Cadre de l'apprentissage statistique, premiers classifieurs

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

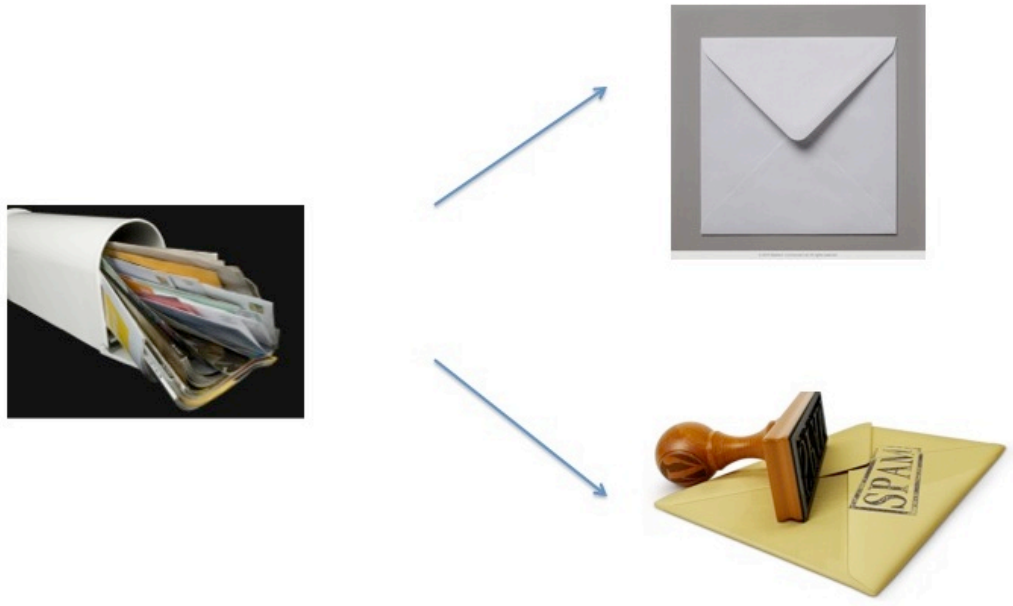
MDI343 - MS BIG DATA, TPT

Outline

Outline

Objectif: détecteur de spam

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.

Apprendre à classer des messages

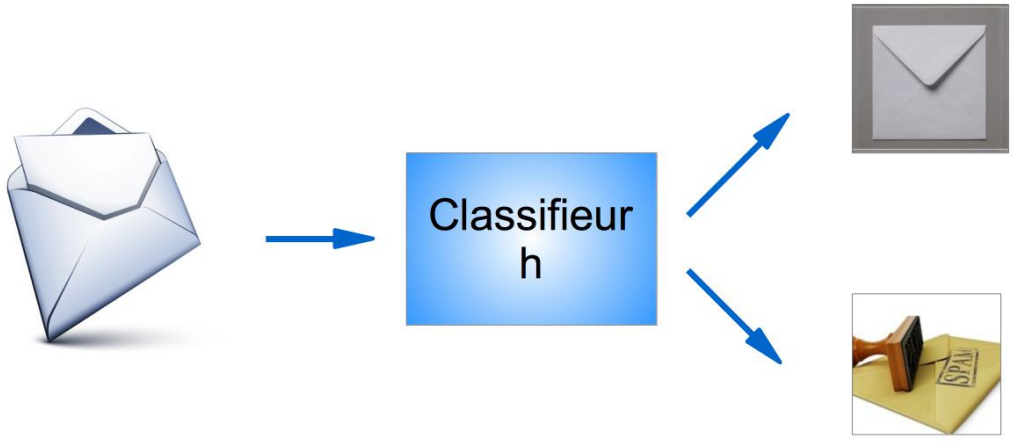
Ensemble d'apprentissage

Algorithme d'apprentissage

→

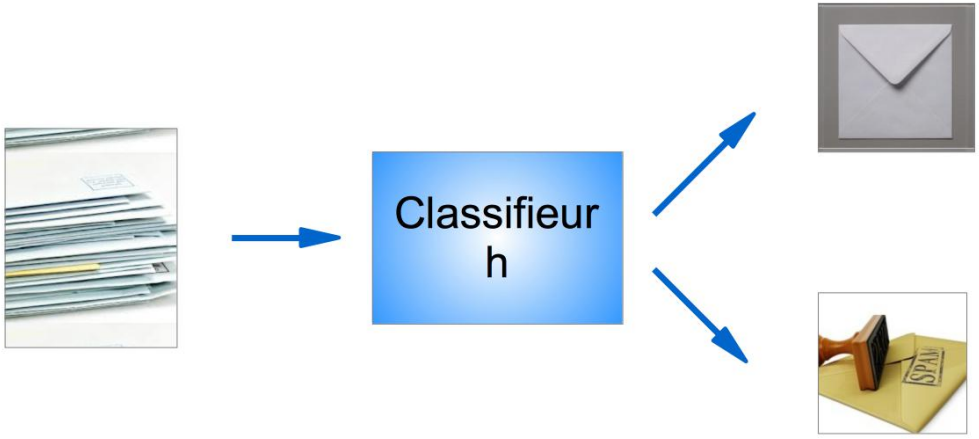
Classifieur h

Classer un nouveau message avec le détecteur de spam



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

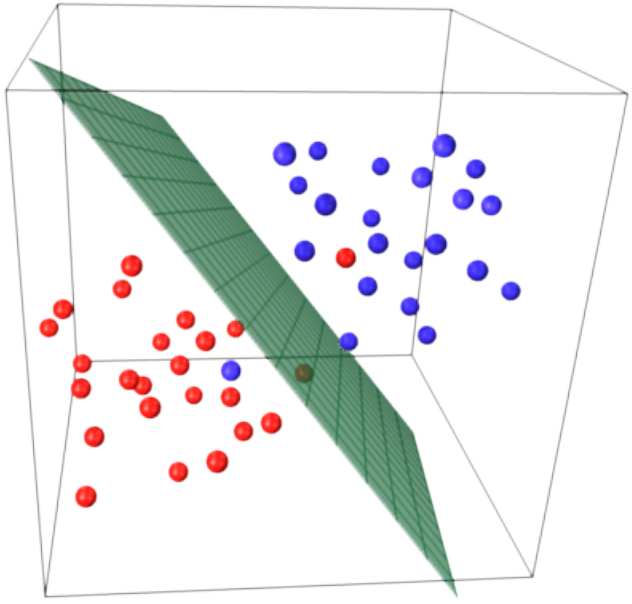
1. Etiquetage des documents (supervision), codage des documents, stockage des documents étiquetés
2. Application d'un algorithme d'apprentissage aux données d'apprentissage: fournit un classifieur
3. Application du résultat de l'apprentissage, c'est-à dire du classifieur à des nouvelles données
4. Evaluation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

Des données au classifieur

- Document : un vecteur x dans $\mathbb{R}^p$
- Classifieur: une fonction à valeurs discrètes de $\mathbb{R}^p$ dans $\{-1, +1\}$
-
- Il existe des classifieurs linéaires (frontière de séparation = hyperplan, ex: perceptron) et des classifieurs non linéaires (par ex: k-plus-proches voisins)



Coder les documents

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

- une collection C de messages (documents)
- un mot $\rightarrow$ 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
- $TF(t,x) = \frac{\text{nb d'occurrence de t dans d}}{\text{nb de termes dans x}}$
- $IDF(t,C) = \log \frac{\text{N}}{\text{nb de documents de } C \text{ où t apparaît}}$

Espace de représentation des messages

Codage TF-IDF d'un message x

- un vecteur x de dimension p
- $x_i = TF(t_i, x).IDF(t_i, C), i = 1, \ldots, p$
- On prend : C $= S_{app}$, documents de l'échantillon d'apprentissage

Espace de représentation des données

- $\mathcal{X} = \mathbb{R}^p$

Classe des fonctions de classification

Au programme

1. Classifieur linéaire : analyse discriminante linéaire, régression logistique linéaire, (perceptron : vu en TP)
2. Classification non linéaire : k- plus-proches voisins

Outline

Classification binaire supervisée

Cadre probabiliste : pas encore de données !

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

Classifieur, perte et risque

Cadre probabiliste : pas encore de données !

