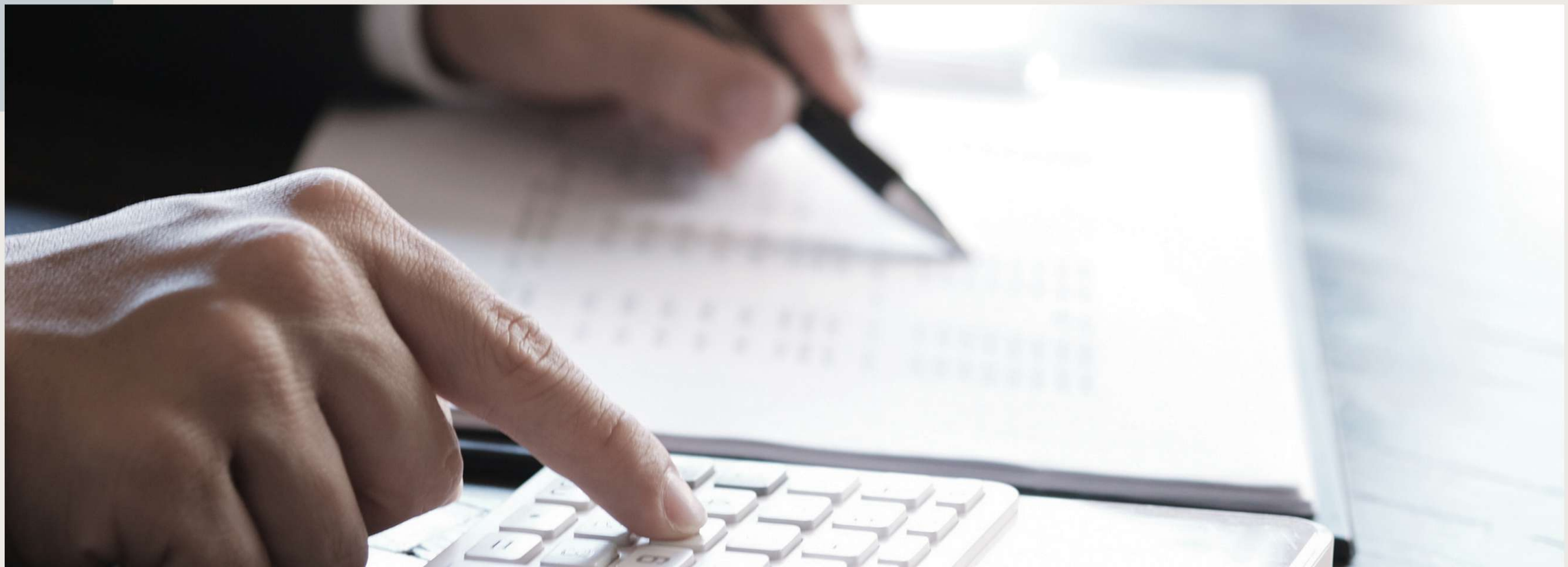
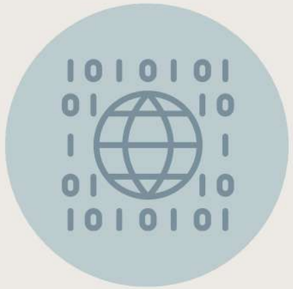
DATA MINING

