

# ENSEMBLES D'ARBRES

## THÉORIE ET APPLICATION AU SCORING

---

Guillaume Ausset - Paris Saclay

30 novembre 2016

M2 MASEF - Crédit Agricole GRO

# INTRODUCTION

- Rappels théoriques sur l'apprentissage

  - Scoring et mesure des performances

  - Théorie de Vapnik-Chervonenkis

- Méthodes ensemblistes

  - Décomposition biais-variance

  - Moyennage de classifieurs

  - Boosting

  - Stacking

  - Combinaisons bayésiennes

- Arbres et forêts

  - Arbres de classification

  - Forêts aléatoires

- Données déséquilibrées

- Résultats

## RAPPELS THÉORIQUES SUR L'APPRENTISSAGE

---

**Classification binaire :** étant donné  $\mathbf{x}_i \in \mathcal{X}$ , déterminer  $y_i \in \mathcal{Y} = \{0, 1\}$

$$\text{Err} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$$

$$\text{Err}_{\text{équilibrée}} = \beta \frac{\text{FP}}{\text{TN} + \text{FP}} + (1 - \beta) \frac{\text{FN}}{\text{TP} + \text{FN}}$$

$$\text{G - mesure} = \sqrt{\frac{\text{FP}}{\text{TN} + \text{FP}} * \frac{\text{FN}}{\text{TP} + \text{FN}}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F - mesure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

*Problème :* mesures pas adaptées à la problématique du scoring.

**Score** : classification binaire à l'aide d'une variable latente sur laquelle la classification en elle-même pourra se faire par analyse discriminante.

$$\begin{cases} y = 1 \text{ si } S(x) \geq s \\ y = 0 \text{ si } S(x) < s \end{cases}$$

On veut que le score soit *bien ordonné* :

$$S(x_1) > S(x_2) \Leftrightarrow \mathbb{P}(y_1 = 1 \mid x_1) > \mathbb{P}(y_2 = 1 \mid x_2) \quad (1)$$

et (si possible) *bien calibré* :

$$S(x) = \mathbb{P}(y = 1 \mid x)$$

Si l'on note  $t$  le seuil de décision et  $T$  la sortie de notre classifieur, on a

$$\mathbb{P}(T > t \mid Y = 1) \simeq \frac{\text{TP}(t)}{\text{TP}(t) + \text{FN}(t)}$$

et de la même façon

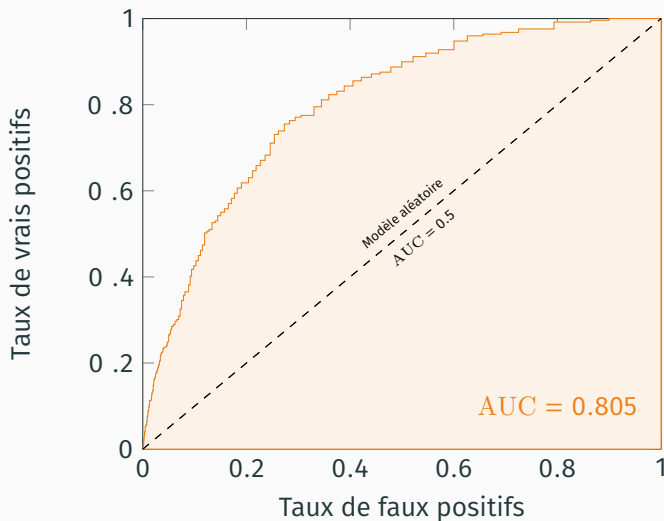
$$\mathbb{P}(T > t \mid Y = 0) \simeq \frac{\text{FP}(t)}{\text{TN}(t) + \text{FP}(t)}$$

La courbe ROC est alors la courbe paramétrée :

$$\text{ROC}(t) = \left[ \begin{array}{c} \mathbb{P}(T > t \mid Y = 0) \\ \mathbb{P}(T > t \mid Y = 1) \end{array} \right]$$

Sous cette forme l'AUC se calcule comme :

$$\begin{aligned} \text{AUC} &= - \int_0^1 \mathbb{P}(T > t \mid Y = 1) \mathbb{P}'(T > t \mid Y = 0) dt \\ &= \int_0^1 \mathbb{P}(T > t \mid Y = 1) p(t \mid Y = 0) dt \end{aligned}$$



