Trường Đại học Khoa học Tự nhiên
Khoa Công nghệ Thông tin

TÀI LIỆU LÝ THUYẾT KTDL & UD

KHAI THÁC MẪU PHỔ BIẾN và LUẬT KẾT HỢP (P2)

Giảng viên: ThS. Lê Ngọc Thành
Email: lnthanh@fit.hcmus.edu.vn

Summer 2012

Nội dung

- Thuật toán FP-Growth
  - Xây dựng cây FP-Growth
  - Phát sinh mẫu phổ biến từ FP-Growth
- So sánh Fp-Growth và Apriori
- Độ do tính lý thú của LKH

Nhắc lại hạn chế Apriori

- Các hạn chế của thuật toán Apriori
  - Phải duyệt CSDL nhiều lần
  - Kiểm tra và tạo lượng lớn tập ứng viên
- Ví dụ:
  - Nếu có 10^4 mẫu phổ biến 1-hạng mục thì cần phát sinh nhiều hơn 10^7 ứng viên 2-hạng mục
  - Để tìm mẫu phổ biến i_1 i_2… i_{100}:
    - Số lần duyệt CSDL: 100
    - Số lượng ứng viên ít nhất: $2^{100}-1 = 10^{30}$

Giới thiệu thuật toán FP-Growth

- Được đề xuất bởi J.Han, J.Pei và Y.Yin trong hội nghị SIGMOD năm 2000.
- Là thuật toán tìm kiếm theo chiều sâu (theo tư tưởng chia để trị)
- Khai thác mẫu phổ biến không sử dụng hàm tạo ứng viên.

Ý tưởng FP-Growth

Nén CSDL thành cấu trúc cây FP (Frequent Patern)

Chia dữ liệu đã nén thành các dữ liệu có điều kiện

Thực hiện khai thác trên mỗi cơ sở dữ liệu riêng rẽ

Thuật toán FP-Growth

- Input: dữ liệu giao tác D và minsup
- Output: tập mẫu phổ biến
- B1: Xây dựng cây FP
- B2: Khai thác cây FP

B1. Xây dựng FP-tree

- ***B1.1:*** (Duyệt CSDL lần 1) Tìm tập phổ biến 1- hạng mục. Sắp xếp tập phổ biến giảm dần vào trong F-list .
- ***B1.2:*** Chọn lọc và sắp xếp mỗi giao tác trong CSDL lại theo thứ tự trong F-list.
- ***B1.3:*** Khởi tạo cây với gốc là "null".
- ***B1.4:*** (Duyệt CSDL lần 2) Với mỗi giao tác ở B1.2, xuất phát từ gốc, lấy từng hạng mục theo thứ tự:
  - Nếu hạng mục chưa có ở vị trí con của node hiện tại thì thêm vào cây với đếm là 1.
  - Nếu hạng mục đã có ở vị trí con của node hiện tại thì tăng đếm lên 1.

Ví dụ xây dựng cây FP (1/3)

TID	*Items bought*	*(ordered) frequent items*
100	{*f, a, c, d, g, i, m, p*}	{*f, c, a, m, p*}
200	{*a, b, c, f, l, m, o*}	{*f, c, a, b, m*}
300	{*b, f, h, j, o, w*}	{*f, b*}
400	{*b, c, k, s, p*}	{*c, b, p*}
500	{*a, f, c, e, l, p, m, n*}	{*f, c, a, m, p*}

minsupp = 3

B1.1: Tìm tập phổ biến 1-hạng mục. Sắp xếp tập phổ biến giảm dần vào trong F-list

F-list=f-c-a-b-m-p

B1.2: Chọn lọc và sắp xếp CSDL theo F-list.

Header Table

Item	*Supp.count*
f	*4*
c	*4*
a	*3*
b	*3*
m	*3*
p	*3*

Ví dụ xây dựng cây FP (2/3)

TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

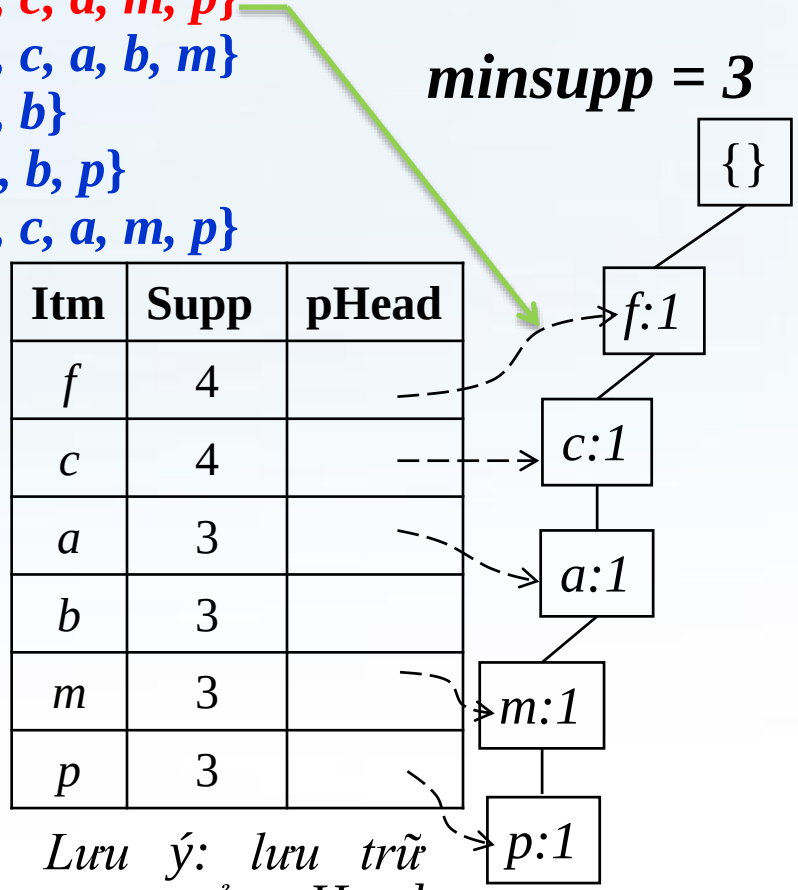
B1.3: Khởi tạo cây với gốc "null"

B1.4: Với mỗi giao tác ở B1.2, xuất phát từ gốc, lấy từng hạng mục theo thứ tự:

- Nếu hạng mục chưa có ở vị trí con của node hiện tại thì thêm vào cây với đếm là 1.
- Nếu hạng mục đã có ở vị trí con của node hiện tại thì tăng đếm lên 1

Itm	Supp	pHead
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

Lưu ý: lưu trữ con trỏ pHead của DSLK để tiện truy cập về sau



Ví dụ xây dựng cây FP (2/3)

TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

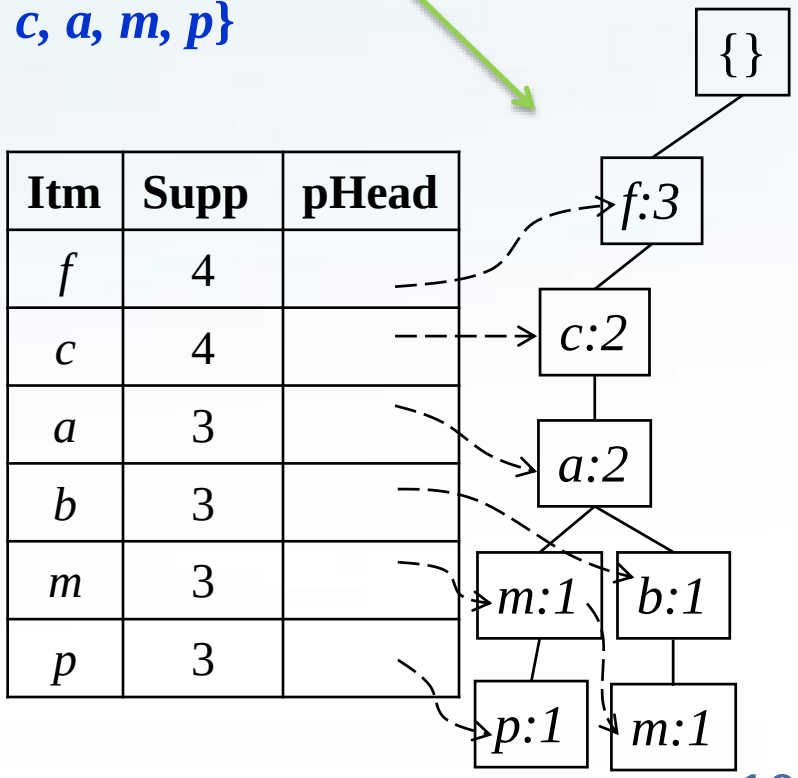
minsupp = 3

B1.3: Khởi tạo cây với gốc "null"

B1.4: Với mỗi giao tác ở B1.2, xuất phát từ gốc, lấy từng hạng mục theo thứ tự:

- Nếu hạng mục chưa có ở vị trí con của node hiện tại thì thêm vào cây với đếm là 1.
- Nếu hạng mục đã có ở vị trí con của node hiện tại thì tăng đếm lên 1

Itm	Supp	pHead
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



Ví dụ xây dựng cây FP (2/3)

TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

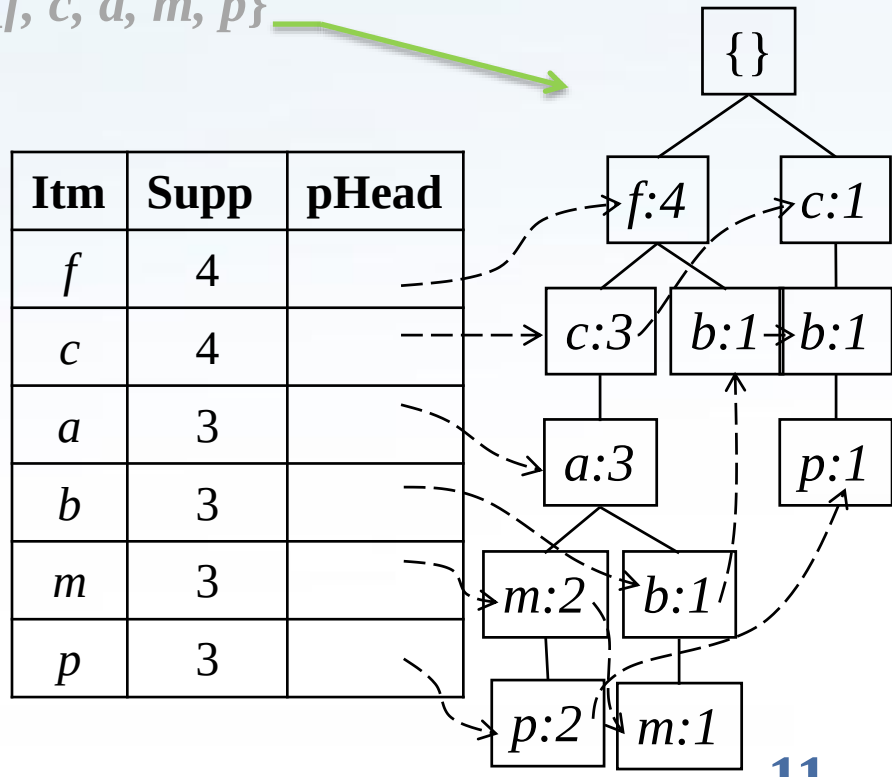
minsupp = 3

B1.3: Khởi tạo cây với gốc "null"

B1.4: Với mỗi giao tác ở B1.2, xuất phát từ gốc, lấy từng hạng mục theo thứ tự:

- Nếu hạng mục chưa có ở vị trí con của node hiện tại thì thêm vào cây với đếm là 1.
- Nếu hạng mục đã có ở vị trí con của node hiện tại thì tăng đếm lên 1

