Cas de 2 classes :

Méthode itérative historique :

Algorithme du Perceptron

INPUT : $w_{(0)}$ et α positifs quelconques

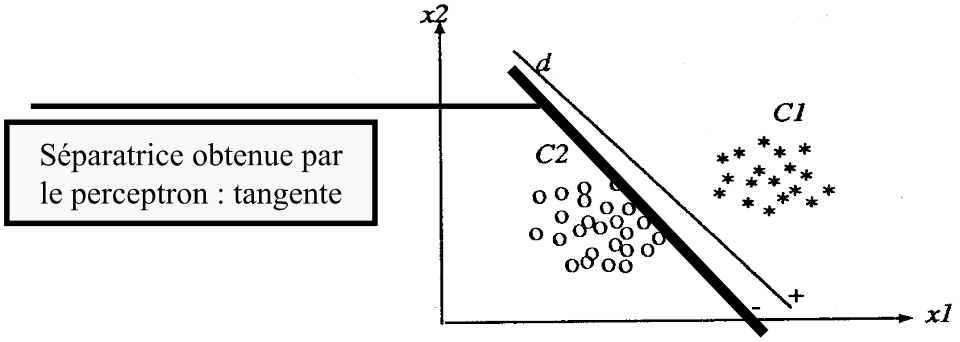
- $t \leftarrow 0$
- Tant que $t \leq t_{max}$ faire
  - $modif \leftarrow 0$
  - Pour *chaque donnée d'apprentissage x* faire
    - Si x est mal classé alors
      - Si $x \in C_1$ alors
        - $modif \leftarrow modif + \alpha x$
      - Sinon
        - $modif \leftarrow modif - \alpha x$
      - Fin si
    - Fin si
  - Fin pour
  - $w_{(t+1)} \leftarrow w_{(t)} + modif$
  - $t \leftarrow t + 1$
- Fin tant que

une époque d'apprentissage : cumul des contributions de chaque exemple de S

OUTPUT : l'hyperplan optimal W

- Convergence,mais lente et loin de l 'optimum

$$W_{t+1}Y_{t+1}^{T} = W_t Y_{t+1}^{T} + \left\| Y_{t+1} \right\|^2$$

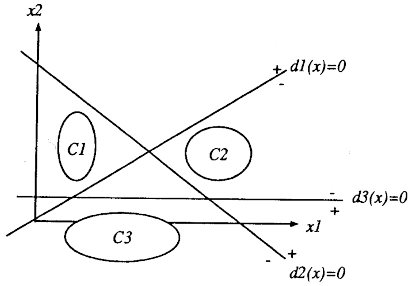


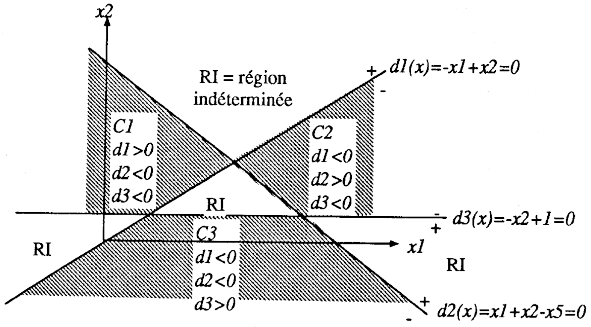
- Minimisation de $J(W) = -\sum_{x \in S \quad mal\ classés\ par\ W} W^T x$

- La base S peut être parcourue plusieurs fois
- Version stochastique mais meilleure généralisation de Ho et Kashyap

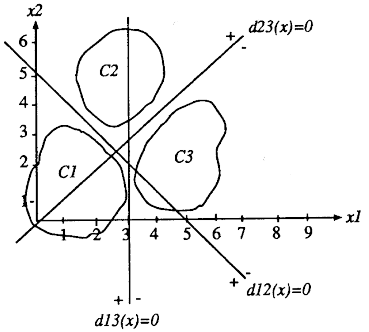
Cas général : M classes C_1, C_2, …C_M

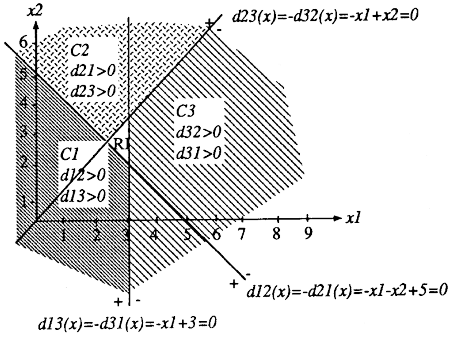
Chaque classe est séparée des autres classes par un hyperpan : au total M hyperplans



