



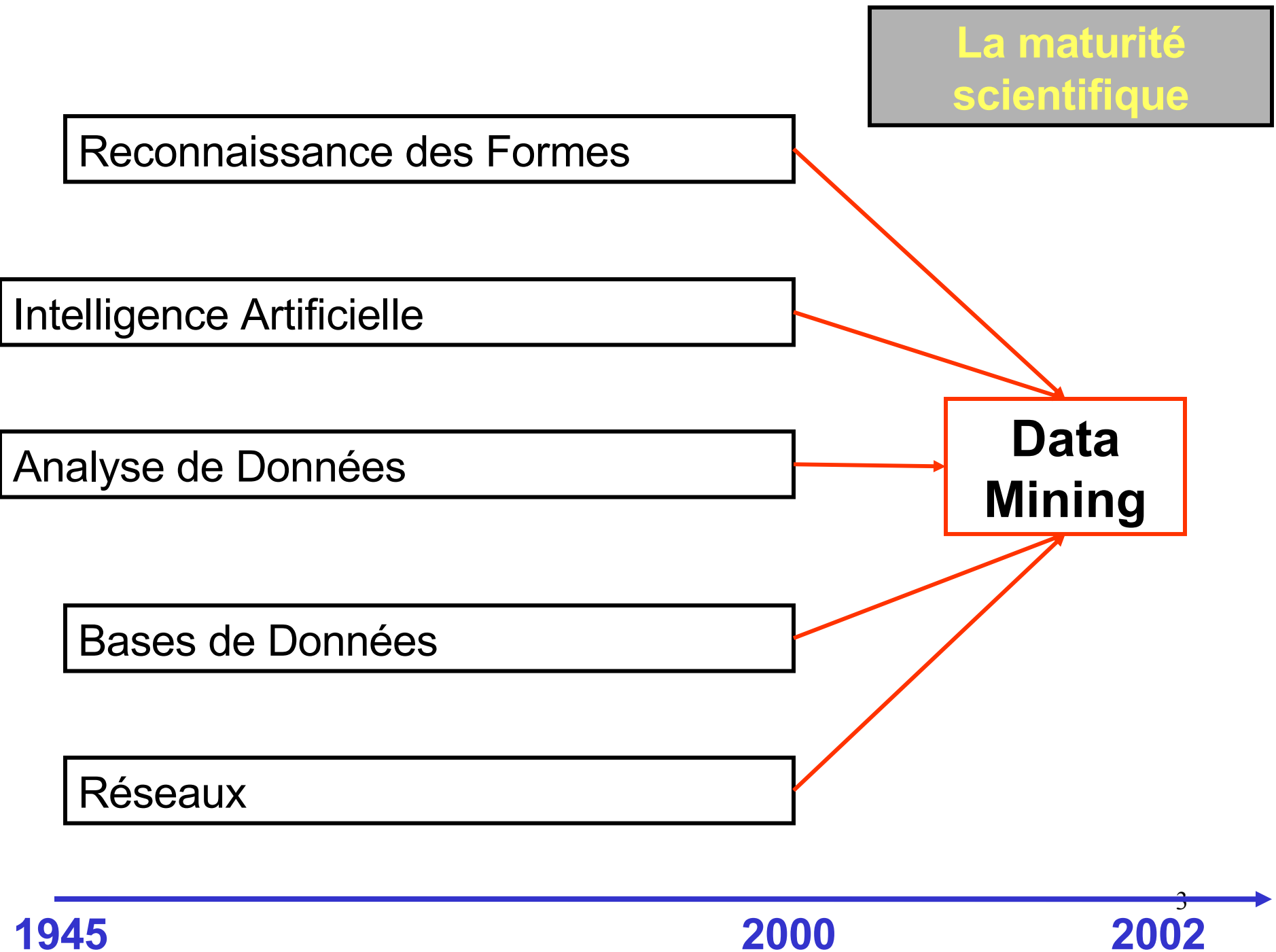
Nicolas Loménie

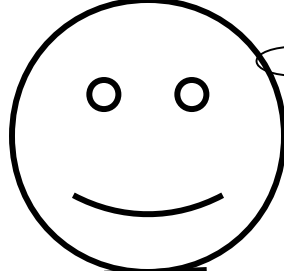
Data Mining



Nicolas Loménie

INTRODUCTION





Représentation et manipulation de connaissances symboliques
Systèmes de mémorisation et de stockage de l'information
Protocoles de communication et d'échange de l'information
Processus de perception et de reconnaissance de formes et de structures

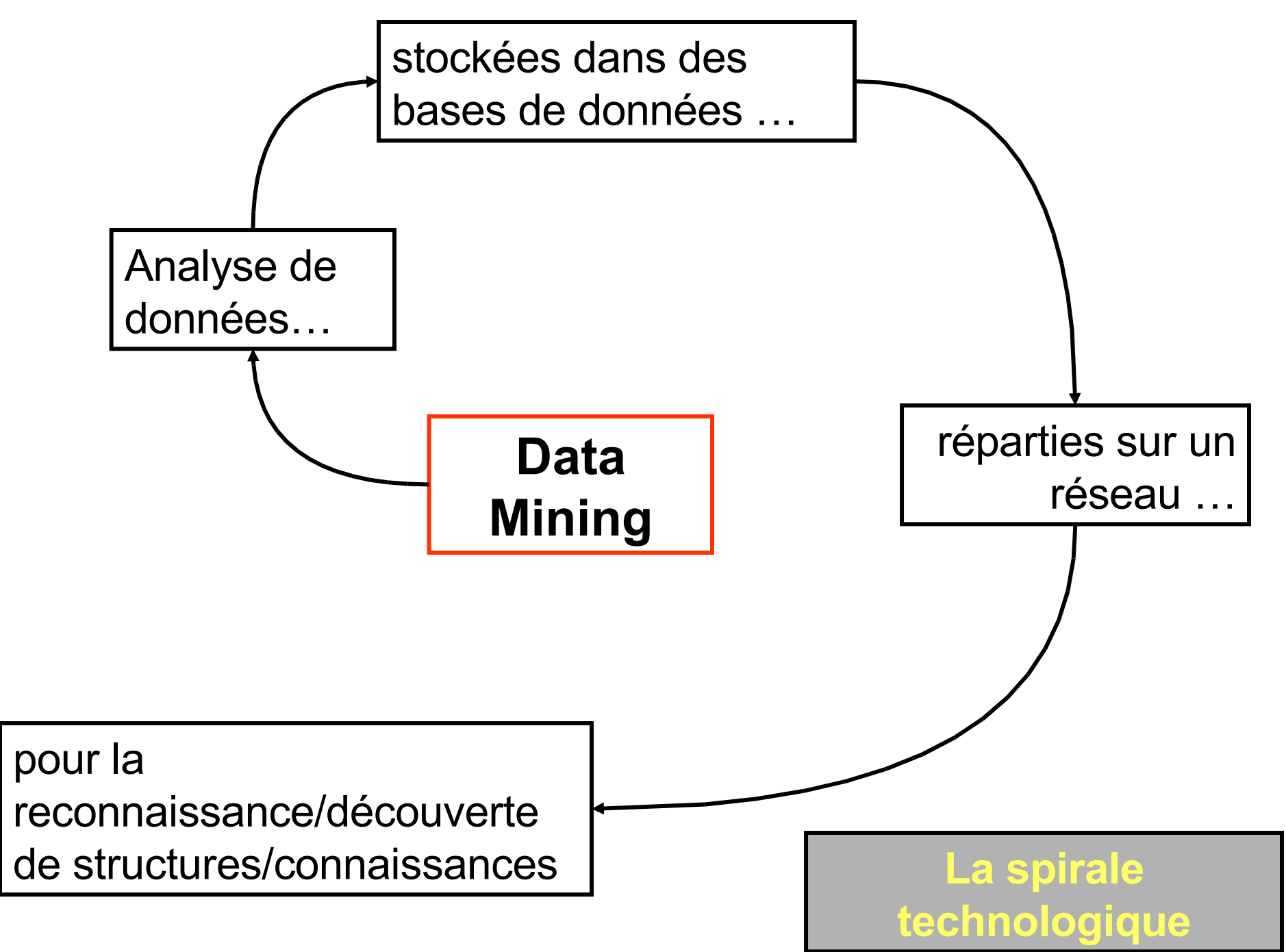
Intelligence Artificielle

Reconnaissance des Formes

Réseaux

Bases de Données

L'édifice cognitif



**Exploration
de
données**

**Data
Mining**

**Fouille
de
Données**

KDD

?

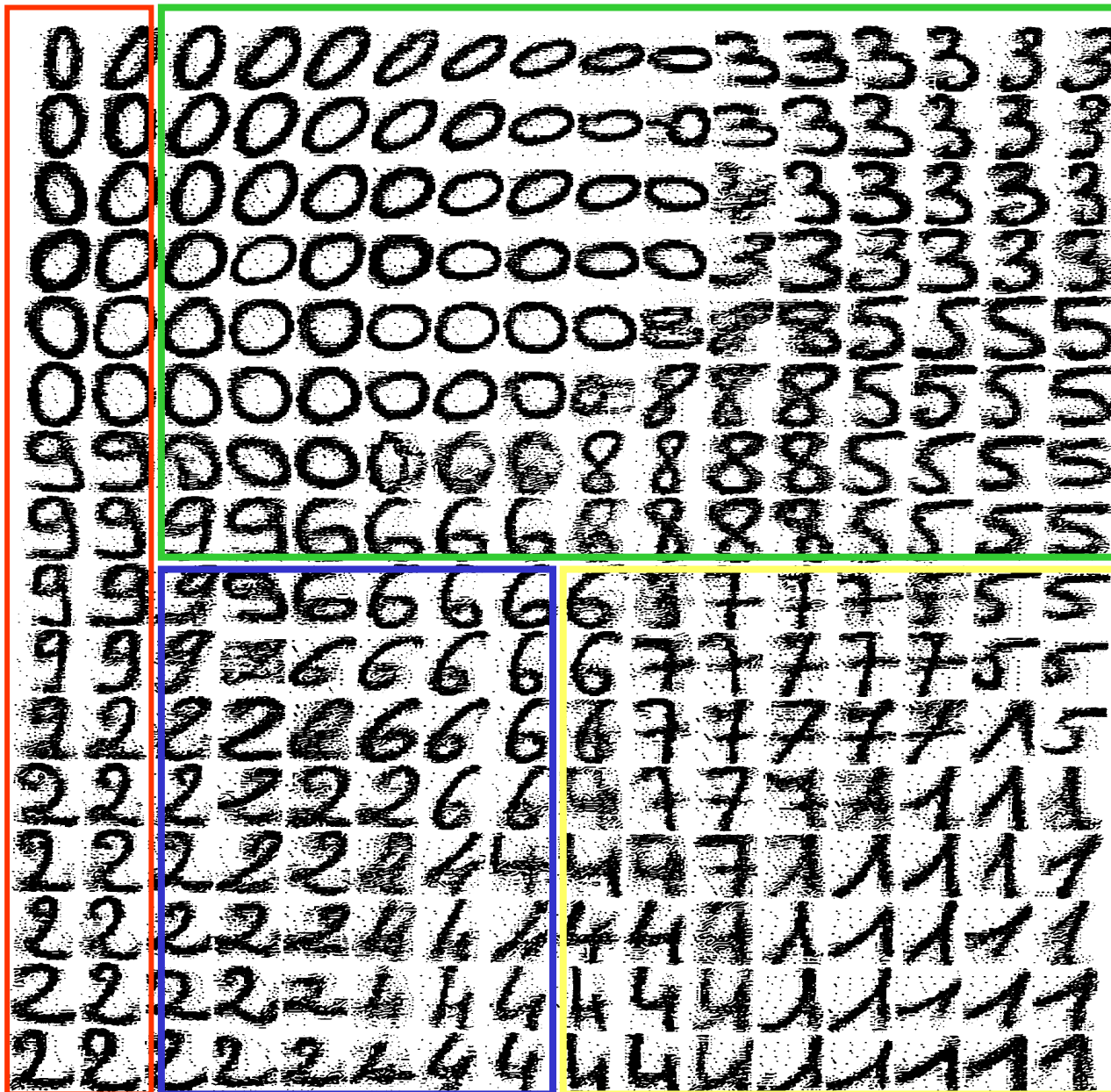
Comment détecter des ressemblances, des structures, des motifs *a priori* ?

Jean-Paul

Samia

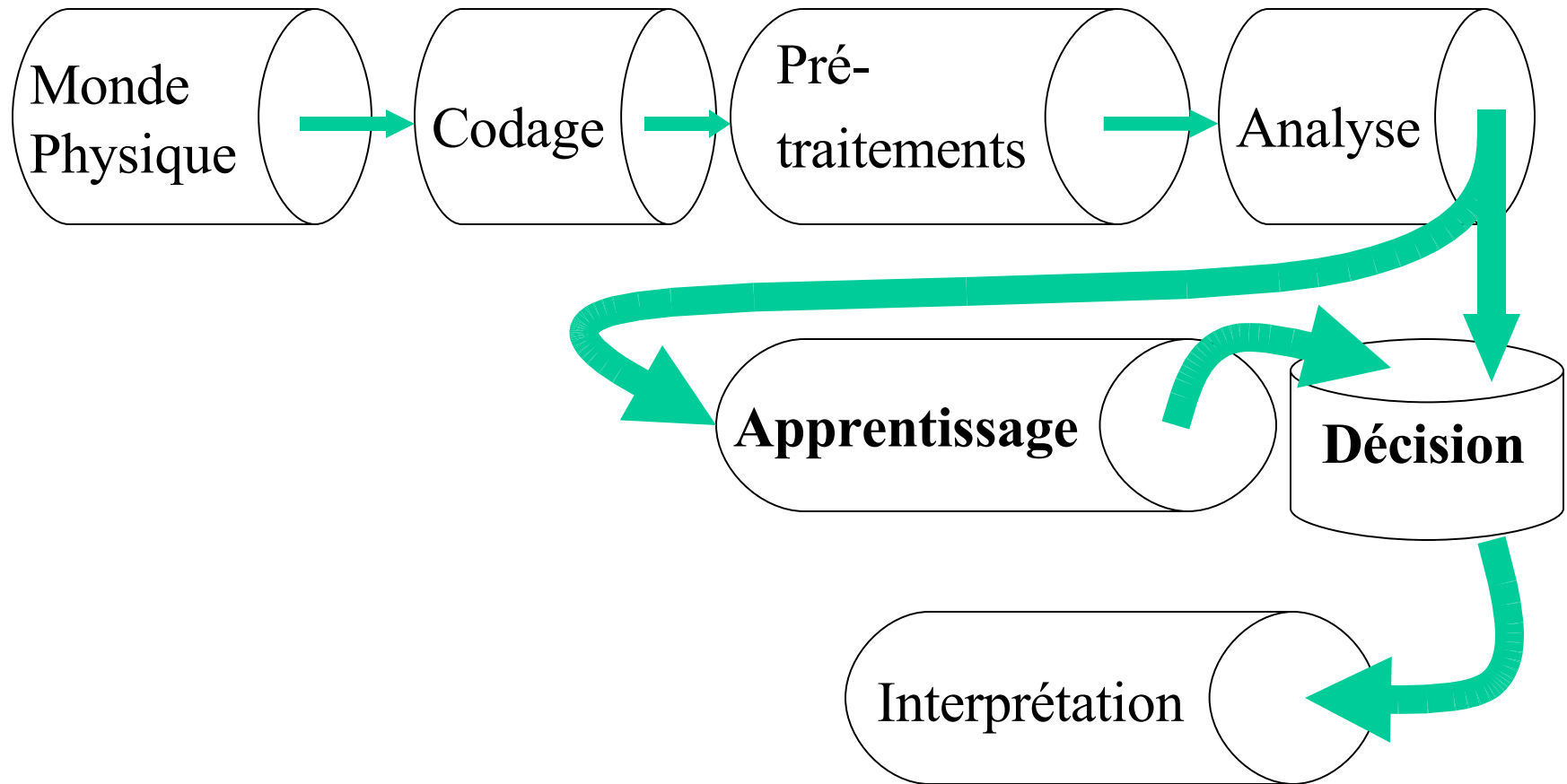
Lin

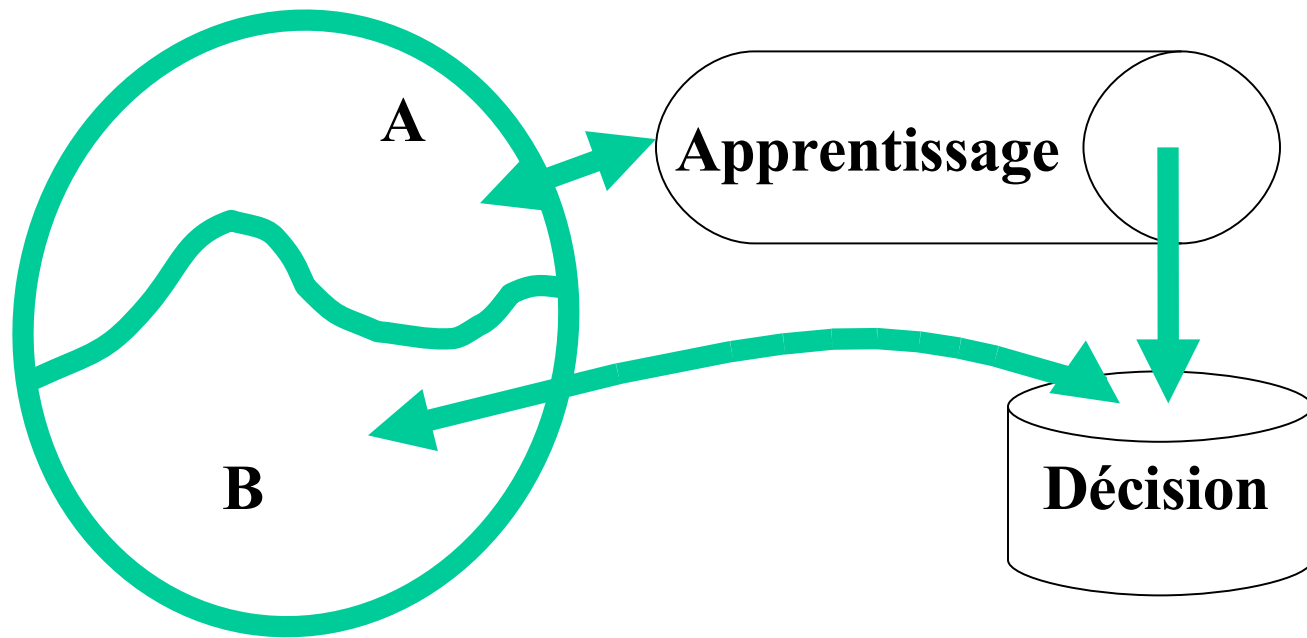
Samy





Systeme de Reconnaissance de Formes Classique

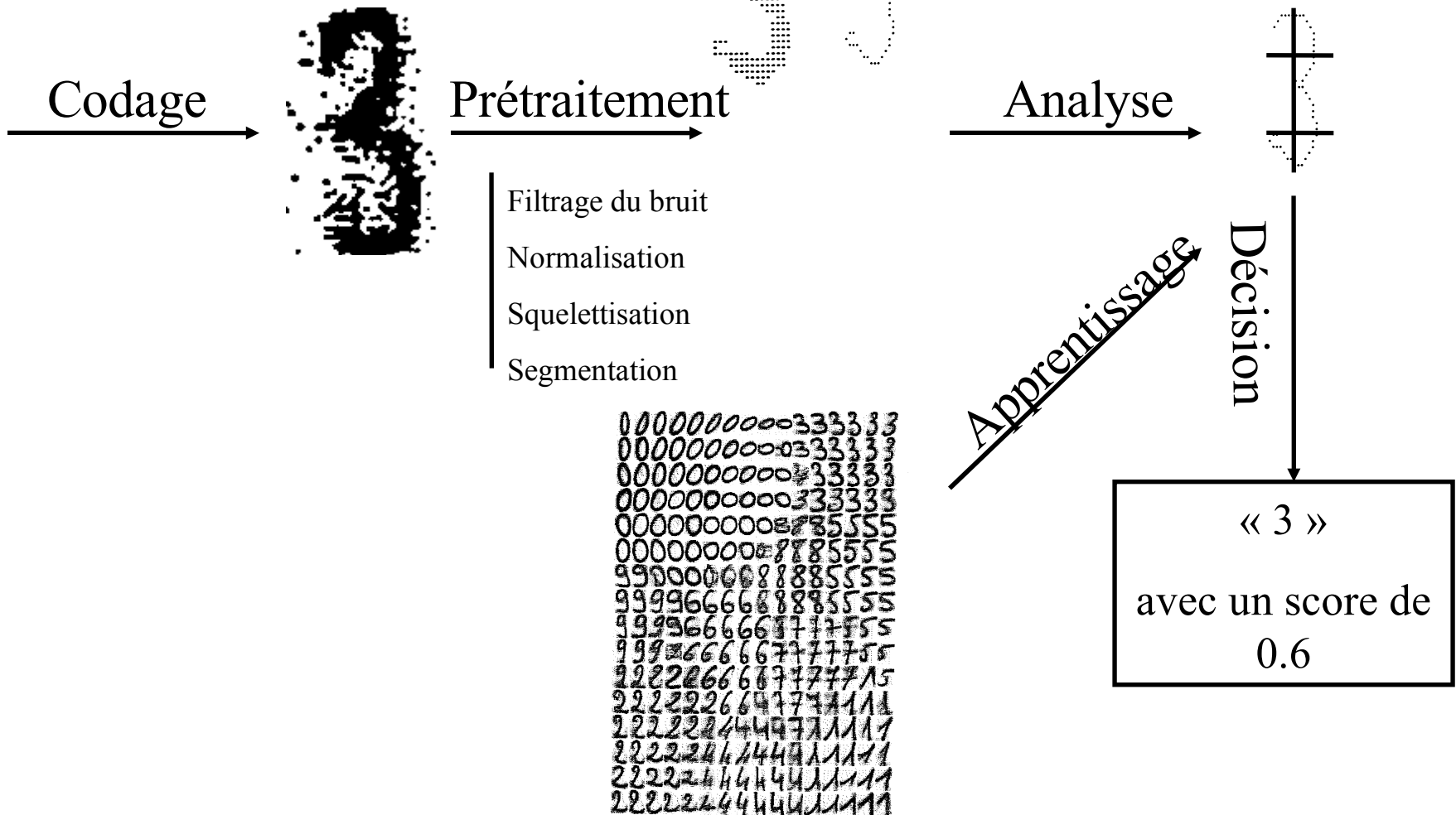




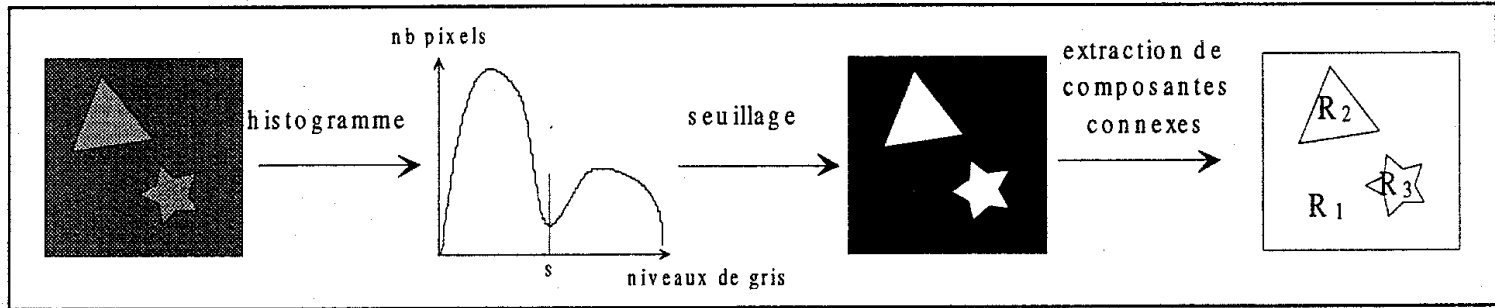
A = Ensemble d'échantillons pour chaque classe

$A = \emptyset \Rightarrow$ Apprentissage Non Supervisé

$A \neq \emptyset \Rightarrow$ Apprentissage Supervisé



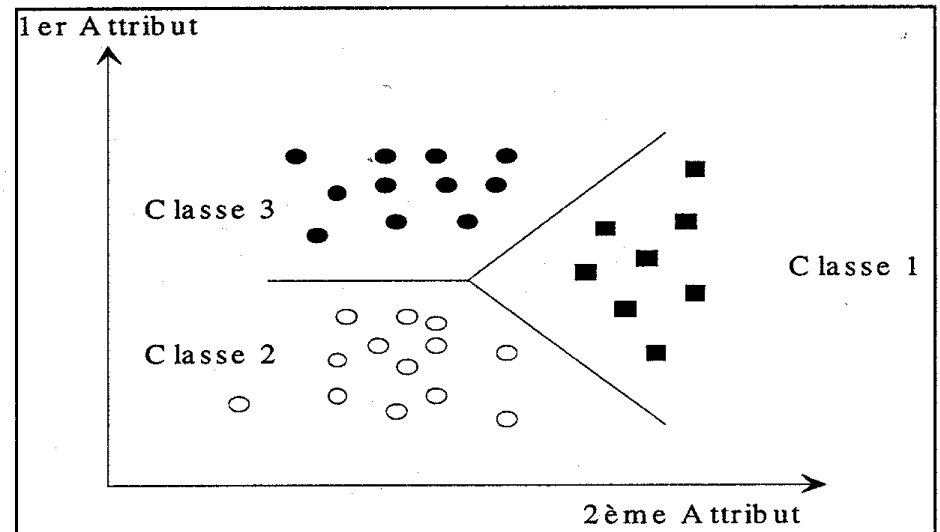
Dans le cas NON supervisé, les techniques spécifiques utilisées sont typiques des applications dites de Fouille de Données



Segmentation en régions par seuillage

- **Classification**

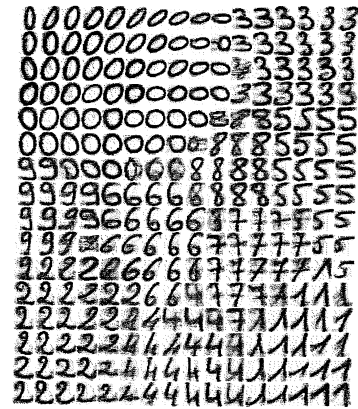
- **Segmentation**



Classification des points dans l'espace des attributs (3 classes, dimension 2 de l'espace)

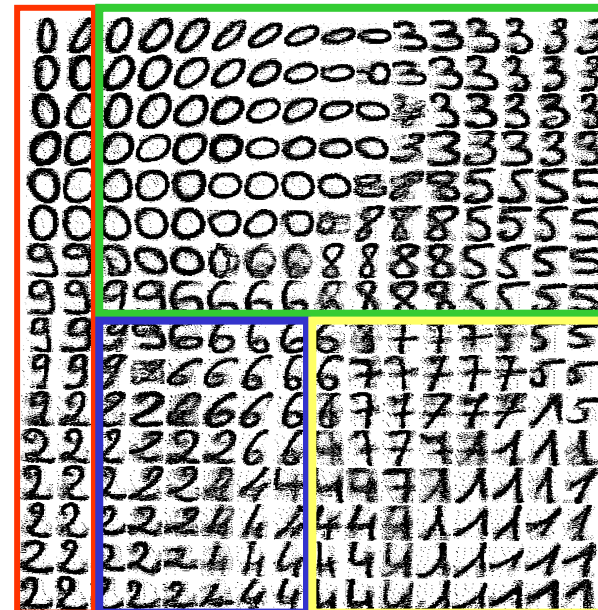
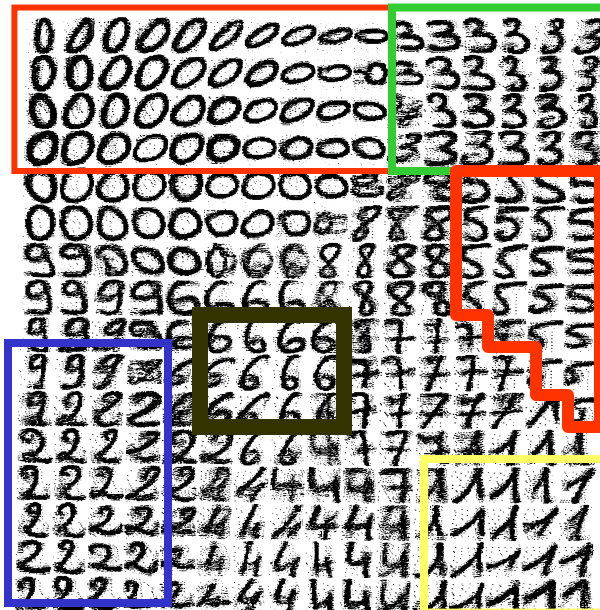
Un problème typique visuel qui pourrait relever de la problématique de la Fouille de Données plus que Reconnaissance des Formes

On donne ces données stockées sur des supports électroniques hétérogènes et non centralisés :



0000000000-3333333333
0000000000-0333333333
0000000000-3333333333
0000000000-3333333333
0000000000-8888555555
0000000000-8888555555
99990000668888555555
999966668888555555
9999666677775555
9999666677777755
2222666677777715
2222226647771111
22222244447711117
2222244444111111
2222244444111111
2222244444111111

Alors sans intervention de type supervisé (cad sans apprentissage avec exemples), le système parvient à détecter (structurer, extraire) la présence de 10 formes différentes sans forcément les reconnaître, ou bien de 4 scripteurs différents sans forcément les identifier dans un premier temps :



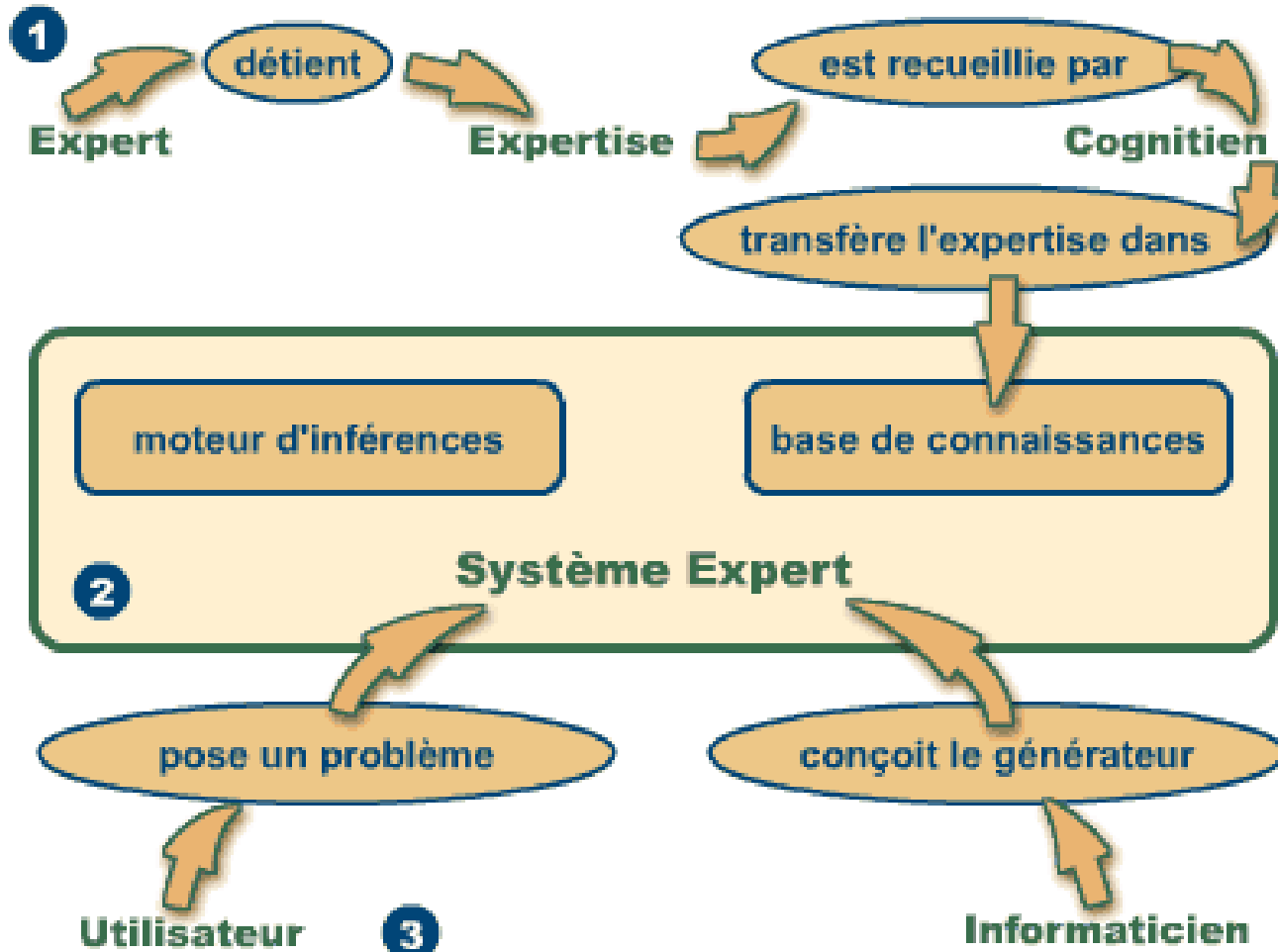
Outre le paradigme de Reconnaissance des Formes, cette intégration nouvelle ou ce paradigme nouveau est la résultante de problématiques arrivées à maturité ou à leur limite comme :

- Les systèmes experts issus de l'IA
- Les bases et les entrepôts de données
- Les protocoles réseaux normalisés

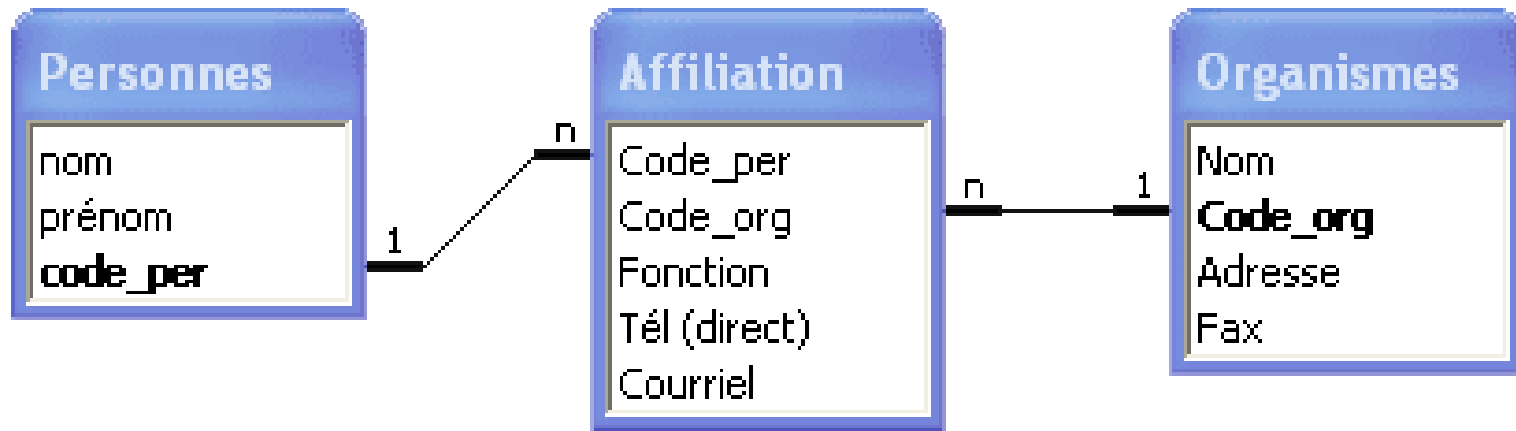
Créer une intelligence des systèmes, avec les potentialités de chacun des outils technologiques intégrés -> le rêve de système pensant plus que pensé

Différence de points de vue entre : SELECTIONNE moi les NOMS des CLIENTS ayant acheté du NUTELLA et du SAVON (requête de type SQL) et je (le logiciel) te (l'utilisateur du logiciel) fais remarquer que les clients qui achète du Nutella achètent aussi du Savon

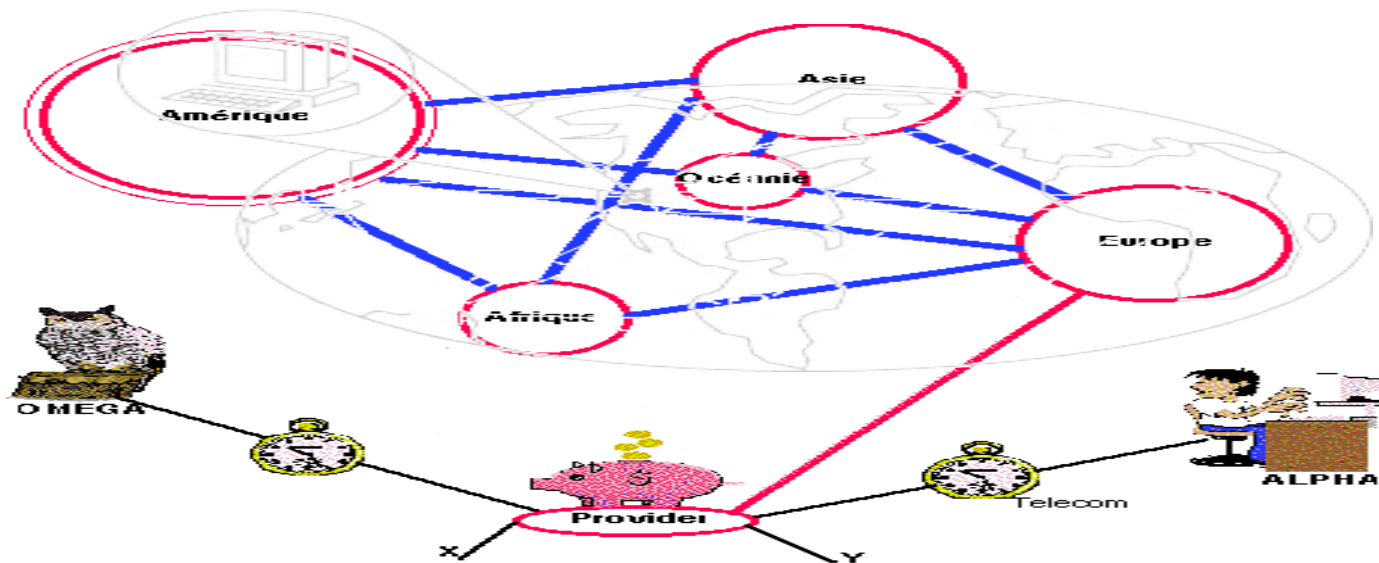
Systeme Expert Classique



Base de Données Classique

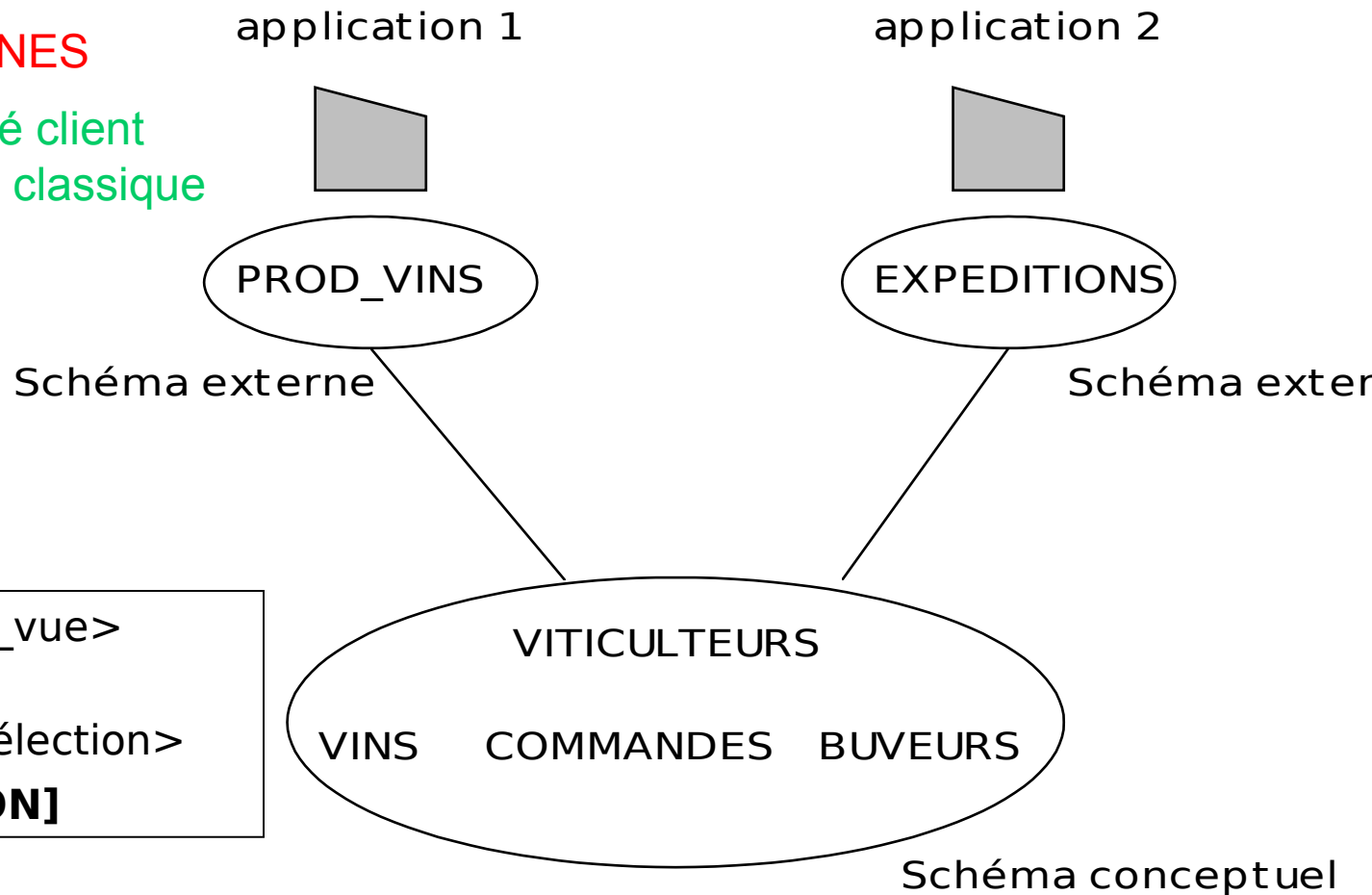


Protocole de Communication Réseau Classique



LES VUES EXTERNES

Les Virtuelles ... côté client
dans le cadre du OLTP classique



```
CREATE VIEW <nom_vue>  
[(liste_attributs)]  
AS <expression_de_sélection>  
[WITH CHECK OPTION]
```

- Recalculé à chaque transaction l'impliquant
- L'expression de sélection peut porter sur des tables de base et/ou des vues
- **[WITH CHECK OPTION]** : à manipuler avec des pincettes car le problème des mises à jour de la base au travers des vues est loin d'être efficacement résolu

LES VUES EXTERNES

Les Concrètes ... côté serveur

Dans le cadre de l'OLAP, ROLAP, MROLAP

Pour le data warehouse

```
CREATE CONCRETE VIEW  
<nom_vue> [(liste_attributs)]  
AS <expression_de_sélection>
```

- Vue Stockée en dur
- Si Vue souvent utilisée;
- Si Tables sources peu modifiées;
- Alors Mise à Jour par TRIGGER ou déclencheurs mais pas automatique;
- Alors Vues orientées objets au-dessus des BDR.

OLTP versus OLAP

Caractéristiques	OLTP	OLAP
Opérations typiques	Mise à jour	Analyse
Type d'accès	Lecture et écriture	Lecture
Niveau d'analyse	Elémentaire	Global
Ecrans	Fixe	Interactif
Quantité d'info échangée	Faible	Importante
Orientation	Ligne	Multi-dimensions
Taille BD	100MB-GB	1GB - TB
Ancienneté des données	Récente	Historique

Motivations des entreprises

- Besoin des entreprises
 - accéder à toutes les données de l'entreprise
 - regrouper les informations disséminées dans les bases
 - analyser et prendre des décisions rapidement (OLAP)
- Exemples d'applications concernées
 - Bancaire : suivi des clients, gestion de portefeuilles
 - mailing ciblés pour le marketing
 - Grande distribution : marketing, maintenance, ...
 - produits à succès, modes, habitudes d'achat
 - préférences par secteurs géographiques
 - Télécommunications : pannes, fraudes, mobiles, ...
 - classification des clients, détection fraudes, fuites de clients, etc.
 - Médecine, Pharmacie, Bourse, Production, ...

L'approche entrepôt de données

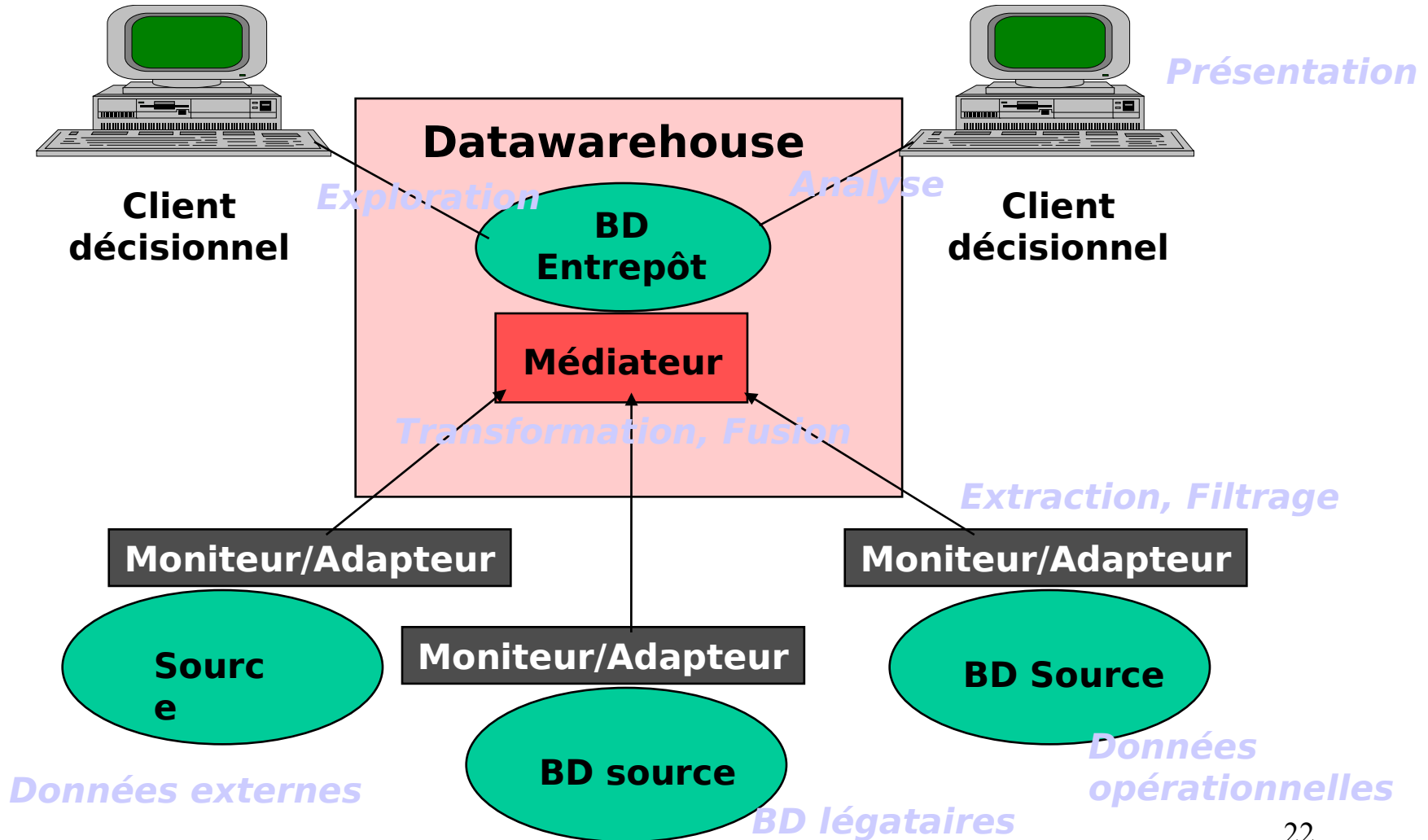
- Datawarehouse

- Ensemble de données historisées variant dans le temps, organisé par sujets, consolidé dans une base de données unique, géré dans un environnement de stockage particulier, aidant à la prise de décision dans l'entreprise.

