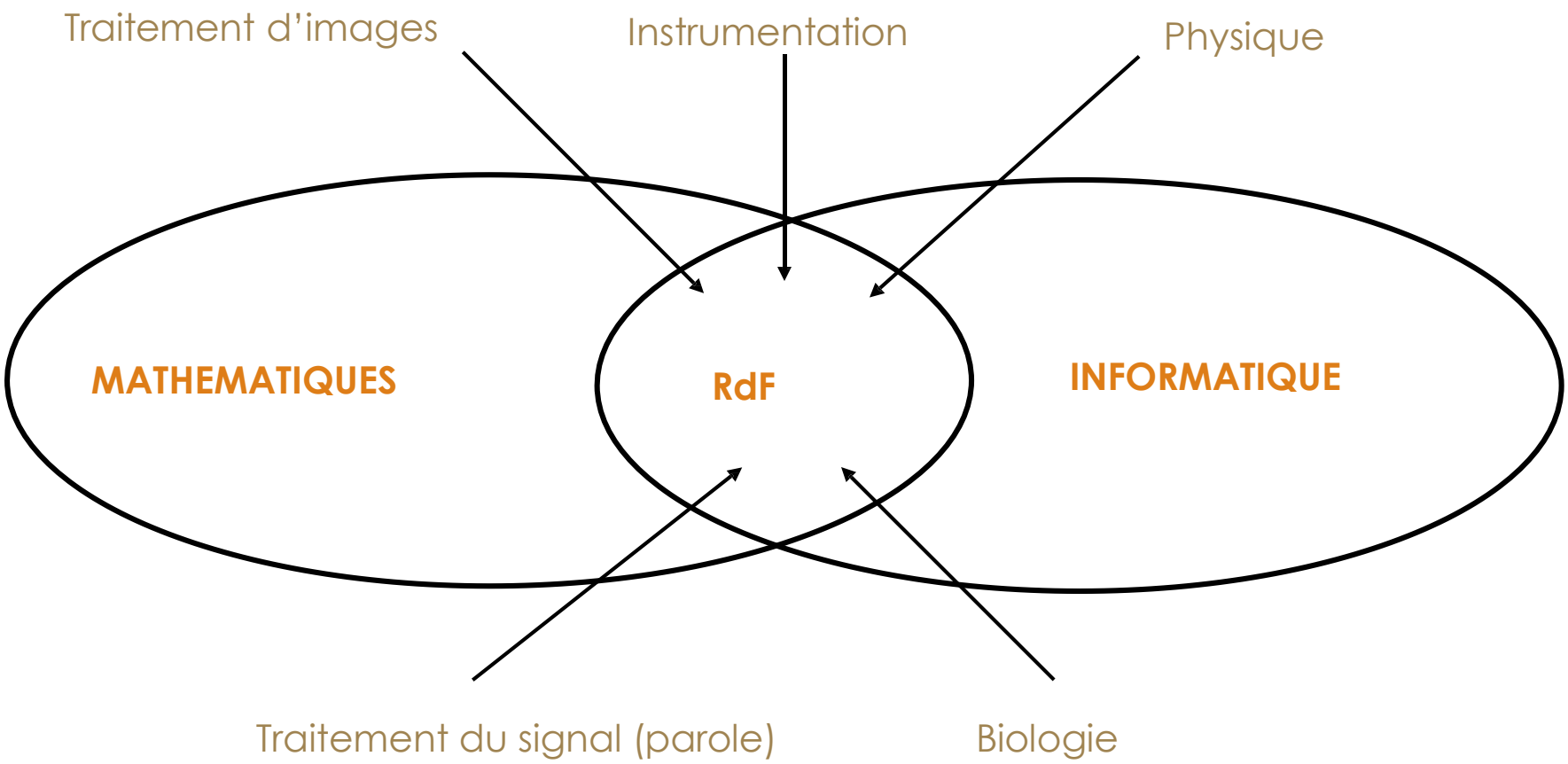
RECONNAISSANCE DES FORMES

Principe


Traitement d'images
Instrumentation
Physique
MATHEMATIQUES
RdF
INFORMATIQUE
Traitement du signal (parole)
Biologie

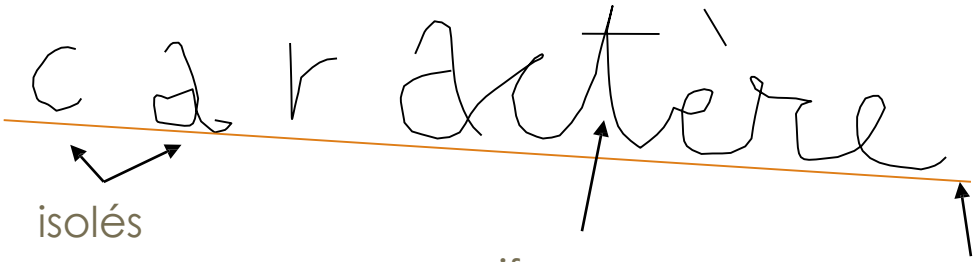
Principe

- On dispose d'un ensemble de formes dont la classe est connue (base d'apprentissage ou de références)

- On met au point une méthode qui, en étudiant cette base, sera ensuite capable de classer des formes inconnues

