Arbres de décisions et forêts aléatoires.

Pierre Gaillard

7 janvier 2014

Plan

Introduction

Introduction

Jeu de données (ex : n emails)

- n observations
- p variables explicatives décrivant ces observations

Variable cible à prévoir, Y.
Elle peut être continue (régression) ou discrète (classification)
ex : spam / non spam

Variables explicatives, $X_1, \ldots, X_p$.
ex : proportions de majuscules, présence de certains mots (remove,free,...), de certains caractère ($, !,...)

Idées

Arbre de décisions. Découper l'espace des variables explicatives en groupes homogènes de façon récursive et dyadique.

Forêts aléatoires. Construire aléatoirement de nombreux arbres et moyenne.

I. Les arbres de décision

Historique et références

Méthode introduite par Leo Breiman en 1984.

Il existe de nombreuses variantes : CART (Classification and regression trees), C4.5, CHAID, ID3

Package R : `tree`, `rpart`.

Références :

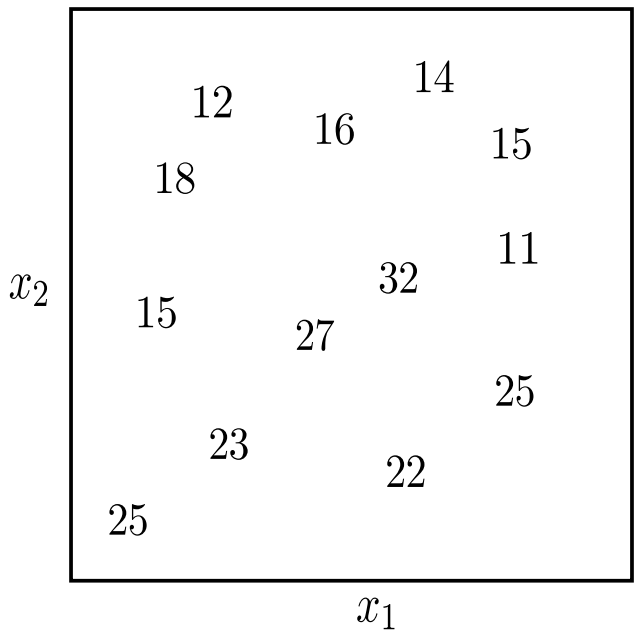
Breiman L. and al (1984) Classification and Regression Trees.

Arbre de régression

Ensemble d'entraînement $(X_t, Y_t)_{t \in S_0}$ indépendantes et identiquement distribuées.

Objectif

Expliquer Y_t en fonction des variables contextuelles $X_t = (X_{1,t}, \ldots, X_{p,t})$.

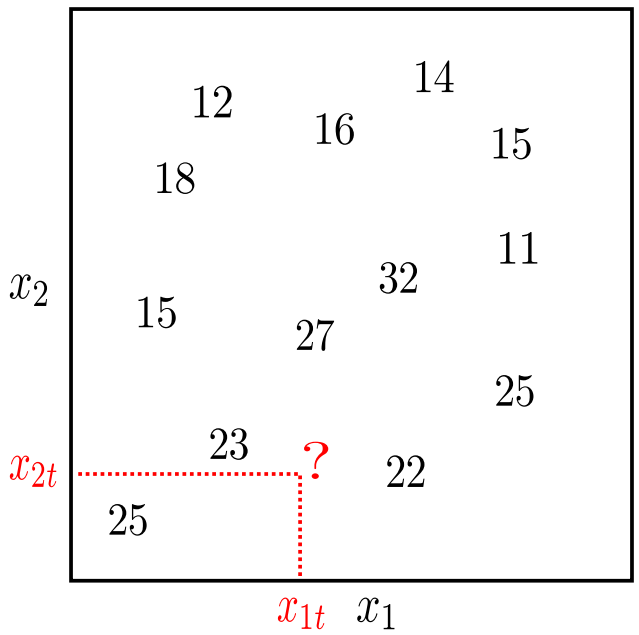


Arbre de régression

Ensemble d'entraînement $(X_t, Y_t)_{t \in S_0}$ indépendantes et identiquement distribuées.

Objectif

Expliquer Y_t en fonction des variables contextuelles $X_t = (X_{1,t}, \ldots, X_{p,t})$.

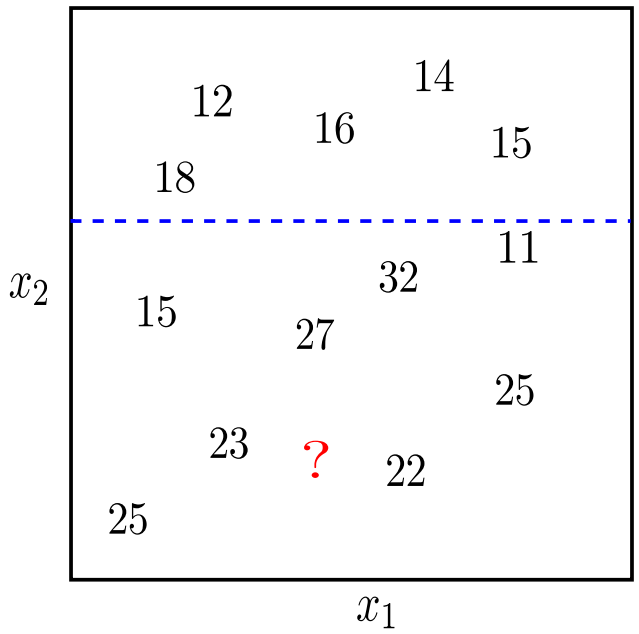


Arbre de régression

Ensemble d'entraînement $(X_t, Y_t)_{t \in S_0}$ indépendantes et identiquement distribuées.

