

Science des Données Apprentissage Statistique & IA

PHILIPPE BESSE & BÉATRICE LAURENT

Université de Toulouse
INSA – Dpt GMM
Institut de Mathématiques – ESP
UMR CNRS 5219

Questions

Apprentissage Statistique ⊂ *Machine Learning* ⊂ IA

- Facteurs de risque épidémiologiques
- Catégorisation de textes
- Reconnaissance d'une activité humaine
- Adaptation statistique en prévision météo
- Score d'appétence ou d'attrition en GRC
- Système de recommandation de ventes en ligne
- Détection de défaillance, fraude, intrusion (anomalies)
- ...
- Estimer un modèle, apprendre un algorithme, prévoir

De la Statistique à l'IA par la *Science des Données*

1930-70 h-Octets Statistique inférentielle

1950 Débuts de l'Intelligence Artificielle : Allan Turing

1970s kO Analyse des données et *exploratory data analysis*

1980s MO Réseaux de neurones, Statistique fonctionnelle

1990s GO *Data mining* : données pré-acquises

2000s TO Bioinformatique : $p >> n$, *Machine Learning*

2008 Science des Données

2010s PO Big Data p et n très grands

2012 *Deep Learning*

2016 Intelligence Artificielle (IA) : AlphaGo

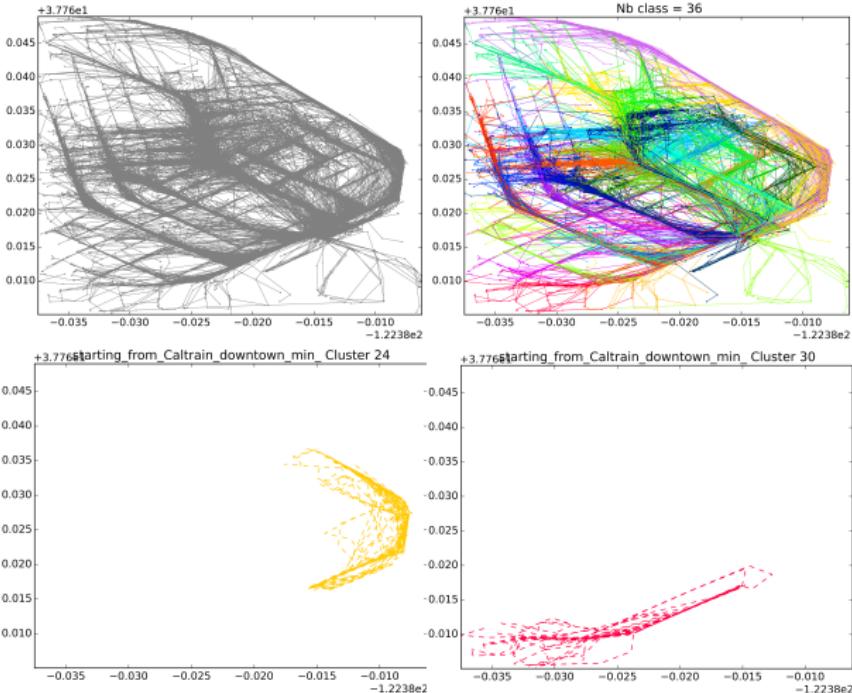
VVV...VV : Volume, Variété, Vélocité... Véracité, Valorisation

Introduction

Stratégie de la *Science des Données*
Méthodes statistiques classiques
Arbres et ensemble d'arbres

Définitions, objectifs

Environnement technologique
Cas d'usage



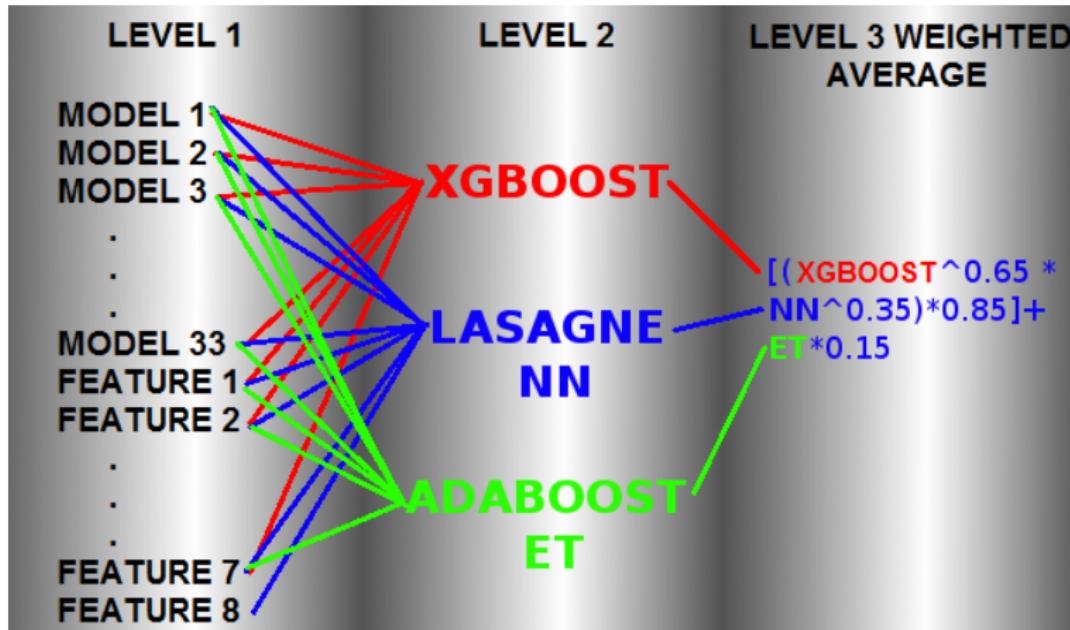
Variété : Classification de trajectoires GPS

Objectif ?

- **Explorer** : représenter, décrire, taxonomie
- **Expliquer** ou tester, prouver
- **Prévoir** et sélectionner, interpréter
- **Prévision "brute"**
- **Détection** d'anomalies

But ?

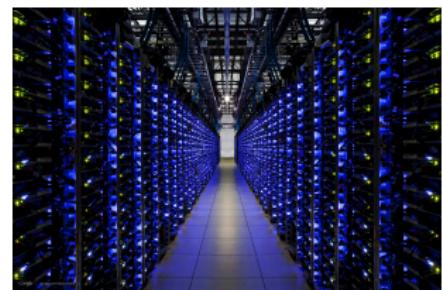
- Publication **académique** (*Benchmarks — UCI repository*)
- Solution **industrielle**
- Concours de type **Kaggle**



Concours Kaggle : Identify people who have a high degree of Psychopathy based on Twitter usage.

Réellement massives ?

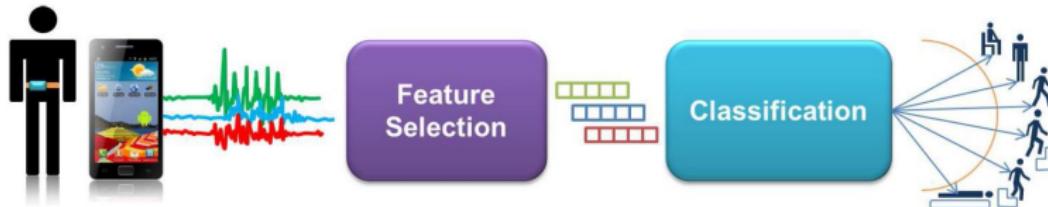
- **Seuils** technologiques (RAM, Disque)
- **Préparation** (*munging*) des données (Python)
- Données **distribuées** :
- **Hadoop, MapReduce, scalability**



Ferme de données

Quand utiliser ?

- R
- Python & Scikit-learn
- **Spark (Hadoop) & MLlib**



Human activity recognition HAR

- Données publiques de l'*UCI repository*
- 6 signaux : accéléromètre et gyroscope en x, y, z
- $p = 128$ mesures : 2.52 secondes à 50 Hz
- Objectif : Identifier l'**activité** : couché, assis, debout, marcher, monter & descendre escalier

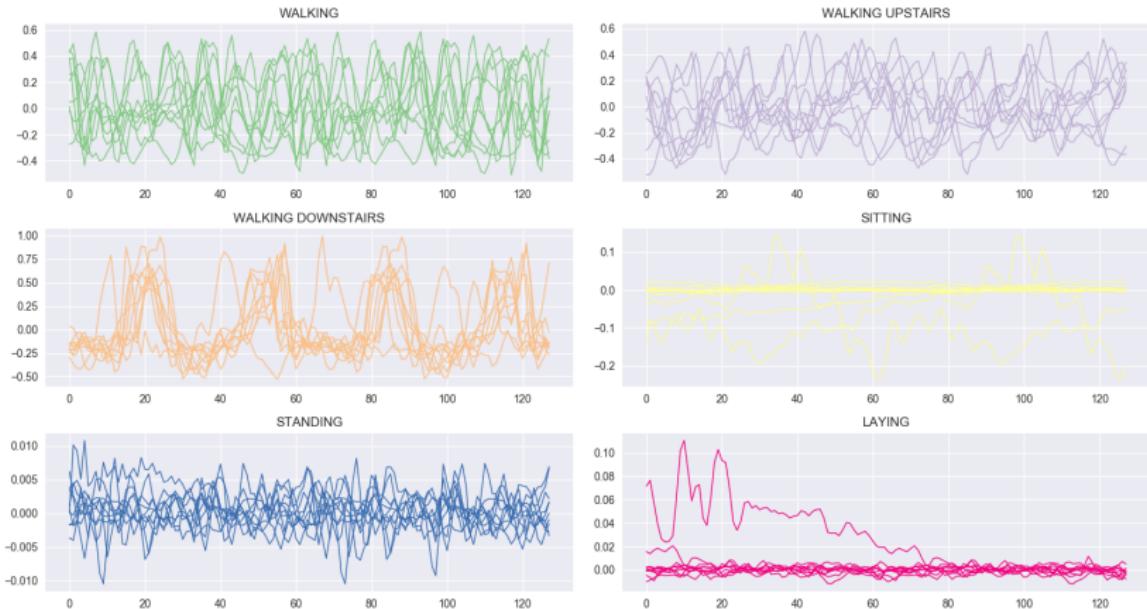
Introduction

Stratégie de la *Science des Données*
Méthodes statistiques classiques
Arbres et ensemble d'arbres

