Phát hiện các luật kết hợp trong cơ sở dữ liệu

Nguyễn Hồng Phương
Bộ môn Hệ thống thông tin
Viện CNTT&TT – trường ĐHBK Hà Nội
phuongnh@soict.hut.edu.vn
http://is.hut.edu.vn/~phuongnh

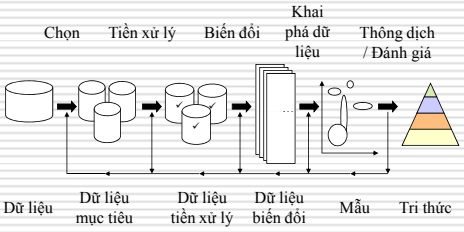
Nội dung trình bày

- 1. Tổng quan
- 2. Phát hiện luật kết hợp trong cơ sở dữ liệu giao dịch
- 3. Phát hiện luật kết hợp trong cơ sở dữ liệu quan hệ
- 4. Một số vấn đề khác

1. Tổng quan

- Khai phá dữ liệu và phát hiện tri thức
- Luật kết hợp: Bài toán "Cái giỏ hàng"
- Một số ứng dụng khác
- Các khái niệm cơ bản

Khai phá dữ liệu và phát hiện tri thức

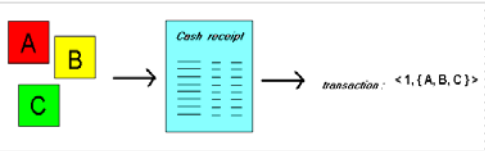


Luật kết hợp: Bài toán "Cái giỏ hàng"

- Phân tích thói quen mua hàng của khách hàng: tìm sự kết hợp và tương quan giữa các mặt hàng khác nhau mà khách hàng đặt vào trong "giỏ hàng" của họ

Phân tích bài toán "Cái giỏ hàng"

- Cho cơ sở dữ liệu gồm các giao dịch của khách hàng, mỗi giao dịch là một tập các mặt hàng
- Tìm các nhóm mặt hàng thường được mua cùng nhau



Một số ứng dụng khác

- Viễn thông
  - Mỗi khách hàng là một giao dịch gồm một tập các cuộc gọi của khách hàng đó
- Hiện tượng khí quyển
  - Mỗi khoảng thời gian quan sát là một giao dịch chứa một tập các sự kiện quan sát được (mưa, gió, mây,...)

Các khái niệm cơ bản

Giao dịch:

Dạng quan hệ <Tid,item>	Dạng thu gọn <Tid,itemset>
<1, item1>	<1, {item1,item2}>
<1, item2>	<2, {item3}>
<2, item3>	

Mục (Item): phần tử đơn,
Tập mục (Itemset): Tập các mục
Độ hỗ trợ của 1 tập mục X - sup(X): Số giao dịch chứa X
Độ hỗ trợ tối thiểu minsup : ngưỡng của độ hỗ trợ
Tập mục thường xuyên : độ hỗ trợ ≥ minsup.

Tập mục thường xuyên

ID giao dịch	Các mặt hàng đã mua
1	Sữa, trứng, đường, bánh mỳ
2	Sữa, trứng, ngũ cốc, bánh mỳ
3	Trứng, đường

- Sup({Sữa, trứng, bánh mỳ})= 2 (66.6%)
- Sup({Trứng, đường})= 2 (66.6%)
- Sup({Ngũ cốc, bánh mỳ})= 1 (33.3%)
- Nếu minsup = 50% thì {Sữa, trứng, bánh mỳ} và {Trứng, đường} là các tập mục thường xuyên còn {Ngũ cốc, bánh mỳ} thì không phải.

Luật kết hợp

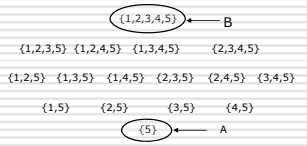
- A, B là tập các mục trong tập mục I
- Luật r = A ⇒ B
- Độ hỗ trợ của r: sup(r)=sup(A∪B)
- Độ tin cậy của r:
  - conf(r) = sup(A∪B)/sup(A)
- r được gọi là luật kết hợp nếu sup(r)≥minsup và conf(r)≥minconf

Độ hỗ trợ tối thiểu

Độ tin cậy tối thiểu

Hai tính chất cơ bản

- Tính chất 1:
  - Nếu một tập mục là không thường xuyên thì các siêu tập của nó cũng không thường xuyên
- Tính chất 2:
  - Nếu một tập mục là thường xuyên thì các tập con của nó cũng thường xuyên



2. Phát hiện luật kết hợp trong CSDL giao dịch

- Phát hiện các tập mục thường xuyên
  - Kiểu Apriori
  - Sử dụng FP-tree
- Phát hiện các luật kết hợp
- Khai phá luật kết hợp đa mức