**FIG. 1 :** Courbe ROC et AUC.



On obtient alors une interprétation de l'AUC qui explique tout son intérêt dans le cas du scoring :

$$\begin{aligned}\mathbb{P}(T^i > T^j \mid Y^i = 1, Y^j = 0) &= \iint_{\{(t^i, t^j) : t^i > t^j\}} p(t^i, t^j \mid Y^i = 1, Y^j = 0) dt^i dt^j \\ &= \int_0^1 \left( \int_{t^j}^1 p(t^i \mid Y^i = 1) dt^i \right) p(t^j \mid Y^j = 0) dt^j \\ &= \int_0^1 \mathbb{P}(T > t^j \mid Y = 1) p(t^j \mid Y^j = 0) dt^j \\ &= \text{AUC}\end{aligned}$$

## Definition (Règle de scoring propre)

Une règle de scoring

$$R : [0, 1]^J \times \mathcal{Y} \rightarrow \mathbb{R}$$

est dite propre si et seulement si

$$(\mathbb{P}(c_1), \dots, \mathbb{P}(c_J)) = \arg \max_p \mathbb{E}_Y [R(p, y)]$$

Règle logarithmique

$$R(p, y) = \log(p_{i_y})$$

Règle quadratique de Brier

$$R(p, y) = 2p_{i_y} - \|p\|_2$$

Règle sphérique

$$R(p, y) = \frac{p_{i_y}}{\|p\|_2}$$

où  $i_y$  est l'indice de la classe de  $y$ .

Problème d'approximation de  $f$  par une famille de fonctions, mais  $f$  connue seulement en certains points. On remplace la distance entre les fonctions par la *perte* aux points  $(\mathbf{x}_i, y_i)$  :

$$L(\varphi_{\mathcal{L}}(y_i, \mathbf{x}_i))$$

On minimise la perte moyenne

$$\text{Err}(\varphi) = \mathbb{E}_{X,Y} [L(\varphi_{\mathcal{L}}(Y, X))]$$

$$\varphi^{\star} = \arg \min_{\varphi_{\mathcal{L}} \in \Phi} \mathbb{E}_{X,Y} [L(Y, \varphi_{\mathcal{L}}(X))]$$

$$\simeq \arg \min_{\varphi_{\mathcal{L}} \in \Phi} \frac{1}{N} \sum_{i=1}^N L(y_i, \varphi_{\mathcal{L}}(\mathbf{x}_i))$$

$$\text{Err}_N(\varphi_{\mathcal{L}}) = \frac{1}{N} \sum_{i=1}^N L(y_i, \varphi_{\mathcal{L}}(\mathbf{x}_i))$$

$$\varphi_{\mathcal{L}}^{\star} = \arg \min_{\varphi \in \Phi} \text{Err}_N(\varphi)$$

## Definition

La procédure ERM est consistante si :

$$\begin{aligned} \text{Err}(\varphi_{\mathcal{L}}^{\star}) &\xrightarrow[|\mathcal{L}| \rightarrow +\infty]{\mathbb{P}} \text{Err}(\varphi^{\star}) \\ \text{Err}_N(\varphi_{\mathcal{L}}^{\star}) &\xrightarrow[|\mathcal{L}| \rightarrow +\infty]{\mathbb{P}} \text{Err}(\varphi^{\star}) \end{aligned}$$

## Théorème (Théorème fondamental de l'apprentissage)

*Pour une fonction de perte  $L$  bornée, la méthode ERM est consistante si et seulement si le risque empirique converge uniformément vers le vrai risque, c'est-à-dire :*

$$\lim_{|\mathcal{L}| \rightarrow +\infty} \mathbb{P} \left( \sup_{\varphi \in \Phi} |\text{Err}(\varphi_{\mathcal{L}}) - \text{Err}_N(\varphi_{\mathcal{L}})| > \varepsilon \right) = 0 \quad (2)$$

### Théorème

*Dans le cas de la classification binaire et avec la perte 0/1 on a :*

$$\mathbb{P} \left( \sup_{\varphi \in \Phi} |\text{Err}_N(\varphi) - \text{Err}(\varphi)| > \varepsilon \right) \leq 8G(\Phi, N)e^{-N\varepsilon^2/32} \quad (3)$$

$$\mathbb{E} \left[ \sup_{\varphi \in \Phi} |\text{Err}_N(\varphi) - \text{Err}(\varphi)| \right] \leq 2\sqrt{\frac{\log G(\Phi, N) + \log 2}{N}} \quad (4)$$

Or,

$$G(\Phi, N) \leq h \left( 1 + \ln \left( \frac{N}{h} \right) \right)$$

Donc il suffit de contrôler  $h$ , la dimension de VC.

