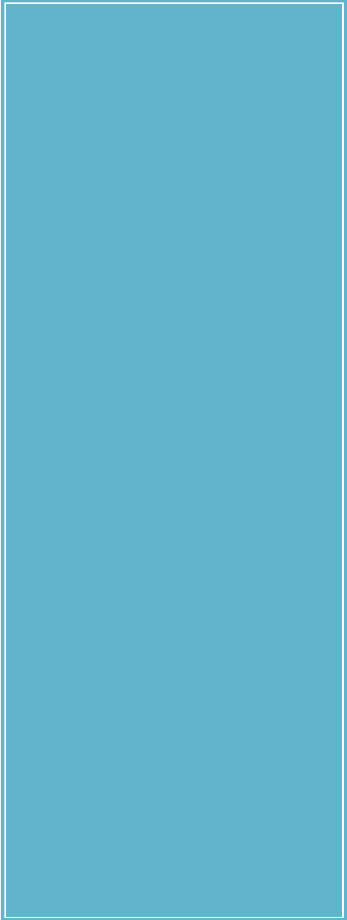


CHAPITRE III :

FOUILLES DE DONNÉES

2. Technique de structuration

Sommaire

- 
- A. **Rappel**
 - B. **Introduction**
 - C. **Méthodes utilisées**
 - D. **Conclusion**



A. Rappel

A. Rappel

Techniques descriptives		Techniques prédictives		
Corrélation simple	Corrélation complexe	Présent		Futur
		Variable cible numérique	Variable cible catégorielle	
1 : Description	2 : Classification	Estimation	Segmentation	Prévision
	3 : Association	4		

A. Rappel

Structuration / Classification

1.

Description :

trouver un résumé des données qui soit plus intelligible

- statistique descriptive
- analyse factorielle

Ex : moyenne d'âge des personnes présentant un cancer du sein

2.

Structuration :

Faire ressurgir des groupes « naturels » qui représentent des entités particulières

- clustering, apprentissage non-supervisé

->Classification

Ex : découvrir une typologie de comportement des clients d'un magasin

4.

Explication :

Prédire les valeurs d'un attribut (endogène) à partir d'autres attributs (exogènes)

- Régression/Estimation et classement/Segmentation
- **apprentissage supervisé**

Ex : prédire la qualité d'un client (rembourse ou non son crédit) en fonction de ses caractéristiques (revenus, statut marital, nombre d'enfants, etc.)

3.

Association :

Trouver les ensembles de descripteurs qui sont le plus corrélés

- **règles d'association**

Ex : rayonnage de magasins, les personnes qui achètent du poivre achètent également du sel

Méthodes de Data Mining

Techniques du Datamining

- **Présentation des techniques**
- **Description de chaque technique :**
 - ✓ **Mesures de Similarités & Types Variables**
 - ✓ **Classification/Structuration /Clustering**
 - ✓ **Association**
 - ✓ **Estimation & Segmentation & Prévision**

Techniques du Datamining

- **Présentation des techniques**
- **Description de chaque technique :**
 - ✓ **Mesures de Similarités & Types Variables**
 - ✓ **Classification/Structuration /Clustering**
 - ✓ **Association**
 - ✓ **Estimation & Segmentation & Prévision**

B. Mésures de Similarités & Types Variables

- Intervalles:
- Binaires:
- catégories, ordinales, ratio:
- Différents types:

Intervalle (discrètes)

- Standardiser les données
 - Calculer l'écart absolu moyen:



où



- Calculer la mesure standardisée (z-score)

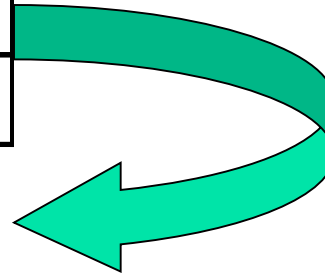
$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

Intervalle (discrètes)

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

$$M_{Age} = 6 \quad S_{Age} = 5$$

~~$M_{Age} = 6$ $S_{Age} = 5$~~



	Age	Salaire
Personne1	-2	-0,5
Personne2	2	0,175
Personne3	0	0,324
Personne4	0	2

Similarité entre objets

- Les distances expriment une similarité
- Ex: la *distance de Minkowski* :



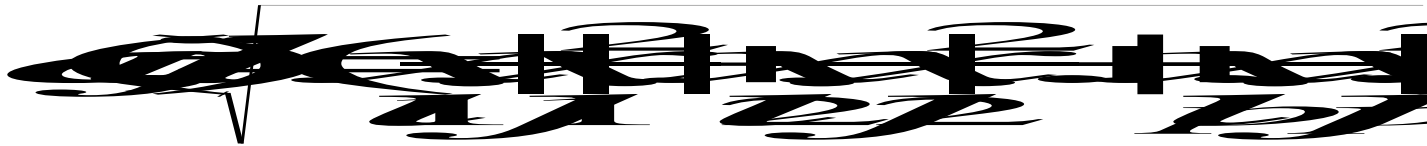
où $i = (x_{i1}, x_{i2}, \dots, x_{ip})$ et $j = (x_{j1}, x_{j2}, \dots, x_{jp})$ sont deux objets p -dimensionnels et q un entier positif

- Si $q = 1$, d est la distance de Manhattan



Similarité entre objets

- Si $q = 2$, d est la distance Euclidienne :



- Propriétés

- $d(i,j) \geq 0$
- $d(i,i) = 0$
- $d(i,j) = d(j,i)$
- $d(i,j) \leq d(i,k) + d(k,j)$

Exemple: distance de Manhattan

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

$$d(p1, p2) = 120$$

$$d(p1, p3) = 132$$

Conclusion: p1 ressemble plus à p2 qu'à p3 ☹

	Age	Salaire
Personne1	-2	-0,5
Personne2	2	0,175
Personne3	0	0,324
Personne4	0	0

$$d(p1, p2) = 4,675$$

$$d(p1, p3) = 2,324$$

Conclusion: p1 ressemble plus à p3 qu'à p2 ☺

Variables binaires

- Une table de contingence pour données binaires

		Objet j		
		1	0	
Objet i	1	a	b	$a+b$
	0	c	d	$c+d$
		$a+c$	$b+d$	p

a = nombre de positions
où i a 1 et j a 1

- Exemple $o_i = (1, 1, 0, 1, 0)$ et $o_j = (1, 0, 0, 0, 1)$

$a=1, b=2, c=1, d=1$

Variables binaires

- Coefficient d'appariement (matching) simple (invariant pour variables symétriques):

$$d = \frac{b + c}{a + b + c}$$

Exemple $o_i = (1, 1, 0, 1, 0)$ et $o_j = (1, 0, 0, 0, 1)$

$$d(o_i, o_j) = 3/5$$

- Coefficient de Jaccard

$$d(o_i, o_j) = 3/4$$

$$d = \frac{b}{a + b}$$

Variables binaires

- Variable symétrique: Ex. le sexe d'une personne, i.e coder masculin par 1 et féminin par 0 c'est pareil que le codage inverse
- Variable asymétrique: Ex. Test HIV. Le test peut être positif ou négatif (0 ou 1) mais il y a une valeur qui sera plus présente que l'autre. Généralement, on code par 1 la modalité la moins fréquente
 - 2 personnes ayant la valeur 1 pour le test sont *plus similaires* que 2 personnes ayant 0 pour le test

Variables binaires

Exemple

Nm	Sexe	Fêve	Toux	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- Sexe est un attribut symétrique
- Les autres attributs sont asymétriques
- Y et P \equiv 1, N \equiv 0, la distance n'est mesurée que sur les asymétriques

$$d_{jackmary} = \frac{0+1}{2+0+1} = 0.33$$

$$d_{jackjim} = \frac{1+1}{1+1+1} = 0.67$$

$$d_{jimmary} = \frac{1+2}{1+1+2} = 0.75$$

Les plus similaires sont Jack et Mary \Rightarrow atteints du même mal

Variables Nominales

- Une généralisation des variables binaires, ex: rouge, vert et bleu
- Méthode 1: Matching simple
 - m : # d'appariements, p : # total de variables

$$\frac{m}{p}$$

- Méthode 2: utiliser un grand nombre de variables binaires
 - Créer une variable binaire pour chaque modalité (ex: variable rouge qui prend les valeurs vrai ou faux)

Variables Ordinales

- Une variable ordinale peut être discrète ou continue
- L'ordre peut être important, ex: classement
- Peuvent être traitées comme les variables intervalles
 - remplacer x_{if} par son rang
 - Remplacer le rang de chaque variable par une valeur dans $[0, 1]$ en remplaçant la variable f dans l'objet I par

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- Utiliser une distance pour calculer la similarité

Techniques du Datamining

- **Présentation des techniques**
- **Description de chaque technique :**
 - ✓ **Mesures de Similarités & Types Variables**
 - ✓ **Classification/Structuration /Clustering**
 - ✓ **Association**
 - ✓ **Estimation & Segmentation & Prévision**

C. Introduction

- Classification/Structuration /Clustering

C. Introduction

□ Problématique :

- Par exemple, on souhaite regrouper une clientèle (14 clients) en 4 classes
 - Probabilité : 10 millions de partitions possibles en 4 classes
- Avec un gros calculateur :
 - examiner toutes les possibilités (10 millions)
 - et ne retenir que la meilleure

} **Impossible !**

□ D'où le besoin d'algorithmes

C. Introduction

□ **Structuration = Classification = “Clustering”**

□ **Définition :**

▣ Consiste à créer des classes (\approx sous-ensembles) de données:

■ similaires entre elles

■ et différentes des données d'une autre classe

➔ L'intersection des classes entre elles doit toujours être vide

▣ Définit les grands types de regroupement et de distinction: on parle de **métatypologie** (type de type)

- Partitionnement logique de la base de données en clusters
- **Clusters** : groupes d'instances ayant les mêmes caractéristiques
- ☐ classes inconnues (*Apprentissage non supervisé*)

C. Introduction

□ **Intérêts :**

- Favoriser la compréhension et la prédiction
- Fixer des segments qui serviront d'ensemble de départ pour des analyses approfondies.
- Réduire les dimensions, c'est-à-dire le nombre d'attributs/variables, quand il y en a trop au départ

□ **Applications :**

- Economie (segmentation de marchés)
- Médecine (localisation de tumeurs dans le cerveau)
- etc.

□ **Exemple :** Métatypologie d'une clientèle en fonction de :

→ l'âge, → les revenus, → le caractère urbain ou rural,
→ la taille des villes, → etc.

D. Méthodes utilisées

I. Classification Non hiérarchique

1. Agrégation autour de centres mobiles (*KMeans*)
2. Nuées Dynamiques (*Self organizing map*)

II. Classification Hiérarchique

1. Ascendante (*agglomerative*)

C. Méthodes utilisées

I. Classification Non hiérarchique

1. Agrégation autour de centres mobiles (*K-Means/K-moyennes*)
2. Nuées Dynamiques (*Self organizing map*)

II. Classification Hiérarchique

1. Ascendante (*agglomerative*)

C. Méthodes utilisées

I. Classification Non hiérarchique

1. Agrégation autour de centre mobiles (*K-Means/K-moyennes*)

- ❑ Définition
- ❑ Etapes
- ❑ L'algorithme de "K-Means"
- ❑ Synthèse

2. Nuées Dynamiques (*Self organizing map*)

II. Classification Hiérarchique

1. Ascendante (*agglomerative*)

C. Méthodes utilisées

I. Classification Non hiérarchique

1. Agrégation autour de centre mobiles (*K-Means/K-moyennes*)

- ❑ Définition
- ❑ Etapes
- ❑ L'algorithme de "K-Means"
- ❑ Synthèse

2. Nuées Dynamiques (*Self organizing map*)

II. Classification Hiérarchique

1. Ascendante (*agglomerative*)







Algorithme k voisins

30

- **Paramètre** : le nombre k de voisins
- **Donnée** : un échantillon de m exemples et leurs classes
 - ▣ La classe d'un exemple X est $c(X)$
- **Entrée** : un enregistrement Y
- 1. Déterminer les k plus proches exemples de Y en calculant les distances
- 2. Combiner les classes de ces k exemples en une classe c
- **Sortie** : la classe de Y est $c(Y)=c$

Exemple







31

Customer	Age	Income	No. credit cards	Loyal
John 	35	35K	3	No
Rachel 	22	50K	2	Yes
Hannah 	63	200K	1	No
Tom 	59	170K	1	No
Nellie 	25	40K	4	Yes
David 	37	50K	2	?

Example

32

$K = 3$

Customer	Age	Income	No. credit cards	Loyal
John 	35	35K	3	No
Rachel 	22	50K	2	Yes
Hannah 	63	200K	1	No
Tom 	59	170K	1	No
Nellie 	25	40K	4	Yes
David 	37	50K	2	Yes

Distance from David
$\text{sqrt} [(35-37)^2 + (35-50)^2 + (3-2)^2] = 15.16$
$\text{sqrt} [(22-37)^2 + (50-50)^2 + (2-2)^2] = 15$
$\text{sqrt} [(63-37)^2 + (200-50)^2 + (1-2)^2] = 152.23$
$\text{sqrt} [(59-37)^2 + (170-50)^2 + (1-2)^2] = 122$
$\text{sqrt} [(25-37)^2 + (40-50)^2 + (4-2)^2] = 15.74$

Définition de l'algorithme “K-Means”

- C'est la méthode la mieux adaptée aux très grands tableaux de données.
- On choisit une métrique pour calculer la distance entre individus.
- On définit à priori un nombre de classes (k).
- On choisit de façon arbitraire k centres de classes. C'est souvent k individus tirés au hasard.
- Les individus seront affectés au centre de classe le plus proche.

