Khoa Khoa Học & Kỹ Thuật Máy Tính
Trường Đại Học Bách Khoa Tp. Hồ Chí Minh

Chương 6: Khai phá luật kết hợp

Khai phá dữ liệu

(Data mining)

Học kỳ 1 – 2017-2018

Nội dung

- 6.1. Tổng quan về khai phá luật kết hợp
- 6.2. Biểu diễn luật kết hợp
- 6.3. Khám phá các mẫu thường xuyên
- 6.4. Khám phá các luật kết hợp từ các mẫu thường xuyên
- 6.5. Khám phá các luật kết hợp dựa trên ràng buộc
- 6.6. Phân tích tương quan
- 6.7. Tóm tắt

Tài liệu tham khảo

Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques", Second Edition, Morgan Kaufmann Publishers, 2006.

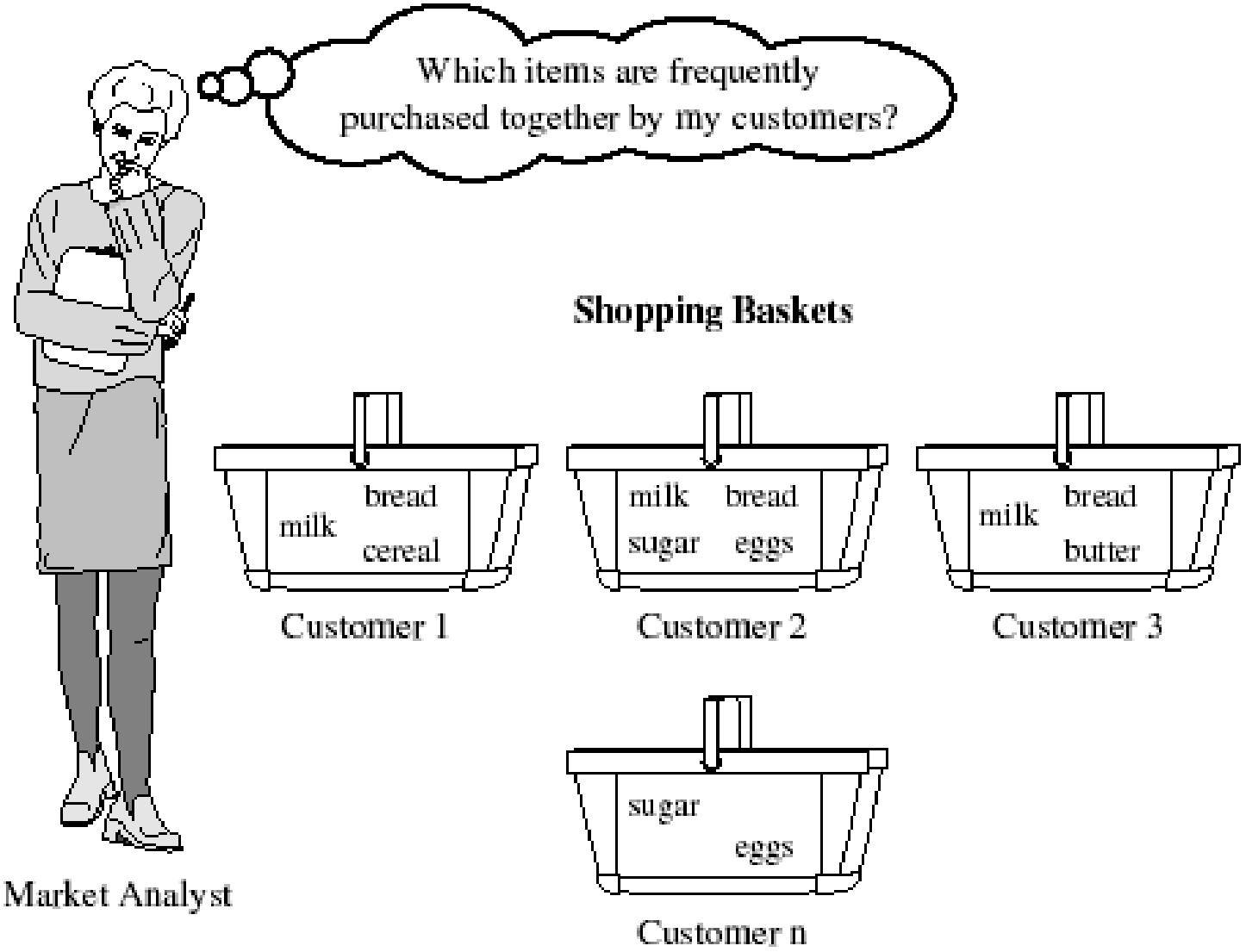
5.1 Basic Concepts and a Road Map (227-232)

5.2 Efficient and Scalable Frequent Itemset Mining Methods (234-248)

5.4 From Association Mining to Correlation Analysis

(259-261)

6.0. Tình huống 1 – Market basket analysis



6.0. Tình huống 2 - Tiếp thị chéo

6.0. Tình huống 2 - Tiếp thị chéo

6.0. Tình huống …

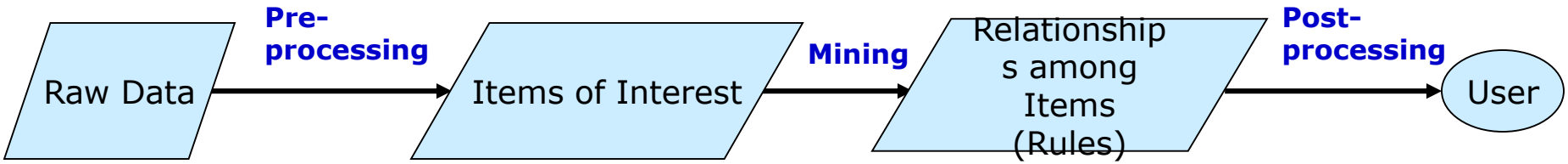
- Phân tích dữ liệu giỏ hàng (basket data analysis)
- Tiếp thị chéo (cross-marketing)
- Thiết kế catalog (catalog design)
- Phân loại dữ liệu (classification) và gom cụm dữ liệu (clustering) với các mẫu phổ biến
- …

