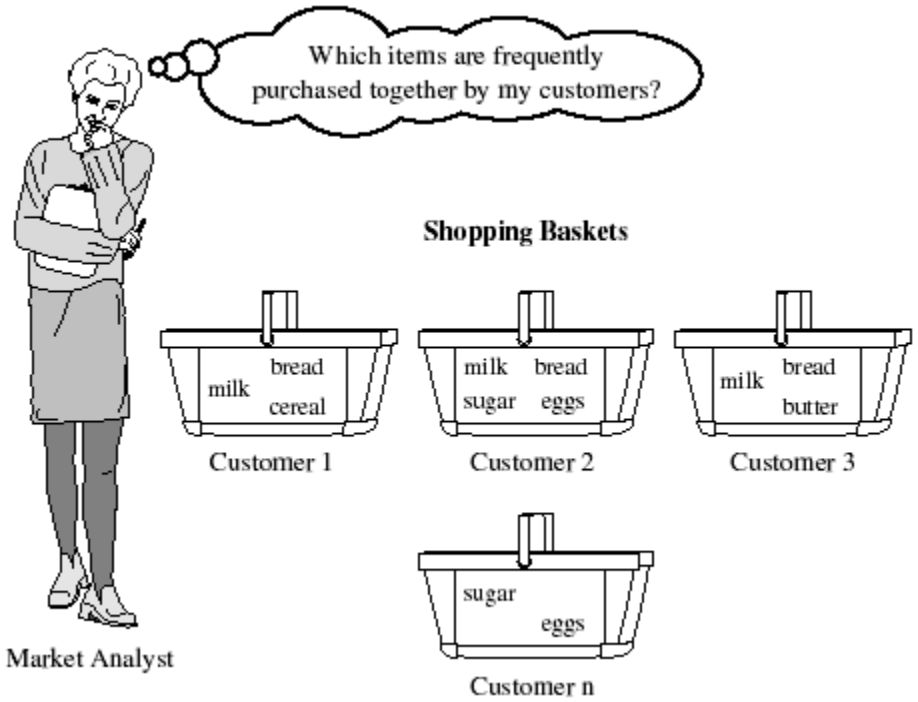
Khai phá luật kết hợp

TS. Nguyễn Quốc Tuấn

Nội dung

- Tổng quan về khai phá luật kết hợp
- Biểu diễn luật kết hợp
- Giải thuật Apriori
- Giải thuật FP_Growth

