REGLE d’ASSOCIATION

Mme Leila HAMDAD
ESI

- Les règles d’association est une tâche descriptive du Data Mining.
- Trouver des associations entre attributs (features) ou items d’objets dans la base de données.
- S’applique à des données en forme d’une table individus-variables:

✓ Les variables continues sont discrétisées .

✓ les variables catégorielles sont mises sous. forme disjonctive complète

Domaines d'applications:

- Marketing,
- Diagnostic médical,
- Bioinformatique,……

En plus détaillé:

- Les Réseaux de télécommunication: filtrage des alarmes non informatives et l'identification des causes d'anomalies [Pasquier, 2000a].
- Le Multi-média et internet:

✓ facilitent l'aide à la navigation dans les systèmes de gestion d'information,

✓ la recherche et la sélection des sites intéressants,

✓ l'aide à l'organisation des sites et ressources par l'historique des accès des usagers [Pasquier, 2000a].

- Les règles d'Association ont été introduite par Agrawal et al en 1993.
- Problème classifié NP-difficile[Jourdan, 2003].
- Plusieurs méthodes abordent dans la littérature l'extraction des RA en un temps minimal.

Mesures de qualité des règles

- Le support: indique l'importance statistique des règles d' association dans l'ensemble de données.
- La confiance: indique les règles crédible.

- Les bonnes règles sont celles avec un grand **support** et une grande **confiance.**

Exemple

- Soient les données de toutes les transactions d'achats d'un magasin sur une période donnée et $\boldsymbol{I = \{i_1, i_2, \ldots, i_N\}}$ l'ensemble de tous les articles vendu dans ce magasin. Pour faire de **meilleurs profits**, il serait important de connaitre les articles qui **s'achètent ensembles** (qui s'associe), par exemple : souvent les clients qui achètent une télévision achètent aussi un magnétoscope. Une telle information peut être utilisée pour faire des promotions ou pour localiser ces articles de manière plus efficace. Elles sont appelées : « règles d'association ».

- Soient I = $\{i_1, i_2, \ldots, i_N\}$ un ensemble d'items. Et X et Y deux sous-ensembles de I.
- Une règle d'association est de la forme :

  « **X => Y »,**avec $X \cap Y = \varnothing$.

- On dit que la règle d'association **« X => Y »** possède une confiance c , si c% de toute les transactions d'achat qui contiennent X contiennent aussi Y :

$$Confiance, c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

On dit que la règle d'association **« X => Y »** possède un support s, si s% de toute les transactions d'achat contiennent à la fois X et Y :

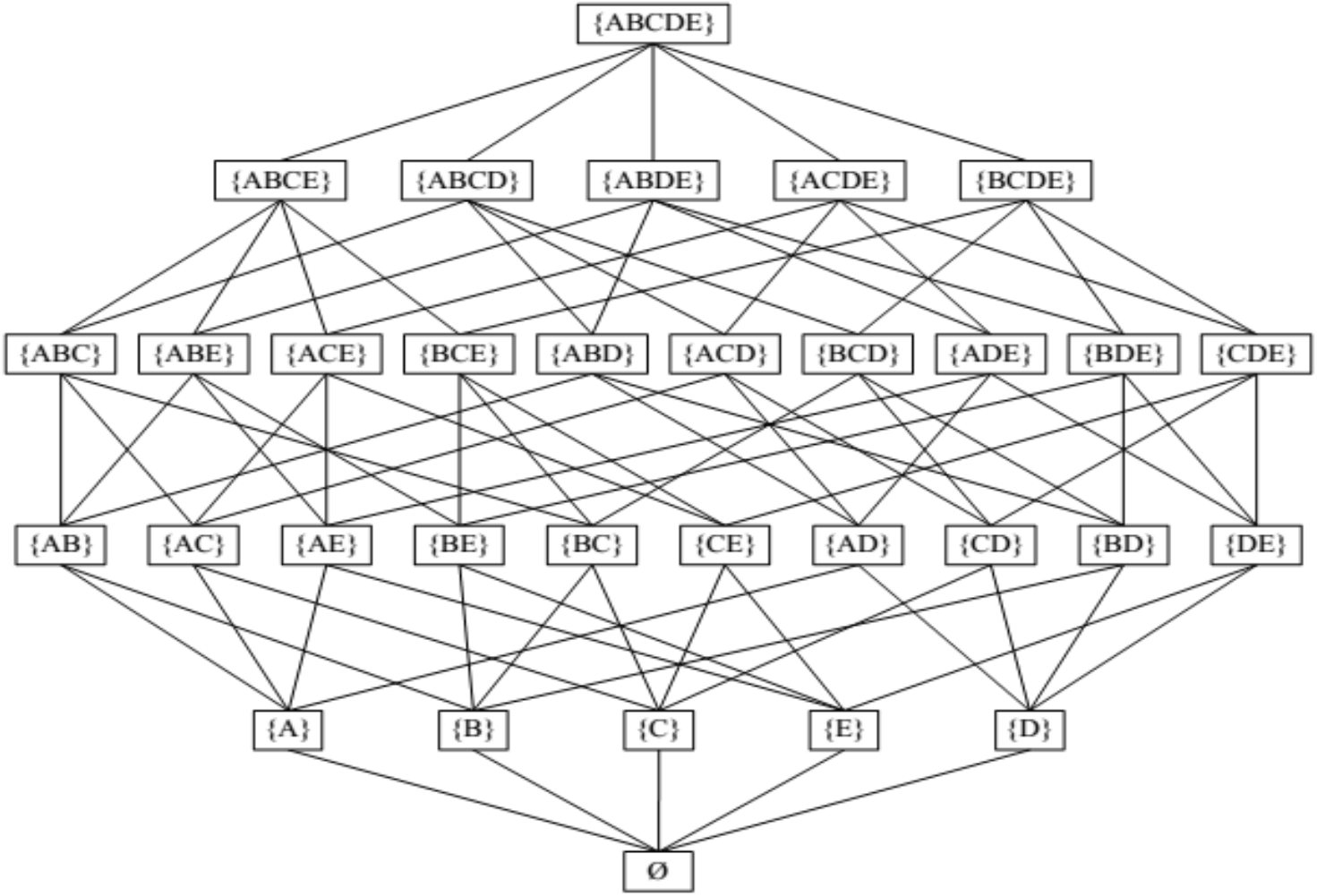
$$Support, s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$$

