

KHAI THÁC ITEMSET PHỔ BIẾN SỬ DỤNG DIFFSET

DATA MINING

HCMUS - 2023



6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

1

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

2

Nội dung

1. Giới thiệu
2. Đặt vấn đề
3. Các phương pháp biểu diễn dữ liệu
4. Thuật toán Eclat
5. Sử dụng Diffset
6. Thuật toán dEclat, dCharm, dGenMax
7. So sánh Tidset và Diffset
8. Nhận xét

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 sự 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 suất thực thi thuật toán.

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

3

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

4

2. Đặt vấn đề

Khai thác itemset phổ biến.

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

Minimum Support = 80%
Minimum Support Count = 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

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

a. Độ support

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

b. Tidset

c. Bitvector

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

5

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

6

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 tác)

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

c. Sử dụng Bitvector (vector nhị phân)

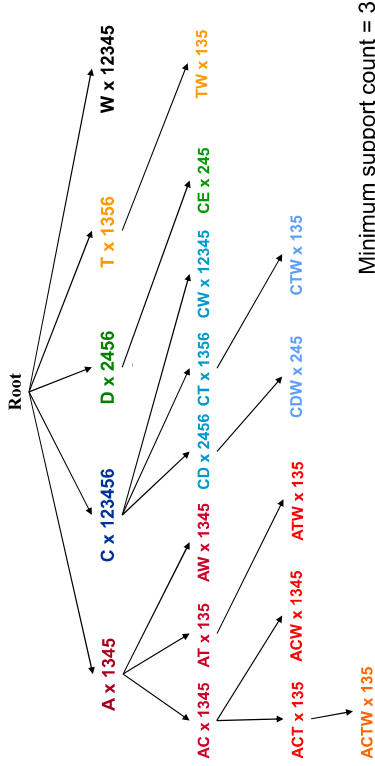
- Tính toán support nhanh bằng phép toán AND.

4. Thuật toán Eclat

0. Eclat([P]):

- for all $X_j \in [P]$ 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 $\sigma(R) \geq \text{min_sup}$ then
- $T_i = T_i \cup \{R\}$; $F_{|R|} = F_{|R|} \cup \{R\}$;
- for all $T_i \neq \emptyset$ do Eclat(T_i);

4. Thuật toán Eclat



Minimum support count = 3

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 $d(PX)$, $d(PY)$.

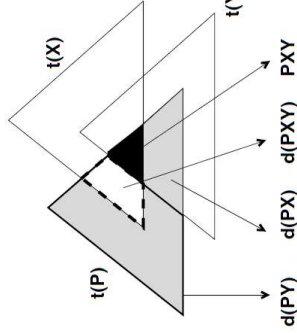
$$d(PXY) = t(PX) - t(PY) + t(P) - t(P)$$

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

$$\Rightarrow d(PXY) = d(PY) - d(PX)$$

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

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



Ví dụ: Với P là tiền tố.
Cho X, Y là các itemset.

- $t(P)$ là tidset của P

- $d(PX)$ là diffset của PX

$$d(PX) = t(P) - t(X)$$

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

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

10

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



Ví dụ: Tính support của itemset AC

$$d(AC) = d(C) - d(A) = \{\emptyset\} - \{2, 6\} = \{\emptyset\}$$

$$\Rightarrow \sigma(AC) = \sigma(A) - |d(AC)| = 4 - 0 = 4$$

6/6/2023

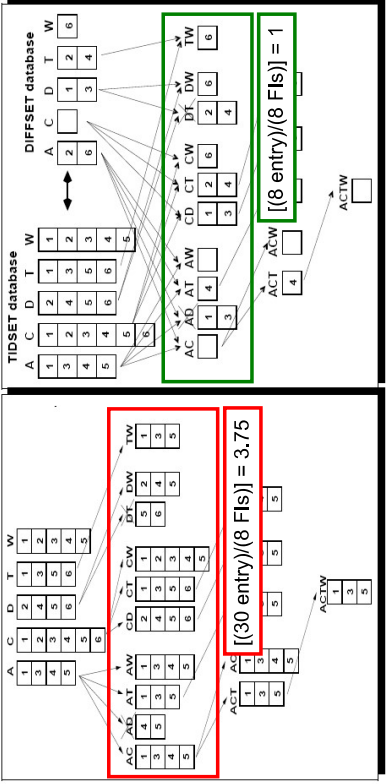
BỘ MÔN KHOA HỌC MÁY TÍNH

11

BỘ MÔN KHOA HỌC MÁY TÍNH

12

5. Sử dụng Tidset và Diffsets (5)



19

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

20

5. Sử dụng Diffset (7)

Thông thường trong quá trình thực thi sẽ có một điểm giao để chuyển đổi giữa tidset và diffset.