Tổng quan về luật kết hợp



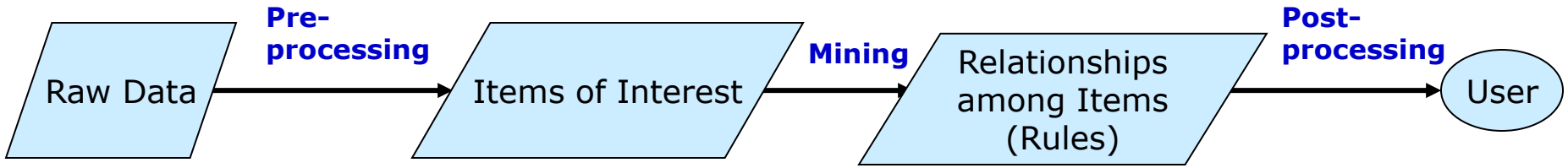

Đắc Nhân Tâm - Cuốn Sách Hay Nhất Của Mọi Thời Đại Đưa Bạn Đến Thành Công - Sách Vinabook.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
(11 người đánh giá)
Số trang: 320
Kích thước: 14.5x20.5 cm
Trọng lượng: 370 gram
(Chi tiết về phí vận chuyển)
Hình thức bìa: Bìa mềm
Ngày xuất bản: 09 - 2008
Số lần xem: 87539
In trang này
Gửi cho bạn bè
Giá bìa: 48.000 VNĐ
Giá bán: 48.000 VNĐ
Giảm giá: (0%)
Vietnam đồng
Xếp hạng: 12 (trong những cuốn Sách bán chạy)
Sách nên mua kèm với sách này
Nên mua cuốn sách trên cùng với cuốn Quẳng Gánh Lo Đi Và Vui Sống - Những Ý Tưởng Tuyệt Vời Để Sống Thanh Thản Và Hạnh Phúc - Nxb Trẻ
Giá bán tất cả: 96.000 VNĐ
Thêm tất cả vào giỏ hàng
Khách hàng mua cuốn sách trên cũng từng mua một trong những cuốn sách dưới đây
Trang: 1 / 10
Người Giỏi Không Phải Là Người Làm ... Tác giả: Donna M. Genett
Giá bán: 24.000 VNĐ
Quẳng Gánh Lo Đi Và Vui Sống - Những Ý ... Tác giả: Dale Carnegie
Giá bán: 48.000 VNĐ
Lập Bản Đồ Tư Duy - Công Cụ Tư Duy ... Tác giả: Tony Buzan
Giá bán: 24.000 VNĐ
Nguyên Lý 80/20 - Bí Quyết Làm Ít Được ... Tác giả: Richard Koch
Giá bán: 70.000 VNĐ
Bài Giảng Cuối Cùng - The Last Lecture ... Tác giả: Randy Pausch
Giá bán: 58.000 VNĐ
Done
Internet

