PARIS DESCARTES

# *Data Mining /Machine Learning/ Big Data*

Biblio :

Bioinformatics: The Machine Learning Approach, Second Edition (Adaptive Computation and Machine Learning) – 2001 par Pierre Baldi et Søren Brunak

Introduction à la bioinformatique - 2001, par Cynthia Gibas et Per Jambeck. (traduit de l'anglais)

Geographic Data Mining and Knowledge Discovery Second Edition publié par Harvey J. Miller,Jiawei Han

Webs:

http://chem-eng.utoronto.ca/~datamining/dmc/data_mining_map.htm

http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0115892

http://svmcompbio.tuebingen.mpg.de/

http://infolab.stanford.edu/~ullman/pub/book.pdf

http://www.math.cmu.edu/~ctsourak/resources.html

## Texte Mining

"Pharmaceuticals often have side effects that go unnoticed until they're already available to the public. Doctors and even the FDA have a hard time predicting what drug combinations will lead to serious problems. But thanks to people scouring the web for the side effects of the drugs they're taking, researchers have now shown that Google and other **search engines** can be mined for dangerous drug combinations. In a new study, scientists tried the approach out on predicting hypoglycemia, or low blood sugar. They found that the data-mining procedure correctly predicted whether a drug combo did or did not cause hypoglycemia about 81% of the time."

http://news.sciencemag.org/math/2013/03/should-you-mix-those-two-drugs-ask-dr.-google?ref=hp

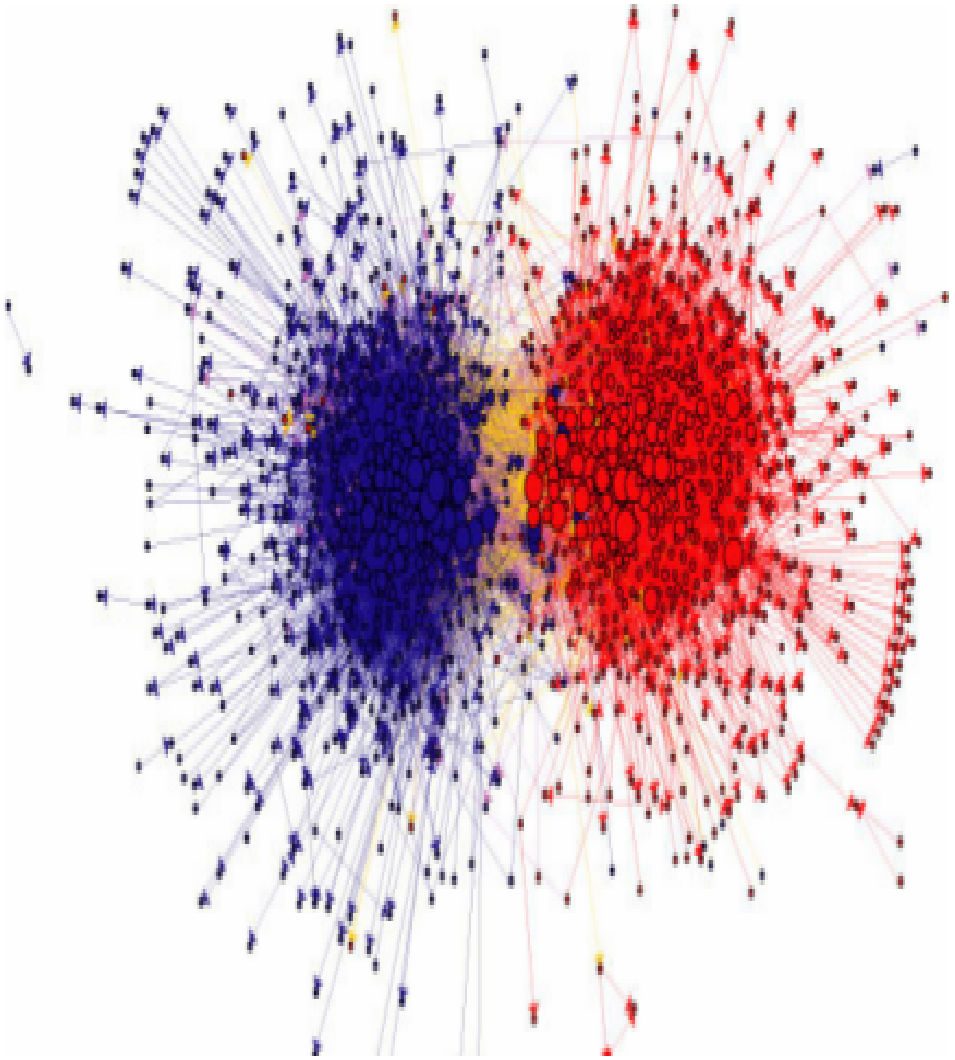
Paroxetine (anti-depresseur) + pravastatin (suppresseur de cholesterol)
→ hyperglycemie (bouche sèche, xerostomie etc.)

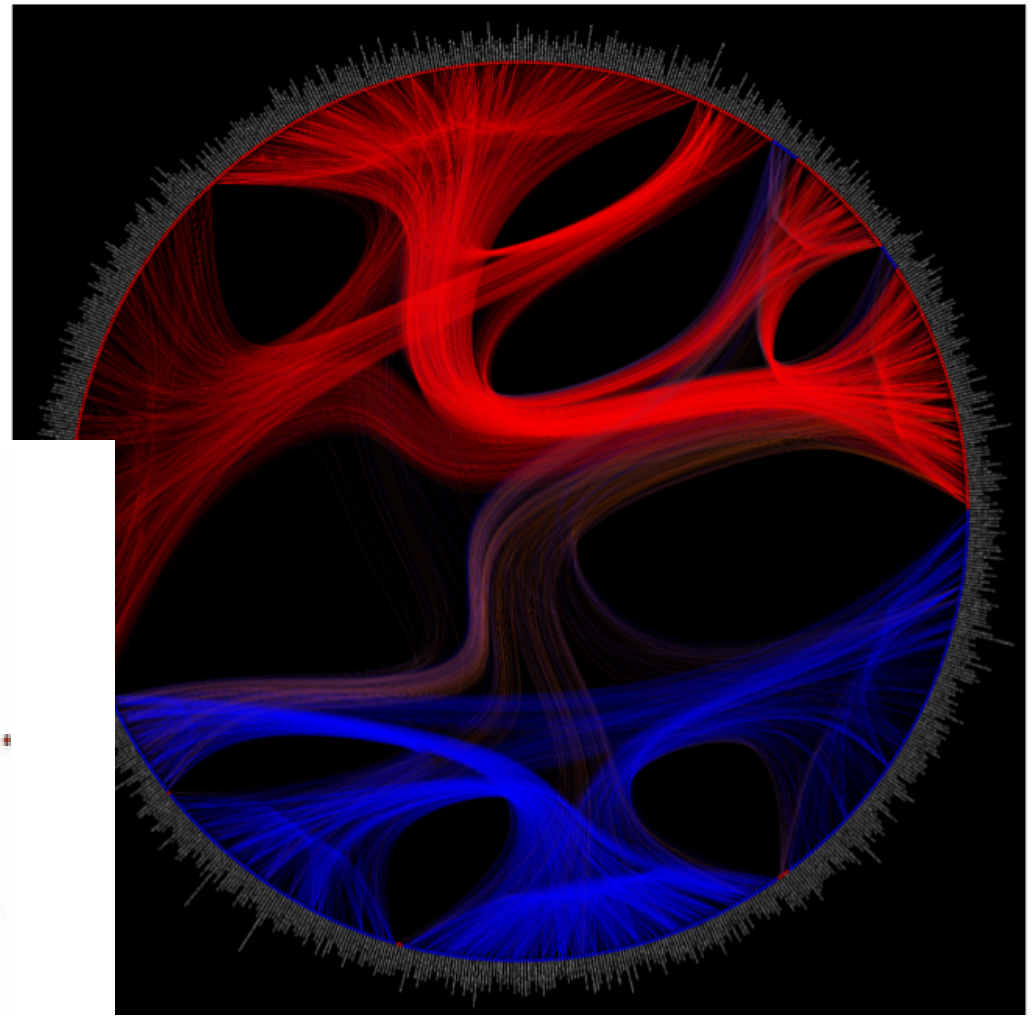
http://watchdoglabs.org/blog/car-resources/analysis/

Blog mining election

http://graphexploration.cond.org/

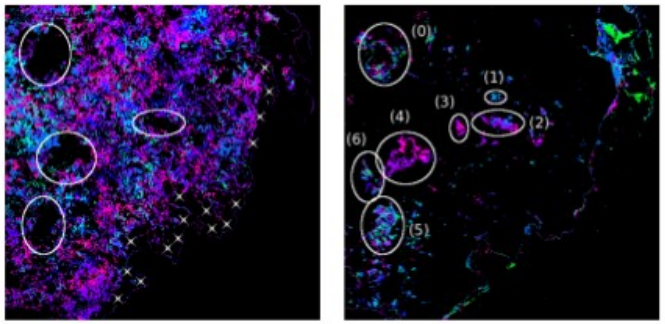
Community detection in networks



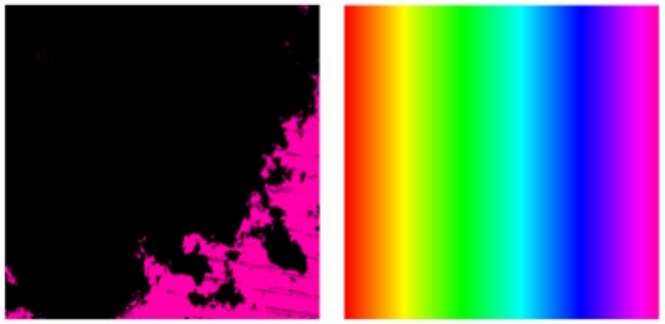


# Image Mining

(a) RGB image - 22 October 2002 (b) RGB image - 27 April 2007




(c) Pattern 2, 2, 3, 2, 2, 2, 3 (d) Pattern 2, 2, 1, 1, 1, 2



(e) Pattern 1, 1, 1, 1,1, 1, 1, 1,1, 1, 1, 1,1, 1, 1 (f) Temporal color scale from red (year 2000) to violet (year 2011)

hdr_Genovesio2013.pdf

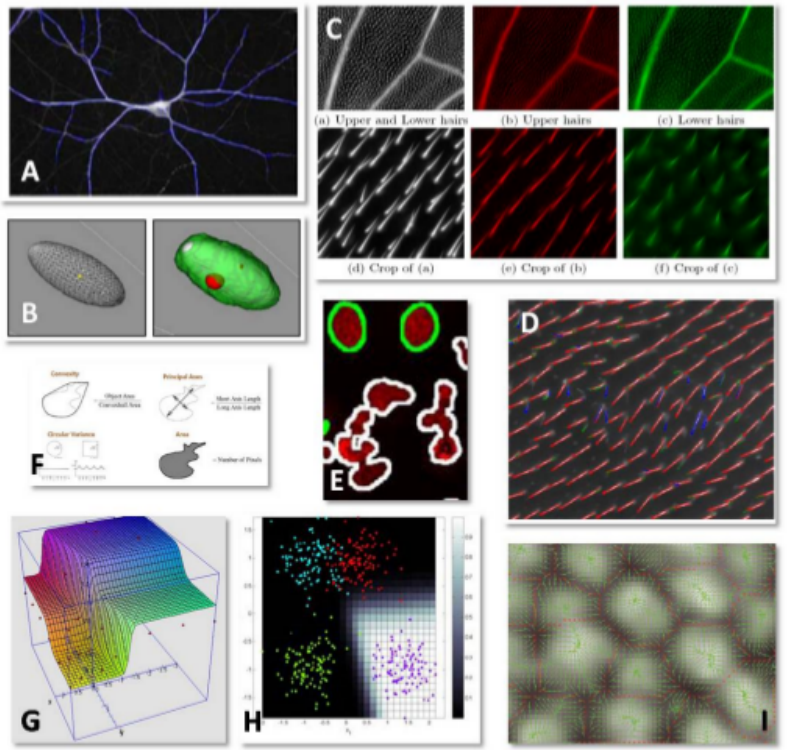



**FIGURE 2.1:** Mise au point de plusieurs méthodes de détection pour des A - cellules neuronales, B - noyaux de cellules tridimensionnels, C, D - pilosité d'ailes de drosophiles mutantes, E,I - agglomération de noyaux de cellules, F - une batterie de descripteur a également été développée pour décrire ensuite les objets détectés et G,H - divers type d'analyse et de classifiers sont ensuite étudiés et utilises pour sélectionner les expériences présentant des phénotypes recherches selon l'application.

**Data Mining**


ENGEES – ASTEE – 20 mars 2014

Gestion des grands volumes de données rivières : structuration, explorations innovantes et utilisations pratiques

## Les besoins opérationnels : comment les outils de fouille de données peuvent y répondre

**Danielle Levet**

ANR

aquascop

fresQueau

*ANR_AquaScop2014.pdf*

« Dans son acception la plus générale, **l'induction** est une opération mentale consistant à généraliser un raisonnement ou une observation à partir de cas singuliers.

En philosophie, l'induction est une démarche intellectuelle qui consiste à procéder par inférence probable, c'est-à-dire à déduire des lois par généralisation des observations. Par exemple, en l'absence de toute connaissance scientifique en astronomie, la plupart des gens s'attendent à voir le soleil se lever le lendemain matin. Dans le contexte scientifique, et à condition de bien en mesurer les limites, l'induction peut trouver sa place. Par exemple, l'accumulation d'études monographiques peut conduire à formuler, par généralisation, des propositions relatives au changement social.

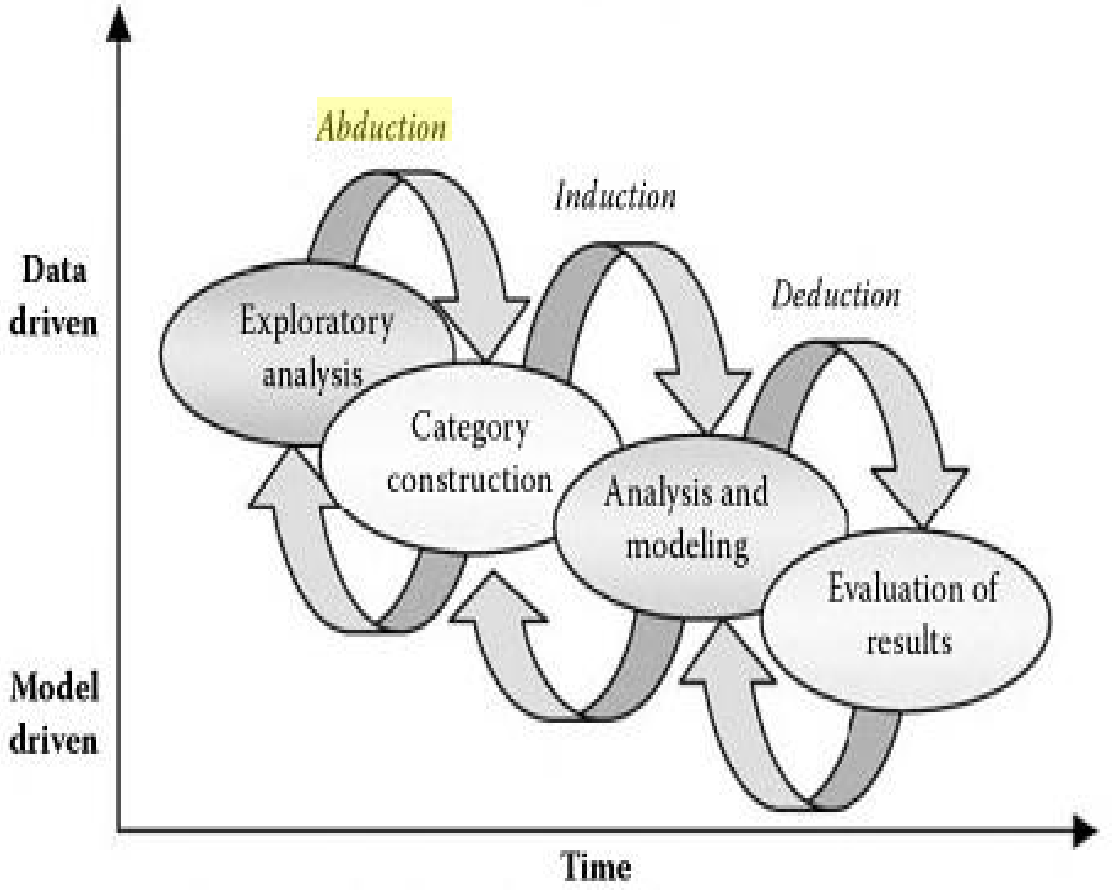
En mathématiques, en logique et en informatique, l'induction complète, aujourd'hui très souvent abrégée en induction, est une autre façon de désigner la récurrence : aussi bien le raisonnement par récurrence que les définitions par récurrence. Le terme est souvent employé pour les généralisations de la récurrence aux bons ordres et relations bien fondées. Le terme d'Ensemble inductif est également employé en liaison avec le lemme de Zorn.

Induction (logique et philosophie), genre de raisonnement qui se propose de chercher des lois générales à partir de l'observation de faits particuliers, sur une base probabiliste. »

https://fr.wikipedia.org/wiki/D%C3%A9duction_et_induction

« L'abduction (du latin « abductio » : emmener) ou inférence de la meilleure explication1 est un mode de raisonnement. Elle consiste, lorsque l'on observe un fait dont on connaît une cause possible, à conclure à titre d'hypothèse que le fait est probablement dû à cette cause-ci.

