



KHAI THÁC TẬP KHÔNG HỮU ÍCH

(MINING ERASABLE ITEMSETS)

Nội dung

- Giới thiệu
- Bài toán khai thác EI
- Thuật toán META
- Tổng kết

Giới thiệu



Sản xuất các loại sản phẩm...

$P_1 (i_2, i_3, i_4, i_6)$

Lợi nhuận: 20 triệu

$P_2 (i_2, i_5, i_7)$

Lợi nhuận: 50 triệu

$P_3 (i_1, i_2, i_3, i_5)$

Lợi nhuận: 30 triệu

*Tổng lợi nhuận khi
bán toàn bộ sản phẩm
100 triệu*

*Nguyên liệu
 $i_1, i_2, i_3, i_4, i_5, i_6, i_7$*

Giới thiệu



Công ty không có đủ tiền mua nguyên liệu...

Công ty phải ngừng sản xuất một số loại sản phẩm và không mua những nguyên vật liệu tương ứng...



Ngừng sản xuất những loại sản phẩm nào?

*Những loại sản phẩm mà không làm giảm tổng lợi nhuận quá một **ngưỡng** nào đó...*

Giới thiệu

Ví dụ:

Với ngưỡng giảm lợi nhuận chấp nhận được là 25%, công ty có thể bỏ loại sản phẩm P_1 và không mua các nguyên liệu i_4 và i_6 .

Sản phẩm	Lợi nhuận
$P_1 (i_2, i_3, i_4, i_6)$	20
$P_2 (i_2, i_5, i_7)$	50
$P_3 (i_1, i_2, i_3, i_5)$	30

$\{i_4, i_6\}$ gọi là một tập thành phần không hữu ích

Vấn đề trở thành tìm
những *itemsets* như vậy...

Nội dung

- Giới thiệu
- Bài toán khai thác EI
- Thuật toán META
- Tổng kết

Bài toán khai thác EI

- Dữ liệu
 - Ngưỡng ξ
 - Tập thành phần $I = \{i_1, i_2, \dots, i_m\}$
 - Cơ sở dữ liệu $DB = \{P_1, P_2, \dots, P_n\}$

Cơ sở dữ liệu thí dụ DB_e gồm 6 loại sản phẩm và 7 thành phần, tổng lợi nhuận là 10000.

<i>Product</i>	<i>PID</i>	<i>Items</i>	<i>Val</i>
P_1	1	$\{i_2, i_3, i_4, i_6\}$	500
P_2	2	$\{i_2, i_5, i_7\}$	200
P_3	3	$\{i_1, i_2, i_3, i_5\}$	500
P_4	4	$\{i_1, i_2, i_4\}$	8000
P_5	5	$\{i_6, i_7\}$	300
P_6	6	$\{i_3, i_4\}$	500

Bài toán khai thác EI

- Định nghĩa 1 (*Gain*)

Cho itemset $A (\subseteq I)$, *Gain* của A được tính như sau:

$$Gain(A) = \sum_{\{P_k | A \cap P_k.Items \neq \emptyset\}} P_k.Val$$

Ví dụ:

$A = \{i_6, i_7\}$, các loại sản phẩm có chứa i_6 hoặc i_7 hay cả hai là P_1, P_2, P_5 , do đó:

$$\begin{aligned} Gain(A) &= P_1.Val + P_2.Val + P_5.Val \\ &= 500 + 200 + 300 = 1000 \end{aligned}$$

P_i	PID	Items	Val
P_1	1	$\{i_2, i_3, i_4, i_6\}$	500
P_2	2	$\{i_2, i_5, i_7\}$	200
P_3	3	$\{i_1, i_2, i_3, i_5\}$	500
P_4	4	$\{i_1, i_2, i_4\}$	8000
P_5	5	$\{i_6, i_7\}$	300
P_6	6	$\{i_3, i_4\}$	500

Bài toán khai thác EI

- Định nghĩa 2 (*EI*)

Cho trước một ngưỡng ξ và tập DB , một tập A gọi là *tập không hữu ích* nếu:

$$Gain(A) \leq (\sum_{P_k \in DB} P_k.Val) \times \xi$$

Ví dụ:

Tổng lợi nhuận 10000, $\xi = 15\%$

$$\begin{aligned} Gain(\{i_6, i_7\}) &= 50 + 20 + 30 \\ &= 100 \leq (10000 \times 15\%) \end{aligned}$$

Do đó $\{i_6, i_7\}$ là một tập không hữu ích.


P_i	PID	Items	Val
P_1	1	$\{i_2, i_3, i_4, i_6\}$	500
P_2	2	$\{i_2, i_5, i_7\}$	200
P_3	3	$\{i_1, i_2, i_3, i_5\}$	500
P_4	4	$\{i_1, i_2, i_4\}$	8000
P_5	5	$\{i_6, i_7\}$	300
P_6	6	$\{i_3, i_4\}$	500

Bài toán khai thác EI

- Phát biểu bài toán: Cho cơ sở dữ liệu sản phẩm DB và một ngưỡng ξ , hãy tìm tất cả các tập không hữu ích trong DB .

PID	$Items$	Val
1	$\{i_2, i_3, i_4, i_6\}$	500
2	$\{i_2, i_5, i_7\}$	200
3	$\{i_1, i_2, i_3, i_5\}$	500
4	$\{i_1, i_2, i_4\}$	8000
5	$\{i_6, i_7\}$	300
6	$\{i_3, i_4\}$	500

$\xi = 15\%$



$Itemset$	$Gain$
$\{i_3\}$	1500
$\{i_5\}$	700
$\{i_6\}$	800
$\{i_7\}$	500
$\{i_5, i_6\}$	1500
$\{i_5, i_7\}$	1000
$\{i_6, i_7\}$	1000
$\{i_5, i_6, i_7\}$	1500

EIs

Nội dung

- Giới thiệu
- Bài toán khai thác EI
- Thuật toán META
- Tổng kết

Thuật toán META

- **META** Mining Erasable iTemsets with the Antimonotone property algorithm

Thuật toán đầu tiên khai thác *EIs* được nhóm tác giả Zhi-Hong Deng giới thiệu vào năm 2009.