- Soit $h : \mathbb{R}^p \to \{-1, +1\}$ une fonction de classification binaire
- Soit $\ell : \{\mathbb{R}^p, -1, +1\} \times \{-1, +1\} \to \mathbb{R}$ une fonction de perte ou coût
- Par exemple, la fonction de perte 0/1 ou coût de prédiction est définie par $\ell_{0/1}(x, y, h(x)) = 1$ si $y \neq h(x)$, 0 sinon.
- 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:

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

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

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

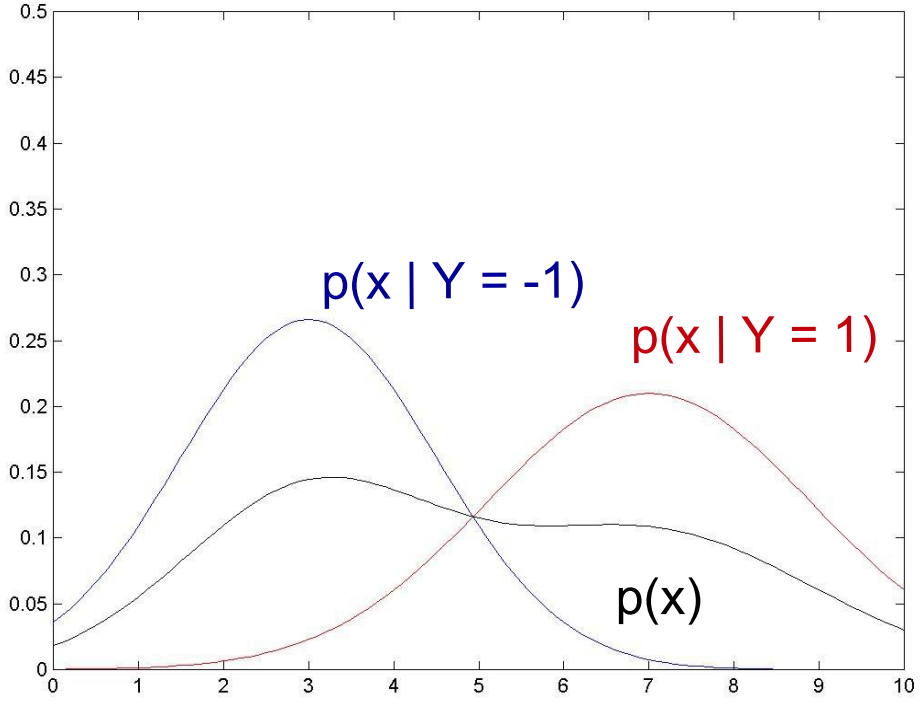
.

Risque bayesien

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

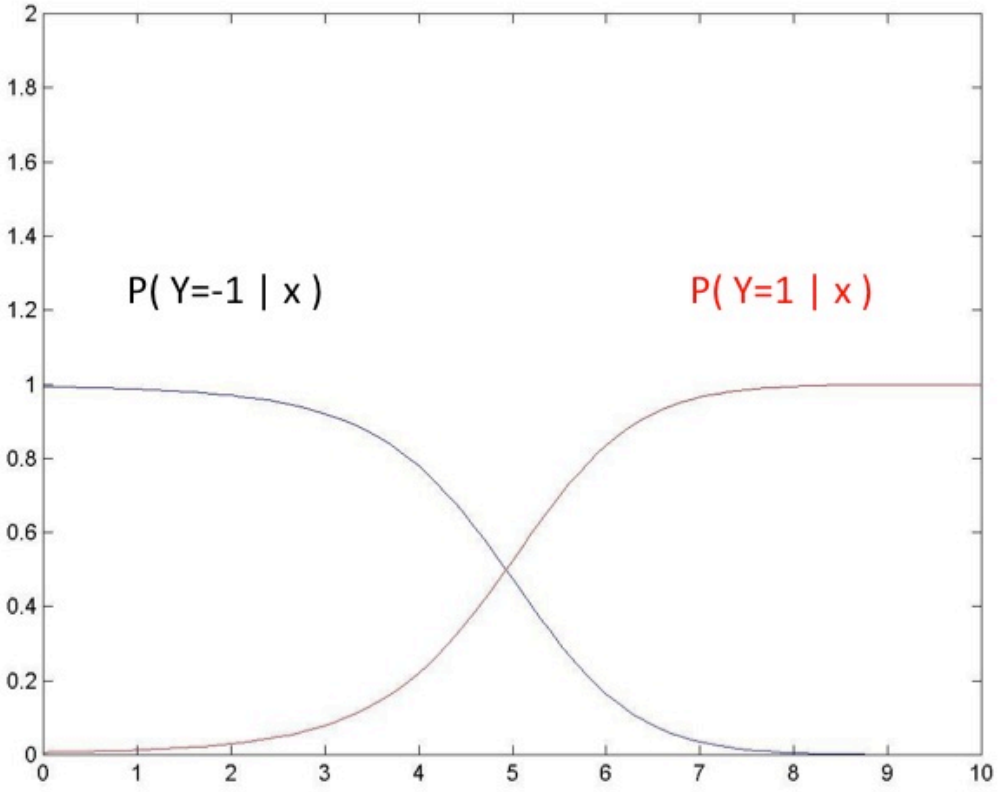
On démontre qu'il s'agit du meilleur classifieur .

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 (la solution !) 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 $S_{app} = S_n = \{(x_i, y_i), i = 1, \ldots, n\}$ 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.

Apprentissage supervisé

On distingue deux types d'approches:

1. les approches dites "génératives": $h(x) = seuil(\hat{P}(Y = 1|x))$ et h est fondée sur la modélisation des probabilités conditionnelles de chaque classe: $p(x|Y = 1)$ et $p(x|Y = -1)$
2. les approches dites "discriminantes": avec $h(x)$ on essaie de discriminer entre les classes sans modélisation des probabilités conditionnelles

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*:

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

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

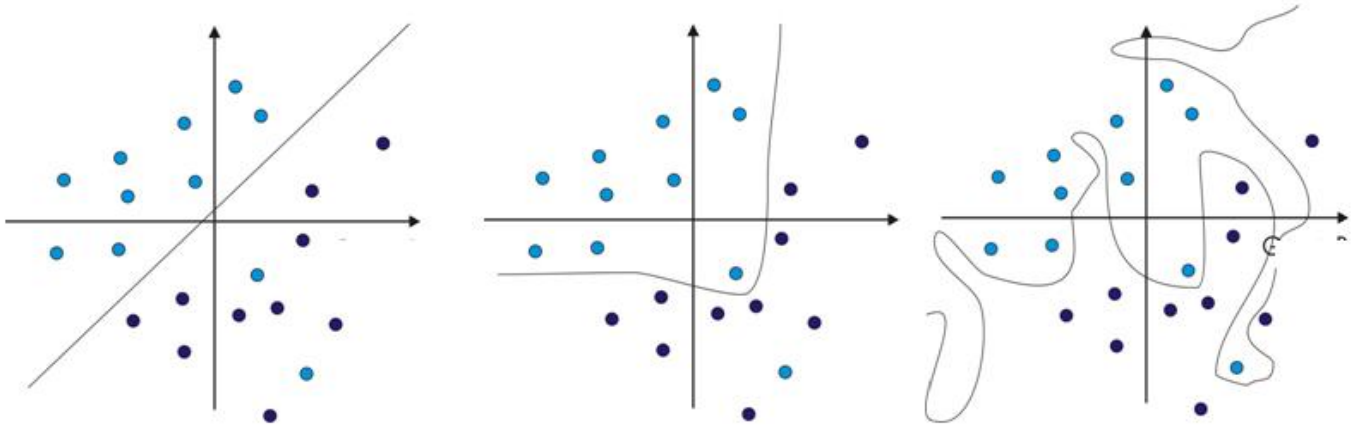
$\mathcal{A}$ est consistant en $\mathcal{H}$ par rapport à une loi P et une perte ℓ si:
pour tout $\epsilon > 0$,
$\mathbb{P}(|\mathbb{E}_P[\ell(X, h_n(X), Y)] - R^\ell_{\mathcal{H}}| \geq \epsilon) \to 0$ quand n tend vers l'infini.

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

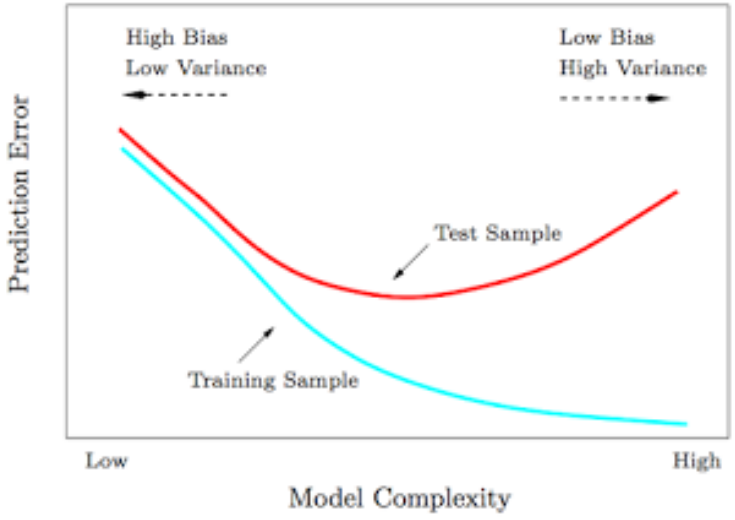
Néanmoins, attention au surapprentissage

A nombre fixé n de données :



le modèle qui ne fait aucune erreur sur les données d'apprentissage n'est pas nécessairement le meilleur !

