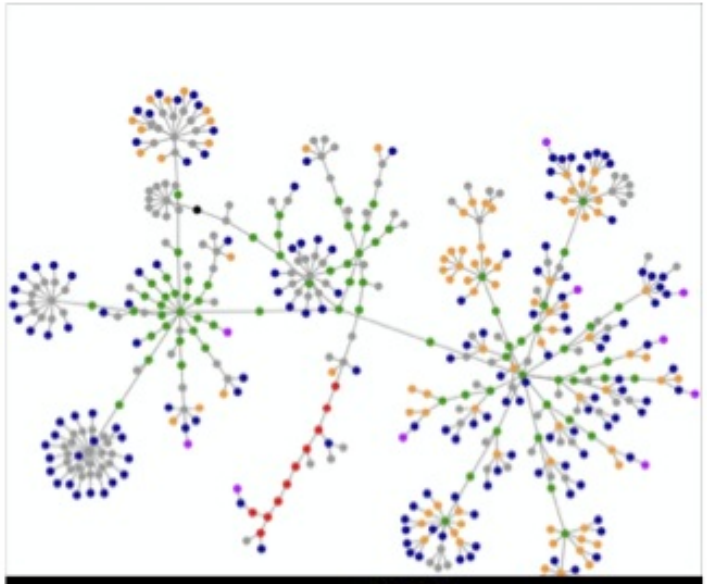
Introduction à l'apprentissage statistique - 1

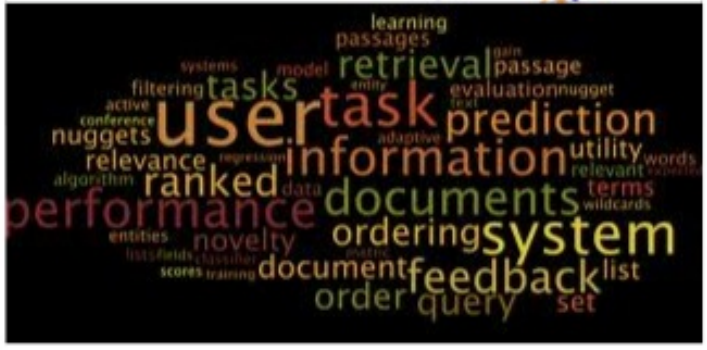
Florence d'Alché,
florence.dalche@telecom-paristech.fr

CES Data Scientist, TPT, Paris, France

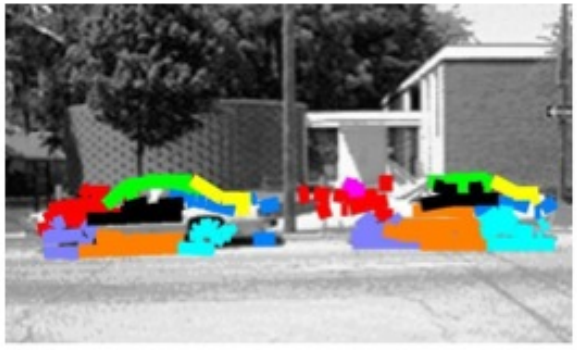
Outline

Motivation

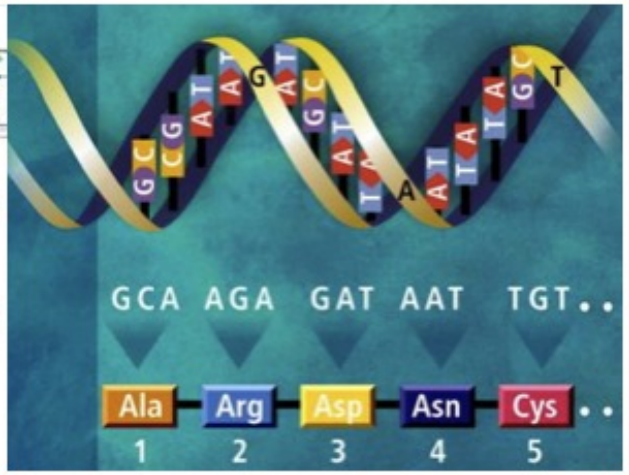


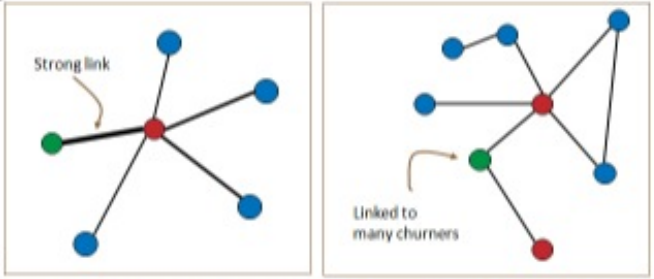

learning
passages
retrieval
passage
tasks
evaluation
nugget
user
task
prediction
nuggets
information
utility
relevance
ranked
terms
performance
documents
ordering
system
novelty
document
feedback
list
order
query
set


Correlations coefficient
1.00
0.75
0.50
0.25
0.00
1996
1998
2000
Stocks, USA and Germany
Stocks, USA and Denmark
10-year government bond, USA and Germany
10-year government bond, Germany and Denmark



Data everywhere !

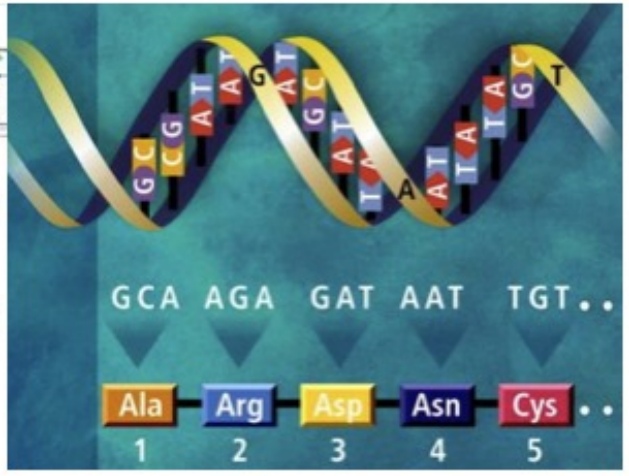


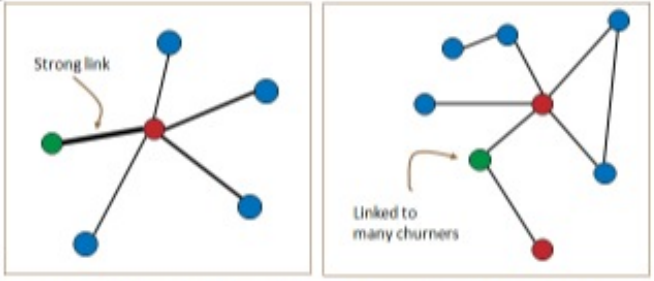


Two features of the call network + seed churners predict future churners

Data

- Vectorial data (ex: explanatory variables to describe a client)
- Unstructured data (ex: texts, images)
- Structured data such XML pages, molecules (graphs), sequences
- Heteregoneous data
- Incomplete data
- Noisy data





Two features of the call network + seed churners predict future churners

Machine Learning everywhere !

