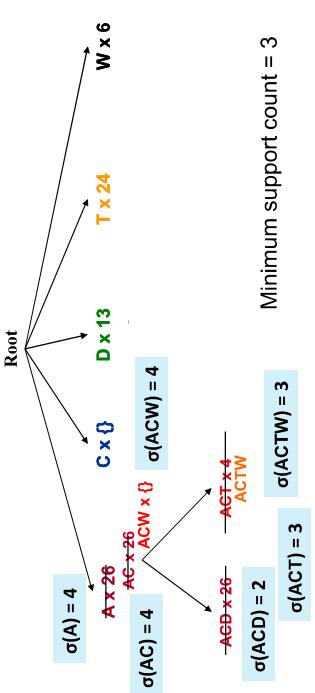
6/6/2023 BO MON KHOA HOC MAY TINH

26

BO MON KHOA HOC MAY TINH

26

6. Thuật toán dGenMax



❖ Thuật toán sử dụng tiến trình **backtrack** để tìm kiếm mẫu tối đại.

❖ Các cải tiến

- Sắp item theo thứ tự **tăng dần** theo kích thước và độ **support** (i.e. đầu tiên khám phá item có kích thước nhỏ trước, ii. Bỏ một node càng sớm càng tốt trong cây tìm kiếm).
- Kiểm tra các tập bao (**superset**) của itemset đang xét.
- **CSDL** theo **chiều dọc** tối ưu việc kiểm tra phỗ biến bằng cách sử dụng tidsset, hoặc cải tiến hơn nữa là **diffsets**.

❖ Bộ nhớ

- Lưu trữ nhiều nhất $k = m + I$ tidssets (diffsets) trong bộ nhớ, với m là độ dài của tập kết hợp dài nhất và I là độ dài của itemset tối dài có chiều dài lớn nhất.

27

BO MON KHOA HOC MAY TINH

27

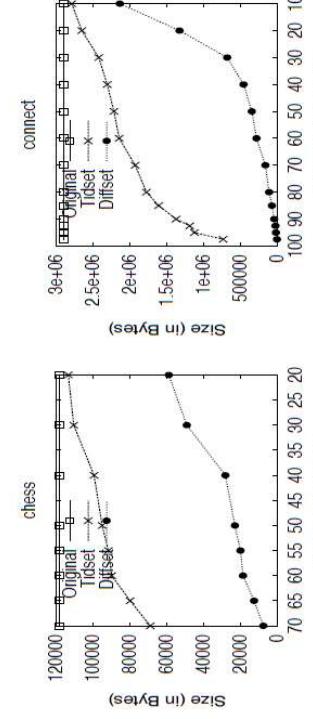
6. Thuật toán dGenMax

6. Thuật toán dGenMax

Procedure *Extend*(I, X, Y)

```
1.  $I$  is the itemset to be extended.  $X$  is the set of items
   // that can be added to  $I$ , i.e., the combine-set.
   //  $Y$  is the set of relevant maximal itemsets found so far
   // i.e., all maximal itemsets which contain  $I$ .
2. Sort items in  $F_1$  in INCREASING cardinality of  $c(i)$ 
   and then INCREASING  $\sigma(i)$ .
3. Sort each  $c(i)$  in order of  $F_1$ .
4.  $c(i) = c(i) - \{j : j < i \text{ in sorted order of } F_1\}$ .
5.  $M = \{\}$ ; // Maximal Frequent Itemsets.
6.  $M = \{\}$ ; // Maximal Frequent Itemsets.
7. for each  $i \in F_1$  do
8.    $Z = \{x \in M : i \in x\}$ 
   for each  $j \in c(i)$  do
9.    $H = \{x : x \text{ is } j \text{ or } x \text{ follows } i \text{ in } c(i)\}$ 
   if  $H$  has a super set in  $Z$  then break
10.   $I = \{i, j\}$ 
11.   $X = c(i) \cap c(j); d(X) = t(i) - t(j)$ 
12.   $Y = \{x \in Z : j \in x\}$ 
13.   $NewI = I \cup \{j\}; d(NewI) = d(j) - d(I)$ 
   if  $(NewI \text{ is frequent})$  then
    14.   $NewX = X \cap c(j)$ 
    15.   $Extend(I, X, Y)$ 
    16.   $Y = Y \cup \{NewY\}$ 
17.   $Z = Z \cup Y$ 
18.   $M = M \cup Z$ 
19. Return  $M$ 
```

7. So sánh Tidsset và Diffset



So sánh bộ nhớ sử dụng
trên dữ liệu đặc Chess và Connect

6/6/2023 BO MON KHOA HOC MAY TINH

29

6/6/2023

29

6. Thuật toán dCharm

- ❖ Thuật toán áp dụng diffset vào thuật toán Charm thay vì sử dụng tidsset.

- ❖ Thuật toán duyệt theo chiều sâu trước (DFS). Bắt đầu với các diffset của các items phô biến.