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



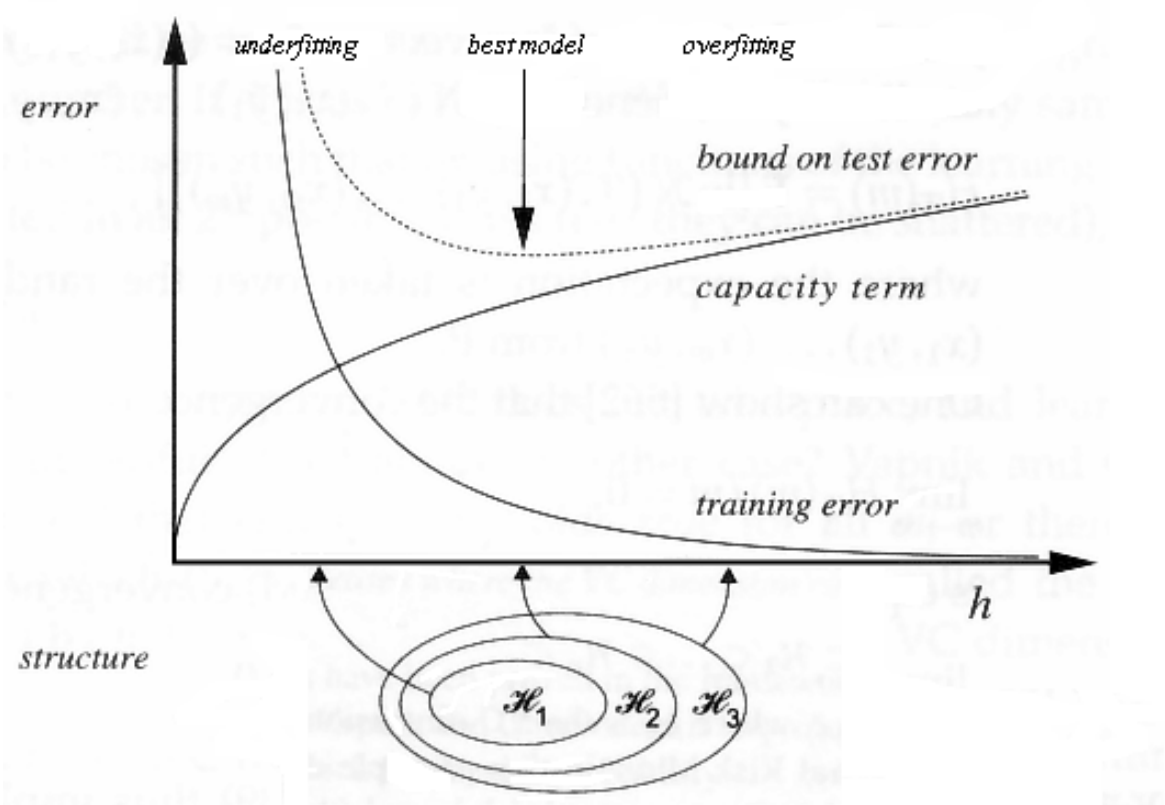
- Si la classe $\mathcal{H}$ est trop petite, on ne peut pas atteindre la cible (biais large)
- Si la classe $\mathcal{H}$ est trop grande, on ne peut pas réduire la variance de l'estimateur (variance petite)

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 $\mathcal{H}$ (par exemple la dimension de Vapnik-Chrevonenkis, voir annexe)
- 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 : approche 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 : $\hat{h} = \arg\min_{h\in\mathcal{H}} R_n(h) + \lambda\Omega(h)$

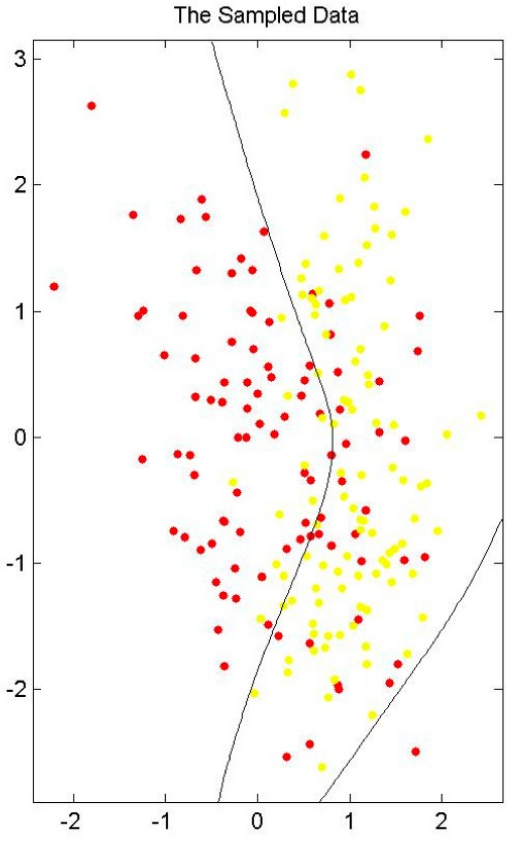
Surapprentissage : approche par régularisation

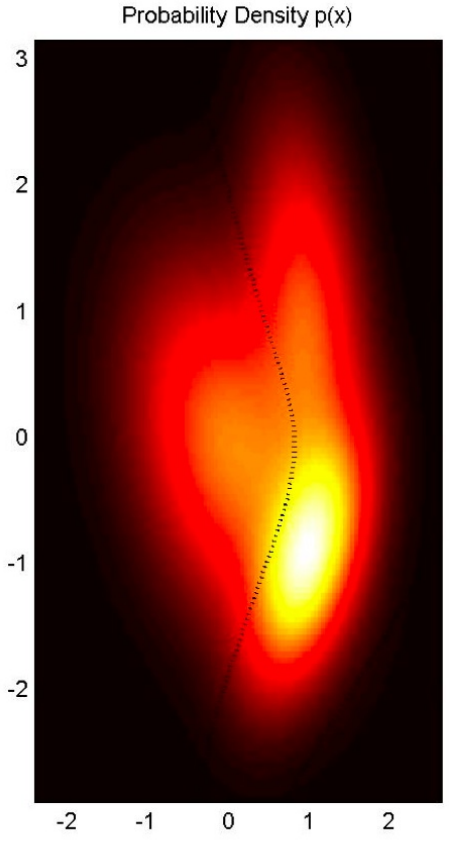
Exemple de l'approche par régularisation

- 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 : $\hat{h} = \arg\min_{h\in\mathcal{H}} R_n(h) + \lambda\Omega(h)$

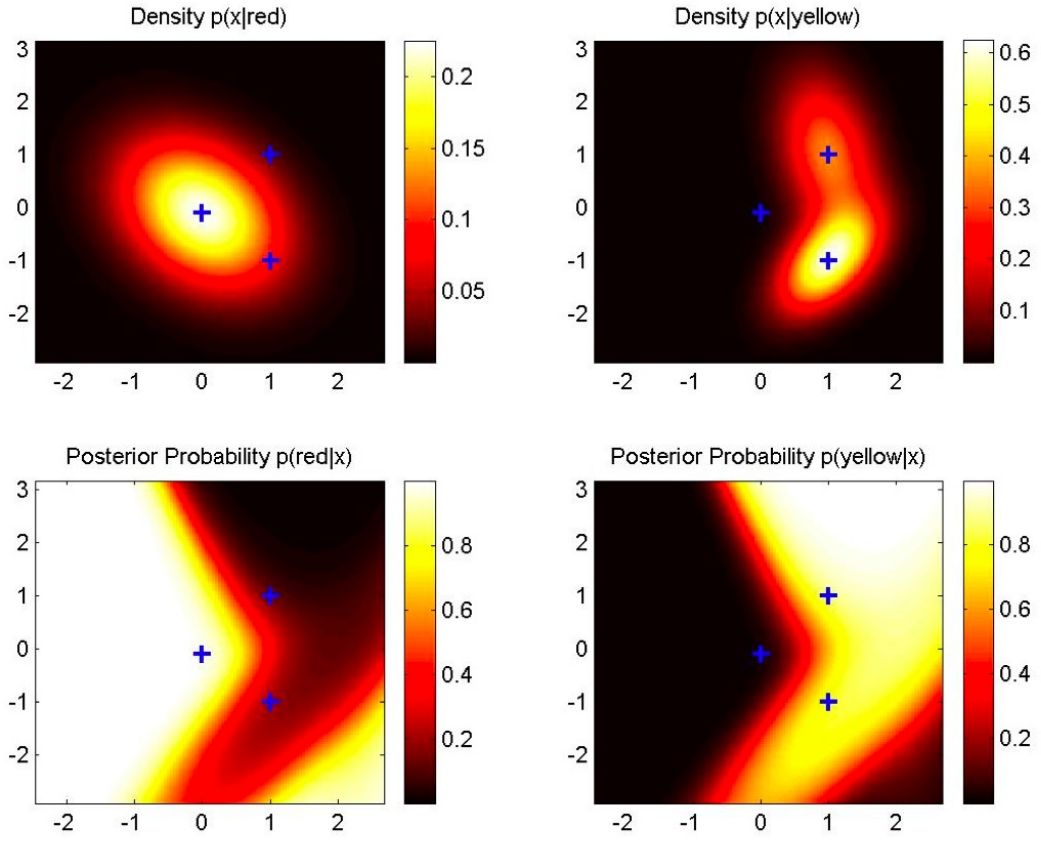
NB : on cherche à obtenir un compromis entre une bonne adéquation aux données et une complexité limitée : $\Omega(h)$ est en général choisi pour renforcer la régularité de la fonction