- Trois fonctions essentielles :

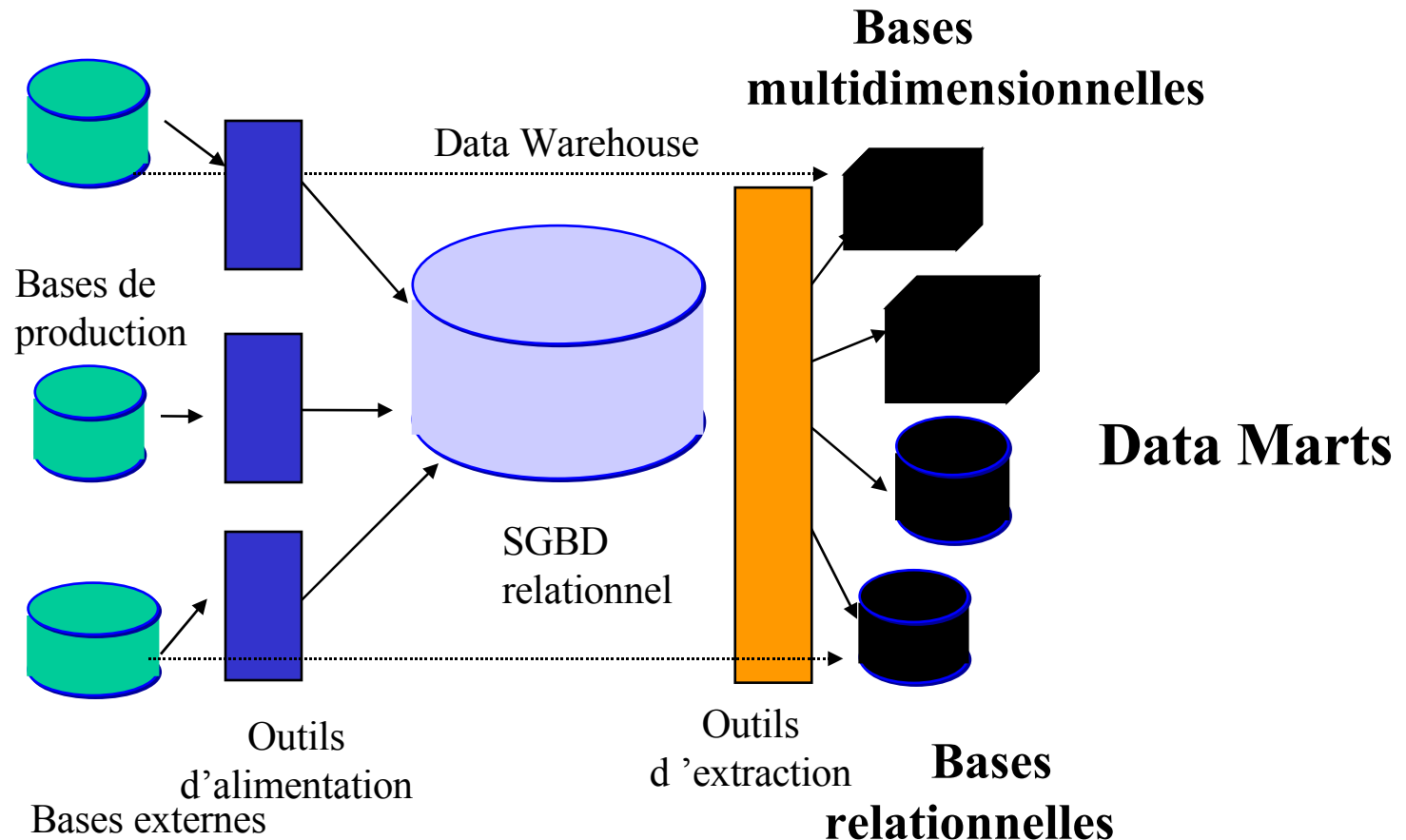
- collecte de données de bases existantes et chargement
 - gestion des données dans l'entrepôt
 - analyse de données pour la prise de décision

Architecture type



Datamart (Magasin de données)

- sous-ensemble de données extrait du datawarehouse et ciblé sur un sujet unique



Modélisation multidimensionnelle

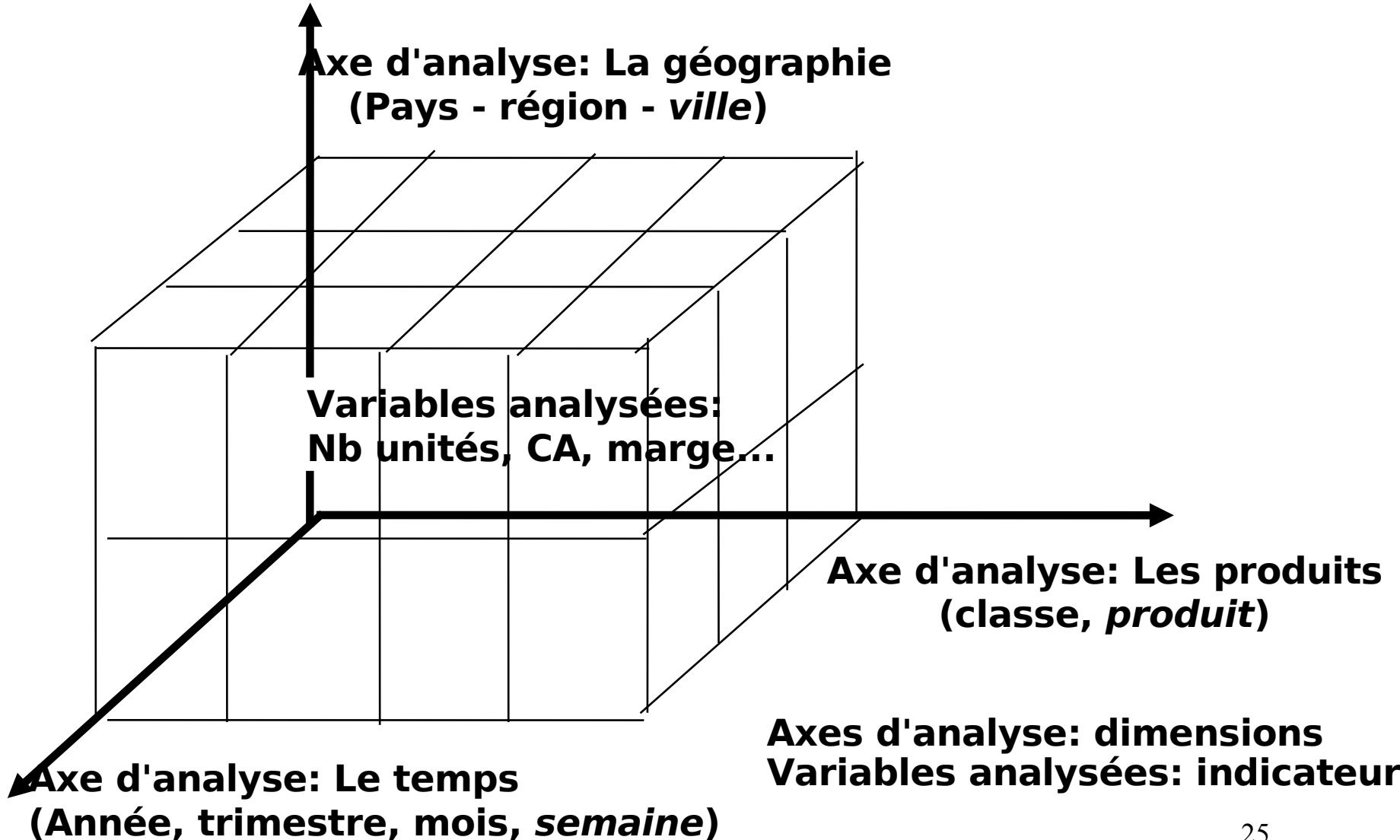
- Dimensions:

- Temps
- Géographie
- Produits
- Clients
- Canaux de ventes.....

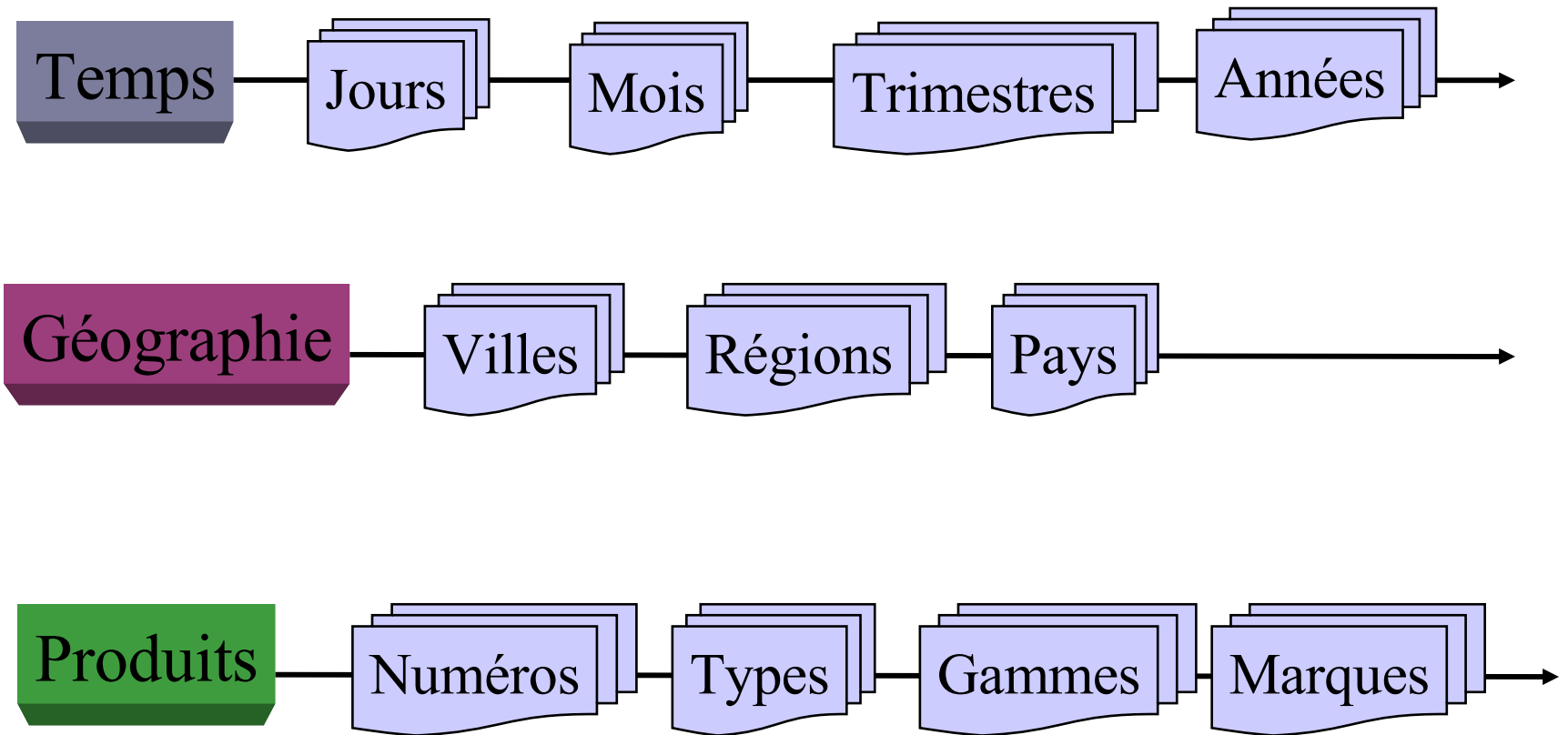
- Indicateurs:

- Nombre d'unités vendues
- CA
- Coût
- Marge.....

Le data cube et les dimensions

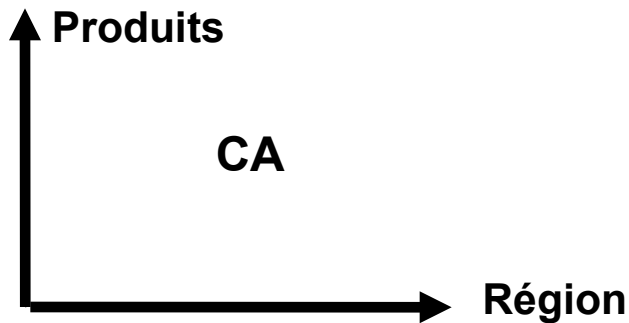


La granularité des dimensions

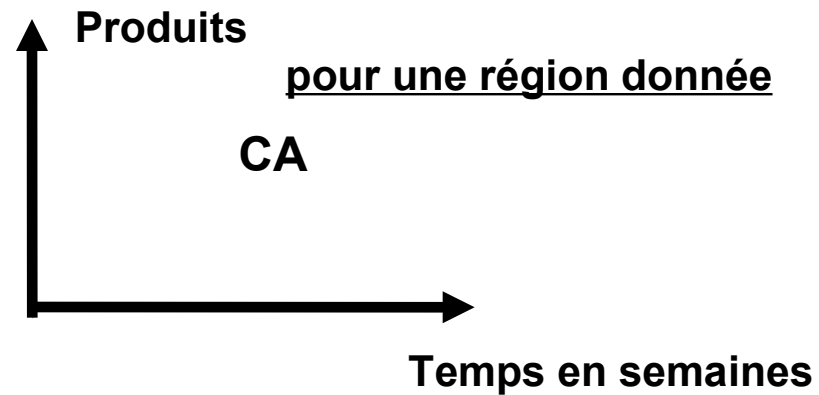


La navigation multidimensionnelle

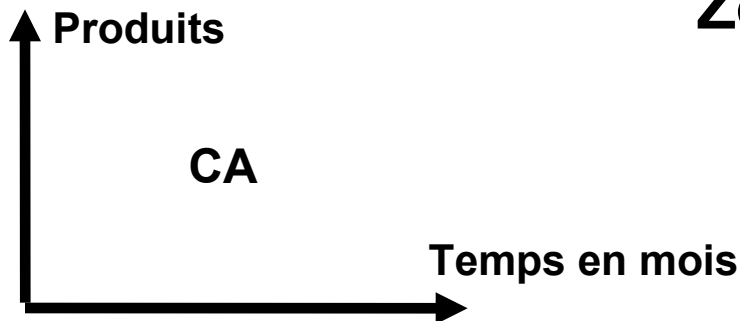
Projection en 2 dimensions



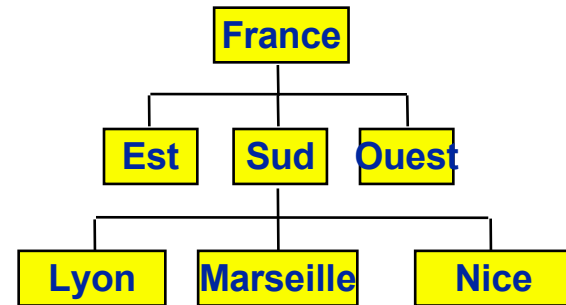
Coupe d'un cube



Réduction selon 1 dimension



Zoom selon une dimension



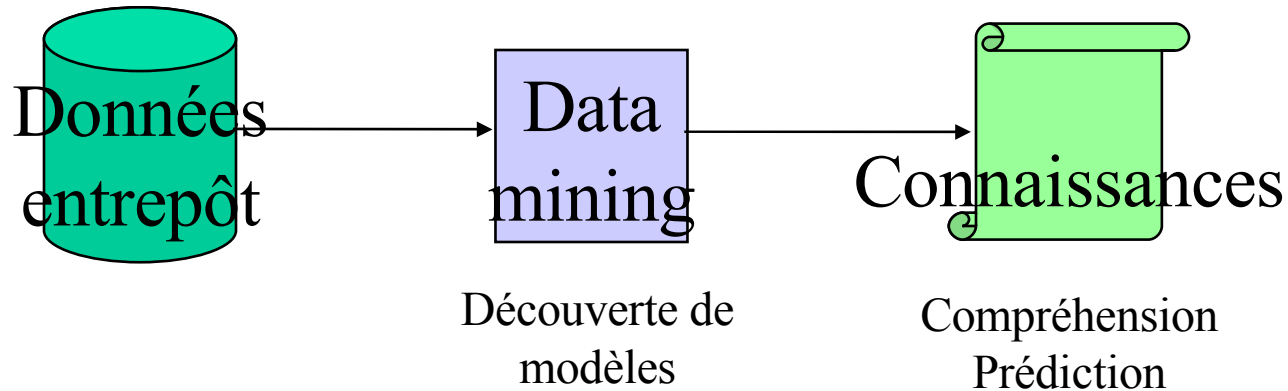
Bilan OLAP

- La modélisation multidimensionnelle est adaptée à l'analyse de données
- Le datacube est au centre du processus décisionnel
 - transformation et visualisation 3D
 - une algèbre du cube :
 - Slice, Dice, Rollup, Drilldown (SQL spécifique)

Qu 'est-ce-que le data mining ?

- Data mining

–ensembles de techniques d'exploration de données afin d'en tirer des connaissances (la substantifique moelle) sous forme de modèles présentées à l 'utilisateur averti pour examen



- Connaissances

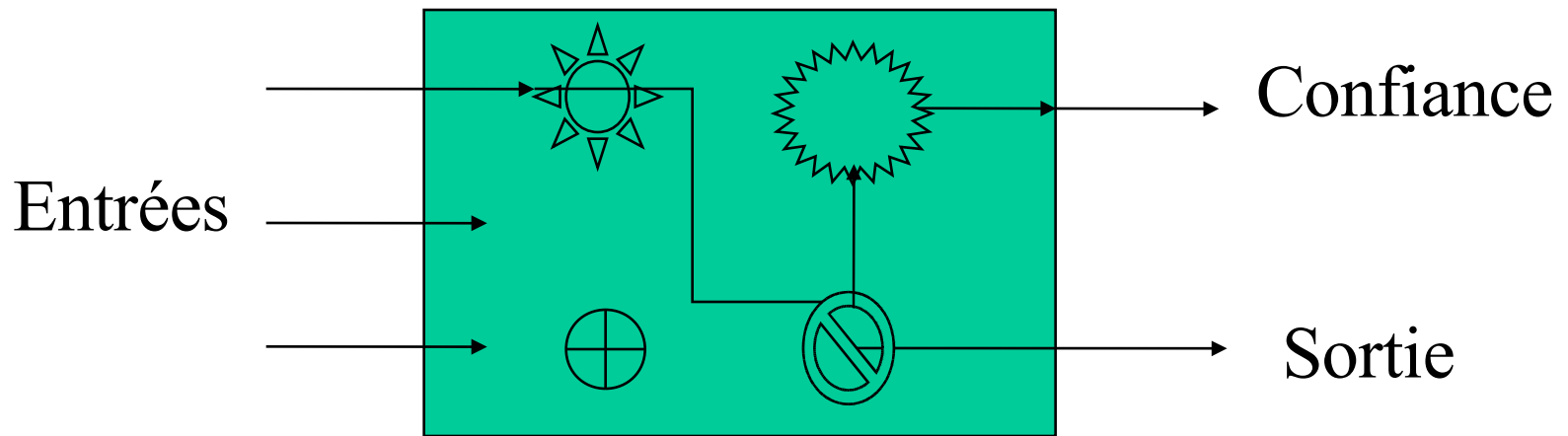
–analyses (distribution du trafic en fonction de l 'heure)
–scores (fidélité d 'un client), classes (mauvais payeurs)
–règles (**si** facture > 10000 **alors** départ à 70%)

Mécanismes de base

- **Déduction** : base des systèmes experts
 - schéma logique permettant de déduire un théorème à partir d'axiomes
 - le résultat est sûr, mais la méthode nécessite la connaissance de règles
- **Induction** : base du data mining
 - méthode permettant de tirer des conclusions à partir d'une série de faits
 - généralisation un peu abusive
 - indicateurs de confiance permettant la pondération

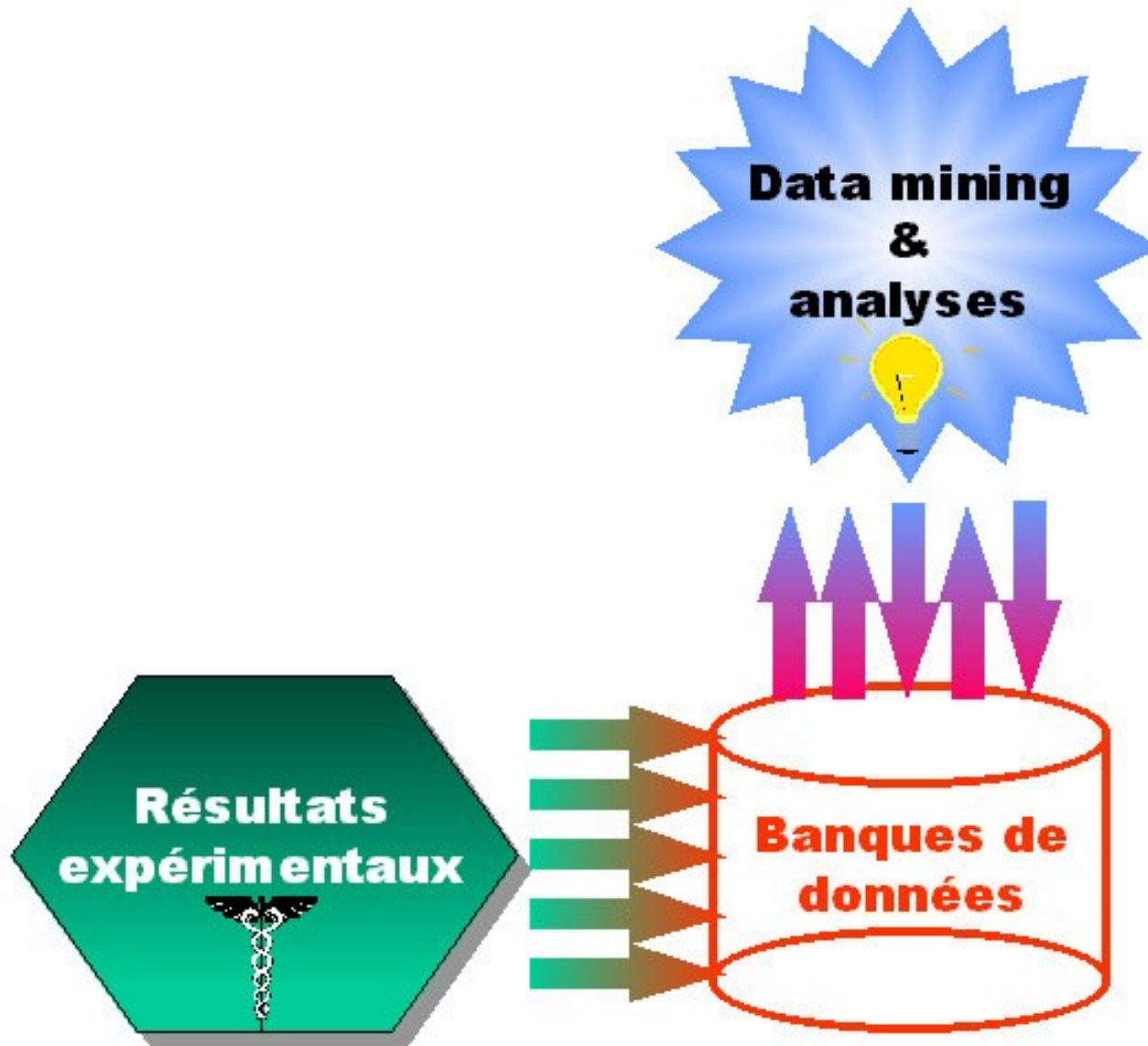
Découverte de modèles

- Description ou prédiction

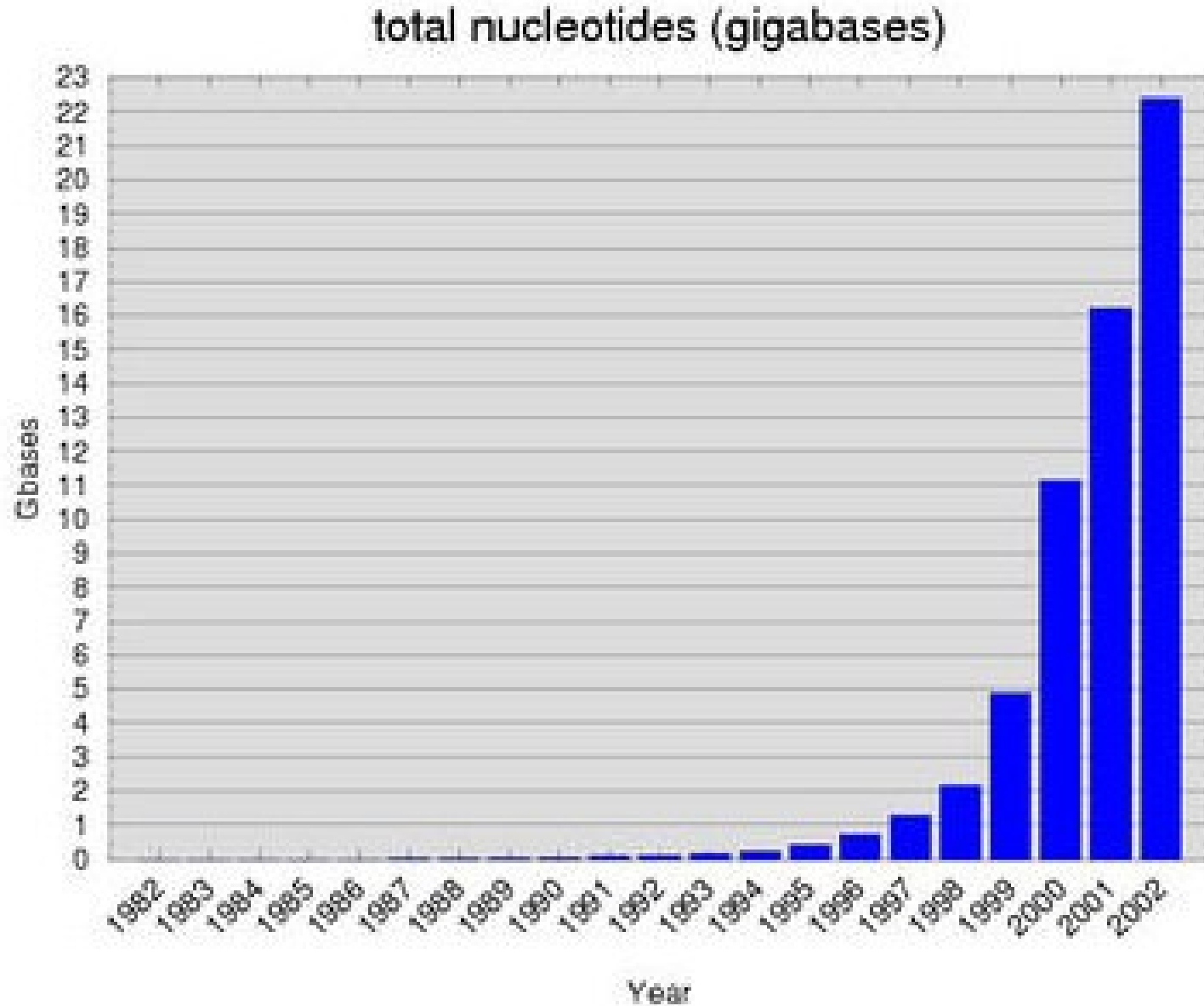


- Apprentissage sur la base
- Utilisation pour prédire le futur
- Exemple : régression linéaire $Y = a X + B$

Le matériel biologique



A ce compte là, il ne s'agit plus d'apprendre donc de reconnaître mais déjà de comprendre donc de structurer



Comment analyser, visualiser, structurer des grandes masses de données réparties, hétérogènes

>cDNA inconnu

AATGCAAGTGCATGCATGCATGCATCGGATCGTACGGATTGCAGTTCGGATTCATAATAA
ATGCGTAAAAACAGTAGTTTCACTAGTTTCAAAGTTGCATAATACTTGCTGTTCTTCTT
GTTTACCCTAACAGTATGGCTGTTTTCGCTGTTGCTGCTGACGGTATACCTTTCCCTTAC
CACGCTAAATACAGTAACGGTGTCTATAAGTCCTCTTCACGTTACTCAAAGTAGTGGTAAC
AGTAGTGTTAAAGCTGAATGGGAACAATGGAAAAGTGCTCACATAACTAGTGACCTTAAC
GGTGCTGGTGGTTACAAATACGTTCAACGTGACATAAACGGTAACACTGACGGTGTTAGT
GAAGGTCTTGGTTACGGTCTTATAGCTACTGTTTGCTTCAACGGTGCTGACAGTAACGCT
CAAACCTTTACGACGGTCTTTACAAATACGTTAAAAGTTTCCCTAGTGCTAACAACCCT
AACCTTATGGGTTGGCACATAAACAGTAGTAACAACATAACTGAAAAAGACGACGGTATA
GGTGCTGCTACTGACGCTGACGAAGACATAGCTGTTAGTCTTATACTTGCTCACAAAAAA
TGGGGTACTAGTGGTAAAATAAACTACCTTAAAGCTGCTCGTGACTACATAAACAAAAAC
ATATACGCTAAAATGGTTGAACCTAACAACTACACTCTTAAACTTGGTGACATGTGGGGT
GGTAACGACTTCAAAAACGCTACTCGTCCTAGTTACTTCGCTCCTGCTCACCTTCGTATA
TTCTACGCTTACACTGGTGACAAAGGTTGGATAAACGTTGCTAACAACTTTACACTACT
GTTAACGAAGTTCGTAACAAATACGCTCCTAAAACCTGGTCTTCTTCCTGACTGGTGCGCT
GCTAACGGTACTCCTGAAAGTGGTCAAAGTTTCGACTACGACTACGACGCTTGCCGTGTT
CAACTTCGTACTGCTATAGACTACAGTTGGTACGGTGACGCTCGTGCTGCTGCTCAAAGT
GACAAAATGAACAGTTTTCATAGCTGCTGACACTGCTAAAACCCTAGTAACATAAAAGAC
GGTTACACTCTTAAACGGTAGTAAAATAAGTAGTAACCACAGTGCTAGTTTCTACAGTCCT
GCTGCTGCTGCTGCTATGACTGGTACTAACACTGCTTTCGCTAAATGGATAAACAGTGGT
TGGGACAAAGTTAAAGACAGTAAAAAATACGGTACTACGGTGACAGTCTTAAAATGCTT
ATAATGCTTTACATAACTGGTAACTTCCCTAACCCCTCTTAGTGACCTTAGTAGTCAACCT
AGTCCTGGTGACCTTAAACGGTGACGGTGAAATAGACGAACTTGACATAGCTGCTCTTAAA
AAAGCTATACTTAAACAAAGTACTAGTAACATAAACCTTACTAACGCTGACATGAACCGT
GACGGTGCTATAGACGCTAGTGACTTCGCTATACTTAAAGTTTACCTTTAAT

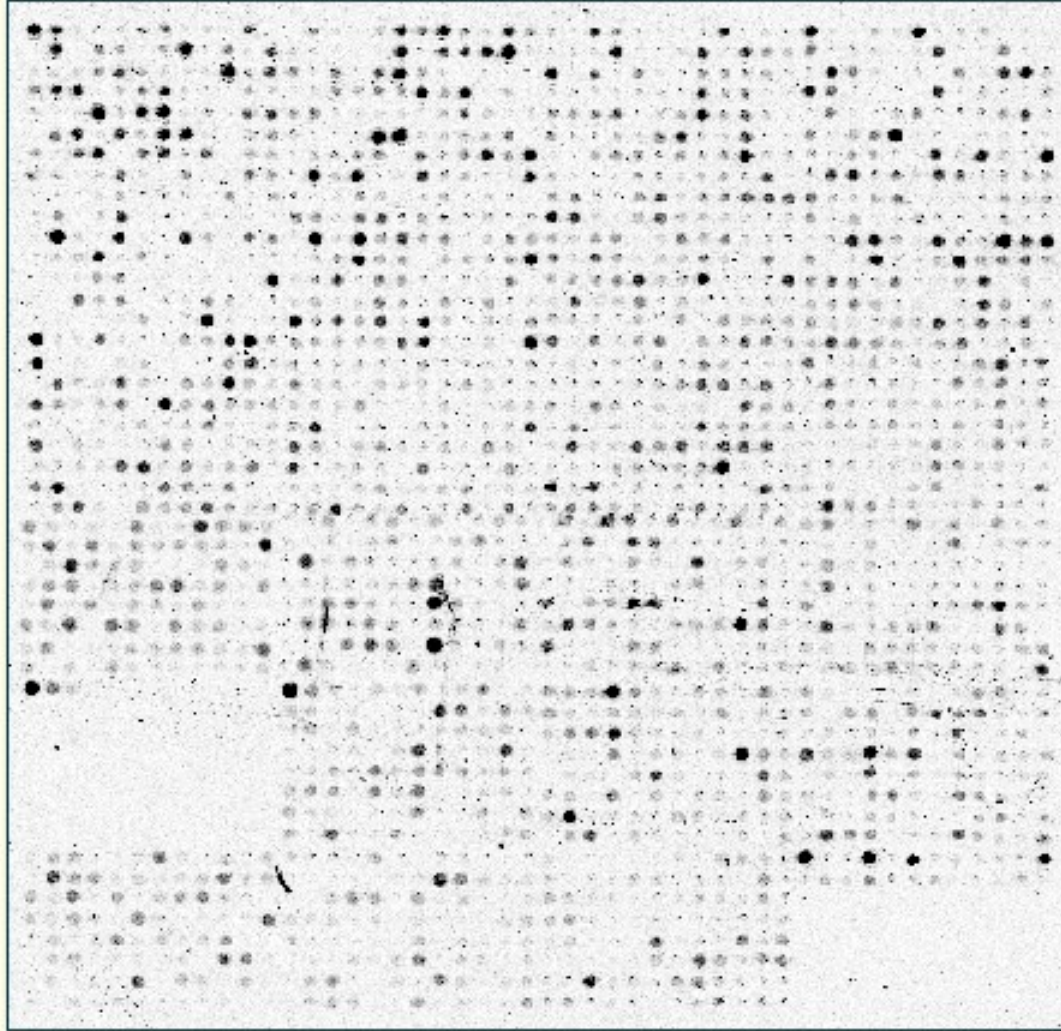
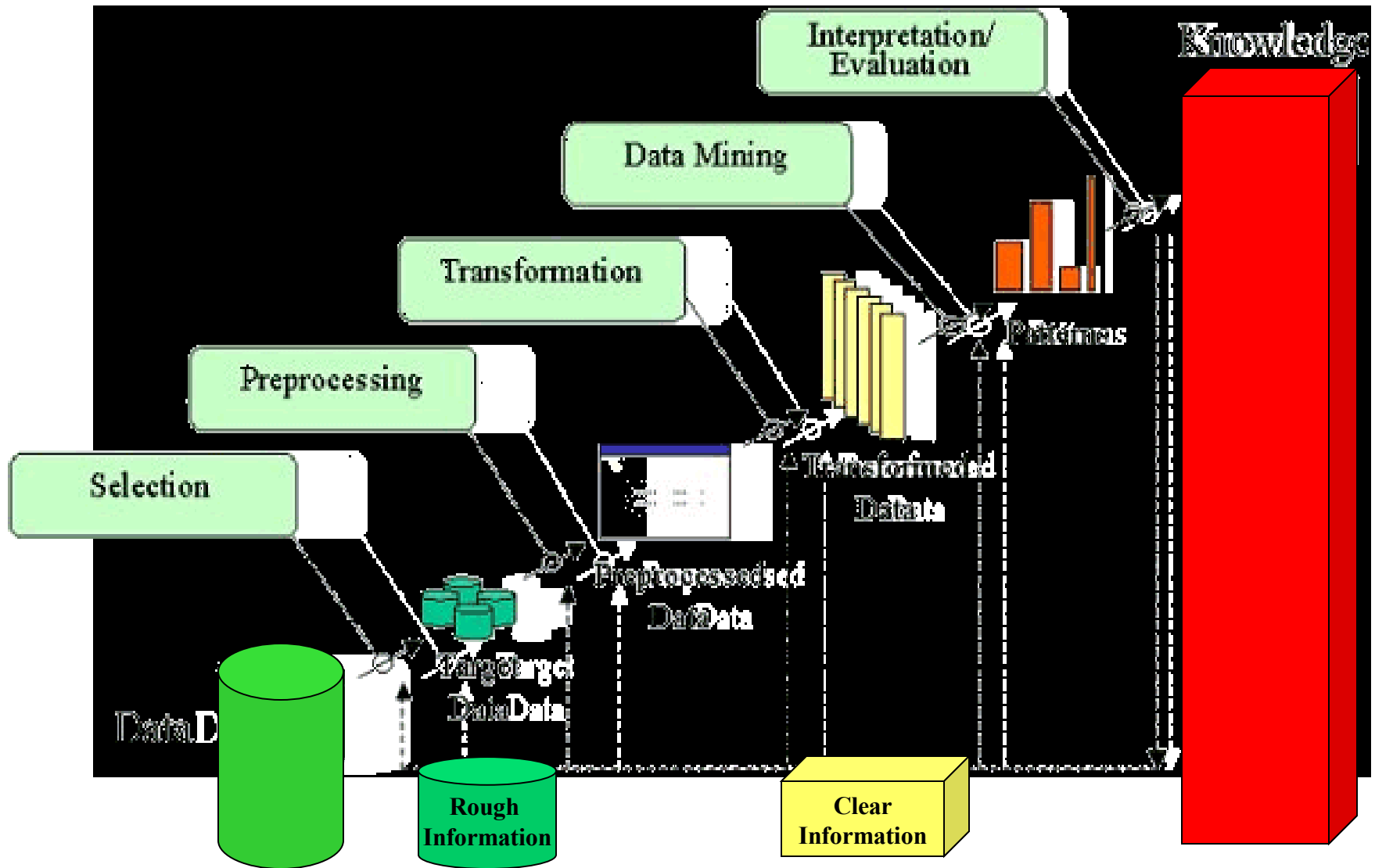


Tableau 1. Tableau récapitulatif des base de données biologiques accessibles sur l'Internet.

Biomedical literature	PubMed	www.ncbi.nlm.nih.gov/entrez/query.fcgi
Nucleic acid sequence	GenBank	Idem
	SRS at EMBL/EBI	http://srs.ebi.ac.uk
Genome Sequence	Entrez Genome	www.tigr.org/tdb
	TIGR databases	
Protein sequence	GenBank	www.expasy.ch/spro/ http://www-nbrf.georgetown.edu
	SWISS-PROT at ExPASy	
	PIR	
Protein Structure	Protein Data Bank	www.rcsb.org/pdp/
Entrez Structure DB Protein and peptide mass spectroscopy	PROWL	http://prowl.rockefeller.edu
Biochemical pathways	PathDB KEGG WIT	www.genome.ad.jp/kegg/ http://wit.mcs.anl.gov/WIT2/
Microarray	Gene Expression Links	http://industry.ebi.ac.uk/~alan/microArray
...		

Un système d'Extraction de Connaissances



Les étapes du processus de KDD

- Comprendre le domaine d'application
- Sélection d'un ensemble de données
- Nettoyage et pré-traitement des données (peut prendre 60% de l'effort)
- Choix des fonctionnalités du data mining
 - classification, consolidation, régression, association, clustering.
- Choix de(s) l'algorithme(s) d'extraction
- Data mining : Recherche des motifs (patterns) intéressants
- Évaluation des Patterns et présentation
 - visualisation, transformation, suppression des patterns redondants, etc.
- Utilisation de la connaissance extraite

Des techniques issus de l'IA et de la RF

Machine learning techniques such as :

- **Arbre de décision**
- **Réseaux de neurones**
- **Règles d'association**
- **Clustering**

Des systèmes combinant les technologies Réseaux et BD

- **SQL**
- **FTP**
- **TCP/IP**
- **Php / mySQL**

Des champs d'applications très diversifiés

- **Commerce – Economie**
- **Bio-informatique**
- **Web Mining et Marketing**
- **Médecine**

Principe global

L'importance pratique et industrielle des procédés d'analyse automatique et intelligente de données, textes, images, sons ou enregistrements électroniques est telle que beaucoup de recherches spécialisées se sont développées.

Nous cherchons ici à en donner une idée et à en dégager les points communs qui sont propre de la méthodologie de la Fouille de Données.

C'est essentiellement dans la conception des processus de discrimination (ou d'affectation à diverses catégories) que l'on retrouve une méthodologie commune, à quelques variantes près.

En gros une telle fonctionnalité est constituée de plusieurs composantes, correspondant à plusieurs phases de traitement. On en distinguera essentiellement deux, les autres pouvant s'échelonner entre les deux extrêmes :

1. Le prétraitement
2. La découverte de catégories proprement dite

Un fait remarquable en FD est que chaque application fait appel à plusieurs techniques parmi celles présentées ici, avec une interrelation parfois surprenante où l'invention et le flair de l'ingénieurs sont rois.

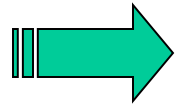
Des algorithmes

Une évolution plus qu'une
révolution

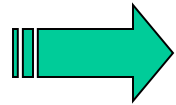
Un cocktail de techniques

Des algorithmes

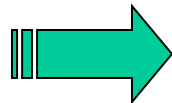
D'inspirations ...



Mathématiques : stat. et AD

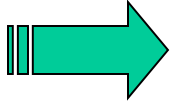


Calculatoires



Biologiques

Des algorithmes



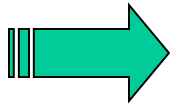
Calculatoires

« *Clustering* »

Arbres de décision

Règles d'association

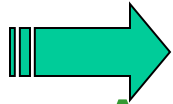
Programmation dynamique



Biologiques

Réseaux de neurones
Algorithmes génétiques

Des algorithmes



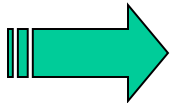
Non Supervisés

Apprentissage *a priori* en mode Découverte

« *Clustering* »

Algorithmes génétiques

Règles d'association



Supervisés

Apprentissage *a posteriori* en mode Reconnaissance -
Prédiction

Réseaux de neurones

Arbres de décision

Programmation dynamique

Cas pratique avec les mains

Cas de marketing classique : identification de profils de clients et organisation d'une campagne de marketing direct

Il s'agit d'un voyageur organisant des circuits touristiques avec 5 types de prestation (A,B,C,D et E). Son directeur marketing souhaite mettre en place une politique de fidélisation.

Décomposition en sous-problèmes précis :

Fidéliser la clientèle ?

Clientèle ?

Typologie de problèmes à résoudre :

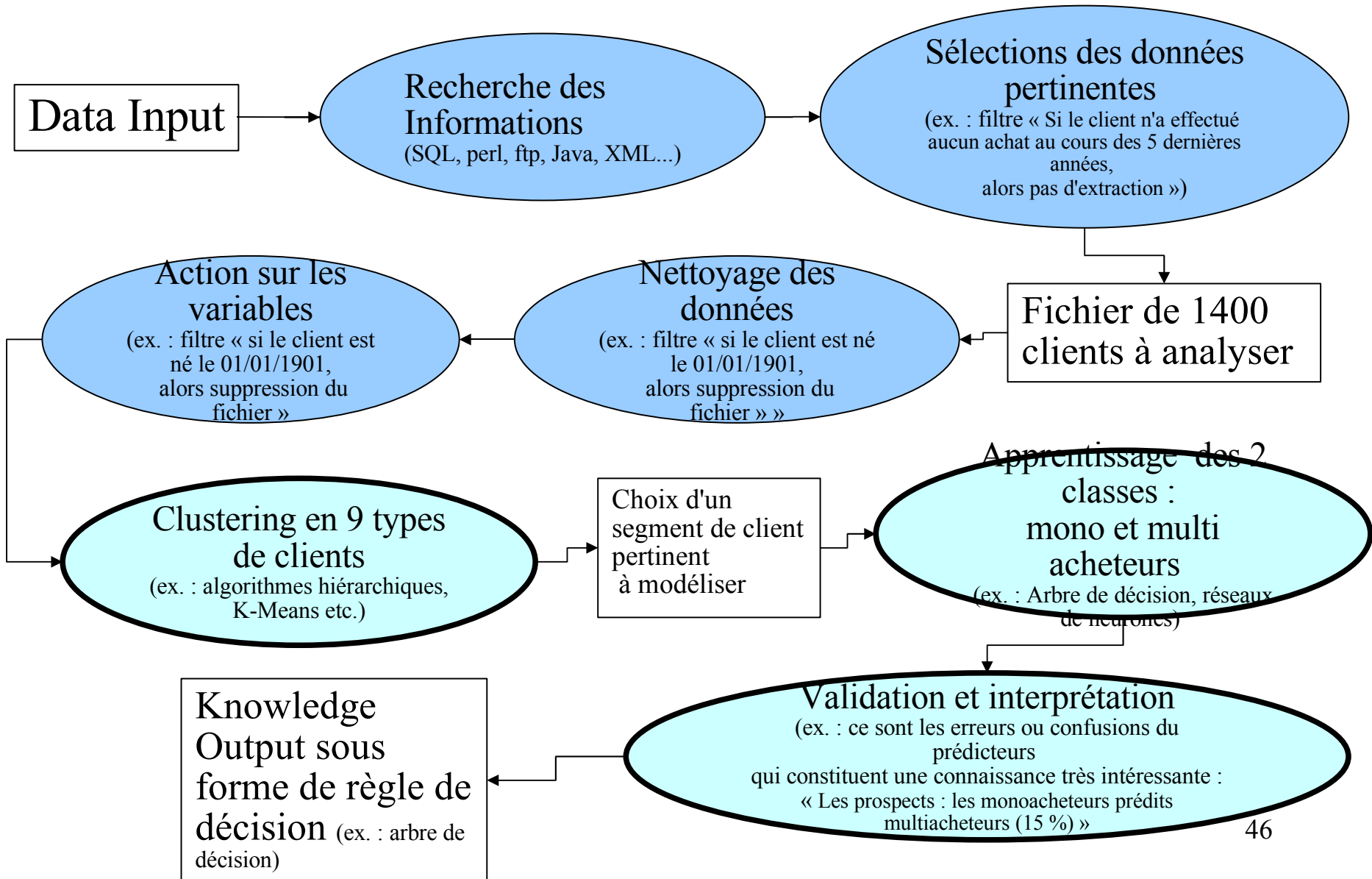
Structuration : qui sont mes clients ?

Affectation : quels sont les clients à contacter ?

Objectifs :

1. Connaître les clients pour revoir les offres et la politique marketing.
2. Fournir à la cellule marketing opérationnelle et aux réseaux de distribution une liste ciblée de clients par requête SQL, ce qui implique des critères compréhensibles

Cas pratique avec les mains



Chapitre I. Apprentissage *a priori* par Similitude

Clustering, Regroupement et Similitude

Voici un des enjeux du processus de data mining : comment comparer des éléments disparates pour les rassembler ou au contraire les différencier ?

Objectifs :

- Comprendre la notion d'apprentissage **NON supervisé**
- Le lier à la notion de **Découverte** de Structures
- Connaître des algorithmes de regroupement
 - Hiérarchiques et leur représentation graphique : **dendrogramme**
 - par Optimisation type **K-Means**, ISODATA
- Comprendre que la notion de Similitude liée à la vaste notion mathématique de Distance est **subjective** mais centrale dans cette problématique
- Savoir **construire un espace** de mesure multi-dimensionnelle et **définir une mesure de similarité** dans cette espace
- Savoir **choisir l'algorithme** à utiliser en fonction des données en entrée

Principes

Contexte non supervisé

« Révéler » l'organisation de motifs
en groupes cohérents

Subjectif

Disciplines

Biologie
Zoologie
Psychiatrie
Sociologie
Géologie
Géographie

Synonymes

Apprentissage non supervisé
Taxonomie
Typologie
Partition

Définition

Le clustering consiste à construire un classificateur intrinsèque, sans classes connues a priori, par apprentissage à partir d'échantillons donnés.

Le nombre de classes possibles est en général connu et on veut construire le classificateur $K : M \rightarrow \{1..c\}$ partitionnant l'espace M des mesures.

