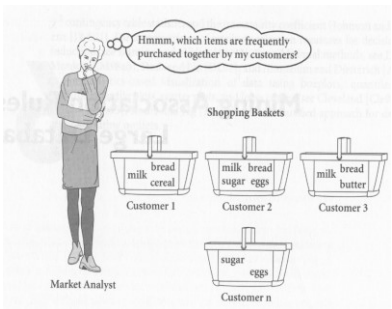
CHƯƠNG 3
KHAI PHÁ LUẬT KẾT HỢP

Nội dung

3.1. Giới thiệu

Bài toán phân tích giỏ hàng

- Phân tích thói quen mua hàng của khách hàng bằng cách tìm ra những "mối kết hợp" giữa những mặt hàng mà khách đã mua
- Mục tiêu giúp gia tăng doanh số, tạo thuận lợi cho khách khi mua hàng trong siêu thị
- Bài toán được Agrawal thuộc nhóm nghiên cứu của IBM đưa ra vào năm 1994



3.1. Giới thiệu

- Bài toán phát hiện luật kết hợp (Association rule mining)
  - Với một tập các giao dịch (transactions) cho trước, cần tìm các luật dự đoán khả năng xuất hiện trong một giao dịch của các mục (items) này dựa trên việc xuất hiện của các mục khác
- Một số ví dụ về *"luật kết hợp" (associate rule)*
- "98% khách hàng *mà* mua tạp chí thể thao *thì đều* mua các tạp chí về ôtô" ⇨ sự ***kết hợp*** giữa "*tạp chí thể thao*" với "*tạp chí về ôtô*"
- "60% khách hàng *mà* mua bia tại siêu thị *thì đều* mua bỉm trẻ em" ⇨ sự ***kết hợp*** giữa "*bia*" với "*bỉm trẻ em*"
- *40% sinh viên học khá môn Cấu trúc dữ liệu thì cũng học khá môn Lập trình hướng đối tượng với độ tin cậy 60%*

3.2. Một số khái niệm

1. **Cơ sở dữ liệu giao dịch** (transaction database)
   - *Giao dịch (giao tác)*: danh sách các mặt hàng (mục: item) trong một phiếu mua hàng của khách hàng. Giao dịch T là một tập mục.

   Ví dụ có các phiếu mua hàng được mô tả như bảng bên

   Phiếu 1: TID=1 có các mặt hàng Milk, Bread, Eggs

   Phiếu 2: TID=2 : Bread, Sugar

   ….

TID	Produces
1	MILK, BREAD, EGGS
2	BREAD, SUGAR
3	BREAD, CEREAL
4	MILK, BREAD, SUGAR
5	MILK, CEREAL
6	BREAD, CEREAL
7	MILK, CEREAL
8	MILK, BREAD, CEREAL, EGGS
9	MILK, BREAD, CEREAL

3.2.Một số khái niệm

TID	Products
1	A, B, E
2	B, D
3	B, C
4	A, B, D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, C

Biến đổi CSDL về dạng nhị phân

TID	A	B	C	D	E
1	1	1	0	0	1
2	0	1	0	1	0
3	0	1	1	0	0
4	1	1	0	1	0
5	1	0	1	0	0
6	0	1	1	0	0
7	1	0	1	0	0
8	1	1	1	0	1
9	1	1	1	0	0

ITEMS:

A = milk
B= bread
C= cereal
D= sugar
E= eggs

3.2.Một số khái niệm

Định nghĩa

- *Mục dữ liệu (Item):* là mặt hàng trong phiếu mua hàng hay thuộc tính
- ***Tập mục:***Kí hiệu $I = \{i_1, i_2, \ldots, i_m\}$ là tập m thuộc tính riêng biệt, mỗi thuộc tính gọi là một mục dữ liệu.
  - Ví dụ I={Milk, Bread, Sugar}
- Một tập mục có k mục gọi là tập *k-mục*
- *Giao dịch(giao tác Transaction)*: tập các mặt hàng mua trong 1 phiếu hàng (có Tid – định danh) : (Tid, tập hạng mục).
  - Ví dụ (2, {Bread, Sugar})
- *Giao dịch t* : tập các hạng mục sao cho $t \subseteq I$
  - Ví dụ:t={Milk, Bread, Eggs}

3.2.Một số khái niệm

Định nghĩa

- CSDL giao dịch D là tập những giao dịch, mỗi giao dịch T là một tập các mục của I, $T \subseteq I$. Mỗi giao dịch có một định danh duy nhất gọi là TID.
- CSDL $D = \{t_1, t_2, \ldots, t_n\}$ trong đó $t_j \subseteq I$
- Ví dụ: CSDL như bảng bên

TID	Items bought
10	A, B, C
20	A, C
30	A, D
40	B, E, F

3.2.Một số khái niệm(tt)

2. Độ hỗ trợ, độ tin cậy tập mục phổ biến

Độ hỗ trợ:

- **Định nghĩa** *Độ hỗ trợ của một tập mục X trong cơ sở dữ liệu D là tỉ số giữa các giao tác T ⊂ D chứa tập X và tổng số giao tác trong D (hay là phần trăm của các giao tác trong D có chứa tập mục X) kí hiệu là Supp(X)*

$$Supp(X) = \frac{|\{T \in D : T \supset X\}|}{|D|}$$

3.2.Một số khái niệm(tt)

- ***Định nghĩa :*** *Độ hỗ trợ của luật $X \rightarrow Y$ là tỉ số của những giao tác có chứa $X \cup Y$ và số giao tác trong cơ sở dữ liệu D, ký hiệu : $Supp(X \rightarrow Y)$.*

$$Supp(X \rightarrow Y) = \frac{|\{T \in D : T \supseteq X \cup Y\}|}{|D|}$$

3.2.Một số khái niệm(tt)

- **Độ tin cậy**
- ***Định nghĩa :*** *Độ tin cậy của một luật $r=X \rightarrow Y$ là tỉ số (phần trăm) của số giao tác trong D chứa $X \cup Y$ với số giao tác trong D có chứa tập mục X. Kí hiệu độ tin cậy của một luật là conf(r), với $0 \leq conf(r) \leq 1$.*

3.2.Một số khái niệm(tt)

Định nghĩa : *Tập mục X được gọi là tập mục phổ biến nếu có Supp(X) ≥ MinSup, với Minsup là ngưỡng độ hỗ trợ cho trước.*

- Tính chất của tập phổ biến.

