Nhập môn Học máy và Khai phá dữ liệu
(IT3190)

Nguyễn Nhật Quang

quang.nguyennhat@hust.edu.vn

Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ thông tin và truyền thông

Năm học 2020-2021

Nội dung môn học:

Giới thiệu về Học máy và Khai phá dữ liệu

Tiền xử lý dữ liệu

Đánh giá hiệu năng của hệ thống

Hồi quy

Phân lớp

Phân cụm

Phát hiện luật kết hợp

- **Bài toán phát hiện luật kết hợp**
- **Giải thuật Apriori**

Phát hiện các luật kết hợp – Giới thiệu

- Bài toán phát hiện luật kết hợp (Association rule mining)
  - Với một tập các giao dịch (transactions) cho trước, cần tìm các luật dự đoán khả năng xuất hiện trong một giao dịch của các mục (items) này dựa trên việc xuất hiện của các mục khác

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Các ví dụ của luật kết hợp:

{Diaper} → {Beer}

{Milk, Bread} → {Eggs, Coke}

{Beer, Bread} → {Milk}

Các định nghĩa cơ bản (1)

- **Tập mục (Itemset)**
  - Một tập hợp gồm một hoặc nhiều mục
    - Ví dụ: {Milk, Bread, Diaper}
  - Tập mục mức *k* (*k*-itemset)
    - Một tập mục gồm *k* mục
- **Tổng số hỗ trợ (Support count) σ**
  - Số lần xuất hiện của một tập mục
  - Ví dụ: σ({Milk, Bread, Diaper}) = 2
- **Độ hỗ trợ (Support) *s***
  - Tỷ lệ các giao dịch chứa một tập mục
  - Ví dụ: s({Milk, Bread, Diaper}) = 2/5
- **Tập mục thường xuyên (Frequent/large itemset)**
  - Một tập mục mà độ hỗ trợ lớn hơn hoặc bằng một giá trị ngưỡng *minsup*

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Các định nghĩa cơ bản (2)

- **Luật kết hợp (Association rule)**
  - Một biểu thức kéo theo có dạng: $X \rightarrow Y$, trong đó X và Y là các tập mục
  - Ví dụ: {Milk, Diaper} $\rightarrow$ {Beer}

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- **Các độ đo đánh giá luật**
  - **Độ hỗ trợ (Support) s**
    - Tỷ lệ các giao dịch chứa cả X và Y đối với tất cả các giao dịch
  - **Độ tin cậy (Confidence) c**
    - Tỷ lệ các giao dịch chứa cả X và Y đối với các giao dịch chứa X

$$\{\text{Milk}, \text{Diaper}\} \rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Phát hiện các luật kết hợp

- Với một tập các giao dịch *T*, mục đích của bài toán phát hiện luật kết hợp là tìm ra tất cả các luật có:
  - độ hỗ trợ ≥ giá trị ngưỡng *minsup*, **và**
  - độ tin cậy ≥ giá trị ngưỡng *minconf*
- Cách tiếp cận vét cạn (Brute-force)
  - Liệt kê tất cả các luật kết hợp có thể
  - Tính toán độ hỗ trợ và độ tin cậy cho mỗi luật
  - Loại bỏ đi các luật có độ hỗ trợ nhỏ hơn *minsup* hoặc có độ tin cậy nhỏ hơn *minconf*

$\Rightarrow$ Phương pháp vét cạn này có chi phí tính toán quá lớn, không áp dụng được trong thực tế!