Hypothèse

Plus deux échantillons sont proches dans M , plus leur probabilité d'appartenir à la même classe est grande.



$\xrightarrow{\text{ko}/(\text{largeur} * \text{hauteur})}$? $\xrightarrow{\text{bmp} \quad \text{gif}}$ Format



%Contour
 ↑
 Ng moyen →



Notion d'espace, de dimension ... Image $\in \mathbb{C} \times \mathbb{N} \times \mathbb{F}$ ⁵²

Cadre numérique

Vecteurs de caractéristiques

$$V(x_1, x_2, x_3, x_4)$$

Différentes natures des caractéristiques

Continue

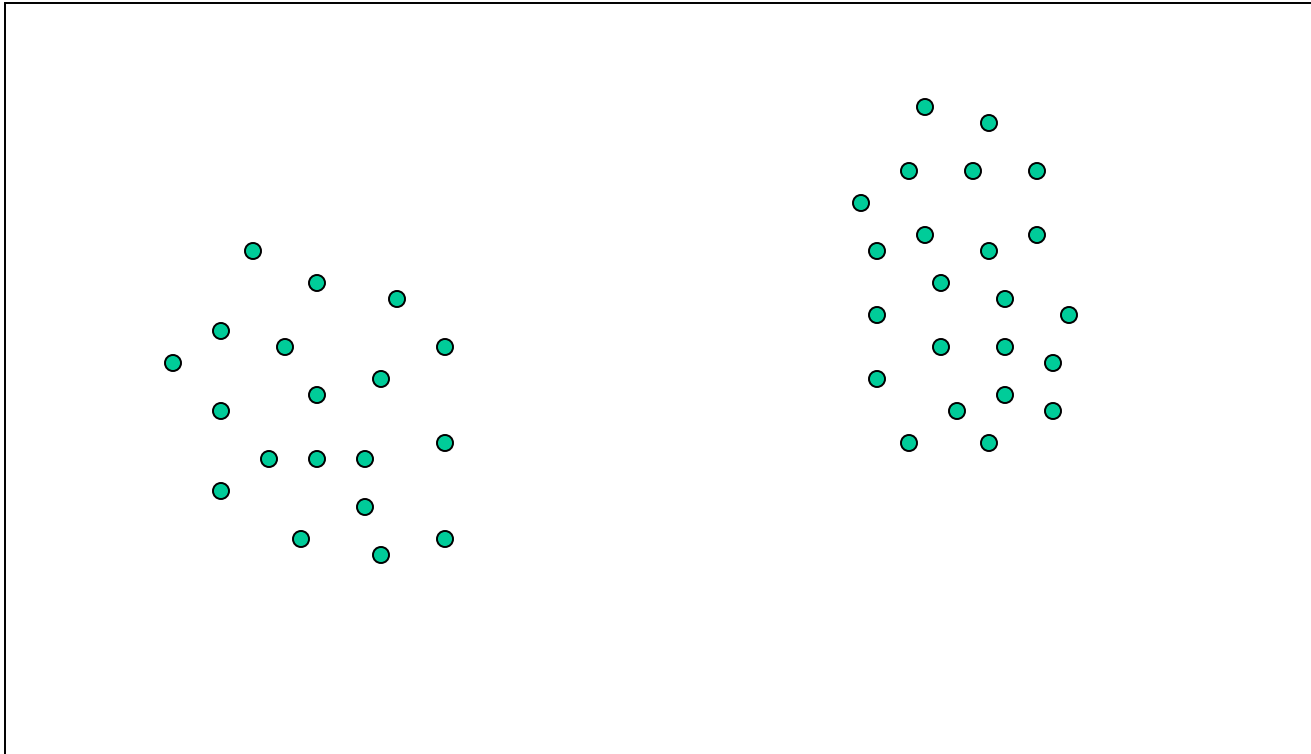
Discrète

Ordinal

Nominal

...

« Clustering »



Imaginez un lien avec une requête SQL



Algorithme Séquentiel Basique

(ASB) :

INPUT : $S = \{x_1, x_2, \dots, x_N\}$, seuil de distance Θ et seuil de nombre de classes q

- $m=1$
- $C_m = \{x_1\}$
- For $i=2$ to N
 - Find $C_k : d(x_i, C_k) = \min_{1 \leq j \leq m} d(x_i, C_j)$
 - If $(d(x_i, C_k) > \Theta)$ AND $(m < q)$ then
 - ❖ $m = m+1$
 - ❖ $C_m = \{x_i\}$
 - Else
 - ❖ $C_k = C_k \cup \{x_i\}$
 - ❖ Where necessary, update representatives.
 - End {If}
- End {For}

OUTPUT : une classification dure $R = \cup C_i$

« Clustering »

Algorithme Séquentiel Basique Modifié (ASBM) :



Cluster Determination

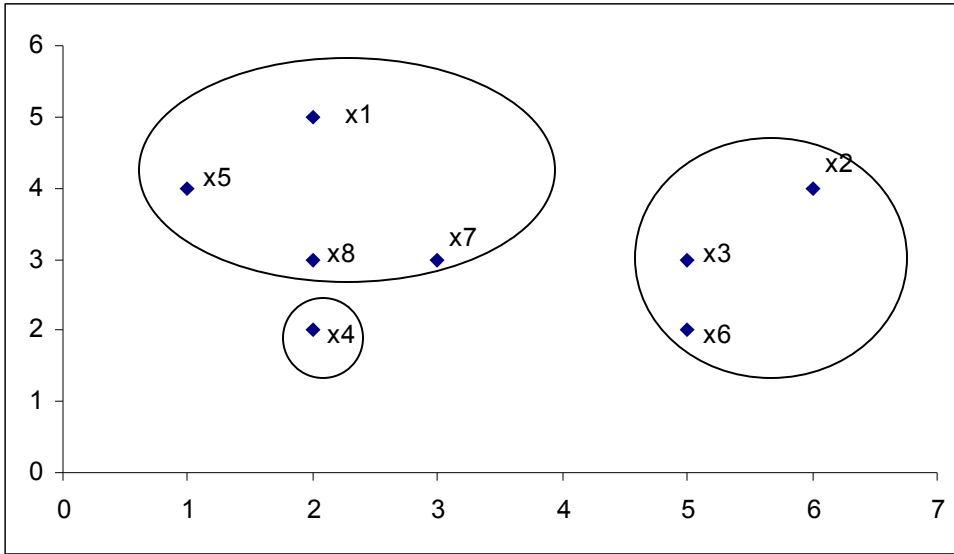
- $m=1$
- $C_m = \{x_i\}$
- For $i=2$ to N
 - Find $C_k : d(x_i, C_k) = \min_{1 \leq j \leq m} d(x_i, C_j)$
 - If $(d(x_i, C_k) > \Theta)$ AND $(m < q)$ then
 - ❖ $m = m+1$
 - ❖ $C_m = \{x_i\}$
 - End {If}
- End {For}

Pattern Classification

- For $i=1$ to N
 - If x_i has not been assigned to a cluster, then
 - ❖ Find $C_k : d(x_i, C_k) = \min_{1 \leq j \leq m} d(x_i, C_j)$
 - ❖ $C_k = C_k \cup \{x_i\}$
 - ❖ Where necessary, update representatives
 - End {If}
- End {For}

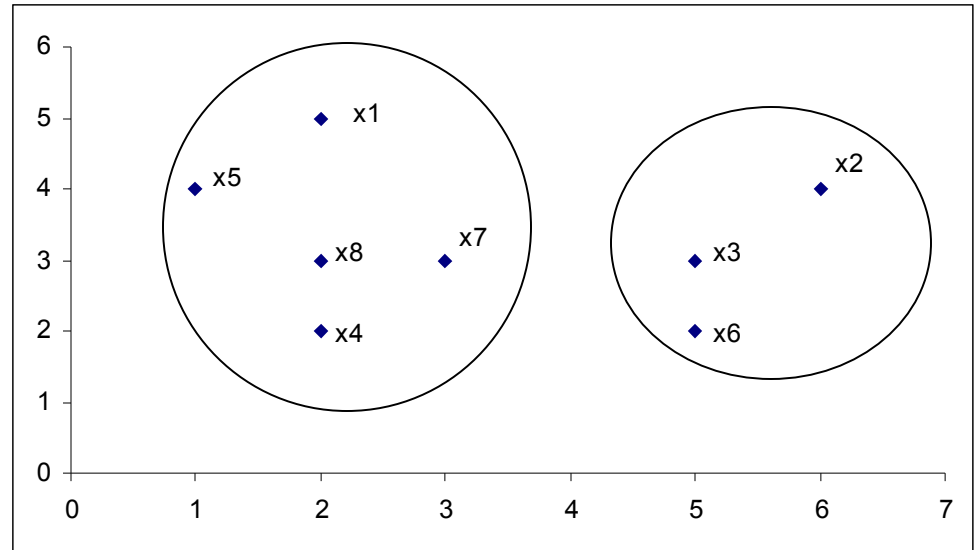
Remarque :

Si $d(x, C) = d(x, m_C)$ alors $m_{C_k}^{new} = \frac{(n_{C_k}^{new} - 1)m_{C_k}^{old} + x}{n_{C_k}^{new}}$



Ordre :
x1,x2,x3,x4,x5,x6,x7,x8

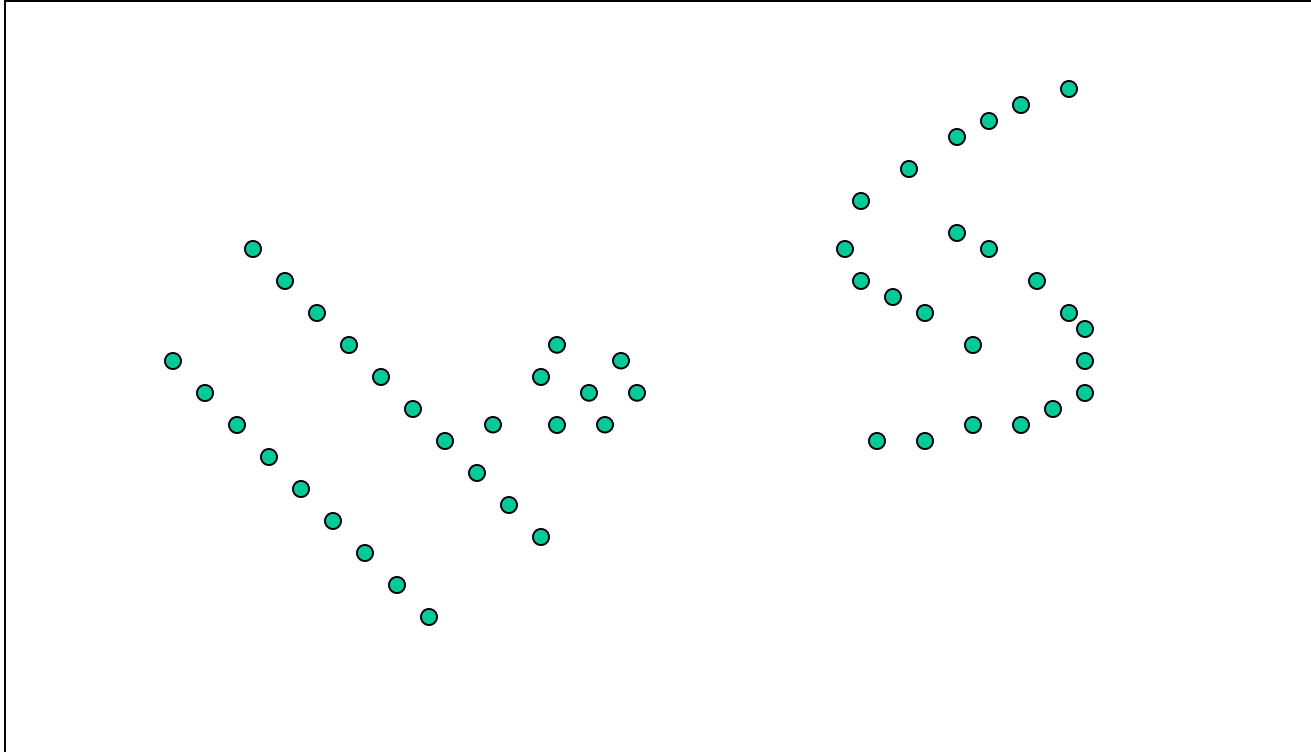
Ordre :
x1,x2,x5,x3,x8,x6,x7,x4



Notes sur l'algorithme ASB :

- Cet algorithme séquentiel est bien adapté pour traiter des échantillons à mesure de leur acquisition (analyse on-line)
- mais fournit des résultats dépendant de l'ordre de présentation (arbitraire)

« Clustering »



« Clustering »

Quantifier la similarité entre :

➡ 2 vecteurs de caractéristiques

➡ 2 ensembles de vecteurs de caractéristiques

« Clustering »

Hard Clustering

Soit $X = \{x_1, \dots, x_N\}$

On appelle *m-clustering* de X la partition de X en m ensembles (clusters) C_1, \dots, C_m tels que :

$$C_i \neq \emptyset, i = 1, \dots, m$$

$$\bigcup_{i=1}^m C_i = X$$

$$C_i \cap C_j = \emptyset, i \neq j, i, j = 1, \dots, m$$

Fuzzy Clustering

Zadeh (1965)

Soit $X = \{x_1, \dots, x_N\}$

Le *m-clustering flou* de X en m clusters est caractérisé par m fonctions d'appartenance u_j avec :

$$u_j : X \rightarrow [0,1], j = 1, \dots, m$$

$$\sum_{j=1}^m u_j(x_i) = 1, i = 1, \dots, N$$

$$0 < \sum_{i=1}^N u_j(x_i) < N, j = 1, \dots, m$$

« Clustering »

Mesures de
proximité

Mesure de (dis)similarité
Mesure de (dis)similarité
métrique ou Distance

Distinction importante surtout d'un point de vue théorique et de propriétés de convergence des algorithmes

Mesure de
(dis)similarité
métrique

$d : X \times X \rightarrow \mathfrak{R}$ telle que :

$$\forall x, y \in X, d(x, y) \geq 0$$

$$d(x, x) = 0 \quad \forall x \in X$$

$$d(x, y) = d(y, x) \quad \forall x, y \in X$$

$$\left\{ \begin{array}{l} d(x, y) = 0 \Leftrightarrow x = y \\ d(x, z) \leq d(x, y) + d(y, z) \quad \forall x, y, z \in X \end{array} \right.$$

« Clustering »

A partir de maintenant,

Point = vecteur de caractéristiques

Ensemble = ensemble de vecteurs de caractéristiques

Mesures de proximité entre 2 points

Valeurs réelles

$$d_p(x, y) = \left(\sum_{i=1}^N w_i |x_i - y_i|^p \right)^{1/p}$$

Distances de
Mahalanobis,
euclidienne,
Manhattan,
infini

$$S_{inner} = x^T y \text{ (corrélation)}$$

$$S_{Tanimoto} = \frac{x^T y}{\|x\|^2 + \|y\|^2 - x^T y}$$

Valeurs discrètes

Les coordonnées des vecteurs appartiennent à un ensemble fini $F = \{0, 1, \dots, k-1\}$, $k \geq 0$

Si $x, y \in F^l$, on définit la matrice

de contingence $A(x, y)_{k \times k} = [a_{ij}]$ par :

a_{ij} = nombre de places où le premier vecteur a le symbole i et l'élément correspondant du second vecteur a le symbole j

$$d_{\text{Hamming}}(x, y) = \sum_{i=0}^{k-1} \sum_{j=0, j \neq i}^{k-1} a_{ij}$$

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 😞

z-scoring

	Age	Salaire
Personne1	-2	-0,5
Personne2	2	0,18
Personne3	0	0,32
Personne4	0	0

—————→ $d(p1,p2)=4,675$

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

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

$$m_{\text{age}} = 60, s_{\text{age}} = 5$$

$$m_{\text{salaire}} = 11074, s_{\text{salaire}} = 48$$

Exemple: le problème de normalisation des données : cas des intervalles

Il faut standardiser les données en calculant une mesure normalisée par la moyenne et l'écart type du *feature* f : ce qu'on appelle le z-score.

Soit la matrice de n données suivantes :

On définit alors la matrice standardisée des z-scores par :

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

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{i1} & x_{i2} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

Avec m_f la moyenne du feature en colonne et s_f son écart type absolu moyen (plus robuste que la variance classique)

Exemple : Variables binaires

- Une table de contingence pour données binaires

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

		Objet <i>j</i>		<i>Sum</i>
		1	0	
Objet <i>i</i>	1	<i>a</i>	<i>b</i>	<i>a+b</i>
	0	<i>c</i>	<i>d</i>	<i>c+d</i>
<i>Sum</i>		<i>a+c</i>	<i>b+d</i>	<i>p</i>

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

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

Exemple : variable binaire et Mesures de distances

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

$$d(i, j) = \frac{(b + c)}{(a + b + c + d)}$$

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 (pour variables asymétriques)

$$d(o_i, o_j) = 3/4 \quad d(i, j) = \frac{(b + c)}{(a + b + c)}$$

Exemple : Variables binaires (I)

- 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

Exemple : Variables binaires(II)

- Exemple

Nom	Sexe	Fièvre	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

$$d(jack, mary) = 0 + 1/2 + 0 + 1 = 0,33$$

$$d(jack, jim) = 1 + 1/1 + 1 + 1 = 0,67$$

$$d(jim, mary) = 1 + 2/1 + 1 + 2 = 0,75$$

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

Exemple : Variables Nominales

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

$$d(i, j) = \frac{(p - m)}{p}$$

- 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)

Exemple : Variables Ordinales

- Une variable ordinale peut être discrète ou continue
- L'ordre peut être important, ex: classement
- Par contre, l'écart entre deux valeurs n'est pas équivalent :
exemple, le degré de douleur de 1 à 10 : l'écart entre 5 et 7 n'a pas la même signification qu'entre 7 et 9.
- Peut-être traitée comme “interval-scaled” (f est une variable)
 - remplace x_{if} par leurs rangs $r_{if} \in \{1, \dots, M_f\}$
 - mappe la dynamique de chaque variable sur $[0, 1]$ en remplaçant *le* i -ème objet dans la f -ième variable by

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

Exemple : Google ou Similarité entre documents

Créer des moteurs de recherche comprenant le langage naturel

-> évolution de *google* au-delà des mots clés

-> le « text-mining » ou le « web-mining »

Collection : ensemble des documents

Vector Space Model (VSM) : espace à N dimensions où N est le nombre de termes utiles dans le langage (« le », « la »... sont inutiles et sont appelés **stop-words**)

Etape 1 : Pour chacun des N termes du langage, création d'un **Inverse Index** qui stocke pour chaque terme du langage les documents l'utilisant

- calcul du Document Frequency = nb doc utilisant ce terme / nb total de document = **df** (Plus df grand, moins le terme a d'importance d'un point de vue informatif)
- soit **idf = log (1/df)**

Etape 2 : Pour chaque document,

- Pour chaque terme, calcul du Term Frequency (Plus **tf** est grand dans un document plus ce terme doit être important par rapport au sujet du document)
- Calcul d'un vecteur caractéristique VSM :
 - Pour i allant de 0 à N, $VSM[i] = tf(\text{terme } i \text{ dans ce document}) * idf(\text{terme } i)$
 - On normalise ce vecteur pour que $\|VSM(\text{document})\|=1$

Chaque requête de l'utilisateur est associée de la même façon à un VSM(requête) et la Similarité est calculée par produit scalaire (voir clustering pour indexation et optimisation, voir ontologies pour web sémantique)

« Clustering »

Si on veut être plus statisticien :

Mesures de proximité entre 2 vecteurs-distributions : le cadre fréquentiste

Dans le cadre du texte mining, la **distance du chi2** est la distance euclidienne entre deux vecteurs-documents normalisés par leur longueur (nombre de mots), pondérée par la masse de chacun des mots par rapport à l'ensemble des textes (nombre total d'un mot dans l'ensemble des textes)

Et pour calculer la distance du χ_2 entre V1 et V3 :

$$d_{euclid}^2(V1, V3) = \sum_{j=1}^L \left(\frac{f_{1j}}{f_{1.}} - \frac{f_{3j}}{f_{3.}} \right)^2$$

$$\chi^2(V1, V3) = \sum_{j=1}^L \frac{f_{.j}}{f_{.j}} \left(\frac{f_{1j}}{f_{1.}} - \frac{f_{3j}}{f_{3.}} \right)^2$$

Où :

f désigne le nombre total de mots dans tous les documents

$f_{D_i j}$ désigne la fréquence du mot j dans le texte D_i

$f_{D_i .}$ désigne le nombre total de mots du texte D_i

$f_{.j}$ désigne la fréquence du mot j dans l'ensemble des textes.

En fait, la distance du chi2 permet de comparer deux histogrammes de valeurs ou encore deux distributions de probabilité

Très utile dès qu'on veut comparer des histogrammes, des distributions de probabilités....

Pour les documents (un feature = un mot)...

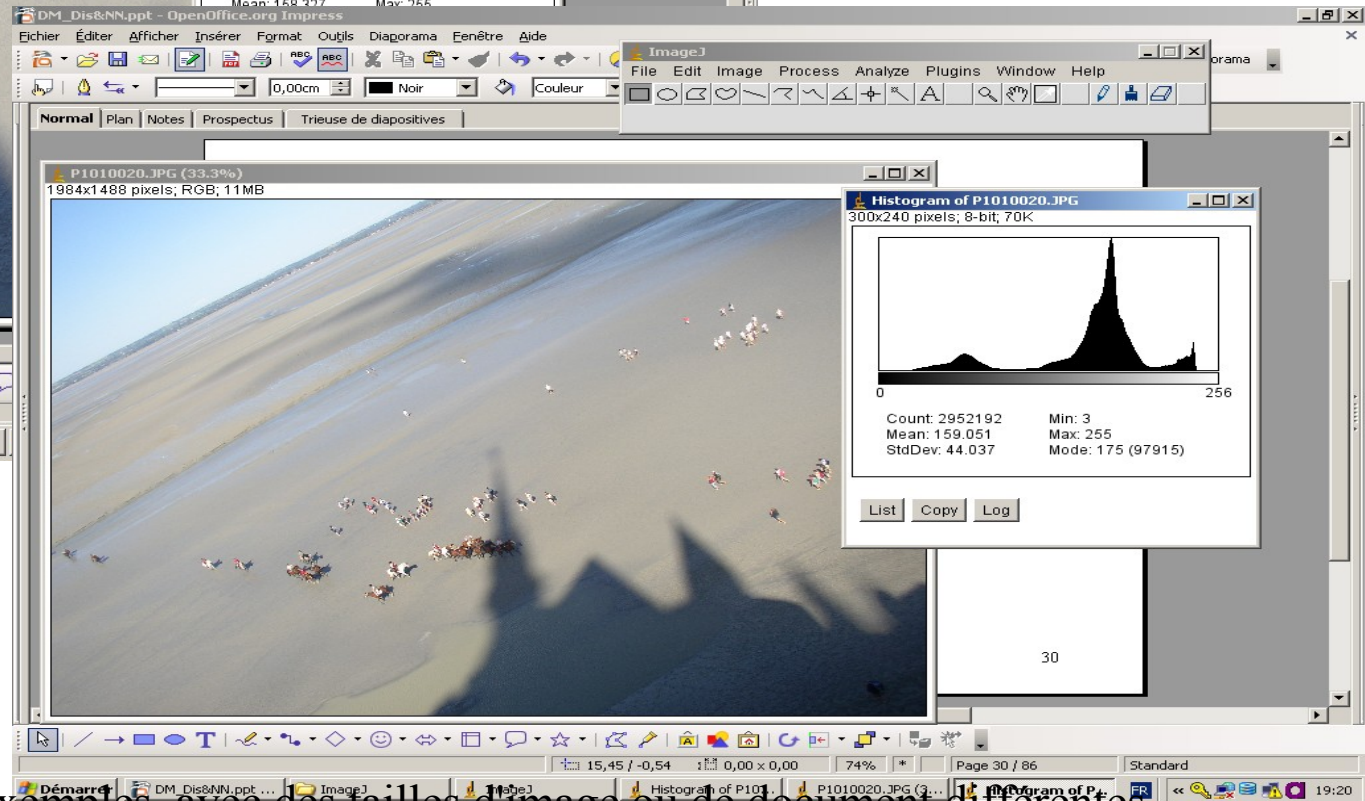
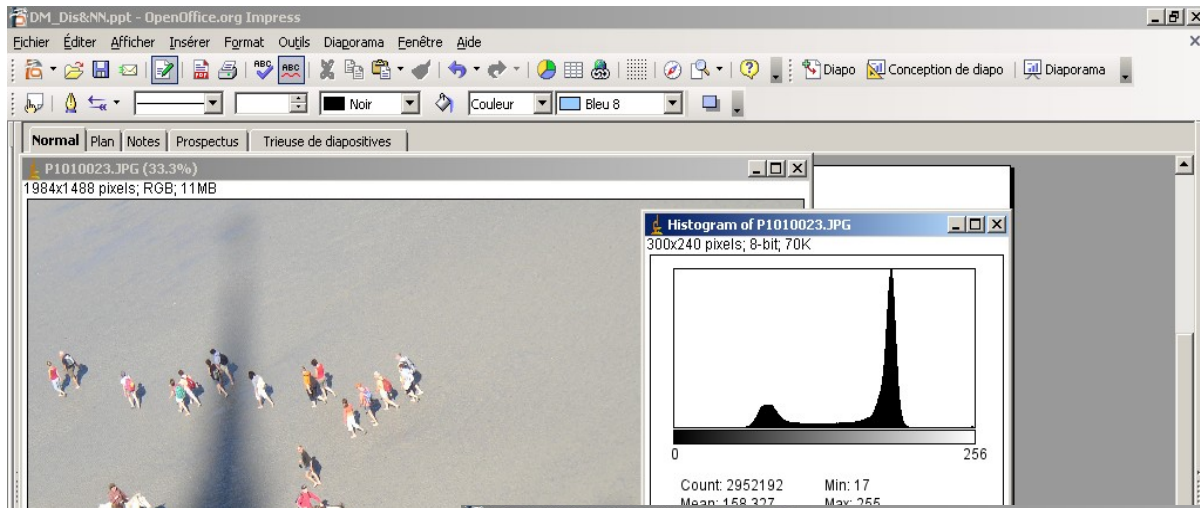
Table 2: Tableau de contingence

		Mots					
		m_1	...	m_j	...	m_L	somme
	D_1	$f_{1,1}$...	f_{1j}	...	f_{1L}	...

Textes	D_i	f_{i1}	...	f_{ij}	...	f_{iL}	$f_{i.} = \sum_{j=1}^L f_{ij}$

	D_D	f_{D1}	...	f_{Dj}	...	f_{DL}	...
	somme	$f_{.j} = \sum_{i=1}^D f_{ij}$	f

...ou pour les images (un feature = un niveau de gris)...



A vous de former des exemples, avec des tailles d'image ou de document différentes.

« Clustering »

Mesures de proximité entre 2 vecteurs-distributions : le cadre probabiliste

La distance de **Kullback-Leibler** entre les documents D1 et D2 :

$$d_{KL}(V1N, V2N) = \frac{1}{f} \sum_{j=1}^L (f_{ij} (\log \frac{f_{ij}}{f_i} - \log \frac{f_{Kj}}{f_K}))$$

Où :

f_{ij} = fréquence du mot j dans le document i

f_i = longueur du document i

$\frac{1}{f}$ = la longueur de l'ensemble des documents (de la classe ou pas encore de la classe)

$$d_{KL}(V1N, V2N) = \frac{1}{f} \sum_{j=1}^L (f_{ij} (\log(\frac{p(j|i)}{p(j|i')})))$$

Pas une vraie distance. Pas symétrique. Appelée divergence.

« Clustering »

Valeurs floues

$$x = (x_i)_{i \in [1, N]}, \quad x_i \in [0, 1]$$

$$s(x_i, y_i) = \max(\min(1 - x_i, 1 - y_i), \min(x_i, y_i))$$

$$s_F^p(x, y) = \left(\sum_{i=1}^N s(x_i, y_i)^p \right)^{1/p}$$

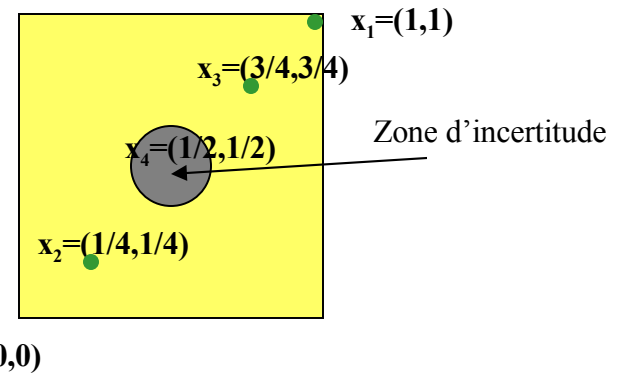
Propriétés :

$$s_{F_{\max}}^p = N^{1/p} \quad \text{et} \quad s_{F_{\min}}^p = 0,5N^{1/p}$$

Si $p \rightarrow \infty$:

$$s_F^\infty(x, y) = \max_{1 \leq i \leq N} s(x_i, y_i)$$

Remarque : $d_F(x, y) = s_F(x, y)$



- $S_F(x_1, x_1) =$
- $S_F(x_2, x_2) =$
- $S_F(x_3, x_3) =$
- $S_F(x_4, x_4) =$
- $S_F(x_1, x_3) =$
- $S_F(x_2, x_4) =$

A comparer aux distances euclidiennes !!

Enfin, dans le cas de mélanges de variables de différents types on calcule une distance entre éléments à partir d'une moyenne pondérée des distances définies pour chaque *feature* (ou dimension d'analyse ou variable) : d'un point de vue mathématique, on parle de combinaison linéaire, d'où en partie l'aspect matriciel des algorithmes

Proposer une mesure de dissimilarité pour les données étudiantes suivantes :

Nom de la variable	Note Examen	Sexe	TOEIC	Personal Computer	Age	Couleur des yeux	Motivation	Origine
Domaine de la variable	[0,20]	M/F	O/N	O/N	[18,40]	{vert,bleu, marron, noir}	[0,10]	{IUT, Licence Générale, Licence Pro, Master, autre}
Type de variable								

$$d(e_1, e_2) =$$

« Clustering »

Mesures de proximité entre 1 point x et 1 ensemble C de points

Sans prototypes

$$\wp^{pe}_{\max}(x, C) = \max_{y \in C} \wp(x, y)$$

$$\wp^{pe}_{\min}(x, C) = \min_{y \in C} \wp(x, y)$$

$$\wp^{pe}_{\text{moy}}(x, C) = \frac{1}{N} \sum_{y \in C} \wp(x, y)$$

Avec Prototypes

On utilise un représentant de la classe.

Dans le cas d'un cluster sphérique C ,

on utilise le point barycentre $m_C = \frac{1}{n_C} \sum_{y \in C} y$

et $d(x, C) = d(x, m_C)$

Dans le cas d'un cluster linéaire C ,

on utilise l'hyperplan (droite) H

d'approximation optimale

et $d(x, C) = d(x, H) = \frac{|a^T x + a_0|}{\|a\|}$

avec $H : \sum_{j=1}^p a_j x_j + a_0 = a^T x + a_0 = 0$

« Clustering »

Mesures de proximité entre 2 ensembles de points

$$\mathcal{D}^{ee}_{\max}(B, C) = \max_{x \in B, y \in C} \mathcal{D}(x, y)$$

...

$$\mathcal{D}^{ee}(B, C) = \sqrt{\frac{n_B n_C}{n_B + n_C}} \mathcal{D}(m_B, m_C)$$

D1={x1,x2,x3,x4} et D2={y1,y2,y3,y4}
X1(0,0), x2(0,2), x3(2,0), x4(2,2)
Y1(-3,0), y2(-5,0), y3(-3,-2), y4(-5,-2)
Avec la distance euclidienne :
$\mathcal{D}^{ee}_{\max}(B, C) = 8$
$\mathcal{D}^{ee}_{\min}(B, C) = 3$
$\mathcal{D}^{ee}_{\text{moy}}(B, C) = 5,6$
$\mathcal{D}^{ee}_p(B, C) = 7,6$

Conclusion (la part d'arbitraire ou d'expertise) : “La seule façon d'arriver un clustering des données adéquat (choix des distances, des prototypes), c'est par essai-erreur et bien sûr, en prenant en compte l'opinion d'un expert dans le champ d'application”

« Clustering »

Nombre de « clustering » possibles S

$$S(15,3) = 2\ 375\ 101$$

$$S(20,4) = 45\ 232\ 115\ 901$$




$$S(25,8) = 690\ 223\ 721\ 118\ 368\ 580$$

$$S(100,5) \approx 10^{68}$$

Si 10^{-12} secondes par cluster formé, il faudrait 10^{48} an/machine. Cela justifie le développement d'algorithmes « informés »

« Clustering »

3 Catégories d'algorithmes :

-  Séquentiels
-  Hiérarchiques
-  Basés sur l'optimisation d'une fonction de coût

Définition

Les méthodes de regroupement hiérarchique proposent une famille de regroupements dont la taille et le nombre des classes varie (inversement l'une de l'autre). Ces regroupements sont énumérés par un paramètre t , le niveau.

Ils forment une hiérarchie dans le sens que si deux échantillons sont regroupés dans une même classe au niveau t , ils le seront à tous les niveaux supérieurs à t .

La représentation graphique idéale pour une telle hiérarchie est le dendrogramme, car on ne produit pas un seul clustering mais une hiérarchie de clustering imbriqués (nested clustering).



Algorithme Hiérarchique Basique (AHB) :

INPUT : $S = \{x_1, x_2, \dots, x_N\}$, Choose $R_0 = \{ C_i = \{x_i\}, i = 1, \dots, N \}$ as the initial clustering, $t = 0$

• *Repeat* :

▪ $t = t + 1$

▪ Among all possible pairs of clusters (C_r, C_s) in R_{t-1} find the one, say (C_i, C_j) , such that $Dis(C_i, C_j) = \min_{r,s} Dis(C_r, C_s)$

▪ Define $C_q = C_i \cup C_j$ and produce the new clustering

$R_t = (R_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$

• *Until all vectors lie in a single cluster*

OUTPUT : Un ensemble de partitions $R = \cup R_i$, avec $R_i = \cup C_j^i$

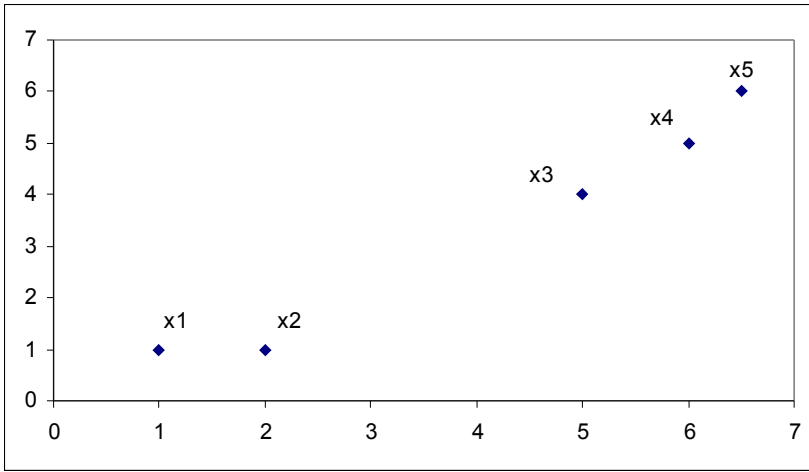
Data Structures

- Data matrix
– (two modes)

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
– (one mode)

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$



Matrice des Motifs

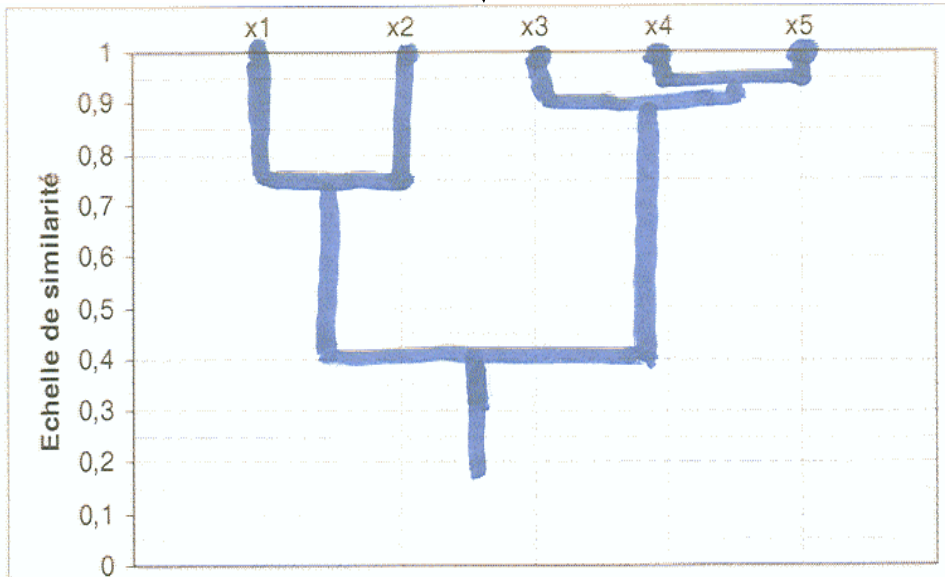
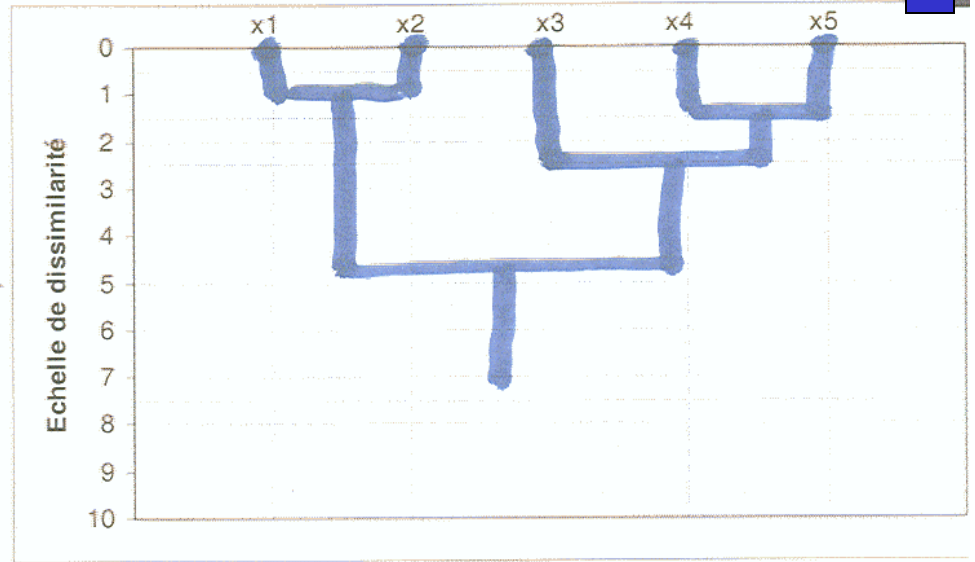
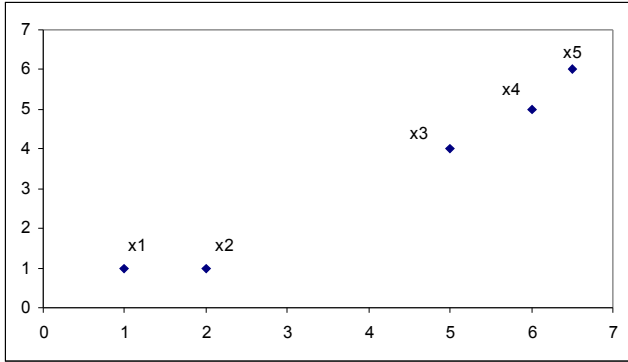
$$D(X) = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 5 & 4 \\ 6 & 5 \\ 6,5 & 6 \end{bmatrix}$$

Matrice des Dissimilarités

$$P_{euclid}(X) = \begin{bmatrix} 0 & 1 & 5 & 6,4 & 7,4 \\ & 0 & 4.2 & 5.7 & 6.7 \\ & & 0 & 1.4 & 2.5 \\ & & & 0 & 1.1 \\ & & & & 0 \end{bmatrix}$$

Matrice des Similarités

$$P_{tanimoto}(X) = \begin{bmatrix} 1 & 0.75 & 0.26 & 0.21 & 0.18 \\ & 1 & 0.44 & 0.35 & 0.2 \\ & & 1 & 0.96 & 0.9 \\ & & & 1 & 0.98 \\ & & & & 1 \end{bmatrix}$$



Les Dendrogrammes

Notion de durée de vie d'un cluster



Algorithme Hiérarchique Matriciel (AHM) :

• **INPUT** : $S = \{x_1, x_2, \dots, x_N\}$, Choose $R_0 = \{C_i = \{x_i\}, i = 1, \dots, N\}$ as the initial clustering, $t = 0$, $P_0 = P(X)$

• *Repeat* :

▪ $t = t + 1$

▪ Find (C_i, C_j) in R_{t-1} such that $Dis(C_i, C_j) = \min_{r,s} Dis(C_r, C_s)$

▪ Define $C_q = C_i \cup C_j$ and produce $R_t = (R_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$

▪ Define the new proximity matrix P_t from P_{t-1} (voir transparent suivant)

• *Until* R_{N-1} clustering is formed

OUTPUT : Un ensemble de partitions $R = \cup R_i$, avec $R_i = \cup C_j^i$



« Clustering »

Algorithme Hiérarchique Matriciel (AHM) :

- *Mise à jour de la matrice P_t avec :*

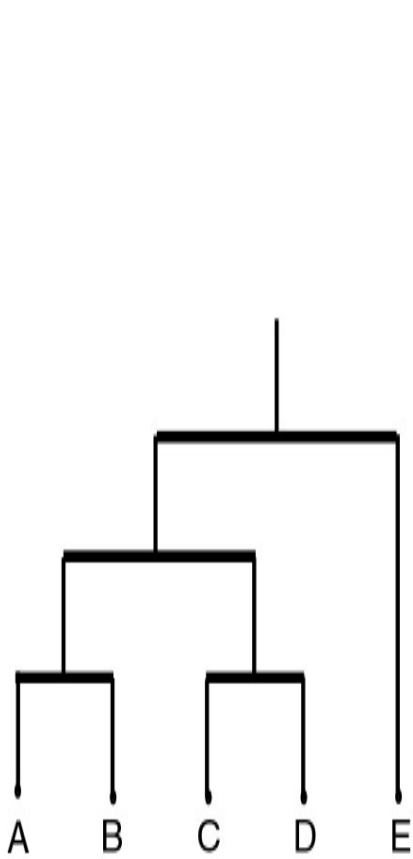
$$Dis(C_q, C_s) = a_i Dis(C_i, C_s) + a_j Dis(C_j, C_s) + b Dis(C_i, C_j) + c |Dis(C_i, C_s) - Dis(C_j, C_s)|$$

- *Single Link Algorithm ou voisin immédiat : $a_i = a_j = \frac{1}{2}$ et $b=0$ et $c = -\frac{1}{2}$*

$$Dis(C_q, C_s) = \text{Min} \{ Dis(C_i, C_s), Dis(C_j, C_s) \}$$

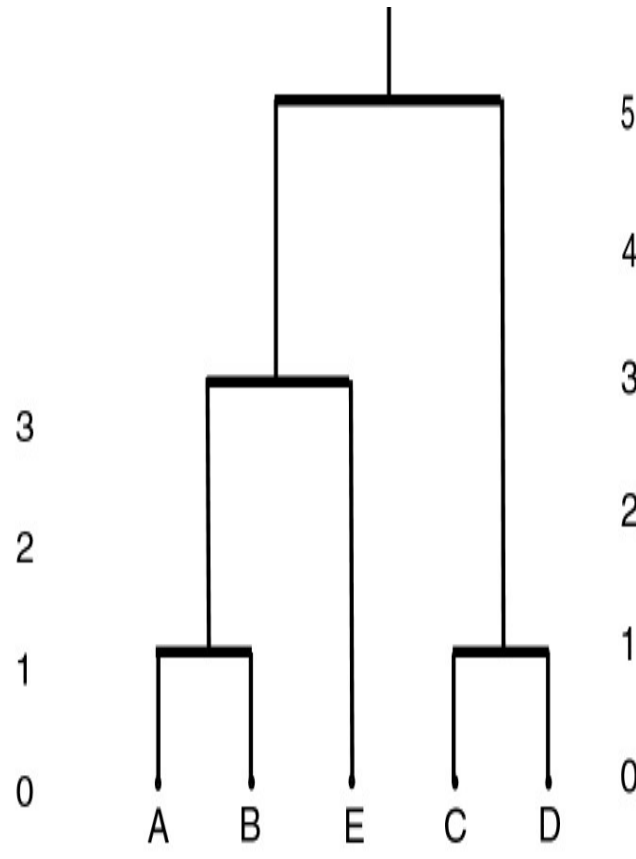
- *Complete Link Algorithm ou voisin éloigné : $a_i = a_j = \frac{1}{2}$ et $b=0$ et $c = \frac{1}{2}$*

$$Dis(C_q, C_s) = \text{Max} \{ Dis(C_i, C_s), Dis(C_j, C_s) \}$$



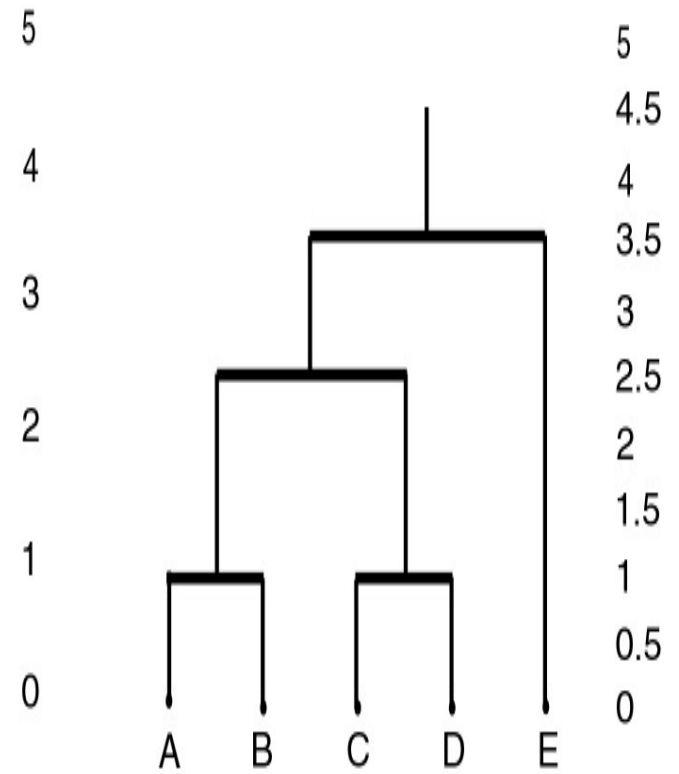
Single Link

$$a_i = a_j = \frac{1}{2} \text{ et } b=0 \text{ et } c = -\frac{1}{2}$$



Complete Link

$$a_i = a_j = \frac{1}{2} \text{ et } b=0 \text{ et } c = \frac{1}{2}$$

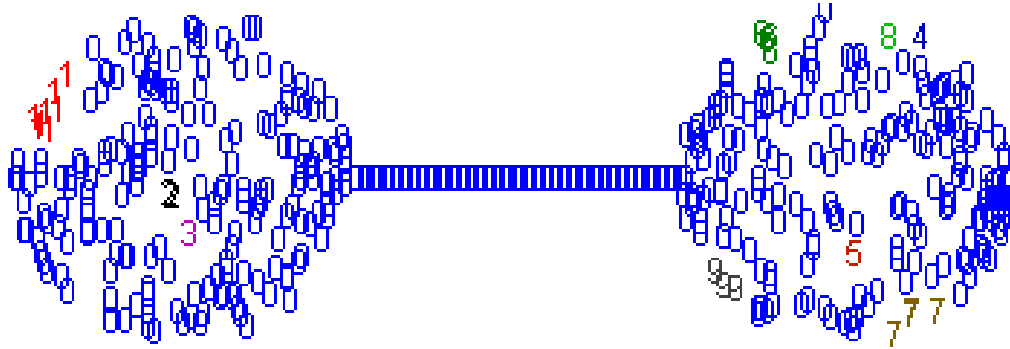


Average Link

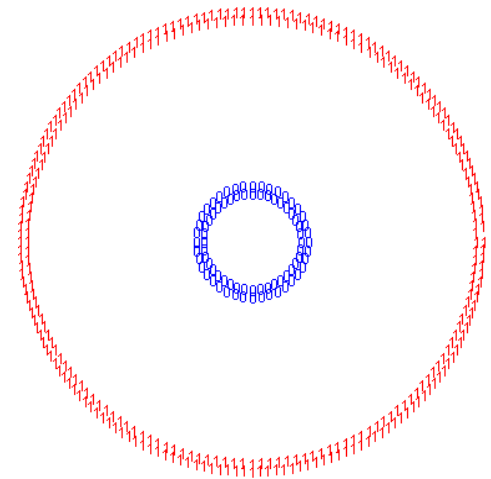
$$a_i = a_j = \frac{1}{2} \text{ et } b=c=0$$