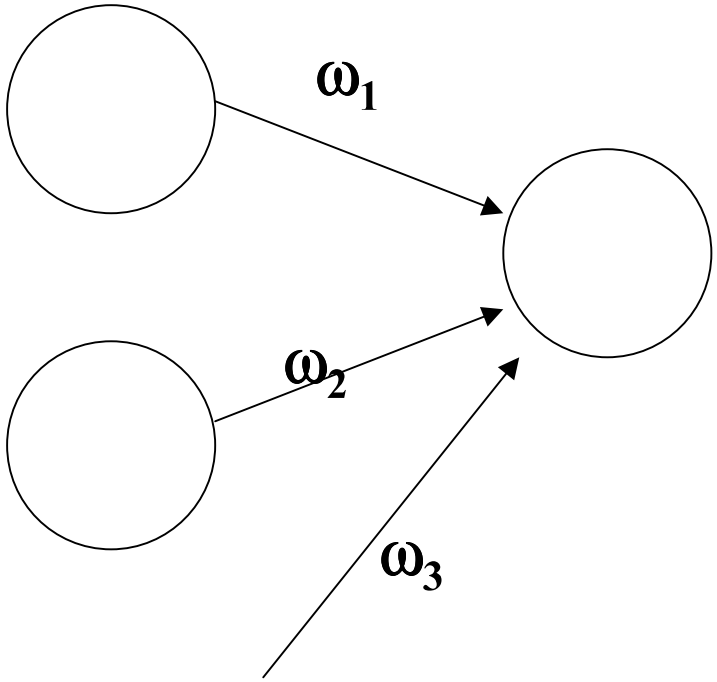


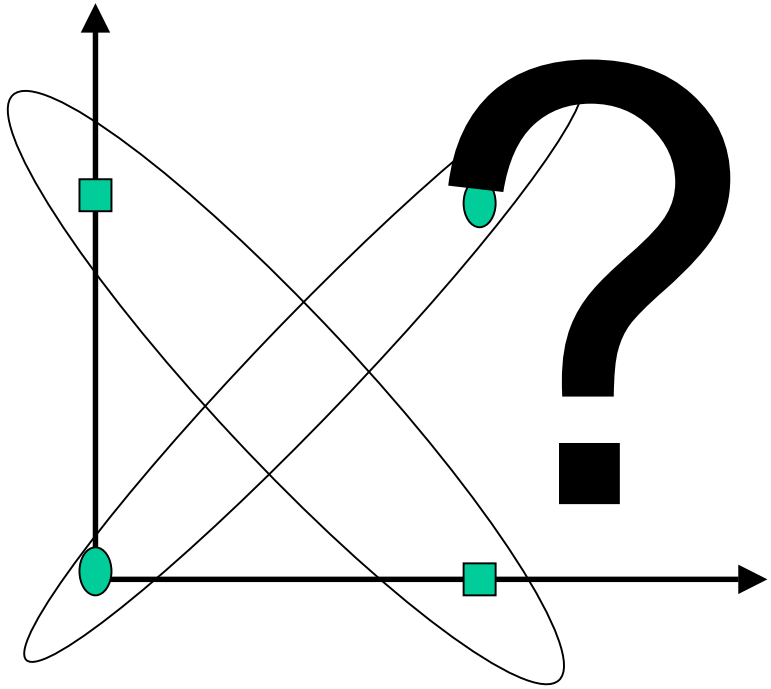
Les classes sont séparées deux à deux : au total M(M-1)/2 hyperplans





Lien avec les réseaux de Neurones





Echec Sur le Problème du XOR

- Neurones Formels et Réseaux de Neurones :

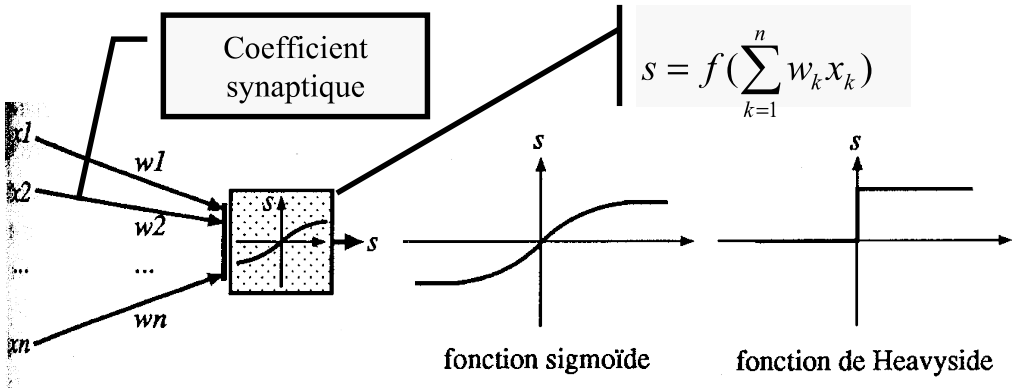
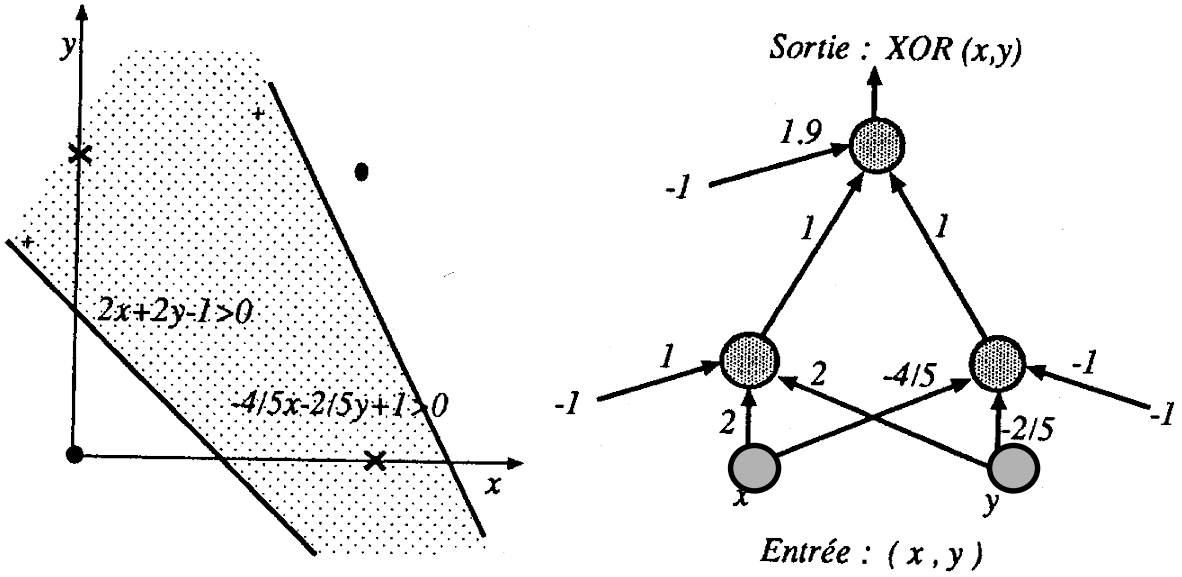


