Université de Montréal

Département d'informatique et de recherche opérationnelle

IFT3390/6390
Fondements de l'apprentissage machine

http://www.iro.umontreal.ca/~vincentp/ift3390

Troisième cours:

Méthodes de type histogramme: curse of dimensionality

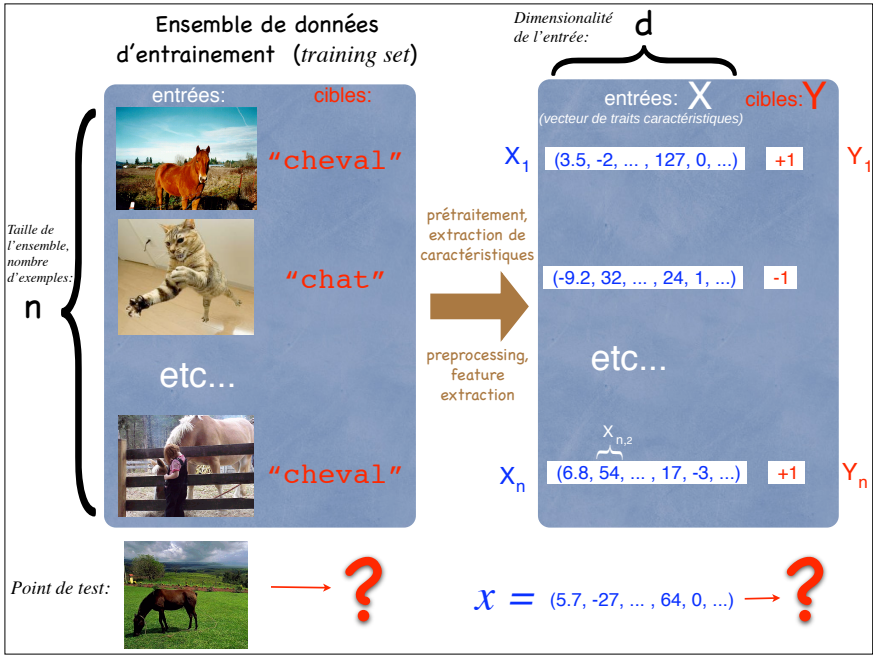
Formalisation du problème de l'apprentissage

Professeur: Pascal Vincent

LISA Laboratoire d'Informatique des Systèmes d'Apprentissage

Au programme aujourd'hui

- Petit rappel de terminologie.
- Méthodes de type histogramme, illustrées pour classification, régression, estimation de densité.
- Malédiction de la dimensionalité.
- Formalisation mathématique du problème de l'apprentissage. Notion de capacité.



Une idée simple: découper l'espace en petits cubes...

Les algorithmes à base de quadrillages de l'espace (de type histogramme)

Une idée simple pour la classification

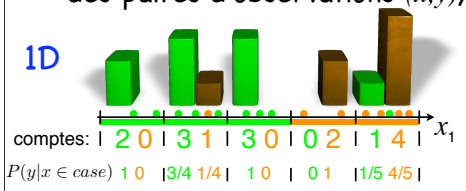
Tout algo d'apprentissage doit pouvoir effectuer une prédiction pour n'importe quel point de test de l'espace d'entrée... (ex: $x \in \mathbb{R}^d$)
Partant de là, voici une idée simple d'algorithme:

- Quadriller l'espace!
- Entraînement: Compter, pour chaque case, combien de points de chaque classe y tombent (parmi les points de l'ensemble d'apprentissage).
- Test: trouver la case dans laquelle tombe le point de test. Répondre la classe majoritaire tombée dans cette case.

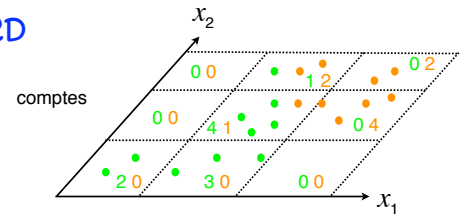
Classification

- On suppose qu'il existe un processus inconnu qui génère des paires d'observations (x,y), ou y indique la classe (• ou •)

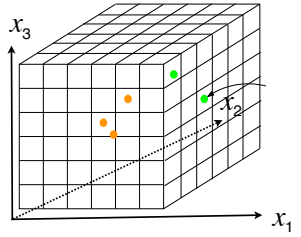
1D



2D



3D

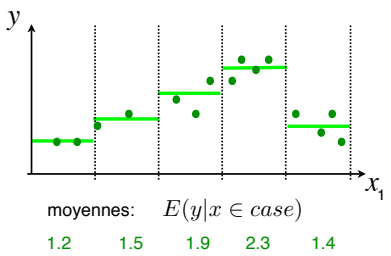


dD ...