Main applications

- Search engine, text-mining
- Social networks analytics
- Recommendation systems
- Diagnosis, Fault detection
- Predictive Maintenance, monitoring
- Business analytics
- Pattern recognition
- Robotics

A definition of Machine Learning

A definition by Tom Mitchell (http://www.cs.cmu.edu/~tom/

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T , as measured by P, improves with experience E.

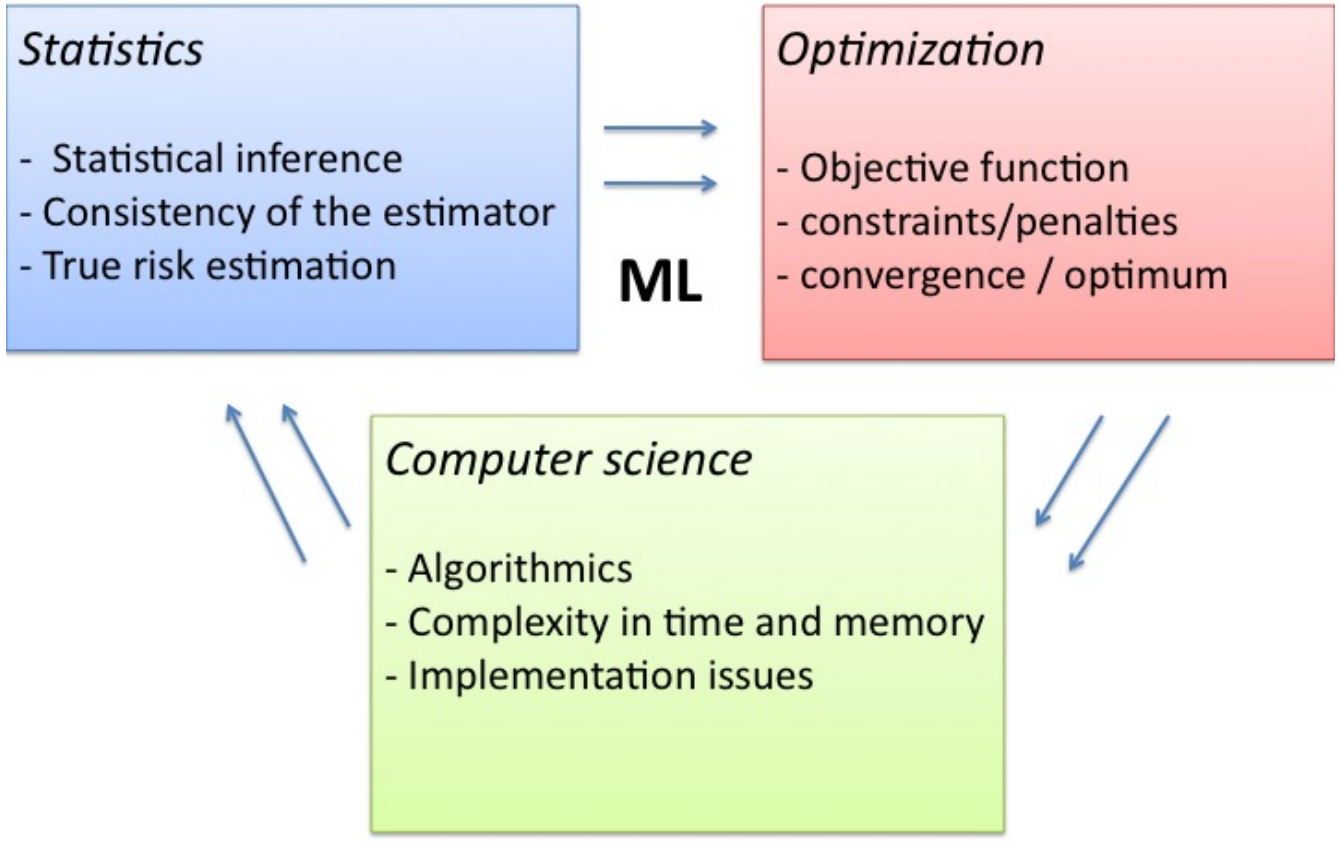
Experience, tasks and performance measure

- **Experience** : data provided off-line or on-line
- **Tasks** : pattern recognition, diagnostic, complex system modelling, game player, robot learning,...
- **Performance measure** : accuracy on new data, ability to generalize

Influences of various domains

- Statistical Inference
- Computer Science
- Artificial intelligence
- Mathematical programming
- Neurophysiology (neural nets)
- Theory of approximation
- Statistical physics
- Even Philosophy of inference and modelling

Machine Learning

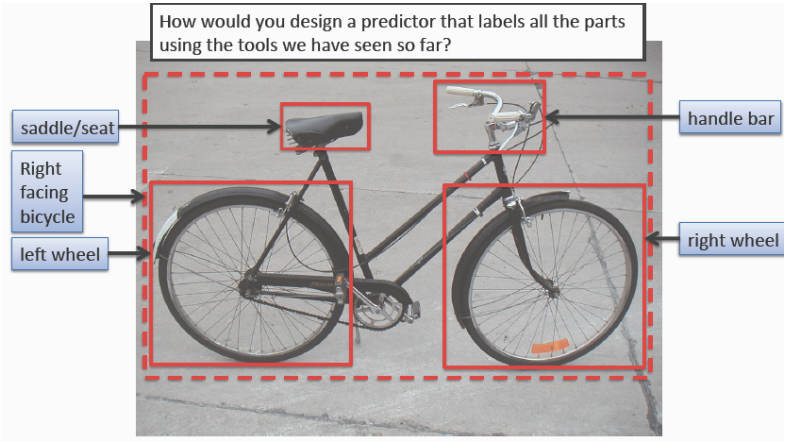


Example 1: a robot that learns

Robot endowed with a set of sensors and a online learning algorithm:

- Sense the environment, act and measure the effect of action
- Goal: play football

Example 2 : object recognition in an image



- Read a data file
- Recognize if parts of the target object are present
- **Goal**: say if an object is present or not in the image.

Two kinds of learning 1/2

Online learning: *the learning algorithm keeps on interacting with the environment*

- robotics
- predictive maintenance
- security in cloud servers
- personalized advertising
- autonomous cars
- personalized healthcare
- security systems

Two kinds of learning 2/2

Offline or batch learning: *the learning algorithm gets a datafile and outputs some function that can be used in turn to new data*

- pattern recognition (a wide panel of applications)
- diagnosis (health, plants)
- link prediction in networks
- data-mining
- social networks analytics

Supervised versus unsupervised learning

- **Supervised Learning (classification, regression)**:
  - Goal: Learn a function f to predict a variable y from an individual x.
  - Data: Learning set (x_i, y_i)
- **Unsupervised Learning (clustering, graphical model)**:
  - Goal: Discover a structure within a set of of individuals $\{x_i\}$.
  - Data: Training set $\{x_i\}$
- First case is better posed.
- Supervised Learning : modules 1, 4 and 5 lecture)
- Note: most of these algorithms can be implemented offline or online.