- Lift: mesure l'amélioration apportée par la règle d'association par rapport à un jeu de transactions aléatoire (où X et Y seraient indépendants). Il est défini par:

$$Lift = \frac{Supp(X \cup Y)}{Supp(X)Supp(Y)}$$

- Un « *lift* » supérieur à 1 traduit une corrélation positive de X et Y → Caractère significatif de l'association.

Treillis des itemsets



Algorithmes classiques

- Ces algorithmes donnent toutes les règles dont le support et la confiance sont supérieurs à des seuils **minsup** et **minconf** respectivement choisis par l'utilisateur.
- Obtenir les K- item frequents à partir des (K-1)- item frequents.
- Calculer le support de l'ensemble candidat dans la base de données scannée.

Phase1: Extraction des itemsets fréquents

- Donner tout les itemsets (ensemble d'items) ayant un support supérieur ou égal à minsup après la génération de tout les itemsets candidats.
- *Ils sont appelés* **itemsets fréquents**
- Cette phase est très couteuse en temps d'exécutions [Agrawal et al.1994].

Phase2: **Génération des RA**

La génération de Règle d'Association valide à partir d'itemsets fréquents**:**

- Les règles sont de la forme:
    - «Si **prémisse** (X) alors **conclusion**(Y)».

Tel que $X \cap Y = \varnothing$.

X est aussi appelé antécédent de la règle et Y la conséquences. Se sont deux itemsets fréquents disjoints.

- Retenir celles dont la confiance est supérieure ou égale à minconf .

Le premier algorithme appelé **Apriori** a été introduit par Agrawal et al en 1993.

Avantages:

-Simple et donne des résultats claire et aucune hypothèse n'est supposée à priori.

-Inconvénients

-Couteux en Temps d'exécution (Jourdan2003).

- Produit beaucoup de règles redondantes et pas nécessaires [Blanchard, 2005] [Pasquier, 2000a].

Plus formellement, l'algorithme a priori se déroule comme suit:

- L'ensemble des 1-itemsets candidats est construit à partir de tout les items. Cet ensemble est évalué pour former l'ensemble des 1-itemsets fréquents (support > minsup).
- A chaque itération k (k démarre à 2), un scanne est effectué pour construire l'ensemble des k-itemsets candidats en regroupant deux ensembles de k-1-itemsets fréquents, et le support est calculé.

- Apriori supprime tous k-itemset non fréquents et ses sur-ensembles (lorsqu'un itemset vérifie une condition prédéfinie alors tous ses sous-ensembles la vérifient aussi) afin d'aboutir à un ensemble de k-itemsets fréquents.

- La deuxième phase consiste à générer des règles à partir des itemsets fréquents de la première phase.

