KHAI PHÁ DỮ LIỆU – CHƯƠNG 6
LUẬT KẾT HỢP VÀ THUẬT TOÁN APRIORI

Tổng hợp kiến thức, câu hỏi tự luận và trắc nghiệm

1. TỔNG QUAN VỀ KHAI PHÁ LUẬT KẾT HỢP

1.1. Tình huống ứng dụng

Phân tích giỏ hàng (Basket Analysis): Xác định các sản phẩm thường được mua cùng nhau trong một giao dịch. Ví dụ: khách hàng mua bia thường mua tã lót cùng lúc vào cuối tuần.

Hệ thống gợi ý (Recommendation): Dựa trên lịch sử mua hàng hoặc hành vi của người dùng tương tự để đề xuất sản phẩm/nội dung phù hợp.

Các ứng dụng khác:

- Thiết kế catalog và bố trí hàng hóa trong siêu thị.
- Cross-marketing – khuyến mãi chéo giữa các sản phẩm.
- Phân loại và phân cụm dựa trên các mẫu phổ biến (frequent patterns).
- Phân tích rủi ro trong tài chính, y tế, an ninh mạng.

1.2. Quy trình khai phá luật kết hợp

1. **Tiền xử lý (Pre-processing):** Làm sạch dữ liệu, chuyển đổi về dạng giao dịch.
2. **Khai phá (Mining):** Tìm các tập phổ biến và sinh luật kết hợp.
3. **Hậu xử lý (Post-processing):** Lọc, đánh giá và trình bày kết quả cho người dùng.

2. CÁC KHÁI NIỆM CƠ BẢN

2.1. Định nghĩa nền tảng

- **Item (mục):** Một đối tượng quan tâm, ví dụ: một sản phẩm, một từ khóa.
- **Tập items (Itemset):** Tập hợp gồm một hoặc nhiều item. **k-itemset** là itemset có đúng k item.
- $J = \{I_1, I_2, \ldots, I_m\}$**:** Tập tất cả m item trong tập dữ liệu.

- **Giao dịch (Transaction):** Một bản ghi trong tập dữ liệu giao dịch; là một tập các item thuộc cùng một giao dịch. Mỗi giao dịch T thỏa $T \subseteq J$.
- **Tập dữ liệu giao dịch D:** Tập hợp tất cả các giao dịch.

2.2. Kết hợp và luật kết hợp

- **Kết hợp (Association):** Sự xuất hiện đồng thời của các item trong cùng một giao dịch.
- **Luật kết hợp (Association rule):** $A \Rightarrow B$, với A và B là các itemset ($A \cap B = \emptyset$), biểu diễn rằng khi A xuất hiện thì B có xu hướng xuất hiện.

2.3. Support (Độ phổ biến)

Support đo lường tần suất xuất hiện của một itemset trong tập dữ liệu:

$$\text{support}(A) = \frac{|\{T \in D \mid A \subseteq T\}|}{|D|}$$

Với luật $A \Rightarrow B$:

$$\text{support}(A \Rightarrow B) = \text{support}(A \cup B) = \frac{|\{T \in D \mid A \cup B \subseteq T\}|}{|D|}$$

Ngưỡng support tối thiểu (MinSup): giá trị support tối thiểu do người dùng định nghĩa để xác định tập phổ biến.

2.4. Confidence (Độ tin cậy)

Confidence đo lường xác suất có điều kiện – tần suất B xuất hiện khi A đã xuất hiện:

$$\text{confidence}(A \Rightarrow B) = P(B \mid A) = \frac{\text{support}(A \cup B)}{\text{support}(A)}$$

Ngưỡng confidence tối thiểu (MinConf): giá trị confidence tối thiểu do người dùng định nghĩa.

2.5. Tập phổ biến và luật kết hợp mạnh

- **Tập phổ biến (Frequent itemset):** Itemset A thỏa $\text{support}(A) \geq \text{MinSup}$.

- **Luật kết hợp mạnh (Strong association rule):** Luật $A \Rightarrow B$ thỏa đồng thời:

$$\text{support}(A \Rightarrow B) \geq \text{MinSup} \quad \text{và} \quad \text{confidence}(A \Rightarrow B) \geq \text{MinConf}$$

2.6. Ví dụ minh họa

Xét tập dữ liệu 9 giao dịch, với các item $\{I1, I2, I3, I4, I5\}$. Nếu MinSup $= 2/9$ và itemset $\{I1, I2\}$ xuất hiện trong 4 giao dịch:

$$\text{support}(\{I1, I2\}) = \frac{4}{9} \approx 44\% \geq \text{MinSup}$$

Luật $I1 \Rightarrow I2$ có confidence:

$$\text{confidence}(I1 \Rightarrow I2) = \frac{\text{support}(\{I1, I2\})}{\text{support}(\{I1\})} = \frac{4/9}{6/9} = \frac{4}{6} \approx 66.7\%$$

3. PHÂN LOẠI LUẬT KẾT HỢP

- **Boolean vs. Quantitative:**
  - **Boolean:** Biểu diễn sự xuất hiện/vắng mặt của item. Ví dụ: *Computer* $\Rightarrow$ *Financial_SW* [sup=50%, conf=60%].
  - **Quantitative:** Liên quan đến các thuộc tính định lượng. Ví dụ: *Age(X,"30..39")* $\Rightarrow$ *Buys(X, "TV")* [sup=50%, conf=60%].
- **Single-dimensional vs. Multidimensional:**
  - **Single-dimensional:** Chỉ liên quan đến một chiều dữ liệu (ví dụ: *buys*).
  - **Multidimensional:** Liên quan đến nhiều chiều (ví dụ: *age, income, buys*).
- **Single-level vs. Multilevel:**
  - **Single-level:** Tất cả item ở cùng mức trừu tượng.
  - **Multilevel:** Item ở nhiều mức trừu tượng khác nhau (ví dụ: *laptop* và *computer* là hai mức của cùng một khái niệm).
- **Association rule vs. Correlation rule:** Luật kết hợp chỉ dựa trên support/confidence; luật tương quan bổ sung thêm các độ đo thống kê (lift, χ^2, ...).