- Các tính chất

Tính chất 1: Cho hai tập $X \subseteq I$ và $Y \subseteq I$. Nếu X là tập con của Y ($X \subseteq Y$) thì $Gain(X) \leq Gain(Y)$.

Tính chất 2 (*anti-monotone*): Cho hai tập $X \subseteq I$ và $Y \subseteq I$. Nếu X không phải là một *EI* và $X \subseteq Y$ thì Y cũng không phải là một *EI*.

Tính chất 3: Nếu X là một *EI* và Y là tập con của X ($Y \subseteq X$) thì Y phải là một *EI*.

Thuật toán META

- Phương pháp: Tìm các *EI*s theo từng cấp độ (*level-wise search*)

Input: Cơ sở dữ liệu $DB = \{P_1, P_2, \dots, P_n\}$; ngưỡng ξ ;

Output: Tập toàn bộ các tập không hữu ích *EI*;

$Sum_val = 0$;

For ($k = 1$; $k \leq n$; $k++$)

$Sum_val = Sum_val + P_k.Val$;

$E_1 = \{EI_1\}$;

For ($k = 2$; $E_{k-1} \neq \emptyset$; $k++$)

$GC_k = \mathbf{Gen_Candidate}(E_{k-1})$;

For each product $P \in DB$ {

For each candidate itemset $C \in GC_k$

If ($C \cap P \neq \emptyset$) then

$C.gain = C.value + P.Val$;

$E_k = \{C \in GC_k \mid C.gain \leq \xi \times Sum_val\}$

Return $EI = \cup_k E_k$;

Thuật toán META

Procedure Gen_Candidate (E_{k-1})

// Các items trong E_{k-1} được sắp theo thứ tự xuất hiện trong I

$Candidates = \emptyset$;

For each $A_1 (= \{x_1, x_2, \dots, x_{k-2}, x_{k-1}\}) \in E_{k-1}$

For each $A_2 (= \{y_1, y_2, \dots, y_{k-2}, y_{k-1}\}) \in E_{k-1}$

If $((x_1=y_1) \wedge (x_2=y_2) \wedge \dots \wedge (x_{k-2}=y_{k-2}) \wedge (x_{k-1} < y_{k-1}))$ then

$X = \{x_1, x_2, \dots, x_{k-2}, x_{k-1}, y_{k-1}\}$;

If **No_Unerasable_Subset** (X, E_{k-1}) then

add X to $Candidates$;

Return $Candidates$;

Procedure No_Unerasable_Subset (X, E_{k-1})

For each $(k-1)$ -subset X_s of X

If $X_s \notin E_{k-1}$ then

Return FALSE; *//anti-monotone*

Return TRUE;

Thuật toán META

- Minh họa

$$I = \{i_1, i_2, i_3, i_4, i_5, i_6, i_7\}$$

$$DB = \{P_1, P_2, P_3, P_4, P_5, P_6\}$$

$$\xi = 18\%$$

<i>PID</i>	<i>Items</i>	<i>Val</i>
1	$\{i_2, i_3, i_4, i_6\}$	500
2	$\{i_2, i_5, i_7\}$	200
3	$\{i_1, i_2, i_3, i_5\}$	500
4	$\{i_1, i_2, i_4\}$	8000
5	$\{i_6, i_7\}$	300
6	$\{i_3, i_4\}$	500

Bước 1: Khởi tạo

- Tính $Sum_val = 10000$
- Tìm E_1

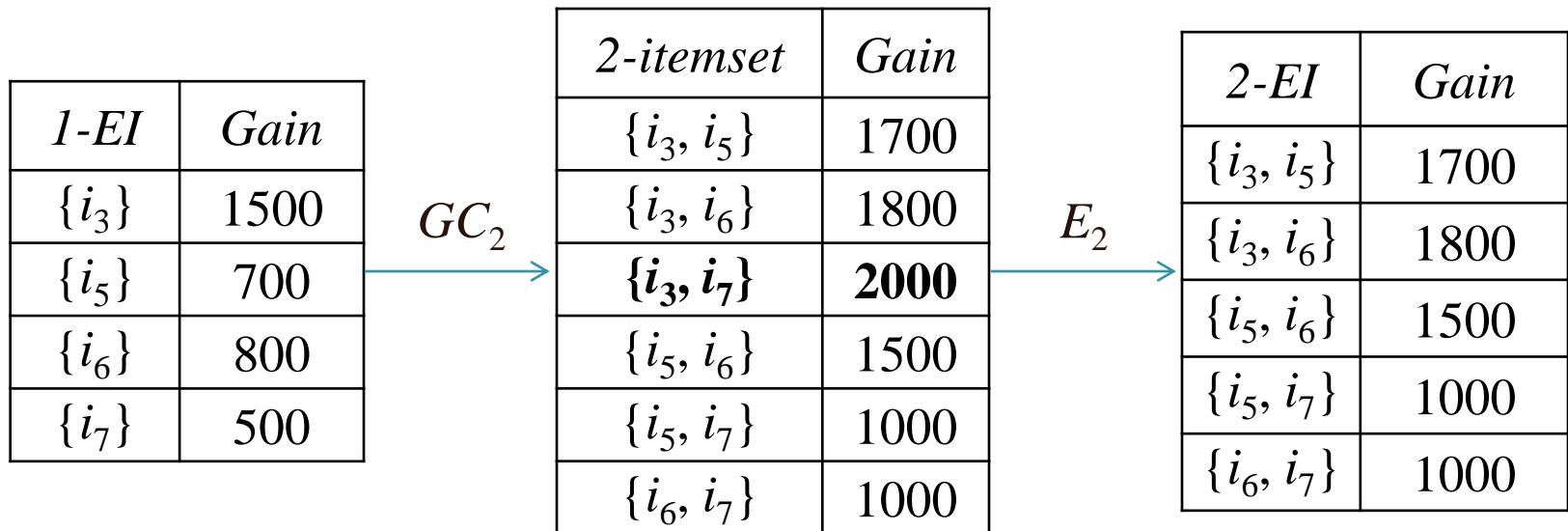
E_1	<i>1-itemset</i>	<i>Gain</i>
	$\{i_3\}$	1500
	$\{i_5\}$	700
	$\{i_6\}$	800
	$\{i_7\}$	500