GROUP 8



NỘI DUNG



I

ASSOCIATION RULE

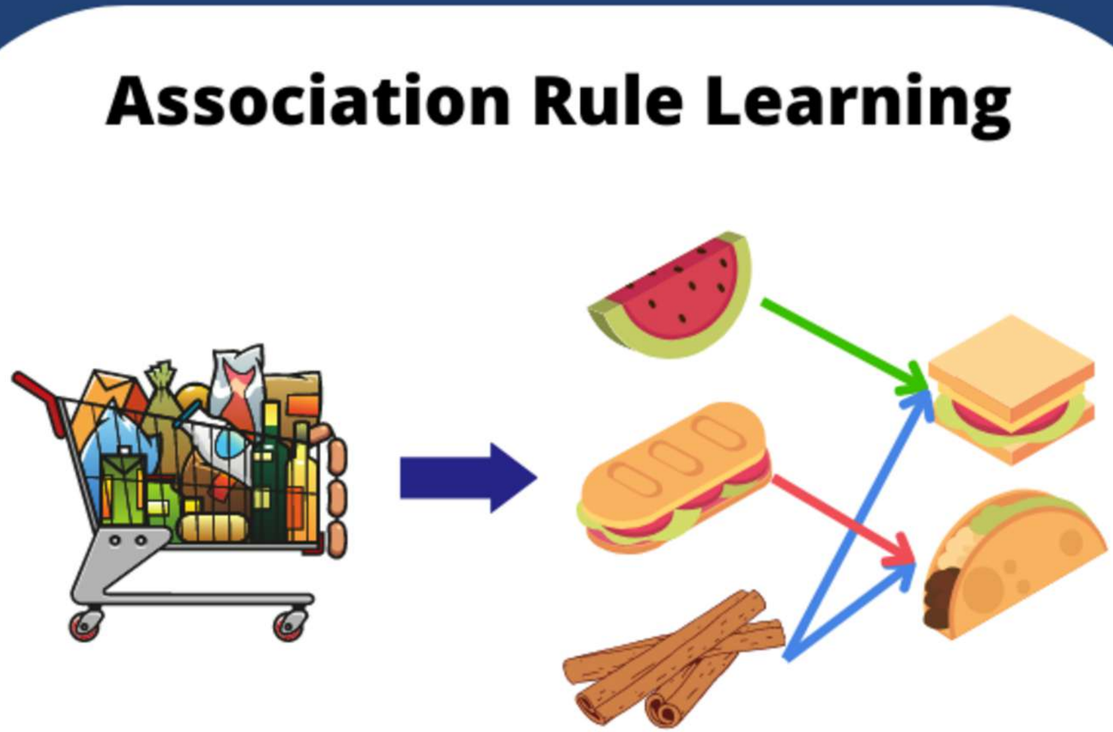
II

APRIORI ALGORITHM

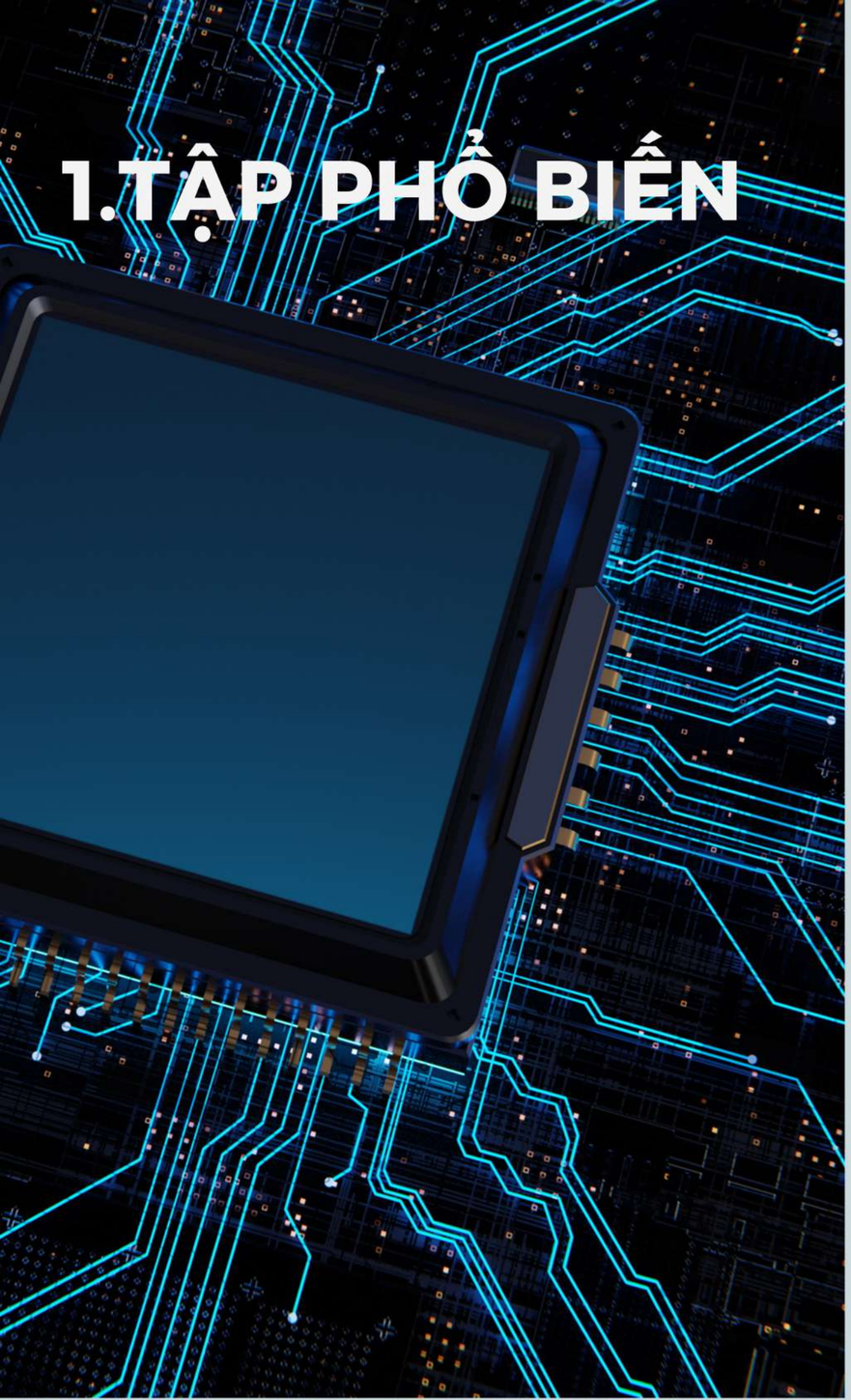
III

FP-GROWTH ALGORITHM

I - ASSOCIATION RULE



1.TẬP PHỔ BIẾN



Khái niệm: Tập phổ biến là một khái niệm quan trọng trong lĩnh vực khai phá dữ liệu. Nó là một tập hợp các mẫu (items) xuất hiện thường xuyên trong một tập dữ liệu.

Mục đích: Tìm ra các quy luật kết hợp giữa các mục (items) trong tập dữ liệu. Các quy luật kết hợp này có thể được sử dụng để phân tích hành vi tiêu dùng, tìm kiếm các mối liên hệ giữa các sản phẩm hoặc dịch vụ, đưa ra các khuyến nghị sản phẩm, và cải thiện hiệu quả kinh doanh.

VŨ THỊ HÀ

MỘT SỐ KHÁI NIỆM

- **Cơ sở dữ liệu giao dịch:** lưu trữ các giao dịch được thực hiện trong một khoảng thời gian nhất định, trong đó mỗi giao dịch là một danh sách các mục (items) mà khách hàng đã mua hoặc sử dụng.
- **Hạng mục (item):** là một đối tượng hoặc một thực thể riêng biệt trong một danh sách hoặc một tập hợp có thể là sản phẩm, mặt hàng (sữa, trứng, bánh mì,.., dịch vụ, hoặc bất kỳ đối tượng nào khác trong tập dữ liệu được phân tích.
- **Tập hạng mục (itemset):** có thể là một tập hợp các hạng mục đơn hoặc một tập hợp các hạng mục kết hợp trong tập dữ liệu. (các hạng mục trong giỏ hàng)

BÀI TOÁN KHAI THÁC TẬP PHỔ BIẾN

- Là bài toán tìm tất cả các hạng mục (itemset) X có độ phổ biến (support) thỏa mãn độ phổ biến tối thiểu (minsupport)

Sup(X) >= minsup

ĐỘ PHỔ BIẾN?