Phát hiện luật kết hợp

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Các luật kết hợp:

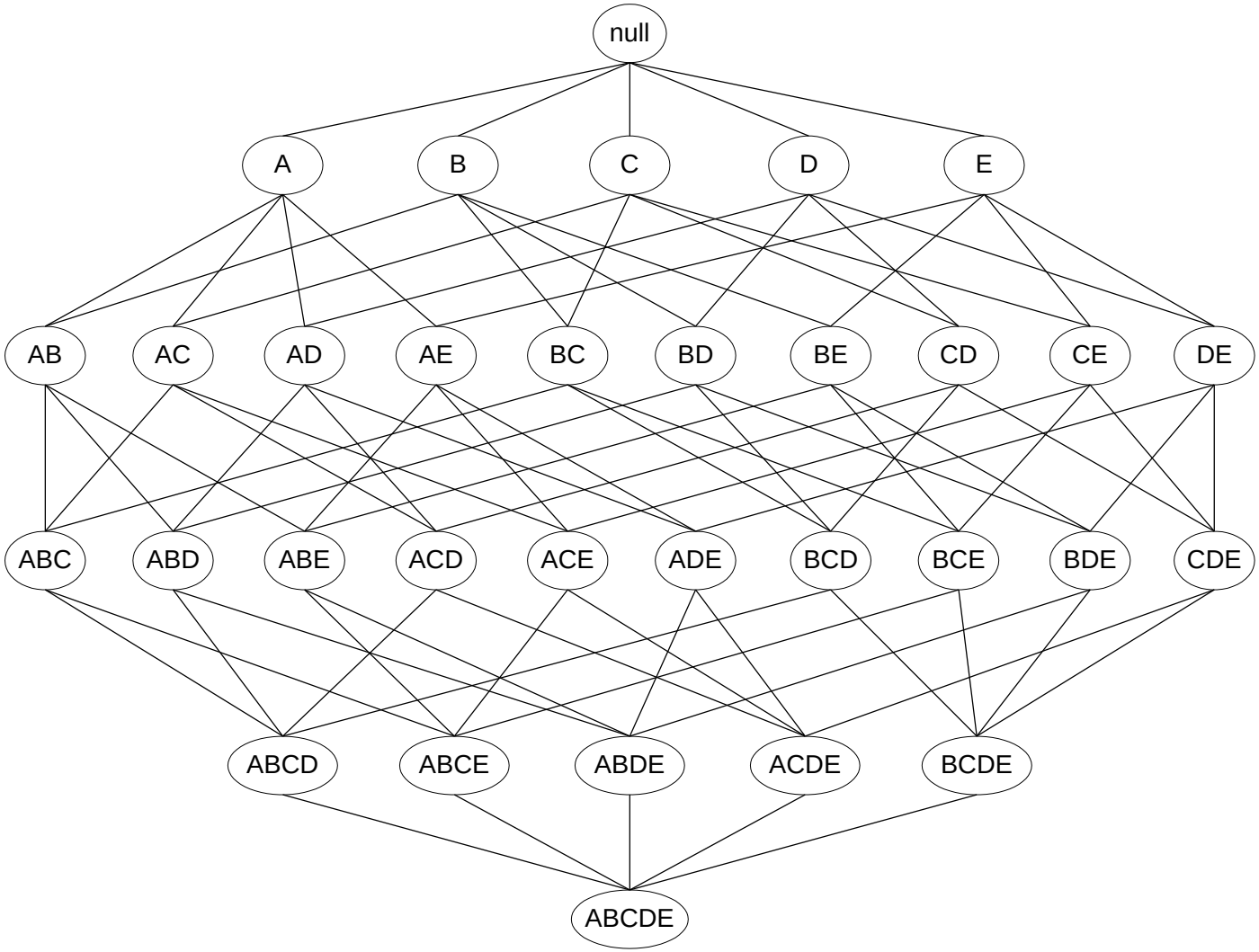
{Milk, Diaper} → {Beer} (s=0.4, c=0.67)
{Milk, Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper, Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk, Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk, Beer} (s=0.4, c=0.5)
{Milk} → {Diaper, Beer} (s=0.4, c=0.5)

- Tất cả các luật trên đều là sự phân tách (thành 2 tập con) của cùng tập mục : {Milk, Diaper, Beer}
- Các luật sinh ra từ cùng một tập mục sẽ có cùng độ hỗ trợ, nhưng có thể khác về độ tin cậy
- Do đó, trong quá trình phát hiện luật kết hợp, chúng ta có thể tách riêng 2 yêu cầu về độ hỗ trợ và độ tin cậy