Applications

Reconnaissance de texte



isolés

cursifs

ligne de base

Variabilité entre scripteurs

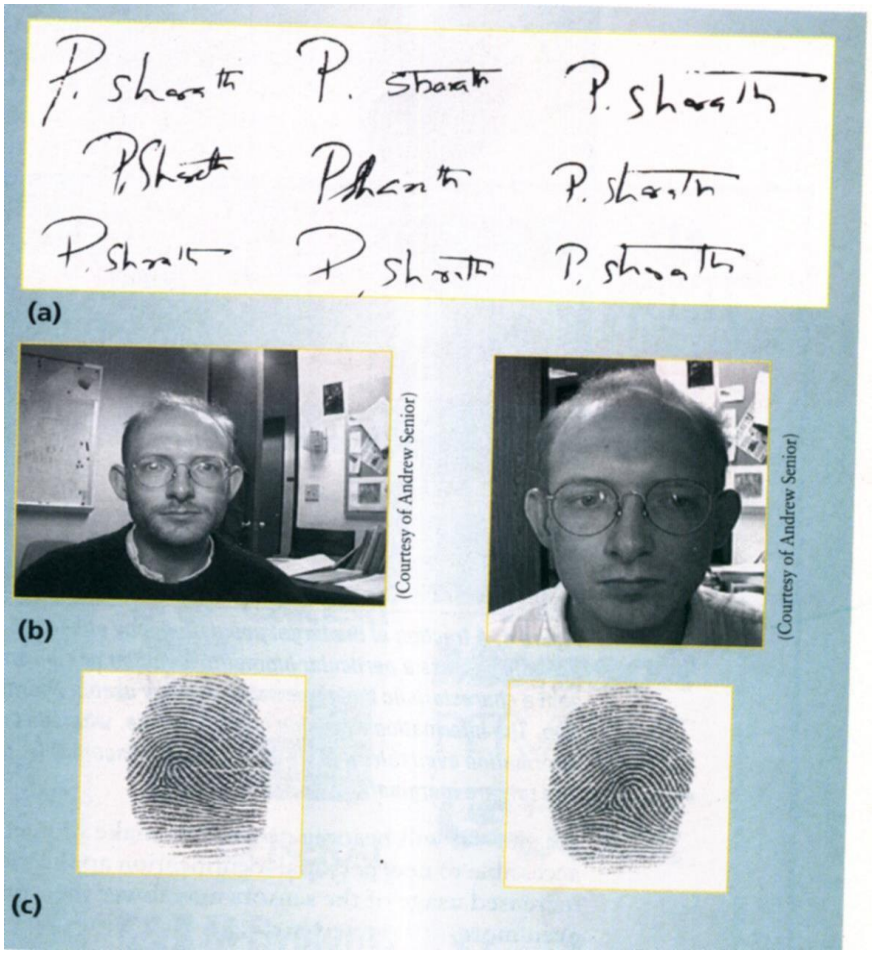
Applications

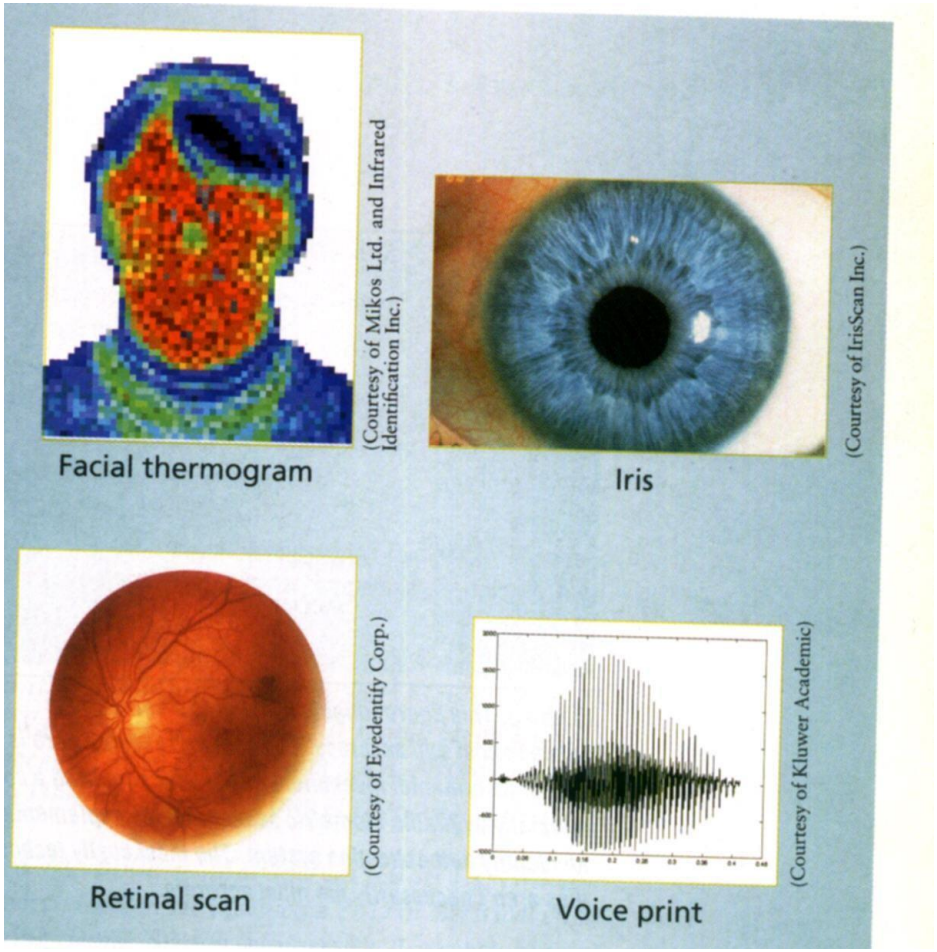
Biométrie

Signature

Visage

Iris

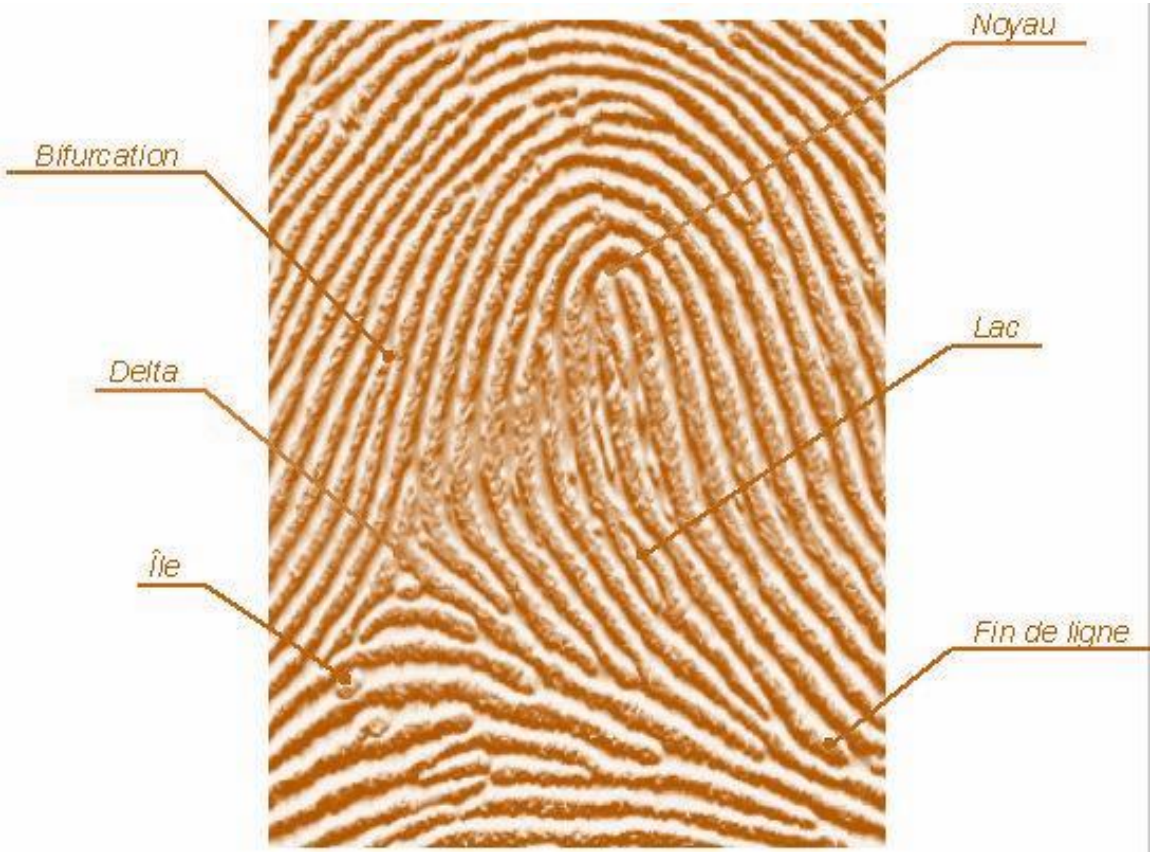




Empreinte digitale

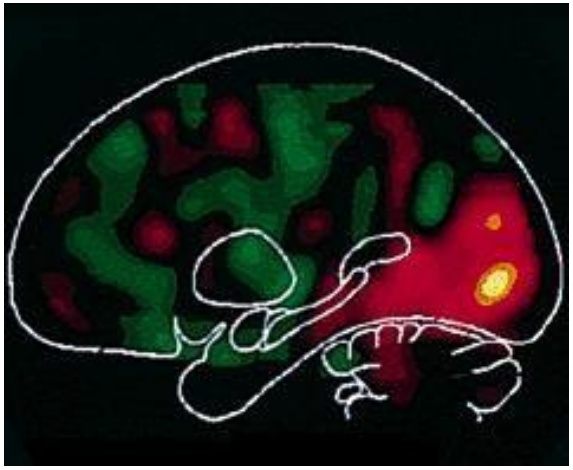
Empreinte vocale

Applications Reconnaissances d'empreintes digitales



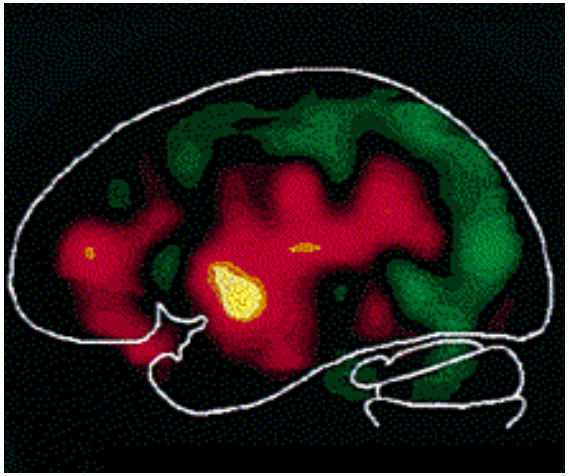
Applications

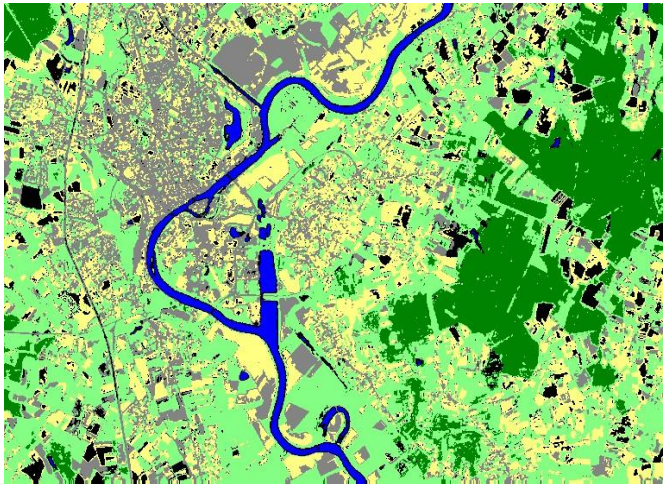
Imagerie



médicale

satellitaire



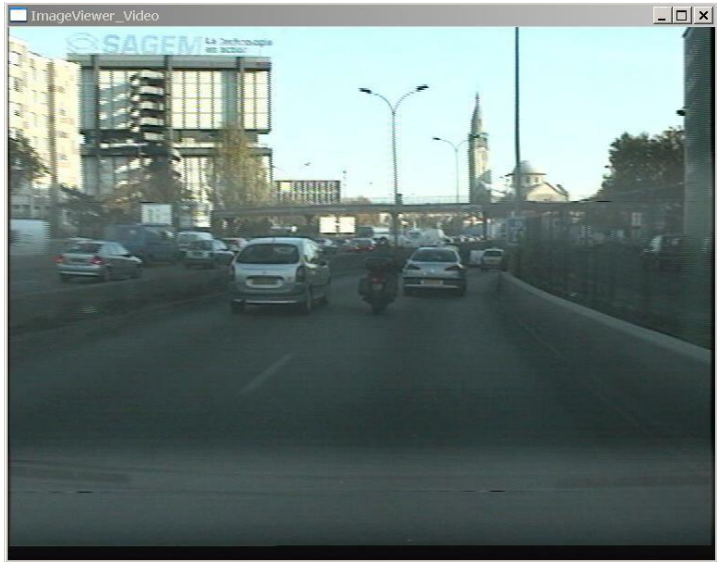


Applications

Analyse de scène

Aide technique

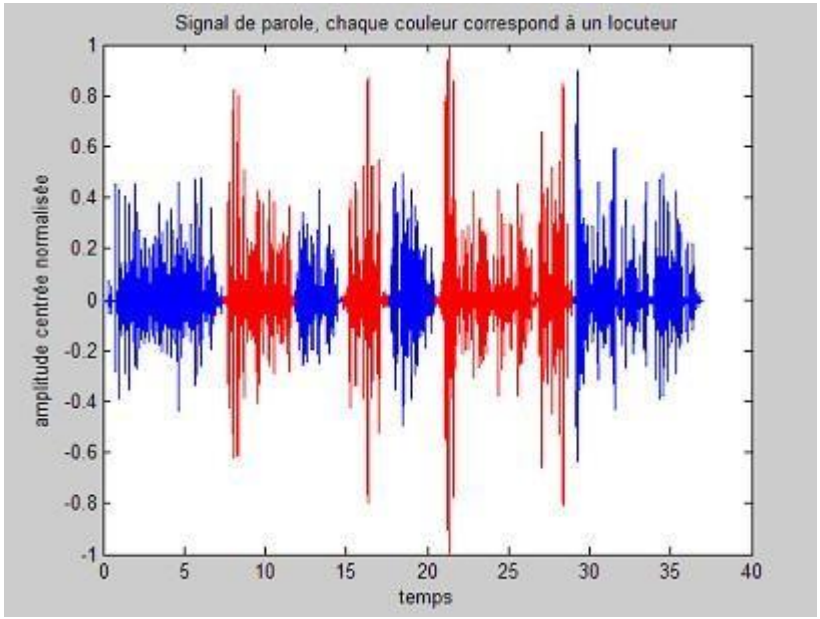
Véhicules intelligents



Applications

Signaux audio

Reconnaissance de locuteurs



Reconnaissance de parole

Applications Signaux divers : diagnostic de panne

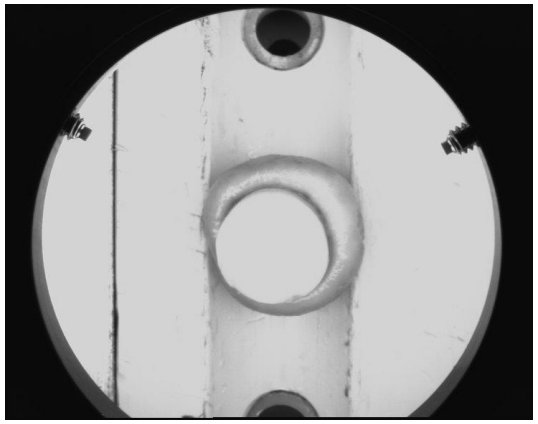
Sur un avion, il y a plusieurs centaines de capteurs qui donnent des signaux en permanence.

Comment faire pour détecter automatiquement une panne et diagnostiquer son origine ?

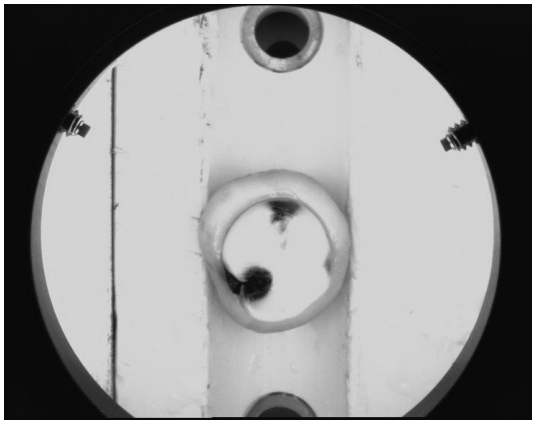
Applications

Contrôle de qualité

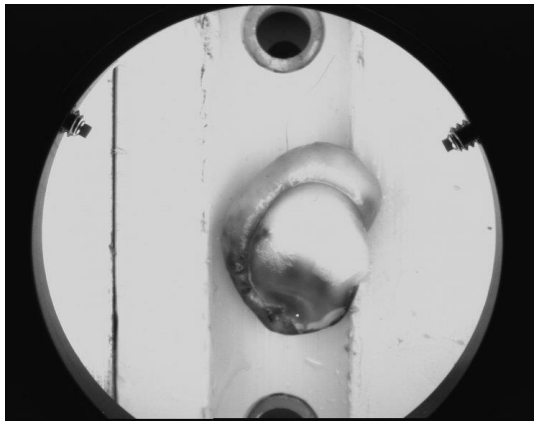
Champignon fermé



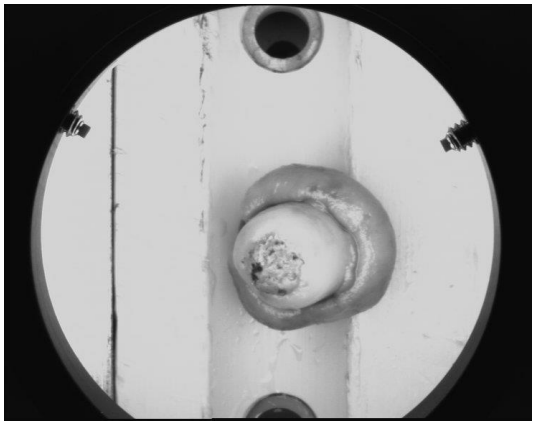
Champignon véreux



Champignon taché

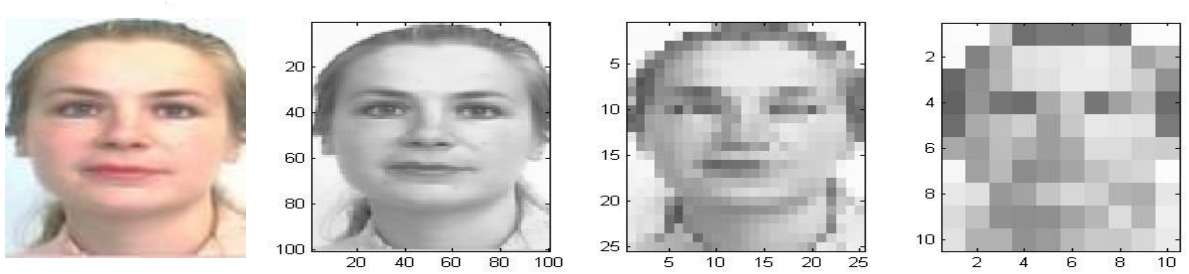


Champignon terreux



Difficultés

Problème de résolution



Problème de pose

Distance entre visage?

Difficultés

Expressions faciales, occlusion

A.M. Martinez and R. Benavente. The AR Face Database. CVC Technical Report #24, June 1998".

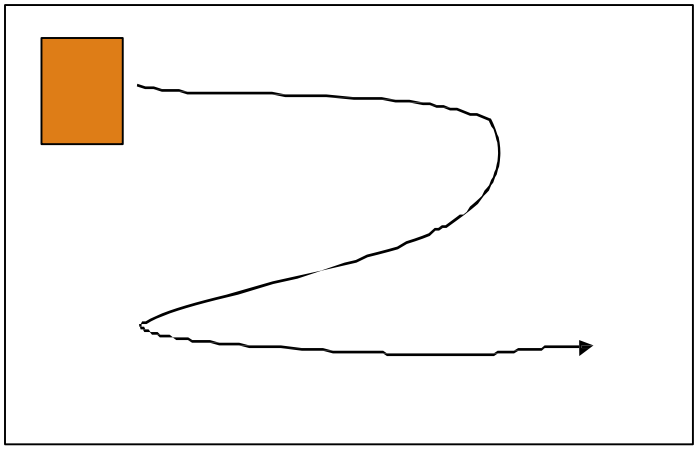
Cas particulier : la détection

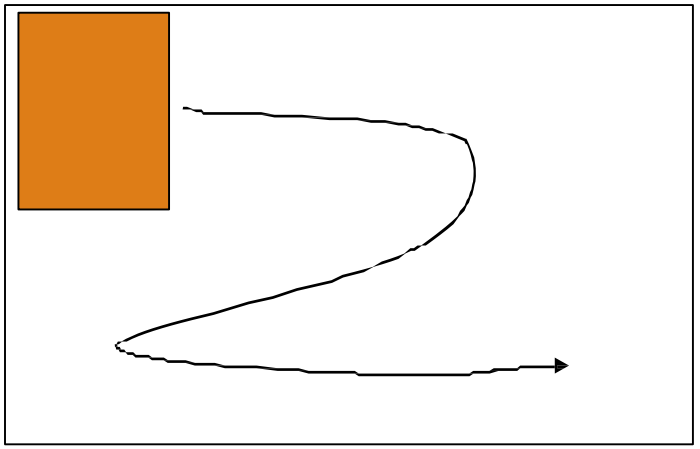
Raisonnons sur un cas concret :
comment détecter un visage dans une image ?

Cas particulier : la détection

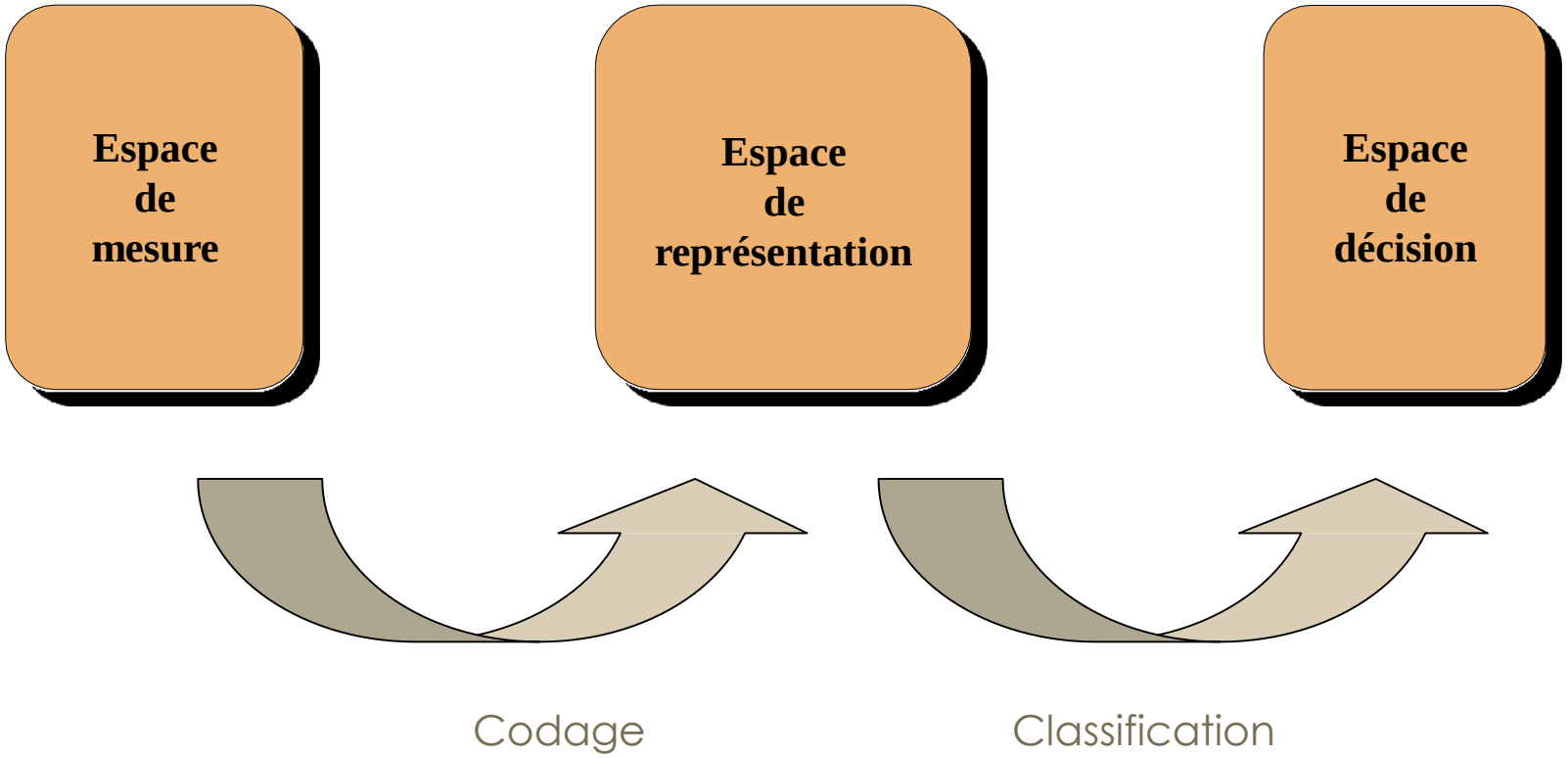
- On change le problème de détection en un problème de classification
- On présente une imagette (une zone de l'image) en entrée d'un système de reconnaissance. Celui-ci nous dit si cette imagette est un visage ou non (sortie binaire)
- Comment savoir où rechercher dans l'image ?
- Comment détecter à plusieurs échelles ?