6.1. Tổng quan về khai phá luật kết hợp

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

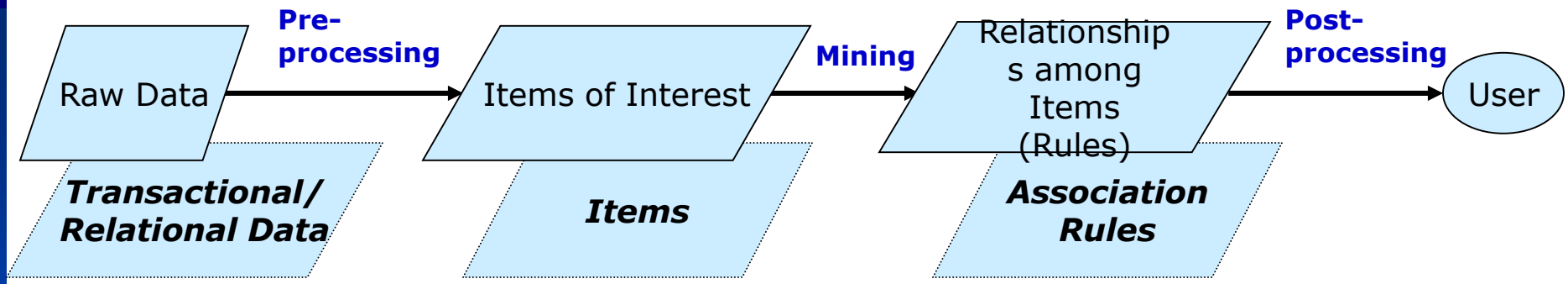
6.1. Tổng quan về khai phá luật kết hợp

- Quá trình khai phá luật kết hợp



6.1. 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%)
…

Bài toán phân tích giỏ thị trường

6.1. Tổng quan về khai phá luật kết hợp

- Dữ liệu mẫu của AllElectronics (sau quá trình tiền xử lý)

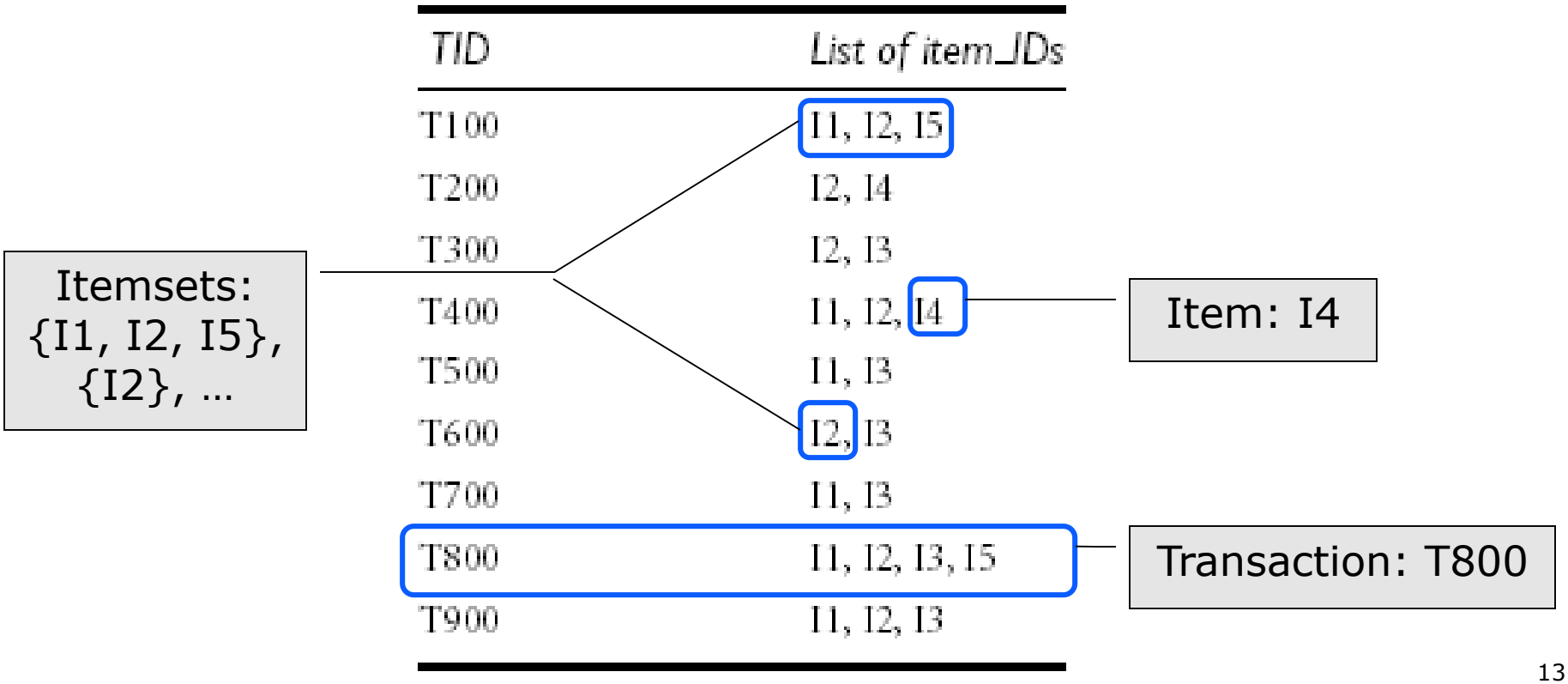
TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

6.1. 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)

6.1. Tổng quan về khai phá luật kết hợp

- Dữ liệu mẫu của AllElectronics (sau quá trình tiền xử lý)



6.1. 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.
    - $I = \{I1, I2, \ldots, Im\}$: 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 (ví dụ: giao dịch "khách hàng mua hàng")
    - Liên hệ với một tập T gồm các phần tử được giao dịch

6.1. Tổng quan về khai phá luật kết hợp

- Các khái niệm cơ bản
  - Association (sự kết hợp) và association rule (luật kết hợp)
    - 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
    - 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.

6.1. 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.
    - Minimum support threshold (ngưỡng hỗ trợ tối thiểu)
      - Giá trị support nhỏ nhất được chỉ định bởi người dùng.
  - 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.
    - Minimum confidence threshold (ngưỡng tin cậy tối thiểu)
      - Giá trị confidence nhỏ nhất được chỉ định bởi người dùng.

6.1. 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 minimum support threshold.
    - Cho A là một itemset
      - A là frequent itemset iff support(A) >= minimum support threshold.
  - 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 strong association rule iff support(A→B) >= minimum support threshold và confidence(A→B) >= minimum confidence threshold.