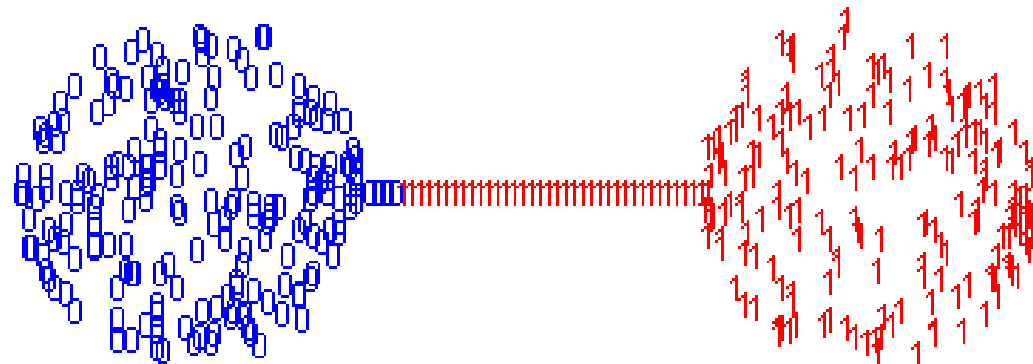
Effet de Chaîne



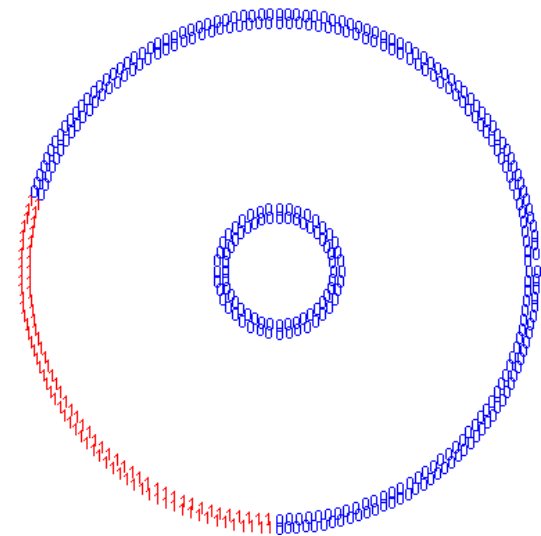
Single-link (10 clusters)



Single-link (2 clusters)



Complete-link (2 clusters)



Complete-link (2 clusters)

2 variantes remarquables :

- *Algorithme de Ward qui minimise la variance : on construit une distance pondérée d'_{ij} :*

$$d'_{ij} = \frac{n_i n_j}{n_i + n_j} d_{ij} \text{ avec } d_{ij} = \|m_i - m_j\|^2$$

$$d'_{qs} = \frac{n_i + n_s}{n_i + n_j + n_s} d'_{is} + \frac{n_j + n_s}{n_i + n_j + n_s} d'_{js} - \frac{n_s}{n_i + n_j + n_s} d'_{ij}$$

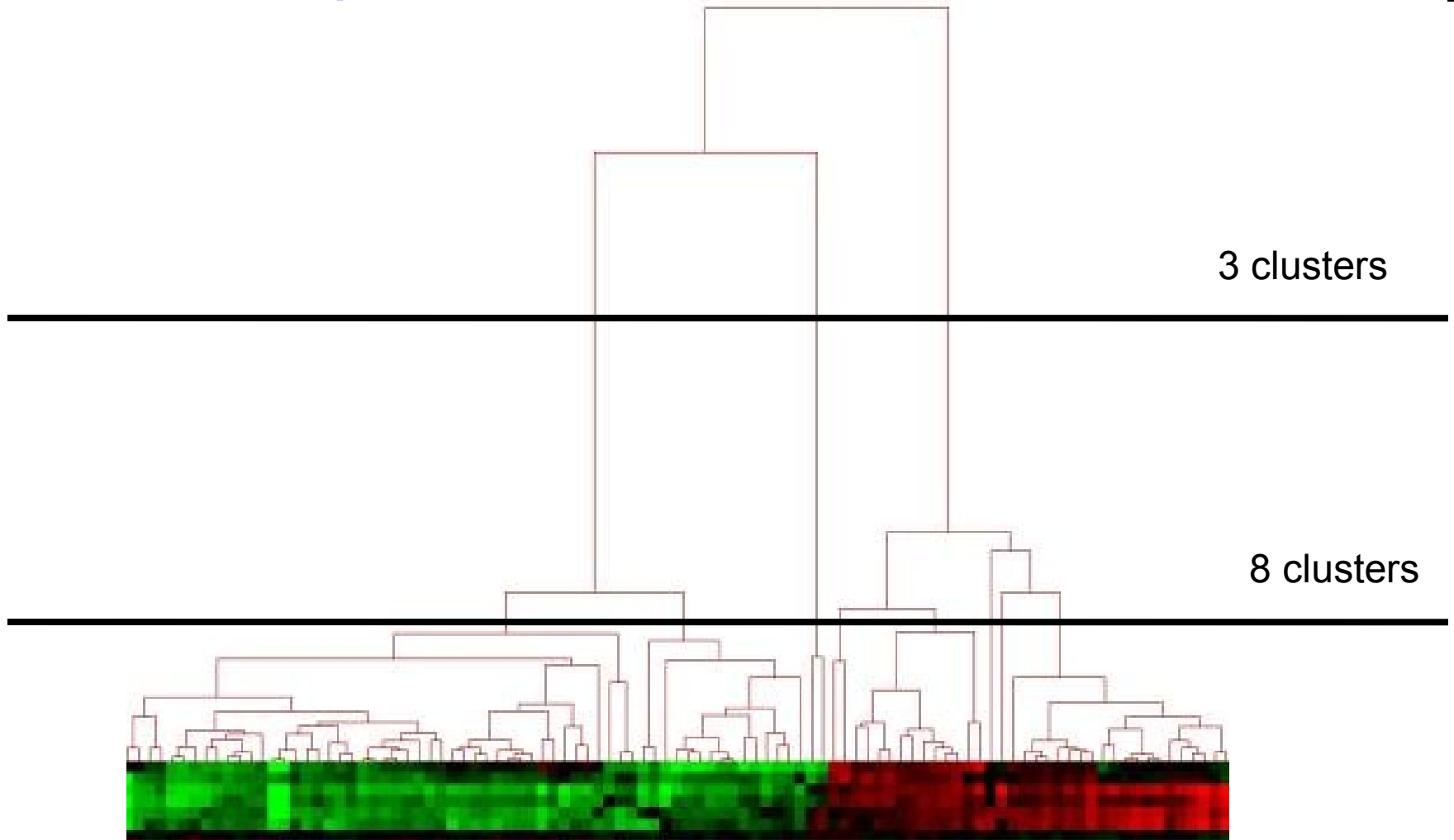
Si $e_r^2 = \sum_{x \in C_r} \|x - m_r\|^2$ est la variance du $r^{\text{ième}}$ cluster

et $E_t = \sum_{r=1}^{N-t} e_r^2$, cet algorithme forme la R_{t+1} partition en fusionnant les 2 clusters

$C_i + C_j$ qui conduisent à l'augmentation minimum de la variance totale E_t

- Si $a_i = n_i / (n_i + n_j)$ $a_j = n_j / (n_i + n_j)$ et $b = -n_i n_j / (n_i + n_j)^2$ et $c = 0$

$$d_{qs} = a_i d_{is} + a_j d_{js} + b d_{ij} = \|m_q - m_s\|^2$$



A dendrogram for a part of Yeast cDNA microarray data set.

Heuristique pour le meilleur nombre de clusters : on s'arrête par exemple à la hiérarchie R_t pour laquelle




$$\exists C_j \in R_{t-1} / h(C_j) > \theta \quad \text{avec } h(C) = \max\{d(x, y), x, y \in C\}$$

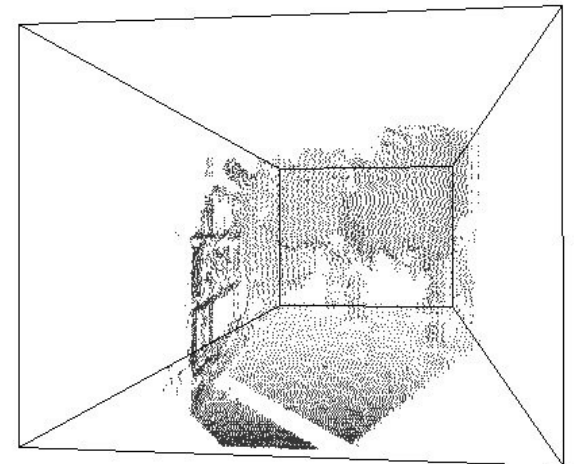
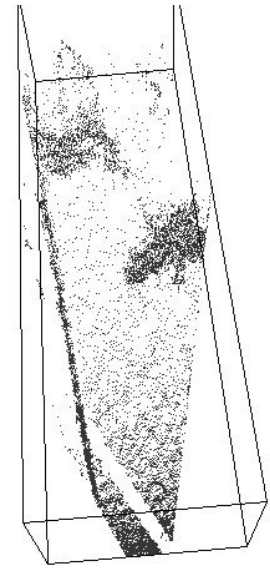
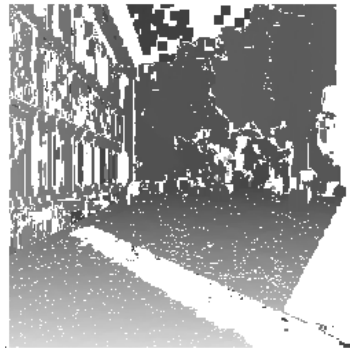
Notes sur l'algorithme hiérarchique :

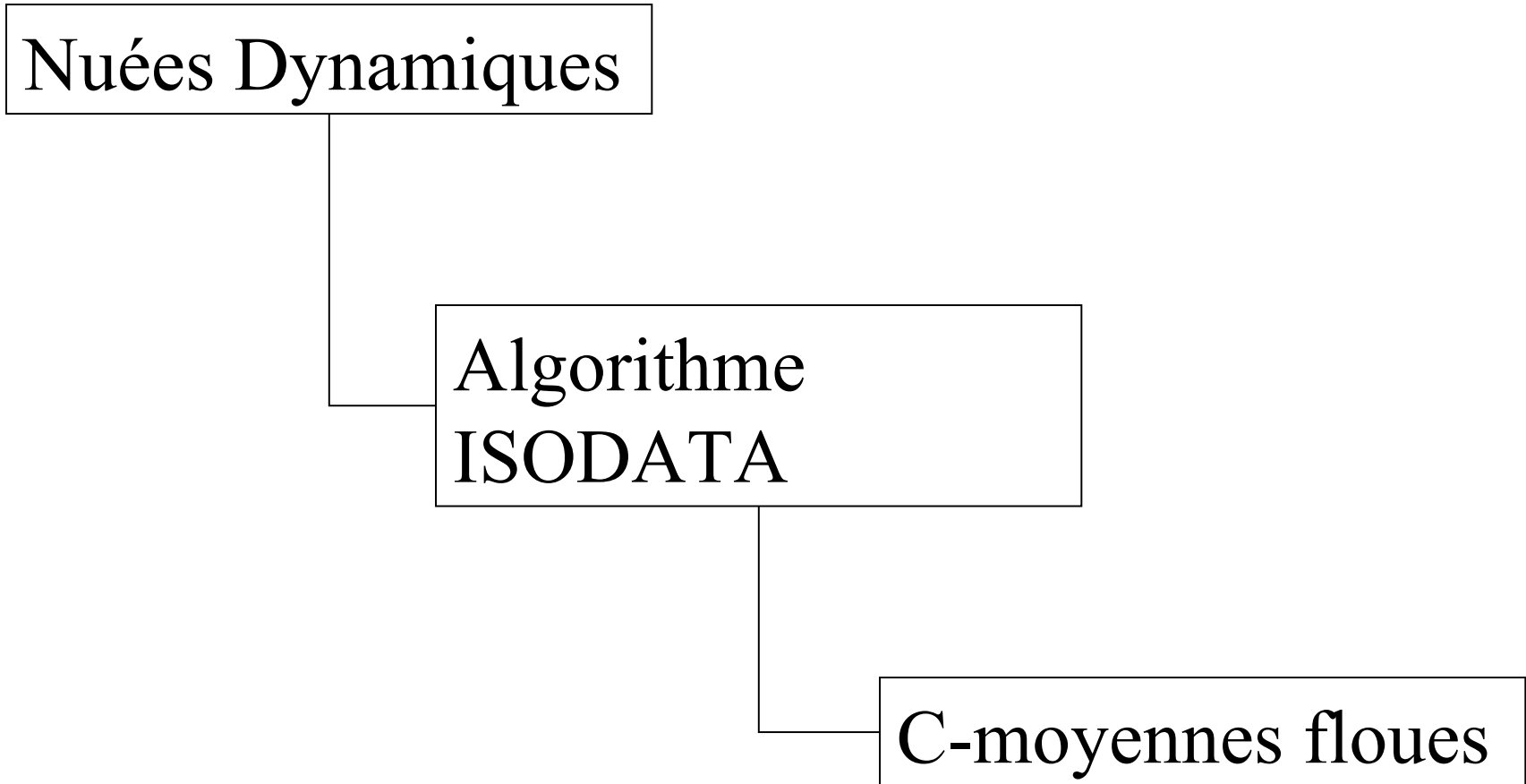
- Cet algorithme hiérarchique est bien adapté pour fournir une segmentation multi-échelle puisqu'il donne en ensemble de partitions possibles en C classes pour 1 variant de 1 à N , nombre de points dans le nuage de points
- Différents niveaux de granularité sont ainsi directement visualisable dans les données
- Peut prendre en entrée des données ou directement une matrice de mesures de proximité sans connaissance sur l'espace des données (voir exercice sur la phonétique).

« Clustering »

3 Catégories d'algorithmes :

-  Séquentiels
-  Hiérarchiques
-  Basés sur l'optimisation d'une fonction de coût







Principe les Nuées Dynamiques

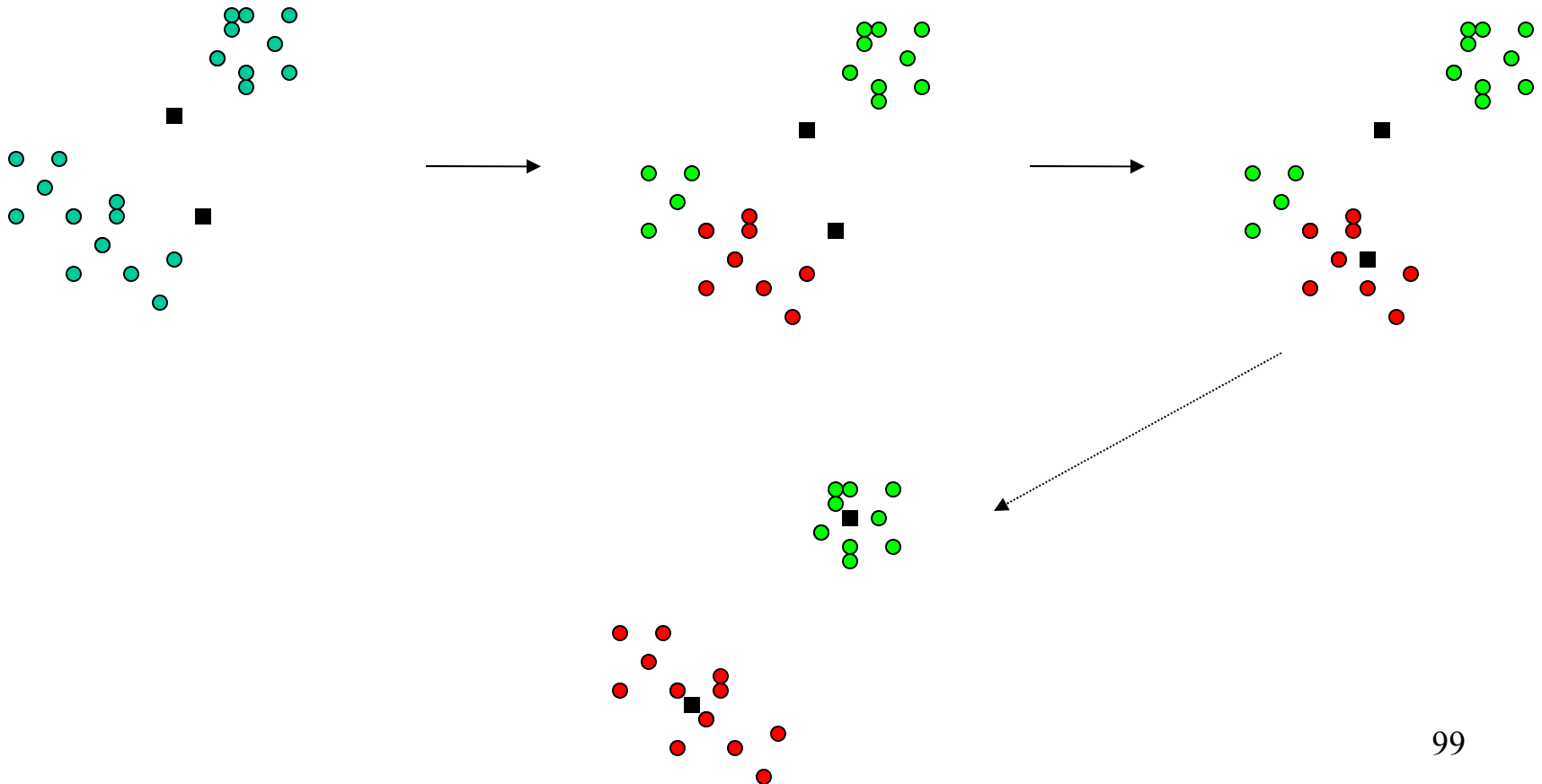


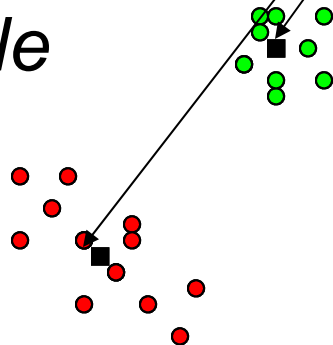


Schéma numérique : Algorithme itératif de type ISODATA

⇒ Nombre de groupements C connu

⇒ V est un vecteur de paramètres de forme : on peut prendre par exemple les barycentres m_c des nuages de points C .

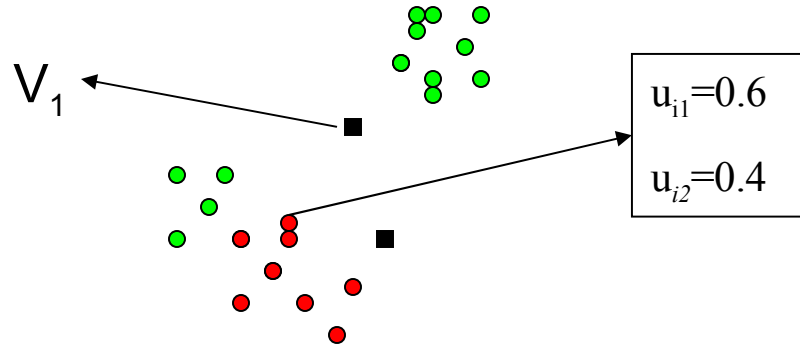
⇒ Minimisation itérative d'une fonctionnelle J qui mesure la valeur d'un regroupement selon un critère variable



Implémentation par les C-moyennes



⇒ Coefficient d'appartenance u_{ij} dans $[0, 1]$



⇒ Minimisation de la fonctionnelle générale :

$$J_m(U; V) = \sum_{j=1}^C \sum_{i=1}^N u_{ij}^m d^2(x_i, V_j)$$

⇒ Dans le cas dur, si $u_{ij} \in \{0, 1\}$:

$$J_m(U; V) = J(V)$$



Algorithme ISODATA Dur (Nuées Dynamiques ou K-Means ou C-moyennes) $u_{ij} \in \{0;1\}$

- **INPUT** : $S = \{x_1, x_2, \dots, x_N\}$, un nombre de classes C et un ensemble de noyaux initial V_i^0 , $i = 1, \dots, C$, itération $n=1$, nombre d'itérations maximum n_0
- **Pour chaque** valeur de k (de 1 à C), calculer
$$C_k^n \leftarrow \left\{ x_i \in S \mid \forall j \neq k, d(x_i, V_k^{n-1}) \leq d(x_i, V_j^{n-1}) \right\}$$

Calcul de V_k^n à partir de C_k^n
- **Si** $\forall k, C_k^n \neq C_k^{n-1}$ *et* $n \leq n_0$,
Sinon arrêt
- **OUTPUT** : *une classification dure* $R = \cup C_i$

« Clustering »

Cas « Fuzzy » : $u_{ij} \in [0, 1]$



⇒ Dans le cas où $m=2$, minimisation de la fonctionnelle suivante :

$$J(U; V) = \sum_{j=1}^C \sum_{i=1}^N u_{ij}^2 d^2(x_i, V_j)$$

⇒ En utilisant la formulation lagrangienne, on montre que minimiser J revient à résoudre ce système couplé :

$$\left\{ \begin{array}{l} \sum_{i=1}^N u_{ij}^2 \frac{\partial d^2(x_i, V_j)}{\partial V_j} = 0 \\ u_{rs} = \frac{1}{\sum_{j=1}^C \frac{d^2(x_i, V_s)}{d^2(x_i, V_j)}} \end{array} \right.$$

Algorithme ISODATA Flou (Nuées Dynamiques ou K-Means ou C-moyennes) $u_{ij} \in [0, 1]$

- **INPUT** : $S = \{x_1, x_2, \dots, x_N\}$, un nombre de classes C et un ensemble de noyaux initial V_i^0 , $i = 1, \dots, C$, itération $t=1$, nombre d'itérations maximum t_0

Répéter

- **Pour chaque** valeur de i (de 1 à N),

Pour chaque valeur de j (de 1 à C), calculer $u_{ij}(t) = \frac{1}{\sum_{k=1}^C d^2(x_i, V_k)}$

Fin pour

Fin pour

- $t = t + 1$

- **Pour chaque** valeur de j (de 1 à C),

Résoudre $\sum_{i=1}^N u_{ij}^2(t-1) \frac{\partial d^2(x_i, V_j)}{\partial V_j} = 0$

Fin pour

Tant que un critère d'arrêt n'est pas atteint

- **OUTPUT** : une classification floue $U = u[i][j]$ pour i de 1 à N et j de 1 à C

« Clustering »

Cas « Fuzzy » : $u_{ij} \in [0, 1]$



Dans le cas classique où on utilise la distance euclidienne

$$d^2(x_i, V_j) = (x_i - V_j)^T (x_i - V_j)$$

On met à jour les prototypes par l'équation suivante :

$$V_j(t) = \frac{\sum_{i=1}^N u_{ij}^2(t-1) x_i}{\sum_{i=1}^N u_{ij}^2(t-1)}$$

Il s'agit de l'algorithme du « Fuzzy K-Means » ou C-moyennes floues classique qui dans ce cas fait l'hypothèse d'une distribution Normale des données dans les classes

« Clustering »

Cas « Fuzzy » : $u_{ij} \in [0, 1]$



Ajoutons que pour $m=1$, il n'y a pas de clustering flou meilleur que le meilleur des « crisp clustering ».

Mais pour $m=2$ (le cas étudié), il y a des cas pour lesquels le clustering flou a de plus petites valeurs pour $J_2(U;V)$

« Clustering »



Algorithme des C-Moyennes Floues Exponentielles (CMFE)

⇒ Prototypes : centroïdes \mathbf{V}_j et matrices de covariance floue \mathbf{F}_j

$$\mathbf{F}_j = \frac{\sum_{i=1}^N u_{ij}^2 (\mathbf{x}_i - \mathbf{V}_j)(\mathbf{x}_i - \mathbf{V}_j)^T}{\sum_{i=1}^N u_{ij}^2}$$

⇒ Distance : *exponentielle* \mathbf{d}_e

$$d_e^2(\mathbf{x}_i, \mathbf{v}_j) = \frac{[\det(\mathbf{F}_j)]^{1/2}}{P_j} \exp \left[\frac{1}{2} (\mathbf{x}_i - \mathbf{v}_j)^T \mathbf{F}_j^{-1} (\mathbf{x}_i - \mathbf{v}_j) \right]$$

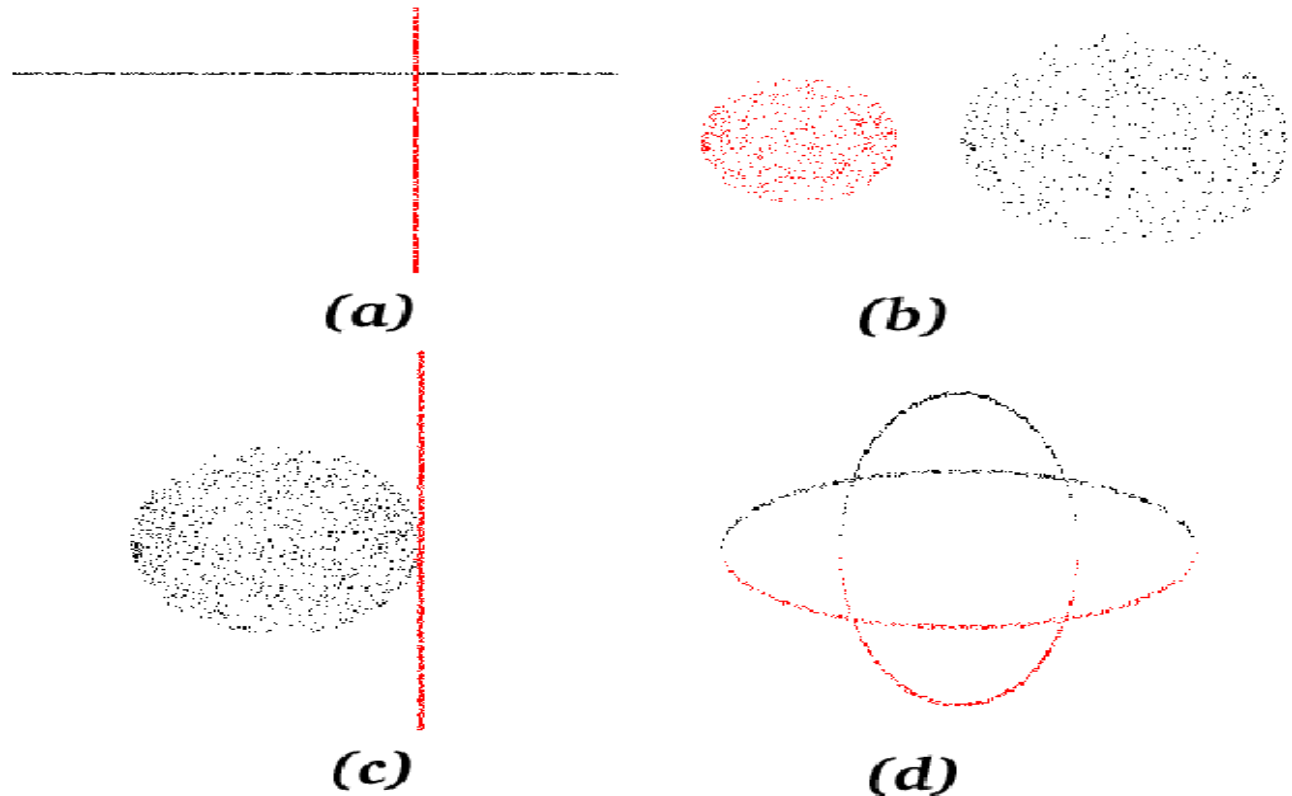
« Clustering »



Algorithme des C-Moyennes Floues Exponentielles (CMFE)

Prend en compte pour chaque cluster :

- La forme
- Le nombre de points
- La densité

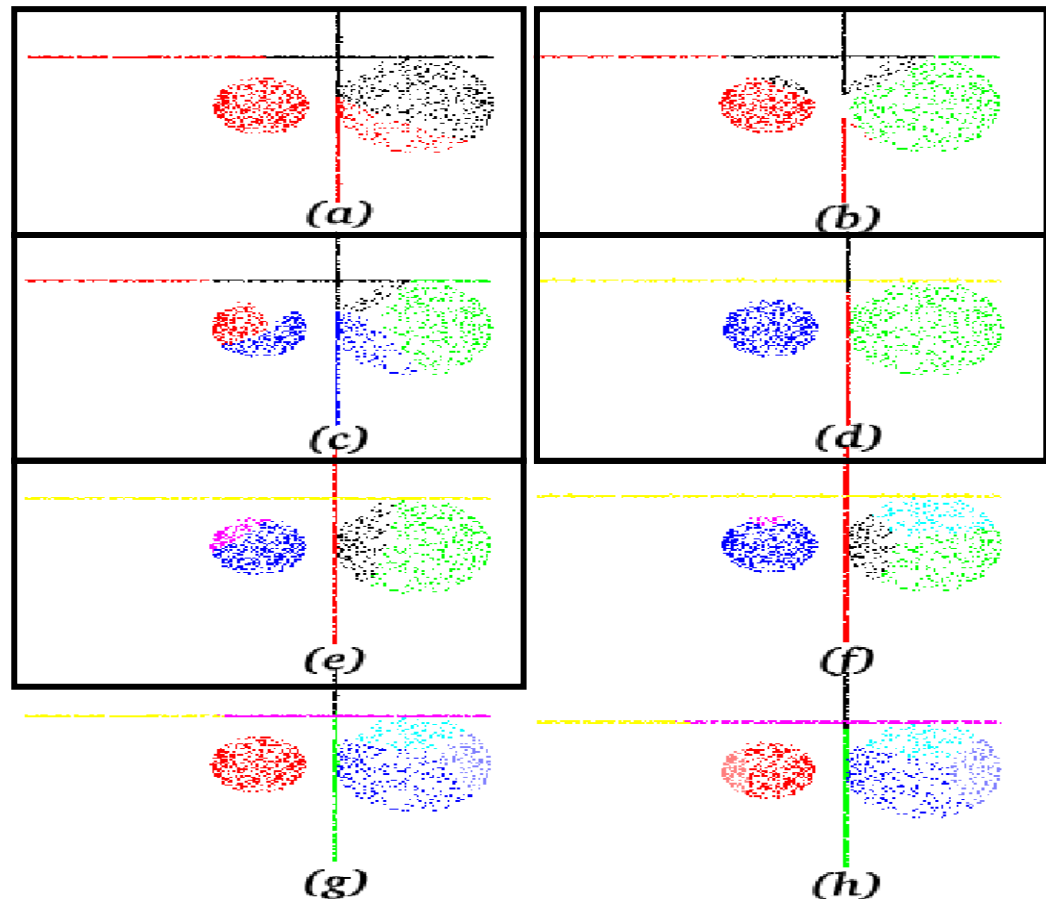


« Clustering »



Comment déterminer
le nombre de groupements C optimal ?

Critère numérique :
la Densité Moyenne
de Partition
(*DMP*) ?



« Clustering »



Critère numérique : la Densité Moyenne de Partition

$$DPM(C) = \frac{1}{C} \sum_{j=1}^C \frac{S_j}{V_j}$$

Avec $X_j = \{x \in X : (x - V_j) F_j^{-1} (x - V_j) < 1\}$

$$S_j = \sum_{x_i \in X_j} u_{ij}$$

et l'hypervolume flou de chaque cluster

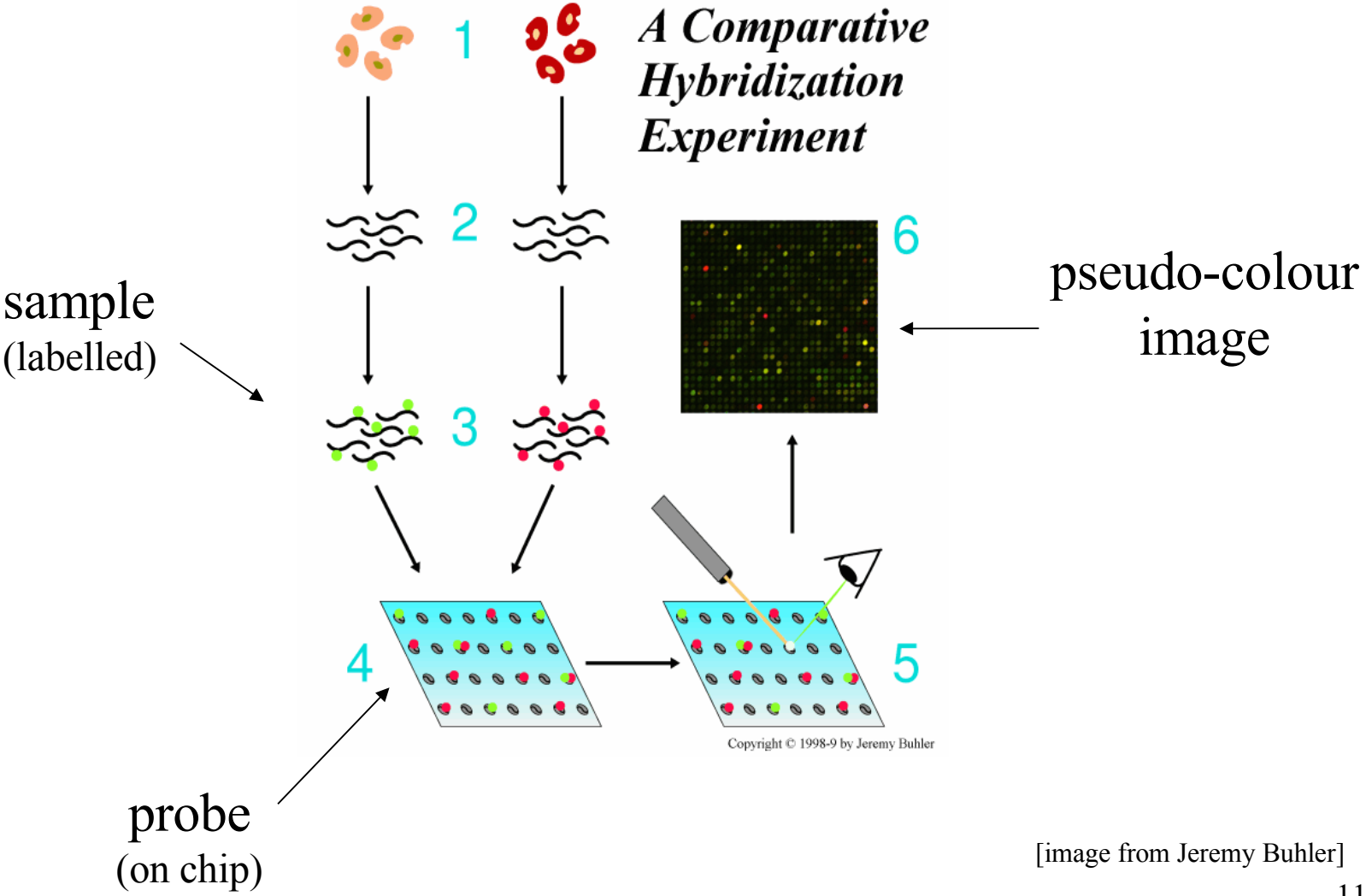
$$V_j = |F_j|^{1/2}$$

Notes sur l'algorithme ISODATA :

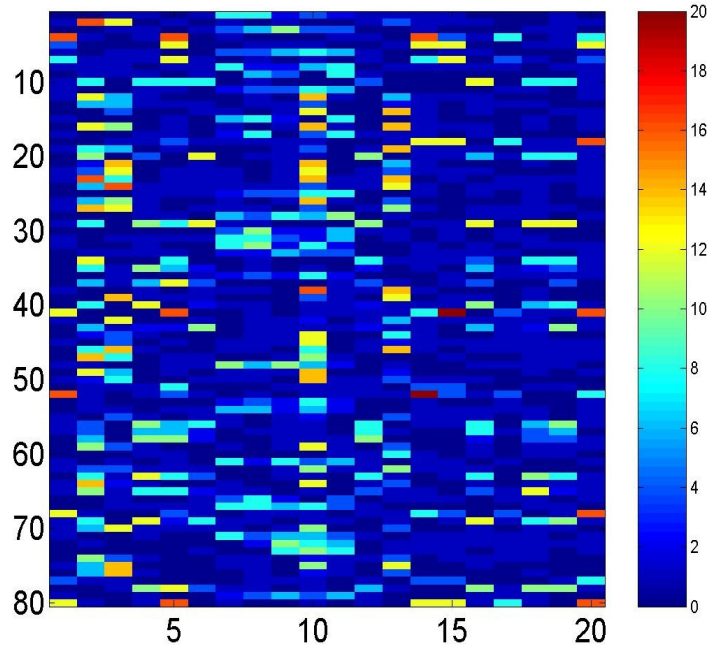
- Cet algorithme procédant par optimisation itérative est très simple et très connu.
- Il a des propriétés de convergence surprenantes **en pratique** (l'initialisation influe assez peu sur le résultat final par exemple) notamment pour traiter un grand nombre de données

Pour une analyse pratique des clusters , voir Section 2.1 de l'étude de cas : [*dm_case_studies.pdf*](#)

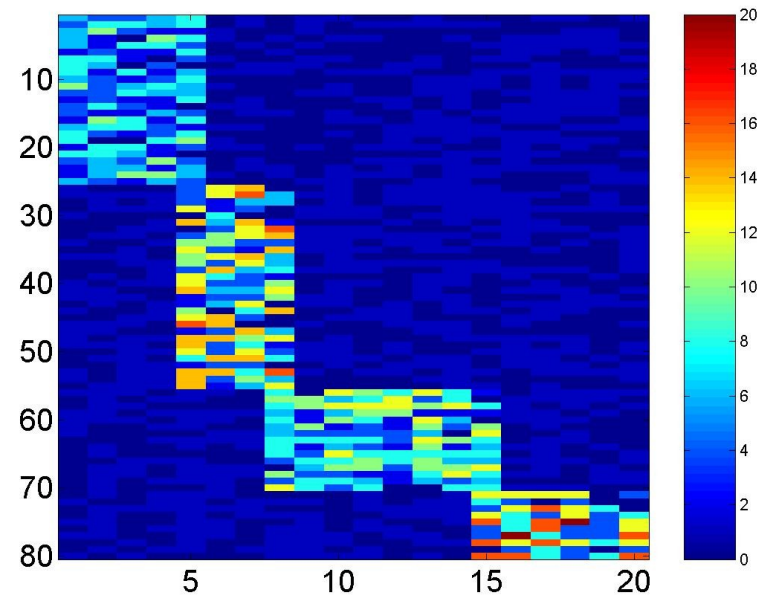
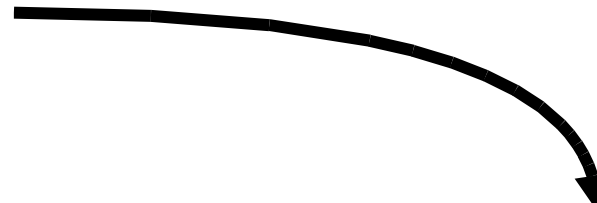
Exemple d'Application en biologie : l'analyse des résultats fournis par les puces ADN



[image from Jeremy Buhler]



Regroupement en familles de gènes



Chapitre I. Apprentissage *a priori* par Induction

Règles d'association

Complément récent aux méthodes statistiques classiques de détection de relations entre attributs

Input : données booléennes

Output : implications logiques entre attributs ou ensemble d'attributs en utilisant la table de vérité de l'implication logique :

x	y	$x \rightarrow y$
V	V	V
F	V	V
F	F	V
V	F	F

Un exemple e (non supervisé) = *enregistrement* d'une table.

e est décrit par d attributs binaires (x_1, x_2, \dots, x_d) ou *champs*.

La valeur à 1 d'un attribut pour un exemple est appelé un *item*.

Un ensemble d'items pour un enregistrement a été appelé un *itemset*.

Soit l'ensemble d'apprentissage ou la base de données suivantes contenant 10 exemples décrits par 5 attributs binaires :

	x1	x2	x3	x4	x5
e1	0	1	0	0	1
e2	0	0	1	0	1
e3	0	0	1	0	0
e4	1	1	1	1	1
e5	1	1	1	1	1
e6	1	1	1	1	0
e7	1	0	1	1	0
e8	1	0	1	1	0
e9	1	0	0	0	1
e10	1	0	0	0	1

Couverture (ou support) de $x_1 \rightarrow x_2$:

la probabilité $P(x_1, x_2)$ que x_1 et x_2 soient VRAI en même temps.

Propriété symétrique : $s_{x_1 \rightarrow x_2} = s_{x_2 \rightarrow x_1}$

Ici, $P(x_1, x_2) = 3/10$

Cas peu intéressants : $s_{x \rightarrow y} \sim 0$ ou 1

Ici, $\{x_2, x_3\}$ et $\{x_1, x_4\}$ sont des itemsets de e_5 .

Association :

Implication disant que deux itemsets sont VRAI ensemble pour un nombre suffisant d'exemples. La couverture de l'association est calculée comme nombre d'itemsets, divisé par le nombre total d'exemples. Quand une couverture est supérieure à une valeur MinCouv fixée à l'avance par l'utilisateur, on dit que l'itemset constitué par cette intersection est fréquent.

Ici, si $\text{MinCouv}=0.3$, les itemsets (x_1, x_2, x_3) et (x_1, x_3, x_4) sont fréquents avec une couverture d'association : $s_{x_1 x_3 x_5}=0.3$ et $s_{x_1 x_3 x_4}=0.5$

Du coup toutes les règles d'association associées ont une couverture fréquente : $s_{x_1 \rightarrow x_3} = \dots = s_{x_1 \text{ et } x_3 \rightarrow x_5} = 0.3$

On peut définir un algorithme rapide comme "A Priori" pour trouver les items fréquents en utilisant $s_{\text{itemset1} \cup \text{itemset2}} \leq s_{\text{itemset1}}$

Algorithme A priori

Créer L_1 , l'ensemble des 1-itemsets fréquents par une consultation de la base de données

Tant que le test d'arrêt n'est pas satisfait faire

Etape 1 : utiliser L_{k-1} pour produire C_k contenant les k-itemsets candidats.