D. Méthodes utilisées

I. Classification Non hiérarchique

1. Agrégation autour de centre mobiles (*K-Means/K-moyennes*)

- ❑ Définition
- ❑ Étapes
- ❑ L'algorithme de "K-Means"
- ❑ Synthèse

2. Nuées Dynamiques (*Self organizing map*)

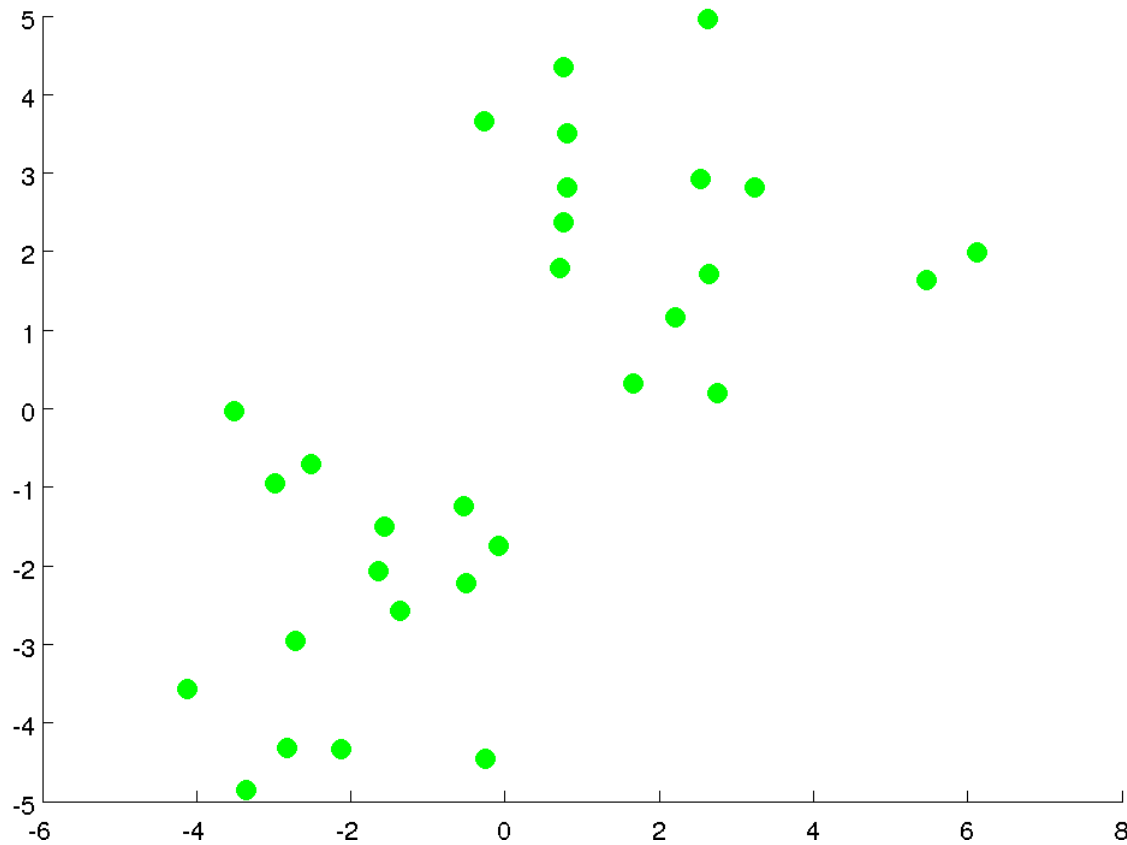
II. Classification Hiérarchique

1. Ascendante (*agglomerative*)

Etapes de l'algorithme "K-Means"

□ Exemple :

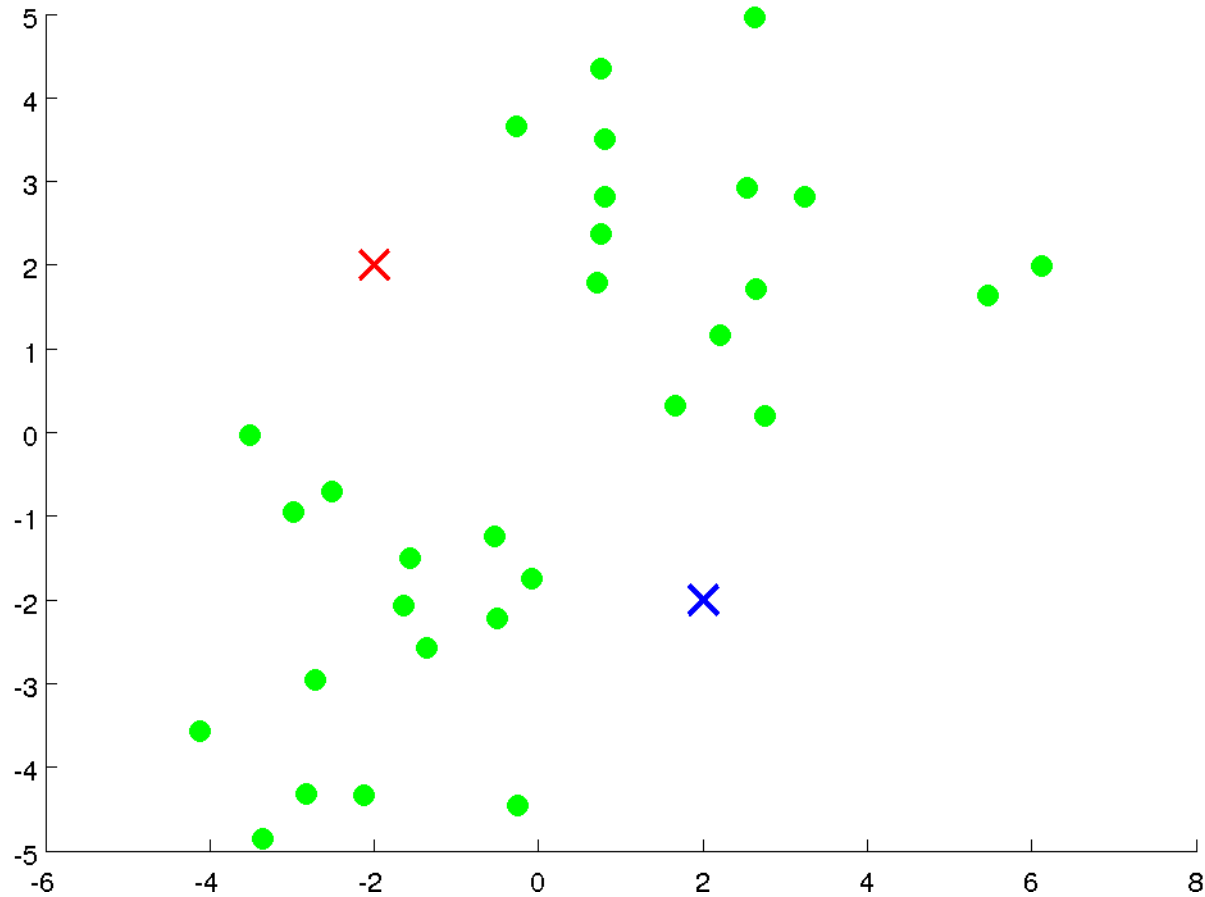
▣ On désire un regroupement en 2 classes.



Etapes de l'algorithme "K-Means"

□ Etape 1:

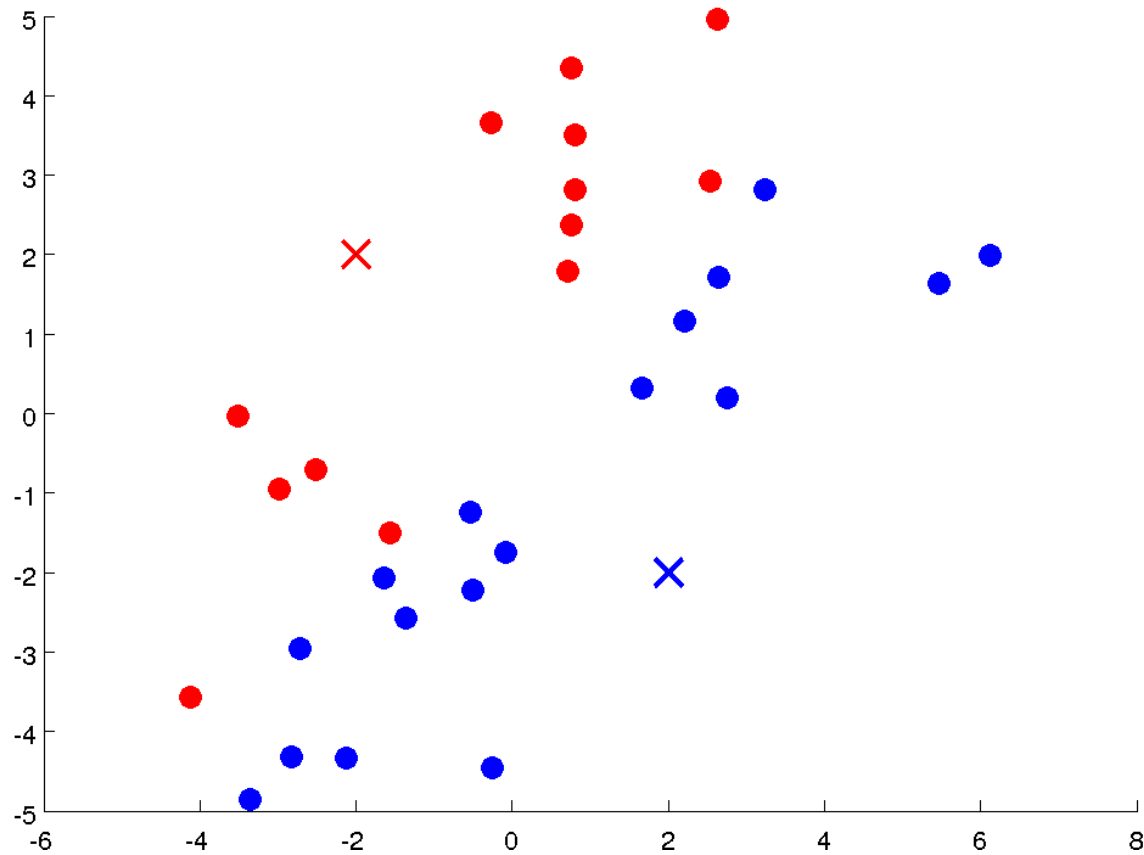
▣ On choisit au hasard 2 centres de classe



Étapes de l'algorithme "K-Means"

□ Étape 2 :

- ▣ Les individus sont affectés au centre de classe le plus proche → Constitution des 2 premières classes.

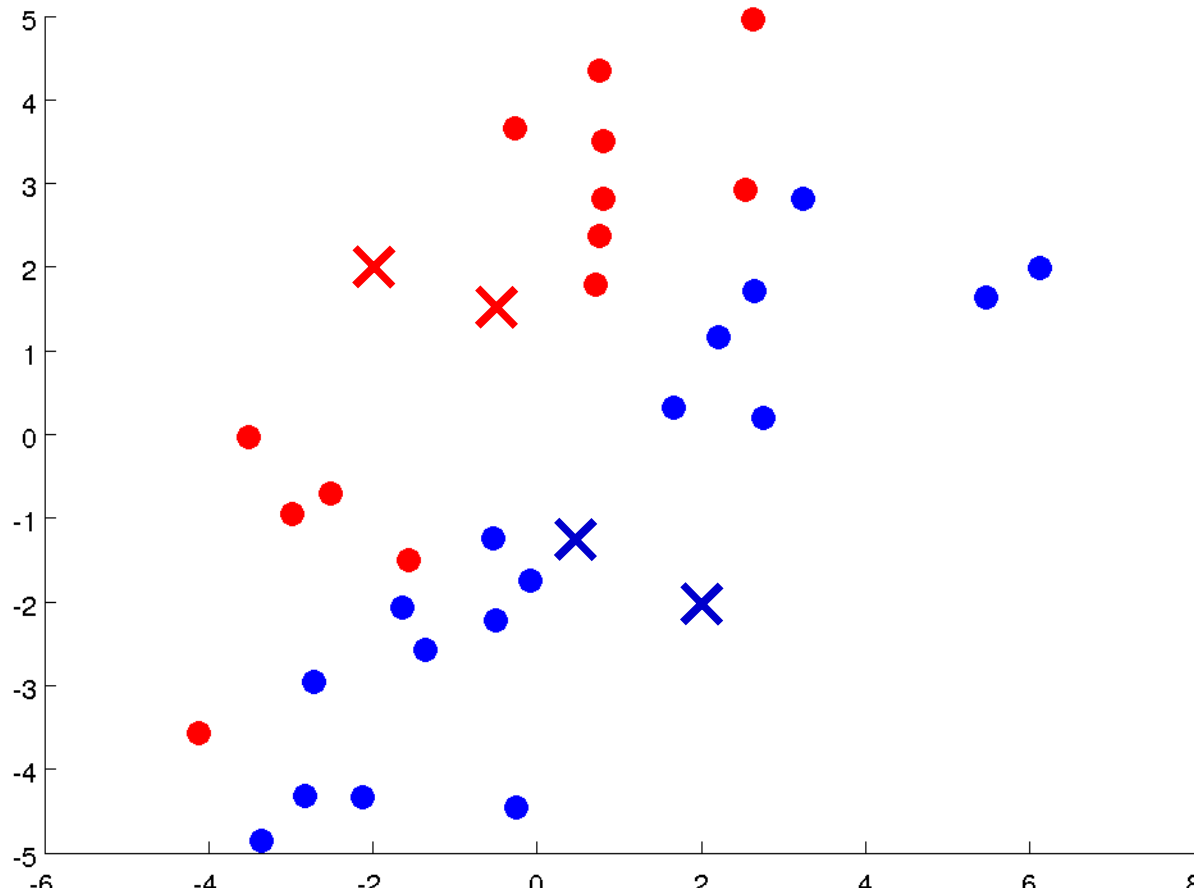


• : Classe 1
• : Classe 2

Étapes de l'algorithme "K-Means"