Itm	Supp	pHead
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



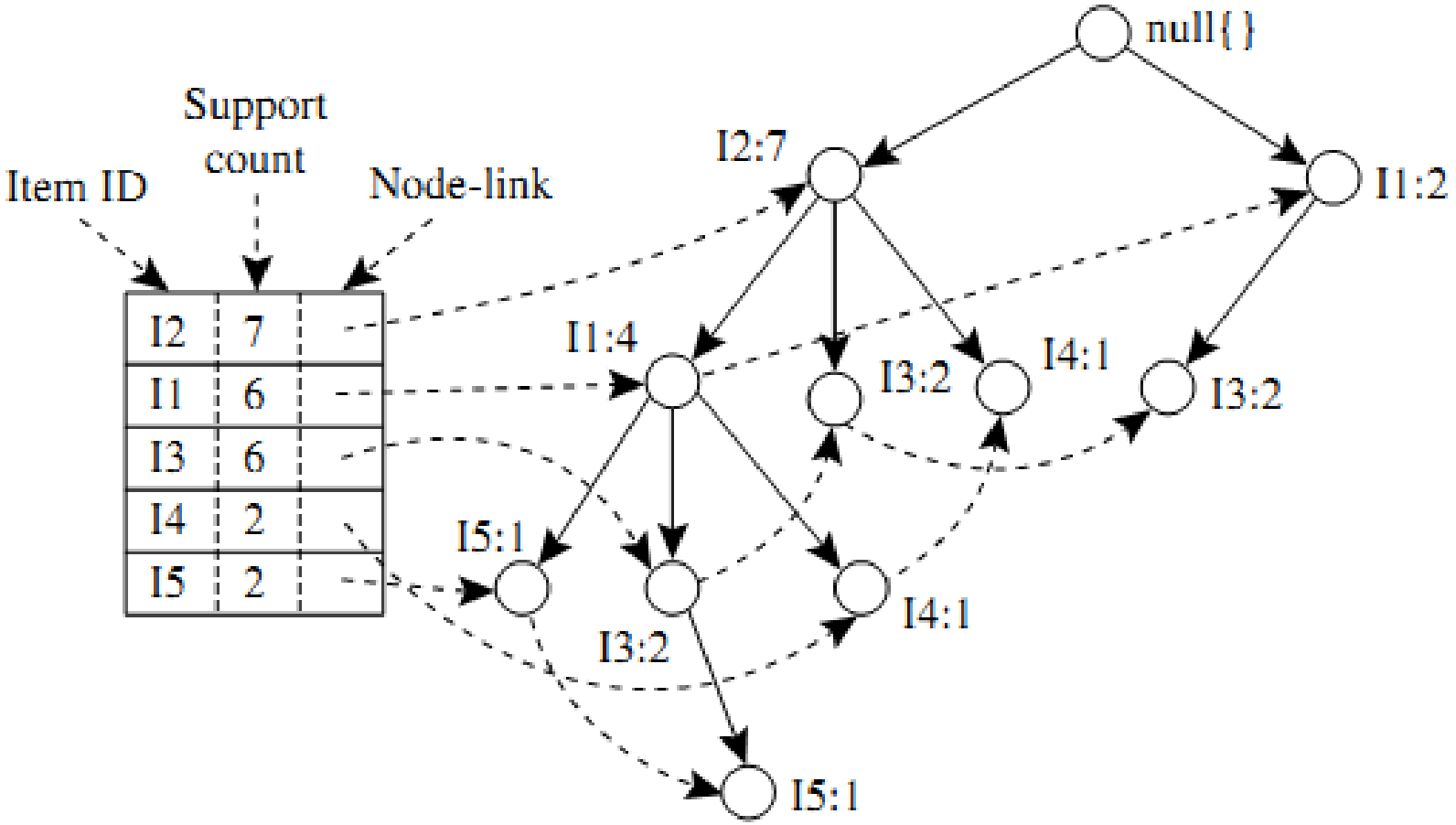
Bài tập áp dụng 1

Xây dựng cây FP cho dữ liệu giao tác sau với minsup = 2 hay 22%:

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Bài tập áp dụng 1 – Đáp án

Cây FP được xây dựng:



B2. Khai thác FP-tree

- FP-tree được khai thác bằng cách gọi hàm FP_growth(FP_tree,null):

```
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); }
```

B2. Khai thác FP-tree

- FP

Với mỗi mẫu phổ biến a_i trong bảng header (từ dưới lên)

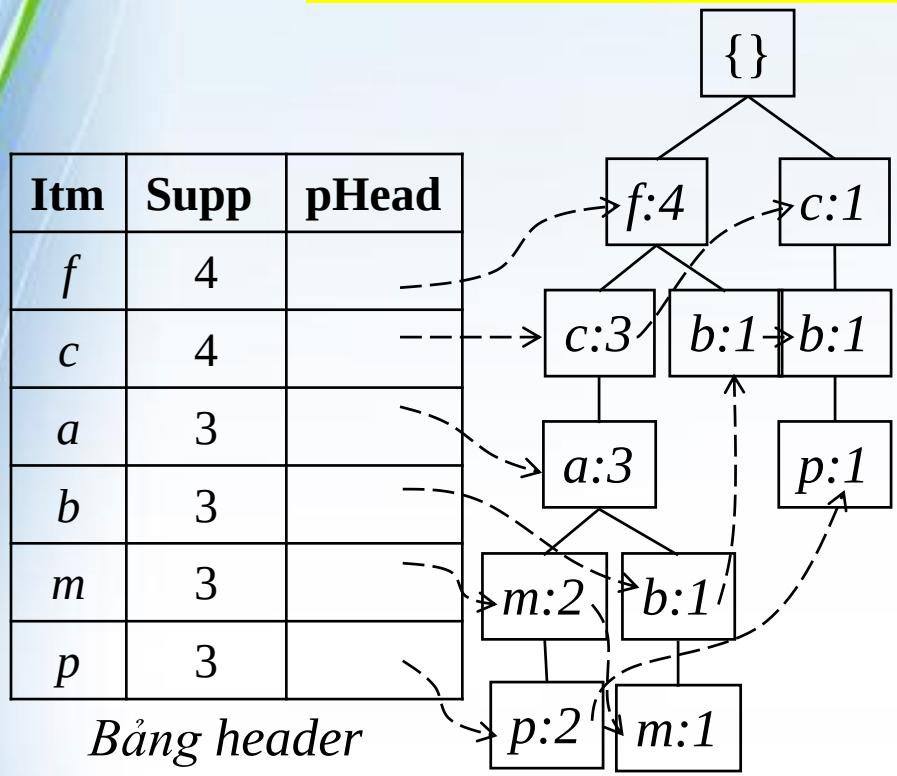
1. Tạo mẫu hậu tố β bằng cách kết hợp mẫu a_i với mẫu điều kiện α trước đó và đếm trợ = đếm trợ của a_i
2. Xây dựng cơ sở mẫu điều kiện β
3. Xây dựng cây FP điều kiện β
4. Gọi đệ quy hàm cho cây FP-điều kiện β nếu cây tồn tại

```
procedure FP_growth
(1)   if Tree contains
(2)      for each con
(3)         generate
(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); }
```

B2.1.Tạo mẫu điều kiện β

Với mỗi hạng mục phổ biến a_i trong bảng header (từ dưới lên)

1. Tạo mẫu điều kiện β bằng cách kết hợp hạng mục a_i với mẫu điều kiện trước đó và đếm trợ = đếm trợ của a_i



Itm	Supp	pHead
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

Bảng header

β	Supp
p	3
m	3
b	3
a	3
c	4
f	4

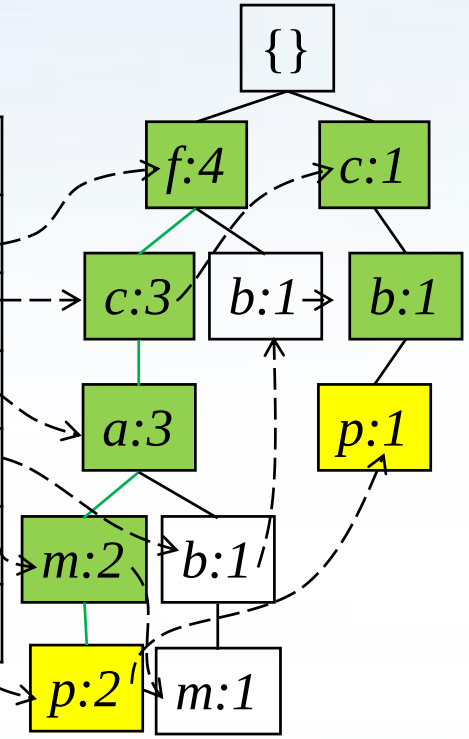
Tạo mẫu điều kiện

B2.2. Xd cơ sở mẫu điều kiện β

- Dựa trên FP-tree và theo kết nối của mỗi mẫu.
- Gom tất cả đường dẫn tiền tố biến đổi (transformed prefix) của mẫu để tạo cơ sở mẫu điều kiện