Outline

Programme CES Data Science

- **Algorithmes "classiques" pour la classification**
  Analyse discriminante linéaire et régression logistique
  Les "plus proches voisins" et variantes
  Méthodes de partitionnement - l'algorithme CART
  Le perceptron mono-couche
- **Sélection de Modèles**
  Planification expérimentale - Validation Croisée - Minimisation Structurelle du Risque - Bootstrap
- **Algorithmes "avancés" pour la classification**
  - Boosting
  - Random Forest
  - SVM - Noyaux
  - Réseaux de neurones et deep learning

Programme CES Data Science

- **D'autres problèmes supervisés**
  Convexification
  Classification multi-label
  Régression ordinale et Régression (Lasso)
  Ranking
- **Programme CES Data Science**
  Clustering
  Analyse en Variables Latentes (kernel PCA, ICA, NMF)
  Minimum Volume Set
- **HMM - Modèles graphiques**

Programme CES Data Science

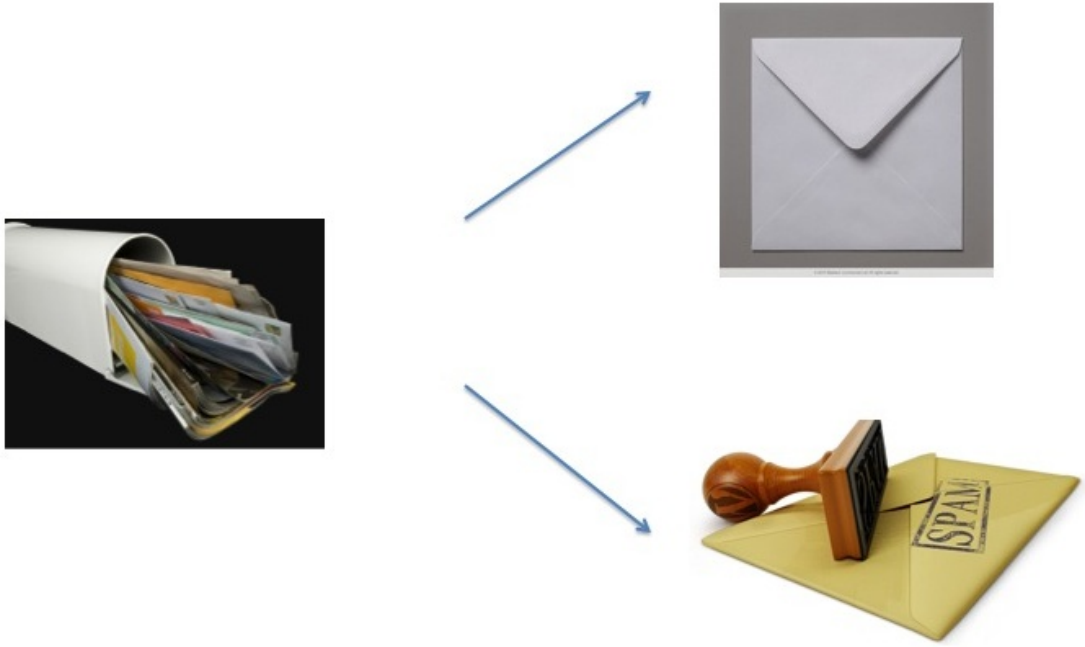
- **Apprentissage et optimisation distribués**
  Programmation MapReduce
- **Graph-Mining**
  Clustering spectral
  Détection de communautés
  Visualisation
- **Apprentissage en ligne**
- **Moteurs de recommandation - Filtrage collaboratif**

Outline

Outline

Objectif: détecteur de spams

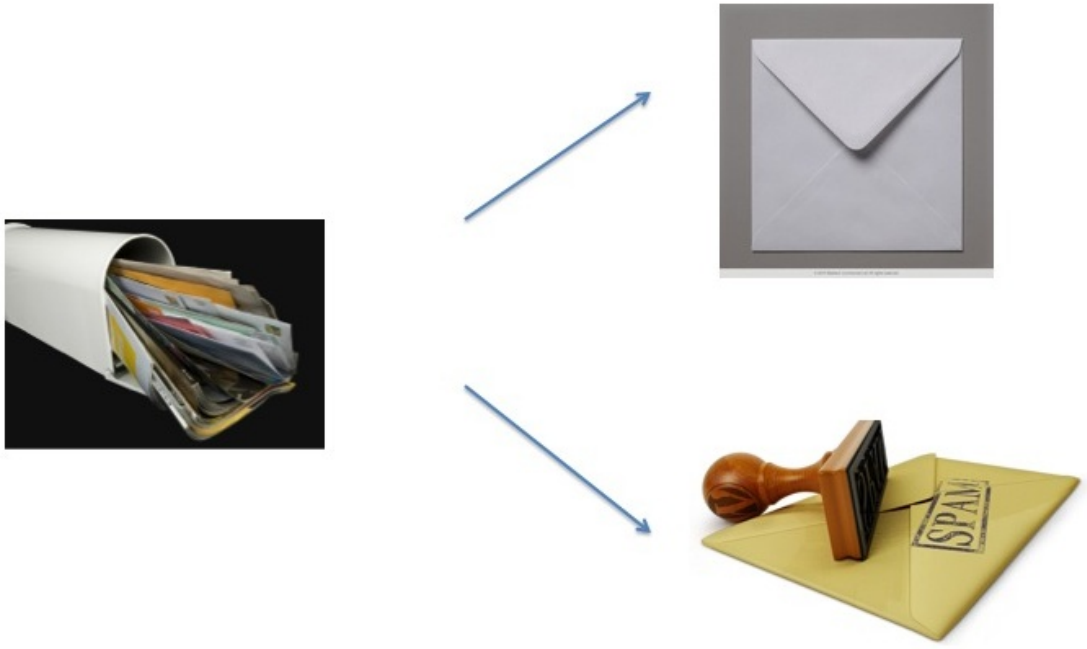
Construire un ensemble d'apprentissage



Par exemple, pendant une semaine je trie mon courrier et je stocke les fichiers des emails et je leur associe une étiquette de classe $+1$ s'ils me paraissent pertinents, -1 s'ils ne le sont pas.

Pre-processing et codage des données

Une étape essentielle ...



Quelles sont les variables explicatives qui vont coder les messages ? Présence de certains mots....Quelles parties du message garde-t-on? Comment faire pour coder un message avec des fautes d'orthographe ?

