

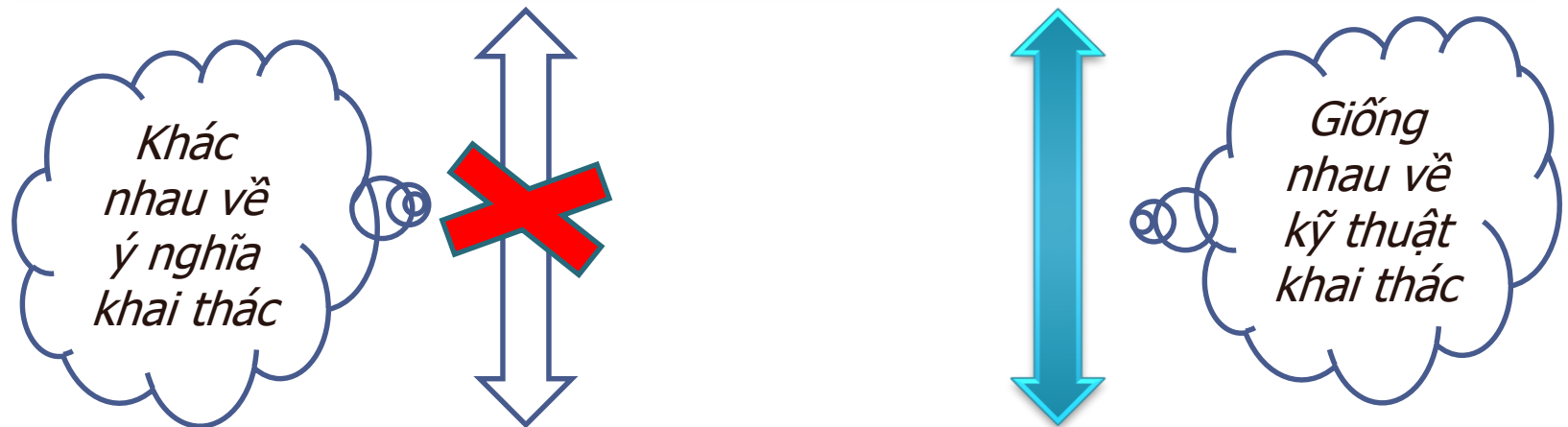


KHAI THÁC DỮ LIỆU

KHAI THÁC CÁC THÀNH PHẦN KHÔNG HỮU ÍCH

KHAI THÁC TẬP THÀNH PHẦN KHÔNG HỮU ÍCH

Khai thác Mẫu phổ biến/Tập thành phần hữu ích
(Frequent Itemsets/High Utility Itemsets)



Khai thác Tập thành phần không hữu ích
(Erased Itemsets)

Một trong những tác vụ mới nổi trong KTDL

Nội dung

1. **Bối cảnh vấn đề**
2. Phát biểu bài toán
3. Biểu diễn dữ liệu
4. Thuật toán VME
5. Tổng kết

1. Bối cảnh vấn đề



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
 $i_1, i_2, i_3, i_4, i_5, i_6, i_7$

1. Bối cảnh vấn đề



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...

1. Bối cảnh vấn đề

Ví dụ:

Với ngưỡng giảm lợi nhuận chấp nhận được là 25%, công ty có thể bỏ loại sản phẩm P_1 và không cần 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

Bài toán trở thành tìm những tập thành phần như vậy...

Nội dung

1. Bối cảnh vấn đề
2. **Phát biểu bài toán**
3. Biểu diễn dữ liệu
4. Thuật toán VME
5. Tổng kết

2. *Phát biểu bài toán*

- Cho các tập sau
 - $I = \{i_1, i_2, \dots, i_m\}$ là tập tất cả các thành phần cấu tạo nên các loại sản phẩm.
 - $DB = \{P_1, P_2, \dots, P_n\}$ là cơ sở dữ liệu sản phẩm.
Mỗi P_i là dữ liệu một loại sản phẩm, có cấu trúc:
 $\langle PID, Items, Val \rangle$
 - PID là mã loại sản phẩm.
 - $Items$ là tập các thành phần cấu tạo nên sản phẩm P_i .
 - Val là lợi nhuận mà công ty đạt được khi bán toàn bộ sản phẩm P_i .