**Conclusion :** Les méthodes introduites plus tard seront consistantes.

# MÉTHODES ENSEMBLISTES

---

Théorème (Décomposition biais-variance, GEMAN et al., 1992)

Dans le cas de la perte  $L^2$  on a :

$$\mathbb{E}_{\mathcal{L}} [\text{Err}(\varphi_{\mathcal{L}}(x))] = \text{bruit}(x) + \text{biais}^2(x) + \text{var}(x) \quad (5)$$

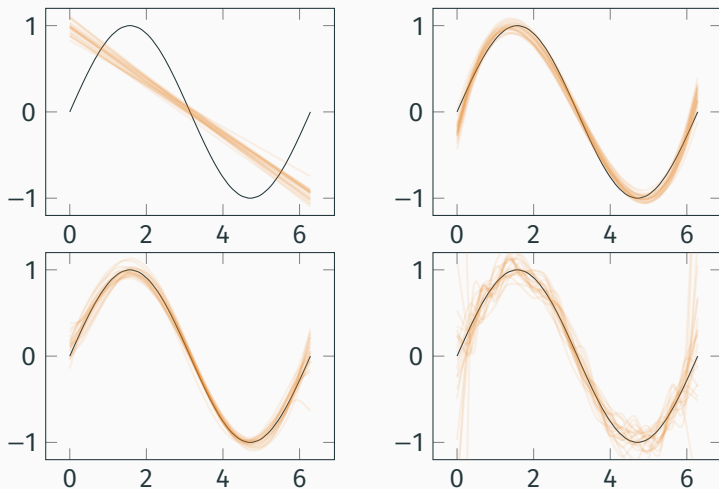
où

$$\text{bruit}(x) = \text{Err}(\varphi_B(x))$$

$$\text{biais}^2(x) = (\varphi_B(x) - \mathbb{E}_{\mathcal{L}} [\varphi_{\mathcal{L}}(x)])^2$$

$$\text{var}(x) = \mathbb{E}_{\mathcal{L}} [(\varphi_{\mathcal{L}}(x) - \mathbb{E}_{\mathcal{L}}[\varphi_{\mathcal{L}}(x)])^2]$$

( Rappel :  $\text{Err}(\varphi_{\mathcal{L}}) = \mathbb{E}_{X,Y} [L(Y, \varphi_{\mathcal{L}}(X))]$  )



**FIG. 2 :** Apprentissage de  $\sin(x)$  sur  $[0, 2\pi]$  à l'aide de polynômes de degrés 1, 3, 5 et 20.



Sous la règle de décision :  $\varphi_{\mathcal{L}}(x) = \arg \max_{c \in \mathcal{Y}} \hat{p}_{\mathcal{L}}(Y = c \mid X = x)$ , on trouve la décomposition :

$$\begin{aligned} & \mathbb{E}_{\mathcal{L}} \left[ \mathbb{E}_{Y|X=x} \left[ \mathbb{1}_{\varphi_{\mathcal{L}}(x) \neq Y} \right] \right] \\ &= \mathbb{P}(\varphi_B(x) \neq Y) + \mathbb{P}_{\mathcal{L}}(\varphi_{\mathcal{L}}(x) \neq \varphi_B(x)) (2\mathbb{P}(\varphi_B(x) = Y) - 1) \end{aligned}$$

Si l'on suppose alors que  $\hat{p}_{\mathcal{L}}(Y = \varphi_B(x)) \sim \mathcal{N}$  :

### Théorème (LOUPPE, 2014)

*Pour la perte 0-1 dans le cas de la classification, l'erreur de généralisation attendue  $\mathbb{E}_{\mathcal{L}} [\text{Err}(\varphi_{\mathcal{L}}(x))]$  en  $X = x$  admet la décomposition suivante :*

$$\begin{aligned} & \mathbb{E}_{\mathcal{L}} [\text{Err}(\varphi_{\mathcal{L}}(x))] = \mathbb{P}(\varphi_B(x) \neq Y) + \dots \\ & \dots + \Phi \left( \frac{0.5 - \mathbb{E}_{\mathcal{L}}[\hat{p}_{\mathcal{L}}(Y = \varphi_B(x))]}{\sqrt{\mathbb{V}_{\mathcal{L}} \hat{p}_{\mathcal{L}}(Y = \varphi_B(x))}} \right) (2\mathbb{P}(\varphi_B(x) = Y) - 1) \end{aligned}$$