Tổng quan về khai phá luật kết hợp

- Quá trình khai phá luật kết hợp
- Các khái niệm cơ bản

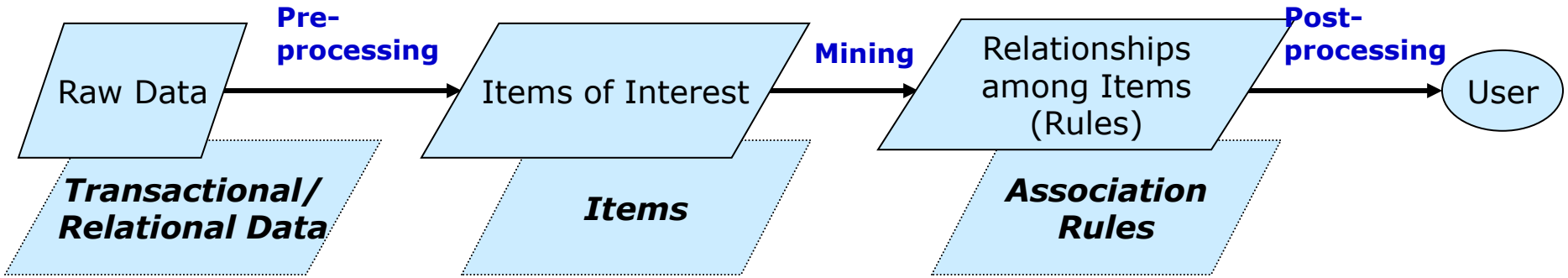
Tổng quan về khai phá luật kết hợp

- Quá trình khai phá luật kết hợp



Tổng quan về khai phá luật kết hợp

- Quá trình khai phá luật kết hợp



Transaction	Items_bought
2000	A, B, C
1000	A, C
4000	A, D
5000	B, E, F
…	

A, B, C, D, F,
…

$A \rightarrow C$ (50%, 66.6%)
…

Tổng quan về khai phá luật kết hợp

- Dữ liệu mẫu của AllElectronics

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

Tổng quan về khai phá luật kết hợp

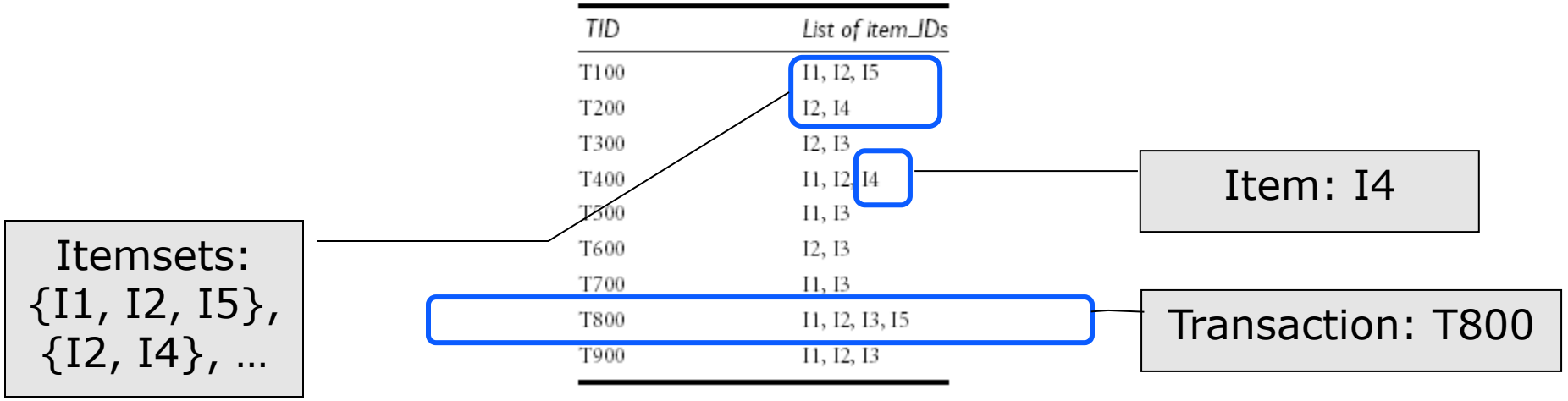
- **Các khái niệm cơ bản**
  - Item (phần tử)
  - Itemset (tập phần tử)
  - Transaction (giao dịch)
  - Association (sự kết hợp) và association rule (luật kết hợp)
  - Support (độ hỗ trợ)
  - Confidence (độ tin cậy)
  - Frequent itemset (tập phần tử phổ biến/thường xuyên)
  - Strong association rule (luật kết hợp mạnh)

Tổng quan về khai phá luật kết hợp

- **Các khái niệm cơ bản**
  - ***Item*** (phần tử)
    - Các phần tử, mẫu, đối tượng đang được quan tâm.
    - I = {I1, I2, …, Im}: tập tất cả m phần tử có thể có trong tập dữ liệu
  - ***Itemset*** (tập phần tử)
    - Tập hợp các items
    - Một itemset có k items gọi là k-itemset.
  - ***Transaction*** (giao dịch)
    - Lần thực hiện tương tác với hệ thống
    - Liên hệ với một tập D gồm các phần tử được giao dịch

Tổng quan về khai phá luật kết hợp

- Dữ liệu mẫu của AllElectronics



Tổng quan về khai phá luật kết hợp

- **Các khái niệm cơ bản**
  - ***Association*** (Sự kết hợp) : các phần tử cùng xuất hiện với nhau trong một hay nhiều giao dịch.
    - Thể hiện mối liên hệ giữa các phần tử/các tập phần tử
  - ***Association rule*** (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
    - Cho A và B là các tập phần tử, luật kết hợp giữa A và B là A → B.
      - B xuất hiện trong điều kiện A xuất hiện.

Tổng quan về khai phá luật kết hợp

- **Các khái niệm cơ bản**
  - ***Support*** (độ hỗ trợ)
    - Độ đo đo tần số xuất hiện của các phần tử/tập phần tử.

      $$\text{Support}(A) = P(A) = \text{support\_count}(A)/|D|$$

    - Minimum support threshold (ngưỡng hỗ trợ tối thiểu) – min_sup
  - ***Confidence*** (độ tin cậy)
    - Độ đo đo tần số xuất hiện của một tập phần tử trong điều kiện xuất hiện của một tập phần tử khác.

    $$\text{Confidence}(A \Rightarrow B) = P(B|A) = \text{support}(A \cup B)/\text{support}(A) = \text{support\_count}(A \cup B)/\text{support\_count}(A)$$

    - Minimum confidence threshold (ngưỡng tin cậy tối thiểu) – min_conf

Tổng quan về khai phá luật kết hợp

- **Các khái niệm cơ bản**
  - ***Frequent itemset*** (Tập phần tử phổ biến)
    - Tập phần tử có support thỏa min_sup.
    - Cho A là một itemset
      - A là tập phổ biến nếu support(A) >= min_sup.
  - ***Strong association rule*** (Luật kết hợp mạnh)
    - Luật kết hợp có support và confidence thỏa minimum support threshold và minimum confidence threshold.
    - Cho luật kết hợp A→B giữa A và B, A và B là itemsets
      - A→B là luật kết hợp mạnh nếu

        support(A→B) >= min_sup và confidence(A→B) >= min_conf.