2. Phát biểu bài toán

- Cơ sở dữ liệu mẫu

Product	PID	Items	Val (<i>Triệu đô la</i>)
P_1	1	$\{i_2, i_3, i_4, i_6\}$	50
P_2	2	$\{i_2, i_5, i_7\}$	20
P_3	3	$\{i_1, i_2, i_3, i_5\}$	50
P_4	4	$\{i_1, i_2, i_4\}$	800
P_5	5	$\{i_6, i_7\}$	30
P_6	6	$\{i_3, i_4\}$	50

2. Phát biểu bài toán

- **Định nghĩa 2.1**

Cho $A (\subseteq I)$ là một tập thành phần, *độ hỗ trợ của A* được tính theo công thức sau.

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

- Ví dụ: Với tập thành phần $P = \{i_6, i_7\}$, các loại sản phẩm có thành phần chứa i_6 hoặc i_7 , hay cả hai là P_1, P_2, P_5 . Vậy độ hỗ trợ của P là:

$$P_1.Val + P_2.Val + P_5.Val = 50 + 20 + 30 = 100$$

2. Phát biểu bài toán

- **Định nghĩa 2.2**

Cho trước một ngưỡng ξ và cơ sở dữ liệu sản phẩm DB , tập thành phần A gọi là một *tập thành phần không hữu ích* nếu:

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


- Ví dụ: Tổng lợi nhuận là 1000, $\xi = 15\%$

$$\begin{aligned} Gain(\{i_6, i_7\}) &= P_1 \cdot Val + P_2 \cdot Val + P_5 \cdot Val = 50 + 20 + 30 \\ &= 100 \leq (1000 \times 15\%) \end{aligned}$$

Nên $\{i_6, i_7\}$ là một tập thành phần không hữu ích.

2. Phát biểu bài toán

- Bài toán:** Cho cơ sở dữ liệu sản phẩm DB và một ngưỡng ξ , tìm tất cả các tập thành phần không hữu ích trong cơ sở dữ liệu.

Product	PID	Items	Val	$\xi = 15\%$	Itemset A	Gain(A)
P_1	1	$\{i_2, i_3, i_4, i_6\}$	50	 Các tập thành phần không hữu ích	$\{i_3\}$	150
P_2	2	$\{i_2, i_5, i_7\}$	20		$\{i_5\}$	70
P_3	3	$\{i_1, i_2, i_3, i_5\}$	50		$\{i_6\}$	80
P_4	4	$\{i_1, i_2, i_4\}$	800		$\{i_7\}$	50
P_5	5	$\{i_6, i_7\}$	30		$\{i_5, i_6\}$	150
P_6	6	$\{i_3, i_4\}$	50		$\{i_5, i_7\}$	100
					$\{i_6, i_7\}$	100
					$\{i_5, i_6, i_7\}$	150

Nội dung

1. Bối cảnh vấn đề
2. Phát biểu bài toán
3. **Biểu diễn dữ liệu**
4. Thuật toán VME
5. Tổng kết

3. Biểu diễn dữ liệu

- Thuật toán đầu tiên là META¹, sử dụng phương pháp quét cơ sở dữ liệu để tính độ hỗ trợ của 1 tập thành phần nên chi phí thời gian là rất lớn.

*Bản chất của
cơ sở dữ liệu
là lưu trữ
theo dòng*

Product	PID	Items	Val
P_1	1	$\{i_2, i_3, i_4, i_6\}$	50
P_2	2	$\{i_2, i_5, i_7\}$	20
P_3	3	$\{i_1, i_2, i_3, i_5\}$	50
P_4	4	$\{i_1, i_2, i_4\}$	800
P_5	5	$\{i_6, i_7\}$	30
P_6	6	$\{i_3, i_4\}$	50

**Để giảm chi phí tính độ hỗ trợ của 1 tập thành phần
cần có phương pháp hiệu quả
và *thay đổi định dạng* của cơ sở dữ liệu sản phẩm.**

¹ Deng, Z., Fang, G., Wang, Z., Xu, X.: Mining Erasable Itemsets. In: 8th IEEE International Conference on Machine Learning and Cybernetics, pp. 67–73. IEEE Press, New York (2009).