Phát hiện luật kết hợp

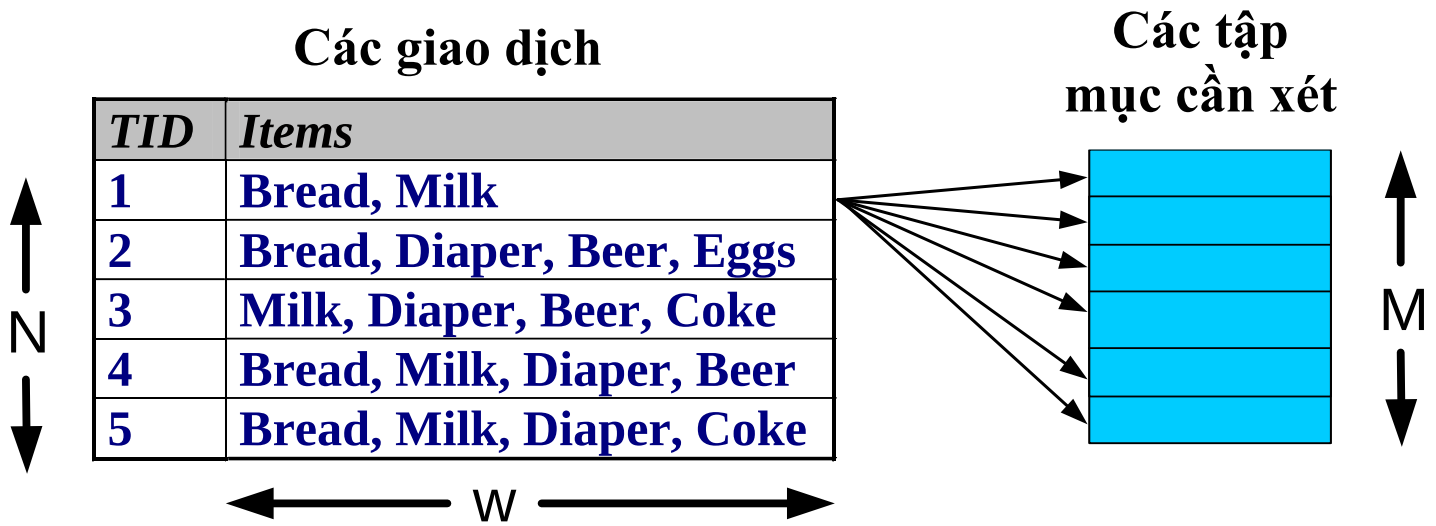
- Quá trình phát hiện luật kết hợp sẽ gồm 2 bước (2 giai đoạn) quan trọng:
  - **Sinh ra các tập mục thường xuyên** (frequent/large itemsets)
    - Sinh ra tất cả các tập mục có độ hỗ trợ ≥ *minsup*
  - **Sinh ra các luật kết hợp**
    - Từ mỗi tập mục thường xuyên (thu được ở bước trên), sinh ra tất cả các luật có độ tin cậy cao (≥ *minconf*)
    - Mỗi luật là một phân tách nhị phân (phân tách thành 2 phần) của một tập mục thường xuyên
- Bước sinh ra các tập mục thường xuyên (bước thứ 1) vẫn có chi phí tính toán quá cao!

Lattice biểu diễn các tập mục cần xét



Với *d* mục, thì phải xét đến 2^d các tập mục có thể!

Sinh ra các tập mục thường xuyên



- Phương pháp vét cạn (Brute-force)
  - Mỗi tập mục trong lattice đều được xét
  - Tính độ hỗ trợ của mỗi tập mục, bằng cách duyệt qua tất cả các giao dịch
  - Với mỗi giao dịch, so sánh nó với mỗi tập mục được xét
  - Độ phức tạp ~ O(N.M.w)
    - Với $M = 2^d$, thì độ phức tạp này là quá lớn!