Exemple d'exécution de l'algorithme APRIORI: MINSUP = 3/6 et MINCONF= 0,8.

TID	ITEMS			
01	A	C	D	
02	B	C	E	
03	A	B	C	E
04	B	E		
05	A	B	C	E
06	B	C	E	

Itemset	Support
{A}	3/6
{B}	5/6
{C}	5/6
{D}	1/6
{E}	5/6

C1

→

Itemset	Support
{A}	3/6
{B}	5/6
{C}	5/6
{E}	5/6

L1

Itemset	Support
{A B}	2/6
{A C}	3/6
{A E}	2/6
{B C}	4/6
{B E}	5/6
{C E}	4/6

C2

→

Itemset	Support
{AC}	3/6
{BC}	4/6
{BE}	5/6
{CE}	4/6

L2

Itemset	Support
{A B C}	2/6
{A C E}	2/6
{A B E}	2/6
{B C E}	4/6

C3

→

Itemset	Support
{B C E}	4/6

L3

Itemset	Support
{A B C E}	2/6

C4

Itemsets fréquent	Règle	Confiance	Règle prise
{A C}	A→C	3/3	Oui
	C→A	3/5	Non
{B C}	B→C	4/5	Oui
	C→B	4/5	Oui
{B E}	B→E	5/5	Oui
	E→B	5/5	Oui
{C E}	C→E	4/5	Oui
	E→C	4/5	Oui
{B C E}	BC→E	4/4	Oui
	BE→C	4/5	Oui
	EC→B	4/4	Oui

La méthode FPgrowth

- FPgrowth a été proposé en 2000 par Han et al.
- Permet d'éviter le processus coûteux de génération et de test des candidats, utilisé par Apriori.
- Pour conserver les itemsets fréquents dans la base de transactions cet algorithme utilise une structure de données compacte appelé Frequent-Pattern tree.

❑Avantages:

○ Les éléments sont triés: accélère la recherche des règles d'association.

○ Il suffit de suivre les liens inter-noeuds pour connaître toutes les associations fréquentes.

Principe

1. Construction de la structure FP-tree

- Balayer la base des transactions pour créer la liste des items fréquents avec leur support
- Trier cette liste en ordre décroissant de support

2. Exploitation récursive du FP-tree

Pour chaque item fréquent :

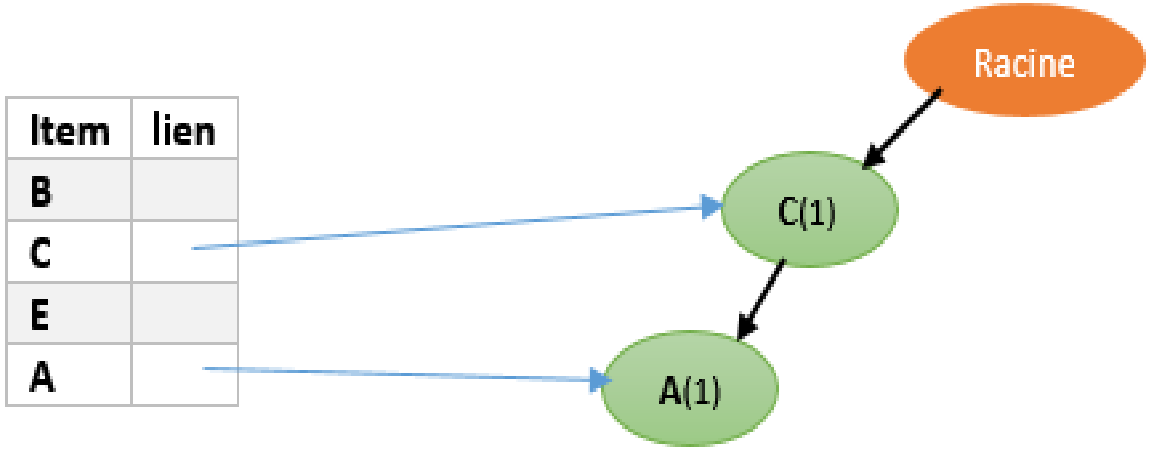
- Construire les chemins préfixes dans le FP-tree
- Fusionner les préfixes identiques et conserver les sous-chemins de support >= seuil
- Générer les ensembles fréquents par combinaison des nœuds des chemins fréquents