4. BIỂU DIỄN LUẬT KẾT HỢP

Dạng chuẩn của luật kết hợp:

$$A \Rightarrow B \quad [\text{support} = s\%,\ \text{confidence} = c\%]$$

trong đó:

- A, B là các tập phổ biến (frequent itemsets), $A \cap B = \emptyset$.
- $\text{support}(A \Rightarrow B) = \text{support}(A \cup B) \geq \text{MinSup}$.
- $\text{confidence}(A \Rightarrow B) = \dfrac{\text{support}(A \cup B)}{\text{support}(A)} \geq \text{MinConf}$.

5. KHAI PHÁ TẬP PHỔ BIẾN – HAI BƯỚC

Khai phá luật kết hợp gồm **hai bước** chính:

1. **Bước 1 – Tìm tập phổ biến:** Tìm tất cả itemset có support $\geq$ MinSup. Đây là bước tốn kém nhất về tính toán.
2. **Bước 2 – Sinh luật kết hợp:** Từ mỗi tập phổ biến L, sinh tất cả các luật $A \Rightarrow (L \setminus A)$ thỏa MinConf.

6. THUẬT TOÁN APRIORI

6.1. Ý tưởng và tính chất Apriori

Apriori (Agrawal & Srikant, VLDB 1994) khai phá tập phổ biến bằng cách tận dụng **tính chất Apriori (Apriori property)**:

"Mọi tập con của một tập phổ biến đều là tập phổ biến."

Hệ quả (Anti-monotone property): Nếu X **không** phải tập phổ biến thì $X \cup Y$ (với Y bất kỳ) cũng **không** phải tập phổ biến. Tính chất này cho phép cắt tỉa không gian tìm kiếm hiệu quả.

6.2. Pseudo-code của Apriori

```
Ck: tập ứng viên k-itemset
Lk: tập phổ biến k-itemset
```

```
L1 = {frequent 1-itemsets};
k = 1;
while (Lk != empty) {
    k++;
    Ck = apriori_gen(Lk-1);      // sinh ứng viên từ Lk-1
    for each transaction t in D {
        Ct = subset(Ck, t);       // ứng viên chứa trong t
        for each candidate c in Ct
            c.count++;            // tăng bộ đếm
    }
    Lk = {c in Ck | c.count >= min_sup_count};
}
return Union(Lk);
```

6.3. Sinh tập ứng viên (apriori_gen)

Gồm hai bước:

Bước 1 – Self-joining (tự nối): Từ L_{k-1}, nối hai itemset có $k-2$ item đầu giống nhau:

```
INSERT INTO Ck
SELECT p.item1, ..., p.itemk-1, q.itemk-1
FROM   Lk-1 p, Lk-1 q
WHERE  p.item1=q.item1, ..., p.itemk-2=q.itemk-2,
       p.itemk-1 < q.itemk-1
```

Bước 2 – Pruning (cắt tỉa): Loại bỏ ứng viên $c \in C_k$ nếu bất kỳ $(k-1)$-subset nào của c không có trong L_{k-1}.

6.4. Ví dụ minh họa Apriori

Tập dữ liệu D gồm 4 giao dịch, MinSup = 2:

TID	Items
100	1, 3, 4
200	2, 3, 5
300	1, 2, 3, 5
400	2, 5

Vòng lặp 1: $C_1 = \{\{1\},\{2\},\{3\},\{4\},\{5\}\}$. Sau khi quét D: $L_1 = \{\{1\}^2,\{2\}^3,\{3\}^3,\{5\}^3\}$ (loại $\{4\}$ vì sup=1 < 2).

Vòng lặp 2: Tự nối L_1 sinh C_2. Sau khi quét: $L_2 = \{\{1,3\}^2, \{2,3\}^2, \{2,5\}^3, \{3,5\}^2\}$.

Vòng lặp 3: Sinh $C_3 = \{\{2,3,5\}\}$ (các ứng viên khác bị cắt tỉa). Sau quét: $L_3 = \{\{2,3,5\}^2\}$.

Vòng lặp 4: $C_4 = \emptyset \Rightarrow$ dừng.

6.5. Đặc điểm và hạn chế của Apriori

- **Ưu điểm:** Dễ hiểu, tận dụng tính chất Apriori để cắt tỉa không gian tìm kiếm.
- **Hạn chế:**
  - Quét toàn bộ tập dữ liệu nhiều lần (mỗi cấp độ k quét một lần).
  - Sinh ra số lượng lớn tập ứng viên, tốn bộ nhớ.
  - Không hiệu quả với tập dữ liệu lớn hoặc MinSup thấp.

7. THUẬT TOÁN FP-GROWTH

FP-Growth (Han, Pei, Yin – SIGMOD 2000) khai phá tập phổ biến **không sinh tập ứng viên**, sử dụng cấu trúc **FP-tree (Frequent Pattern tree)**.

Ý tưởng chính:

1. **Xây dựng FP-tree:** Nén tập dữ liệu vào một cây tiền tố (prefix tree). Chỉ cần quét dữ liệu **2 lần**.
2. **Khai phá FP-tree:** Phân tích từng nhánh của cây theo phương pháp "divide and conquer" để tìm tập phổ biến mà không sinh ứng viên.