□ Étape 3 :

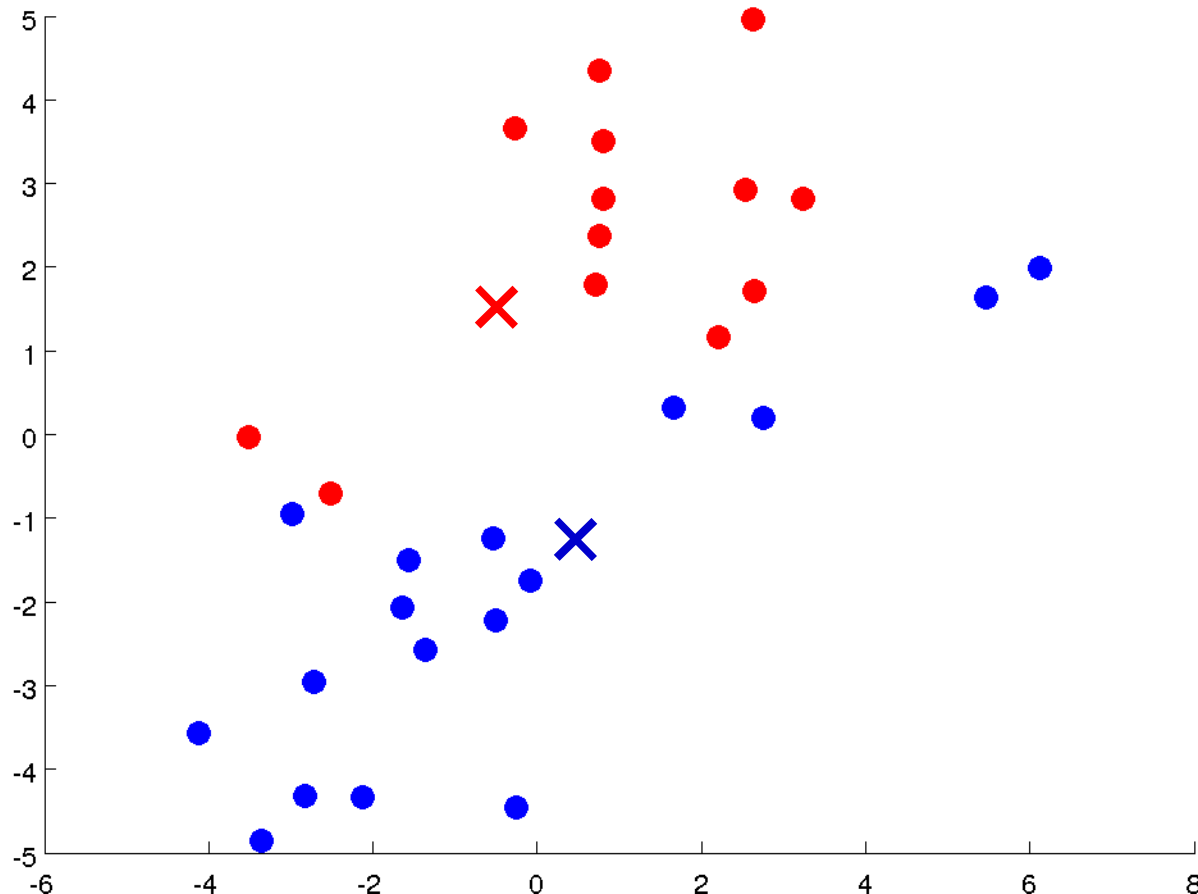
▣ Calcul des nouveaux centres de classes



Étapes de l'algorithme "K-Means"

□ Étape 4 :

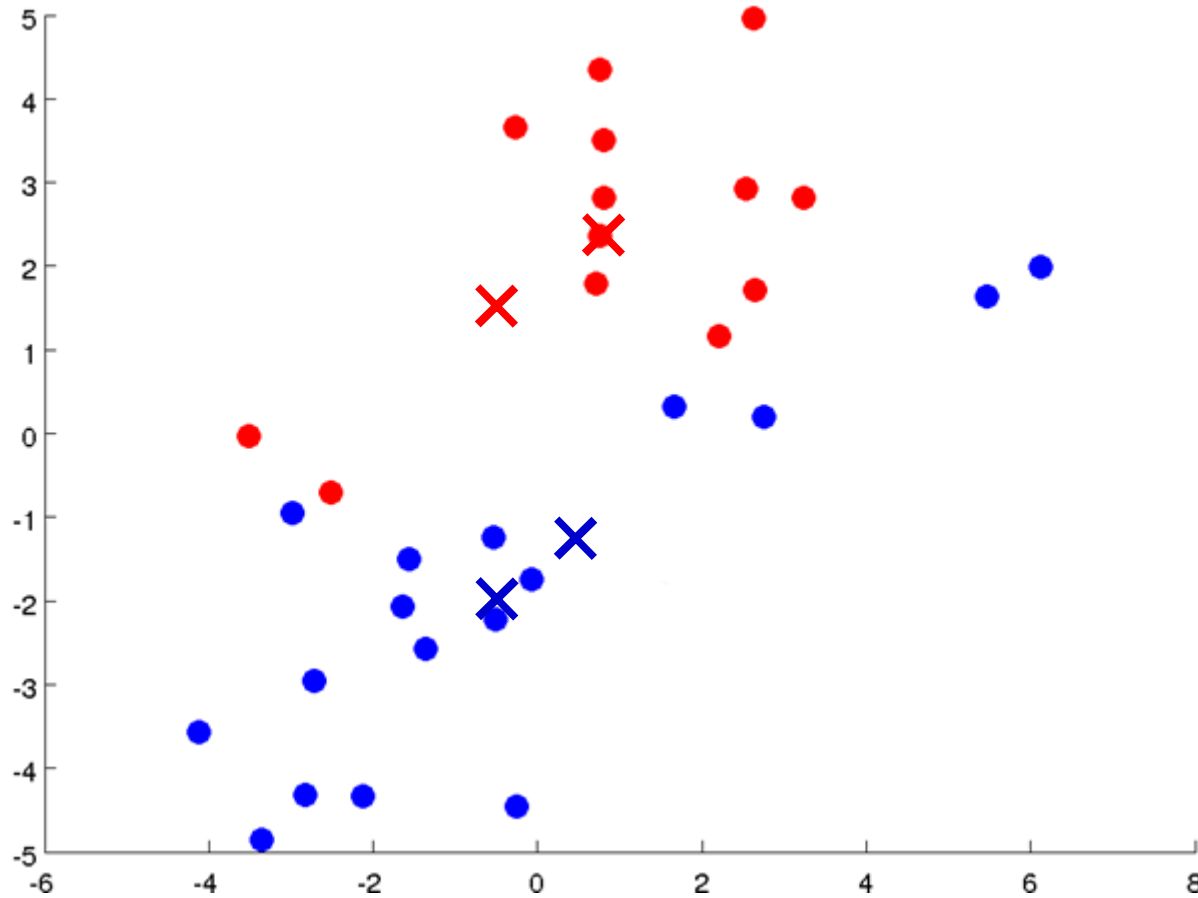
▣ Affectation des individus aux nouveaux centres de classes



Étapes de l'algorithme "K-Means"

□ Étape 5 :

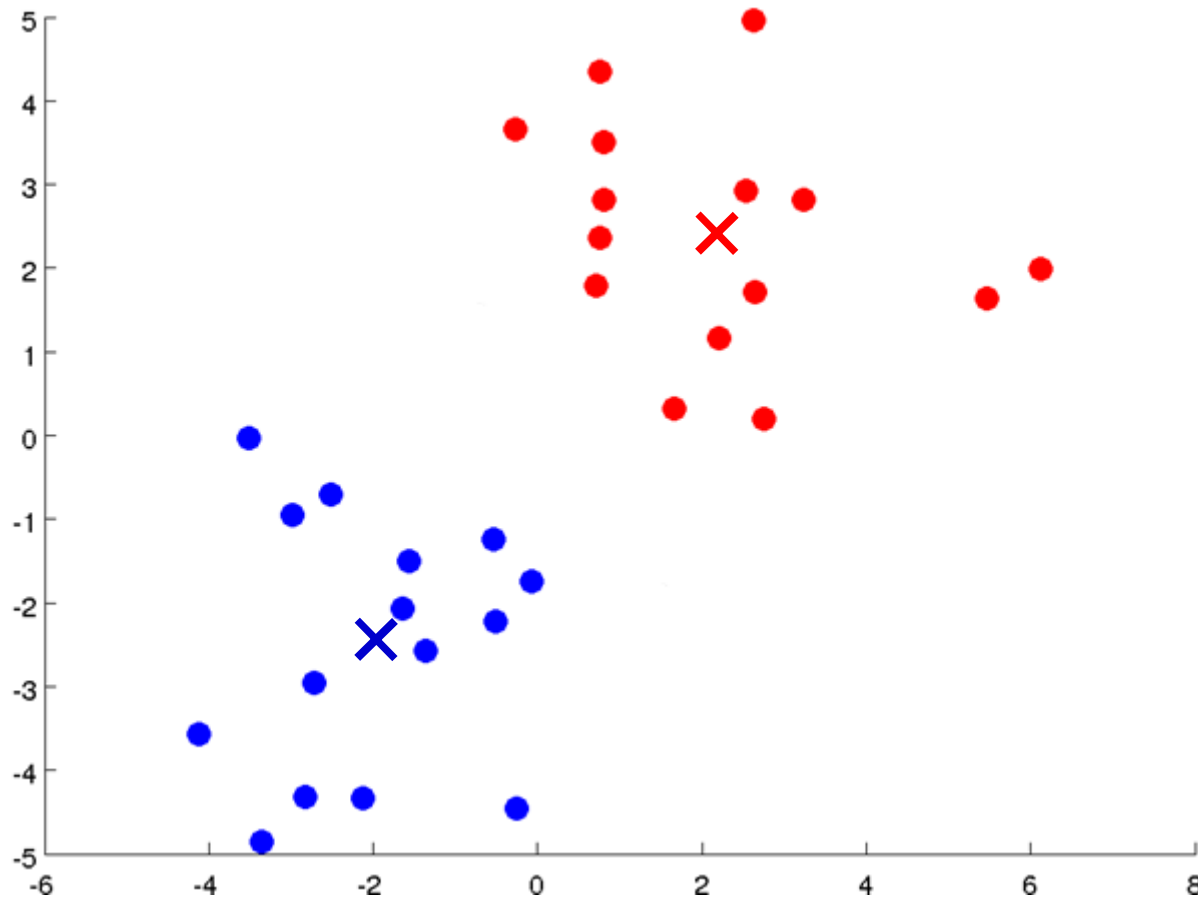
▣ Calcul des nouveaux centres de classes



Étapes de l'algorithme "K-Means"

□ Étape 6 : (dernière étape)

- ▣ Affectation des individus aux nouveaux centres de classe



Etapes de l'algorithme "K-Means"

- L'algorithme s'arrête quand :
 - ▣ La variance intra classe cesse de décroître ou,
 - ▣ La variance inter classe cesse d'augmenter ou,
 - ▣ L'affectation des individus aux classes ne change plus ou,
 - ▣ On a atteint un nombre maximum d'itérations

D. Méthodes utilisées

I. Classification Non hiérarchique

1. Agrégation autour de centre mobiles (*K-Means/K-moyennes*)

- ❑ Définition
- ❑ Etapes
- ❑ L'algorithme de "K-Means"
- ❑ Synthèse

2. Nuées Dynamiques (*Self organizing map*)

II. Classification Hiérarchique

1. Ascendante (*agglomerative*)

L'algorithme de "K-Means"