Figure 5.35. *Neurone formel et deux fonctions de seuillage.*

- Un Neurone Formel effectue une discrimination linéaire
- Un réseau de Neurones effectue une discrimination linéaire par morceaux

Succès sur le Problème du XOR avec une couche cachée :



Divers Points de Vue sur les Réseaux de Neurones :

- Point de Vue Informatique :

  Parallélisme à grains très fins;

- Point de Vue de la Classification :

  Discriminateur Linéaire Complexe;

Avantages Principaux :

- Inclus dans son traitement Analyse et Prétraitement
- Grande Capacité Autonome d 'Apprentissage

➡ *Algorithme de Rétropropagation*

- Problème de la Généralisation
- Extension non linéaire et bien fondée théoriquement donnée par les SVM : Séparateurs à Vastes Marges ou Support Vector Machines -> kernel-based machine learning

- Modèle Probabiliste de Représentation des Formes ?
- Assure en théorie le Minimum d'Erreur en Classification

Propriété :

La règle de décision bayésienne est la règle de décision optimale au sens où elle minimise le risque réel en utilisant l'information disponible de manière optimale.

Etant donné que l'on m'a fourni un échantillon de données, comment cela doit-il modifier mes croyances antérieures sur le monde ?

Règle de Bayes de révision des probabilités :

- x : représentation de la forme, ω : classe

$$P(x/\omega) = \frac{P(x,\omega)}{p(\omega)}$$

$$p(\omega/x) = \frac{P(x,\omega)}{P(x)}$$

Théorème de Bayes

mesurable — estimable

$$p(\omega/x) = \frac{P(x/\omega)\,p(\omega)}{P(x)}$$

Difficile à calculer

Probabilité *a posteriori*

Probabilités *a priori*

Cas de 2 classes :

ω_1 = *oiseau* et ω_2 = *canard* après observation de la variable x=*couleur_aile*

$$p(oiseau = canard / couleur\_aile = noire) = \frac{P(couleur\_aile = noire / oiseau = canard)\, p(oiseau = canard)}{P(couleur\_aile = noire)}$$

Risque bayésien :

Le but de l'agent est de prendre une décision *(faire un diagnostic vital par exemple)* minimisant l'espérance de risque.

On définit une fonction de décision S : X ->H, où H est vu comme un ensemble de décisions à prendre.

Dans le cas de N classes, H peut comprendre les N classes possibles Ω en décision de classification plus une classe de rejet quand l'incertitude est trop forte et ne permet pas de prendre de décision.

Avant toute observation sur le monde, et en prenant seulement en compte les connaissances *a priori*, l'espérance de risque associée à une décision *h* peut s'écrire : $R(h) = \sum_{\omega \in \Omega} l(h|\omega)\, p(\omega)$

où $p(\omega)$ dénote la probabilité *a priori* que le monde soit dans l'état ω, tandis que $l(h|\ \omega)$ est le coût ou perte encouru lorsque la décision h est prise alors que l'état du monde est ω.

En général,

$$l(h|\omega) = \begin{cases} 0 & si \quad h = \omega \ \ (décision\ correcte) \\ 1 & si \quad h \neq \omega \ \ (décision\ incorrecte) \\ r & si \quad h = rejet\ (doute\ trop\ important) \end{cases}$$

Mais , le coût de ne pas diagnostiquer à tort une tumeur est bien plus élevé que de faire un faux diagnostic. Le coût de la décision incorrecte est ajustable.

Principe de la Décision Bayésienne :

⇨ Choisir pour l'observation x la classe ω qui minimise l'espérance de risque.

$$h^* = \underset{h \in H}{ArgMin} \sum_{\omega \in \Omega} l(h|\omega)\, p(\omega) P(x|\omega)$$

Principe de la Décision Bayésienne : cas particuliers

Règle du Maximum A Posteriori (MAP) :

Lorque les coûts de mauvaise classification sont égaux, la règle de décision bayésienne de risque minimal devient la règle du MAP :