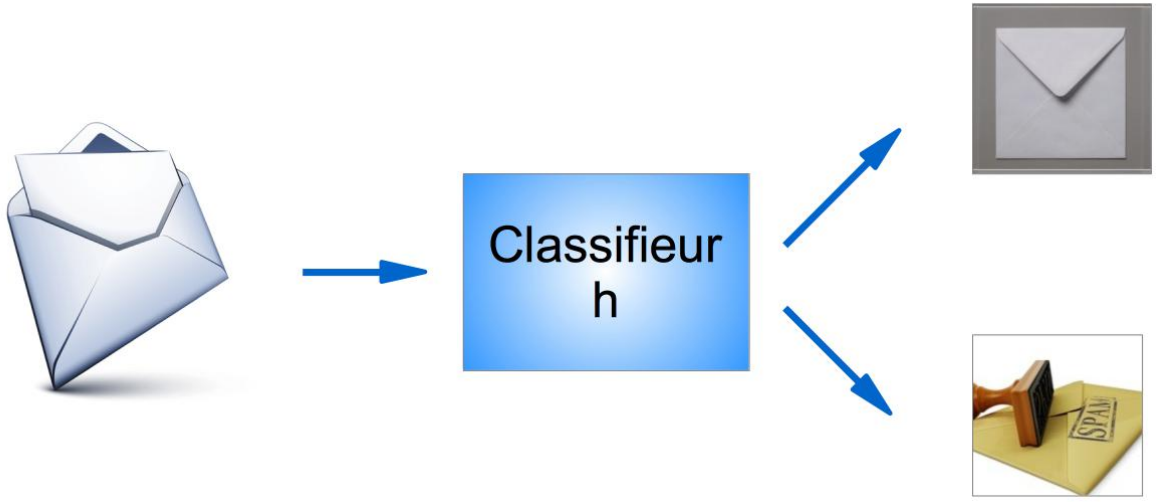
Apprendre à classer des messages

Ensemble d'apprentissage

Algorithme d'apprentissage

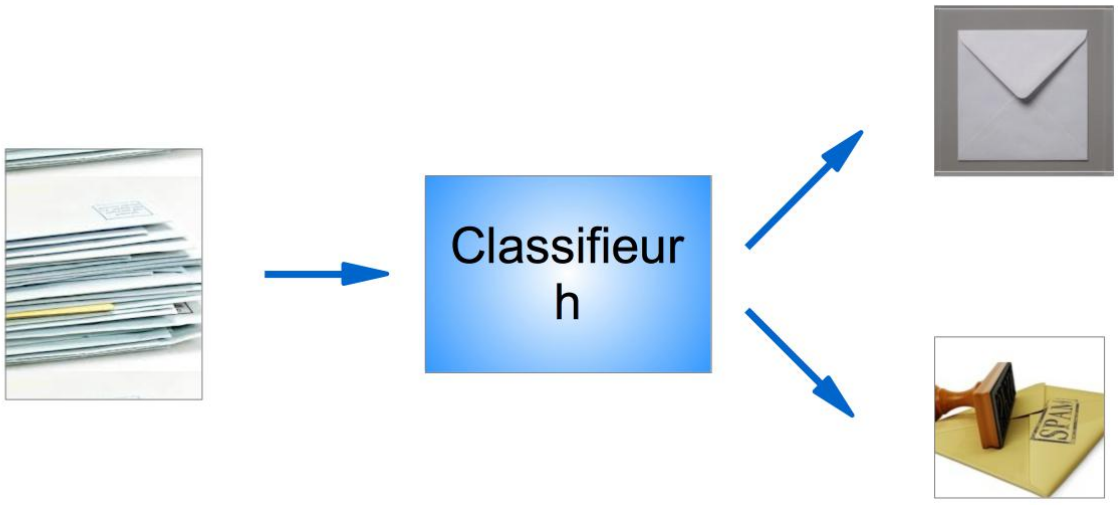
Classifieur h

Classer un nouveau message



Evaluer le détecteur de SPAM

- Mesurer le nombre d'erreurs commises par h sur un ensemble de messages jamais vus par l'algorithme d'apprentissage



Des données au classifieur

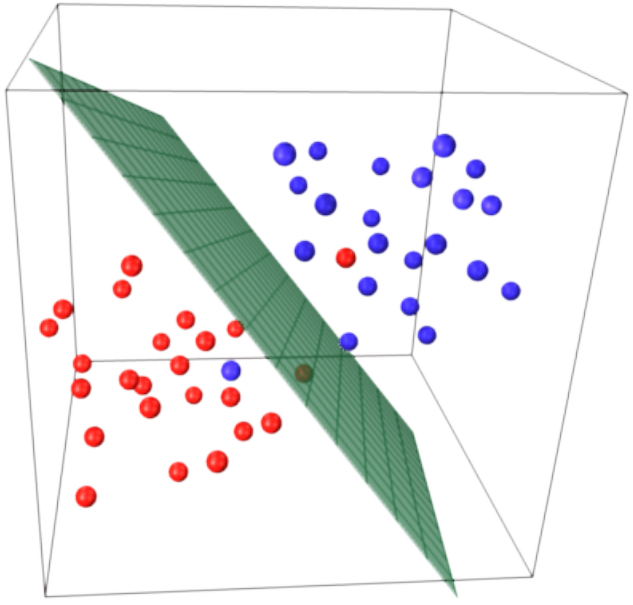
1. Etiquetage des documents (supervision), codage des documents, stockage des documents étiquetés
2. Application d'un algorithme d'apprentissage aux données d'apprentissage: fournit un classifieur
3. Application du résultat de l'apprentissage, c'est-à dire du classifieur à des nouvelles données
4. Evaluation : calcul du nombre d'erreurs commises par le classifieur

Des données au classifieur

1. Etiquetage des documents (supervision), codage des documents, stockage des documents étiquetés
2. Application d'un algorithme d'apprentissage aux données d'apprentissage: fournit un classifieur
3. Application du classifieur à des nouvelles données : fournit des prédictions de classe
4. Evaluation : fournit une mesure d'erreur

Données et classifieur : visualisation

- Document : un vecteur x dans $\mathbb{R}^p$
- Classifieur: une fonction à valeurs discrètes de $\mathbb{R}^p$ dans $\{-1, +1\}$
- Interpétation géométrique: un classifeur définit une frontière de séparation entre les données
- Exemple : classifieur linéaire



Coder les documents

Codage Term-Frequency-Inverse Document Frequency (TF-IDF)

- une collection C de messages (documents)
- un mot = un terme
- à définir : un dictionnaire D de p termes apparaissant dans C
- un message (document) $x \rightarrow$ un ensemble de termes avec leur occurrence (bag of words)
- C: a collection of N documents
- $TF(t,x) = \frac{\text{nb d'occurrence de t dans x}}{\text{nb de termes dans x}}$
- $IDF(t,C) = \log \frac{\text{N}}{\text{nb de documents de } C \text{ où t apparaît}}$

Espace de représentation des messages

Codage TF-IDF d'un message x

- un vecteur x de dimension p
- $x_i = TF(t_i, x).IDF(t_i, C), i = 1, \ldots, p$
- à chaque donnée d'entrée x_i est associée une étiquette de classe $y_i \in \{-1, +1\}$
- On prend : C $= S_{app}$, documents de l'ensemble d'apprentissage

N.B: la dimension p peut être trs grande (jusqu'à 30 000 mots), même en réduisant, on obtiendra des vecteurs trs creux, c'est-à dire avec beaucoup de zeros.