Objectif

Expliquer Y_t en fonction des variables contextuelles $X_t = (X_{1,t}, \ldots, X_{p,t})$.

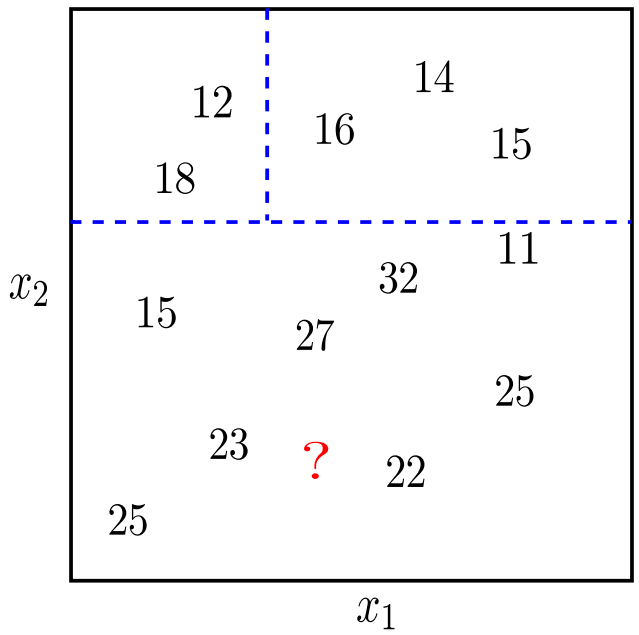


Arbre de régression

Ensemble d'entraînement $(X_t, Y_t)_{t \in S_0}$ indépendantes et identiquement distribuées.

Objectif

Expliquer Y_t en fonction des variables contextuelles $X_t = (X_{1,t}, \ldots, X_{p,t})$.

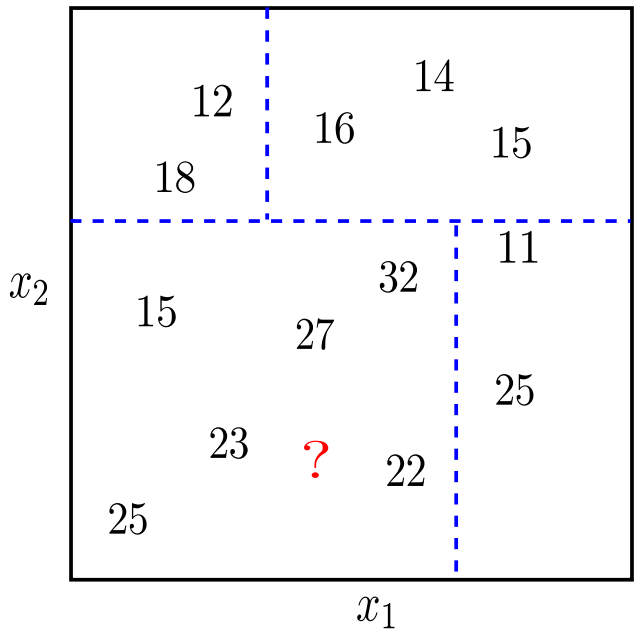


Arbre de régression

Ensemble d'entraînement $(X_t, Y_t)_{t \in S_0}$ indépendantes et identiquement distribuées.

Objectif

Expliquer Y_t en fonction des variables contextuelles $X_t = (X_{1,t}, \ldots, X_{p,t})$.

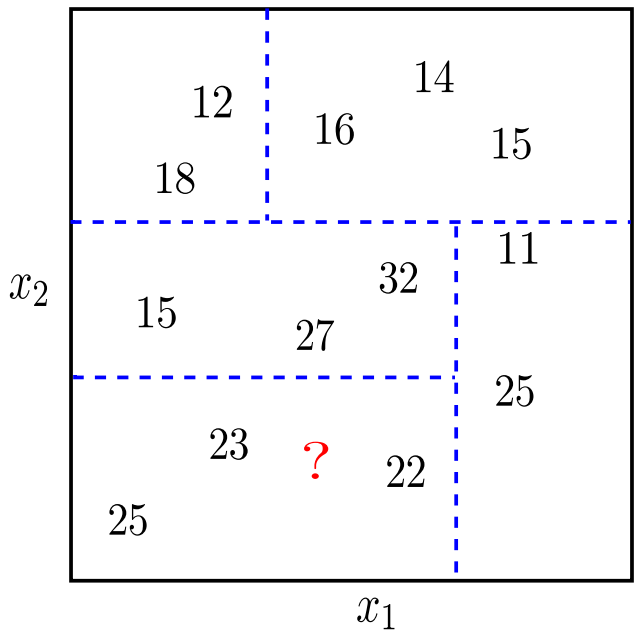


Arbre de régression

Ensemble d'entraînement $(X_t, Y_t)_{t \in S_0}$ indépendantes et identiquement distribuées.