L'abduction est communément admise, avec la déduction et l'induction, comme l'un des trois types majeurs d'inférence. C'est une forme de raisonnement utilisé dans le processus de découverte par sérendipité. »




*Exploratory visual analysis, visual data mining*

*Visual knowledge construction & refinement*

*Visual model tracking/ model steering*

*Data presentation, Visualization of uncertainty*

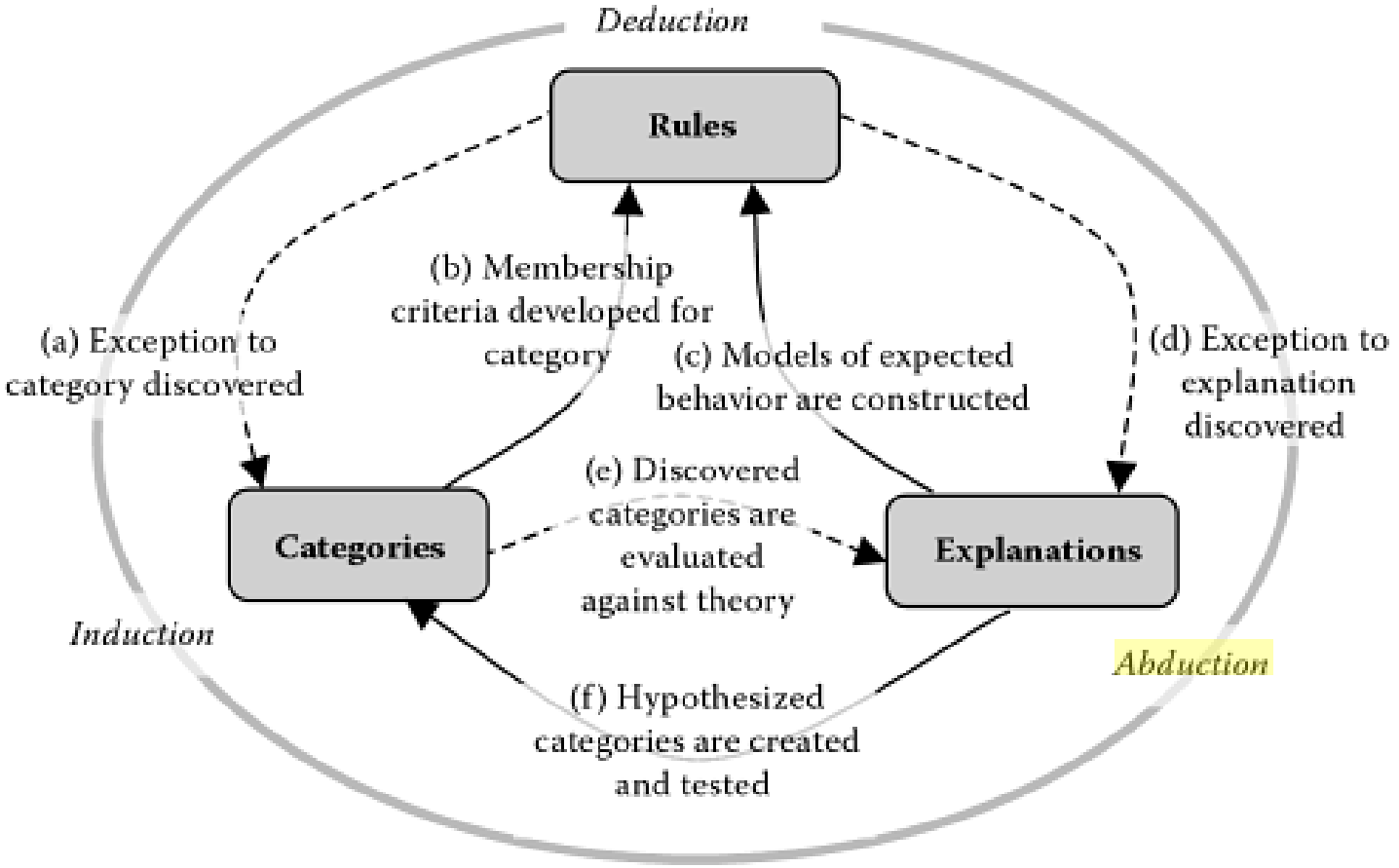



**FIGURE 11.11** A roadmap for discovery-based science. Transformations such as these occur when normal science breaks down. They result in the creation, modification, and retirement of conceptual structures such as rules, categories, and explanations. Constructive activities are shown as solid arrows, revision activities as dashed arrows.

PARIS DESCARTES

# Chapitre I. Clustering 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

- 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

# 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->{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.

ko/(largeur*hauteur) ? bmp gif Format

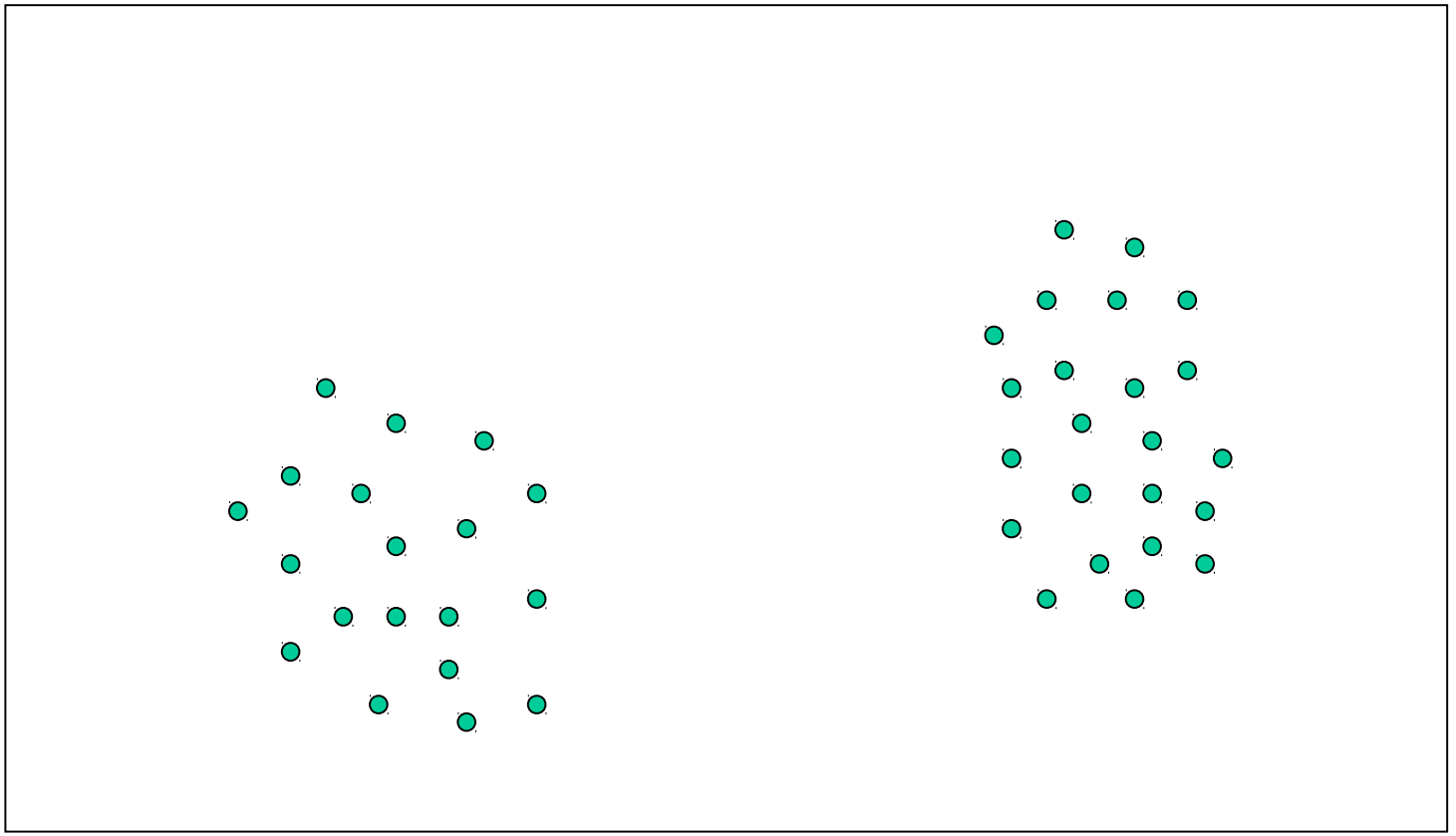

%Contour

intensité moyenne



Notion d'espace, de dimension ...

# « Clustering »



En Data Mining, on travaille en général dans des espaces de très grande dimension mais dans ces transparents on représente toujours les cas en deux dimensions par simplicité.

# Algorithme Séquentiel Basique (ASB) :

INPUT : *S={x1,x2,…,xN}, seuil de distance $\Theta$ et seuil de nombre de classes q*

- *m=1*
- $C_m=\{x_1\}$
- *For i=2 to N*
  - *Find* $C_k$ *:* $d(x_i, C_k) = \min_{1 \le j \le m} d(x_i, C_j)$
  - *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

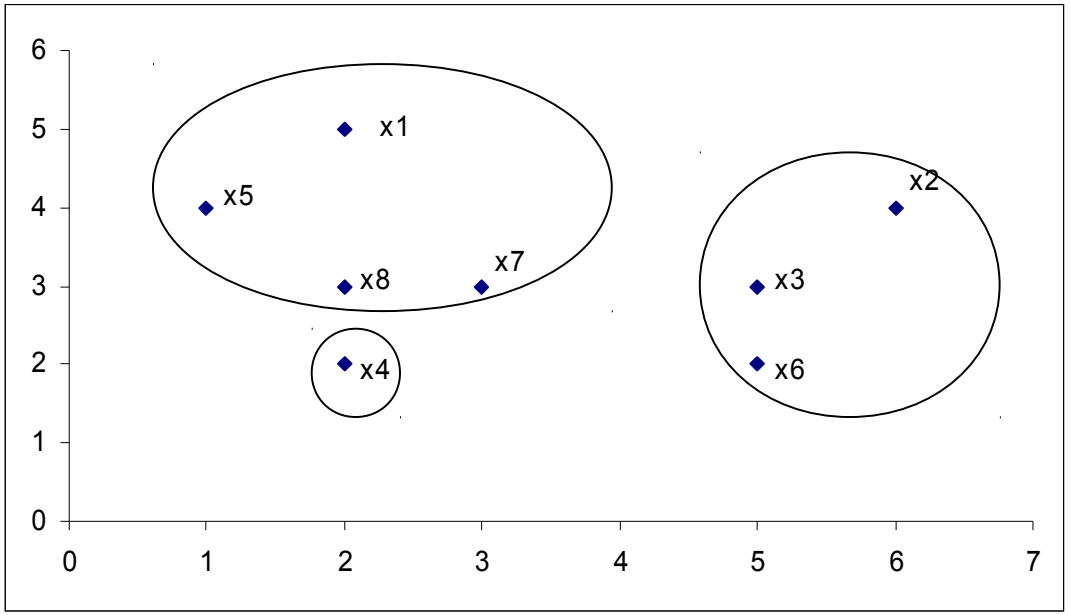
- *m=1*
- *$C_m$={$x_1$}*
- *For i=2 to N*
  - *Find $C_k$ : d ( $x_i$ , $C_k$) = min $_{1\le j\le m}$ d($x_i$,$C_j$)*
  - *If ( d( $x_i$ , $C_k$) > $\Theta$ ) AND ( m < q ) then*
    - ❖ *m = m+1*
    - ❖ *$C_m$ = {$x_i$}*
  - *End {If}*
- *End {For}*

### Pattern Classification

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

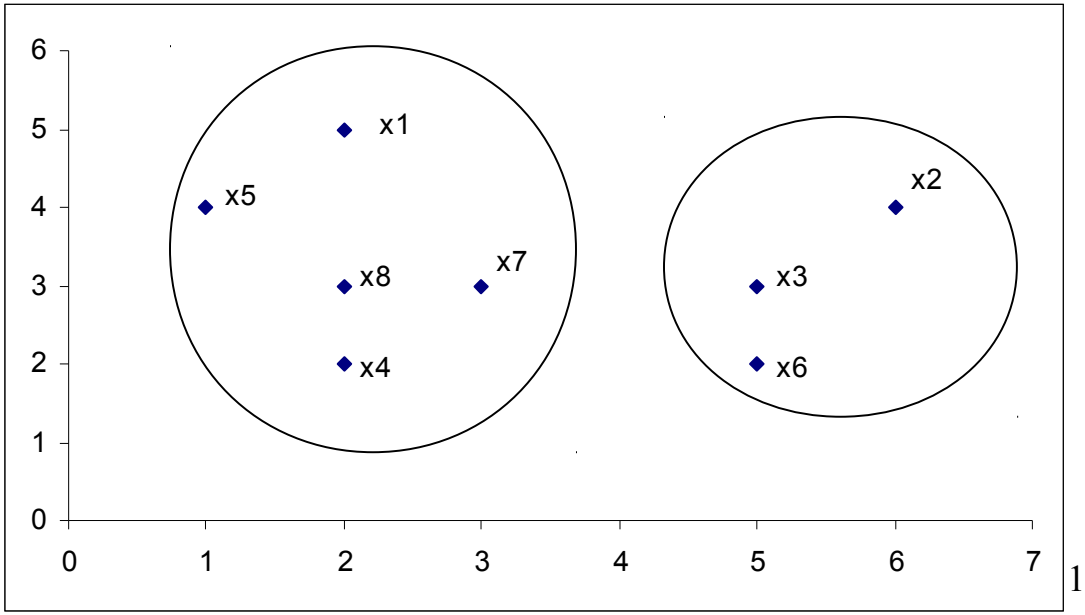
→ Remarque :

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




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

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



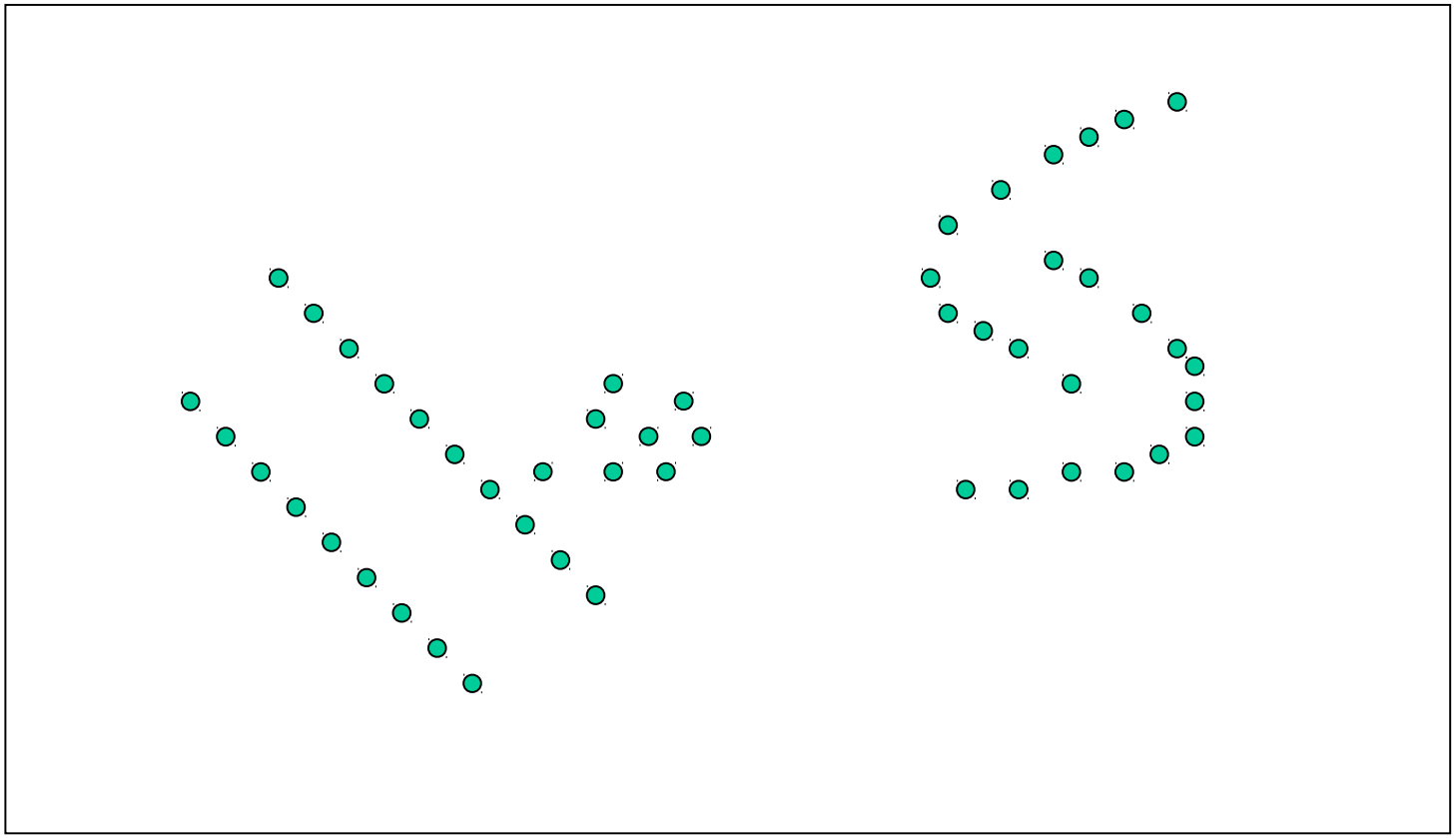
## Notes sur l'algorithme ASB :

- 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'orde de présentation (arbitraire)

- Et aussi de la mesure de proximité d(x,C) adoptée... dont la définition est l'objet de ce chapitre

# « Clustering »



# « Clustering »

A partir de maintenant,
Point = vecteur de caractéristiques
Ensemble = ensemble de vecteurs de caractéristiques

## Mesures de proximité entre 2 points

### Valeurs réelles

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

Distances de Mahalanobis, euclidienne, Manhattan, infini

$$s_{inner} = x^T y \ (corrélation)$$

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

### Valeurs discrètes

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

*Si* $x,y \in F^l$, *on définit la matrice de contingence* $A(x,y)_{k \times k} = [a_{ij}]$ *par* :