Cho tập I = {i1, i2, …, in}

- Hạng mục (item): i1, i2, …, in
- Tập hạng mục (itemset):Tập X I

Cho tập D = {T1, T2, …, Tn} T1:

- Một giao dịch (transaction)
- D: CSDL giao dịch
- |D|: Số giao dịch trong D

Độ phổ biến: sup(X)=C(X)/|D|

X={A, C}
sup(X)=3/5 (60%)

- sup({A})=5/5
- sup({C})=3/5

I = {A, B, C, D, E}
Ex. tập hạng mục: X={A, E, D}
|D| = 5
X={A, C}
sup(X)=3/5 (60%)

(1) Minsup=60%

(2) Minsup=70%

T_1	{A, B, C, D}
T_2	{A, C, E}
T_3	{A, E}
T_4	{A, B, E}
T_5	{A, B, C, D, E}

T_1	{A, B, C, D}
T_2	{A, C, E}
T_3	{A, E}
T_4	{A, B, E}
T_5	{A, B, C, D, E}

Minsup=60%

VŨ THỊ HÀ

Tính chất tập phổ biến

- **Tập con của tập phổ biến là tập phổ biến.**
- **Tập bao của tập KHÔNG phổ biến là tập KHÔNG phổ biến.**
- **Các tính chất của độ phổ biến trong data mining bao gồm:** tính mô tả, tính ứng dụng, tính tương quan

VÍ DỤ

T_1	{A, B, C, D}
T_2	{A, C, E}
T_3	{A, E}
T_4	{A, B, E}
T_5	{A, B, C, D, E}

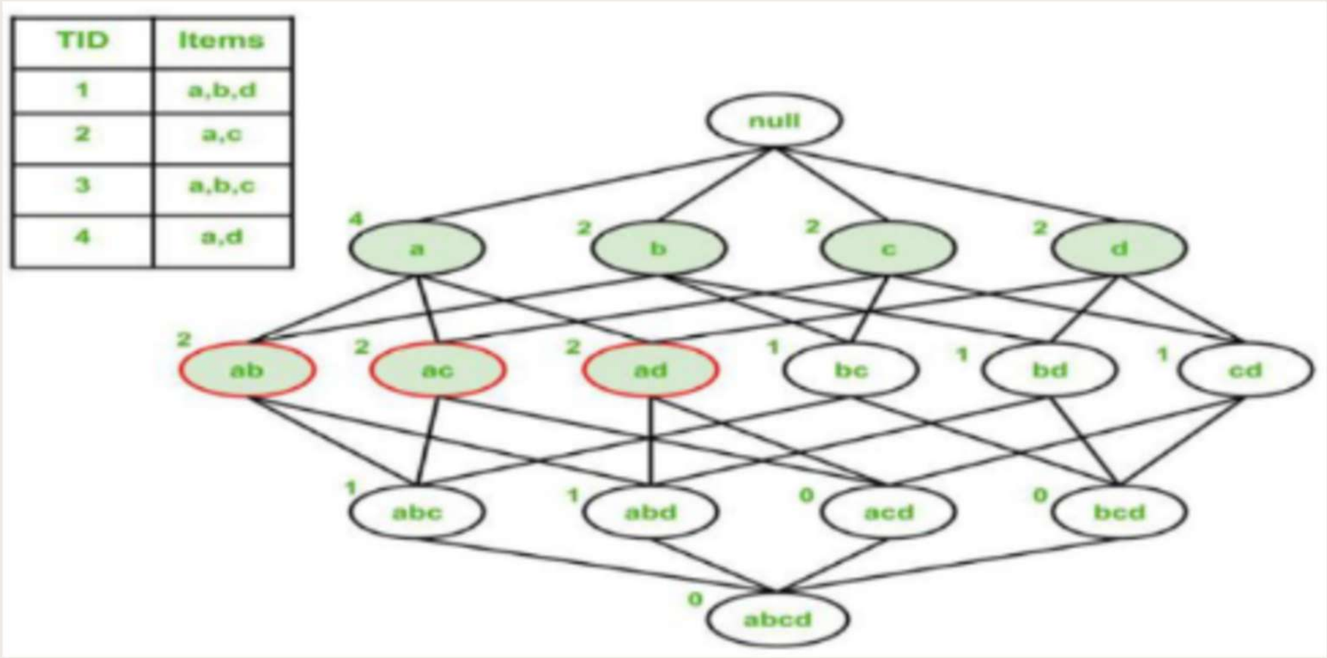
Minsup=60%

Nếu tập {A,D} KHÔNG phổ biến, thì các tập chứa nó như {A,B,C,D} cũng KHÔNG phổ biến (sup({D})=2/5 <minsup: tập không phổ biến).

=> {A, D}: không phải là tập phổ biến.

