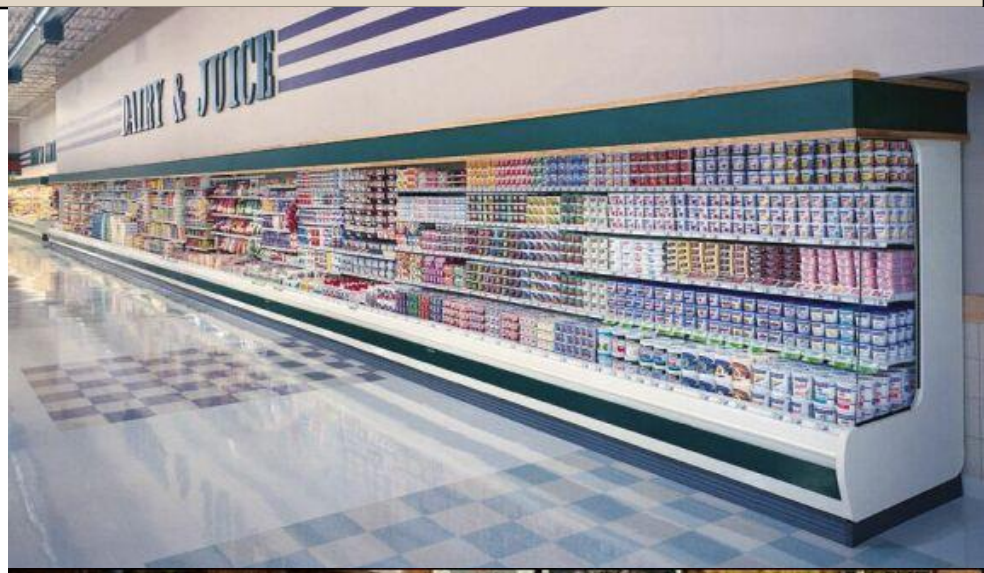
Introduction au Data Mining

Règles d'Association

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

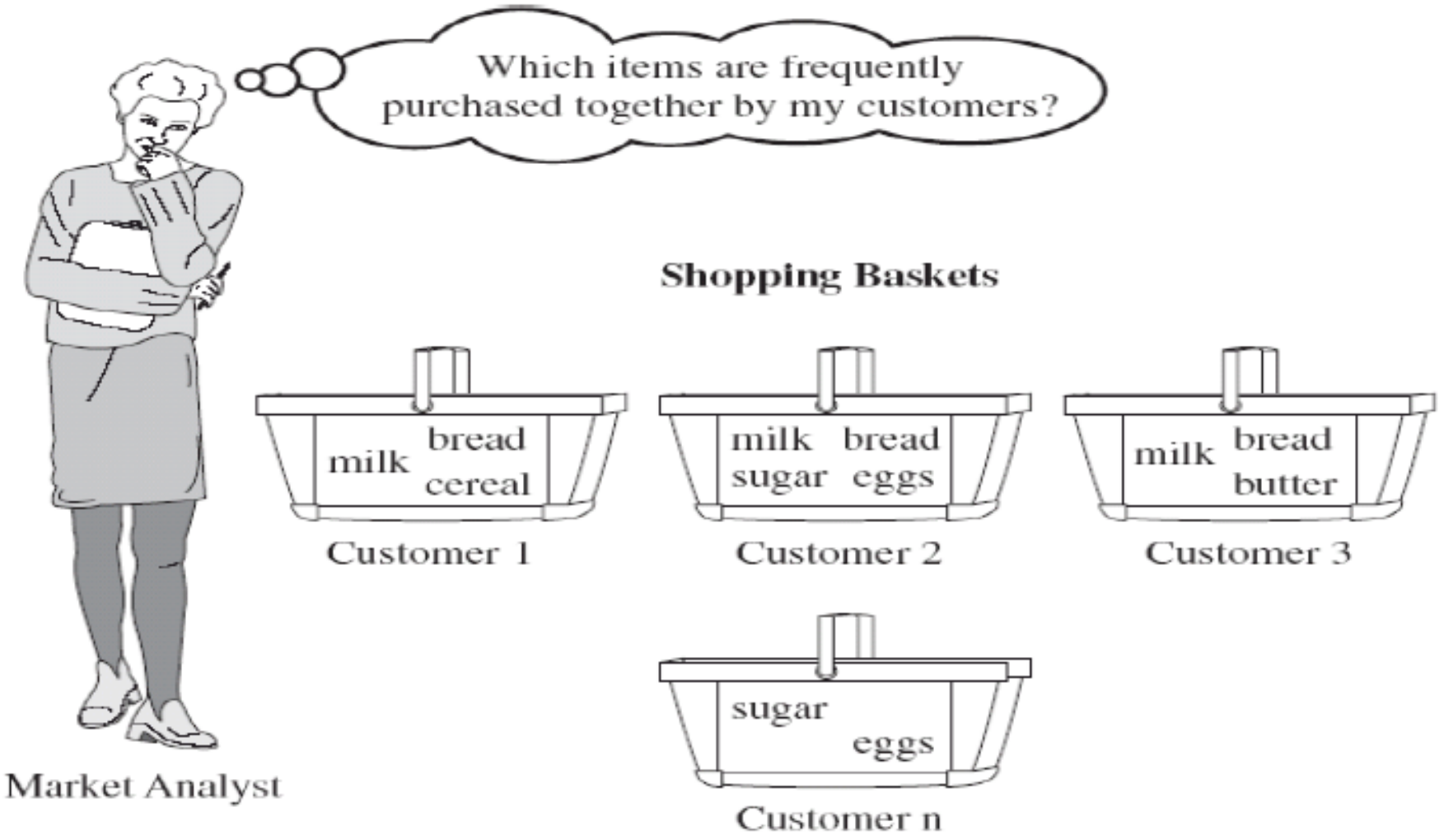
(1) les objets (items) sont fréquemment vendus ensemble au supermarché

