

**Université Mohamed Boudiaf – M'sila**  
**Département Informatique**  
**Master Intelligence Artificielle**  
**2020-2021**

**Chapitre 5**  
**Règles d'association**

Dr. Mehenni Tahar

# Règles d'Association

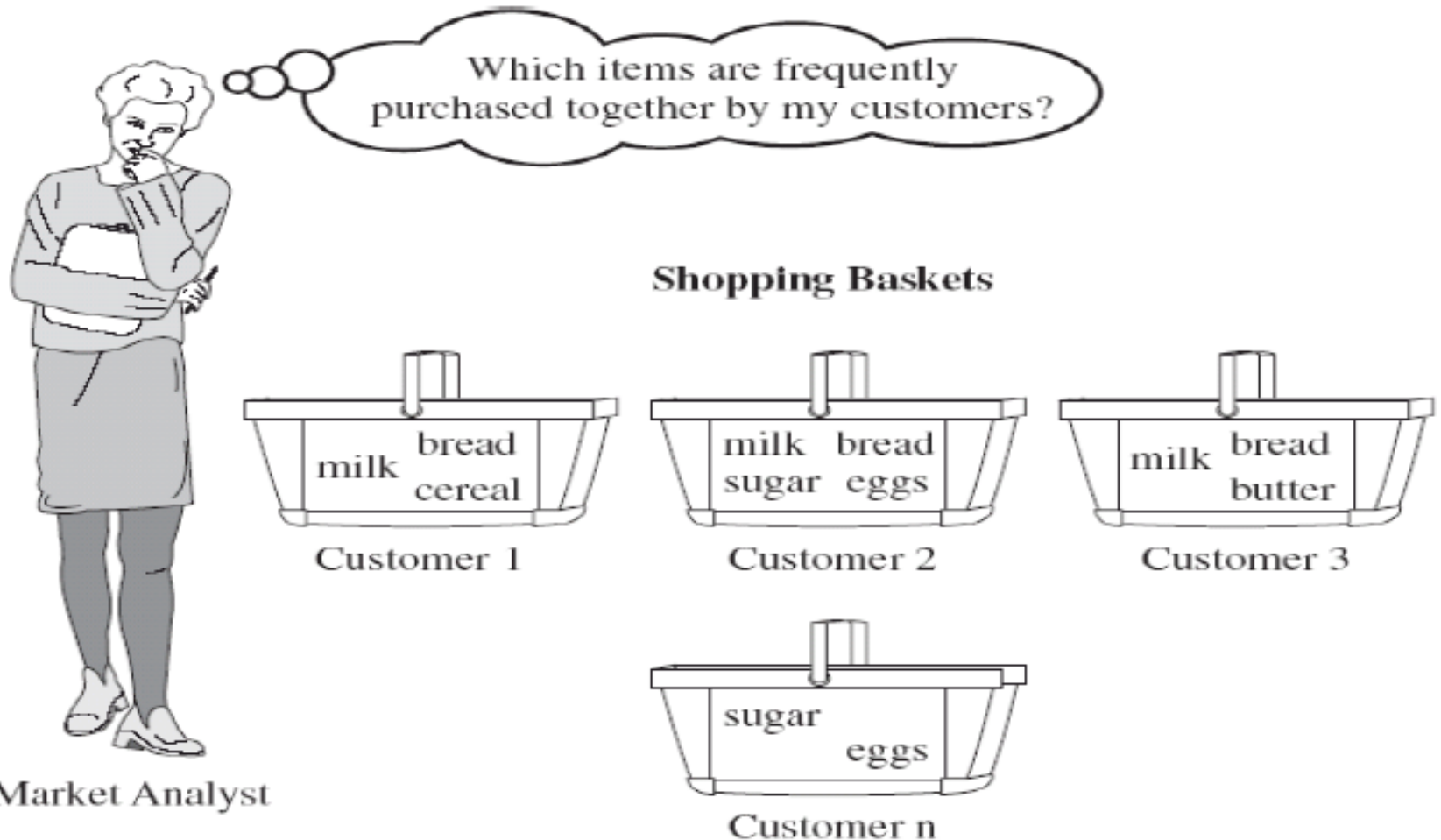
Une application habituelle est l'*analyse du panier de consommation (market basket analysis)* où

- (1) les objets (items) sont fréquemment vendus ensemble au supermarché
- (2) les items ayant même promotion sont arrangés ensemble.