Những tập con của tập phổ biến cũng phải phổ biến

Tập con không phổ biến vậy tập cha của nó có phổ biến hay không?

3.2.Một số khái niệm(tt)

Luật kết hợp

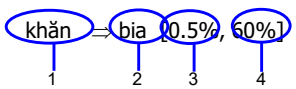
- Luật kết hợp: qui tắc kết hợp có điều kiện giữa các tập phần tử.
- Thể hiện mối liên hệ (có điều kiện) giữa các tập phần tử

Định nghĩa : Một *luật kết hợp* là một quan hệ có dạng $X \rightarrow Y$, trong đó $X, Y \subset I$ là các tập mục, và X được gọi là tiền đề, Y là mệnh đề kết quả. Hai thông số quan trọng của luật kết hợp là *độ hỗ trợ* (*supp*) và *độ tin cậy* (*conf*).

3.2.Một số khái niệm(tt)

- **Biểu diễn đặc trưng cho các luật kết hợp:**
  - khăn ⇒ bia [0.5%, 60%]
  - mua:khăn ⇒ mua:bia [0.5%, 60%]
  - "**Nếu** mua khăn **thì** mua bia trong 60% trường hợp. Khăn và bia được mua chung trong 0.5% dòng dữ liệu."
- **Các biểu diễn khác:**
  - mua(x, "khăn") ⇒ mua(x, "bia") [0.5%, 60%]
  - khoa(x, "CS") ^ học(x, "DB") ⇒ điểm(x, "A") [1%, 75%]

3.2.Một số khái niệm(tt)



"**NẾU** mua khăn
THÌ mua bia
trong 60% trường hợp
trên 0.5% số dòng dữ liệu"

1 **Tiền đề**, vế trái luật
2 **Mệnh đề kết quả**, vế phải luật
3 **Support**, tần số, độ hỗ trợ ("trong bao nhiêu phần trăm dữ liệu thì những điều ở vế trái và vế phải cùng xảy ra")
4 **Confidence**, độ mạnh, độ tin cậy ("nếu vế trái xảy ra thì có bao nhiêu khả năng vế phải xảy ra")

Ví dụ tìm luật kết hợp

Transaction-id	Items bought
10	A, B, C
20	A, C
30	A, D
40	B, E, F

Min. support 50%
Min. confidence 50%

Tập mục t. xuyên	Support
{A}	75%=3/4
{B}	50%
{C}	50%
{A, C}	50%

X={AC}
Count(X)=2, |D|=4
Sup(x)=2/4=50%
For rule $A \Rightarrow C$:
Sup($A \Rightarrow C$) = count ($\{A\} \cup \{C\}$)/|D| =2/4 50%
confidence = count($\{A\} \cup \{C\}$)/count($\{A\}$) =2/3= 66.6%

Bài tập

- Cho csdl sau
- Minsup=30%
- Minconf=40%
- Tìm các tập mục phổ biến và các luật thỏa mãn 2 điều kiện trên.

TID	Các mục trong giao tác
T1	A,B,C
T2	A,C
T3	A,B,D
T4	B,C
T5	B,D

3.3. Mô tả bài toán khai phá luật kết hợp

Tìm tần số mẫu, mối kết hợp, sự tương quan, hay các cấu trúc nhân quả giữa các tập đối tượng trong các cơ sở dữ liệu giao tác, cơ sở dữ liệu quan hệ, và những kho thông tin khác đảm bảo được:**Tính hiểu được, Tính sử dụng được:** Cung cấp thông tin thiết thực

- **Tính hiệu quả:** Đã có những thuật toán khai thác hiệu quả

Bài toán :Với một cơ sở dữ liệu giao dịch D có I= $\{i_1, i_2, \ldots, i_m\}$ tập m thuộc tính của dữ liệu, ngưỡng độ hỗ trợ tối thiểu *minsup*, ngưỡng độ tin cậy *mincof*,

Bài toán khai phá luật kết hợp là tìm tất cả các luật kết hợp dạng $X \rightarrow Y$ trên D (X, Y $\subset$I và X$\cap$Y ={})thoả mãn điều kiện

Support($X \rightarrow Y$) >= *minsup* và Confidence($X \rightarrow Y$) >= *mincof*.

3.3. Mô tả bài toán khai phá luật kết hợp

Quy trình tìm luật kết hợp từ cơ sở dữ liệu:

Dữ liệu vào : *I, D, minsup, minconf;*

Dữ liệu ra: *tất cả những luật kết hợp thỏa minsup và minconf*

Phương pháp:

1. Tìm tất cả những tập mục phổ biến có độ hỗ trợ lớn hơn hay bằng ngưỡng hỗ trợ cho trước *minsup*.
2. Từ các tập mục phổ biến sinh các luật kết hợp thỏa mãn *minsup* và *minconf*.

3.4 Tìm tập mục phổ biến

- Bài toán tìm tập phổ biến là bài toán rất quan trọng lĩnh vựcKPDL
- Bài toán tìm tập phổ biến là bài toán tìm tất cả các tập các tập mục S(hay tập phổ biến S) có độ hỗ trợ thỏa mãn độ hộ trợ tối thiểu *minsupp*
  - *supp(S)* $\geq$ *minsupp*
- Cách giải quyết: dựa trên tính chất của tập phổ biến
- Tìm kiếm theo chiều rộng : Thuật toán Apriori (1994)
- Phát triển mẫu: Thuật toán FP-Growth(2000)

3.4.1 Thuật toán Apriori

Thuật toán sử dụng tính chất tập con của một tập phổ biến là phổ biến

- Duyệt qua dữ liệu nhiều lần
- Lần duyệt đầu tiên – tính độ hỗ trợ cho mỗi mục và xác định tập mục nào phổ biến
- Lần duyệt thứ k
  - Sinh ứng cử viên C_k có k-mục từ các tập mục phổ biến k-1 mục ở bước trước.
  - Xác định tập mục phổ biến L_k từ tập ứng cử C_k.
    Lặp lại quá trình cho đến khi không tìm được các tập mục phổ biến mới.
  - Quá trình này gồm 2 bước như sau