*aij* = 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 |

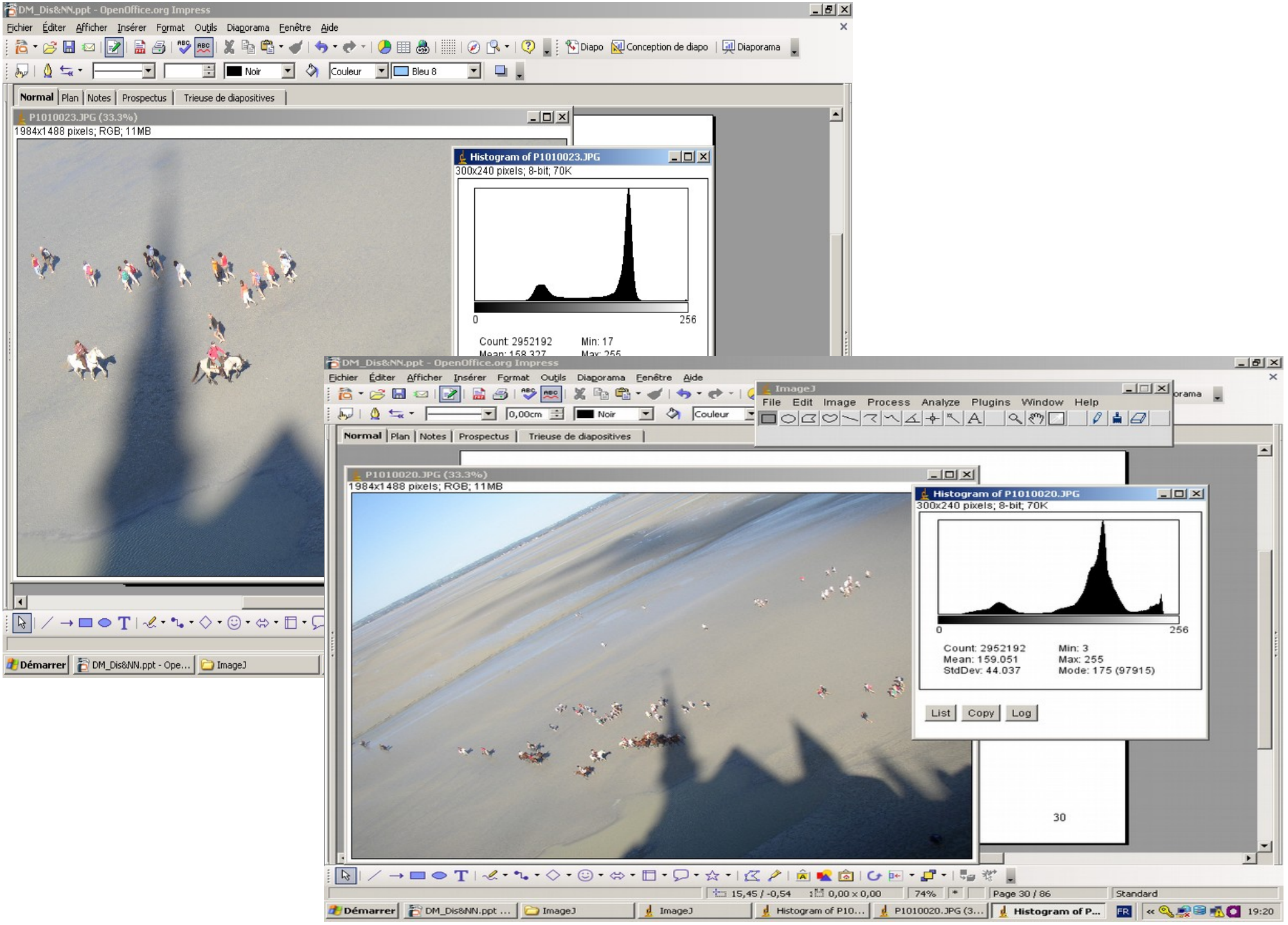
$m_{age}$=60, $s_{age}$=5

$m_{salaire}$=11074 $s_{salaire}$=48

→ d(p1,p2)=4,675

d(p1,p3)=2,324

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




Distance entre Histogrammes ?

Finalement, 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 »

Nombre de « clustering » possibles S

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

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

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

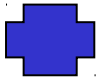
$$S(100,5) \approx 10^{68}$$

Si $10^{-12}$ secondes par cluster formé, il faudrait $10^{48}$ an/machine. Cela justifie le développement d'algorithme "informé"

# « 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={$x_1$,$x_2$,…,$x_N$}, Choose $R_0$ = { $C_i$ = {xi} , i = 1, …, 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*

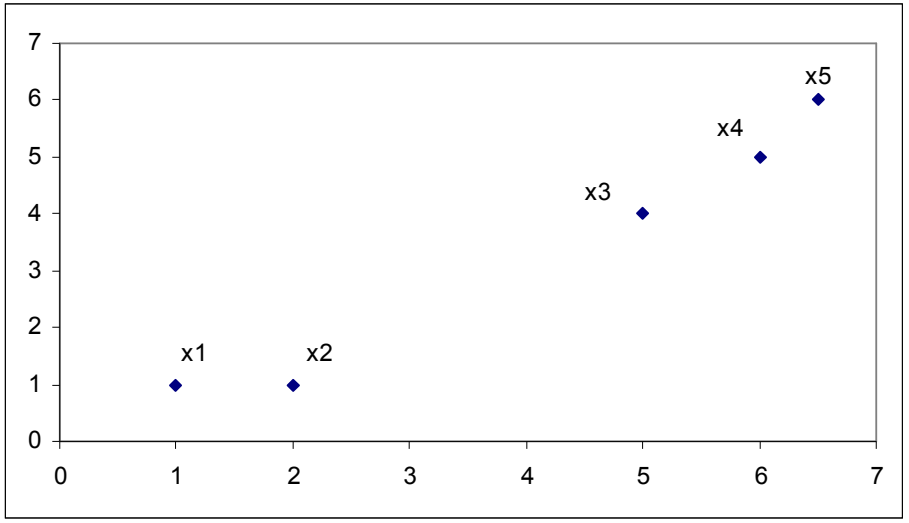
OUTPUT : Un ensemble de partitions R= $\cup R_i$ , avec $R_i$ = $\cup C^i_j$

# Data Structures

- Data matrix
  - (two modes)
- Dissimilarity matrix
  - (one mode)

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

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




## Matrice des Motifs

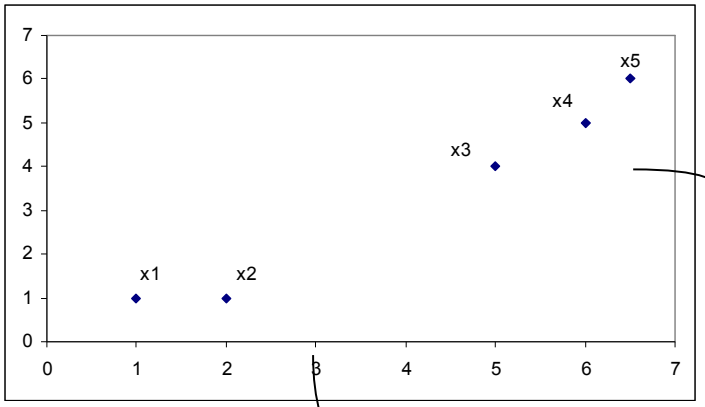
$$D(X) = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 5 & 4 \\ 6 & 5 \\ 6{,}5 & 6 \end{bmatrix}$$

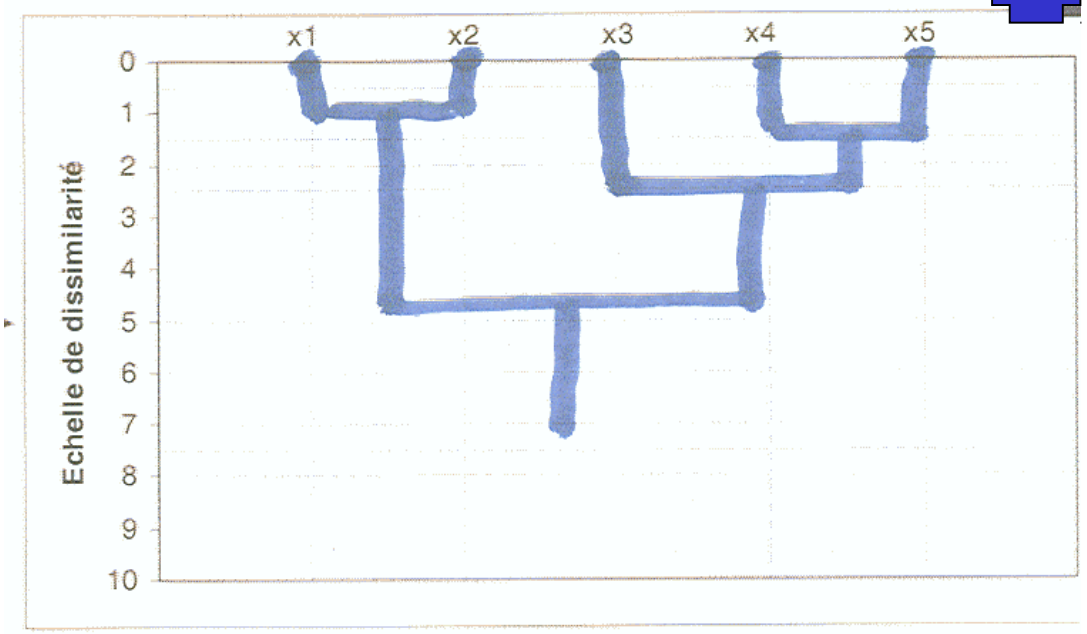
## Matrice des Dissimilarités

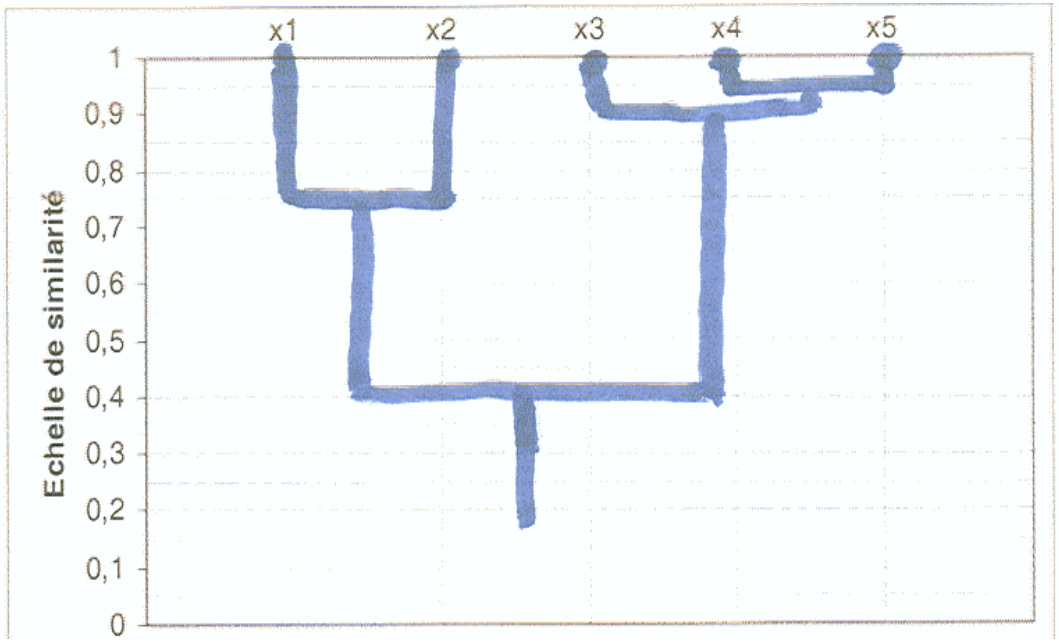
$$P_{euclid}(X) = \begin{bmatrix} 0 & 1 & 5 & 6{,}4 & 7{,}4 \\ & 0 & 4.2 & 5.7 & 6.7 \\ & & 0 & 1.4 & 2.5 \\ & & & 0 & 1.1 \\ & & & & 0 \end{bmatrix}$$

## Matrice des Similarités

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



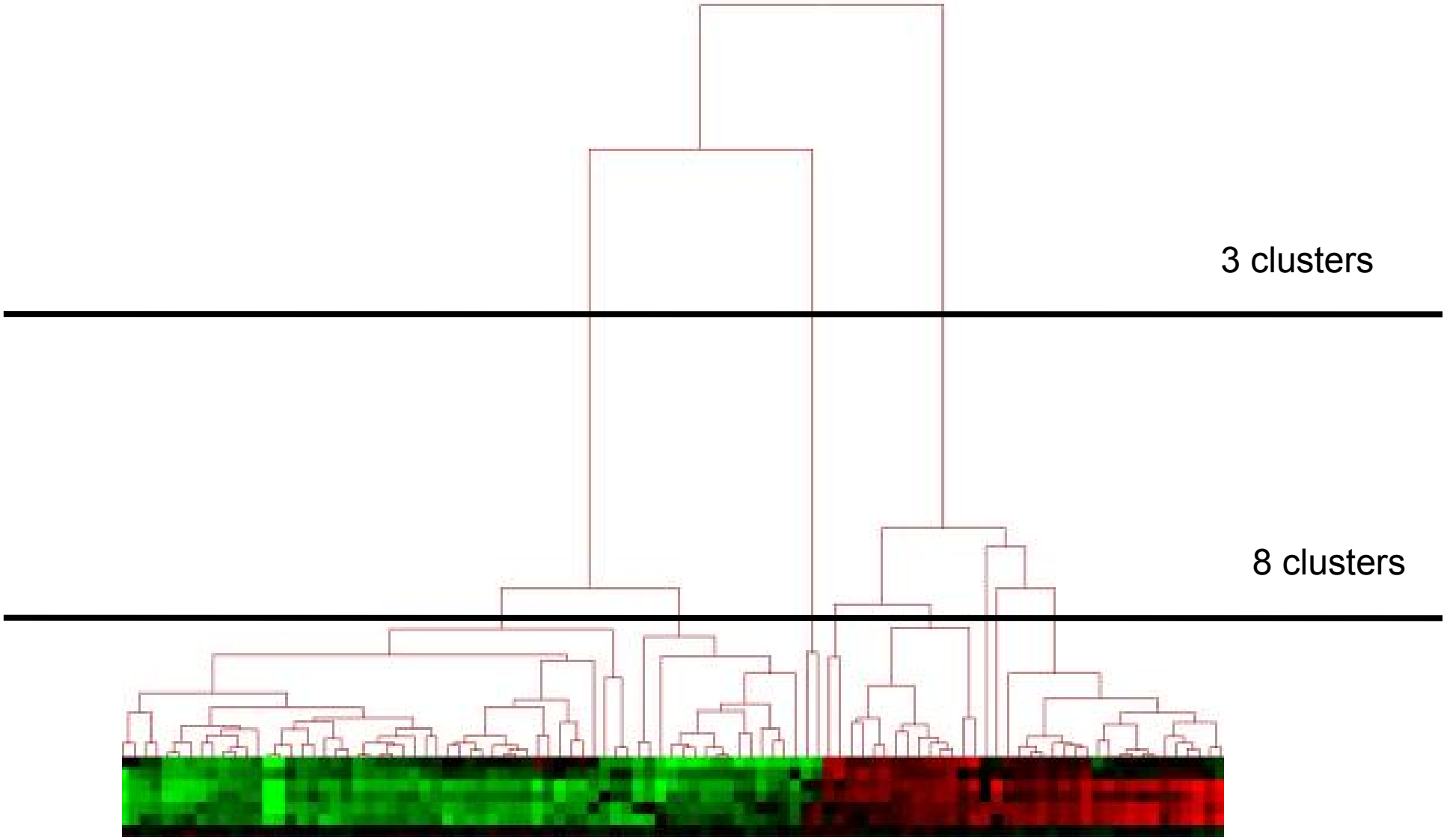




# Les Dendrogrammes

Notion de durée de vie d'un cluster

http://www.cs.umd.edu/hcil/hce/




*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

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

## 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

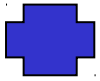
•Différents niveaux de granularité sont ainsi directement visualisable dans les données

•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 »

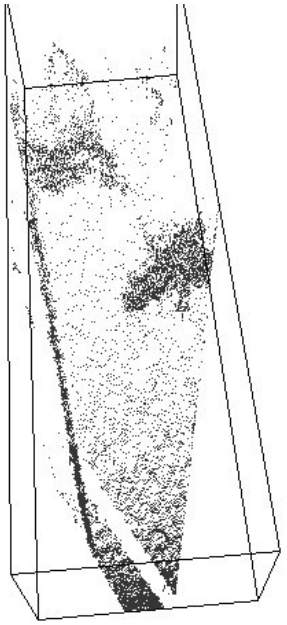
3 Catégories d'algorithmes :