- **Entrée**: un échantillon de m enregistrements x_1, \dots, x_m
- 1. Choisir k centres initiaux c_1, \dots, c_k
- 2. Répartir chacun des m enregistrements dans le groupe i dont le centre c_i est le plus proche.
- 3. Si aucun élément ne change de groupe alors **arrêt** et sortir les groupes
- 4. Calculer les nouveaux centres : pour tout i , c_i est la moyenne des éléments du groupe i
- Aller en 2

D. Méthodes utilisées

I. Classification Non hiérarchique

1. Agrégation autour de centre mobiles (*K-Means/K-moyennes*)

- ❑ Définition
- ❑ Etapes
- ❑ L'algorithme de “K-Means”
- ❑ Exemple

2. Nuées Dynamiques (*Self organizing map*)

II. Classification Hiérarchique

1. Ascendante (*agglomerative*)

Exemple

- $A=\{1,2,3,6,7,8,13,15,17\}$. Créer 3 clusters à partir de A
- On prend 3 objets au hasard. Supposons que c'est 1, 2 et 3. Ça donne $C_1=\{1\}$, $M_1=1$, $C_2=\{2\}$, $M_2=2$, $C_3=\{3\}$ et $M_3=3$
- Chaque objet O est affecté au cluster au milieu duquel, O est le plus proche. 6 est affecté à C_3 car $\text{dist}(M_3,6) < \text{dist}(M_2,6)$ et $\text{dist}(M_3,6) < \text{dist}(M_1,6)$
On a $C_1=\{1\}$, $M_1=1$,
 $C_2=\{2\}$, $M_2=2$
 $C_3=\{3, 6,7,8,13,15,17\}$, $M_3=69/7=9.86$

Exemple

- $\text{dist}(3, M_2) < \text{dist}(3, M_3) \rightarrow 3$ passe dans C_2 . Tous les autres objets ne bougent pas. $C_1 = \{1\}$, $M_1 = 1$, $C_2 = \{2, 3\}$, $M_2 = 2.5$, $C_3 = \{6, 7, 8, 13, 15, 17\}$ et $M_3 = 66/6 = 11$
- $\text{dist}(6, M_2) < \text{dist}(6, M_3) \rightarrow 6$ passe dans C_2 . Tous les autres objets ne bougent pas. $C_1 = \{1\}$, $M_1 = 1$, $C_2 = \{2, 3, 6\}$, $M_2 = 11/3 = 3.67$, $C_3 = \{7, 8, 13, 15, 17\}$, $M_3 = 12$
- $\text{dist}(2, M_1) < \text{dist}(2, M_2) \rightarrow 2$ passe en C_1 . $\text{dist}(7, M_2) < \text{dist}(7, M_3) \rightarrow 7$ passe en C_2 . Les autres ne bougent pas. $C_1 = \{1, 2\}$, $M_1 = 1.5$, $C_2 = \{3, 6, 7\}$, $M_2 = 5.34$, $C_3 = \{8, 13, 15, 17\}$, $M_3 = 13.25$
- **$\text{dist}(3, M_1) < \text{dist}(3, M_2) \rightarrow 3$ passe en 1. $\text{dist}(8, M_2) < \text{dist}(8, M_3) \rightarrow 8$ passe en 2**
 $C_1 = \{1, 2, 3\}$, $M_1 = 2$, $C_2 = \{6, 7, 8\}$, $M_2 = 7$, $C_3 = \{13, 15, 17\}$, $M_3 = 15$

Plus rien ne bouge

Synthèse

□ **Avantages :**

- **Relativement extensible** : dans le traitement d'ensembles de taille importante (BD volumineuse)
- **Relativement efficace**
- **Produit généralement un optimum** : local ; un optimum global peut être obtenu en utilisant d'autres techniques telles que: algorithmes génétiques, ...

Synthèse

□ Inconvénients :


- **Applicable seulement** dans le cas où la moyenne des objets est définie
- **Besoin de spécifier k** , le nombre de clusters/groupes/classes, a priori
- **Incapable** de traiter les données **bruitées** (noisy)
- **Non adapté** pour découvrir des clusters avec structures non-convexes, et des clusters de tailles différentes
- **Les points isolés** sont mal gérés (doivent-ils appartenir obligatoirement à un cluster ?) -probabiliste

C. Méthodes utilisées

I. Classification Non hiérarchique

1. Agrégation autour de centres mobiles (*KMeans*)
2. Nuées Dynamiques (*Self organizing map*)

II. Classification Hiérarchique

1. ndante (*agglomerative*)
 - ✓ Exemple simple
 - ✓ Exemple complexe & ses étapes
 - ✓ Algorithme
 - ✓ Synthèse

Exemple simple

□ Énoncé :

Nom	Age	Enfants	Pointure	Taille
Alain	45	3	45	182
Martine	28	1	36	165
Pierre	22	0	43	172

□ Questions :

- Comment mesurer la distance entre deux individus ?
- De qui Martine est-elle la plus proche ?

Exemple simple

□ Distance :

▣ On utilisera la **distance euclidienne**.

▣ Distance entre *Alain* et *Martine* :

$$d_{1*2} = \sqrt{(45-28)^2 + (3-1)^2 + (45-36)^2 + (182-165)^2}$$

▣ Distance entre *Martine* et *Pierre* :

$$d_{2*3} = \sqrt{(28-22)^2 + (1-0)^2 + (36-43)^2 + (165-172)^2}$$

▣ Distance entre *Alain* et *Pierre* :

$$d_{1*3} = \sqrt{(45-22)^2 + (3-0)^2 + (45-43)^2 + (182-172)^2}$$

Exemple simple

□ **Tableau des distances :**

- ▣ De ces distances euclédiennes, on obtient le tableau des distances :

	Alain	Martin	Pierre
Alain	0		
Martin	25,74	0	
Pierre	25,33	11,61	0

Exemple simple

□ Remarque :

- ▣ Les deux variables *Pointure* et *Enfants* n'ont que **peu de poids** dans le calcul de la distance.
- ▣ C'est pour remédier à ça, qu'on doit centrer et réduire les données. (*On verra plus tard*)

Id	Age	Enfants	Pointure	Taille
1	45	3	45	182
2	28	1	36	165
3	22	0	43	172

C. Méthodes utilisées

I. Classification Non hiérarchique

1. Agrégation autour de centres mobiles (*KMeans*)
2. Nuées Dynamiques (*Self organizing map*)

II. Classification Hiérarchique

1. **Ascendante (agglomerative)**



- ✓ Exemple simple
- ✓ **Exemple complexe & ses étapes**
- ✓ Algorithme
- ✓ Synthèse

Exemple complexe & ses étapes

□ Énoncé :

▣ 1 tableau de données :

- Variables : $V1$, $V2$
- Individus : A , B , C , D , E

	$V1$	$V2$
A	2	5
B	7	8
C	3	3
D	8	9
E	4	5

□ Questions :

- ▣ A partir des variables $V1$ & $V2$, “Qui est proche de qui” ?

Exemple complexe & ses étapes

□ Étape 1 :

▣ On calcule les **distances euclédiennes**

■ $d_{A*B} = \sqrt{(2-7)^2 + (5-8)^2} \rightarrow d_{A*B} = 5.83$

■ $d_{A*C} = \sqrt{(2-3)^2 + (5-3)^2} \rightarrow d_{A*C} = 2.23$

■ $d_{A*D} = \sqrt{(2-8)^2 + (5-9)^2} \rightarrow d_{A*D} = 7.21$

■ ...

Exemple complexe & ses étapes

□ Étape 1 : (suite)

- ▣ De ces distances euclésiennes, on obtient le **tableau des distances** :

	A	B	C	D	E
A	0				
B	5,83	0			
C	2,23	6,40	0		
D	7,21	1,41	7,81	0	
E	2,00	4,24	2,23	5,65	0

A la main →

A l'aide d'un
Logiciel →

	A	B	C	D
B	5.830952			
C	2.236068	6.403124		
D	7.211103	1.414214	7.810250	
E	2.000000	4.242641	2.236068	5.656854

Exemple complexe & ses étapes

□ Étape 2 :

- On regroupe les individus les plus proches
→ **B** et **D** sont les individus les plus proches

Pourquoi ?

- Car la distance minimum est d_{B*D} (indice de niveau 1.41)
- On se retrouve avec les 4 groupes suivants :
 - A
 - BD
 - C
 - E

Exemple complexe & ses étapes

□ Étape 3 :

▣ On calcule la :

■ distance Euclidienne : d_{A*C} d_{A*E} d_{C*E}

■ distance moyenne : d_{A*BD} d_{BD*C} d_{BD*E}

■
$$d_{A*BD} = \frac{d_{AB} + d_{AD}}{2} \rightarrow d_{A*BD} = 6.52$$

■
.....

□ Étape 3 : (suite)

■ De ces distances, on obtient le **tableau des distances** :

	A	BD	C	E
A	0			
BD	6,52	0		
C	2,23	7,05	0	
E	2,00	4,94	2,23	0

□ Étape 4 :

- On regroupe les individus les plus proches
→ **A** et **E** sont les individus les plus proches

Pourquoi ?

- Car la distance minimum est d_{A^*E} (indice de niveau 2,00)
- On se retrouve avec les 3 groupes suivants :
 - AE
 - BD
 - C

□ Étape 5 :

- On calcule la distance moyenne : d_{AE*C} d_{AE*BD} d_{BD*C}

$$■ d_{AE*C} = \frac{d_{AC} + d_{EC}}{2} \rightarrow d_{AE*C} = 2.23$$

$$■ d_{AE*BD} = \frac{d_{A*BD} + d_{E*BD}}{2} \rightarrow d_{AE*BD} = 5.73$$

- On obtient le tableau des distances

	AE	BD	C
AE	0		
BD	5,73	0	
C	2,23	7,05	0

Exemple complexe & ses étapes

□ Étape 6 :

- On regroupe les individus les plus proches
→ **AE** et **C** sont les individus les plus proches

Pourquoi ?

- Car la distance minimum est d_{C*AE} (indice de niveau 2,23)
- On se retrouve avec les 3 groupes suivants :
 - CAE
 - BD

Exemple complexe & ses étapes

□ Étape 7 :

▣ On calcule la distance moyenne : d_{AEC*BD}

$$■ d_{AEC*BD} = \frac{d_{AE*BD} + d_{C*BD}}{2} \rightarrow d_{AEC*BD} = 6.39$$

▣ On obtient le tableau des distances

	AEC	BD
AE	0	
BD	6,39	0

Exemple complexe & ses étapes

- **Étape 8 :**

- On regroupe les individus **AEC** et **BD**

Exemple complexe & ses étapes

- **Résumé des 8 étapes précédentes :**

- Etape 1&2 : **B** et **D** sont regroupés à l'indice de niveau **1,41**

- Etape 3&4 : **A** et **E** sont regroupés à l'indice de niveau **2,00**

- Etape 5&6 : **AE** et **C** sont regroupés à l'indice de niveau **2,23**

- Etape 7&8 : **AEC** et **BD** sont regroupés à l'indice de niveau **6,39**

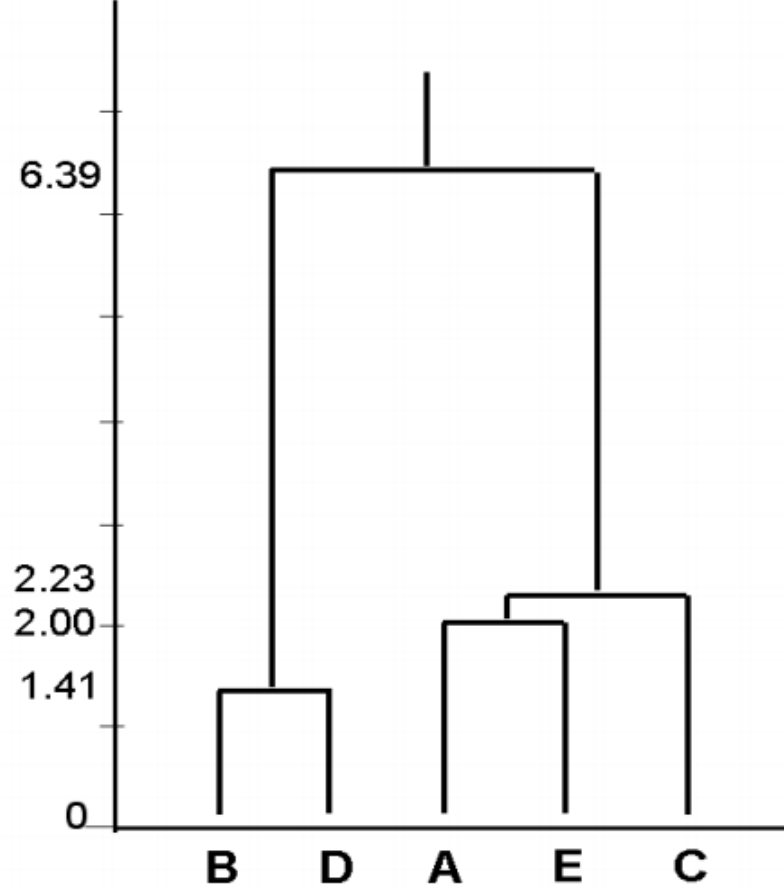
- Pour y voir plus clair, il reste à faire un graphique