Définitions, objectifs

Environnement technologique

Cas d'usage



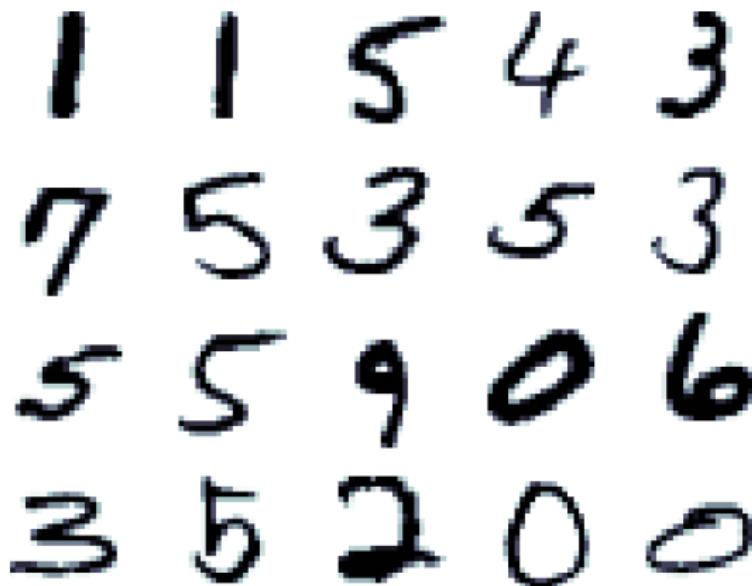
Human activity recognition : accéléromètre en y réparti par activités

HAR Phase 1 : variables "métier"

- $p = 561$ Nouvelles variables (*features*)
 - Domaine **temporel** : min, max, moyennes, variances, corrélations...
 - Domaine **fréquentiel** : plus grande, moyenne, énergies par bande...
- Base d'apprentissage : $n = 10300$

HAR ... à suivre

- Phase 2 : signaux bruts et *deep learning*
- Phase 3 : reconnaissance *en ligne*



MNIST : quelques exemples d'images de caractères

MNIST : État de l'art

- Site de Yann le Cun
- 60 000 caractères, $28 \times 28 = 784$ pixels
- Test : 10 000 images
- Méthodes **classiques** (k -nn, RF)
- Prétraitement de normalisation des images
- Complétion des données
- Distance spécifique (tangentielle) avec propriétés d'invariance
- Apprentissage Profond : *TensorFlow, Keras*

Cdiscount : données textuelles

- Données **publiques** du concours [datascience.net](#), [kaggle.com](#)
- **15M** de produits (3.5 Go), 3 niveaux : 5789 classes
- Classes **déséquilibrées**
- **Pyramide** (Python) de régressions logistiques
- Simplifier : prévoir le **1er niveau** : 47 classes
- Comparaison de Python `Scikit-learn` **vs.** `Spark`
- **Trois phases** : Nettoyage, Vectorisation, Apprentissage

Préparation des données ou *data munging*

- Extraction, nettoyage
- Statistiques élémentaires univariées, bivariées
- Valeurs atypiques & incohérences
- Transformation
- Données manquantes
- Nouvelles variables ou caractéristiques (*features*)

Natural Language Processing (NLP)

- Préparation ou *data munging*
 - Nettoyages (ponctuation, erreurs de code, casse...)
 - Suppression des mots "vides" (*stopwords*)
 - Racinisation (*stemming*) : $\text{card}(\text{Dictionnaire}) = N$
- Vectorisation de (grosses ?) données
 - Hashage (*hashing trick*) : $n_hash < N$
 $i = h(j) \quad h : \{1, \dots, N\} \longmapsto \{1, \dots, n_hash\}$
 - Xgram : *h* appliquée à un mot ou couple (bigram) ou...
 - TF-IDF
 - Word2vec

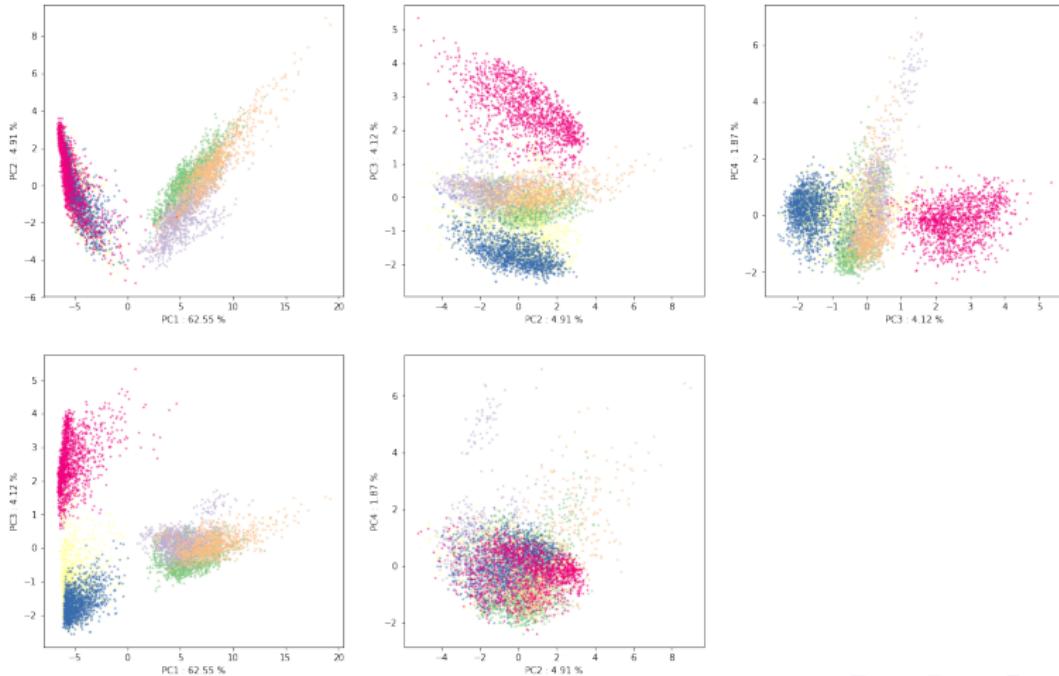
NLP : TF-IDF

- Matrice creuse : D : nombre de documents $\times N$ ou n_hash
- Importance relative de chaque mot (ou xgram) dans un document par rapport à l'ensemble des documents.
- $TF(m, d)$: nombre d'occurrences du mot m dans le document d
- $f(m)$: nombre de documents contenant le mot m
- $IDF(m) = \log \frac{D+1}{f(m)+1}$ (version smooth)
- TF-IDF : nouvelles variables ou *features* par pondération des effectifs conjoints :
 $V_m(d) = TF(m, d) \times IDF(m)$

Exploration multidimensionnelle des données

- Réduction de dimension, représentations
 - Analyse en composantes principales
 - Analyse factorielle discriminante
 - Analyse des correspondances
 - *Multidimensional scaling*
 - t-SNE (*t-Distributed Stochastic Neighbor Embedding*)
- Classification non supervisée (*clustering*)
 - Classification ascendante hiérarchique
 - Réallocation dynamique : k -means, PAM
 - DBSCAN, SOM...

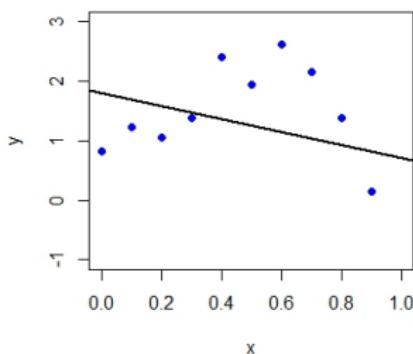
HAR : Analyse en composantes principales sur les variables "métier"



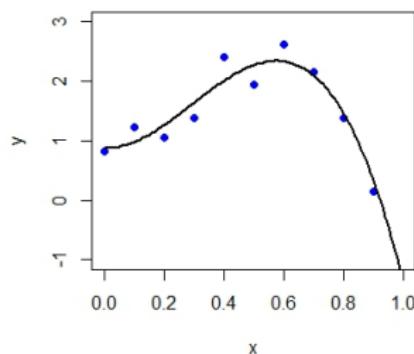
Objectif de l'apprentissage supervisé

- **Données** : $p + 1$ variables $Y, X_{j=1,p}^j$ sur n individus
- Apprendre ou estimer : $Y = f(X)$
- **Minimiser** risque ou erreur de **prévision**
 - Y quantitative (régression) : erreur quadratique moyenne
 - Y qualitative (discrimination) : nb de mal classés
- **Attention**
Distinguer erreur d'**ajustement** et erreur de **prévision**
- **Minimiser l'erreur de prévision**
 - Optimiser la **complexité** du modèle (parcimonie)
 - **Meilleur compromis** Biais – Variance

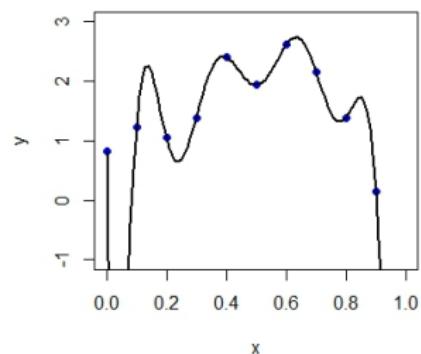
Modèle linéaire; R²=0.11



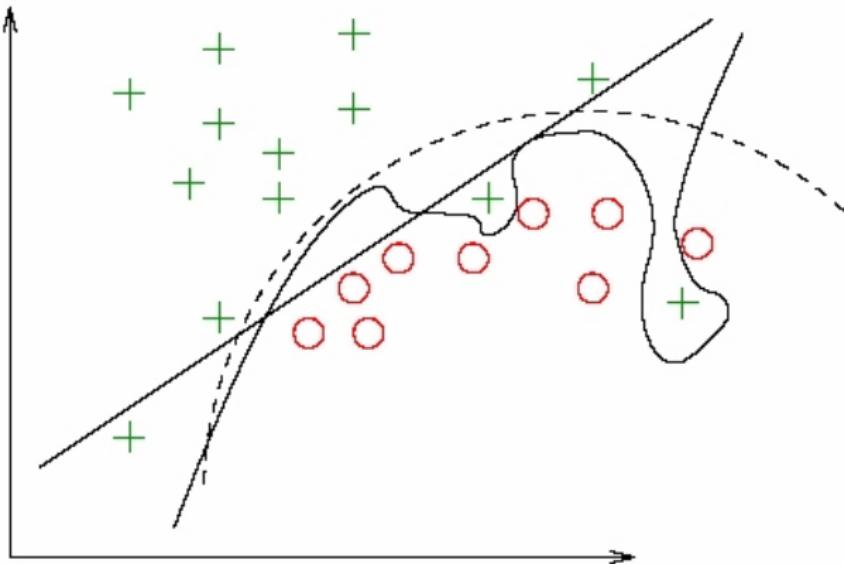
Modèle cubique; R²=0.95



Degré 10; R²=1



Sur-apprentissage en régression polynomiale



Sur-apprentissage en classification supervisée

Méthodes et algorithmes d'apprentissage

- Modèle linéaire avec sélection ou régularisation
- Régression PLS avec pénalisation
- Modèle linéaire binomial (logistique) avec sélection ou régularisation
- Analyse discriminante, k plus proches voisins
- Arbres binaires de décision (CART)
- Réseaux de neurones, perceptron, apprentissage profond
- Agrégation de modèles : *random forest, boosting...*
- SVM ou séparateurs à vaste marge
- ...
- Imputation de données manquantes
- Détection d'anomalies ou d'atypiques