- Séquentiels

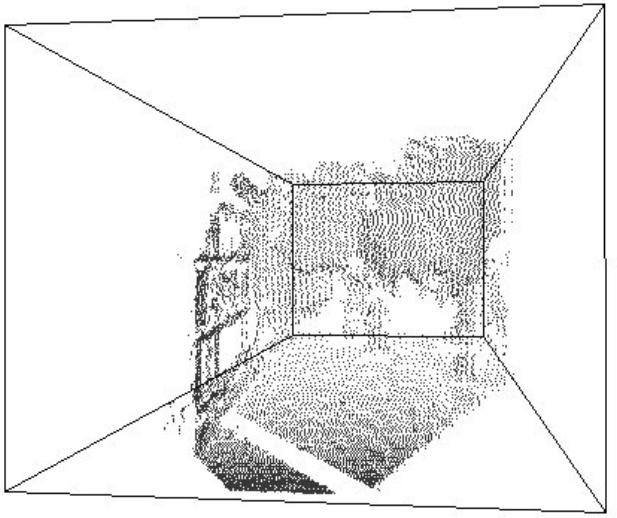


- Hiérarchiques

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





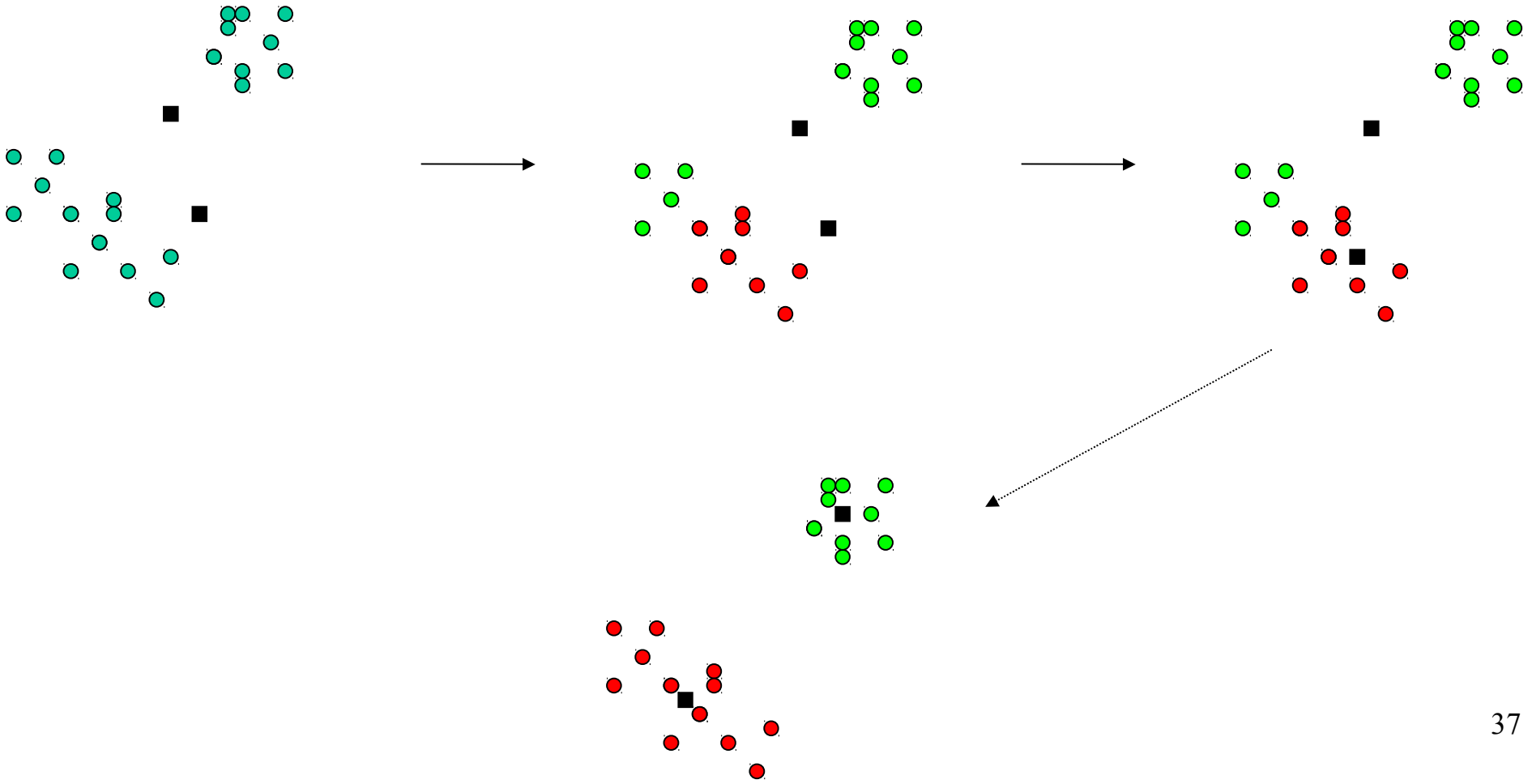


Nuées Dynamiques

Algorithme
ISODATA

C-moyennes floues

# Principe
# les Nuées Dynamiques



# Schéma numérique :
# Algorithme itératif de type ISODATA

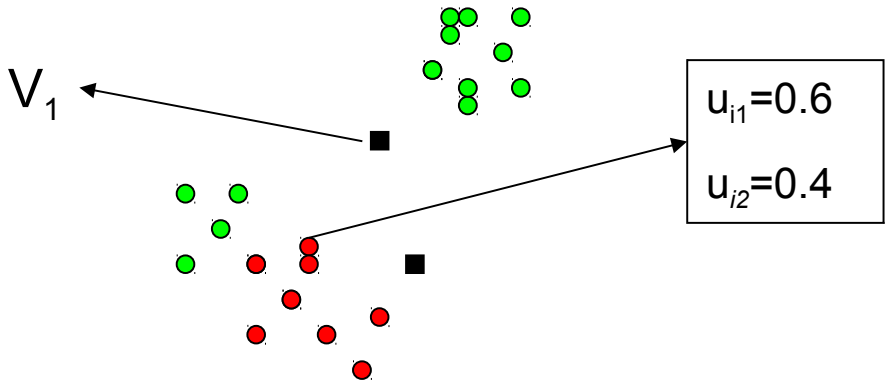
⇨ Nombre de groupements *C* connu

⇨*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={$x_1$,$x_2$,…,$x_N$}, un nombre de classes C et un ensemble de noyaux initial $V^0_i$, i = 1, …, C, itération n=1, nombre d'itérations maximum $n_0$

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

$$C_k^n \leftarrow \left\{ x_i \in S \middle| \forall j \neq k, d(x_i, V_k^{n-1}) \leq d(x_i, V_j^{n-1}) \right\}$$

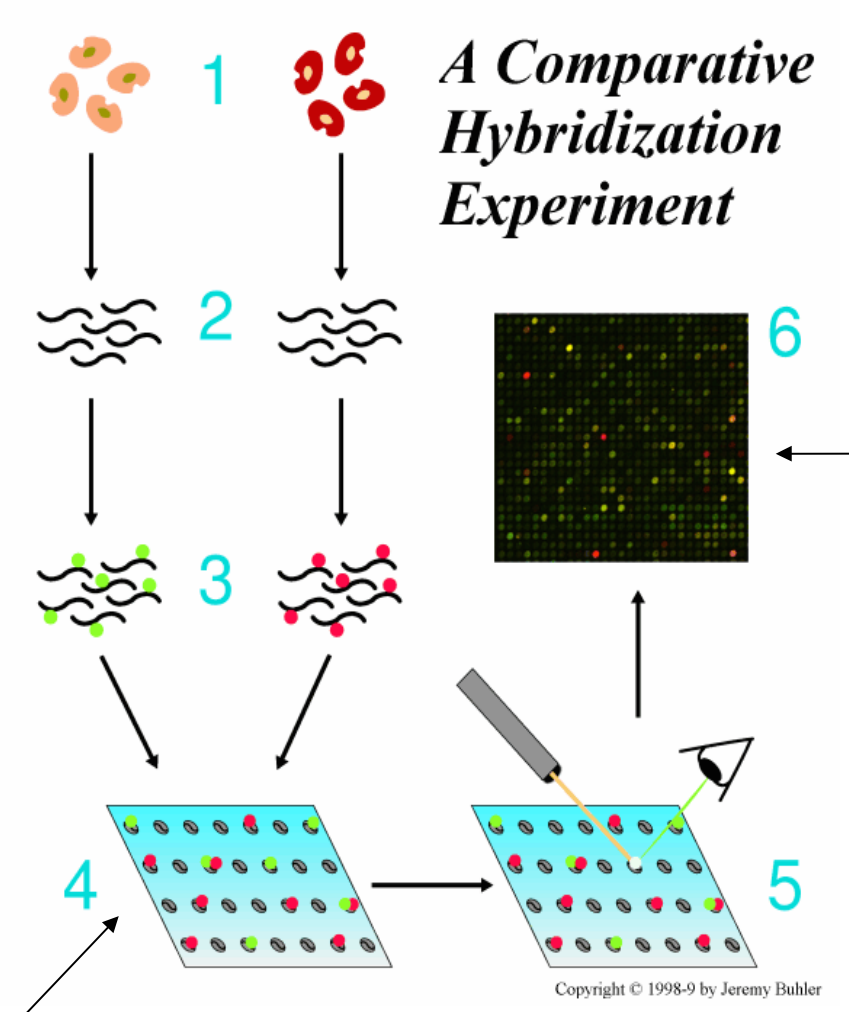
$$Calcul \quad de \quad V_k^n \, à \quad partir \quad de \, C_k^n$$

- **Si** $\forall k, C_k^n \neq C_k^{n-1} \quad et \quad n \leq n_0,$ retourner en 1

  **Sinon arrêt**

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

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

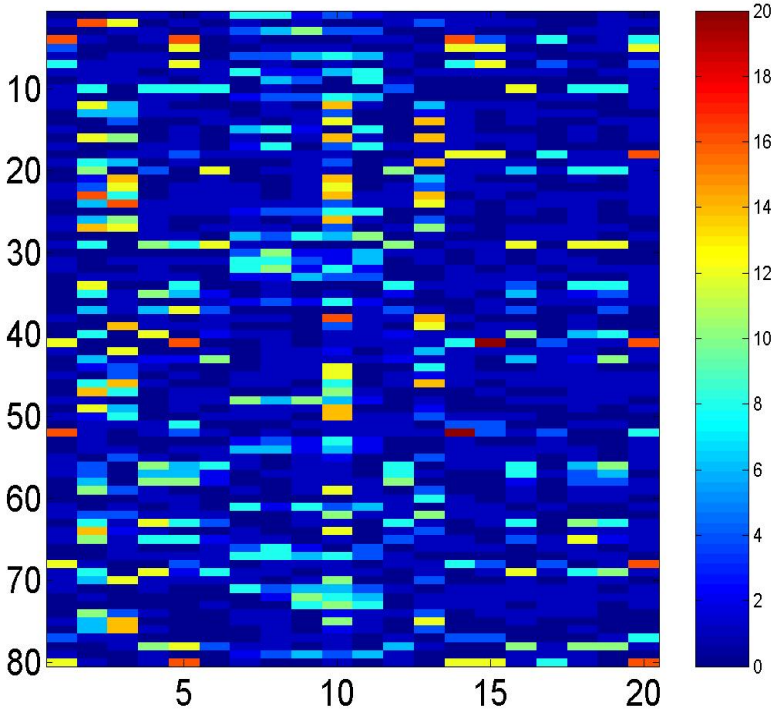



sample (labelled)

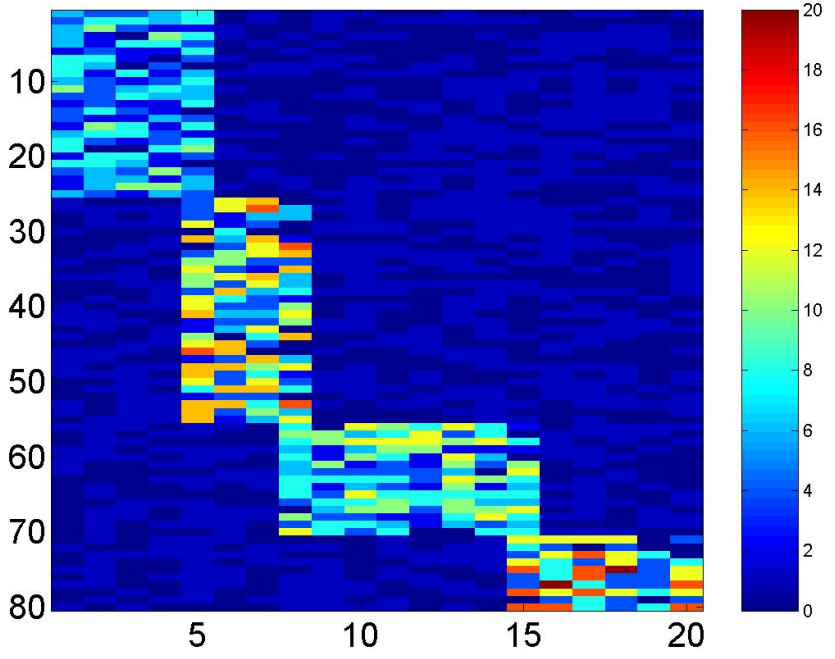
pseudo-colour image

probe (on chip)

[image from Jeremy Buhler]