Objectif

Expliquer Y_t en fonction des variables contextuelles $X_t = (X_{1,t}, \ldots, X_{p,t})$.

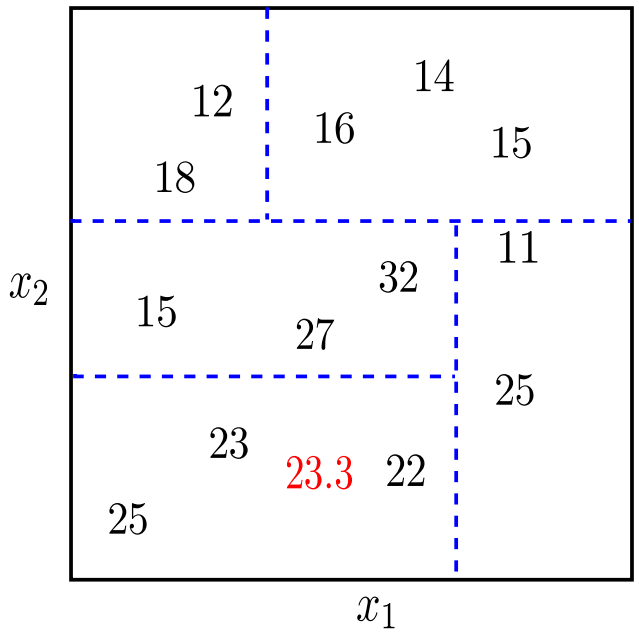


Arbre de régression

Ensemble d'entraînement $(X_t, Y_t)_{t \in S_0}$ indépendantes et identiquement distribuées.

Objectif

Expliquer Y_t en fonction des variables contextuelles $X_t = (X_{1,t}, \ldots, X_{p,t})$.



Construction de l'arbre

On découpe l'ensemble des variable explicative en deux de façon récursive :

1. Choix d'une variable explicative
2. Choix d'un seuil de coupe
3. Séparer les données en deux sous-ensemble et recommencer sur chaque sous-ensemble jusqu'à un critère d'arrêt.

Exemple numérique : prévision de la qualité de l'air

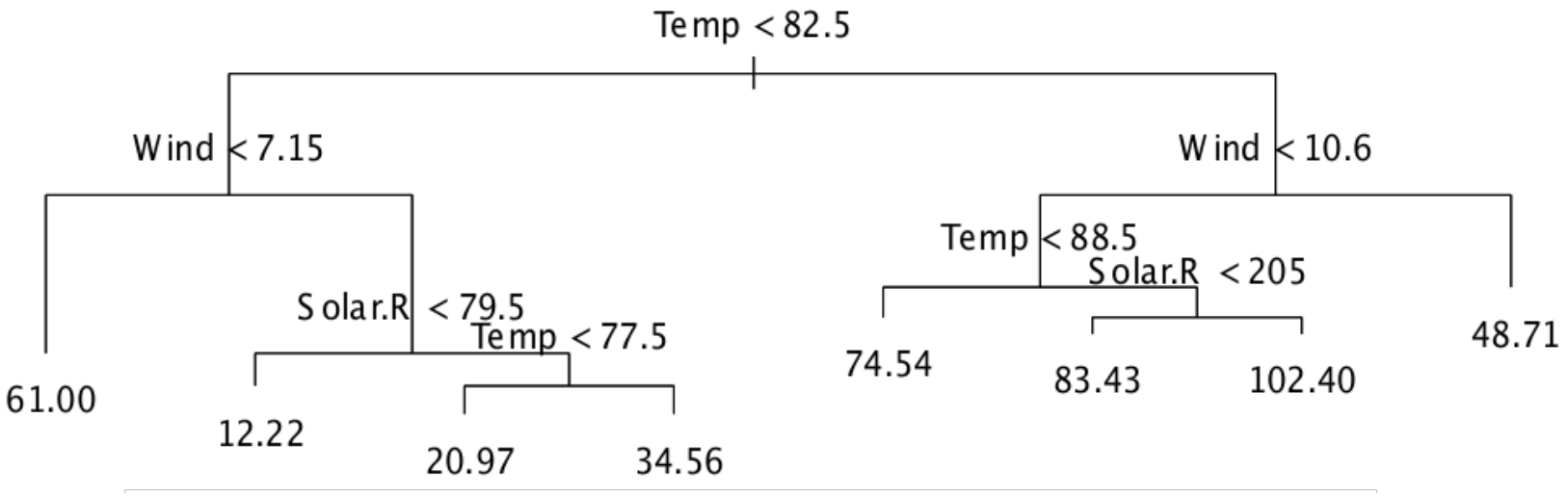
Relevé quotidien de la qualité de l'air (concentration d'ozone) à New York de mai à septembre 1973.

Jeu de données. 154 observations, 6 variables explicatives.

Variables explicatives. Vitesse moyenne du vent, température maximale, jour de la semaine, mois de l'année, ensoleillement.

Références. Sous `R`. Jeu de données `airquality`.

