REsearch and methodology in Data Science

Cours 2 – Méthodes d'ensemble

Olivier Schwander <olivier.schwander@lip6.fr>

Master DAC Data Science
UPMC - LIP6

2020-2021

Rappel (ou pas) sur les arbres de décision

Contexte

- Données numériques dans $\mathbb{R}^N$
- Sortie: 1 classe parmi K
- Apprentissage: trouver le meilleur vecteur de paramètres θ pour l fonction $f_\theta(x)$ qui associe une catégorie à un vecteur x
- Multiples modèles: plusieurs choix possibles pour f

Rappel (ou pas) sur les arbres de décision

Arbres de décision

- Modèle: ensemble de décisions binaires organisées sous forme d'arbre
- *Noeuds*: **test sur une features** :$x_3 > 0.6$
- *Feuilles*: décisions dans $1, 2, ..., K$
- Construction de haut en bas: choix d'une feature, choix d'un seuil, et récursivement
- Pleins d'algorithmes: CART, C4.5, ID3, ...

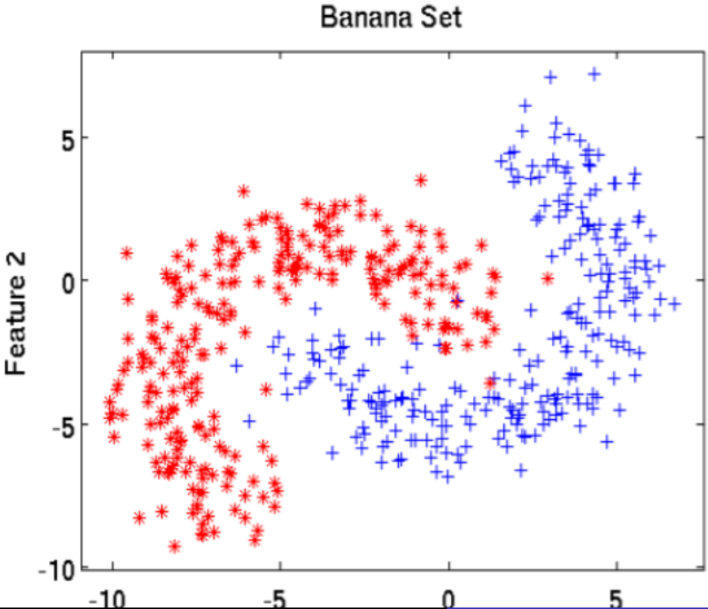
Post-traitement

- Élagage pour améliorer la généralisation

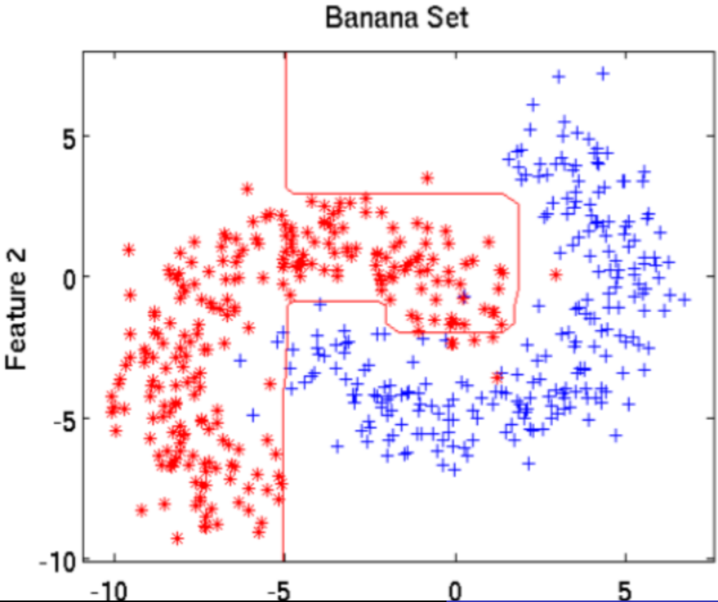
Inférence

- Descente dans l'arbre du nouveau point

Rappel (ou pas) sur les arbres de décision



Rappel (ou pas) sur les arbres de décision

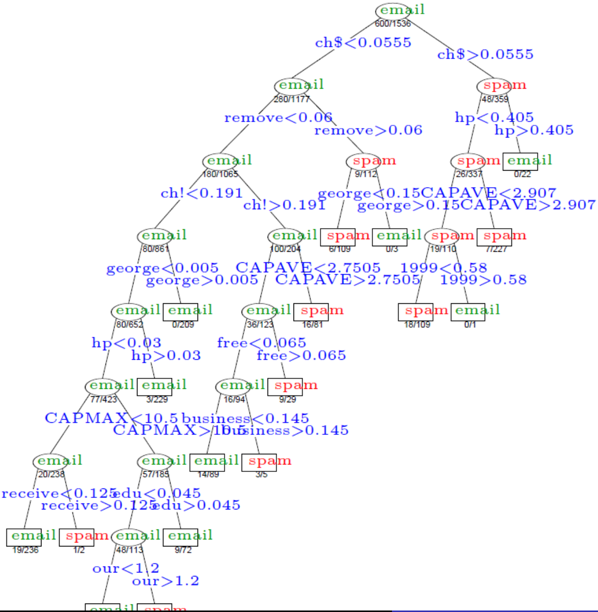


Rappel (ou pas) sur les arbres de décision

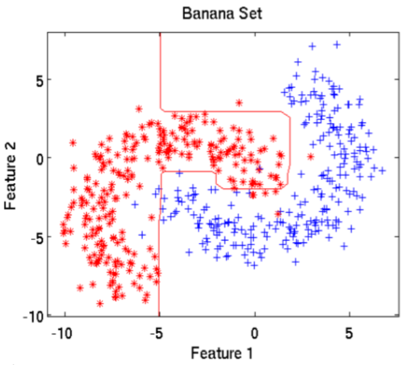
Avantages

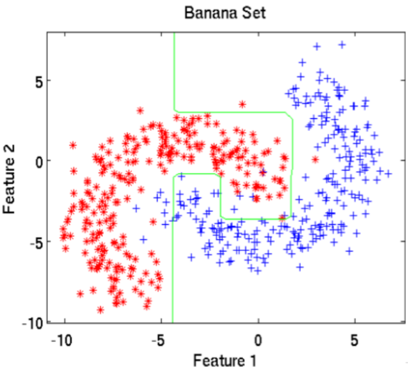
- Capable de digérer de grands jeux de données
- Interprétables
- Autres avantages: variables manquantes, variables redondantes, entrées qualitatives et quantitatives

Rappel (ou pas) sur les arbres de décision



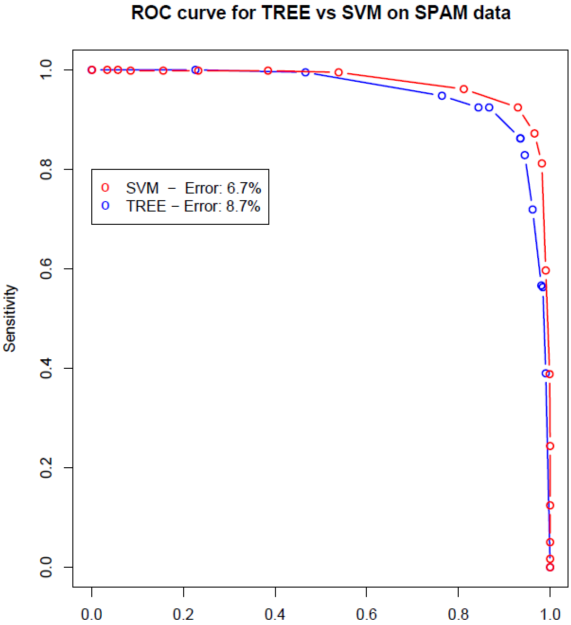
Rappel (ou pas) sur les arbres de décision





Inconvénient: instable

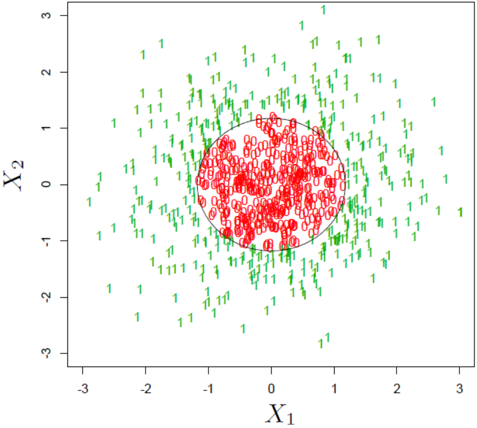
Rappel (ou pas) sur les arbres de décision