Solution: En testant toutes les combinaisons possibles

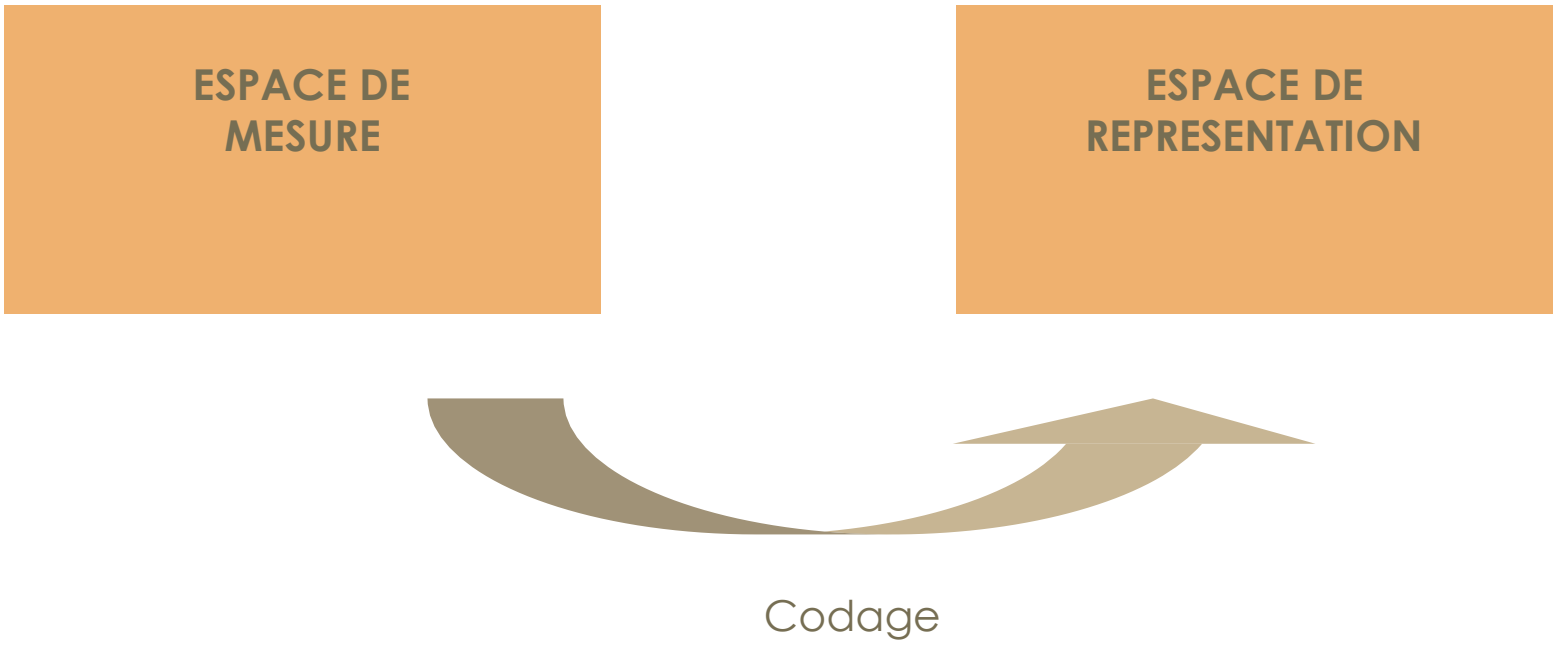




Les étapes de la reconnaissance de formes



Prétraitements et codage



Caractéristique des codages

- **Pouvoir discriminant**

Le codage doit être différent pour des exemples de classes différentes → forte variance inter-classes

- **Pouvoir unifiant**

Le codage doit être à peu près le même pour tous les exemples d'une même classe → faible variance intra-classe

Stabilité/invariance

- **Codage le plus insensible possible au bruit**

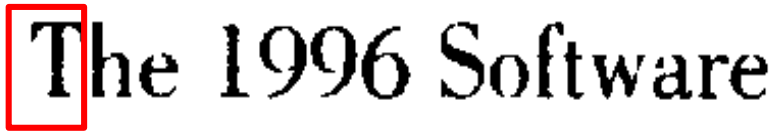
En fonction des applications, invariance en translation, rotation, changement d'echelle

- **Faible dimension**
  - Temps de calcul seront faibles
  - Augmenter la dimension peut détériorer les résultats

Prétraitement

But : isoler la forme à reconnaître

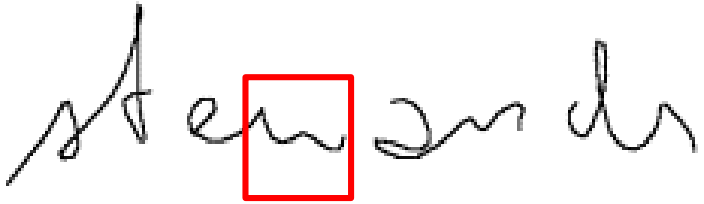
Exemple : isoler les lettres



Paradoxe

Il faut segmenter pour reconnaître et …

reconnaître pour segmenter



Prétraitement

Action (iii), namely proposing new words, is based on the *visually similar neighbourhood* (VSN) of a word choice. The VSN of a word is computed by the same process that is used by the WR to reduce a lexicon based on wholistic properties of a word. In cases where the reduced lexicon is still too large (over 200 words), more stringent constraints (such as word length) are applied in order to reduce the size even further. The VSN is computed automatically from the ASCII representation of a word. For example, if the correct words are "nuclear power" and the set of word choices result in "nucleus power" and "mucus power", collocational analysis results in the additional word choice "nuclear". This is based on the fact that (i) "nuclear" is in the VSN of "nucleus" and (ii) the words "nuclear power" constitute a strong collocation. This method is currently being attempted for only a small set of strong collocations.

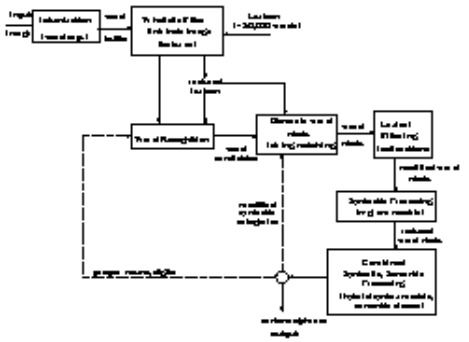


Figure 2: Overall Model of Sentence/Phrase Recognition

but the word 'by' may be preceded by virtually anything. Such analysis extends the utility of collocational analysis but comes with a risk of promoting incorrect word choices.

Action (iii), namely proposing new words, is based on the visually similar neighbourhood (VSN) of a word choice. The VSN of a word is computed by the same process that is used by the WR to reduce a lexicon based on wholistic properties of a word. In cases where the reduced lexicon is still too large (over 200 words), more stringent constraints (such as word length) are applied in order to reduce the size even further. The VSN is computed automatically from the ASCII representation of a word. For example, if the correct words are "nuclear power" and the set of word choices result in "nucleus power" and "mucus power", collocational analysis results in the additional word choice "nuclear". This is based on the fact that (i) "nuclear" is in the VSN of "nucleus" and (ii) the words "nuclear power" constitute a strong collocation. This method is currently being attempted for only a small set of strong collocations.

5. SYNTACTIC MODELS: USING POS TAGS TO REDUCE WORD NEIGHBOURHOODS

The performance of a WR system can be improved by incorporating statistical information at the word sequence level. The performance improvement derives from selection of lower-rank words from the WR output when the surrounding context indicates such selection makes the entire sentence more probable. Given a set of output words $\hat{X}$ which emanate from a noisy channel (such as the WR), N-gram word models (6) can be used to determine the string of words $\hat{W}$ which most probably gave rise to it. This amounts to finding the string $\hat{W}$ for which the a posteriori probability

$$P(\hat{W} \mid \hat{X}) = \frac{P(\hat{W}) \cdot P(\hat{X} \mid \hat{W})}{P(\hat{X})}$$

is maximum, where $P(\hat{X} \mid \hat{W})$ is the probability of observing $\hat{X}$ when $\hat{W}$ is the true word sequence, $P(\hat{W})$ is the a priori probability of $\hat{W}$ and $P(\hat{X})$ is the probability of string $\hat{X}$. The values for each of the $P(\hat{X}_i \mid \hat{W}_i)$ are known as the channel (or confusion) probabilities and can be estimated empirically. If we assume that words are generated by an mth order Markov source, then the a priori probability $P(\hat{W})$ can be estimated as

$$P(\hat{W}) = P(\hat{W}_{m+1} \mid \hat{W}_{m+1-m}) \cdots P(\hat{W}_2 \mid \hat{W}_1) \cdot P(\hat{W}_1)$$

where $P(\hat{W}_n \mid \hat{W}_{n-m} \cdots \hat{W}_{n-1})$ is called the mth-order transitional probability. The Viterbi algorithm (7) is a dynamic method of finding optimal solutions to the above quantity.

The problem with such approaches is that as the number of words grow in the vocabulary, estimating the parameters reliably becomes difficult. More specifically, the number of low or zero-valued entries in the transition matrix starts to rise exponentially. (8) reports that of the 6.799×10^{14} 2-grams that could possibly occur in a 365,351,251 word corpus (consisting of 260,740 unique words), only 14,494,217 actually occurred, and of these, 8,045,024 occurred only once.

In n-gram class models, words are mapped into syntactic (9) classes. In this situation, $p(w_i \mid w_{i-1})$ becomes

$$p(w_i \mid w_{i-1}) = p(w_i \mid G(w_i)) \cdot p(G(w_i) \mid G(w_{i-1}))$$

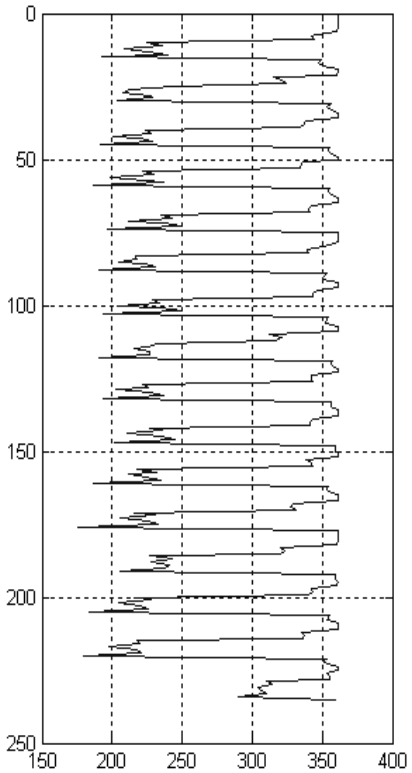
where $p(G(w_i) \mid G(w_{i-1}))$ is the probability to get to the class $G(w_i)$ following the class $G(w_{i-1})$ and $p(w_i \mid G(w_i))$ is the probability to get the word w_i among the words of the class $G(w_i)$.

The research described here uses n-gram class models where part-of-speech (POS) tags are used to classify words. We use the notation $A : B$ to indicate the case where word A has been assigned the tag B. For each syntactic analysis, we

Prétraitement

Segmentation

Projection selon y → segmentation en lignes

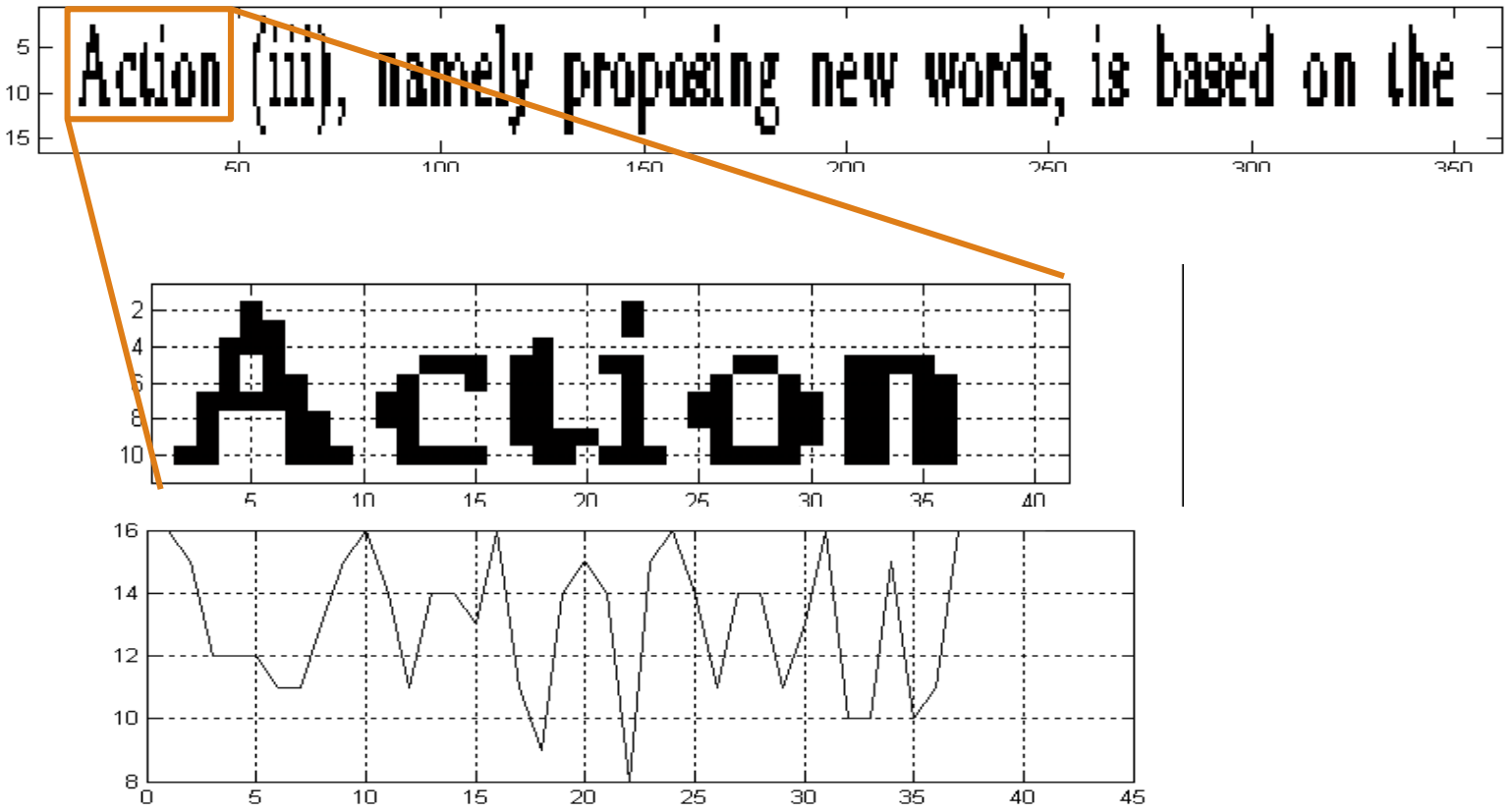


Action (iii), namely proposing new words, is based on the *visually similar neighbourhood* (VSN) of a word choice. The VSN of a word is computed by the same process that is used by the WR to reduce a lexicon based on wholistic properties of a word. In cases where the reduced lexicon is still too large (over 200 words), more stringent constraints (such as word length) are applied in order to reduce the size even further. The VSN is computed automatically from the ASCII representation of a word. For example, if the correct words are "nuclear power" and the set of word choices result in "nucleus power" and "mucus power", collocational analysis results in the additional word choice "nuclear". This is based on the fact that (i) "nuclear" is in the VSN of "nucleus" and (ii) the words "nuclear power" constitute a strong collocation. This method is currently being attempted for only a small set of strong collocations.

Prétraitement

Segmentation

Projection selon x → segmentation en lettres



Codage global vs structurel

Codage global

On code toute la forme sans en extraire d'éléments spécifiques.
La forme peut être représentée par un vecteur de paramètres.