3.4.1 Thuật toán Apriori

Kết nối

- Để tìm tập các tập mục phổ biến k-mục L_k chúng ta xuất phát từ các tập mục phổ biến L_{k-1}, sinh tập ứng cử k-mục C_k bằng cách kết nối các phần tử L_{k-1}.
- Nối các phần tử trong L_{k-1} được thực hiện như sau: các phần tử trong L_{k-1} được kết nối với nhau nếu chúng có chung (k-2) mục đầu tiên giống nhau tức là:
- $(l_1[1]=l_2[1])\wedge(l_1[2]=l_2[2])\wedge\ldots\wedge(l_1[k-2]=l_2[k-2])\wedge(l_1[k-1]<l_2[k-1])$
- Điều kiện $(l_1[k-1]<l_2[k-1])$ là để đảm bảo không sinh thừa các tập ứng cử. Tập kết quả nhận được có dạng: $l_1[1]\ l_1[2]\ l_1[3]\ \ldots l_1[k-2]\ l_1[k-1] l_2[k-1]$.

3.4.1 thuật toán Apriori

- Kết nối

l_1 và l_2 là 2 tập mục phổ biến có k-2 mục đầu tiên giống nhau

```
for (moi l1 ∈ Lk-1 ) Do
    for (moi l2 ∈ Lk-1 ) Do {
        If (l1[1]=l2[1]) (l1[2]=l2[2]) &
           (l1[k-2]=l2[k-2]) (l1[k-1]<l2[k-1])Then
        c = l1 x l2
```

Kết nối l_1 với l_2

3.4.1 thuật toán Apriori

- **Bước tỉa**: Để rút gọn kích thước của C_k sử dụng tính chất bất kì tập con *(k-1)-mục* nào không phổ biến thì không thể là tập con của tập *k-mục* phổ biến. Do đó, nếu bất kỳ tập k-mục nào mà có tập *(k-1)-mục* không phổ biến thì sẽ bị loại bỏ khỏi C_k. Việc kiểm tra này có thể thực hiện nhanh bằng cách duy trì một cây băm cho tất cả các tập mục phổ biến.

3.4.1 Thuật toán Apriori

- Bước tỉa

for moi $c \in C_k$ do

for moi tap muc $k-1muc\ s\ cua\ c$ do

if $(s \notin L_{k-1})$ then

delete $c\ khoi\ C_k$

Kiểm tra tất cả các tập con và xóa đi những ứng cử có tập con không phổ biến

- Ví dụ

L_2 = { {A B}, { A C}, { A E},{B C}, {B D}, {B E} }

Sau khi kết nối

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

{D E},{ C D} và {C E} Không có trong L_2

Sau khi tỉa

C_3 = {{A B C}, {A B E}}

3.4.1 Thuật toán Apriori

L_1 = { 1-muc}

For ($k = 2; L_{k-1} \neq \phi ; k++$) {

C_k = apriori- gen (L_{k-1});

for (moi giao tac $t \in D$) {

C_t = subset(C_k,t)

for(moi ung cu $c \in C_t$)do

c.count ++;

}

L_k = { $c \in C_k | c.count \geq minsup$}

}

$L = \bigcup_k L_k$;

Tìm những tập phổ biến một mục

Sinh ứng cử mới k-mục

Tính độ hỗ trợ cho tất cả các ứng cử

Lấy những tập k-mục phổ biến

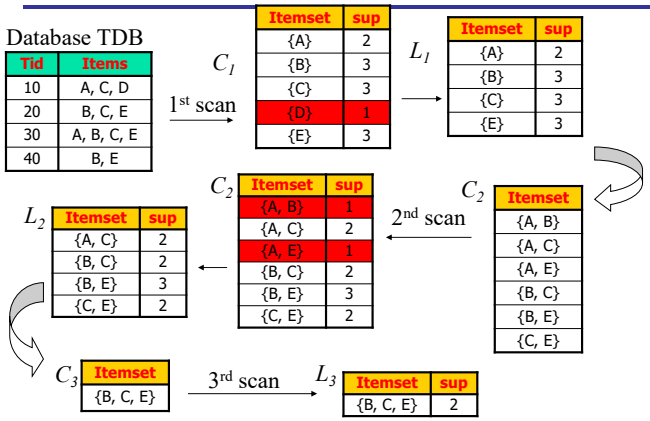
3.4.1 Thuật toán Apriori

```
procedure      apriori-gen(L_{k-1}; minsup){
  for (mỗi l_1 ∈ L_{k-1} ) Do
     for (mỗi l_2 ∈ L_{k-1} ) Do {
        If (l_1[1]=l_2[1])∧(l_1[2]=l_2[2])∧...∧
          (l_1[k-2]=l_2[k-2])∧(l_1[k-1]<l_2[k-1])Then
              c = l_1 x l_2
  if (has_inrequent_subset(c, L_{k-1}) Then
     xoá c;
  else thêm c vào C_k
}
return C_k}
```


3.4.1 Thuật toán Apriori

```
Function     has_inrequent_subset(c:
  ứng cử k-mục, L_{k-1})
  For (mỗi tập con (k-1)-mục s của c)
  Do
     If  s ∉ L_{k-1} Then
         return TRUE
return FALSE
```

Một ví dụ thuật toán Apriori (s=50%)



Bài tập

Cho CSDL giao dịch bên
Sử dụng thuật toán Apriori để tìm các tập phổ biến với minsupp = 22%
Tìm tất cả các luật kết hợp thỏa mãn
. *Minconf = 50 %*
. *Minconf = 70%*

Tid	Items
100	M1, M2, M5
200	M2, M4
300	M2, M3
400	M1, M2, M4
500	M1, M3
600	M2, M3
700	M1, M3
800	M1, M2, M3, M5
900	M1, M2, M3