Các chiến lược sinh tập mục thường xuyên

- Giảm bớt **số lượng các tập mục cần xét (M)**
  - Tìm kiếm (xét) đầy đủ: $M=2^d$
  - Sử dụng các kỹ thuật cắt tỉa (pruning) để giảm giá trị M
- Giảm bớt **số lượng các giao dịch cần xét (N)**
  - Giảm giá trị N, khi kích thước (số lượng các mục) của tập mục tăng lên
- Giảm bớt **số lượng các so sánh (matchings/comparisons) giữa các tập mục và các giao dịch (N.M)**
  - Sử dụng các cấu trúc dữ liệu phù hợp (hiệu quả) để lưu các tập mục cần xét hoặc các giao dịch
  - Không cần phải so sánh mỗi tập mục với mỗi giao dịch

Giảm bớt số lượng các tập mục cần xét

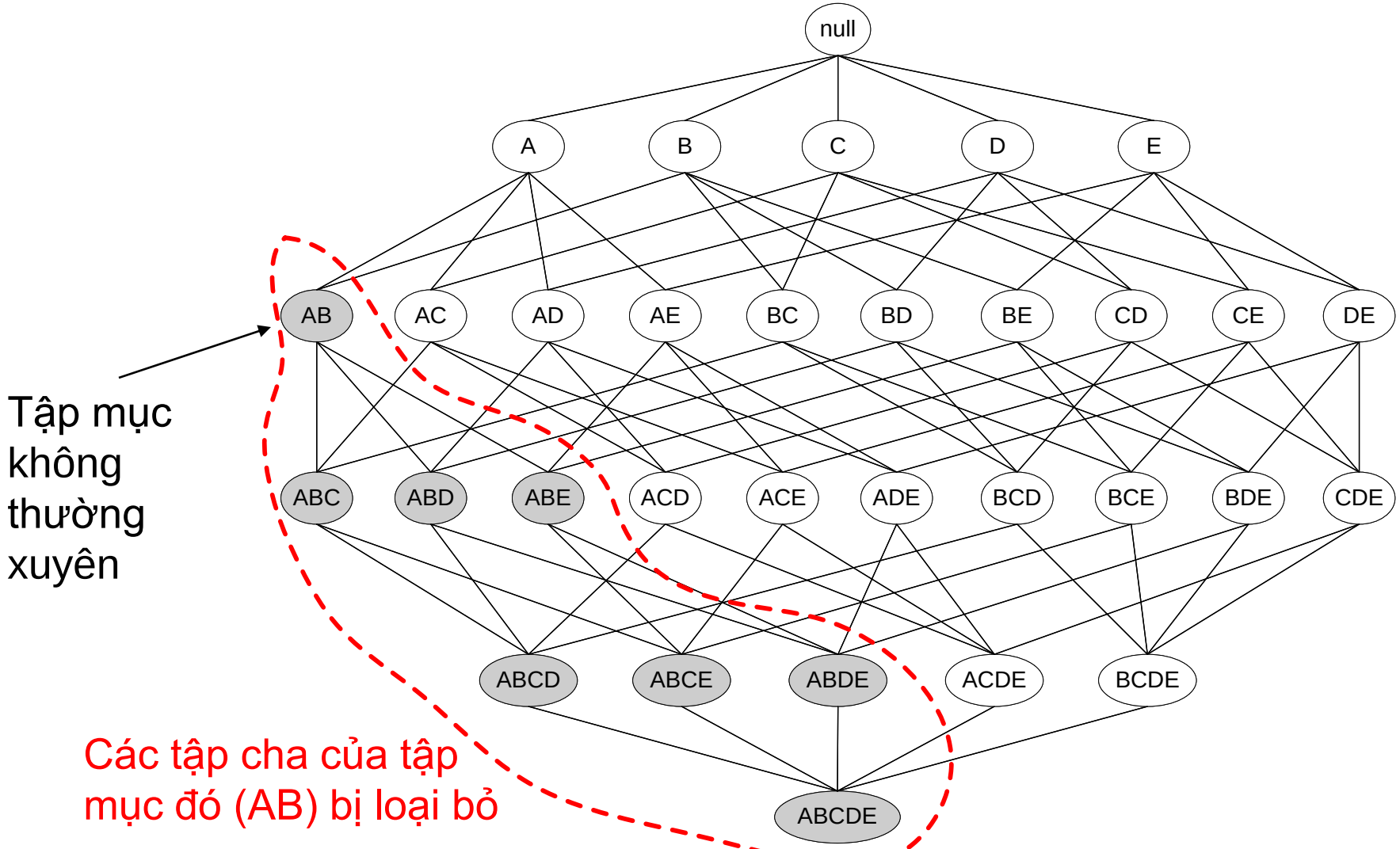
- Nguyên tắc của giải thuật Apriori – Loại bỏ (prunning) dựa trên độ hỗ trợ
  - Nếu một tập mục là thường xuyên, thì tất cả các tập con (subsets) của nó đều là các tập mục thường xuyên
  - Nếu một tập mục là không thường xuyên (not frequent), thì tất cả các tập cha (supersets) của nó đều là các tập mục không thường xuyên