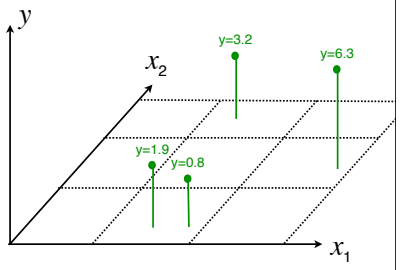
Régression

- On suppose qu'il existe un processus inconnu qui génère des paires d'observations (x,y), avec y réel.

1D



2D

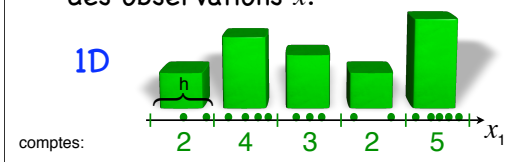


3D ... dD ...

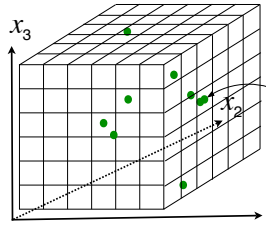
Estimation de densité

- On suppose qu'il existe un processus inconnu qui génère des observations x.

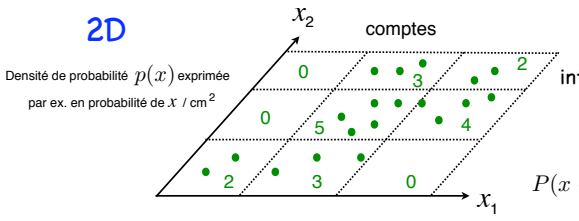
1D



3D



2D



La densité de prob. doit intégrer à 1 sur tout l'espace:

$$\int p(x)dx = 1$$

$$P(x \in region) = \int_{region} p(x)dx$$

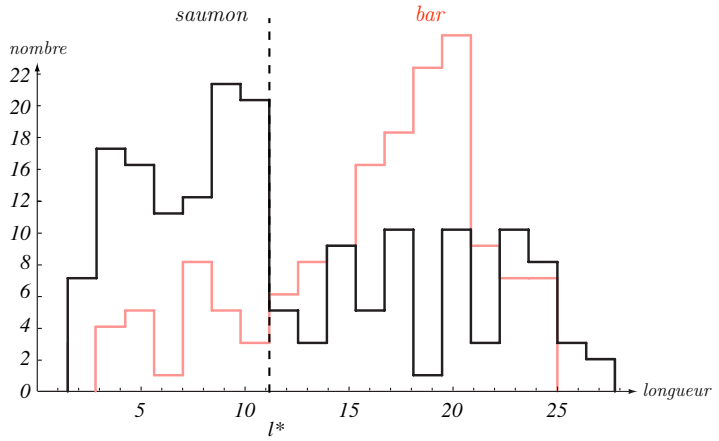
Quelle dimensionalité? exemple...

Exemple de classification

- Séparer deux types de poissons (saumon et bar) sur un tapis roulant
  - entrée des données (caméra)
  - traitement d'image
  - extraction des caractéristiques/traits (largeur, longueur, luminosité, etc.)
  - design d'une fonction de classification:

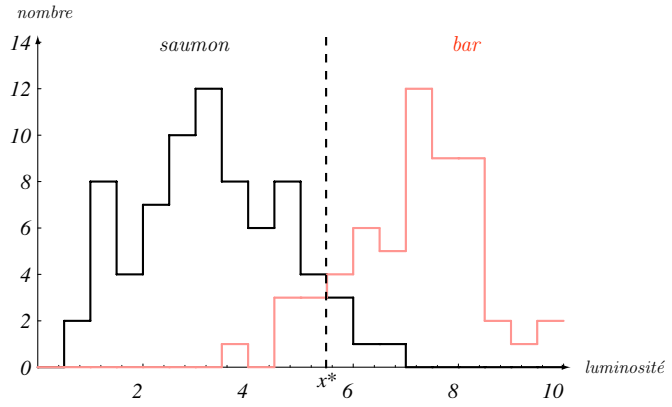
$$f : \{\text{vecteur des traits}\} \mapsto \{\text{saumon}, \text{bar}\}$$

