Data Mining

Apprentissage et Fouille de données

Dr. Sana Hamdi sana.hamdi@fst.utm.tn

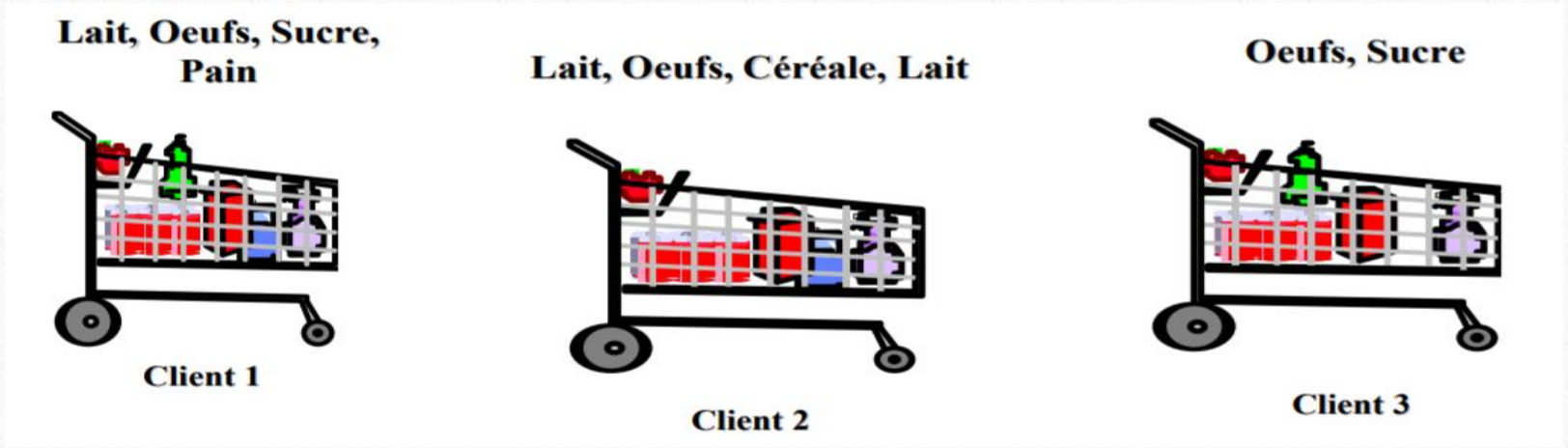
Les règles d’association

Recherche d'associations

- **Règles d'association** : consiste à déterminer les items qui sont associées.
  - **Motifs de la forme :** prémisse ⇒ conclusion
  - **Exemple type** : Détermination des articles qui se retrouvent ensemble sur un même ticket de supermarché (achète(x, "fromage") ⇒ achète(x, "pain"))
    - ✓ Intéressant pour identifier des opportunités de vente croisée et concevoir des groupements attractifs de produit.
    - ✓ Nécessité de très grands jeux de données.

Exemple: Analyse du panier de la ménagère

- La base de données a pour tuples les consommateurs et chaque tuple est un ensemble d'items.



- Déterminer des groupes d'items qui sont **fréquemment** achetés simultanément

Recherche d'associations

Numéro de la transaction	Contenu de caddie		
1	P1	P2	P3
2	P1	P3	
3	P1	P2	P3
4	P1	P3	
5	P2	P3	
6	P4		

Dès qu'on a des données binaires, il est possible de construire des **règles d'association.**

Transaction	P1	P2	P3	P4
1	1	1	1	0
2	1	0	1	0
3	1	1	1	0
4	1	0	1	0
5	0	1	1	0
6	0	0	0	1

Recherche d'associations

Observation	Taille	Corpulence
1	petit	mince
2	grand	enveloppé
3	grand	mince

Dès qu'on a des données binaires, il est possible de construire des **règles d'association.**

Observation	Taille = petit	Taille = grand	Corpulence = mince	Corpulence = enveloppé
1	1	0	1	0
2	0	1	0	1
3	0	1	1	0

Recherche d'associations

Objectifs:

- Mettre en évidence les produits achetés ensemble.
- Transcrire la connaissance sous forme de règle d'association.