- Nguyên tắc của giải thuật Apriori dựa trên **đặc tính không đơn điệu (anti-monotone) của độ hỗ trợ**

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

  - Độ hỗ trợ của một tập mục nhỏ hơn độ hỗ trợ của các tập con của nó

Apriori: Loại bỏ dựa trên độ hỗ trợ



Apriori: Loại bỏ dựa trên độ hỗ trợ

Item	Count
Bread	**4**
Coke	**2**
Milk	**4**
Beer	**3**
Diaper	**4**
Eggs	**1**

Các tập mục mức 1 (1-itemsets)

Itemset	Count
{Bread,Milk}	**3**
{Bread,Beer}	**2**
{Bread,Diaper}	**3**
{Milk,Beer}	**2**
{Milk,Diaper}	**3**
{Beer,Diaper}	**3**

Các tập mục mức 2 (2-itemsets)

(Không cần xét các tập mục có chứa mục *Coke* hoặc *Eggs*)

minsup = 3

Các tập mục mức 3 (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	**3**

- Nếu xét tất cả các tập mục có thể:
  $^6C_1 + {}^6C_2 + {}^6C_3 = 41$
- Với cơ chế loại bỏ dựa trên độ hỗ trợ:
  6 + 6 + 1 = 13

Giải thuật Apriori

- Sinh ra tất cả các tập mục thường xuyên mức 1 (frequent 1-itemsets): các tập mục thường xuyên chỉ chứa 1 mục
- Gán k = 1
- Lặp lại, cho đến khi không có thêm bất kỳ tập mục thường xuyên nào mới
  - Từ các tập mục thường xuyên mức k (chứa k mục), sinh ra các tập mục mức (k+1) cần xét
  - Loại bỏ các tập mục mức (k+1) chứa các tập con là các tập mục không thường xuyên mức k
  - Tính độ hỗ trợ của mỗi tập mục mức (k+1), bằng cách duyệt qua tất cả các giao dịch
  - Loại bỏ các tập mục không thường xuyên mức (k+1)
  - Thu được các tập mục thường xuyên mức (k+1)

Apriori: Các yếu tố ảnh hưởng độ phức tạp

- Lựa chọn **giá trị ngưỡng *minsup***
  - Giá trị minsup quá thấp sẽ sinh ra nhiều tập mục thường xuyên
  - Điều này có thể làm tăng số lượng các tập mục phải xét và độ dài (kích thước) tối đa của các tập mục thường xuyên
- **Số lượng các mục** trong cơ sở dữ liệu (các giao dịch)
  - Cần thêm bộ nhớ để lưu giá trị độ hỗ trợ đối với mỗi mục
  - Nếu số lượng các mục (tập mục mức 1) thường xuyên tăng lên, thì chi phí tính toán và chi phí I/O (duyệt các giao dịch) cũng tăng
- **Kích thước của cơ sở dữ liệu** (các giao dịch)
  - Giải thuật Apriori **duyệt cơ sở dữ liệu nhiều lần**. Do đó, chi phí tính toán của Apriori tăng lên khi số lượng các giao dịch tăng lên
- **Kích thước trung bình của các giao dịch**
  - Khi kích thước (số lượng các mục) trung bình của các giao dịch tăng lên, thì độ dài tối đa của các tập mục thường xuyên cũng tăng, và chi phí duyệt cây băm cũng tăng