Exemple : prévision de la qualité de l'air



Construction de l'arbre

Nous devons répondre aux trois questions suivantes :

- Comment choisir la variable à segmenter parmis les variables explicatives disponibles ?
- Lorsque la variable est continue, comment déterminer le seuil de coupe ?
- Comment déterminer la bonne taille de l'arbre ?

Choisir la variable et trouver la bonne coupe

Le choix de la variable explicative j où couper, et de la coupe correspondante τ partitionne les données en deux ensembles :

$$E_{j,\tau}^{-} = \{i \in \{1, \ldots, n\} \ tq \ X_{i,j} \leqslant \tau\}$$

$$E_{j,\tau}^{+} = \{i \in \{1, \ldots, n\} \ tq. \ X_{i,j} > \tau\}$$

Exemple : prévision de la qualité de l'air

On peut décider de couper selon que la température est supérieure ou inférieure à 30°C.

Choisir la variable et trouver la bonne coupe

Le choix de la variable explicative j où couper, et de la coupe correspondante τ partitionne les données en deux ensembles :

$$E_{j,\tau}^{-} = \{i \in \{1, \dots, n\} \ tq \ X_{i,j} \leqslant \tau\}$$

$$E_{j,\tau}^{+} = \{i \in \{1, \dots, n\} \ tq. \ X_{i,j} > \tau\}$$

Idée. Obtenir deux groupes les plus homogènes possible pour la variable à prévoir Y. On minimise leur variance :

$$(j^\star, \tau^\star) = \arg\min_{j,\tau} \left\{ Var(E_{j,\tau}^{-}) + Var(E_{j,\tau}^{+}) \right\}$$

où $Var(E) = \min_{y \in \mathbb{R}} \sum_{i \in E} (Y_i - y)^2$

Choisir la variable et trouver la bonne coupe

Le choix de la variable explicative j où couper, et de la coupe correspondante τ partitionne les données en deux ensembles :

$$E_{j,\tau}^{-} = \{i \in \{1, \ldots, n\} \ tq \ X_{i,j} \leqslant \tau\}$$

$$E_{j,\tau}^{+} = \{i \in \{1, \ldots, n\} \ tq. \ X_{i,j} > \tau\}$$

Exemple : prévision de la qualité de l'air

On décide de couper selon que la température est supérieure ou inférieure à 30°C, si les données qui ont une température inférieure à 30°C ont une qualité de l'air similaire (de même pour celle supérieures).

Trouver la bonne coupe pour une variable catégorique

Pour des variables catégoriques non ordonnée, on ne peut pas trouver de seuil τ, pour diviser les données facilement en deux ensembles.

Exemple : prévision de la qualité de l'air

On peut avoir une variable renseignant si le temps est ensoleillé, nuageux, pluvieux, ou neigeux.

Si la variable prend ses valeurs dans un ensemble de k catégories, il y a $2^{k-1} - 1$ partitions correspondantes. C'est trop !

Solution. Ordonner les catégories suivant la moyenne des Y_i leur appartenant.

À quel point doit on laisser l'arbre grandir ?

Un arbre trop grand risque de suraprendre les données.
Un trop petit risque de ne pas capter leur structure.

La taille optimal est un paramètre à calibrer sur les données.

Solutions :

- Descendre tant que la variance au sein des feuilles décroit plus qu'un certain seuil.
  Vision à trop court terme : une mauvaise coupe peut parfois mener à de très bonnes coupes.
- Élagage. On fait grandir l'arbre au maximum puis on supprime les noeuds inintéressants.

Élagage (pruning)

On réexamine l'arbre et on retire les noeuds causés par le bruit présent dans l'ensemble d'entrainement.

On suppose qu'un ensemble de validation (test) est accessible. On minimise alors sur l'ensemble de test, l'erreur quadratique $R(T)$ pénalisée par la taille de l'arbre $|T|$:

$$R_\alpha(T) = R(T) + \alpha|T|$$

Le paramètre de complexité α pénalise les grands arbres.

Le cas $\alpha = 0$ revient à retirer les noeuds de l'arbre qui augmentent l'erreur de validation.

Avantages

- Facilement interprétable : règles de décisions simples
- Prévision rapide
- Méthode générique
  - facile à implémenter sur de nouvelles données
  - non paramétrique
  - peu d'hypothèse sur les variables
- Faible perturbation des valeurs extrêmes (outsiders) qui se retrouvent isolées dans de petites feuilles.
- Peu sensible au bruit des variables non discriminantes : non choisies lors des coupes et donc non introduites dans le modèle $\rightarrow$ possibilité d'un grand nombre de covariables.
- Utilisation de tous types de variables (catégoriques, discrètes, continues)

Inconvénients

- Instabilité (effet papillon) ! ! La modification d'une variable dans l'arbre transforme l'arbre complètement.
- Arbres non optimaux : règles heuristiques/empiriques.
- Long à entrainer : choix des coupes, élagage, ...

Exemple numérique : prévision de la qualité de l'air

Relevé quotidien de la qualité de l'air (concentration d'ozone) à New York de mai à septembre 1973.

Jeu de données. 154 observations, 6 variables explicatives.

Variables explicatives. Vitesse moyenne du vent, température maximale, jour de la semaine, mois de l'année, ensoleillement.