# Regroupement en familles de gènes

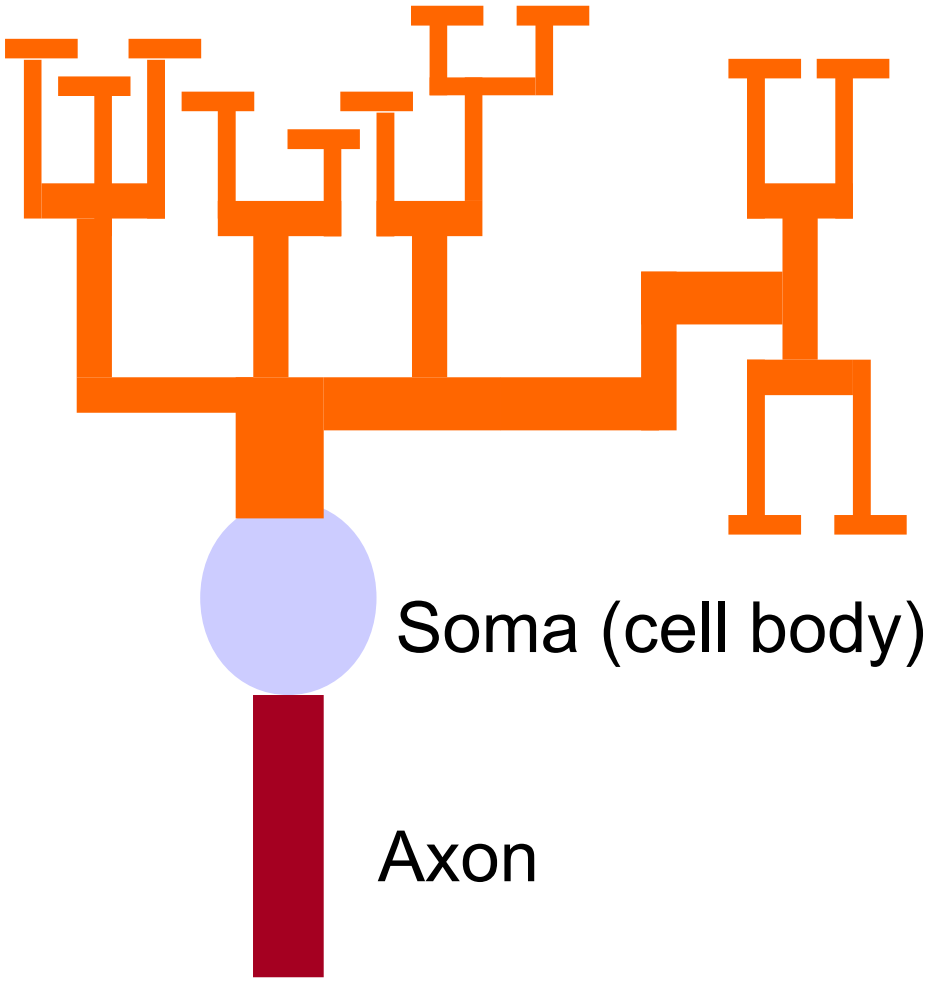


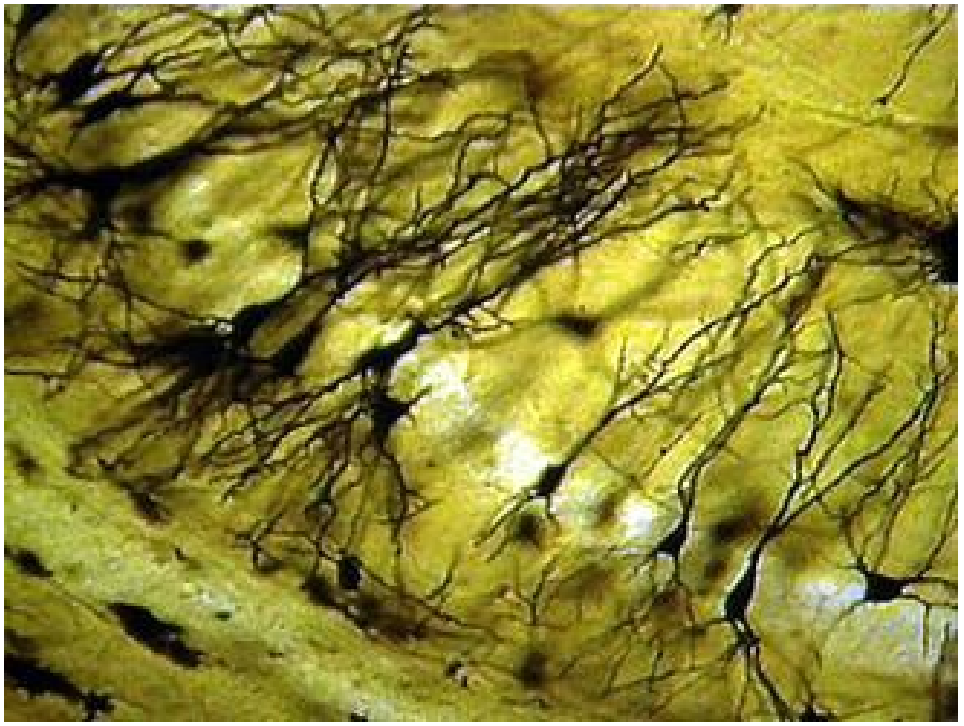
PARIS DESCARTES

# Chapitre II. Apprentissage *a posteriori* par Séparatrices Linéaires

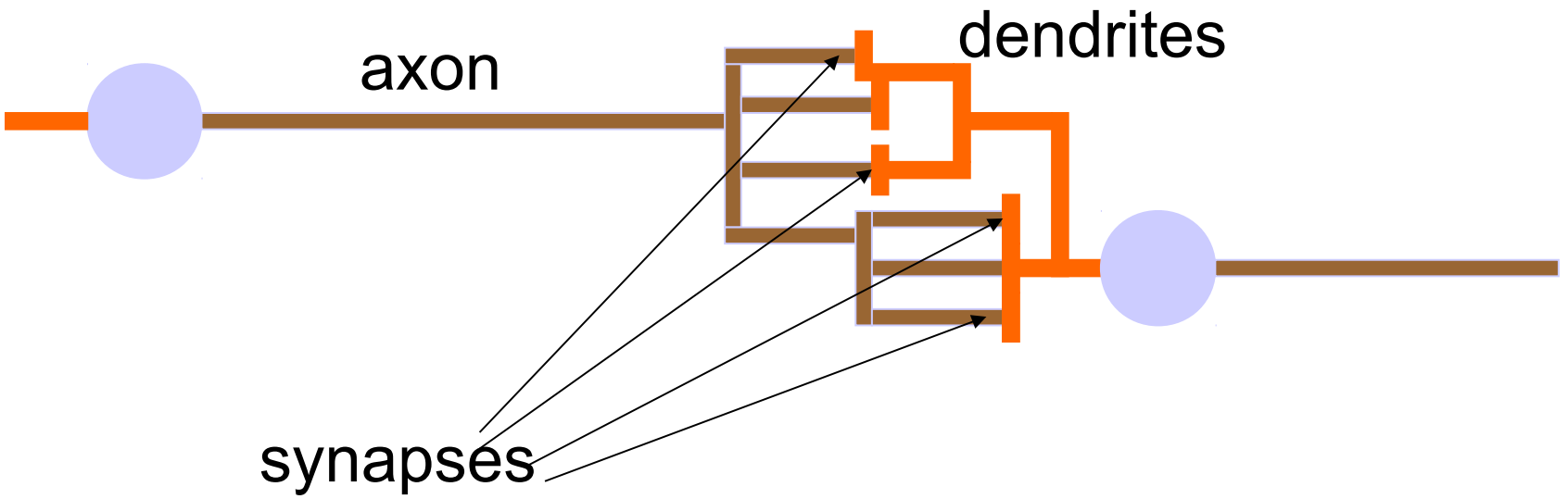
## Réseaux de Neurones

# Inspiration Biologique



# Inspiration Biologique



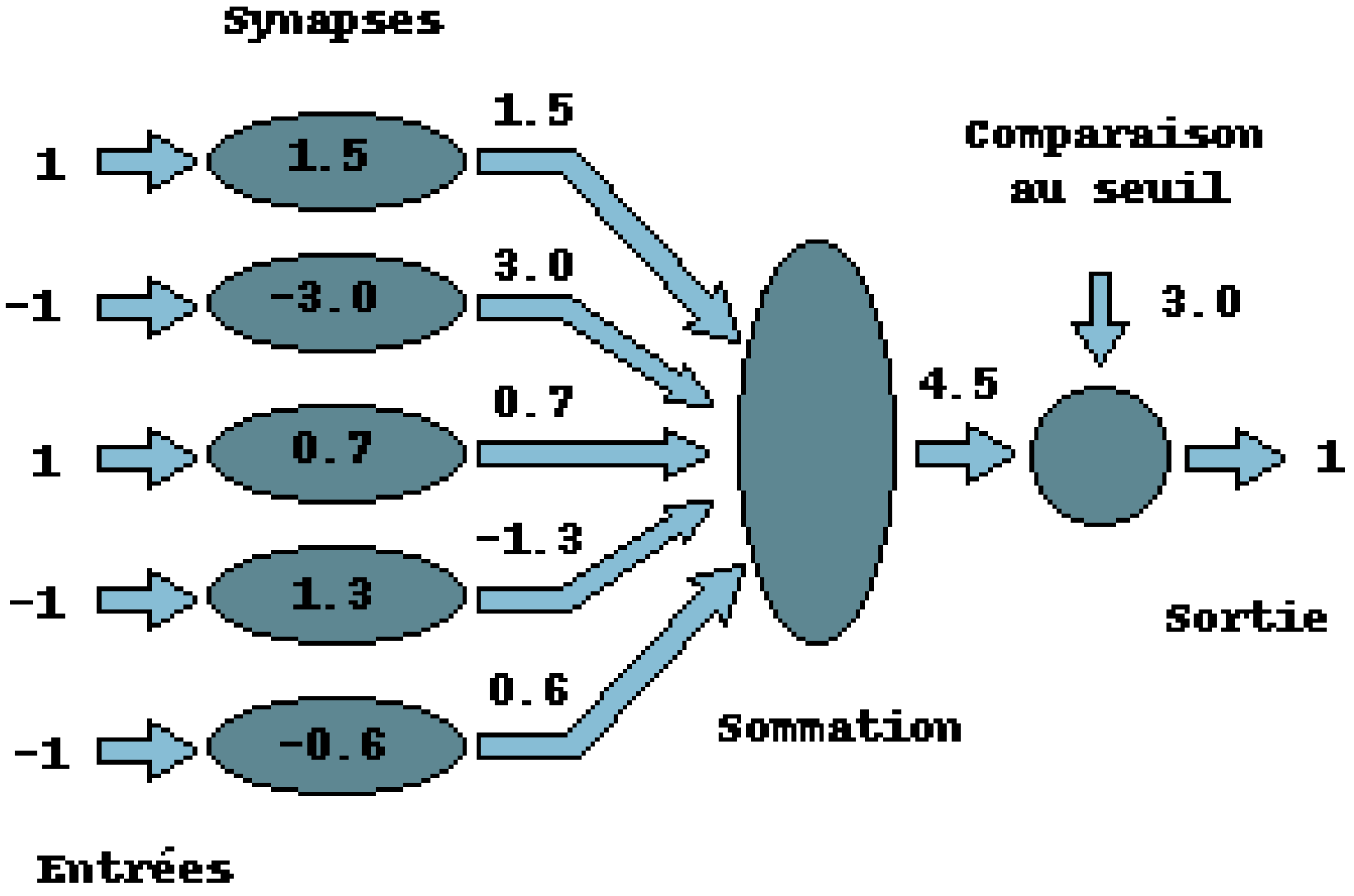

La transformation de l'information symbolique
a lieu au niveau des synapses sous forme
d'impulsions électriques quantifiées (numérisables)

# Inspiration Biologique

- Cerveau humain : 100 billions de neurones ($10^{11}$) et des centaines de types différents.

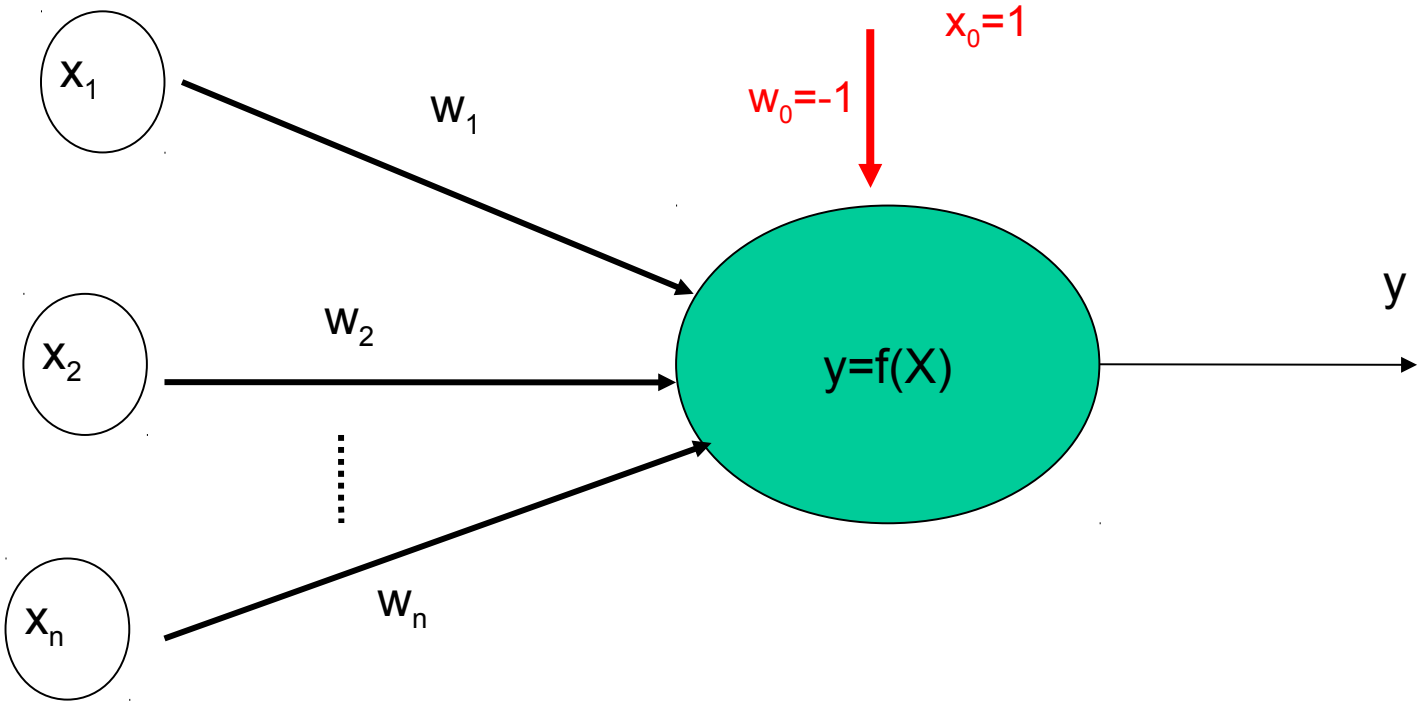
- Les neurones se rassemblent en couches, contenant chacune des milliers de neurones fortement interconnectés.

## Une modélisation simplifiée de la réalité




http://diwww.epfl.ch/mantra/tutorial/french/

# Le modèle de neurone artificiel de McCullogh et Pitts

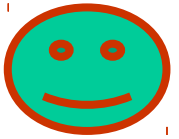



$$y = f\left(\sum_{i=1}^{n} w_i x_i - u\right) = f\left(\sum_{i=0}^{n} w_i x_i\right)$$

# Histoire

- 1943 - McCullock et Pitts : le problème

  [(x AND y) OR (x OR y)]



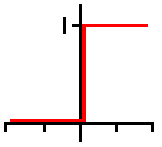
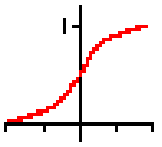
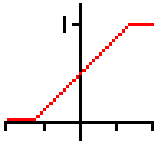
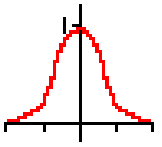
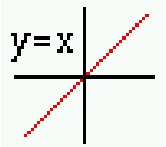
- 1950's - Hodgkin et Huxley : prix Nobel.
- 1969 - Minsky et Papert, *Perceptrons,* le problème non résolu : le XOR :

  [(x AND NOT y) OR (y AND NOT x)]

- 1987 - Robert Hecht-Nielsen

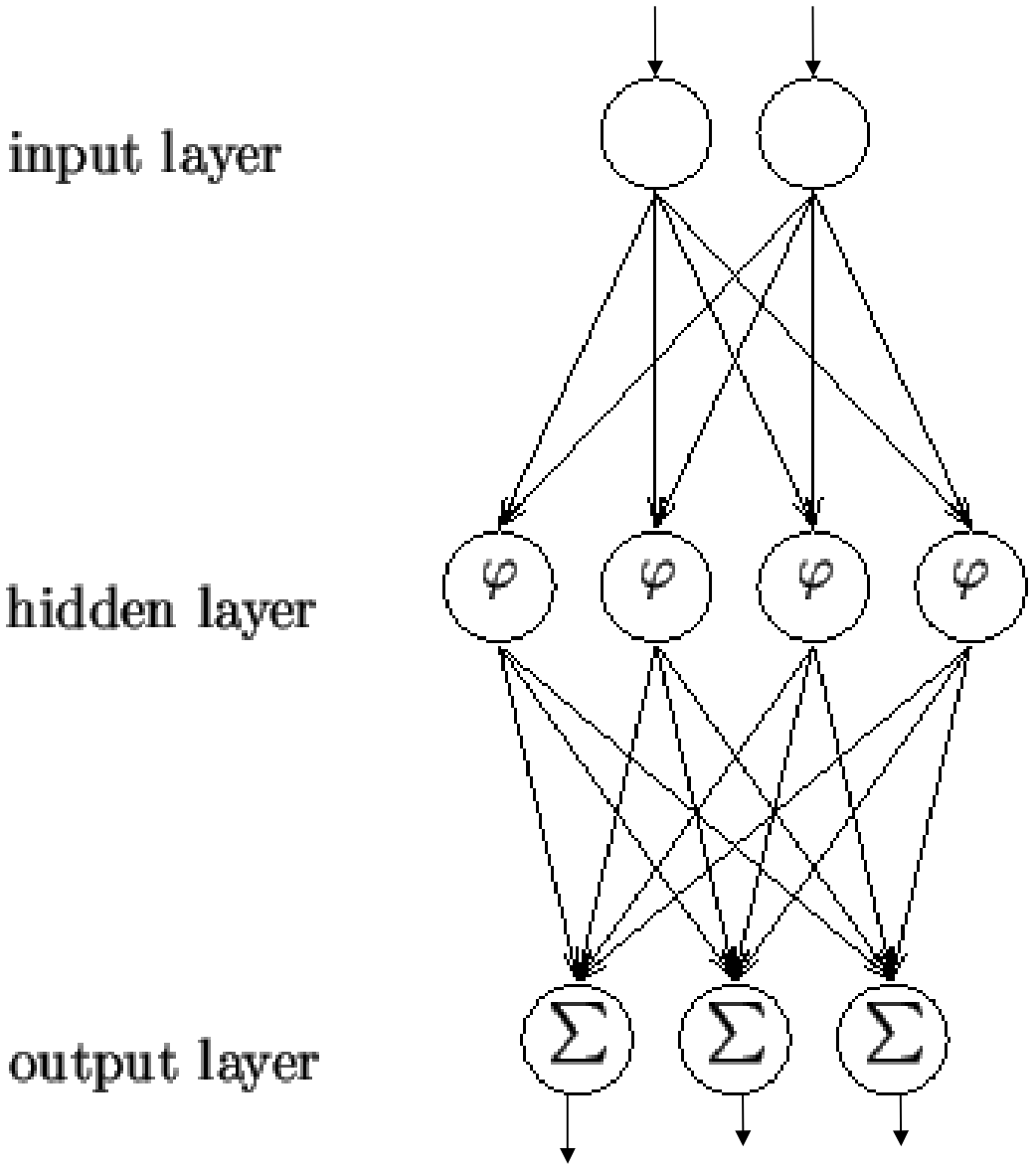
# Principe

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)



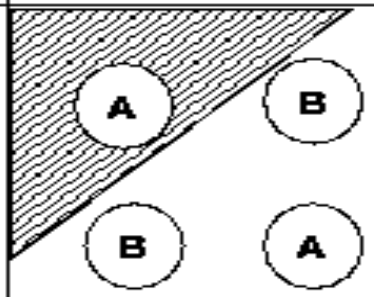
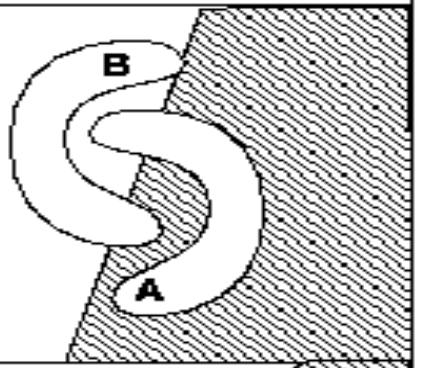
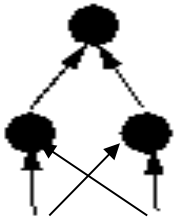
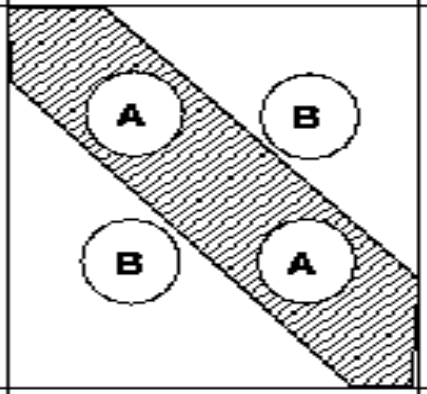
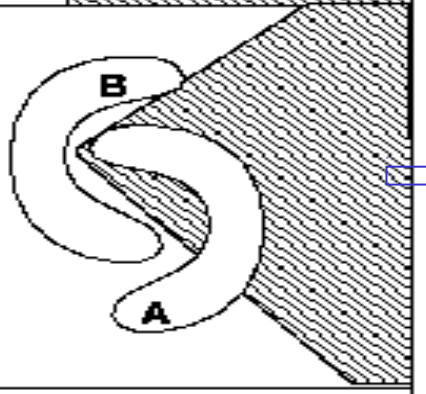
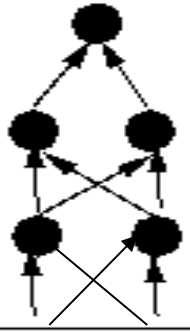
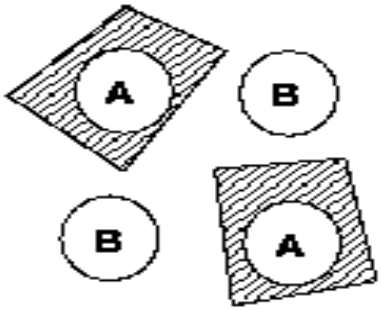
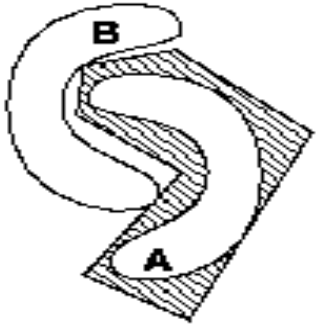

Fonctions d'activation :

Non linéaire

Linéaire

# Principe

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

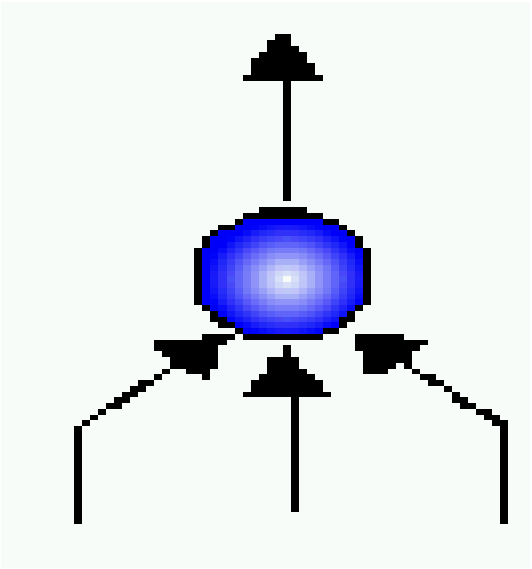
| Structure | Regions | XOR | Meshed regions |
|---|---|---|---|
| single layer  | Half plane bounded by hyper-plane |   |   |
| two layer  | Convex open or closed regions |   |   |
| three layer  | Arbitrary (limited by # of nodes) |   |   |

Une couche cachée

Deux couches cachées

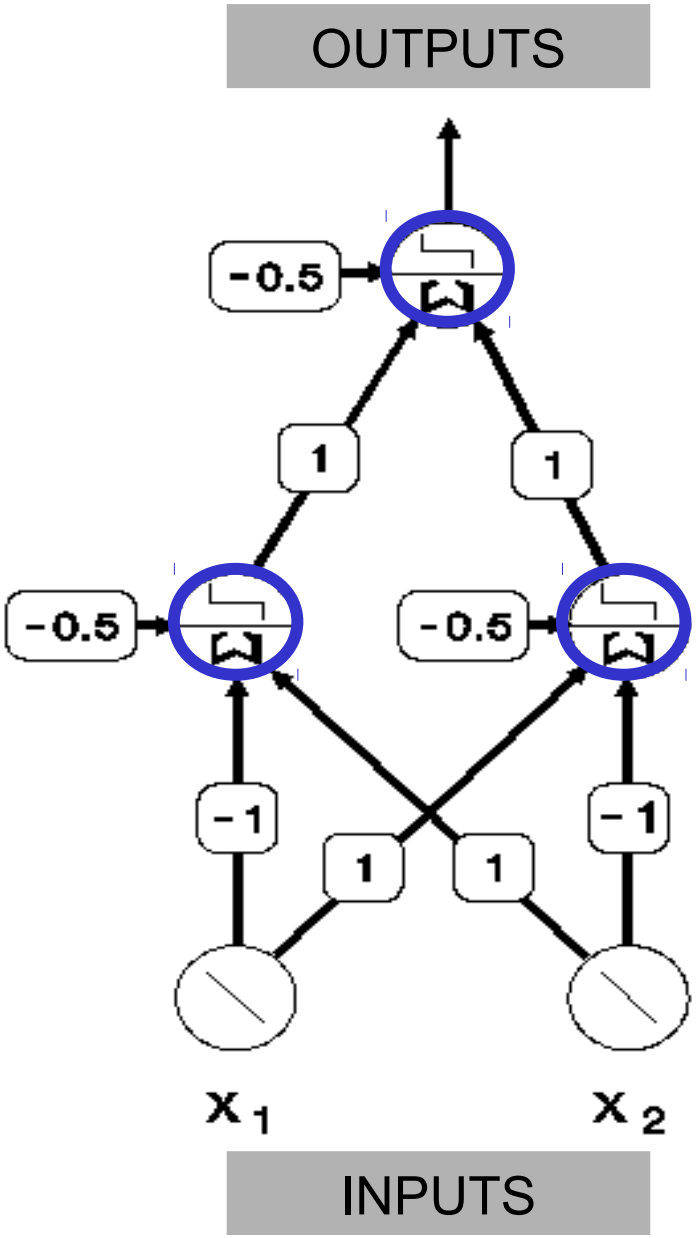
# Principe

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).