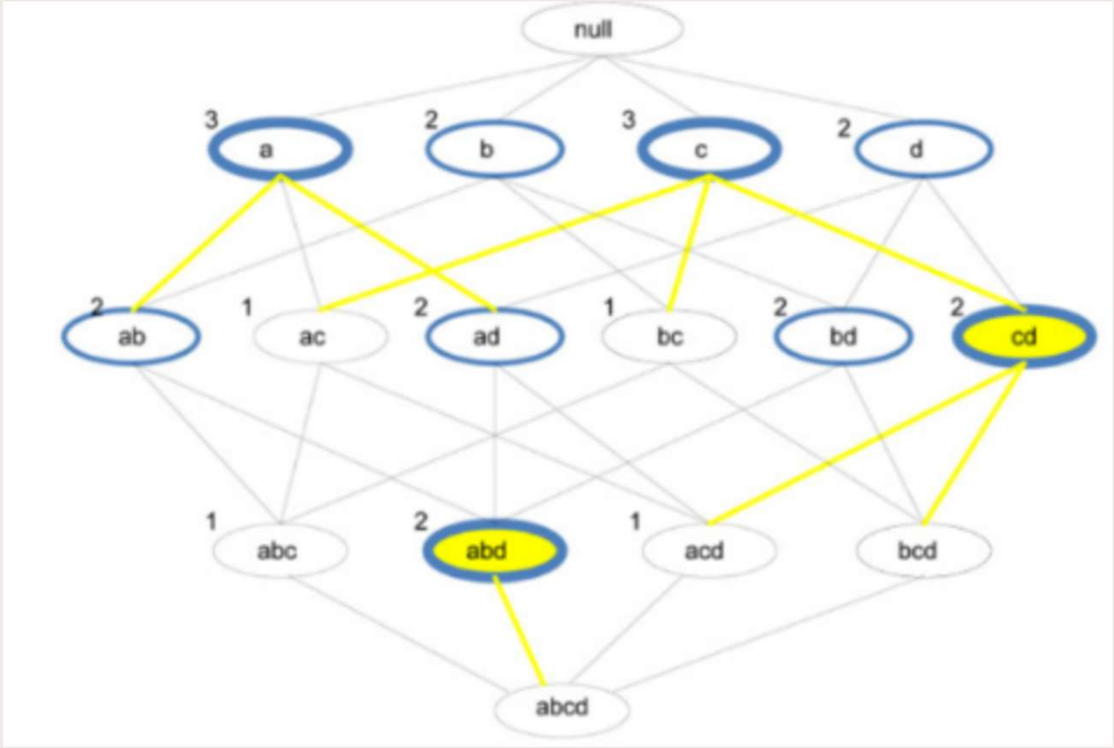
Tập phổ biến tối đại
(Maximal frequent itemset)

Là tập phổ biến và KHÔNG tồn tại tập bao nó là tập phổ biến



Tập bao phổ biến
(Closed frequent itemset)

Là tập phổ biến và không tồn tại tập nào bao nó có cùng độ phổ biến.

