Chương 4

KHAI PHÁ DỮ LIỆU

Nội dung

1. Tiền xử lý dữ liệu.
2. Phương pháp khai phá bằng luật kết hợp.
3. Phương pháp cây quyết định.
4. Các phương pháp phân cụm.
5. Các phương pháp khai phá dữ liệu phức tạp.

Tập phổ biến và luật kết hợp

- Cho tập $I = \{i_1, i_2, \ldots, i_n\}$,
  - Mục (item): $i_1, i_2, \ldots$
  - Tập mục (item set): Tập $X \subseteq I$
- Cho tập $D = \{T_1, T_2, \ldots, T_n\}$,
  - T_i: Một giao dịch (transaction),
  - T_i: Các tập con của I,
  - D: Cơ sở dữ liệu giao dịch.
  - |D|: Số giao dịch trong D.

Tập phổ biến và luật kết hợp

- Ví dụ:
  - Cho tập I = {A, B, C, D, E},
  - Tập mục: X = {A, D, E},
  - Cơ sở dữ liệu giao dịch D:

T_1	{A, B, C, D}
T_2	{A, C, E}
T_3	{A, E}
T_4	{A, B, E}
T_5	{A, B, C, D, E}

  - D có 5 giao dịch.

Tập phổ biến và luật kết hợp

- Độ hỗ trợ (support) ứng với một tập mục:
  - ✓ Là xác suất xuất hiện của X trong cơ sở dữ liệu giao dịch D.
  - ✓ Công thức: $\sup(X) = \frac{C(X)}{|D|}$
  - ✓ C(X) là số giao dịch có chứa X.
  - ✓ Ví dụ: X = {A, C},
    - → C(X) = 3,
    - → Sup(X) = 3/5 (=60%)

T_1	{A, B, C, D}
T_2	{A, C, E}
T_3	{A, E}
T_4	{A, B, E}
T_5	{A, B, C, D, E}

- Các tập mục có độ hỗ trợ lớn hơn một giá trị ngưỡng minsup nào đó cho trước gọi là tập phổ biến.

Luật kết hợp (Association Rule)

- Cho hai tập mục X, Y ⊆ I, X ∩ Y = ϕ.
- Luật kết hợp ký hiệu X → Y, là mối ràng buộc của tập mục Y theo tập mục X,
- X xuất hiện trong cơ sở dữ liệu giao dịch sẽ kéo theo sự xuất hiện của Y với một một tỷ lệ nào đấy.
- Luật kết hợp được đặc trưng bởi:
  - ✓ Độ hỗ trợ của luật (xác suất cả X và Y cùng xuất hiện trong một giao dịch):

$$\sup(X \to Y) = \sup(X \cup Y) = \frac{C(X \cup Y)}{|D|}$$

Luật kết hợp (Association Rule)

- Độ tin cậy của luật (tỷ lệ các giao dịch chứa cả X và Y so với các giao dịch chứa X):

$$conf(X \rightarrow Y) = \frac{C(X \cup Y)}{C(X)} = \frac{\sup(X \rightarrow Y)}{\sup(X)}$$

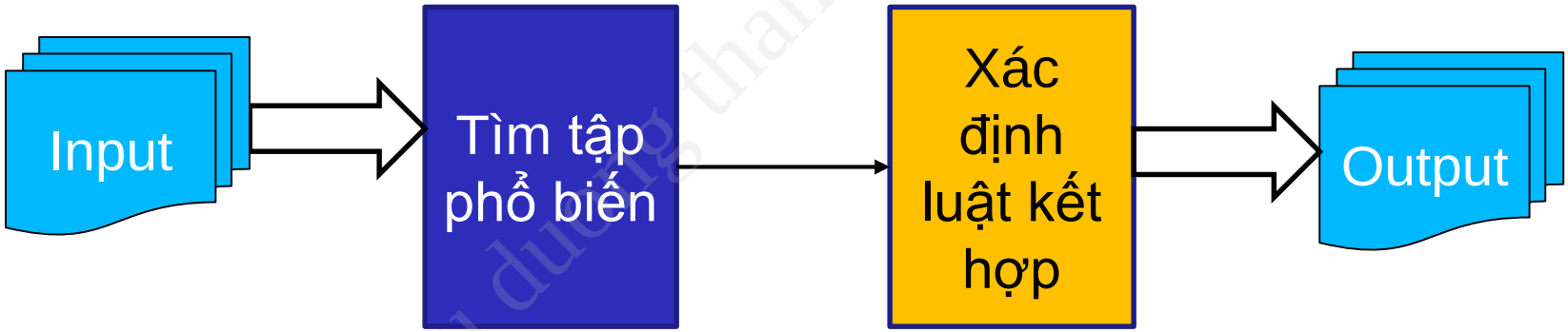
- Trong đó: C(X∪Y) là số giao dịch chứa cả X và Y,
  C(X) là số giao dịch có chứa X

Luật kết hợp (Association Rule)

- Luật mạnh: Các luật có độ hỗ trợ lớn hơn một giá trị ngưỡng minsup và độ tin cậy lớn hơn một giá trị ngưỡng minconf cho trước được gọi là luật mạnh, hay luật có giá trị (strong association rules).
  - Nếu đồng thời $sup(X \rightarrow Y) \geq minsup$ thì $X \rightarrow Y$ và $conf(X \rightarrow Y) \geq minconf$ được gọi là luật mạnh.