$$h^* = \underset{h \in \Omega}{ArgMax}\ p(\omega)P(x|\omega)$$

Remarque : cette règle minimise le nombre d'erreurs en classification

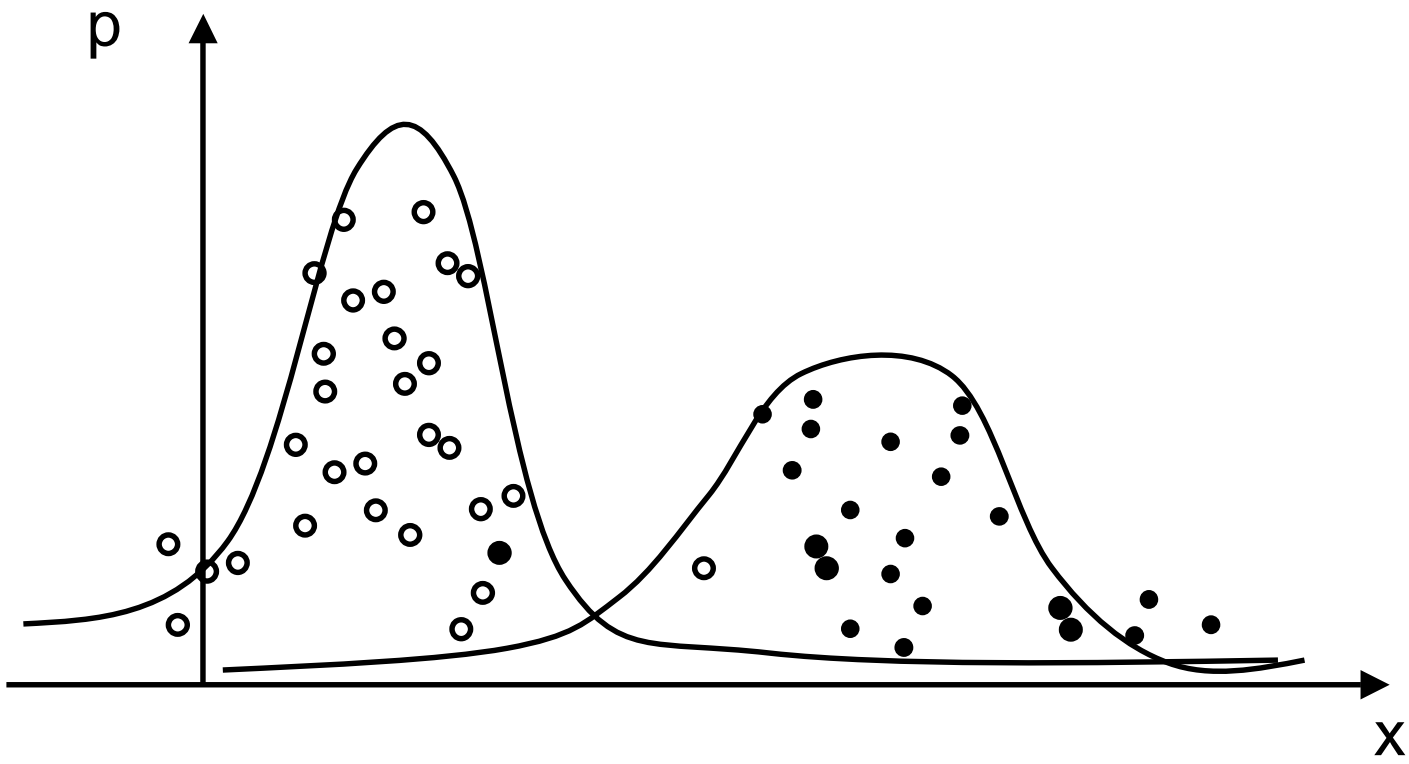
Principe de la Décision Bayésienne : cas particuliers

Règle du Maximum de Vraisemblance Maximum Likelihood (ML) :

Si de plus toutes les hypothèses ont la même probabilité a priori, alors la règle du MAP devient la règle du Maximum de Vraisemblance :

$$h^* = \underset{h \in \Omega}{ArgMax}\, P(x|\omega)$$

Principe de la Décision Bayésienne : cas particuliers



Principe de la Décision Bayésienne : cas particuliers

Nous supposons maintenant que la tâche d'apprentissage consiste à discriminer les formes observées x en 2 classes : H = Ω = $\{\omega_1, \omega_2\}$
Etant donnée l'observation x, les espérances de risque associés à chaque décision sont respectivement en notant $l(\omega_i|\omega_j)=l_{ij}$:

$R(\omega_1)=l_{11}\, p(\omega_1 |x) + l_{12}\, p(\omega_2 |x)$
$R(\omega_2)=l_{21}\, p(\omega_1 |x) + l_{22}\, p(\omega_2 |x)$

La règle de Bayes stipule de choisir l'hypothèse d'espérance de risque minimal. Il faut donc atrribuer la forme x à la classe ω_1 ssi :

$(l_{21}-l_{11})\, p(\omega_1 |x) \geq (l_{12}-l_{22})\, p(\omega_2 |x)$,

Que l'on peut écrire en appliquant la formule de Bayes de révision des probabilités :

$(l_{21}-l_{11})\, p(x|\, \omega_1)\, p(\omega_1) \geq (l_{12}-l_{22})\, p(x|\, \omega_2)\, p(\omega_2)$, *soit :*

$$d(x) = \log \frac{p(x|\omega_1)}{p(x|\omega_2)} + \log \frac{(l_{21}-l_{11})\, p(\omega_1)}{(l_{12}-l_{22})\, p(\omega_2)} \geq 0$$