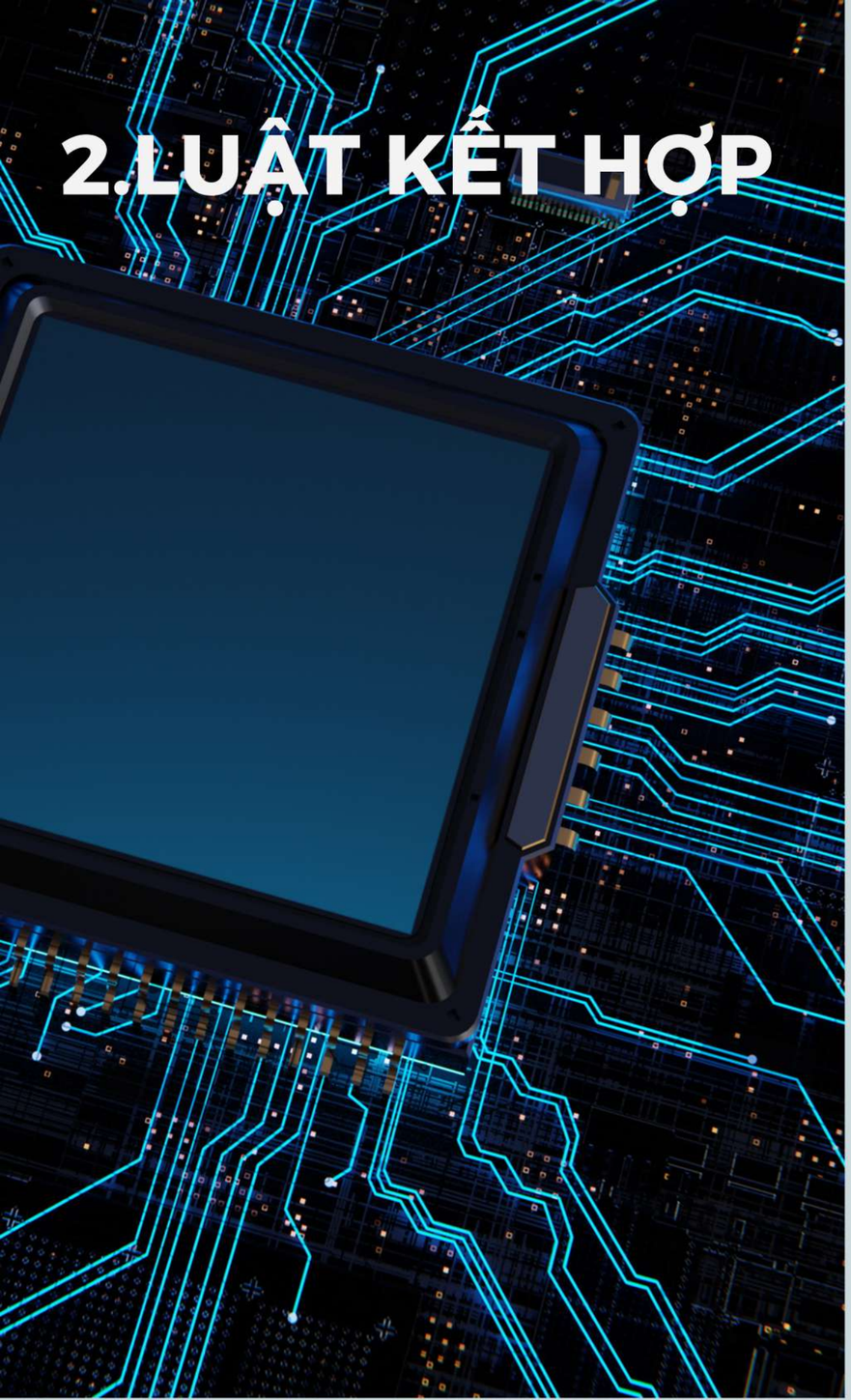


SO SÁNH CÁC TẬP

TẬP PHỔ BIẾN (PATTERN BASE)	TẬP PHỔ BIẾN TỐI ĐA (MAXIMAL PATTERN BASE)	TẬP BAO PHỔ BIẾN (CLOSED PATTERN BASE)
Tập phổ biến là tập hợp các mục (items) xuất hiện trong Luật kết hợp của một tập dữ liệu. **VD**: Nếu tập dữ liệu có các giao dịch mua hàng, tập PB có thể chứa các mục như {sữa, bánh mì}, {sữa, trái cây}, {sữa, thịt}.	Tập phổ biến tối đa là một tập phổ biến mà không thể mở rộng thêm bằng cách thêm bất kỳ mục nào khác. **VD**: Nếu tập PB gốc chứa các tập {A, B} và {A, B, C}, thì chỉ tập {A, B} được coi là tối đa.	Tập bao phổ biến là một tập phổ biến mà không có tập PB khác chứa nó nhưng có cùng hỗ trợ (support). **VD**: Nếu tập PB gốc chứa các tập {A, B}, {A, B, C}, và {A, B, D}, thì chỉ tập {A, B} được coi là tập bao PB.

PHƯƠNG UYÊN

2.LUẬT KẾT HỢP



Khái niệm: Luật kết hợp là mối quan hệ giữa các tập thuộc tính trong cơ sở dữ liệu. Luật kết hợp là phương tiện hữu ích để khám phá các mối liên kết trong dữ liệu. Có 2 độ đo quan trọng đối với luật kết hợp: Độ hỗ trợ (support) và độ tin cậy (confidence).

ĐỘ PHỔ BIẾN

$$sup(X \rightarrow Y) = P(X \cup Y) = sup(X \cup Y)$$

Độ phổ biến của một luật kết hợp X => Y là tỷ lệ giữa số lượng các bản ghi chứa tập hợp X => Y, so với tổng số các bản ghi trong D - Ký hiệu sup(X => Y).

ĐỘ TIN CẬY

$$Conf(X \rightarrow Y) = P(Y|X) = sup(X \cup Y) / sup(X)$$

Độ tin cậy (confidence) của luật X → Y trong D, ký hiệu conf (X → Y , D), là tỉ lệ giữa số giao dịch chứa cả X và Y trên số giao dịch chỉ chứa X.

VÍ DỤ

Transaction ID	Items bought
1	Tea, Cake, Cold Drink
2	Tea, Coffee, Cold Drink
3	Eggs, Tea, Cold Drink
4	Cake, Milk, Eggs
5	Cake, Coffee, Cold Drink, Milk, Eggs

X → Y	sup	conf
Tea → Cold Drink	60%	100%
Cake → Milk	40%	67%
Tea → Milk	0%	0%

QUY TRÌNH KHAI
THÁC LUẬT KẾT HỢP

B1: Tìm tất cả các tập PB (theo ngưỡng minsup)

B2: Tạo ra các luật từ các tập PB

- Mỗi tập phổ biến X, tạo các tập con khác rỗng của X

• Đối với mỗi tập con (khác rỗng) A của X, B=X-A

A → B là LKH nếu conf(A → B) >=minconf

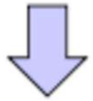
conf(A→ B)=sup(A B)/sup(A)=sup(X)/sup(A)