Itm	Supp	pHead
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

Bảng header



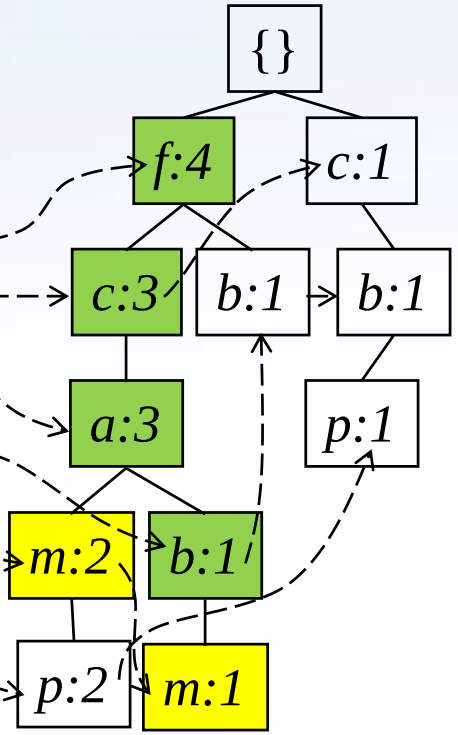
β	Supp	Cond.Pattern Base
p	3	{*fcam:2*}, {*cb:1*}
m	3	
b	3	
a	3	
c	4	
f	4	

B2.2. Xd cơ sở mẫu điều kiện β

- Dựa trên FP-tree và theo kết nối của mỗi mẫu.
- Gom tất cả đường dẫn tiền tố biến đổi (transformed prefix) của mẫu để tạo cơ sở mẫu điều kiện

Itm	Supp	pHead
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

Bảng header



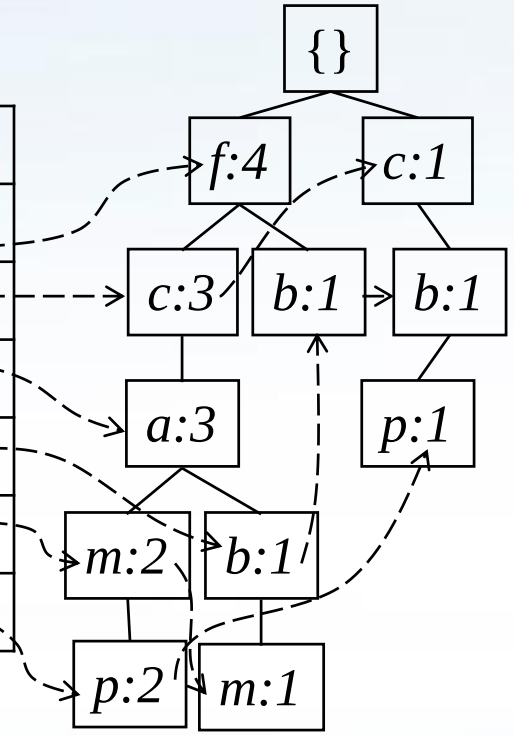
β	Supp	Cond.Pattern Base
p	3	{*fcam:2*}, {*cb:1*}
m	3	{*fca:2*}, {*fcab:1*}
b	3	
a	3	
c	4	
f	4	

B2.2. Xd cơ sở mẫu điều kiện β

- Dựa trên FP-tree và theo kết nối của mỗi mẫu.
- Gom tất cả đường dẫn tiền tố biến đổi (transformed prefix) của mẫu để tạo cơ sở mẫu điều kiện

Itm	Supp	pHead
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

Bảng header



β	Supp	Cond.Pattern Base
p	3	{*fcam:2*}, {*cb:1*}
m	3	{*fca:2*}, {*fcabm:1*}
b	3	{*fca:1*}, {*f:1*}, {*c:1*}
a	3	{*fc:3*}
c	4	{*f:3*}
f	4	{ }

B2.3. Xd FP-tree điều kiện β

- Xem cơ sở mẫu điều kiện như là những giao tác.
- Thực hiện xây dựng cây FP-tree cho từng cơ sở như bước B1 (chọn lọc hạng mục phổ biến, x/d) cây từ mỗi giao tác đã chọn lọc

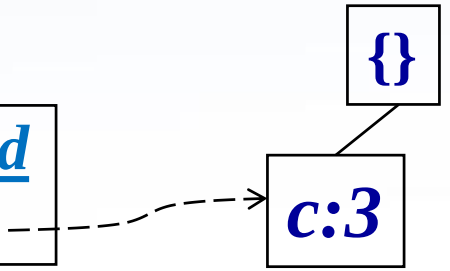
minsupp = 3

β	Supp	Cond.Pattern Base
p	3	{*fcam:2*}, {*cb:1*}
m	3	{*fca:2*}, {*fcab:1*}
b	3	{*fca:1*}, {*f:1*}, {*c:1*}
a	3	{*fc:3*}
c	4	{*f:3*}
f	4	{ }

***p-conditional* FP-tree**

Bảng header

Itm	***Supp***	***pHead***
c	***3***	



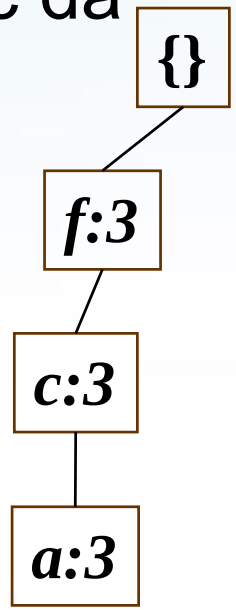
f,a,m,b không thỏa minsup nên bị loại

B2.3. Xd FP-tree điều kiện β

- Xem cơ sở mẫu điều kiện như là những giao tác.
- Thực hiện xây dựng cây FP-tree cho từng cơ sở như bước B1 (chọn lọc hạng mục phổ biến, x/d) cây từ mỗi giao tác đã chọn lọc

minsupp = 3

β	Supp	Cond.Pattern Base	Cond.FP-Tree
p	3	{*fcam:2*}, {*cb:1*}	<*c:3*>
m	3	{*fca:2*}, {*fcab:1*}	<*f:3,c:3,a:3*>
b	3	{*fca:1*}, {*f:1*}, {*c:1*}	
a	3	{*fc:3*}	
c	4	{*f:3*}	
f	4	{ }	



B2.3. Xd FP-tree điều kiện β

- Xem cơ sở mẫu điều kiện như là những giao tác.
- Thực hiện xây dựng cây FP-tree cho từng cơ sở như bước B1 (chọn lọc hạng mục phổ biến, x/d) cây từ mỗi giao tác đã chọn lọc

minsupp = 3

β	Supp	Cond.Pattern Base	Cond.FP-Tree
p	3	{*fcam:2*}, {*cb:1*}	<*c:3*>
m	3	{*fca:2*}, {*fcab:1*}	<*f:3,c:3,a:3*>
b	3	{*fca:1*}, {*f:1*}, {*c:1*}	< >
a	3	{*fc:3*}	<*f:3,c:3*>
c	4	{*f:3*}	<*f:3*>
f	4	{ }	< >

B2. Khai thác FP-tree

- FP-tree được khai thác bằng cách gọi hàm FP_growth(FP_tree,null):

```
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
(5)     generate p
(6)     construct
(7)     if Tree_β ≠ ∅ then
(8)       call FP_growth(Tree_β, β); }
```