(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.**

TID	*Items*
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 → Head [support, confidence]".

buys(X, `butter') → buys(X, "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, … ,b_k): pourcentage des transactions contenant (b_1, b_2, … ,b_k) par rapport au nombre total de transactions.

Support :

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

Confiance :

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

Amélioration :

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

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%

Définition 1

- ***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.*

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

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

Preuve :

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

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 : *I1, I2, …, I5*

Neuf transactions.

Support minimal : 2/9

t1	I1, I2, I5
t2	I2, I4
t3	I2, I3
t4	I1, I2, I4
t5	I1, I3
t6	I2, I3
t7	I1, I3
t8	I1 ,I2 ,I3,I5
t9	I1 ,I2 ,I3

1-itemsets fréquents

Article	Frequence
I1	6
I2	7
I3	6
I4	2
I5	2

Candidats 2-itemsets

- On construit tous les couples de 1-itemsets fréquents (I_i, I_j)

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

	I1	I2	I3	I4	I5
I1	X	4	4	1	2
I2	X	X	4	2	2
I3	X	X	X	0	1
I4	X	X	X	X	0
I5	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*

	I1, I2	I1, I3	I1, I5	I2, I3	I2, I4	I2, I5
I1, I2	X	ok	ok	deja vu	non	deja vu
I1, I3	X	X	deja vu	deja vu	X	X
I1, I5	X	X	X	X	X	deja vu
I2, I3	X	X	X	X	non	non
I2, I4	X	X	X	X	X	non

4-itemsets

Un seul candidat : (*I1, I2, I3, I5),m*ais (*I2, I3, I5)* 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 $a_{\gamma(1)} \wedge \ldots a_{\gamma(i)}$ ALORS $a_{\gamma(i+1)} \wedge \ldots a_{\gamma(k)}$

où γ 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
I1, I2	I1 ⇒ I2	4/6	4/9
I1, I2	I2 ⇒ I1	4/7	4/9
I2, I4	I2 ⇒ I4	2/7	2/9
I2, I4	I4 ⇒ I2	2/2	2/9
...	...	...	...
...	...	...	...

- A partir des triplets

(on ne fait que (*I1, I2, I5*)

Regle	Confiance
I1 ^ I2 ⇒ I5	2/4
I1 ^ I5 ⇒ I2	2/2
I2 ^ I5 ⇒ I1	2/2
I1 ⇒ I2 ^ I5	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.