***Si** antécédent **Alors** conséquent*

liste des produits

Transaction	P1	P2	P3	P4
1	1	1	1	0
2	1	0	1	0
3	1	1	1	0
4	1	0	1	0
5	0	1	1	0
6	0	0	0	1

Mesures: Support et Confiance

- **Support:** un indicateur de fiabilité de la règle (Si X alors Y): la proportion de transactions qui contiennent à la fois X et Y

$$\text{Support } (X \Rightarrow Y) = P(X \text{ et } Y)$$

- **Confiance:** un indicateur de précision de la règle (Si X alors Y): la proportion de transactions contenant X, contiennent aussi Y

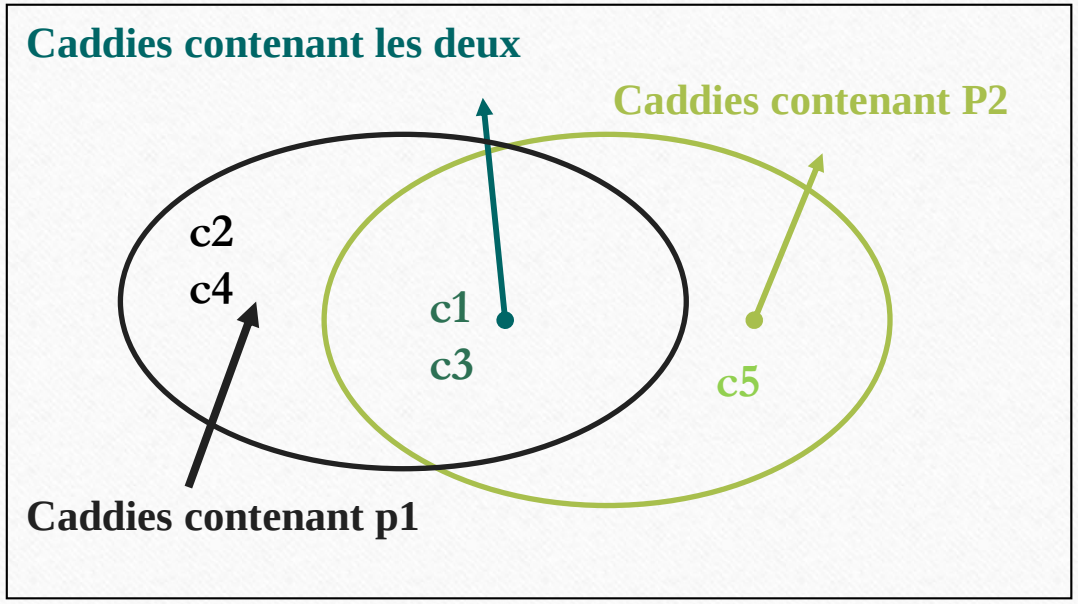
$$\text{Confiance } (X \Rightarrow Y) = P(Y \mid X) = P(X \text{ et } Y) \,/\, P(X)$$

$$= \text{Support } (X \Rightarrow Y) \,/\, \text{Support } (X)$$

- Bonne règle possède un support et confiance élevés.

Mesures: Support et Confiance

- Soit la règle d'association **R1: Si p1 alors p2**



Caddie	P1	P2	P3	P4
c1	1	1	1	0
c2	1	0	1	0
c3	1	1	1	0
c4	1	0	1	0
c5	0	1	1	0
c6	0	0	0	1

Mesures: Support et Confiance

- Soit la règle d'association **R1: Si p1 alors p2**

  - **Support(R1) = 2/6 = 33%**
  - **Confiance(R1) = (2/6) /(4/6) = 50%**

Caddie	P1	P2	P3	P4
c1	1	1	1	0
c2	1	0	1	0
c3	1	1	1	0
c4	1	0	1	0
c5	0	1	1	0
c6	0	0	0	1

Recherche d'associations: Notions utiles

- **Objectif:** trouver toutes les règles associatives, respectant MinSup et MinConf
- **Seuils minimaux de support et confiance donnés par l'utilisateur:**
  - MinSup
  - MinConf
- **Décomposition du problème:**
  1. Détermination des itemsets fréquents (support ≥ MinSup)
  2. Génération des règles associatives (confiance ≥ MinConf)

Recherche d'associations

- MinSup
  - Elevé ⇒ peu d'itemsets fréquents
    ⇒ peu de règles valides qui ont été souvent vérifiées
  - Réduit ⇒ plusieurs règles valides qui ont été rarement vérifiées
- MinConf
  - Elevée ⇒ peu de règles, mais toutes « pratiquement » correctes
  - Réduite ⇒ plusieurs règles, plusieurs d'entre elles sont « incertaines »
- Valeurs utilisées: MinSup 2 % -10 %, MinConf = 70 % - 90 %

Approche 1: 1.Extraction des ensembles fréquents

- **D**: une base de transactions
- **I**: ensemble de tous les items avec |I|=n

***Algorithme 1:* Extraction des ensembles fréquents**

```
Fréquents = ∅
   Pour chaque ensemble J ⊆ I Faire
      count(J)=0
      Pour chaque transaction t∈D Faire
         Si J ⊆ t.items Alors
            count(J)= count(J)+1
       Si count(J) ≥ min_support Alors
          Fréquents += J
   Fin
```

Approche 1: 2.Extraction des règles

- **D**: une base de transactions
- **I**: ensemble de tous les items avec |I|=n

***Algorithme 2:* Extraction des règles**

```
Règles = ∅
Pour chaque J dans Fréquents
  Pour chaque règle r: s ⇒ J\s  avec s ⊂ J, s≠∅
     Si confiance(r) ≥ min_confiance Alors
        Règles += r
Fin
```

Approche1: Estimation du coût

Algorithme 1: très coûteux.