Théorème (LOUPPE, 2014, cas de la régression  $L^2$ )

$$\psi_{\mathcal{L},1,\dots,M}(x) = \frac{1}{M} \sum_{m=1}^M \varphi_{\mathcal{L},m}(x)$$

$$\mathbb{E}_{\mathcal{L}} [\text{Err}(\psi_{\mathcal{L},1,\dots,M}(x))] = \text{bruit}(x) + \text{biais}^2(x) + \text{var}(x)$$

$$\text{bruit}(x) = \text{Err}(\varphi_B(x))$$

$$\text{biais}^2(x) = \left( \varphi_B(x) - \mathbb{E}_{\mathcal{L},i} [\varphi_{\mathcal{L},i}(x)] \right)^2$$

$$\text{var}(x) = \rho(x) \sigma_{\mathcal{L},i}^2(x) + \frac{1 - \rho(x)}{M} \sigma_{\mathcal{L},i}^2(x)$$

$$\sigma_{\mathcal{L},i}^2(x) = \mathbb{V}_{\mathcal{L},i}[\varphi_{\mathcal{L},i}(x)]$$

$$\mu_{\mathcal{L},i}(x) = \mathbb{E}_{\mathcal{L},i}[\varphi_{\mathcal{L},i}(x)]$$

$$\rho(x) = \frac{\mathbb{E}_{\mathcal{L},i,i'}[\varphi_{\mathcal{L},i}(x) \varphi_{\mathcal{L},i'}(x)] - \mu_{\mathcal{L},i}^2(x)}{\sigma_{\mathcal{L},i}^2(x)}$$

Le but est d'approximer  $f$  par une combinaison linéaire de fonctions  $b(x, \gamma_k)$ , c'est-à-dire

$$f(x) = \sum_{k=1}^K \beta_k b(x, \gamma_k)$$

---

## Algorithme 1 Gradient Boosting

---

**procedure** GRADIENT BOOSTING( $\mathcal{L}, L$ )

$f_0(x) \leftarrow \arg \min_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$

**pour**  $k = 1, \dots, K$  **faire**

$[r_m]_i \leftarrow - \left[ \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x_i)=f_{k-1}(x_i)}$

Entraîne  $B_k(x)$  sur les pseudo-résidus  $\{(x_i, [r_k]_i)\}$

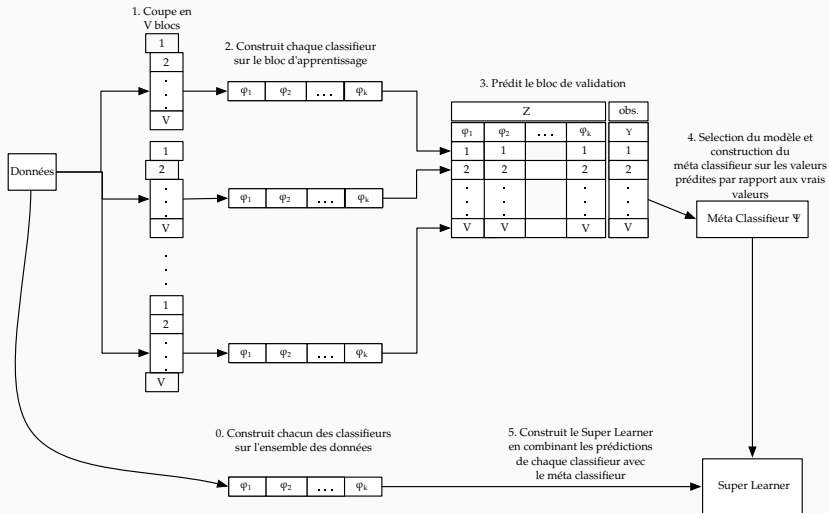
$\gamma_k \leftarrow \arg \min_{\gamma} \sum_{i=1}^N L(y_i, f_{k-1}(x_i) + \gamma B_k(x_i))$

$f_k(x) \leftarrow f_{k-1} + \gamma_k B_k(x)$

**fin pour**

**renvoyer**  $f_K(x)$

**fin procedure**



**FIG. 3 : Fonctionnement de l'algorithme SuperLearner, adapté de van der LAAN et al. (2007)**

On veut prendre en compte l'incertitude quant au « vrai » modèle :

$$\mathbb{P}(\zeta \mid \mathcal{L}) = \sum_{m=1}^M \mathbb{P}(\zeta \mid \varphi_m, \mathcal{L}) \mathbb{P}(\varphi \mid \mathcal{L}) \quad (6)$$

où  $\zeta$  est une quelconque valeur d'intérêt, dans la plupart des cas  $\zeta = y$  et on a donc :

$$\mathbb{P}(y_i \mid x_i, \mathcal{L}) = \sum_{m=1}^M \mathbb{P}(y_i \mid x_i, \varphi_m, \mathcal{L}) \mathbb{P}(\varphi \mid \mathcal{L}) \quad (7)$$

En utilisant encore une fois le théorème de Bayes, on peut aussi écrire :

$$\mathbb{P}(\varphi \mid \mathcal{L}) \propto \mathbb{P}(\varphi_m) \mathbb{P}(\mathcal{L} \mid \varphi_m) \quad (8)$$

$$\propto \mathbb{P}(\varphi_m) \int \mathbb{P}(\mathcal{L} \mid \theta_m, \varphi_m) \mathbb{P}(\theta_m \mid \varphi_m) d\theta_m \quad (9)$$

où  $\mathbb{P}(\theta_m \mid \varphi_m)$  est définie *a priori*.