Luật kết hợp (Association Rule)

- Mô hình khai phá dữ liệu bằng luật kết hợp.



Luật kết hợp (Association Rule)

- Mô hình khai phá dữ liệu bằng luật kết hợp
  - Input: Cơ sở dữ liệu giao dịch, trị ngưỡng minsup, minconf.
  - Tìm tập phổ biến: Sinh tất cả các luật kết hợp có thể có bằng Apriori, FP-Growth, …
  - Xác định luật kết hợp: Tách tập phổ biến tìm được thành 2 tập không giao nhau X và Y. Tính độ tin cậy của X $\rightarrow$ Y, nếu trên ngưỡng minconf thì đó là luật mạnh.
  - Output: Tất cả các luật mạnh.

Luật kết hợp (Association Rule)

- Mô hình khai phá dữ liệu bằng luật kết hợp
  - ✓ Nếu tập M phổ biến có n phần tử thì số tập con của M sẽ là 2^n - 2. Vì vậy, với M ta sẽ có nhiều nhất 2^n – 2 luật.
  - ✓ Khi một giải thuật không dựa vào Độ hỗ trợ mà dựa vào Số lần xuất hiện (support count) thì giá trị ngưỡng mincount để một tập là phổ biến được xác định:

$$mincount = minsup * |D|$$

Luật kết hợp (Association Rule)

- Nguyễn lý Apriori:

"*Nếu một tập mục là phổ biến thì mọi tập con khác rỗng bất kỳ của nó cũng là tập phổ biến*".

→ Tìm ra tất cả các tập phổ biến có thể có.

Luật kết hợp (Association Rule)

- Nguyễn lý Apriori:

Chứng minh:

Xét X’$\subseteq$ X. Gọi p là ngưỡng độ hỗ trợ minsup. Một tập mục xuất hiện bao nhiêu lần thì các tập con chứa trong nó cũng xuất hiện ít nhất bấy nhiêu lần. Do đó:

$$C(X') \geq C(X) \quad (1)$$

X là tập phổ biến nên: $$\sup(X) = \frac{C(X)}{|D|} \geq p \Rightarrow C(X) \geq p|D| \quad (2)$$

Từ (1) và (2) ta có: $$C(X') > p|D| \Rightarrow \sup(X') = \frac{C(X')}{|D|} \geq p$$

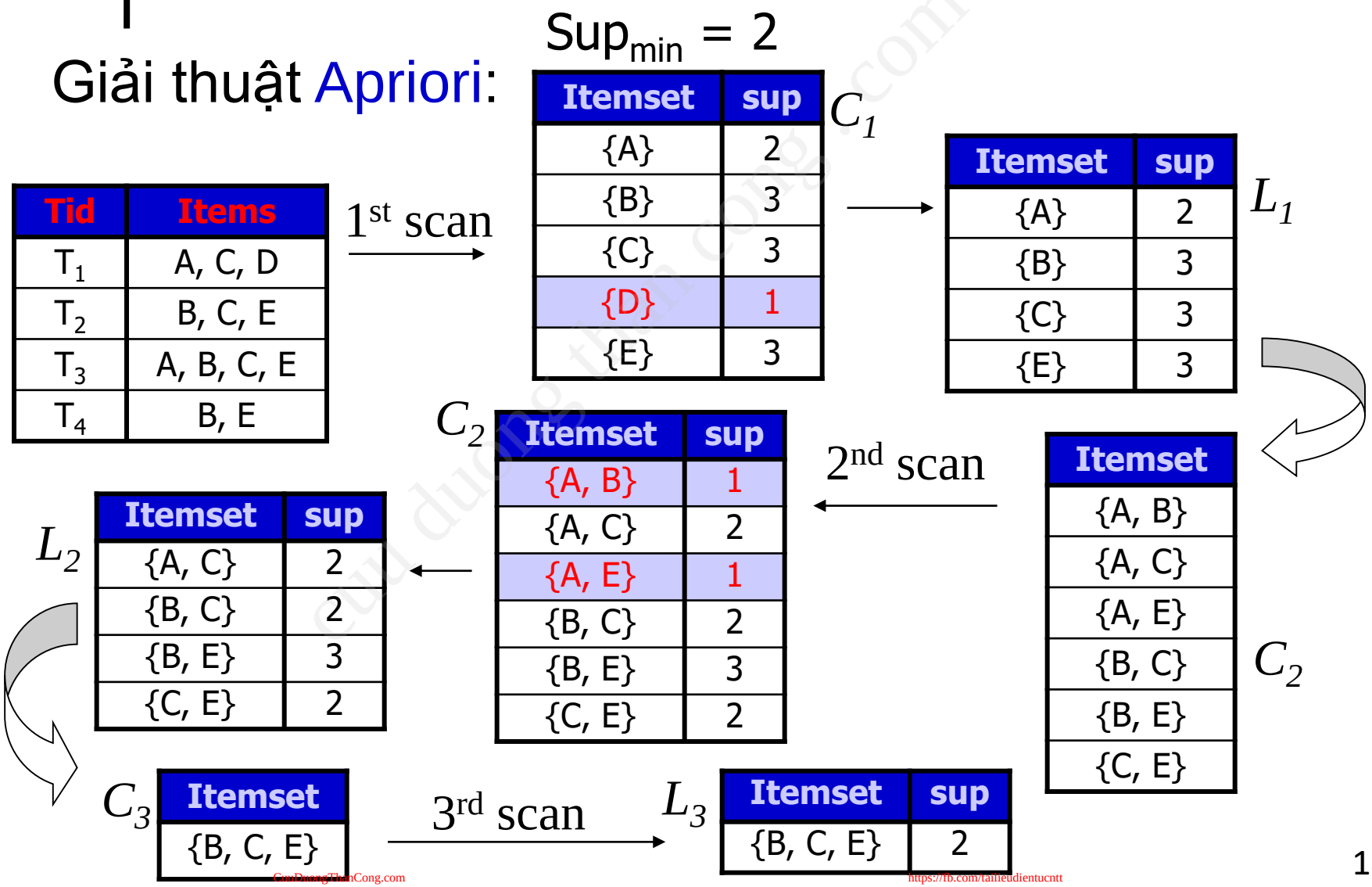
Vậy X’ cũng là tập phổ biến.