Stratégie de l'Apprentissage

- ➊ Partition aléatoire de l'échantillon : apprentissage, (validation), test
- ➋ Pour chacune des méthodes considérées :
 - Apprentissage (estimation) fonction de θ (complexité)
 - Optimisation de θ par validation croisée (VC)
- ➌ Comparaison des méthodes : erreur de prévision sur échantillon test
- ➍ Itération éventuelle (VC Monte Carlo)
- ➎ Choix de la méthode (prévision vs. interprétabilité).
- ➏ Ré-estimation du modèle, exploitation

Possible : combinaison de modèles

Estimer sans biais une erreur de prévision

- Partager l'échantillon
- Pénalisation de l'erreur empirique (Cp, AIC, BIC) dans les modèles statistiques
- Simulation
 - Validation croisée *Monte Carlo*
 - *V-fold cross validation*
 - Échantillons *Bootstrap*

Notations

- D_n observations d'un n -échantillon
 $D_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ de loi conjointe inconnue P sur $\mathcal{X} \times \mathcal{Y}$
- x observation de la variable X multidimensionnelle
- D_n est appelé **échantillon d'apprentissage**
- D_n est supposé indépendant de (X, Y)
- Une **règle de prévision** (ou prédicteur) est une fonction
 $f : \mathcal{X} \rightarrow \mathcal{Y}, x \mapsto f(x)$
- Une fonction $l : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ est une **fonction de perte** si
 $l(y, y) = 0$ et $l(y, y') > 0$ pour $y \neq y'$
- Si f est une règle de prévision, $l(y, f(x))$ mesure la perte de f en x

Définitions

- **Régression réelle** : pertes \mathbb{L}^p ($p \geq 1$) : $l(y, y') = |y - y'|^p$
perte absolue si $p = 1$, perte quadratique si $p = 2$
- **Discrimination binaire** : $l(y, y') = \mathbf{1}_{y \neq y'} = \frac{|y-y'|}{2} = \frac{(y-y')^2}{4}$
- **Risque d'une règle** f : $R_P(f) = \mathbb{E}_{(X,Y) \sim P}[l(Y, f(X))]$
- **Risque empirique** associé à \mathbf{D}_n :
$$\widehat{R}_n(f, \mathbf{D}_n) = \frac{1}{n} \sum_{i=1}^n l(Y_i, f(X_i))$$
- **Minimisation** du risque empirique sur un sous-ensemble F (un modèle) de \mathcal{F} : $\hat{f}_F(\mathbf{D}_n) \in \operatorname{argmin}_{f \in F} \widehat{R}_n(f, \mathbf{D}_n)$
- **Problème** : choix de F !
- La règle oracle est telle que : $R_P(f^*) = \inf_f R_P(f)$

Décomposition du risque empirique

$$R_P(\hat{f}_F(\mathbf{D}_n)) - R_P(f^*) =$$
$$\underbrace{\left\{ R_P(\hat{f}_F(\mathbf{D}_n)) - \inf_{f \in F} R_P(f) \right\}}_{\text{Erreur d'estimation} \quad (\text{Variance})} + \underbrace{\left\{ \inf_{f \in F} R_P(f) - R_P(f^*) \right\}}_{\text{d'approximation} \quad (\text{Biais})}$$

↗ ↘ (taille de F)

- **Plus** le modèle F est complexe ou flexible,
- plus le biais est réduit mais
- plus la partie variance risque d'augmenter
- **Enjeu** : meilleur compromis biais / variance

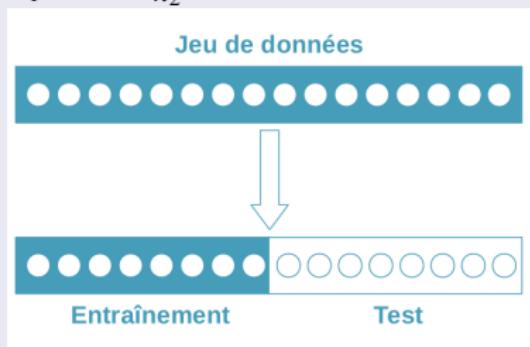
Risque empirique ou qualité d'ajustement

$$\widehat{R}_n(\widehat{f}(\mathbf{D}_n), \mathbf{D}_n) = \frac{1}{n} \sum_{i=1}^n l(y_i, \widehat{f}(\mathbf{D}_n)(\mathbf{x}_i))$$

- Minimum des moindres carrés dans le cas quantitatif
- Taux de mal classés dans le cas qualitatif
- Estimation biaisée, par optimisme

Estimation sans biais sur un échantillon indépendant

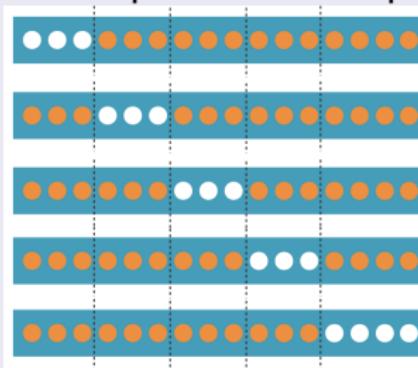
- Partition : $D_n = D_{n_1}^{\text{Appr}} \cup D_{n_2}^{\text{Test}}$ avec $n = n_1 + n_2$



- $\widehat{R}_n(\widehat{f}(D_{n_1}^{\text{Appr}}), D_{n_1}^{\text{Appr}})$ pour estimer un modèle choisi $\widehat{f}(D_{n_1}^{\text{Appr}})$
- $\widehat{R}_n(\widehat{f}, D_{n_2}^{\text{Test}})$ pour comparer les meilleurs modèles

V-fold cross validation

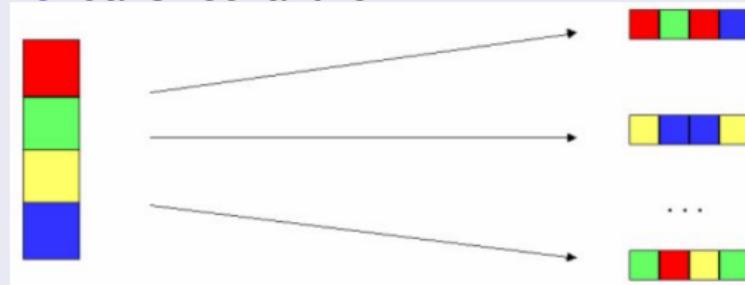
- Permutation aléatoire et séparation en V parts



- V estimations du modèle et de l'erreur
- Moyenne des V erreurs : $\widehat{R}_{\text{CV}} = \frac{1}{n} \sum_{i=1}^n l(y_i, \widehat{f}^{(-\tau(i))}(\mathbf{x}_i))$
- Choix de V : n (variance), petit (biais), 10 par défaut
- Choix de modèle : $\widehat{\theta} = \arg \min_{\theta} \widehat{R}_{\text{CV}}(\theta)$

Échantillons *Bootstrap* ou ré-échantillonnage

- Échantillonner dans l'échantillon



- B fois n tirages avec remise
- $\widehat{R}_{\text{Boot}} = \frac{1}{B} \sum_{b=1}^B \frac{1}{n} \sum_{i=1}^n l(y_i, f_{z^{*b}}(\mathbf{x}_i))$
- Erreur *out of bag* :
$$\widehat{R}_{\text{oob}} = \frac{1}{n} \sum_{i=1}^n \frac{1}{B_i} \sum_{b \in K_i} l(y_i, f_{z^{*b}}(\mathbf{x}_i))$$

Objectifs

- Expliquer Y quantitative avec X^1, \dots, X^p
- Modèle gaussien et linéaire général
- Choix de modèle par sélection de variables
- Choix de modèle par pénalisation (*ridge*, Lasso)

Hypothèses du Modèle linéaire

- Échantillon taille n : $(x_i^1, \dots, x_i^p, y_i); i = 1, \dots, n$

$$Y_i = \beta_0 + \beta_1 X_i^1 + \beta_2 X_i^2 + \dots + \beta_p X_i^p + \varepsilon_i; i = 1, \dots, n$$

- Hypothèses

- $E(\varepsilon_i) = 0, \text{Var}(\varepsilon) = \sigma^2 \mathbf{I}$
- X^j déterministes ou bien ε indépendant des X^j
- β_0, \dots, β_p constants
- Option $\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$

Expression matricielle

- $E(\varepsilon_i) = 0, \text{Var}(\varepsilon) = \sigma^2 \mathbf{I}$
- $\mathbf{X}(n \times (p + 1))$ de terme général x_i^j avec $\mathbf{x}^0 = \mathbf{1}$
- \mathbf{Y} de terme général y_i
- $\varepsilon = [\varepsilon_1 \cdots \varepsilon_p]'$
- $\beta = [\beta_0 \beta_1 \cdots \beta_p]'$