3. Biểu diễn dữ liệu

Product	PID	Items	Val
P_1	1	$\{i_2, i_3, i_4, i_6\}$	50
P_2	2	$\{i_2, i_5, i_7\}$	20
P_3	3	$\{i_1, i_2, i_3, i_5\}$	50
P_4	4	$\{i_1, i_2, i_4\}$	800
P_5	5	$\{i_6, i_7\}$	30
P_6	6	$\{i_3, i_4\}$	50

*Cơ sở dữ liệu
thông thường
(Theo dòng)*

*Biểu diễn
cơ sở dữ liệu theo
dạng đảo
(Theo cột)*

Thành phần	Danh sách đảo
i_1	<3, 50>, <4, 800>
i_2	<1, 50>, <2, 20>, <3, 50>, <4, 800>
i_3	<1, 50>, <3, 50>, <6, 50>
i_4	<1, 50>, <4, 800>, <6, 50>
i_5	<2, 20>, <3, 50>
i_6	<1, 50>, <5, 30>
i_7	<2, 20>, <5, 30>

3. Biểu diễn dữ liệu

- Cho tập thành phần $X = \{i_{x1}, i_{x2}, \dots, i_{xk}\} \subseteq I$, với $i_{xj} \in I$ ($1 \leq j \leq k$) và $x_s < x_t$ ($1 \leq s < t \leq k$). Nghĩa là các thành phần được sắp theo thứ tự xuất hiện trong I .

- **Định nghĩa 3.1**

Với mỗi $y \in I$, **PID_list** của tập 1-thành phần $\{y\}$ là:

$\{<P_{y1}.PID, P_{y1}.Val>, <P_{y2}.PID, P_{y2}.Val>, \dots, <P_{yk}.PID, P_{yk}.Val>\}$

Trong đó y là một thành phần cấu tạo sản phẩm P_{yj} ($1 \leq j \leq k$), $y \in P_{yj}.Items$, và $P_{yv}.PID < P_{yu}.PID$ với $v < u$. Nghĩa là các phần tử của PID_list được sắp theo thứ tự tăng của PID .

- Ví dụ PID_list của $\{i_2\}$:

$\{<1, 50>, <2, 20>, <3, 50>, <4, 800>\}$

3. Biểu diễn dữ liệu

- **Tính chất 3.1**

Cho $\{y\}$ là tập 1-thành phần có PID_list là:

$$\{ \langle P_{y1}.PID, P_{y1}.Val \rangle, \langle P_{y2}.PID, P_{y2}.Val \rangle, \dots, \langle P_{yk}.PID, P_{yk}.Val \rangle \}$$

Độ hỗ trợ của $\{y\}$ được tính như sau:

$$Gain(\{y\}) = \sum_{j=1}^k P_{yj}.Val$$

- Ví dụ tập $\{i_2\}$ có PID_list là:

$$\{ \langle 1, 50 \rangle, \langle 2, 20 \rangle, \langle 3, 50 \rangle, \langle 4, 800 \rangle \}$$

$$\Rightarrow Gain(\{i_2\}) = 50 + 20 + 50 + 800 = 920 \text{ (triệu)}$$

3. Biểu diễn dữ liệu

- **Định nghĩa 3.2**

Cho tập k -thành phần $X = \{i_{x1}, i_{x2}, \dots, i_{xk}\}$. Với mỗi i_{xj} ($1 \leq j \leq k$), ký hiệu PID_list của $\{i_{xj}\}$ là $PID_list(\{i_{xj}\})$.

PID_list của tập k -thành phần X , ký hiệu $PID_list(X)$ là:

$$\{ \langle P_{x1}.PID, P_{x1}.Val \rangle, \langle P_{x2}.PID, P_{x2}.Val \rangle, \dots, \langle P_{xs}.PID, P_{xs}.Val \rangle \}$$

thỏa 3 điều kiện sau:

- (1) $P_{xv}.PID < P_{xu}.PID$ với $1 \leq v < u \leq s$
- (2) $\forall \langle P_{xj}.PID, P_{xj}.Val \rangle (1 \leq j \leq s) \in PID_list(X)$
 $\Rightarrow \exists i_{xv} \in X, \langle P_{xj}.PID, P_{xj}.Val \rangle \in PID_list(\{i_{xv}\})$
- (3) Với mỗi $i_{xv} \in X$ ($1 \leq v \leq k$), $\forall \langle P_j.PID, P_j.Val \rangle \in PID_list(\{i_{xv}\})$
 $\Rightarrow \langle P_j.PID, P_j.Val \rangle \in PID_list(X)$.