Sinh ra các luật kết hợp (1)

- Với mỗi tập mục thường xuyên L, cần tìm tất cả các tập con khác rỗng $f \subset L$ sao cho: $f \rightarrow \{L \setminus f\}$ thỏa mãn điều kiện về độ tin cậy tối thiểu
  - Vd: Với tập mục thường xuyên {A,B,C,D}, các luật cần xét gồm có:

$$ABC \rightarrow D, \quad ABD \rightarrow C, \quad ACD \rightarrow B, \quad BCD \rightarrow A,$$
$$A \rightarrow BCD, \quad B \rightarrow ACD, \quad C \rightarrow ABD, \quad D \rightarrow ABC$$
$$AB \rightarrow CD, \quad AC \rightarrow BD, \quad AD \rightarrow BC, \quad BC \rightarrow AD,$$
$$BD \rightarrow AC, \quad CD \rightarrow AB,$$

- Nếu $|L| = k$, thì sẽ phải xét $(2^k - 2)$ các luật kết hợp có thể (bỏ qua 2 luật: $L \rightarrow \varnothing$ và $\varnothing \rightarrow L$)

Sinh ra các luật kết hợp (2)

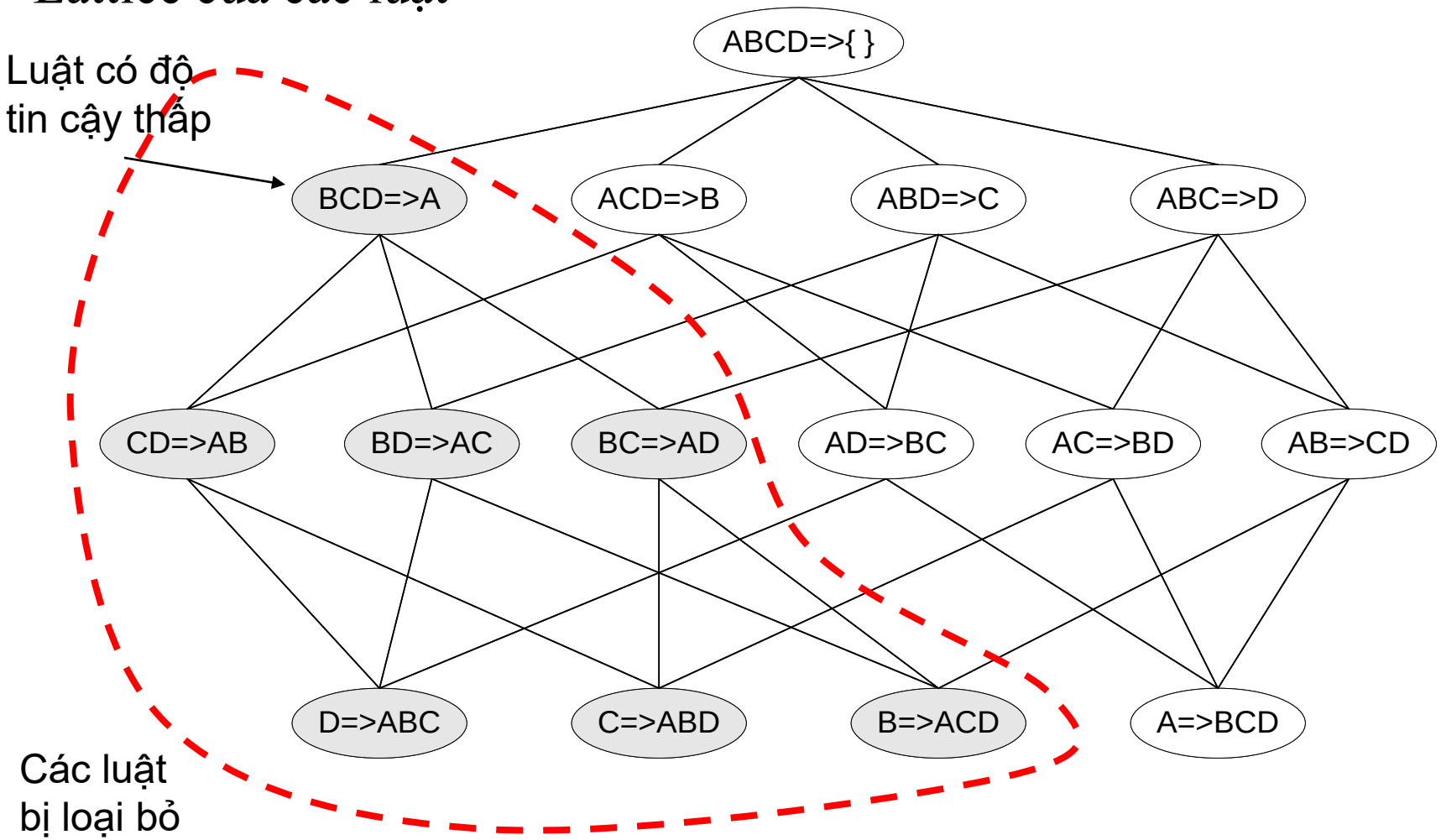
- Làm thế nào để sinh ra các luật từ các tập mục thường xuyên, một cách có hiệu quả?
- Xét tổng quát, **độ tin cậy không có đặc tính không đơn điệu (anti-monotone)**