Thuật toán META

Bước 2: Khai thác

$$\underline{k = 2}$$

- Xây dựng GC_2 dựa trên E_1
- Tìm tập E_2 dựa trên GC_2

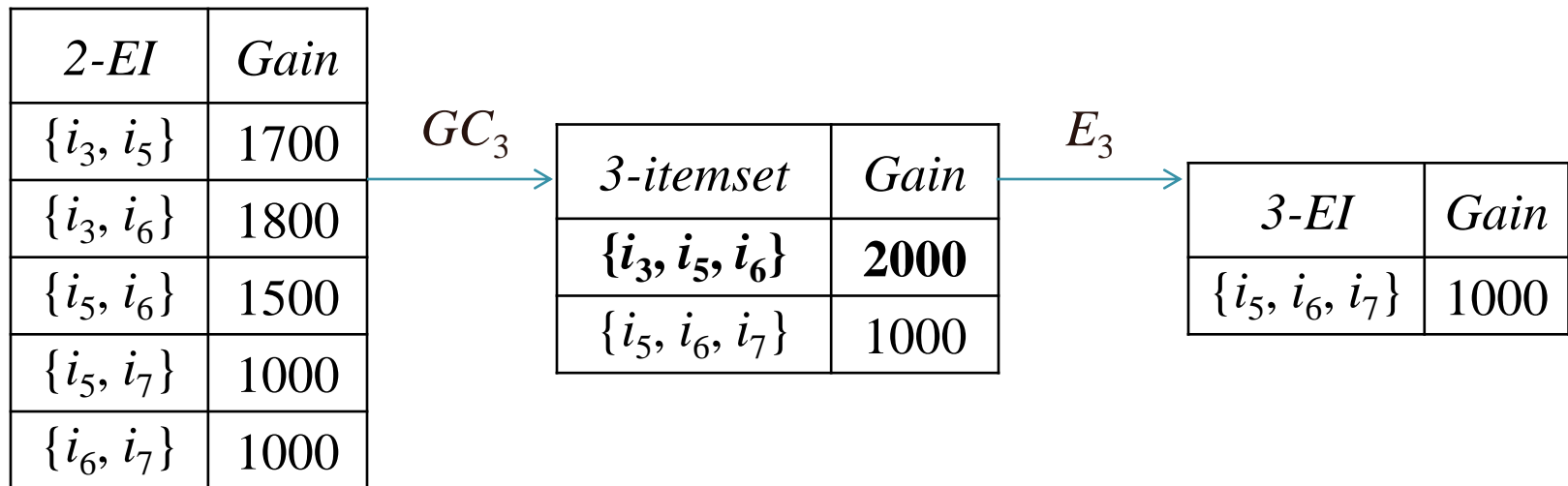


Thuật toán META

Bước 2: Khai thác

$$\underline{k = 3}$$

- Xây dựng GC_3 dựa trên E_2
- Tìm tập E_3 dựa trên GC_3



Thuật toán META

Bước 2: Khai thác

$$\underline{k = 4}$$

$E_4 = \emptyset \rightarrow$ Thuật toán dừng

Bước 3: Trả về kết quả

$$EI = E_1 \cup E_2 \cup E_3$$

Thuật toán META

- Nhận xét

- Đầu tiên, thuật toán duyệt *DB* tính tổng lợi nhuận. Trong k bước lặp tiếp theo, thuật toán tiếp tục duyệt *DB* để tính lợi nhuận của các itemset. Do đó chi phí thời gian rất lớn.

- Thuật toán không loại bỏ được dữ liệu dư thừa. Ví dụ xét itemset $\{i_3\}$, các loại sản phẩm chứa i_3 là P_1, P_3 và P_6 , nhưng khi tính *Gain* của $\{i_3\}$ phải duyệt toàn bộ *DB*.

<i>PID</i>	<i>Items</i>	<i>Val</i>
1	$\{i_2, i_3, i_4, i_6\}$	500
2	$\{i_2, i_5, i_7\}$	200
3	$\{i_1, i_2, i_3, i_5\}$	500
4	$\{i_1, i_2, i_4\}$	8000
5	$\{i_6, i_7\}$	300
6	$\{i_3, i_4\}$	500

Nội dung

- Giới thiệu
- Bài toán khai thác EI
- Thuật toán META
- Tổng kết

Tổng kết

- Khai thác các tập thành phần không hữu ích là một trong những tác vụ mới trong khai thác dữ liệu.
- Về mặt kỹ thuật, khai thác *EI* cũng tương tự khai thác mẫu phổ biến *FP*. Cả hai cùng khai thác các itemset quan tâm.
- Tuy nhiên, khai thác *EI* và khai thác *FP* có sự khác biệt.
 - Khai thác *FP* ra đời trong bối cảnh một siêu thị bán lẻ muốn tìm mối quan hệ giữa các mặt hàng được khách hàng mua.
 - Khai thác *EI* xuất hiện trong bối cảnh công ty sản xuất sản phẩm cần lập kế hoạch sản xuất phù hợp khi nền kinh tế rơi vào suy thoái.



Thuật toán ILA

Thuật toán nâng cao (Đại học Thủy lợi)



Scan to open on Studocu

Thuật toán ILA(Inductive Learning Algorithm) được dùng để xác định các luật phân loại cho tập hợp các mẫu học. Thuật giải này thực hiện theo cơ chế lặp, để tìm luật riêng đại diện cho tập mẫu của từng lớp. Sau khi xác định được luật, ILA loại bỏ các mẫu liên quan khỏi tập mẫu, đồng thời thêm luật mới vào tập luật. Kết quả có được là một danh sách có thứ tự các luật chứ không là một cây quyết. Các ưu điểm của thuật giải này có thể được trình bày như sau:

-Dạng các luật sẽ phù hợp cho việc khảo sát dữ liệu, mô tả mỗi lớp một cách đơn giản để dễ phân biệt với các lớp khác.