6.1. Tổng quan về khai phá luật kết hợp

- Phân loại luật kết hợp
  - Boolean association rule (luật kết hợp luận lý)/quantitative association rule (luật kết hợp lượng số)
  - Single-dimensional association rule (luật kết hợp đơn chiều)/multidimensional association rule (luật kết hợp đa chiều)
  - Single-level association rule (luật kết hợp đơn mức)/multilevel association rule (luật kết hợp đa mức)
  - Association rule (luật kết hợp)/correlation rule (luật tương quan thống kê)

6.1. Tổng quan về khai phá luật kết hợp

- Phân loại luật kết hợp
  - Boolean association rule (luật kết hợp luận lý)/quantitative association rule (luật kết hợp lượng số)
    - Boolean association rule: luật mô tả sự kết hợp giữa sự hiện diện/vắng mặt của các phần tử.
      - Computer → Financial_management_software [support=2%, confidence=60%]
    - Quantitative association rule: luật mô tả sự kết hợp giữa các phần tử/thuộc tính định lượng.
      - Age(X, "30..39") ∧ Income(X, "42K..48K") → buys(X, high resolution TV)

6.1. Tổng quan về khai phá luật kết hợp

- Phân loại luật kết hợp
  - Single-dimensional association rule (luật kết hợp đơn chiều)/multidimensional association rule (luật kết hợp đa chiều)
    - Single-dimensional association rule: luật chỉ liên quan đến các phần tử/thuộc tính của một chiều dữ liệu.
      - Buys(X, “computer”) → Buys(X, “financial_management_software”)
    - Multidimensional association rule: luật liên quan đến các phần tử/thuộc tính của nhiều hơn một chiều.
      - Age(X, “30..39”) → Buys(X, “computer”)

6.1. Tổng quan về khai phá luật kết hợp

- Phân loại luật kết hợp
  - Single-level association rule (luật kết hợp đơn mức) /multilevel association rule (luật kết hợp đa mức)
    - Single-level association rule: luật chỉ liên quan đến các phần tử/thuộc tính ở một mức trừu tượng.
      - Age(X, “30..39”) → Buys(X, “computer”)
      - Age(X, “18..29”) → Buys(X, “camera”)
    - Multilevel association rule: luật liên quan đến các phần tử/thuộc tính ở các mức trừu tượng khác nhau.
      - Age(X, “30..39”) → Buys(X, “laptop computer”)
      - Age(X, “30..39”) → Buys(X, “computer”)

6.1. Tổng quan về khai phá luật kết hợp

- Phân loại luật kết hợp
  - Association rule (luật kết hợp)/correlation rule (luật tương quan thống kê)
    - Association rule: strong association rules A→B (association rules đáp ứng yêu cầu minimum support threshold và minimum confidence threshold).
    - Correlation rule: strong association rules A → B đáp ứng yêu cầu về sự tương quan thống kê giữa A và B.

6.2. Biểu diễn luật kết hợp

- Dạng luật: A→B [support, confidence]
  - Cho trước minimum support threshold (min_sup), minimum confidence threshold (min_conf)
  - A và B là các itemsets
    - Frequent itemsets/subsequences/substructures
    - Closed frequent itemsets
    - Maximal frequent itemsets
    - Constrained frequent itemsets
    - Approximate frequent itemsets
    - Top-k frequent itemsets

6.2. Biểu diễn luật kết hợp

- Frequent itemsets/subsequences/substructures
  - Itemset/subsequence/substructure X là frequent nếu support(X) >= min_sup.
    - Itemsets: tập các items
    - Subsequences: chuỗi tuần tự các events/items
    - Substructures: các tiểu cấu trúc (graph, lattice, tree, sequence, set, …)

6.2. Biểu diễn luật kết hợp

- Closed frequent itemsets
  - Một itemset X closed trong *J* nếu không tồn tại tập cha thực sự Y nào trong *J* có cùng support với X.
    - $X \subseteq J$, X closed iff $\forall$ $Y \subseteq J$ và $X \subset Y$: support(Y) <> support (X).
  - X là closed frequent itemset trong *J* nếu X là frequent itemset và closed trong *J*.
- Maximal frequent itemsets
  - Một itemset X là maximal frequent itemset trong *J* nếu không tồn tại tập cha thực sự Y nào trong *J* là một frequent itemset.
    - $X \subseteq J$, X là maximal frequent itemset iff $\forall$ $Y \subseteq J$ và $X \subset Y$: Y không phải là một frequent itemset.

6.2. Biểu diễn luật kết hợp

- Constrained frequent itemsets
  - Frequent itemsets thỏa các ràng buộc do người dùng định nghĩa.
- Approximate frequent itemsets
  - Frequent itemsets dẫn ra support (xấp xỉ) cho các frequent itemsets sẽ được khai phá.
- Top-k frequent itemsets
  - Frequent itemsets có nhiều nhất k phần tử với k do người dùng chỉ định.

6.2. Biểu diễn luật kết hợp

- Luật kết hợp luận lý, đơn mức, đơn chiều giữa các tập phần tử phổ biến: A→B [support, confidence]
  - A và B là các frequent itemsets
    - single-dimensional
    - single-level
    - Boolean
  - Support(A→B) = Support(A U B) >= min_sup
  - Confidence(A→B) = Support(A U B)/Support(A) = P(B|A) >= min_conf

6.3. Khám phá các mẫu thường xuyên

- Giải thuật Apriori: khám phá các mẫu thường xuyên với tập dự tuyển
  - R. Agrawal, R. Srikant. Fast algorithms for mining association rules. In VLDB 1994, pp. 487-499.
- Giải thuật FP-Growth: khám phá các mẫu thường xuyên với FP-tree
  - J. Han, J. Pei, Y. Yin. Mining frequent patterns without candidate generation. In MOD 2000, pp. 1-12.

6.3. Khám phá các mẫu thường xuyên

- Giải thuật Apriori
  - Dùng tri thức biết trước (prior knowledge) về đặc điểm của các frequent itemsets
  - Tiếp cận lặp với quá trình tìm kiếm các frequent itemsets ở từng mức một (level-wise search)
    - k+1-itemsets được tạo ra từ k-itemsets.
    - Ở mỗi mức tìm kiếm, toàn bộ dữ liệu đều được kiểm tra.
  - Apriori property để giảm không gian tìm kiếm: All nonempty subsets of a frequent itemset must also be frequent.
    - Chứng minh???
    - Antimonotone: if a set cannot pass a test, all of its supersets will fail the same test as well.