Nếu cây FP chứa một nhánh đơn P

i. Với mỗi cách kết hợp các node trong nhánh P (kí hiệu β) Phát sinh mẫu phổ biến = mẫu β hợp với mẫu điều kiện α trước đó và độ trợ = độ trợ của node nhỏ nhất trong β

B2.i. Phát sinh mẫu phổ biến

- Nếu cây FP chỉ có một nhánh thì phát sinh mẫu phổ biến bằng cách kết hợp các tổ hợp trên nhánh với mẫu điều kiện.
  - Độ trợ của mẫu là độ trợ nhỏ nhất trong tổ hợp
  - Lưu ý: với tổ hợp rỗng thì mẫu phổ biến chính là mẫu điều kiện và độ trợ bằng độ trợ mẫu điều kiện
- Nếu cây FP có nhiều nhánh thì quá trình đệ qui như các bước trước (x/d mẫu điều kiện, cơ sở mẫu điều kiện, cây FP điều kiện)

B2.i. Phát sinh mẫu phổ biến

- Nếu cây FP chỉ có một nhánh thì phát sinh mẫu phổ biến bằng cách kết hợp các tổ hợp trên nhánh với mẫu điều kiện.
  - Độ trợ của mẫu là độ trợ nhỏ nhất trong tổ hợp
  - Lưu ý: với tổ hợp rỗng thì mẫu phổ biến chính là mẫu điều kiện và độ trợ bằng độ trợ mẫu điều kiện

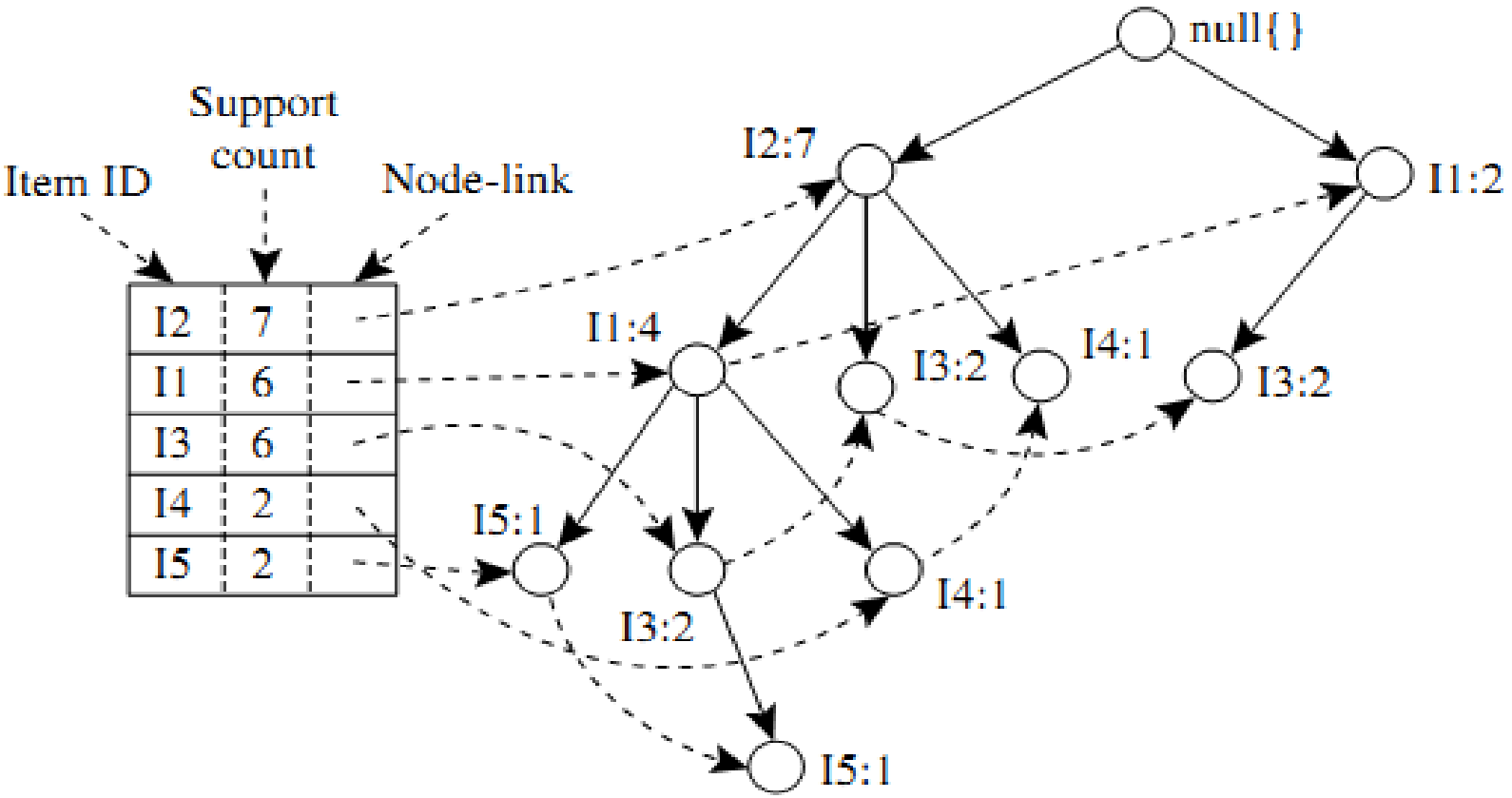
α	Supp	Cond.Pattern Base	Cond.FP-Tree	Frequent Patterns
p	3	{*fcam:2*}, {*cb:1*}	<*c:3*>	{*p:3*}, {*cp:3*}
m	3	{*fca:2*}, {*fcab:1*}	<*f:3,c:3,a:3*>	
b	3	{*fca:1*}, {*f:1*}, {*c:1*}	< >	
a	3	{*fc:3*}	<*f:3,c:3*>	
c	4	{*f:3*}	<*f:3*>	
f	4	{ }	< >	

B2.i. Phát sinh mẫu phổ biến

α	Supp	Cond.Pattern Base	Cond.FP-Tree	Frequent Patterns
p	3	{*fcam:2*}, {*cb:1*}	<*c:3*>	{*p:3*}, {*cp:3*}
m	3	{*fca:2*}, {*fcab:1*}	<*f:3,c:3,a:3*>	{*m:3*}, {*fm:3*}, {*cm:3*}, {*am:3*}, {*fcm:3*}, {*fam:3*}, {*cam:3*}, {*fcam:3*}
b	3	{*fca:1*}, {*f:1*}, {*c:1*}	< >	{*b:3*}
a	3	{*fc:3*}	<*f:3,c:3*>	{*a:3*}, {*fa:3*}, {*ca:3*}, {*fca:3*}
c	4	{*f:3*}	<*f:3*>	{*c:4*}, {*fc:3*}
f	4	{ }	< >	{*f:4*}