VÍ DỤ

minsup=60%,
minconf=80%

T_1	{A, B, C, D}
T_2	{A, C, E}
T_3	{A, E}
T_4	{B, C, E}
T_5	{A, B, C, D, E}

minsup=60%,

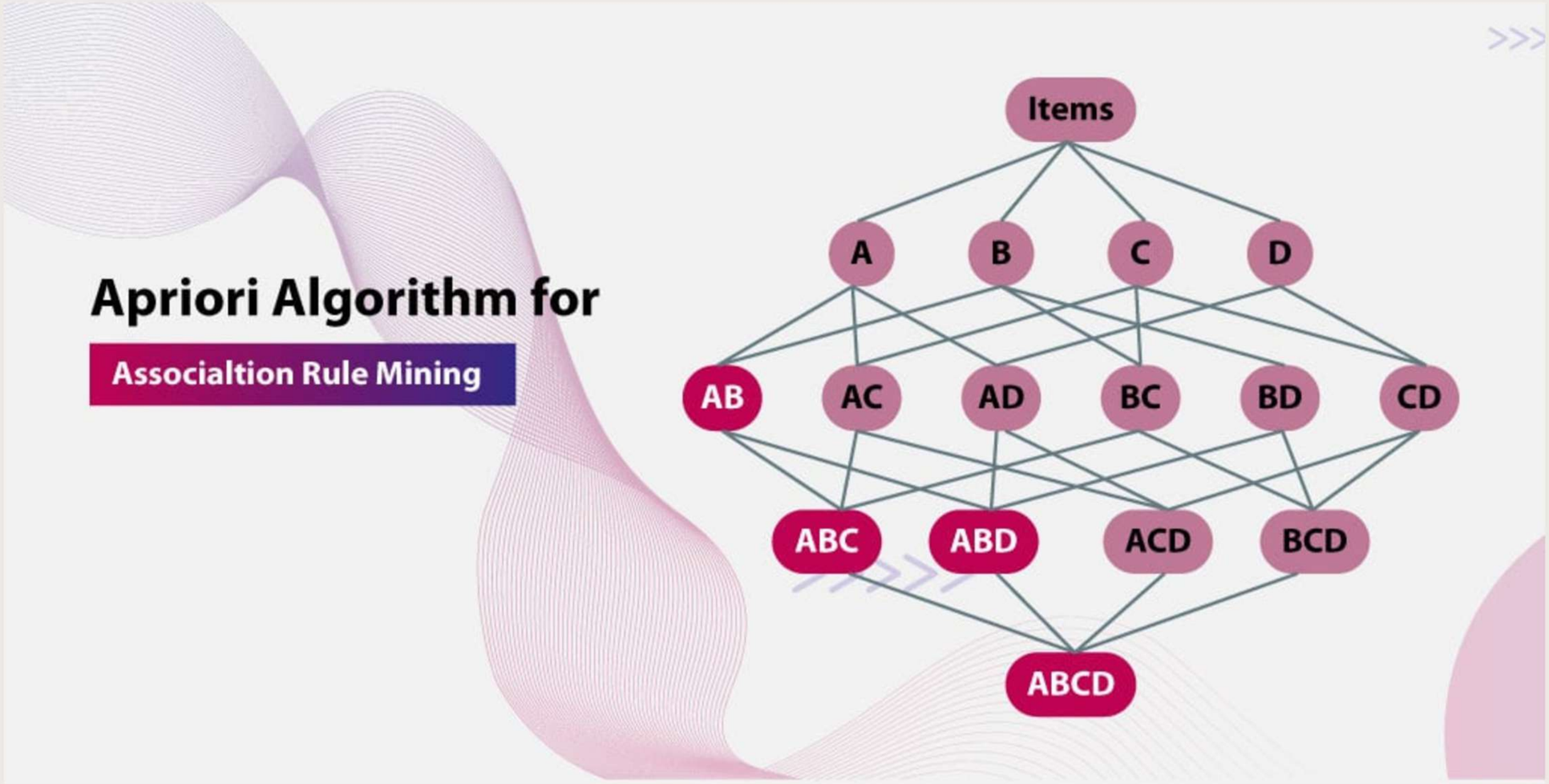


Frequent itemset	Support (%)
{A}, {B}, {C}, {E}	80, 60, 80, 80
{A, C}, {A, E}, {B, C}, {C, E}	60, 60, 60, 60

minconf=80%

Luật	conf(X→Y)	Chọn/Loại
A → C	sup({A}∪{C})/sup({A})=3/4=75%	Loại
C → A	sup({C}∪{A})/sup({C})=3/4=75%	Loại
A → E	sup({A}∪{E})/sup({A})=3/4=75%	Loại
E → A	sup({E}∪{A})/sup({E})=3/4=75%	Loại
B → C	sup({B}∪{C})/sup({B})=3/3=100%	**Chọn**
C → B	sup({C}∪{B})/sup({C})=3/4=75%	Loại
C → E	sup({C}∪{E})/sup({C})=3/4=75%	Loại
E → C	sup({E}∪{C})/sup({E})=3/4=75%	Loại

II - APRIORI ALGORITHM



Khái niệm: Thuật toán Apriori là một thuật toán được sử dụng trong khai phá dữ liệu để tìm kiếm các tập hợp mục thường xuyên và học các quy tắc kết hợp trong cơ sở dữ liệu quan hệ.

Cách thức hoạt động:

1. Xác định ngưỡng hỗ trợ và ngưỡng tin cậy
2. Tìm tất cả các tập phổ biến 1 hạng mục
3. Tạo các tập ứng viên k+1 hạng mục từ tập phổ biến có k hạng mục
4. Kiểm tra độ phổ biến của các ứng viên
5. Dừng khi không tạo được tập phổ biến hoặc tập ứng viên