Item	support
A	3/6
B	5/6
C	5/6
D	1/6
E	5/6

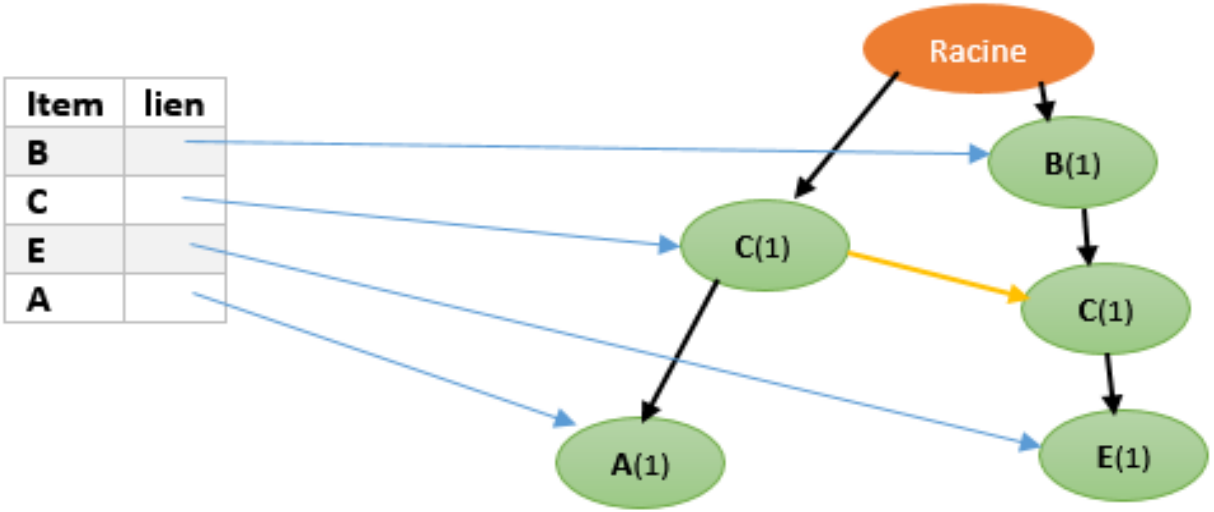
→

Item	support
B	5/6
C	5/6
E	5/6
A	3/6

TID	Items	Items fréquents ordonnés
01	A C D	C A
02	B C E	B C E
03	A B C E	B C E A
04	B E	B E
05	A B C E	B C E A
06	B C E	B C E

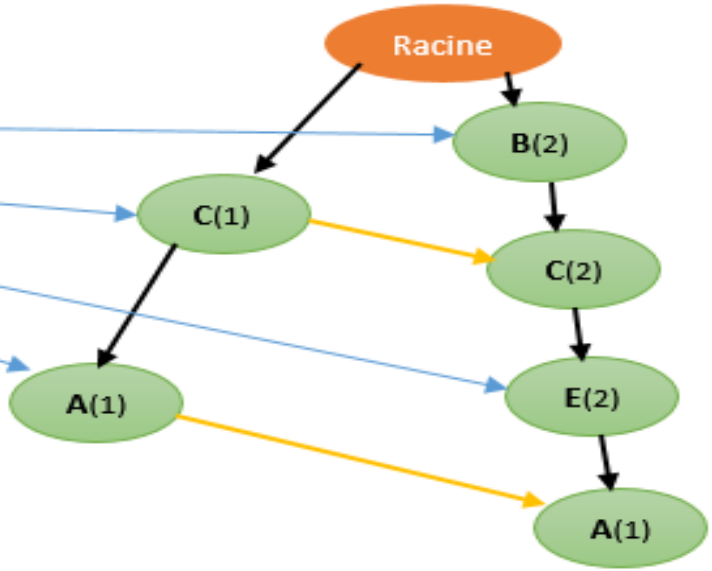


Transaction « 02 »



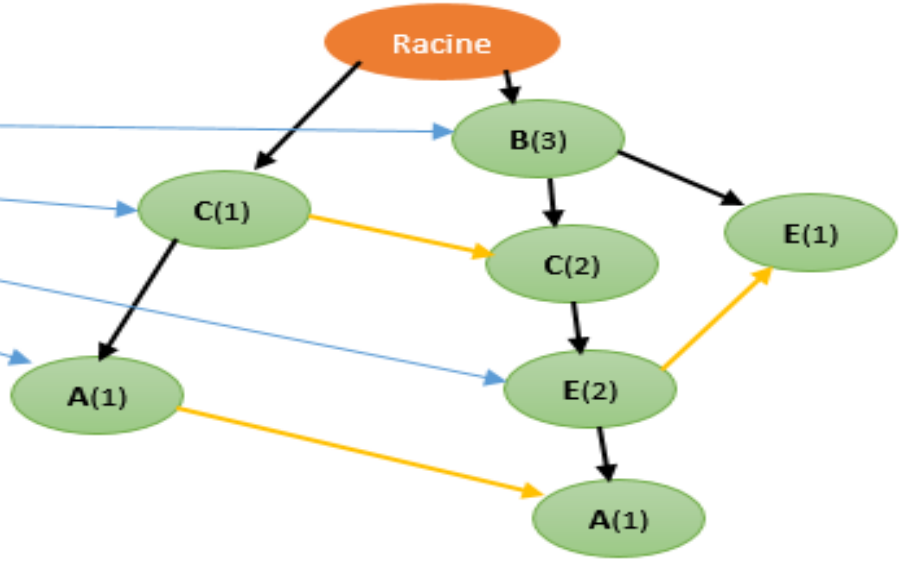
Transaction « 03 »