Exemple en 2D

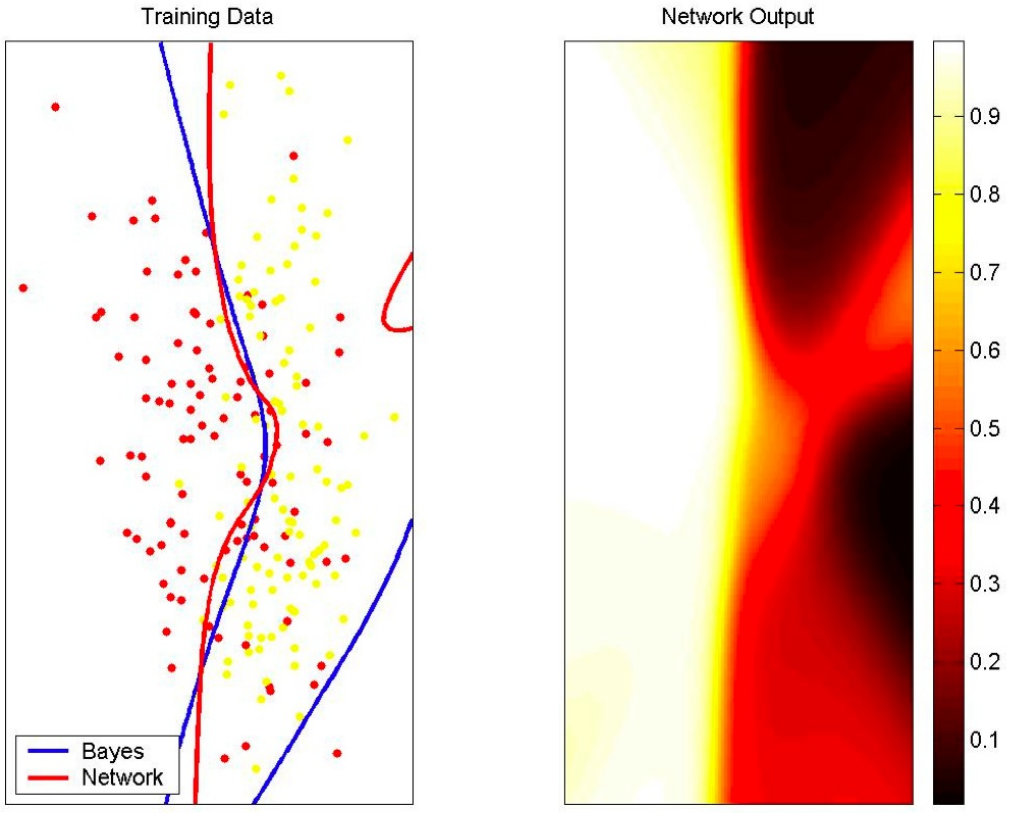




Exemple en 2D



En utilisant un ensemble d'apprentissage



Et en pratique, comment fait-on ?

Méthodologie pour développer une approche discriminante

- Définir
  - l'**espace de représentation** des entrées

Et en pratique, comment fait-on ?

Méthodologie pour développer une approche discriminante

- Définir
  - l'**espace de représentation** des entrées
  - la **classe des fonctions** de classification binaire considérées

Et en pratique, comment fait-on ?

Méthodologie pour développer une approche discriminante

- Définir
  - l'**espace de représentation** des entrées
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe

Et en pratique, comment fait-on ?

Méthodologie pour développer une approche discriminante

- Définir
  - l'**espace de représentation** des entrées
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe
  - l'**algorithme de minimisation** de cette fonction de coût

Et en pratique, comment fait-on ?

Méthodologie pour développer une approche discriminante

- Définir
  - l'**espace de représentation** des entrées
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe
  - l'**algorithme de minimisation** de cette fonction de coût
  - une **méthode de sélection de modèle** pour définir les hyperparamètres

Outline

Classe des fonctions de classification

Aujourd'hui, au programme : les premiers classifieurs (historiquement)

- Classifieur linéaire
  - Un premier exemple d'approche génératrice : l'analyse discriminante linéaire (sous hypothèse de normalité des classes)
  - Un second exemple d'approche génératrice : la régression logistique linéaire
- Classification non linéaire : k- plus-proches voisins

Analyse Discriminante Linéaire (en anglais, LDA) : 2 classes

La plus simple des approches génératrices !
ICI : $\mathcal{X} = ^p$

LDA

1. $p(x|Y = +1)$ and $p(x|Y = -1)$, densités supposés gaussiennes de matrice de covariance gales
2. $P(Y = +1) = 1 - P(Y = -1)$ supposés connus

$$h_{LDA}(x) = 1 \text{ if } \log\left(\frac{P(Y=+1|x)}{P(Y=-1|x)}\right) \geq 0, \text{ -1, sinon}$$

Analyse discriminante linéaire : 2 classes

Question: quelle est la forme géométrique de la frontière de décision définie par le classifieur LDA ?

Notations et définitions

- $\mu_+ \in \mathbb{R}^p$, $\mu_- \in \mathbb{R}^p$
- Σ: matrice symétrique définie positive
- $$p(x|Y=+1) = \frac{1}{2\pi^{p/2}|\Sigma|^{1/2}} \exp(-(x-\mu_+)^T\Sigma^{-1}(x-\mu_+)) \quad (4)$$
- $P(Y=+1) = p_1$
- $$p(x|Y=-1) = \frac{1}{2\pi^{p/2}|\Sigma|^{1/2}} \exp(-(x-\mu_-)^T\Sigma^{-1}(x-\mu_-)) \quad (5)$$
- $P(Y=-1) = 1-p_1$

Réponse

Formule de Bayes:

$$P(Y = i|x) = \frac{p(x|Y = i)P(Y = i)}{p(x)}$$

Puis, on cherche à définir la frontière de décision induite par le classifieur LDA:

Analyse Discriminante Linéaire

$$\log\left(\frac{P(Y=+1|x)}{P(Y=-1|x)}\right)=0$$

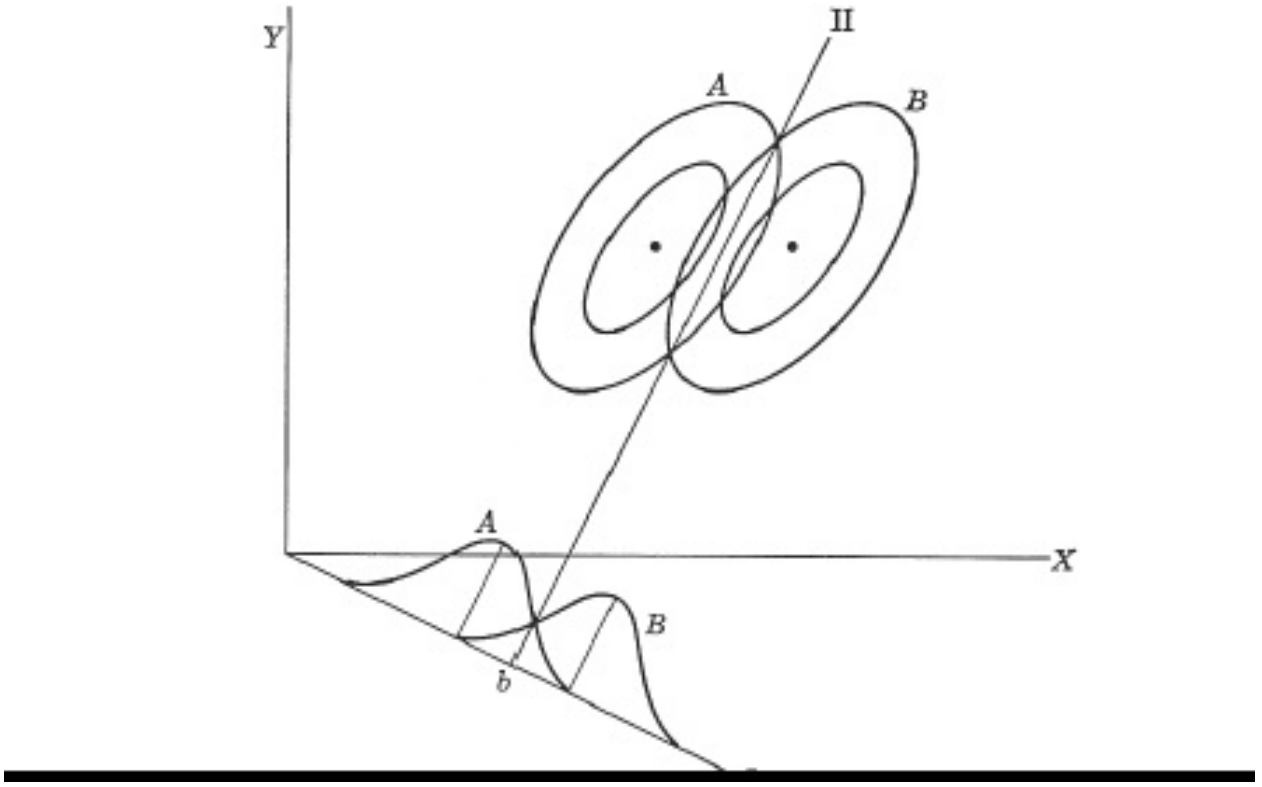
$$\text{soit } \log\left(\frac{p(x|Y=1)P(Y=1)}{p(x|Y=-1)P(Y=-1)}\right)=0$$

$$\log(\frac{p_1}{1-p_1})+\log(\frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}})-\frac{1}{2}(x-\mu_+)^T\Sigma^{-1}(x-\mu_+)-\log(\frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}})+\frac{1}{2}(x-\mu_-)^T\Sigma^{-1}(x-\mu_-)=0$$

$$x^T\Sigma^{-1}(\mu_+-\mu_-)+\log(\frac{p_1}{1-p_1})-\frac{1}{2}(\mu_+-\mu_-)^T\Sigma^{-1}(\mu_+-\mu_-)=0$$