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon$$

Estimateur des moindres carrés

$$\min_{\beta \in \mathbb{R}^{p+1}} \|\mathbf{Y} - \mathbf{X}\beta\|^2$$

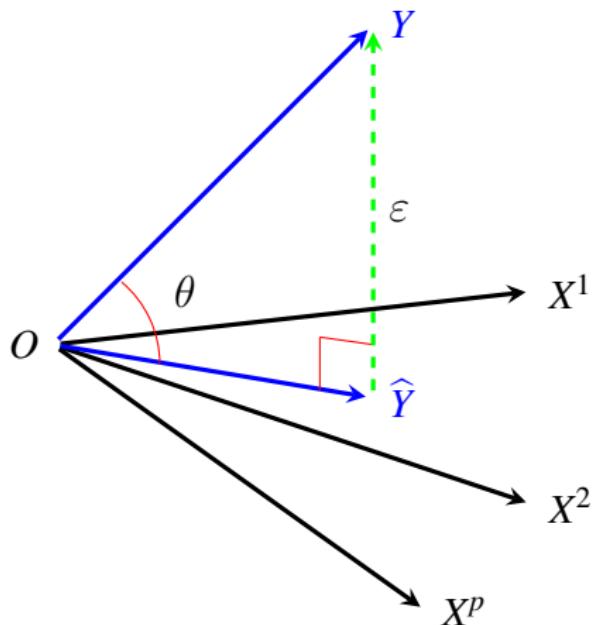
- Equations normales : $\mathbf{X}'\mathbf{Y} - \mathbf{X}'\mathbf{X}\beta = 0$
- et si $\mathbf{X}'\mathbf{X}$ inversible
- Estimation de β : $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$
- Prédiction de \mathbf{Y} : $\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} = \mathbf{HY}$
- $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$: projection orthog. sur Vect(\mathbf{X})
- Résidus : $\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{X}\hat{\beta} = (\mathbf{I} - \mathbf{H})\mathbf{Y}$

Covariances des estimateurs

$$\begin{aligned} E[(\hat{\beta} - \beta)(\hat{\beta} - \beta)'] &= \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \\ E[(\hat{\mathbf{Y}} - \mathbf{X}\beta)(\hat{\mathbf{Y}} - \mathbf{X}\beta)'] &= \sigma^2\mathbf{H} \\ E[\mathbf{e}\mathbf{e}'] &= \sigma^2(\mathbf{I} - \mathbf{H}) \end{aligned}$$

Estimation de σ^2

$$s^2 = \frac{\|\mathbf{Y} - \mathbf{X}\hat{\beta}\|^2}{n-p-1} = \frac{\text{SSE}}{n-p-1}$$



Projection \hat{Y} de Y sur l'espace vectoriel $\text{Vect}\{\mathbf{1}, X^1, \dots, X^p\}$

Inférence sur la Prévision

Pour \mathbf{x}_0 :

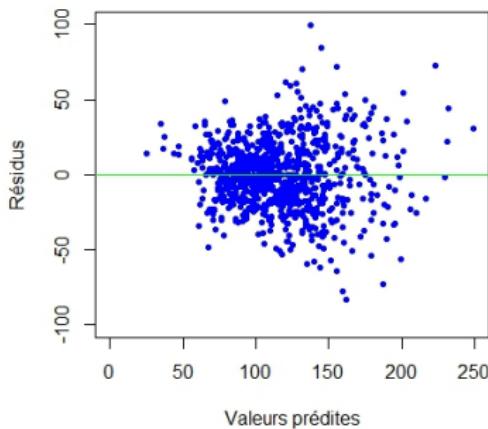
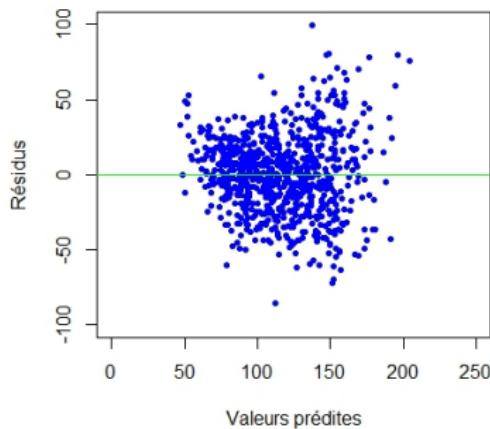
$$\hat{y}_0 = b_0 + b_1 x_0^1 + \cdots + b_p x_0^p.$$

Intervalles de confiance des prévisions de Y et $E(Y)$

$$\hat{y}_0 \pm t_{\alpha/2; (n-p-1)} s(1 + \mathbf{v}'_0 (\mathbf{X}' \mathbf{X})^{-1} \mathbf{v}_0)^{1/2}$$

$$\hat{y}_0 \pm t_{\alpha/2; (n-p-1)} s(\mathbf{v}'_0 (\mathbf{X}' \mathbf{X})^{-1} \mathbf{v}_0)^{1/2}$$

avec $\mathbf{v}_0 = (1 | \mathbf{x}'_0)' \in \mathbb{R}^{p+1}$



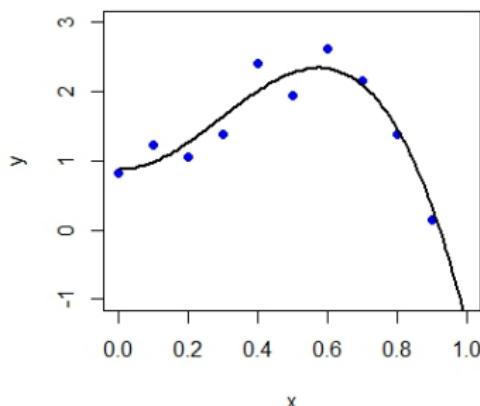
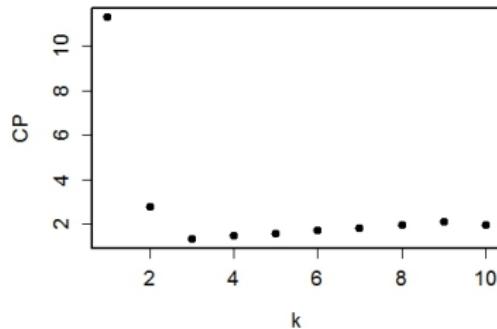
Ozone : Résidus des modèles linéaire et quadratique.

Critères de sélection de variables

- Tous les critères sont équivalents avec q fixé
- Problème : optimisé le choix de q
- C_p de Mallow $\text{MSE}(\hat{y}_i) = \text{Var}(\hat{y}_i) + [\text{Biais}(\hat{y}_i)]^2$
On suppose le modèle complet sans biais

$$C_j = (n - j - 1) \frac{\text{MSE}_j}{\text{MSE}} - [n - 2(j + 1)]$$

- $C_p = \widehat{R}_n(\widehat{f}(\mathbf{d}^n), \mathbf{d}^n) + 2\frac{d}{n}\widehat{\sigma}^2$
- $\text{AIC} = -2\mathcal{L} + 2\frac{d}{n}$
- $\text{BIC} = -2\mathcal{L} + \log(n)\frac{d}{n}$



Régression polynomiale : minimisation du C_p de Mallows.

Sélection de modèle par sélection de variables

Rechercher dans le graphe des 2^p modèles possibles

- Sélection ascendante (*forward*)
- Élimination descendante (*backward*)
- Mixte (pas à pas ou *step wise*)
- Modèle linéaire général
 - AIC mais pas le C_p
 - Interactions et effets principaux

Sélection de modèle par régularisation *ridge*

$$\tilde{\boldsymbol{\beta}} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \cdot \\ \cdot \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \beta_p \end{pmatrix}$$

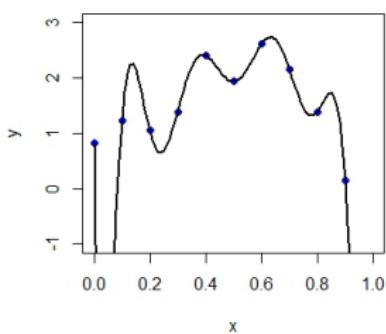
- $X^0 = (1, 1, \dots, 1)'$, et \mathbf{X} la matrice $\tilde{\mathbf{X}}$ privée de X^0
- \mathbf{Y} et $bm\mathbf{X}$ sont **centrés**
- $\mathbf{Y} = \mathbf{X}\tilde{\boldsymbol{\beta}} + \epsilon$
- $\hat{\boldsymbol{\beta}}_{\text{Ridge}} = \operatorname{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^n (Y_i - \sum_{j=0}^p X_i^{(j)} \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2$
- λ paramètre positif à optimiser
- $\hat{\boldsymbol{\beta}}_{\text{Ridge}} = (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}'\mathbf{Y}$



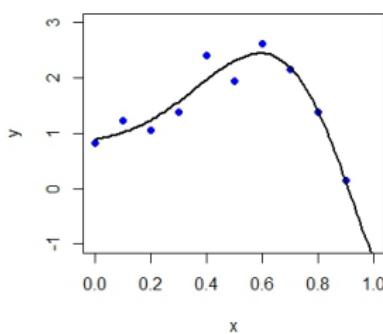
Propriétés de la régression ridge

- ① $\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p$ est inversible, mieux conditionnée
- ② β_0 n'intervient pas : centrer \mathbf{X}
- ③ Dépend des unités : réduire \mathbf{X}
- ④ Forme équivalente :
$$\hat{\beta}_{\text{Ridge}} = \arg \min_{\beta} \left\{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 ; \|\beta\|^2 < c \right\}$$
- ⑤ Chemin de régularisation
- ⑥ Optimisation de λ par V -fold validation croisée

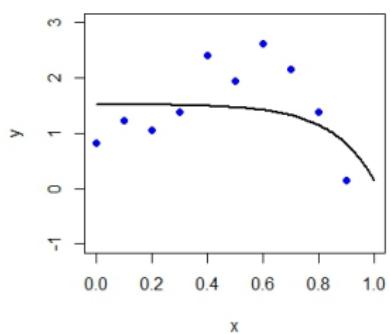
Régression Ridge, $\lambda=0$



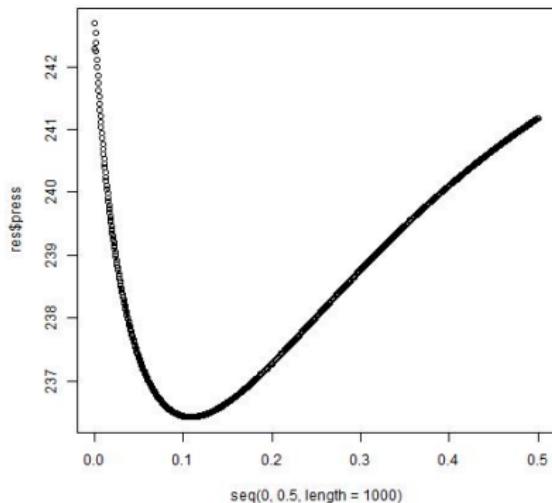
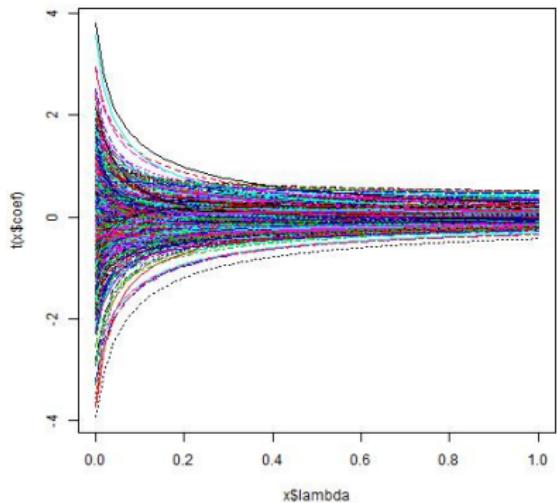
Régression Ridge, $\lambda=1.e-4$