Phát hiện các tập mục thường xuyên

- Giải thuật Apriori
- Sử dụng FP-tree

Giải thuật Apriori

Đầu vào: Cơ sở dữ liệu các giao dịch D và s_{min}
Đầu ra: Tập Answer chứa tất cả các tập mục thường xuyên của D
Giải thuật:

```
1)  L1 = {large 1-itemsets};
2)  for(k=2; Lk-1≠∅; k++) do begin
3)      Ck = AprioriGen(Lk-1); // New candidate
4)      forall transactions t∈D do begin
5)          Ct = Subset(Ck, t); // Candidates contained in t
6)          forall candidates c∈Ct do
7)              c.count++
8)      end
9)      Lk = {c∈Ck | c.count ≥ smin}
10) end
11) Answer = ∪k Lk;
```


Hàm AprioriGen

Đầu vào: Một tập L_{k-1} chứa tất cả các (k-1)-tập mục thường xuyên
Đầu ra: Tập C_k ứng cử là một siêu tập chứa tất cả các k-tập mục thường xuyên
Giải thuật:

```
1)  Function AprioriGen(Lk-1: tập (k-1)-tập mục thường xuyên):tập k-tập mục thường xuyên
2)      // Pha kết nối
3)      insert into Ck
4)      select p.item1, p.item2,...,p.itemk-1, q.itemk-1
5)      from Lk-1 p, Lk-1 q
6)      where p.item1 = q.item1,..., p.itemk-2 = q.itemk-2, p.itemk-1 < q.itemk-1
7)      // Pha cắt tỉa
8)      forall itemsets c∈Ck do
9)          forall (k-1)-subsets s of c do
10)             if(s∉Lk-1) then delete c from Ck;
11)  return Ck;
12) end;
```


Vấn đề của giải thuật kiểu Apriori

- Chi phí cho việc kiểm soát một số lượng lớn các tập mục ứng cử
  - 10^4 1-tập mục thường xuyên sẽ sinh 10^7 tập ứng cử kích thước 2
  - Lặp nhiều lần việc duyệt CSDL để kiểm tra các tập ứng cử

Tránh việc sinh quá nhiều tập ứng cử

Sử dụng cấu trúc cây mẫu thường xuyên

Xây dựng cây mẫu thường xuyên

- FP-tree (Frequent Pattern tree)
- Các bước xây dựng:
  - Duyệt DB lần 1 để sinh ra danh sách L

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

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

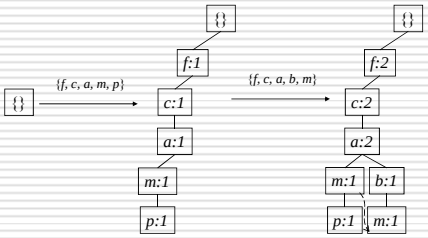
Xây dựng cây mẫu thường xuyên

- Duyệt DB lần 2, sắp xếp lại các giao dịch theo danh sách L

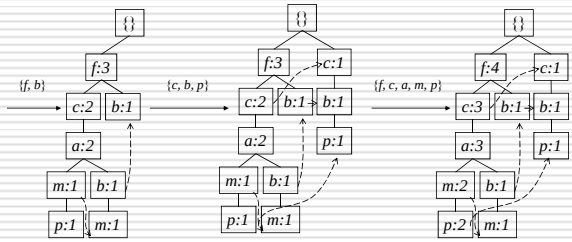
TID	Items	Các mục đã sắp xếp
100	f, a, c, d, g, i, m, p	f, c, a, m, p
200	a, b, c, f, l, m, o	f, c, a, b, m
300	b, f, h, j, o	f, b
400	b, c, k, s, p	c, b, p
500	a, f, c, e, l, p, m, n	f, c, a, m, p

Xây dựng cây mẫu thường xuyên

- Tiến hành xây dựng cây

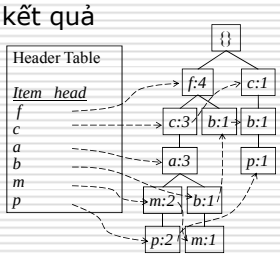


Xây dựng cây mẫu thường xuyên



Xây dựng cây mẫu thường xuyên

- Cây kết quả



- Khai phá mẫu thường xuyên?