- ❖ Sử dụng diffset đối với CSDL dày đặc.
 - ❖ Bắt đầu với tidset đối với CSDL thưa thớt, và ở các giai đoạn sau có thể chuyển sang diffset.
- ⇒ **Khi nào nên chuyển đổi giữa tidset và diffset?**

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 $r = t(PXY) / d(PXY)$
- Đối với diffset sẽ có lợi nếu như $r \geq 1$ hoặc $t(PXY) / d(PXY) \geq 1$

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

21

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

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 $\frac{1}{t(PX) - t(PXY)} \geq 1$
- Sau khi đơn giản được $t(PX) / t(PXY) \leq 2$

⇒ **Điều đó có nghĩa là nếu độ hỗ trợ của PXY bằng ít nhất 1/2 của PX thì ta chuyển sang sử dụng diffset.**

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 phổ biến.
- ❖ Vòng lặp đệ quy để tìm tất cả các itemset phổ biến ở cấp hiện tại. Tiến trình lặp lại cho đến khi tất cả các itemset phổ biến được khai thác.

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

23

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

24

6. Thuật toán dEclat

- 0. $dEclat([P])$:
- 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. $d(R) = d(X_j) - d(X_i)$;
- 5. **if** $\sigma(R) \geq min_sup$ **then**
- 6. $T_i = T_i \cup \{R\}$; $// T_i$ initially empty
- 7. **if** $T_i \neq \emptyset$ **then** $dEclat(T_i)$;

6. Thuật toán dCharm

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

6. Thuật toán dCharm

```

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  $\sigma(R) \geq \text{min\_sup}$  then
6.       if  $d(X_j) = d(X_i)$  then
7.         Remove  $X_j$  from  $[P]$ ;
8.         Replace all  $X_j$  with  $R$ ;
9.       else if  $d(X_j) > d(X_i)$  then
10.        Replace all  $X_j$  with  $R$ ;
11.       else if  $d(X_j) < d(X_i)$  then
12.        Remove  $X_j$  from  $[P]$ ;
13.        Add  $R$  to  $\text{NewN}$ ;
14.       else if  $d(X_i) \neq d(X_j)$  then
15.        Add  $R$  to  $\text{NewN}$ ;
16.     Subsumption-Check( $C, R$ );
17. if  $\text{NewN} \neq \emptyset$  then dCharm( $\text{NewN}$ );

```

6/6/2023

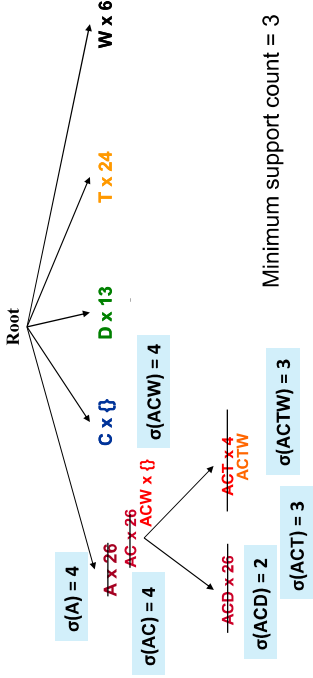
BỘ MÔN KHOA HỌC MÁY TÍNH

25

BỘ MÔN KHOA HỌC MÁY TÍNH

26

6. Thuật toán dCharm



Minimum support count = 3

27

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

28

6. Thuật toán dGenMax

```

GenMax (Dataset  $T$ ):
1. Calculate  $F_1$ . Calculate  $F_2$ .
2. For each item  $i \in F_1$  calculate  $c(i)$ , its combine-set.
3. Sort items in  $F_1$  in INCREASING cardinality of  $c(i)$  and then INCREASING  $\sigma(i)$ .
4. Sort each  $c(i)$  in order of  $F_1$ .
5.  $c(i) = c(i) - \{j : j < i \text{ in sorted order of } F_1\}$ .
6.  $M = \{\}; // \text{Maximal Frequent Itemsets.}$ 
7. for each  $i \in F_1$  do
8.    $Z = \{x \in M : i \in x\}$ 
9.   for each  $j \in c(i)$  do
10.     $H = \{x : x \text{ is } j \text{ or } x \text{ follows } j \text{ in } c(i)\}$ 
11.    if  $H$  has a super set in  $Z$  then break
12.     $I = \{i, j\}$ 
13.     $X = c(i) \cap c(j); d(X) = t(i) - t(j)$ 
14.     $Y = \{x \in Z : j \in x\}$ 
15.     $Extend(I, X, Y)$ 
16.     $Z = Z \cup Y$ 
17.     $M = M \cup Z$ 
18.   return  $M$ 