$\Rightarrow PID_list(X)$ là hội các PID_list của từng thành phần trong tập X , với các phần tử được sắp tăng dần theo PID .

3. Biểu diễn dữ liệu

- Ví dụ PID_list của tập k -thành phần X :
Xét các tập 1-thành phần $\{i_3\}$, $\{i_5\}$, $\{i_6\}$ có PID_list như sau:

1-thành phần	PID_list
$\{i_3\}$	$\{<1, 50>, <3, 50>, <6, 50>\}$
$\{i_5\}$	$\{<2, 20>, <3, 50>\}$
$\{i_6\}$	$\{<1, 50>, <5, 30>\}$

⇒ PID_list của tập 3-thành phần $\{i_3, i_5, i_6\}$ là:

$\{<1, 50>, <2, 20>, <3, 50>, <5, 30>, <6, 50>\}$

3. Biểu diễn dữ liệu

- **Tính chất 3.2**

Cho tập k -thành phần X và $PID_list(X)$ là:

$$\{ \langle P_{x_1}.PID, P_{x_1}.Val \rangle, \langle P_{x_2}.PID, P_{x_2}.Val \rangle, \dots, \langle P_{x_s}.PID, P_{x_s}.Val \rangle \}$$

Độ hỗ trợ của X được tính như sau:

$$Gain(X) = \sum_{j=1}^s P_{x_j}.Val$$

- Ví dụ tập 3-thành phần $\{i_3, i_5, i_6\}$ có PID_list là:

$$\{ \langle 1, 50 \rangle, \langle 2, 20 \rangle, \langle 3, 50 \rangle, \langle 5, 30 \rangle, \langle 6, 50 \rangle \}$$

$$\begin{aligned} \Rightarrow Gain(\{i_3, i_5, i_6\}) &= 50 + 20 + 50 + 30 + 50 \\ &= 200 \text{ (triệu)} \end{aligned}$$

3. Biểu diễn dữ liệu

Product	PID	Items	Val
P_1	1	$\{i_2, i_3, i_4, i_6\}$	50
P_2	2	$\{i_2, i_5, i_7\}$	20
P_3	3	$\{i_1, i_2, i_3, i_5\}$	50
P_4	4	$\{i_1, i_2, i_4\}$	800
P_5	5	$\{i_6, i_7\}$	30
P_6	6	$\{i_3, i_4\}$	50

Khi tính độ hỗ trợ của $\{i_3\}$, không cần xét các sản phẩm P_2 , P_4 và P_5 .

Thành phần	Danh sách đảo
i_1	<3, 50>, <4, 800>
i_2	<1, 50>, <2, 20>, <3, 50>, <4, 800>
i_3	<1, 50>, <3, 50>, <6, 50>
i_4	<1, 50>, <4, 800>, <6, 50>
i_5	<2, 20>, <3, 50>
i_6	<1, 50>, <5, 30>
i_7	<2, 20>, <5, 30>

Cấu trúc **PID_list** cho phép **tính độ hỗ trợ** của một tập thành phần rất nhanh, đồng thời **loại bỏ dữ liệu dư thừa** một cách tự nhiên...