Références. Sous `R`. Jeu de données `airquality`.

Exemple numérique. Code R

Prévision de la qualité de l'air

```
# Chargement de la library
    library("tree")

# Chargement des données
    data(airquality)

# Construction de l'arbre
    ozone.tr <- tree(Ozone ~ ., data=airquality)

# Élagage
    ozone.tr2 <- prune.tree(ozone.tr)
```

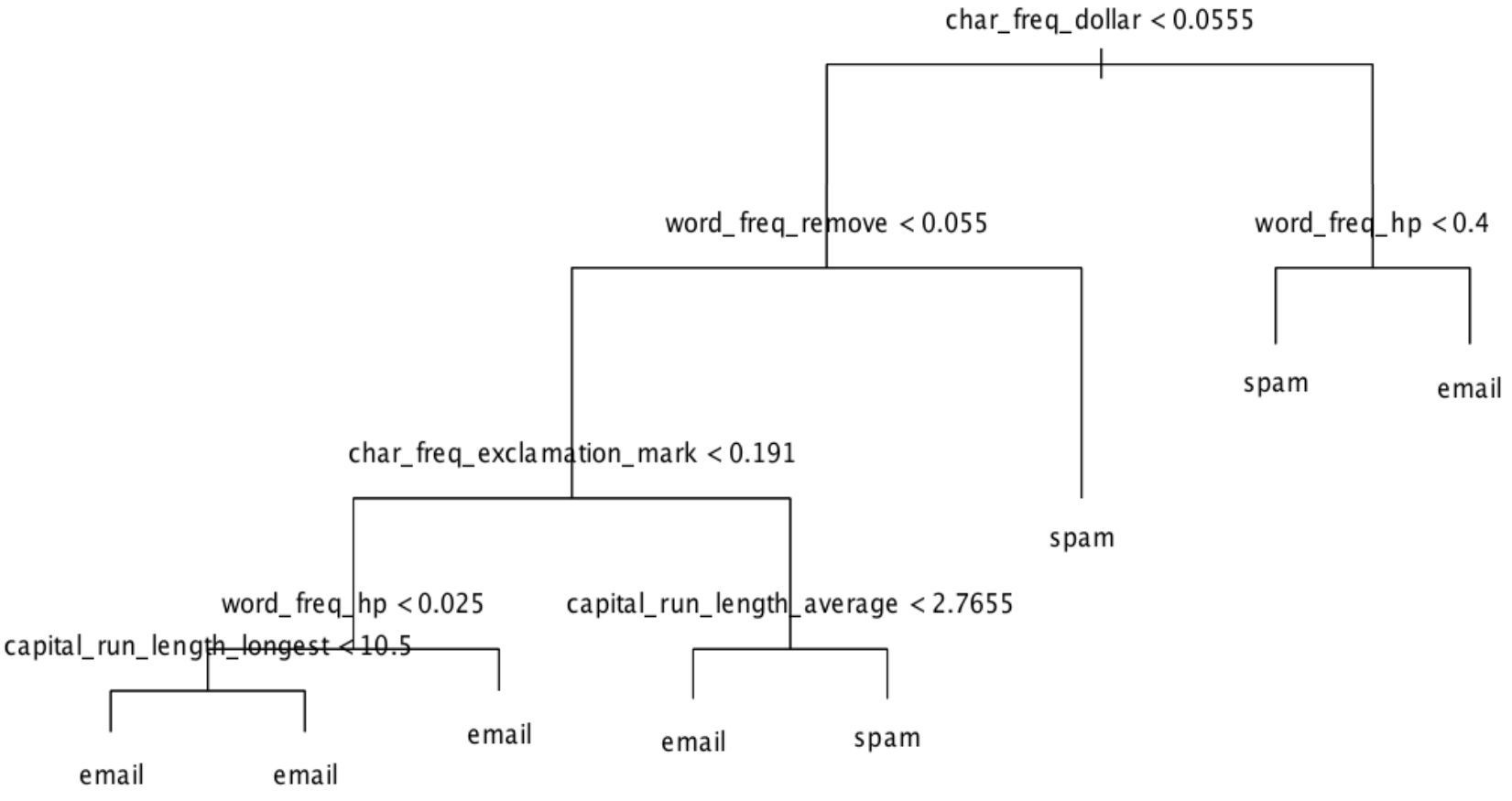
La classification

La variable à expliquer prend ses valeurs dans un ensemble de catégories.

Exemples.

Détection de spam. Reconnaissance d'objet, de races d'animaux, de variétés de plantes à partir de mesures,...