  $c(ABC \rightarrow D)$ có thể lớn hơn hoặc nhỏ hơn $c(AB \rightarrow D)$
- Nhưng, **độ tin cậy của các luật được sinh ra từ cùng một tập mục thường xuyên thì lại có đặc tính không đơn điệu**
  - Ví dụ: Với L = {A,B,C,D}:

    $c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$
  - Độ tin cậy có đặc tính không đơn điệu đối với số lượng các mục ở vế phải của luật

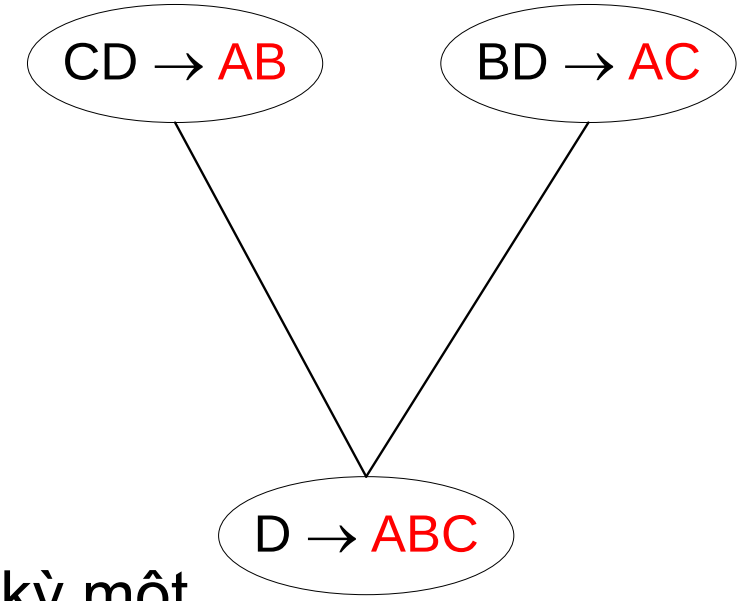
Apriori: Sinh ra các luật (1)

Lattice của các luật



Apriori: Sinh ra các luật (2)

- Các luật cần xét được sinh ra bằng cách kết hợp 2 luật có cùng tiền tố (phần bắt đầu) của phần kết luận (rule consequent)



- Ví dụ: Kết hợp 2 luật
  (CD → AB, BD → AC)
  sẽ sinh ra luật cần xét
  D → ABC

- Loại bỏ luật D → ABC nếu bất kỳ một luật con của nó (AD → BC, BCD → A, …) không có độ tin cậy cao (< *minconf*)

Tài liệu tham khảo

- P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining (chapter 6)*. Addison-Wesley, 2005.