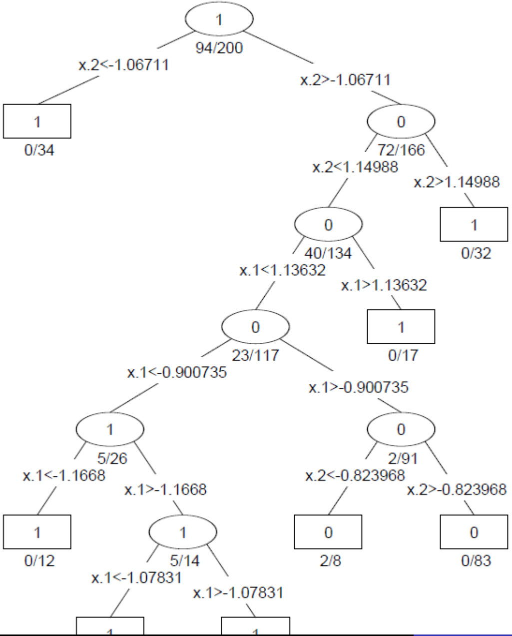
Rappel (ou pas) sur les arbres de décision

Sphères imbriquées

- Deux sphère l'une dans l'autre, en n dimension
- Sans bruit
- Erreur de Bayes nulle

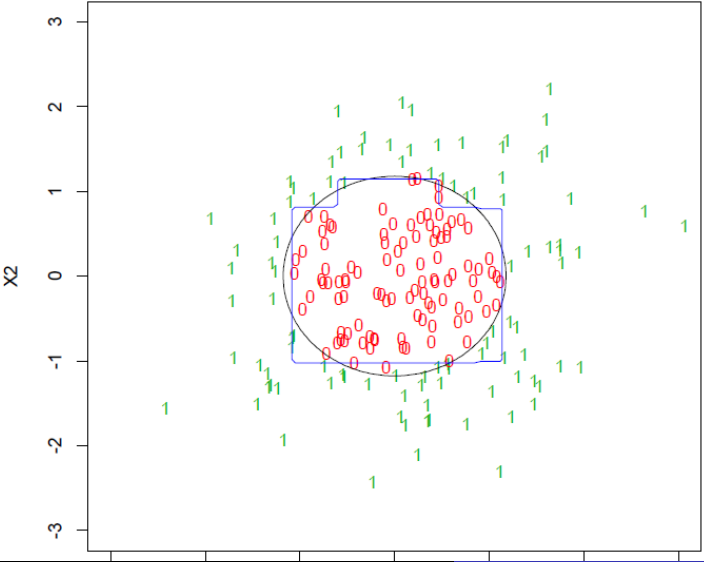


Rappel (ou pas) sur les arbres de décision



Rappel (ou pas) sur les arbres de décision

Dimension 10: erreur > 0.3



Décomposition Biais-Variance

Biais

- Le biais mesure la qualité d'un prédicteur. Un grand biais signifie que le modèle n'est pas performant, et provient souvent du fait de mauvaises hypothèses dans la classe de fonctions utilisées.
- Biais élevé: sous-apprentissage

Variance

- La variance mesure la sensibilité du classifieur à de petites fluctuations dans l'ensemble d'apprentissage. Une grande variance correspond à une mauvaise généralisation.
- Variance élevée: sur-apprentissage

Compromis Biais-Variance

- Idéalement, on souhaite minimiser les deux simultanément. *Mais* il y a un compromis à trouver !

Problème clef de l'apprentissage statistique

Décomposition Biais-Variance

- Soit un ensemble de points d'apprentissage $S = x^1, ..., x^n$ et les sorties associées y^i.
- Soite $y = f_\theta(x) + \epsilon$ où ϵ est un bruit gaussien de moyenne 0 et de variance σ^2
- On cherche la fonction $\hat{f}$ qui approche f au sens de l'erreur des moindres carrées $(y - \hat{f}(x))^2$
- Etant donné un nouveau point x, y, on analyse le comportement du modèle sur ce nouveau point.
- Supposons que S est tiré selon la loi de probabilité P, nous allons calculer la valeur suivante:

$$E_P[(y - \hat{f}(x))^2]$$

Décomposition Biais-Variance

Soit Z une variable aléatoire et $\bar{Z} = E_P[Z]$ sa moyenne.

$$\begin{aligned}
E[(Z-\bar{Z})^2] &= E[Z^2 - 2Z\bar{Z} + \bar{Z}^2] \\
&= E[Z^2] - 2E[Z]\bar{Z} + \bar{Z}^2 \\
&= E[Z^2] - 2\bar{Z}^2 + \bar{Z}^2 \\
&= E[Z^2] - \bar{Z}^2
\end{aligned}$$