- Pour chaque ensemble d'items J, il faut parcourir la base D pour compter le nombre de ses occurrences.
- Si n est le nombre d'items, il existe 2^n ensembles J → 2^n parcours de D
- Accès successif à la base de données → Nombre de calcul énorme

Algorithme 2: On peut supposer que l'ensemble « Fréquents » réside en mémoire et count(J) est une information déjà calculée par algorithme 1 → pas d'accès à D

→ **Coût global: 2^n parcours de D**

Algorithme Apriori (Agrawal et Srikant, 1994)

Principe : Si un ensemble est non fréquent, alors tous ses sur-ensembles (super-set) ne sont pas fréquents, i.e.,

- Si **{ AB }** est fréquent alors {A} et {B} le sont
- Si **{A}** n'est pas fréquent alors {AB} ne peut pas l'être

- Itérativement, trouver les itemsets fréquents dont la cardinalité varie de 1 à k (k-itemset)
- Utiliser les itemsets fréquents pour générer les règles d'association

Algorithme Apriori (Agrawal et Srikant, 1994)

- Étape de jointure: C_{k+1} est généré en joignant F_k avec lui même **% *F_k : itemset fréquent de taille k (k-itemset)***
- Étape d'élagage: Chaque (k)-itemset qui n'est pas fréquent ne peut être un sous ensemble d'un (k+1)-itemset fréquent

F_1 = {items fréquents};

for (k = 1; F_k !=∅; k++) do

C_{k+1} = candidats générés à partir de F_k ***% jointure de F_k***

for each transaction *t* dans la base do

incrémenter le* COUNT *des candidats de C_{k+1} qui sont dans t

F_{k+1} = candidats dans C_{k+1} dont COUNT > support_min

end for

end for

return $\cup_k F_k$

Génération de candidats: étape de jointure

- **Self-join de F_k :**

insert into C_{k+1}

select p[1], p[2], …, p[k], q[k] **from** p, q (appartenant à F_k)

where p[1] = q[1], …, p[k − 1] = q[k − 1], p[k] < q[k]

- **Exemple:**
- si F_3 = {{123}, {124}, {134}, {135}, {234}}

→ la phase de jointure donne comme résultat C_4 = {{1234}, {1345}}

Génération de candidats: élagage de C_k

- La deuxième phase élagage de C_k : effacer les éléments qui ne vérifient pas la propriété des sous ensemble fréquents.

Pour chaque *itemset c dans C_k Faire*

Pour chaque *(k-1)-sous-ensemble s de c* Faire

Si *(s n'est pas dans F_{k-1})* Alors supprimer c de C_k

Exemple: si F_3 = {{123}, {124}, {134}, {135}, {234}}

→ la phase joindre donne comme résultat C4 = {{1234}, {1345}}

→ ensuite la phase d'élagage donne le résultat : C4 = {{1234} car l'élément {145} n'est pas dans F_3 et donc {1345} est effacé.