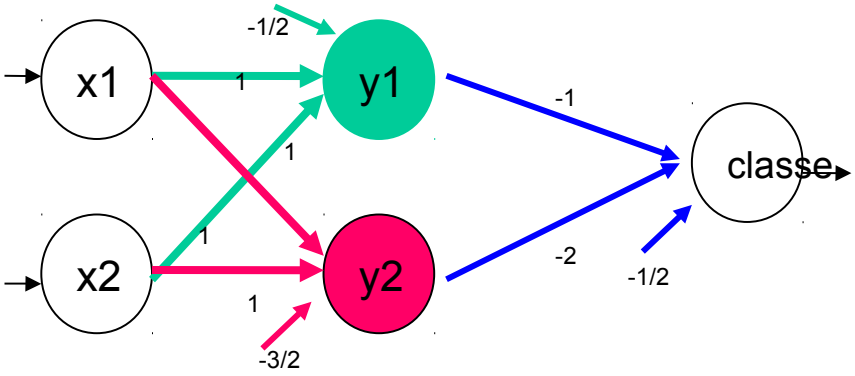
Le perceptron simple

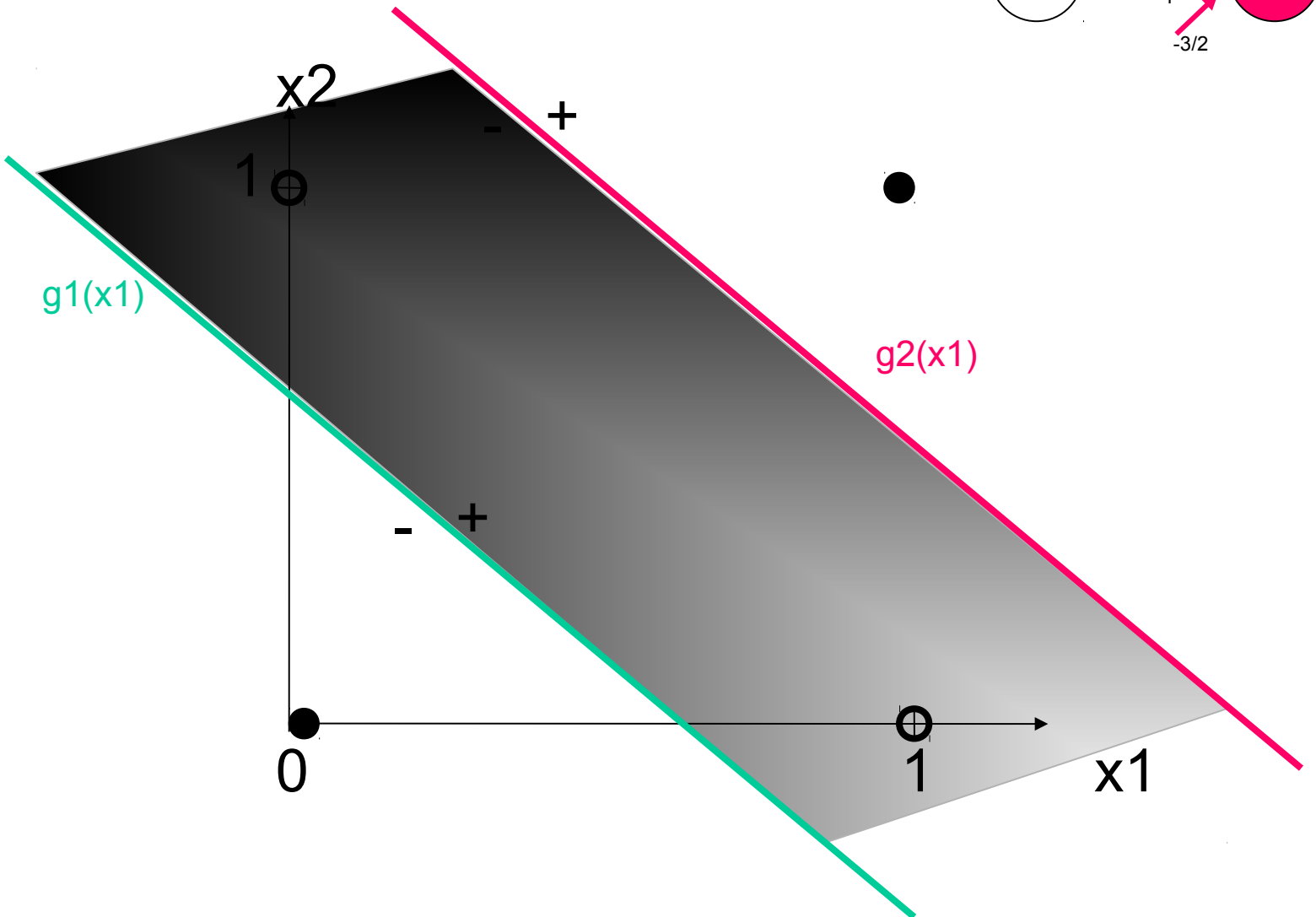
Le perceptron à 1 couche cachée



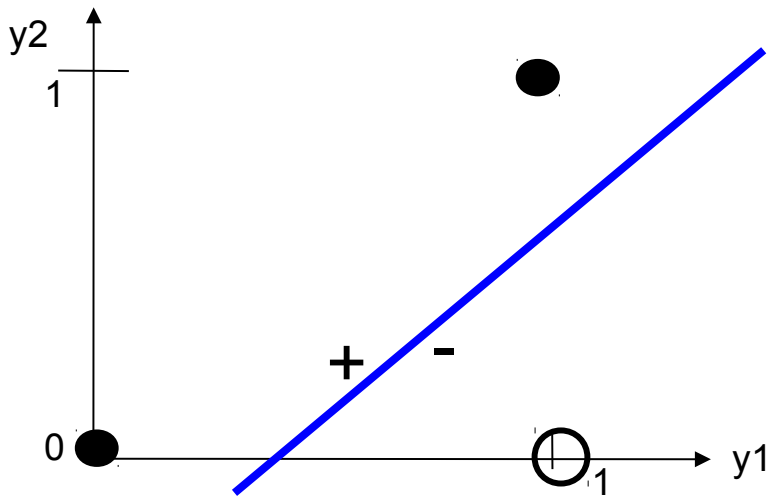
# Problème du XOR

| x1 | x2 | Classe |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |

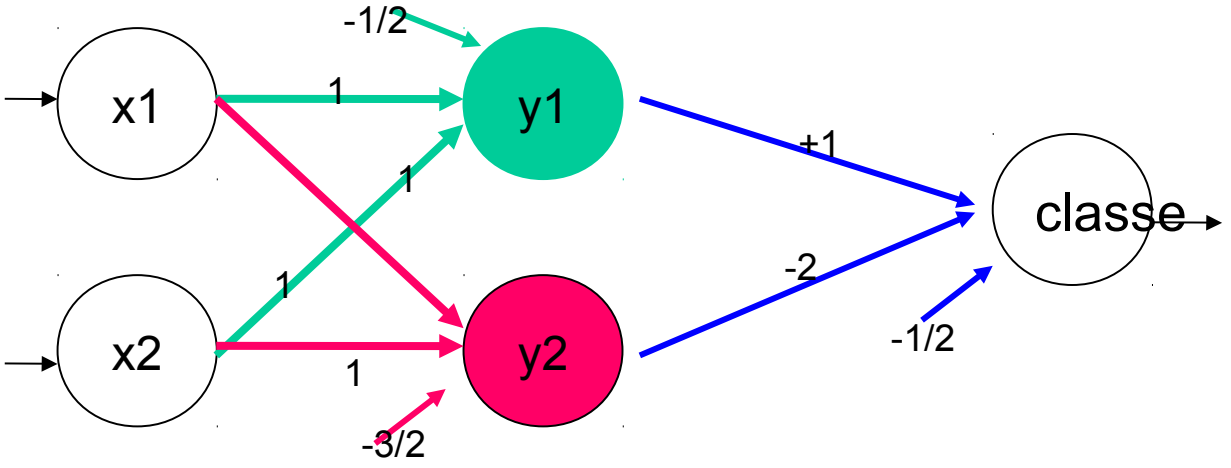


Projection dans un espace où le problème est linéairement séparable

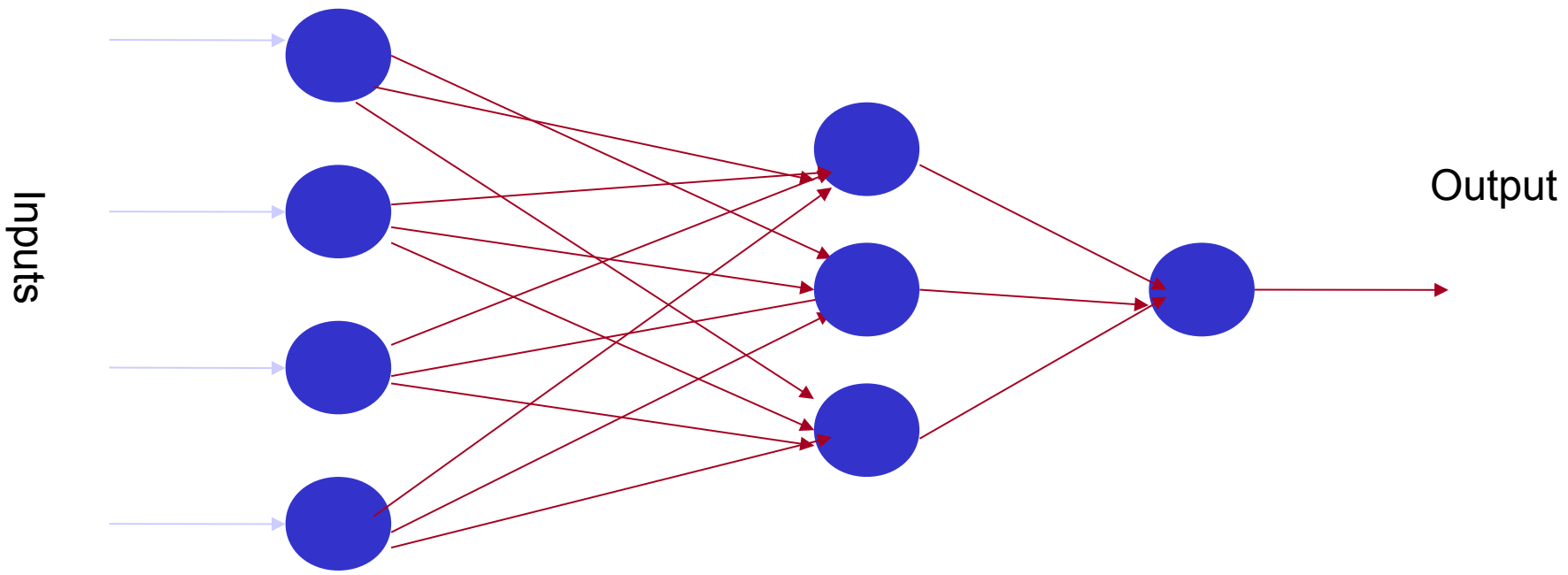



Réseau de neurones à une couche cachée.




Projection du repère (x1,x2) vers un hypercube de $R^2$ ! Quelle est la fonction d'activation dans ce cas ?

| x1 | x2 | y1 | y2 | Classe |
|---|---|---|---|---|
| 0 | 0 | 0(-) | 0(-) | 0 |
| 1 | 1 | 1(+) | 1(+) | 0 |
| 1 | 0 | 1(+) | 0(-) | 1 |
| 0 | 1 | 1(+) | 0(-) | 1 |




$$y_1^1 = f(x_1, w_1^1)$$
$$y_2^1 = f(x_2, w_2^1)$$
$$y_3^1 = f(x_3, w_3^1)$$
$$y_4^1 = f(x_4, w_4^1)$$

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

# 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.”

## 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 $x_0$, $x_1, \ldots x_{N-1}$ and specify the desired outputs $d_0$, $d_1, \ldots d_{M-1}$. 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 $y_0$, $y_1 \ldots y_{M-1}$.

**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, $\eta$ is a gain term, and $\delta_j$ 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

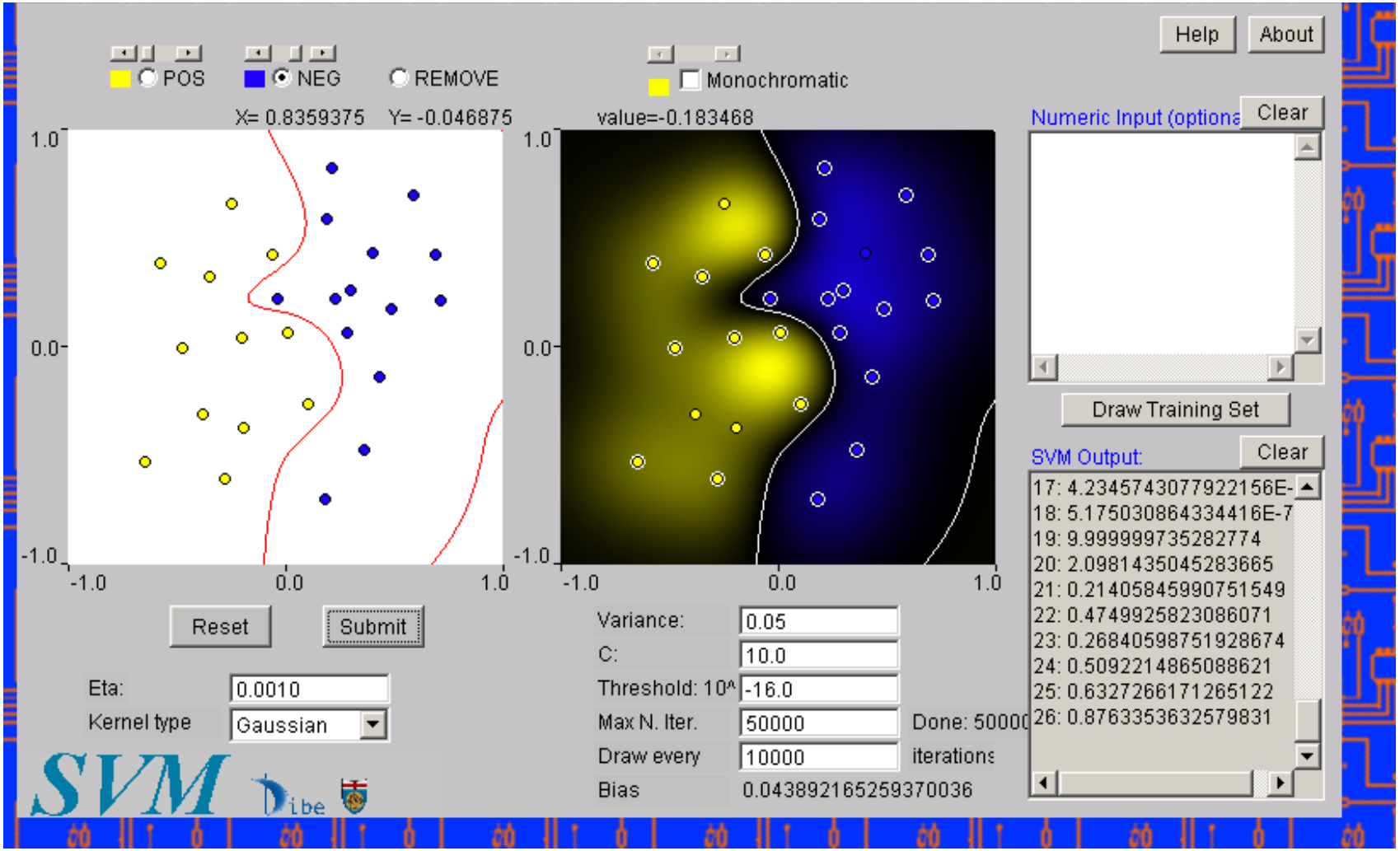
Nouvelle modélisation des *Neural Networks* :
les SVM ou Support Vector Machine

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

# Carte de Kohonen

Auto-organisation de données

https://www.dtreg.com/methodology
http://rgm3.lab.nig.ac.jp/RGM/R_rdfile?f=CMA/man/svmCMA.Rd&d=R_BC