Classe des fonctions de classification

Au programme pendant ces deux jours

1. Classifieur linéaire : analyse discriminante linéaire, régression logistique linéaire, perceptron
2. Classification non linéaire : k- plus-proches voisins

Algorithme d'apprentissage

- $\mathcal{A} : (\{(x_i, y_i), i = 1, \ldots, n\}) \rightarrow h \in \mathcal{H}$

N.B:

1. Il s'agit bien d'un algorithme et il doit être implémenté dans un langage informatique (en scikitlearn, classifier.fit(...))
2. En apprentissage statistique, l'algorithme réalise une estimation à partir des données

Classifieur

Une fois ses paramètres définis, et pour une nouvelle donnée d'entrée, le classifieur fournit une prédiction (en scikitlearn classifier.predict(...))

Approche statistique de l'apprentissage

- Dans la suite de ce cours, nous introduisons l'approche statistique de l'apprentissage automatique qui est la plus performante et la plus utilisée
- Il est bon de savoir qu'il existe une autre approche symbolique et logique qui s'intéresse à construire des formules logiques qui décrivent les classes. Ces approches bien que trs interprétables présentent deux inconvénients majeurs: elles sont peu robustes au bruit et elles souffrent d'un problème de transition de phase lors de l'apprentissage

Outline

Classification binaire supervisée

Cadre probabiliste : pas encore de données !

- Soit X un vecteur aléatoire de $\mathcal{X} = \mathbb{R}^p$
- X décrit ici les caractéristiques ("features") d'un message ou document
- Y une variable aléatoire discrète $\mathcal{Y} = \{-1, 1\}$
- Soit P la loi de probabilitéé jointe de (X,Y)

Classifieur, perte et risque

Cadre probabiliste : pas encore de données !

- Soit $\ell : \{-1, +1\} \times \{-1, +1\} \to \mathbb{R}$ une fonction de perte ou coût
- Par exemple, la fonction de perte $0-1$ ou coût de prédiction est définie par $\ell_{0-1}(y, y') = 1$ si $y \neq y'$, 0 sinon.
- Soit $h : \mathbb{R}^p \to \{-1, +1\}$ une fonction de classification binaire ou classifieur binaire
- on définit le risque de h par:
  - $R(h) = \mathbb{E}_P[\ell(Y, h(X))]$
- Dans le cas de la perte $0-1$, $R(h) = \mathbb{P}(h(X) \neq Y)$ est la probabilité que h se trompe - sous-entendu sur des (x,y) distribués selon P.

Risque $R(h)$

Espérance du loi jointe mixte:

$$R(h) = \sum_{y=-1,1} P(Y = y) \int_{\mathbb{R}^p} \ell(y, h(x)) p(x|Y = y) dx$$

Meilleur classifieur

Existe-t-il un classifieur h^* qui minimise R

- Etant donnée $P(X, Y)$ la loi de probabilité jointe, existe-t-il un classifieur h^* tel que:

$$h^* = \arg\min_h R(h)? \tag{1}$$

Réponse: oui, le classifieur de Bayes

Classifieur de Bayes

$$h_{Bayes}(x) = \arg \max_{y \in \{-1,+1\}} P(Y = y|x)$$

On utilise la formule de Bayes pour le définir:

- $P(Y = k|x) = \frac{p(x|Y=k)P(Y=k)}{p(x|Y=-1).P(Y=-1)+p(x|Y=1).P(Y=1)}$

Formule de Bayes

- $P(Y = k|x) = \frac{p(x|Y=k)P(Y=k)}{p(x|Y=-1).P(Y=-1)+p(x|Y=1).P(Y=1)}$

Classifieur bayésien

Definition

$$h_{Bayes}(x) = argmax_{k=1,-1} P(Y = k|x)$$

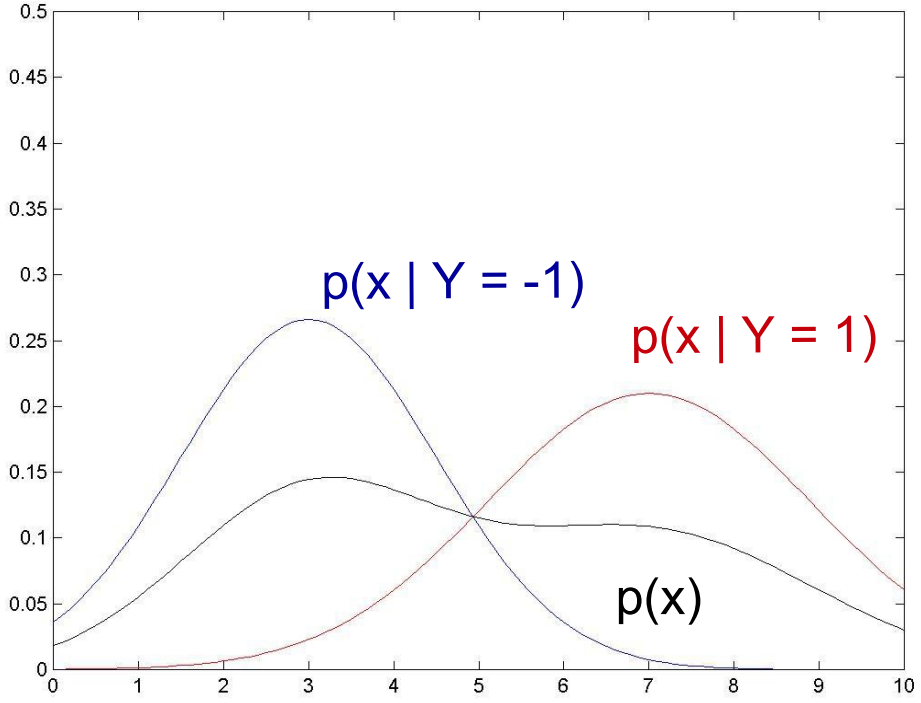
.