- ➔ **Dendrogramme** \approx Arbre hiérarchique

Exemple complexe & ses étapes

A la main

Indices de niveaux

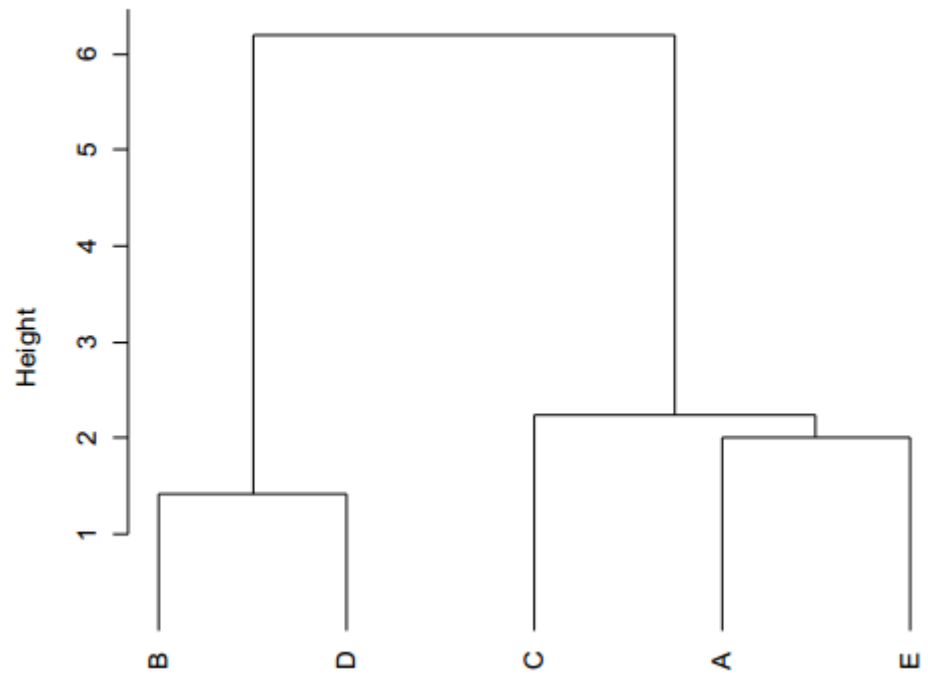


A l'aide d'un Logiciel

Paramètres :

- calcul des distances : distance **euclidienne**
- critère d'agrégation : distance **moyenne** (average)

Cluster Dendrogram



C. Méthodes utilisées

I. Classification Non hiérarchique

1. Agrégation autour de centres mobiles (*KMeans*)
2. Nuées Dynamiques (*Self organizing map*)

II. Classification Hiérarchique

1. **Ascendante** (*agglomerative*)



- ✓ Exemple simple
- ✓ Exemple complexe & ses étapes
- ✓ **Algorithme**
- ✓ Synthèse

Algorithme de la classif. hiérarchique ascendante

- **1^{ère} phase :** Initialisation de l'algorithme
 - ▣ Les classes initiales = n singletons individus.
 - ▣ Calcul de la matrice des distances des individus 2 à 2

- **2^{ème} phase :** Itération des étapes suivantes.
 - ▣ Regrouper les 2 éléments (individus ou groupes) les plus proches au sens d'un critère choisi
 - ▣ Mise à jour du tableau des distances en remplaçant les deux éléments regroupés par le nouveau et en recalculant sa distance avec les autres classes

- **Fin de l'itération :**
 - ▣ agrégation de tous les individus en une seule classe

C. Méthodes utilisées

I. Classification Non hiérarchique

1. Agrégation autour de centres mobiles (*KMeans*)
2. Nuées Dynamiques (*Self organizing map*)

II. Classification Hiérarchique

1. **Ascendante** (*agglomerative*)

- ✓ Exemple simple
- ➔ ✓ Exemple complexe & ses étapes
- ✓ Algorithme
- ✓ **Synthèse**

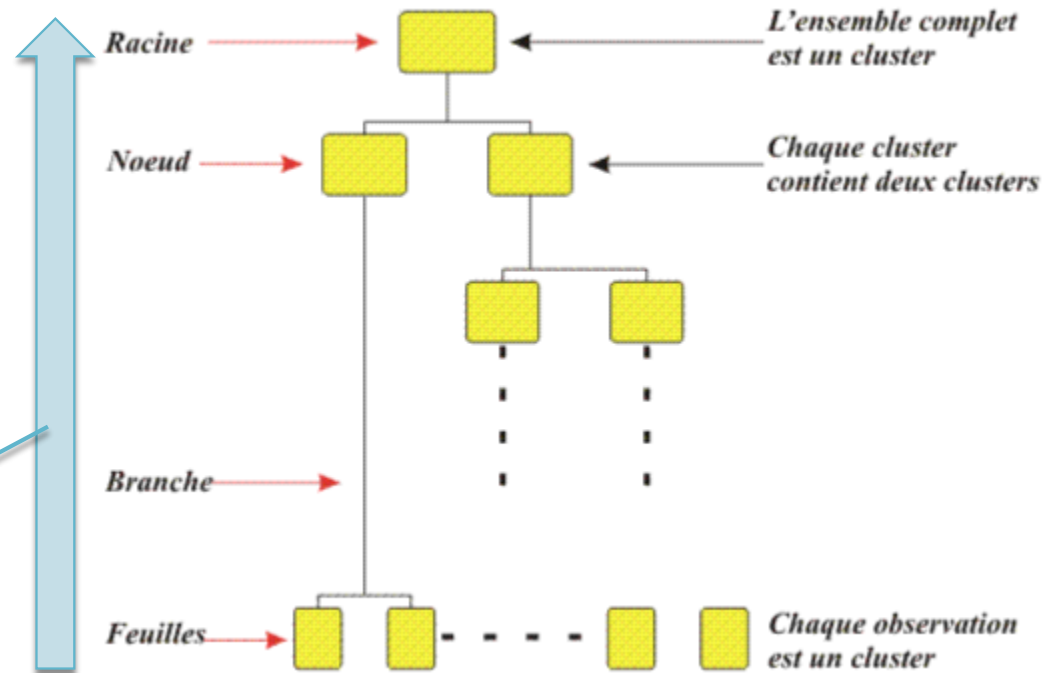
Synthèse

□ Objectif :

- ▣ Fournir un ensemble de partitions de moins en moins fines obtenus par regroupement successifs de parties.
- ▣ Obtenir une hiérarchie, une collection de groupes d'observations

- Les feuilles de l'arbre
= objets/observation
- Les noeuds intermédiaires
de l'arbre = clusters

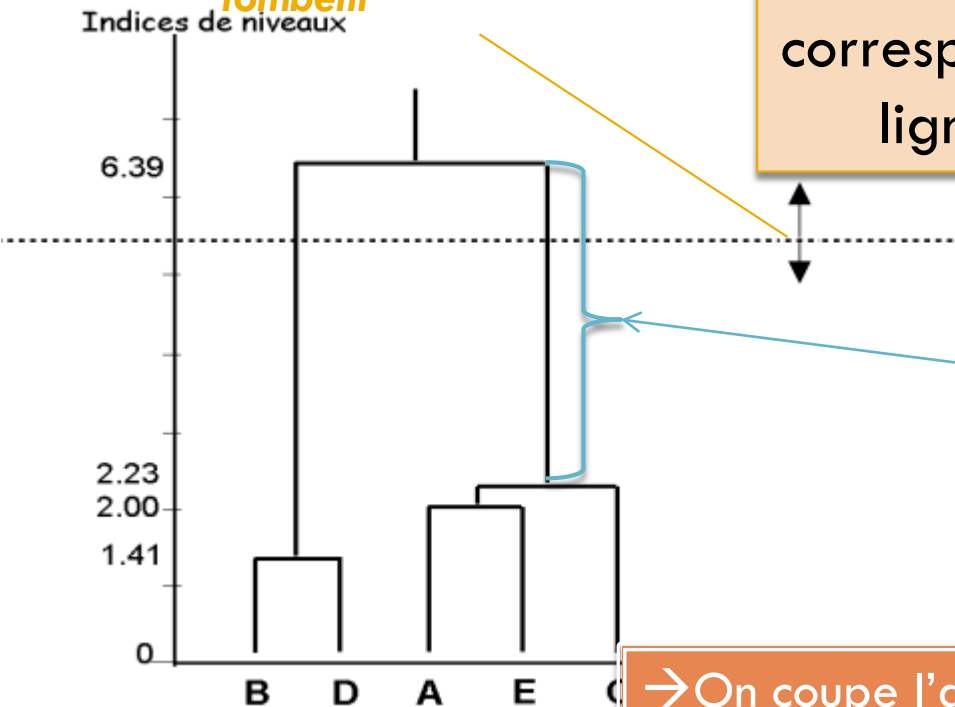
Une fois cet algorithme terminé, on ne récupère donc pas directement une partition, mais une hiérarchie de partitions en $n, \dots, 1$ classes, avec diminution de l'inertie inter-classes à chaque agrégation



Synthèse

□ Lectures des arbres

*on coupe l'arbre et on
regarde les branches qui
tombent*



Le nb de classes
correspond aux nb de
lignes coupées

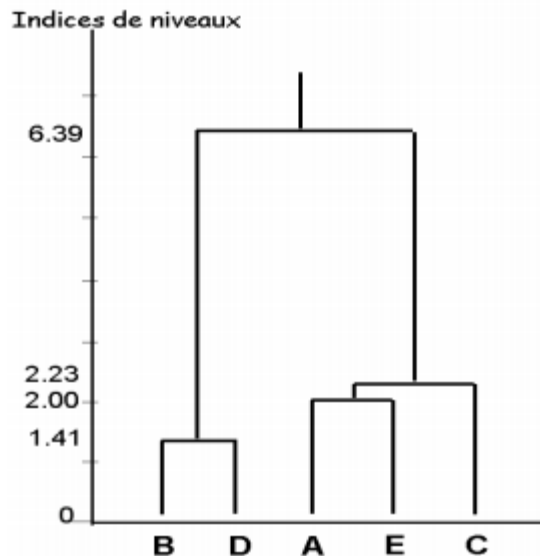
La hauteur d'une branche
est proportionnelle à la
distance entre les deux
objets regroupés

→ On coupe l'arbre avant une perte trop
importante de de distance

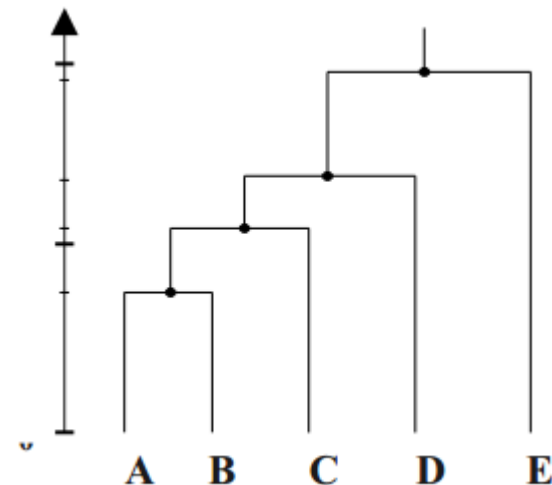
Synthèse

□ Lectures des arbres

- Il existe deux formes d'arbre très reconnaissables



Données classifiables
En 2 ou 3 classes

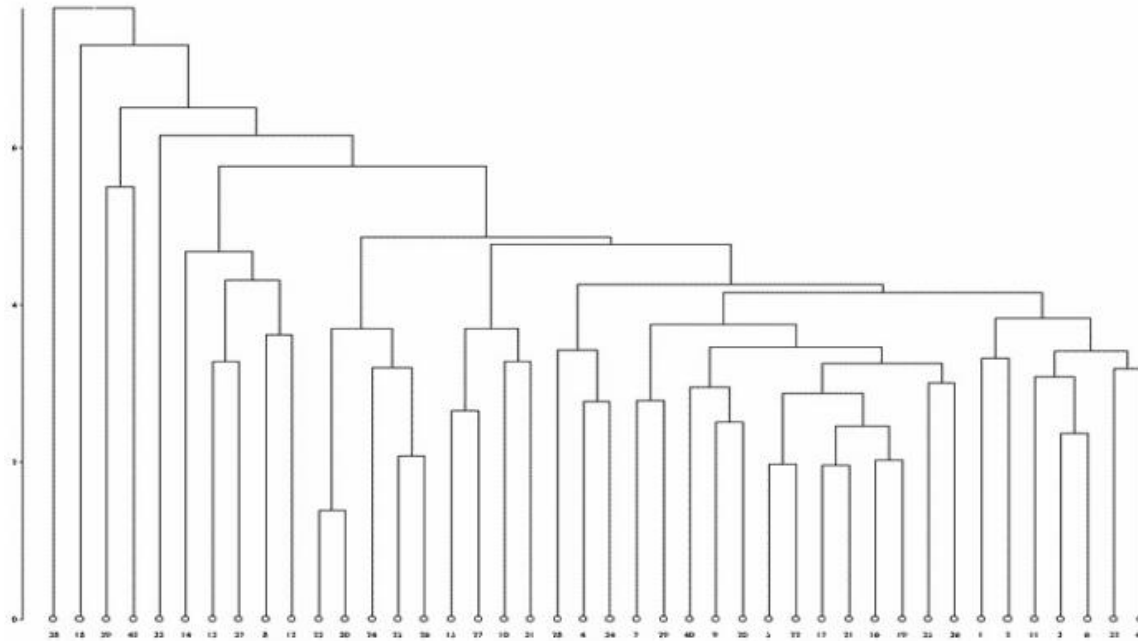


Données non classifiables
Continuum !!