NB. : Ceci se fait sans consulter la base de données

Etape 2 : Ne conserver que les itemsets de C_k qui sont fréquents : ils

constituent L_k

NB. : Ceci demande une consultation de la base de Données

Fin tant que

Chapitre I. Apprentissage *a priori* par Induction

Information Retrieval

Créer des moteurs de recherche comprenant le langage naturel

-> évolution de *google* au-delà des mots clés

-> le « text-mining » ou le « web-mining »

Collection : ensemble des documents

Vector Space Model (VSM) : espace à N dimensions où N est le nombre de termes utiles dans le langage (« le », « la »... sont inutiles et sont appelés **stop-words**)

Etape 1 : Pour chaque terme du langage,

- création d'un **Inverse Index** qui stocke pour chaque terme du langage les documents l'utilisant
- calcul du Document Frequency = nb doc utilisant ce terme / nb total de document = **df** (Plus df grand moins le terme a d'importance d'un point de vue informatif)
- soit **idf = log (1/df)**

Etape 2 : Pour chaque document,

- Pour chaque terme, calcul du Term Frequency (Plus **tf** est grand dans un document plus ce terme doit être important par rapport au sujet du document)
- Calcul d'un vecteur caractéristique VSM :
 - Pour i allant de 0 à N $VSM[i] = tf(\text{terme } i \text{ dans ce document}) * idf(\text{terme } i)$
 - On normalise ce vecteur pour que $\|VSM(\text{document})\|=1$

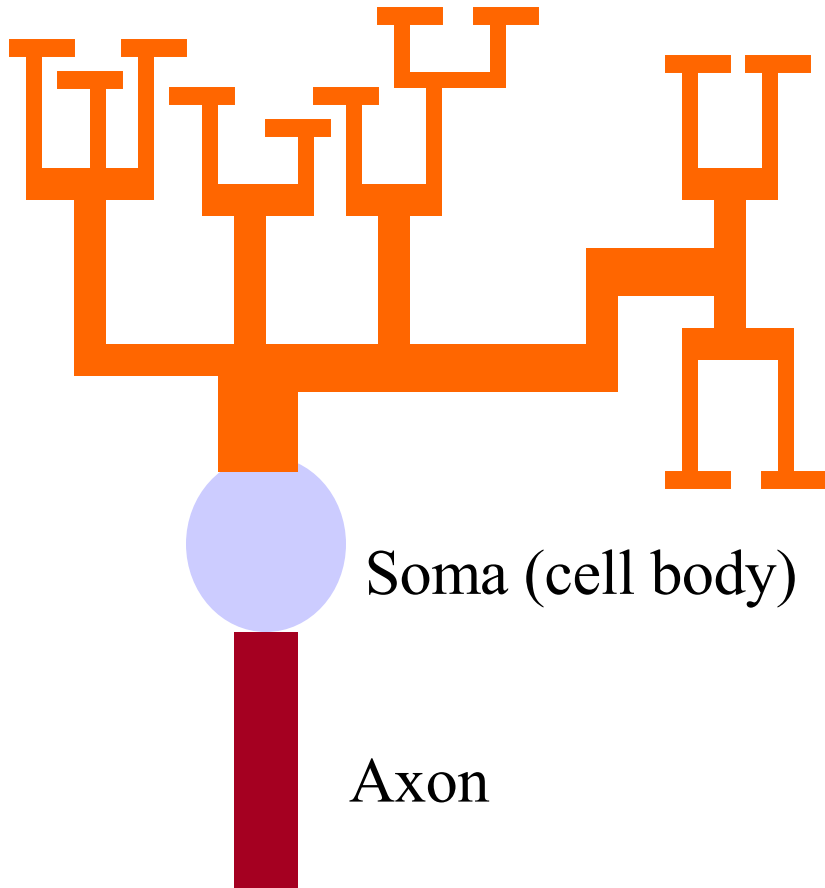
Chaque requête de l'utilisateur est associée de la même façon à un VSM(requête) et la Similarity est calculé par produit scalaire (voir clustering pour indexation et optimisation, voir ontologies pour web sémantique) 120

Chapitre II. Apprentissage *a posteriori* par Séparatrices Linéaires

Réseaux de Neurones

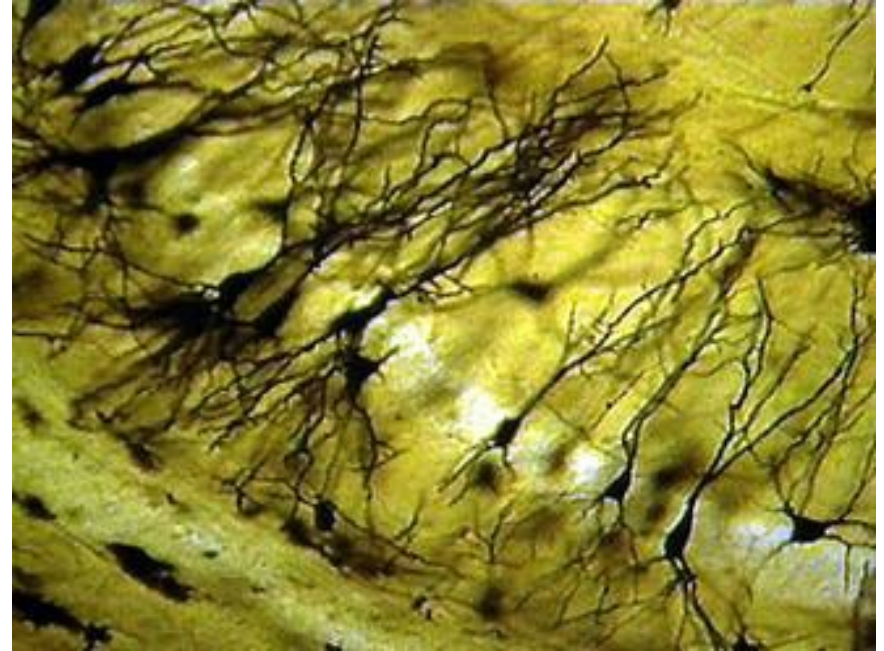
Inspiration Biologique

Dendrites

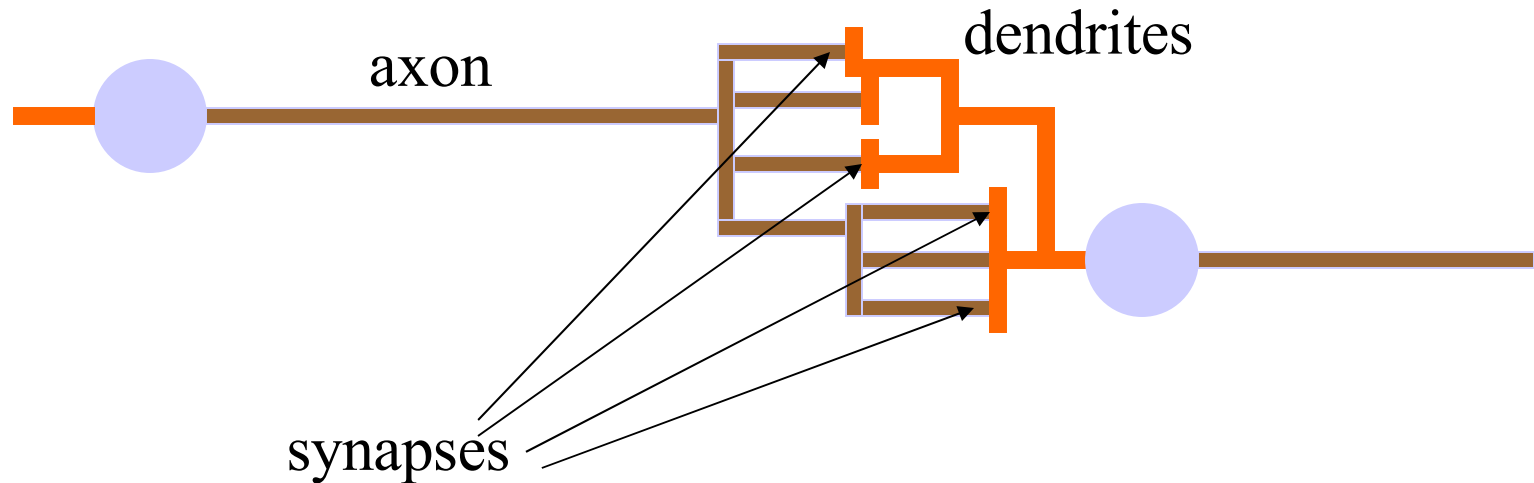


Soma (cell body)

Axon



Inspiration Biologique

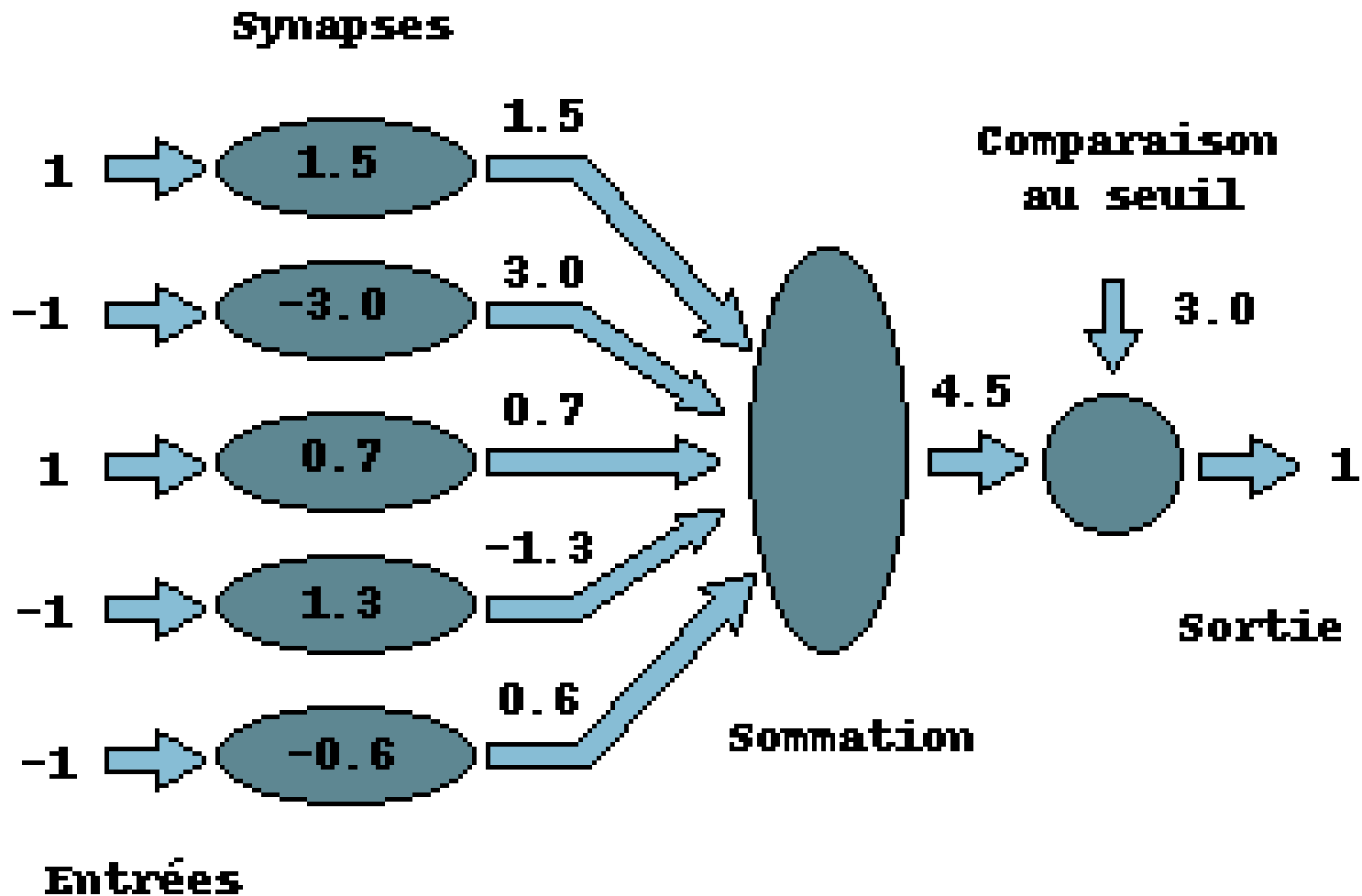


La transformation de l'information symbolique a lieu au niveau des synapses sous forme d'impulsions électriques quantifiées (numérisables)

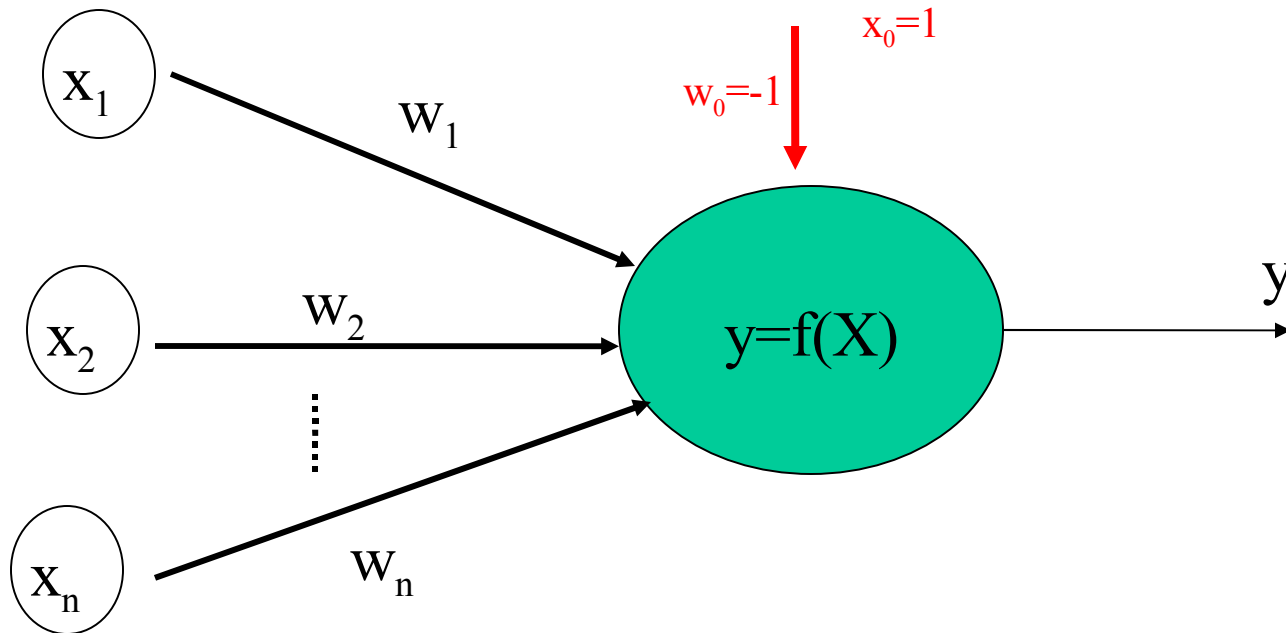
Inspiration Biologique

- Cerveau humain : 100 billions de neurones (10^{11}) et des centaines de types différents.
- Les neurones se rassemblent en **couches**, contenant chacune des milliers de neurones fortement interconnectés.

Une modélisation simplifiée de la réalité



Le modèle de neurone artificiel de McCulloch et Pitts



$$y = f\left(\sum_{i=1}^n w_i x_i - u\right) = f\left(\sum_{i=0}^n w_i x_i\right)$$

Histoire

- 1943 - McCullock et Pitts : le problème

$$[(x \text{ AND } y) \text{ OR } (x \text{ OR } y)]$$



- 1950's - Hodgkin et Huxley : prix Nobel.

- 1969 - Minsky et Papert, *Perceptrons*, le problème non résolu : le XOR :

$$[(x \text{ AND NOT } y) \text{ OR } (y \text{ AND NOT } x)]$$

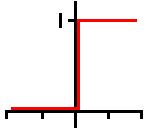
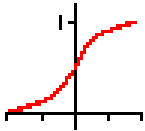
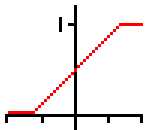
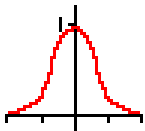
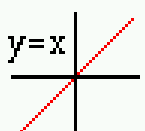


- 1987 - Robert Hecht-Nielsen



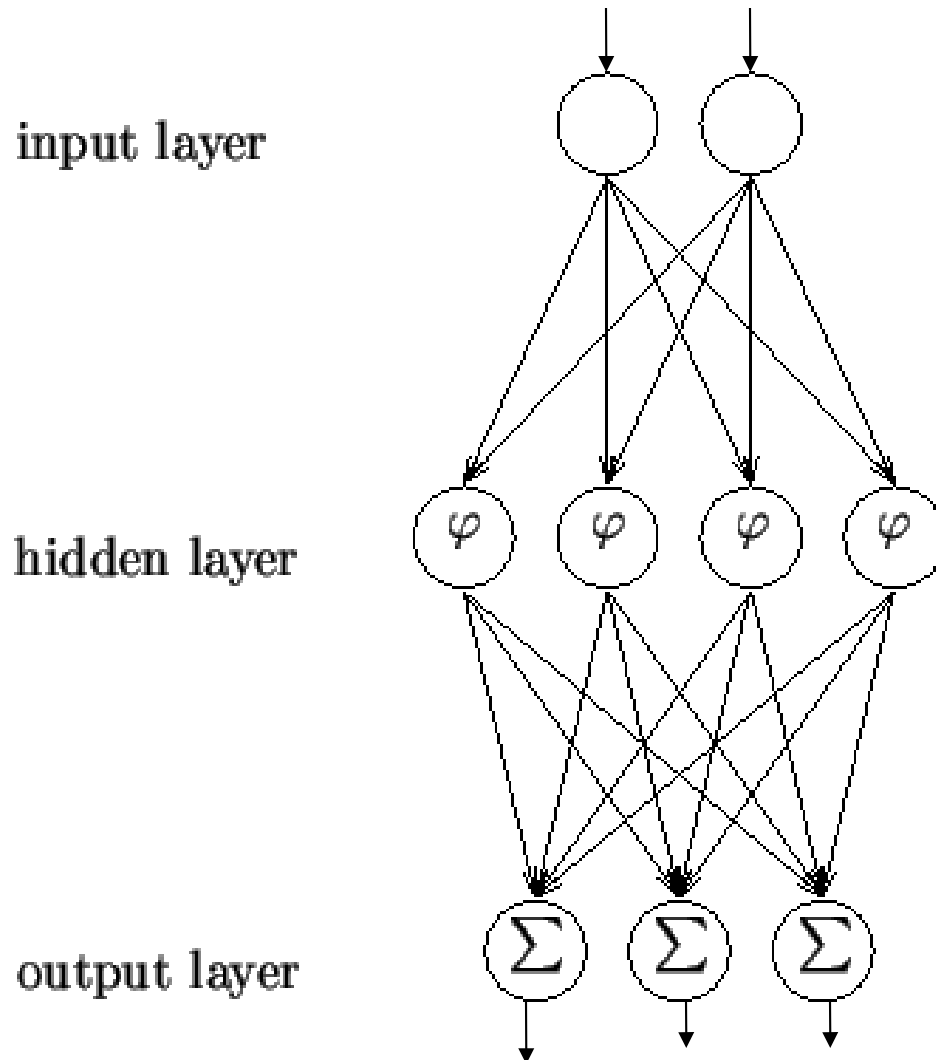
Principe

Fonctions d'activation : linéaire contre non linéaire

Pas unitaire		$f(x) = \begin{cases} 0 & \text{if } 0 > x \\ 1 & \text{if } x \geq 0 \end{cases}$
Sigmoïde		$f(x) = \frac{1}{1+e^{-\beta x}}$
Linéaire Seuillée		$f(x) = \begin{cases} 0 & \text{if } x \leq x_{min} \\ mx+b & \text{if } x_{max} > x > x_{min} \\ 1 & \text{if } x \geq x_{max} \end{cases}$
Gaussienne		$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
Identité		$f(x) = x$

Principe

Réseau Multi-couches (MLP en anglais pour Multi-Layer Perceptron)




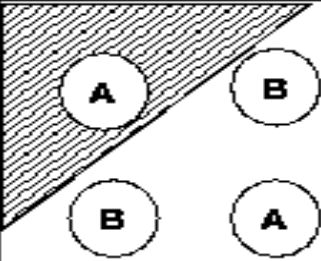
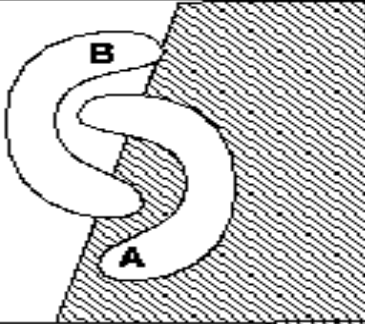
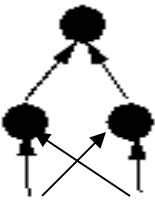
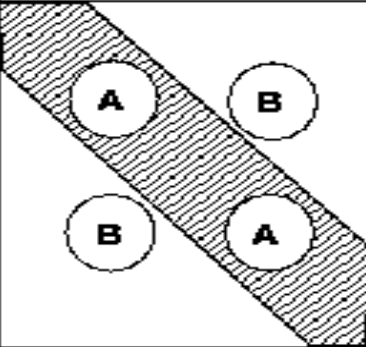
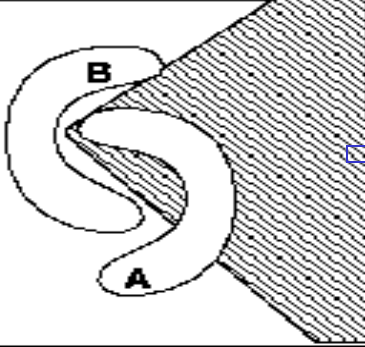
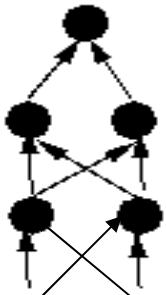
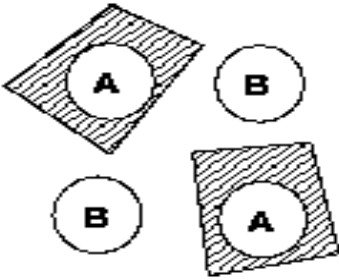
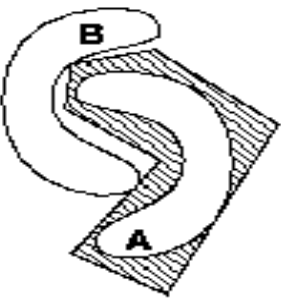
Fonctions d'activation :

φ Non linéaire

Σ Linéaire

Principe

Frontières que l'on peut obtenir dans le cas d'une fonction d'activation ϕ linéaire par morceaux

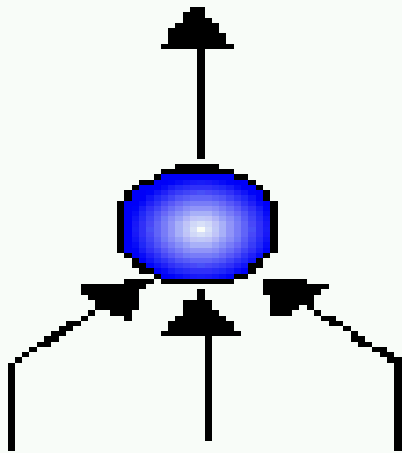
Structure	Regions	XOR	Meshed regions
single layer 	Half plane bounded by hyper-plane		
two layer 	Convex open or closed regions		
three layer 	Arbitrary (limited by # of nodes)		

Une couche cachée

Deux couches cachées

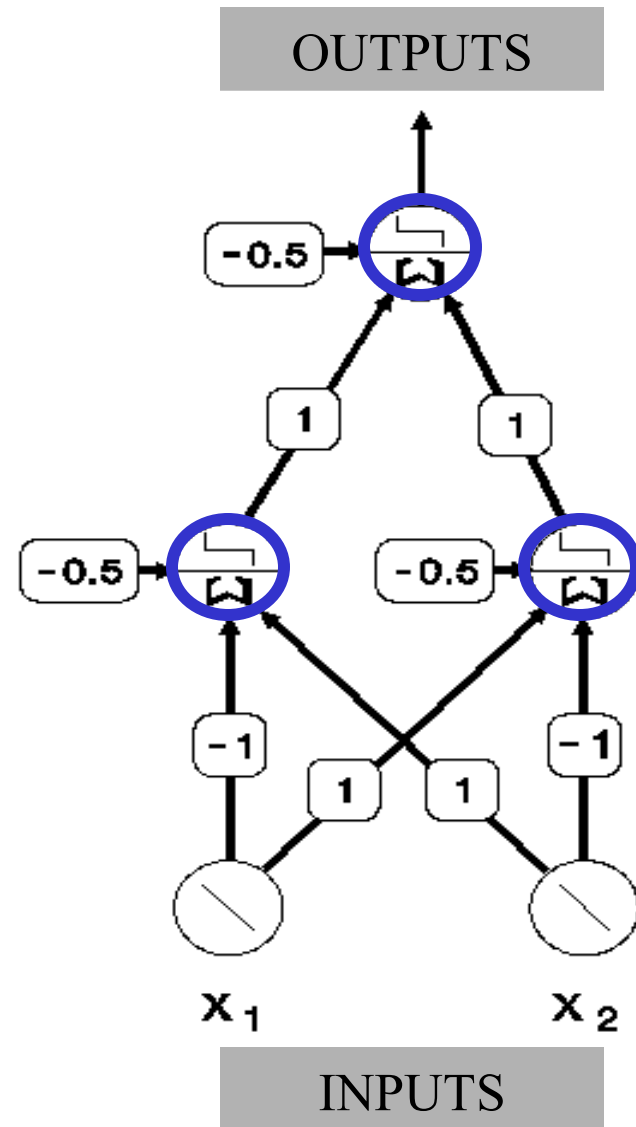
Principe

Mais en général, **une seule couche cachée suffit** à résoudre tout problème de classification pourvu que les neurones de la couche cachée possèdent une fonction d'activation NON linéaire (par exemple, la fonction sigmoïde).



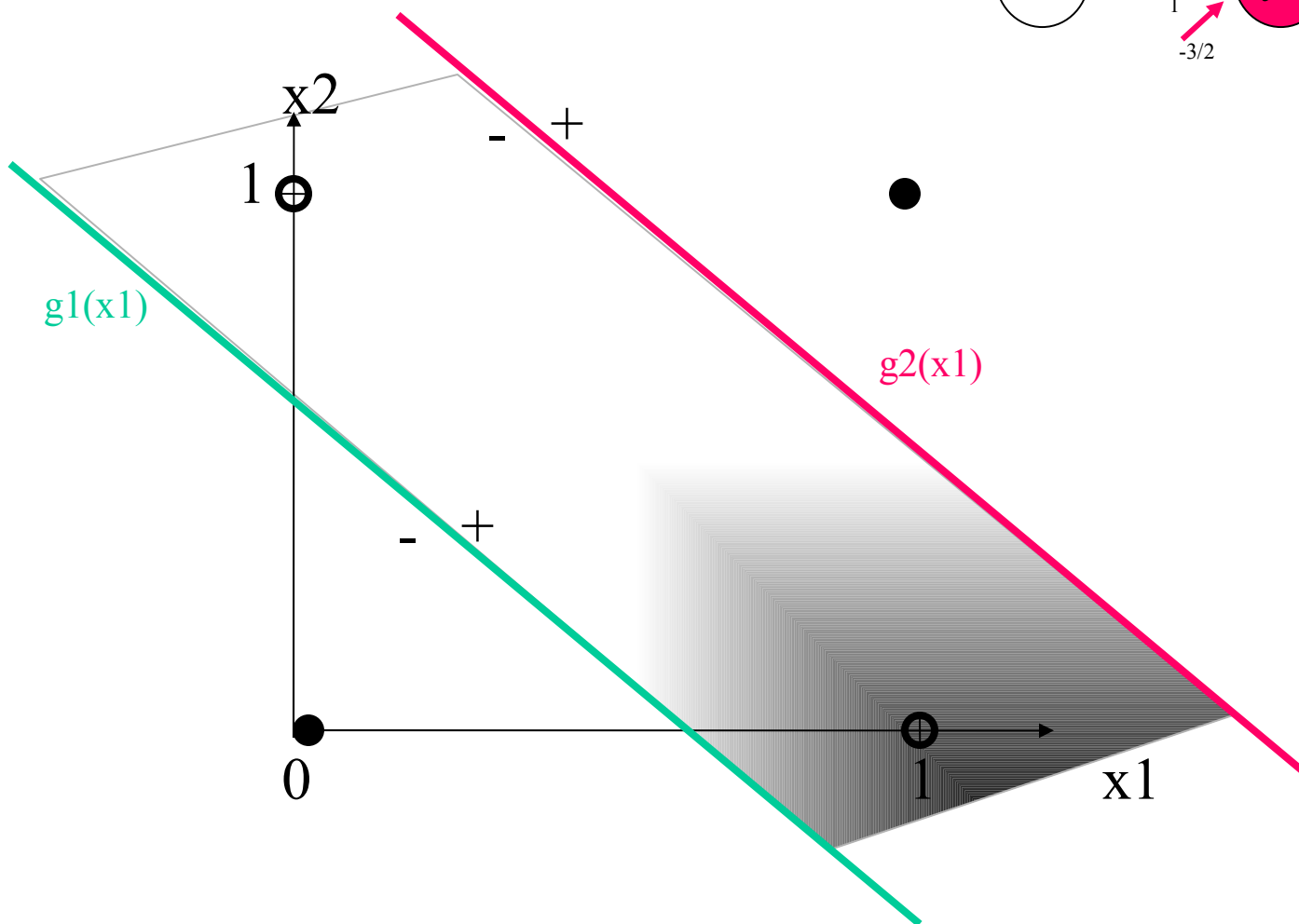
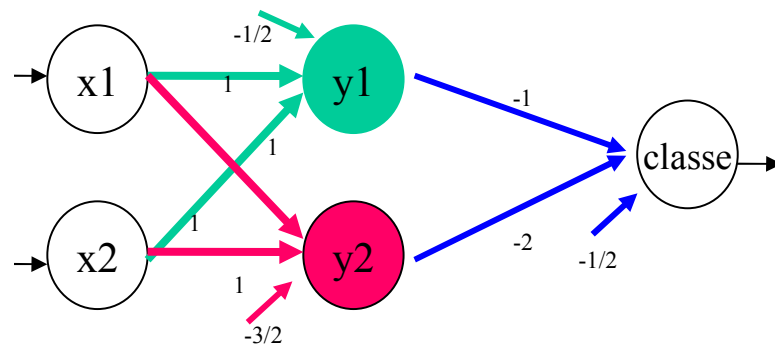
Le perceptron simple

Le perceptron à 1 couche cachée

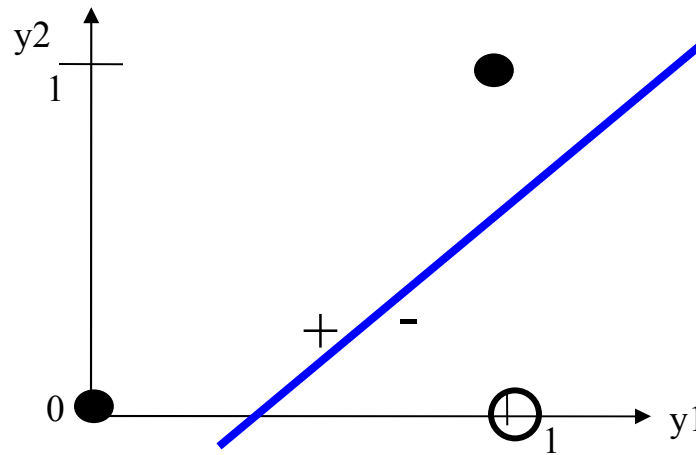


Problème du XOR

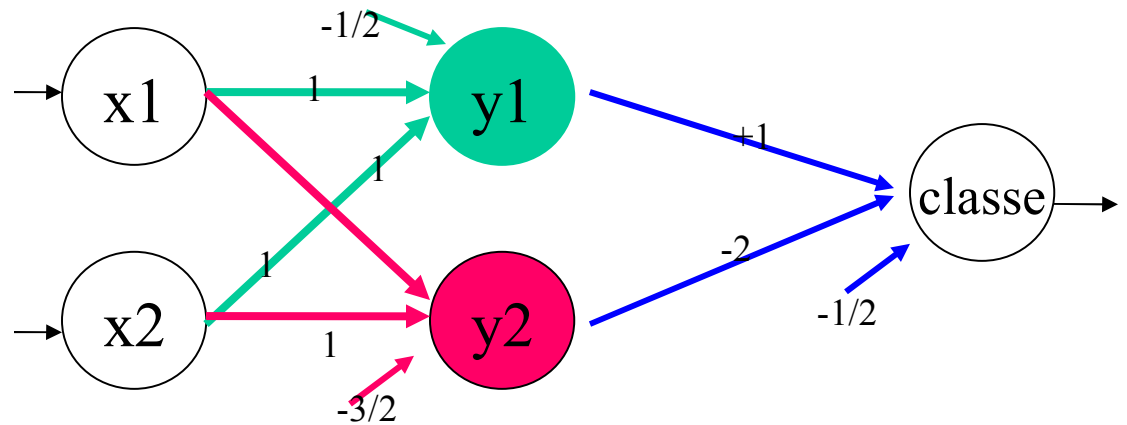
x1	x2	Classe
0	0	0
1	1	0
1	0	1
0	1	1



Projection dans un espace où le problème est linéairement séparable

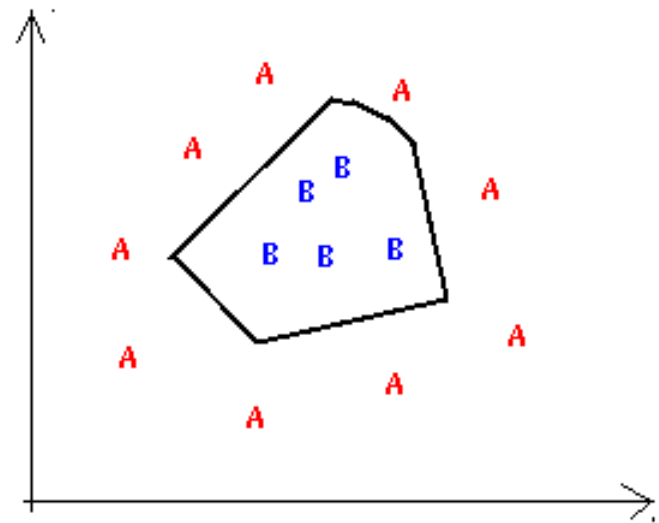
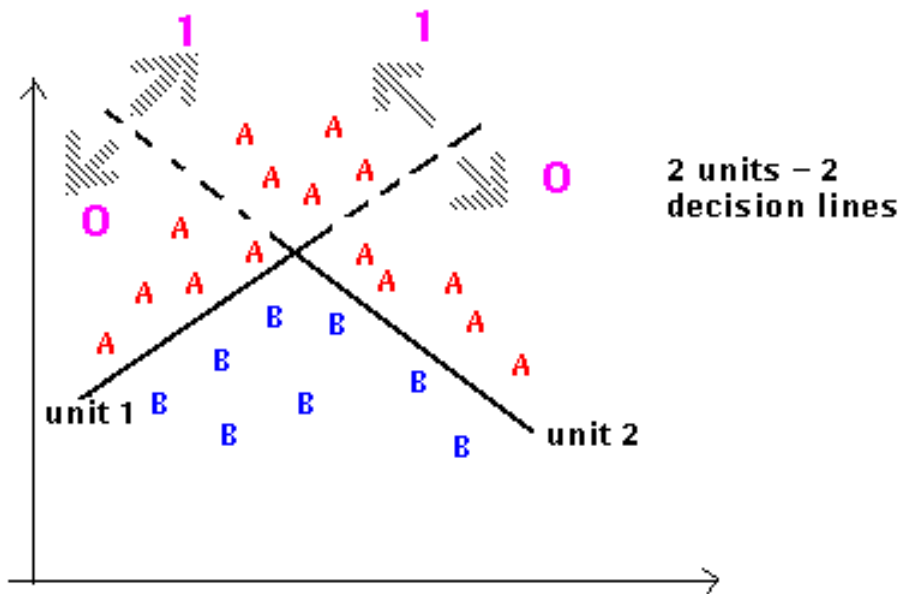


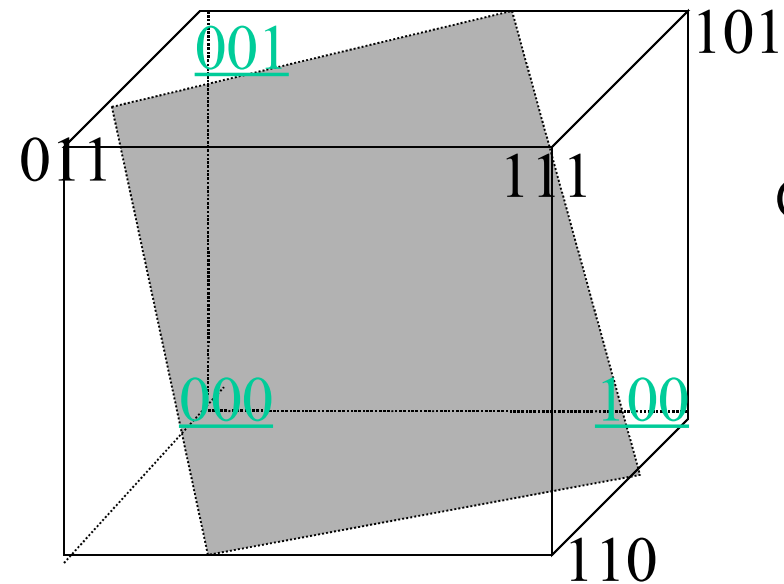
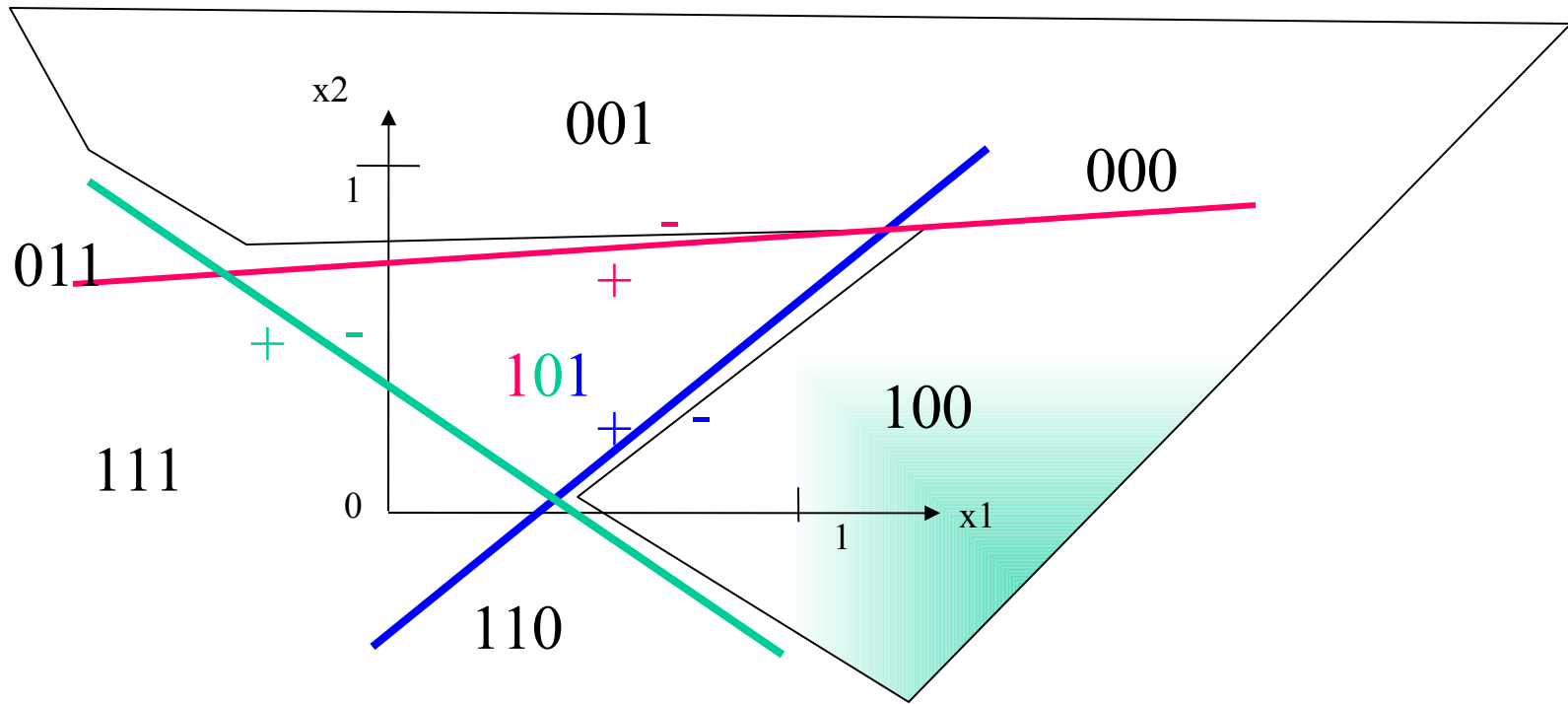
Réseau de neurones à une couche cachée.



Projection du repère (x_1, x_2) vers un hypercube de \mathbb{R}^2 !
 Quelle est la fonction d'activation dans ce cas ?

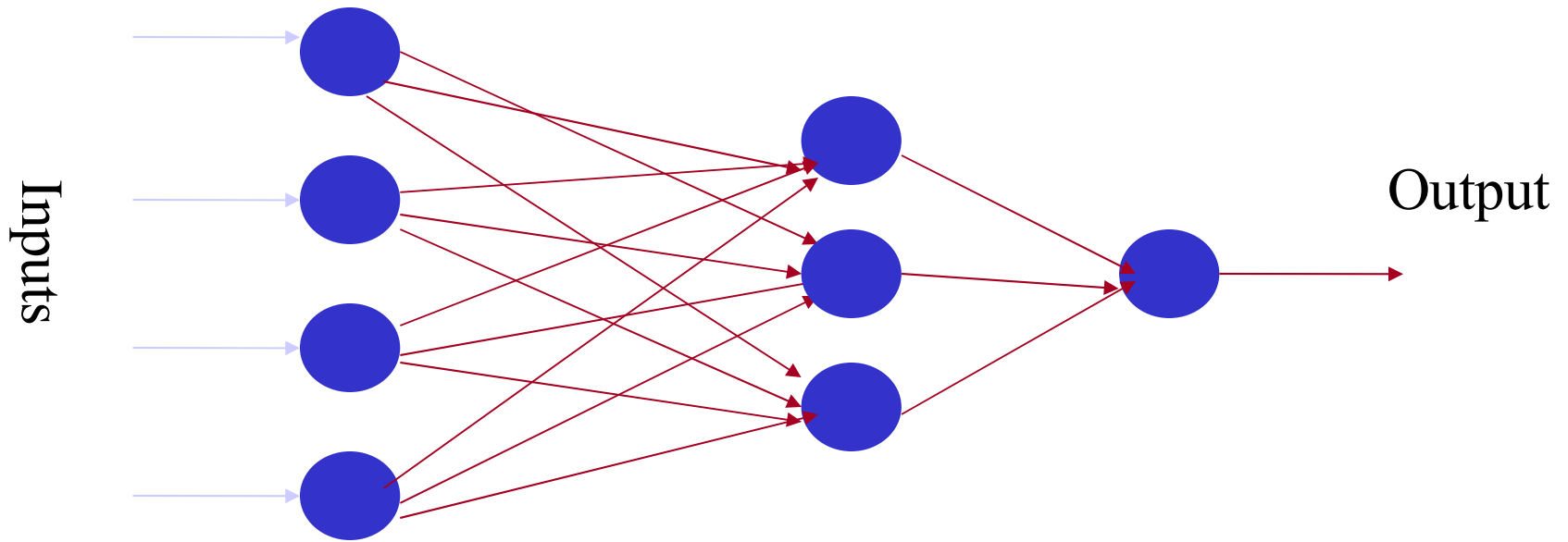
x_1	x_2	y_1	y_2	Classe
0	0	0(-)	0(-)	0
1	1	1(+)	1(+)	0
1	0	1(+)	0(-)	1
0	1	1(+)	0(-)	1





Quelle structure de réseau avons-nous *a priori* ?

Et si la région 111 avait été grisée, cette structure aurait-elle suffi ?



$$y_1^1 = f(x_1, w_1^1)$$

$$y_2^1 = f(x_2, w_2^1)$$

$$y_3^1 = f(x_3, w_3^1)$$

$$y_4^1 = f(x_4, w_4^1)$$

$$y^1 = \begin{pmatrix} y_1^1 \\ y_2^1 \\ y_3^1 \\ y_4^1 \end{pmatrix}$$

Apprentissage biologique

Apprentissage par adaptation :

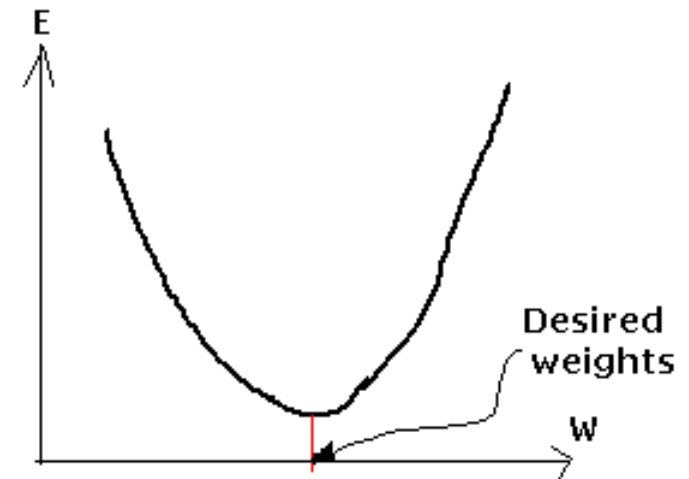
“The young animal learns that the green fruits are sour, while the yellowish/reddish ones are sweet. The learning happens by adapting the fruit picking behaviour.

At the neural level the learning happens by changing of the synaptic strengths, eliminating some synapses, and building new ones.”

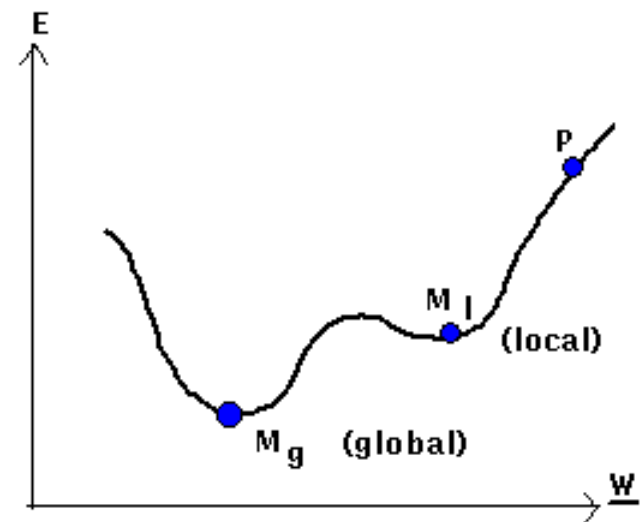
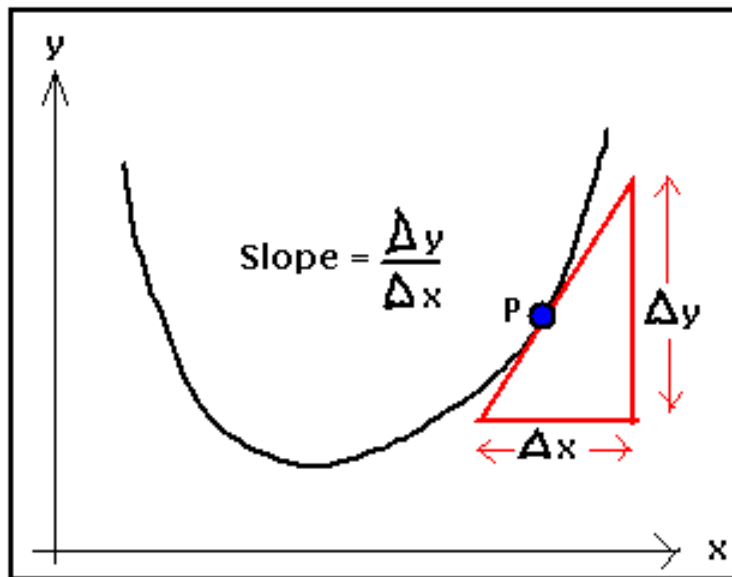
Principe mathématique

Énergie ou Erreur à minimiser

$$E = \frac{1}{N} \sum_{t=1}^N (F(x_t; W) - y_t)^2$$



<http://www.dice.ucl.ac.be/~verleyse/lectures/elec2870/elec2870.htm>



Un principe d'optimisation numérique, si l'Énergie ou Erreur est :

$$E = \frac{1}{N} \sum_{t=1}^N (F(x_t; W) - y_t)^2$$

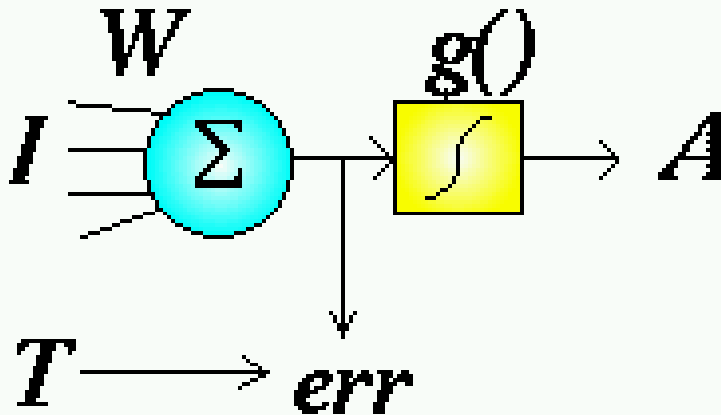
une mise à jour des poids efficace à chaque itération suit la loi

$$\Delta w_i^j = -c \cdot \frac{\partial E}{\partial w_i^j} (W)$$

$$w_i^{j, new} = w_i^j + \Delta w_i^j$$

jusqu'à convergence...

Résolution informatique



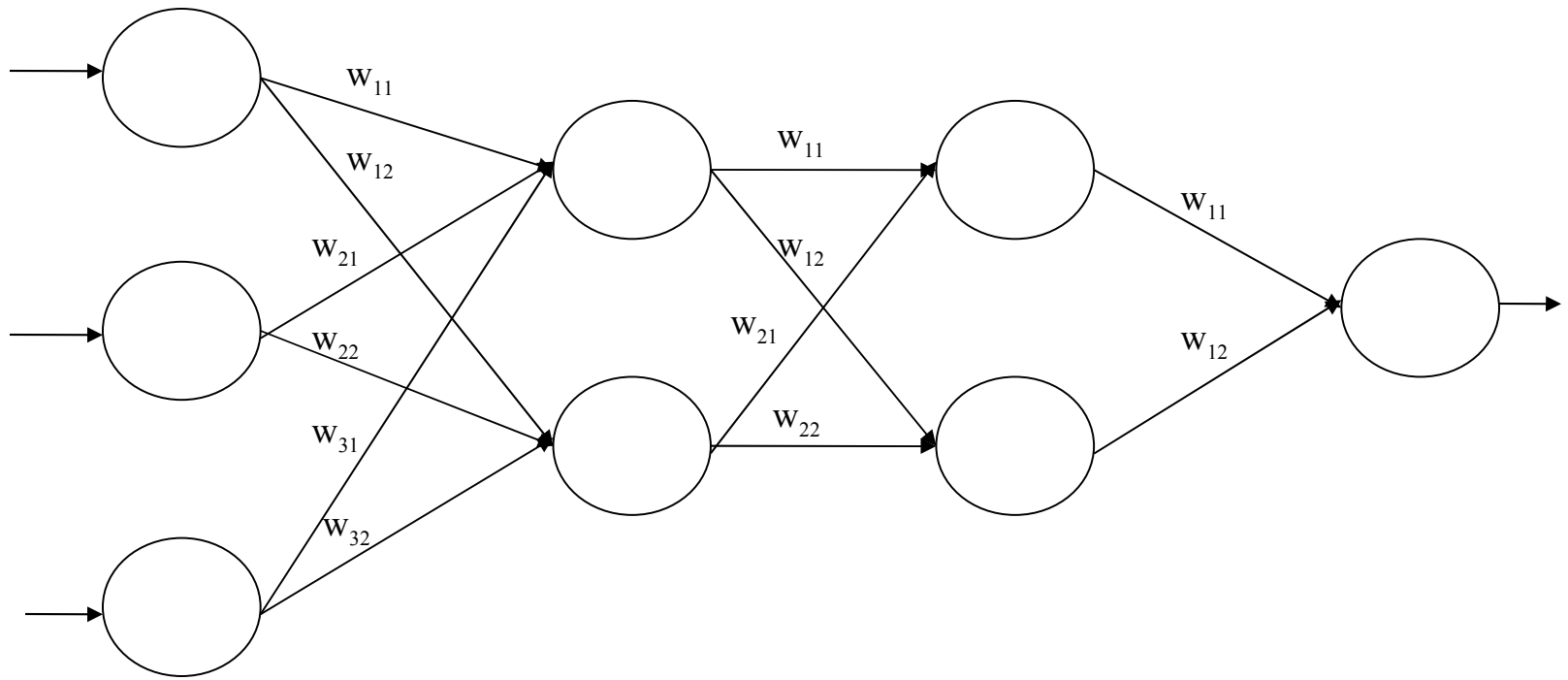
Cas simple:
ADALINE

Mise à jour des poids

$$w_j(t+1) = w_j(t) + \eta(d - y)x$$

d : sortie désirée
 y : sortie obtenue
 x : entrée fournie
 η : pas
d'apprentissage
(learning rate)

Comment faire sur un MLP ?