Régression Ridge, $\lambda=100$



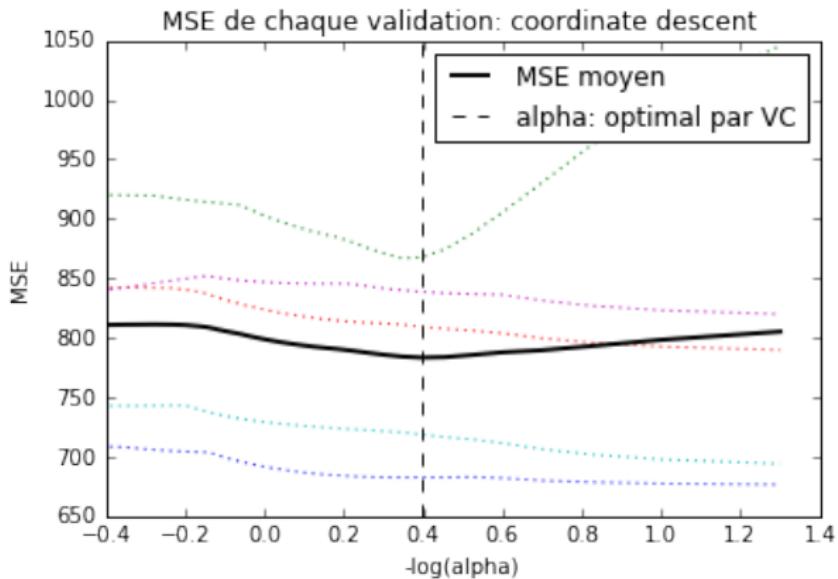
Optimisation (validation croisée) de la régression ridge polynomiale



Cookies : Régression avec pénalisation ridge de données NIR.

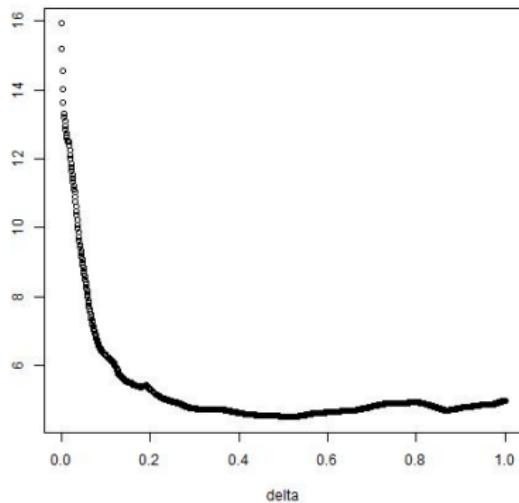
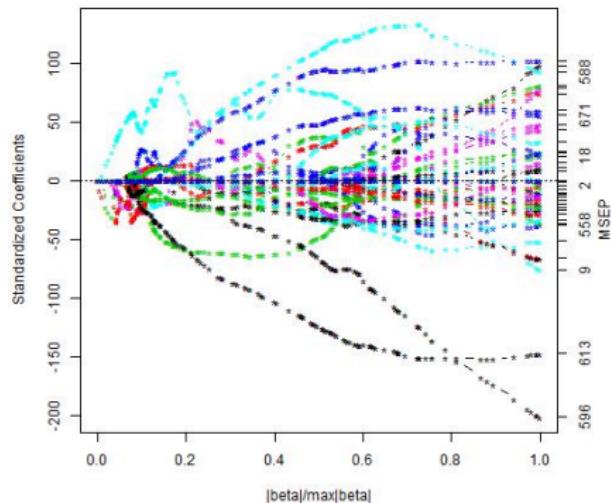
Sélection de modèle par pénalisation LASSO

- Ridge toujours calculable mais problème d'interprétation
- Objectif : associer pénalisation et sélection
- $\hat{\beta}_{\text{Lasso}} = \operatorname{argmin}_{\beta \in \mathbb{R}^p} \left\{ \sum_{i=1}^n (Y_i - \sum_{j=0}^p X_i^{(j)} \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}$
- $\hat{\beta}_{\text{Lasso}} = \operatorname{argmin}_{\beta, \|\beta\|_1 \leq t} (\|\mathbf{Y} - \mathbf{X}\beta\|^2)$
- λ est le paramètre de pénalisation
 - $\lambda = 0$: estimateur des moindres carrés.
 - λ tend vers l'infini, $\hat{\beta}_j = 0, j = 1, \dots, p$.
- $\beta_j = \operatorname{signe}(\hat{\beta}_j) (|\hat{\beta}_j| - \lambda) \mathbf{1}_{|\hat{\beta}_j| \geq \lambda}$



Ozone : optimisation de régularisation lasso par validation croisée.

LASSO



Cookies : Régression Lasso de données NIR.

Sélection *elastic net*

$$\sum_{i=1}^n (Y_i - \beta_0 - \beta_1 X_i^{(1)} - \beta_2 X_i^{(2)} - \dots - \beta_p X_i^{(p)})^2 + \lambda \left(\alpha \sum_{j=1}^p |\beta_j| + (1 - \alpha) \sum_{j=1}^p \beta_j^2 \right)$$

- Pour $\alpha = 1$, régression Lasso
- Pour $\alpha = 0$, régression ridge

Objectif de la régression logistique

- Expliquer Z qualitative à 2 modalités $\{0, 1\}$ ou Y nombre de "succès" de Z par $\{X^1, \dots, X^p\}$ qualitatives et quantitatives
- Prédicteur linéaire $X\beta$ inadapté
- Cas particulier du MLG : **modèle binomial**
- Méthode sans doute la plus utilisée (médical, marketing)
- Modèle **explicable**
- Passe à l'échelle **volume**

Notations

- Z variable qualitative à 2 modalités : 1 ou 0...
- $\mathbf{X}\beta$ prend ses valeurs dans \mathbb{R}
- Modéliser $\pi = P[Z = 1]$ ou plutôt
- $g(\pi_i) = \mathbf{x}_i'\beta$ avec $g : [0, 1] \mapsto \mathbb{R}$
- g est appelée fonction lien
 - *probit* : g fonction inverse de la fonction de répartition d'une loi normale (pas explicite).
 - *log-log* : $g(\pi) = \ln[-\ln(1 - \pi)]$ (dissymétrique)
 - *logit* : $g(\pi) = \text{logit}(\pi) = \ln \frac{\pi}{1-\pi}; \quad g^{-1}(x) = \frac{e^x}{1+e^x}$
- La *régression logistique* est une modélisation linéaire du *log odd*
- Les coefficients expriment des *odds ratio*

Estimation

- Estimation \mathbf{b} de β par maximisation de la log-vraisemblance
- Méthodes numériques itératives (Newton Raphson, Scores de Fisher)
- Prévisions des probabilités π_i : $\hat{\pi}_i = \frac{e^{\mathbf{x}'_i \mathbf{b}}}{1+e^{\mathbf{x}'_i \mathbf{b}}}$
- et des effectifs $\hat{y}_i = n_i \hat{p}_i$
- Sélection de modèle : Ridge ou Lasso
- Cas multi-classe : par défaut une contre les autres

Erreur de prévision et matrice de confusion

Prévision : Si $\hat{\pi}_i > s$, $\hat{y}_i = 1$ sinon $\hat{y}_i = 0$

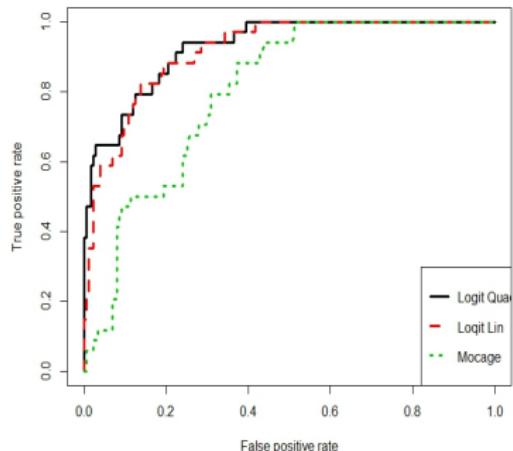
Prévision	Observation		Total
	$Y = 1$	$Y = 0$	
$\hat{y}_i = 1$	$n_{11}(s)$	$n_{10}(s)$	$n_{1+}(s)$
$\hat{y}_i = 0$	$n_{01}(s)$	$n_{00}(s)$	$n_{0+}(s)$
Total	n_{+1}	n_{+0}	n

- Vrais positifs les $n_{11}(s)$ bien classées ($\hat{y}_i = 1$ et $Y = 1$)
- Vrais négatifs les $n_{00}(s)$ bien classées ($\hat{y}_i = 0$ et $Y = 0$)
- Faux négatifs les $n_{01}(s)$ mal classées ($\hat{y}_i = 0$ et $Y = 1$)
- Faux positifs les $n_{10}(s)$ mal classées ($\hat{y}_i = 1$ et $Y = 0$)
- Le taux d'erreur : $t(s) = \frac{n_{01}(s)+n_{10}(s)}{n}$



- Taux de vrais positifs ou
sensibilité = $\frac{n_{11}(s)}{n_{+1}}$
- Taux de vrais négatifs ou
spécificité = $\frac{n_{00}(s)}{n_{+0}}$
- Taux de faux positifs
 $= 1 - \text{Spécificité} = \frac{n_{10}(s)}{n_{+0}}$
- AUC : aire sous la courbe
- Score de Pierce ($H - F$)
- Log loss