- Tập luật được sắp thứ tự, riêng biệt cho phép quan tâm đến một luật tại thời điểm bất kỳ. Khác với việc xử lý luật theo phương pháp cây quyết định, vốn rất phức tạp trong trường hợp các nút cây trở nên khá lớn.

Bước 1: Chia mẫu ban đầu thành n bảng con. Mỗi bảng con ứng với một giá trị của thuộc tính quyết định của tập mẫu

Thực hiện lần lượt các bước từ 2 đến 8 cho mỗi bảng con có được

Bước 2: $j=1$ (j là số thuộc tính của tổ hợp T)

Bước 3: Trên mỗi bảng con đang khảo sát, chia danh sách các thuộc tính thành các tổ hợp khác nhau, mỗi tổ hợp bao gồm j thuộc tính

Bước 4: Với mỗi tổ hợp thuộc tính có được tính số lần giá trị thuộc tính xuất hiện theo cùng tổ hợp thuộc tính trong các dòng còn lại của bảng con đang xét (mà đồng thời không xuất hiện tổ hợp giá trị này trên tất cả các bảng còn lại).

Tổ hợp T ^{Thuộc tính}
Giá trị của thuộc tính

Gọi tổ hợp đầu tiên(trong bảng con) có số lần xuất hiện nhiều nhất là tổ hợp lớn nhất.

Bước 5: Nếu tổ hợp lớn nhất có giá trị bằng 0, tăng j lên 1 và quay lại bước 3.

Bước 6:Loại bỏ các dòng thỏa mãn tổ hợp lớn nhất ra khỏi bảng con đang xử lý

Bước 7: Thêm luật mới vào tập luật R , với vế trái là tập các thuộc tính của tổ hợp lớn nhất(Kết hợp các thuộc tính bằng toán tử AND) và vế phải là giá trị thuộc tính quyết định tương ứng.

Bước 8: Nếu tất cả các dòng đều đã được loại bỏ, tiếp tục thực hiện bước 2 cho các bảng còn lại. Ngược lại(nếu còn dòng chưa bị loại bỏ) thì quay lại bước 4. Nếu tất cả các dòng con đã được xét thì kết thúc. Tập R chính là tập luật cần tìm.

2. Minh họa thuật toán:

Minh họa giải thuật ILA cho bảng dữ liệu sau đây:

Tên	Nước tiểu	Giảm cân	Đau lưng	Sốt	Kết quả
An	Vàng	Không	Không	Có	Không bệnh sỏi thận
Cường	Vàng	Không	Không	Không	Không bệnh sỏi thận
Châu	Có vôi	Không	Không	Có	Bệnh sỏi thận
Dung	Có máu	ít	Không	Có	Bệnh sỏi thận
Diễm	Có máu	Nhanh	Có	Có	Bệnh sỏi thận
Hương	Có máu	Nhanh	Có	Không	Không bệnh sỏi thận
Hoa	Có vôi	Nhanh	Có	Không	Bệnh sỏi thận
Phương	Vàng	ít	Không	Có	Không bệnh sỏi thận
My	Vàng	Nhanh	Có	Có	Bệnh sỏi thận
Nhung	Có máu	ít	Có	Có	Bệnh sỏi thận
Thu	Vàng	ít	Có	Không	Bệnh sỏi thận
Thương	Có vôi	ít	Không	Không	Bệnh sỏi thận
Tuấn	Có vôi	Không	Có	Có	Bệnh sỏi thận
Tùng	Có máu	ít	Không	Không	Không bệnh sỏi thận

Chia bảng mẫu thành 2 bảng con bởi 2 loại quyết định: “Bệnh sỏi thận” và “Không bệnh sỏi thận” như sau:

Bảng 1: “Bệnh sỏi thận”

Tên	Nước Tiểu	Giảm Cân	Đau Lưng	Sốt	Kết Quả
Châu	Có vôi	Không	Không	Có	Bệnh Sỏi Thận
Dung	Có máu	Ít	Không	Có	Bệnh Sỏi Thận
Diễm	Có máu	Nhanh	Có	Có	Bệnh Sỏi Thận
Hoa	Có vôi	Nhanh	Có	Không	Bệnh Sỏi Thận
My	Vàng	Nhanh	Có	Có	Bệnh Sỏi Thận
Nhung	Có máu	Ít	Có	Có	Bệnh Sỏi Thận
Thu	Vàng	Ít	Có	Không	Bệnh Sỏi Thận
Thương	Có vôi	Ít	Không	Không	Bệnh Sỏi Thận
Tuấn	Có vôi	Không	Có	Có	Bệnh Sỏi Thận

Bảng 2: “Không bệnh sỏi thận”

Tên	Nước Tiểu	Giảm Cân	Đau Lưng	Sốt	Kết Quả
An	Vàng	Không	Không	Có	Không Bệnh Sởi Thận
Cường	Vàng	Không	Không	Không	Không Bệnh Sởi Thận
Hương	Có máu	Nhanh	Có	Không	Không Bệnh Sởi Thận
Phương	Vàng	Ít	Không	Có	Không Bệnh Sởi Thận
Tùng	Có máu	Ít	Không	Không	Không Bệnh Sởi Thận

Bảng 1: “Bệnh sởi thận”

Tổ hợp: T

Với $j=1$, có 4 tổ hợp:

- {Nước tiểu}
 - {Giảm cân}
 - {Đau lưng}
 - {Sốt}
- Với tổ hợp {Nước tiểu}: Thuộc tính “Có vôi” xuất hiện 4 lần trong bảng 1 và không xuất hiện trong bảng 2. Thuộc tính “Có máu” và “Vàng” xuất hiện trên cả hai bảng.
- $$T_{\text{Nước tiểu}}^{\text{Có vôi}} = 4$$
- $$T_{\text{Nước tiểu}}^{\text{Có máu}} = 0$$
- $$T_{\text{Nước tiểu}}^{\text{Vàng}} = 0$$
- Với tổ hợp {Giảm cân}: Thuộc tính “Không”, “Nhanh”, “Ít” xuất hiện trên cả hai bảng.
- $$T_{\text{Giảm cân}}^{\text{Không}} = 0$$
- $$T_{\text{Giảm cân}}^{\text{Nhanh}} = 0$$
- $$T_{\text{Giảm cân}}^{\text{Ít}} = 0$$
- Với tổ hợp {Đau lưng}: Thuộc tính “Không”, “Có” xuất hiện trên cả hai bảng.
- $$T_{\text{Đau lưng}}^{\text{Không}} = 0$$
- $$T_{\text{Đau lưng}}^{\text{Có}} = 0$$
- Với tổ hợp {Sốt}: Thuộc tính “Không”, “Có” xuất hiện trên cả hai bảng.