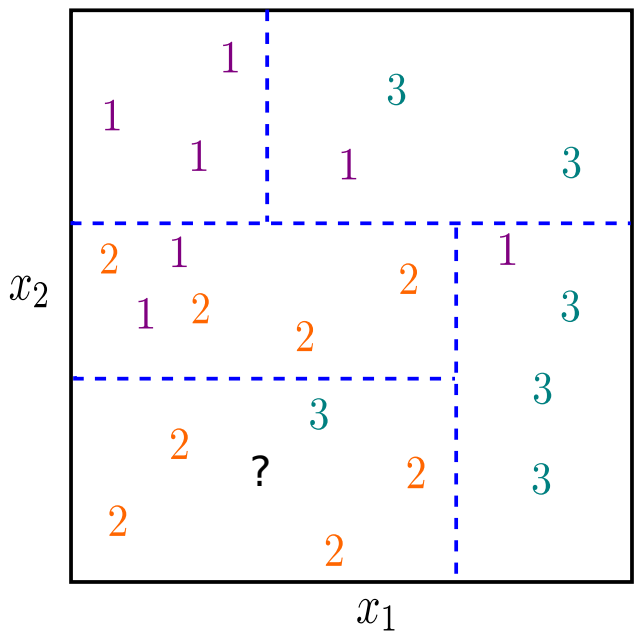
Arbre de classification. Détection de spams



Arbre de classification

Différences avec l'arbre de régression :

- au lieu de prévoir comme la moyenne des éléments de la feuille, on prévoit comme la majorité.
- le critère de variance pour sélectionner la variable et la coupe est légèrement différents.



Trouver la bonne coupe pour un arbre de classification

Encore une fois, on choisit ensuite la coupe de façon à maximiser l'homogénéité au seins des groupes.

On note $p(j|E)$ la proportion d'observations ayant le label j dans l'ensemble E. On remplace le critère de variance par

- l'entropie : $C(E) = -\sum_j p(j|E) \log p(j|E)$
- le critère de Gini : $C(E) = 1 - \sum_j p(j|E)^2$

On minimise ensuite le critère :

$$(j^\star, \tau^\star) = \arg\min_{j,\tau} \left\{ C(E^-_{j,\tau}) + C(E^+_{j,\tau}) \right\}$$

Exemple numérique. Détection de spams

Jeu de données contenant 2788 emails et 1813 spams en anglais.
Variables contextuelles disponibles :

- `word_freq_...` : 48 réels donnant le pourcentage de certains mots dans l'email (ex : `word_freq_free`)
- `char_freq_...` : 6 réels donnant le pourcentage de certains caractères (ex : $, !,...)
- `capital_run_length_average` : durée moyenne de suite de majuscules consécutives.
- `capital_run_length_longest` : durée de la plus grande suites de majuscules consécutives.
- `capital_run_length_total` : hombre total de majuscules.

II. Les méthodes d'ensembles et les forêts aléatoires

Pourquoi combiner ? Une intuition...

Classification binaire $Y \in \{-1, +1\}$, variables exogènes $X \in \mathbb{R}^p$.
On dispose d'un ensemble de K méthodes de classification initiales (h_k) indépendantes telles que

$$P\{h_k(X) \neq Y\} = \varepsilon$$

Alors en moyennant ces méthodes et en prévoyant

$$H(X) = \text{sign}\left(\sum_{k=1}^{K} h_k(X)\right)$$

par l'inégalité de Hoeffding, la probabilité d'erreur de H est

$$P\{H(X) \neq Y\} \leqslant \exp\left(-\frac{1}{2}K(2\varepsilon - 1)^2\right)$$

qui décroit vers 0 exponentiellement en K.

Le bagging (Bootstrap AGGregatING)

Introduit par Breiman (1996).
Deux ingrédients clefs : bootstrap et aggregation..

On sait que l'agrégation de méthode de prévision initiales indépendantes (base learners) mène à une réduction importante de l'erreur de prévision.
$\Rightarrow$ obtenir des méthodes initiales aussi indépendantes que possible.

Idée naive : entrainer nos "base learners" (ex : CART) sur des sous-ensembles d'observations disjoints de l'ensemble d'entrainement.

Problème : le nombre d'observations de l'ensemble d'entrainement n'est pas infini $\rightarrow$ les "base learners" auront trop peu de données et de mauvaises performances.

Idée du bagging

Le bagging crée des sous-ensembles d'entrainement à l'aide d'échantillonnage bootstrap [Elfron et Tibshirani, 1993].

Pour créer un nouveau "base learner"

1. on tire aléatoirement avec remise n observations de l'ensemble d'entrainement.
2. on entraine notre méthode (ex : CART) sur cet ensemble d'observations

Chaque "base learner" contient ainsi environ 36.8% des observations de l'ensemble d'entrainement.

La performance d'un "base learner" est obtenu par l'erreur "out-of-bag".

Les forêts aléatoires

Méthode introduite par Leo Breiman en 2001.

Idées plus anciennes : Bagging (1996), arbres de décisions CART (1984)

Preuves de convergences récentes (2006,2008)

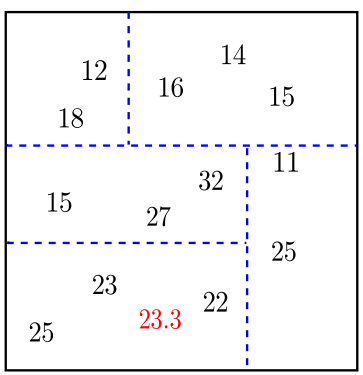
Un site web utile :

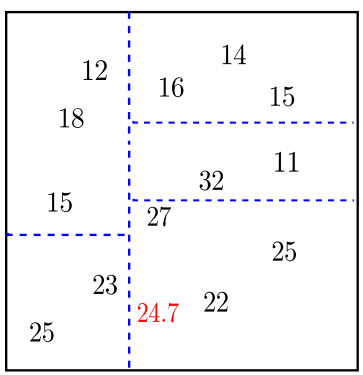
`http://www.stat.berkeley.edu/~breiman/RandomForests`

Forêts aléatoires

Les forêts aléatoires consistent à faire tourner en parallèle un grand nombre (≈ 400) d'arbres de décisions construits aléatoirement, avant de les moyenner.