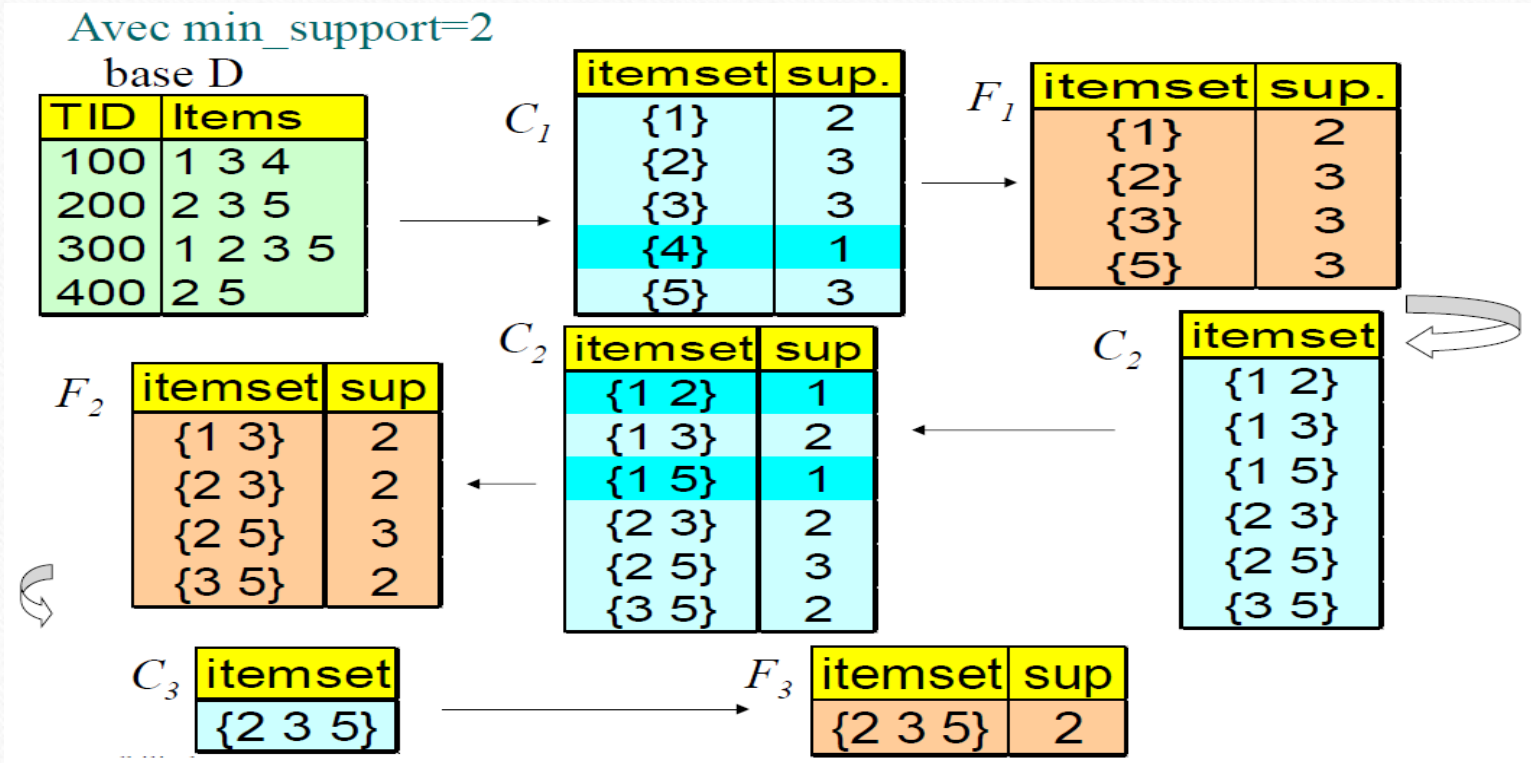
Exemple1: Extraction des ensembles fréquents

- Avec une valeur de MinSup =2, appliquer l'algorithme "apriori" pour déterminer les ensembles fréquents à partir des transactions suivantes:

base D

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

Exemple1: Extraction des ensembles fréquents



Génération des règles d'association

Algorithme :

pour chaque item**set** fréquent f

  pour chaque sous ensemble s ⊂ f, avec s≠∅

    si confiance(s ⇒ f\s) > min_conf **alors**

      afficher(s ⇒ f\s)

  fin pour

fin pour

confiance (X ⇒Y)
=
support (X ⇒Y) / support (X)