- Histogramme obtenu de l'ensemble d'entraînement
  - erreur d'entraînement



Nombre d'erreurs: 26+69 = 95 *pour un classifieur linéaire*

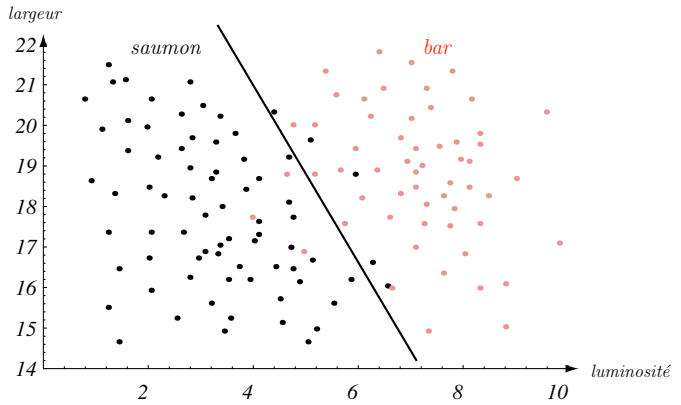
- Une autre variable/trait
  - coût de la mauvaise classification



Nombre d'erreurs: 7+5 = 12

- Deux variables
  - vecteurs de traits, espace de traits, frontière de décision

pour un classifieur linéaire



Nombre d'erreurs: 3+5 = 8

- Plus de dimensions (traits caractéristiques) c'est (généralement) plus d'information pour prendre la bone décision.
- Les classes en sont plus facilement séparables
- C'est bien mais....

Malédiction (fléau) de la dimensionalité
CURSE OF DIMENSIONALITY

Ex: combien de cases pour un quadrillage découpé en 10 en dimension d ?

- d=1 : 10 cases
- d=2 : 10x10=100 cases
- d=3 : 10x10x10=1000 cases
- d=10 : 10^{10} = 10 000 000 000 cases
  dix milliards!
- Pour un quadrillage où chaque dimension est découpé en m, on a d^m cases.

La "taille" de l'espace explorable à modéliser croît exponentiellemet avec la dimensionalité !

Si on a n=100 000 points d'entrainement répartis ± uniformément

- d=1 : 100 000/10 = 10000 points/case
- d=2 : 100 000/100 = 1000 points/case
- d=3 : 100 000/1000 = 100 points/case
- d=10 : 100 000/10^{10} = 10^{-5} points/case
- d=100 : 100 000/10^{100} = 10^{-95} points/case
- En haute dimension, la plupart des cases (où risque d'apparaître un point de test...) vont êtes vide!!!

Sensibilité à la malédiction

- Les méthodes de type histogramme (quadrillage) fonctionnent bien en faible dimension (1, 2, voire 3)
- Mais sont catastrophiques en haute dimension!
- La malédiction de la dimensionalité affecte ± tous les algorithmes d'apprentissage, mais certains y sont beaucoup plus sensible que d'autres.

Formalisation du problème de l'apprentissage

Machine d'apprentissage

- Formellement

$$g(\text{données}, \text{observation}) \mapsto \text{classe}$$

  - données d'entraînement

$$\text{données} = \{(\text{observation}_1, \text{classe}_1), \ldots, (\text{observation}_n, \text{classe}_n)\}$$