6.3. Khám phá các mẫu thường xuyên

- 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;
```

6.3. Khám phá các mẫu thường xuyên

- 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;
```

6.3. Khám phá các mẫu thường xuyên

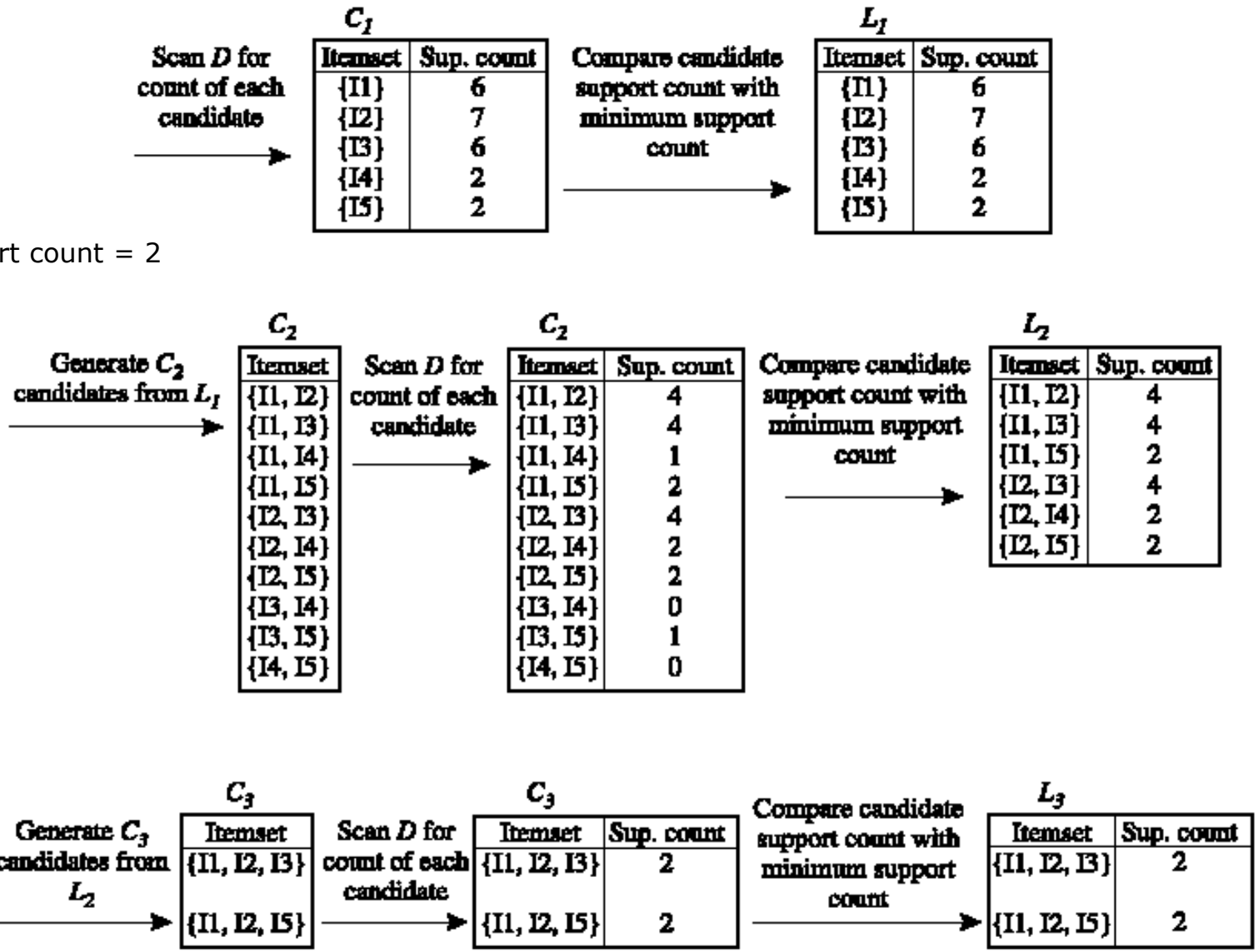
- Dữ liệu mẫu của AllElectronics (sau quá trình tiền xử lý)

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

6.3. Khám phá các mẫu thường xuyên

min_sup = 2/9

minimum support count = 2



6.3. Khám phá các mẫu thường xuyên

- Giải thuật Apriori
  - Đặc điểm
    - Tạo ra nhiều tập dự tuyển
      - 10^4 frequent 1-itemsets → nhiều hơn 10^7 ($\approx 10^4(10^4-1)/2$) 2-itemsets 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.

6.3. Khám phá các mẫu thường xuyên

- Giải thuật Apriori
  - Các cải tiến của giải thuật Apriori
    - Kỹ thuật dựa trên bảng băm (hash-based technique)
      - Một k-itemset ứng với hashing bucket count nhỏ hơn minimum support threshold không là một frequent itemset.
    - Giảm giao dịch (transaction reduction)
      - Một giao dịch không chứa frequent k-itemset nào thì không cần được kiểm tra ở các lần sau (cho k+1-itemset).
    - Phân hoạch (partitioning)
      - Một itemset phải frequent trong ít nhất một phân hoạch thì mới có thể frequent trong toàn bộ tập dữ liệu.
    - Lấy mẫu (sampling)
      - Khai phá chỉ tập con dữ liệu cho trước với một trị support threshold nhỏ hơn và cần một phương pháp để xác định tính toàn diện (completeness).
    - Đếm itemset động (dynamic itemset counting)
      - Chỉ thêm các itemsets dự tuyển khi tất cả các tập con của chúng được dự đoán là frequent.

6.3. Khám phá các mẫu thường xuyên

- 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
      - Infrequent items 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 frequent itemsets với FP-tree
  - Tránh tạo ra các tập dự tuyển
    - Mỗi lần kiểm tra một phần tập dữ liệu

6.3. Khám phá các mẫu thường xuyên

- 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 (frequency, tần số 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.

6.3. Khám phá các mẫu thường xuyên

- Giải thuật FP-Growth

Input:

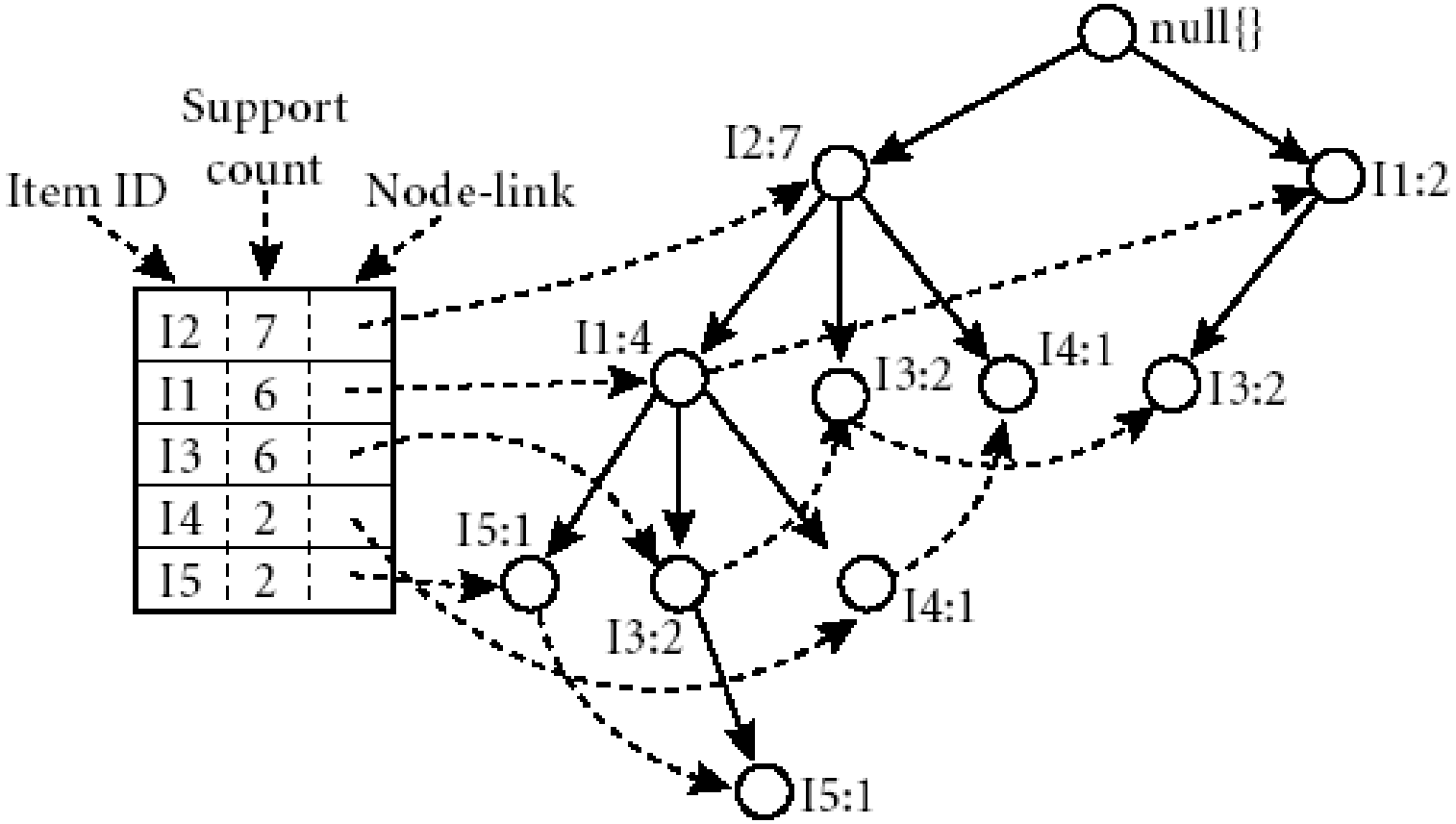
- D, a transaction database;
- *min_sup*, the minimum support count threshold.

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.

6.3. Khám phá các mẫu thường xuyên



6.3. Khám phá các mẫu thường xuyên

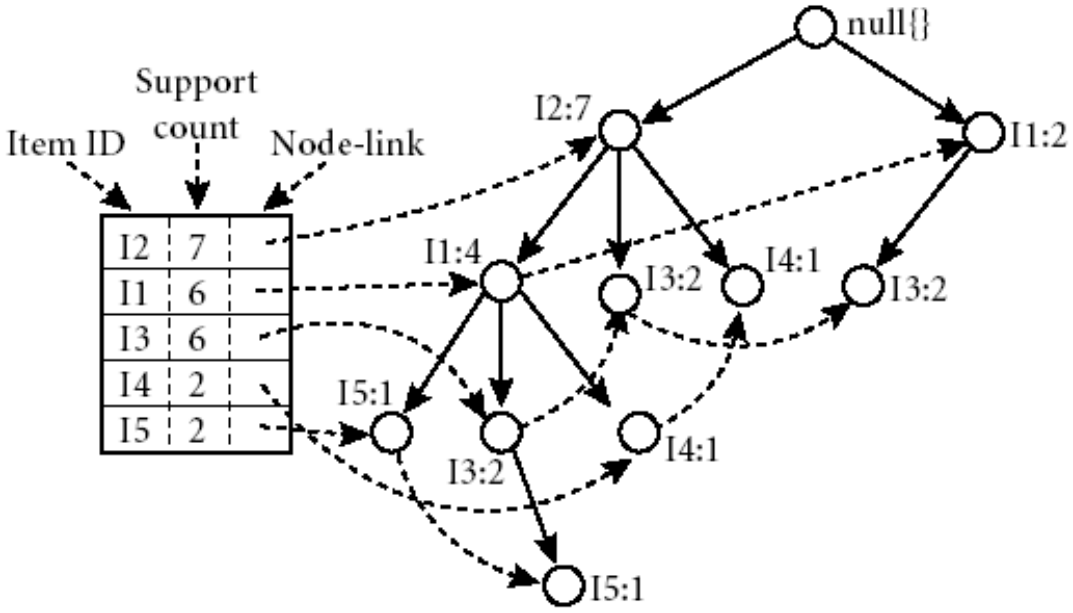
- Giải thuật FP-Growth
  - 2. Khám phá frequent itemsets với FP-tree
    - 2.1. Tạo conditional pattern base cho mỗi node của FP-tree
      - Tích luỹ các prefix paths with frequency của node đó
    - 2.2. Tạo conditional FP-tree từ mỗi conditional pattern base
      - Tích lũy frequency cho mỗi item trong mỗi base
      - Xây dựng conditional FP-tree cho frequent items của base đó
    - 2.3. Khám phá conditional FP-tree và phát triển frequent itemsets một cách đệ qui
      - Nếu conditional FP-tree có một path đơn thì liệt kê tất cả các itemsets.

