

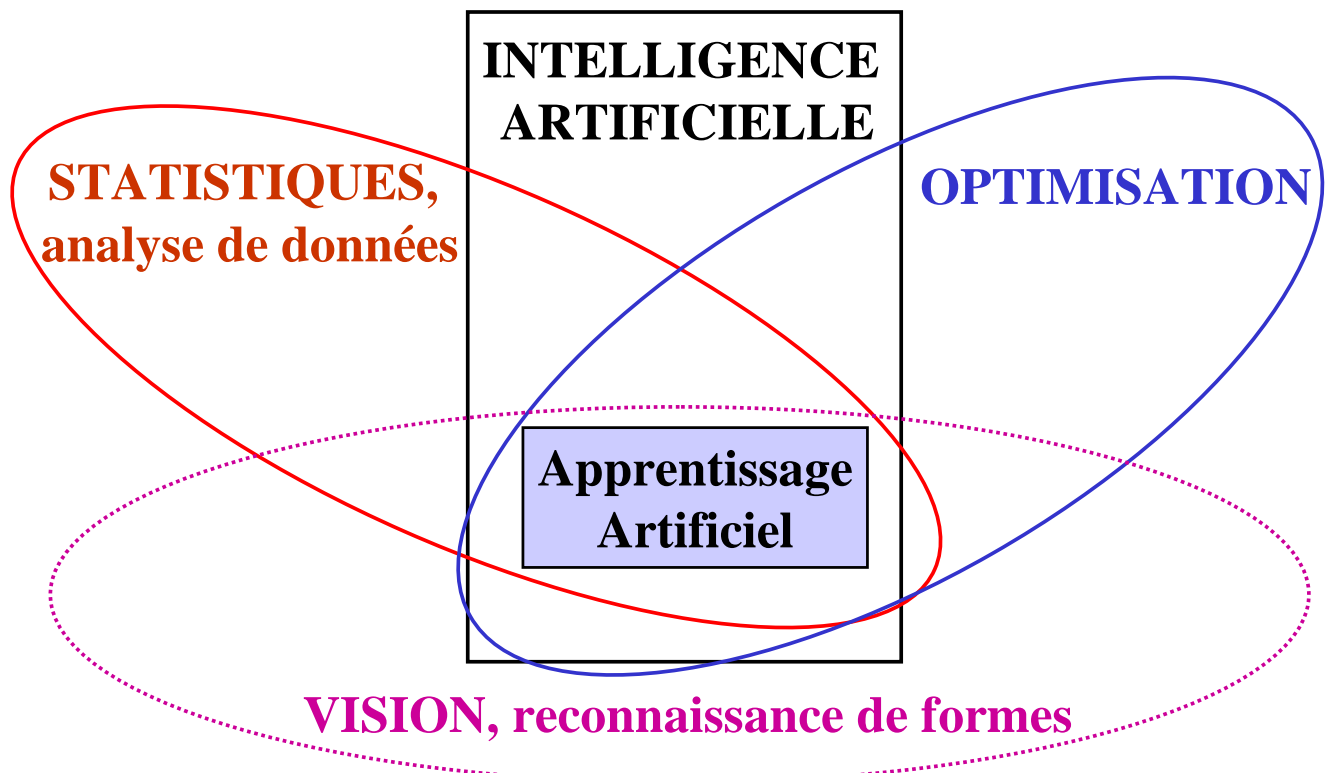
APPRENTISSAGE ARTIFICIEL/STATISTIQUE (« Machine-Learning »)

1. Introduction et Généralités

Fabien Moutarde
Centre de Robotique (CAOR)
MINES ParisTech (Ecole des Mines de Paris)