$$T_{\text{Không}}^{\text{Sốt}} = 0$$

$$T_{\text{Có}}^{\text{Sốt}} = 0$$

⇒ Ta có $T_{\text{Có vôi}}^{\text{Nước tiểu}} = 4$ là lớn nhất. Ta chọn $T_{\text{Có vôi}}^{\text{Nước tiểu}}$

RULE 1: IF Nước tiểu = có vôi THEN Kết quả = Bệnh sỏi thận

Tiếp theo ta loại bỏ những dòng thỏa mãn tổ hợp lớn nhất tương ứng với Nước tiểu = Có vôi ra khỏi bảng 1 ta có bảng sau:

Tên	Nước tiểu	Giảm cân	Đau lưng	Sốt	Kết quả
Dung	Có máu	Ít	Không	Có	Bệnh sỏi thận
Diễm	Có máu	Nhanh	Có	Có	Bệnh sỏi thận
My	Vàng	Nhanh	Có	Có	Bệnh sỏi thận
Nhung	Có máu	Ít	Có	Có	Bệnh sỏi thận
Thu	Vàng	Ít	Có	Không	Bệnh sỏi thận

Các dòng trong bảng trên mọi giá trị của thuộc tính đều xuất hiện trong cả hai bảng(mọi giá trị T đều bằng 0) nên ta sẽ tăng j lên 1.

Với j=2, có 6 tổ hợp:

- {Nước tiểu, Giảm cân }
- {Nước tiểu, Đau lưng}
- {Nước tiểu, Sốt}
- {Giảm cân, Đau lưng}
- {Giảm cân, Sốt}
- {Đau lưng, Sốt}

$$T_{\text{Có máu}, \text{Ít}}^{\text{Nước tiểu, Giảm cân}} = 0$$

$$T_{\text{Có máu}, \text{Nhanh}}^{\text{Nước tiểu, Giảm cân}} = 0$$

$$T_{\text{Vàng}, \text{Ít}}^{\text{Nước tiểu, Giảm cân}} = 0$$

$$T_{\text{Vàng}, \text{Nhanh}}^{\text{Nước tiểu, Giảm cân}} =$$

$$T_{\text{Có máu}, \text{Không}}^{\text{Nước tiểu, Đau lưng}} = 0$$

$$T_{\text{Có máu}, \text{Có}}^{\text{Nước tiểu, Đau lưng}} = 0$$

T Nước tiểu Vàng , Đau lưng Có = 1

T Nước tiểu Vàng , Sốt Không = 0

T Nước tiểu Có máu , Sốt Có = 3

T Giảm cân Ít , Đau lưng Không = 0

T Giảm cân Nhanh , Đau lưng Có = 0

T Giảm cân Ít , Đau lưng Có = 2

T Giảm cân Ít , Sốt Có = 0

T Giảm cân Ít , Sốt Không = 0

T Giảm cân Nhanh , Sốt Có = 1

T Đau lưng Không , Sốt Có = 0

T Đau lưng Có , Sốt Có = 2

T Đau lưng Có , Sốt Không = 0

⇒ Ta có T Nước tiểu Có máu , Sốt Có = 3 là lớn nhất, ta chọn T Nước tiểu Có máu , Sốt Có

RULE 2: IF Nước tiểu= Có máu AND Sốt = Có THEN Kết Quả = Bệnh sỏi thận

Kể tiếp, loại bỏ những dòng ứng với Nước tiểu = có máu và sốt = có ra khỏi bảng ta được:

Tên	Nước tiểu	Giảm cân	Đau lưng	Sốt	Kết quả
My	Vàng	Nhanh	Có	Có	Bệnh sỏi thận
Thu	Vàng	Ít	Có	Không	Bệnh sỏi thận

Với j=2, ta có 6 tổ hợp mỗi tổ hợp gồm 2 thuộc tính:

- {Nước tiểu, Giảm cân }
- {Nước tiểu, Đau lưng}
- {Nước tiểu, Sốt}
- {Giảm cân, Đau lưng}
- {Giảm cân, Sốt}
- {Đau lưng, Sốt}
-

T Nước tiểu vàng, giảm cân ít=0
T Nước tiểu vàng, giảm cân Nhanh=1

T Nước tiểu vàng, đau lưng có=2

T Nước tiểu vàng, sốt không=0
T Nước tiểu vàng, sốt Có=0

T Giảm cân ít, đau lưng có=1

T Giảm cân ít, sốt Không=0
T Giảm cân Nhanh, sốt Có=1

T Đau lưng có, sốt không=0
T Đau lưng có, sốt có=1

⇒ Ta có T Nước tiểu vàng, đau lưng có=2 là lớn nhất. Ta chọn T Nước tiểu vàng, đau lưng có và ta có luật:

RULE 3: IF Nước tiểu = Vàng AND Đau lưng=Có THEN Kết quả=Bệnh sỏi thận

Loại bỏ các dòng tương ứng với Nước tiểu = vàng, đau lưng=có, như vậy tất cả các dòng trong bảng 1 bị loại bỏ

Bảng 2: “Không bệnh sỏi thận”

Tên	Nước Tiểu	Giảm Cân	Đau Lưng	Sốt	Kết Quả
An	Vàng	Không	Không	Có	Không Bệnh Sỏi Thận
Cường	Vàng	Không	Không	Không	Không Bệnh Sỏi Thận
Hương	Có máu	Nhanh	Có	Không	Không Bệnh Sỏi Thận
Phương	Vàng	Ít	Không	Có	Không Bệnh Sỏi Thận
Tùng	Có máu	Ít	Không	Không	Không Bệnh Sỏi Thận

Trong bảng 2, mọi giá trị của thuộc tính đều xuất hiện trong cả hai bảng(mọi giá trị T đều bằng 0) nên ta sẽ tăng j lên 1.