Algorithme de rétro-propagation du gradient

- RPROP (BackProp) : algorithme d'apprentissage pour un réseau de type MLP
- But : Pour un ensemble d'éléments (d'apprentissage), trouver les poids du réseau MLP qui fournissent la réponse attendue pour chacun des éléments

Box 6. The Back-Propagation Training Algorithm

The back-propagation training algorithm is an iterative gradient algorithm designed to minimize the mean square error between the actual output of a multilayer feed-forward perceptron and the desired output. It requires continuous differentiable non-linearities. The following assumes a sigmoid logistic non-linearity is used where the function $f(\alpha)$

is

$$f(\alpha) = \frac{1}{1 + e^{-(\alpha-\theta)}}$$

Step 1. Initialize Weights and Offsets

Set all weights and node offsets to small random values.

Step 2. Present Input and Desired Outputs

Present a continuous valued input vector x_0, x_1, \dots, x_{N-1} and specify the desired outputs d_0, d_1, \dots, d_{M-1} . If the net is used as a classifier then all desired outputs are typically set to zero except for that corresponding to the class the input is from. That desired output is 1. The input could be new on each trial or samples from a training set could be presented cyclically until weights stabilize.

Step 3. Calculate Actual Outputs

Use the sigmoid nonlinearity from above to calculate outputs $y_0,$

y_1, \dots, y_{M-1} .

Step 4. Adapt Weights

Use a recursive algorithm starting at the output nodes and working back to the first hidden layer. Adjust weights by

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x_i'$$

In this equation $w_{ij}(t)$ is the weight from hidden node i or from an input to node j at time t , x_i' is either the output of node i or is an input, η is a gain term, and δ_j is an error term for node j . If node j is an output node, then

$$\delta_j = y_j(1 - y_j)(d_j - y_j),$$

where d_j is the desired output of node j and y_j is the actual output.

If node j is an internal hidden node, then

$$\delta_j = x_j'(1 - x_j') \sum_k \delta_k w_{jk},$$

where k is over all nodes in the layers above node j . Internal node thresholds are adapted in a similar manner by assuming they are connection weights on links from auxiliary constant-valued inputs. Convergence is sometimes faster if a momentum term is added and weight changes are smoothed by

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x_i' + \alpha(w_{ij}(t) - w_{ij}(t-1)),$$

where $0 < \alpha < 1$.

Step 5. Repeat by Going to Step 2



- Très efficace
- Sans modélisation a priori
- Simple à implémenter



- Un maximum d'exemples (comportement statistique, loi des grands nombres)
- Effet boîte noire : comportement interne difficile à expliquer, modéliser

SVM

Nouvelle modélisation des Neural Network :
les SVM ou Support Vector Machine

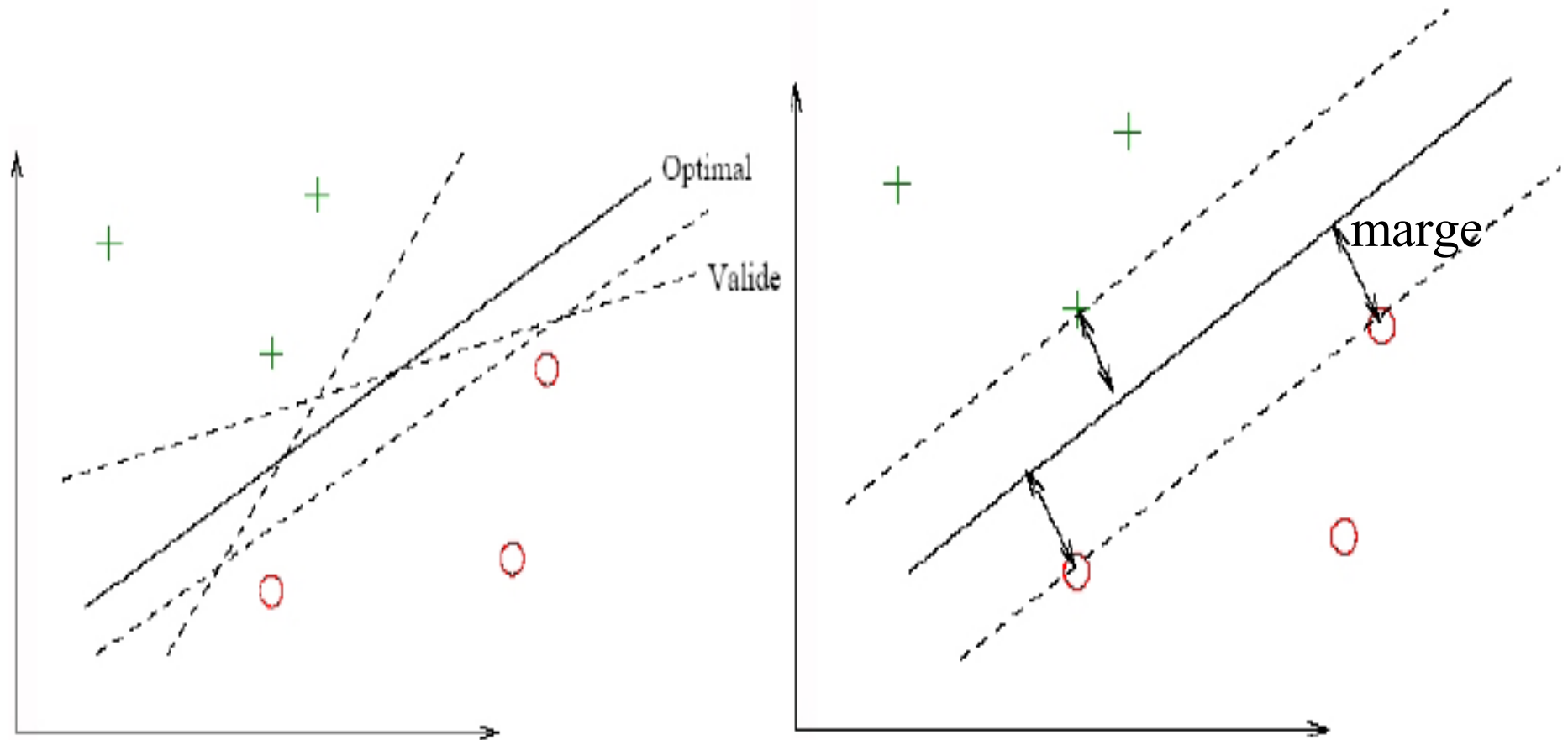
Utile quand peu d'exemples d'apprentissage et
problème à 2 classes

Carte de Kohonen

Auto-organisation de données

Dans le cadre des SVM, on cherche l'hyperplan de marge optimale pour séparer correctement les données tout en étant éloigné le plus possible de toutes les observations.

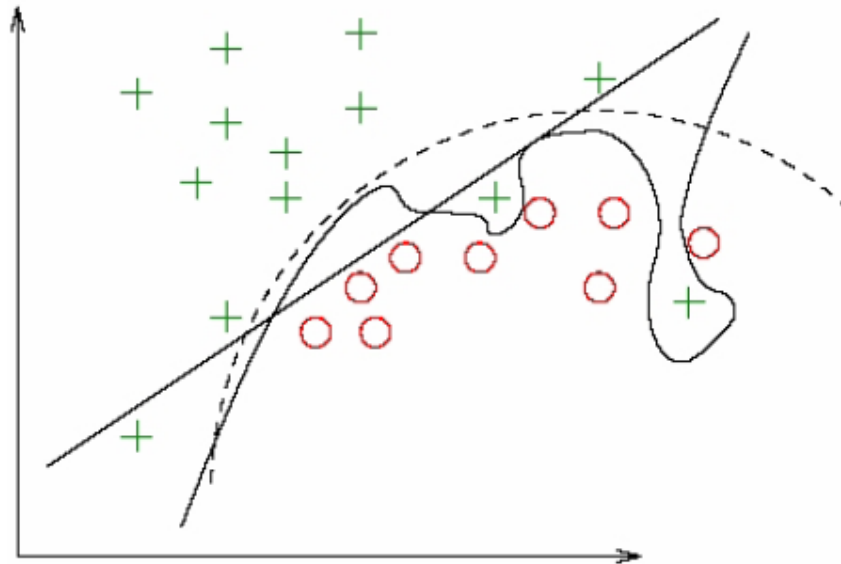
On cherche donc à trouver un classifieur ou une fonction de discrimination dont la capacité de généralisation (qualité de la prédiction) est la plus grande possible.



SVM = Séparateurs à vastes marges
= Support Vector Machines
= Machine à noyaux ou kernel machines

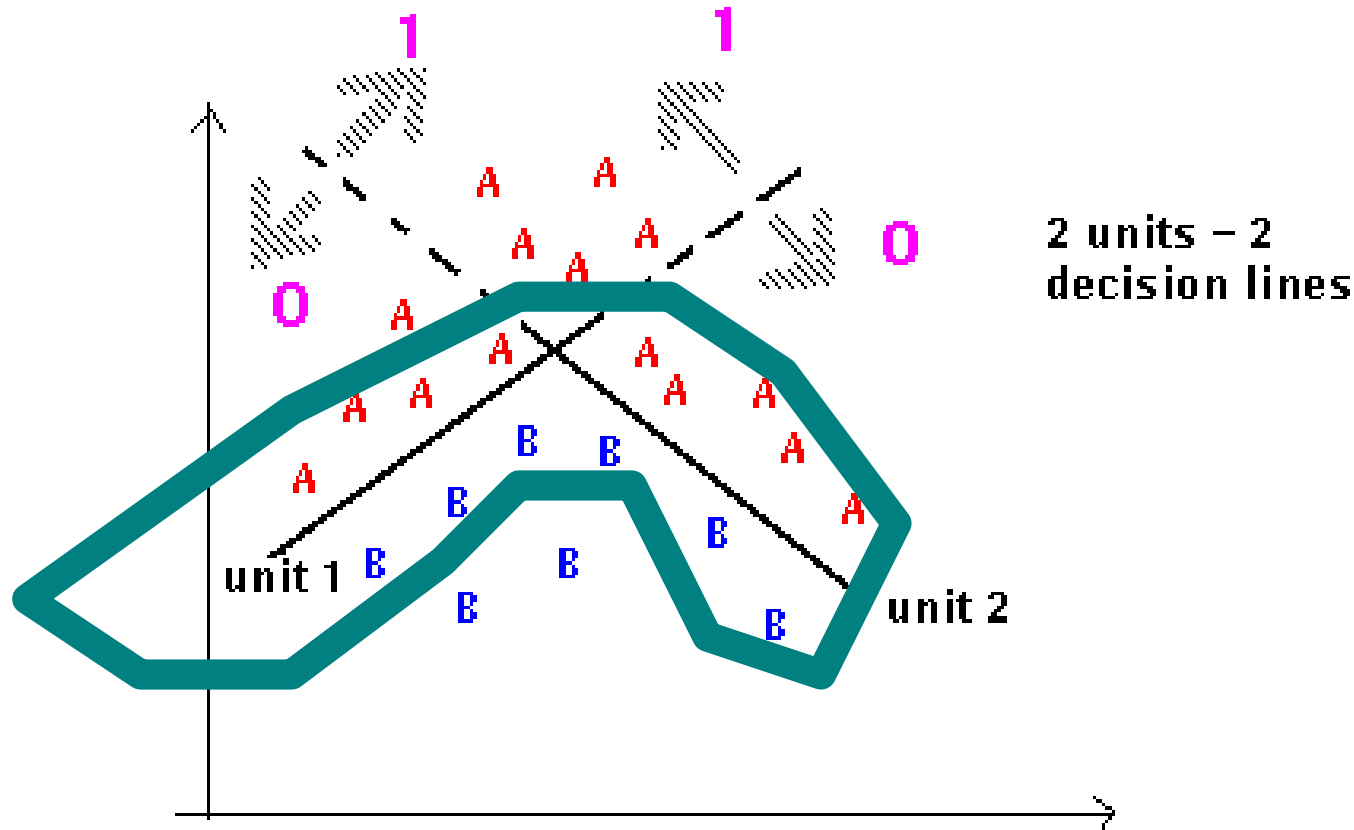
Liés aux fonctions noyaux pour introduire de la non linéarité dans des espaces de plus grandes dimensions en particulier (cf. fonction d'activation sigmoïde du MLP) et à la dimension de Vapnik Chernovenkis

Compromis entre **complexité** de la frontière (capacité d'ajustement du modèle) et qualités de **généralisation** ou prédiction de ce modèle



Sous-ajustement linéaire et sur-ajustement local (proches voisins) d'un modèle quadratique.

Dans le cas de SVM, on va surtout s'appuyer sur les « support vectors » les plus proches de la frontière pour les calculs et la modélisation de la frontière optimale.



Pour les software et tutorial,
<http://www.kernel-machines.org/>

Cas à 2 classes :

Soit une variable Y à prédire prenant ses valeurs dans $\{-1,1\}$.
Soit $X = X^1, \dots, X^p$ les variables explicatives ou prédictives.
 $X \in \mathbb{R}^p$ ou plus généralement $X \in F$ quelconque

Soit $m(X)$ un modèle pour Y .
On note $z = \{(x_1, y_1), \dots, (x_n, y_n)\}$ un échantillon statistique de taille n .

L'objectif est de construire une estimation m^e de m , fonction de F dans $\{-1,1\}$ de sorte que la probabilité $P(m(X) \neq Y)$ soit minimale.

L'astuce est de rechercher plutôt qu'une fonction m^e à valeurs dans $\{-1,1\}$ une fonction réelle f dont le signe fournira la prédiction :
$$m^e = \text{signe}(f)$$

L'erreur s'exprime alors comme la quantité :
$$P(m(X) \neq Y) = P(Yf(X) \leq 0)$$

De plus, la quantité $|Yf(X)|$ fournit un indicateur sur la confiance à accorder au résultat du classement. $Yf(X)$ est la marge de f en (X, Y) .

Espace intermédiaire :

une première étape consiste à transformer les valeurs de X , cad les objets de F par une fonction T à valeurs dans un espace H intermédiaire (feature space) muni d'un produit scalaire et d'une plus grande dimension.

Cette transformation est fondamentale dans le principe des SVM, elle prend en compte l'éventuelle non linéarité du problème posé et le ramène à la résolution d'une séparation linéaire.
(cf. projection 2D vers 3D une dizaine de transparents précédemment).

En pratique, il n'est pas nécessaire d'explicitier la transformée T ce qui serait souvent impossible, à condition de savoir exprimer les produits scalaires dans H à l'aide d'une fonction $k:F \times F \rightarrow \mathbb{R}$ symétrique appelé noyau de sorte que :

$$k(x, x') = \langle T(x), T(x') \rangle$$

Bien choisi, le noyau permet de matérialiser une notion de « proximité » adaptée au problème de discrimination et à sa structure de donnée.

Exemple

Prenons le cas trivial où $\mathbf{x} = (x_1, x_2)$ dans \mathbb{R}^2 et $\Phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$ est explicite. Dans ce cas, \mathcal{H} est de dimension 3 et le produit scalaire s'écrit :

$$\begin{aligned}\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle &= x_1^2x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2x_2'^2 \\ &= (x_1x_1' + x_2x_2')^2 \\ &= \langle \mathbf{x}, \mathbf{x}' \rangle^2 \\ &= k(\mathbf{x}, \mathbf{x}').\end{aligned}$$

Le calcul du produit scalaire dans \mathcal{H} ne nécessite pas l'évaluation explicite de Φ . D'autre part, le plongement dans $\mathcal{H} = \mathbb{R}^3$ peut rendre possible la séparation linéaire de certaines structures de données (cf. figure 10.3).

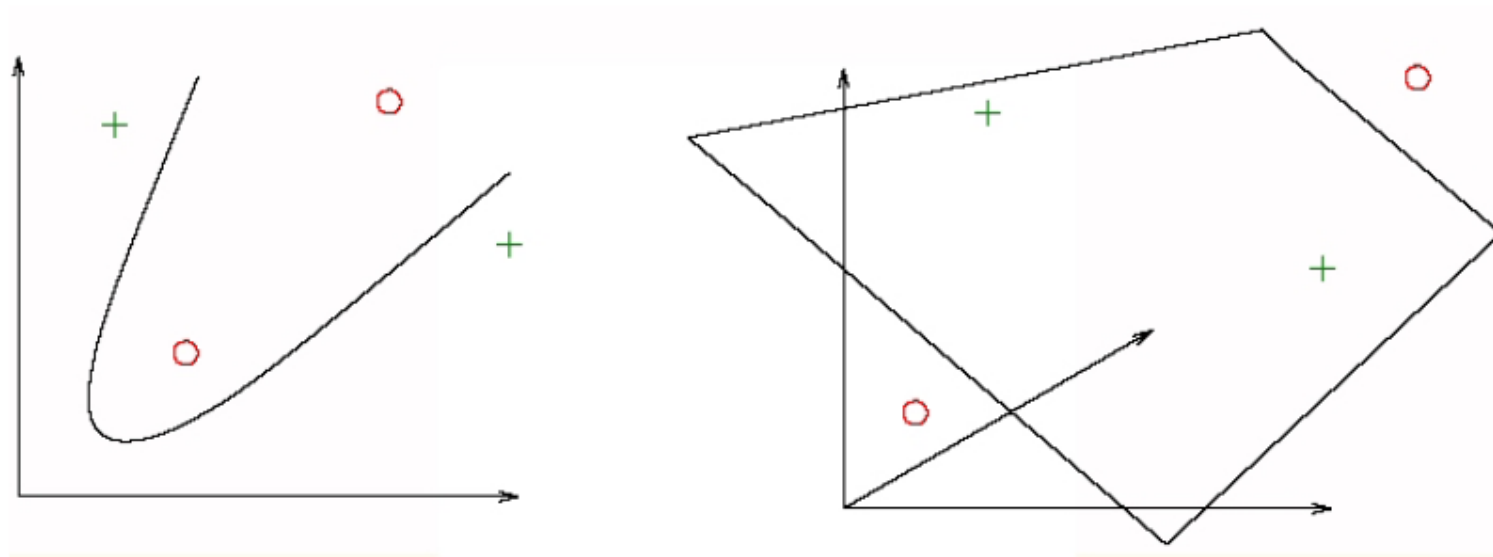


FIG. 10.3 – Rôle de l'espace intermédiaire dans la séparation des données.

Exemples de noyaux parmi une infinité (c'est la difficulté de construction et d'utilisation grand public de l'outil SVM)

- Linéaire

$$k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$$

- Polynômial

$$k(\mathbf{x}, \mathbf{x}') = (c + \langle \mathbf{x}, \mathbf{x}' \rangle)^d$$

- Gaussien

$$k(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}}$$

D'où, procédure de stabilisation par essai erreur en utilisant une bonne méthode d'évaluation des erreurs de prédiction par exemple par validation croisée.

Par rapport aux réseaux de neurones, les SVM mettent en oeuvre des stratégies d'optimisation spécifiques.

<http://svm.dcs.rhbnc.ac.uk/>

<http://www-ai.cs.uni-dortmund.de/SOFTWARE/YALE/index.html>

<http://www.smartlab.dibe.unige.it/smartlab/software/software.html>

POS NEG REMOVE Monochromatic

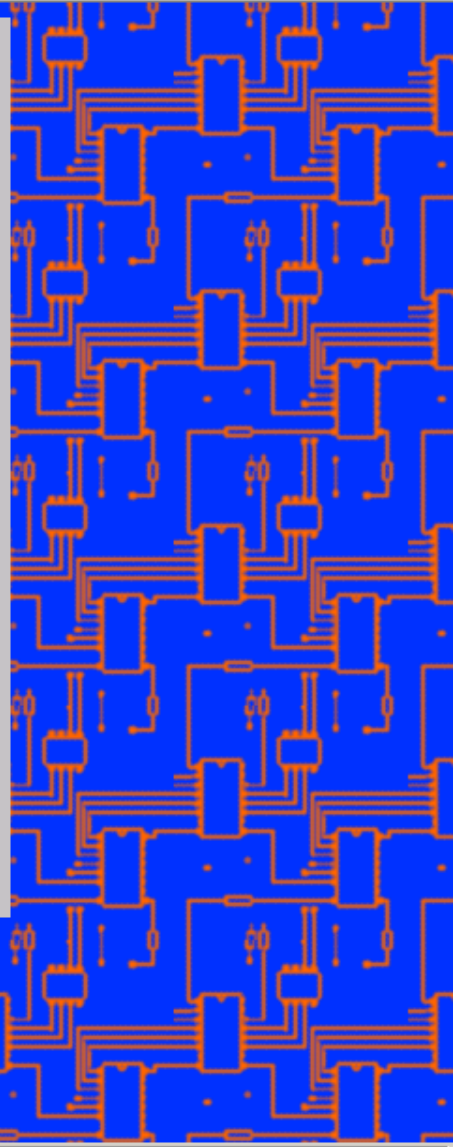
X= 0.8359375 Y= -0.046875 value=-0.183468

Eta:
 Kernel type:

Variance:
 C:
 Threshold: $10^{-16.0}$
 Max N. Iter. Done: 50000
 Draw every iterations
 Bias

Numeric Input (options)

SVM Output:
 17: 4.2345743077922156E-
 18: 5.175030864334416E-7
 19: 9.999999735282774
 20: 2.0981435045283665
 21: 0.21405845990751549
 22: 0.4749925823086071
 23: 0.26840598751928674
 24: 0.5092214865088621
 25: 0.6327266171265122
 26: 0.8763353632579831



Applications

Taches que peut résoudre un Réseau de Neurones Artificiels:

- contrôler le mouvement d'un robot en se fondant sur la perception;
- décider de la catégorie de certaines nourritures (comestibles ou non) dans les mondes artificiels (jeux);
- prédire si une séquence ADN est une séquence codante correspondant à un gène;
- prédire le comportement de valeurs boursières;
- prédire le comportement de futurs utilisateurs d'un service ...

DNA Sequence Analysis

Eukaryotic Gene Structure

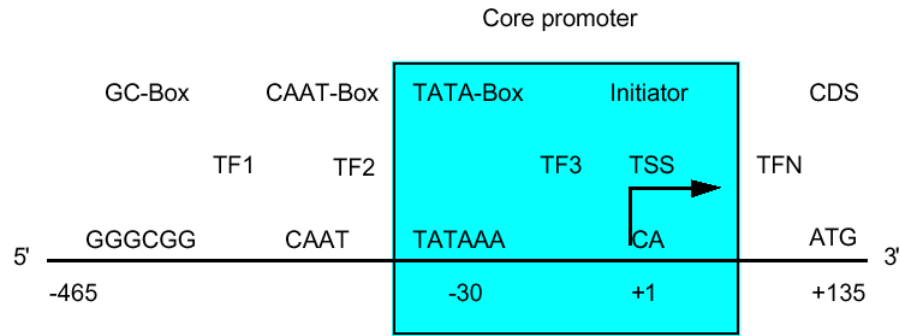


transcription



translation

protein



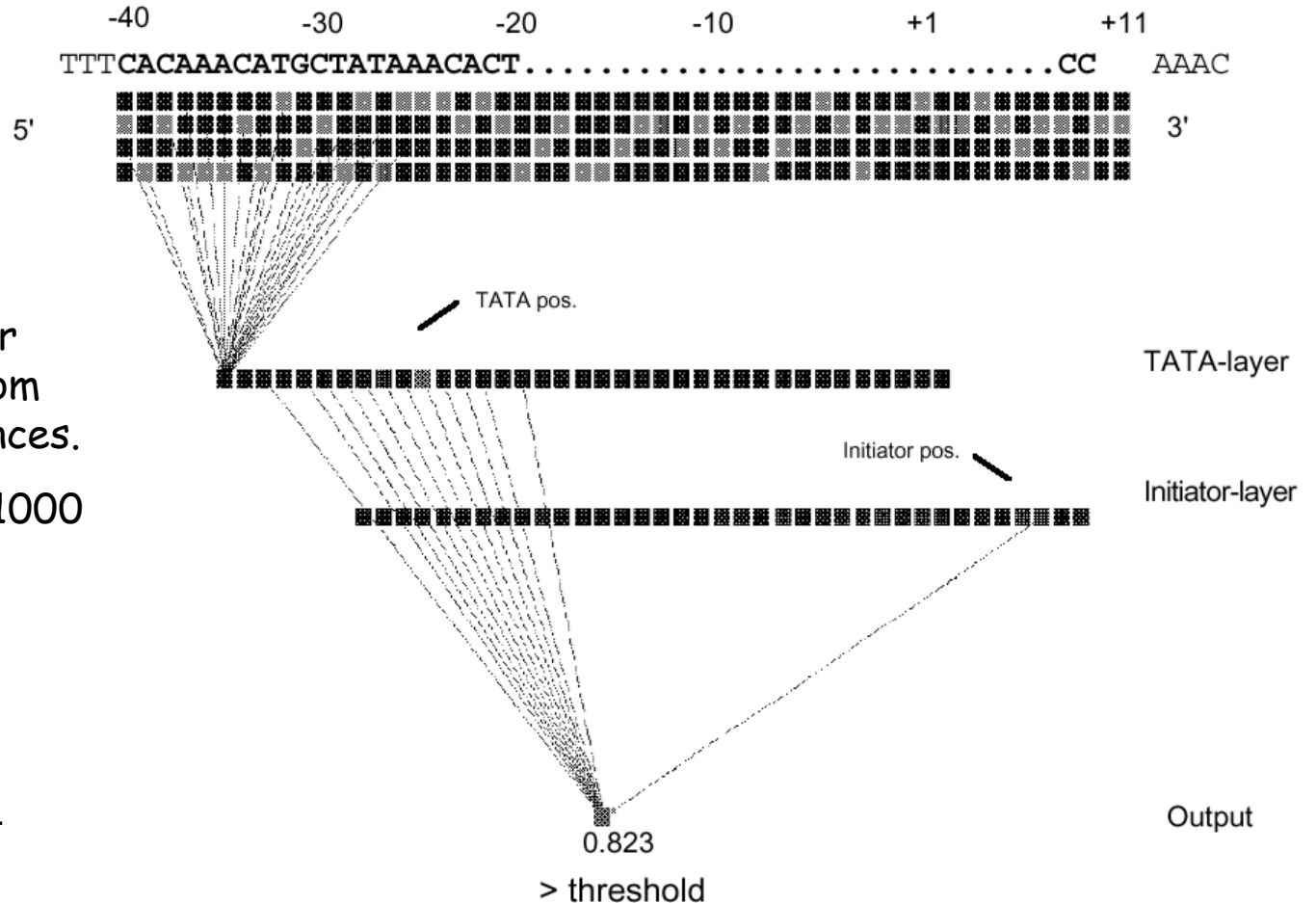
TDNN

(Time-delay

Neural Network)

Training: 300 promoter sequences, 3000 random NON-promoter sequences.

Test: 129 promoters, 1000 random sequences.



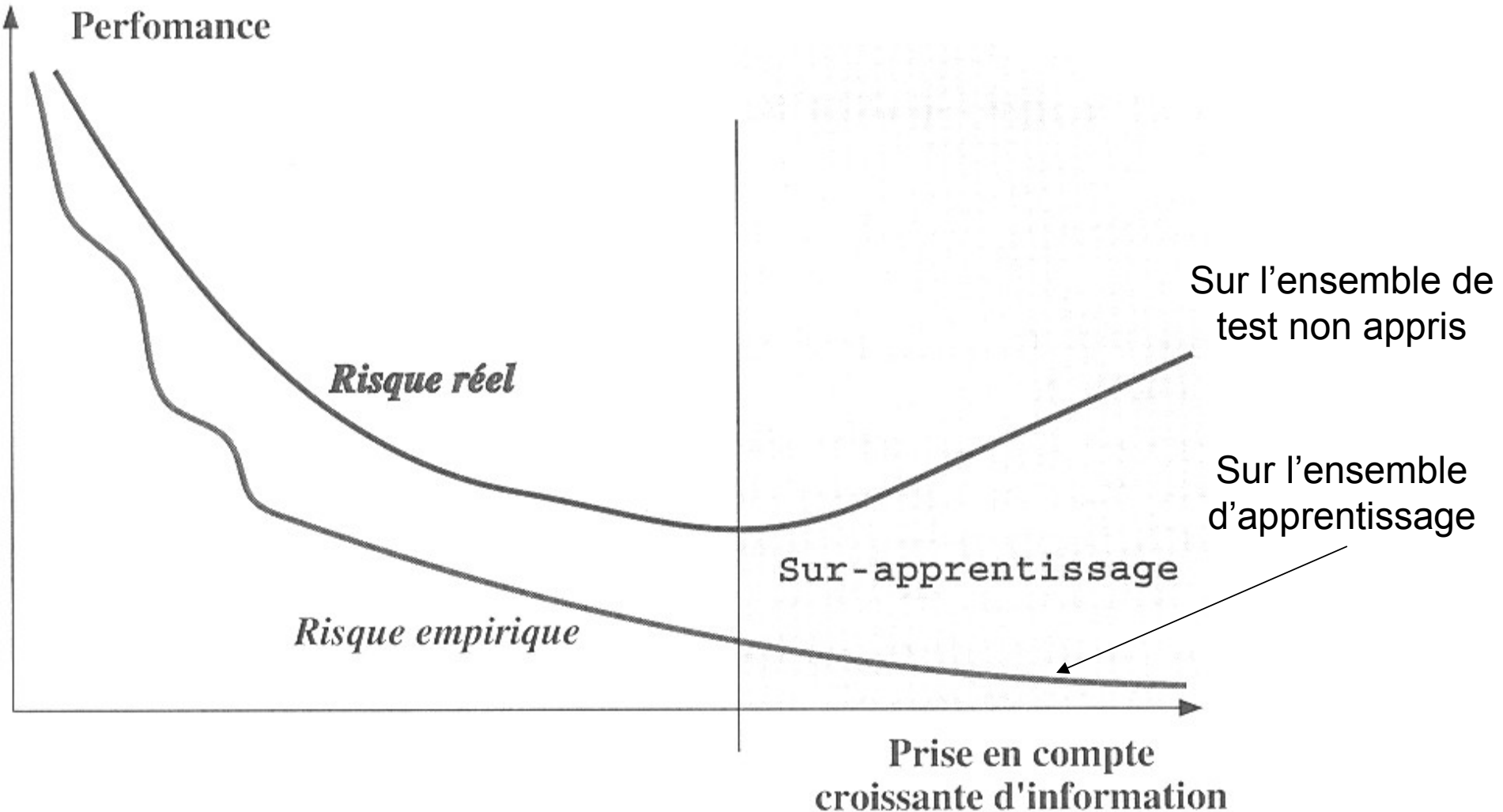
(BDGP Server)
 Martin G. Reese and Frank H. Eeckman.

Evaluation de l'apprentissage supervisé

Input : algorithme paramétrable + ensemble d'exemples A

Output: hypothèse h

Question : évaluer la performance de cette hypothèse



On travaille en général sur deux ensembles : un ensemble d'apprentissage A et un ensemble de test T.

Une estimation du risque réel de l'hypothèse h proposée sur l'ensemble de test T peut être obtenue à partir de la matrice dite de confusion.

Dans le cas binaire par exemple, c'est-à-dire dans le cas du test d'une hypothèse (une classe) indépendamment des autres classes (hypothèses), on a

	'+' prédit	'-' prédit
'+' réel	Vrais positifs	Faux positifs
'-' réel	Faux négatifs	Vrais négatifs

Risque Réel (h) = Somme des termes non diagonaux / Nombre d'exemples
= Somme des exemples mal classés / Nombre d'exemples

Considérons le cas où je possède un ensemble E de 1000 exemples pour apprendre. Pour valider l'apprentissage, j'ai plusieurs choix pour l'ensemble d'apprentissage et l'ensemble de test. Par exemple :

- A = 2/3 de E et T = le 1/3 restant
- A = 1/2 de E et T = la 1/2 restante

On voit bien que les risques réels mesurés $R(h,T)$ dans chacun des cas seront différents et on sent bien que plus T sera grand et plus la mesure réelle du risque sera proche de sa véritable valeur.

Mais, plus T est grand et plus A est petit puisque les ensembles doivent rester décorrélés et donc moins l'apprentissage sera efficace.

Conclusion : cette méthode (dite **hold-out**) de validation est correcte si E possède beaucoup d'exemples.

Dans le cas contraire, on utilisera d'autres méthodes statistiquement correcte pour estimer la validité d'un apprentissage sur un ensemble réduit d'exemples.

L'estimation par validation croisée (N-fold cross-validation)

- Diviser A en N sous-échantillons de tailles égales
 - Retenir l'un de ces échantillons N_i pour le test et apprendre sur les N-1 autres
 - Mesurer le taux d'erreurs $R(h_i, N_i)$ sur N_i
 - Recommencer n fois en faisant varier l'échantillon i de 1 à N
- L'erreur estimée finale est la moyenne des $R(h_i, N_i)$ pour i de 1 à N.

Souvent N varie entre 5 et 10.

On refait souvent un apprentissage global sur A tout entier (plutôt que de choisir une des hypothèse h_i). Mais la procédure précédente est utile pour avoir une bonne mesure de la validité ou du taux d'erreur de la méthode d'apprentissage choisie.

A l'extrême, quand A est très petit, le **leave-one-out** mais moins performant.

Une technique aléatoire sensée être statistiquement encore plus performante : le **bootstrap**.

Enfin, jusqu'ici on essayait d'estimer au mieux la performance d'une méthode mais pour comparer différentes méthodes entre elles ou différents jeux de paramètres d'une même méthode, on est amené à considérer 3 ensembles : l'ensemble A d'apprentissage, l'ensemble T de test, et l'ensemble V de validation.

La matrice de confusion : mesure globale de la performance de la classification en toutes les classes de l'ensemble de validation

Grab the column header separators to resize the columns

Region/Class	[2] Land	[1] Sea	[3] Forrest
Land	151	0	3
Sea	0	984	0
Cloud	17	0	390

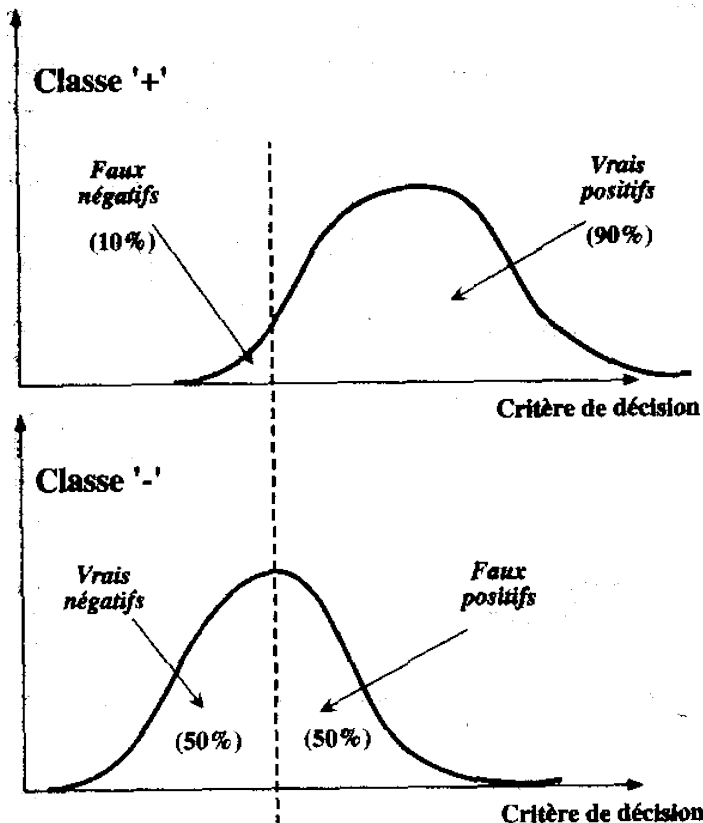
La courbe ROC (Receiver Operating Characteristics -> voir radar/World War))

Dans un contexte de prise de décision, la performance intrinsèque en terme de taux d'erreur indifférencié n'est pas suffisante.

Les taux de "faux positifs" et de "faux négatifs" sont des estimateurs précieux : faire une erreur sur la prédiction d'une maladie grave n'est pas équivalent selon qu'on la laisse passer (faux négatif) ou qu'on la détecte ("faux positif").

Ces taux sont disponibles à partir de la matrice de confusion.

La courbe ROC est utilisée dans le cadre de classification à 2 classes.

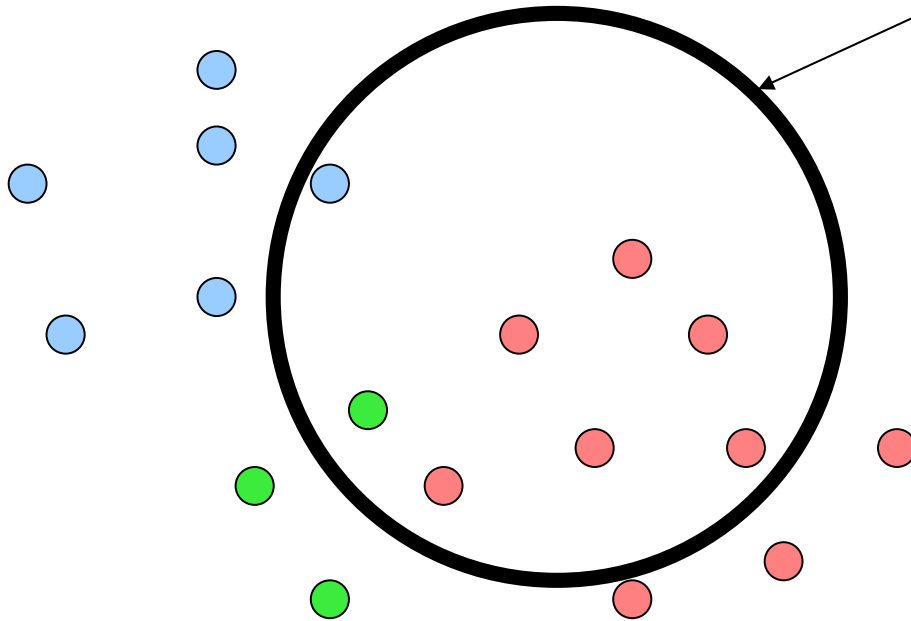


Remarque : classe = hypothèse = test

	'+'	'-'
'+'	Vrais positifs	Faux positifs
'-'	Faux négatifs	Vrais négatifs

Mesures de la qualité de prédiction d'une classe par rapport à toutes les autres

Ensemble des N éléments déclarés roses en phase de décision (test), c'est-à-dire qui répondent positivement au test (hypothèse) de la classe rose après apprentissage.



$$\textit{Précision} = \frac{(6 \textit{ roses dans cercle noir})}{(6 + 2 \textit{ non roses dans cercle noir})}$$

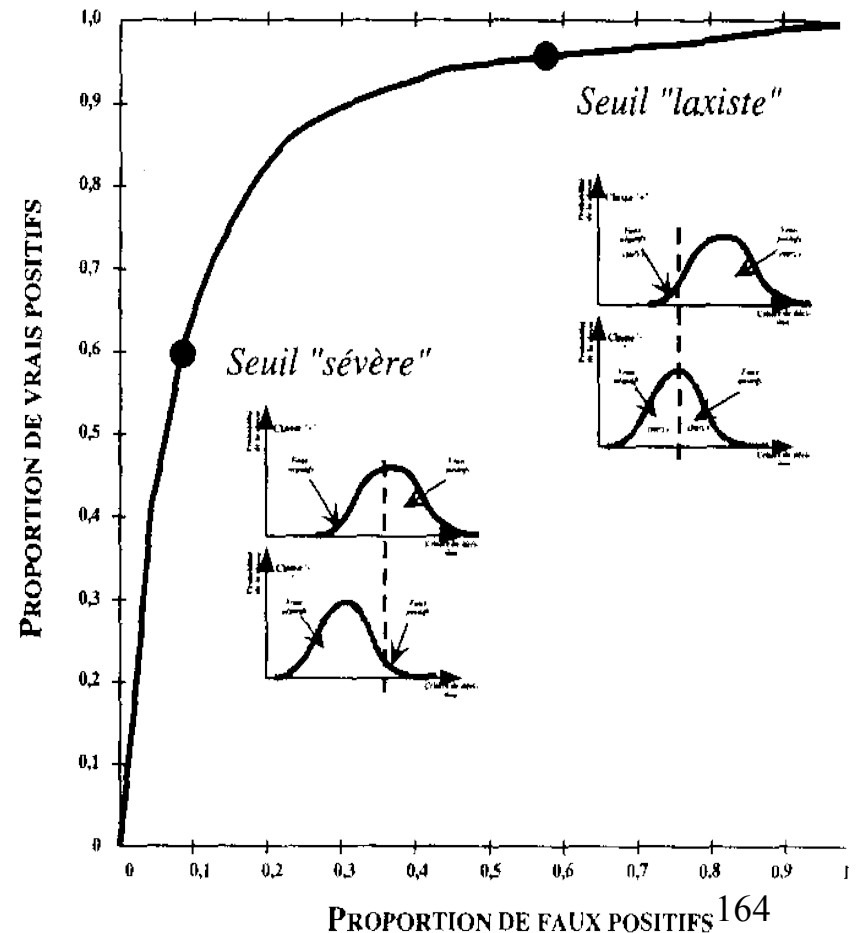
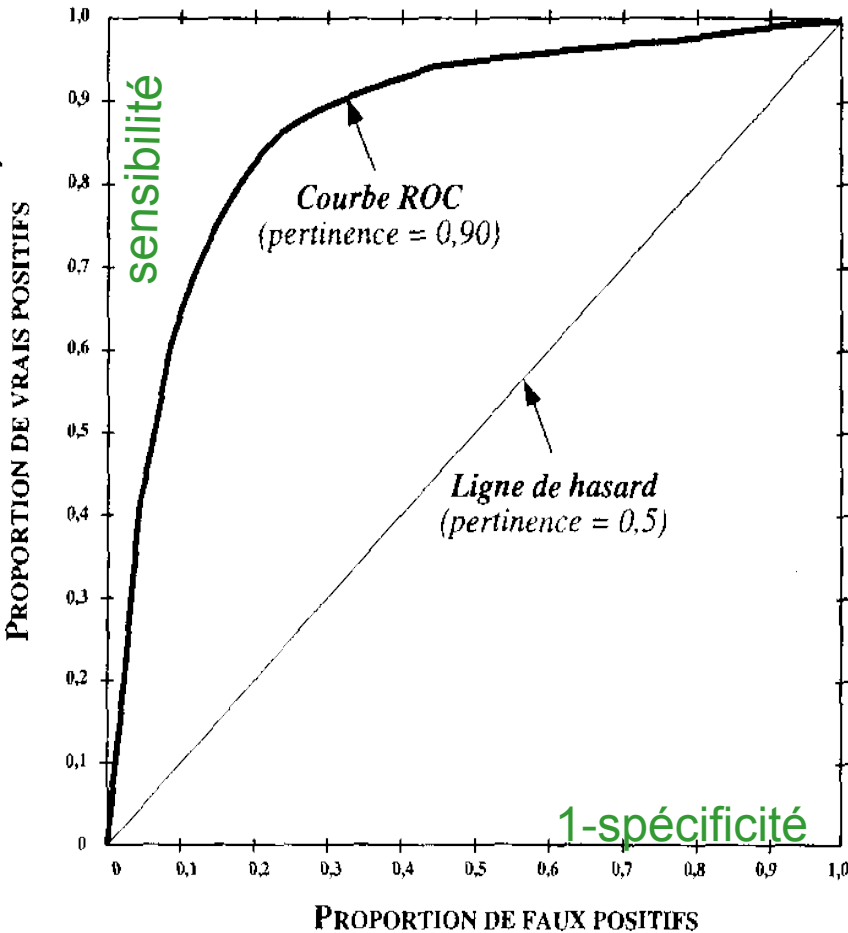
$$\textit{Recall} = \frac{(6 \textit{ roses dans cercle noir})}{(6 + 3 \textit{ roses hors du cercle noir})}$$

Sensibilité = taux de détection ou reconnaissance = Recall = $TP/(TP+FN) = TP/\{\text{Exemples positifs}\}$

Spécificité = 1-taux de fausses alarmes = $TN/(TN+FP) = TN/\{\text{Exemples négatifs}\}$

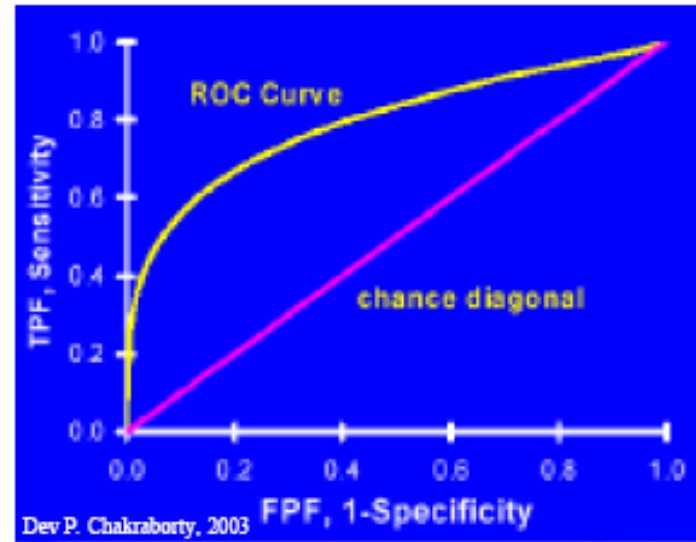
Taux de fausses alarmes = $FP/(TN+FP) = 1 - \text{spécificité}$

Précision = $TP/(TP+FP)$ (utilisée en médecine essentiellement)



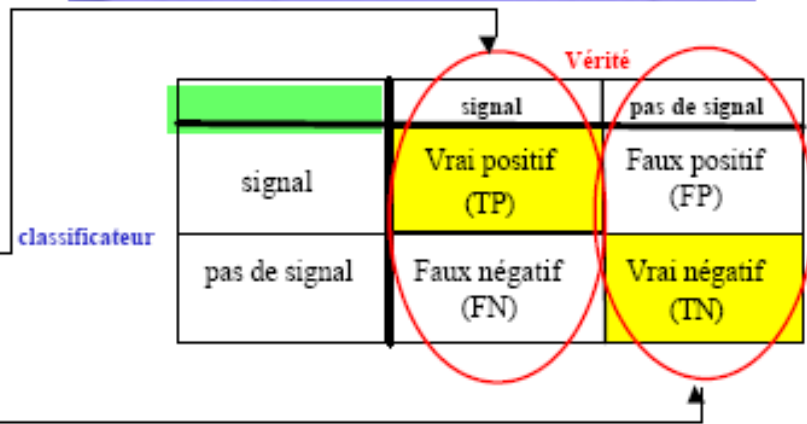
Courbes ROC

- 2e étape:
 - Taux de détection (sensitivité)
 - Taux de fausses alarmes (1 - spécificité)



Sensitivité: $TPF = \frac{TP}{TP + FN}$ Total positif

Spécificité: $TNF = \frac{TN}{TN + FP}$ Total négatif

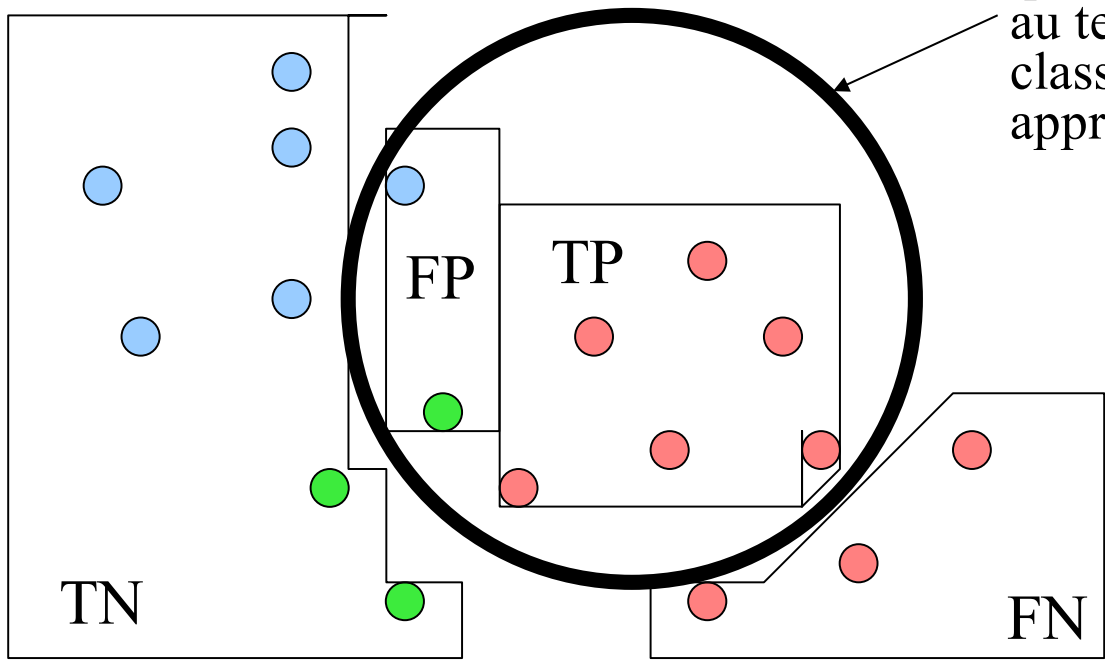


Summary of measures

	Domain	Plot	Explanation
Lift chart	Marketing	TP Subset size	TP $(TP+FP)/$ $(TP+FP+TN+FN)$
ROC curve	Communications	TP rate FP rate	$TP/(TP+FN)$ $FP/(FP+TN)$
Recall- precision curve	Information retrieval	Recall Precision	$TP/(TP+FN)$ $TP/(TP+FP)$

Mesures de la qualité de prédiction d'une classe par rapport à toutes les autres

Ensemble des N éléments déclarés roses en phase de décision (test), c'est-à-dire qui répondent positivement au test (hypothèse) de la classe rose après apprentissage.