Exemple: pour une personne le poids et la taille

Codage structurel

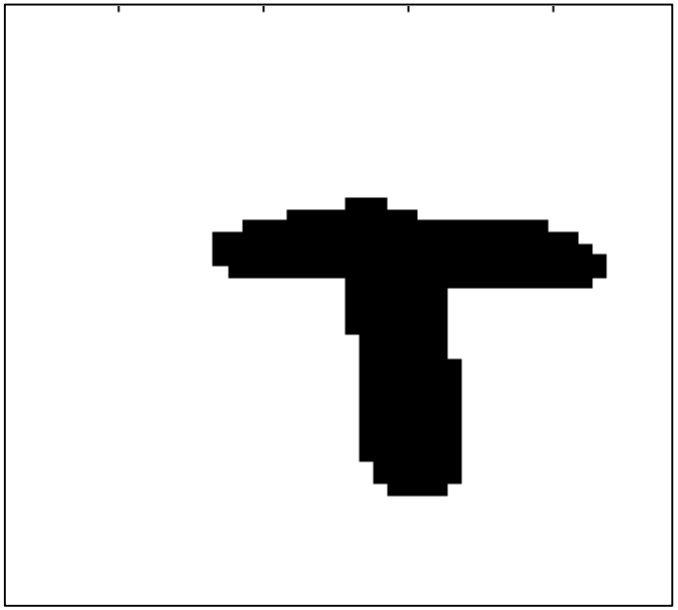
On extrait des éléments spécifiques de la forme et leur relation.

Exemples :

- pour la personne 1 : pull rouge au dessus d'un pantalon bleu au dessus de chaussures noires
- Pour reconnaitre un 'L' : contour d'abord vertical puis horizontal

Codage rétinien

On garde toute l'information directement dans une rétine :

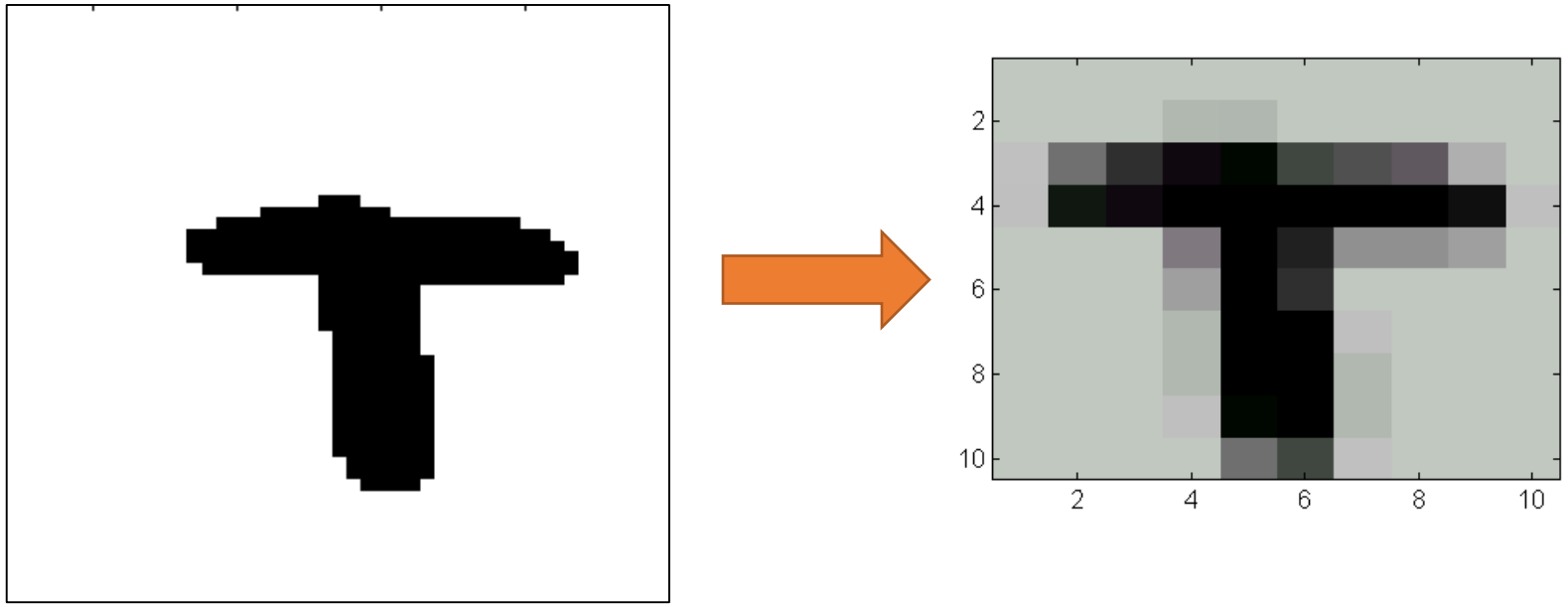


Problème :

- La lettre n'est pas toujours à la même position
- La résolution de l'image n'est pas toujours la même

Codage rétinien

- Calcul du centre de gravité
- Sélection du plus petit carré centré en G et englobant tous les pixels
- Réduction de la dimension (attention, pas binaire !!)

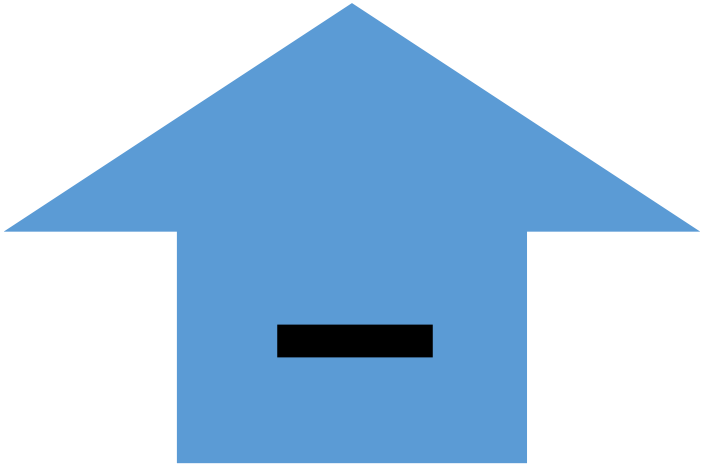


Codage rétinien

Codage quasiment neutre

- Pas de perte d'information
- Laisse le classifieur travailler
- Efficace si base d'exemples importante

Ne tolère ni les transformations ni les déformations



Moments géométriques

- Ils codent n'importe quelle forme, même non binaire
- Soit I(x,y) une image Moment d'ordre p,q

$$M_{pq} = \sum_x \sum_y x^p y^q I(x, y)$$

Aire :

$$M_{00} = \sum_x \sum_y I(x, y)$$

Centre de masse :

$$M_{01} = \sum_x \sum_y y^q I(x, y) \quad M_{10} = \sum_x \sum_y x^p I(x, y)$$

$$\bar{x} = \frac{M_{10}}{M_{00}} \quad et \quad \bar{y} = \frac{M_{01}}{M_{00}}$$

Moments géométriques

Orientation:

$$\theta = \frac{1}{2}\arctan\left(\frac{2M_{11}}{M_{20} - M_{02}}\right)$$

Moments centrés pour être invariant en translation

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y)$$

Moments normalisés pour être invariant en changement d'échelle

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\chi}} \; avec \chi = 1 + \frac{p+q}{2}$$

Moments géométriques

Moments invariants en rotation : Moment de Hu

$$h_0 = \eta_{20} + \eta_{02}$$

$$h_1 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$h_2 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$h_3 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$h_4 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

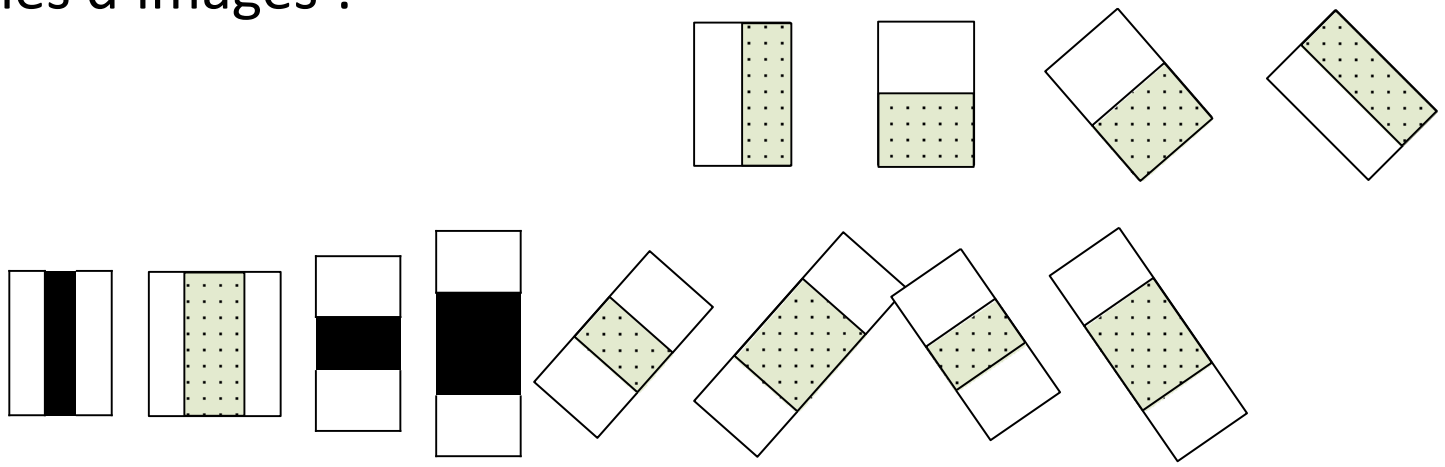
$$h_5 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})]$$

$$h_6 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

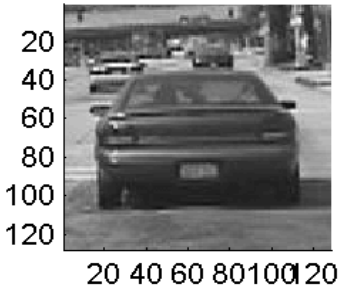
Filtres de Haar

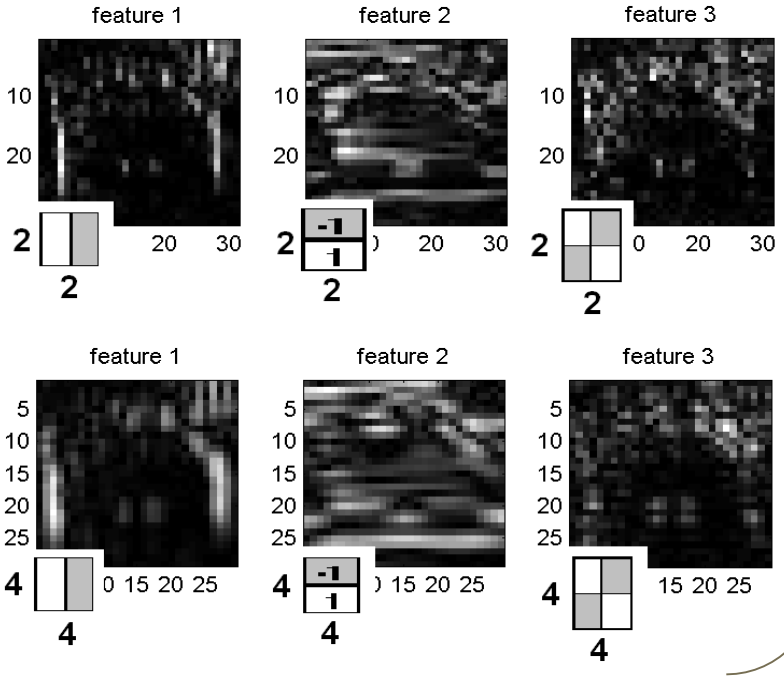
Plusieurs filtres à plusieurs échelles, plusieurs orientations, estimés en plusieurs positions de l'image.

On ne peu pas utiliser toutes ces valeurs, le codage serait bien trop grand (plusieurs dizaines d'images !

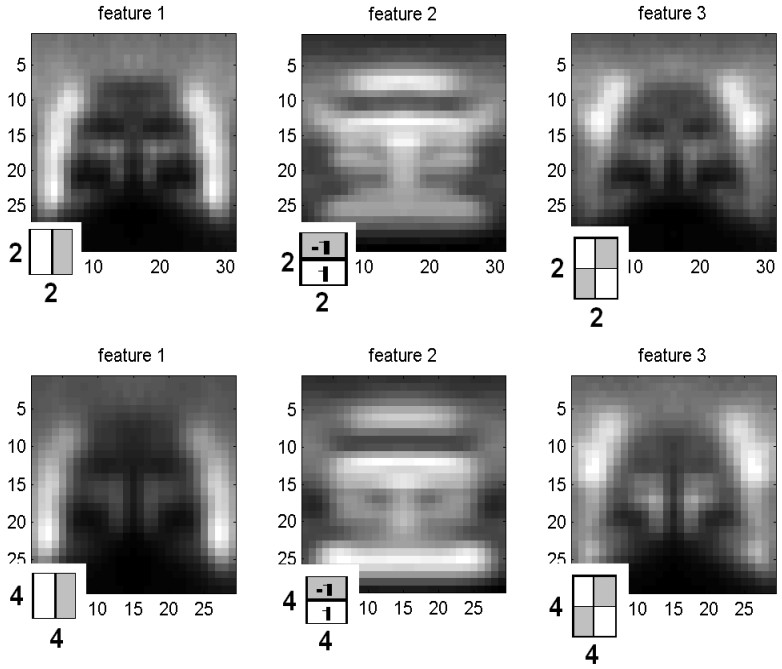


Filtres de Haar





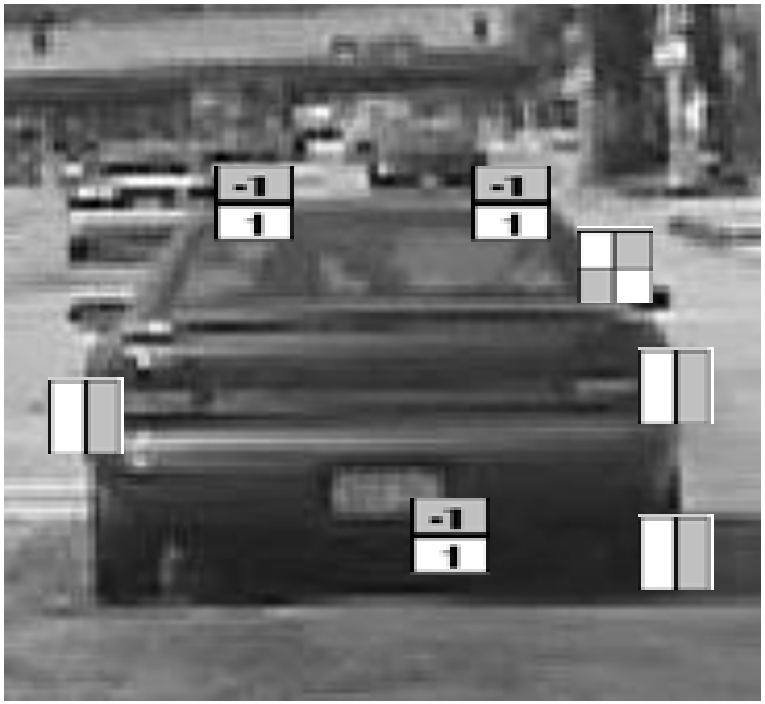
En faisant des statistiques sur plusieurs milliers d'images d'une même classe, on détermine quelles sont les tailles, orientations et positions pertinents pour faire le codage. moyenne pour plusieurs images



Filtres de Haar

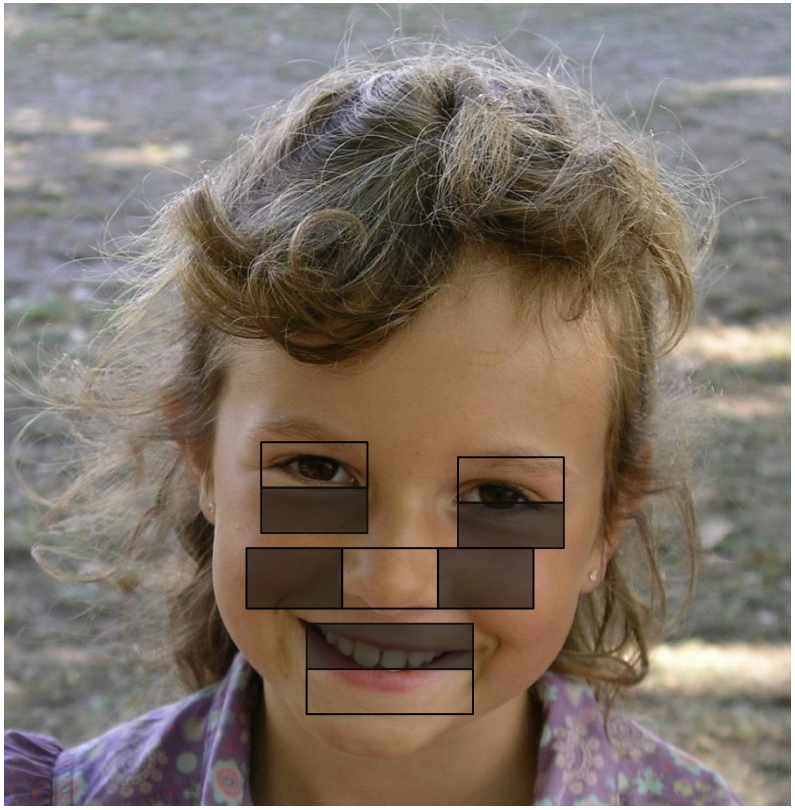
- On en déduit, sur une grande base de données (apprentissage) quels sont les filtres à utiliser et où.