Luật kết hợp (Association Rule)

- Giải thuật Apriori:
  - Dựa trên nguyên lý Apriori.
  - Dựa vào quy hoạch động:
    - Xét các tập $F_i = \{c_i | c_i$ là tập phổ biến, $c_i = i\}$, gồm mọi tập mục phổ biến có độ dài i, $i \in [1;k]$.
    - Tìm tập F_{k+1} gồm mọi tập mục phổ biến có độ dài k+1.
    - Các mục $I_1, I_2, \ldots, I_n$ trong tập I được sắp theo một trật tự nhất định.

Luật kết hợp (Association Rule)

- Giải thuật Apriori:



Luật kết hợp (Association Rule)

- Giải thuật Apriori:

→ Các tập phổ biến:

F = {{A}, {B}, {C}, {E},

{A, C}, {B, C}, {B, E}, {C, E},

{B, C, E}}

Luật kết hợp (Association Rule)

C_k: Candidate itemset of size k

L_k: frequent itemset of size k

L_1 = {frequent items};

for (k = 1; L_k !=∅; k++) **do begin**

C_{k+1} = candidates generated from L_k;

for each transaction t in database do

increment the count of all candidates in C_{k+1} that are contained in t

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\cup_k L_k$;

Luật kết hợp (Association Rule)

- Sinh luật kết hợp:
  - Với mỗi tập phổ biến X $\in$ F, ta xác định các tập mục khác $\varnothing$ (rỗng) là con của X.
  - Với mỗi tập mục con không rỗng của X ta sẽ thu được một luật kết hợp S $\rightarrow$ (X\S). Nếu độ tin cậy của luật thỏa mãn ngưỡng minconf thì luật đó là luật mạnh.

$$conf(S \rightarrow (X \setminus S)) = \frac{C(X)}{C(S)} \geq \min conf$$

Luật kết hợp (Association Rule)

- Sinh luật kết hợp:

function Rules_Gen (F: Tập các tập phổ biến)

{

R = $\varnothing$;

F = F\F$_1$; //*Các tập phổ biến độ dài 1 không sinh luật*

for each X ∈ F

for each S ⊂ X

if conf(S→(X\S)) ≥ minconf **then**

R = R ∪ { S→(X\S)};

return R;

} // Output: Tập các luật kết hợp mạnh

Luật kết hợp (Association Rule)

- Ví dụ:

Cho I = {A, B, C, D, E, F} và cơ sở dữ liệu giao dịch D:

Chọn ngưỡng minsup = 25% và minconf = 75%. Hãy xác định các luật kết hợp mạnh.