# Découverte des Règles d'Association

## Market basket analysis



# Association Rule Discovery

- Given a set of records each of which contain some number of items from a given collection;
  - Produce dependency rules which will predict occurrence of an item based on occurrences of other items.



<i>TID</i>	<i>Items</i>
1	Bread, Coke, Milk
2	Butter, Bread
3	Butter, Coke, Diaper, Milk
4	Butter, Bread, Diaper, Milk
5	Coke, Diaper, Milk

## Rules Discovered:

**{Milk} --> {Coke}**

**{Diaper, Milk} --> {Butter}**

# Association Rule Discovery

Market basket:

Rule form: "Body  $\rightarrow$  Head [support, confidence]"

$\text{buys}(X, \text{'butter'}) \rightarrow \text{buys}(X, \text{'snacks'})$   
[1%, 60%]

- (a) If a customer X purchased 'butter', 60% of them purchased 'snacks'
- (b) 1% of all transactions contain the items 'butter' and 'snacks' together



# Notation

$\#(b_1, b_2, \dots, b_k)$ : pourcentage des transactions contenant  $(b_1, b_2, \dots, b_k)$  par rapport au nombre total de transactions.

**Support** :

$$\#(a_1, a_2, \dots, a_n, b_1, \dots, b_k)$$

**Confiance** :

$$\frac{\#(a_1, a_2, \dots, a_n, b_1, \dots, b_k)}{\#(a_1, a_2, \dots, a_n)}$$

**Amélioration** :

$$\frac{\text{confiance}(a_1, a_2, \dots, a_n, b_1, \dots, b_k)}{\#(b_1, \dots, b_k)}$$

# Algorithme A-priori

- Algorithme permettant de trouver *toutes* les règles d'association pour :
  - Un support minimal fixé.
  - Une confiance minimale fixée.
- **Exemple:**
- Trouver toutes les règles s'appliquant à au moins 30% des transactions, et ayant une confiance supérieure à 80%

# Définitions

- ***itemset*** : ensemble d'articles.
- ***k-itemset*** : ensemble contenant exactement  $k$  articles.
- ***itemset fréquent*** : itemset dont la fréquence dans la liste des transactions est supérieure au seuil fixé par le support minimal.



# L'algorithme A-Priori

## Deux phases :

1. Trouver tous les itemsets fréquents.
2. A partir de ces itemsets, trouver toutes les règles ayant une confiance suffisante.

## Trouver les itemsets fréquents

- Compter combien de fois chaque sous-ensemble d'articles apparaît dans la liste des transactions ?
- Impossible :  $n$  articles, donc  $2^n$  sous-ensembles à explorer.
- $n$  très grand.
- La plupart des sous-ensembles ne sont jamais présents dans aucune transaction.

# Propriété A-Priori

## ***Propriété:***

*Pour qu'un  $k$ -itemset soit fréquent, il faut que tous les  $(k-1)$ -itemsets qu'il contient soient aussi fréquents*

## **Preuve :**

- Soit  $(a_1, a_2, \dots, a_{n-1})$  un  $(n-1)$ -itemset non fréquent.
- Sa fréquence est donc inférieure au seuil.
- Le  $n$ -itemset  $((a_1, a_2, \dots, a_{n-1}, a_n))$  est au plus aussi fréquent que  $(a_1, a_2, \dots, a_{n-1})$
- Il n'est donc pas fréquent.

# Etapes de l'Algorithme

1. Créer l'ensemble de tous les 1-itemsets fréquents.  
( $k=1$ )
2. Tant que de nouveaux  $k$ -itemsets ont été créés faire :
  - Créer tous les candidats  $(k+1)$ -itemsets à partir des  $k$ -itemsets fréquents (*union d'ensembles*)
  - Parcourir la liste des transactions pour éliminer les  $(k+1)$ -itemsets non fréquents.
3. *construire les règles*

# Exemple détaillé

Cinq articles :  $l_1, l_2, \dots, l_5$

Neuf transactions.