Risque bayesien

$$\begin{aligned} R(h_{bay}) &= \int_{R_1} P(h_{bay}(x) \neq 1)p(x)dx + \int_{R_{-1}} P(h_{bay}(x) \neq -1)p(x)dx \quad (2) \\ &= \int_{R_1} P(y = -1|x)p(x)dx + \int_{R_{-1}} P(y = 1|x)p(x)dx \quad (3) \end{aligned}$$

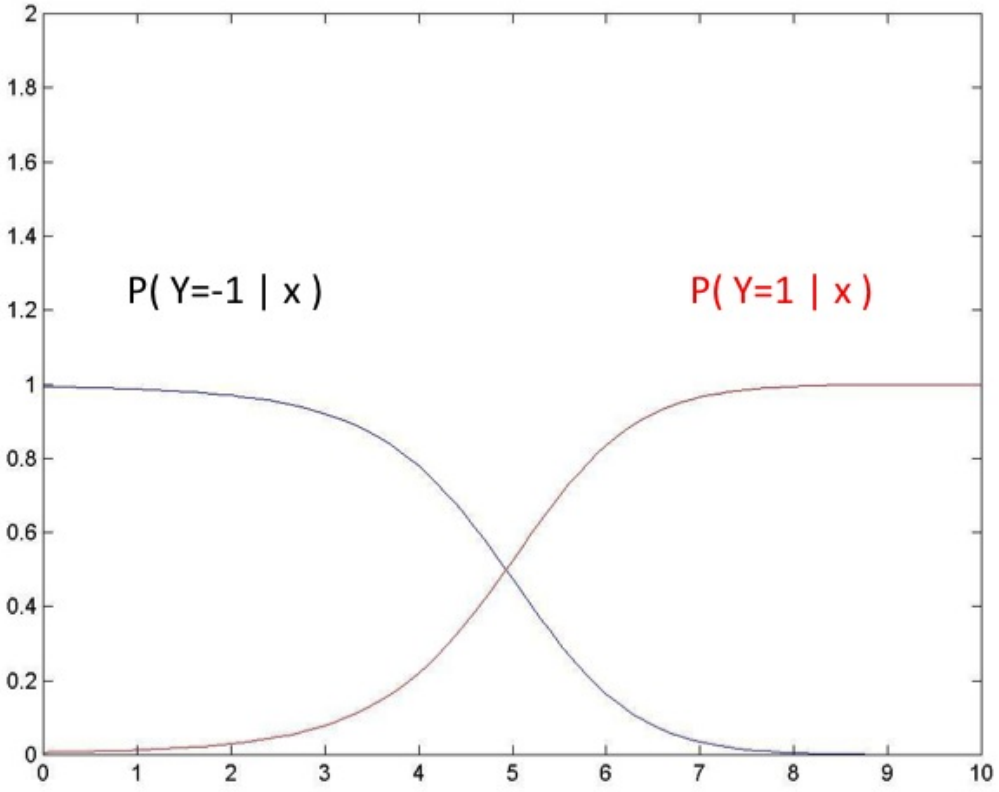
On démontre qu'il s'agit du meilleur classifieur .

Exemple en 1D avec des lois conditionnelles gaussiennes et $P(Y = +1) = P(Y = -1)$



Classifieur bayesien

Classes gaussiennes et $P(Y = +1) = P(Y = -1)$



Take-home message

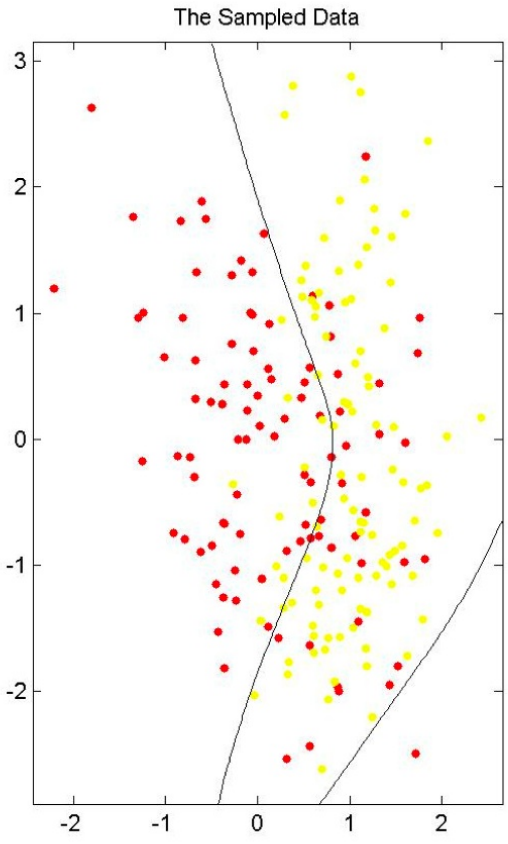
- La fonction cible pour la perte $0-1$ en classification supervisée est le classifieur de Bayes
- On ne peut pas obtenir un risque plus petit que le risque bayesien: $R(h_{Bayes})$ qui est une caractéristique du problème
- NB : nous verrons plus tard qu'en régression, la fonction cible pour la perte quadratique est l'espérance conditionnelle $h(x) = \mathbb{E}[Y|x]$

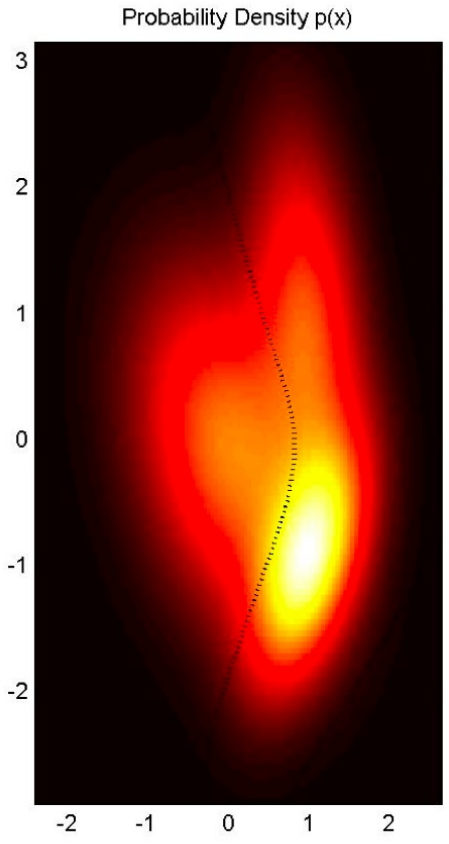
Classification binaire supervisée

Cadre probabiliste et statistique: voici les données !