Fabien.Moutarde@mines-paristech.fr
<http://www.mines-paristech.fr/~moutarde>

Un domaine interdisciplinaire

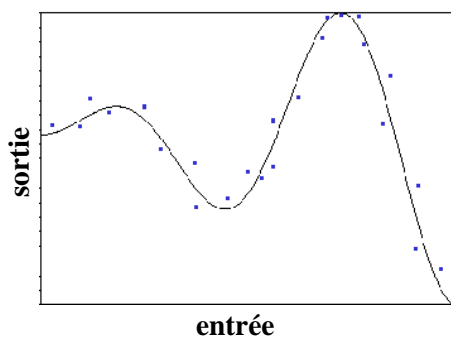


« Capacité d'un système à améliorer ses performances via des interactions avec son environnement »

- **Quel « système » ?**
→ types de modèle (Ad hoc ? Issu d'une famille particulière de fonctions mathématiques [tq splines, arbre de décision, réseau de neurones, arbre d'expression, machine à noyau...] ?)
- **Quelles « interactions avec l'environnement » ?**
→ apprentissage « hors-ligne » v.s. « en-ligne »
→ apprentissage « supervisé » ou non, « par renforcement »
- **Quelles « performances » ?**
→ fonction de coût, objectif, critère implicite, ...
- **Comment améliorer ?**
→ type d'algorithme (gradient, résolution exacte problème quadratique, heuristique, ...)

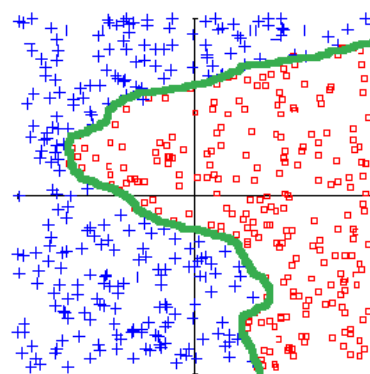
APPRENTISSAGE SUPERVISÉ : régression et classification

Régression (approximation)



points = exemples → courbe = régression

Classification ($y_i = \ll \text{étiquettes} \gg$)

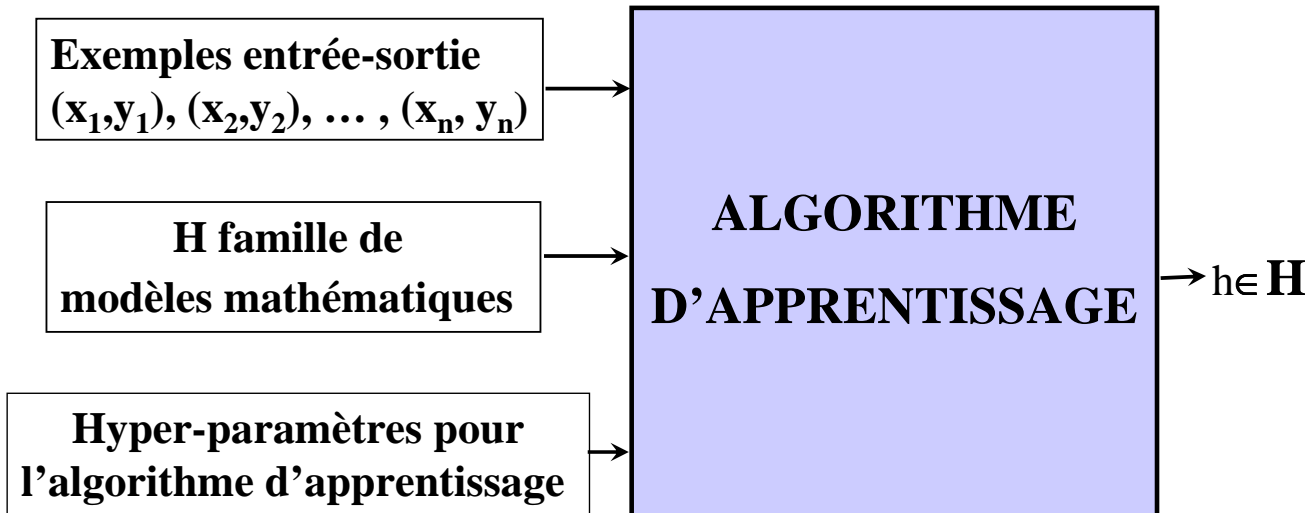


entrée =
position point
sortie désirée =
classe ($\square = -1, + = +1$)

↓
Fonction
étiquette = $f(x)$
(et frontière de
séparation)

Problème

- Ensemble d'entraînement $\mathcal{S} = \{(x_1, y_1), \dots, (x_n, y_n)\}$.
- Construire un classifieur h à partir de \mathcal{S} , tel que $\hat{y} = h(x)$ prédise la classe d'un x inconnu le plus précisément possible.