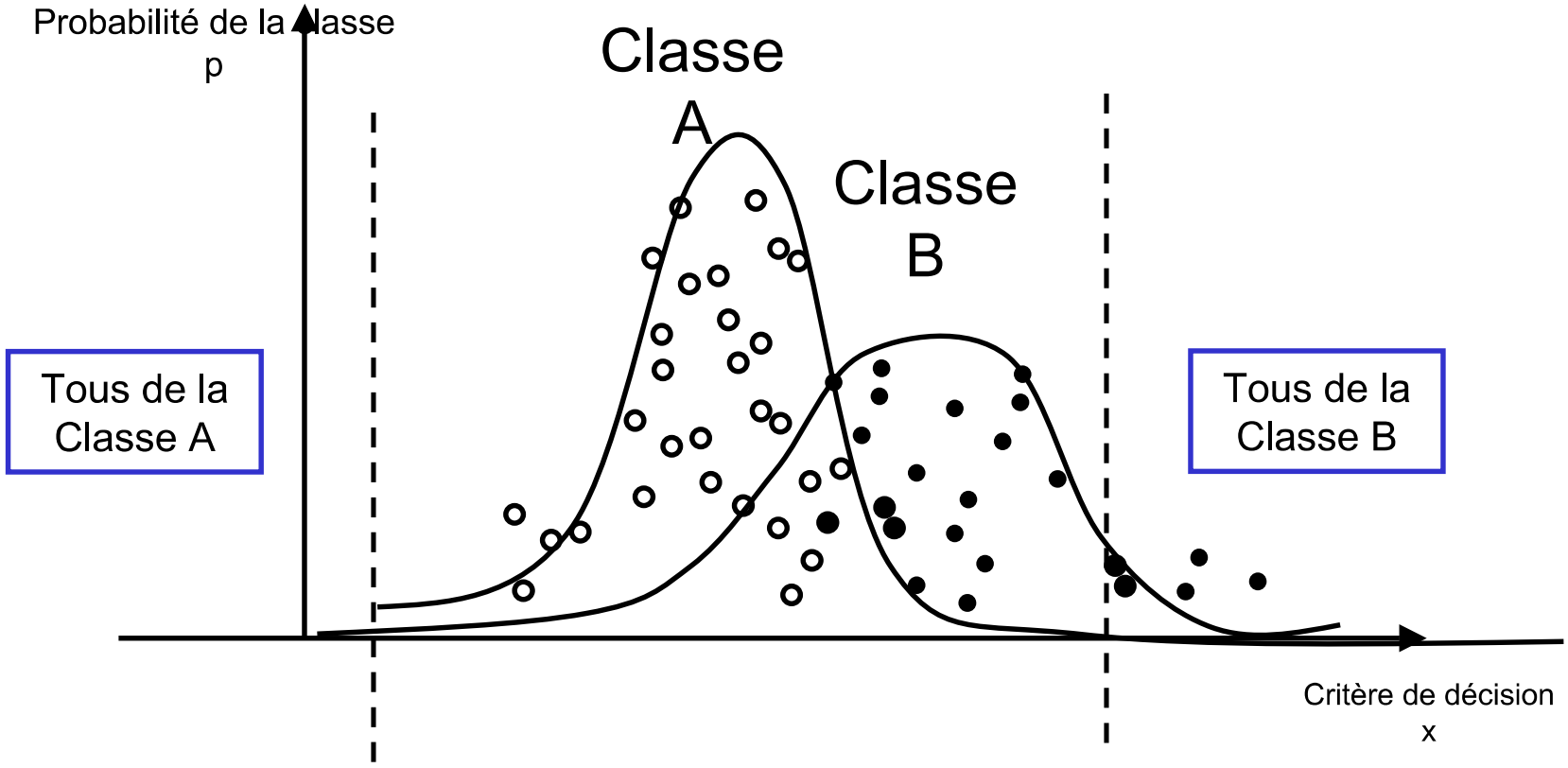
Principe de la Décision Bayésienne : cas particuliers

La règle de décision bayésienne se traduit ainsi par une fonction de discrimination ou de décision d décrivant une frontière ou surface de décision dans l'espace X.