Et donc $E[Z^2] = E[(Z-\bar{Z})^2] + \bar{Z}^2$

Décomposition Biais-Variance

$$
\begin{aligned}
E[((\hat{f}(x) - y)^2] &= E[\hat{f}(x)^2 - 2\hat{f}(x)y + y^2] \\
&= E[\hat{f}(x)^2] - 2E[\hat{f}(x)]E[y] + E[y^2] \\
&= E[(\hat{f}(x) - \bar{\hat{f}}(x))^2] + \bar{\hat{f}}(x))^2 \\
&\quad - 2\bar{\hat{f}}(x)f(x) \\
&\quad + E[(y - f(x))^2] + f(x)^2 \\
&= E[(\hat{f}(x) - \bar{\hat{f}}(x))^2] + (\bar{\hat{f}}(x) - f(x))^2 \\
&\quad + E[(y - f(x))^2]
\end{aligned}
$$

$$
\underbrace{E[(\hat{f}(x) - \bar{\hat{f}}(x))^2]}_{\text{Variance}(\hat{f}(x))} + \underbrace{(\bar{\hat{f}}(x) - f(x))^2}_{\text{Biais}(\hat{f}(x))^2} + \underbrace{E[(y - f(x))^2]}_{\text{Bruit}(\sigma^2)}
$$

Décomposition Biais-Variance

Estimation

- Un seul S disponible
- Simuler plusieurs S par tirage avec remise (bootstrap)

Bagging

Bootstrap AGGregatING: méthode pour réduire la variance par moyennage

Combinaisons de modèles

Soit $\hat{f}_1, ..., \hat{f}_B$ un ensemble de modèles, on peut constuire un modèle aggrégé par:

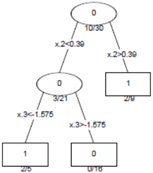
- Moyenne des prédiction des modèles (régression)
- Vote majoritaire (classification)

Bagging

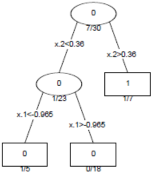
- Bootstrap pour avoir plusieurs ensembles d'apprentissage
- Apprentissage d'un modèle sur chaque ensemble
- Combinaison

Bagging

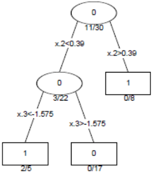
Original Tree



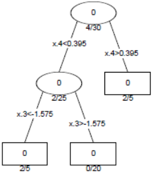
Bootstrap Tree 1



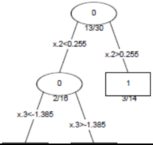
Bootstrap Tree 2



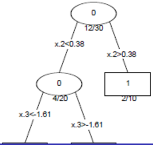
Bootstrap Tree 3



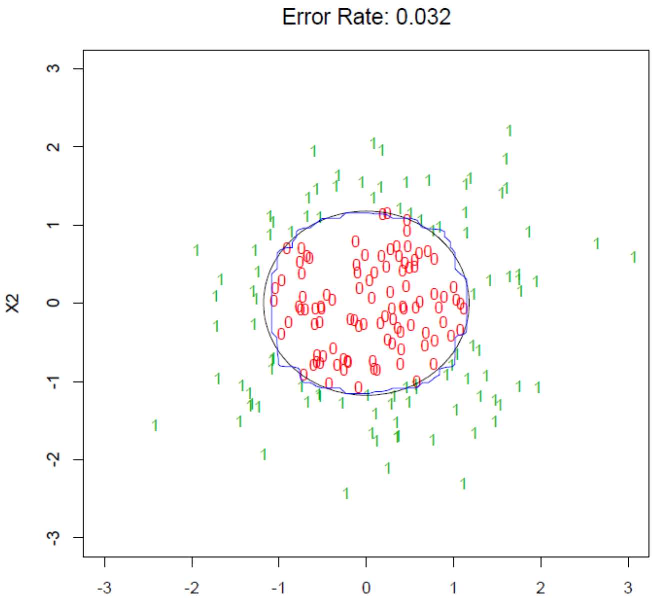
Bootstrap Tree 4



Bootstrap Tree 5



Bagging



Bagging

Modèle appris par bagging:

$$\hat{f}(x) = \frac{1}{B} \sum_i \hat{f}_i(x)$$

Rappel:

- Biais $= (\bar{\hat{f}}(x) - f(x))^2$
- Variance $= E[(\hat{f}(x) - \bar{\hat{f}}(x)^2]$

Le bagging réduit la variance, et augmente le biais légèrement.