Paradigme d'apprentissage

Chaque paradigme se caractérise par :

Un modèle, le plus souvent paramétrique

+

Une façon d'interagir avec l'environnement

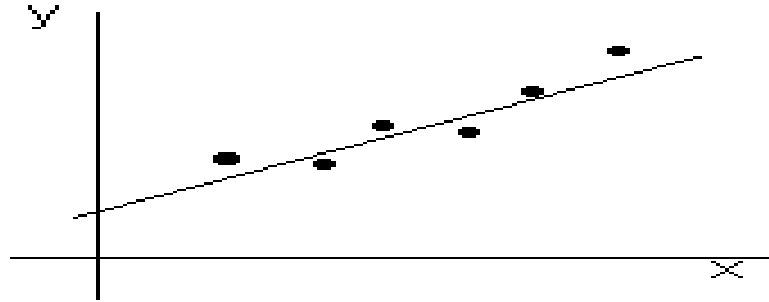
+

Une « fonction de coût » à minimiser (sauf exceptions)

+

**Un algorithme pour adapter le modèle,
en utilisant les données issues de l'environnement,
de façon à optimiser la fonction de coût**

Exemple trivial : régression linéaire par moindres carrés



- **Modèle** : droite $y=ax+b$ (2 paramètres a et b)
- **Interaction** : collecte *préalable* de n points $(x_i, y_i) \in \mathbb{R}^2$
- **Fonction de coût** : somme des carrés des écarts à la droite $\kappa = \sum_i (y_i - a \cdot x_i - b)^2$
- **Algorithme** : résolution directe (ou itérative) du système linéaire

$$\begin{pmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & n \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n y_i \end{pmatrix}$$

Nombreux paradigmes

- **Régression linéaire par moindre carrés**
- **Plus Proches Voisins (kPPV, alias kNN)**
- **Algo ID3 ou CART pour arbres de décision**
- **Méthodes probabilistes**
- ...
- **Rétropropagation du gradient sur réseau neuronal à couches**
- **Cartes topologiques de Kohonen**
- **Support Vector Machines**
- **Boosting de classifieurs faibles**
- ...

- **Résolution système linéaire** (régression, Kalman, ...)
- **Algos classiques d'optimisation**
 - Descente de gradient, gradient conjugué, ...
 - Optimisation sous contrainte
 - ...
- **Heuristiques diverses :**
 - Algo d'auto-organisation non supervisée de Kohonen
 - Algorithmes évolutionnistes (GA, GP, ...)
 - « colonies de fourmis » (Ant Colony Optimization)
 - Optimisation par Essaim Particulaire (OEP)
 - Renforcement (Q-learning, ...)