T_1	{A, B, C, D}
T_2	{A, C, E}
T_3	{B, C, E}
T_4	{B, C, E}
T_5	{A, B, C, D, E}

minsup= 60% ; minconf = 80%

A: Bánh mì

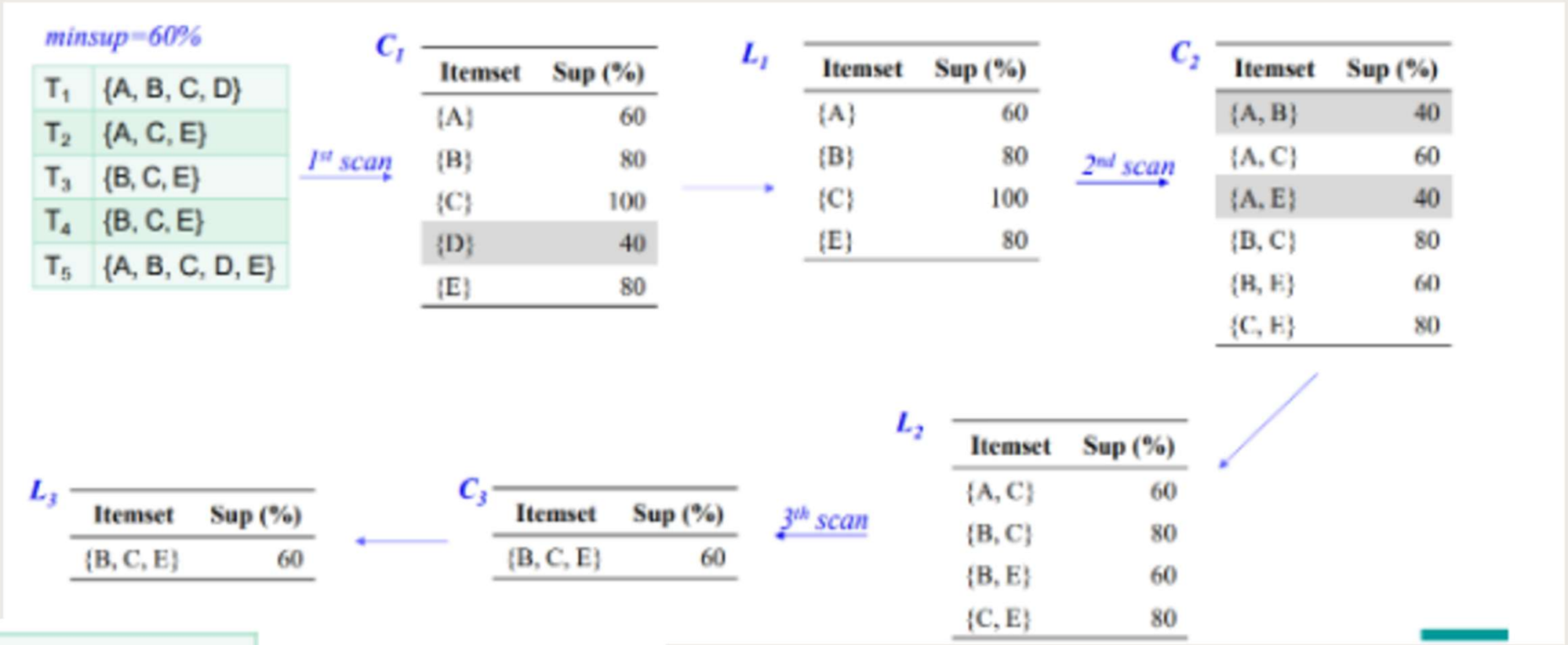
B: Sữa

C: Bơ

D: Táo

E: Kem

KIỀU DOAN



T_1	{A, B, C, D}
T_2	{A, C, E}
T_3	{B, C, E}
T_4	{B, C, E}
T_5	{A, B, C, D, E}

Frequent Itemset	Sup (%)
{A, C}	60
{B, C}	80
{B, E}	60
{C, E}	80
{B, C, E}	60

Từ các bước vừa rồi, ta được bảng Frequent Itemset từ các mục L2 và L3

KIỀU DOAN

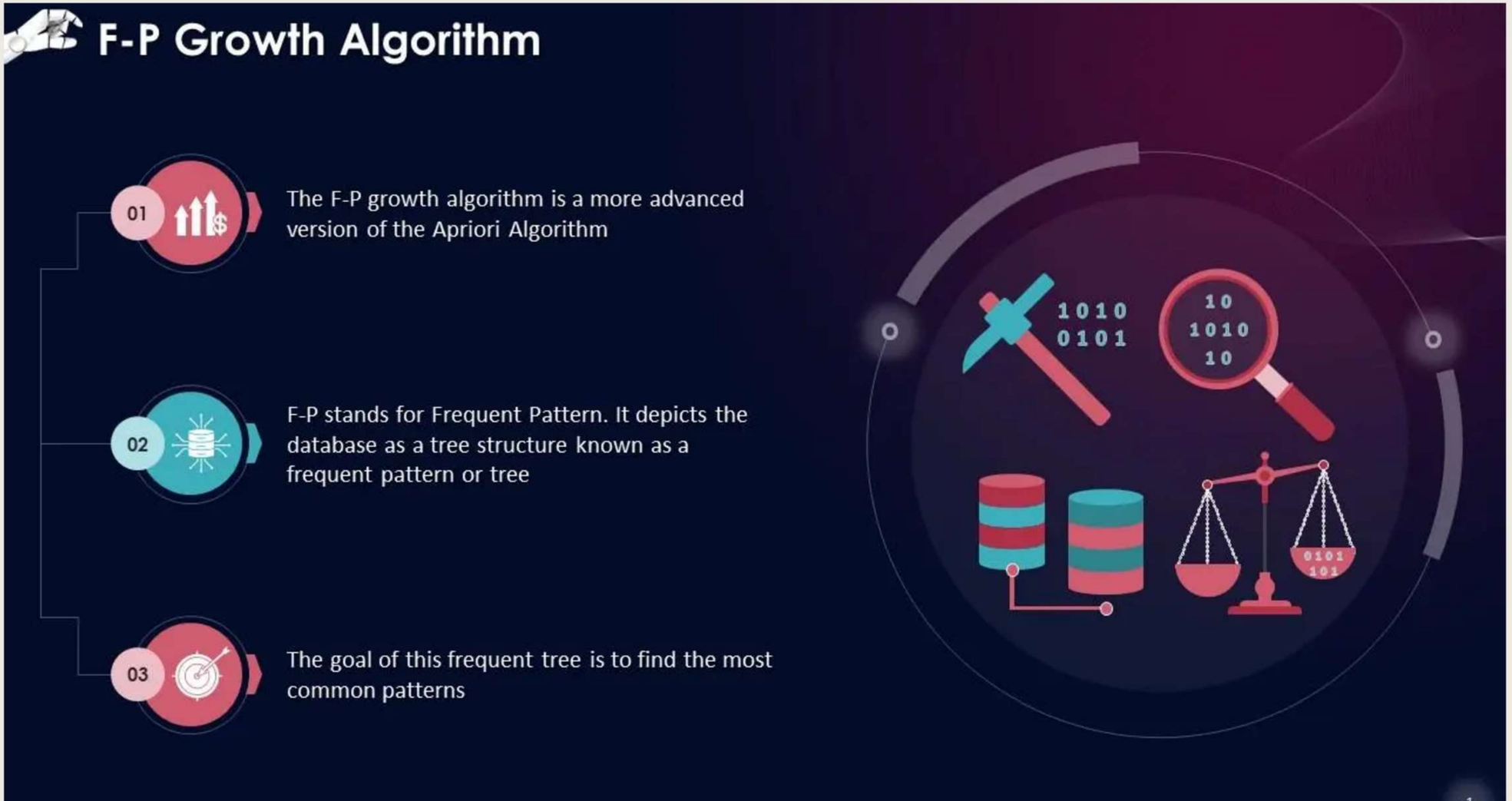
AR	conf (%)
{A} → {C}	100
{C} → {A}	60
{B} → {C}	100
{C} → {B}	80
{B} → {E}	75
{E} → {B}	75
{C} → {E}	80
{E} → {C}	100
{B} → {C, E}	75
{C} → {B, E}	60
{E} → {B, C}	75
{B, C} → {E}	75
{B, E} → {C}	100
{C, E} → {B}	75

Tính độ tin cậy và so với mincof =80% và sẽ có 6 luật kết hợp như hình.