Với j=2, có 6 tổ hợp mỗi tổ hợp có 2 thuộc tính:

- {Nước tiểu, Giảm cân }
- {Nước tiểu, Đau lưng}
- {Nước tiểu, Sốt}
- {Giảm cân, Đau lưng}
- {Giảm cân, Sốt}
- {Đau lưng, Sốt}

$$T(\text{nước tiểu có máu, giảm cân ít})=0$$

$$T(\text{nước tiểu có máu, giảm cân nhanh})=0$$

$$T(\text{nước tiểu vàng, giảm cân ít})=0$$

$$T(\text{nước tiểu vàng, giảm cân không})=2$$

$$T(\text{nước tiểu có máu, đau lưng không})=0$$

$$T(\text{nước tiểu có máu, đau lưng có})=0$$

$$T(\text{nước tiểu vàng, đau lưng không})=3$$

$$T(\text{nước tiểu vàng, sốt không})=0$$

$$T(\text{nước tiểu vàng, sốt có})=0$$

$$T(\text{nước tiểu có máu, sốt không})=2$$

$$T(\text{giảm cân ít, đau lưng không})=0$$

$$T(\text{Giảm cân nhanh, đau lưng có})=0$$

$$T(\text{giảm cân không, đau lưng không})=0$$

$$T(\text{giảm cân không, sốt có})=0$$

$$T(\text{giảm cân không, sốt không})=1$$

$$T(\text{giảm cân nhanh, sốt không})=0$$

$$T(\text{giảm cân ít, sốt có})=0$$

T giảm cân ít, sốt không = 0

T đau lưng không, sốt có = 0

T đau lưng không, sốt không = 0

T đau lưng có, sốt không = 0

⇒ Ta có T nước tiểu vàng, đau lưng không = 3 là lớn nhất. ta chọn T nước tiểu vàng, đau lưng không và ta có luật

RULE 4: IF Nước tiểu = Vàng AND Đau lưng = Không THEN Kết Quả = Không bệnh sỏi thận

Kết tiếp ta loại bỏ những dòng ứng với Nước tiểu=vàng và Đau lưng = Không ra khỏi Bảng 2:

Tên	Nước tiểu	Giảm cân	Đau lưng	Sốt	Kết quả
Hương	Có máu	Nhanh	Có	Không	Không bệnh sỏi thận
Tùng	Có máu	Ít	Không	Không	Không bệnh sỏi thận

Với j=2, có 6 tổ hợp gồm 2 thuộc tính:

- {Nước tiểu, Giảm cân}
- {Nước tiểu, Đau lưng}
- {Nước tiểu, Sốt}
- {Giảm cân, Đau lưng}
- {Giảm cân, Sốt}
- {Đau lưng, Sốt}

T nước tiểu có máu, giảm cân ít = 0

T nước tiểu có máu, giảm cân nhanh = 0

T nước tiểu có máu, đau lưng không = 0

T nước tiểu có máu, đau lưng có = 0

T nước tiểu có máu, sốt không = 2

T giảm cân ít, đau lưng không = 0

T giảm cân nhanh, đau lưng có = 0

T giảm cân nhanh, sốt không = 0

T giảm cân ít, sốt không =0

T đau lưng không, sốt không =0

T đau lưng có, sốt không =0

⇒ Ta có T nước tiểu có máu, sốt không =2 là lớn nhất. ta chọn T nước tiểu có máu, sốt không và ta sẽ có luật:

RULE 5: IF Nước tiểu=Có máu AND Sốt = Không THEN Kết Quả=Không bệnh sỏi thận

Kết tiếp, loại bỏ những dòng ứng với Nước tiểu= có máu và sốt = không ra khỏi bảng

⇒ Như vậy ta đã loại bỏ tất cả các dòng trong bảng 2

⇒ Thuật toán kết thúc vì tất cả các bảng đã được xét đến và các dòng trong các bảng đã được loại bỏ.

Tổng hợp các luật:

RULE 1: IF Nước tiểu = có vôi THEN Kết quả = Bệnh sỏi thận

RULE 2: IF Nước tiểu= Có máu AND Sốt = Có THEN Kết Quả = Bệnh sỏi thận

RULE 3: IF Nước tiểu = Vàng AND Đau lưng=Có THEN Kết quả=Bệnh sỏi thận

RULE 4: IF Nước tiểu = Vàng AND Đau lưng = Không THEN Kết Quả = Không bệnh sỏi thận

RULE 5: IF Nước tiểu=Có máu AND Sốt = Không THEN Kết Quả=Không bệnh sỏi thận

3. Cài đặt ứng dụng minh họa

Hai chuyên đề nổi bật là giải thuật để xây dựng cây định danh và tìm ra tri thức cho

mẫu dữ liệu thực tế là giải thuật

Quinlan và giải thuật ILA. Trong phần này ứng dụng

chỉ minh họa cho giải thuật ILA để tìm ra tri thức cho bảng dữ liệu.

Phương pháp Naive Bayes với Laplace Correction

Bước 1: Xác suất phân loại của bộ dữ liệu

Gọi N là tổng số mẫu, C là số lượng lớp và N_c là số mẫu thuộc lớp c . Xác suất tiên nghiệm của lớp c được tính như sau:

$$P(c) = \frac{N_c + 1}{N + C}.$$

Bước 2: Xác suất có điều kiện cho thuộc tính

Cho thuộc tính A và lớp c , xác suất có điều kiện $P(A|c)$ được ước lượng bằng:

$$P(A|c) = \frac{N_{A,c} + 1}{N_c + C_A},$$

trong đó: - $N_{A,c}$ là số mẫu có thuộc tính A và thuộc lớp c . - C_A là số giá trị khả dĩ của thuộc tính A .