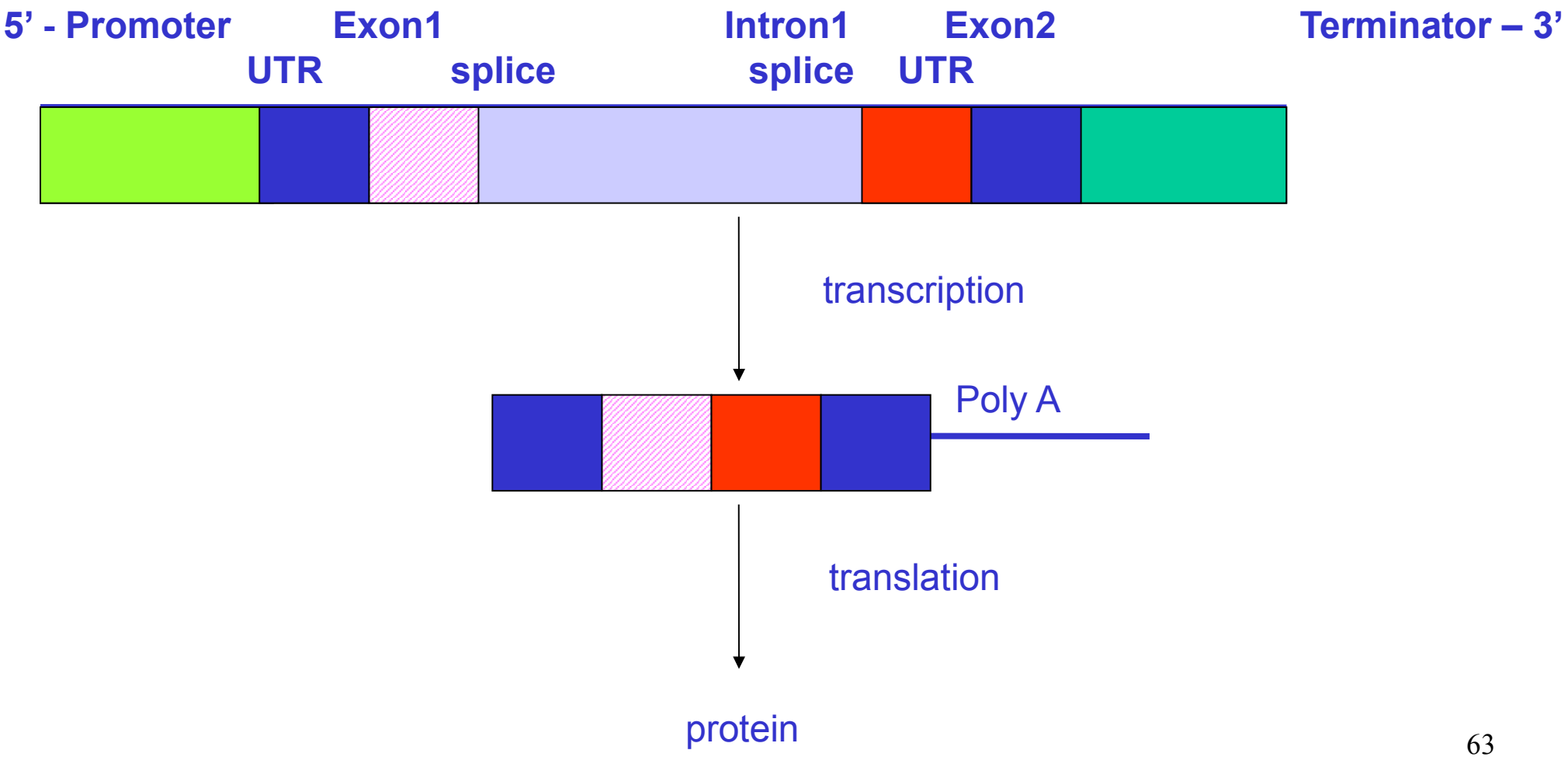
# Applications

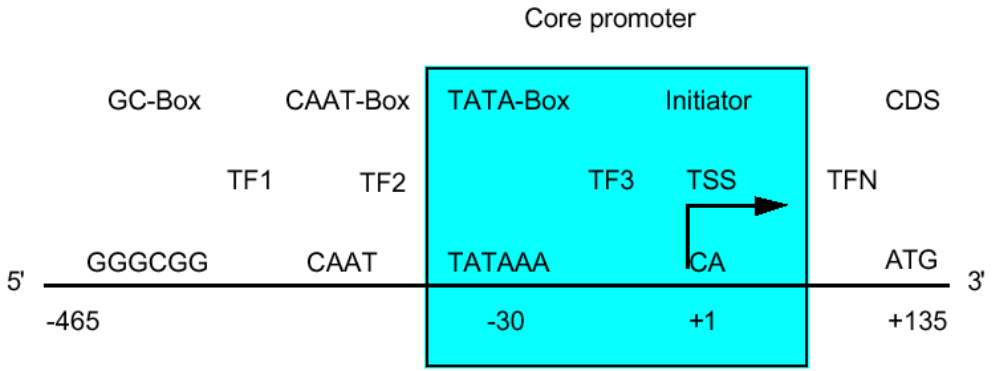
Tâches que peut résoudre un Réseau de Neurones Artificiels:

- 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

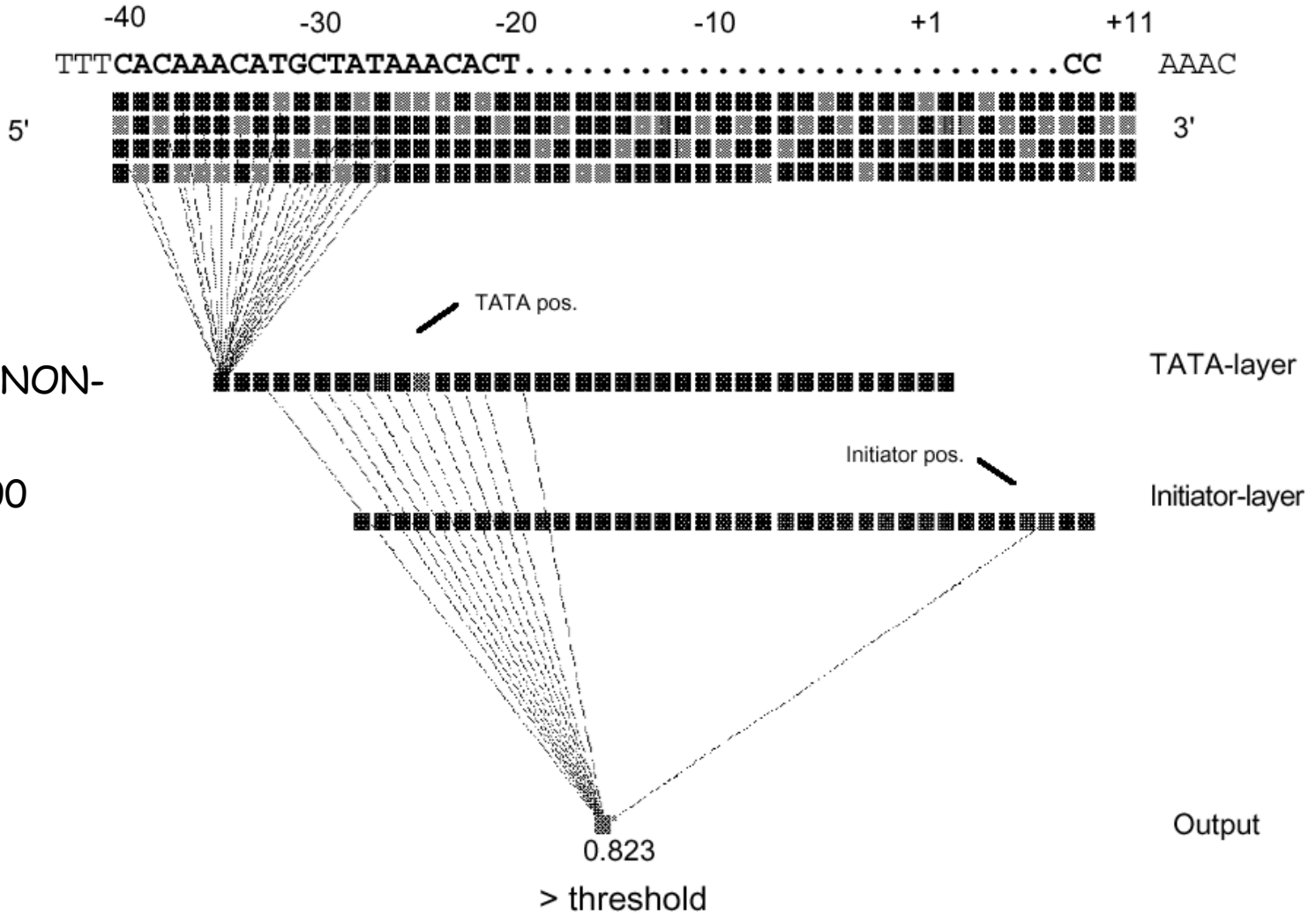





TDNN

(Time-delay

Neural Network)



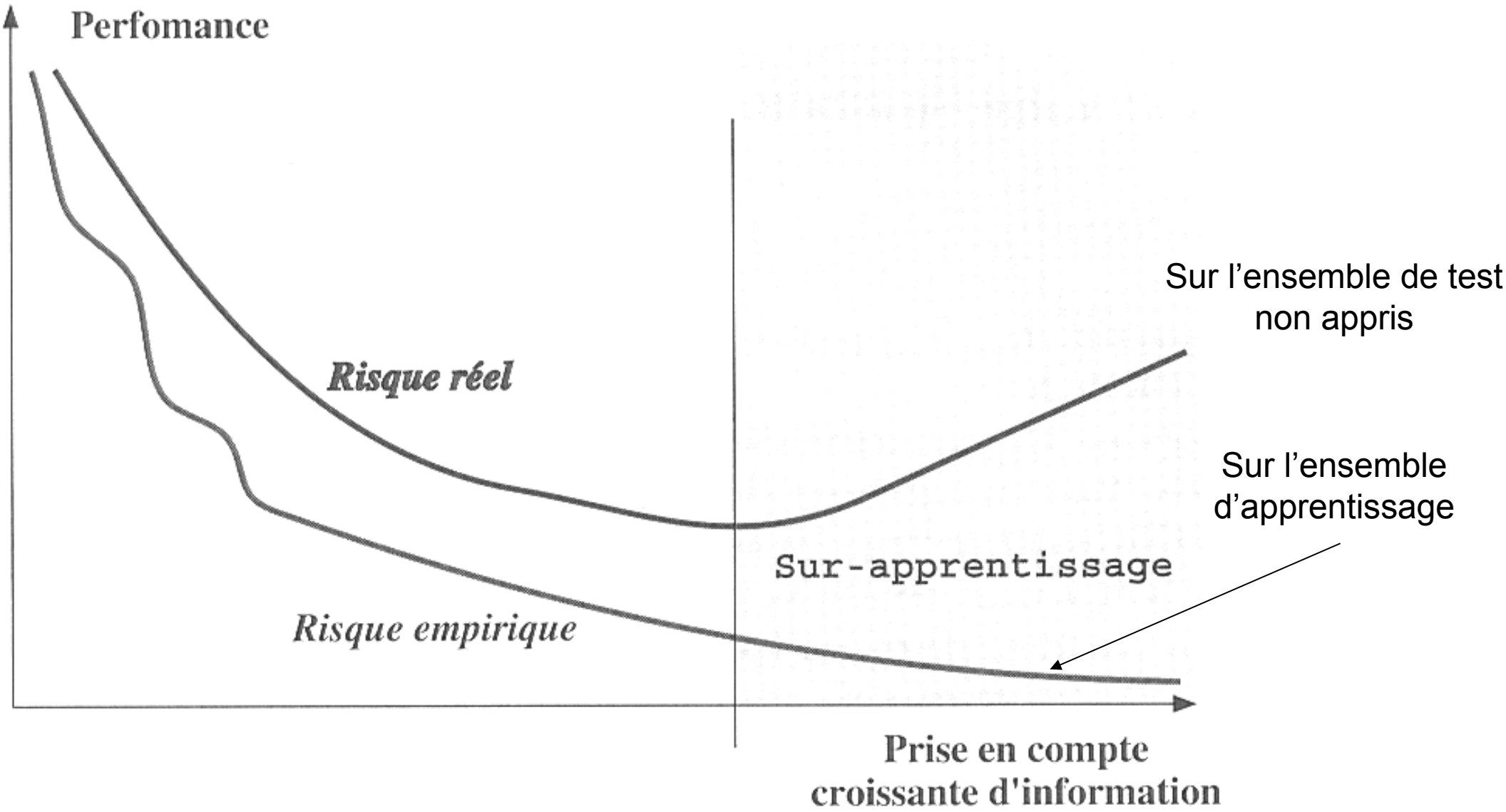

Training: 300 promoter sequences, 3000 random NON-promoter sequences.

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 négatifs** |
| **'-' réel** | **Faux positifs** | **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 Ni pour le test et apprendre sur les N-1 autres
- Mesurer le taux d'erreurs R(hi,Ni) sur Ni
- Recommencer n fois en faisant varier l'échantillon i de 1 à N

L'erreur estimée finale est la moyenne des R(hi,Ni) pour i de 1 à N.

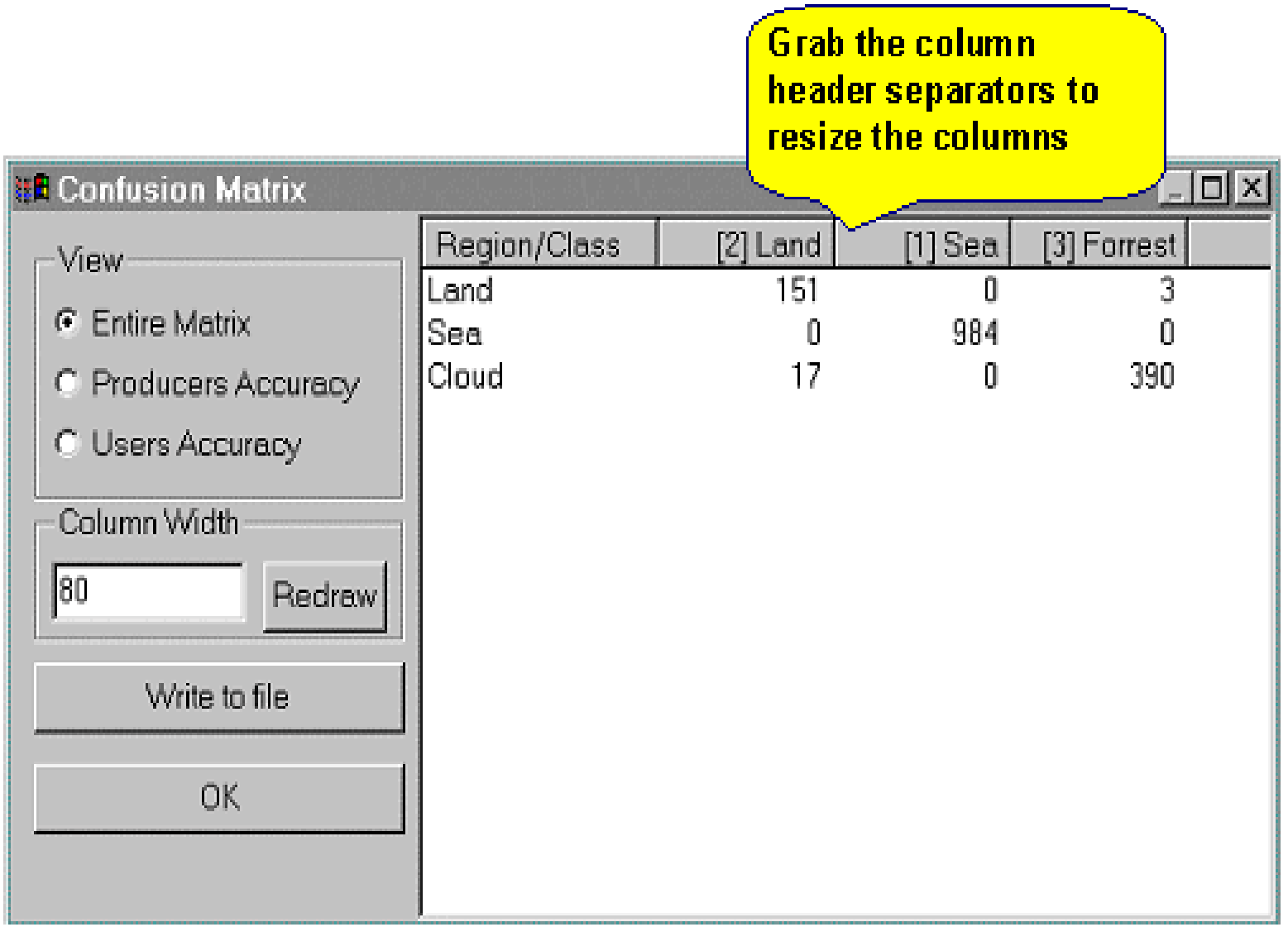
Souvent N varie entre 5 et 10.
On refait souvent un apprentissage global sur A tout entier (plutôt que de choisir une des hypothèse hi). Mais la procédure précédente est utile pour avoir une bonne mesure de la validité ou du taux d'erreur de la méthode d'apprentissage choisie.

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



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

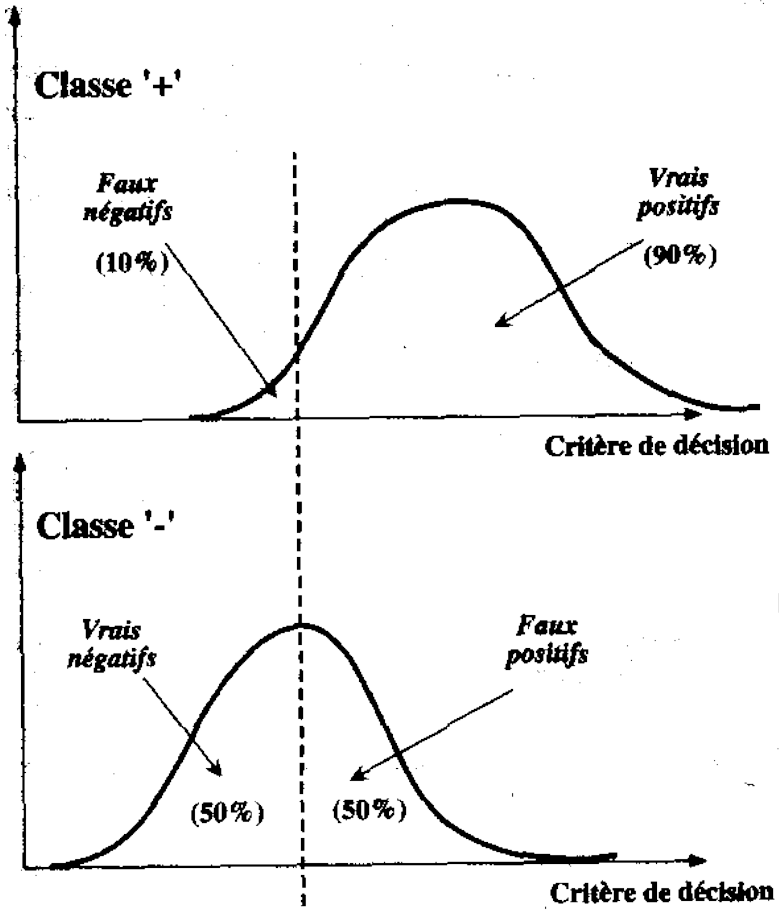
Dans un contexte de prise de décision, la performance intrinsèque en terme de taux d'erreur indifférencié n'est pas suffisante.