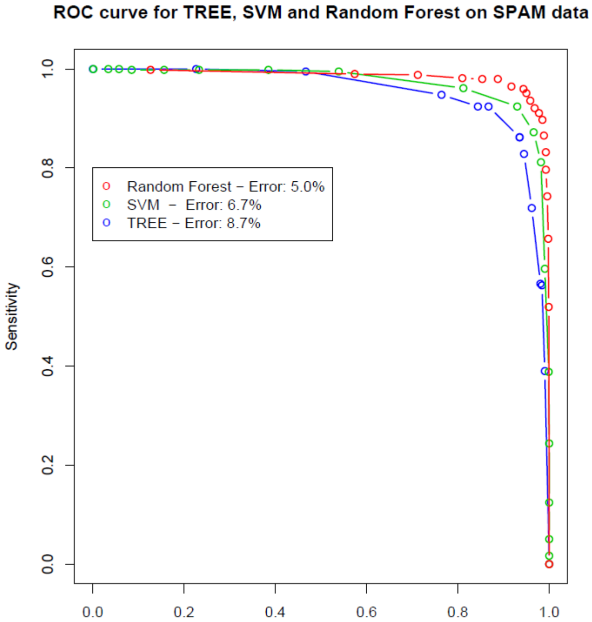
Fôrets aléatoires

Random Forests

- Bagging d'arbres de décision
- À chaque *split*, un échantillon aléatoire de m features est tiré (décorrélation des arbres). (Typiquement, $m = \sqrt{n}$ ou $\log_2(n)$)
- Chaque arbre est appris sur un *bootstrap* de l'échantillon original.

L'erreur est évaluées sur les points qui n'ont pas été pris dans les échantillons samplés

Fôrêts aléatoires



Boosting

Classifieur faible

- Accuracy strictement supérieure à 50%
- Pas forcément beaucoup plus

Boosting

- Terme générique pour la combinaison de classifieur faibles
- Combinaison: classifieur très performant

Idée

- Apprentissage succesif de modèles
- Pondération des exemples d'apprentissage:
  - Points bien prédits $\Rightarrow$ poids faible
  - Points mal prédits $\Rightarrow$ poids fort
- Focalisation sur les parties de l'espace mal prédits.

Boosting: AdaBoost

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
- Update:

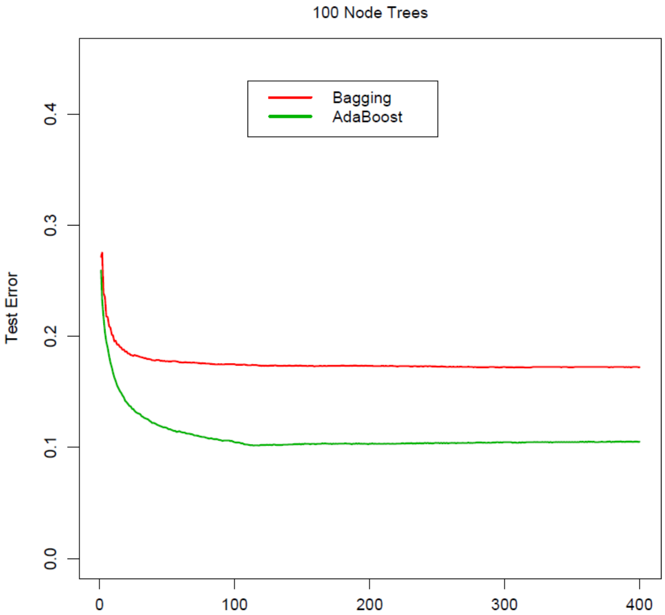
$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

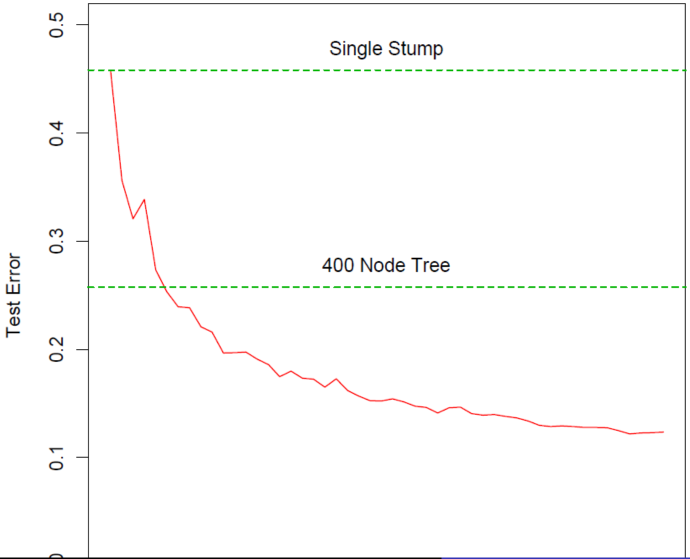
$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Boosting