Item	lien
B	
C	
E	
A	



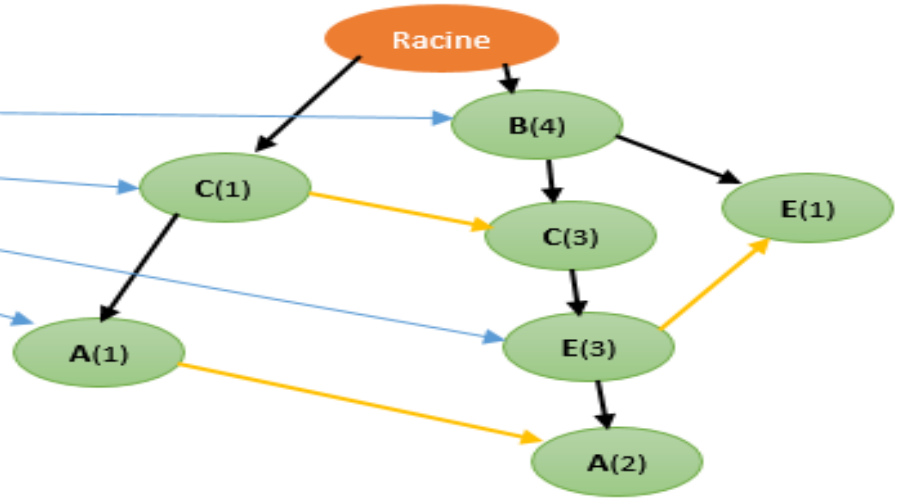
Transaction « 04 »

Item	lien
B	
C	
E	
A	



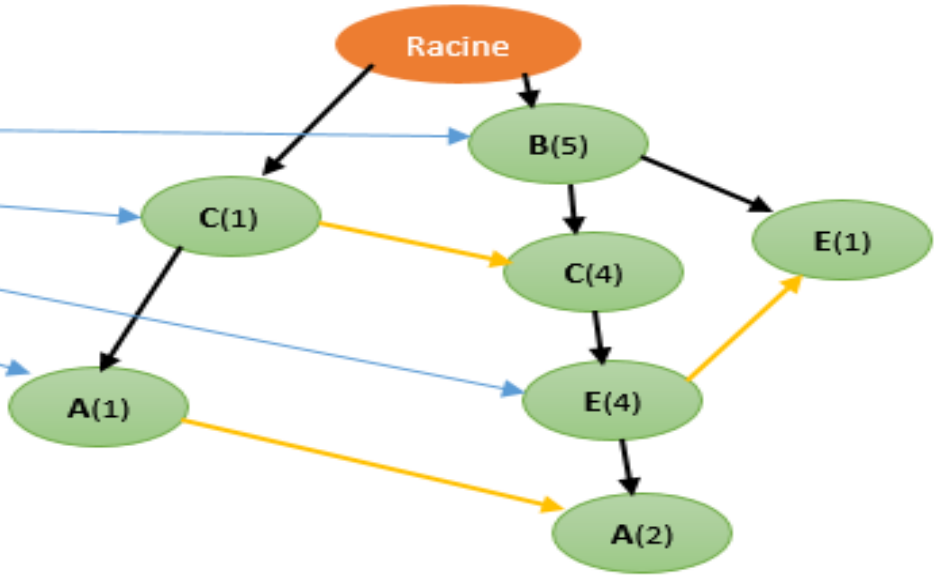
Transaction « 05 »

Item	lien
B	
C	
E	
A	



Transaction « 06 »

Item	lien
B	
C	
E	
A	



Item	Somme Compteur	Support
B	5	5/6
C	5	5/6
E	5	5/6
A	3	3/6