- ❖ Thuật toán sử dụng các bước tia nhánh dựa vào mối quan hệ của các tập con

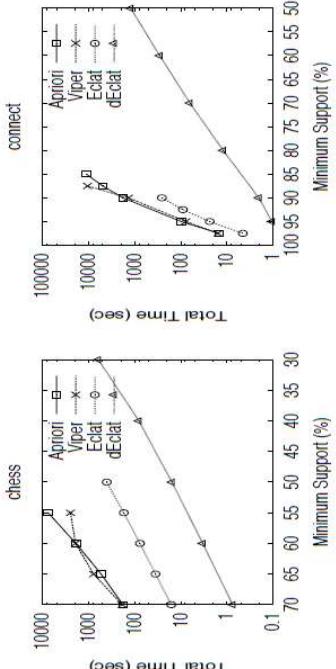
0. dCharm([P]):

```
1. for all  $X_i \in [P]$  do
2.   for all  $X_j \in [P]$ , with  $j > i$  do
3.      $R = X_i \cup X_j;$ 
4.      $d(R) = d(X_i) - d(X_j);$ 
5.     if  $d(R) \geq \min_{sup}$  then
       Remove  $X_j$  from  $[P]$ ;
       Replace all  $X_i$  with  $R$ ;
6.   else if  $d(R) > d(X_j)$  then
       Replace all  $X_i$  with  $R$ ;
7.   else if  $d(R) \subset d(X_j)$  then
       Remove  $X_i$  from  $[P]$ ;
8.   else if  $d(R) \subset d(X_i)$  then
       Replace all  $X_j$  with  $R$ ;
9.   else if  $d(R) \neq d(X_i)$  then
       Add  $R$  to  $NewN$ ;
10.  else if  $d(R) \neq d(X_j)$  then
      Add  $R$  to  $NewN$ ;
11. if  $NewN \neq \emptyset$  then dCharm( $NewN$ );
```

```
12. if  $NewN \neq \emptyset$  then dCharm( $NewN$ );
```

7. So sánh Tidset và Diffset

7. So sánh Tidset và Diffset



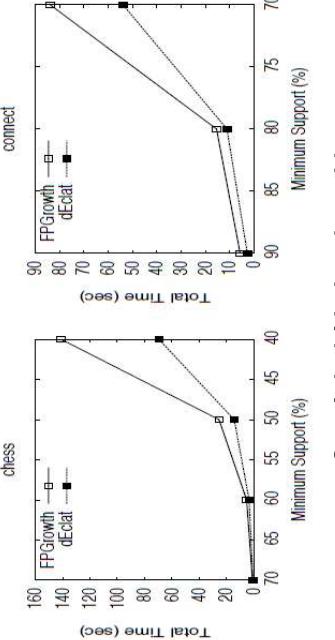
6/6/2023 BO MÔN KHOA HỌC MÁY TINH 31

BO MÔN KHOA HỌC MÁY TINH

32 BO MÔN KHOA HỌC MÁY TINH

7. So sánh Tidset và Diffset

7. So sánh Tidset và Diffset



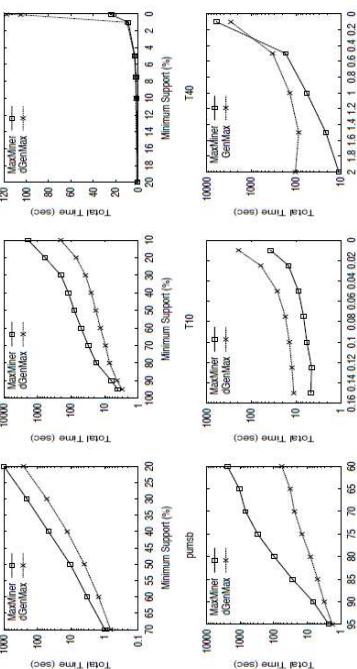
6/6/2023 BO MÔN KHOA HỌC MÁY TINH 33

BO MÔN KHOA HỌC MÁY TINH

34 BO MÔN KHOA HỌC MÁY TINH

7. So sánh Tidset và Diffset

7. So sánh Tidset và Diffset



35 BO MÔN KHOA HỌC MÁY TINH

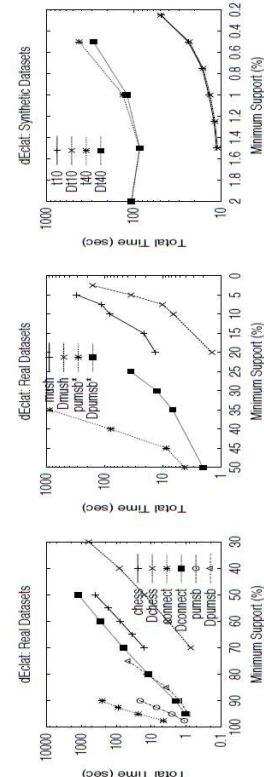
BO MÔN KHOA HỌC MÁY TINH

36 BO MÔN KHOA HỌC MÁY TINH

Kích thước trung bình vòng lặp: Tidset vs Diffset

7. So sánh Tidset và Diffset

8. Nhận xét



So sánh thời gian thực thi trên CSDL dày đặc (Real datasets) và mỏng (Synthetic datasets)

6/6/2023 BO MÔN KHOA HỌC MÁY TINH 37

- ❖ Diffset giảm đáng kể kích thước bộ nhớ cần để lưu trữ trực tiếp kết quả.
- ❖ Diffset tăng hiệu quả thực thi khi đưa vào phương pháp khai thác dữ liệu theo chiều dọc.
- ❖ Diffset cung cấp tầm quan trọng về cải tiến hiệu xuất so với các phương pháp tốt trước đó.

Tài liệu tham khảo

- [1] M.J. Zaki and K. Gouda, *Fast Vertical Mining Using Diffsets*, Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, Aug. 2003.
- [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 C.-J. Hsiao, *Efficient Algorithms for Mining Closed Itemsets and their Lattice Structure*, IEEE Transactions on Knowledge and Data Engineering, 17(4), Apr 2005, pp. 462-478.
- [4] 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.

6/6/2023 BO MÔN KHOA HỌC MÁY TINH 39

Thanks for your listening !!
Q & A

6/6/2023 BO MÔN KHOA HỌC MÁY TINH 38