Ưu điểm so với Apriori:

- Không sinh tập ứng viên $\Rightarrow$ giảm đáng kể bộ nhớ.
- Chỉ quét dữ liệu 2 lần $\Rightarrow$ hiệu quả hơn với tập dữ liệu lớn.
- Hiệu quả với MinSup thấp.

8. KHAI PHÁ LUẬT KẾT HỢP TỪ TẬP PHỔ BIẾN

Từ mỗi tập phổ biến L (với $|L| \geq 2$), sinh tất cả các luật $A \Rightarrow (L \setminus A)$ $(A \neq \emptyset)$ và giữ lại những luật thỏa MinConf.

Tính chất đơn điệu của confidence: Với luật $A \Rightarrow (L \setminus A)$, nếu luật không thỏa MinConf thì bất kỳ luật nào có vế trái là tập con của A cũng không thỏa. Tính chất này cho phép cắt tỉa bước sinh luật.

Ví dụ: Từ $L_3 = \{2,3,5\}$ (sup = 2), sinh các luật:

- $\{2,3\} \Rightarrow \{5\}$: conf = $2/2 = 100\%$
- $\{2,5\} \Rightarrow \{3\}$: conf = $2/3 = 66.7\%$
- $\{3,5\} \Rightarrow \{2\}$: conf = $2/2 = 100\%$
- $\{2\} \Rightarrow \{3,5\}$: conf = $2/3 = 66.7\%$
- $\{3\} \Rightarrow \{2,5\}$: conf = $2/3 = 66.7\%$
- $\{5\} \Rightarrow \{2,3\}$: conf = $2/3 = 66.7\%$

9. KHAI PHÁ CÓ RÀNG BUỘC (CONSTRAINT-BASED MINING)

Trong thực tế, người dùng chỉ quan tâm đến một tập con các luật thỏa mãn điều kiện cụ thể. Các loại ràng buộc:

- **Knowledge type constraint:** Chỉ khai phá loại luật nhất định (ví dụ: chỉ luật Boolean).
- **Data constraint:** Giới hạn trên tập item cụ thể (ví dụ: chỉ sản phẩm điện tử).
- **Dimension/level constraint:** Giới hạn chiều dữ liệu hoặc mức trừu tượng.
- **Rule constraint:** Ràng buộc trực tiếp trên cấu trúc luật (ví dụ: vế trái chứa ít nhất 2 item).
- **Interestingness constraint:** Ràng buộc trên support, confidence hoặc các độ đo khác.

Ràng buộc **anti-monotone** (như support) có thể dùng để cắt tỉa sớm trong quá trình khai phá.

10. PHÂN TÍCH TƯƠNG QUAN (CORRELATION ANALYSIS)

Luật kết hợp mạnh (thỏa MinSup và MinConf) chưa chắc đã biểu diễn quan hệ nhân quả hoặc tương quan thực sự. Cần bổ sung thêm các độ đo tương quan:

10.1. Lift (Độ nâng)

$$\text{lift}(A \Rightarrow B) = \frac{\text{confidence}(A \Rightarrow B)}{\text{support}(B)} = \frac{\text{support}(A \cup B)}{\text{support}(A) \cdot \text{support}(B)}$$

- lift > 1: A và B **tương quan thuận** (positively correlated).
- lift $= 1$: A và B **độc lập** (independent).
- lift < 1: A và B **tương quan nghịch** (negatively correlated).

10.2. Độ đo χ^2 (Chi-square)

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

trong đó O là tần suất quan sát, E là tần suất kỳ vọng. Nếu $\chi^2 = 0$: hai item độc lập; $\chi^2 > 0$: có tương quan.

10.3. Cosine similarity

$$\text{cosine}(A,B) = \frac{\text{support}(A \cup B)}{\sqrt{\text{support}(A) \cdot \text{support}(B)}}$$

11. TÓM TẮT

- **Luật kết hợp** $A \Rightarrow B$ được đặc trưng bởi support và confidence.
- **Khai phá** gồm 2 bước: (1) tìm tập phổ biến; (2) sinh luật từ tập phổ biến.
- **Apriori:** Khai phá tập phổ biến dựa trên tính chất anti-monotone; quét dữ liệu nhiều lần.
- **FP-Growth:** Không sinh ứng viên, chỉ quét dữ liệu 2 lần; hiệu quả hơn Apriori.
- **Luật kết hợp mạnh** chưa đủ; cần bổ sung **lift,** χ^2 để đánh giá tương quan thực sự.
- **Ràng buộc** giúp người dùng tập trung vào các luật có ý nghĩa thực tiễn.

12. CÂU HỎI TỰ LUẬN

Câu 1. Khai phá luật kết hợp (association rule mining) là gì? Trình bày quy trình 3 bước của khai phá luật kết hợp và cho ví dụ ứng dụng trong thực tiễn.

Câu 2. Giải thích các khái niệm: **item, itemset, k-itemset, transaction, tập dữ liệu giao dịch**. Cho ví dụ minh họa với một tập dữ liệu giao dịch cụ thể.

Câu 3. Định nghĩa **support** và **confidence** của một luật kết hợp $A \Rightarrow B$. Viết công thức và giải thích ý nghĩa của từng độ đo.

Câu 4. **Tập phổ biến (frequent itemset)** và **luật kết hợp mạnh (strong association rule)** được định nghĩa như thế nào? Tại sao cần cả hai điều kiện MinSup và MinConf?

Câu 5. Trình bày **tính chất Apriori (Apriori property)** và **tính anti-monotone**. Tính chất này được sử dụng như thế nào để giảm không gian tìm kiếm trong thuật toán Apriori?

Câu 6. Mô tả chi tiết các bước của **thuật toán Apriori**. Giải thích bước sinh tập ứng viên (apriori_gen) bao gồm self-joining và pruning. Minh họa bằng ví dụ.