APPRENTISSAGE SUPERVISÉ : définition formelle

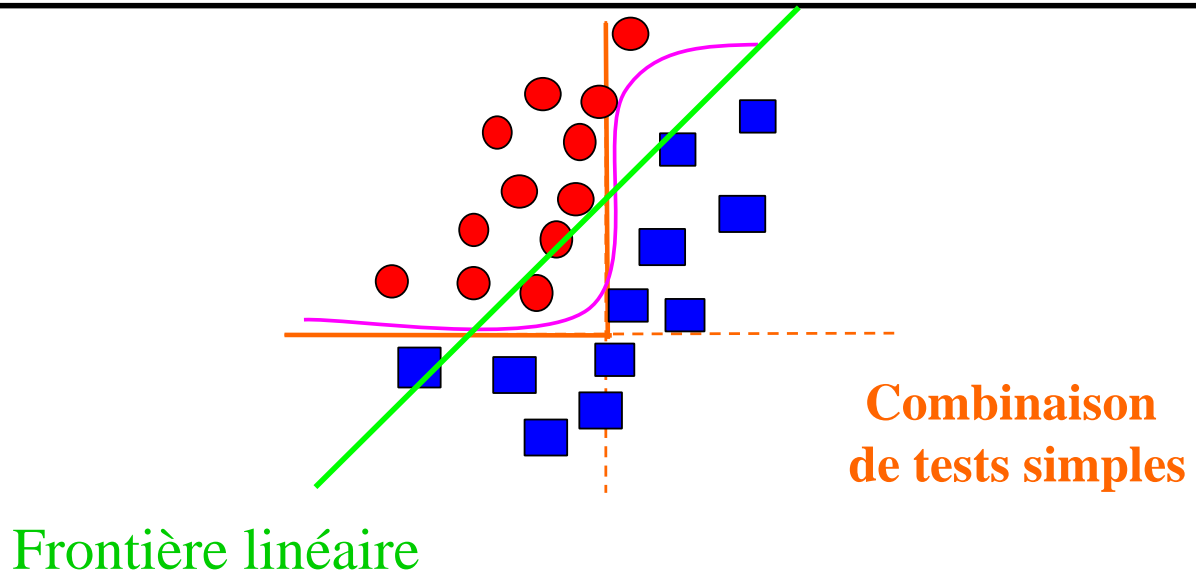
« **APPRENDRE = INFERER/INDUIRE + GENERALISER** »

- Etant donné un ensemble *fini* d'exemples $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, où $x_i \in \mathbb{R}^d$ vecteurs d'entrée, et $y_i \in \mathbb{R}^s$ sorties désirées (fournies par le « superviseur »), trouver une fonction h qui « approxime et généralise au mieux » la fonction sous-jacente f telle que $y_i = f(x_i) + \text{bruit}$

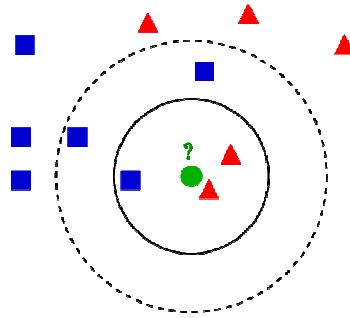
⇒ but = minimiser erreur de généralisation

$$E_{\text{gen}} = \int \|h(\mathbf{x}) - f(\mathbf{x})\|^2 p(\mathbf{x}) d\mathbf{x}$$

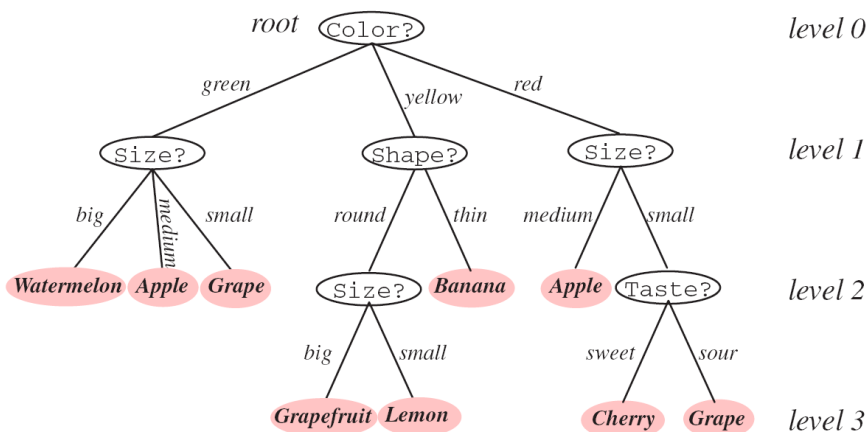
(où $p(\mathbf{x}) = \text{distrib. de proba de } \mathbf{x}$)