Phát hiện các luật kết hợp

- Giải thuật đơn giản để sinh các luật

Đầu vào: Tập tất cả các tập mục thường xuyên có nhiều hơn một mục

$\overline{F} = \bigcup_{k\geq 2} F_k = F \setminus F_1$

Đầu ra: Tất cả các luật kết hợp

Phương pháp:

1) forall $f_k \in \overline{F}$ do
2) GenRules(f_k, f_k);

Phát hiện các luật kết hợp

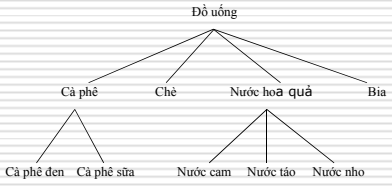
- Thủ tục GenRules

Đầu vào: Hai tập mục thường xuyên f_k và l_m, và một ngưỡng độ tin cậy c_{min}
Đầu ra: Các luật kết hợp với nhiều nhất m-1 mục ở phần đầu luật (m>2)
Phương pháp:

```
1)  procedure GenRules(fk: k-tập mục thường xuyên, lm: m-tập mục thường xuyên)
2)      L ← {các (m-1)-tập mục lm-1 | lm-1 ⊂ lm}
3)      forall lm-1 ∈ L do begin
4)          c ← s(fk)/s(lm-1); // Độ chắc chắn của luật
5)          if c ≥ cmin then begin
6)              output luật lm-1 ⇒ (fk \ lm-1);
7)              if m-1 ≥ 1 then
8)                  GenRules(fk, lm-1);
9)          end;
10)     end;
11) end;
```


Vấn đề khai phá luật kết hợp đa mức

- Phân cấp khái niệm trên các mục của CSDL



Thuật toán cơ bản khai phá luật kết hợp đa mức

L_1:={các 1-tập mục thường xuyên};
k:=2;
while ($L_{k-1} \neq \emptyset$) **do**
begin
  C_k:=các ứng cử viên mới kích thước k được sinh ra từ L_{k-1}
  forall giao dịch t∈D **do**
  begin
    Thêm tất cả các tổ tiên của từng mục trong t vào t, loại bỏ sự trùng lặp
    Tăng bộ đếm của tất cả các ứng viên trong C_k mà có mặt trong t
  end
  L_k:=Tất cả ứng viên trong C_k đạt độ hỗ trợ tối thiểu
  k:=k+1;
end
Câu trả lời := $\bigcup_k L_k$

3. Phát hiện luật kết hợp trong CSDL quan hệ

- CSDL quan hệ: các quan hệ thường chứa các thuộc tính định lượng, phạm trù.
  - Xử lý những thuộc tính định lượng:
    - Phân vùng rõ

⇒ **Khai phá luật kết hợp định lượng**

    - Phân vùng mờ

⇒ **Khai phá luật kết hợp mờ**

Khai phá luật kết hợp định lượng

- Phân vùng Equi-Depth: Các vùng có kích thước như nhau
  - Phân vùng dựa trên các giá trị có thể có của thuộc tính. Ví dụ: nếu kiểu thuộc tính có giá trị từ 1 đến 15 và depth d=3 thì sinh ra các khoảng [1,3], [4,6], [7,9], [10,12], [13,15]
  - Phân vùng dựa trên các giá trị có thực trong CSDL: d giá trị đầu được đặt vào khoảng thứ nhất, d giá trị tiếp theo được đặt vào khoảng thứ hai,…

Khai phá luật kết hợp định lượng

- Phân vùng dựa trên khoảng cách: có xem xét tính chất định lượng và ngữ nghĩa của dữ liệu. Khoảng cách giữa các điểm dữ liệu càng nhỏ thì chúng càng nên thuộc về 1 nhóm

Lương	equi-depth	distance-based
18	[18, 30]	[18, 18]
30		[30, 31]
31	[31, 80]	
80		[80, 82]
81	[81, 82]	
82		

Khai phá luật kết hợp định lượng

- Các bước:

recordId	age	married	numCars
100	23	no	1
200	25	yes	1
300	29	no	0
400	34	yes	2
500	38	yes	2

Interval
[20, 24]
[25, 29]
[30, 34]
[35, 39]

age	Integer
[20, 24]	1
[25, 29]	2
[30, 34]	3
[35, 39]	4

married	Integer
yes	1
no	2

recordId	age	married	numCars
100	1	2	1
200	2	1	1
300	2	2	0
400	3	1	2
500	4	1	2

Khai phá luật kết hợp định lượng

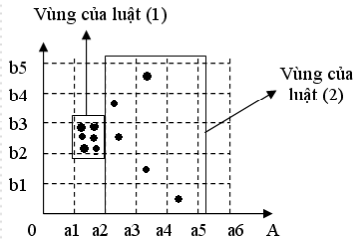
recordId	<age, [20,29]>	<age, [30,39]>	<married, yes>	<married, no>	<numCars, 0>	<numCars, 1>	<numCars, 2>
100	1	0	0	1	0	1	0
200	1	0	1	0	0	1	0
300	1	0	0	1	1	0	0
400	0	1	1	0	0	0	1
500	0	1	1	0	0	0	1