```

```

Procedure Extend( $I, X, Y$ )
//  $I$  is the itemset to be extended.  $X$  is the set of items
// that can be added to  $I$ , i.e., the combine set, and
//  $Y$  is the set of relevant maximal itemsets found so far
// i.e., all maximal itemsets which contain  $I$ .
1. extendig = 0
2. for each  $j \in X$  do
3.   if  $|Y| > 0$  then
4.      $G = \{x : x \text{ is } j \text{ or } x \text{ follows } j \text{ in } X\}$ 
5.     if  $G$  has super set in  $Y$  then
6.       extendig = 1; break;
7.    $NewI = I \cup \{j\}; d(NewI) = d(I) - d(j)$ 
8.   if  $(NewI \text{ is frequent})$  then
9.      $NewX = X \cap c(j)$ 
10.    extendig = 1
11.    if  $(NewX == \emptyset)$  then
12.       $Y = Y \cup \{NewI\}$ 
13.    else
14.       $NewY = \{x \in Y : j \in x\}$ 
15.       $Extend(NewI, NewX, NewY)$ 
16.       $Y = Y \cup \{NewY\}$ 
17.      if  $(extendig == 0 \text{ and } |Y| == 0)$  then
18.         $Y = Y \cup \{I\}$ 

```

6/6/2023

BỘ MÔN KHOA HỌC MÁY TÍNH

29

BỘ MÔN KHOA HỌC MÁY TÍNH

30

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

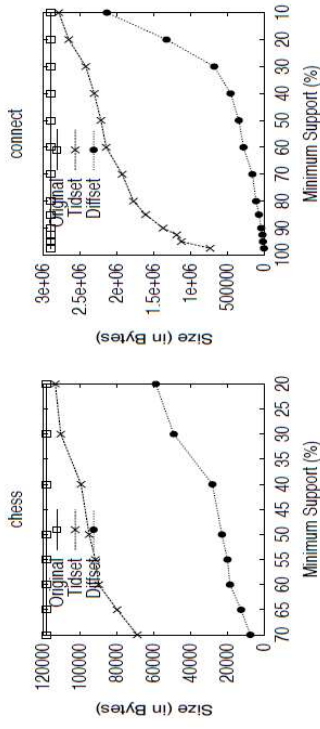
❖ Các cải tiến

- Sắp item theo thứ tự **tăng dần** theo kích thước và độ support (i. 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 phổ biến bằng cách sử dụng tidset, hoặc cải tiến hơn nữa là **diffsets**.

❖ Bộ nhớ

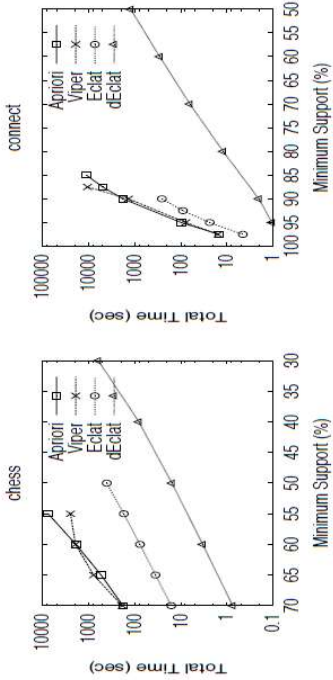
- Lưu trữ nhiều nhất $k = m + l$ tidsets (diffsets) trong bộ nhớ, với m là độ dài của tập kết hợp dài nhất và l là độ dài của itemset tối đại có chiều dài lớn nhất.

7. So sánh Tidset và Diffset



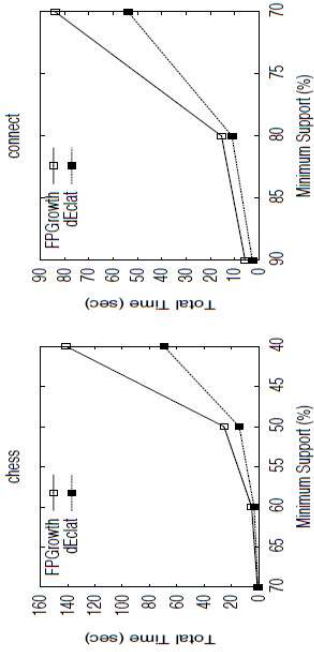
So sánh bộ nhớ sử dụng
trên dữ liệu đặc Chess và Connect