Nội dung

- Tổng quan về khai phá luật kết hợp
- **Biểu diễn luật kết hợp**
- Giải thuật Apriori
- Giải thuật FP_Growth

Biểu diễn luật kết hợp

- Luật kết hợp giữa các tập phần tử phổ biến: A→B [support, confidence]
  - A và B là các frequent itemsets
  - Support(A→B) = Support(A U B) >= min_sup
  - Confidence(A→B) = Support(A U B)/Support(A) >= min_conf

Biểu diễn luật kết hợp

- Quá trình tạo các luật kết hợp mạnh từ các tập phổ biến
  - Cho mỗi tập phổ biến *l*, tạo các tập con không rỗng của *l*.
    - Support(*l*) >= min_sup
  - Cho mỗi tập con không rỗng *s* của *l*, tạo ra luật "*s* → (*l*-*s*)" nếu Support_count(*l*)/Support_count(*s*) >= min_conf

Nội dung

- Tổng quan về khai phá luật kết hợp
- Biểu diễn luật kết hợp
- **Giải thuật Apriori**
- Giải thuật FP_Growth

Giải thuật Apriori

- Tính chất của thuật toán: Mọi tập con khác rỗng của một tập phổ biến cũng phải là tập phổ biến.
- Mô tả thuật toán:
  - Bước 1: Duyệt (Scan) toàn bộ transaction database để có được tập ứng viên của 1-itemset (C_1), so sánh C_1 với min_sup, để có được 1-itemset (L_1).
  - Bước 2: Sử dụng L_{k-1} nối (join) L_{k-1} để sinh ra candidate k-itemset. Loại bỏ các itemsets không phải là frequent itemsets thu được k-itemset.
  - Bước 3: Scan transaction database để có được support của mỗi candidate k-itemset, so sánh C_k với min_sup để thu được frequent k –itemset (L_k).
  - Bước 4: Lặp lại từ bước 2 cho đến khi tập ứng viên trống (không tìm thấy frequent itemsets).

Giải thuật Apriori

Input:

- D, a database of transactions;
- *min_sup*, the minimum support count threshold.

Output: L, frequent itemsets in D.

Method:

```
(1)   L1 = find_frequent_1-itemsets(D);
(2)   for (k = 2; Lk−1 ≠ ϕ; k++) {
(3)       Ck = apriori_gen(Lk−1);
(4)       for each transaction t ∈ D { // scan D for counts
(5)           Ct = subset(Ck, t); // get the subsets of t that are candidates
(6)           for each candidate c ∈ Ct
(7)               c.count++;
(8)       }
(9)       Lk = {c ∈ Ck | c.count ≥ min_sup}
(10)  }
(11)  return L = ∪k Lk;
```

Giải thuật Apriori

```
procedure apriori_gen(L_{k-1}:frequent (k-1)-itemsets)
(1)     for each itemset l_1 ∈ L_{k-1}
(2)         for each itemset l_2 ∈ L_{k-1}
(3)             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 {
(4)                 c = l_1 ⋈ l_2; // join step: generate candidates
(5)                 if has_infrequent_subset(c, L_{k-1}) then
(6)                     delete c; // prune step: remove unfruitful candidate
(7)                 else add c to C_k;
(8)             }
(9)     return C_k;

procedure has_infrequent_subset(c: candidate k-itemset;
            L_{k-1}: frequent (k-1)-itemsets); // use prior knowledge
(1)     for each (k-1)-subset s of c
(2)         if s ∉ L_{k-1} then
(3)             return TRUE;
(4)     return FALSE;
```