3.4.1 Thuật toán Apriori

- **Phần cốt lõi của thuật toán Apriori:**
  - Dùng các tập phổ biến kích thước (k – 1) để tạo các tập phổ biến kích thước k ứng viên
  - Duyệt CSDL và đối sánh mẫu để đếm số lần xuất hiện của các tập ứng viên trong các giao tác
- **Tình trạng nghẽn cổ chai của thuật toán *Apriori*: việc tạo ứng viên**
  - Các tập ứng viên đồ sộ:
    - 10^4 tập phổ biến kích thước 1 sẽ tạo ra 10^7 tập ứng viên kích thước 2
    - Để phát hiện một mẫu phổ biến kích thước 100, ví dụ $\{a_1, a_2, \ldots, a_{100}\}$, cần tạo $2^{100} \approx 10^{30}$ ứng viên.
  - Duyệt CSDL nhiều lần:
    - Cần duyệt (n +1) lần, n là chiều dài của mẫu dài nhất

3.4.1 Thuật toán Apriori

Hạn chế của thuật toán Apriori

- **Thực tế:**
  - Đối với tiếp cận Apriori căn bản thì số lượng thuộc tính trên dòng thường khó hơn nhiều so với số lượng dòng giao tác.
  - Ví dụ:
    - 50 thuộc tính mỗi cái có 1-3 giá trị, 100.000 dòng (không quá tệ)
    - 50 thuộc tính mỗi cái có 10-100 giá trị, 100.000 dòng (hơi tệ)
    - 10.000 thuộc tính mỗi cái có 5-10 giá trị, 100 dòng (quá tệ...)
  - Lưu ý:
    - Một thuộc tính có thể có một vài giá trị khác nhau
    - Các thuật toán luật kết hợp có đặc trưng là xem một cặp thuộc tính-giá trị là một thuộc tính (2 thuộc tính mỗi cái có 5 giá trị => "10 thuộc tính")
- **Cách khắc phục vấn đề ?**

3.4.1 Thuật toán Apriori

Cải tiến thuật toán Apriori: ý tưởng chung

Giảm số lần duyệt CSDL

Giảm số ứng cử viên

Qui trình tính độ hỗ trợ đơn giản hơn

3.4.2.Thuật toán FP-Growth

- Nén tập dữ liệu vào cấu trúc cây (Frequent Pattern tree, FP-tree)
  - Giảm chi phí cho toàn tập dữ liệu dùng trong quá trình khai phá
    - Những tập mục không phổ biến bị loại bỏ sớm.
  - Đảm bảo kết quả khai phá không bị ảnh hưởng
- Phương pháp chia-để-trị (divide-and-conquer)

3.4.2.Thuật toán FP-Growth

- Quá trình khai phá
  1. Xây dựng FP-tree
  2. Khám phá frequent itemsets với FP-tree
     - B1 : Thiết lập cơ sở mẫu điều kiện (Conditional Pattern Bases) cho mỗi hạng mục phổ biến (mỗi nút trên cây FP).
     - B2 : Thiết lập cây FP điều kiện (Conditional FP tree) từ mỗi cơ sở mẫu điều kiện
     - B3 : Khai thác đệ qui cây FP điều kiện và phát triển mẫu phổ biến cho đến khi cây FP điều kiện chỉ chứa 1 đường dẫn duy nhất - tạo ra tất cả các tổ hợp của mẫu phổ biến

3.4.2.Thuật toán FP-Growth

Định nghĩa:

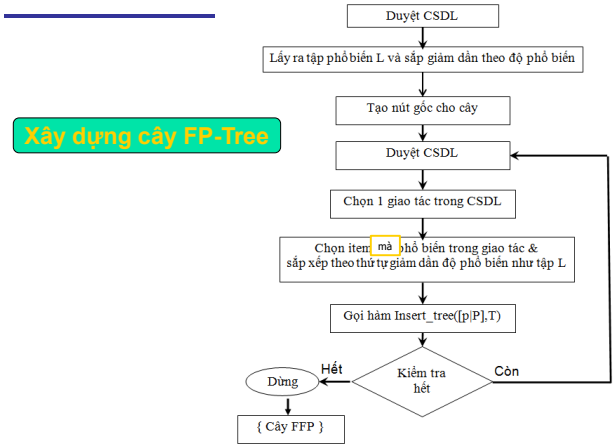
- FP_Tree bao gồm nút gốc có nhãn "Null", tập các cây con tiền tố (prefix) như là cây con của nút gốc và một bảng tiêu đề mục phổ biến.
- Mỗi nút của cây con tiền tố có 3 trường: *Item_name, count, nút liên kết (node link)*; với item_name là nhãn của nút, count: số giao tác mà mục này xuất hiện, node_link dùng để liên kết với nút tiếp theo trong cây nếu có cùng Item_name hay là Null nếu không có.
- Mỗi lối vào trong bảng tiêu đề có hai trường: Item_name và node_link, nút liên kết trỏ tới nút đầu tiên trong FP_tree có chứa nhãn Item_name.

3.4.2.Thuật toán FP-Growth

Các bước xây dựng cây FP

- **Bước 1:** Tìm tập F các mục phổ biến một mục, Duyệt csdl lần 1.
- **Bước 2** *Sắp xếp các mục trong tập F theo thứ tự giảm dần của độ phổ biến, ta được tập kết quả là L.*
- **Bước 3: Xây dựng cây FP:**
  - Tạo nút gốc cho cây T, và tên của nút gốc sẽ là Null.
  - Duyệt CSDL lần thứ hai. Ứng với mỗi giao tác trong CSDL ta thực hiện 2 công việc sau:
    - Chọn các mục phổ biến trong các giao tác và sắp xếp chúng theo thứ tự giảm dần độ phổ biến trong tập L
    - Gọi hàm Insert_tree([p|P],T) để đưa các mục vào trong cây T

Các bước xây dựng cây FP-Tree



Các bước xây dựng cây FP-Tree

Thủ tục chèn 1 nút vào cây FP

```
procedure Insert_Tree(string [p| P], Tree T)
(trong đó p là mục đầu tiên của dãy và P là
  danh sách còn lại)
{ IF  cây T có nút con N mà N.Item_name=p
  Then
      N.count++;
  Else
      Tạo nút mới N;
      N.Item_Name:= p; N.count:=1;
      Thay đổi nút liên kết cho p.
  IF  P  ≠  ∅ Then
  Insert_Tree(P,N);
}
```


Các bước xây dựng cây FP-Tree

ví dụ 2: Cho cơ sở dữ liệu với các giao tác, ngưỡng hỗ trợ MinSupp = 2/9.