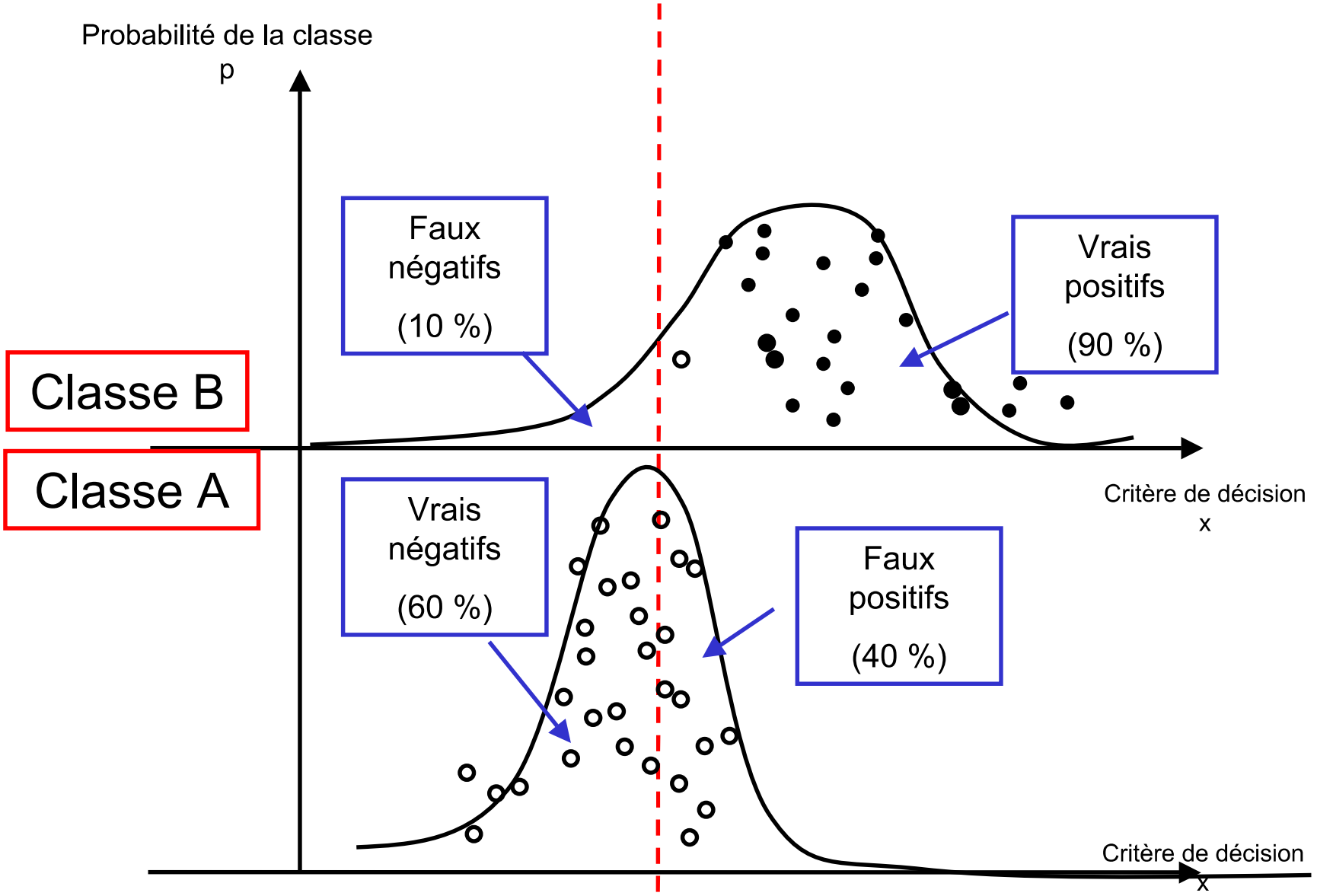
On peut donc essayer d'apprendre directement cette frontière de décision plutôt que les probabilités, plus difficiles à estimer -> voire les méthodes de classifications fonctionnelles et le connexionnisme.
Dans le cas particulier de la discrimination entre deux classes, de distribution normale gaussienne de moyennes μ_1 et μ_2 avec des matrices de covariances égales Σ, la fonction de décision *d(x)* est une fonction linéaire :

$$d(x) = (x - \frac{1}{2(\mu_1 - \mu_2)})^T \Sigma^{-1} (\mu_1 - \mu_2) + \log \frac{(l_{21} - l_{11}) p(\omega_1)}{(l_{12} - l_{22}) p(\omega_2)}$$