Giải thuật Apriori

- Dữ liệu mẫu của AllElectronics

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

Giải thuật Apriori

- Min_sup=20%

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

Scan D for count of each candidate →

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count →

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for count of each candidate →

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count →

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Generate C_3 candidates from L_2 →

C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

Scan D for count of each candidate →

C_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Compare candidate support count with minimum support count →

L_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Khai phá luật kết hợp từ tập phổ biến

- Min_conf =50%

$l = \{I1, I2, I5\}$

nonempty subsets of l are $\{I1, I2\}$, $\{I1, I5\}$, $\{I2, I5\}$, $\{I1\}$, $\{I2\}$, and $\{I5\}$

$I1 \wedge I2 \Rightarrow I5$, $confidence = 2/4 = 50\%$
$I1 \wedge I5 \Rightarrow I2$, $confidence = 2/2 = 100\%$
$I2 \wedge I5 \Rightarrow I1$, $confidence = 2/2 = 100\%$
$I1 \Rightarrow I2 \wedge I5$, $confidence = 2/6 = 33\%$
$I2 \Rightarrow I1 \wedge I5$, $confidence = 2/7 = 29\%$
$I5 \Rightarrow I1 \wedge I2$, $confidence = 2/2 = 100\%$

Min_conf = 50%

$I1 \wedge I2 \Rightarrow I5$
$I1 \wedge I5 \Rightarrow I2$
$I2 \wedge I5 \Rightarrow I1$
$I5 \Rightarrow I1 \wedge I2$

Giải thuật Apriori

- Đặc điểm
  - Tạo ra nhiều tập dự tuyển
    - Một k-itemset cần ít nhất 2^k -1 itemsets dự tuyển trước đó.
  - Kiểm tra tập dữ liệu nhiều lần
    - Chi phí lớn khi kích thước các itemsets tăng lên dần.
    - Nếu k-itemsets được khám phá thì cần kiểm tra tập dữ liệu k+1 lần.

- Bài tập 1: Cho I = {A, B, C, D, E, F} và cơ sở dữ liệu giao dịch sau:

T1	{A, B, C, F}
T2	{A, B, E, F}
T3	{A, C}
T4	{D, E}
T5	{B, F}

- Tìm tập phổ biến và tập luật kết hợp mạnh với min_sup=25% và min_conf=75%

$mincount = \lceil min\,sup * |D| \rceil = \lceil 25\% * 5 \rceil = \lceil 1.25 \rceil = 2$

Tập mục	Số lần xuất hiện
{A}	3
{B}	3
{C}	2
~~{D}~~	~~1~~
{E}	2
{F}	3

Sinh các tập phổ biến có độ dài 1

L_1	Số lần xuất hiện
{A}	3
{B}	3
{C}	2
{E}	2
{F}	3

Sinh các tập có độ dài 2 bằng cách nối các tập có độ dài 1

Tập mục
{A, B}
{A, C}
{A, E}
{A, F}
{B, C}
{B, E}
{B, F}
{C, E}
{C, F}
{E, F}

C_2	Số lần xuất hiện
{A, B}	2
{A, C}	2
~~{A, E}~~	~~1~~
{A, F}	2
~~{B, C}~~	~~1~~
~~{B, E}~~	~~1~~
{B, F}	3
~~{C, E}~~	~~0~~
~~{C, F}~~	~~1~~
~~{E, F}~~	~~1~~

L_2	Số lần xuất hiện
{A, B}	2
{A, C}	2
{A, F}	2
{B, F}	3

Sinh các tập mục có độ dài 3 từ tập phổ biến L_2

Tập mục
~~{A, B, C}~~
{A, B, F}
~~{A, C, F}~~

Loại các tập mục không thỏa mãn nguyên lý Apriori

C_3	Số lần xuất hiện
{A, B, F}	2

L_3	Số lần xuất hiện
{A, B, F}	2

T1	{A, B, C, F}
T2	{A, B, E, F}
T3	{A, C}
T4	{D, E}
T5	{B, F}

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

Nội dung

- Tổng quan về khai phá luật kết hợp
- Biểu diễn luật kết hợp
- Giải thuật Apriori
- **Giải thuật FP_Growth**