- Le code de l'image (vecteur de caractéristiques) est composé de la sortie de tous ces filtres locaux



Vecteur de caractéristiques

Filtres de Haar



Même chose pour un visage

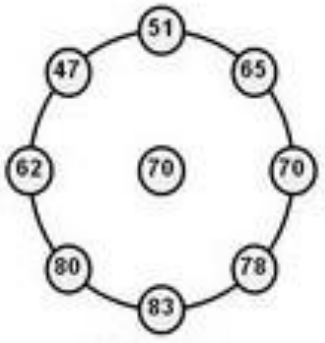
Un visage est représenté par la réponse de plusieurs filtres en différentes positions

Local Binary Patterns (LBP)

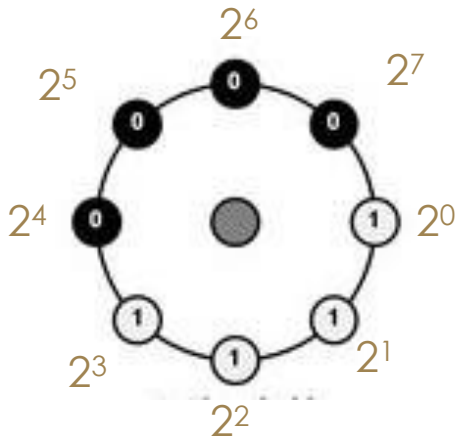
- Idée : obtenir un descripteur complètement invariant à l'éclairage
- Solution : codage des relations d'ordre

  Comparaison du niveau de gris d'un point avec ses voisins
  Chaque comparaison renvoie un nombre binaire
  Le mot binaire obtenu avec les 8 voisins est codé en décimal
  L'histogramme de cette valeur pour tous les points d'une zone forme le descripteur

Exemple
Une portion d'image :
un pixel et ses 8 voisins



Le résultat de comparaison



Le code binaire

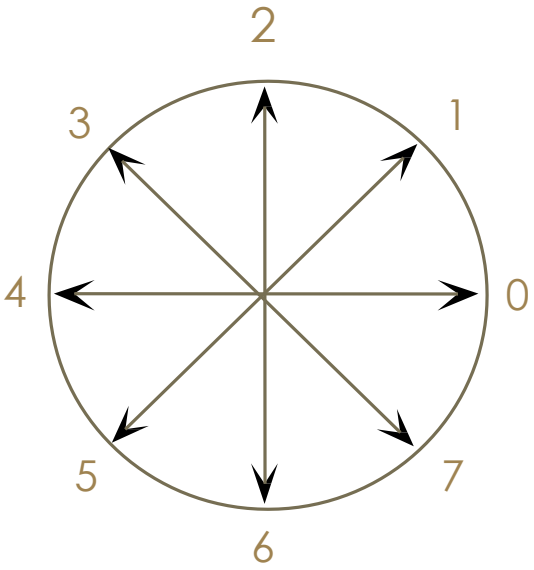
$1*2^0+$
$1*2^1+$
$1*2^2+$
$1*2^3+$
$0*2^4+$ = 15
$0*2^5+$
$0*2^6+$
$0*2^7+$

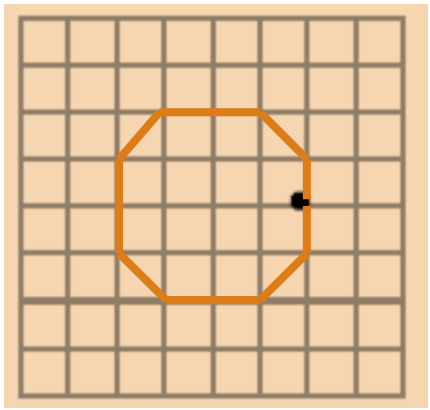
Codage des contours de freeman

On code les contours à partir d'une liste chainée d'angles discrétisés

<u>Exemple :</u>

2 3 4 4 5 6 6 7 0 0 1 2



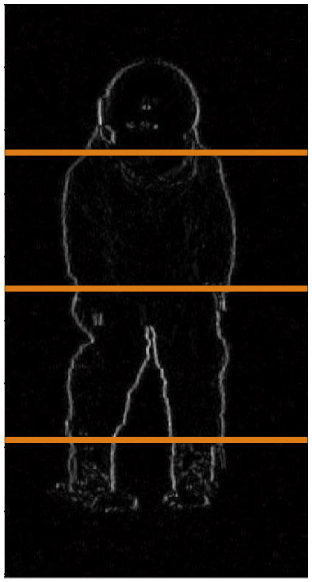


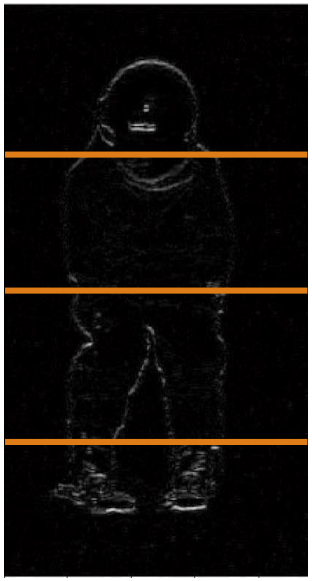
Histogrammes des orientations de gradients

On peut observer les probabilités des orientations du gradient dans différentes zones de l'image

Pour un piéton :

	|Ix|	|Iy|





Histogrammes des orientations de gradients

Etapes:

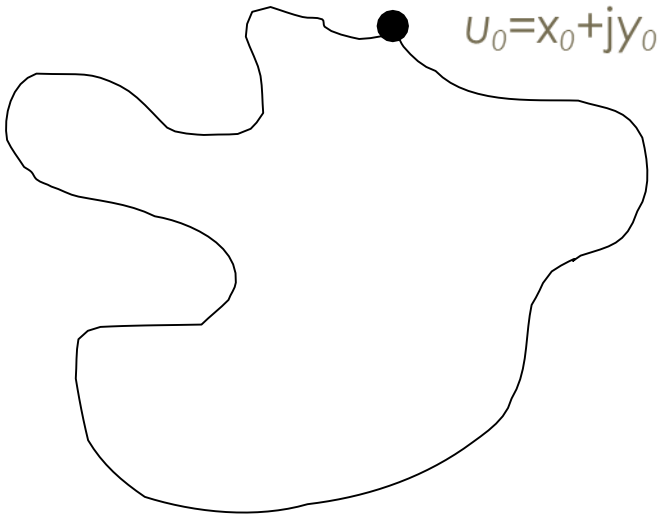
- Calcul des gradients horizontaux et verticaux Gx et Gy avec [-1 0 1] et [-1 0 1]T

$$\theta = \operatorname{atan}\left(Gy / Gx\right) \quad \theta \in [0, \pi]$$

- Calcul de l'orientation du gradient
- Construction de l'histogramme d'orientation de gradient pour différentes zones (souvent 8 bins)
- Concaténation des différents histogrammes
- Variante : pondération des votes par l'amplitude du gradient

Descripteurs de Fourier

Le contour est décrit par la liste chaînée des points qui le constitue. Les coordonnées {xi,yi} des points de cette liste sont transformées en un complexe ui=xi+jyi



Descripteurs de Fourier

$$a_n = \sum_i u_i e^{-\frac{j2\pi ni}{N}}$$

La majorité des informations est contenue dans les basses fréquences les premières valeurs de an suffisent à caractériser le signal et composent le vecteur de caractéristiques

Propriétés:

- Translation : modification de a0 seulement
- Rotation : modification de la phase
- Changement d'échelle : multiplication des an par une constante

Comment trouver un bon codage ?

Méthodes empiriques :

- Choix des caractéristiques pertinentes

Méthodes exploratoire:

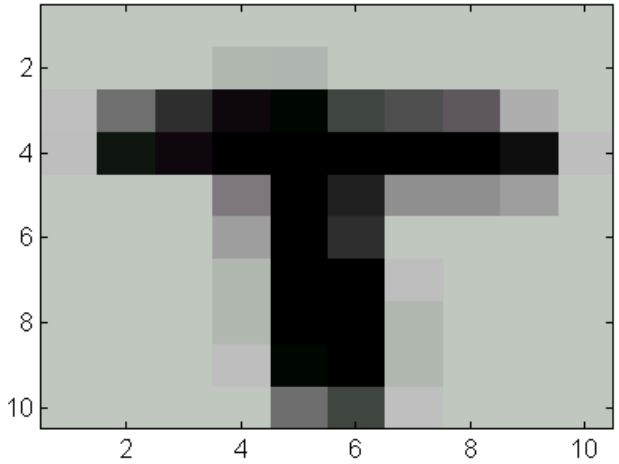
- Algorithmes génétiques

Méthodes statistiques (réduire la dimension ou la taille des données)

- Analyse en composantes principals
- Analyse discriminante linéaire
- Sélection/extraction de caractéristiques

Problème des grandes dimensions

La dimension des données n est la dimension du code les représentant Ex : n = 100



Rétine 10 X 10:

(après centrage et réduction)

Complexité algorithmique linéaire ***f(n²)*** **ou** ***f(n³)*** **ou même, exponentielle**

Problème des grandes dimensions

Plus la dimension n est grande, plus la base de données devra être de grande dimension

Ex : avec deux exemples par dimension

- Si n = 2, 2^2 = 4 données
- Si n = 3, 2^3 =8 données
- Si n = 4 , 2^4 =16 données
- Si n = 20, 2^{20} =1.048.576 données

Problème des grandes dimensions

Considérons un nombre fixé N de points uniformément répartis dans un hyper-cube de dimension n.

Plus n augmente, plus la variance des distances entre points diminue.

En grande dimension, le voisinage immédiat d'une donnée est très peu occupé tandis que la plupart des autres données se trouvent à des distances très comparables de cette dernière.

D'une manière générale, les distances entre données de grande dimension sont très concentrées autour de leur moyenne.

Sélection/extraction de caractéristiques

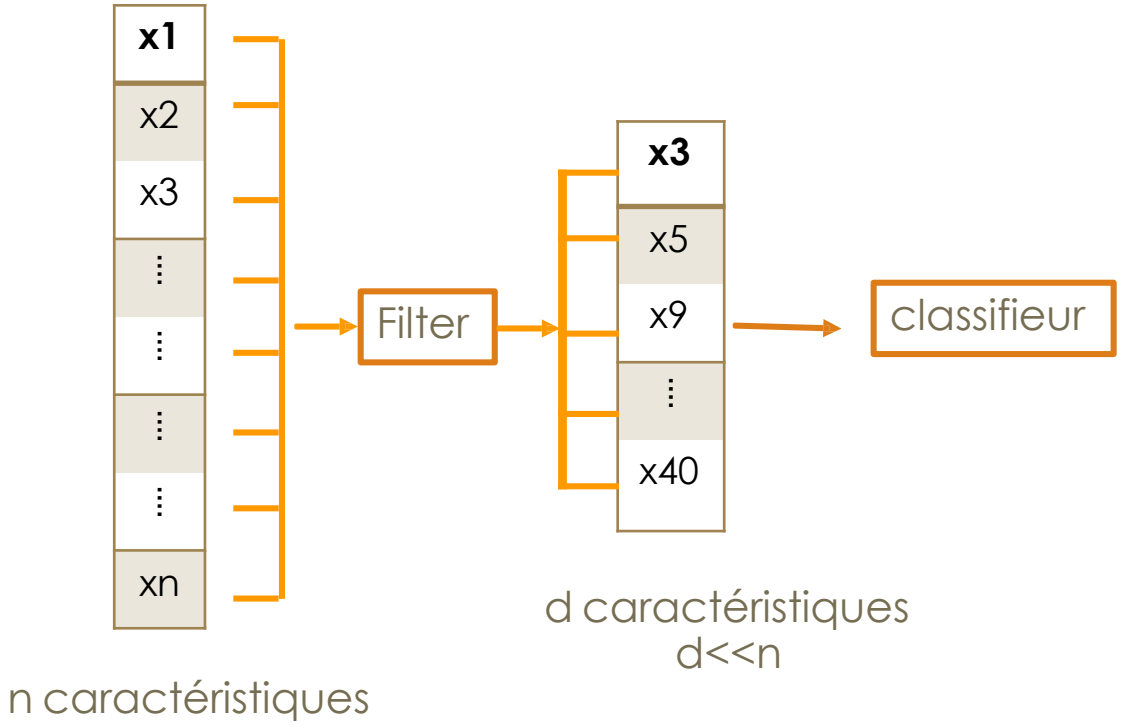
Les méthodes de sélection de caractéristiques peuvent être classées en trois catégories principales :

- Filter
- Wrapper
- Embedded

Sélection/extraction de caractéristiques

Les méthodes filter travaillent en amont de la classification : on étudie les n dimensions (ou caractéristiques) et on en sélectionne d en fonction d'un critère donné.

Par exemple, on garde les caractéristiques qui ont la plus forte corrélation possible avec les étiquettes.