Bước 3: Xác suất hậu nghiệm của lớp

Với một tập thuộc tính $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$, xác suất hậu nghiệm $P(c|\mathbf{A})$ được tính như sau:

$$P(c|\mathbf{A}) \propto P(c) \prod_{i=1}^n P(A_i|c).$$

Bước 4: Phân loại

Dự đoán lớp \hat{c} bằng cách chọn lớp có xác suất hậu nghiệm lớn nhất:

$$\hat{c} = \arg \max_{c \in C} P(c|\mathbf{A}).$$

BÀI 4 (Bài 1 Đề Cơ sở Trí tuệ nhân tạo năm 2023-2024). Cho bảng quan sát về thời tiết như sau:

# ex.	Weather	Parents	Cash	Exam	Decision
1	sunny	visit	rich	yes	cinema
2	sunny	no-visit	rich	no	tennis
3	windy	visit	rich	no	cinema
4	rainy	visit	poor	yes	cinema
5	rainy	no-visit	rich	no	stay-in
6	rainy	visit	poor	no	cinema
7	windy	no-visit	poor	yes	cinema
8	windy	no-visit	rich	yes	shopping
9	windy	visit	rich	no	cinema
10	sunny	no-visit	rich	no	tennis
11	sunny	no-visit	poor	yes	???

a) Sử dụng độ đo sau để xây dựng cây định danh và tìm bộ luật để phân lớp.
Độ đo **Information Gain (IG)**:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Trong đó:

- $Value(A)$ là tập tất cả các giá trị có thể có đối với thuộc tính A và S_v là tập con của S mà A có giá trị là v
- Với S bao gồm c lớp, thì Entropy của S được tính bằng công thức sau:

$$Entropy(S) = - \sum_{i=1}^c p_i \log_2 p_i$$

Ở đây p_i là tỉ lệ của các mẫu thuộc lớp i trong tập S .

- b) Cho biết lớp (Class) của mẫu 11 dựa vào tập luật vừa tìm được?
c) So sánh kết quả ở câu (b) với phương pháp **Naive Bayes**

Công thức Naive Bayes

$$P(C_i | X) = \frac{P(X | C_i) \cdot P(C_i)}{P(X)}$$

Với:

- $P(C_i)$: Xác suất tiên nghiệm của lớp C_i .
- $P(X | C_i)$: Xác suất có điều kiện của các thuộc tính X dựa trên C_i .
- $P(X)$: Xác suất xảy ra của X (là hằng số, không cần tính để so sánh).

Áp dụng Laplace Correction

$$P(\text{Attribute} | C_i) = \frac{N_{\text{Attribute}, C_i} + k}{N_{C_i} + k}$$

Với:

- $N_{\text{Attribute}, C_i}$: Số lượng mẫu trong lớp C_i có giá trị thuộc tính tương ứng.
- N_{C_i} : Tổng số mẫu trong lớp C_i .
- k : Số lượng giá trị khác nhau của thuộc tính.

Bước 1: Tính cho lớp cinema

- Tổng số mẫu thuộc lớp "cinema": $N_{\text{cinema}} = 6$
- Số giá trị khác nhau cho mỗi thuộc tính:
 - Weather: $k = 3$ (sunny, rainy, windy)
 - Parents: $k = 2$ (visit, no-visit)
 - Cash: $k = 2$ (rich, poor)
 - Exam: $k = 2$ (yes, no)

Tính từng thành phần:

$$\begin{aligned} P(\text{sunny} | \text{cinema}) &= \frac{N_{\text{sunny}, \text{cinema}} + 1}{N_{\text{cinema}} + k} = \frac{1 + 1}{6 + 3} = \frac{2}{9} \\ P(\text{no-visit} | \text{cinema}) &= \frac{N_{\text{no-visit}, \text{cinema}} + 1}{N_{\text{cinema}} + k} = \frac{1 + 1}{6 + 2} = \frac{1}{4} \\ P(\text{poor} | \text{cinema}) &= \frac{N_{\text{poor}, \text{cinema}} + 1}{N_{\text{cinema}} + k} = \frac{3 + 1}{6 + 2} = \frac{1}{2} = 0.5 \\ P(\text{yes} | \text{cinema}) &= \frac{N_{\text{yes}, \text{cinema}} + 1}{N_{\text{cinema}} + k} = \frac{3 + 1}{6 + 2} = \frac{1}{2} = 0.5 \end{aligned}$$

Tổng hợp lại:

$$P(X | \text{cinema}) = \frac{2}{9} \cdot \frac{1}{4} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{72} \approx 0.014$$

Bước 2: Tính cho lớp shopping

- Tổng số mẫu thuộc lớp "shopping": $N_{\text{shopping}} = 1$
- Số giá trị khác nhau cho mỗi thuộc tính:
 - Weather: $k = 3$ (sunny, rainy, windy)
 - Parents: $k = 2$ (visit, no-visit)
 - Cash: $k = 2$ (rich, poor)
 - Exam: $k = 2$ (yes, no)

Tính từng thành phần:

$$\begin{aligned}
 P(\text{sunny} \mid \text{shopping}) &= \frac{N_{\text{sunny,shopping}} + 1}{N_{\text{shopping}} + k} = \frac{0 + 1}{1 + 3} = \frac{1}{4} \\
 P(\text{no-visit} \mid \text{shopping}) &= \frac{N_{\text{no-visit,shopping}} + 1}{N_{\text{shopping}} + k} = \frac{1 + 1}{1 + 2} = \frac{2}{3} \\
 P(\text{poor} \mid \text{shopping}) &= \frac{N_{\text{poor,shopping}} + 1}{N_{\text{shopping}} + k} = \frac{0 + 1}{1 + 2} = \frac{1}{3} \\
 P(\text{yes} \mid \text{shopping}) &= \frac{N_{\text{yes,shopping}} + 1}{N_{\text{shopping}} + k} = \frac{1 + 1}{1 + 2} = \frac{2}{3}
 \end{aligned}$$

Tổng hợp lại:

$$P(X \mid \text{shopping}) = \frac{1}{4} \cdot \frac{2}{3} \cdot \frac{1}{3} \cdot \frac{2}{3} = \frac{4}{108} = \frac{1}{27} \approx 0.037$$