Giải thuật 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
    - Tập mục không thường xuyên bị loại bỏ sớm.
  - Đảm bảo kết quả khai phá không bị ảnh hưởng
- Phương pháp chia-để-trị (divide-and-conquer)
  - Quá trình khai phá được chia thành các công tác nhỏ.
    - 1. Xây dựng FP-tree
    - 2. Khám phá tập mục thường xuyên với FP-tree
- Tránh tạo ra các tập dự tuyển
  - Mỗi lần kiểm tra một phần tập dữ liệu

Giải thuật FP-Growth

1. Xây dựng FP-tree
   1.1. Kiểm tra tập dữ liệu, tìm frequent 1-itemsets
   1.2. Sắp thứ tự frequent 1-itemsets theo sự giảm dần của support count (tần suất xuất hiện)
   1.3. Kiểm tra tập dữ liệu, tạo FP-tree
   - Tạo root của FP-tree, được gán nhãn “null” {}
   - Mỗi giao dịch tương ứng một nhánh của FP-tree.
   - Mỗi node trên một nhánh tương ứng một item của giao dịch.
     - Các item của một giao dịch được sắp theo giảm dần.
     - Mỗi node kết hợp với support count của item tương ứng.
   - Các giao dịch có chung items tạo thành các nhánh có prefix chung.

Giải thuật FP-Growth

2. Khám phá tập phổ biến với FP-tree

2.1. Tạo cơ sở mẫu điều kiện (Conditional pattern base) cho mỗi node của FP-tree

- Tích luỹ các prefix paths with frequency của node đó

2.2. Tạo cây FP điều kiện (Conditional FP-tree) từ mỗi tập mẫu điều kiện

- Tích lũy frequency cho mỗi item trong mỗi tập mẫu.
- Xây dựng cây FP điều kiện cho tập phổ biến của tập mẫu đó

2.3. Khám phá cây FP điều kiện và phát triển tập phổ biến một cách đệ qui

- Nếu cây FP điều kiện có một đường đi đơn thì liệt kê tất cả các itemsets.

Giải thuật FP-Growth

Input:

- D, a transaction database;
- *min_sup*, the minimum support count threshold.

Output: The complete set of frequent patterns.

Method:

1. The FP-tree is constructed in the following steps:
   (a) Scan the transaction database D once. Collect F, the set of frequent items, and their support counts. Sort F in support count descending order as L, the *list* of frequent items.
   (b) Create the root of an FP-tree, and label it as "null." For each transaction *Trans* in D do the following. Select and sort the frequent items in *Trans* according to the order of L. Let the sorted frequent item list in *Trans* be $[p|P]$, where p is the first element and P is the remaining list. Call insert_tree$([p|P], T)$, which is performed as follows. If T has a child N such that $N.item\text{-}name = p.item\text{-}name$, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same *item-name* via the node-link structure. If P is nonempty, call insert_tree(P, N) recursively.
2. The FP-tree is mined by calling FP_growth$(FP_tree, null)$, which is implemented as follows.

Giải thuật FP-Growth

```
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); }
```