Sélection/extraction de caractéristiques

Avantage des méthodes de type filter

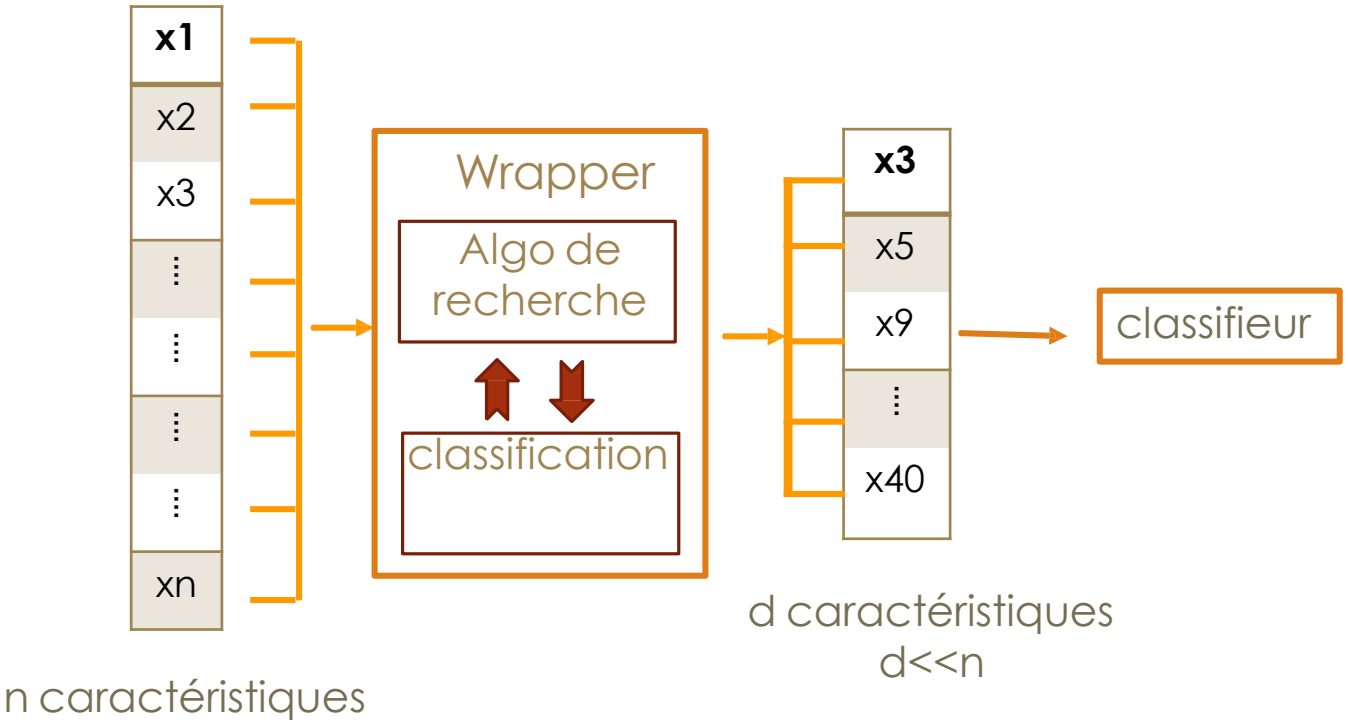
- efficacité calculatoire

Inconvénient des méthodes de type filter

- ne tiennent pas compte des interactions entre caractéristiques
- tendent a sélectionner des caractéristiques redondantes plutôt que complémentaires.
- ne tiennent pas compte de la performance des méthodes de classification

Sélection/extraction de caractéristiques

Les méthodes wrapper évaluent un sous-ensemble de caractéristiques par sa performance de classification en utilisant un algorithme d'apprentissage



La complexité de l'algorithme d'apprentissage rend les méthodes "wrapper" très coûteuses en temps de calcul → stratégie de recherche exhaustive impossible

Sélection/extraction de caractéristiques

Exemple :

on commence par un ensemble vide de caractéristiques. A chaque itération, la meilleure caractéristique parmi celles qui restent est sélectionnée

Avantage des méthodes de type wrapper

Capable de sélectionner des sous-ensembles de caractéristiques de petite taille qui sont performants pour le classificateur

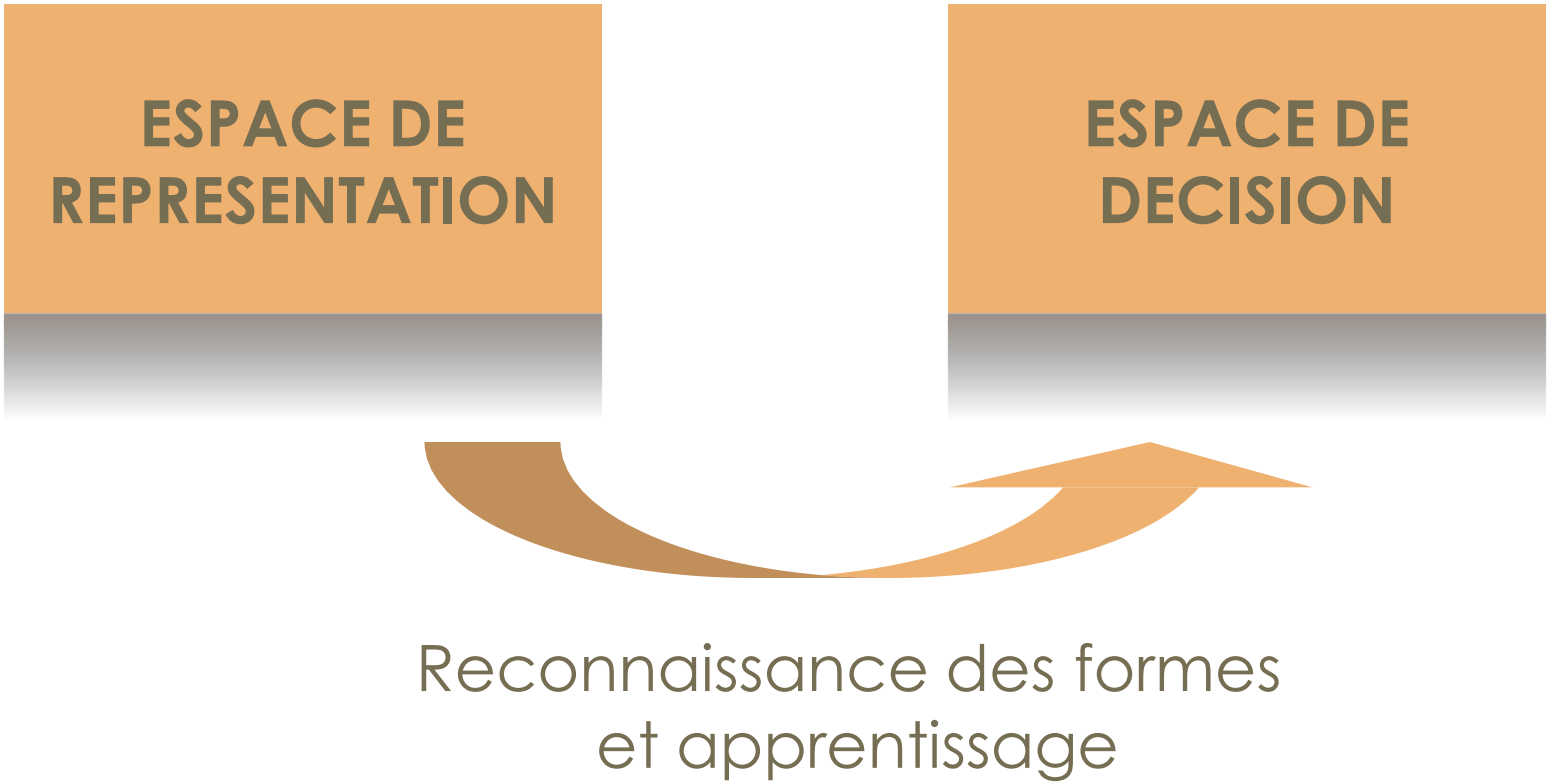
Inconvénient des méthodes de type filter

- Très longues en temps de calcul car beaucoup d'apprentissages nécessaires pour sélectionner le bon sous-ensemble de caractéristiques
- Sous-ensemble dépendant du classificateur choisi

Sélection/extraction de caractéristiques

Les méthodes Embedded ou intégrées incorporent la sélection de variables lors du processus d'apprentissage (boosting, arbre de décision)

Introduction



Introduction

Classifier - Estimer

=

associer une **classe** C ou une **valeur**

à un **vecteur de caractéristiques**

$X=[x_1, x_2,\ldots x_n]$ de dimension n

Vecteur de caractéristique

X = forme + variabilité + bruit de mesure

Introduction

Connaissances disponibles

- Informations fournies par un « expert »
- Modèles explicites (méthode structurelle)
- Cas le plus général : **base de données étiquetées ou non**

Généralisation

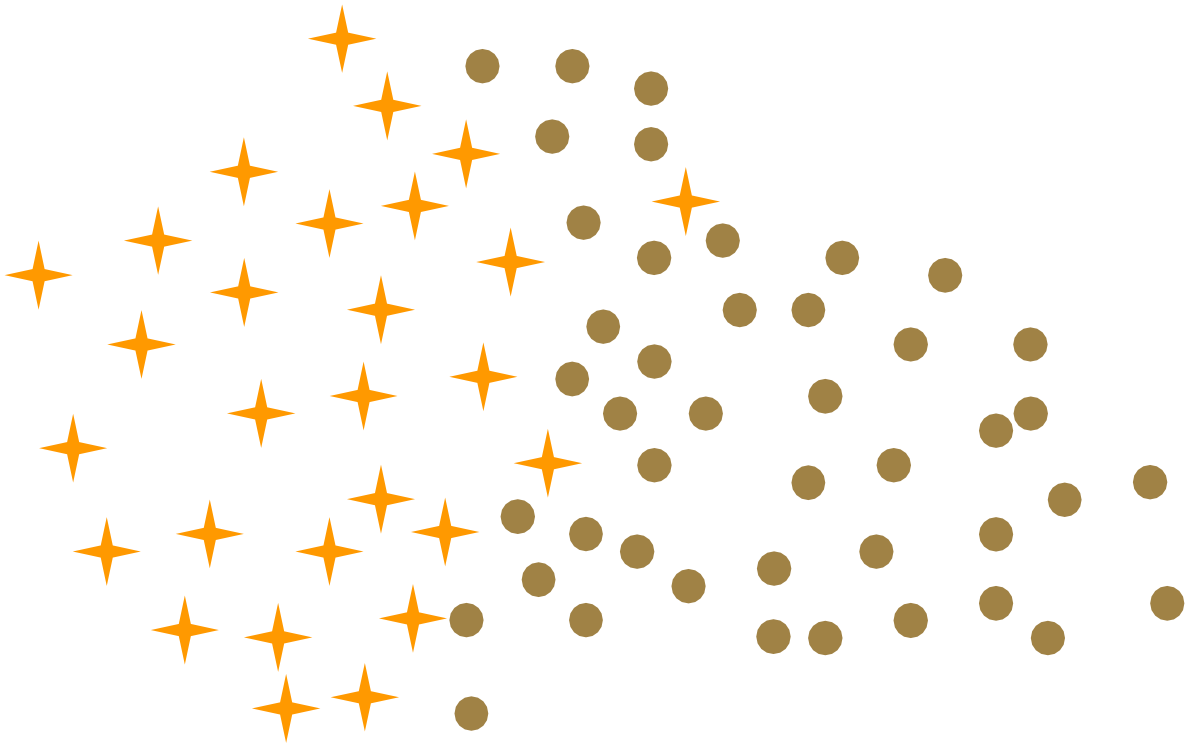
Condition requise

Bonne généralisation → Capacité du classificateur/estimateur à reconnaître/estimer des exemples qu'il n'a pas appris

Attention: Ne pas apprendre par cœur…

Généralisation

Bonne généralisation, où est la frontière ?



Estimation des probabilités

Connaissant un ensemble de N échantillons xi générés selon la loi de probabilité $p(x)$, comment estimer la densité de probabilité $p(x)$?

Il existe deux grands types d'approches :

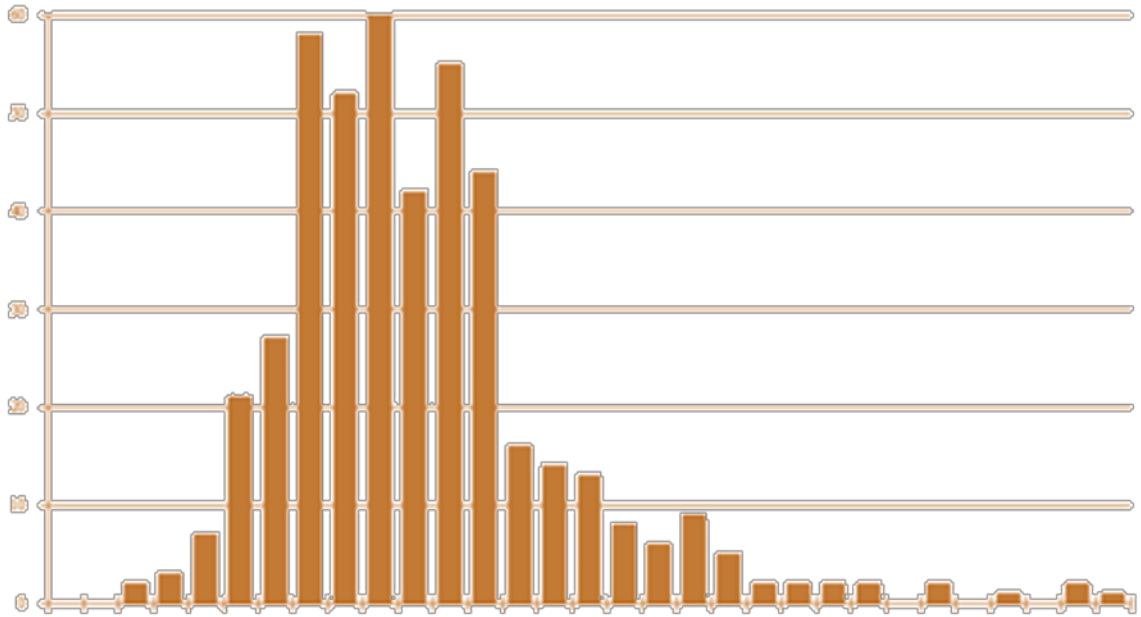
- les méthodes non paramétriques
- les méthodes paramétriques (la loi est fixée a priori et on en recherche les paramètres)

Estimation non paramétrique

Histogramme (non paramétrique)

Une des méthodes les plus simples consiste à estimer l'histogramme de l'ensemble de données.

- On divise chaque dimension en cases (bins) de largeur $\boldsymbol{h}$
- On compte le nombre d'échantillons xi par case

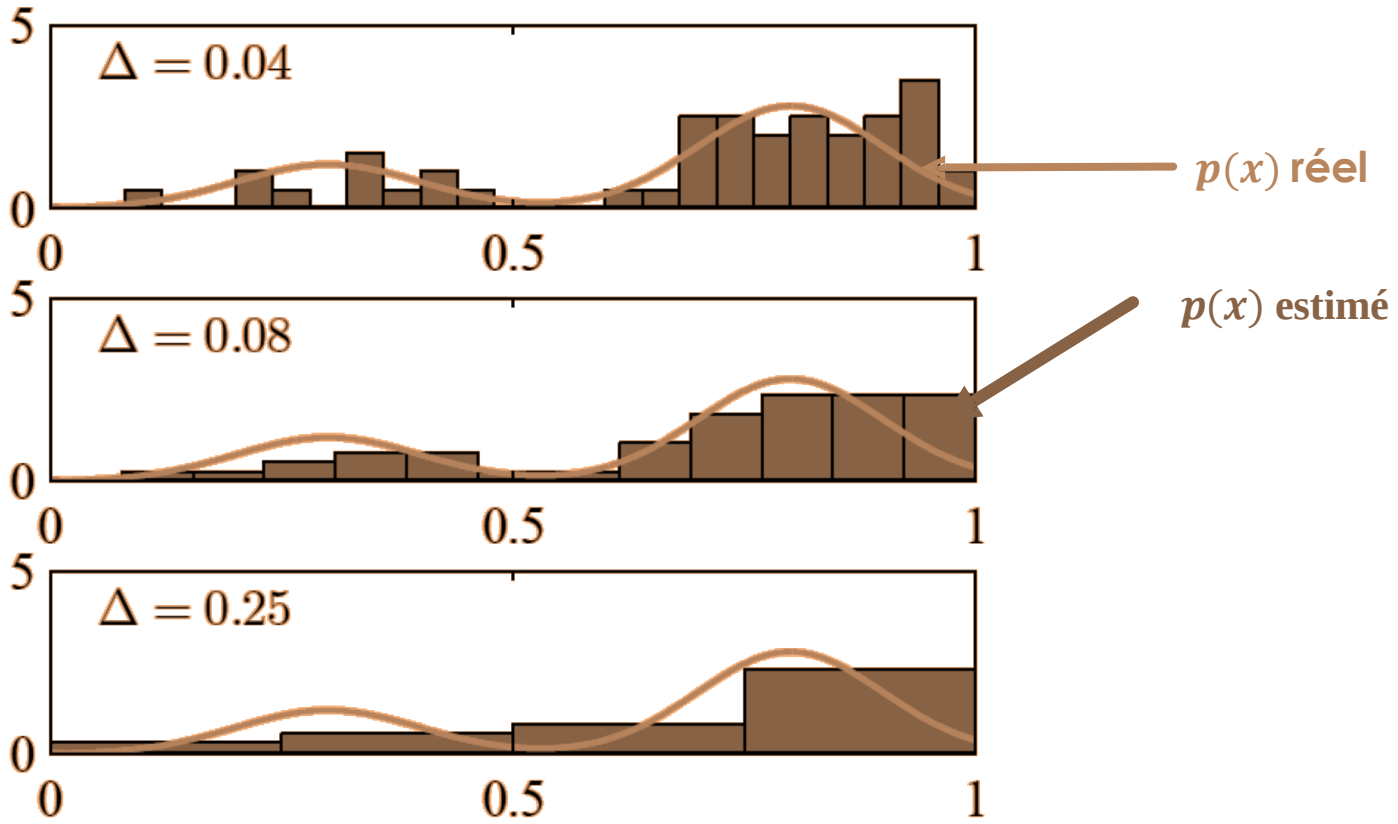


Estimation non paramétrique

Histogramme (non paramétrique)