6.3. Khám phá các mẫu thường xuyên

- 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); }
```

6.3. Khám phá các mẫu thường xuyên



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}

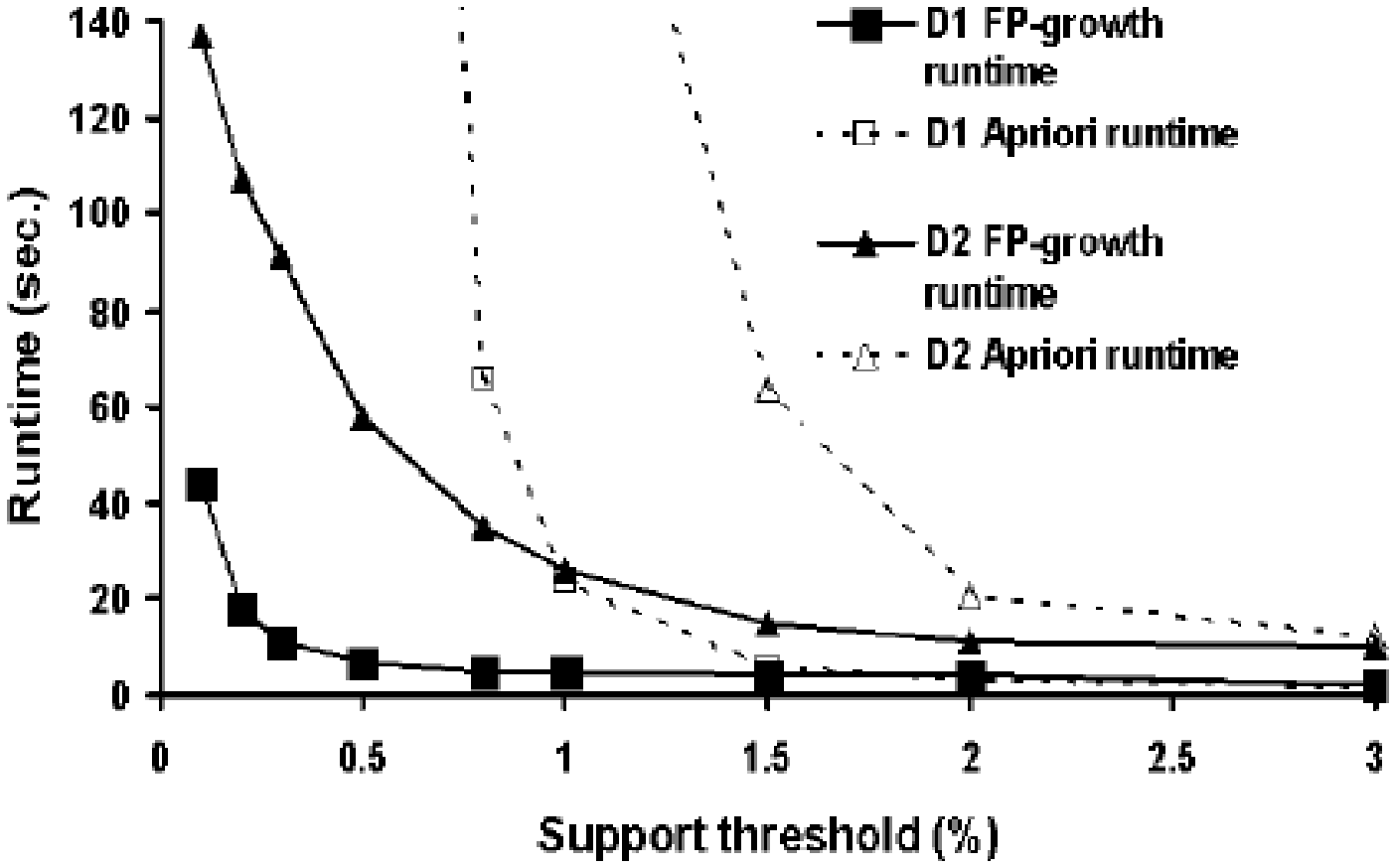
6.3. Khám phá các mẫu thường xuyên

- 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

6.3. Khám phá các mẫu thường xuyên

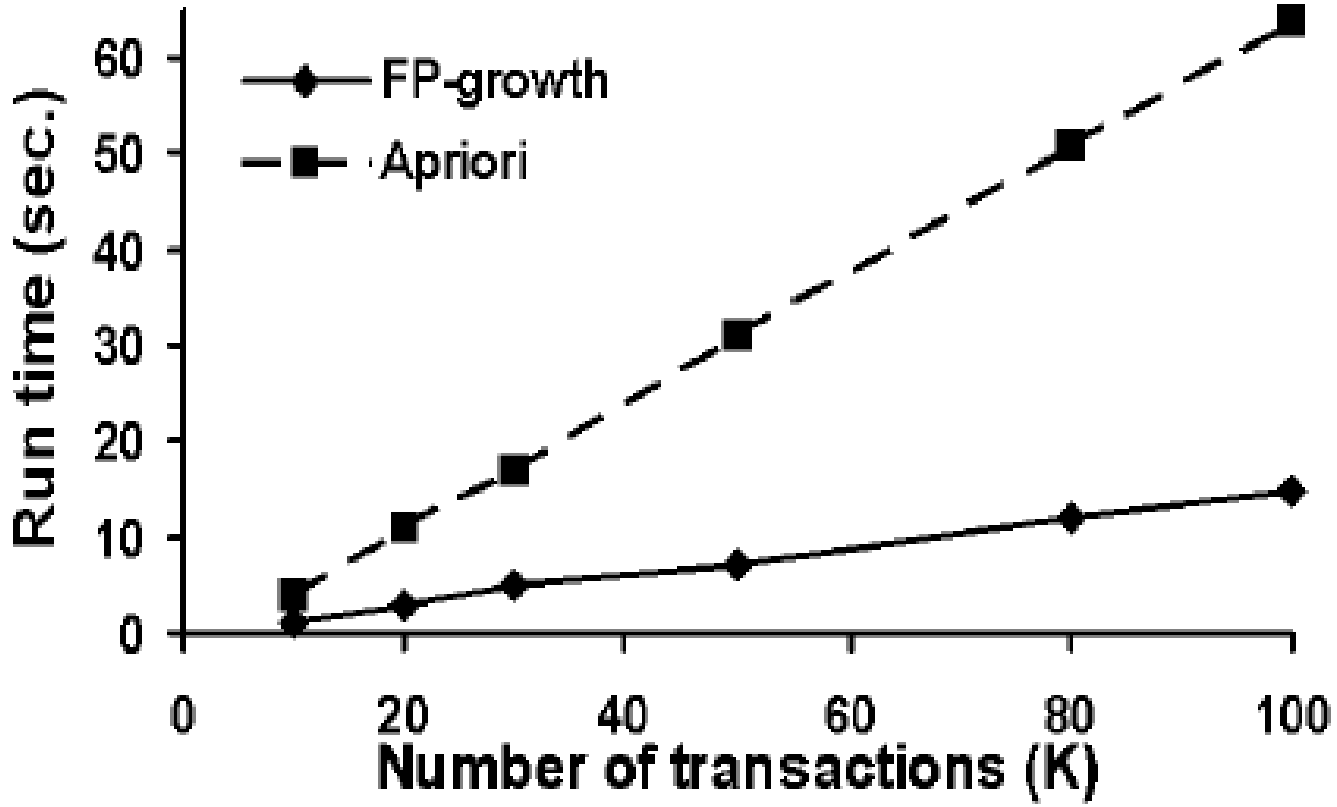
- So sánh giữa giải thuật Apriori và giải thuật FP-Growth



Co giãn với support threshold

6.3. Khám phá các mẫu thường xuyên

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



Co giãn tuyến tính với số giao dịch

6.4. Khám phá các luật kết hợp từ các mẫu thường xuyên

- Strong association rules A→B
  - Support(A→B) = Support(A U B) >= min_sup
  - Confidence(A→B) = Support(A U B)/Support(A) = P(B|A) >= min_conf
  - → Support(A→B) = Support_count(A U B) >= min_sup
  - → Confidence(A→B) = P(B|A) = Support_count(AUB)/Support_count(A) >= min_conf

6.4. Khám phá các luật kết hợp từ các mẫu thường xuyên

- Quá trình tạo các strong association rules từ tập các frequent itemsets
  - Cho mỗi frequent itemset *l*, tạo các tập con không rỗng của *l*.
    - Support_count(*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

6.4. Khám phá các luật kết hợp từ các mẫu thường xuyên

Frequent Patterns Generated
{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
{I2, I4: 2}
{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
{I2, I1: 4}

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

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%

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

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Ràng buộc (constraints)
  - Hướng dẫn quá trình khai phá mẫu (patterns) và luật (rules)
  - Giới hạn không gian tìm kiếm dữ liệu trong quá trình khai phá
  - Các dạng ràng buộc
    - Ràng buộc kiểu tri thức (knowledge type constraints)
    - Ràng buộc dữ liệu (data constraints)
    - Ràng buộc mức/chiều (level/dimension constraints)
    - Ràng buộc liên quan đến độ đo (interestingness constraints)
    - Ràng buộc liên quan đến luật (rule constraints)

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Ràng buộc kiểu tri thức (knowledge type constraints)
  - Luật kết hợp/tương quan
- Ràng buộc dữ liệu (data constraints)
  - Task-relevant data (association rule mining)
- Ràng buộc mức/chiều (level/dimension constraints)
  - Chiều (thuộc tính) dữ liệu hay mức trừu tượng/ý niệm
- Ràng buộc liên quan đến độ đo (interestingness constraints)
  - Ngưỡng của các độ đo (thresholds)
- Ràng buộc liên quan đến luật (rule constraints)
  - Dạng luật sẽ được khám phá

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Khám phá luật dựa trên ràng buộc
  - Quá trình khai phá dữ liệu tốt hơn và hiệu quả hơn (more effective and efficient).
    - Luật được khám phá dựa trên các yêu cầu (ràng buộc) của người sử dụng.
      - More effective
    - Bộ tối ưu hóa (optimizer) có thể được dùng để khai thác các ràng buộc của người sử dụng.
      - More efficient

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Khám phá luật dựa trên ràng buộc liên quan đến luật (rule constraints)
  - Dạng luật (meta-rule guided mining)
    - Metarules: chỉ định dạng luật (về cú pháp – syntactic) mong muốn được khám phá
  - Nội dung luật (rule content)
    - Ràng buộc giữa các biến trong A và/hoặc B trong luật A → B
      - Quan hệ tập hợp cha/con
      - Miền trị
      - Các hàm kết hợp (aggregate functions)

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Metarules
  - Chỉ định dạng luật (về cú pháp – syntactic) mong muốn được khám phá
  - Dựa trên kinh nghiệm, mong đợi và trực giác của nhà phân tích dữ liệu
  - Tạo nên giả thuyết (hypothesis) về các mối quan hệ (relationships) trong các luật mà người dùng quan tâm