Giao dịch	Tập mục
T01	I1, I2, I5
T02	I2, I4
T03	I2, I3
T04	I1, I2, I4
T05	I1, I3
T06	I2, I3
T07	I1, I3
T08	I1, I2, I3, I5
T09	I1, I2, I3

Ví dụ xây dựng cây FP-Tree

- B1. Duyệt qua CSDL để tìm các mục phổ biến

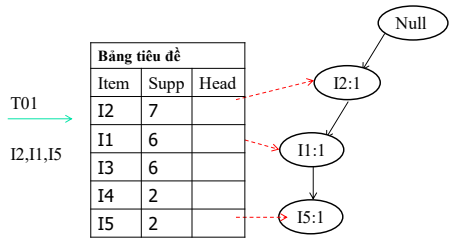
F={{I2:7}, {I1:6}, {I3:6}, {I4:2}, {I5:2}}

- B2. sắp xếp giảm dần theo độ hỗ trợ:

Mục	Số lần xuất hiện
I2	7
I1	6
I3	6
I4	2
I5	2

41

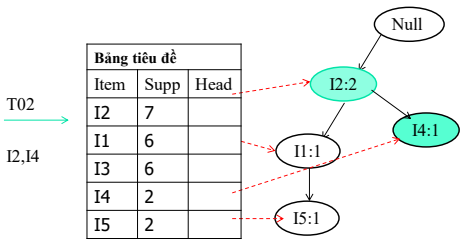
Ví dụ xây dựng cây FP-Tree



42

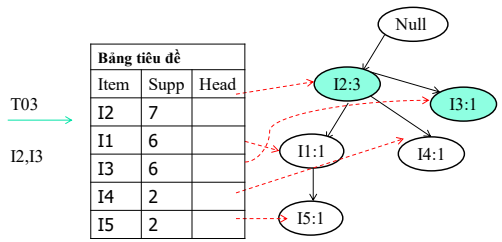
Ví dụ xây dựng cây FP-Tree

- Quá trình xây dựng cây FP-Tree
- Duyệt qua csdl để xây dựng cây



43

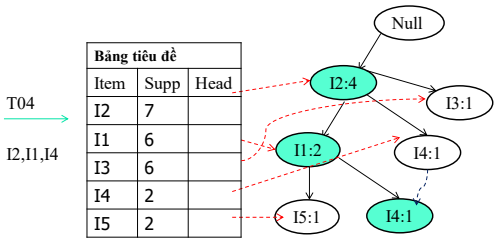
Ví dụ xây dựng cây FP-Tree



44

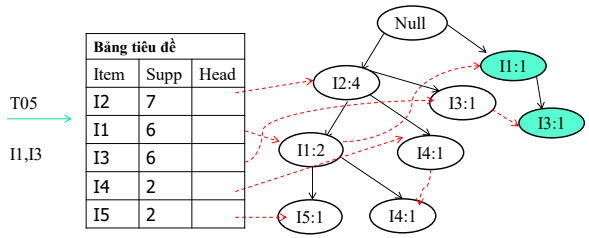
3.4.2.Thuật toán FP-Growth

- Quá trình xây dựng cây FP-Tree
- Duyệt qua csdl để xây dựng cây

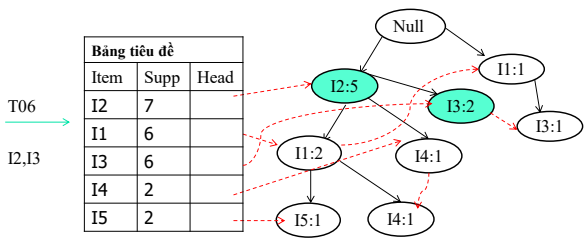


Ví dụ xây dựng cây FP-Tree

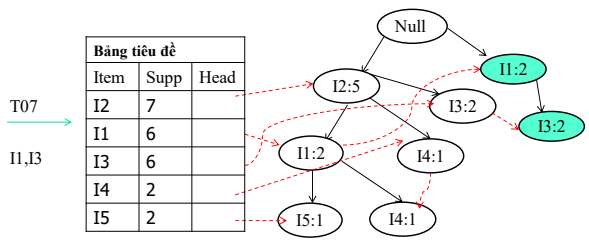
- Quá trình xây dựng cây FP-Tree
- Duyệt qua csdl để xây dựng cây



Ví dụ xây dựng cây FP-Tree



Ví dụ xây dựng cây FP-Tree

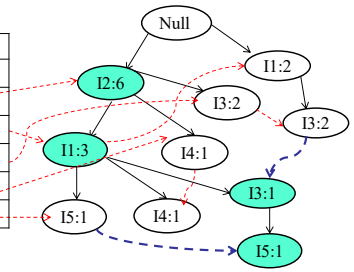


Ví dụ xây dựng cây FP-Tree

T08

I2,I1,I3,I5

Bảng tiêu đề		
Item	Supp	Head
I2	7	
I1	6	
I3	6	
I4	2	
I5	2	

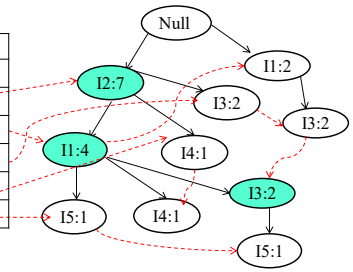


Ví dụ xây dựng cây FP-Tree

T09

I2,I1,I3

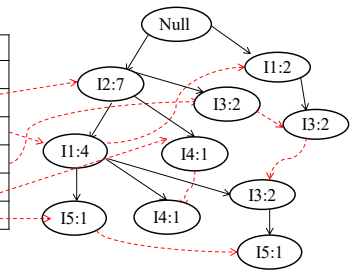
Bảng tiêu đề		
Item	Supp	Head
I2	7	
I1	6	
I3	6	
I4	2	
I5	2	