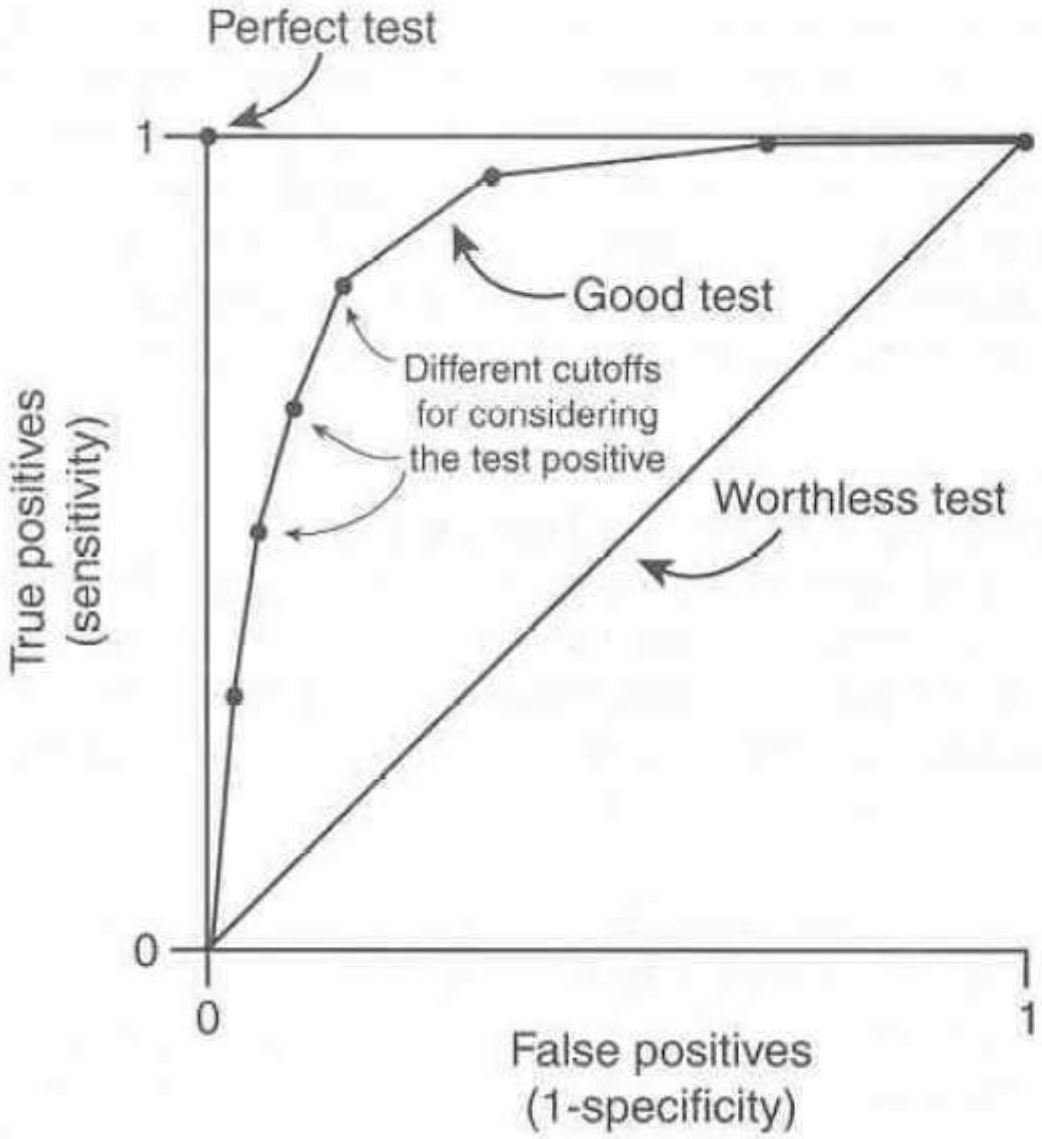
Validation de l'apprentissage



Validation de l'apprentissage



Validation de l'apprentissage



Construction et signification de la courbe ROC

Apprentissage :

• $X_n=\{x_1,x_2,\ldots,x_n\}$: échantillons d 'une même classe

• $f(x/\omega)=f(x,\Theta)$ où $\Theta=\{\Theta_1, \Theta_2,\ldots, \Theta_m\}$ paramètres

•Choisir Θ Tel que

$$f(X_n,\Theta)=\prod_{k=1}^{n} f(x_k,\Theta)=\sum_{k=1}^{n}\log(f(x_{k,\Theta}))\ \textit{Maximum}$$

•f est supposée Gaussienne $\rightarrow \Theta=\{\mu,\sigma\}$ et

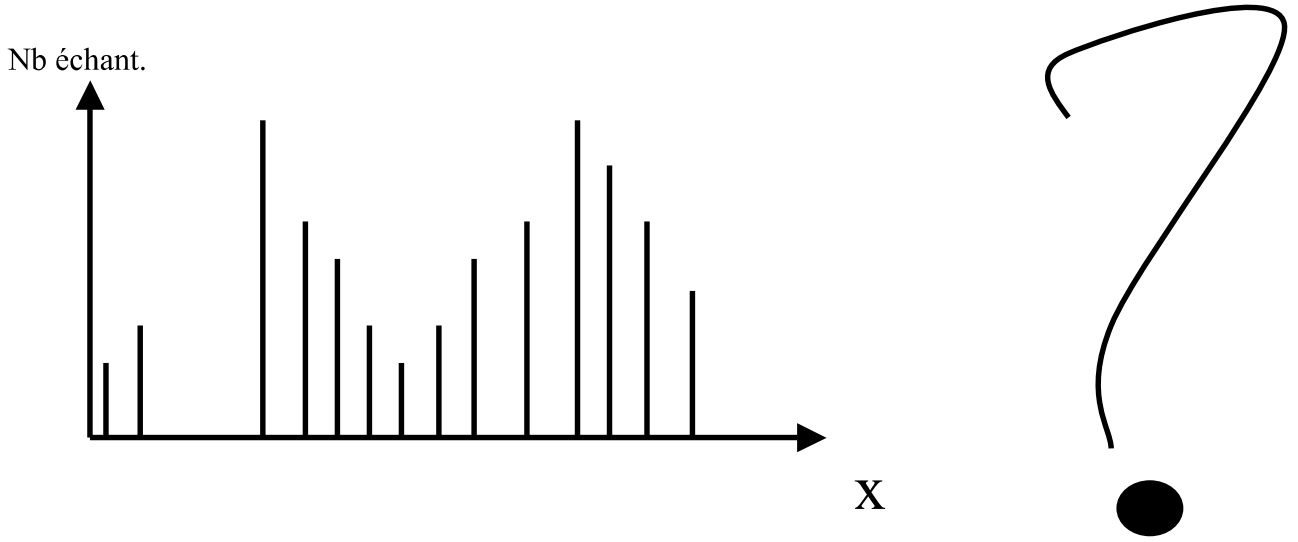
$$\begin{cases} \overline{\mu}=\dfrac{1}{n}\displaystyle\sum_{k=1}^{n} x_k \\ \overline{\sigma}=\dfrac{1}{n}\displaystyle\sum_{k=1}^{n}(x_k-\overline{\mu})^2 \end{cases}$$

Inconvénients :

⇨Difficulté d ’Estimation

⇨ Hypothèse Gaussienne trop Simple

⇨ Nécessité de Beaucoup d ’Echantillons



- « k plus proches voisins » à rapprocher :

  ⇨ de la Classification Automatique (notion de « Proximité »)

  ⇨ des Techniques Bayésiennes (hypothèses sur la forme des classes en moins)

- Convergence « remarquable » quand nb échant.$\rightarrow\infty$

- Simplicité et Qualité $\Rightarrow$ Méthode de référence pour l ’évaluation des méthodes connexionnistes

Principe :

Déterminer la classe de chacun des k points les plus proches de x dans $\mathcal{R}^n$ parmi les formes d 'apprentissage et affecter x à la classe la plus représentée

Inconvénients :

- Temps de calcul en Décision

  ⇨ Calcul de *N* distances dans un espace à *m* dimensions

- Organiser l 'espace de représentation des formes (pavage, tri, hiérarchie...)