- **Par analogie → Plus Proches Voisin (PPV)**
- **Par combinaison de tests élémentaires :**
 - **Arborescence → Arbre de Décision Binaires (ADB)**
 - **Vote pondéré → boosting (dopage)**
- **Par approche probabiliste (avec hypothèses sur distribution des classes) → méthodes bayésiennes**
- **Par minimisation de l'erreur (descente de gradient, etc..) → Réseaux de neurones (MLP), etc...**
- **Par maximisation de la « marge » → Support Vector Machines (SVM)**



Exemple de classification par les plus proches voisins

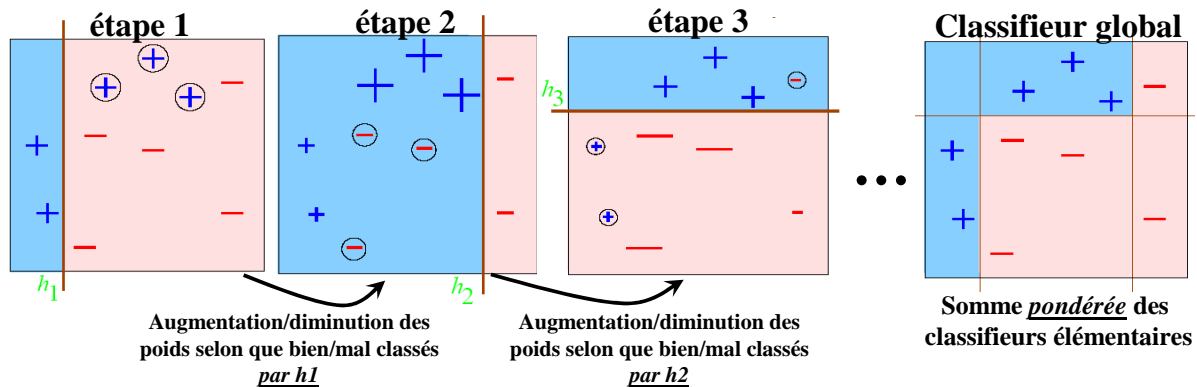


Exemple de classification par Arbre de Décision Binaire

Principe : vote (pondéré) de nombreuses règles simples

- cheval qui a gagné le + souvent récemment
- cheval qui a la meilleur cote pour la course
- cheval qui a eu meilleurs classements sur cet hippodrome
- ...

Illustration



Méthodes probabilistes

Principe : probabilité de classe C connaissant features F1, F2, ..., Fn

$$p(C|F_1, \dots, F_n) = \frac{p(C) p(F_1, \dots, F_n|C)}{p(F_1, \dots, F_n)} \quad (\text{théorème de Bayes})$$

$$\propto p(C, F_1, \dots, F_n)$$

$$= \frac{1}{Z} p(C) \prod_{i=1}^n p(F_i|C) \quad (\text{hypothèse indépendance})$$

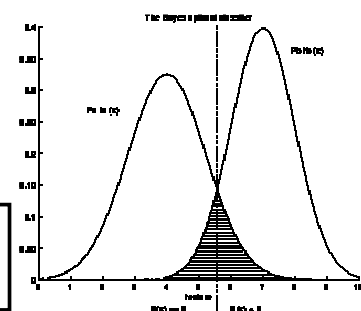
→ Classifieur MAP (Maximum A Posteriori)

$$\text{classify}(f_1, \dots, f_n) = \underset{c}{\operatorname{argmax}} p(C=c) \prod_{i=1}^n p(F_i=f_i|C=c)$$

Proba « A priori »
(prior)