{A} → {C} là Bánh mì → Bơ: Khách hàng mua Bánh mì khả năng cao mua Bơ

{B, E} → {C} là Sữa, Kem → Bơ: Khách hàng mua Sữa, Kem khả năng cao mua Bơ

III - FP-GROWTH ALGORITHM



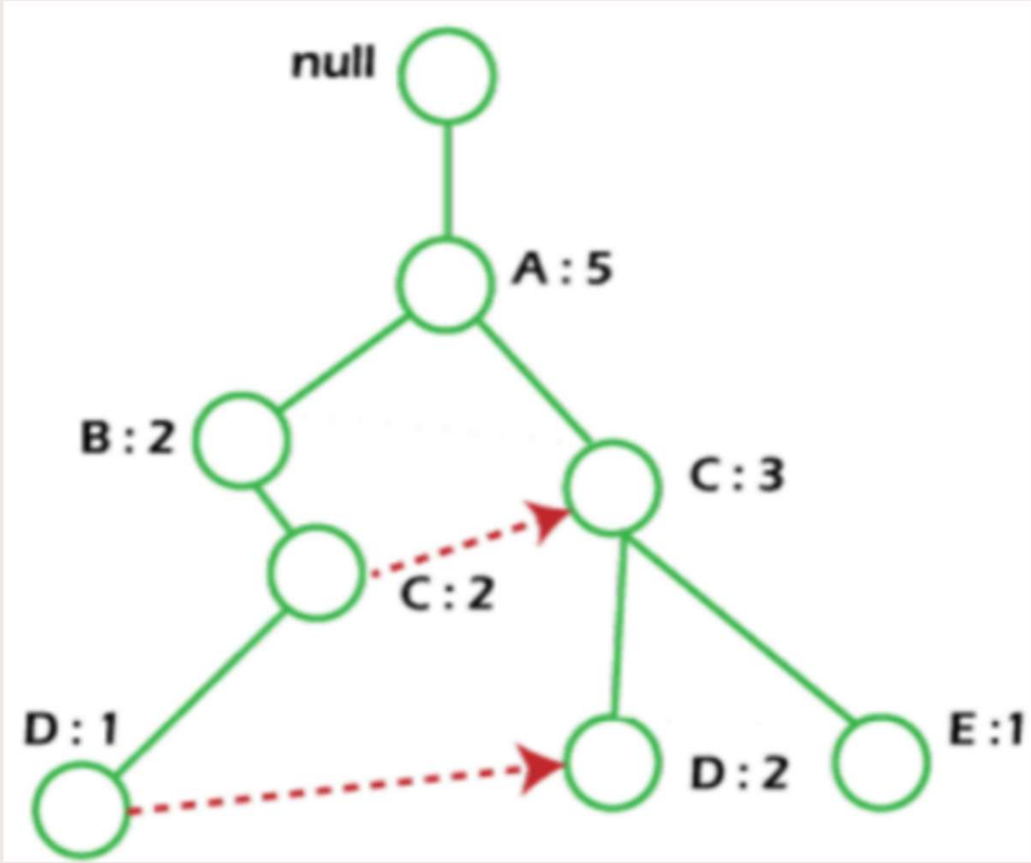
KHÁI NIỆM

- **Thuật toán tăng trưởng FP** là một cách khác để tìm các tập mục phổ biến mà không cần sử dụng các thế hệ ứng viên, do đó cải thiện hiệu suất. Đối với rất nhiều, nó sử dụng chiến lược chia để trị. Cốt lõi của phương pháp này là việc sử dụng một cấu trúc dữ liệu đặc biệt có tên là cây mẫu phổ biến (cây FP), giữ lại thông tin liên kết của tập mục.
- **Các hoạt động của thuật toán:**
- Đầu tiên, nó nén cơ sở dữ liệu đầu vào tạo ra một thể hiện cây FP để biểu diễn các mục thường xuyên.
- Sau bước đầu tiên này, nó chia cơ sở dữ liệu đã nén thành một tập hợp các cơ sở dữ liệu có điều kiện, mỗi cơ sở dữ liệu được liên kết với một mẫu phổ biến.
- Cuối cùng, mỗi cơ sở dữ liệu như vậy được khai thác riêng.

CÂY FP

- **Cây mẫu phổ biến (FP-tree)** là một cấu trúc dữ liệu nhỏ gọn lưu trữ thông tin định lượng về
- **Các mẫu phổ biến trong cơ sở dữ liệu.**
- **Cây mẫu phổ biến được tạo với các bộ mục ban đầu của cơ sở dữ liệu.**

Mục đích: Mục đích của cây FP là khai thác mẫu thường xuyên nhất. Mỗi nút của cây FP đại diện cho một mục của tập mục.



ƯU ĐIỂM

- Cần quét cơ sở dữ liệu hai lần so với Apriori (quét các giao dịch cho mỗi lần lặp lại).
- Việc ghép nối các mục không được thực hiện trong thuật toán này, làm cho nó nhanh hơn.
- Cơ sở dữ liệu được lưu trữ trong một phiên bản nhỏ gọn trong bộ nhớ.
- Nó hiệu quả và có khả năng mở rộng để khai thác cả các mẫu phổ biến dài và ngắn.

NHƯỢC ĐIỂM

- FP Tree cồng kềnh và khó xây dựng hơn Apriori.
- Nó có thể đắt tiền.
- Thuật toán có thể không vừa với bộ nhớ dùng chung khi cơ sở dữ liệu lớn.

THANK YOU FOR ATTENTION