T_1	{A, B, C, F}
T_2	{A, B, E, F}
T_3	{A, C}
T_4	{D, E}
T_5	{B, F}

Luật kết hợp (Association Rule)

C_1

Tập mục	Số lần xuất hiện
{A}	3
{B}	3
{C}	2
{D}	1
{E}	2
{F}	3

Sinh các tập phổ biến có độ dài bằng 1

L_1

F_1	Số lần xuất hiện
{A}	3
{B}	3
{C}	2
{E}	2
{F}	3

Kết hợp các tập của L_1 để có C_2

C_2

Tập mục
{A, B}
{A, C}
{A, E}
{A, F}
{B, C}
{B, E}
{B, F}
{C, E}
{C, F}
{E, F}

Luật kết hợp (Association Rule)

C_2

Tập mục
{A, B}
{A, C}
{A, E}
{A, F}
{B, C}
{B, E}
{B, F}
{C, E}
{C, F}
{E, F}

C_2

Tập mục	Số lần xuất hiện
{A, B}	2
{A, C}	2
{A, E}	1
{A, F}	2
{B, C}	1
{B, E}	1
{B, F}	3
{C, E}	0
{C, F}	1
{E, F}	1

Tập mục	Số lần xuất hiện
{A, B}	2
{A, C}	2
{A, F}	2
{B, F}	3

L_2

Kết hợp các tập của L_2 để có C_3

Tập mục
{A, B, C}
{A, B, F}
{A, C, F}

C_3

Luật kết hợp (Association Rule)

C_3

Tập mục
{A, B, C}
{A, B, F}
{A, C, F}

C_3

Tập mục	Số lần xuất hiện
{A, B, C}	1
{A, B, F}	2
{A, C, F}	1

L_3

Tập mục	Số lần xuất hiện
{A, B, F}	2

L_3 chỉ có một phần tử nên không thể tiếp tục kết nối để sinh L_4.

Thuật toán kết thúc.

Ta có tập các tập phổ biến là:

F ={{A}, {B}, {C}, {E}, {F}, {A, B}, {A, C}, {A, F}, {B, F}, {A, B, F}}

Luật kết hợp (Association Rule)

{A, B} có thể sinh các luật: {A} → {B} và {B} → {A}

$$conf(\{A\} \rightarrow \{B\}) = \frac{C(\{A,B\})}{C(\{A\})} = \frac{2}{3} = 66.7\%$$

$$conf(\{B\} \rightarrow \{A\}) = \frac{C(\{A,B\})}{C(\{B\})} = \frac{2}{3} = 66.7\%$$

{A, C} có thể sinh ra các luật: {A} → {C} và {C} → {A}

$$conf(\{A\} \rightarrow \{C\}) = \frac{C(\{A,C\})}{C(\{A\})} = \frac{2}{3} = 66.7\%$$

$$conf(\{C\} \rightarrow \{A\}) = \frac{C(\{A,C\})}{C(\{C\})} = \frac{2}{2} = 100\%$$

{A, F} có thể sinh ra các luật: {A} → {F} và {F} → {A}

$$conf(\{A\} \rightarrow \{F\}) = \frac{C(\{A,F\})}{C(\{A\})} = \frac{2}{3} = 66.7\%$$

$$conf(\{F\} \rightarrow \{A\}) = \frac{C(\{A,C\})}{C(\{F\})} = \frac{2}{3} = 66.7\%$$

Luật kết hợp (Association Rule)

{A, F} có thể sinh ra các luật: {A} → {F} và {F} → {A}

$$conf(\{B\} \rightarrow \{F\}) = \frac{C(\{B,F\})}{C(\{B\})} = \frac{3}{3} = 66.7\%$$

$$conf(\{F\} \rightarrow \{B\}) = \frac{C(\{B,C\})}{C(\{F\})} = \frac{3}{3} = 66.7\%$$

{A, B, F} có thể sinh ra các luật: {A} → {B, F}, {A, B} → {F}, {B} → {A, F}, {B, F} → {A}, {F} → {A, B}, {A, F} →{B}

$$conf(\{A\} \rightarrow \{B,F\}) = \frac{C(\{A,B,F\})}{C(\{A\})} = \frac{2}{3} = 66.7\%$$

$$conf(\{A,B\} \rightarrow \{F\}) = \frac{C(\{A,B,F\})}{C(\{A,B\})} = \frac{2}{2} = 100\%$$

$$conf(\{B\} \rightarrow \{A,F\}) = \frac{C(\{A,B,F\})}{C(\{B\})} = \frac{2}{3} = 66.7\%$$

$$conf(\{B,F\} \rightarrow \{A\}) = \frac{C(\{A,B,F\})}{C(\{B,F\})} = \frac{2}{3} = 66.7\%$$

Luật kết hợp (Association Rule)

$$conf(\{F\} \rightarrow \{A,B\}) = \frac{C(\{A,B,F\})}{C(\{F\})} = \frac{2}{3} = 66.7\%$$

$$conf(\{A,F\} \rightarrow \{B\}) = \frac{C(\{A,B,F\})}{C(\{A,F\})} = \frac{2}{2} = 100\%$$

Các luật kết hợp mạnh thu được gồm:

1. {C} → {A}
2. {B} → {F}
3. {F} → {B}
4. {A, B} → {F}
5. {A, F} → {B}

Luật kết hợp (Association Rule)

- Bài tập:

  Cho I = {A, B, C, D, E, F} và cơ sở dữ liệu giao dịch D:

  Chọn ngưỡng minsup = 20% và minconf = 70%. Hãy xác định các luật kết hợp mạnh.