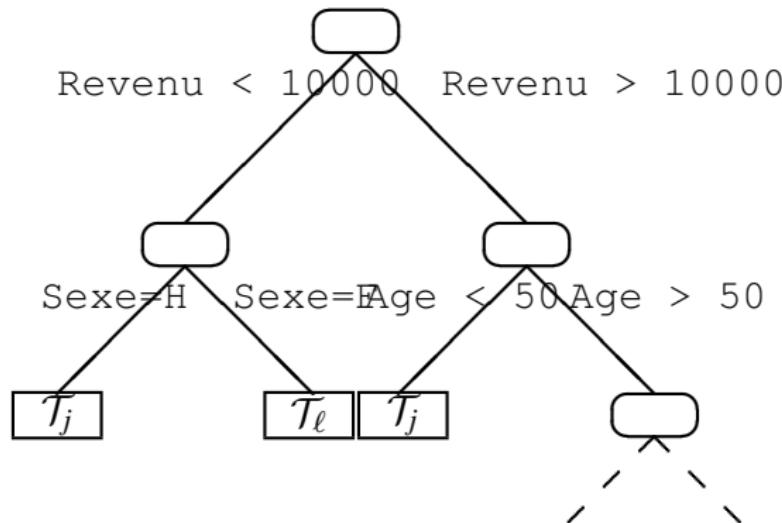
$$\text{LI} = -\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^L y_{ik} \log(\hat{\pi}_{ik})$$



Ozone : Courbes ROC et aire sous la courbe

Introduction aux arbres binaires de décision

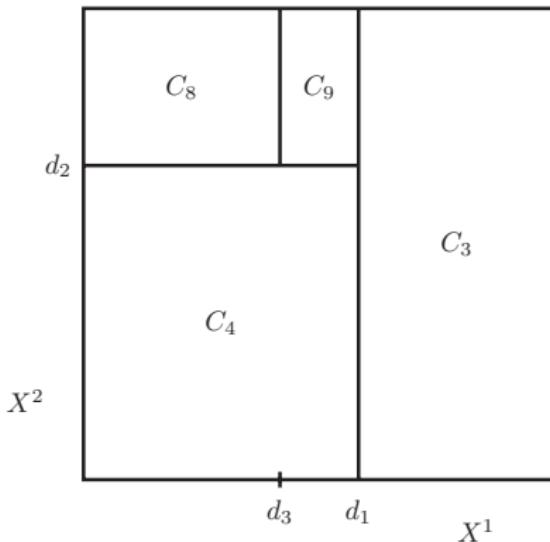
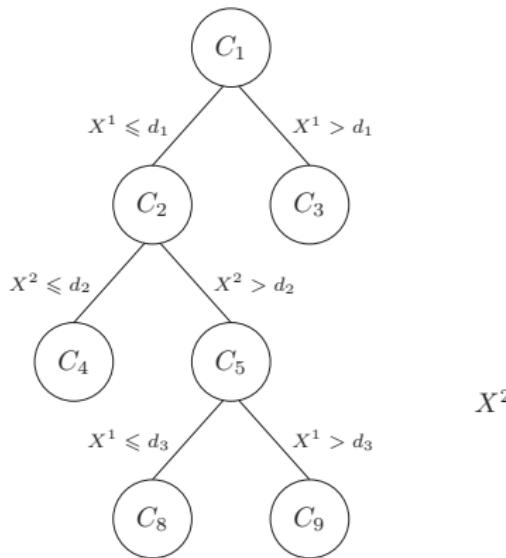
- Classification and regression trees (**CART**)
- Breiman et col. (1984)
- X^j explicatives quantitatives ou qualitatives
- Y quantitative : **regression tree**
- Y qualitative à m modalités $\{\mathcal{T}_\ell; \ell = 1 \dots, m\}$: **classification tree**
- **Objectif** : construction d'un arbre de décision binaire simple à interpréter
- Méthodes **calculatoires** : pas d'hypothèses mais des données



Exemple fictif : arbre binaire de classification

Définitions

- Déterminer une séquence **itérative** de *nœuds*
- *Racine* : nœud **initial** ou ensemble de l'échantillon
- *Feuille* : nœud **terminal**
- *Nœud* : choix d'une **variable** et d'une **division**
sous-ensemble auquel est appliquée une **dichotomie**
- *Division* : valeur seuil ou groupes des modalités



Exemple fictif : pavage dyadique de l'espace

Règles

- Choix nécessaires :
 - 1 Critère de la “meilleure” division parmi celles *admissibles*
 - 2 Règle de nœud terminal : *feuille*
 - 3 Règle d'affectation à une classe T_ℓ ou une valeur de Y
- Division *admissible* : descendants $\neq \emptyset$
- X^j réelle ou ordinaire : $(c_j - 1)$ divisions possibles
Attention effets incontrôlables si c grand
- X^j nominale : $2^{(c_j-1)} - 1$ divisions
- Fonction d'hétérogénéité D_κ d'un nœud
 - 1 Nulle : une seule modalité de Y ou Y constante
 - 2 Maximale : modalités de Y équiréparties ou grande variance

Division optimale

- Notation

- κ : numéro d'un nœud
- κ_G et κ_D les nœuds fils

- L'algorithme retient la division rendant minimales

$$D_{\kappa_G} + D_{\kappa_D}$$

- Chaque étape κ de construction de l'arbre :

$$\max_{\{Divisions \; de \; X^j; j=1,p\}} D_\kappa - (D_{(\kappa_G} + D_{\kappa_D})$$

Feuille et affectation

- Un Nœud est terminal ou feuille, si :
 - Homogène
 - Plus de partition admissible
 - Nombre d'observations inférieur à un seuil
- Affectation
 - Y quantitative, la valeur est la moyenne des observations
 - Y qualitative, chaque feuille est affectée à une classe T_ℓ de Y en considérant le mode conditionnel :
 - la classe la mieux représentée dans le nœud
 - la classe *a posteriori* la plus probable si des *a priori* sont connus
 - la classe la moins coûteuse si des coûts de mauvais classement sont donnés

Y quantitative : hétérogénéité en régression

Hétérogénéité du nœud κ :

$$D_\kappa = \frac{1}{|\kappa|} \sum_{i \in \kappa} (y_i - \bar{y}_\kappa)^2$$

où $|\kappa|$ est l'effectif du nœud κ

Minimiser la variance intra-classe

Les nœud fils κ_G et κ_D minimisent :

$$\frac{|\kappa_G|}{n} \sum_{i \in \kappa_G} (y_i - \bar{y}_{\kappa_G})^2 + \frac{|\kappa_D|}{n} \sum_{i \in \kappa_D} (y_i - \bar{y}_{\kappa_D})^2.$$

Hétérogénéité et déviance dans le cas gaussien
(Breiman et al. 1984)



Y qualitative : hétérogénéité en discrimination

Hétérogénéité du nœud κ :

- Entropie avec la notation $0 \log(0) = 0$

$$D_\kappa = -2 \sum_{\ell=1}^m |\kappa| p_\kappa^\ell \log(p_\kappa^\ell)$$

p_κ^ℓ : proportion de la classe \mathcal{T}_ℓ de Y dans κ .

- Concentration de Gini : $D_\kappa = \sum_{\ell=1}^m p_\kappa^\ell (1 - p_\kappa^\ell)$

Entropie et déviance d'un modèle multinomial (Breiman et al. 1984)

Sélection de l'arbre optimal : élagage par validation croisée

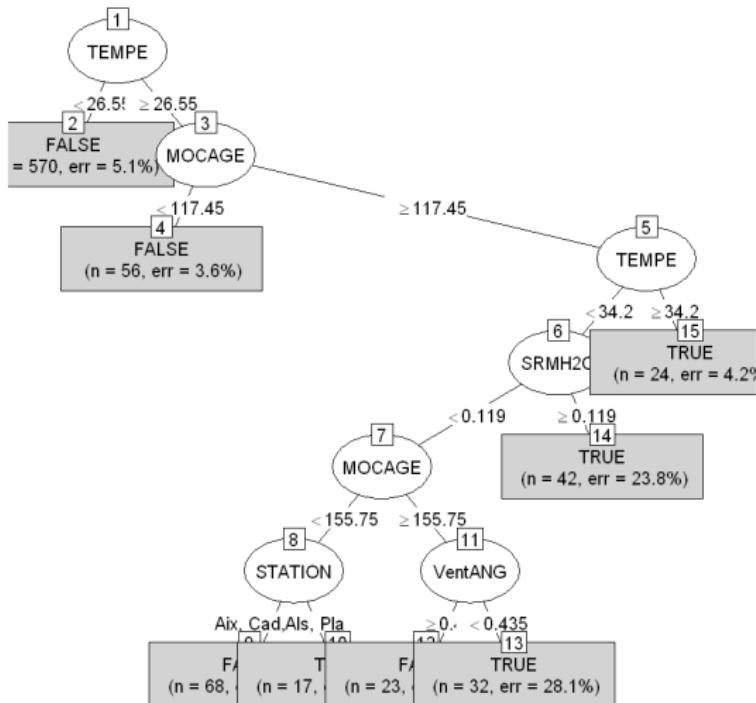
- Arbre maximal A_{\max}

$$C(A) = D(A) + \gamma \times K_A$$

- Séquence $A_K \dots A_1$ emboîtée associée à
- Séquence des valeurs γ_κ
- For $v = 1, \dots, V$
 - Estimation de la séquence d'arbres associée à γ_κ
 - Estimation de l'erreur par validation croisée
- Séquence des moyennes de ces erreurs
- γ_{Opt} optimal
- Arbre associé à γ_{Opt} dans $A_K \dots A_1$

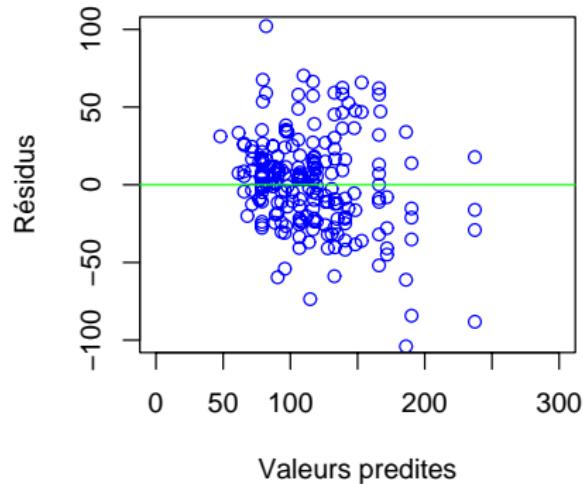
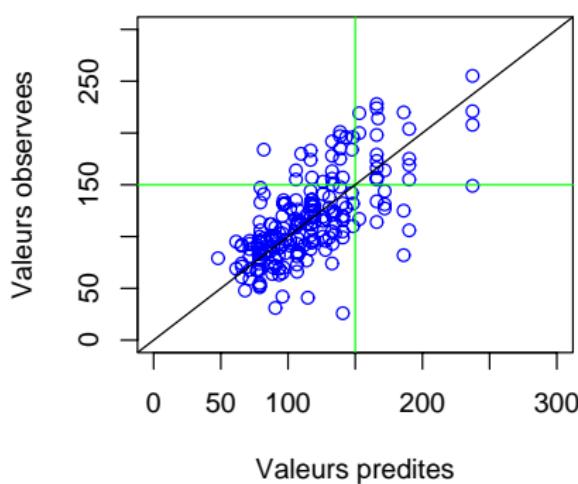
Attention : Séquences d'arbres différentes, même séquence γ_κ