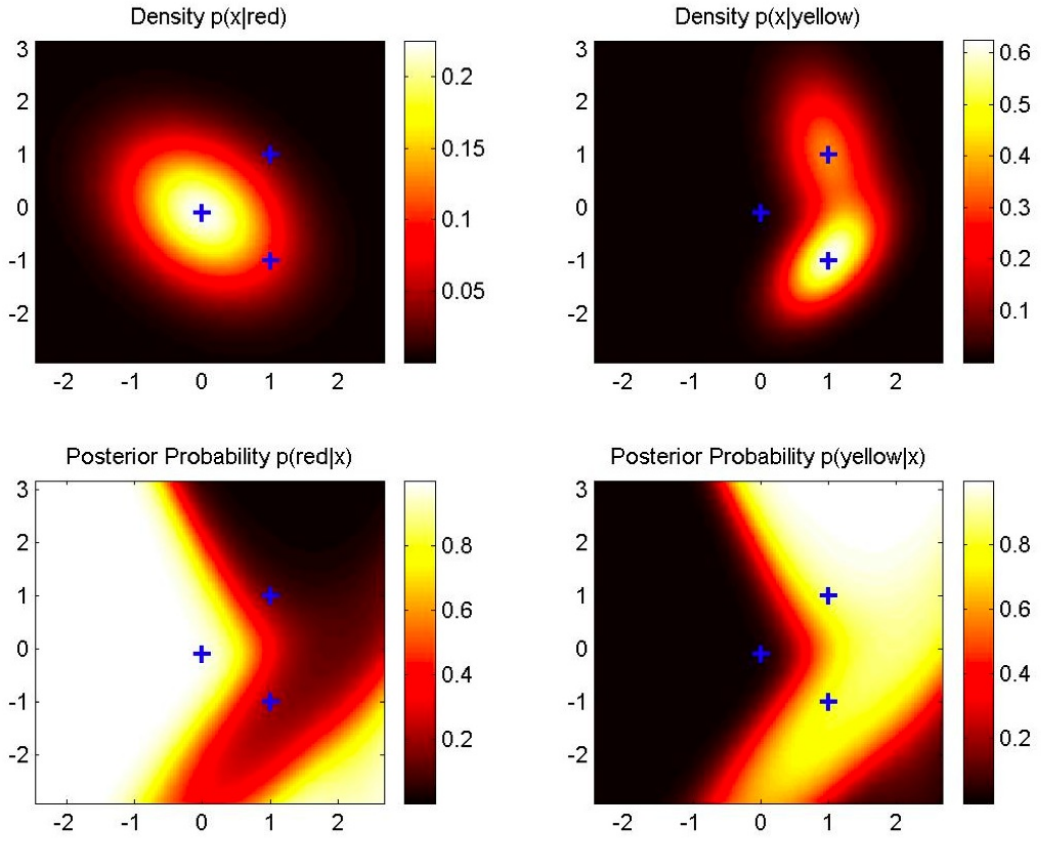
- Nous supposons que $S_n = \{(x_i, y_i), i = 1, \ldots, n\}$ est un échantillon i.i.d. tiré de la loi de probabilité jointe $P(X, Y)$
- P est fixée mais inconnue
- A partir de S_n, déterminer la fonction $h_n \in \mathcal{H}$ qui minimise le risque $R(h)$ pour $h \in \mathcal{H}$, une classe de fonctions.

Exemple en 2D

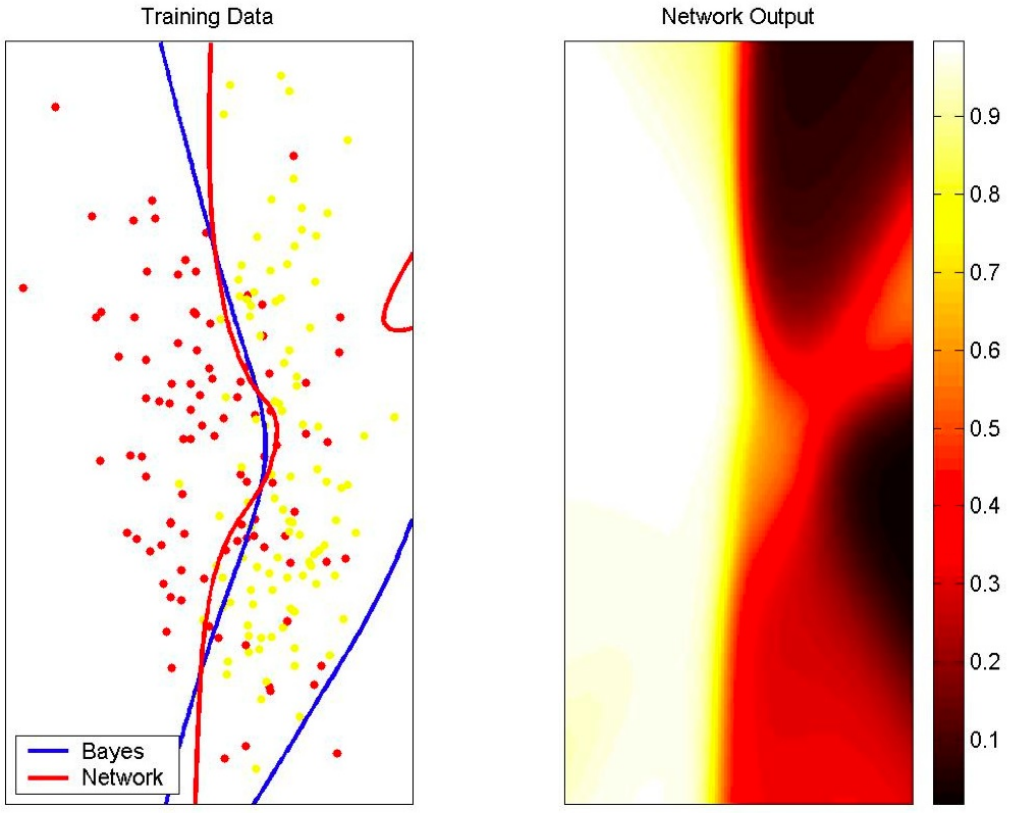




Exemple en 2D



En utilisant un ensemble d'apprentissage



Apprentissage statistique - approches discriminantes

Pb: la loi jointe n'est pas connue : on ne peut pas calculer $R(h)$

Minimisation du risque empirique

A la place de l'espérance, on minimise la moyenne empirique , appelée *risque empirique*:

$$\mathrm{R}_n(h) = \frac{1}{n} \sum_i \ell(x_i, y_i, h(x_i))$$

Erreur d'excès, erreur d'approximation et erreur d'estimation

Considérons ici la perte $0-1$: Soit $R^* = \inf_h R(h)$, le risque de Bayes. Soit $R_{\mathcal{H}} = \inf_{h \in \mathcal{H}} R(h)$.
Supposons $h_n \in \mathcal{H}$ est le classifieur estimé à partir des données S_n par minimisation du risque empirique ou par tout autre principe employant les données.

Erreur d'excès, erreur d'approximation et erreur d'estimation

$$R(h_n) - R^* = R(h_n) - R_{\mathcal{H}} + R_{\mathcal{H}} - R^*$$

L'excès d'erreur que fait h_n pa rapport au risque de Bayes est égal à la somme de deux termes:

- $R(h_n) - R_{\mathcal{H}}$: l'erreur d'estimation, mesurant à quel point on s'approche de l'optimum dans $\mathcal{H}$
- $R_{\mathcal{H}} - R^*$: l'erreur d'approximation, inhérente à la classe de fonctions choisie. par exemple, si la frontiere de séparation est non linéaire et que je me restreins à un classifieur linéaire.