T_1	{D, E}
T_2	{A, B, D, E}
T_3	{A, B, D}
T_4	{C, D, E}
T_5	{F}
T_6	{B, C, D}

Thuật giải FP-GROWTH

- Thuật giải FP-GROWTH cho phép phát hiện ra các tập phổ biến mà không cần khởi tạo các ứng viên
  - Xây dựng một cấu trúc dữ liệu thu gọn gọi là cây FP.
  - Kết xuất các mục phổ biến dựa trên cây FP.

Thuật giải FP-GROWTH – B1

Input:

- D, a transaction database;
- *min_sup*, the minimum support count threshold.

Output: The complete set of frequent patterns.

Method:

1. The FP-tree is constructed in the following steps:
   (a) Scan the transaction database D once. Collect F, the set of frequent items, and their support counts. Sort F in support count descending order as L, the *list* of frequent items.
   (b) Create the root of an FP-tree, and label it as "null." For each transaction *Trans* in D do the following. Select and sort the frequent items in *Trans* according to the order of L. Let the sorted frequent item list in *Trans* be $[p|P]$, where p is the first element and P is the remaining list. Call **insert_tree**$([p|P], T)$, which is performed as follows. If T has a child N such that $N.item\text{-}name = p.item\text{-}name$, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same *item-name* via the node-link structure. If P is nonempty, call **insert_tree**(P, N) recursively.
2. The FP-tree is mined by calling **FP_growth**(*FP_tree*, *null*), which is implemented as follows.

Thuật giải FP-GROWTH – B1

- Duyệt CSDL giao dịch và đếm số lần xuất hiện ứng với mỗi mục.
- Loại bỏ các mục không phổ biến.
- Sắp lại thứ tự các mục trong mỗi giao dịch theo thứ tự giảm dần của số lần xuất hiện.
- Mỗi nút của cây tương ứng với một mục và được gắn trọng số là số lần xuất hiện.
- Giải thuật FP-Growth đọc lần lqợt từng giao dịch và ánh xạ tương ứng với mỗi đừờng đi (xuất phát từ nút gốc) trên cây FP.

Thuật giải FP-GROWTH – B1

- Thứ tự sắp xếp của các mục được tuân thủ trong suốt quá trình xây dựng cây FP.
- Các đường đi có thể có thể có những đoạn trùng nhau do các giao dịch có các phần tử chung (chung tiền tố trong dãy). Mỗi lần có phần tử trùng thì trọng số của đỉnh ở vị trí trùng được tăng lên 1.
- Con trỏ được sử dụng để duy trì danh sách kết nối đơn giữa các nút đại diện cho cùng một mục.

Thuật giải FP-GROWTH – B2

```
procedure FP_growth(Tree, α)
(1)   if Tree contains a single path P then
(2)     for each combination (denoted as β) of the nodes in the path P
(3)       generate pattern β ∪ α with support_count = minimum support count of nodes in β;
(4)   else for each a_i in the header of Tree {
(5)     generate pattern β = a_i ∪ α with support_count = a_i.support_count;
(6)     construct β's conditional pattern base and then β's conditional FP_tree Tree_β;
(7)     if Tree_β ≠ ∅ then
(8)       call FP_growth(Tree_β, β); }
```

Thuật giải FP-GROWTH – B1

Ứng với mỗi mục phổ biến I_j:

- Xây dựng tập các cơ sở mẫu có điều kiện (conditional pattern base). Mỗi mẫu có điều kiện là một đường đi nối từ đỉnh gốc tới đỉnh cha kề với đỉnh có chứa mục I_j. Mỗi mẫu được gán trọng số bằng với trọng số của đỉnh có chứa mẫu I_j ở cuối đường đi.
- Xây dựng cây FP có điều kiện (conditional FP-tree) dựa trên việc kết hợp các mẫu có chung tiền tố (nếu có). Khi đó trọng số ứng với mỗi đỉnh là tổng các trọng số được ghép.
- Duyệt cây FP có điều kiện để sinh các tập phổ biến có hậu tố là I_j.

Thuật giải FP-GROWTH

Ví dụ: Cho cơ sở dữ liệu giao dịch D gồm các giao dịch như bảng dưới. Biết ngưỡng minsup = 60%. Hãy tìm các tập phổ biến.

T	Items
T100	{*f, a, c, d, g, i, m, p*}
T200	{*a, b, c, f, l, m, o*}
T300	{*b, f, h, j, o*}
T400	{*b, c, k, s, p*}
T500	{*a, f, c, e, l, p, m, n*}

Thuật giải FP-GROWTH

Duyệt CSDL để xác định tần suất xuất hiện của mỗi mục.