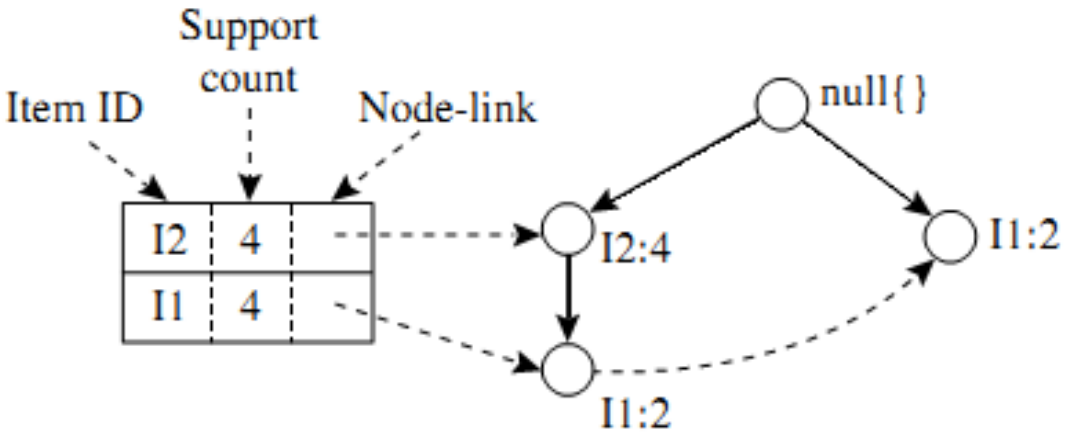
Bài tập áp dụng 2

Tìm mẫu phổ biến từ cây FP đã xây dựng trong bài tập 1 (minsup = 2):



Bài tập áp dụng 2 – Đáp án

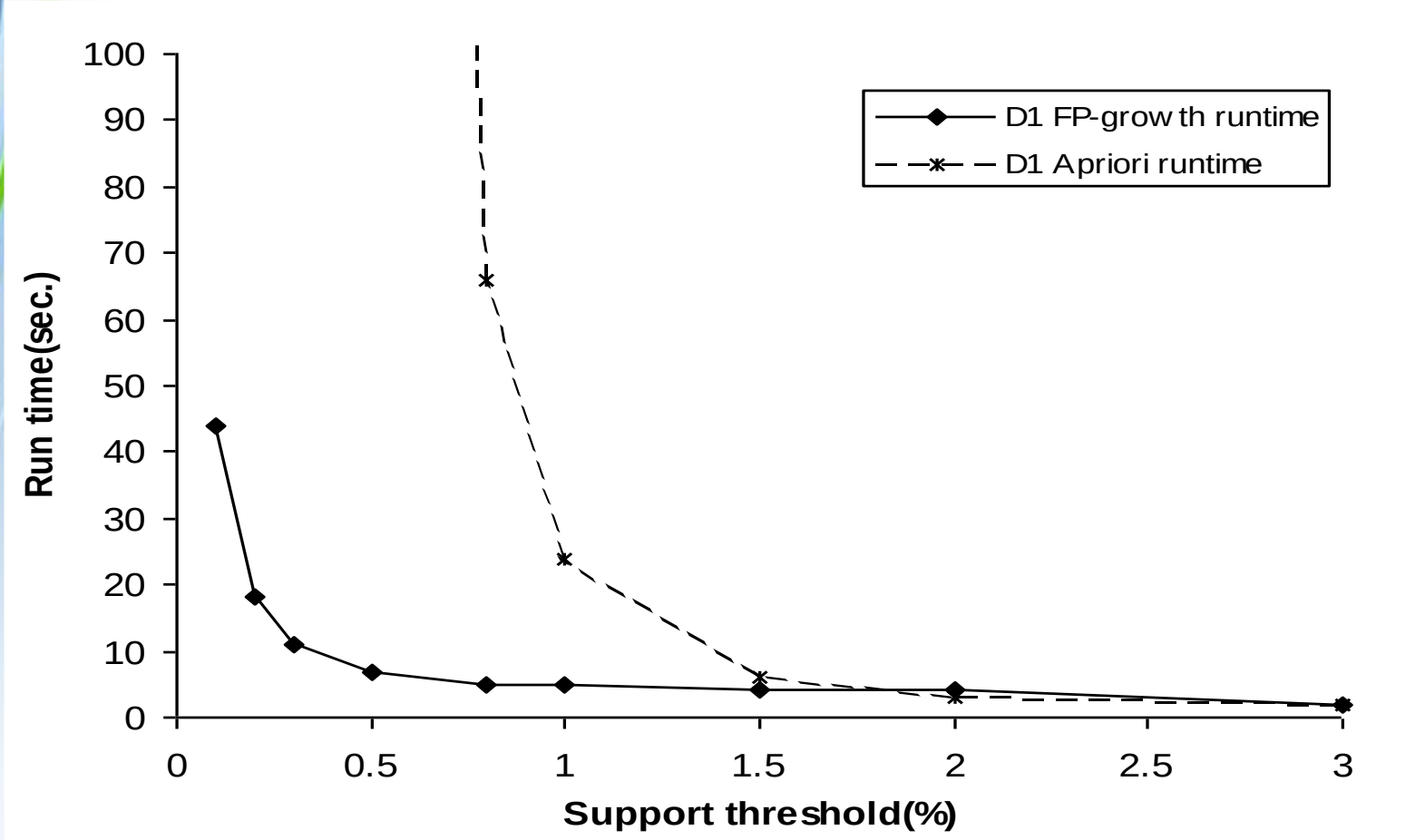
Item	*Conditional Pattern Base*	*Conditional FP-tree*	*Frequent Patterns Generated*
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}



Cây triển khai tiếp từ I3

So sánh Apriori và FP-Growth

Data set T25I20D10K



Nội dung

- Thuật toán FP-Growth
  - Xây dựng cây FP-Growth
  - Phát sinh mẫu phổ biến từ FP-Growth
- So sánh Fp-Growth và Apriori
- Độ do tính lý thú của LKH

Vấn đề khi phát sinh luật

- Thuật toán khai thác LKH có xu hướng sinh ra quá nhiều luật khi minsup và minconf thấp.
- Trong đó có nhiều luật không hay hoặc bị thừa, thậm chí là sai.

Ví dụ 1 về v/đ phát sinh luật

Thống kê số giao tác/hóa đơn của một cửa hàng bán nước:

	$Coffee$	$\overline{Coffee}$	$\Sigma_{\text{dòng}}$
Tea	15	5	20
$\overline{Tea}$	75	5	80
$\Sigma_{\text{cột}}$	90	10	100

Luật kết hợp: **Tea → Coffee** [sup = 15%; conf = 75%]

→ Mặc dù độ tin cậy cao nhưng luật gây ra sự hiểu lầm vì thực tế P(coffee) = 90% còn cao hơn conf của luật.

$$P(coffee|\overline{tea}) = \frac{75}{80} = 0.9375$$

Ví dụ 2 về v/đ phát sinh luật

Thống kê về mối liên hệ giữa SV chơi bóng rổ và ăn sáng với ngũ cốc:

	$Chơi\ bóng$	$\overline{Ko\ chơi\ bóng}$	$\Sigma_{dòng}$
$Ăn$	4000	3500	7500
$\overline{Ko\ ăn}$	2000	500	2500
$\Sigma_{cột}$	6000	4000	10,000

LKH:**Chơi bóng rổ → Ăn ngũ cốc** [sup=40%; conf=66%]

- Ta thấy, phần trăm SV ăn ngũ cốc P(Ăn ngũ cốc) = 75% lại còn cao hơn cả conf của luật.
- Thực tế, LKH: Chơi bóng rổ → Không ăn ngũ cốc [20%; 34%] chính xác hơn.