Analyse Discriminante Linéaire

Le cas de deux classes aux matrices de covariances identiques



Estimation des paramètres (LDA)

Maximum de vraisemblance pour chaque sous-échantillon correspondant à une classe.
On se rappelle que la moyenne empirique (resp. la covariance empirique) est exactement l'estimateur de la moyenne par Maximum de vraisemblance.

Estimation des paramètres (LDA)

- Prendre les estimations empiriques définies à partir des données
- $S_+ = \{(x_i, y_i) \in S,\ s.t\ y_i = 1\}$
- $S_- = \{(x_i, y_i) \in S,\ s.t\ y_i = -1\}$

$$\hat{\mu_+} = \frac{1}{|S_+|} \sum_{x_i \in S_+} x_i$$

$$\hat{\mu_-} = \frac{1}{|S_-|} \sum_{x_i \in S_-} x_i$$

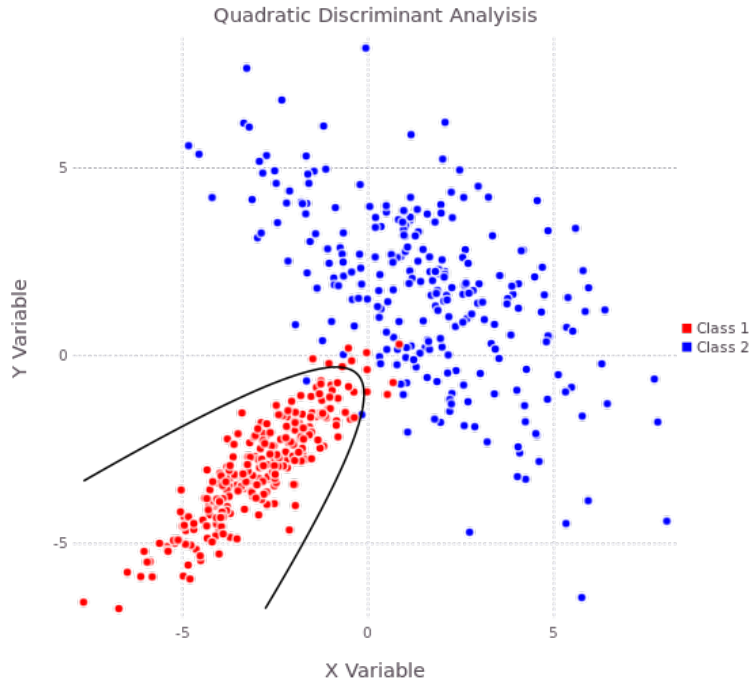
$$\hat{\Sigma} = \frac{1}{2} \left(\frac{1}{|S_+|} \sum_{x_i \in S_+} (x_i - \hat{\mu}_+)(x_i - \hat{\mu}_+)^T + \frac{1}{|S_-|} \sum_{x_i \in S_-} (x_i - \hat{\mu}_-)(x_i - \hat{\mu}_-)^T \right)$$

Calculs pour LDA

1. Transformer les données (sphere the data) selon la covariance commune diagonalisée:
   - $\widehat{\Sigma} = UDU^T$
   - $X^* = D^{-1/2}U^T X$
2. Les données de chaque classe ont pour covariance l'identité: simplification de la fonction f_{LDA} avec covariance = identité
3. Alors, associer un nouveau point au "centre" de la classe la plus proche modulo l'effet des *a priori*.

Analyse Discriminante Quadratique

Cas de matrices de covariance différentes.
Le terme quadratique en x reste dans l'équation.



Outline

Régression logistique 1 / 3

On cherche à modèliser:

$$\eta(x) = \mathbb{P}(Y = 1|x)$$

C'est une probabilité, on impose qu'elle soit entre 0 et 1 en fixant la forme suivante:

$$\eta_\theta(x) = \frac{\exp(g_\theta(x))}{1 + \exp(g_\theta(x))},$$

où f_θ est une fonction à choisir.

Régression logistique 2/3

La transformation *logistique* ou *logit* est définie par:

$$\text{logit}(\eta(x)) = \log[\frac{\eta(x)}{1-\eta(x)}] = g(x) \tag{6}$$

La transformation inverse: $\eta(x) = \frac{\exp((g(x))}{1+\exp(g(x))}$

On parle de *régression logistique* lorsque $g(x) = g_\theta(x) = \beta_0 + \beta_1^T x$:

$$\log \frac{\mathbb{P}(Y=1|x)}{\mathbb{P}(Y=-1|x)} = \beta_0 + \beta_1^T x \tag{7}$$

Régression logistique 2/3

Au final, pendant l'apprentissage, j'utilise:

$$f_{log}(x) = \eta_\beta(x) = \frac{\exp(\beta_0 + \beta_1^T x)}{1 + \exp(\beta_0 + \beta_1^T x)}$$

Remarquons que :

$$f_{log}(x) = \eta_\beta(x) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1^T x))}$$

Pour prendre une décision : x est-il au dessus ou au dessous de l'hyperplan d'équation: $\beta_0 + \beta_1^T x = 0$?

$$h_{log}(x) = \text{signe}(f(x) - 1/2)$$

Apprendre = Estimer une régression logistique

Prenons : $y_i \in \{0, 1\}, i = 1, \ldots, n$. Dans notre modèle, $Y|x$ suit une loi binômiale avec n=1 et k=1. Maximiser la log-vraisemblance conditionnelle :

$$\mathcal{L}(\beta) = \log \prod_{i=1}^{n} \hat{\mathbb{P}}(Y = 1|x_i)^{y_i} (1 - \hat{\mathbb{P}}(Y = 1|x_i))^{(1-y_i)}$$

Soit à maximiser:

$$\mathcal{L}(\beta) = \sum_{i=1}^{n} y_i \log(\eta_\beta(x_i)) + (1 - y_i) \log(1 - \eta_\beta(x_i))$$

Apprendre une régression logistique

$$\mathcal{L}(\beta) = \sum_{i=1}^{n} \log(1 - \eta_\beta(x_i)) + \sum_{i=1}^{n} y_i \log \frac{\eta_\beta(x_i)}{1 - \eta_\beta(x_i)}$$

D'après Eq. (6) et la définition de η_β, on a:

$$\begin{aligned} \mathcal{L}(\beta) &= \sum_{i=1}^{n} \log(1 - \eta_\beta(x_i)) + \sum_{i=1}^{n} y_i(\beta_0 + \beta_1^T x) \\ &= -\sum_{i=1}^{n} \log(1 + \exp(\beta_0 + \beta_1^T x_i)) + \sum_{i=1}^{n} y_i(\beta_0 + \beta_1^T x) \end{aligned}$$

On écrit les conditions d'optimalité du premier ordre :

$$\frac{\partial \mathcal{L}}{\partial \beta_j} = -\sum_{i=1}^{n} (y_i - \eta_\beta(x_i)) x_{ij} = 0$$

Optimisation

Pas de solution explicite, alors on implémente la méthode de Newton-Raphson: On part de β_0 (initialisation aléatoire). Itérativement,

$$\beta^{new} = \beta^{old} - (\frac{\partial^2 \mathcal{L}(\beta)}{\partial \beta \beta^T})^{-1} \frac{\partial \mathcal{L}(\beta)}{\partial \beta},$$

$(\frac{\partial^2 \mathcal{L}(\beta)}{\partial \beta \beta^T})^{-1}$ est l'inverse de la matrice hessienne