Chapitre II. Apprentissage *a posteriori* par Induction

Arbres de décision

Principe : Construire des règles

Si Age > 65 ans
Et Sexe Feminin
Alors Pas d'achat (à 87 %)

...à partir d'un arbre

Algorithme ID3 ou C4.5, Algorithme CART

En théorie,

Principe de diminution du désordre
et lien avec la théorie de l'information

Un pont vers les Systèmes Experts

On veut classer les pièces anciennes et modernes en fonction de

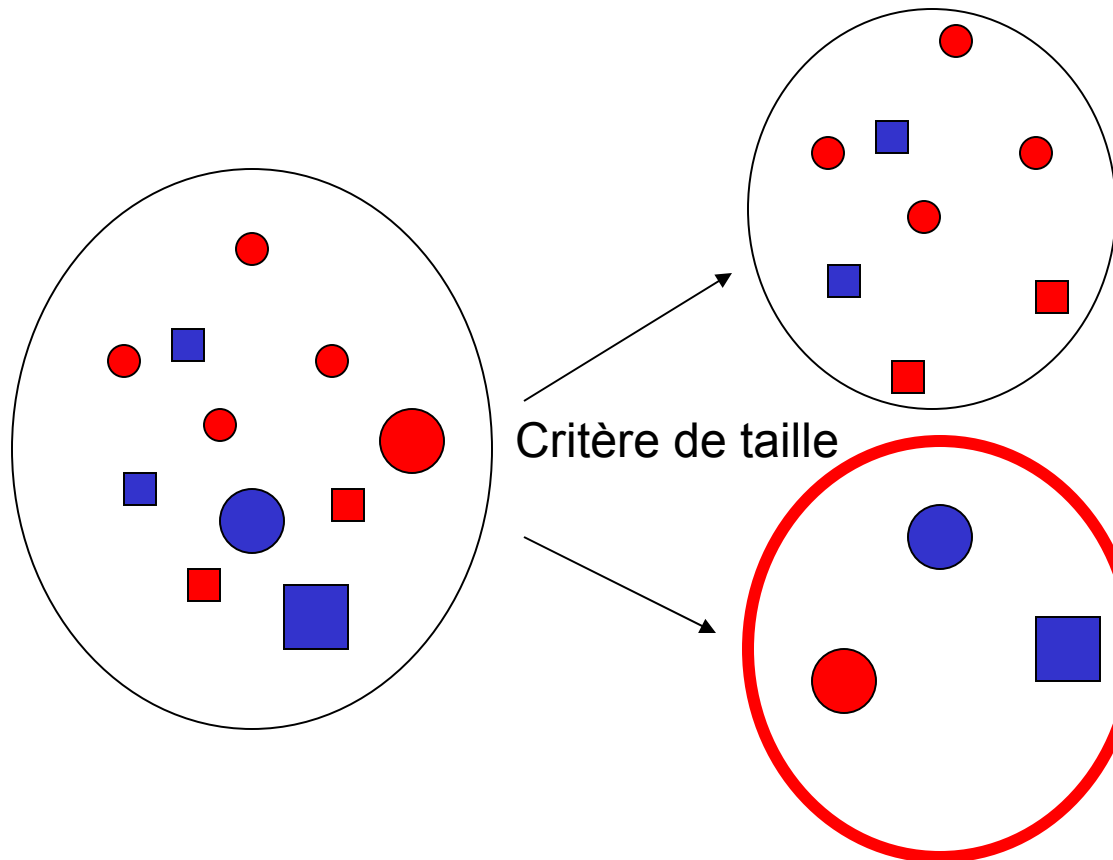
critères explicites et pertinents :



Pièces anciennes



Pièces modernes



Critère de taille

Si on tire au hasard une pièce de ce sac de petites pièces, la probabilité de tomber sur une pièce ancienne est de $1/4$ et moderne de $3/4$

Désordre estimable à $= -1/4\log(1/4)-3/4\log(3/4) = 0,24$

Si on tire au hasard une pièce de ce sac de grandes pièces, la probabilité de tomber sur une pièce ancienne est de $2/3$ et moderne de $1/3$:

Désordre estimable à $= -1/3\log(1/3)-2/3\log(2/3) = 0,27$

Et en prenant le critère de forme ?

Modélisation dans le cas binaire

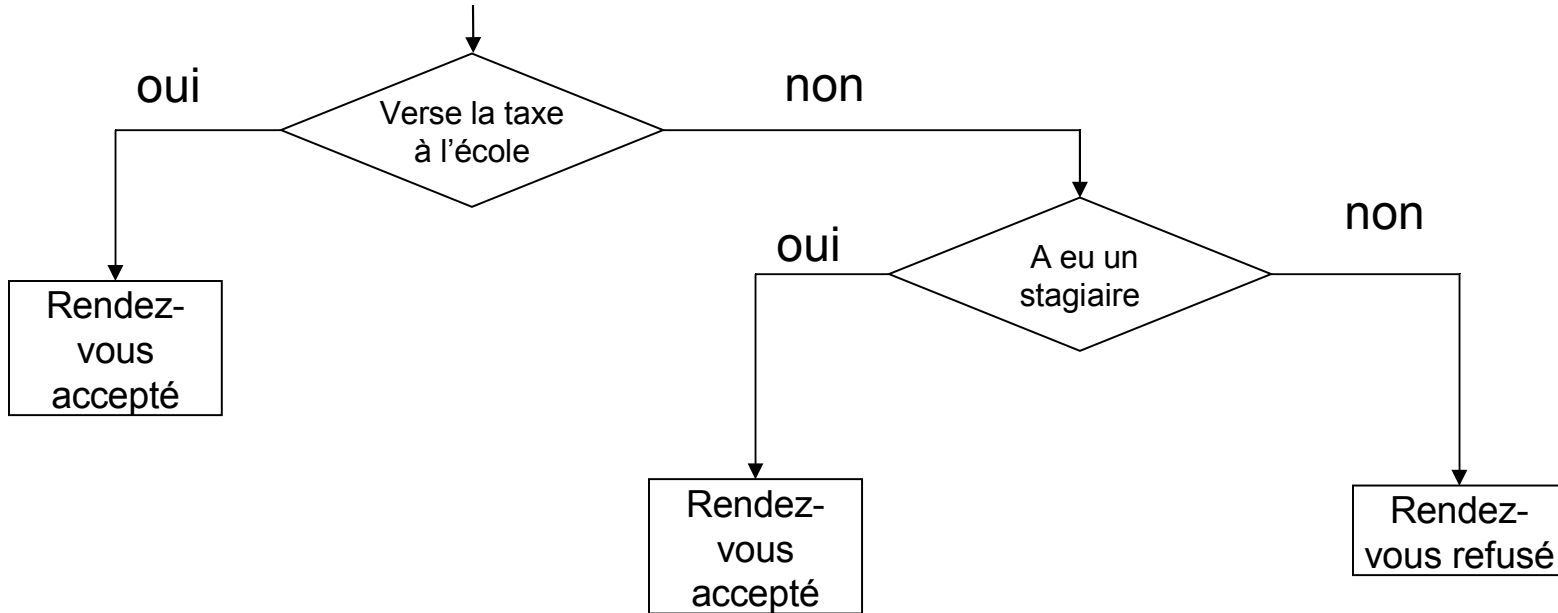
		E1	E2	E3	E4	E5	E6	d_H	P_m
Q1	Connaît l'école	Oui	Oui	Non	Oui	Non	Non	2	2
Q2	A eu un stagiaire	Oui	Non	Non	Non	Non	Non	2	2
Q3	A embauché un étudiant	Non	Oui	Non	Oui	Non	Oui	4	2
Q4	Verse la taxe	Non	Oui	Oui	Non	Non	Non	1	1
Q5	A participé à un évènement	Oui	Oui	Oui	Oui	Oui	Oui	3	3
R	Rendez-vous	Oui	Oui	Oui	Non	Non	Non		

But : Trouver le facteur question Q_i le plus pertinent pour classer les exemples E_i par rapport à une mesure du désordre.

On utilisera par exemple, comme mesure de désordre dans le cas binaire, $P_m(Q_i)$ définie à partir de la distance de Hamming $d_H(R, Q_i)$:

$$P_m : \text{Pseudo-Métrique de Hamming} \\ = \\ \min(\text{nb_exemples} - d_H, d_H)$$

Arbre de décision correspondant :



Simplicité

Lisibilité

Multiplicité des arbres possible

Taille de l'arbre libre

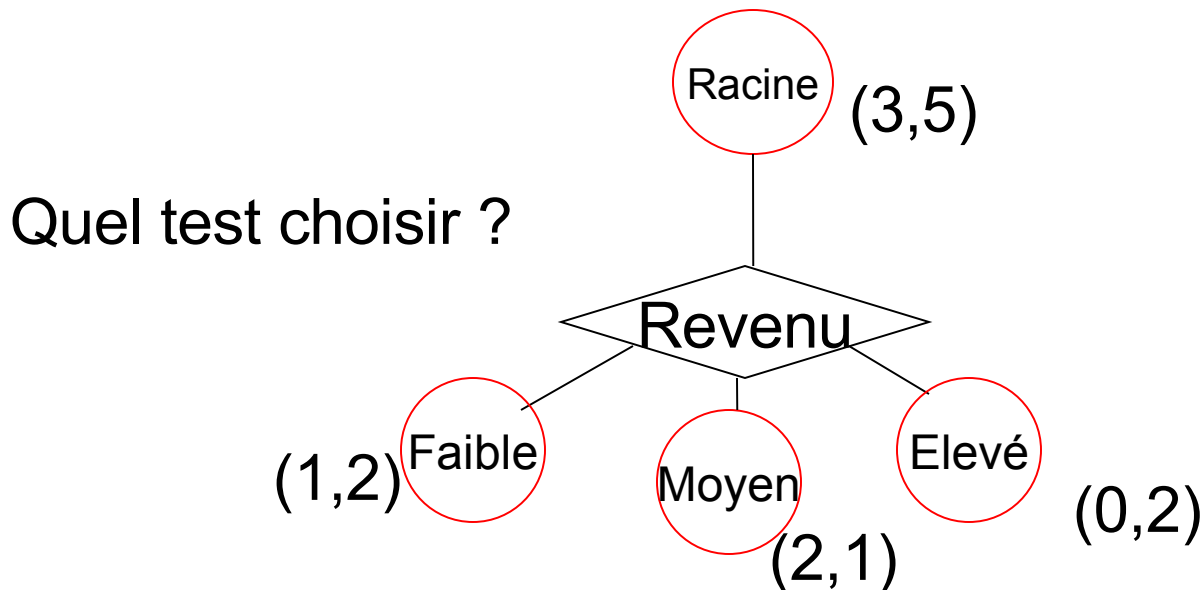
Mais pour des Variables qualitatives, comment faire ?

client	<i>Revenu</i>	<i>Age</i>	<i>Résidence</i>	<i>Etudes</i>	<i>Internet</i>
1	moyen	moyen	village	oui	oui
2	élevé	moyen	bourg	non	non
3	faible	âgé	bourg	non	non
4	faible	moyen	bourg	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevé	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non

Pour chaque nœud, on répartit les différentes classes.

Un nœud est terminal s'il ne contient que des individus appartenant à une seule classe (en l'occurrence ici, qui consulte ses comptes sur internet).

Ainsi le nœud racine de notre arbre n'est pas terminal.



- Entropie (noeud p) = $-\sum_{k=1}^C p(k/p) \log p(k/p)$

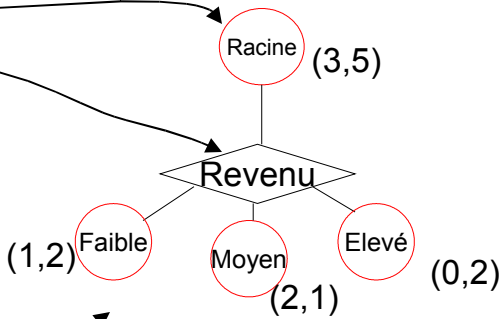
Avec $p(k/p) = N(k/p) / N(p)$
 = proportion d'éléments de classe k à la position p

Entropie(Racine) = $-3/8 \log(3/8) - 5/8 \log(5/8) = 0.954$
 Entropie(Etudes/oui) = $-3/5 \log(3/5) - 2/5 \log(2/5) = 0.970$
 Entropie(Etudes/non) = 0

Ainsi le gain au niveau du désordre entre le noeud courant **p** de l'arbre et l'embranchement de test choisi t vaut :

- Gain(**p**,t) = Entropie(**p**) - $\sum_{k=1}^n P_j Entropie(p_j)$

Avec
 $p_j = j^{\text{ème}}$ noeud créé
 et $P_j =$ proportion d'éléments pour ce noeud

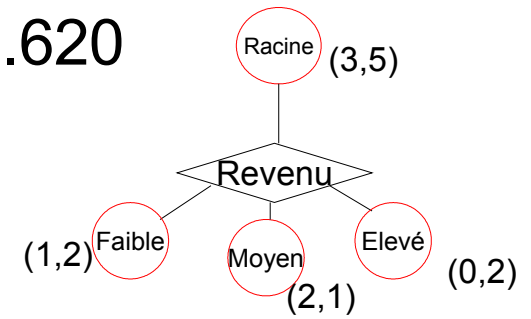


$$\text{Gain}(\text{Racine}, \text{Revenu}) = \text{Entropie}(\text{Racine}) - 0.620$$

$$= \text{Entropie}(\text{Racine}) - \frac{3}{8}[-\frac{1}{3}\log(\frac{1}{3}) - \frac{2}{3}\log(\frac{2}{3})]$$

$$- \frac{3}{8}[-\frac{2}{3}\log(\frac{2}{3}) - \frac{1}{3}\log(\frac{1}{3})]$$

$$- \frac{2}{8}[-\frac{0}{2}\log(\frac{0}{2}) - \frac{2}{2}\log(\frac{2}{2})]$$

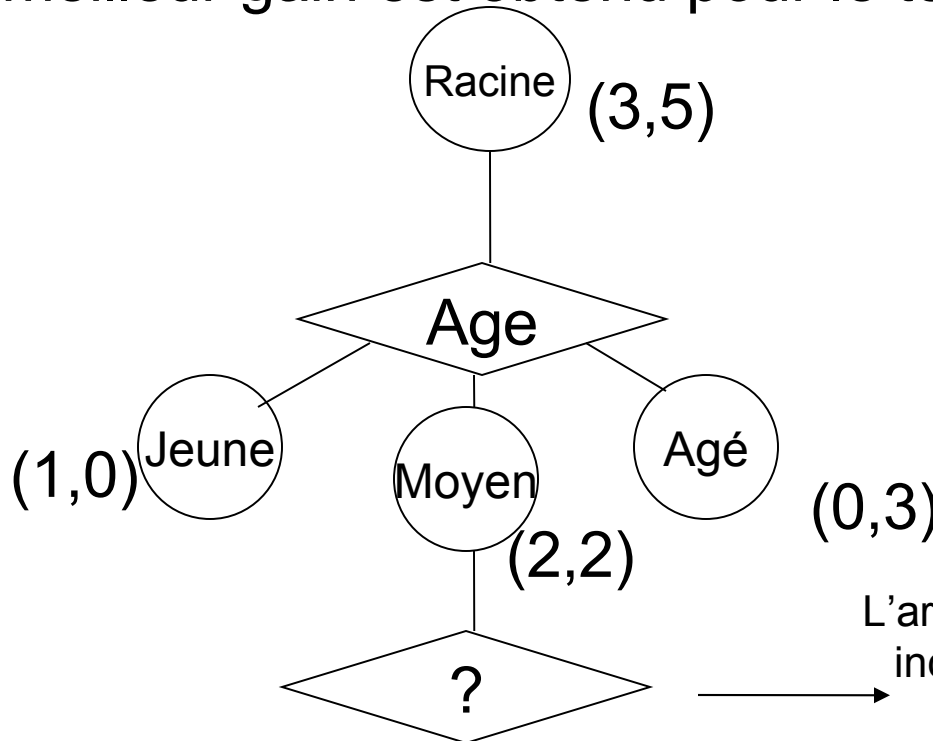


$$\text{Gain}(\text{Racine}, \text{Age}) = \text{Entropie}(\text{Racine}) - 0.5$$

$$\text{Gain}(\text{Racine}, \text{Etudes}) = \text{Entropie}(\text{Racine}) - 0.607$$

$$\text{Gain}(\text{Racine}, \text{Résidence}) = \text{Entropie}(\text{Racine}) - 0.870$$

Le meilleur gain est obtenu pour le test sur l'âge



L'arbre se construit ainsi
incrémentalement en
poursuivant

Variables quantitatives : méthode des grappes

client	<i>Revenu</i>	<i>Age</i>	<i>Résidence</i>	<i>Etudes</i>	<i>Internet</i>
1	2000	30	1000	4	oui
2	4500	34	500	0	non
3	600	56	300	1	non
4	1000	41	350	3	oui
5	1500	20	10000	2	oui
6	3000	65	5000	3	non
7	1800	59	4000	3	non
8	600	32	5600	1	non

Toujours la même chose : compromis entre généralisation et complexité :
-> Elagage *a posteriori* d'un arbre de décision par algorithme *glouton*
(équivalent du *pruning* pour le réseau neuronal)

Algorithme 11.2 Elagage d'un arbre de décision

Procédure : élaguer(T_{max})

$k \leftarrow 0$

$T_k \leftarrow T_{max}$

tant que T_k a plus d'un nœud **faire**

pour chaque nœud ν de T_k **faire**

 calculer le critère $\varpi(T_k, \nu)$ sur l'ensemble d'apprentissage

fin pour

 choisir le nœud ν_m pour lequel le critère est maximum

T_{k+1} se déduit de T_k en y remplaçant ν_m par une feuille

$k \leftarrow k + 1$

fin tant que

Dans l'ensemble des arbres $\{T_{max}, T_1, \dots, T_k, \dots, T_n\}$, choisir celui qui a la plus petite erreur de classification sur l'ensemble de validation.

$$\varpi(T_k, \nu) = \frac{MC_{\text{éla}}(\nu, k) - MC(\nu, k)}{n(k) \cdot (nt(\nu, k) - 1)} \quad (11.8)$$

- $MC_{\text{éla}}(\nu, k)$ est le nombre d'exemples de l'ensemble d'apprentissage mal classés par le nœud ν de T_k dans l'arbre élagué à ν .
- $MC(\nu, k)$ est le nombre d'exemples de l'ensemble d'apprentissage mal classés sous le nœud ν dans l'arbre non élagué
- $n(k)$ est le nombre de feuilles de T_k
- $nt(\nu, k)$ est le nombre de feuilles du sous-arbre de T_k situé sous le nœud ν .

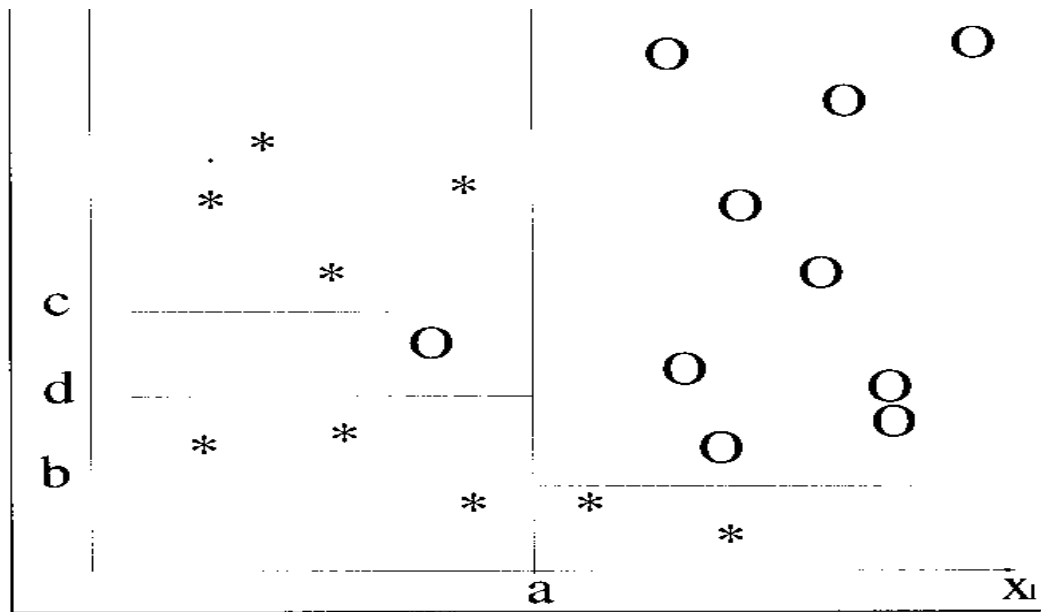


FIG. 11.3 - L'arbre de décision géométrique.

Partie élaguée
après l'algo

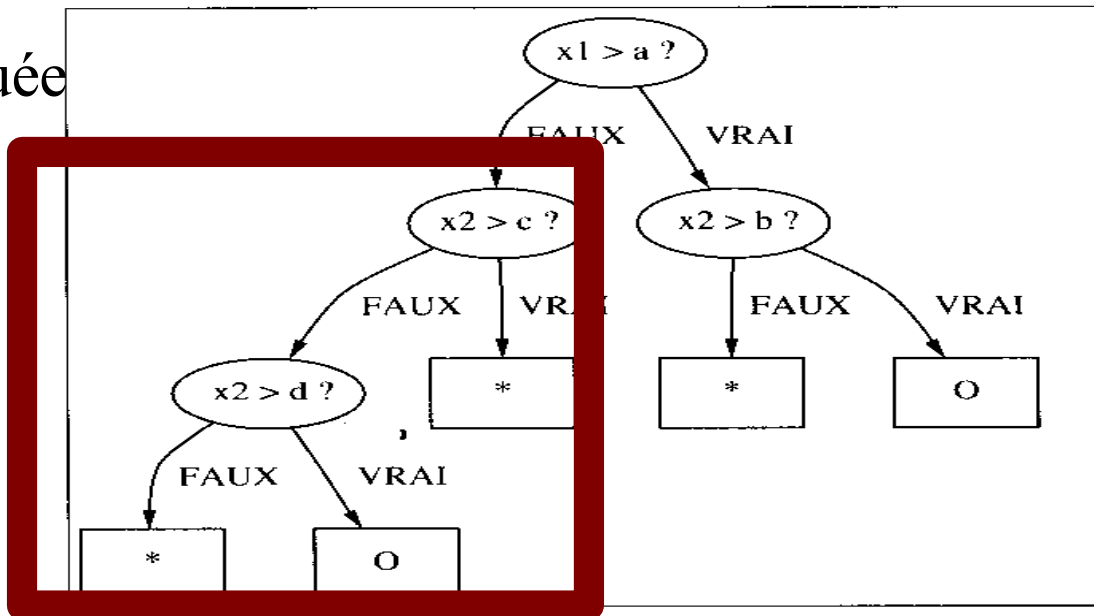


FIG. 11.4 - L'arbre de décision logique T_{max} .

Boosting ou dopage : théorème du classifieur faible

ADABOOST Algorithm ou la combinaisons des classifieurs faibles

Technique du Bagging

Nicolas Loménie

Chapitre III. Algorithmes
évolutionnaires

Algorithmes évolutionnaires

Le problème de l'optimisation en général

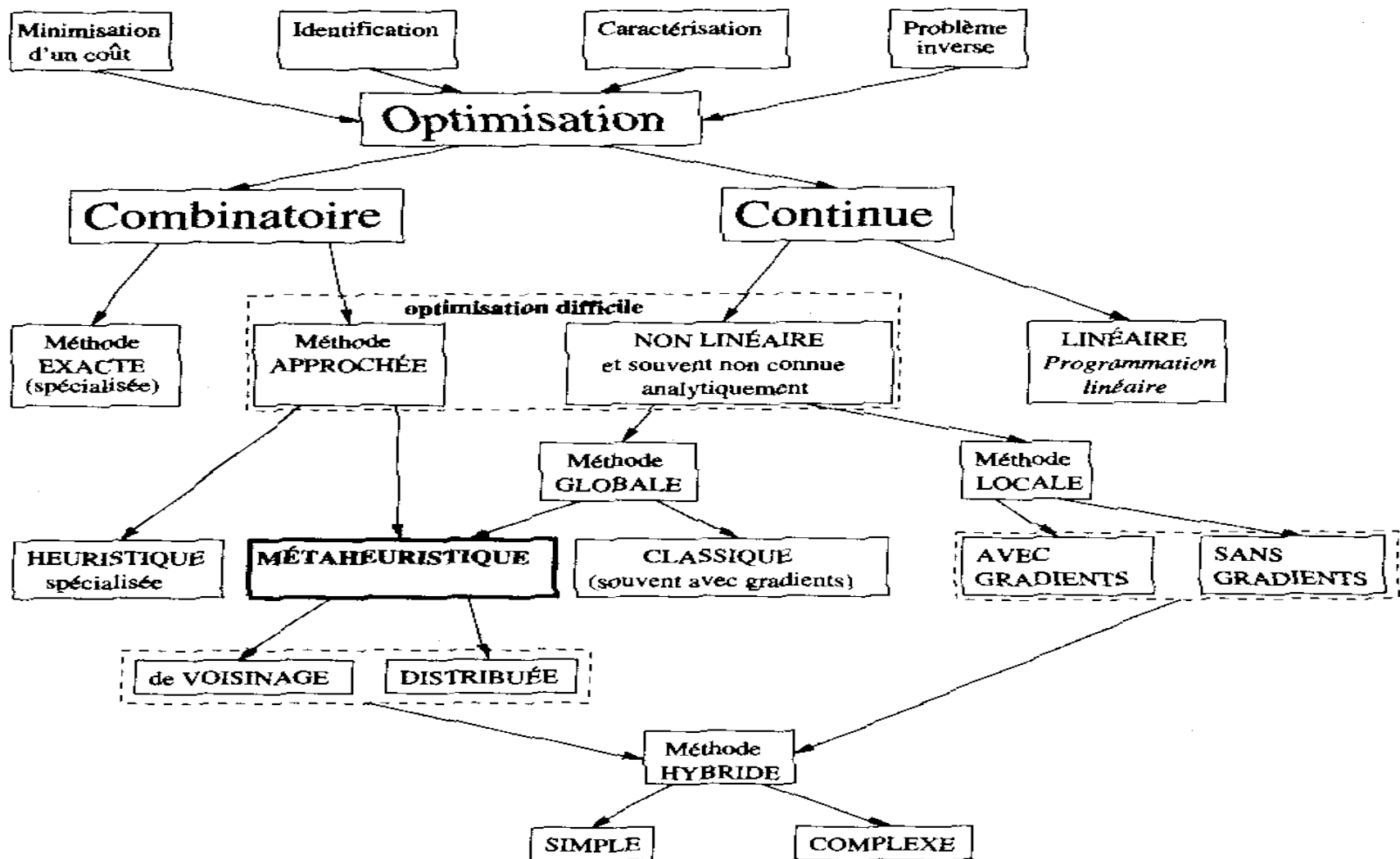
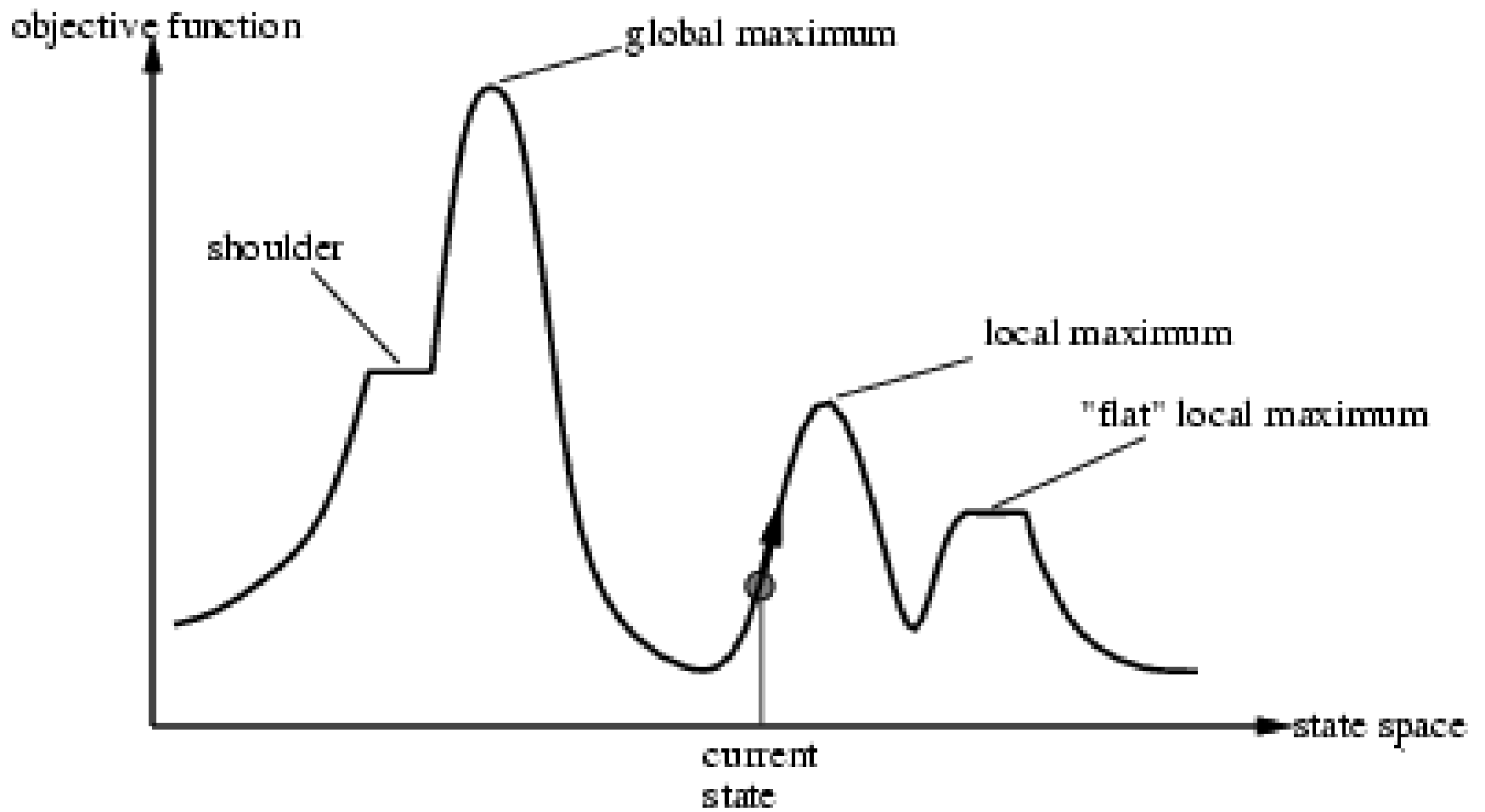


FIG. 11 – Classification générale des méthodes d'optimisation mono-objectif. 184



- Optimiser une fonction de coût ou d'erreur $J(W)$: maximiser ou minimiser !!
- Descente de gradient (voir Réseau de Neurones), qui donne vectoriellement :
$$W^{t+1} = W^t - c * \text{Gradient}_W(J)$$

Pour sortir de minima locaux, on peut par exemple faire varier c avec les itérations : un des principes du “recuit simulé” ‘simulated annealing”

$$W^{t+1} = W^t + c(t) * \Delta W$$

En général, le calcul du gradient n'est pas toujours possible facilement notamment quand la fonction de coût n'est pas la classique erreur aux moindres carrés -> solution par processus stochastique type Monte Carlo.

Enfin, l'espoir d'arriver en un temps raisonnable à un minimum correct s'efface quand le nombre de ces minima locaux augmente.

Simulated annealing search

- Idea: escape local maxima by allowing some "bad" moves but **gradually decrease** their frequency

```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
  inputs: problem, a problem
           schedule, a mapping from time to "temperature"
  local variables: current, a node
                   next, a node
                   T, a "temperature" controlling prob. of downward steps

  current ← MAKE-NODE(INITIAL-STATE[problem])
  for t ← 1 to ∞ do
    T ← schedule[t]
    if T = 0 then return current
    next ← a randomly selected successor of current
    ΔE ← VALUE[next] - VALUE[current]
    if ΔE > 0 then current ← next
    else current ← next only with probability  $e^{\Delta E/T}$ 
```

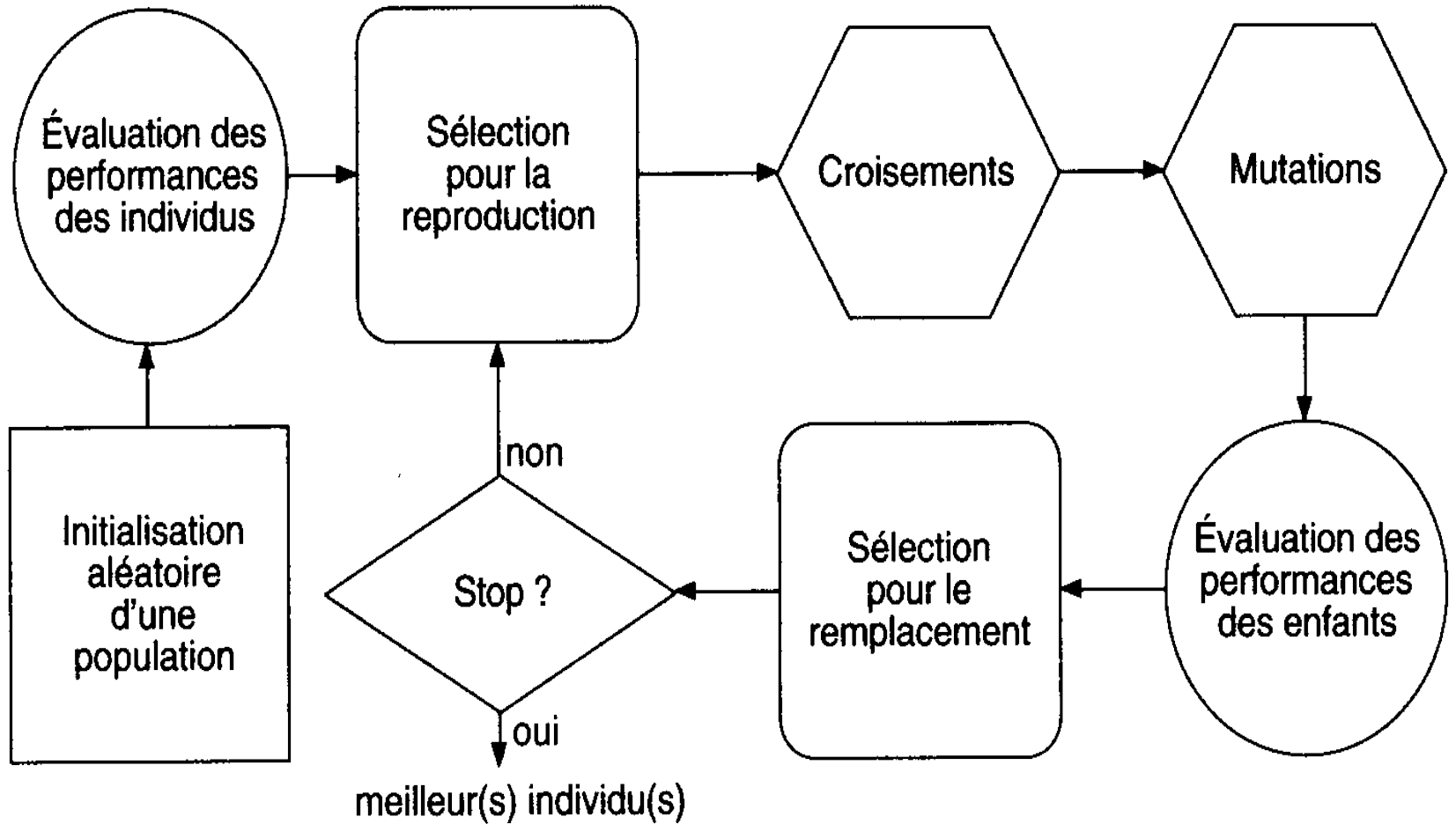


FIG. 9 – Principe d'un algorithme évolutionnaire.

Les Algorithmes génétiques

L'évolution dans la nature survient quand des entités ont la capacité de se **reproduire**, qu'il existe une **population** de ces entités, qu'il existe une **variété** (diversité) à travers ces entités, et que la **survie** des entités dépend des différences entre elles. Toute entité vivante possède un **génotype** et le **phénotype**

Le génotype.

Le génotype est constitué de gènes situés sur des chromosomes stockés dans le noyau des cellules sous la forme d'une longue chaîne d'acide déoxyribonucléique (ADN). Dans la nature, l'ADN est un polymère constitué par l'enchaînement de quatre molécules, les nucléotides adénine (A), cytosime (C), guanine (G) et la thymine (T). On peut donc décrire l'ADN par des chaînes de quatre caractères ACGT. L'ADN constitue l'ensemble des chromosomes, ou le génome d'un individu.

Exemple : la suite **ACCTGAGGGTA**

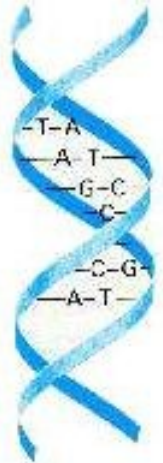
Informatique : le codage binaire d'une solution

Le phénotype.

Le phénotype est l'ensemble des protéines et des enzymes qui peuvent être fabriqués à partir de l'ADN. En fait, l'ADN est copiée par un messenger (ARN) qui au niveau du ribosome, se traduit en chaînes d'acides aminés formant les protéines et les enzymes. En général, on compte une protéine (un enzyme) par gène. Ce sont les protéines et les enzymes qui dictent la structure et le comportement des cellules qui définissent les caractéristiques physiques d'un individu et permettent à un individu de réaliser des tâches dans son environnement, de survivre et de se reproduire à des taux différents. **Ensemble des manifestations observables du génotype.**

Exemple : le gène **AGTAGT** code les **yeux verts**.

Informatique : une solution du problème dans une représentation « naturelle » obtenue après décodage du génotype



La reproduction se traduit par la transmission du génome aux individus de la progéniture ce qui permet de préserver les gènes menant à des performances supérieures. Occasionnellement, un processus naturel, la **mutation** génétique, introduit une variation dans les chromosomes.

Or les individus les mieux adaptés, c'est-à-dire capables de mieux effectuer les tâches nécessaires à leur survie, se reproduisent à des taux les plus élevés, alors que les individus les moins adaptés se reproduisent à des taux plus faibles. Ce sont les **principes de survie et reproduction décrit par Charles Darwin** dans « On the Origin of Species By Means of Natural Selection » en 1859. Il s'avère alors qu'une population ayant une grande variété va, de génération en génération, contenir des individus dont le génotype se traduit par une meilleure adaptation, et ceci à cause de la contrainte de la sélection naturelle.

L'algorithme génétique ne fait que transposer ce que fait la nature à des systèmes artificiels. Il simule les processus évolutifs Darwiniens et génétiques s'appliquant aux chromosomes. Il transforme une population d'individus souvent représentés par des chaînes de caractères pour imiter les chaînes d'ADN, chacun ayant une valeur d'adaptation, en une nouvelle population. L'algorithme fait donc appel à quatre opérateurs de base :

- **l'évaluation** du niveau d'adaptation d'un individu.
- **la sélection** : c'est le choix des individus en fonction du niveau d'adaptation.
- **le croisement** : c'est le mélange des bagages génétiques et c'est l'opérateur de recherche essentiel.
- **la mutation** : le bagage génétique est modifié abruptement (utilisé à un faible taux et pour assurer un certain degré de diversité dans la population).

Auxquels il faut ajouter le **codec génotype<->phénotype** approprié au problème.

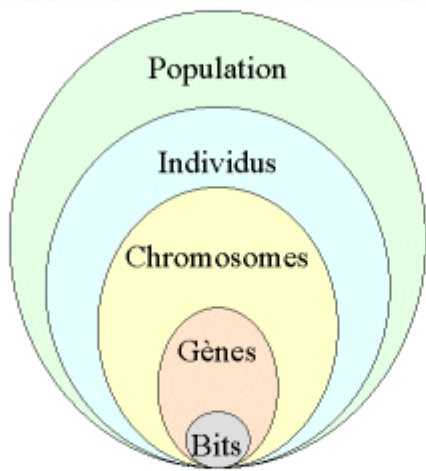


Figure 3 : les cinq niveaux d'organisation de notre Algorithme Génétique.

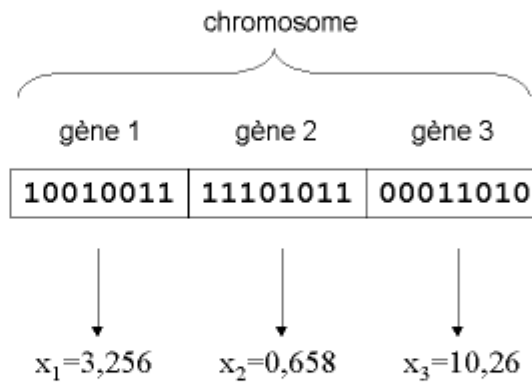


Figure 4 : illustration schématique du codage des variables d'optimisation x_i .

Afin de coder nos variables réelles en binaire, nous discrétisons l'espace de recherche. Ainsi un codage sur 32 bits implique une discrétisation des intervalles en $g_{max} = 2^{32} - 1 = 4\,294\,967\,295$ valeurs discrètes. Notons au passage que si cette discrétisation est plus fine que celle du modèle physique utilisé, la fonction est assimilable à une fonction escalier, si on la considère à une échelle suffisamment petite.

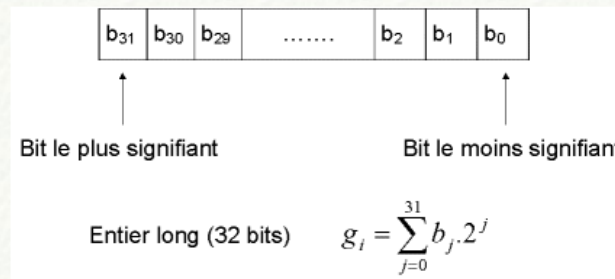


Figure 5 : chaque gène (chaque paramètre du dispositif) est codé par un entier long (32 bits).

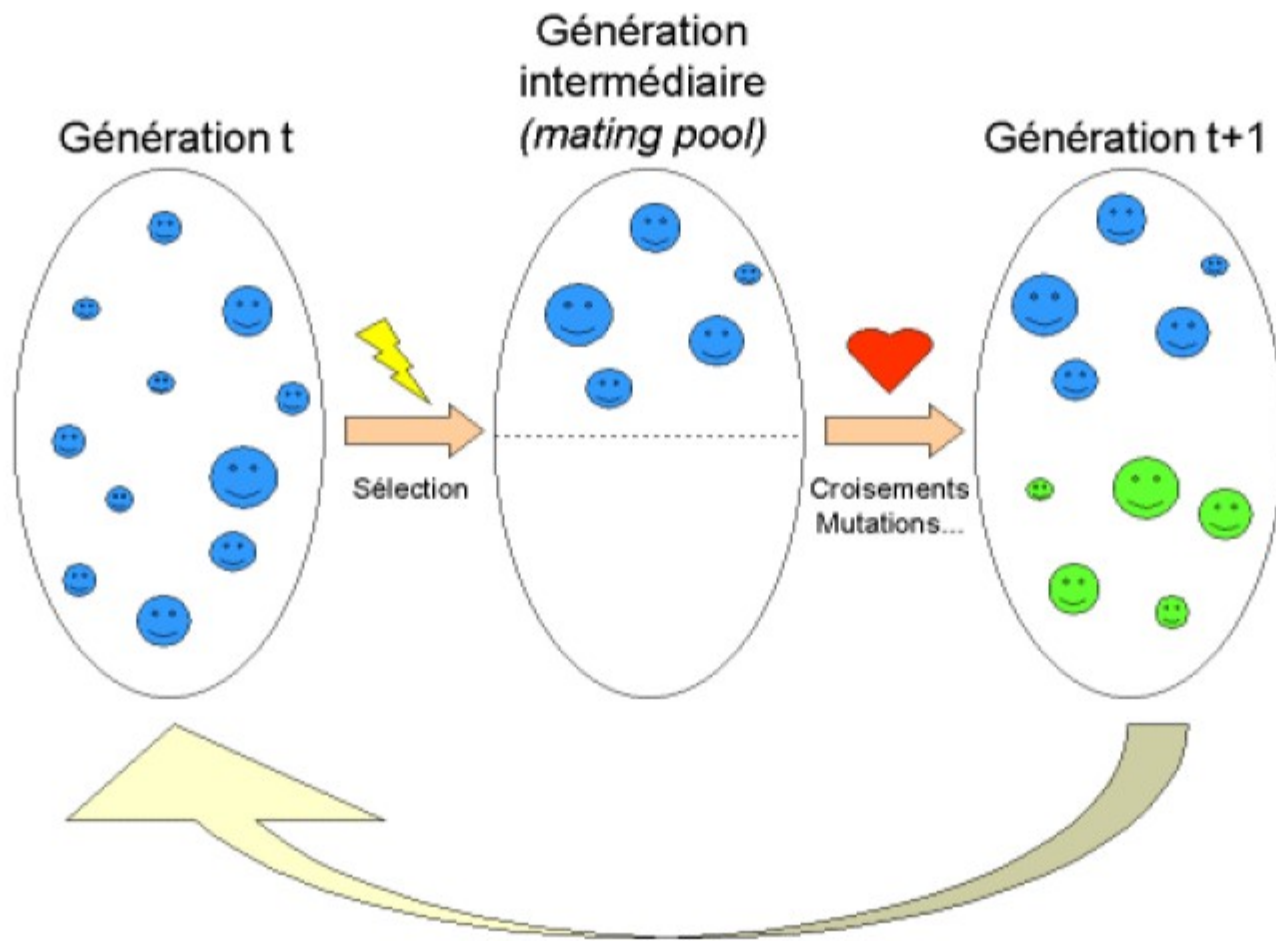
A chaque variable réelle x_i on associe donc un entier long g_i :

$$0 \leq g_i \leq g_{max} \quad \forall i \in [1, n]$$

Les formules de codage et décodage sont alors les suivantes :

$$g_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \cdot g_{max}$$

$$x_i = x_{min} + (x_{max} - x_{min}) \cdot \frac{g_i}{g_{max}} \quad (5)$$



Le passage d'une génération à l'autre se fait par sélection des individus les plus adaptés, puis par reproduction.