Synthèse

□ Lectures des arbres

Données non classifiables



- Données non classifiables : Inutile d'insister. On peut tester d'autres critères d'agrégation

Synthèse

□ Interprétation

- ▣ Les programmes fournissent souvent des aides interpréter les classes.
- ▣ Elles ne sont pas forcément utiles...

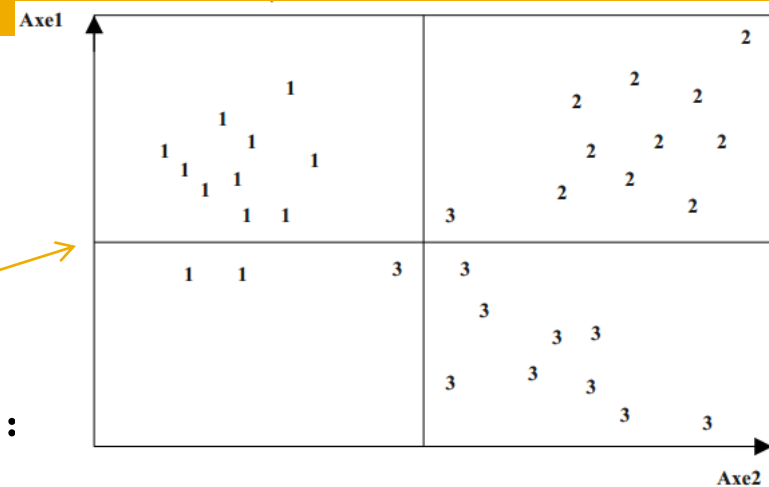
▣ 2 méthodes générales d'aide à l'interprétation :

■ 1ère méthode :

- Projeter les numéros de classes (à la place des codes des individus) sur un plan issu d'une analyse en composantes principales ou d'une analyse factorielle des correspondances

■ 2ème méthode :

- Affecter à chaque individu le numéro de classe auquel il appartient
- Puis, dessiner des histogrammes des variables pour chaque classe
- Calculer des moyennes par classe
- Croiser les variables qualitatives avec les classes (tableaux de fréquences)
- ..



Synthèse

□ **Avantages :**

- Conceptuellement **simple**
- Les propriétés théoriques sont bien connues
- Quand les clusters sont groupés, la décision est définitive \Rightarrow le nombre d'alternatives différentes à examiner est réduit

□ **Inconvénients :**

- **Groupement** de clusters est **définitif** \Rightarrow décisions erronées sont **impossibles** à **modifier** ultérieurement
- Méthodes **non extensibles** pour des ensembles de données de **grandes tailles**

Techniques du Datamining

- **Présentation des techniques**
- **Description de chaque technique :**
 - ✓ **Mesures de Similarités & Types Variables**
 - ✓ **Classification/Structuration /Clustering**
 - ✓ **Association**
 - ✓ **Estimation & Segmentation & Prévision**

E. Méthodes utilisées

Regles D'associations

- ✓ **Introduction**
- ✓ Indicateurs d'évaluations
- ✓ Algorithme Apriori

règles d'associations

□ Règles d'association :

Une règle d'association a la forme logique générale suivante

si Conditions alors Résultats

$$C \rightarrow R$$

■ motifs de la forme : Corps \rightarrow Tête

■ Exemple : achète(x, "fraises") \rightarrow achète(x, "Pommes")

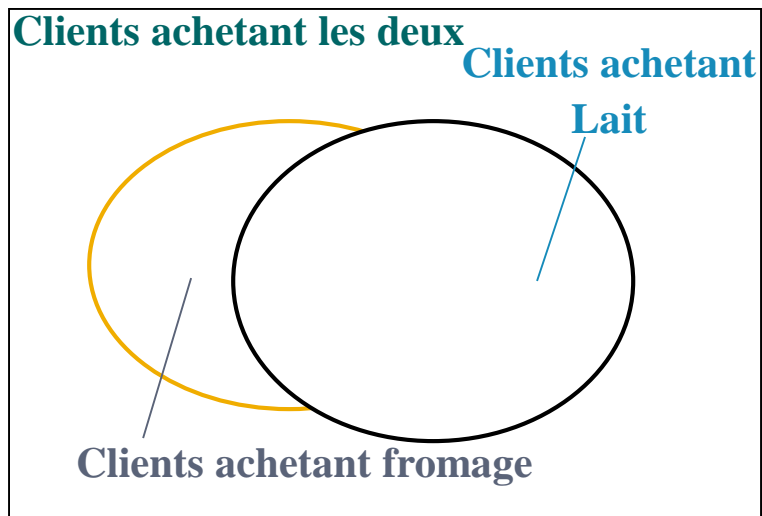
□ Etant donnés: (1) une base de transactions, (2) chaque transaction est décrite par un identifiant et une liste d'items

■ Trouver: toutes les règles qui expriment une corrélation entre la présence d'un item avec la présence d'un ensemble d'items

■ Ex., 98% des personnes qui achètent des fraises achètent des pommes

règles d'associations

- Mesures: Support et Confiance



pour une règle du type : $C \rightarrow R$

☒ Indice de confiance

$$IC = p(C,R) / p(C)$$

Avec:

$$p(C,R) = \text{nb} (C = \text{vrai et } R = \text{vrai}) / \text{nb total}$$

$$p(C) = \text{nb} (C = \text{vrai}) / \text{nb total}$$

On arrive à :

$$IC = \text{nb} (C = \text{vrai et } R = \text{vrai}) / \text{nb} (C = \text{vrai})$$

☒ Indice de Support

IS = probabilité (condition)

$$IS = p(C)$$

Ou encore :

$$IS = \text{nb} (C = \text{vrai}) / \text{nb total}$$

Soit support minimum 50%, et confiance minimum 50%,

$$A \Rightarrow C \text{ (50\%, 66.6\%)}$$

$$C \Rightarrow A \text{ (50\%, 100\%)}$$

ID Transaction	Items
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

règles d'associations

□ Extraction de règles

Transaction ID	Items
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%
Min. confiance 50%

Itemsets fréquents	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

Pour $A \Rightarrow C$:

support = support({A, C}) = 50%

confiance = support({A, C})/support({A}) = 66.6%

règles d'associations

□ Exemple

Le tableau suivant présente 10 tickets de caisse pour lesquels il y a eu, ou pas, achat de pomme et de fraises.

Pommes	1	1	1	1	1	0	0	0	0	0
Fraises	1	1	1	1	0	0	0	0	1	1

□ Calcul de l'indice de support :

$$\mathbf{IS = p(pomme) = 5/10 = 50\%}.$$

Donc, 50% des individus sont concernés par cette règle.

□ Calcul de l'indice de confiance de la règle « pomme → fraise » :

$$p(\text{pomme}) = 5/10$$

$$p(\text{pomme, fraise}) = 4/10$$

$$\mathbf{IC = (4/10) / (5/10) = 4 / 5 = 80\%}.$$

Dans 80% des cas où il y a « pomme », il y a aussi « fraise ».

règles d'associations

Les indicateurs d'évaluation d'une règle

□ **$IC * IS$ ou $p(C,R)$**

Le pourcentage de **réussite de la règle dans la population totale** est un indicateur de la valeur d'une règle.

$IC * IS$ donne cette mesure :

$$\mathbf{IC * IS = p(C,R)}$$

L'exemple précédent donne :

$$\mathbf{IC * IS = 80\% * 50\% = 40\%}$$

Cet indicateur montre le taux d'individus (40%) concernés par la règle.

règles d'associations

Les indicateurs d'évaluation d'une règle

□ Le « lift »

Pour qu'une règle soit intéressante, l'**indice de confiance de la règle doit être supérieur à la probabilité absolue du résultat.**

Pour qu'une règle soit intéressante, il faut donc que :

indice de confiance > probabilité (résultat)

$$IC > p(R)$$

Le « lift », c'est le **taux de progression de la probabilité du résultat grâce à la règle :**

$$\text{lift} = IC / p(R)$$

Un lift intéressant est > 1

□ On peut aussi calculer la **progression brute** :

$$PB = IC - p(R)$$

Une progression brute intéressante est > 0

règles d'associations

Les indicateurs d'évaluation d'une règle

□ Exemple

Pommes	1	1	1	1	1	0	0	0	0	0
Fraises	1	1	1	1	0	1	1	1	1	1

L'indice de confiance de la règle « pomme → fraise » vaut : $(4/10) / (5/10)$ soit 80%.

Donc, dans 80% des cas où il y a « pomme », on trouve aussi « fraise ».

Mais la probabilité des fraises vaut : $(9/10)$ soit 90 %.

Le lift vaut : $80 / 90 = 8/9$.

La progression brute vaut : $80 - 90 = -10\%$.

Lift < 1 et progression brute < 0 : la règle « pomme ® fraise » est donc sans intérêt.

règles d'associations

Les indicateurs d'évaluation d'une règle

□ Règle inverse

Si une règle est inutile (le lift est < 1) alors, la règle inverse est utile :

si $C \rightarrow R$ est inutile alors $C \rightarrow \text{non}(R)$ est utile

□ Exemple

Dans l'exemple précédent, la règle « pomme \rightarrow non(fraise) » est utile.

L'indice de confiance de la règle « pomme \rightarrow non(fraise) » vaut : $(1/10) / (5/10)$ soit 20%.

La probabilité des non(fraise) vaut : $(1/10)$ soit 10%.

Le lift vaut : $20\% / 10\% = 2$.

De 10% de chances on passe à 20 % de chance.

On a 100% ($10 + X*10 = 20$) de chance en plus de ne pas avoir des fraises quand on a pris des pommes que dans tous les cas.

règles d'associations

□ **La capacité de déploiement**

La capacité de déploiement, c'est le pourcentage de ceux qui vérifient les conditions mais pas encore les résultats.

Autrement dit, c'est le pourcentage de la population qui est potentiellement concerné par l'application de la règle.

$$\text{Capacité de déploiement} = p(C, \text{non } R)$$

□ **Exemple :**

On reprend le tableau précédent :

Pommes	1	1	1	1	1	0	0	0	0	0
Fraises	1	1	1	1	0	0	0	0	1	1

La capacité de déploiement de la règle « pomme ® fraise » vaut : (1/10) soit 10%.

règles d'associations

□ **Type des variables**

On a vu dans les exemples que la recherche d'associations se fait sur des variables booléennes.

Cependant, les types continus (numériques) et catégoriels peuvent aussi être pris en compte.

Les types continus doivent d'abord être discrétisés, c'est-à-dire transformés en types catégoriels.

Les types catégoriels sont traités en transformant chaque catégorie (chaque valeur possible pour la variable) en une nouvelle variable qui est donc booléenne.

règles d'associations

□ Exemple

Soit le tableau suivant :

Client	Economie	Capital	Revenu
1	Moyen	Élevé	Élevé
2	Faible	Faible	Moyen
3	Élevé	Moyen	Faible
4	Moyen	Moyen	Moyen
5	Faible	Moyen	Très élevé
6	Élevé	Élevé	Faible
7	Faible	Faible	Faible
8	Moyen	Moyen	Élevé

règles d'associations

On va le transformer en matrice creuse :

Client	Eco = Faible	Eco = Moyen	Eco = Élevé	Cap = Faible	Cap = Moyen	Cap = Élevé	Rev = Faible	Rev = Moyen	Rev = Élevé	Rev = Très élevé
1		1				1			1	
2	1			1				1		
3			1		1		1			
4		1			1			1		
5	1				1					1
6			1			1	1			
7	1			1			1			
8		1			1				1	

règles d'associations

règle d'association et modèle entité-association

La recherche d'associations traite surtout d'associations entre les produits dans un magasin : son premier domaine d'application est l'analyse du ticket de caisse.

Les variables en jeu sont des variables qui identifient le produit. Dans nos exemples, la variable, c'est le produit lui-même.