Bước 3: Tính cho lớp tennis

- Tổng số mẫu thuộc lớp "tennis": $N_{\text{tennis}} = 2$
- Số giá trị khác nhau cho mỗi thuộc tính:
 - Weather: $k = 3$ (sunny, rainy, windy)
 - Parents: $k = 2$ (visit, no-visit)
 - Cash: $k = 2$ (rich, poor)
 - Exam: $k = 2$ (yes, no)

Tính từng thành phần:

$$\begin{aligned}
 P(\text{sunny} \mid \text{tennis}) &= \frac{N_{\text{sunny,tennis}} + 1}{N_{\text{tennis}} + k} = \frac{2 + 1}{2 + 3} = \frac{3}{5} \\
 P(\text{no-visit} \mid \text{tennis}) &= \frac{N_{\text{no-visit,tennis}} + 1}{N_{\text{tennis}} + k} = \frac{2 + 1}{2 + 2} = \frac{3}{4} \\
 P(\text{poor} \mid \text{tennis}) &= \frac{N_{\text{poor,tennis}} + 1}{N_{\text{tennis}} + k} = \frac{0 + 1}{2 + 2} = \frac{1}{4} \\
 P(\text{yes} \mid \text{tennis}) &= \frac{N_{\text{yes,tennis}} + 1}{N_{\text{tennis}} + k} = \frac{0 + 1}{2 + 2} = \frac{1}{4}
 \end{aligned}$$

Tổng hợp lại:

$$P(X \mid \text{tennis}) = \frac{3}{5} \cdot \frac{3}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} = \frac{9}{320} \approx 0.028125$$

Bước 4: Tính cho lớp stay-in

Tổng số mẫu thuộc lớp "stay-in": $N_{\text{stay-in}} = 1$. Tính từng thành phần:

$$\begin{aligned} P(\text{sunny} \mid \text{stay-in}) &= \frac{N_{\text{sunny, stay-in}} + 1}{N_{\text{stay-in}} + k} = \frac{0 + 1}{1 + 3} = \frac{1}{4} \\ P(\text{no-visit} \mid \text{stay-in}) &= \frac{N_{\text{no-visit, stay-in}} + 1}{N_{\text{stay-in}} + k} = \frac{1 + 1}{1 + 2} = \frac{2}{3} \\ P(\text{poor} \mid \text{stay-in}) &= \frac{N_{\text{poor, stay-in}} + 1}{N_{\text{stay-in}} + k} = \frac{0 + 1}{1 + 2} = \frac{1}{3} \\ P(\text{yes} \mid \text{stay-in}) &= \frac{N_{\text{yes, stay-in}} + 1}{N_{\text{stay-in}} + k} = \frac{0 + 1}{1 + 2} = \frac{1}{3} \end{aligned}$$

Tổng hợp lại:

$$P(X \mid \text{stay-in}) = \frac{1}{4} \cdot \frac{2}{3} \cdot \frac{1}{3} \cdot \frac{1}{3} = \frac{2}{108} = \frac{1}{54} \approx 0.0185$$

Bước 5: Tính P(X) với Laplace Correction

Đầu tiên, tính lại xác suất tiên nghiệm $P(C_i)$ với Laplace correction:

$$P(C_i) = \frac{N_{C_i} + 1}{N_{\text{total}} + k}$$

Với:

- $N_{\text{total}} = 10$ (tổng số mẫu)
- $k = 4$ (số lớp khác nhau: cinema, shopping, tennis, stay-in)

Tính xác suất tiên nghiệm cho từng lớp:

$$\begin{aligned} P(\text{cinema}) &= \frac{6 + 1}{10 + 4} = \frac{7}{14} = 0.5 \\ P(\text{shopping}) &= \frac{1 + 1}{10 + 4} = \frac{2}{14} \approx 0.143 \\ P(\text{tennis}) &= \frac{2 + 1}{10 + 4} = \frac{3}{14} \approx 0.214 \\ P(\text{stay-in}) &= \frac{1 + 1}{10 + 4} = \frac{2}{14} \approx 0.143 \end{aligned}$$

Tính lại $P(X)$ với xác suất tiên nghiệm đã điều chỉnh:

$$\begin{aligned} P(X) &= P(X \mid \text{cinema})P(\text{cinema}) + P(X \mid \text{shopping})P(\text{shopping}) \\ &\quad + P(X \mid \text{tennis})P(\text{tennis}) + P(X \mid \text{stay-in})P(\text{stay-in}) \\ &= 0.014 \cdot 0.5 + 0.037 \cdot 0.143 + 0.028125 \cdot 0.214 + 0.0185 \cdot 0.143 \\ &= 0.007 + 0.005291 + 0.006019 + 0.002646 \\ &= 0.020956 \end{aligned}$$

Bước 6: Tính $P(C_i|X)$

Áp dụng công thức Bayes với xác suất tiên nghiệm đã tính

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

Ta tính được các xác suất hậu nghiệm sau:

$$\begin{aligned}P(\text{cinema}|X) &= \frac{0.014 \cdot 0.5}{0.020956} \approx 0.334 \\P(\text{shopping}|X) &= \frac{0.037 \cdot 0.143}{0.020956} \approx 0.252 \\P(\text{tennis}|X) &= \frac{0.028125 \cdot 0.214}{0.020956} \approx 0.287 \\P(\text{stay-in}|X) &= \frac{0.0185 \cdot 0.143}{0.020956} \approx 0.126\end{aligned}$$

Kết luận

Sau khi áp dụng **Laplace correction** cho cả xác suất tiên nghiệm, ta có kết quả:

- $P(\text{cinema}|X) \approx 0.334$ (33.4%)
- $P(\text{shopping}|X) \approx 0.252$ (25.2%)
- $P(\text{tennis}|X) \approx 0.287$ (28.7%)
- $P(\text{stay-in}|X) \approx 0.126$ (12.6%)

Với **Laplace correction** áp dụng cho cả xác suất tiên nghiệm, "**cinema**" là lựa chọn có xác suất cao nhất (33.4%).