100 Node Trees
Bagging
AdaBoost
Test Error
0.0
0.1
0.2
0.3
0.4
0
100
200
300
400

Boosting: stumps

Stump: arbre de décision à un nœud



Boosting: interprétation

Règle de décision

$$H = \text{signe}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

- Hyperplan en dimension T

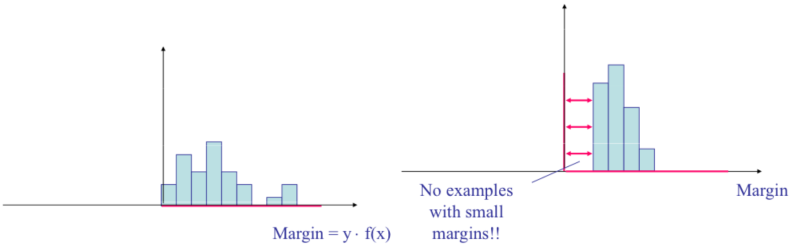
Apprentissage de représentation

- Plongement dans un espace de dimension T
- Décision linéaire dans cet espace

Boosting: succès

Chaque étape

- Augmente le poids là où la marge est la plus faible
- Continue à augmenter la marge globale



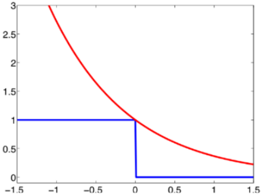
Hypothèse finale

- Complexe
- Mais proche d'une hypothèse simple

Boosting: coefficient

Surrogate

- Majoration de la fonction d'erreur
- Coût exponentiel



$$\ell(h(\mathbf{x}), y) \;=\; e^{-y \cdot h(\mathbf{x})}$$

Boosting: coefficient

Classifieur à l'étape t

$$H_{(t-1)}(x) = \alpha_1 h_1(x) + \cdots + \alpha_{m-1} h_{t-1}(x)$$

Nouveau classifieur faible h_t

$$H_t(x) = H_{t-1}(x) + \alpha_t h_t(x)$$

Risque empirique

$$\begin{aligned}
R(H_t) &= \sum_i \exp\left(-y_i(H_{t-1}(x_i) + \alpha_t h_t(x_i))\right) \\
&= \sum_i \exp\left(-y_i H_{t-1}(x_i)\right) \exp\left(-y_i \alpha_t h_t(x_i)\right) \\
&= \sum_{x_i \text{ mal classés}} W_{t-1} \exp\left(\alpha_t\right) + \\
&\quad \sum_{x_i \text{ bien classés}} W_{t-1} \exp\left(-\alpha_t\right))
\end{aligned}$$

Gradient-Boosting

- Invention de Adaboost (1996, 1997)
- Formulation de Adaboost comme un problème de descente de gradient pour un loss particulier (Breiman et al. 1998/1999)
- Généralisation de Adaboost au Gradient Boosting pour toute une variété de fonction de loss (Friedman et al. 2000, 2001)

Boosting: compromis

Avantages

- Un paramètre: nombre d'étapes
- Pas trop de sur-apprentissage
- Applicable à plein de classifieurs faibles
- Garanties théoriques

Inconvénients

- Pas adapté avec peu de données
- Pas adapté à des classifieurs trop stables
- Pas adapté à des classifieurs trop fort: risque de sur-apprentissage

Conclusion

Autres méthodes d'ensemble

- Classifieurs en cascade
- Hiérarchies d'experts
- ...

Sources

- *A short introduction to boosting* - Yoav Freund and Robert E. Schapire
- *Trees, Bagging, Random Forests and Boosting* - Trevor Hastier - Standford University
- *Bias-Variance Tradeoff and Ensemble Methods* - Tom Dietterich, Rich Maclin
- Cours Antoine Cornuéjols https://www.lri.fr/~antoine/Courses/ENSTA/Tr-boosting-2013(ensta)x4.pdf
- Cours Ricco Rakotomalala