**Support minimal : 2/9**

**1-itemsets fréquents**

Article	Frequence
$l_1$	6
$l_2$	7
$l_3$	6
$l_4$	2
$l_5$	2

$t_1$	$l_1, l_2, l_5$
$t_2$	$l_2, l_4$
$t_3$	$l_2, l_3$
$t_4$	$l_1, l_2, l_4$
$t_5$	$l_1, l_3$
$t_6$	$l_2, l_3$
$t_7$	$l_1, l_3$
$t_8$	$l_1, l_2, l_3, l_5$
$t_9$	$l_1, l_2, l_3$

**Candidats 2-itemsets**

On construit tous les couples de 1-itemsets fréquents  $(l_i, l_j)$

$i \neq j$ : il y en a 10.

	$l_1$	$l_2$	$l_3$	$l_4$	$l_5$
$l_1$	X	4	4	1	2
$l_2$	X	X	4	2	2
$l_3$	X	X	X	0	1
$l_4$	X	X	X	X	0
$l_5$	X	X	X	X	X

# Exemple détaillé

## 3-itemsets candidats

- Construire les candidats à partir des 2-itemsets fréquents.
- Si les 2-itemsets sont dans l'ordre lexicographique, il suffit de considérer les couples de 2-itemsets ne différant que par le *dernier article*

	$l_1, l_2$	$l_1, l_3$	$l_1, l_5$	$l_2, l_3$	$l_2, l_4$	$l_2, l_5$
$l_1, l_2$	X	ok	ok	deja vu	non	deja vu
$l_1, l_3$	X	X	deja vu	deja vu	X	X
$l_1, l_5$	X	X	X	X	X	deja vu
$l_2, l_3$	X	X	X	X	non	non
$l_2, l_4$	X	X	X	X	X	non

## 4-itemsets

Un seul candidat :  $(l_1, l_2, l_3, l_5)$ , mais  $(l_2, l_3, l_5)$  n'est pas fréquent (i.e. n'a pas été gardé à l'étape précédente. . .)

Fin de la recherche des itemsets fréquents.

## On a finalement :

- Six 2-itemsets fréquents.
- Deux 3-itemsets fréquents

# Construire des règles

**Principe :** A partir d'un k-itemset fréquent :

- Construire toutes les règles de la forme

SI  $a_{\gamma(1)} \wedge \dots \wedge a_{\gamma(i)}$  ALORS  $a_{\gamma(i+1)} \wedge \dots \wedge a_{\gamma(k)}$

où  $\gamma$  est une permutation et  $1 \leq i < k$

- Calculer la confiance de chacune de ces règles.
- Garder les meilleures ou les classer.

# Construire des règles

- A partir des six 2-itemsets, 12 règles possibles

Couple	Regle	Confiance	Support
$l_1, l_2$	$l_1 \Rightarrow l_2$	$4/6$	$4/9$
$l_1, l_2$	$l_2 \Rightarrow l_1$	$4/7$	$4/9$
$l_2, l_4$	$l_2 \Rightarrow l_4$	$2/7$	$2/9$
$l_2, l_4$	$l_4 \Rightarrow l_2$	$2/2$	$2/9$
...	...	...	...
...	...	...	...

- A partir des triplets  
(on ne fait que  $(l_1, l_2, l_5)$ )

Regle	Confiance
$l_1 \wedge l_2 \Rightarrow l_5$	$2/4$
$l_1 \wedge l_5 \Rightarrow l_2$	$2/2$
$l_2 \wedge l_5 \Rightarrow l_1$	$2/2$
$l_1 \Rightarrow l_2 \wedge l_5$	$2/6$

## Remarques

1. Si on crée un k-itemset, il faut parcourir la liste des transactions k fois (pour vérifier la fréquence des candidats).
2. Mais : les itemsets fréquents deviennent de plus en plus rares.