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.

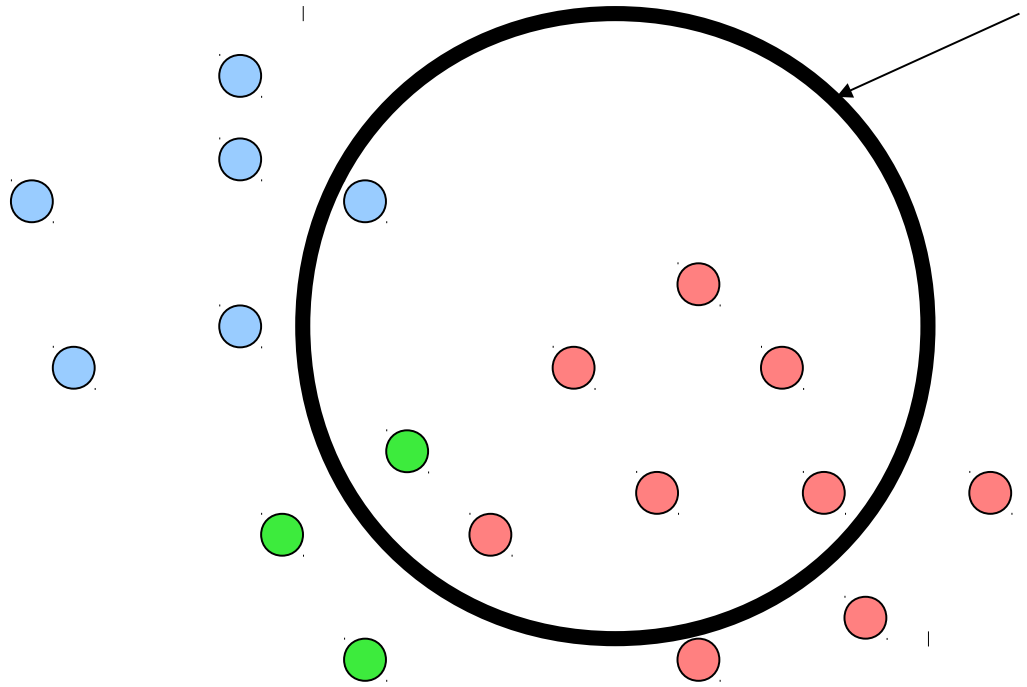
La courbe ROC est utilisée dans le cadre de classification à 2 classes.

Remarque : classe = hypothèse = test




| | '+' | '-' |
|---|---|---|
| '+' | Vrais positifs | Faux négatifs |
| '-' | Faux positifs | 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.

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

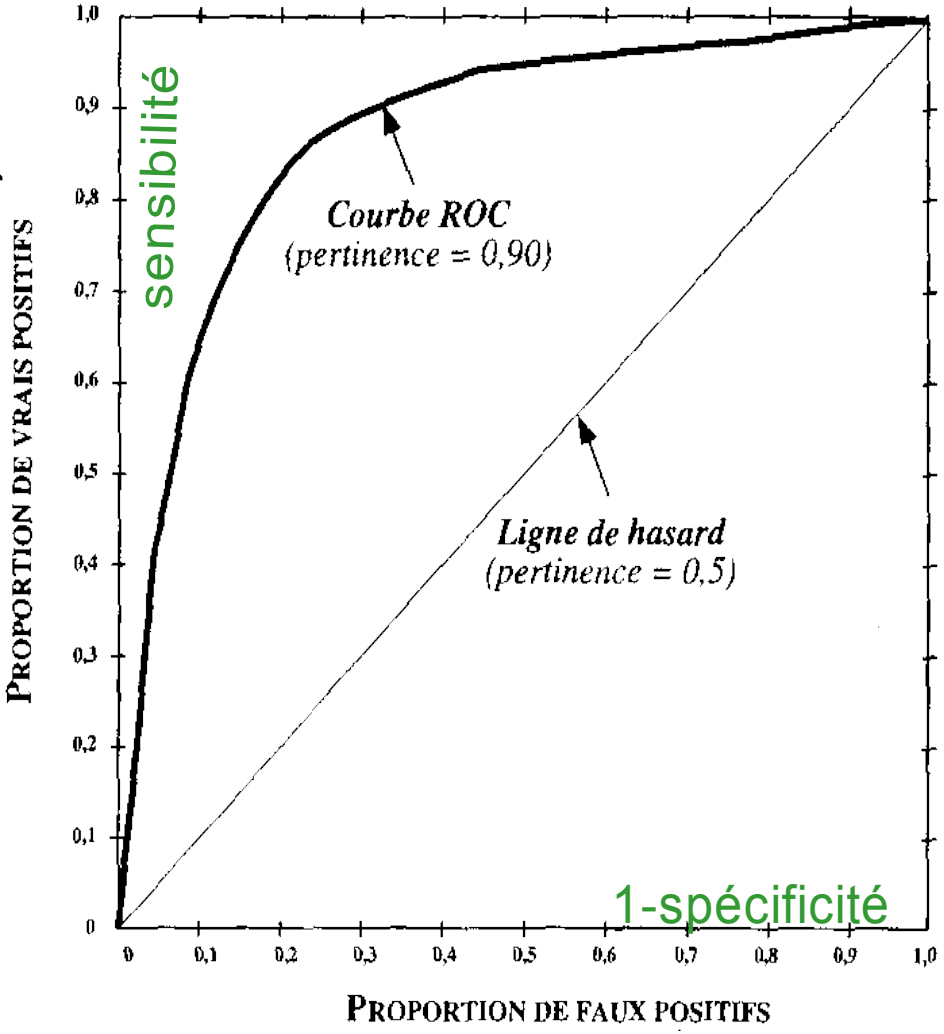
$$Recall = \frac{(6\,roses\,dans\,cercle\,noir)}{(6+3\,roses\,hors\,du\,cercle\,noir)}$$

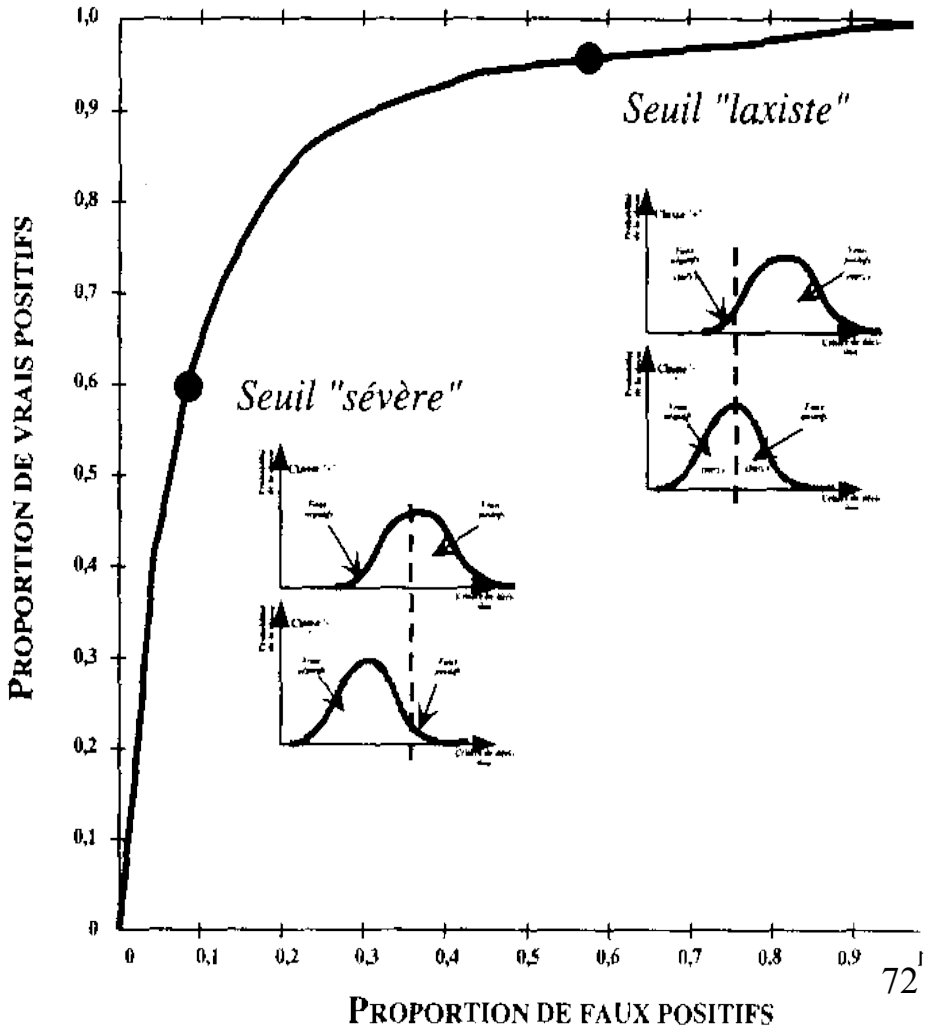
Sensibilité = taux de détection ou reconnaissance = Recall =TP/(TP+FN)=TP/{Exemples positifs}

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

Taux de fausses alarmes = FP/(TN+FP)=1-spécificité

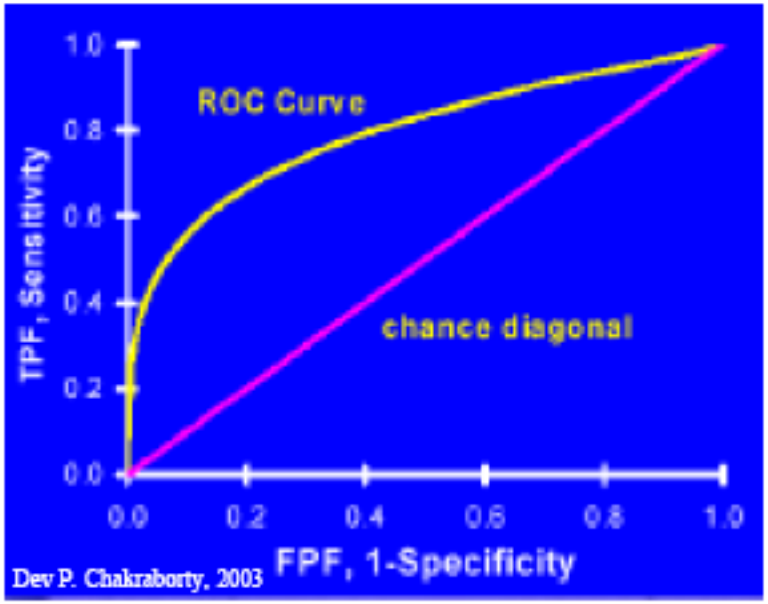
Précision = TP/(TP+FP) (utilisée en médecine essentiellement)



# 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

classificateur

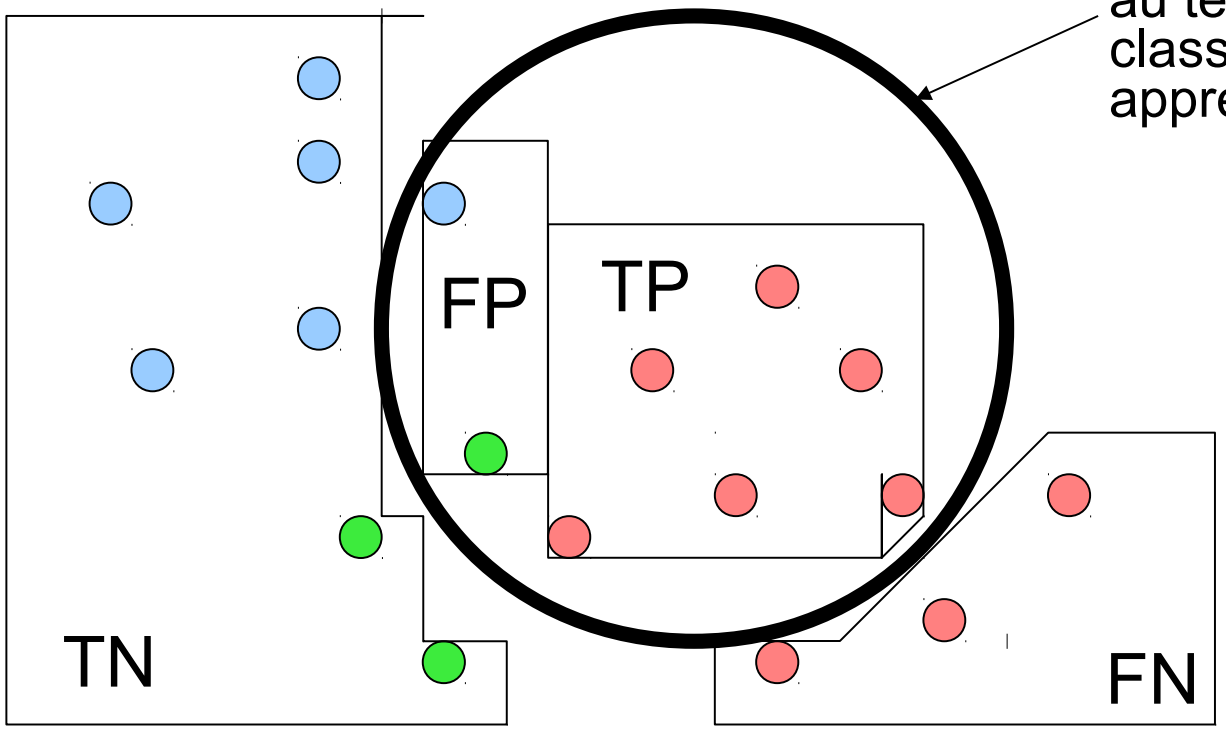
Vérité

| | signal | pas de signal |
|---|---|---|
| signal | Vrai positif (TP) | Faux positif (FP) |
| pas de signal | Faux négatif (FN) | Vrai négatif (TN) |

# Summary of measures

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

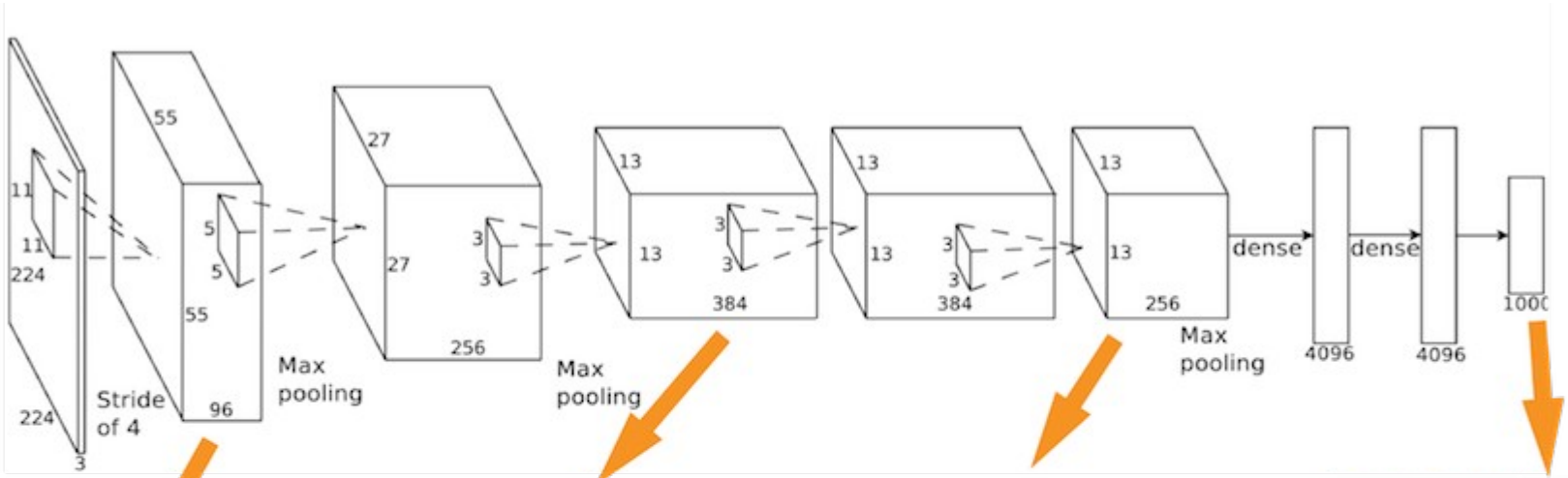
# Mesures de la qualité de prédiction d'une classe par rapport à toutes les autres

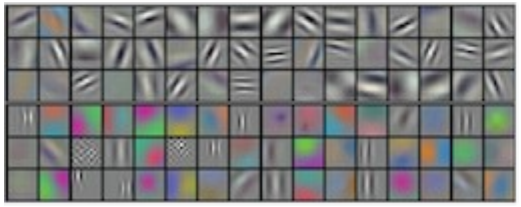



Ensemble des **N** éléments déclarés roses en phase de décision (test), c'est-à-dire qui répondent positivement au test (hypothèse) de la classe rose après apprentissage.

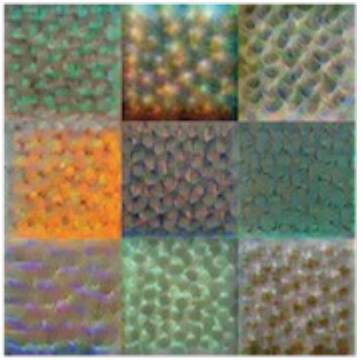
# Deep learning ??

http://www.computervisionblog.com/2015/11/the-deep-learning-gold-rush-of-2015.html



Conv 1: Edge+Blob



Conv 3: Texture

Numerical

Data-driven

Conv 5: Object Parts


Fc8: Object Classes

https://medium.com/deep-math-machine-learning-ai/chapter-8-0-convolutional-neural-networks-for-deep-learning-364971e34ab2