Exemple 1: Génération des règles d'association

MinConf =75%

F_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

F_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

F_3

itemset	sup
{2 3 5}	2

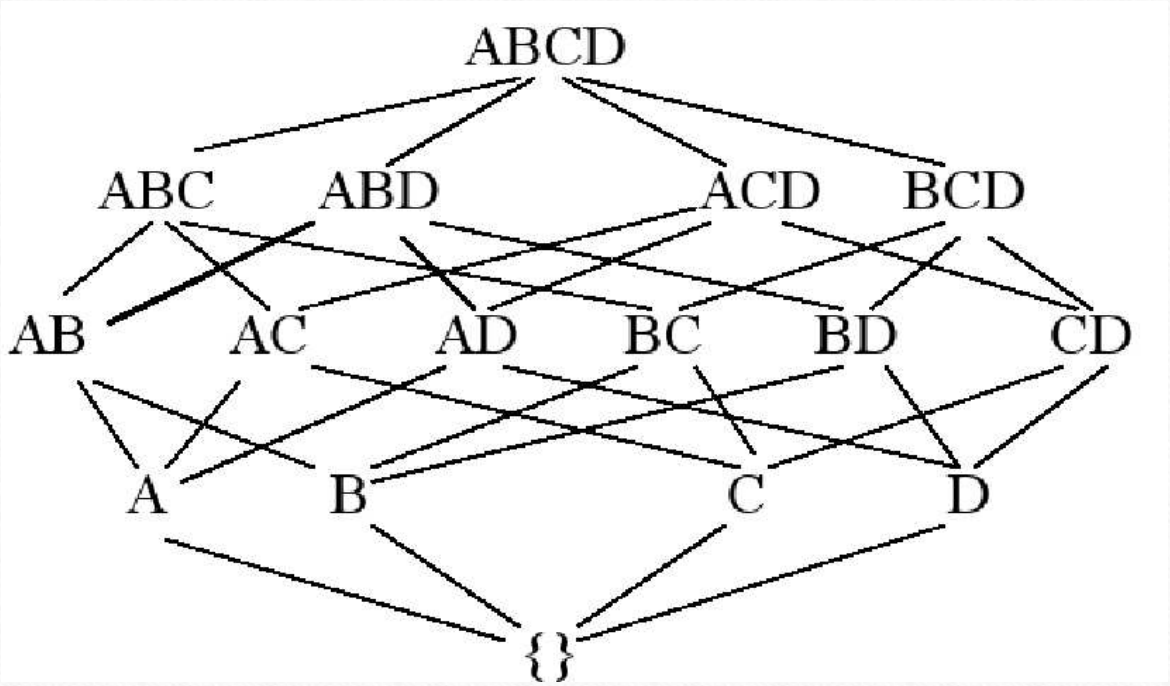
Règles	Conf.
1➡3	100%
3➡1	66%
2➡3	66%
3➡2	66%
2➡5	100%
5➡2	100%
3➡5	66%
5➡3	66%

Règles	Conf.
2,3➡5	100%
2,5➡3	66%
3,5➡2	100%
2➡3,5	66%
3➡2,5	66%
5➡2,3	66%

Apriori: Exemple2

MinSup = 2

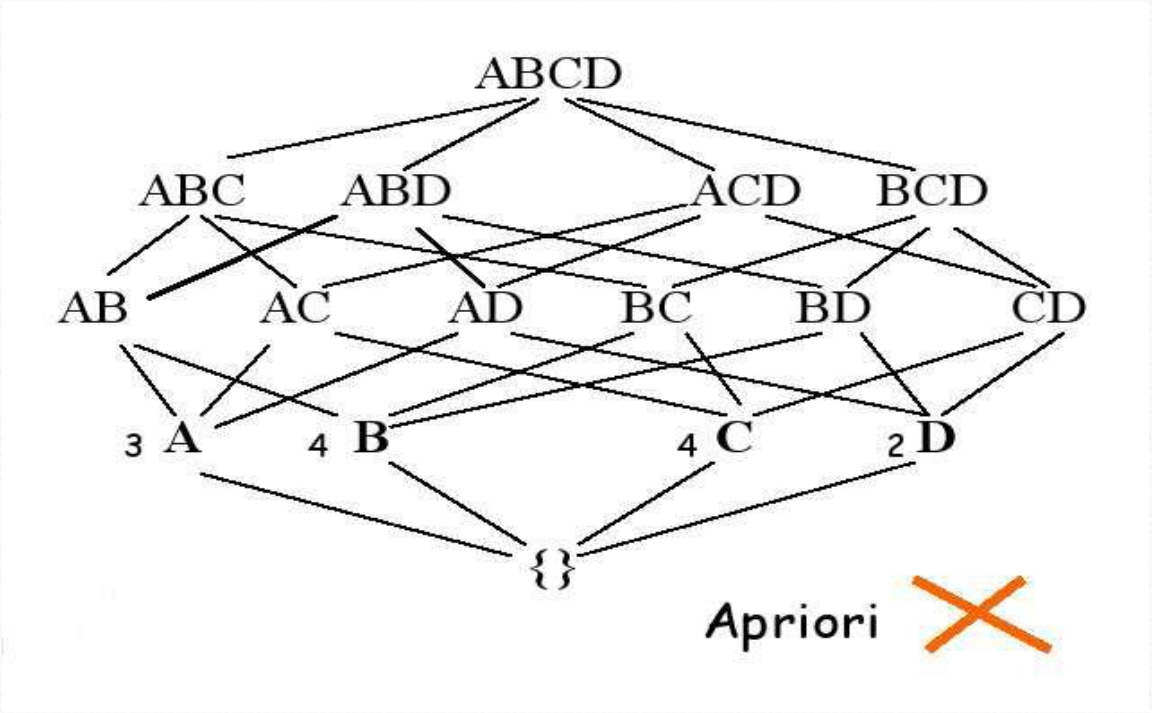
A	B	C	D
1	0	1	0
1	1	1	0
0	1	1	1
0	1	0	1
1	1	1	0