En termes statistiques, si les arbres sont décorrélés, cela permet de réduire la variance des prévisions.



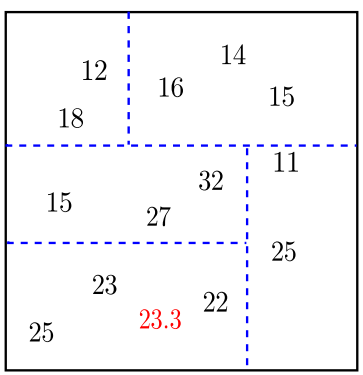


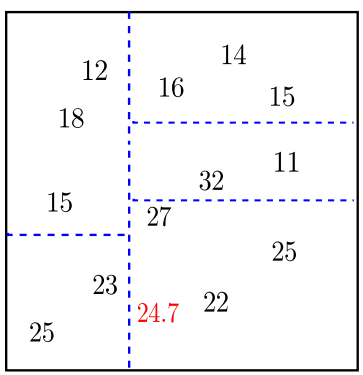
prévoit $\frac{24.7+23.3}{2} = 24$

Forêts aléatoires

Intuition : si K arbres sont identiquement distribués, de variance σ^2, avec un coefficient de corrélation deux à deux ρ, la variance de leur moyenne est alors

$$\bar{\sigma}^2 = \frac{1}{K}(K\sigma^2 + K(K-1)\rho\sigma^2) = \rho\sigma^2 + \frac{1-\rho}{K}\sigma^2$$





prévoit $\frac{24.7+23.3}{2} = 24$

Créer des arbres peu corrélés

Bootstrapping. Plutôt qu'utiliser toutes les données pour construire les arbres. On choisit pour chaque arbre aléatoirement un sous-ensemble (avec répétition possibles) des données.

Choix aléatoire de la variable explicative à couper.

À la différence des arbres Cart, on ne fait pas d'élagage.

Sélection aléatoire des variables de coupe

q : paramètre contrôlant l'aléatoire

Pour couper un noeud :

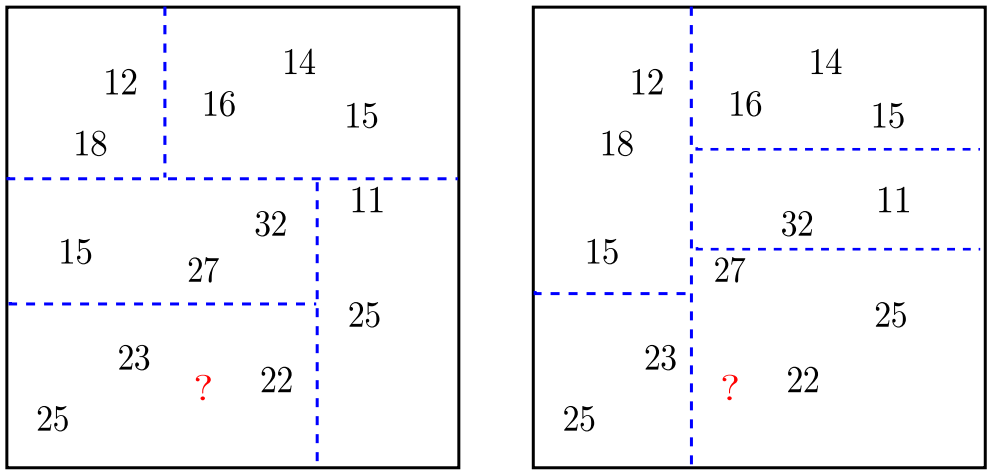
1. on choisit aléatoirement un sous-ensemble de q variables explicatives potentielles parmi les p disponibles
2. on choisit la variable à couper et le seuil de coupe en minimisant le critère de variabilité (Variance, Entropie, Gini) parmis ce sous-emsemble.

- si $q = p$: pas d'aléatoire. On retrouve le choix déterministe de CART
- si $q = 1$: aléatoire total dans le choix de la variable (mais pas dans le seuil de la coupe $\rightarrow$ VR-trees).