  - algorithme de classification

$$\textsc{Algo}(\text{données}) \mapsto f$$

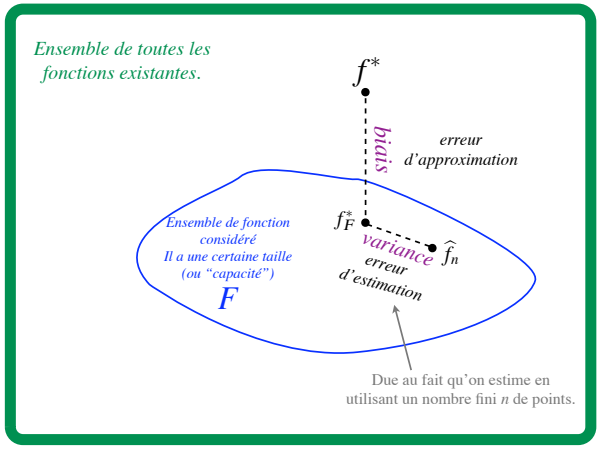
  - fonction de décision/classification

$$f(\text{observation}) \mapsto \text{classe}$$

Machine d'apprentissage

- Attributs des algorithmes
  - classe (parametrisée) de fonctions
    (linéaire, mélange de noyaux, etc.)
  - fonction d'objectif/cible/erreur
    ($0-1$, absolu, quadratique, etc.)
  - méthodes d'optimisation
    (descente de gradient, EM, optimisation quadratique, etc.)

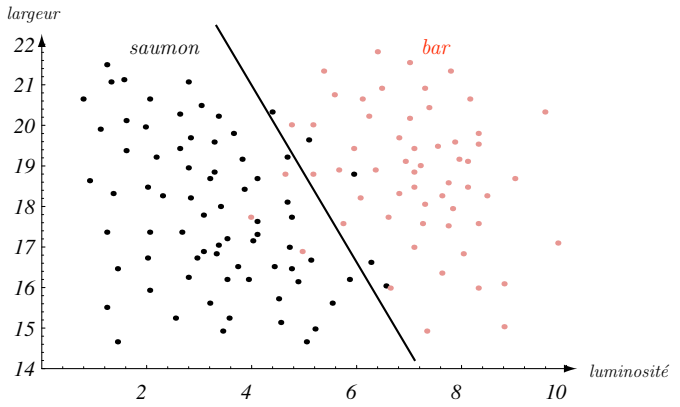
Le dilemme biais–variance



Les notions de capacité et de surapprentissage

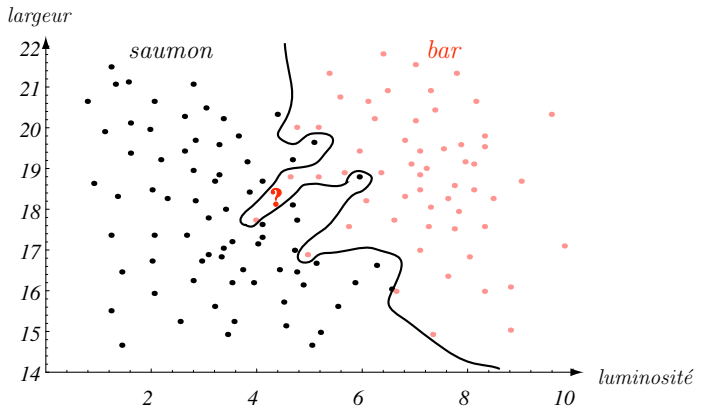
- Deux variables
  - vecteurs de traits, espace de traits, frontière de décision

pour un classifieur linéaire



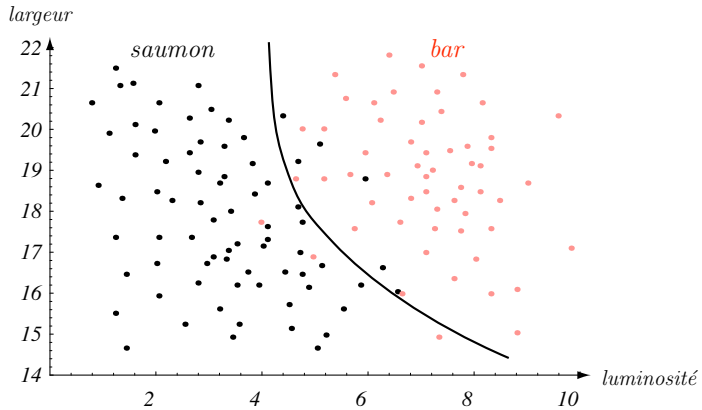
Nombre d'erreurs d'entraînement: 3+5 = 8

- Choix de fonction
  - surapprentissage (overfitting): classe de fonctions trop riche



Nombre d'erreurs d'entrainement: 0

- Choix de fonction
  - capacité optimale



Nombre d'erreurs d'entrainement: 3+6 = 9

- Choix de fonction
  - équilibre entre l'erreur d'entraînement et le complexité de décision
    capacité de l'ensemble de fonction
  - dilemme biais-variance
  - malédiction de la dimensionnalité