Deux inconvénients majeurs mis en avant dans MINKA (2002).

1. L'espace d'hypothèses sur lequel on cherche la solution est nettement plus restreint que l'espace des combinaisons linéaires
2. Tendance à converger vers le modèle le plus probable ; tous les poids sont concentrés sur un unique modèle.

Pour pallier ces 2 problèmes, MONTEITH et al. (2011) proposent une modification du BMA. Au lieu de considérer l'espace des modèles possibles  $\mathcal{M}$ , ils considèrent l'espace des combinaisons de  $\mathcal{M}$ ,  $\mathcal{C}(\mathcal{M})$ .

$$\mathbb{P}(y_i | x_i, \mathcal{L}) = \sum_{c \in \mathcal{C}(\mathcal{M})} \mathbb{P}(y_i | x_i, \varphi_m, \mathcal{L}) \mathbb{P}(c | \mathcal{L}) \quad (10)$$

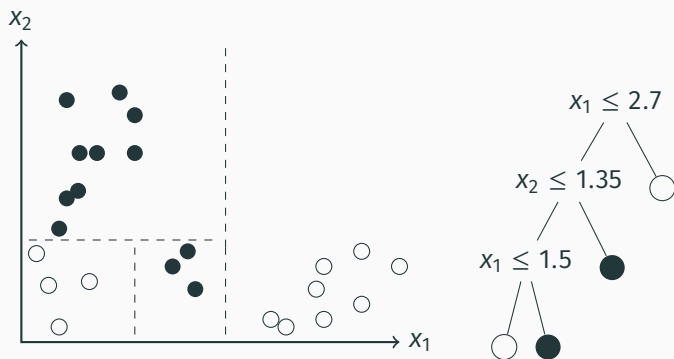
# ARBRES ET FORÊTS

---

- Ils sont non-paramétriques.
- Ils peuvent traiter les données hétérogènes (mélange de variables qualitatives et quantitatives) ainsi que les données manquantes.
- Ils effectuent automatiquement une sélection des variables et sont donc en partie résistants aux variables inutiles ou bruitées.
- Ils sont résistants aux outliers ou erreurs d'étiquetage.
- Ils sont faciles d'interprétation.



Un *arbre de décision* est un classifieur  $\varphi : \mathcal{X} \rightarrow \mathcal{Y}$  représenté par un arbre enraciné où chaque nœud  $t$  représente un sous-espace  $\mathcal{X}_t \subseteq \mathcal{X}$  avec la racine  $t_0$  représentant  $\mathcal{X}$  tout entier. Les fils  $t_i, \dots, t_j$  de  $t$  représentent alors une partition disjointe de  $\mathcal{X}_t$ .



**FIG. 4 :** Équivalence entre la partition du plan et un arbre de décision binaire

## Definition (Réduction de l'impureté)

La *réduction d'impureté pondérée* d'une partition binaire  $s \in Q$  divisant  $t$  en  $t_L$  et  $t_R$  est :

$$\Delta i(s, t) = i(t) - p_L i(t_L) - p_R i(t_R)$$

où  $p_L$  (resp.  $p_R$ ) est la proportion de l'échantillon dans  $\mathcal{L}_t$  allant dans  $t_L$  (resp.  $t_R$ ).

Notre problème de minimisation de l'*impureté* est alors :

$$\begin{cases} s_j^* = \underset{s_j \in Q(X_j)}{\operatorname{argmax}} \Delta i(s_j, t) \\ j=1, \dots, p \end{cases}$$

### Théorème (BREIMAN et al., 1984)

Soit  $\Phi(p_1, \dots, p_J)$  une fonction strictement concave définie sur  $0 \leq p_k \leq 1$ , pour  $k = 1, \dots, J$  et  $\sum_{k=1}^J p_k = 1$  telle que :

- $\Phi(1, \dots, 0) = \Phi(0, 1, \dots, 0) = \dots = \Phi(0, \dots, 1)$  soit minimale
- $\Phi\left(\frac{1}{J}, \dots, \frac{1}{J}\right)$  soit maximale

Alors pour  $i(t) := \Phi(p(c_1 | t), \dots, p(c_J | t))$  et pour n'importe quel  $s$ ,  
$$\Delta i(s, t) \geq 0$$

avec égalité si et seulement si  $p(c_k | t_L) = p(c_k | t_R) = p(c_k | t)$  pour  $k = 1, \dots, J$ .