  → Quá trình khám phá luật kết hợp + quá trình tìm kiếm luật trùng với metarules cho trước

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Metarules
  - Mẫu luật (rule template): $P_1 \wedge P_2 \wedge \ldots \wedge P_l \Rightarrow Q_1 \wedge Q_2 \wedge \ldots \wedge Q_r$
    - $P_1, P_2, \ldots, P_l, Q_1, Q_2, \ldots, Q_r$: vị từ cụ thể (instantiated predicates) hay biến vị từ (predicate variables)
    - Thường liên quan đến nhiều chiều/thuộc tính
  - Ví dụ của metarules
    - Metarule

      $P_1(X, Y) \wedge P_2(X, W) \Rightarrow buys(X, \text{“office software”})$
    - Luật thỏa metarule

      $age(X, \text{“30..39”}) \wedge income(X, \text{“41k..60k”}) \Rightarrow buys(X, \text{“office software”})$

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Ràng buộc giữa các biến S1, S2, … trong A và/hoặc B trong luật A → B
  - Quan hệ tập hợp cha/con: S1 $\subseteq/\subset$ S2
  - Miền trị
    - S1 θ value, $\theta \in$ {=, <>, <, <=, >, >=}
    - value $\in/\notin$ S1
    - ValueSet θ S1 hoặc S1 θ ValueSet, $\theta \in$ {=, <>, $\subseteq$, $\subset$, $\not\subset$}
  - Các hàm kết hợp (aggregate functions)
    - Agg(S1) θ value, Agg() $\in$ {min, max, sum, count, avg}, $\theta \in$ {=, <>, <, <=, >, >=}

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Tính chất của các ràng buộc
  - Anti-monotone
  - Monotone
  - Succinctness
  - Convertible

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Tính chất của các ràng buộc
  - Anti-monotone
    - "A constraint C_a is anti-monotone iff. for any pattern S not satisfying C_a, none of the super-patterns of S can satisfy C_a".
    - Ví dụ: sum(S.Price) <= value
  - Monotone
    - "A constraint C_m is monotone iff. for any pattern S satisfying C_m, every super-pattern of S also satisfies it".
    - Ví dụ: sum(S.Price) >= value

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Tính chất của các ràng buộc
  - Succinctness
    - "A subset of item I_s is a succinct set, if it can be expressed as $\sigma_p(I)$ for some selection predicate p, where σ is a selection operator".
    - "$SP \subseteq 2^I$ is a succinct power set, if there is a fixed number of succinct set I_1, …, $I_k \subseteq I$, s.t. SP can be expressed in terms of the strict power sets of I_1, …, I_k using union and minus".
    - "A constraint C_s is succinct provided $SAT_{Cs}(I)$ is a succinct power set".

    → Có thể tạo tường minh và chính xác các tập thỏa succinct constraints.
    - Ví dụ: min(S.Price) <= value

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Tính chất của các ràng buộc
  - Convertible
    - Các ràng buộc không có các tính chất anti-monotone, monotone, và succinctness
    - Các ràng buộc hoặc là anti-monotone hoặc là monotone nếu các phần tử trong itemset đang kiểm tra có thứ tự.
    - Ví dụ:
      - Nếu các phần tử sắp theo thứ tự tăng dần thì avg(I.price) <= 100 là một convertible anti-monotone constraint.
      - Nếu các phần tử sắp theo thứ tự giảm dần thì avg(I.price) <= 100 là một convertible monotone constraint.

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

Constraint	Antimonotonic	Monotonic	Succinct
$v \in S$	no	yes	yes
$S \supseteq V$	no	yes	yes
$S \subseteq V$	yes	no	yes
$min(S) \leq v$	no	yes	yes
$min(S) \geq v$	yes	no	yes
$max(S) \leq v$	yes	no	yes
$max(S) \geq v$	no	yes	yes
$count(S) \leq v$	yes	no	weakly
$count(S) \geq v$	no	yes	weakly
$sum(S) \leq v\ (\forall a \in S, a \geq 0)$	yes	no	no
$sum(S) \geq v\ (\forall a \in S, a \geq 0)$	no	yes	no
$range(S) \leq v$	yes	no	no
$range(S) \geq v$	no	yes	no
$avg(S)\ \theta\ v, \theta \in \{\leq, \geq\}$	convertible	convertible	no
$support(S) \geq \xi$	yes	no	no
$support(S) \leq \xi$	no	yes	no
$all\_confidence(S) \geq \xi$	yes	no	no
$all\_confidence(S) \leq \xi$	no	yes	no

6.5. Khám phá các luật kết hợp dựa trên ràng buộc