Apriori: Exemple2

MinSup = 2

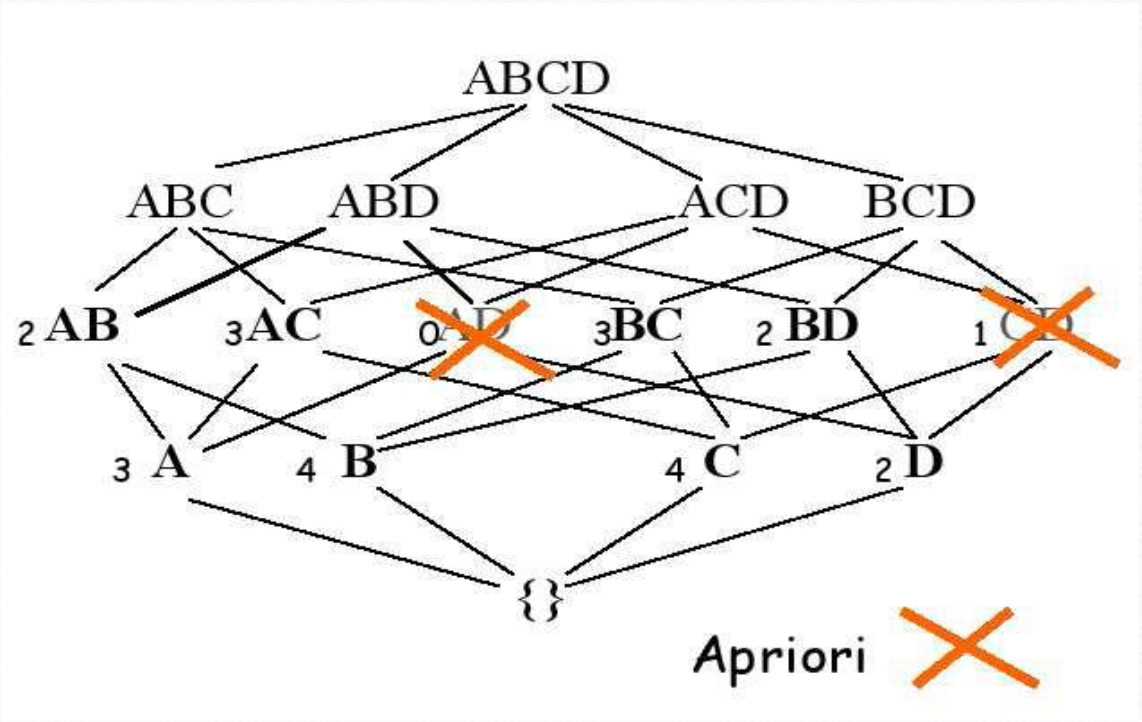
A	B	C	D
1	0	1	0
1	1	1	0
0	1	1	1
0	1	0	1
1	1	1	0



Apriori: Exemple2

MinSup = 2

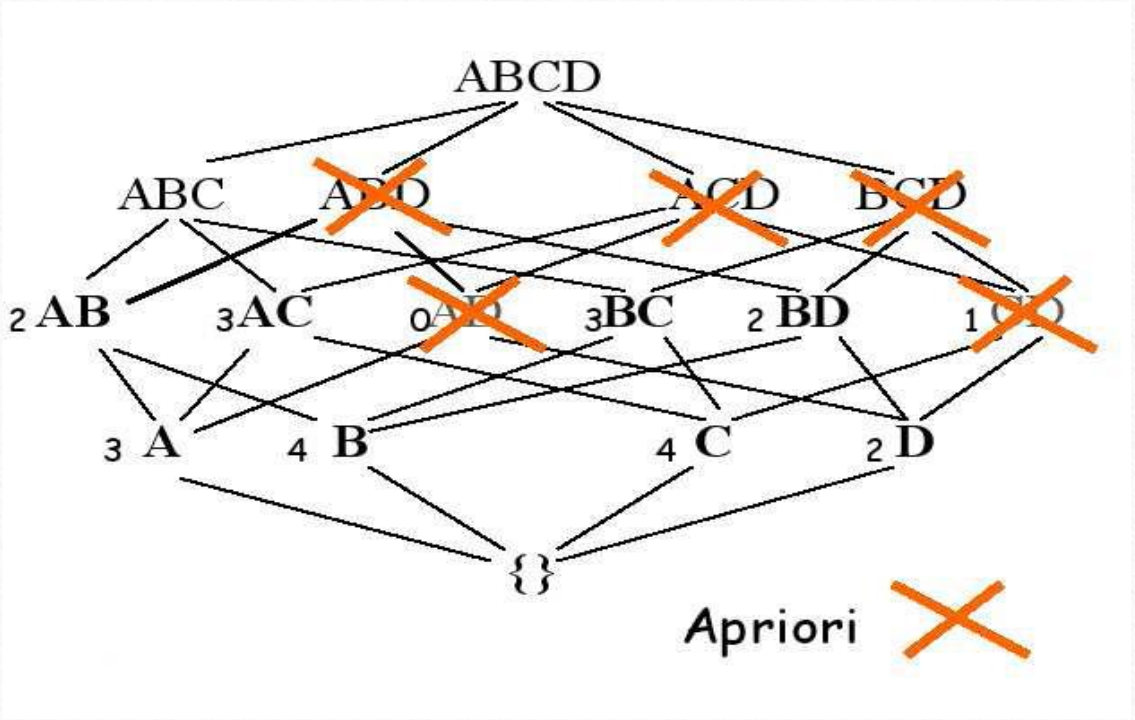
A	B	C	D
1	0	1	0
1	1	1	0
0	1	1	1
0	1	0	1
1	1	1	0