Problème:

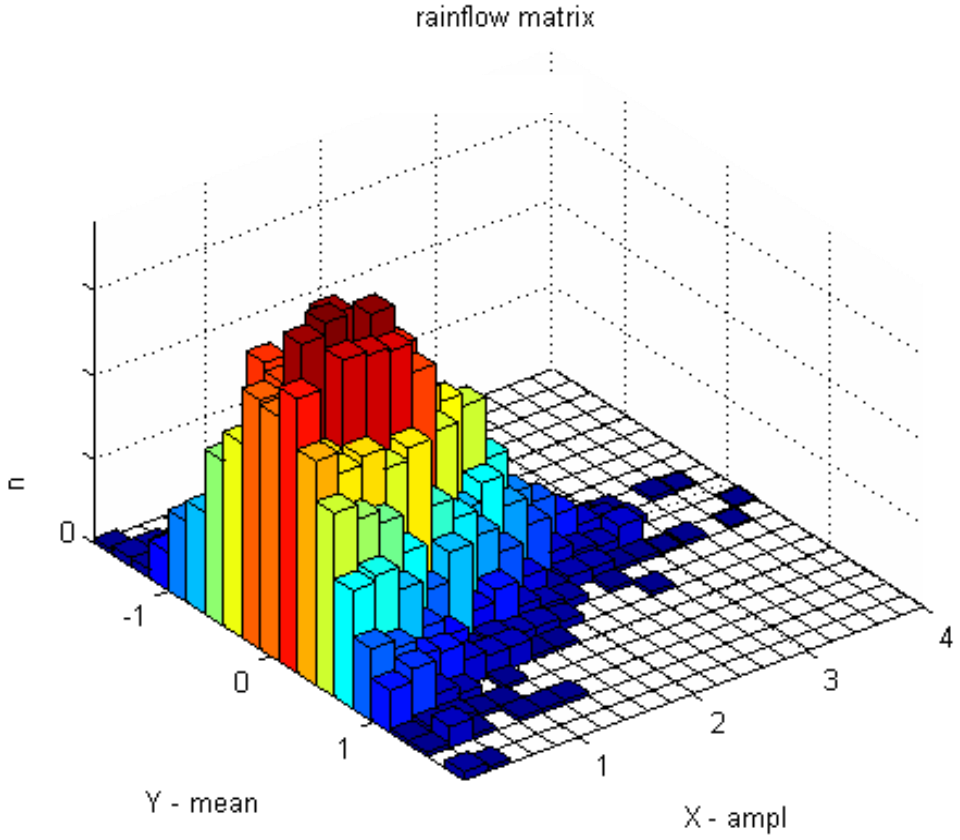
- le choix de l'origine peut changer l'estimation de $p(x)$
- Comment choisir h ?



Estimation non paramétrique

Histogramme (non paramétrique)

En deux dimensions, On peut reprendre la même formulation que précédemment étendue à une dimension 2 puis n dans le cas général,



Estimation non paramétrique

Histogramme (non paramétrique)

Problème de l'origine

Problème du choix de h (discrétisation)

Problème des grandes dimensions

Supposons que l'on ait des données de dimension 20 et que chaque dimension puisse prendre 5 valeurs, L'histogramme aura en tout $5^{20}=9.10^{13}$ cases.

➔ Il faudra une base de donnée énorme pour estimer correctement $p(x)$. Si la dimension est plus grande, le problème devient encore plus difficile

Estimation non paramétrique

Estimation par noyau (non paramétrique)

Pour remédier au problème de l'origine,

$$p(x) = \frac{nombre\ d'échantillons\ \ dans\ [x-h, x+h]}{N2h}$$

Où N est le nombre d'échantillons total
Ceci s'exprime mathématiquement en 1D par :

$$p(x) = \frac{1}{N2h} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right)$$

Avec $K\ (u) = \begin{matrix} 1\ si\ \ u\ < 1 \\ 0\ sinon \end{matrix}$

Ceci revient à compter le nombre d'échantillons tombant dans un hyper-cube de largeur h centré en x

Cette estimation est continue, elle est faite pour tout x

Estimation non paramétrique

Kernel Density Estimation (non paramétrique)

Pour remédier aux discontinuités liées à la discrétisation : **fenêtres de Parzen**. On estime toujours la densité de probabilité avec:

$$p(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right)$$

Mais K() peut être un noyau quelconque,

Exemple avec un noyau gaussien en 2D :

$$p(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{(2\pi)^{1/2} h} exp\left(-\frac{\|x - x_i\|^2}{2h^2}\right)$$

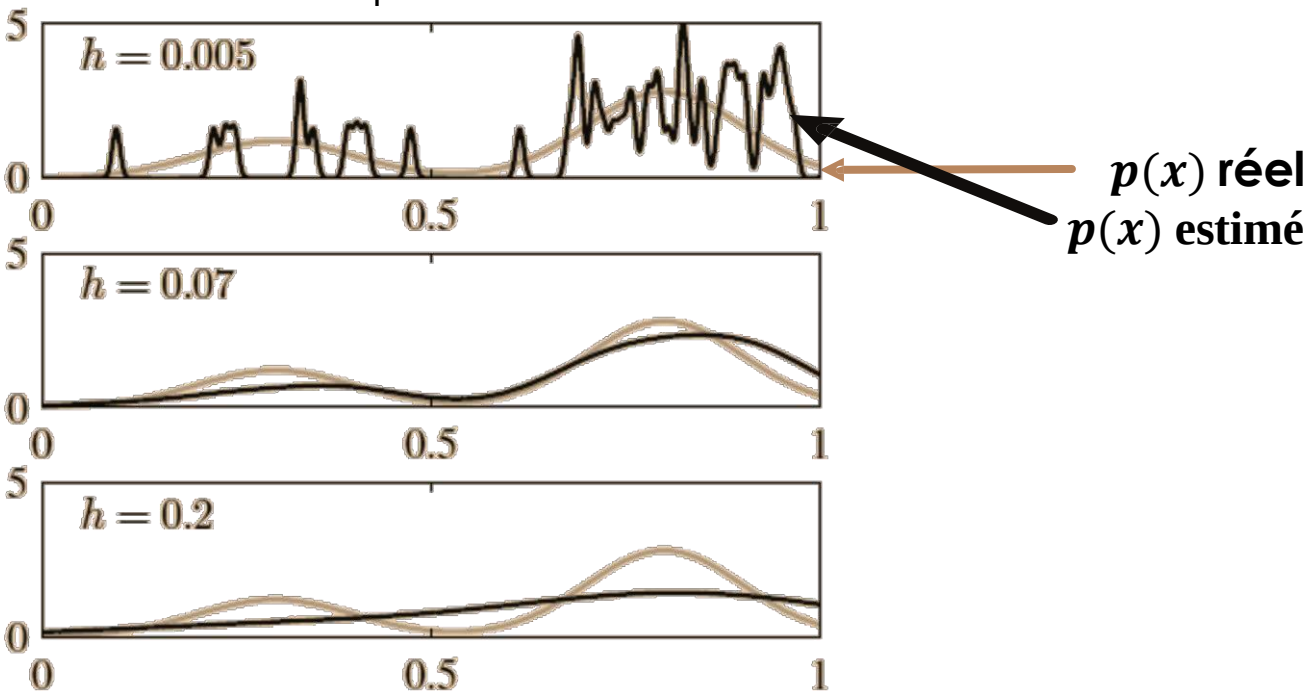
Ceci revient à placer une gaussienne autour de chaque point et à sommer leur contribution

Estimation non paramétrique

Kernel Density Estimation (non paramétrique)

Estimation de la densité de probabilité sur les mêmes données que pour l'histogramme

- h trop petit ➔ estimation très bruitée
- h trop grand ➔ estimation trop lisse



Estimation paramétrique

On souhaite comme précédemment estimer la densité de probabilité $p\ (\ x)$ à partir d'une réalisation de N échantillons,

Le problème est très difficile quand on a un faible nombre d'échantillons de dimension élevée.

La difficulté du problème est réduite si on connait a priori une forme paramétrique de la loi. Dans ce cas, il n'y a plus qu'à estimer les paramètres de la loi.

Ce problème est soluble par l'estimation du maximum de vraisemblance.

Estimation paramétrique

Estimation du maximum de vraisemblance

Nous disposons de N échantillons xi tirés à partir de la loi $p(x)$ D'autre part, $p\ (x)$ a une forme connue, dépendant de paramètres θ : $p\ (x)= f(x, \theta)$

Nous recherchons les paramètres θ qui maximise la vraisemblance des observations définie par:

$$L\ (\theta / x)=\prod_{i=1}^{N} p(x_i) = \prod_{i=1}^{N} f(x_i, \theta)$$

Il est souvent plus simple de travailler avec le logarithme de cette vraisemblance appelée log-vraisemblance:

$$l(\theta/x) = ln\left(\prod_{i=1}^{N} p(x_i)\right) = \sum_{i=1}^{N} ln\,(p(x_i)) = \sum_{i=1}^{N} \ln(f\,(x_i, \theta))$$

Estimation paramétrique

Loi binomial
Loi de Bernouilli
Loi uniforme
Loi normale mono variable
Loi normale multi variables
Mixture de gaussiennes

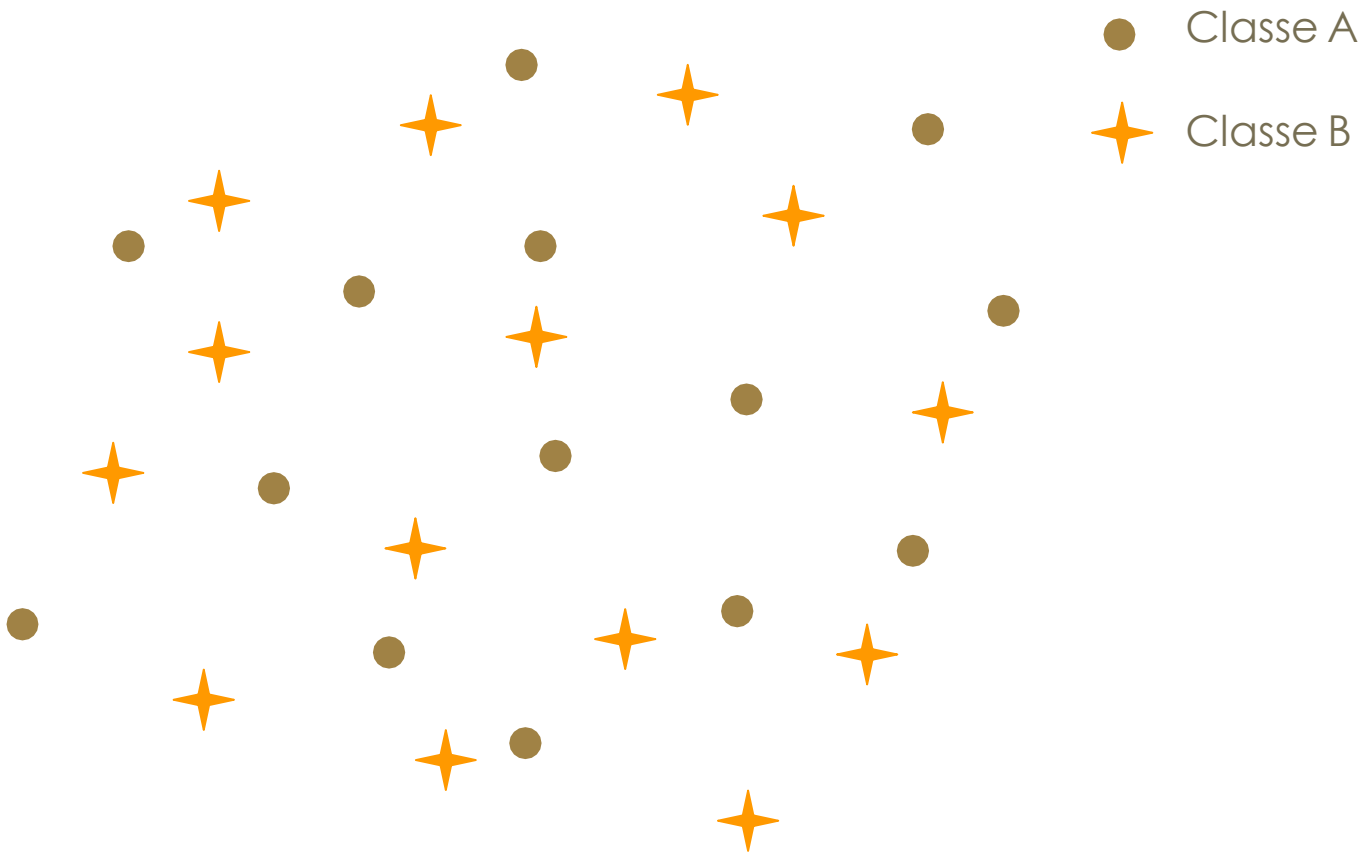
Qualité de la base de données

Plusieurs problèmes apparaissent:

- ✓Données inadaptées
- ✓Données aberrantes (outliers)
- ✓Données manquantes

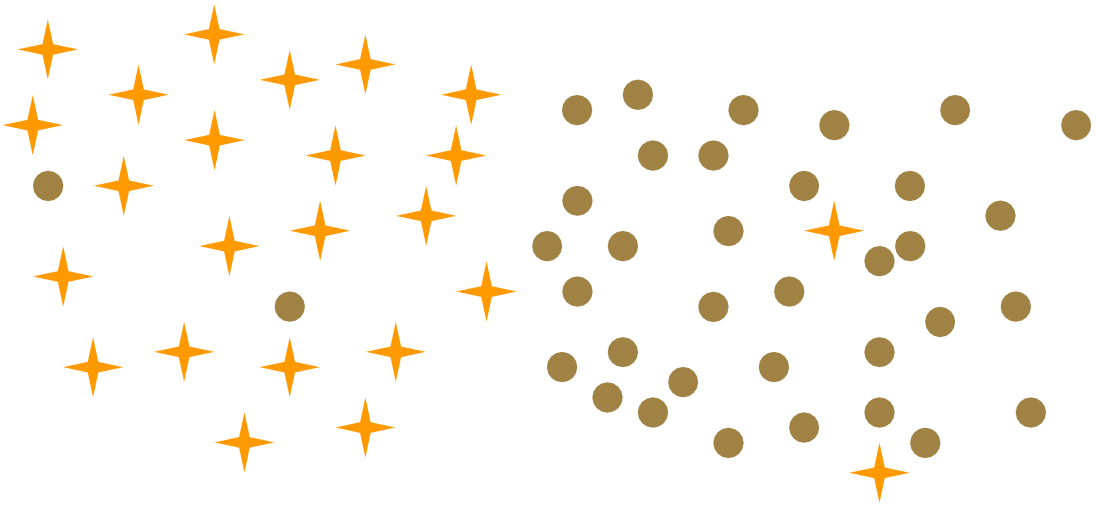
Qualité de la base de données

✓Données inadaptées: aucune cohérence n'apparaît



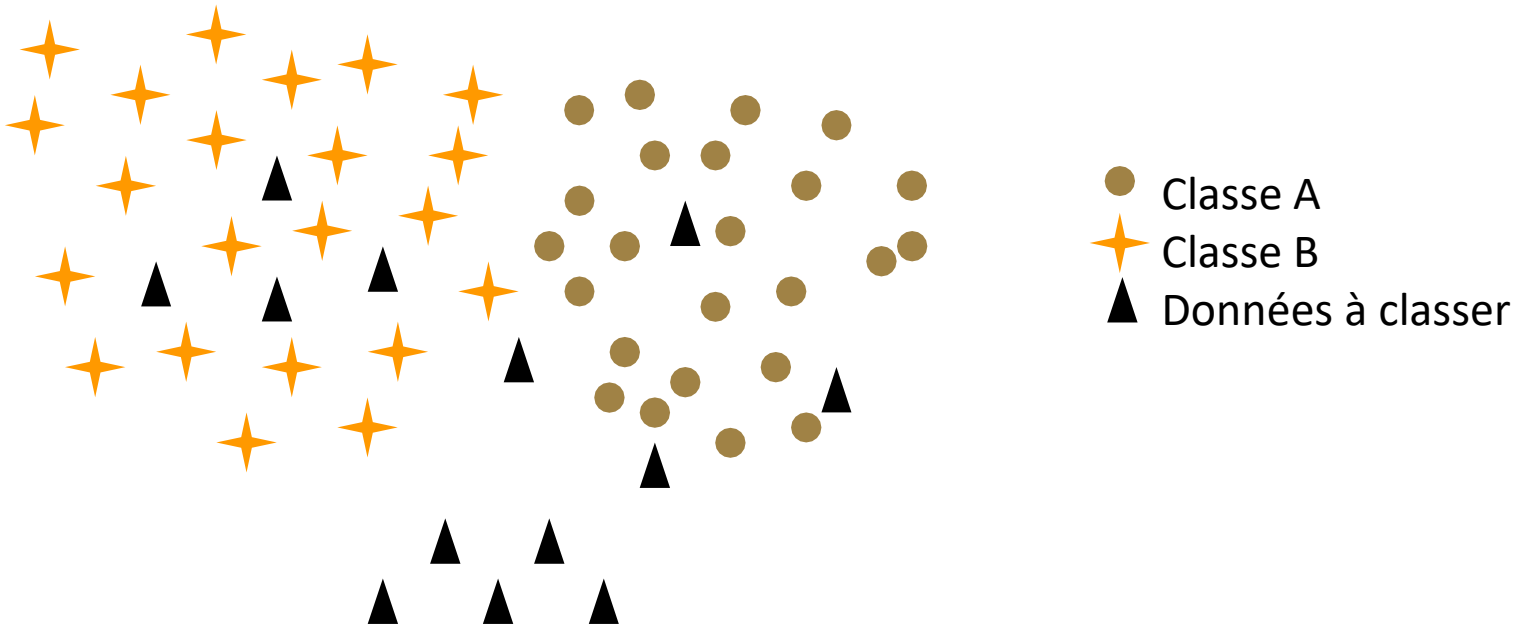
Qualité de la base de données

✓Données aberrantes



Qualité de la base de données

Données manquantes : les données ne recouvrent pas l'ensemble des configurations



Performances d'un classificateur

En RdF, 3 bases :

-**Une base de reference ou d'apprentissage** utilisée pour apprendre le classificateur

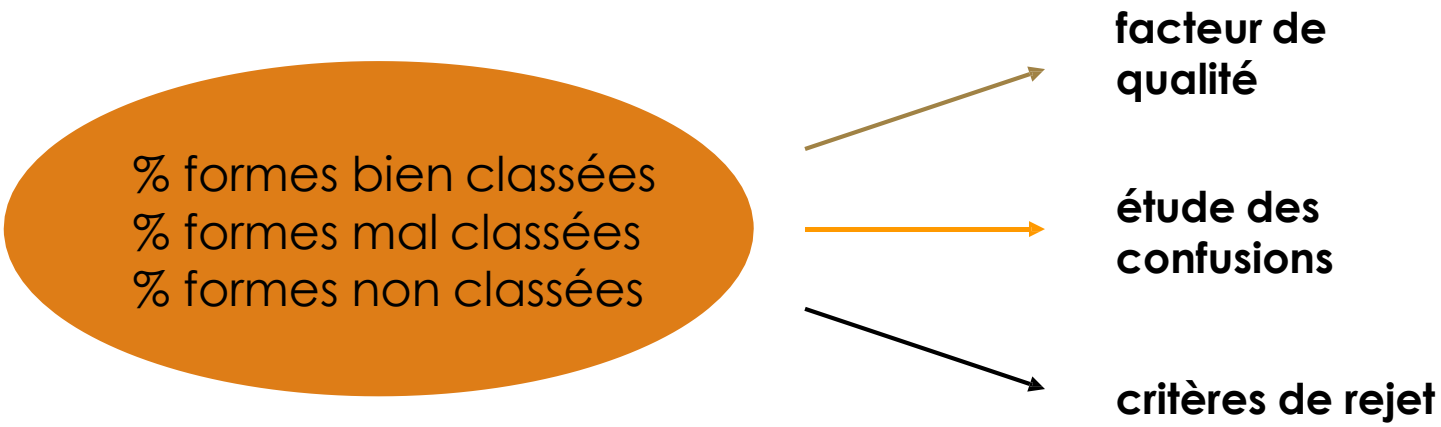
-**Une base de validation** pour déterminer les paramètres du classifieur

-**Une base de test** : exemples jamais vus au préalable pour évaluer le classificateur

Pourquoi ?

Performances d'un classificateur

En fonction des statistiques sur la base de test, on va pourvoir définir:



Performances d'un classificateur

Taux de bonne classification sans coûts

Sans rejet

Taux de bonne classification $$Tb_s = \frac{\text{Nb d'exemples bien classés}}{\text{Nb d'exemples}}$$

Taux d'erreur $$Te_s = 1 - Tb_s$$

Avec rejet

Taux de rejet $$Tr = \frac{\text{Nb d'exemples non classés}}{\text{Nb d'exemples}}$$

Taux de bonne classification $$Tb_a = \frac{\text{Nb exemples bien classés}}{\text{Nb exemples}}$$

Taux d'erreur $$Te_a = 1 - Tb_a - Tr$$

Performances d'un classificateur

Taux de bonne classification sans coûts

Problème :
Il s'agit d'une mesure faible qui **ne tient pas compte de la distribution des classes**

Exemple :
En diagnostic médical, très peu de personnes sont malades (5%?). On a donc des taux très bons en disant que la personne est saine. Or, ce que l'on souhaite, c'est ne pas rater ces 5% et donc, associer un mauvais taux au classificateur qui dirait toujours 'personne saine'.

Exemple sur 100 personnes

	malade	**sain**
malade	0	5
sain	0	95

Tbs=95%

Solution : On tient compte de la répartition des classes et on construit une **matrice de confusion normalisée**

Performances d'un classificateur

Problème 2 : Les performances dépendent des applications.

Exemple : en surveillance médicale, on préfère détecter à tort des maladies plutôt que de risquer d'en laisser passer ➔ on admet beaucoup de fausses alarmes mais pas de manque de détection

➔ On introduit une **matrice des coûts**

Problème : Comment définir les coûts ???

Performances d'un classificateur

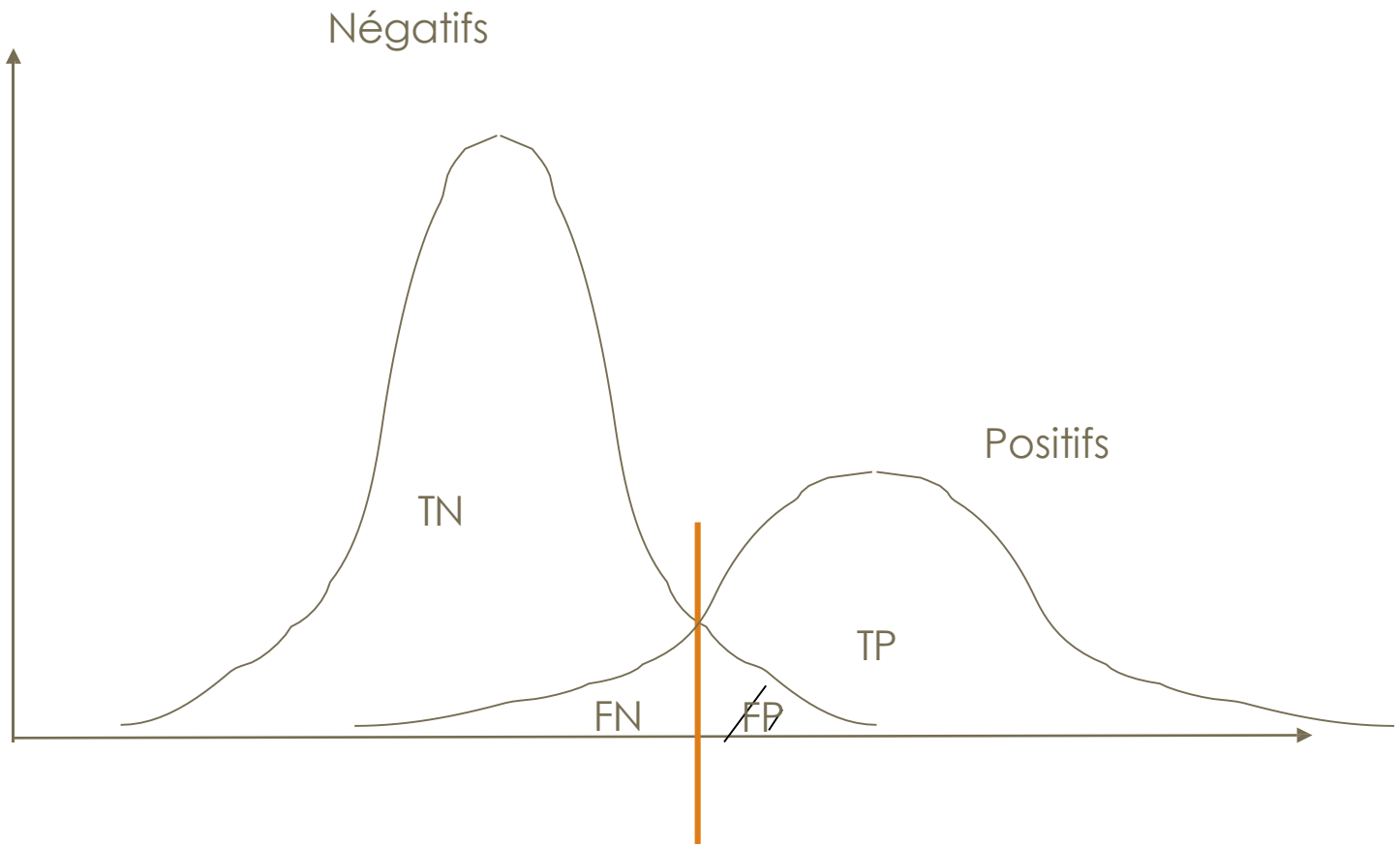
Comment comparer plusieurs classificateurs indépendamment du seuil ?
Problèmes à 2 classes

On définit les :
- ✓Vrai Positif (True Positive)
- ✓Vrai Négatif (True Négatif)
- ✓Faux Négatif (False Négatif)
- ✓Faux Positif (False Positif)

		Trouvé par le classificateur	
		+	-
réel	+	TP	FN
	-	FP	TN

Performances d'un classificateur

Comment comparer plusieurs classificateurs indépendamment du seuil ?
Problèmes à 2 classes

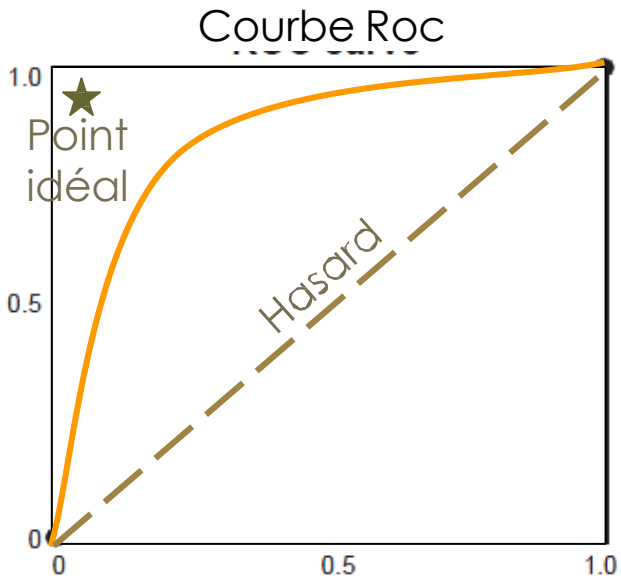


Performances d'un classificateur

Courbe ROC (Receiver Operating Characteristic)

Pour des problèmes binaires

décision / étiquette	+	-
+	TP	FN
-	FP	TN



$$\text{Sensibilité} = \frac{TP}{TP + FN} = \text{Parmi les positifs de la base, \% de corrects}$$

$$\textit{Spécificité} = \frac{TN}{FP + TN} = \text{parmi les négatifs de la base, \% de corrects}$$

Toutes les courbes ROC passent par l'origine et par le point (1,1)

Un bon classificateur devra être
✓sensible : détecter les positifs
✓spécifique : ne pas détecter aussi les négatifs

Généralement, plus un classificateur est sensible, moins il est spécifique et vice versa

Performances d'un classificateur

Cas multi-classes Matrice de confusion:

	Trouvé par le classificateur		
réel	C0	C1	C2
C0	70	11	35
C1	17	73	8
C2	45	5	53

$$\text{Taux de reconnaissance} = \frac{\text{Somme des éléments sur la diagonale}}{\text{Somme des éléments}}$$

Méthodes génératives/discriminatives

3 approches différentes

- Approche générative
- Approche discriminative
- Fonction discriminante

Méthodes génératives/discriminatives

Approche générative : déterminer les densités de probabilités conditionnelles $p(x|C_k)$ et les densités de probabilités a priori $p(C_k)$ pour chaque classe individuellement. Puis utiliser le théorème de Bayes.

Approche discriminative : Déterminer directement $p(C_k|x)$ et décider de la classe

Fonction discriminante: trouver une fonction $f(x)$ reliant directement les données aux classes. Ex : pour un problème a deux classes, $f(x)$ est une fonction à valeur binaire tq $f(x) = 0$ pour la première classe et $f(x) = 1$ pour la seconde (aucune notion de probabilité)

Méthodes génératives/discriminatives

Avantage/inconvénient

Approche générative

+ $p(x)$ est estimée. On peut considéré $p(x)$ comme la probabilité que x soit bien modélisé par le modèle. Ceci permet de faire du rejet.

+ $p(x|C_k)$ peut être utilisée pour générer des données

+ Permet à un système d'utiliser une seule classe. Ex : la teinte chaire

- Trouver $p(x|C_k)$ pour chaque classe est très couteux en temps de calcul, surtout quand x est de grande dimension

- Nécessite une grande base de données, surtout quand x est de grande dimension

Approche discriminative

+ Il est beaucoup plus rapide de déterminer $p(C_k|x)$ car la dimension de C_k est bien souvent beaucoup plus faible que celle de x

Fonction discriminante

+ Modélisation et décision sont combinées dans un seul apprentissage

- $p(C_k|x)$ n'est pas estimé. On ne pourra donc (*i*) ni faire du rejet; (*ii*) ni combiner plusieurs classificateurs; (*iii*) ni compenser différentes probabilités *a priori* des classes

Méthodes génératives/discriminatives

Méthodes génératives	Méthodes discriminatives
Classification bayésienne	K plus proches voisins
Modélisation gaussienne	Arbres de décision
GMM (Gaussian Mixture Model)	Régression linéaire
HMM (Hidden Markov Model)	SVM (Support Vector Machine
Réseaux bayésiens	RVM (Relevance Vector Machine)
MRF (Markov Random Fields)	Réseaux de neurones
	CRF (Conditional Random fields)