Câu 7. Với tập dữ liệu D gồm các giao dịch: T1={A,B,C}, T2={A,C}, T3={A,D}, T4={B,E,F}, T5={A,B,C}. Với MinSup = 2, hãy thực hiện vòng lặp đầu tiên của Apriori: tìm C_1 và L_1.

Câu 8. So sánh **Apriori** và **FP-Growth** về: số lần quét dữ liệu, việc sinh tập ứng viên, bộ nhớ sử dụng và hiệu quả khi MinSup thấp.

Câu 9. **FP-tree** là gì? Trình bày ý tưởng xây dựng FP-tree và cách FP-Growth khai phá tập phổ biến từ FP-tree mà không sinh tập ứng viên.

Câu 10. Từ một tập phổ biến $L = \{A, B, C\}$, hãy liệt kê tất cả các luật kết hợp có thể sinh ra. Giải thích cách áp dụng MinConf để lọc các luật không đủ điều kiện.

Câu 11. Phân biệt **luật kết hợp Boolean** và **luật kết hợp định lượng (quantitative)**. Cho ví dụ cụ thể cho mỗi loại.

Câu 12. Giải thích sự khác biệt giữa **luật kết hợp đơn chiều (single-dimensional)** và **đa chiều (multidimensional)**. Luật đa chiều hữu ích trong tình huống nào?

Câu 13. **Luật kết hợp đa mức (multilevel association rule)** khác với luật đơn mức như thế nào? Ưu điểm của khai phá đa mức là gì? Thách thức nào phát sinh khi khai phá đa mức?

Câu 14. Trình bày các loại ràng buộc (constraints) trong khai phá luật kết hợp dựa trên ràng buộc. Ràng buộc **anti-monotone** khác với ràng buộc thông thường như thế nào?

Câu 15. **Lift** được định nghĩa như thế nào? Giải thích ý nghĩa của lift > 1, lift $= 1$ và lift < 1. Tại sao lift quan trọng hơn chỉ dùng confidence?

Câu 16. Một luật kết hợp mạnh (thỏa MinSup và MinConf) có thể **không có ý nghĩa thực sự** không? Giải thích bằng ví dụ cụ thể và nêu cách khắc phục dùng lift hoặc χ^2.

Câu 17. Trình bày độ đo χ^2 (chi-square) trong phân tích tương quan. Khi nào $\chi^2 = 0$ và điều đó có nghĩa gì?

Câu 18. Giải thích **cosine similarity** trong đánh giá luật kết hợp. So sánh cosine similarity với lift về ưu và nhược điểm.

Câu 19. Nêu và phân tích **các hạn chế của thuật toán Apriori**. Đề xuất ít nhất hai hướng cải tiến Apriori để tăng hiệu quả.

Câu 20. Khai phá luật kết hợp gồm **hai bước**: tìm tập phổ biến và sinh luật. Tại sao bước tìm tập phổ biến được coi là bước **tốn kém nhất**? Phân tích độ phức tạp tính toán của bước này.

13. CÂU HỎI TRẮC NGHIỆM

Câu 1. Khai phá luật kết hợp thuộc loại bài toán nào trong khai phá dữ liệu?

A. Học có giám sát (supervised learning).

B. Học không giám sát (unsupervised learning).

C. Học bán giám sát (semi-supervised learning).

D. Học tăng cường (reinforcement learning).

Câu 2. Phân tích giỏ hàng (basket analysis) trong khai phá luật kết hợp nhằm mục đích:

A. Phân nhóm khách hàng theo hành vi mua hàng.

B. Phát hiện các sản phẩm thường được mua cùng nhau trong một giao dịch.

C. Dự đoán giá sản phẩm trong tương lai.

D. Phân loại khách hàng theo thu nhập.

Câu 3. Một **k-itemset** là:

A. Một giao dịch chứa đúng k item.

B. Một tập hợp gồm đúng k item.

C. Một luật có k item ở vế trái.

D. Một tập phổ biến xuất hiện ít nhất k lần.

Câu 4. Support của luật $A \Rightarrow B$ được tính bằng:

A. $P(A \mid B)$

B. $P(A \cup B)$

C. $P(B \mid A)$

D. $P(A) \cdot P(B)$

Câu 5. Confidence của luật $A \Rightarrow B$ được tính bằng:

A. $\dfrac{\text{support}(A \cup B)}{\text{support}(A)}$

B. $\dfrac{\text{support}(A)}{\text{support}(B)}$

C. $\dfrac{\text{support}(A \cup B)}{|D|}$

D. $\text{support}(A) + \text{support}(B)$

Câu 6. Một itemset được gọi là **tập phổ biến (frequent itemset)** khi:

A. support $\geq$ MinConf.

B. confidence $\geq$ MinSup.

C. support $\geq$ MinSup.

D. lift ≥ 1.

Câu 7. Một luật $A \Rightarrow B$ được gọi là **luật kết hợp mạnh** khi:

A. support $\geq$ MinSup hoặc confidence $\geq$ MinConf.

B. support $\geq$ MinSup và confidence $\geq$ MinConf.

C. lift > 1 và confidence $\geq$ MinConf.

D. support $\geq$ MinSup và lift > 1.

Câu 8. Nếu support($\{A, B\}$) = 0.4 và support($\{A\}$) = 0.5, thì confidence($A \Rightarrow B$) bằng:

A. 0.2

B. 0.4

C. 0.8

D. 0.9

Câu 9. Tính chất Apriori (Apriori property) phát biểu rằng:

A. Mọi superset của tập phổ biến đều là tập phổ biến.

B. Mọi tập con của tập phổ biến đều là tập phổ biến.

C. Mọi tập con của tập phổ biến đều không phổ biến.

D. Chỉ itemset có đúng 1 phần tử mới là tập phổ biến.

Câu 10. Tính anti-monotone của Apriori nói rằng:

A. Nếu X phổ biến thì $X \cup Y$ cũng phổ biến.

B. Nếu X không phổ biến thì $X \cup Y$ cũng không phổ biến.

C. Support tăng khi kích thước itemset tăng.

D. Confidence không thể giảm khi thêm item vào vế trái.

Câu 11. Bước **self-joining** trong sinh tập ứng viên C_k từ L_{k-1} thực hiện:

A. Loại bỏ ứng viên có subset không phổ biến.

B. Nối hai itemset trong L_{k-1} có $k-2$ item đầu giống nhau.

C. Đếm support của tất cả ứng viên trong C_k.

D. Lọc ứng viên có support $<$ MinSup.

Câu 12. Bước **pruning** trong sinh tập ứng viên C_k thực hiện:

A. Nối các itemset trong L_{k-1}.

B. Đếm support của từng ứng viên.

C. Loại bỏ ứng viên $c \in C_k$ nếu có $(k-1)$-subset của c không thuộc L_{k-1}.

D. Sinh ra tất cả $(k+1)$-itemset từ C_k.

Câu 13. Độ phức tạp của Apriori liên quan đến số lần quét tập dữ liệu là:

A. 1 lần (một lần quét duy nhất).

B. 2 lần.

C. Bằng độ dài tập phổ biến dài nhất ($k_{\max}$ lần).

D. n lần với n là số giao dịch.

Câu 14. Với $L_3 = \{abc, abd, acd, ace, bcd\}$, sau bước self-joining, tập ứng viên C_4 ban đầu (trước pruning) gồm:

A. $\{abcd\}$

B. $\{abcd, acde\}$

C. $\{abcd, abce, abde\}$

D. $\{abcde\}$

Câu 15. Sau bước pruning trên $C_4 = \{abcd, acde\}$ (với L_3 như câu 14), kết quả là:

A. $C_4 = \{abcd, acde\}$ (không đổi).

B. $C_4 = \{abcd\}$ vì $acde$ bị loại do $ade \notin L_3$.

C. $C_4 = \emptyset$ vì cả hai đều bị loại.

D. $C_4 = \{acde\}$ vì $abcd$ bị loại.

Câu 16. FP-Growth khác Apriori ở điểm quan trọng nào?

A. FP-Growth cần xác định số cụm k trước.

B. FP-Growth không sinh tập ứng viên và chỉ quét dữ liệu 2 lần.

C. FP-Growth chỉ hoạt động với dữ liệu nhị phân.

D. FP-Growth cần MinConf nhưng không cần MinSup.

Câu 17. FP-tree (Frequent Pattern tree) được sử dụng trong FP-Growth để:

A. Lưu trữ danh sách tất cả tập ứng viên.

B. Nén tập dữ liệu vào cấu trúc cây tiền tố, tránh sinh ứng viên.

C. Biểu diễn dendrogram trong phân cụm.

D. Tính toán confidence nhanh hơn.

Câu 18. Từ tập phổ biến $L = \{A, B, C\}$ (sup = 3), support($\{A, B\}$) = 3. Confidence($\{A, B\} \Rightarrow \{C\}$) bằng:

A. $3/9 = 33\%$

B. $3/3 = 100\%$

C. $3/5 = 60\%$

D. Không tính được từ thông tin đã cho.

Câu 19. Khi sinh luật từ tập phổ biến, **tính đơn điệu của confidence** cho phép:

A. Tính confidence mà không cần quét dữ liệu thêm.

B. Cắt tỉa các luật có vế trái là tập cha của một luật không đủ MinConf.

C. Tự động xác định giá trị MinConf tối ưu.

D. Đảm bảo mọi luật sinh ra đều thỏa MinConf.

Câu 20. Luật "*Age(X, "30..39")* $\Rightarrow$ *Buys(X, "TV")*" thuộc loại:

A. Luật kết hợp Boolean đơn chiều.

B. Luật kết hợp định lượng đa chiều.

C. Luật kết hợp Boolean đa mức.

D. Luật kết hợp tương quan.

Câu 21. Luật kết hợp **đa mức (multilevel)** có đặc điểm:

A. Chỉ liên quan đến một chiều dữ liệu.

B. Liên quan đến item ở nhiều mức trừu tượng khác nhau.

C. Chỉ áp dụng với dữ liệu định lượng.

D. Không cần MinSup để khai phá.

Câu 22. Ràng buộc **anti-monotone** trong khai phá có ràng buộc có nghĩa là:

A. Ràng buộc làm tăng số lượng luật tìm thấy.

B. Nếu một itemset vi phạm ràng buộc thì superset của nó cũng vi phạm, cho phép cắt tỉa sớm.

C. Ràng buộc chỉ áp dụng cho vế phải của luật.

D. Ràng buộc không ảnh hưởng đến hiệu quả khai phá.

Câu 23. **Lift** của luật $A \Rightarrow B$ được tính bằng:

A. $\dfrac{\text{support}(A \cup B)}{\text{support}(A) \cdot \text{support}(B)}$

B. $\dfrac{\text{confidence}(A \Rightarrow B)}{\text{support}(A)}$

C. $\text{support}(A \cup B) - \text{support}(A) \cdot \text{support}(B)$

D. $\dfrac{\text{support}(A)}{\text{support}(B)}$

Câu 24. Nếu $\text{lift}(A \Rightarrow B) = 1$ thì:

A. A và B tương quan thuận.

B. A và B tương quan nghịch.

C. A và B độc lập với nhau.

D. Luật $A \Rightarrow B$ không phải luật mạnh.

Câu 25. Nếu $\text{lift}(A \Rightarrow B) < 1$ thì:

A. A và B tương quan thuận.

B. A và B tương quan nghịch.

C. A và B độc lập.

D. Confidence của luật bằng 0.

Câu 26. Độ đo $\chi^2 = 0$ trong phân tích tương quan có nghĩa là:

A. Hai item hoàn toàn phụ thuộc nhau.

B. Hai item hoàn toàn độc lập với nhau.

C. Support của cả hai item bằng 0.

D. Luật không thỏa MinConf.

Câu 27. Lý do chính cần bổ sung **lift** hoặc χ^2 bên cạnh support và confidence là:

A. Để tính toán nhanh hơn.

B. Luật kết hợp mạnh có thể biểu diễn quan hệ **không có ý nghĩa thực** hoặc tương quan nghịch.

C. Để giảm số lượng tập ứng viên.

D. Để tự động chọn MinSup và MinConf tối ưu.

Câu 28. Trong Apriori, C_k ký hiệu cho:

A. Tập phổ biến k-itemset.

B. Tập ứng viên k-itemset.

C. Tập giao dịch có độ dài k.

D. Số lần quét dữ liệu thứ k.

Câu 29. Trong Apriori, L_k ký hiệu cho:

A. Tập ứng viên k-itemset.

B. Tập phổ biến k-itemset (sau khi lọc theo MinSup).

C. Tập giao dịch có ít nhất k item.

D. Tập luật kết hợp có vế trái là k-itemset.

Câu 30. Khai phá luật kết hợp gồm hai bước; bước nào tốn kém nhất về tính toán?

A. Bước 2: sinh luật từ tập phổ biến.

B. Bước 1: tìm tất cả tập phổ biến.

C. Cả hai bước có độ phức tạp như nhau.

D. Bước tiền xử lý dữ liệu.

Câu 31. MinSup càng **thấp** thì:

A. Số tập phổ biến càng ít.

B. Số tập phổ biến càng nhiều, tính toán càng tốn kém.

C. Thuật toán Apriori càng nhanh.

D. Không ảnh hưởng đến số tập phổ biến.

Câu 32. Với tập dữ liệu D (4 giao dịch), MinSup = 2. Item $\{4\}$ xuất hiện trong 1 giao dịch. Item $\{4\}$ có phải tập phổ biến không?

A. Có, vì support = $1/4 > 0$.

B. Không, vì support = $1 <$ MinSup = 2 (tính theo count).

C. Có, nếu confidence $\geq$ MinConf.

D. Không đủ thông tin để kết luận.

Câu 33. Giả sử $\{A,B\}$ không phổ biến. Theo tính chất anti-monotone, $\{A,B,C\}$:

A. Có thể phổ biến hoặc không.

B. Chắc chắn phổ biến vì có thêm item C.

C. Chắc chắn không phổ biến.

D. Phổ biến nếu $\{C\}$ phổ biến.

Câu 34. Số lượng luật có thể sinh ra từ tập phổ biến L có n item là:

A. n

B. $2^n - 2$

C. n^2

D. $n!$

Câu 35. Cosine similarity trong đánh giá luật kết hợp được tính bằng:

A. $\dfrac{\text{support}(A \cup B)}{\text{support}(A) + \text{support}(B)}$

B. $\dfrac{\text{support}(A \cup B)}{\sqrt{\text{support}(A) \cdot \text{support}(B)}}$

C. $\dfrac{\text{confidence}(A \Rightarrow B)}{\text{confidence}(B \Rightarrow A)}$

D. $\sqrt{\text{support}(A) \cdot \text{support}(B)}$

Câu 36. Trong khai phá luật kết hợp dựa trên ràng buộc, ràng buộc nào sau đây là **anti-monotone**?

A. Luật phải có ít nhất 3 item ở vế trái (min-constraint).

B. Tổng giá trị item ≤ 100 (sum-constraint).

C. Support $\geq$ MinSup.

D. Confidence $\geq$ MinConf.

Câu 37. Tại sao FP-Growth hiệu quả hơn Apriori với MinSup thấp?

A. FP-Growth không cần MinSup.

B. Khi MinSup thấp, số tập phổ biến lớn $\Rightarrow$ Apriori sinh rất nhiều ứng viên; FP-Growth tránh được điều này.

C. FP-Growth quét dữ liệu ít hơn 1 lần so với Apriori.

D. FP-Growth chỉ hoạt động khi MinSup $< 10\%$.

Câu 38. Khai phá luật kết hợp **đa mức** thường phải đối mặt với thách thức:

A. Cần chọn k cụm trước khi khai phá.

B. Cần định nghĩa MinSup khác nhau cho từng mức trừu tượng.

C. Không thể áp dụng tính chất Apriori.

D. Luôn sinh ra quá ít luật ở mức chi tiết.

ĐÁP ÁN

Câu hỏi tự luận – Hướng dẫn trả lời

Câu	Nội dung cần trình bày
1	Khai phá luật kết hợp: tìm các quan hệ "nếu...thì..." giữa các item. Quy trình: tiền xử lý $\rightarrow$ khai phá $\rightarrow$ hậu xử lý. Ví dụ: basket analysis, recommendation, cross-marketing.
2	Item: đối tượng quan tâm. Itemset: tập item. k-itemset: itemset có k phần tử. Transaction: bản ghi chứa tập item. Dataset D: tập tất cả giao dịch. Ví dụ bảng giao dịch cụ thể.
3	Support($A \Rightarrow B$) = support($A \cup B$) = $\frac{\lvert T \in D : A \cup B \subseteq T \rvert}{\lvert D \rvert}$. Confidence($A \Rightarrow B$) = $P(B\lvert A)$ = support($A \cup B$)/support(A). Support đo tần suất, confidence đo độ tin cậy điều kiện.
4	Tập phổ biến: support $\geq$ MinSup. Luật mạnh: support $\geq$ MinSup và confidence $\geq$ MinConf. Cần cả hai vì luật cần vừa phổ biến (xuất hiện đủ nhiều) vừa đáng tin cậy (vế phải xuất hiện thường xuyên khi vế trái xảy ra).
5	Apriori property: mọi tập con của tập phổ biến đều phổ biến. Anti-monotone: nếu X không phổ biến thì $X \cup Y$ không phổ biến $\Rightarrow$ cắt tỉa toàn bộ superset của X khỏi không gian tìm kiếm.
6	(1) L_1 = tập phổ biến 1-itemset. (2) Sinh C_k bằng self-join L_{k-1} + pruning. (3) Đếm support trong D. (4) Lọc thành L_k. (5) Lặp đến khi $L_k = \emptyset$. Self-join: nối 2 itemset có $k-2$ item đầu giống. Pruning: loại ứng viên có $(k-1)$-subset không trong L_{k-1}.
7	C_1 = {{A},{B},{C},{D},{E},{F}}. Đếm support: A=4, B=3, C=3, D=1, E=1, F=1. L_1 (MinSup=2) = $\{\{A\}^4, \{B\}^3, \{C\}^3\}$. Loại D, E, F.
8	Apriori: nhiều lần quét ($k_{\max}$ lần), sinh nhiều ứng viên, tốn bộ nhớ, kém hiệu quả khi MinSup thấp. FP-Growth: 2 lần quét, không sinh ứng viên, bộ nhớ hiệu quả hơn (FP-tree), tốt với MinSup thấp.
9	FP-tree là cây tiền tố nén tập dữ liệu. Xây dựng: (1) quét lần 1 tìm L_1; (2) quét lần 2 chèn giao dịch theo thứ tự giảm dần của support. FP-Growth đào conditional FP-tree theo từng item (divide-and-conquer) mà không sinh ứng viên.
10	$L = \{A, B, C\}$: các luật = $A \Rightarrow BC$, $B \Rightarrow AC$, $C \Rightarrow AB$, $AB \Rightarrow C$, $AC \Rightarrow B$, $BC \Rightarrow A$. Giữ luật nào có confidence = support(L)/support(vế trái) $\geq$ MinConf.

Câu	Nội dung cần trình bày
11	Boolean: sự có/vắng mặt của item, ví dụ "mua máy tính $\Rightarrow$ mua phần mềm". Quantitative: liên quan đến giá trị định lượng, ví dụ "tuổi 30-39 $\Rightarrow$ mua TV cao cấp".
12	Single-dimensional: một chiều (ví dụ: chỉ "buys"). Multidimensional: nhiều chiều (ví dụ: age + income + buys). Hữu ích khi muốn tìm quan hệ phức tạp giữa nhiều thuộc tính nhân khẩu học và hành vi mua hàng.
13	Multilevel: item ở nhiều mức trừu tượng (ví dụ: "laptop" và "computer"). Ưu: tìm được quan hệ tổng quát hơn. Thách thức: cần xác định MinSup phù hợp cho từng mức (mức tổng quát cần MinSup thấp hơn).
14	Data constraint: giới hạn item. Knowledge type: loại luật. Dimension/level: chiều/mức dữ liệu. Rule constraint: cấu trúc luật. Interestingness: ngưỡng support/confidence/lift. Anti-monotone: nếu itemset vi phạm thì superset cũng vi phạm $\Rightarrow$ cắt tỉa sớm.
15	lift = confidence($A \Rightarrow B$)/support(B) = support($A \cup B$)/(support(A)·support(B)). lift>1: tương quan thuận; lift$=1$: độc lập; lift<1: tương quan nghịch. Lift quan trọng hơn confidence vì xét đến tần suất nền của B.
16	Ví dụ: support(trà) = 90%, support(cà phê) = 90%, support(trà$\cup$cà phê) = 80%. Confidence(trà$\Rightarrow$cà phê) = 89% (mạnh). Nhưng lift = $80\%/(90\%\times 90\%) = 0.99 < 1 \Rightarrow$ tương quan nghịch thực sự. Khắc phục: dùng lift hoặc χ^2.
17	$\chi^2 = \sum(O-E)^2/E$. O: tần suất quan sát, E: tần suất kỳ vọng nếu độc lập. $\chi^2 = 0$: hai item hoàn toàn độc lập ($O = E$). $\chi^2 > 0$ và test có ý nghĩa thống kê: có tương quan.
18	$\text{cosine}(A,B) = \text{support}(A \cup B)/\sqrt{\text{support}(A)\cdot\text{support}(B)}$. Ưu điểm: chuẩn hóa về $[0,1]$, đối xứng. Nhược: không phân biệt tương quan thuận/nghịch. Lift phân biệt được nhưng không bị chặn trong $[0,1]$.
19	Hạn chế: (1) quét dữ liệu nhiều lần; (2) sinh nhiều ứng viên tốn bộ nhớ. Cải tiến: (1) FP-Growth – không sinh ứng viên; (2) Hash-based – dùng bảng băm đếm ứng viên hiệu quả hơn; (3) Sampling – lấy mẫu dữ liệu; (4) Partition – chia dữ liệu thành phân vùng.
20	Bước 1 tốn kém vì số itemset ứng viên có thể lên đến 2^m (với m item). Cần quét D nhiều lần ($k_{\max}$ lần). Độ phức tạp phụ thuộc: số item m, số giao dịch n, MinSup. MinSup thấp $\Rightarrow$ nhiều tập phổ biến $\Rightarrow$ nhiều ứng viên $\Rightarrow$ tốn kém hơn nhiều.