Nội dung

1. Bối cảnh vấn đề
2. Phát biểu bài toán
3. Biểu diễn dữ liệu
4. Thuật toán VME
5. Tổng kết

4. Thuật toán VME

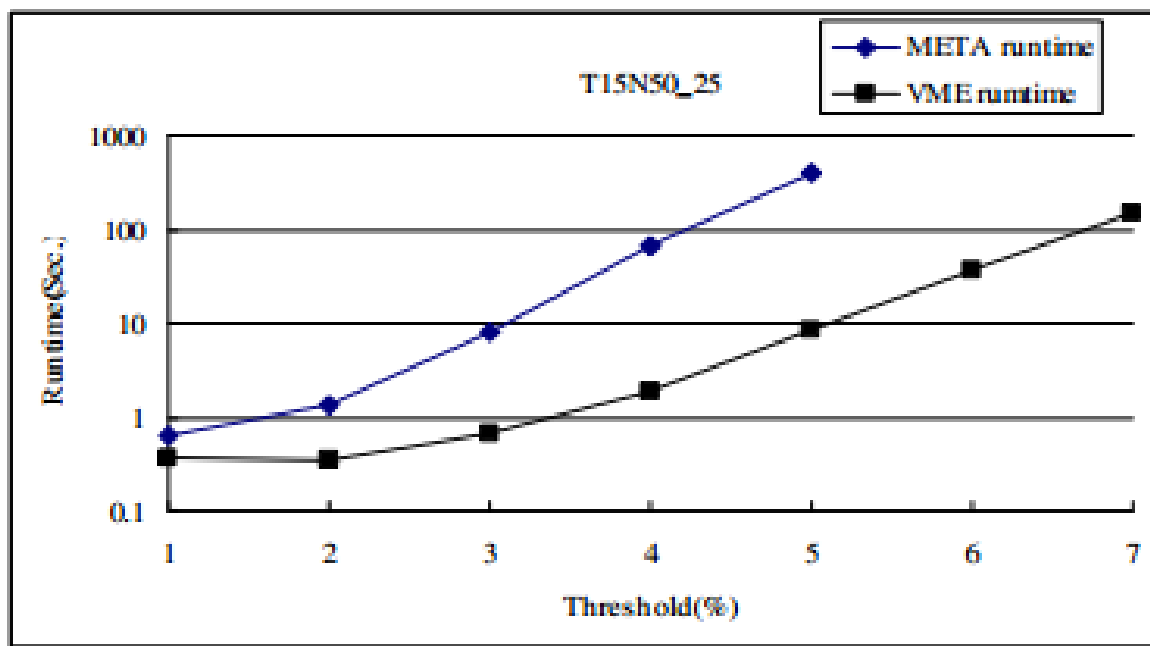
Vertical-format-based algorithm for **M**ining **E**rasable Itemsets

Ý tưởng tìm kiếm theo từng cấp độ (level-wise)

VME

Khai thác các tập thành phần không hữu ích dựa trên cấu trúc PID_list

Hiệu quả trung bình gấp 2 lần so với thuật toán META



4. Thuật toán VME – Các khái niệm

- Bổ đề 4.1

Cho hai tập thành phần $X(\subseteq I)$ và $Y(\subseteq I)$. Nếu $X \subseteq Y$ thì $Gain(X) \leq Gain(Y)$.

Chứng minh:

Xét P_k tùy ý thỏa $P_k.Items \cap X \neq \emptyset$. Do $Y \supseteq X$ nên $P_k.Items \cap Y \neq \emptyset$. Vậy:

$$\{P_k \mid P_k.Items \cap X \neq \emptyset\} \subseteq \{P_k \mid P_k.Items \cap Y \neq \emptyset\}$$

Theo Định nghĩa 2.1:

$$\sum_{\{P_k \mid X \cap P_k.Items \neq \emptyset\}} P_k.Val \leq \sum_{\{P_k \mid Y \cap P_k.Items \neq \emptyset\}} P_k.Val$$

Do đó $Gain(X) \leq Gain(Y)$. (đpcm)

- Ví dụ: Tổng lợi nhuận của các tập $\{i_6\}$, $\{i_6, i_7\}$, $\{i_3, i_6, i_7\}$ là 80, 100, 200 $\Rightarrow Val(\{i_6\}) \leq Val(\{i_6, i_7\}) \leq Val(\{i_3, i_6, i_7\})$.

4. Thuật toán VME – Các khái niệm

- Bổ đề 4.2

Nếu tập thành phần X là không hữu ích và $Y \supseteq X$, thì Y cũng là một tập thành phần không hữu ích.

Chứng minh:

Cho $Y \supseteq X$. Giả thiết Y là tập thành phần hữu ích.

Theo Định nghĩa 2.2: $Gain(Y) \leq (\sum_{P_k \in DB} P_k.Val) \times \xi$

Do X là tập không hữu ích nên: $Gain(X) > \sum_{P_k \in DB} P_k.Val \times \xi$

$\Rightarrow Gain(X) > Gain(Y)$. Theo Bổ đề 4.1 thì $Gain(X) \leq Gain(Y)$, dẫn đến mâu thuẫn với giả thiết. Vậy Y là tập thành phần không hữu ích. (đpcm)

4. Thuật toán VME – Mã giả

- Thuật toán

Input: Cơ sở dữ liệu DB , tập các thành phần I , ngưỡng ξ .