Itemset	Support
{<age, [20, 29]>}	3
{<age, [30, 39]>}	2
{<married, yes>}	3
{<married, no>}	2
{<numCars, [0, 1]>}	3
{<age, [30, 39]>,<married, yes>}	2

Rule	Support	Confidence
{<age, [30, 39]>, <married,yes>} → {<numCars, [2, 2]>}	0.40	1.00
{<age, [20,29]>} → {<numCas, [0, 1]>}	0.60	0.67

Khai phá luật kết hợp định lượng

- Cách tiếp cận khối dày đặc



Khai phá luật kết hợp mờ

- Khái niệm luật kết hợp mờ
  - Nếu X = {x1, x2, ..., xp} là A = {a1, a2, ..., ap} thì Y = {y1, y2, ..., yq} là B = {b1, b2, ..., bq}
  - X, Y là tập các thuộc tính
  - x1, x2,...,y1,y2,... là các thuộc tính
  - A, B là tập các tập mờ
  - a1,a2,...,b1,b2,...là các tập mờ
- Công thức tính độ hỗ trợ mờ

$$FS_{<X,A>} = \frac{\sum_{t_i \in D} \prod_{x_j \in X} d_{x_j}(a_j, t_i.x_j)}{|D|}$$

Khai phá luật kết hợp mờ

- Công thức tính độ tin cậy mờ

$$FC_{<<X,A>,<Y,B>>} = \frac{FS_{<Z,C>}}{FS_{<X,A>}} = \frac{\sum_{t_i \in D} \prod_{z_j \in Z} d_{z_j}(c_j, t_i.z_j)}{\sum_{t_i \in D} \prod_{x_j \in X} d_{x_j}(a_j, t_i.x_j)}$$

- Ví dụ: Có X = {Balance, Income}, A = {medium, high}, Y = {Credit}, B = {high}

Balance, medium	Credit, high	Income, high
0.5	0.6	0.4
0.8	0.9	0.4
0.7	0.8	0.7
0.9	0.8	0.3
0.9	0.7	0.6

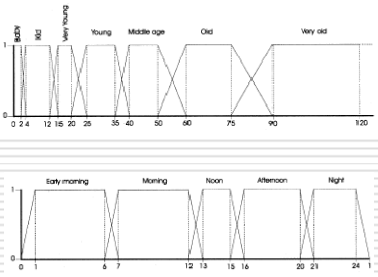
$FS_{<X,A>}=0.364$

$FC_{<<X,A>,<Y,B>>}=0.766$

Khai phá luật kết hợp mờ

- Các bước

	Age	Hour
t_1	60	20:15
t_2	80	23:45
t_3	22	15:30
t_4	55	01:00
t_5	3	19:30
t_6	18	06:51

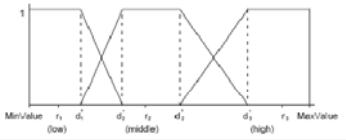


Khai phá luật kết hợp mờ

	<Age, Baby>	<Age, Kid>	<Age, Very young>	<Age, Young>	<Age, Middle Age>	<Age, Old>	<Age, Very old>	<Hour, Early Morning>	<Hour, Morning>	<Hour, Noon>	<Hour, After Noon>	<Hour, Night>
t_1	0	0	0	0	0	1	0	0	0	0	0.75	0.25
t_2	0	0	0	0	0	0.67	0.33	0	0	0	0	1
t_3	0	0	0.6	0.4	0	0	0	0	0	0.5	0.5	0
t_4	0	0	0	0	0.5	0.5	0	1	0	0	0	0
t_5	0.5	0.5	0	0	0	0	0	0	0	0	1	0
t_6	0	0	1	0	0	0	0	0.85	0.15	0	0	0

Khai phá luật kết hợp mờ

- Phân vùng mờ miền thuộc tính?
- Attila Gyenesei giới thiệu kỹ thuật phân vùng mờ dựa trên chỉ số độ tốt (Goodness Index)
- Tìm tâm, các cận các nhóm



- Tính hàm độ thuộc

4. Một số vấn đề khác

- Phát hiện luật có yếu tố thời gian
- Phát hiện luật trên nhiều quan hệ
- Phân loại luật kết hợp

Lời hay ý đẹp

Thành công, đó là cách khuyến khích ta cố gắng làm những việc lớn lao hơn nữa. Thất bại, đó là cách cổ vũ ta làm lại việc đã làm với nhiều hi vọng hơn.

Gabriel Palau