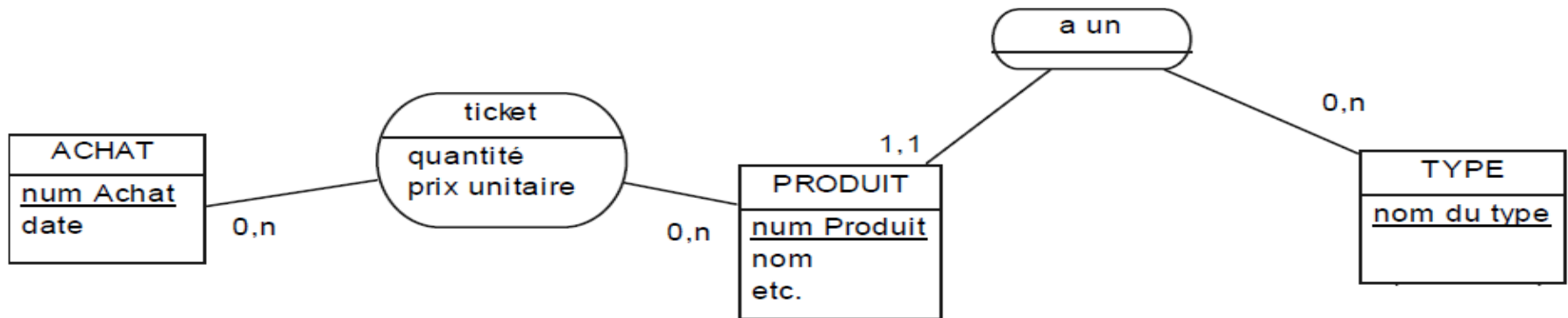
On utilise aussi la taxinomie du produit, c'est-à-dire les différents genres dans lesquels on peut classer le produit (par exemple : tel produit est un yaourt, un produit laitier, un dessert, etc.).

Cette situation est différente de celle de la classification et de l'analyse des données car les attributs avec lesquels on travaille sont souvent des clés étrangères, dans la terminologie du modèle entité-association (MEA).

règles d'associations

□ Exemple :

La table du fichier d'analyse des règles d'association du ticket de caisse correspond au MEA suivant :



Les règles d'association du data mining s'appliquent à l'association non-hiérarchique « ticket ».

Cette association correspond à la table suivante :

Ticket (#numAchat, #numProd, quantité, prix unitaire)

En considérant chaque valeur possible de la variable numProd comme une variable booléenne, on va fabriquer la matrice creuse qui va servir pour la recherche des associations.

règles d'associations

Dans la table ticket, on trouve par exemple (en considérant nomFruit comme l'équivalent de numProd) :

La transformation en matrice creuse donne le résultat suivant :

numT \ numF	Fraise	Poire	Banane	Citron	Pomme	Etc.
1	1	1	1	0	0	0
2	0	0	1	1	1	0
3	0	0	1	0	0	0
etc.	etc.	etc.	etc.	etc.	etc.	etc.

NumTicket	nomFruit
1	Fraise
1	Poire
1	Banane
2	Citron
2	Pomme
2	Banane
3	Banane
Etc.	

A la place des catégories de produit (nom des fruits), on pourrait s'intéresser à des types : fruits exotiques, agrumes, etc.

règles d'associations

Algorithme a priori

Principe de l'algorithme

Pour trouver des règles d'association :

1. On commence par fixer une fréquence minimum pour la recherche : un **seuil de fréquence**.
2. **On calcule la fréquence de chaque n-uplet de variables en partant des n-uplets à une seule variable (singleton), puis en passant à 2, puis à 3, etc.**
3. **On réduit le problème en utilisant la propriété suivante :**

Quel que soit le n-uplet de variables N et quelle que soit la variable V , la fréquence de N est inférieure à celle de $N \cup V$ (N union V).

On peut donc commencer par éliminer tous les singletons de fréquence inférieure au seuil, et donc tous les n-uplets contenant ces singletons, puis tous les doublons de fréquence inférieure au seuil, et donc tous les n-uplets contenant ces doublons, etc.

Cette technique permet de réduire considérablement l'ensemble des n-uplets de variables pouvant être la condition d'une règle d'association.

règles d'associations

Algorithme a priori

Exemple

Matrice creuse de départ

On reprend le tableau des ventes de fruits en le complétant :

numT \ numF	Fraise	Poire	Banane	Citron	Pomme	Raisin	Orange
1	1	1	1				
2			1	1	1		
3			1		1	1	1
4			1			1	1
5	1			1			1
6				1	1	1	1
7			1			1	
8	1					1	
9				1	1		1
10			1				1
11	1	1			1		1
12				1	1		1
13			1	1	1	1	1
14	1	1	1				1

Tableau n°1 : matrice creuse des transactions

règles d'associations

Algorithme a priori

Détermination des n-uplet candidats pour générer des règles d'association

□ *Première réduction du nombre de variables : choix d'un seuil de fréquence*

On fixe un seuil de fréquence : on choisit 4. Ce choix est arbitraire. On considère qu'à moins de $4 / 14 = 29 \%$, on ne s'intéressera pas aux associations.

□ *Calcul de la fréquence pour chaque variable*

Ensuite, on somme les colonnes pour avoir la fréquence par fruit :

<div>numT \ numF</div>	Fraise	Poire	Banane	Citron	Pomme	Raisin	Orange
Fréquence	5	3	8	6	7	6	10
%age	35,7 %	21,4 %	57,1 %	42,9 %	50,0 %	35,7 %	78,6 %

Tableau n°2 : fréquence et %age de fréquence par fruit. Singleton candidat pour les règles d'association.

Toutes les variables sont sélectionnées, sauf la Poire.

règles d'associations

Algorithme a priori

□ *Calcul de la fréquence pour chaque couple de variables*

Ensuite on croise toutes les variables sélectionnées entre elles et on compte le nombre d'occurrences :

	Fraise	Banane	Citron	Pomme	Raisin
Banane	2				
Citron	1	2			
Pomme	1	2	5		
Raisin	1	4	2	3	
Orange	3	5	5	6	4

Tableau n°3 : couples candidats pour les règles d'association.

On peut supprimer la première ligne (fraise) et la dernière colonne (orange) du tableau.

Il n'y a que 6 couples qui ont une fréquence suffisante.

A noter le rôle prépondérant des oranges.

règles d'associations

Algorithme a priori

Calcul de la fréquence pour chaque triplet de variables

On croise tous les couples trouvés avec tous les singletons.

Toutefois, il n'est pas nécessaire de croiser tous les couples trouvés avec tous les singletons.

En effet, on constate que tous les couples avec fraise sont éliminés : on peut donc retirer la fraise des singletons.

Ensuite, pour une variable donnée, les secondes variables intervenant dans les couples éliminés seront éliminées des singletons. Dans le tableau, pour passer tous les couples en revue, on prend toutes les variables « en ligne et en colonne » (banane-fraise, banane-banane, banane-citron, banane pomme, etc.)

	Fraise	Banane	Citron	Pomme	Raisin
Banane	2				
Citron	1	2			
Pomme	1	2	5		
Raisin	1	4	2	3	
Orange	3	5	5	6	4
Triplets candidats		B-R-O	C-P-O	P-C-O	R-B-O

N°	Triplets	Fréquence
1	Banane, Raisin, Orange	3
2	Citron, Pomme, Orange	4
3	Pomme, Citron, Orange (vu en 2)	
4	Raisin, Banane, Orange (vu en 1)	

règles d'associations

Algorithme a priori

Principe de la constitution des règles

Rappelons que, pour la règle $C \Rightarrow R$:

- $IC = p(C,R) / p(C) = f(C,R) / f(C)$
- $IS = p(C) = f(C) / \text{nb total}$
- $p()$ étant la probabilité et $f()$ la fréquences ou nombre d'occurrences.
- $\text{lift} = IC / p(R)$ (un lift intéressant est > 1)
- $\text{Progression Brute} = IC - p(R)$ (une progression brute intéressante est > 0)
- $\text{Capacité de déploiement} = p(C, \text{non } R),$

Le prochain tableau a été construit avec les données des tableaux n° 1 et 2.

Dans les colonnes « lift », on trouve

- Le « lift » : $IC / p(R)$
- La progression brute : $IC - p(R)$
- Dans les colonnes « déploiement », on trouve
- $f(C, nR)$: fréquence des conditions avec la négation du résultat
- $p(C, nR)$: probabilité des conditions avec la négation du résultat

règles d'associations

Algorithme a priori

Conditions	Résultat	Ind. de support		Ind. de conf.		S * C			Lift		Déploiement	
		f(C)	p(C)	f(C,R)	IC		f(R)	p(R)	IC-p(R)		p(C,nR)	
Banane	Raisin	8	57,1%	4	50,0 %	28,6%	6	42,9%	1,2%	7,1%	4	28,6%
Banane	Orange	8	57,1%	5	62,5 %	35,7%	10	71,4%	0,9%	-8,9%	3	21,4%
Citron	Pomme	6	42,9%	5	83,3 %	35,7%	7	50,0%	1,7%	33,3%	1	7,1%
Citron	Orange	6	42,9%	5	83,3 %	35,7%	10	71,4%	1,2%	11,9%	1	7,1%
Pomme	Orange	7	50,0%	6	85,7 %	42,9%	10	71,4%	1,2%	14,3%	1	7,1%
Raisin	Orange	6	42,9%	4	66,7 %	28,6%	10	71,4%	0,9%	-4,8%	2	14,3%
Pomme	Citron	7	50,0%	5	71,4 %	35,7%	6	42,9%	1,7%	28,6%	2	14,3%
Raisin	Banane	6	42,9%	4	66,7 %	28,6%	8	57,1%	1,2%	9,5%	2	14,3%
Orange	Banane	10	71,4%	5	50,0 %	35,7%	8	57,1%	0,9%	-7,1%	5	35,7%
Orange	Citron	10	71,4%	5	50,0 %	35,7%	6	42,9%	1,2%	7,1%	5	35,7%
Orange	Pomme	10	71,4%	6	60,0 %	42,9%	7	50,0%	1,2%	10,0%	4	28,6%
Orange	Raisin	10	71,4%	4	40,0 %	28,6%	6	42,9%	0,9%	-2,9%	6	42,9%
Citron, Pomme	Orange	5	35,7%	4	80,0 %	28,6%	10	71,4%	1,1%	8,6%	1	7,1%
Citron, Orange	Pomme	5	35,7%	4	80,0 %	28,6%	7	50,0%	1,6%	30,0%	1	7,1%
Pomme, Orange	Citron	6	42,9%	4	66,7 %	28,6%	6	42,9%	1,6%	23,8%	2	14,3%

règles d'associations

Algorithme a priori

Interprétation des règles d'association

Classifications

Pour l'interprétation, on va classer le tableau des règles selon les 3 indicateurs de qualité : par

- ❑ **IC * IS** : réussite de la règle dans la population totale
- ❑ **lift** : taux de progression de la probabilité du résultat
- ❑ **déploiement** : pourcentage de la population concerné par l'application de la règle

On commence par le lift car cet indicateur permet d'éliminer des règles.

règles d'associations

Algorithme a priori

➤ Classement par lift

Conditions	Résultat	Ind. de support		Ind. de conf.		IS * IC			Lift		Déploiement	
		F(C)	p(C)	f(C,R)	IC		f(R)	p(R)	IC-p(R)		p(C.nR)	
Citron	Pomme	6	42,90%	5	83,30%	35,70%	7	50,00%	1,70%	33,30%	1	7,10%
Citron, Orange	Pomme	5	35,70%	4	80,00%	28,60%	7	50,00%	1,60%	30,00%	1	7,10%
Pomme	Citron	7	50,00%	5	71,40%	35,70%	6	42,90%	1,70%	28,60%	2	14,30%
Pomme, Orange	Citron	6	42,90%	4	66,70%	28,60%	6	42,90%	1,60%	23,80%	2	14,30%
Pomme	Orange	7	50,00%	6	85,70%	42,90%	10	71,40%	1,20%	14,30%	1	7,10%
Citron	Orange	6	42,90%	5	83,30%	35,70%	10	71,40%	1,20%	11,90%	1	7,10%
Orange	Pomme	10	71,40%	6	60,00%	42,90%	7	50,00%	1,20%	10,00%	4	28,60%
Raisin	Banane	6	42,90%	4	66,70%	28,60%	8	57,10%	1,20%	9,50%	2	14,30%
Citron, Pomme	Orange	5	35,70%	4	80,00%	28,60%	10	71,40%	1,10%	8,60%	1	7,10%
Orange	Citron	10	71,40%	5	50,00%	35,70%	6	42,90%	1,20%	7,10%	5	35,70%
Banane	Raisin	8	57,10%	4	50,00%	28,60%	6	42,90%	1,20%	7,10%	4	28,60%
Orange	Raisin	10	71,40%	4	40,00%	28,60%	6	42,90%	0,90%	-2,90%	6	42,90%
Raisin	Orange	6	42,90%	4	66,70%	28,60%	10	71,40%	0,90%	-4,80%	2	14,30%
Orange	Banane	10	71,40%	5	50,00%	35,70%	8	57,10%	0,90%	-7,10%	5	35,70%
Banane	Orange	8	57,10%	5	62,50%	35,70%	10	71,40%	0,90%	-8,90%	3	21,40%

Les « lift » < 1 (ce qui équivaut à une progression brute négative) sont des règles sans intérêt.
 Les 7 premières règles se résument à l'association : Citron – Pomme – Orange. Il ne reste qu'une seule autre association : Raisin – Banane.