Output: Tập các tập thành phần không hữu ích EI .

- Phương pháp

Duyệt DB tính tổng lợi nhuận Sum_val ;

Duyệt DB một lần nữa tìm tập tất cả các tập 1-thành phần không hữu ích E_1 và PID_list tương ứng;

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

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

$E_k = \emptyset$;

 For each k -itemset $P \in GC_k$ {

 Tính $P.gain$ theo Tính chất 3.2;

 If $P.gain \leq \xi \times Sum_val$ then $E_k = E_k \cup \{P\}$; // Định nghĩa 2.1

 }

}

Return $EI = \cup_k E_k$;

4. Thuật toán VME – Mã giả

// Sinh các tập thành phần ứng viên và PID_list tương ứng

Procedure Gen_Candidate (E_{k-1})

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 {

$X.PID_list = A_1.PID_list \cup A_2.PID_list$;

Candidates = Candidates $\cup \{ (X, X.PID_list) \}$;

}

}

}

}

Return Candidates;

4. Thuật toán VME – Mã giả

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

// X : tập ứng viên k -thành phần

// E_{k-1} : tập các tập $(k-1)$ -thành phần không hữu ích

For each $(k-1)$ -itemset $X_s \subset X$

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

 Return FALSE;

Return TRUE;

4. Thuật toán VME – Minh họa

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

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

$$\xi = 15\%$$

Product	PID	Items	Val
P_1	1	$\{i_2, i_3, i_4, i_6\}$	50
P_2	2	$\{i_2, i_5, i_7\}$	20
P_3	3	$\{i_1, i_2, i_3, i_5\}$	50
P_4	4	$\{i_1, i_2, i_4\}$	800
P_5	5	$\{i_6, i_7\}$	30
P_6	6	$\{i_3, i_4\}$	50

Bước 1: Khởi tạo

$$\text{Sum_val} = 1000$$

$$E_1 = \{\{i_3\}, \{i_5\}, \{i_6\}, \{i_7\}\}$$

PID_list _{E1}	PID_list
$\{i_3\}$	$\{<1, 50>, <3, 50>, <6, 50>\}$
$\{i_5\}$	$\{<2, 20>, <3, 50>\}$
$\{i_6\}$	$\{<1, 50>, <5, 30>\}$
$\{i_7\}$	$\{<2, 20>, <5, 30>\}$

4. Thuật toán VME – Minh họa

$$Sum_val = 1000$$

$$E_1 = \{\{i_3\}, \{i_5\}, \{i_6\}, \{i_7\}\}$$

PID_list _{E1}	PID_list
{i ₃ }	{<1, 50>, <3, 50>, <6, 50>}
{i ₅ }	{<2, 20>, <3, 50>}
{i ₆ }	{<1, 50>, <5, 30>}
{i ₇ }	{<2, 20>, <5, 30>}

Bước 2: Tìm các tập không hữu ích k -thành phần

$k = 2$

$$GC_2 = \{ \{i_3, i_5\}, \{i_3, i_6\}, \{i_3, i_7\}, \{i_5, i_6\}, \{i_5, i_7\}, \{i_6, i_7\} \}$$

PID_list _{GC2}	PID_list
{i ₃ , i ₅ }	{<1, 50>, <2, 20>, <3, 50>, <6, 50>}
{i ₃ , i ₆ }	{<1, 50>, <3, 50>, <5, 30>, <6, 50>}
{i ₃ , i ₇ }	{<1, 50>, <2, 20>, <3, 50>, <5, 30>, <6, 50>}
{i ₅ , i ₆ }	{<1, 50>, <2, 20>, <3, 50>, <5, 30>}
{i ₅ , i ₇ }	{<2, 20>, <3, 50>, <5, 30>}
{i ₆ , i ₇ }	{<1, 50>, <2, 20>, <5, 30>}