Ozone : arbre de discrimination élagué par validation croisée





Ozone : observations et résidus en fonction des prévisions

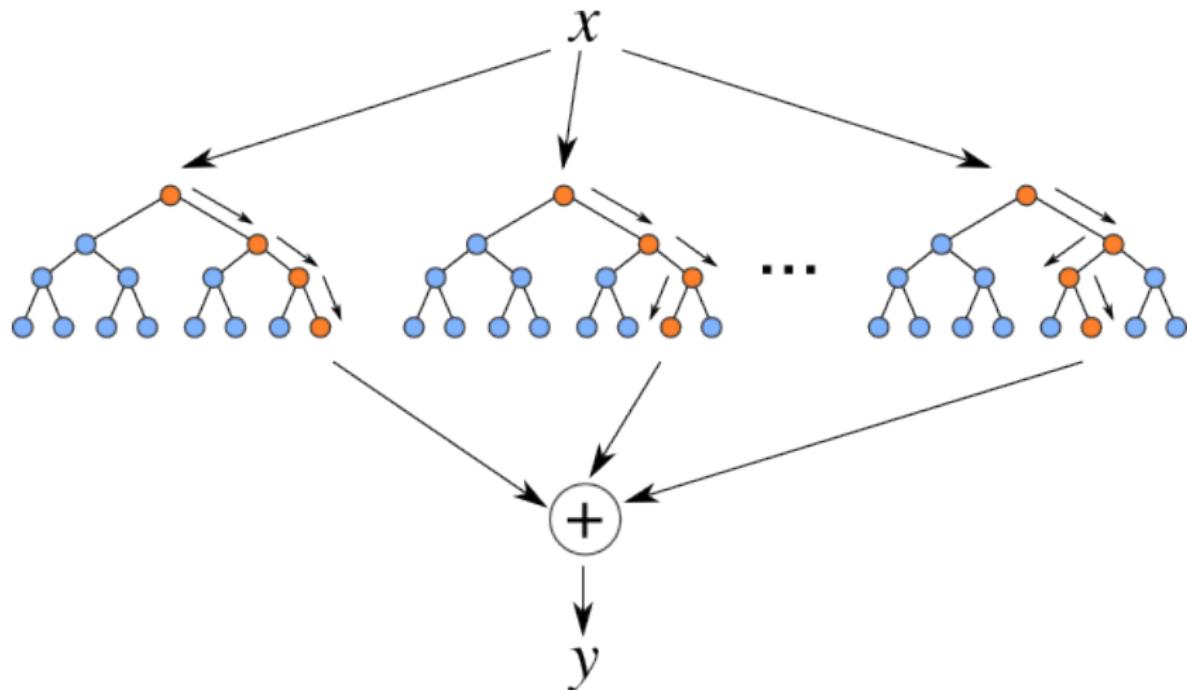
Ensemble d'arbres : introduction historique

- Stratégies adaptatives (**boosting**) ou aléatoires (**bagging**)
- Combinaison ou **agrégation** de modèles (presque) sans **sur-ajustement**
- Apprentissage machine (**machine learning**) et Statistique
- Comparatifs heuristiques et propriétés théoriques
- **Bagging** pour **bootstrap aggregating** (Breiman, 1996)
- Forêts aléatoires (**random forests**) (Breiman, 2001)
- Du **Boosting** (Freund et Shapiro, 1996) déterministe et adaptatif à l'**extrem gradient boosting**
- Toute méthode de modélisation **non linéaire**
- Méthodes **efficaces** : Fernandez-Delgado et al. (2014), *Kaggle*



Bootstrap aggregating : principe

- Soit Y une variable à expliquer quantitative ou qualitative
- X^1, \dots, X^p les variables explicatives
- $f(\mathbf{x})$ un modèle fonction de $\mathbf{x} = \{x^1, \dots, x^p\} \in \mathbb{R}^p$
- $\mathbf{z} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ échantillon de loi F et de taille n
 - $f(\cdot) = E_F(\hat{f}_{\mathbf{z}})$ estimateur sans biais de variance nulle
 - B échantillons indépendants $\{\mathbf{z}_b\}_{b=1,B}$
 - Y quantitative : $\hat{f}_B(\cdot) = \frac{1}{B} \sum_{b=1}^B \hat{f}_{\mathbf{z}_b}(\cdot)$ (moyenne)
 - Y qualitative : $\hat{f}_B(\cdot) = \arg \max_j \text{card} \left\{ b \mid \hat{f}_{\mathbf{z}_b}(\cdot) = j \right\}$ (vote)
- **Principe** : Moyenner des prévisions indépendantes pour réduire la variance
- B échantillons indépendants remplacés par B réplications bootstrap

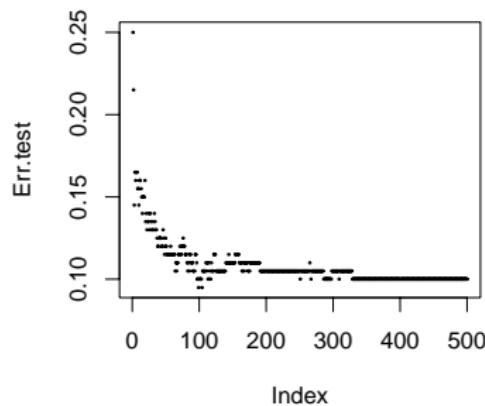
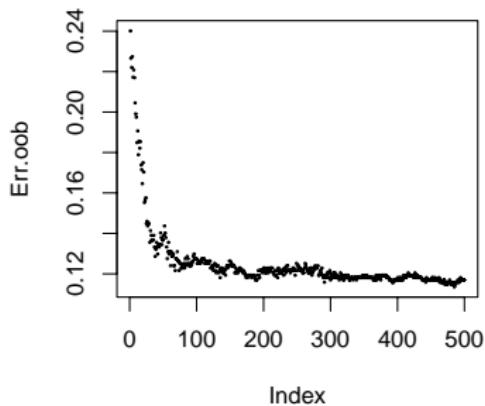


Forêts aléatoires : principe

- Amélioration du **bagging** d'arbres binaires
- Variance de B variables corrélées : $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$
- Ajout d'une **randomisation** pour rendre les arbres plus indépendants
- Choix **aléatoire** des variables
- Intérêt : grande dimension

Forêts aléatoires : algorithme

- Soit \mathbf{x}_0 à prévoir et $\mathbf{z} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ un échantillon pour $b = 1$ à B
 - Tirer un échantillon bootstrap \mathbf{z}_b^*
 - Estimer un arbre avec randomisation des variables :
 - Pour chaque nœud, tirage aléatoire de m prédicteurs
- Calculer l'estimation moyenne $\widehat{f}_B(\mathbf{x}_0) = \frac{1}{B} \sum_{b=1}^B \widehat{f}_{\mathbf{z}_b}(\mathbf{x}_0)$ ou le vote



Banque : Évolution du taux de mal classés "out-of-bag" et sur l'échantillon test en fonction du nombre d'arbres de la forêt

Forêts aléatoires : utilisation

- **Élagage** : Arbres de taille q , ou complet.
- La sélection **aléatoire** des m prédicteurs ($m = \sqrt{p}$ en classification, $\frac{p}{3}$ en régression) accroît la **variabilité**
- Chaque **modèle de base** est moins performant mais l'**agrégation** est performante
- Évaluation itérative de l'erreur **out-of-bag**

Aide à l'interprétation

- Indices d'importances
- Mean Decrease Accuracy
- Mean Decrease Gini
- Var used

Forêts aléatoires : implémentations

- R
 - randomForest : interface du programme Fortran77
 - ranger
- Weka version en java
- Scikit-learn analogue de la version originale
- Spark/MLlib]
 - arbre "scalable" du projet PLANET
 - Deux paramètres supplémentaires
 - `subsamplingRate = 1.0`
 - `maxBins=32`

Forêts aléatoires : autres utilisations

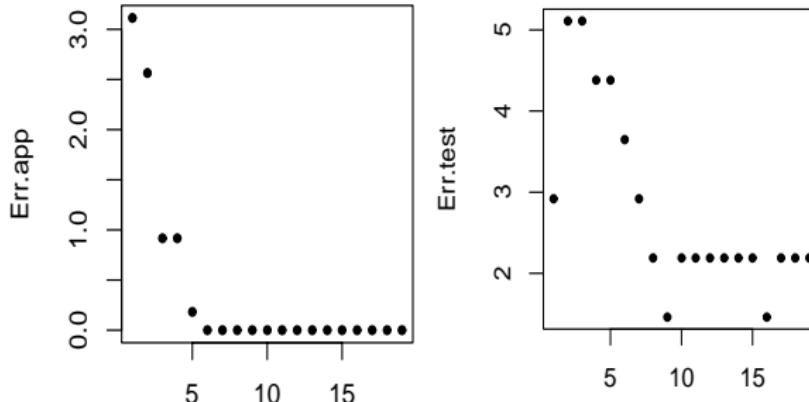
- **Proximités** ou similarités des observations
- **Atypiques** ou anomalies
- **Classification** non supervisée
- **Imputation** de données manquantes missForest
- **Durée de vie** *survival forest*
- ...