Apriori: Exemple2

MinSup = 2

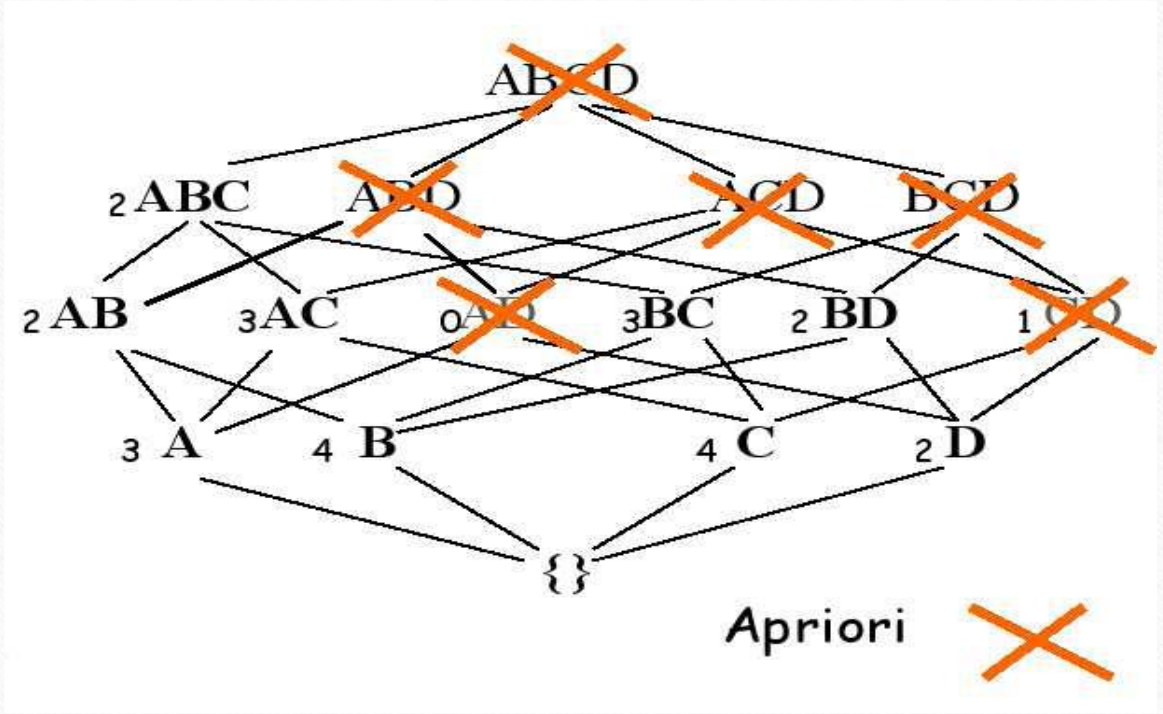
A	B	C	D
1	0	1	0
1	1	1	0
0	1	1	1
0	1	0	1
1	1	1	0