4. Thuật toán VME – Minh họa

Bước 2: Tìm các tập không hữu ích k -thành phần
 $k = 2$ (tt)

$$GC_2 = \{ \{i_3, i_5\}, \{i_3, i_6\}, \{i_3, i_7\}, \{i_5, i_6\}, \{i_5, i_7\}, \{i_6, i_7\} \}$$

PID_list _{GC2}	PID_list	P.Gain
$\{i_3, i_5\}$	$\{<1, 50>, <2, 20>, <3, 50>, <6, 50>\}$	170
$\{i_3, i_6\}$	$\{<1, 50>, <3, 50>, <5, 30>, <6, 50>\}$	180
$\{i_3, i_7\}$	$\{<1, 50>, <2, 20>, <3, 50>, <5, 30>, <6, 50>\}$	200
$\{i_5, i_6\}$	$\{<1, 50>, <2, 20>, <3, 50>, <5, 30>\}$	150
$\{i_5, i_7\}$	$\{<2, 20>, <3, 50>, <5, 30>\}$	100
$\{i_6, i_7\}$	$\{<1, 50>, <2, 20>, <5, 30>\}$	100

$$\Rightarrow E_2 = \{ \{i_5, i_6\}, \{i_5, i_7\}, \{i_6, i_7\} \}$$

PID_list _{E2}	PID_list
$\{i_5, i_6\}$	$\{<1, 50>, <2, 20>, <3, 50>, <5, 30>\}$
$\{i_5, i_7\}$	$\{<2, 20>, <3, 50>, <5, 30>\}$
$\{i_6, i_7\}$	$\{<1, 50>, <2, 20>, <5, 30>\}$

4. Thuật toán VME – Minh họa

PID_list _{E2}	PID_list
$\{i_5, i_6\}$	$\{<1, 50>, <2, 20>, <3, 50>, <5, 30>\}$
$\{i_5, i_7\}$	$\{<2, 20>, <3, 50>, <5, 30>\}$
$\{i_6, i_7\}$	$\{<1, 50>, <2, 20>, <5, 30>\}$

Bước 2: Tìm các tập không hữu ích k -thành phần

$k = 3$

$$GC_3 = \{ \{i_5, i_6, i_7\} \}$$

PID_list _{GC3}	PID_list
$\{i_5, i_6, i_7\}$	$\{<1, 50>, <2, 20>, <3, 50>, <5, 30>\}$

PID_list _{GC3}	PID_list	P.Gain
$\{i_5, i_6, i_7\}$	$\{<1, 50>, <2, 20>, <3, 50>, <5, 30>\}$	150

$$\Rightarrow E_3 = \{ \{i_5, i_6, i_7\} \}$$

PID_list _{E2}	PID_list
$\{i_5, i_6, i_7\}$	$\{<1, 50>, <2, 20>, <3, 50>, <5, 30>\}$

4. Thuật toán VME – Minh họa

PID_list _{E3}	PID_list
$\{i_5, i_6, i_7\}$	$\{<1, 50>, <2, 20>, <3, 50>, <5, 30>\}$

Bước 2: Tìm các tập không hữu ích k -thành phần

$k = 4$

$$GC_4 = \emptyset \Rightarrow E_4 = \emptyset$$

Bước 3: Trả về kết quả

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

$$= \{\{i_3\}, \{i_5\}, \{i_6\}, \{i_7\}, \{i_5, i_6\}, \{i_5, i_7\}, \{i_6, i_7\}, \{i_5, i_6, i_7\}\}$$

Nội dung

1. Bối cảnh vấn đề
2. Phát biểu bài toán
3. Biểu diễn dữ liệu
4. Thuật toán VME
5. **Tổng kết**

5. Tổng kết

- Tác vụ khai thác các tập thành phần không hữu ích.
- Nhược điểm của những thuật toán trước đó là truy xuất cơ sở dữ liệu theo dòng.
- Cấu trúc dữ liệu mới PID_list cho phép biểu diễn dữ liệu sản phẩm theo cột.
- Thuật toán VME khai thác hiệu quả các tập thành phần không hữu ích, dựa trên cấu trúc PID_list.
- Tính độ hỗ trợ của một tập thành phần rất nhanh và tự động loại bỏ dữ liệu dư thừa.
- Hiệu quả về thời gian gấp 2 lần so với thuật toán đầu tiên trong lĩnh vực này là META.