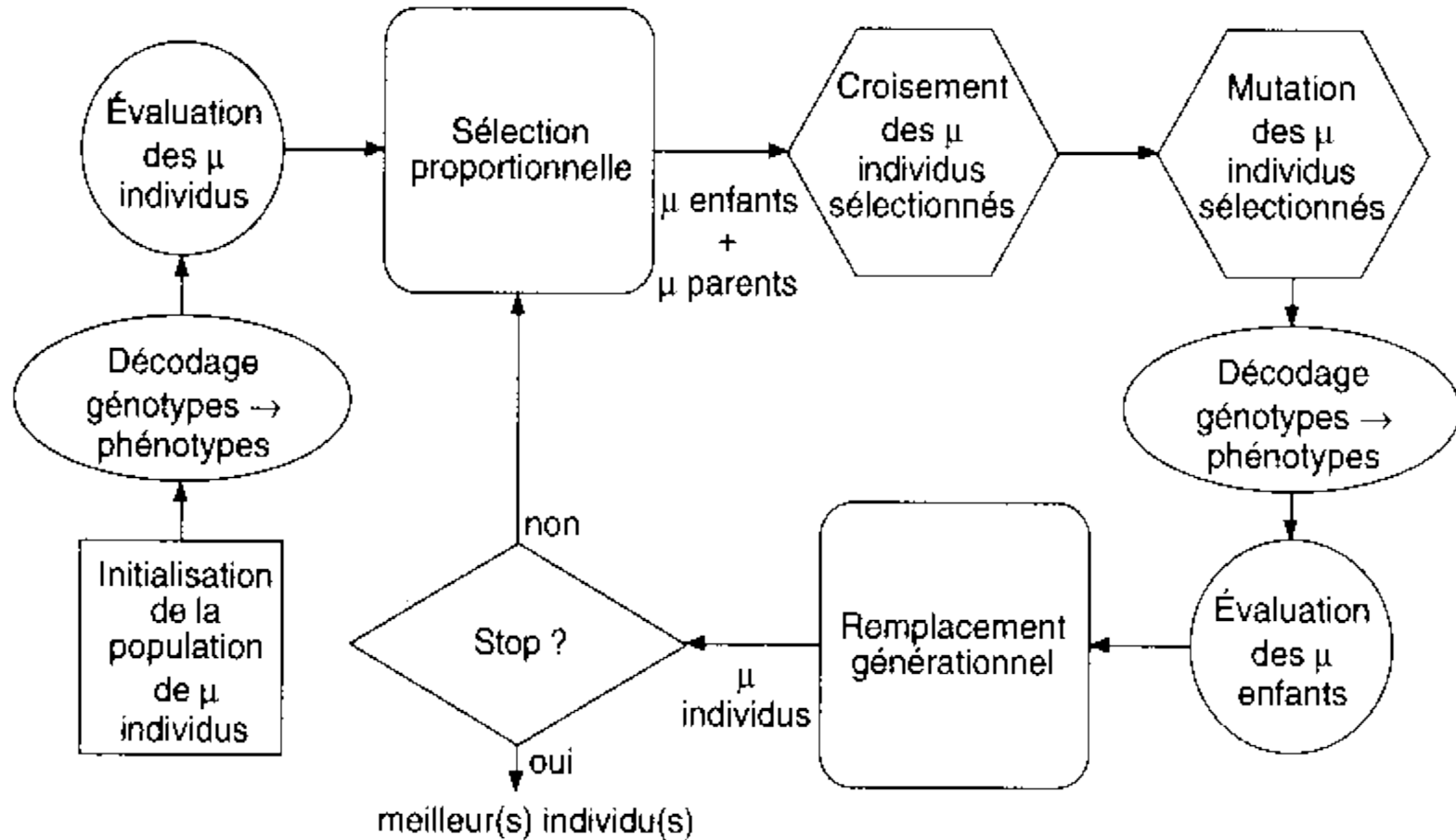


FIG. 3.22 – Un algorithme génétique simple.

Étape 1 : *évaluation des individus*

- ▷ fonction d'objectif d'un individu i : f_i

Dépend du problème : valeur de la fonction à maximiser, coût d'une tournée, poids d'une affectation, temps global d'une exécution, etc.

- ▷ fonction d'adéquation de l'individu (fitness)

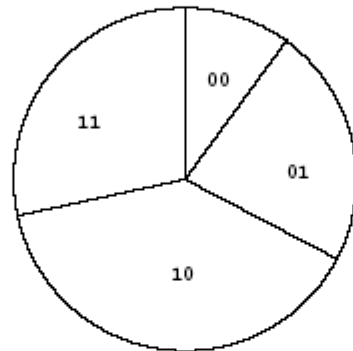
$$\frac{f_i}{\text{moyenne des } f}$$

Étape 2: sélection

Les individus sont retenus de manière probabiliste en fonction de leur adéquation.

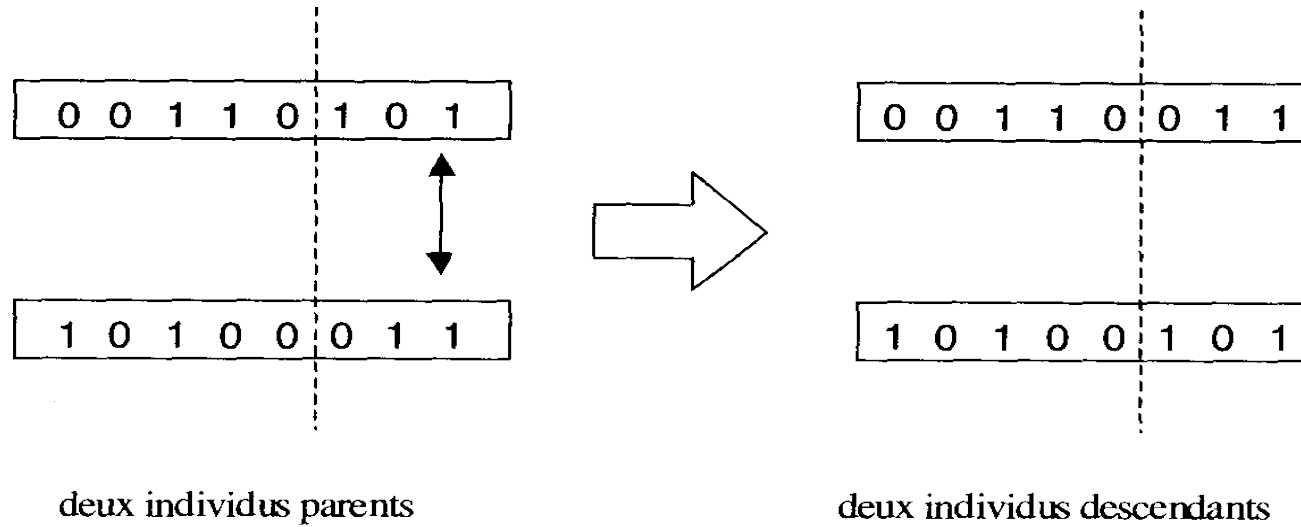
individus	0 0	0 1	10	11
adéquations	0.3	0.8	1.7	1.2

Sélection simple

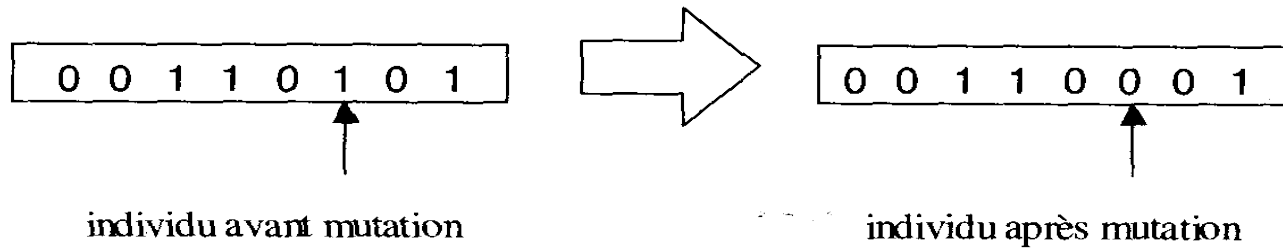


- ▷ On tourne la roue autant de fois que l'on veut d'individus.
- ▷ À chaque tirage, un individu est sélectionné avec une probabilité proportionnelle à son adéquation.

Etape 3 : Création de la génération t+1

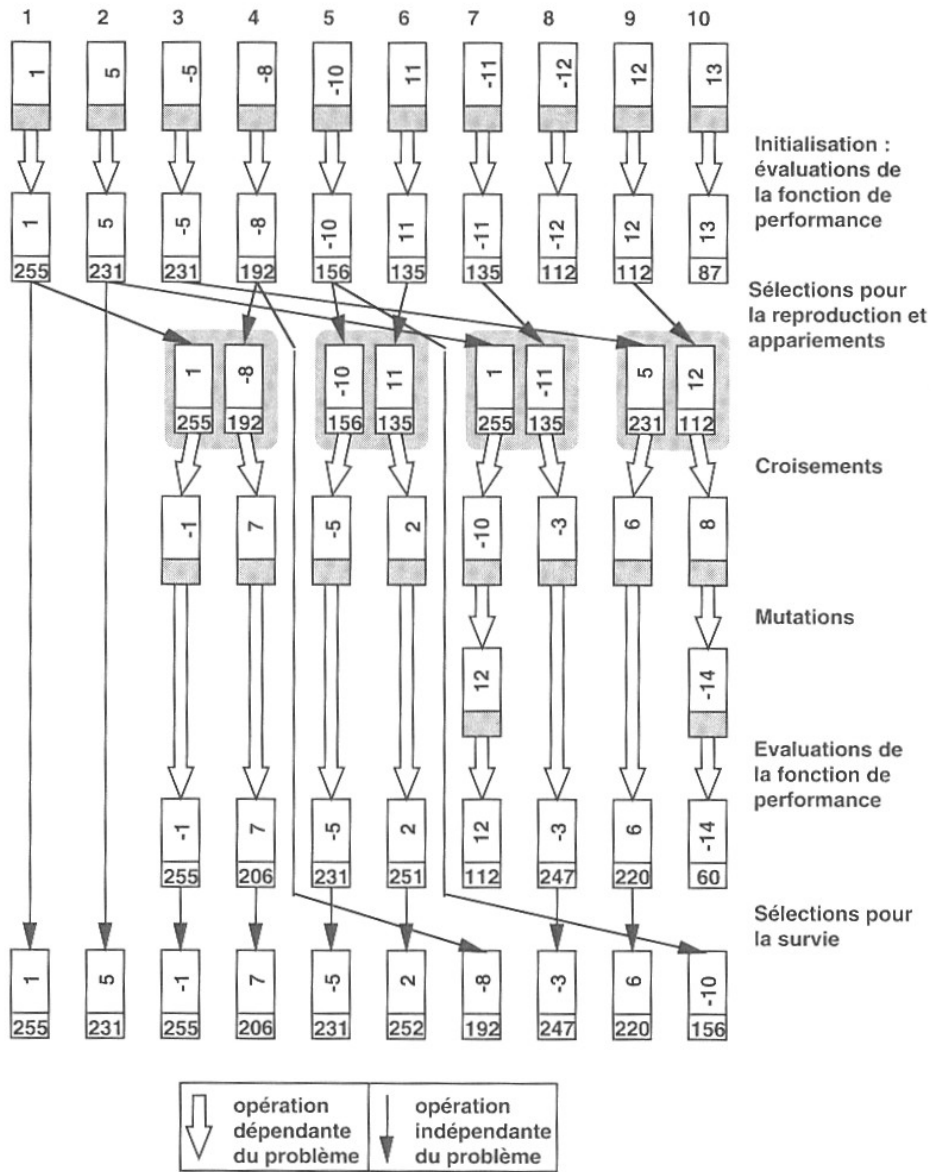


a – Croisement (mono-point).



b – Mutation (d'un seul bit).

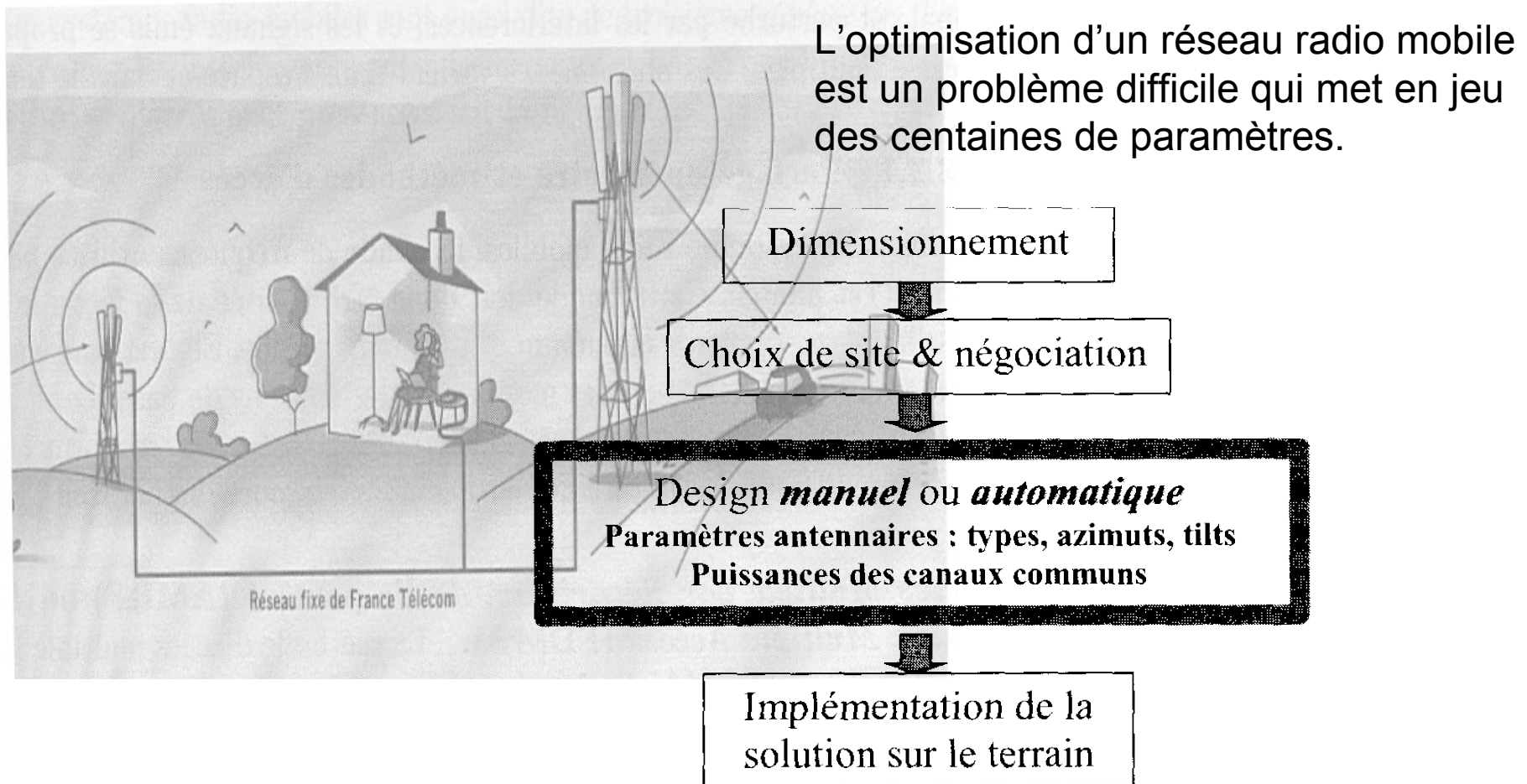
FIG. 8 – Exemples d'opérateurs de croisement et de mutation, dans le cas d'individus représentés par des chaînes de 8 nombres binaires.



Maximisation de $C(x) = 256 - x^2$

. 3.2 – Application d'un algorithme évolutionnaire sur une population de $\mu = 10$ parents e 8 enfants.

Optimisation de réseaux UMTS



. 8.10 – Étapes de la planification d'un réseau radio mobile.

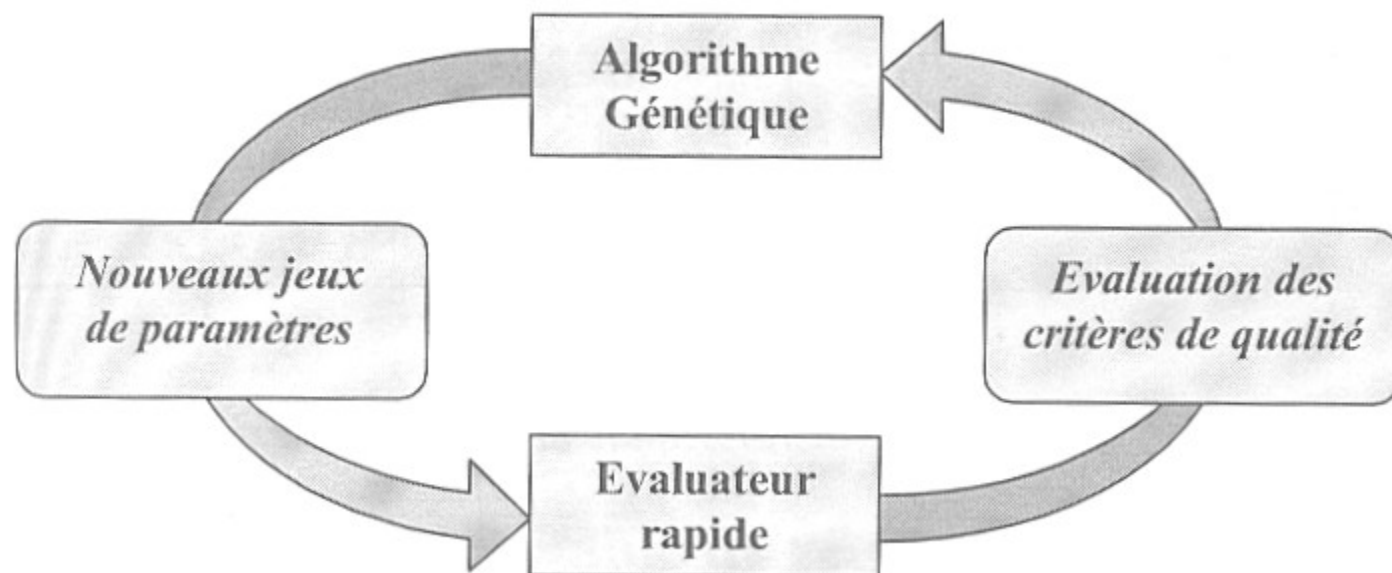


FIG. 8.13 – Schéma de principe de l'outil de planification automatique.

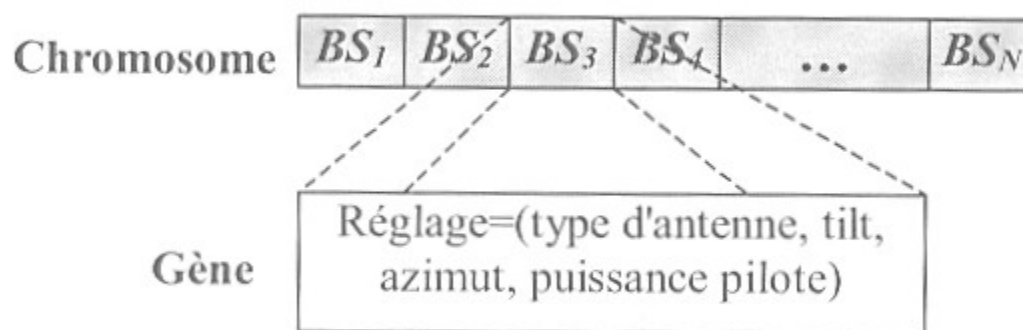


FIG. 8.14 – Codage des différents paramétrages du réseau.

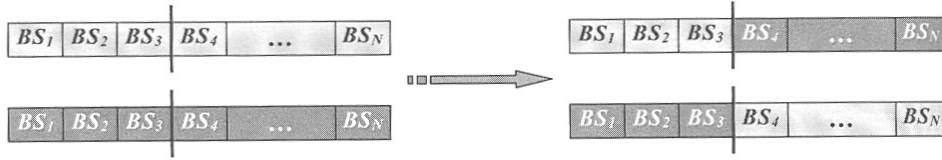


FIG. 8.15 – Croisement en un point.

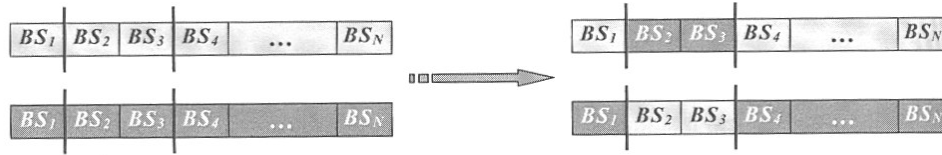


FIG. 8.16 – Croisement en deux points.

Opérateurs de variation

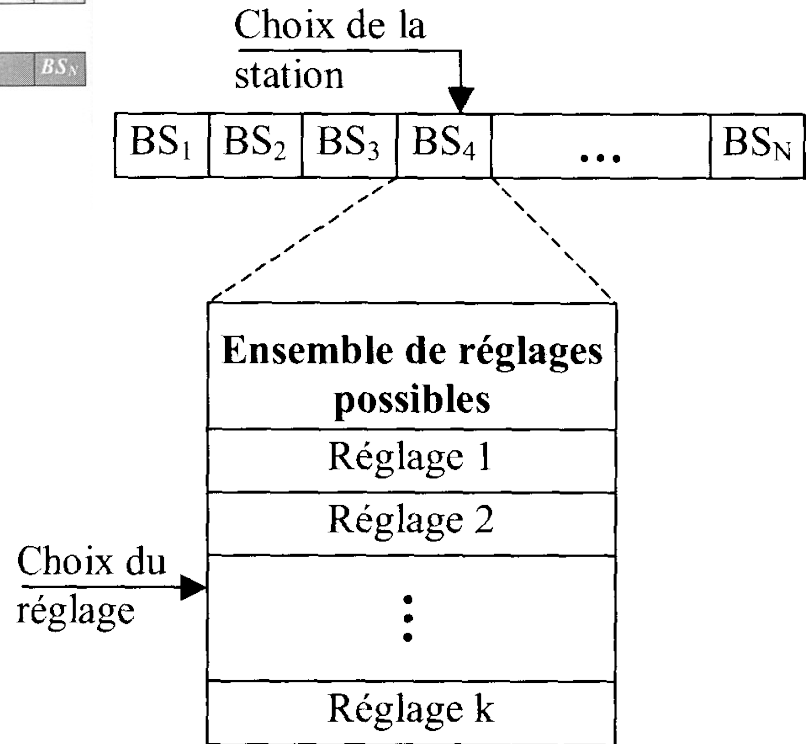
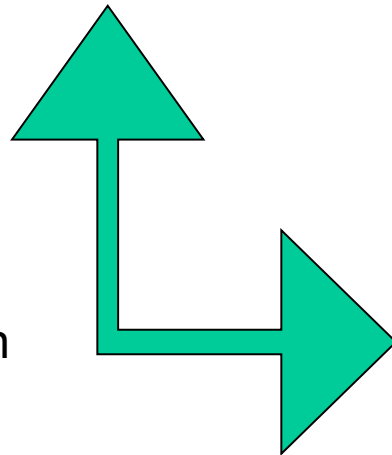


FIG. 8.18 – L'opérateur de mutation agit en deux temps.

Une fonction de coût agrégeant les objectifs pertinents est utilisée :

- Couverture
- Capacité
- Continuité et qualité de service
- Coût d'implémentation

Un algorithme génétique guide une génération de réseaux vers une solution globale correspondante à un optimum global ou à un bon optimum local.

A chaque itération, un évaluateur rapide de réseau UMTS calcule les différents critères de qualité qui permettent de “noter” les paramétrages des réseaux proposés.

Ensuite, l'AG utilise ces “notes” et propose un nouveau jeu de paramètres pour le réseau.

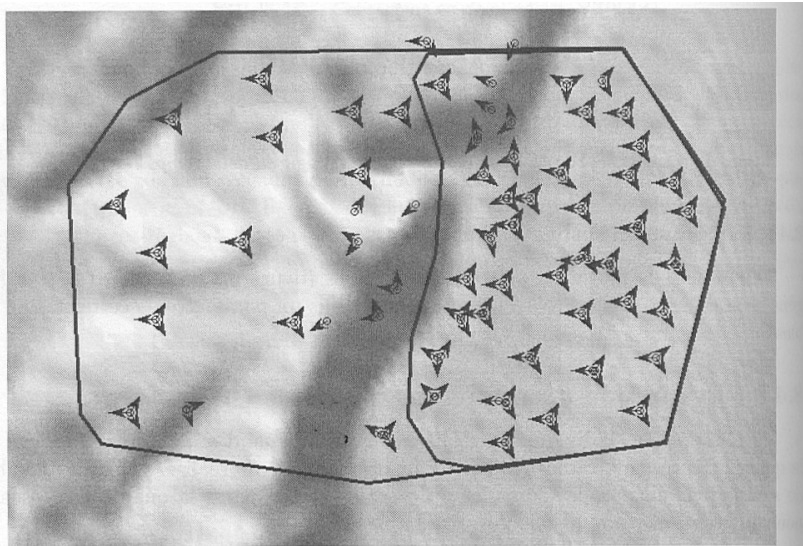
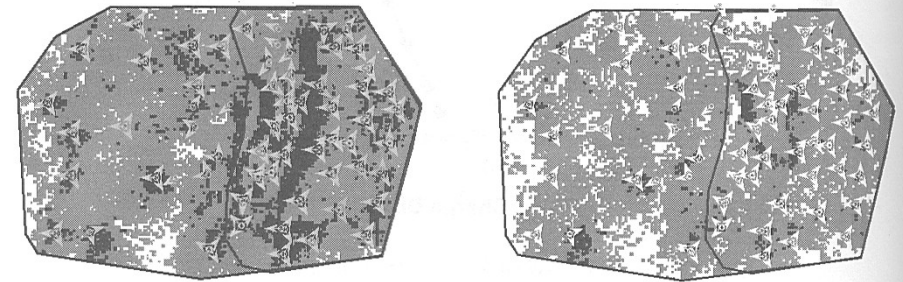
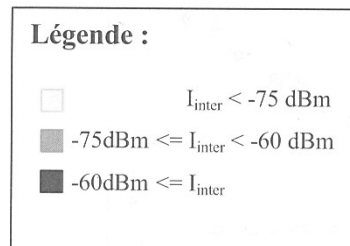


FIG. 8.19 – Réseau UMTS avec 172 stations de base dans un environnement hétérogène, urbain et dense urbain. La plupart des sites sont tri-sectoriels (trois stations de base sont installées sur le même site).

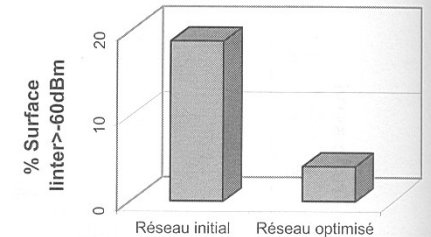


(a)

(b)



(c)



(d)

FIG. 8.25 – Interférences inter-cellulaires du réseau initial (a) et du réseau optimisé (b). La légende est présentée sur (c) et la comparaison entre les surfaces hautement brouillées avec $I_{inter} \geq -60 \text{ dBm}$ sur (d).

Convergence et temps de calcul

On peut constater figure 11 que l'amélioration de la population est très rapide au début (*recherche globale*) et devient de plus en plus lente à mesure que le temps passe (*recherche locale*). Le bruit dans la moyenne est essentiellement dû aux mutations.

On voit que la valeur moyenne de la fonction d'adaptation a tendance à se rapprocher de celle de l'individu le plus adapté. Cela correspond à une uniformisation croissante de la population. Nous avons donc introduit dans notre logiciel une fenêtre graphique (Figure 12) permettant de visualiser la totalité de la population [Dessales, 1996]. Chaque ligne représente le génotype d'un individu, autrement dit les bits qui conduisent aux valeurs des paramètres d'un composant. Chaque pixel représente la valeur d'un bit dans son chromosome principal (blanc pour 0, et noir pour 1). Chaque groupe de 32 bits, entre deux graduations de la Figure 12, correspond à un gène. Les individus sont triés selon leur fonction d'adaptation : les plus adaptés correspondent aux lignes du haut, les moins adaptés aux dernières lignes. Dans cet exemple, nous avons représenté une population aux générations 1, 3, 10, 20, 50 et 130 (de gauche à droite, et de haut en bas). La taille de la population est de 200 individus (dispositifs) et le chromosome principal comprend 8 gènes (paramètres) de 32 bits.

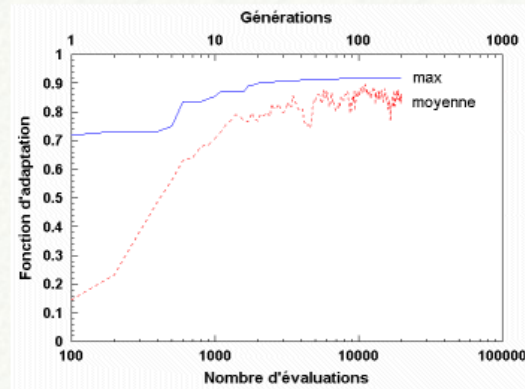


Figure 11 : exemple de convergence de l'AG. On a reporté la valeur de la fonction d'adaptation de l'individu le plus adapté de chaque génération (trait), et la moyenne des fonctions d'adaptation (pointillés), pour une population de 200 individus.

Pourquoi ça marche ?

Tentative d'explication

- ▷ Le fait de travailler à partir d'une *population*, et non d'un seul individu, et de disposer de l'opérateur de mutation permet d'éviter de s'enfermer dans un optimum local.

De la rencontre fortuite de deux bonnes idées peut naître une excellente bonne idée.

- ▷ Les individus les mieux adaptés ont des "points communs": de courts motifs, les **briques élémentaires**.

****101***, *****10**

- ▷ La reproduction par croisement permet de propager ces briques élémentaires, et même de les multiplier.

****101*10**

Mise en pratique

- ▷ "kit heuristique" universel

Peut s'appliquer à tous les problèmes d'optimisation, y compris les problèmes NP-complets, sans analyse particulière du problème

- ▷ pas de garantie

Aucune prédiction sur le comportement de l'algorithme ou sur la qualité du résultat n'est possible.

- ▷ souvent très lent

Il ne faut pas avoir recours à un algorithme génétique si une méthode exacte efficace existe

- ▷ l'approche brute est souvent peu fructueuse

Ajouter des opérateurs spécifiques, changer la représentation des données

⇒ **Algorithmes évolutionnaires, hybrides**

Chapitre IV. Systèmes et Processus de Data Mining

Exemples d'applications

Comprendre les facteurs explicatifs d'un niveau d'appel.

Variable expliquée : durée mensuelle de consommation.

La base de données :

Département
Type de client
Profession
Revenu
Situation matrimoniale
Âge des enfants
Options : renvoi, double appel, etc.
Heure d'appel
Code destination : étranger, local

4 étapes :

1. Préparation des données :

Définir la nature catégorique ou numérique des variables

2. Enrichissement des données :

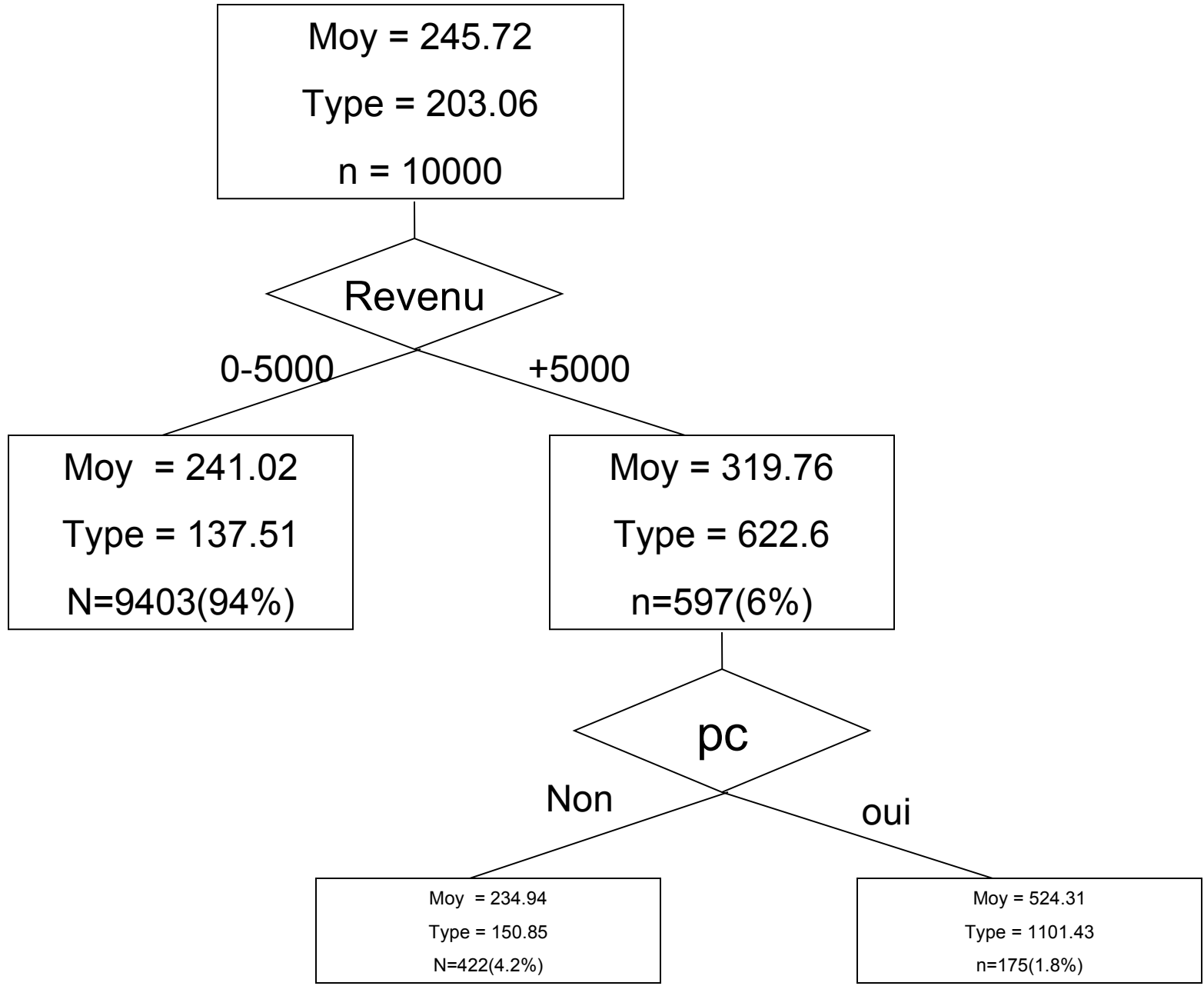
Organiser en taxonomie, typologie, combiner les variables

3. Création de l'arborescence

Manuel, Semi-automatique ou automatique

4. Validation de l'arborescence

Soit sur un mode statistique, soit sur un mode opérationnel



REGLE_17 : SI

Pc = Oui

Revenu = 0 or 30 000 +

ALORS

durée_appel : moyenne 524.309, écart type : 1101,43

REGLE_10 : SI

Marié(e) = Oui

Propriétaire_maison = Oui

Membres5-18 = 2 or 3

Console_jeux = Oui

Satellite = Oui

Revenu = 10-20 000, 0- 10 000 or 20-30 000

ALORS

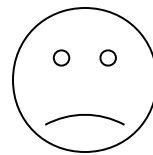
durée_appel : moyenne 376.478, écart type : 131,73

Simplicité

Lisibilité

Multiplicité des arbres possibles

Taille de l'arbre



Data Mining has been identified as one of the ten emergent technologies of the 21st century (MIT Technology Review, 2001). This discipline aims at discovering knowledge relevant to decision making from large amounts of data. After some knowledge has been discovered, the final user (a decision-maker or a data-analyst) is unfortunately confronted with a major difficulty in the validation stage: he/she must cope with the typically numerous extracted pieces of knowledge in order to select the most interesting ones according to his/her preferences. For this reason, during the last decade, the designing of quality measures (or interestingness measures) has become an important challenge in Data Mining.

Alchimie de la transformation des données en connaissances

Nécessité et Proposition d'un cadre méthodologique pour la
KDD : Knowledge Discovery in Database

Phase 1 / POSER LE PROBLEME

Phase 2 / LA RECHERCHE DES DONNEES

Phase 3 / LA SELECTION DES DONNEES PERTINENTES

Phase 4 / LE NETTOYAGE DES DONNEES

Phase 5 / LES ACTIONS SUR LES VARIABLES

Phase 6 / LA RECHERCHE DU MODELE (tout ce qui précède dans le cours)

Phase 7 / L'EVALUATION DU RESULTAT

Phase 8 / L'INTEGRATION DE LA CONNAISSANCE

Phase 1 / POSER LE PROBLEME

Cas de marketing classique : identification de profils de clients et organisation d'une campagne de marketing direct

Il s'agit d'un voyageur organisant des circuits touristiques avec 5 types de prestation (A,B,C,D et E). Son directeur de marketing souhaite mettre en place une politique de fidélisation.

Décomposition en sous-problèmes précis :

Fidéliser la clientèle ?

Clientèle ?

Typologie de problèmes à résoudre :

Structuration : qui sont mes clients ?

Affectation : quels sont les clients à contacter ?

Objectifs :

1. Connaître les clients pour revoir les offres et la politique marketing.
2. Fournir à la cellule marketing opérationnelle et aux réseaux de distribution une liste ciblée de clients par requête SQL, ce qui implique des critères compréhensibles

Phase 2 / LA RECHERCHE DES DONNEES

La base d'informations à disposition

Informations sur le client :

- âge;
- sexe;
- situation matrimoniale : marié ou non;
- nombre d'enfants à charge;
- catégorie socioprofessionnelle;
- nombre d'année dans son emploi;

Informations sur le produit acheté :

- produit A avec la date de 1er achat;
- produit B avec la date de 1er achat;
- produit C avec la date de 1er achat;
- produit D avec la date de 1er achat;
- produit E avec la date de 1er achat;

Informations comptables :

- montant des achats;
- date du dernier achat;
- type de paiement;
- statut financier du client : bon, moyen ou mauvais;

Informations collectées par questionnaires et enquêtes :

- centre d'intérêts;

SGBD Entreprise

INSEE

Informations géographiques :

- code commune;
- taille de la commune;
- type d'habitat;

Relier toutes ces bases d'information ? Jointures ? Manuelles ou Automatiques ?

Phase 3 / LA SELECTION DES DONNEES PERTINENTES

Procédure d'extraction des échantillons représentatifs des enjeux marketing : accroître le CA :

- Si le client n'a effectué aucun achat au cours des 5 dernières années, alors pas d'extraction;
- Si le client a acheté pour plus de 3000 Euros , alors on tire aléatoirement un enregistrement sur 3;
- Sinon, on tire aléatoirement un enregistrement sur 10

Phase 4 / LE NETTOYAGE DES DONNEES

Pour l'heure, après extraction et sélection, environ un fichier de 1400 clients à analyser.

Les données sont forcément "gâtées" par le cheminement qui a conduit à leur stockage et notamment par la saisie manuelle par des opérateurs différents. Les erreurs, oublis, approximations ponctuels ou systématiques doivent être détectés et corrigés le mieux possible.

Choix de traitement automatique des :

- Valeurs aberrantes;
 - Analyse des valeurs min et max au 2^e et 98^e centiles;
 - Analyse de la distribution afin de vérifier son homogénéité;
 - Contrôle de cohérence de certaines informations;
- Valeurs manquantes;
- Valeurs nulles.

Phase 5 / LES ACTIONS SUR LES VARIABLES

Adaptation aux contraintes de la modélisation par :

- Enrichissement :

- le croisement de la Date du premier achat et de la Date du dernier achat permet de déterminer la longévité du client dans la compagnie de voyages;
- Le croisement des variables Type d'habitat et Taille de la commune permet de caractériser le style d'habitat :
 - oPetite ville + individuel = rural;
 - oGrande ville + individuel = banlieue chic;
 - oGrande ville + collectif = forte concentration, etc.

- Normalisation des distributions

Phase 6 / LA RECHERCHE DU MODELE (tout ce qui précède dans le cours)

Le fichier est nettoyé et complété.

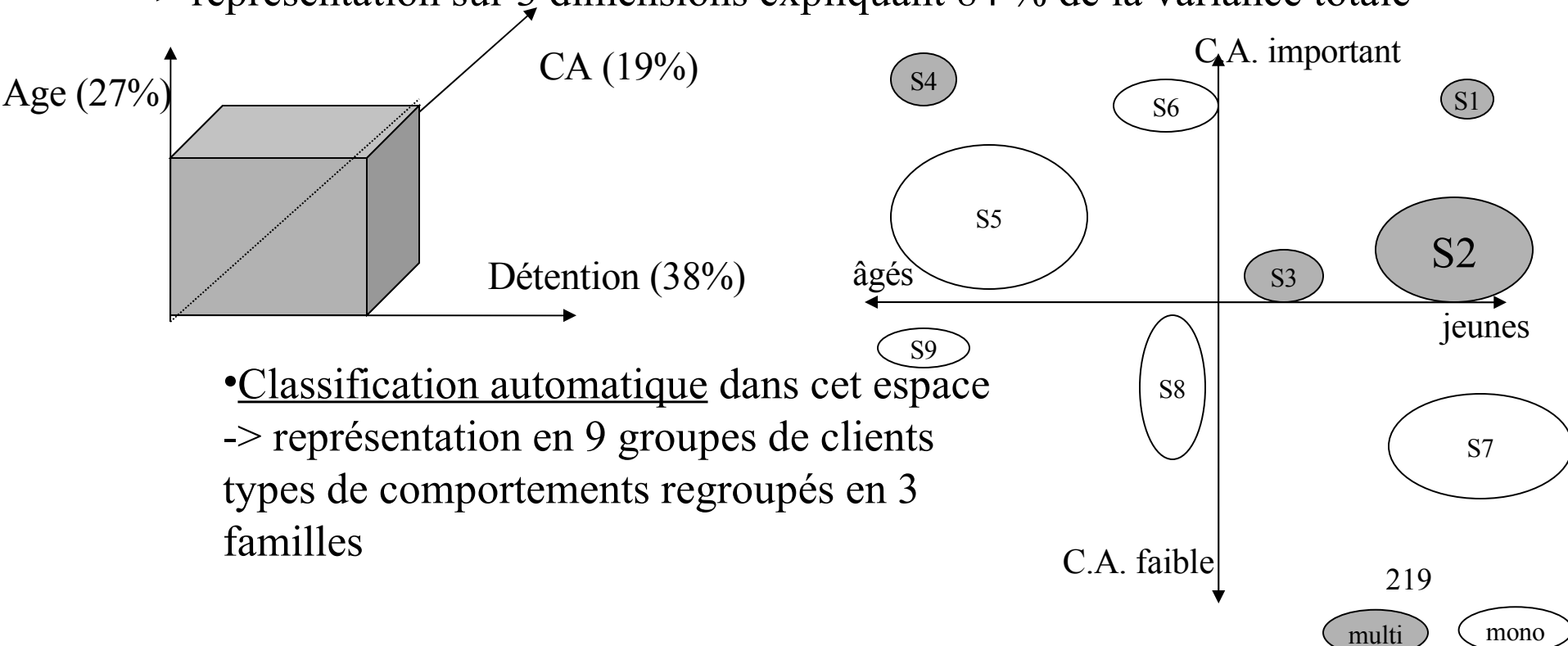
Il reste à trouver les critères de structuration et d'affectation.

- La recherche des facteurs pertinents :

Recherche des typologie de clients par :

- Analyse factorielle pour la compréhension des principaux facteurs de différenciation des clients

-> représentation sur 3 dimensions expliquant 84 % de la variance totale



- Classification automatique dans cet espace

-> représentation en 9 groupes de clients
types de comportements regroupés en 3 familles

Phase 6 / LA RECHERCHE DU MODELE (tout ce qui précède dans le cours)

- La recherche des modèles de ventes croisées :

Il faut construire trois approches, différenciées par la notion d'âge des clients : quels sont les facteurs comportementaux qui permettent de caractériser les gros chiffres d'affaires parmi les clients jeunes, middle age, et âgés. Exemple, sur les clients jeunes par :

- La préparation par réseau de neurones qui répartit notre population en 4 classes :

- Les multi : les multiacheteurs prédits multiacheteurs (45 %);
- Les mono : les monoacheteurs prédits monoacheteurs (30 %). Ces deux catégories expliquent notre modèle à 75 %;
- Les prospects : les monoacheteurs prédits multiacheteurs (15 %);
- Les erreurs : les multiacheteurs prédits monoacheteurs (10 %).

Conclusion : les prospects représentent une part importante des monoacheteurs ce qui constitue un point positif en terme de CA.

- La formalisation de la connaissance par arbre de décision.
- La synthèse.

Phase 7 / L'EVALUATION DU RESULTAT

- Analyse statistique et écart entre taux de classification constaté sur la base d'apprentissage et sur la base de test.
- Croisement avec les connaissances des experts : ici, commerciaux et spécialistes du marketing : être prêt à répondre à tous les résultats qui vont à l'encontre des intuitions de ces derniers -> nécessite une vraie expertise dans le domaine du Data Mining.

Phase 8 / L'INTEGRATION DE LA CONNAISSANCE

- Dans sa fonction de communication;
- Dans sa fonction de production-logistique;

Le tout, permet de juger sa capacité à s'adapter aux différents types de clients révélés par le processus de Data Mining.

Intelligent Miner, d'IBM

Clementine, de SPSS

SAS Enterprise Miner, de SAS

TeraMiner, de NCR

KXEN Components, de KXEN

SPAD, Start Miner, Alice d'Isoft, NeuroText de Grimmer

Bibliographie :

- *Data mining : Gestion de la relation client, Personnalisation de Sites Web*, par René Lefébure et Gilles Venturi, Edition Eyrolles (pour la partie appliquée surtout)
- *Mastering Data Mining : the art and the science of customer relationship management*, par M. J.A. Berry et G. S. Linoff, Edition Wiley (ouvrage très complet)
- *Apprentissage artificiel : concepts et algorithmes*, par A. Cornuéjols et L. Miclet, Edition Eyrolles (pour la partie théorique du cours)

#####

Appel à soumissions pour un
numéro spécial de la Revue RNTI

FOUILLE DES DONNEES D'OPINIONS

http://www.lirmm.fr/~mroche/FODOP08/Appel_RevueRNTI_opinions.pdf

Objectifs de ce numéro spécial :

Pour faire suite à l'atelier FODOP'08

(<http://www.lirmm.fr/~mroche/FODOP08/>) sur la problématique de la fouille des données d'opinions, un numéro spécial est proposé sur cette même thématique. De plus en plus de documents contenant des informations exprimant des opinions ou des sentiments apparaissent sur le Web (i.e. commentaires ou évaluations de produits par des clients, forums, groupe de discussion, blogs). Aujourd'hui, la détection ou l'extraction automatique d'opinions devient un domaine de recherche particulièrement prometteur notamment avec le développement du Web 2.0. Elle devient essentielle pour de très nombreux domaines d'applications. Nous pouvons citer, par exemple, le développement de tâches de veille (technologique, marketing concurrentielle, sociétale), l'évaluation d'un produit par la communauté avant un achat, l'image que les clients peuvent se faire d'une entreprise, la détection de rumeurs (buzz) sur le web, détection d'opinions émergentes et/ou significatives dans les forums, etc. Ainsi, les approches traditionnelles de fouille de données doivent être adaptées à un contexte dans lequel il faut appréhender de gros volumes de données particulièrement hétérogènes. L'objectif principal de ce numéro spécial est alors de décrire les différents traitements des données d'opinions.

Thèmes développés (liste non exhaustive) :

-
- Catégorisation automatique de textes d'opinion
 - Ontologies et données d'opinion
 - Aide à la décision pour les données d'opinion
 - Visualisation pour une analyse des données d'opinion
 - Interrogation des données d'opinion
 - Acquisition de bases de données d'opinion
 - Veille technologique pour l'analyse des opinions
 - Détection de tendances dans les opinions
 - Fouille du Web pour acquérir et/ou analyser des textes d'opinion
 - TAL à partir de textes d'opinion
 - Analyse de discours