Câu hỏi trắc nghiệm – Đáp án

Câu	ĐA	Câu	ĐA	Câu	ĐA	Câu	ĐA
1	B	11	B	21	B	31	B
2	B	12	C	22	A	32	B
3	B	13	C	23	C	33	B
4	B	14	B	24	B	34	C
5	A	15	B	25	C	35	B
6	C	16	B	26	B	36	B
7	B	17	B	27	B	37	B
8	C	18	B	28	B	38	C
9	B	19	B	29	B	39	B
10	B	20	B	30	B	40	B

Giải thích đáp án trắc nghiệm

Câu	Giải thích
1	(B) – Khai phá luật kết hợp là học không giám sát: tìm quan hệ ẩn mà không có nhãn lớp.
2	(B) – Basket analysis: phát hiện sản phẩm mua cùng nhau trong cùng giao dịch.
3	(B) – k-itemset là tập hợp gồm đúng k item.
4	(B) – support($A \Rightarrow B$) = support($A \cup B$) = $P(A \cup B)$.
5	(A) – confidence($A \Rightarrow B$) = support($A \cup B$)/support(A) = $P(B\|A)$.
6	(C) – Frequent itemset: support $\geq$ MinSup (không phải MinConf).
7	(B) – Luật mạnh: thỏa đồng thời cả MinSup và MinConf.
8	(C) – confidence = 0.4/0.5 = 0.8 = 80%.
9	(B) – Apriori property: mọi tập con của tập phổ biến đều phổ biến.
10	(B) – Anti-monotone: nếu X không phổ biến thì $X \cup Y$ cũng không phổ biến.
11	(B) – Self-joining: nối hai L_{k-1} itemset có $k-2$ item đầu giống nhau.
12	(C) – Pruning: loại $c \in C_k$ nếu có $(k-1)$-subset của c không có trong L_{k-1}.
13	(C) – Apriori quét dữ liệu $k_{\max}$ lần (một lần cho mỗi cấp độ k).

Câu	Giải thích
14	(B) – Self-join(L_3): $abc + abd \rightarrow abcd$; $acd + ace \rightarrow acde$. Kết quả: $\{abcd, acde\}$.
15	(B) – Pruning $acde$: kiểm tra 3-subset $ade \notin L_3 \Rightarrow$ loại $acde$. Giữ $abcd$.
16	(B) – FP-Growth không sinh ứng viên và chỉ quét dữ liệu 2 lần (lần 1 tìm L_1, lần 2 xây FP-tree).
17	(B) – FP-tree nén tập dữ liệu vào cây tiền tố để khai phá không cần sinh ứng viên.
18	(B) – confidence = support($\{A,B,C\}$)/support($\{A,B\}$) = 3/3 = 100%.
19	(B) – Tính đơn điệu của confidence: luật có vế trái là tập cha của luật vi phạm cũng vi phạm $\Rightarrow$ cắt tỉa.
20	(B) – Luật có age và buys là hai chiều khác nhau $\Rightarrow$ multidimensional quantitative rule.
21	(B) – Multilevel: item ở nhiều mức trừu tượng (laptop là mức con, computer là mức cha).
22	(A) – Ràng buộc anti-monotone: vi phạm tập con $\Rightarrow$ vi phạm superset $\Rightarrow$ cắt tỉa sớm.
23	(A) – lift = support($A \cup B$)/(support(A)·support(B)).
24	(C) – lift = 1: $P(A \cup B) = P(A) \cdot P(B) \Rightarrow A$ và B độc lập.
25	(B) – lift < 1: tần suất đồng xuất hiện thấp hơn kỳ vọng $\Rightarrow$ tương quan nghịch.
26	(B) – $\chi^2 = 0$ khi $O = E$ (tần suất quan sát = kỳ vọng) $\Rightarrow$ hai item độc lập.
27	(B) – Luật mạnh không đảm bảo tương quan thực; cần lift/χ^2 để phát hiện tương quan giả.
28	(B) – C_k: candidate k-itemsets (tập ứng viên).
29	(B) – L_k: frequent k-itemsets (tập phổ biến sau khi lọc MinSup).
30	(B) – Bước 1 (tìm tập phổ biến) tốn kém nhất vì không gian tìm kiếm lên đến 2^m.
31	(B) – MinSup thấp $\Rightarrow$ nhiều itemset vượt ngưỡng $\Rightarrow$ số tập phổ biến tăng, tính toán tốn kém hơn.
32	(B) – MinSup = 2 tính theo count. Item $\{4\}$ có count = $1 < 2 \Rightarrow$ không phổ biến.
33	(C) – Theo anti-monotone: $\{A,B\}$ không phổ biến $\Rightarrow \{A,B,C\}$ chắc chắn không phổ biến.

Câu	Giải thích
34	(B) – Số luật từ itemset n phần tử: $2^n - 2$ (trừ $\emptyset$ và bản thân L).
35	(B) – $\text{cosine}(A,B) = \text{support}(A \cup B) / \sqrt{\text{support}(A) \cdot \text{support}(B)}$.
36	(B) – Support constraint (support $\geq$ MinSup) là anti-monotone: tập cha có support $\leq$ tập con.
37	(B) – MinSup thấp $\Rightarrow$ rất nhiều tập phổ biến $\Rightarrow$ Apriori sinh hàng triệu ứng viên; FP-Growth tránh hoàn toàn việc sinh ứng viên.
38	(C) – Khai phá đa mức: mỗi mức có tần suất khác nhau, cần MinSup riêng (mức tổng quát thường cần MinSup cao hơn).
39	(B) – Câu 39 giải thích lý do FP-Growth hiệu quả hơn Apriori khi MinSup thấp (đã bao phủ ở câu 37). Ở đây liên quan đến constraint anti-monotone.
40	(B) – Khai phá đa mức cần MinSup khác nhau cho từng mức trừu tượng để tránh quá nhiều hoặc quá ít luật ở mỗi mức.