Độ đo tính lý thú của luật

- Luật thế nào là tốt/lý thú (interesting)?

→ tùy vào dữ liệu và mang tính chủ quan (lý thú với người này nhưng lại không lý thú với người khác)

→ Một cách tương đối, bằng pp thống kê hướng đến loại bỏ các luật không lý thú.

- Các độ đo lý thú là gì?
  - Độ đo tương quan giữa tiền đề và kết luận (correlation)
  - Độ đo khả năng dẫn xuất (implication)

Các độ đo mối tương quan

symbol	measure	range	formula
ϕ	ϕ-coefficient	-1...1	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
Q	Yule's Q	-1 ...1	$\frac{P(A,B)P(\overline{A},\overline{B})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{A},\overline{B})+P(A,\overline{B})P(\overline{A},B)}$
Y	Yule's Y	-1 ...1	$\frac{\sqrt{P(A,B)P(\overline{A},\overline{B})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{A},\overline{B})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}}$
k	Cohen's	-1 ...1	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
PS	Piatetsky-Shapiro's	-0.25 ...0.25	$P(A,B)-P(A)P(B)$
F	Certainty factor	-1 ...1	$\max(\frac{P(B\|A)-P(B)}{1-P(B)}, \frac{P(A\|B)-P(A)}{1-P(A)})$
AV	added value	-0.5 ...1	$\max(P(B\|A)-P(B), P(A\|B)-P(A))$
K	Klosgen's Q	-0.33 ...0.38	$\sqrt{P(A,B)}\max(P(B\|A)-P(B), P(A\|B)-P(A))$
g	Goodman-kruskal's	0 ...1	$\frac{\Sigma_j \max_k P(A_j,B_k)+\Sigma_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-max_k P(B_k)}$
M	Mutual Information	0 ...1	$\frac{\Sigma_i\Sigma_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\Sigma_i P(A_i)\log P(A_i), -\Sigma_i P(B_i)\log P(B_i))}$
J	J-Measure	0 ...1	$\max(P(A,B)\log(\frac{P(B\|A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}))$ $P(A,B)\log(\frac{P(A\|B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})$
G	Gini index	0 ...1	$\max(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A}[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B}[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]-P(A)^2-P(\overline{A})^2)$
s	support	0 ...1	$P(A,B)$
c	confidence	0 ...1	$max(P(B\|A), P(A\|B))$
L	Laplace	0 ...1	$\max(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2})$
IS	Cosine	0 ...1	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
γ	coherence(Jaccard)	0 ...1	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
α	all_confidence	0 ...1	$\frac{P(A,B)}{\max(P(A),P(B))}$
o	odds ratio	0 ...∞	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(\overline{A},B)P(A,\overline{B})}$
V	Conviction	0.5 ...∞	$\max(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})})$
λ	lift	0 ...∞	$\frac{P(A,B)}{P(A)P(B)}$
S	Collective strength	0 ...∞	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})}\times\frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
χ^2	χ^2	0 ...∞	$\Sigma_i\frac{(P(A_i)-E_i)^2}{E_i}$

Thảo luận độ đo *lift* (1/2)

- Độ đo ***lift*** cho luật A→ B:

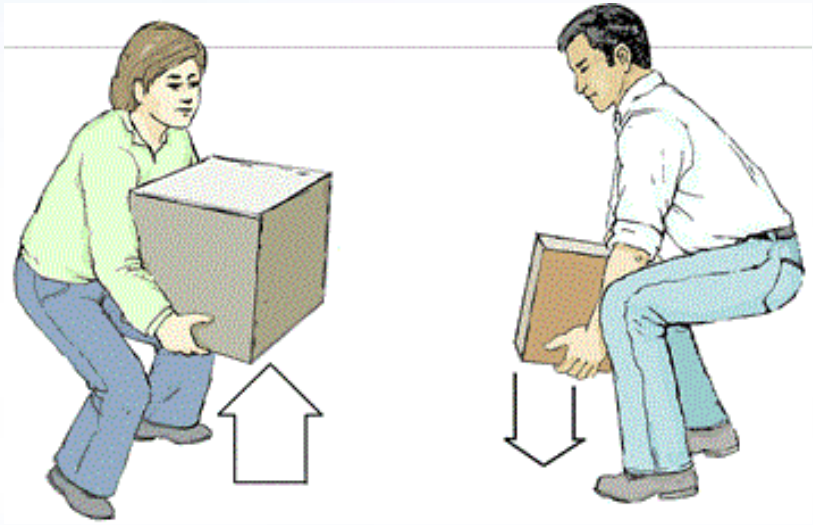
$$lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)}$$

- Thể hiện hạng mục A có xảy ra độc lập với hạng mục B hay không?
  - Nếu độc lập: P(A,B) = P(A).P(B), hay ***lift*** (A,B) = 1.
  - Tương quan nghịch (negatively correlated): ***lift*** (A,B) < 1 nghĩa là cái này xảy ra dẫn đến cái kia sẽ không xảy ra.
  - Tương quan thuận (positively correlated): ***lift*** (A,B) > 1, nghĩa là cái này xảy ra dẫn đến cái kia cũng xảy ra.

Thảo luận độ đo ***lift*** (1/2)

- Nói cách khác:
  - A tăng cường hay "nâng" (***lift***) khả năng xảy ra của B với hệ số là giá trị của biểu thức ***lift***(A,B)

$$lift(A,B) = \frac{P(A,B)}{P(A)P(B)} = \frac{P(B|A)}{P(B)} = \frac{conf(A \rightarrow B)}{sup(B)}$$



Ví dụ độ đo *lift*

	$Chơi\ bóng$	$\overline{Ko\ chơi\ bóng}$	$\Sigma_{\text{dòng}}$
$Ăn$	4000	3500	7500
$\overline{Ko\ ăn}$	2000	500	2500
$\Sigma_{\text{cột}}$	6000	4000	10,000

lift (Chơi bóng rổ $\rightarrow$ Ăn ngũ cốc) =

$$\frac{P(Chơi\ bóng,Ăn)}{P(Chơi\ bóng)\times P(Ăn)} = \frac{\frac{4000}{10000}}{\frac{6000}{10000}\times\frac{7500}{10000}} = 0.89 < 1$$

lift (Chơi bóng rổ $\rightarrow \overline{Ăn\ ngũ\ cốc}$= ?