### Classification

$$i_H(t) = -\frac{1}{2} \sum_{k=1}^J p(c_k | t) \log_2 (p(c_k | t))$$

$$i_G(t) = \sum_{k=1}^J p(c_k | t) (1 - p(c_k | t))$$

### Regression

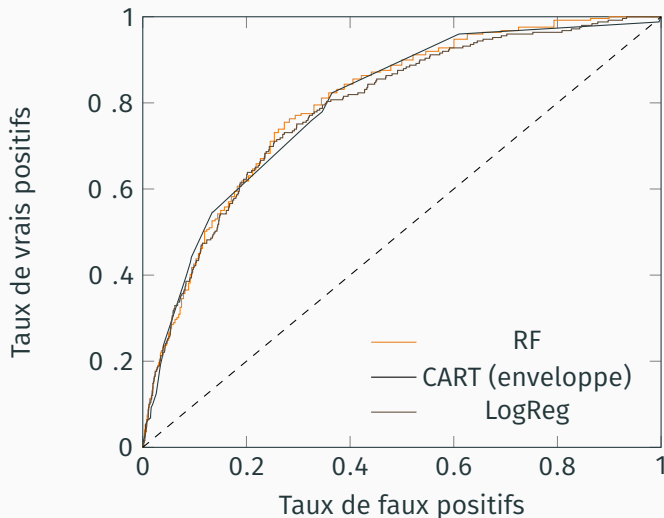
$$i_R(t) = \frac{1}{N_t} \sum_{x,y \in \mathcal{L}_t} (y - \hat{y}_t)^2$$

- Agrégation par vote majoritaire/moyenne.
- À chaque étape de création d'une coupure, on tire aléatoirement un sous-ensemble de  $K$  variables de  $\mathcal{X}$  puis on applique la procédure de choix de la coupe.
- Un échantillon bootstrap de l'échantillon d'apprentissage est tiré avec remise et l'arbre est construit sur ce nouvel échantillon tiré aléatoirement.

BREIMAN, 2004 montre que la convergence des forêts aléatoires ne dépend que des variables dites *fortes* et non pas des variables *faibles* bruitées. Il montre que la convergence pour une version simplifiée des forêts aléatoires vers l'erreur de Bayes est de l'ordre de

$$N^{-\frac{3}{4|S|+3}}$$

avec  $S$  les variables fortes définies telles que  $\mathbb{E}[y \mid \mathbf{x}]$  ne dépend que de  $S$ .



**FIG. 5 :** Performances de RF ( $AUC = 0.805$ ) par rapport à la régression logistique ( $AUC = 0.7889$ ) et un arbre seul ( $AUC = 0.79649$ ).

## Lemme (Johnson-Lindenstrauss en distribution)

$\forall \varepsilon > 0, \delta < 1/2$  et  $d \in \mathbb{N}^+$  il existe une distribution  $\mu$  sur  $\mathcal{R}^{k \times d}$  avec  $k = O(\varepsilon^{-2} \log(1/\delta))$  et  $A \sim \mu$  tel que :

$$\forall x, \|x\| = 1 : \mathbb{P} (|\|Ax\|_2^2 - 1| > \varepsilon) < \delta$$

Un choix possible pour  $A$  qui possède l'avantage d'accélérer grandement le produit matriciel est donné par ACHLIOPTAS (2003) :

$$A_{i,j} = \sqrt{3} \begin{cases} +1 \text{ avec probabilité } 1/6 \\ 0 \text{ avec probabilité } 2/3 \\ -1 \text{ avec probabilité } 1/6 \end{cases}$$

---

## Algorithme 2 Randomer Forest

---

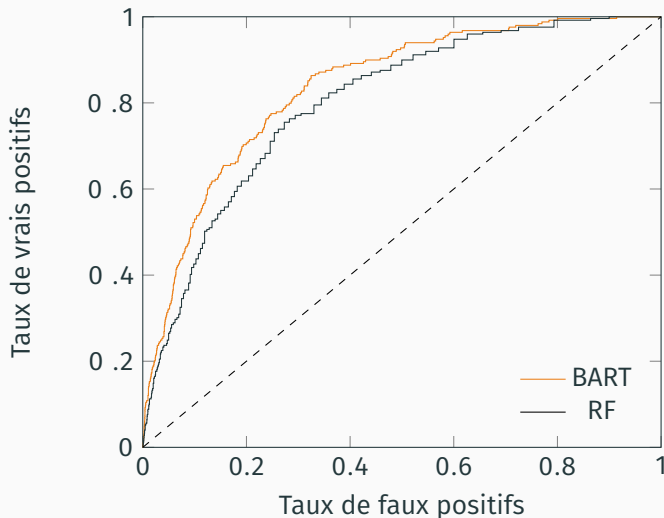
```
procedure RANDOMER FOREST( $\mathcal{L}, \mu(\mathcal{L})$ )  
  pour chaque arbre  $n$  faire  
    Création du sous-échantillon  $\mathcal{L}_n$   
    pour chaque nœud  $i$  de l'arbre faire  
       $\tilde{X}_{i,n} \leftarrow A_{i,n}^\top X_n$  avec  $A_{i,n} \sim \mu(\mathcal{L}_{i,n})$   
       $s_{i,n}^* \leftarrow \text{MeilleureCoupure}(\tilde{X}_{i,n})$   
      Coupe  $\mathcal{L}_{i,n}$  en  $\mathcal{L}_{l,n}$  et  $\mathcal{L}_{r,n}$  selon la règle de coupure  $s_{i,n}^*$   
    fin pour  
  fin pour  
  renvoyer ( $\text{Arbres}_n, (A_{i,n})$ )  
fin procedure
```