T	Items
T100	{*f, a, c, d, g, i, m, p*}
T200	{*a, b, c, f, l, m, o*}
T300	{*b, f, h, j, o*}
T400	{*b, c, k, s, p*}
T500	{*a, f, c, e, l, p, m, n*}

Items	frequency
a	3
b	3
c	4
f	4
m	3
p	3

==> mincount = 3

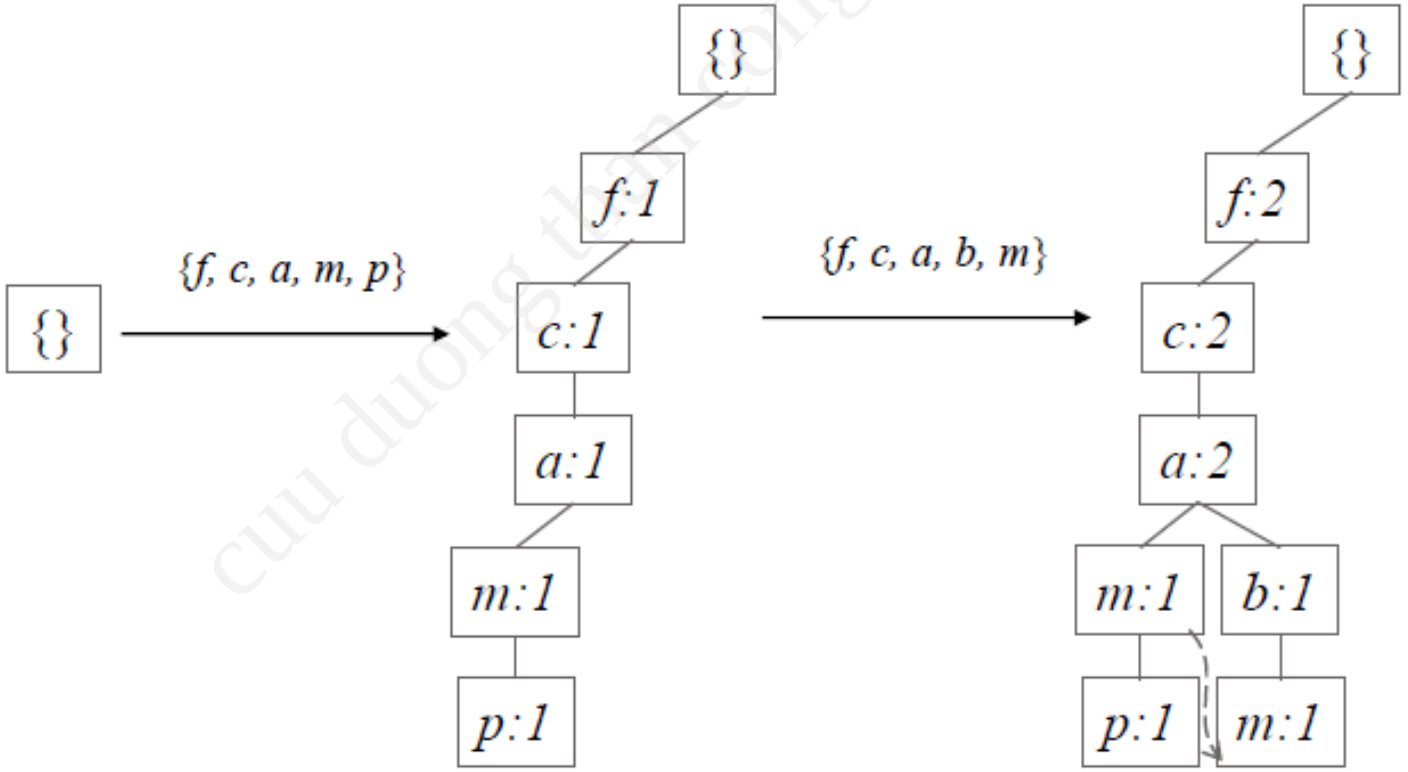
Thuật giải FP-GROWTH

- Loại bỏ các mục không phải là phổ biến.
- Sắp các mục trong mỗi giao dịch theo thứ tự giảm dần của support count

T	Items	Sort
T100	{*f, a, c, d, g, i, m, p*}	{*f, c, a, m, p*}
T200	{*a, b, c, f, l, m, o*}	{*f, c, a, b, m*}
T300	{*b, f, h, j, o*}	{*f, b*}
T400	{*b, c, k, s, p*}	{*c, b, p*}
T500	{*a, f, c, e, l, p, m, n*}	{*f, c, a, m, p*}