Bài tập áp dụng 3

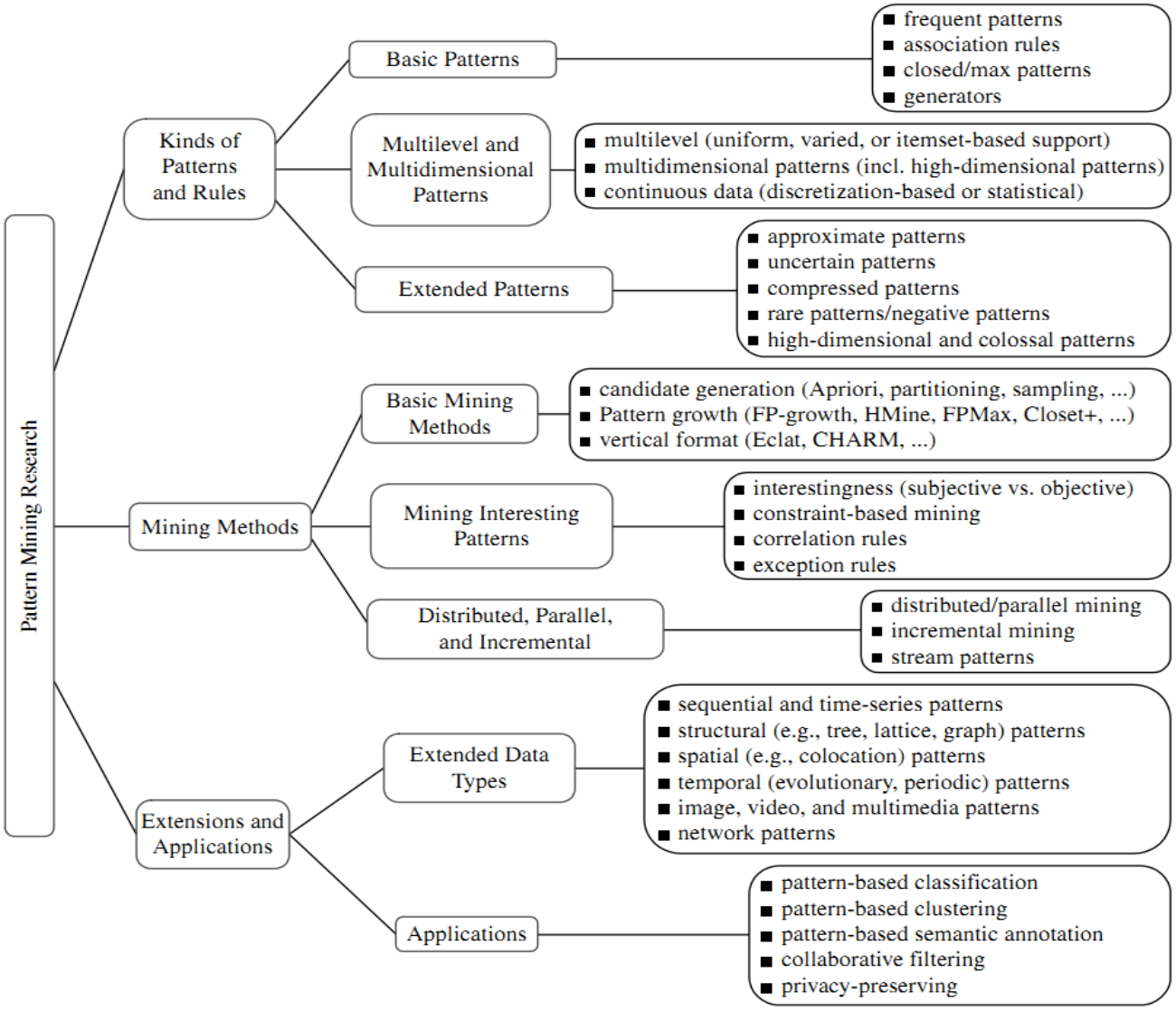
- Phát sinh các luật dựa trên các mẫu phổ biến trong bài tập 2 thỏa minsup = 22% và minconf = 70%
- Tính độ đo ***lift*** của từng luật trên.

Item	*Conditional Pattern Base*	*Conditional FP-tree*	*Frequent Patterns Generated*
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}

So sánh các độ đo tương quan

- Đọc thêm trong phần 6.3.2 và 6.3.3 cuốn Data Mining: Concepts and Techniques (3rd Edition, J.Han) để nắm các độ đo tương quan khác và việc đánh giá các độ đo này.

Road-map lĩnh vực khai thác mẫu



- Đọc thêm chương 7 cuốn Data Mining: Concepts and Techniques (3rd Edition, J.Han) để nắm thêm các hướng nghiên cứu liên quan đến khai thác mẫu.

Bài tập tổng hợp 1

- Cho CSDL sau và minsup = 50%, minconf = 80%

TID	Date	Items_bought
100	15/1/03	K, A, D, B, C, I
200	15/1/03	D, A, C, E, B
300	19/1/03	C, A, B, E, D
400	25/1/03	B, A, D, I

a. Tìm tất cả các tập phổ biến, tập phổ biến tối đại, tập phổ biến đóng sử dụng thuật toán Apriori và FP-Growth
b. Xây dựng LKH thỏa minsup và minconf
c. Tính độ đo tương quan lift của các luật tìm được ở câu b

Bài tập tổng hợp 2

Transaction	Items
t_1	Blouse
t_2	Shoes,Skirt,TShirt
t_3	Jeans,TShirt
t_4	Jeans,Shoes,TShirt
t_5	Jeans,Shorts
t_6	Shoes,TShirt
t_7	Jeans,Skirt
t_8	Jeans,Shoes,Shorts,TShirt
t_9	Jeans
t_{10}	Jeans,Shoes,TShirt
t_{11}	TShirt
t_{12}	Blouse,Jeans,Shoes,Skirt,TShirt
t_{13}	Jeans,Shoes,Shorts,TShirt
t_{14}	Shoes,Skirt,TShirt
t_{15}	Jeans,TShirt
t_{16}	Skirt,TShirt
t_{17}	Blouse,Jeans,Skirt
t_{18}	Jeans,Shoes,Shorts,TShirt
t_{19}	Jeans
t_{20}	Jeans,Shoes,Shorts,TShirt

- Cho CSDL bên và minsup=30%, minconf=50%.
- Yêu cầu tương tự bài tập tổng hợp 1

Tóm tắt

- Các hạn chế của Apriori
- Thuật toán FP-Growth giảm thiểu việc duyệt dữ liệu nhiều lần bằng cách hình thành cây FP, dựa trên cây FP để chia dữ liệu thành các vùng nhỏ để tìm mẫu phổ biến.
- Các luật sinh ra có thể thừa hay gây nhầm lẫn, việc loại bỏ nó thông qua đánh giá các độ đo tương quan.

Tài liệu tham khảo

1. J. Han, J. Pei, and Y. Yin. *Mining frequent patterns without candidate generation*. SIGMOD'00, 1-12, Dallas, TX, May 2000
2. Improvements on FP-growth, http://www.cs.sfu.ca/CourseCentral/741/jpei/slides/FP-growth-Improvements.pdf
3. J.Han, M.Kamber, Chương 6 – Mining Frequent Patterns, Associations, and Correlations và Chương 7 - Advanced Pattern Mining cuốn "*Data mining: Basic Concepts and Methods*", 3rd edition

Hỏi & Đáp