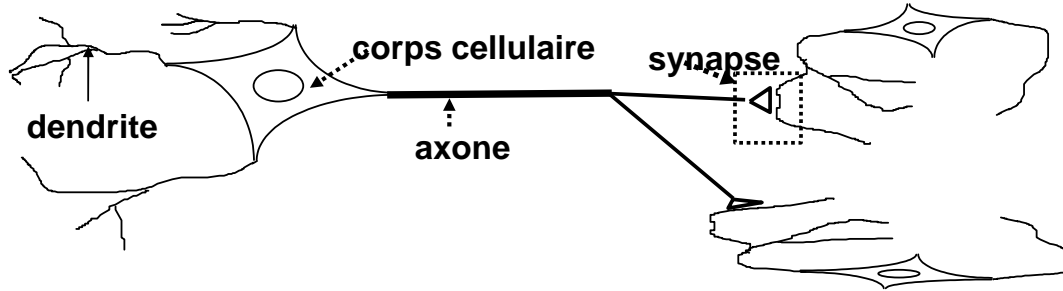
Vraisemblance
(likelihood)

Estimées (avec ou sans hypothèse sur leur forme)
sur l'ensemble des exemples d'apprentissage

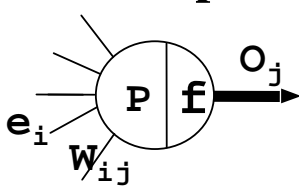


RESEAUX NEURONAUX

- Inspirés de l'architecture et fonctionnement cerveau



- Modèle mathématique paramétré simple + algos d'adaptation des paramètres



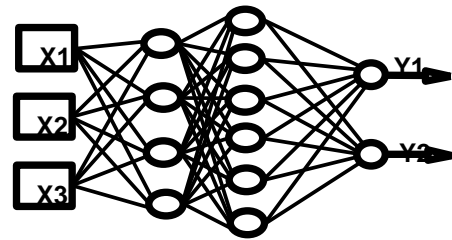
$$O_j = f(P(\vec{e}, \vec{W}_j))$$

avec par exemple

$$P(\vec{e}, \vec{W}_j) = \sum_i e_i W_{ij}$$

$$f(p) = \tanh(p)$$

neurone « formel »