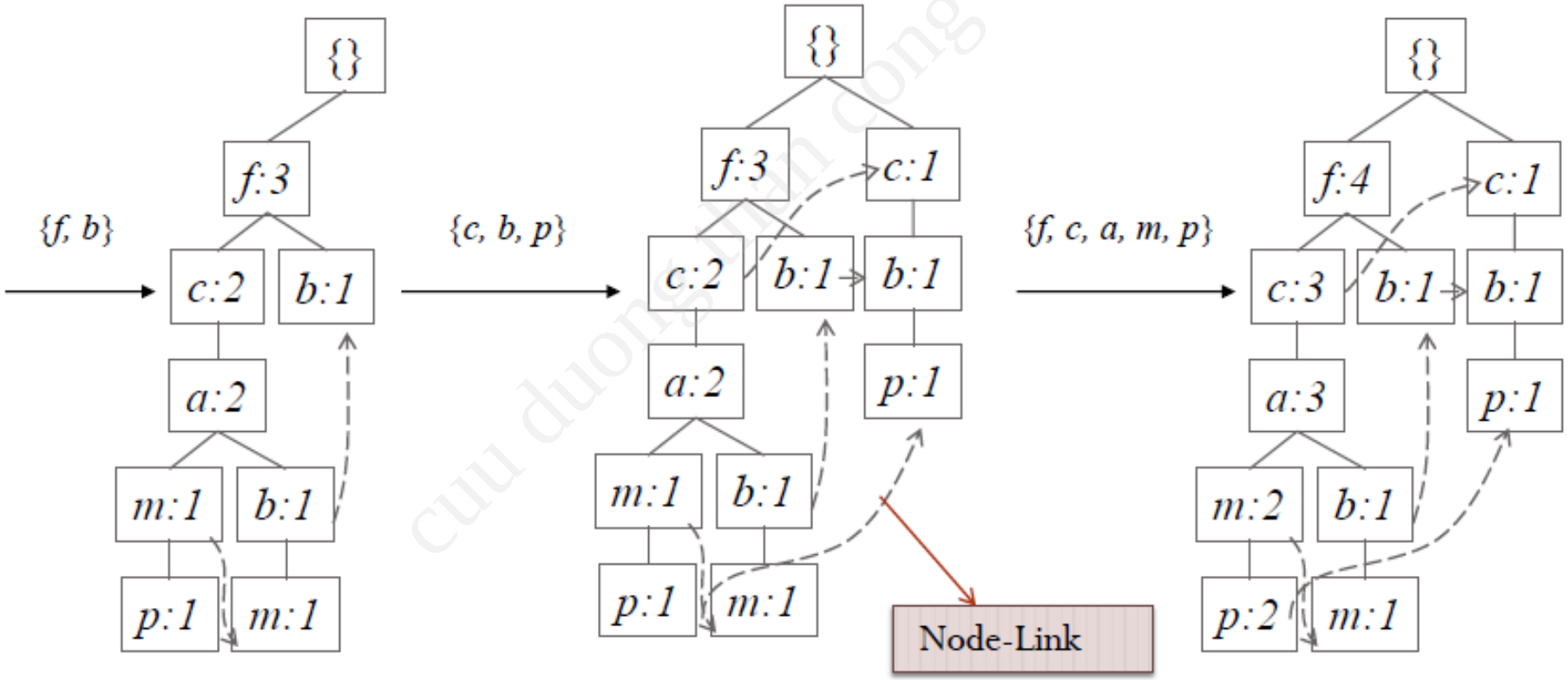
Thuật giải FP-GROWTH

- Đọc từng giao dịch và ánh xạ vào cây FP:



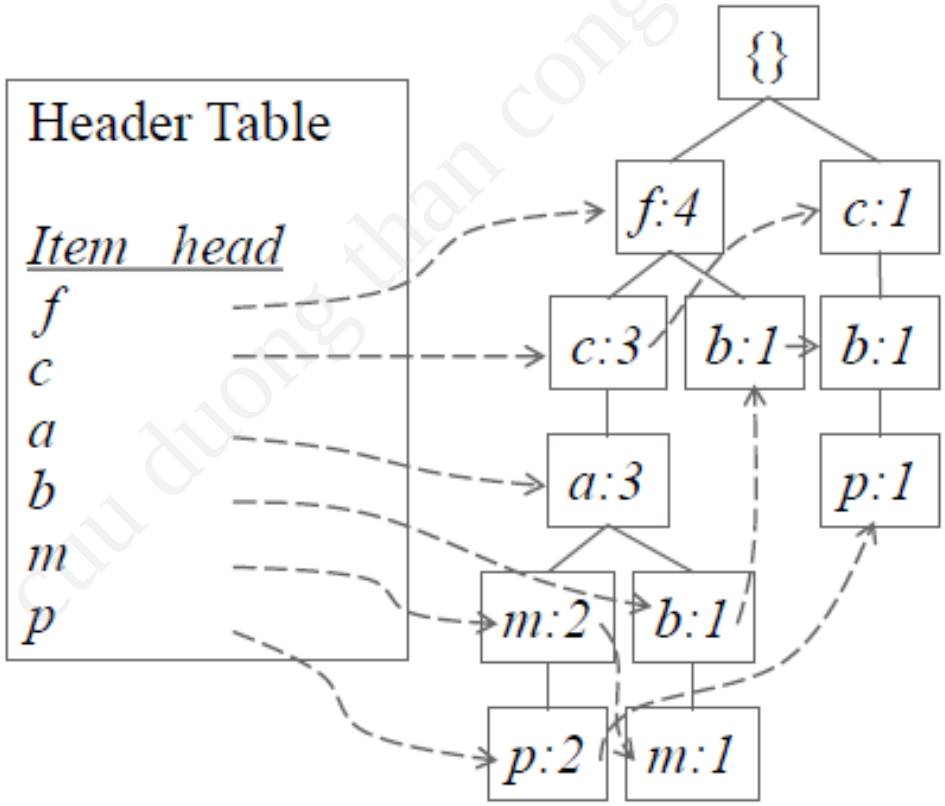
Thuật giải FP-GROWTH

- Đọc từng giao dịch và ánh xạ vào cây FP:



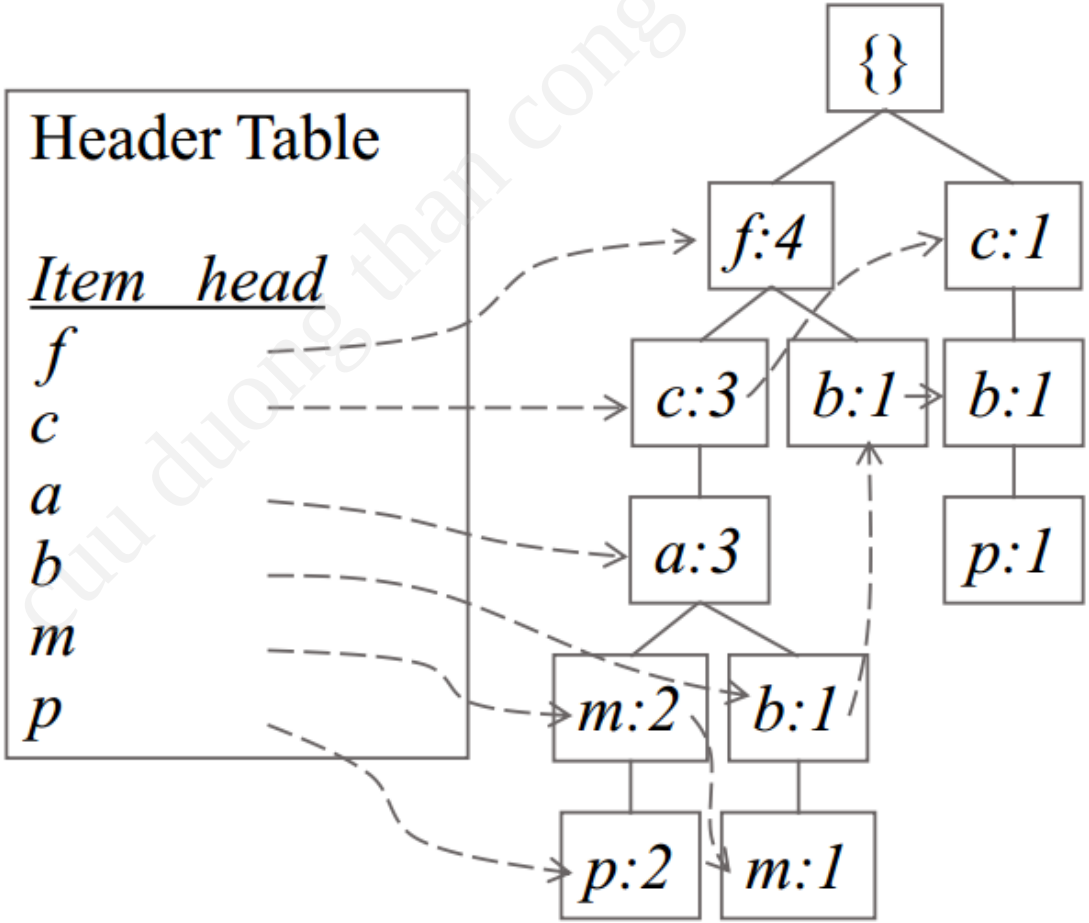
Thuật giải FP-GROWTH

❖ Cây FP hoàn chỉnh:



Thuật giải FP-GROWTH

❖ Cây FP hoàn chỉnh:



Thuật giải FP-GROWTH

Mục	Cơ sở mẫu có điều kiện	Cây FP có điều kiện	Tập phổ biến
p	***fcam:2, cb:1***	***{c:3}***	***p:3, cp:3***
m	***fca:2, fcab:1***	***{f:3, c:3, a:3}***	***m:3, fm:3, cm:3, am:3, fcm:3, fam:3, cam:3***
b	***fca:1, f:1, c:1***	***Null***	***b:3***
a	***fc:3***	***{f:3, c:3}***	***a:3, fa:3, ca:3***
c	***f:3***	***{f:3}***	***c:3, fc:3***
f	***Null***	***Null***	***f:3***