- Khám phá luật (rules)/tập phần tử phổ biến (frequent itemsets) thỏa các ràng buộc
  - Cách tiếp cận trực tiếp
    - Áp dụng các giải thuật truyền thống
    - Kiểm tra các ràng buộc cho từng kết quả đạt được
      - Nếu thỏa ràng buộc thì trả về kết quả sau cùng.
  - Cách tiếp cận dựa trên tính chất của các ràng buộc
    - Phân tích toàn diện các tính chất của các ràng buộc
    - Kiểm tra các ràng buộc càng sớm càng tốt trong quá trình khám phá rules/frequent itemsets
      - Không gian dữ liệu được thu hẹp càng sớm càng tốt.

6.6. Phân tích tương quan

- Strong association rules A $\Rightarrow$ B
  - Dựa trên tần số xuất hiện của A và B (*min_sup*)
  - Dựa trên xác suất có điều kiện của B đối với A (*min_conf*)
  - → Các độ đo *support* và *confidence* dựa vào sự chủ quan của người sử dụng
    - → Lượng rất lớn luật kết hợp có thể được trả về.
  - → Trong số 10,000 giao dịch, 6,000 giao dịch cho *computer games*, 7,500 cho *videos*, và 4,000 cho cả *computer games* và *videos*
    - → Buys(X, "*computer games*") $\Rightarrow$ Buys (X, "*videos*")

6.6. Phân tích tương quan

- Phân tích tương quan cho luật kết hợp A ⇒ B
  - Kiểm tra sự tương quan và phụ thuộc lẫn nhau giữa A và B
  - Dựa vào thống kê về dữ liệu
  - Các độ đo khách quan, không phụ thuộc vào người sử dụng
  - → Trong số 10,000 giao dịch, 6,000 giao dịch cho *computer games*, 7,500 cho *videos*, và 4,000 cho cả *computer games* và *videos*
    - → Buys(X, "*computer games*") ⇒ Buys (X, "*videos*") [support = 40%, confidence = 66%]
    - → P("*videos*") = 75% > 66%: "*computer games*" và "*videos*" tương quan nghịch với nhau.

6.6. Phân tích tương quan

- Luật tương quan (correlation rules): A $\Rightarrow$ B [support, confidence, correlation]
  - correlation: độ đo đo sự tương quan giữa A và B.
    - Các độ đo correlation: *lift*, χ^2 (Chi-square), *all_confidence*, *cosine*
      - *lift*: kiểm tra sự xuất hiện độc lập giữa A và B dựa trên xác suất (khả năng)
      - χ^2 (Chi-square): kiểm tra sự độc lập giữa A và B dựa trên giá trị mong đợi và giá trị quan sát được
      - *all_confidence*: kiểm tra luật dựa trên trị support cực đại
      - *cosine*: giống *lift* tuy nhiên loại bỏ sự phụ thuộc vào tổng số giao dịch hiện có
    - → *all_confidence* và *cosine* tốt cho tập dữ liệu lớn, không phụ thuộc các giao dịch mà không chứa bất kì itemsets đang kiểm tra (null-transactions).
    - → *all_confidence* và *consine* là các độ đo null-invariant.

6.6. Phân tích tương quan

- Độ đo tương quan *lift*
  - *lift*(A, B) < 1: A tương quan nghịch với B
  - *lift*(A, B) > 1: A tương quan thuận với B
  - *lift*(A, B) = 1: A và B độc lập nhau, không có tương quan

$$lift(A,B) = \frac{P(A \cup B)}{P(A)P(B)} = P(B \mid A)/P(B) = confidence(A => B)/support(B)$$

	$game$	$\overline{game}$	Σ_{row}
$video$	4,000	3,500	7,500
$\overline{video}$	2,000	500	2,500
Σ_{col}	6,000	4,000	10,000

$$P(\{game\}) = 0.60.$$

$$P(\{video\}) = 0.75$$

$$P(\{game, video\}) = 0.40.$$

$$P(\{game, video\})/(P(\{game\}) \times P(\{video\})) = 0.40/(0.60 \times 0.75) = 0.89.$$

lift({game}=>{video}) = 0.89 < 1 → {game} và {video} tương quan nghịch.

6.7. Tóm tắt

- Khai phá luật kết hợp
  - Được xem như là một trong những đóng góp quan trọng nhất từ cộng đồng cơ sở dữ liệu trong việc khám phá tri thức
- Các dạng luật: luật kết hợp luận lý/luật kết hợp lượng số, luật kết hợp đơn chiều/luật kết hợp đa chiều, luật kết hợp đơn mức/luật kết hợp đa mức, luật kết hợp/luật tương quan thống kê
- Các dạng phần tử (item)/mẫu (pattern): Frequent itemsets/subsequences/substructures, Closed frequent itemsets, Maximal frequent itemsets, Constrained frequent itemsets, Approximate frequent itemsets, Top-k frequent itemsets
- Khám phá các frequent itemsets: giải thuật Apriori và giải thuật FP-Growth dùng FP-tree