En pratique, Breiman (2001) propose $q \approx \log(p)$

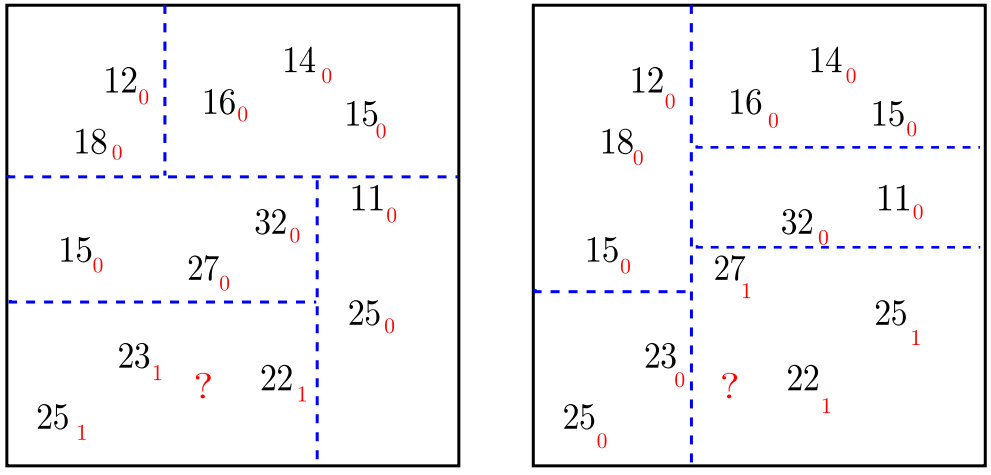
Une notion importante : la proximité

Intuition : Tomber souvent dans les même feuilles des arbres → expliquer la sortie Y de façon similaire.



Une notion importante : la proximité

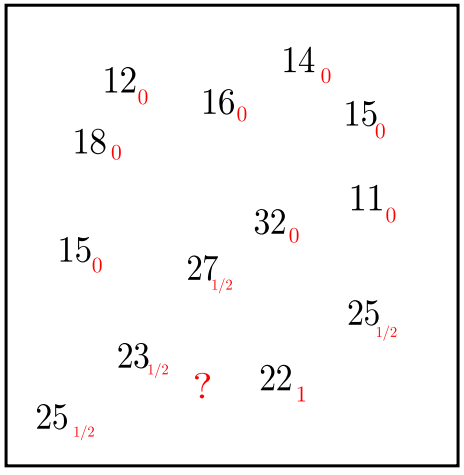
Intuition : Tomber souvent dans les même feuilles des arbres $\rightarrow$ expliquer la sortie Y de façon similaire.



$$\text{prox}(X_t, X_s) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1} \left\{ \begin{array}{c} X_t \text{ et } X_s \text{ tombent dans la} \\ \text{même feuille dans l'arbre } k \end{array} \right\}.$$

La notion de proximité

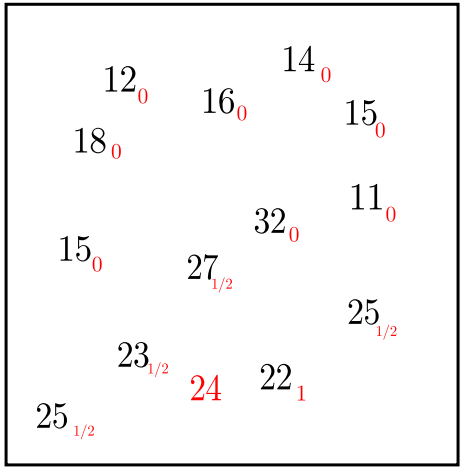
Intuition : Tomber souvent dans les même feuilles des arbres $\rightarrow$ expliquer la sortie Y de façon similaire.



$$\text{prox}(X_t, X_s) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\left\{ \begin{array}{c} X_t \text{ et } X_s \text{ tombent dans la} \\ \text{même feuille dans l'arbre } k \end{array} \right\}.$$

La notion de proximité

Intuition : Tomber souvent dans les même feuilles des arbres $\rightarrow$ expliquer la sortie Y de façon similaire.



On prédit ensuite par exemple par

$$\arg\min_{a\in\mathbb{R}} \sum_{X_s\in E_t} \text{prox}(X_t, X_s)\,(Y_s - a)^2\,.$$

Détection d'anomalies

Les forêts aléatoires se prêtent bien à la détection de valeurs extrêmes [Liu et al. 2008].

Celles-ci se retrouvent en effet isolées rapidement dans une feuille à part.

Le score d'anomalie $s(x)$ d'une observation x est déterminé approximativement par la longueur moyenne du chemin de x aux feuilles des arbres de la forêts.

Plus le chemin est court plus l'observation est susceptible d'être atypique.

Importance des variables explicatives

Les forêts aléatoires permettent de classer les variables explicatives par ordre d'importance dans la prévision.

Tout d'abord,

- on construit la forêt aléatoire
- on calcule l'erreur E "out-of-bag" de la forêt

Le score d'une variable explicative X_i est calculé comme suit

- on permute aléatoirement les valeurs de la variable explicative parmi les observations de l'ensemble d'entrainement
- on calcule à nouveau l'erreur out-of-bag et on fait la différence avec E.

On renormalise les scores.

Avantages-Inconvénients

Avantages

- pas de sur-apprentissage
- en général : meilleure performance que les arbres de décision
- calcul de l'erreur "Out-of-Bag" direct. Validation croisée non nécessaire
- paramètres faciles à calibrer

Inconvénients

- boite noire : difficilement interprétable
- entrainement plus lent

Exemple numérique. Code R

Prévision de la qualité de l'air

```
# Chargement de la library
    library("randomForest")

# Chargement des données
    data(airquality)

# Construction de l'arbre
    ozone.rf <- randomForest(Ozone ~ ., data=airquality, mtry=3,
                             importance=TRUE, na.action=na.omit)
```