Ví dụ xây dựng cây FP-Tree

Thu được cây hoàn chỉnh như sau

Bảng tiêu đề		
Item	Supp	Head
I2	7	
I1	6	
I3	6	
I4	2	
I5	2	



Bài tập

- Xây dựng cây FP từ CSDL bên với minsup=25%

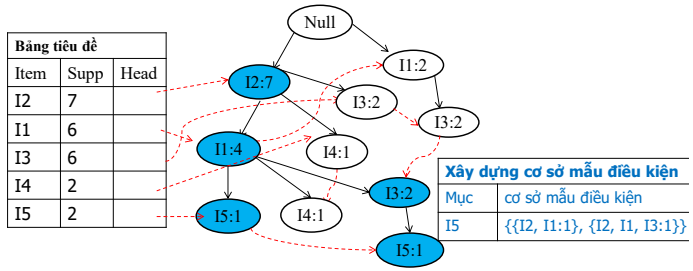
TID	Tập mục
1	A,B
2	B,C,A
3	A,B,D
4	A,B,E
5	A,C
6	B,C
7	B,C,D
8	B,E
9	A,E
10	A,C,E
11	A,D,E

Tìm tập mục phổ biến

- Xác định cơ sở điều kiện
- Bắt đầu từ tập mục phổ biến ở cuối bảng của cây FP.
- *Mỗi mục A dùng nút liên kết để duyệt qua tất cả các nút trên cây mà xuất hiện A, với mỗi nút N có N.Item_name=A tìm tất cả các đường dẫn tiền tố của nút N đó xuất phát từ gốc của cây tới nút N.*
- Xác định tất cả các đường dẫn tiền tố của hạng mục A sao cho tần xuất của các hạng mục trong mỗi đường dẫn bằng tần xuất của A trong đường dẫn đó để tạo cơ sở mẫu điều kiện cho A

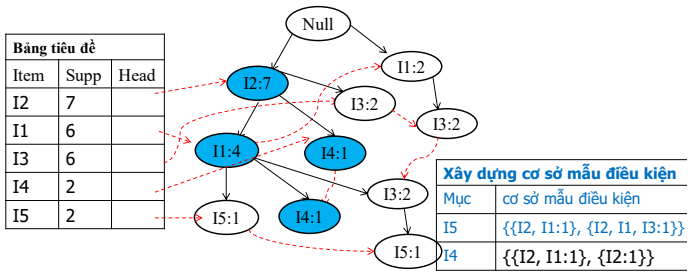
Ví dụ Xác định cơ sở điều kiện

- Bắt đầu từ tập mục phổ biến ở cuối bảng của cây FP. Mục I5
- *Duyệt qua cây FP theo kết nối của mỗi hạn mục phổ biến I5*
- Xác định tất cả các đường dẫn tiền tố của hạng mục *I5, cập nhật tần xuất các hạng mục bằng tần xuất với I5 ở đường dẫn đó.*



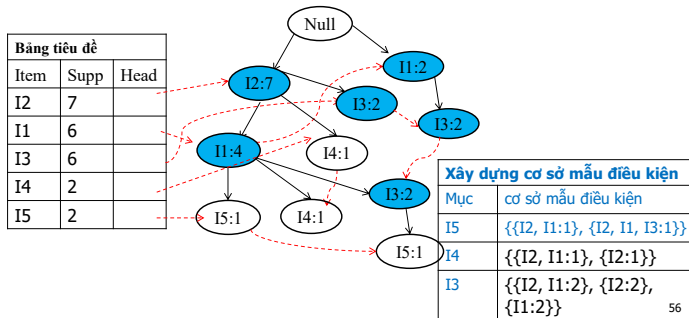
Ví dụ Xác định cơ sở điều kiện

- Mục I4
- *Duyệt qua cây FP theo kết nối của mỗi hạn mục phổ biến I4*
- Xác định tất cả các đường dẫn tiền tố của hạng mục *I4, cập nhật tần xuất các hạng mục bằng tần xuất với I4 ở đường dẫn đó.*



Ví dụ Xác định cơ sở điều kiện

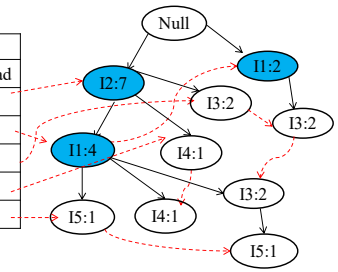
- Mục I3
- *Duyệt qua cây FP theo kết nối của mỗi hạn mục phổ biến I3*
- Xác định tất cả các đường dẫn tiền tố của hạng mục *I3, cập nhật tần xuất các hạng mục bằng tần xuất với I3 ở đường dẫn đó.*



Ví dụ Xác định cơ sở điều kiện

- Mục I1
- *Duyệt qua cây FP theo kết nối của mỗi hạn mục phổ biến I1*
- Xác định tất cả các đường dẫn tiền tố của hạng mục *I1, cập nhật tần xuất các hạng mục bằng tần xuất với I1 ở đường dẫn đó.*

Bảng tiêu đề		
Item	Supp	Head
I2	7	
I1	6	
I3	6	
I4	2	
I5	2	



- Kết quả thu được

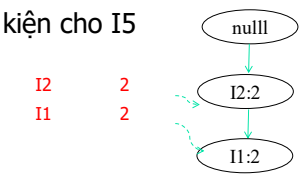
Xây dựng cơ sở mẫu điều kiện	
Mục	cơ sở mẫu điều kiện
I5	{{I2, I1:1}, {I2, I1, I3:1}}
I4	{{I2, I1:1}, {I2:1}}
I3	{{I2, I1:2}, {I2:2}, {I1:2}}
I1	{{I2:4}}

Xây dựng cây FP điều kiện

- Với mỗi mẫu cơ sở điều kiện:
  - Đếm số lượng mỗi mục trong cơ sở mẫu điều kiện. Xác định tập phổ biến 1 mục của mẫu cơ sở.
  - Xây dựng cây FP-điều kiện cho tập phổ biến của mẫu cơ sở điều kiện