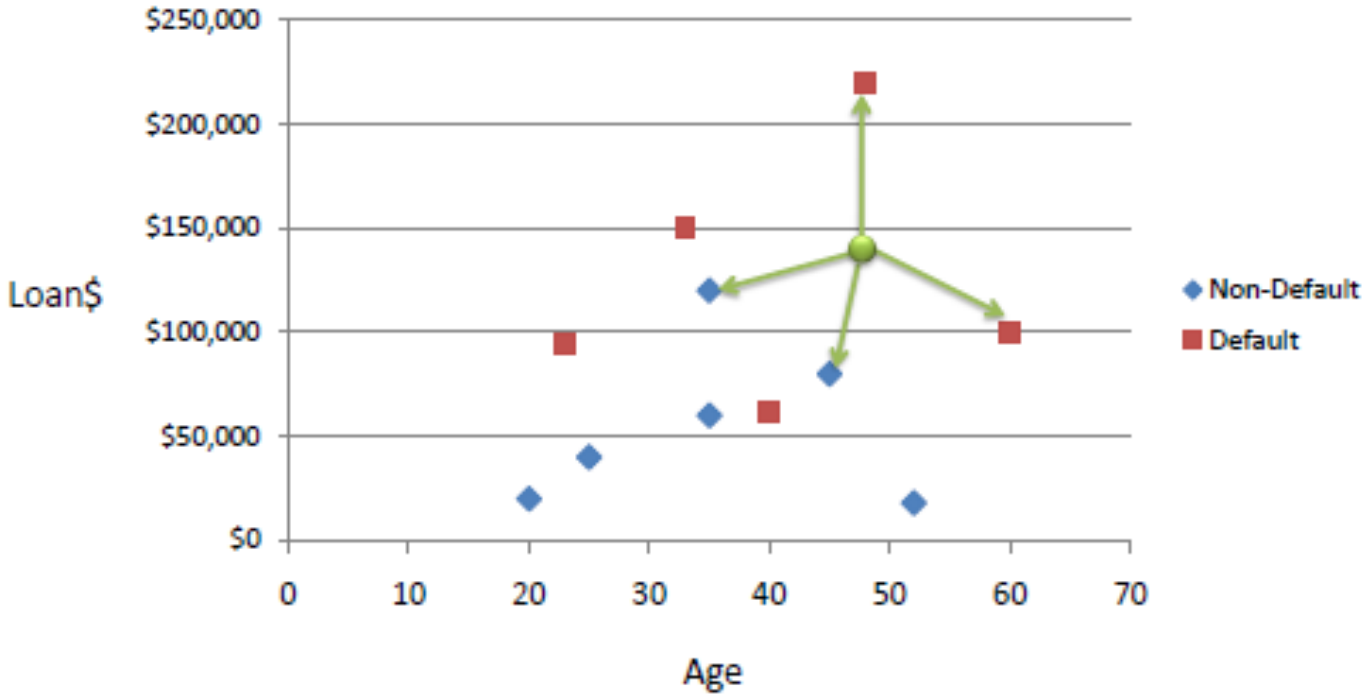
Alternative: une autre fonction de transformation "probit"

Probit model:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-t^2/2)dt$$

$$\phi^{-1}(\eta(x)) =_0 + \beta_1^T x$$

Algorithme des K-plus-proches voisins



Algorithme des K-plus-proches voisins

K-PPV (en anglais K-Nearest neighbors: K-NN)

Cas 2 classes:

$$h_{KNN}(x) = \arg \max_{y \in \{-1,1\}} \frac{N_y^K(x)}{K},$$

avec :

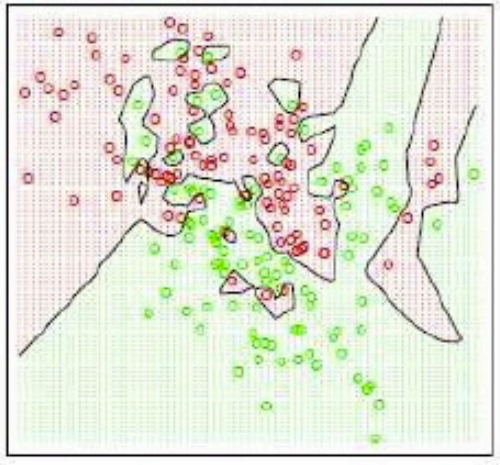
- Soit K un entier strictement positif.
- Soit d une métrique définie sur $x \times x$
- $S = \{(x_i, y_i), i = 1, \ldots, n\}$
- Pour une donnée x, on définit σ la permutation d'indices dans $\{1, \ldots, n\}$ telle que:
  - $d(x, x_{\sigma(1)}) \leq d(x, x_{\sigma(1)}) \ldots \leq d(x, x_{\sigma(n)})$
- $S_x^K = \{x_{\sigma(1)}, \ldots, x_{\sigma(K)}\}$: K premiers voisins de x
- $N_y^K(x) = |\{x_i \in S_x^K, y_i = y\}|$

Le paramètre de lissage K

K: trop petit : la fonction f est trop sensible aux données : large variance

K : trop large : la fonction f devient trop peu sensible aux donnes : biais important

K=1



K=15

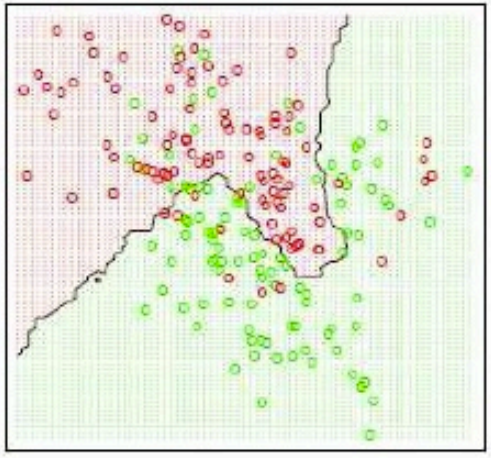


Fig 2.2, 2.3 of HTF01

Book

Décomposition biais-variance

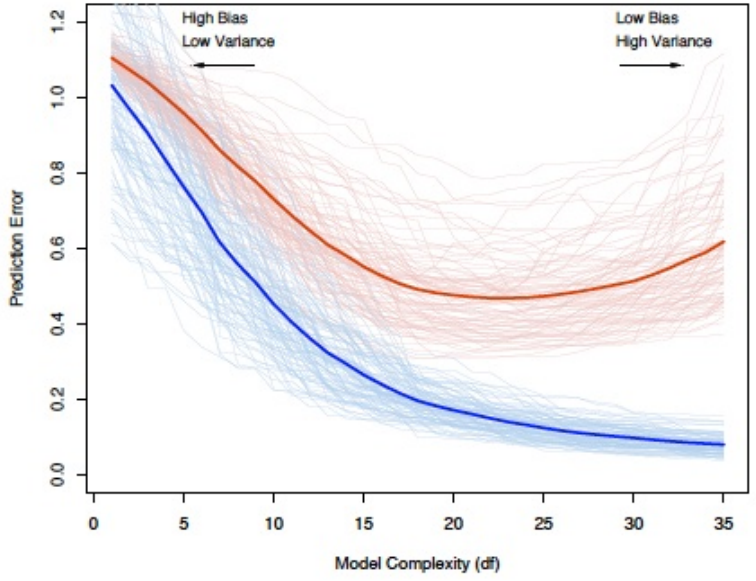
On suppose: $Y = f(X) + \epsilon$ avec ϵ centré et de variance σ_ϵ^2. x est fixé.

$$\begin{aligned}
E_{S,Y}[(Y - \hat{f}(x))^2] &= E_{S,Y}[Y^2 + \hat{f}(x) - 2Y\hat{f}(x)] \\
&= E[Y^2] + E_S[\hat{f}(x)^2] - 2E_S[Y\hat{f}] \\
&= VarY + E[Y]^2 + Var\hat{f}(x) +_S [\hat{f}]^2 - 2E[f(x) + \epsilon]E_S[\hat{f}(x)] \\
&= \sigma_\epsilon^2 + E[f(x) + \epsilon]^2 + E_S[\hat{f}]^2 - 2E_S[f(x)]E_S[\hat{f}(x)] + Var\hat{f}(x) \\
&= \sigma_\epsilon^2 + E_S[\hat{f}(x) - f(x)]^2 + Var\hat{f}(x) \\
&= \sigma_\epsilon^2 + \text{Biais}^2 + \text{variance}
\end{aligned}$$

Terme incompressible : bruit des données
Biais au carré: mesure à quel point $\hat{f}$ est loin de la cible
Variance de $\hat{f}(x)$: mesure à quel point $\hat{f}(x)$ est sensible aux données

Biais variance

Soit M datasets $S_1, \ldots, S_M$ de même taille n. Apprenons pour chacun d'entre eux, une fonction $\hat{f}$ sous différentes contrainte de complexité (ici nombre degr/'e de libertés).On a tracé sur cette figure la courbe des erreurs en apprentissage (bleue) et des erreurs en test (rouge) pour chacune des fonctions construites:



Book of Hastie, Tibshirani and Friedman (The elements of statistical learning, Springer)