7. So sánh Tidset và Diffset



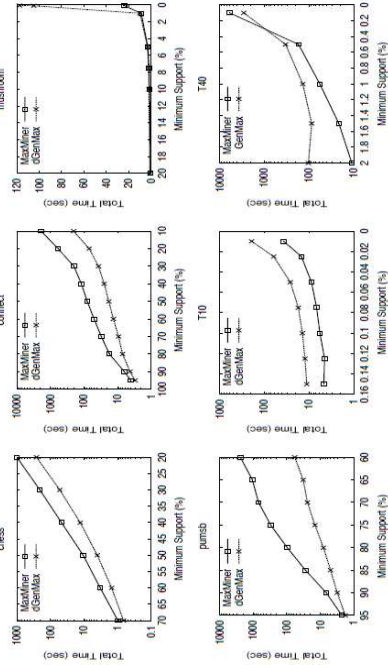
*So sánh thời gian thực thi
trên dữ liệu đặc Chess và Connect*

7. So sánh Tidset và Diffset

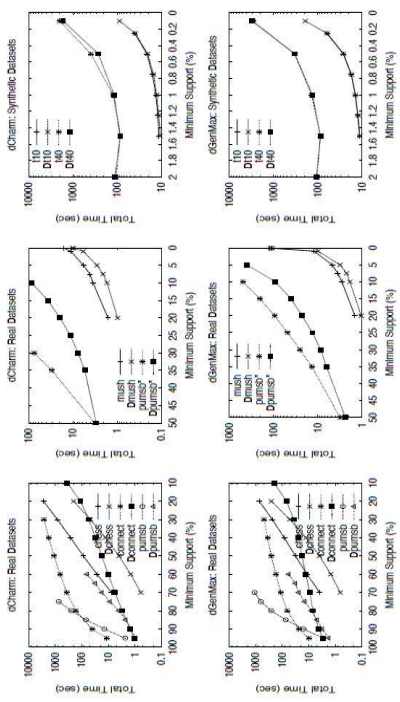


*So sánh thời gian thực thi
giữ thuật toán FPGrowth và dEclat*

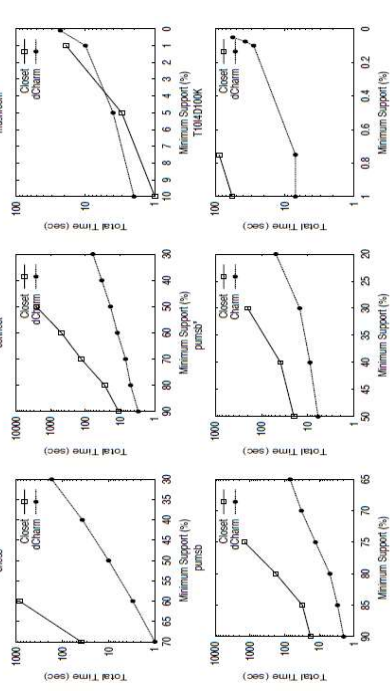
7. So sánh Tidset và Diffset



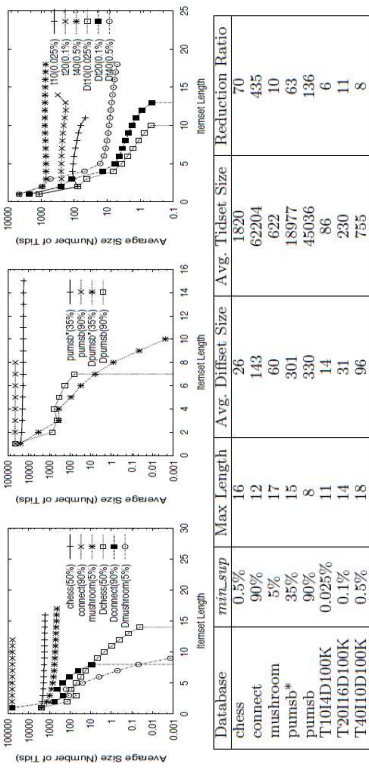
7. So sánh Tidset và Diffset



7. So sánh Tidset và Diffset

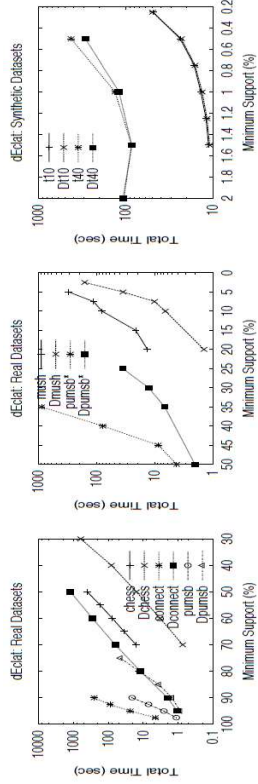


7. So sánh Tidset và Diffset



Kích thước trung bình vòng lặp: Tidset vs Diffset

7. So sánh Tidset và Diffset



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

8. Nhận xét

- ❖ 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 suất 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.

Thanks for your listening !!
Q & A