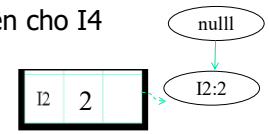
Ví dụ-Xây dựng cây FP điều kiện

- Với mỗi mẫu cơ sở điều kiện của I5 là: {{I2, I1:1}, {I2, I1, I3:1}}
- Đế số lượng mỗi mẫu trong cở sở mẫu điều kiện là: I2:2, I1:2, I3:1 và với minsup=2 nên I2:2, I1:2 phổ biến trên cơ sở mẫu điều kiện của I5. (I3 bị loại vì không thỏa minsup)
- Thiết lập cây FP điều kiện cho I5



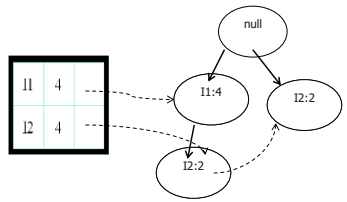
Ví dụ-Xây dựng cây FP điều kiện

- Với mỗi mẫu cơ sở điều kiện của I4 là: {{I2, I1:1}, {I2:1}}
- Đếm số lượng mỗi mẫu trong cở sở mẫu điều kiện là: I2:2, I1:1 và thỏa minsup thì phải có count>=2 nên I2:2 phổ biến trên cơ sở mẫu điều kiện của I4.
- Thiết lập cây FP điều kiện cho I4



Ví dụ-Xây dựng cây FP điều kiện

- Với mỗi mẫu cơ sở điều kiện của I3 là: {{{I2, I1:2}, {I2:2}, {I1:2}}
- Đếm số lượng mỗi mẫu trong cở sở mẫu điều kiện là: I2:4, I1:4 và với minsup=2 nên I2:4, I1:4 phổ biến trên cơ sở mẫu điều kiện của I3.
- Thiết lập cây FP điều kiện cho I3



- Làm tương tự cho I1.

- Kết quả thu được cơ sở mẫu điều kiện và cây FP điều kiện như bảng sau:

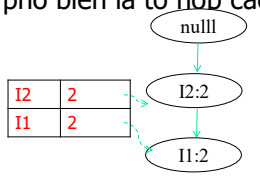
Mục	Cơ sở mẫu có điều kiện	Cây FP có điều kiện
I5	{{I2, I1:1}, {I2, I1, I3:1}}	<I2:2, I1:2>|I5
I4	{{I2, I1:1}, {I2:1}}	<I2:2>|I4
I3	{{I2, I1:2}, {I2:2}, {I1:2}}	<I2:4, I1:2>, <I1:2>|I3
I1	{{I2:4}}	<I2:4>|I1

Tìm tập mục phổ biến

- Nếu cây cây fp- điều kiện này là cây đơn thì tập mục phổ biến là tổ hợp của nút A với các nút trong cây này
- Ngược lại thì thực hiện phân chia cây này thành cây có một đường dẫn đơn bằng cách làm đệ qui lại trên cây fp-điều kiện này.

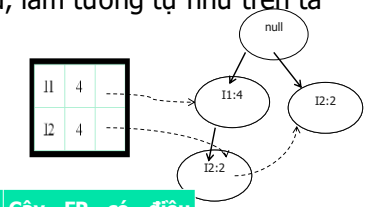
Ví dụ

- Với mục I5 cây điều kiện có 1 đường dẫn đơn
- Có tất cả các tập mục mục phổ biến là tổ hợp các mục trên cây với mục I5
- I5:2,
- I2,I5:2,
- I1,I5:2
- I2,I1,I5:2



Ví dụ

- Cây điều kiện có nhiều đường dẫn
- Tìm cơ sở mẫu điều, làm tương tự như trên ta được



Mục	Cơ sở mẫu có điều kiện	Cây FP có điều kiện
I2I3	{ I1:2}	< I1:2>|I2I3
I1I3	{}	< >

Ví dụ

Tiếp tục với các mục phổ biến còn lại ta có bảng các mẫu cơ sở điều kiện và cây điều kiện, tập mục phổ biến

Mục	Cơ sở mẫu có điều kiện	Cây FP có điều kiện	Mẫu phổ biến được tạo
I5	{{I2,I1:1}, {I2, I1, I3:1}}	<I2:2, I1:2>|I5	{I5:2},{I2,I5:2}, {I1,I5:2}, {I2, I1, I5:2}
I4	{{I2, I1:1}, {I2:1}}	<I2:2>|I4	{I4:2},{I2, I4:2}
I3	{{I2, I1:2}, {I2:2}, {I1:2}}	<I2:4,I1:2>, <I1:2>|I3	{I3:6},{I2, I3:4}, {I1, I3:4}, {I2, I1, I3:2}
I1	{{I2:4}}	<I2:4>|I1	{I1:6},{I2, I1:4}
I2	{}	{}	{I2:7}

3.4.2.Thuật toán FP-Growth

```
Procedure FP-growth(Tree, α)
{
If (Cây FP có chứa một đường đi đơn P) then
      For mỗi tổ hợp (kí hiệu β) của các nút trong
            đường        đi P Do
(3)         Sinh mẫu βUα, support = min(support
            của các nút trong β);
(4) Else ứng với mỗi a_i trong Header của cây Do
      {Sinh mẫu β = a_iUα,
       support = a_i.support
(5)    Tìm cơ sở mẫu điều kiện (phụ thuộc) cho β và
       xây dựng cây FP Treeβ điều kiện của β;
 (7)  If Treeβ ≠ ∅ Then
 (8)         FP-growth(Treeβ, β)
      }
}
```

Bài tập

Sử dụng các ngưỡng *MinSup* = 30% sử dụng thuật toán FP-Growth tìm các tập phổ biến