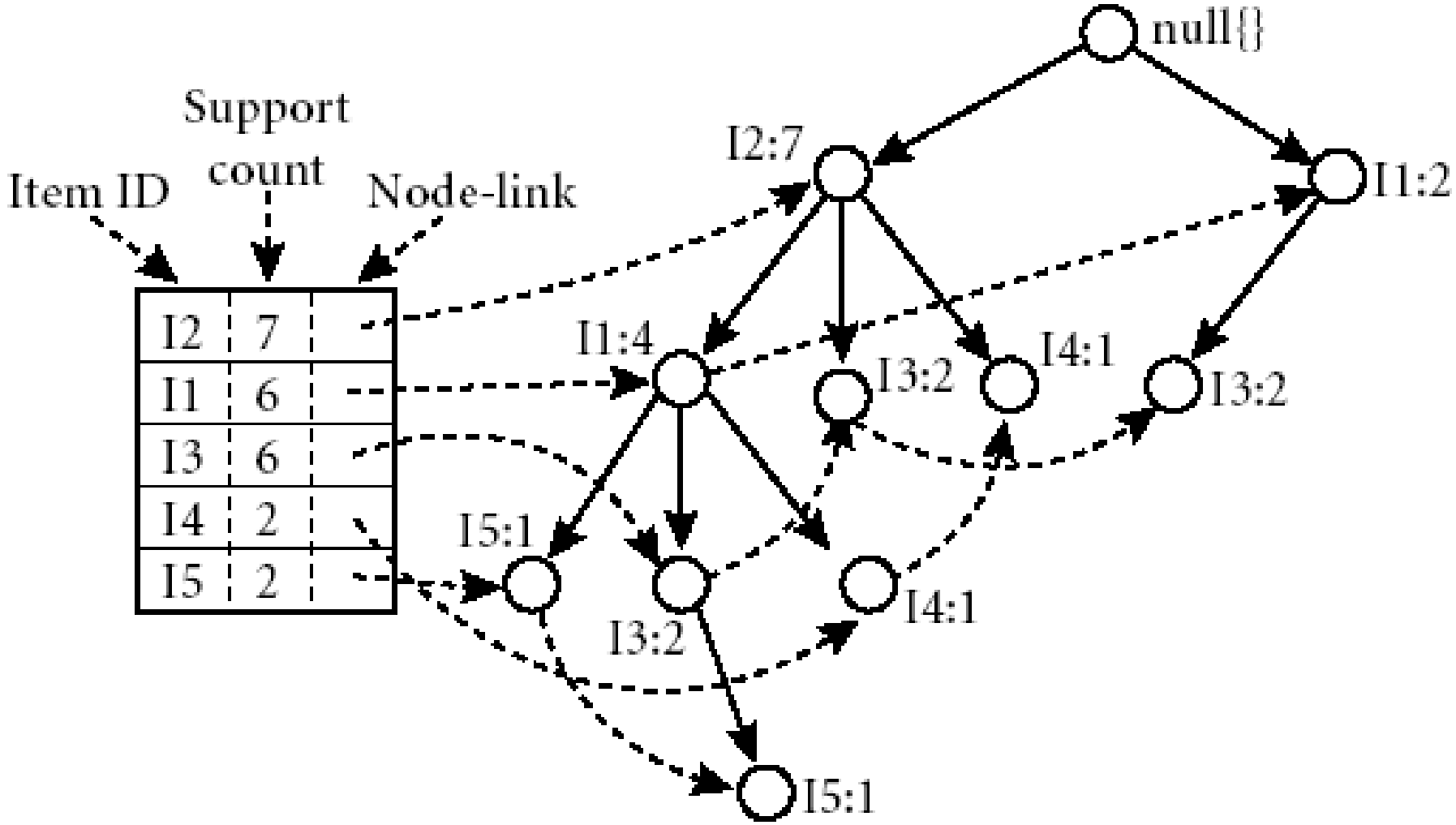
Giải thuật FP-Growth

ID	List item
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

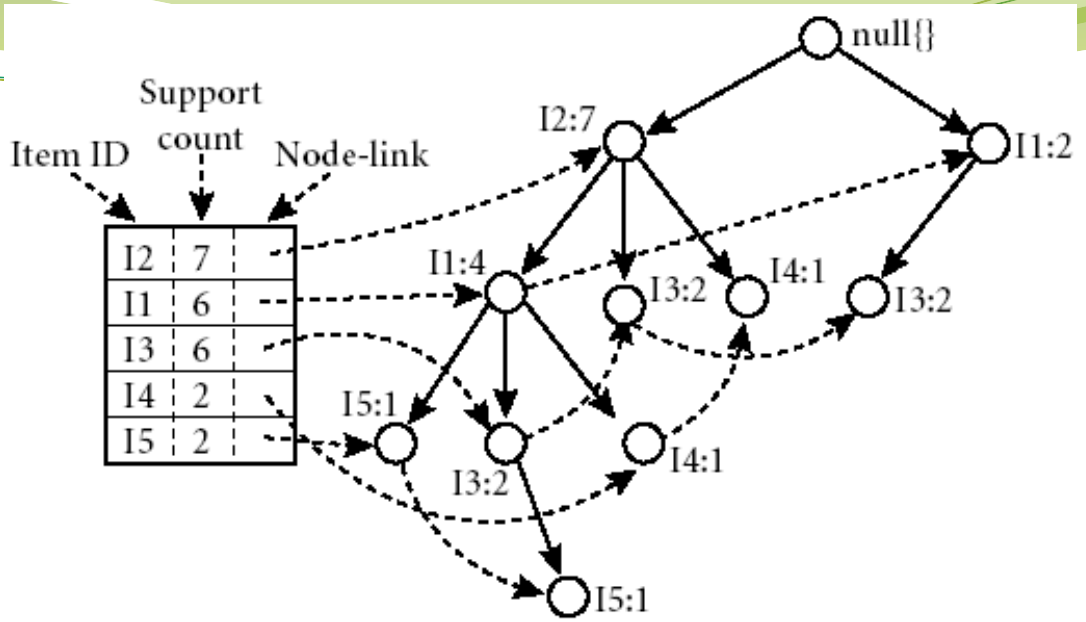
I2	7
I1	6
I3	6
I4	2
I5	2

ID	List item
T100	I2,I1,I5
T200	I2,I4
T300	I2,I3
T400	I2,I1,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I2,I1,I3,I5
T900	I2,I1,I3

Giải thuật FP-Growth



Giải thuật FP-Growth



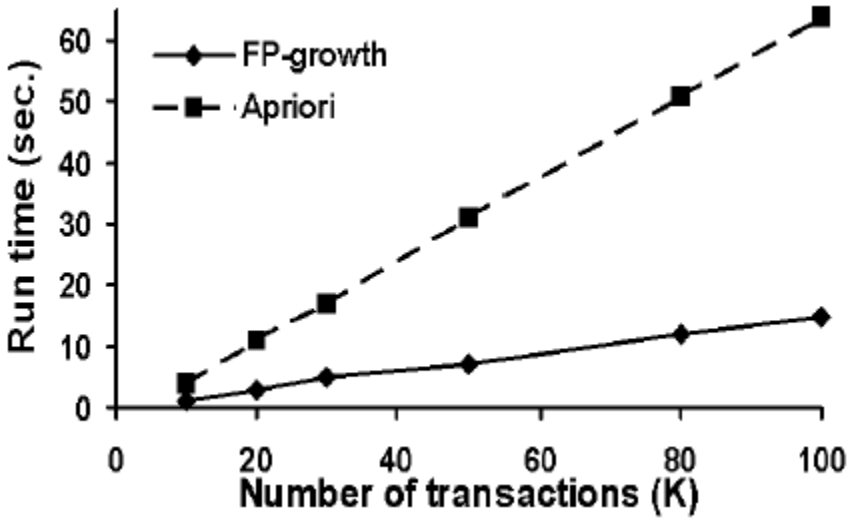
Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2,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

Giải thuật FP-Growth

- Đặc điểm
  - Không tạo tập itemsets dự tuyển
    - Không kiểm tra xem liệu itemsets dự tuyển có thực là frequent itemsets
  - Sử dụng cấu trúc dữ liệu nén dữ liệu từ tập dữ liệu
  - Giảm chi phí kiểm tra tập dữ liệu
  - Chi phí chủ yếu là đếm và xây dựng cây FP-tree lúc đầu

→ Hiệu quả và co giãn tốt cho việc khám phá các frequent itemsets dài lẫn ngắn

So sánh giữa giải thuật Apriori và giải thuật FP-Growth



Co giãn tuyến tính với số giao dịch