---



- On modélise les données comme une somme d'arbres.
- On définit un a priori sur la forme des arbres, un a priori sur les feuilles et un a priori sur l'erreur.
- Tirage par Bayesian Backfitting c'est-à-dire un mélange de Gibbs et Metropolis-Hastings qui exploite la forme de la somme.

Le choix des a priori est vital pour la facilité des calculs et la capacité de mélange de la chaîne de Markov : choix de lois conjuguées et chaînes réversibles quand possible.



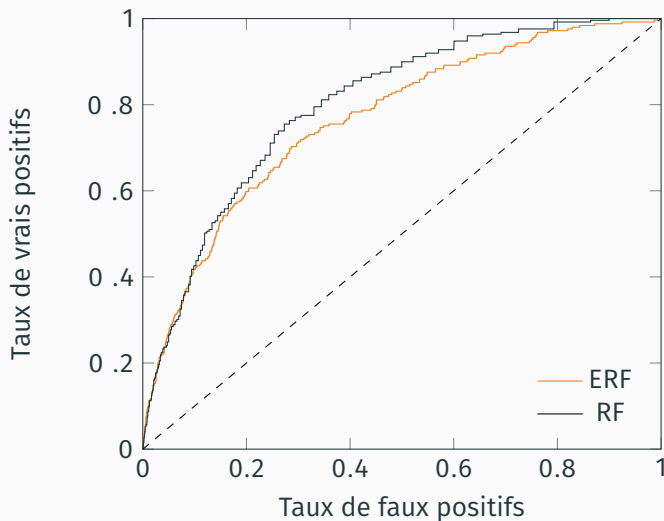
**FIG. 6 :** Performances de BART (AUC = 0 .8365).

On veut séparer le processus de création de la structure d'arbre de celui d'apprentissage des étiquettes.

On peut transformer le problème d'apprentissage des feuilles pour exploiter la structure de forêt :

$$\arg \min_{\omega} \left\{ \frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T l(y_i^t, \hat{y}_i) \right. \\ \left. y_i^t = \omega_t \varphi_t(x_i), \forall i, t \right\} \longrightarrow \arg \min_W \left\{ \frac{1}{2} \|W\|_2^2 + \frac{C}{N} \sum_{i=1}^N l(y_i, \hat{y}_i) \right. \\ \left. y_i = W \Phi(x_i), \forall i \right\}$$

On se ramène donc à un problème de SVM simple. On peut également construire une procédure d'effeuillage globale en réunissant les couples de feuilles de normes faibles.



**FIG. 7 :** Performances de ERF ( $AUC = 0.7664$ ).

On peut aussi chercher à apprendre les feuilles de façon à optimiser l'AUC.

$$\text{AUC}(\varphi, \mathcal{L}) = \frac{\sum_{k_0 \in \mathcal{L}_0} \sum_{k_1 \in \mathcal{L}_1} \mathbb{1}_{\varphi(x_0) < \varphi(x_1)}}{|\mathcal{L}_0| |\mathcal{L}_1|}$$

Mais l'AUC n'est pas dérivable. On va alors optimiser un substitut :

$$\text{AUC}(\varphi, \mathcal{L}) = \frac{\sum_{k_0 \in \mathcal{L}_0} \sum_{k_1 \in \mathcal{L}_1} f(\varphi(x_1) - \varphi(x_0))}{|\mathcal{L}_0| |\mathcal{L}_1|}$$

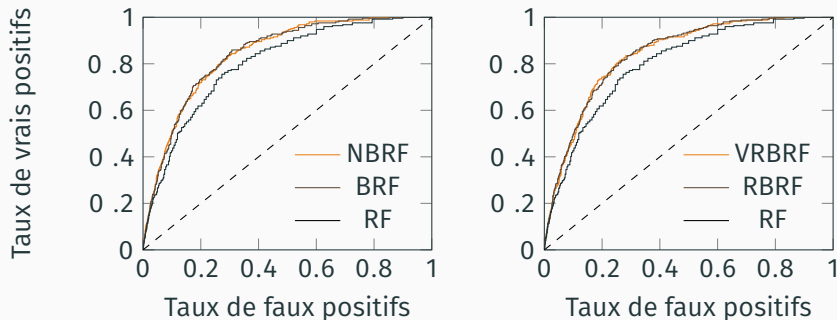
avec  $f$  dérivable.