Apriori: Exemple2

MinSup = 2

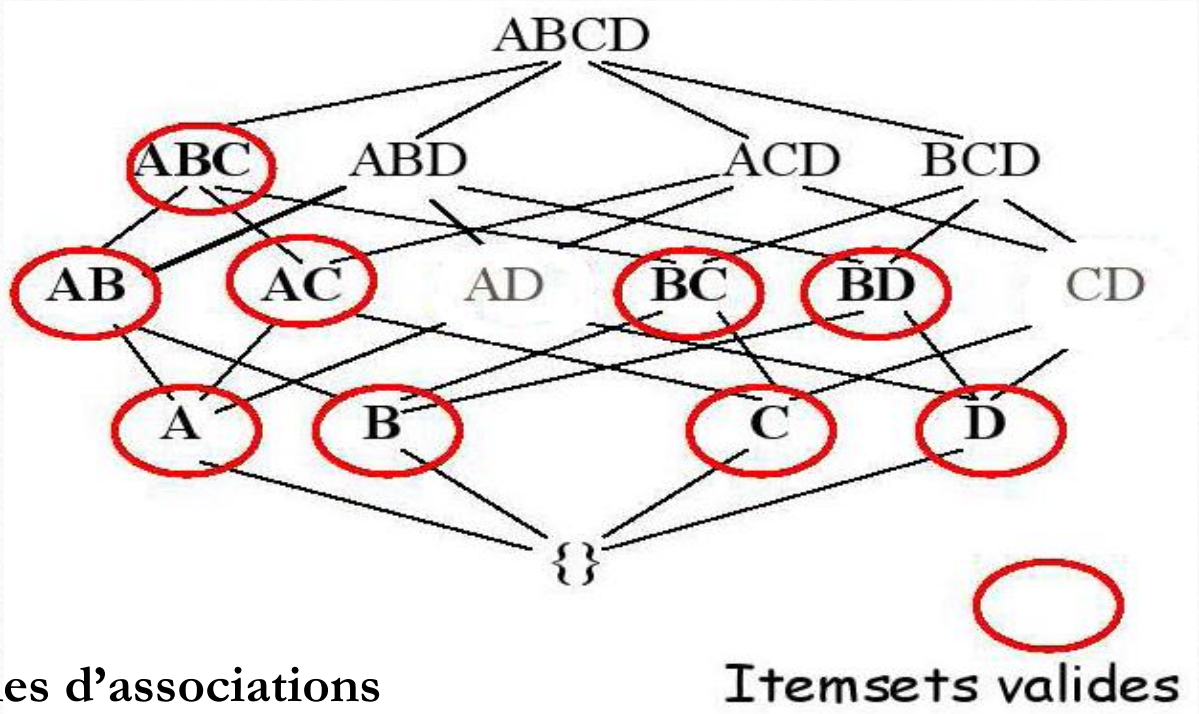
A	B	C	D
1	0	1	0
1	1	1	0
0	1	1	1
0	1	0	1
1	1	1	0



Apriori: Exemple2

MinSup = 2

A	B	C	D
1	0	1	0
1	1	1	0
0	1	1	1
0	1	0	1
1	1	1	0



Soit MinConf = 60%, donnez les règles d'associations

Exercice: relations entre les mots

- On cherche des relations entre les mots d'un corpus sous forme de règles:

  Si les mots $M_1, \ldots, M_m$ apparaissent dans un texte **alors**

  les mots $M_{m+1}, \ldots, M_n$ apparaissent aussi dans le texte

- Dans le tableau suivant, nous avons 4 textes indexés par les mots clés A, B, C et D.

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

Exercice: relations entre les mots

- Appliquer l'algorithme "apriori" pour déterminer les règles d'association valides avec un **SupMin = 40%** et **ConfMin = 50%**

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

Exercice: relations entre les mots

- Avec **SupMin** = 40% et **ConfMin** = 50%: les deux règles valides sont :

  $D \Rightarrow B$ et $B \Rightarrow D$.

- Avec **ConfMin** = 100%:

  seule la règle: $D \Rightarrow B$ est valide

Apport d'information d'une règle

- Parfois une règle peut avoir d'excellents support et confiance, MAIS, sans être autant « intéressante »

Exemple:

X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

- X et Y positivement corrélés,
- X et Z négativement corrélés
- Les support et confiance de X ⇒ Z dominent
- Nous avons besoin d'une mesure de corrélation $Corr_{(A,B)} = P(A \cap B) / P(A).P(B)$

Apport d'information d'une règle

- Cette mesure est appelé: Intérêt (corrélation) ou aussi le Lift:

  $$\text{Lift}(A \Rightarrow B) = \text{confiance}(A \Rightarrow B) / \text{Support}(B) = P(A \cap B) / P(A).P(B)$$

  - ✓ prendre en compte P(A) et P (B)
  - ✓ Si Lift =1, alors la règle ne sert absolument à rien: A et B sont indépendants:

    $$P(A \cap B) = P(A) \cdot P(B) \text{ , } P(A), P(B) > 0$$

  - ✓ Si Lift $(A \Rightarrow B) < 1$, alors A et B sont négativement corrélés
  - ✓ Si Lift $(A \Rightarrow B) > 1$, alors A et B sont positivement corrélés (A est susceptible d'être acheté avec B)

Apport d'information d'une règle

X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

Itemset	Support	Intérêt
X,Y	25%	2
X,Z	37,50%	0,9
Y,Z	12,50%	0,57