- Prise en Compte du Contexte
- Chaînes de Markov
- Modèles de Markov Caché
- En Parole et Ecriture Cursive

- Réaliser un appariement élastique
- Trouver le prototype nécessitant le minimum de déformation
- Vocabulaire limité

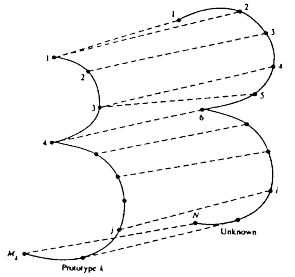


Figure 3.2 : Appariement élastique

- Utilisées en reconnaissance de la parole (recalage temporelle) et en reconnaissance de l ’écriture cursive (recalage spatiale)

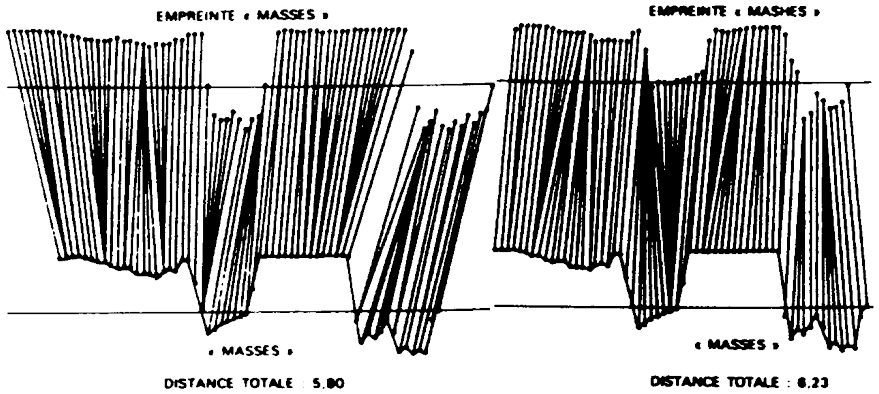


Figure 5.2. *Comparaison entre les mots "masses" et "mashes" donnée dans [Lev84].*

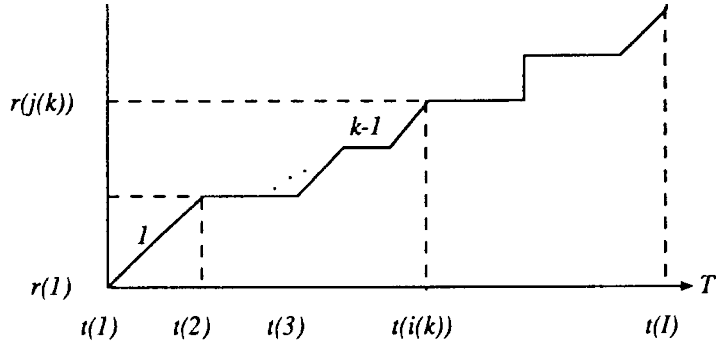


Figure 5.3. *Exemple de chemin de recalage.*

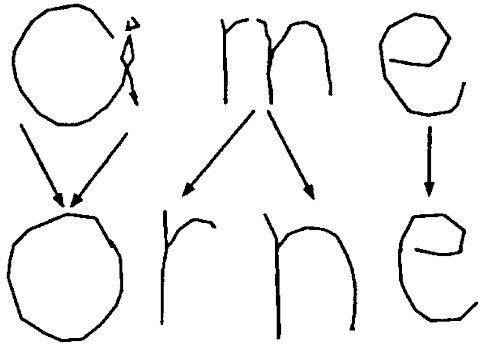
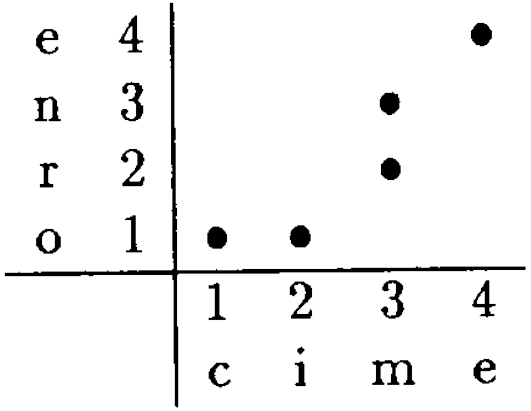


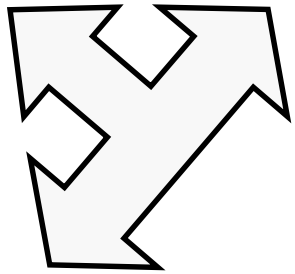
Figure 6.53. *Appariement entre les mots "cime" et "orne".*



Approches Syntaxique et Structurelle

Comparaisons de Chaînes

- Programmation Dynamique
- Distances de Chaînes



Isomorphismes de Graphe

Grammaires de Langages

- **Noam Chomsky**

- Taille de l ’ensemble d ’apprentissage / Dimension de l ’espace de représentation ?

- Choix des Traits Caractéristiques ou des Primitives ?

- Tests et Validations :
  - ⇨ Resubstitution
  - ⇨ « Hold-out »
  - ⇨ « Leave-one-out »
  - ⇨ « Bootstrap »
- Type d ’Erreurs :
  - ⇨ Faux Rejet
  - ⇨ Fausse Reconnaissance
  - ⇨ Confusion

• Taux d ’erreur = taux de rejet (faux rejet) + taux de confusion (fausses reconnaissances et confusions)

• Taux de reconnaissance = 100% - taux d ’erreur

• Bornes d ’erreurs (avec un nombre de formes infini)

$$err_B << ... E_k << E_{k-1} ... << E_2 << E_1 << 2err_B$$