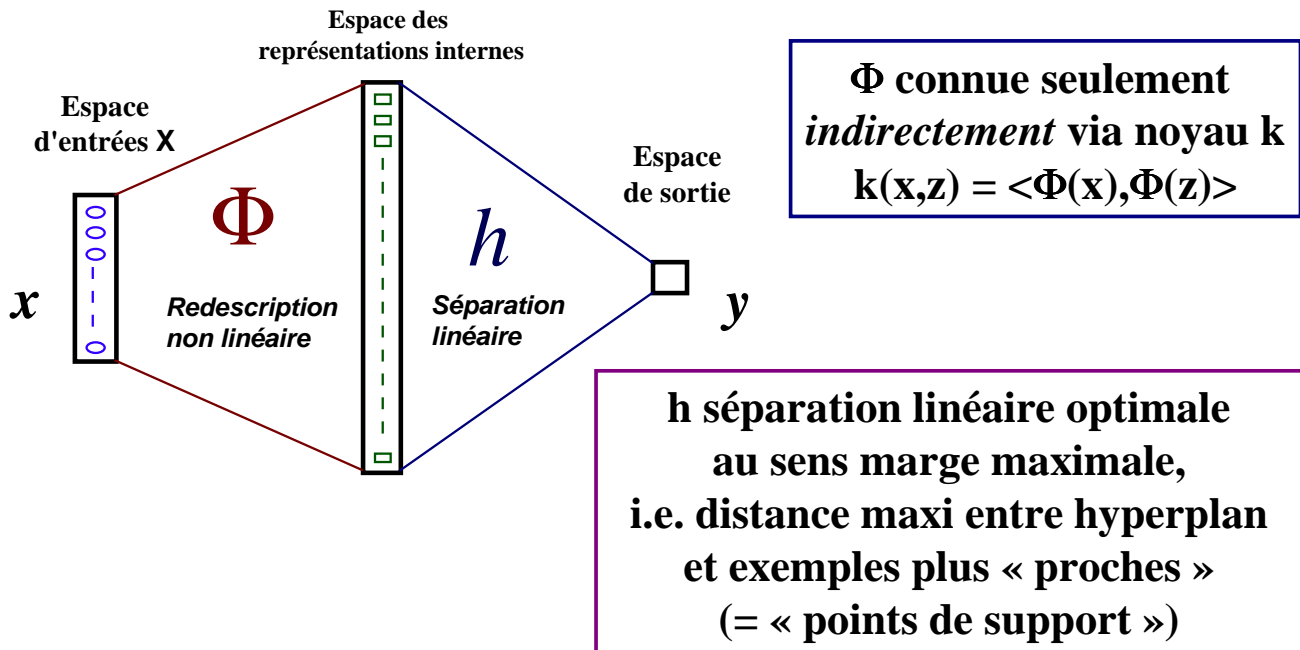


Réseau =
assemblage de neurones

RESEAUX NEURONAUX (2)

- Apprentissage = à partir d'exemples de couples (entrée, sortie), le réseau modifie :
 - les paramètres W (poids des connexions)
 - éventuellement son architecture A
(en créant/éliminant neurones ou connexions)

Plus de détails sur divers types de neurones, de réseaux et les algorithmes d'apprentissage dans le cours dédié aux réseaux neuronaux...



Plus de détails dans partie du cours consacrée à cette technique

Spécificité de la classification d'images

- Etape préalable : quelles caractéristiques (features) mettre en entrée du classifieur ??
 - les pixels (après quels redimensionnement/normalisation ?)



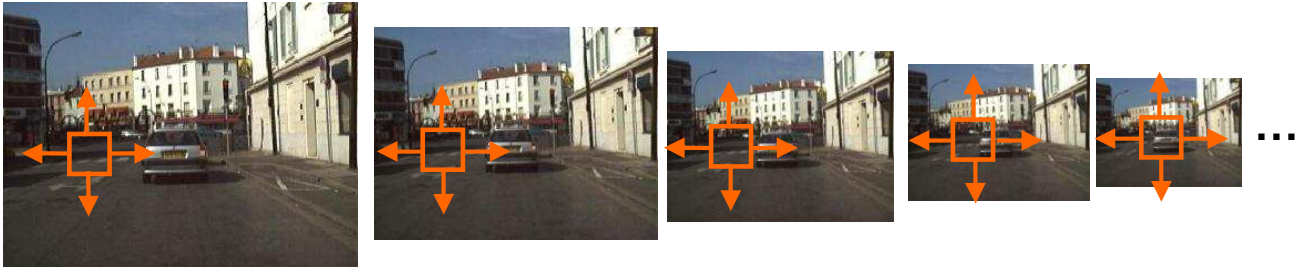
Egalisation histogramme

- des histogrammes ?
- des infos fréquentielles (FFT, etc...) ?
- des résultats de filtres (eg Gabor pour texture) ?

➔ Chaque exemple (image) devient un vecteur de \mathcal{R}^d

Détection multi-résolutions avec un classifieur unique (« window-scanning »)

- Pour chaque image traitée de la vidéo :
 - construire une pyramide de ~12 images par sous-échantillonnages
 - scanner chaque image de la pyramide avec une fenêtre de détection de taille fixe (e.g. 36x36 pixels pour la détection de vue arrière de voiture)
→ plusieurs dizaines de milliers d'images correspondant à des sous-fenêtres de tailles et positions diverses dans l'image initiale



- avec un unique classifieur, évaluer pour chacune de ces images si elle est correctement centrée sur un objet du type recherché (e.g. vue arrière d'une voiture)

→ Besoin uniquement d'un classifieur objet_cherché/autre pour images de taille fixée (e.g. 36x36 pixels)

QUELQUES REFERENCES SUR L'APPRENTISSAGE STATISTIQUE

- *Apprentissage artificiel : concepts et algorithmes*, A. Cornuéjols, L. Miclet & Y. Kodratoff, Eyrolles, 2002.
- *Pattern recognition and Machine-Learning*, Christopher M. Bishop, Springer, 2006.
- *Introduction to Data Mining*, P.N. Tan, M. Steinbach & V. Kumar, AddisonWesley, 2006.
- *Machine Learning*, Thomas Mitchell, McGraw-Hill Science/Engineering/Math, 1997.