TID	Items
T01	A1, B1, C2
T02	A2, C1, D1
T03	B2, C2, E2
T04	B1, C1, E1
T05	A3, C3, E2
T06	C1, D2, E2

3.5. Sinh luật từ tập mục phổ biến

- Với mỗi tập mục phổ biến L, cần tìm tất cả các tập con khác rỗng f ⊂ L sao cho: f → {L \ f} thỏa mãn điều kiện về độ tin cậy tối thiểu
- Vd: Với tập mục phổ biến {A,B,C,D}, các luật cần xét gồm có:
- ABC → D, ABD → C, ACD → B, BCD → A,
- A → BCD B → ACD C → ABD D → ABC A → BCD, B → ACD,
- C → ABD, D → ABC
- AB → CD, AC → BD, AD → BC, BC → AD,
- BD → AC, CD → AB,

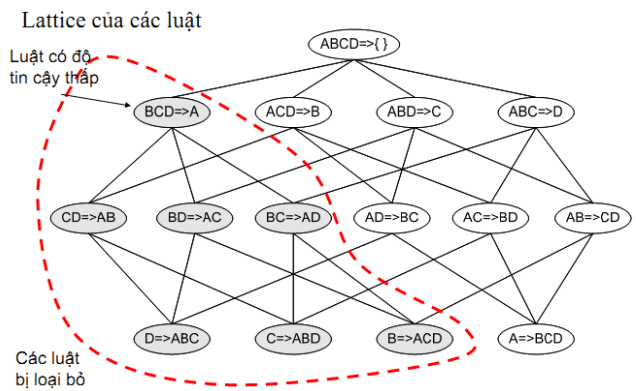
3.5. Sinh luật từ tập mục phổ biến

Sử dụng thuật toán nhanh hơn

Với mỗi tập mục phổ biến l_k

- Sinh tất cả các luật có 1 mục ở phần kết luận
- Sử dụng kết luận của các luật này và hàm apriori-gen trong thuật toán Apriori để sinh tất cả kết luận của luật bao gồm 2-mục, 3-mục…

3.5. Sinh luật từ tập mục phổ biến



3.5. Sinh luật từ tập mục phổ biến

```
for tap muc k-muc l_k , k ≥ 2 do {
        H_1 = {cac ket luan 1−muc cua cac luat sinh tu l_k };
        ap-genrules(l_k,H_1);
}

procedure ap-genrules (l_k:frequent k-items , H_m: frequent m-items )
if (k > m+1) then {
        H_{m+1} = apriori-gen(H_m);
        for(all h_{m+1} ∈ H_{m+1}) do {
                conf = supp(l_k)/supp(l_k − h_{m+1});
                if (conf ≥ minconf) then
                  output the rule (l_k − h_{m+1}) ⇒ h_{m+1}
                else
                        delete h_{m+1} from H_{m+1}
                }
        }
        call ap-genrules(l_k,h_{m+1});
}
```

Tìm tất cả các kết luận một mục

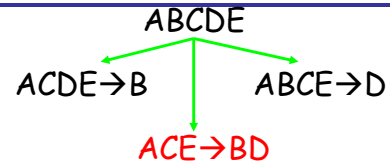
Sinh kết luận mới có (m+1)-mục

Kiểm tra độ tin cậy của luật mới

Tiếp tục cho những luật có kết

Nếu không thỏa độ tin cậy

Ví dụ



Ví dụ

- Lấy kết quả các tập mục phổ biến trong ví dụ 1 để sinh các luật thỏa mãn độ tin cậy minsup=50%, minconf=70%
- Ta có các tập mục phổ biến 2 mục
  - AC, BC, BE,CE

  Ta có các luật sau

A->C conf=2/2=100% sup=2/4=50%

C->A

B->C conf= 2/3=66.67% sup=2/4=50%

C->B

B->E conf= 3/3=100% sup= 3/4= 75%

E->B

C->E conf= 2/3=66.67% sup= 2/4=50%

E->C

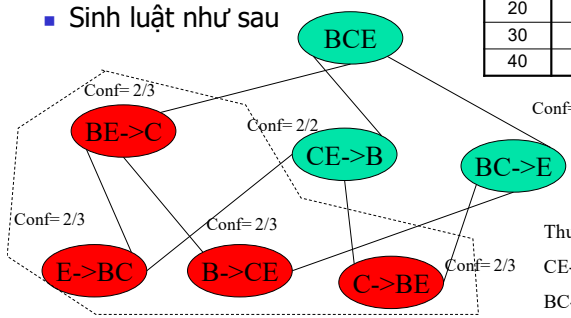
Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

- TÍNH TIẾP ĐỘ HỖ TRỢ VÀ ĐỘ TIN CẬY

Ví dụ

- Tập mục phổ biến 3-mục BCE
- Sinh luật như sau

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E



Thu được 2 luật

CE->B conf=100%

BC->E conf=100%

BÀI TẬP

Bài 1

- Cho CSDL giao dịch sau và *minsupp = 40%, minconf = 80%*
- 1.Hãy sử dụng thuật toán **Apriori** và **FP-Growth** để tìm tất cả các tập phổ biến
- 2. Tìm các luật kết hợp được xây dựng từ các tập phổ biến thỏa mãn các ngưỡng *minsupp, minconf* đã cho

TID	Items
100	K, D, A, B, C,F
200	A, H, C, D
300	C, I, D, E, G, F
400	B, C, H, A, I D, F, G
500	F, C, K, E, G

BÀI TẬP

Bài 2:

- Sử dụng thuật toán Apriori
  - Tìm các tập phổ biến có ngưỡng *MinSup=60%*
  - Tìm các luật kết hợp có ngưỡng MinSup=60% và MinConf >= 70%

TID	Items
100	f,a,b,d,g,i,m,p
200	a,b,c,f,l,m,o
300	a,c,h,j,o
400	b,c,k,s,p
500	a,f,b,c,l,p,m,n

BÀI TẬP

Bài 3

Cho csdl sau với *minsupp=20%, minconf=50%*

a)Tìm tất cả các tập ứng cử viên và tập phổ biến sử dụng lần lượt thuật toán Apriori và thuật toán Fp-Growth

b) *Liệt kê tất cả LKH thỏa mãn ngưỡng đã cho*

TID	Tập mục
T1	A,B,E
T2	B, D
T3	B, C
T4	A,B, D
T5	A, C
T6	B, C
T7	A, C
T8	A, B, C, E
T9	A, B, C