règles d'associations

Algorithme a priori

Classement par $IC * IS$

Conditions	Résultat	Ind. de support		Ind. de conf.		IS * IC			Lift		Déploiement	
		F(C)	p(C)	f(C,R)	IC		f(R)	p(R)	IC-p(R)		p(C,nR)	
Orange	Pomme	10	71,40%	6	60,00%	42,90%	7	50,00%	1,20%	10,00%	4	28,60%
Pomme	Orange	7	50,00%	6	85,70%	42,90%	10	71,40%	1,20%	14,30%	1	7,10%
Orange	Citron	10	71,40%	5	50,00%	35,70%	6	42,90%	1,20%	7,10%	5	35,70%
Orange	Banane	10	71,40%	5	50,00%	35,70%	8	57,10%	0,90%	-7,10%	5	35,70%
Banane	Orange	8	57,10%	5	62,50%	35,70%	10	71,40%	0,90%	-8,90%	3	21,40%
Pomme	Citron	7	50,00%	5	71,40%	35,70%	6	42,90%	1,70%	28,60%	2	14,30%
Citron	Pomme	6	42,90%	5	83,30%	35,70%	7	50,00%	1,70%	33,30%	1	7,10%
Citron	Orange	6	42,90%	5	83,30%	35,70%	10	71,40%	1,20%	11,90%	1	7,10%
Orange	Raisin	10	71,40%	4	40,00%	28,60%	6	42,90%	0,90%	-2,90%	6	42,90%
Banane	Raisin	8	57,10%	4	50,00%	28,60%	6	42,90%	1,20%	7,10%	4	28,60%
Pomme, Orange	Citron	6	42,90%	4	66,70%	28,60%	6	42,90%	1,60%	23,80%	2	14,30%
Raisin	Banane	6	42,90%	4	66,70%	28,60%	8	57,10%	1,20%	9,50%	2	14,30%
Raisin	Orange	6	42,90%	4	66,70%	28,60%	10	71,40%	0,90%	-4,80%	2	14,30%
Citron, Orange	Pomme	5	35,70%	4	80,00%	28,60%	7	50,00%	1,60%	30,00%	1	7,10%
Citron, Pomme	Orange	5	35,70%	4	80,00%	28,60%	10	71,40%	1,10%	8,60%	1	7,10%

On obtient les mêmes résultat qu'avec le lift.

règles d'associations

Algorithme a priori

Classement par capacité de déploiement

Conditions	Résultat	Ind. de support		Ind. de conf.		IS * IC			Lift		Déploiement	
		F(C)	P(C)	f(C,R)	IC		f(R)	p(R)	IC-p(R)		p(C,nR)	
Orange	Raisin	10	71,40%	4	40,00%	28,60%	6	42,90%	0,90%	-2,90%	6	42,90%
Orange	Citron	10	71,40%	5	50,00%	35,70%	6	42,90%	1,20%	7,10%	5	35,70%
Orange	Banane	10	71,40%	5	50,00%	35,70%	8	57,10%	0,90%	-7,10%	5	35,70%
Orange	Pomme	10	71,40%	6	60,00%	42,90%	7	50,00%	1,20%	10,00%	4	28,60%
Banane	Raisin	8	57,10%	4	50,00%	28,60%	6	42,90%	1,20%	7,10%	4	28,60%
Banane	Orange	8	57,10%	5	62,50%	35,70%	10	71,40%	0,90%	-8,90%	3	21,40%
Pomme	Citron	7	50,00%	5	71,40%	35,70%	6	42,90%	1,70%	28,60%	2	14,30%
Pomme, Orange	Citron	6	42,90%	4	66,70%	28,60%	6	42,90%	1,60%	23,80%	2	14,30%
Raisin	Banane	6	42,90%	4	66,70%	28,60%	8	57,10%	1,20%	9,50%	2	14,30%
Raisin	Orange	6	42,90%	4	66,70%	28,60%	10	71,40%	0,90%	-4,80%	2	14,30%
Citron	Pomme	6	42,90%	5	83,30%	35,70%	7	50,00%	1,70%	33,30%	1	7,10%
Citron, Orange	Pomme	5	35,70%	4	80,00%	28,60%	7	50,00%	1,60%	30,00%	1	7,10%
Pomme	Orange	7	50,00%	6	85,70%	42,90%	10	71,40%	1,20%	14,30%	1	7,10%
Citron	Orange	6	42,90%	5	83,30%	35,70%	10	71,40%	1,20%	11,90%	1	7,10%
Citron, Pomme	Orange	5	35,70%	4	80,00%	28,60%	10	71,40%	1,10%	8,60%	1	7,10%

L'observation des capacités de développement montre que le triplet Citron – Pomme – Orange est intéressant à déployer, tandis que le couple Raisin – Banane l'est moins.

règles d'associations

Algorithme a priori

- Support et confiance ne sont pas toujours suffisants
- Ex : Soient les 3 articles A, B et C

article	A	B	C	A et B	A et C	B et C	A, B et C
fréquence	45%	42,5%	40%	25%	20%	15%	5%

- Règles à 3 articles : même support 5%

Confiance

- Règle : Si A et B alors C = 0.20
- Règle : Si A et C alors B = 0.25
- Règle : Si B et C alors A = 0.33

règles d'associations

Algorithme a priori

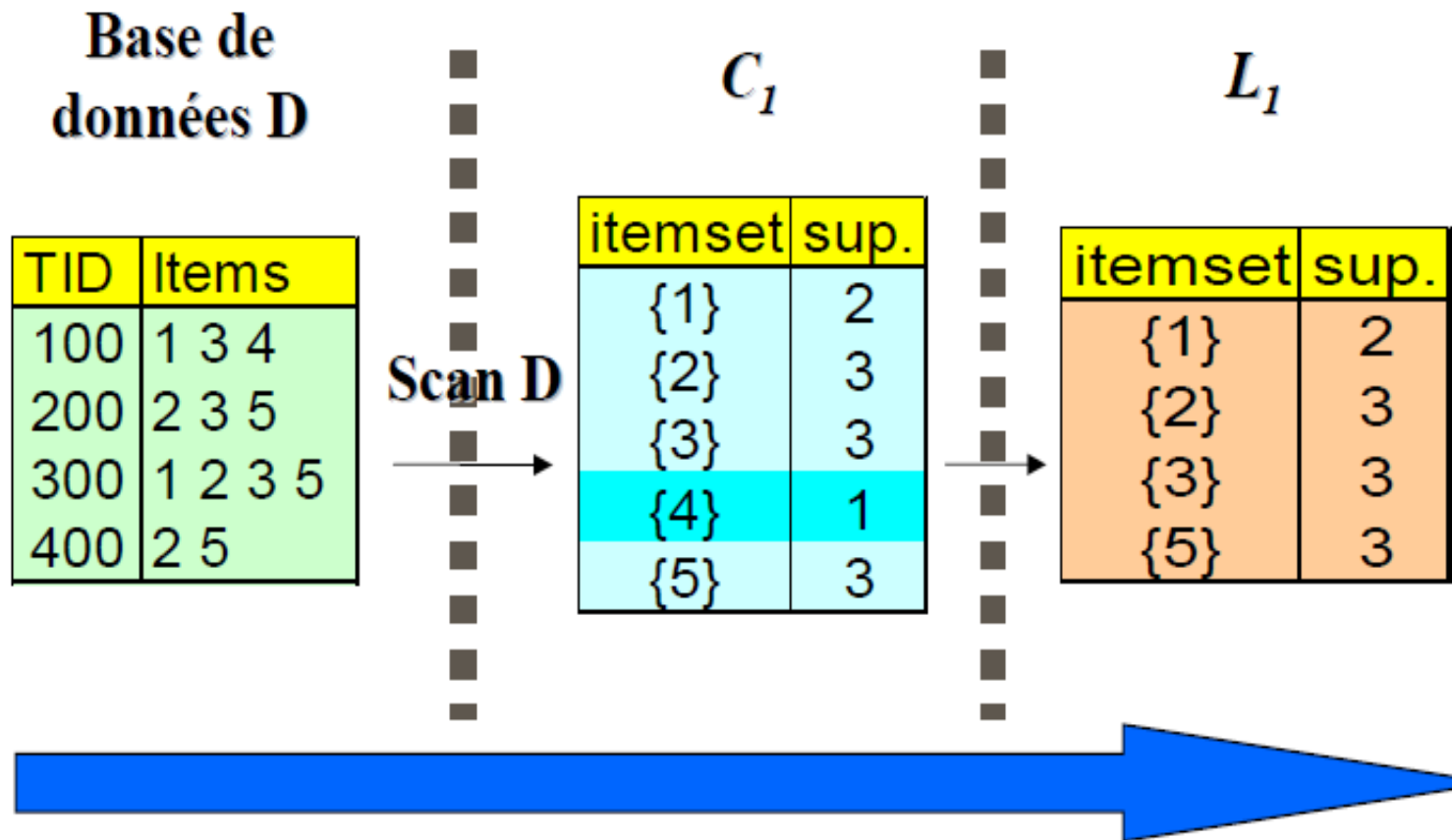
- Amélioration = confiance / fréq(résultat)
- Comparer le résultat de la prédiction en utilisant la règle avec la prédiction sans la règle
- Règle intéressante si Amélioration > 1

Règle	Confiance	F(résultat)	Amélioration
Si A et B alors C	0.20	40%	0.50
Si A et C alors B	0.25	42.5%	0.59
Si B et C alors A	0.33	45%	0.74

- Règle : Si A alors B ; support=25% ; confiance=55% ; Amélioration = 1.31 donc Meilleure règle

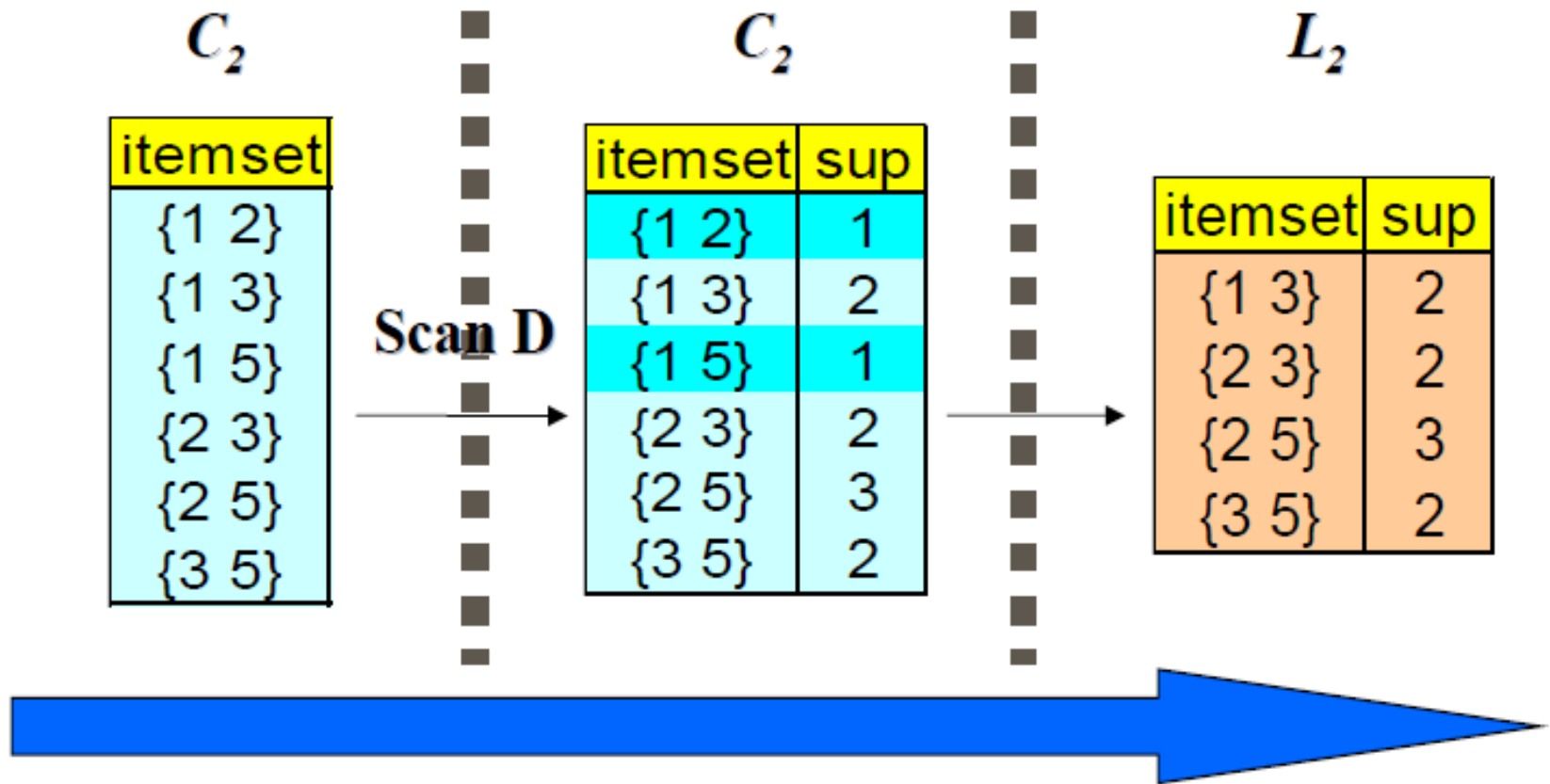
règles d'associations

Algorithme a priori



règles d'associations

Algorithme a priori



règles d'associations

Algorithme a priori