Boosting : principe

- Améliorer les compétences d'un **faible classifieur** (Schapire, 1990 ; Freund et Schapire, 1996)
- **AdaBoost (Adaptive boosting)** prévision d'une variable binaire
- Réduire la **variance** mais aussi le **biais** de prévision
- Meilleure méthode **off-the-shelf**
- Agrégation d'une famille de modèles **récurents**
Chaque modèle est une version adaptative du précédent en donnant plus de poids, lors de l'estimation suivante, aux observations mal ajustées
- **Variantes** : type de la variable à prédire (binaire, k classes, réelles), fonction perte (robustesse)

AdaBoost discret

- Fonction δ de discrimination $\{-1, 1\}$
- Soit \mathbf{x}_0 à prévoir et
- $\mathbf{z} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ un échantillon
- Initialiser les poids $\mathbf{w} = \{w_i = 1/n ; i = 1, \dots, n\}$
- pour $m = 1$ à M
 - Estimer δ_m sur l'échantillon pondéré par \mathbf{w}
 - Calculer le taux d'erreur apparent : $\widehat{\mathcal{E}}_p = \frac{\sum_{i=1}^n w_i \mathbf{1}\{\delta_m(\mathbf{x}_i) \neq y_i\}}{\sum_{i=1}^n w_i}$
 - Calculer les logit : $c_m = \log((1 - \widehat{\mathcal{E}}_p)/\widehat{\mathcal{E}}_p)$
 - Nouvelles pondérations (normalisation) :
 $w_i \leftarrow w_i \cdot \exp [c_m \mathbf{1}\{\delta_m(\mathbf{x}_i) \neq y_i\}] ; i = 1, \dots, n$
- Résultat du vote : $\widehat{f}_M(\mathbf{x}_0) = \text{signe} \left[\sum_{m=1}^M c_m \delta_m(\mathbf{x}_0) \right]$



Cancer : Evolution des taux d'erreur de l'apprentissage et du test en fonction du nombre d'arbres dans AdaBoost

Gradient boosting machine : Principe 1

- Gradient Boosting Models (Friedman, 2002-2009)
- dans le cas d'une fonction perte différentiable
- **Principe :**
 - Construire une **séquence** de modèles de sorte qu'à chaque **étape**, chaque **modèle** ajouté à la **combinaison**, apparaisse comme un **pas** vers une **meilleure solution**
 - Ce **pas** est franchi dans la direction du **gradient** de la **fonction perte** approché par un arbre de régression

GBM : Principe 2

- Modèle adaptatif précédent :

$$\widehat{f}_m(\mathbf{x}) = \widehat{f}_{m-1}(\mathbf{x}) + c_m \delta(\mathbf{x}; \gamma_m)$$

- Transformé en une descente de gradient

$$\widehat{f}_m = \widehat{f}_{m-1} - \gamma_m \sum_{i=1}^n \nabla_{f_{m-1}} l(y_i, f_{m-1}(x_i)).$$

- Recherche d'un meilleur pas de descente γ :

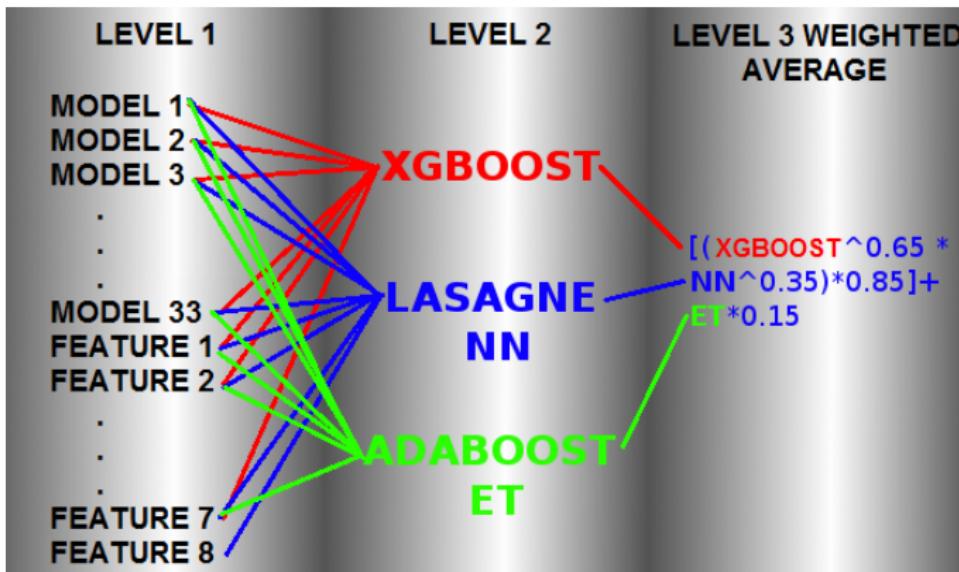
$$\min_{\gamma} \sum_{i=1}^n \left[l \left(y_i, f_{m-1}(x_i) - \gamma \frac{\partial l(y_i, f_{m-1}(x_i))}{\partial f_{m-1}(x_i)} \right) \right].$$

GBM en régression : algorithme

- Soit \mathbf{x}_0 à prévoir
- Initialiser $\hat{f}_0 = \arg \min_{\gamma} \sum_{i=1}^n l(y_i, \gamma)$
- pour $m = 1$ à M
 - Calculer $r_{mi} = - \left[\frac{\delta l(y_i, f(\mathbf{x}_i))}{\delta f(\mathbf{x}_i)} \right]_{f=f_{m-1}} ; \quad i = 1, \dots, m$
 - Ajuster un arbre de régression δ_m aux (\mathbf{x}_i, r_{mi})
 - Calculer $\gamma_{jm} = \arg \min_{\gamma} \sum_{i=1}^n l(y_i, f_{m-1}(\mathbf{x}_i) + \gamma \delta_m(\mathbf{x}_i))$
 - Mise à jour : $\hat{f}_m(\mathbf{x}) = \hat{f}_{m-1}(\mathbf{x}) + \gamma_m \delta_m(\mathbf{x})$
- Résultat : $\hat{f}_M(\mathbf{x}_0)$

Extrem Gradient Boosting : motivation

- **Algorithme** *XGBoost* de Chen et Guestrin (2016)
- **Pénalisation** supplémentaire pour contrôle du sur-apprentissage
- **Problème** : nombre de paramètres à optimiser
- **Astuces** d'implémentation pour parallélisation
- **Environnements** : R (*caret*), Python, Julia, GPU, *Amazon Web Service*, Spark...
- **Solutions** gagnantes des concours *Kaggle*



Kaggle : Identify people who have a high degree of Psychopathy based on Twitter usage

XGBoost : pénalisation

- Fonction perte L par pénalisation de l

$$\mathcal{L}(f) = \sum_{i=1}^n l(\hat{y}_i, y_i) + \sum_{m=1}^M \Omega(\delta_m)$$

$$\Omega(\delta) = \alpha|\delta| + \frac{1}{2}\beta||\mathbf{w}||^2$$

- $|\delta|$ nombre de feuilles de l'arbre δ
- \mathbf{w} vecteur des valeurs attribuées à chaque feuille
- Ω mélange de pénalisation l_1 et l_2

XGBoost : astuces

- Approximation du gradient par développement de Taylor : sommes et parallélisation
- Complexité des divisions : quantiles des distributions
- Algorithme tolérant aux **Données manquantes** : gradient calculé sur les valeurs présentes
- Indicateur d'**importance** des variables
- Gestion des **matrices creuses**

XGBoost : paramètres supplémentaires

`alpha` pénalisation de type Lasso (l_1) sur la complexité de l'arbre de régression estimant le gradient

`lambda` pénalisation de type *ridge* (l_2)

`gamma` réduction minimale de la perte pour accepter une division

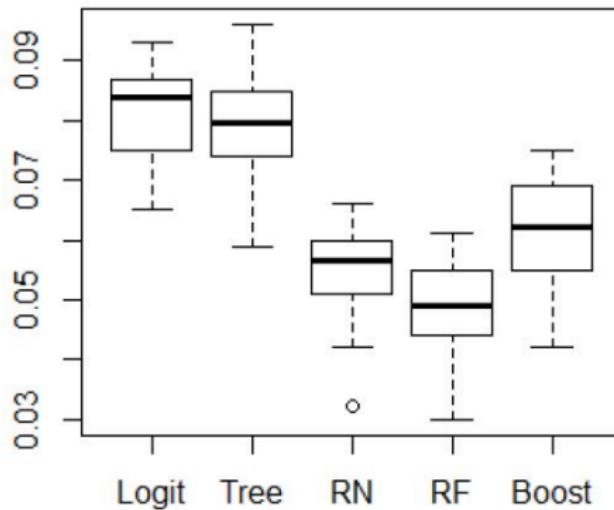
`tree_method` algorithme glouton de recherche des divisions ou simplification (quantiles ou regroupement de classes)

`sketch_eps` contrôle le nombre de classes

`scale_pos_weight` à prendre en compte pour des classes déséquilibrées

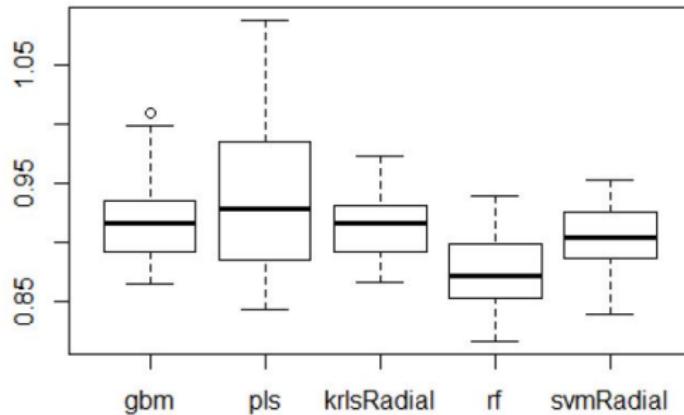
Autres paramètres d'optimisation des performances

	Logit	Tree	RN	RF	Boost
Moyenne	0.0820	0.078	0.055	0.049	0.061



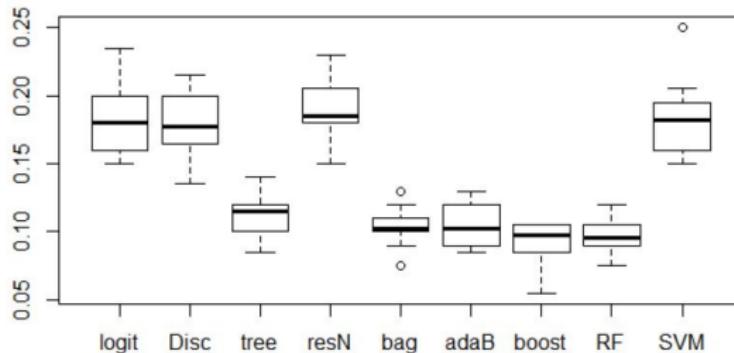
Spams : Comparaison par validation croisée Monte Carlo des erreurs de prévision de détections de pourriels

	gbm	pls	krlsRadial	rf	svmradial
Moyenne	0.92	0.94	0.92	0.87	0.90



Criblage virtuel de molécule : prévision de la capacité d'une molécule à traverser la barrière du cerveau

	logit	Disc	tree	resN	bag	adaB	boost	RF	SVM
	0.19	0.18	0.11	0.19	0.10	0.11	0.09	0.10	0.19



Marketing bancaire : Score d'appétance de la carte Visa Premier