Consistance statistique

En statistique, on s'intéresse au comportement de l'algorithme d'apprentissage en tant que procédure d'estimation. $h_n = \mathcal{A}(S_n)$

Consistance statistique

Consistance en $\mathcal{H}$ par rapport à une loi P et une perte ℓ

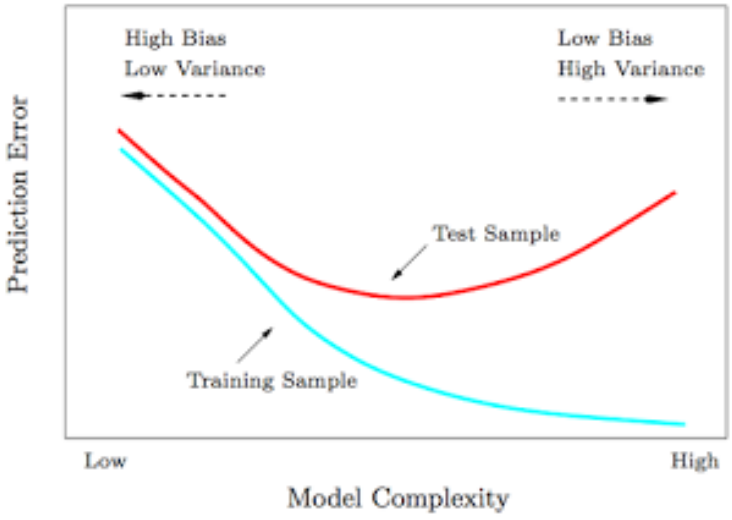
$\mathcal{A}$ est consistant en $\mathcal{H}$ par rapport à une loi P et une perte ℓ si:
pour tout $\epsilon > 0$,
$\mathbb{P}_{S_n}(\mathbb{E}_P[\ell(h_n(X), Y)] - R^\ell_{\mathcal{H}} \geq \epsilon) \to 0$ quand n tend vers l'infini.

Lorsque $\mathcal{A}$ est consistant pour toutes les distributions de probabilités P, on dit que $\mathcal{A}$ est universellement consistant en $\mathcal{H}$ par rapport à ℓ. L'algorithme de minimisation du risque empirique est universellement consistant

Néanmoins, attention au surapprentissage

A nombre n de donnés fixé:

Compromis biais /variance : comment choisir $\mathcal{H}$?



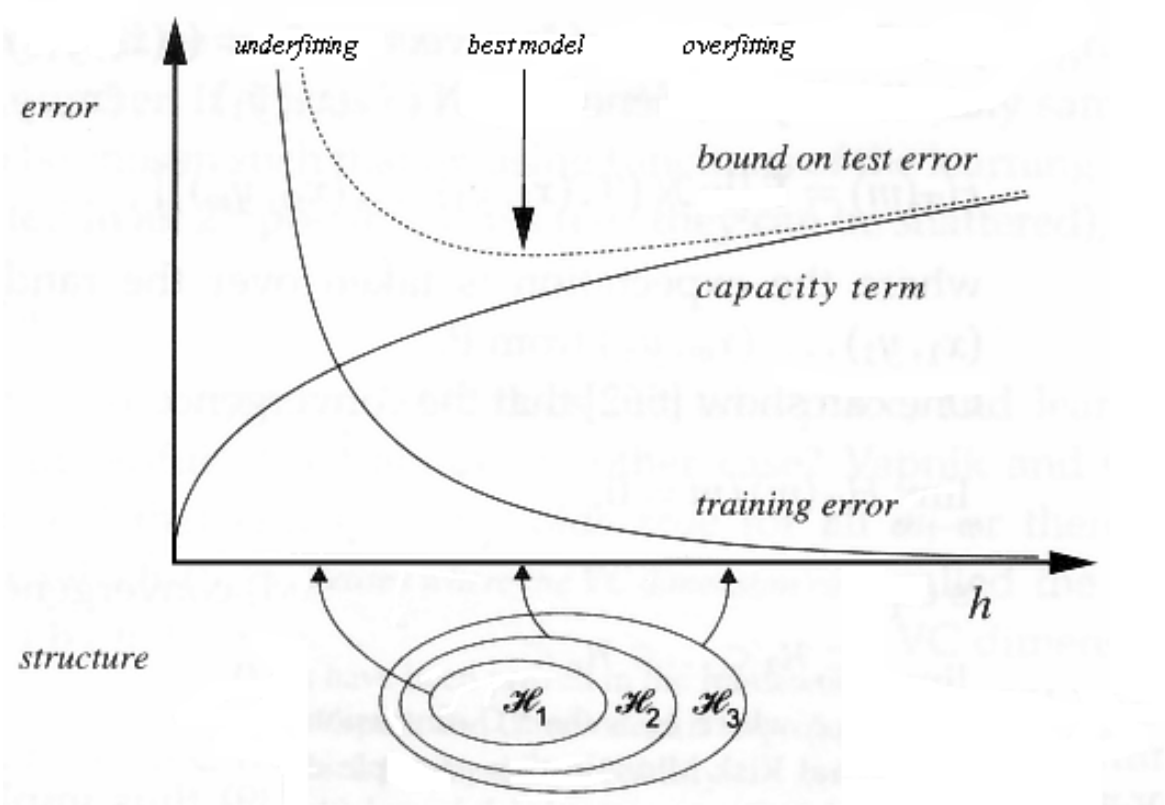
- Si la classe $\mathcal{H}$ est trop petite, on ne peut pas atteindre la cible (biais large)
- Si la classe $\mathcal{H}$ est trop grande, on ne peut pas réduire la variance de l'estimateur (variance petite)

Comportement du risque empirique

Résultats de vapnik et Chervonenkis

- $\forall \mathbb{P}, \mathcal{S}_n$ i.i.d from $\mathbb{P}, \forall h \in \mathcal{H}, R(h) \leq R_n(h) + \mathcal{B}(d, n)$
- où d est une mesure de complexité de $\mathcal{H}$
- si n augmente, $\mathcal{B}(d, n)$ diminue
- si d augmente, $\mathcal{B}(d, n)$ augmente

Surapprentissage : minimisation du risque structurel pour l'éviter



Surapprentissage : approcher par régularisation

Exemple de l'approche par régularisation

- A la place de $R(h)$, on minimise la somme de deux termes:
- le risque empirique $R_n(h) = \frac{1}{n}\sum_i \ell(y_i, h(x_i))$ et un terme régularisateur $\Omega(h)$ qui mesure la *complexité* de h.
- On cherche : $\hat{h} = \arg\min_{h\in\mathcal{H}} R_n(h) + \lambda\Omega(h)$

Surapprentissage : approcher par régularisation

Exemple de l'approche par régularisation

- A la place de $R(h)$, on minimise la somme de deux termes:
- le risque empirique $R_n(h) = \frac{1}{n}\sum_i \ell(y_i, h(x_i))$ et un terme régularisateur $\Omega(h)$ qui mesure la *complexité* de h.
- On cherche : $\hat{h} = \arg\min_{h\in\mathcal{H}} R_n(h) + \lambda\Omega(h)$

NB : on cherche à obtenir un compromis entre une bonne adéquation aux données et une complexité limitée : $\Omega(h)$ est en général choisi pour renforcer la régularité de la fonction

Outline

Refs

- The elements of statistical learning, Hastie, Tibshirani, Friedman, Springer.
- Foundations of machine learning, Mohri, MIT press.