Décomposition biais-variance des k-plus-proches voisins

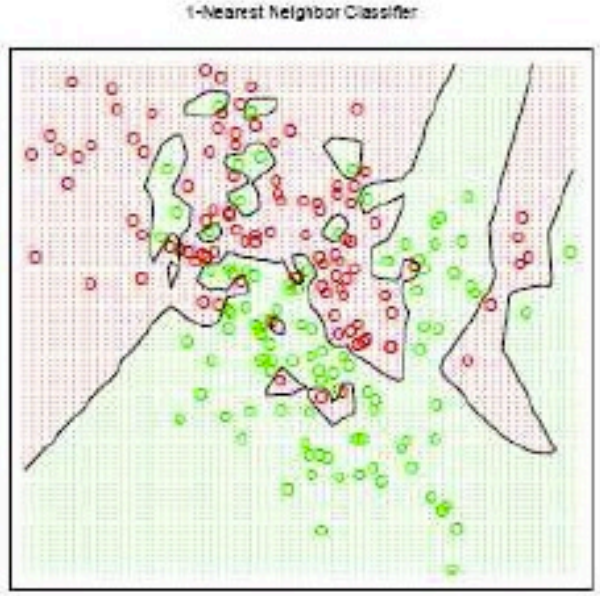
Posons x_0. L'aléa vient de l'échantillon utilis pour apprendre $\hat{f}$ et de Y. On peut montrer que:

$$E_{S,Y}[(Y - \hat{f}(x_0))^2] = \sigma_\epsilon^2 + (f(x_0) - \frac{1}{K}\sum_{\ell=1}^{K} f(x_{(\ell)})^2 + \frac{\sigma_\epsilon}{K}$$

K contrôle le terme de variance : plus grande est la valeur de K, plus la variance décroit; mais K contrôle aussi le biais, plus petite est la valeur de K, plus petit est le biais : nous sommes en plein *dilemne biais-variance*.

Le paramètre de lissage K

K=1



K=15

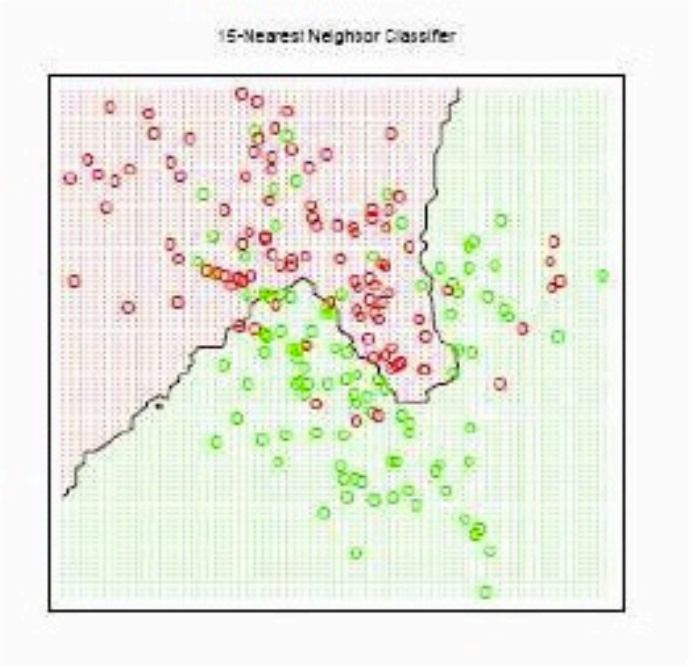
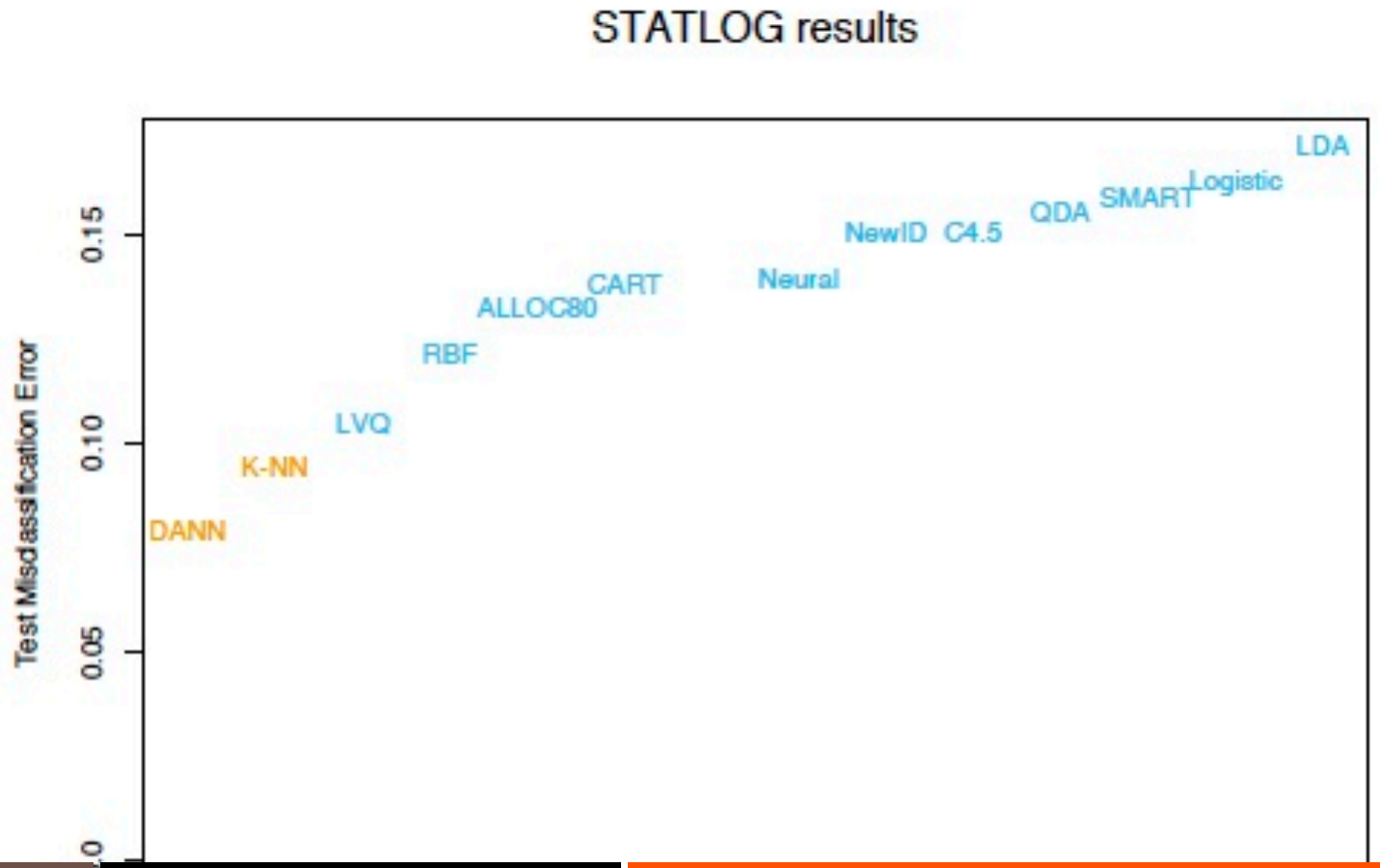


Fig 2.2, 2.3 of HTF01

Comparaison: statlog data

LANDSAT images fo classsification Hastie, Tibshirani, Friedman's book.



Importance de la métrique

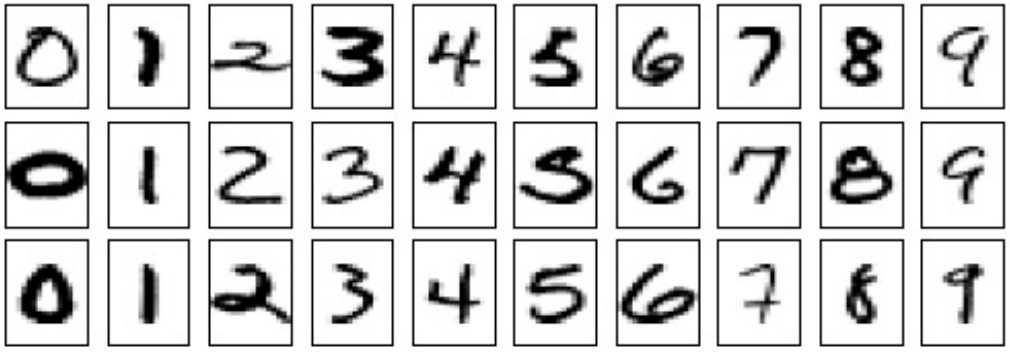
Rappel:

2 questions duales : choix de représentation des entrées x et choix de la classe de fonctions

Dans les K-plus-proches voisins, c'est le choix de la métrique qui définit le modèle avec le nombre K

Importance de la métrique

Un des meilleurs scores pour la base de données USPOST en classification de caractères manuscrits: k-NN + tangent distance Simard et al. 2000 et 2012: `http://link.springer.com/chapter/10.1007/978-3-642-35289-8_17#page-1`



Distance Tangente

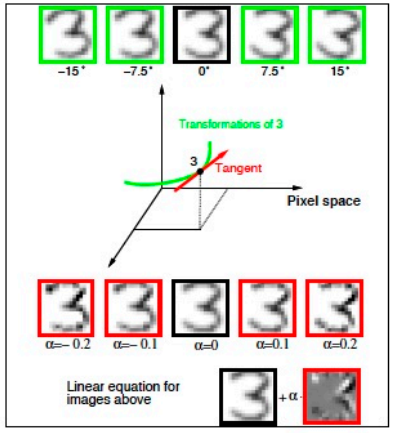


FIGURE 13.10. *The top row shows a "3" in its original orientation (middle) and rotated versions of it. The green curve in the middle of the figure depicts this set of rotated "3" in 256-dimensional space. The red line is the tangent line to the curve at the original image, with some "3"s on this tangent line, and its equation shown at the bottom of the figure.*

La distance entre deux images est la distance euclidienne la plus courte entre n'importe quelle image transformée de l'image 1 par une rotation et n'importe quelle image transformée de l'image 2.

Outline

References

- W. Vapnik, An overview of statistical learning theory,IEEE Trans. Neural Networks, 2001.
- HTF: The elements of statistical learning: chapters 1, 2 and 4
- Chapitre 4, Pattern Recognition and Neurla Networks, C. Bishop, Springer, 2006.

Outline

Vapnik-Chervonenkis dimension

Definition: **VC-dimension**
The VC-dimension of a hypothesis set $\mathcal{H}$ is the size of the largest set that can be fully shattered by $\mathcal{H}$:

$$VCdim(\mathcal{H}) = max\{m : \exists(\mathbf{x}_1, \ldots, \mathbf{x}_m) \in \mathcal{X}^m \text{that are shattered by } \mathcal{H}\}$$

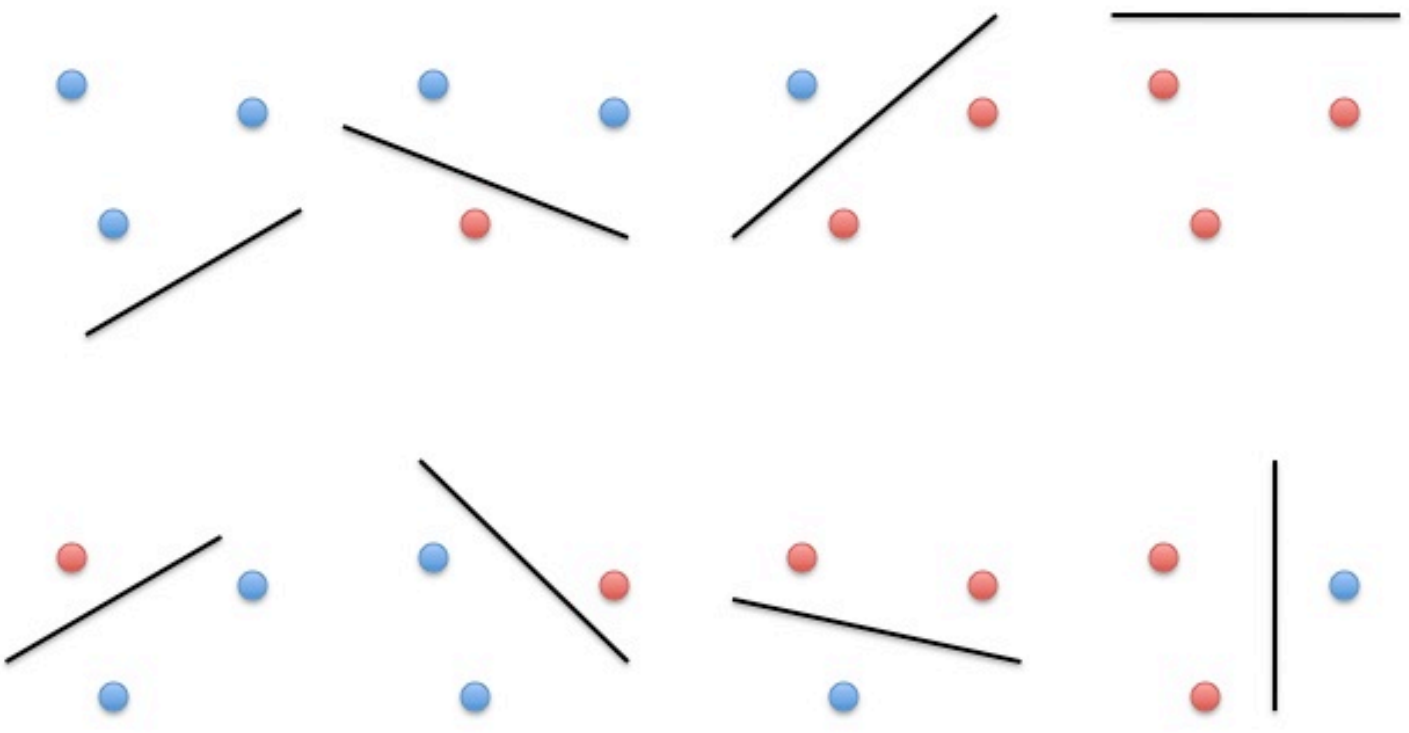
N.B.: if VCdim($\mathcal{H}$) = d, then there exists a set of d points that is fully shattered by $\mathcal{H}$, but this DOES NOT imply that all sets of dimension d or less are fully shattered !

VC-dimension of Hyperplanes

What is the VC-dimension of hyperplanes in $\mathbb{R}^2$ (denoted $\mathcal{H}_2$) ?
Obviously $\text{VCdim}(\mathcal{H}_2) \geq 2$
Let us try with 3 points :

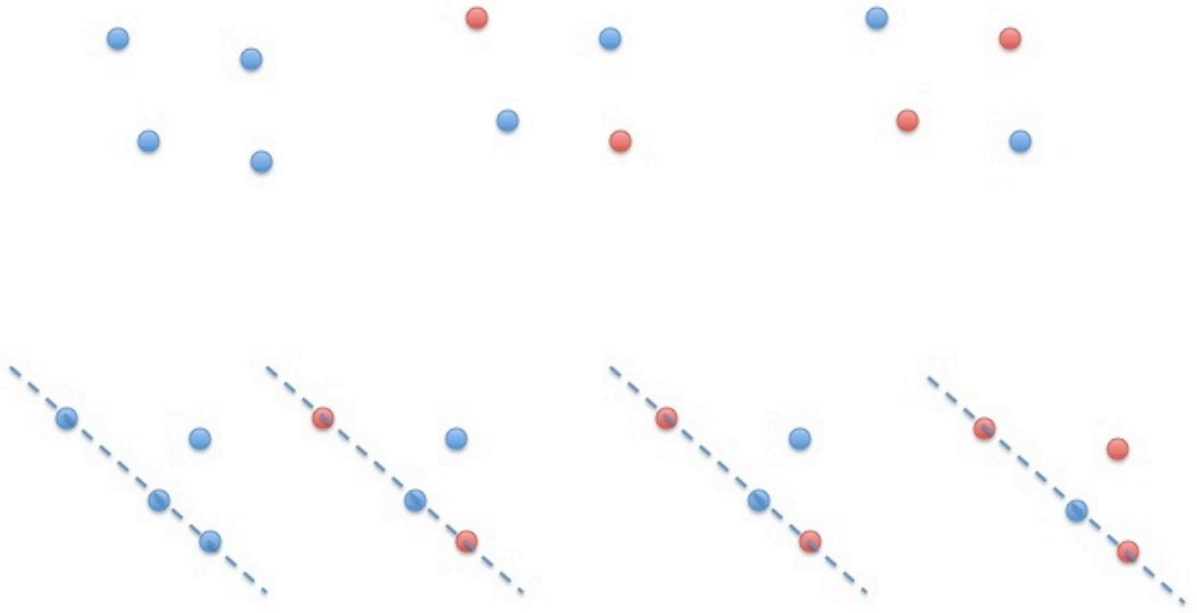
VC-dimension of Hyperplanes

What is the VC-dimension of hyperplanes in $\mathbb{R}^2$ (denoted $\mathcal{H}_2$) ?
Let us consider the following triplet of points



VC-dimension of Hyperplanes

What is the VC-dimension of hyperplanes in $\mathbb{R}^2$ (denoted $\mathcal{H}_2$) ?
For any set of 4 points, either 3 of them (at least) are aligned or no triplet of points is aligned.



We can show that it is not possible for $\mathcal{H}_2$ to shatter 4 points.
Then $\text{VCdim}(\mathcal{H}_2) = 3$.

VC-dimension of Hyperplanes

More generally, one can prove :

$$VCdim(\mathcal{H}_d) = d + 1$$

VC-dimension generalization bounds

Theorem:
Let $\mathcal{H}$ be a family of functions taking values in $\{-1, +1\}$ with VC-dimension d. Then, for any $\delta > 0$, with probability at least $1 - \delta$, over a random sampling $\mathcal{S} \sim D^n$, the following holds for all $h \in \mathcal{H}$:

$$R_D(h) \leq R_{\mathcal{S}}(h) + \sqrt{\frac{2d \log(\frac{em}{d})}{n}} + \sqrt{\frac{\log(\frac{1}{\delta})}{2n}}$$