# DONNÉES DÉSÉQUILIBRÉES

---

Le sous-échantillonnage permet d'éviter le sur-apprentissage de la classe majoritaire et implicitement l'attribution de poids plus importants à la classe majoritaire. De plus le sous-échantillonnage a pour effet secondaire d'accélérer l'apprentissage.

- On peut effectuer un sous-échantillonnage directement durant l'étape bootstrap en forçant l'équilibre.
- On peut également équilibrer « en moyenne » afin de diminuer encore plus la corrélation entre les arbres tout en gardant un certain équilibre.
- La proportion optimale n'est pas forcément 50/50. On peut donc également effectuer un sous-échantillonnage non équilibré.  
(Proportion  $\frac{1}{2} + \frac{2q_a - 1}{4}$ )



**FIG. 8 :** Performances de BRF (AUC = 0.8432), NBRF (AUC = 0.8427), RBRF (AUC = 0.8432) et VRBRF (AUC = 0.8421)



Puisque les classes majoritaires et minoritaires n'ont pas la même importance, on introduit un coût de mauvaise classification. On cherche alors à minimiser le risque conditionnel

$$R(i | x) = \sum_j \mathbb{P}(j | x) C(i, j) \quad (11)$$

On peut alors :

1. Utiliser notre classifieur pour déterminer la probabilité inconnue (par bagging par exemple si le classifieur ne donne pas de probabilité naturellement).
2. Résoudre l'équation 11
3. Remplacer dans l'échantillon les vraies étiquettes par les étiquettes « idéales » trouvées précédemment.
4. Apprendre notre classifieur final sur ce nouvel échantillon.

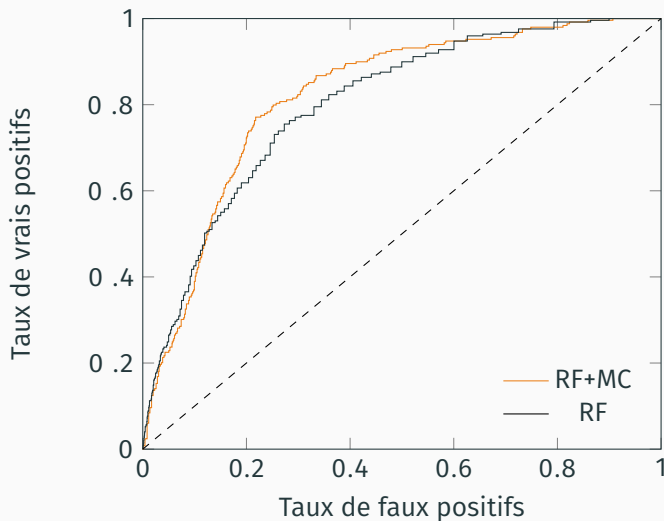
---

### Algorithme 3 Algorithme Metacost pour les forêts aléatoires

---

```
procédure METACOST-RF( $S, C$ )  
   $RF \leftarrow \text{ForêtsAléatoire}(S)$   
  pour  $(y, x) \in S$  faire  
    pour chaque classe  $j$  faire  
      si  $x \in \text{OOB}(RF)$  alors  
         $\hat{p}(j \mid x) \leftarrow \mathbb{P}_{\text{OOB}}(j \mid x)$   
      sinon  
         $\hat{p}(j \mid x) \leftarrow RF(j \mid x)$   
      fin si  
    fin pour  
     $y \leftarrow \arg \min_i \sum_j \hat{p}(j \mid x) C(i, j)$   
  fin pour  
  renvoyer  $S$   
fin procédure
```

---



**FIG. 9** : Performances de MetaCost + RF ( $AUC = 0.8217$ ).

## RÉSULTATS

---

# RÉSULTATS

Classifieur	AUC	Brier	AUC	Brier
Régression logistique	0.789	0.031	0.673	0.024
CART	0.796	0.032	0.654	0.025
Forêts aléatoires	0.805	0.029	0.699	0.023
Boosting Arbres	0.810	0.030	0.635	0.025
Stacking ~ Réseau de neurones	0.737	0.040	0.632	0.025
Balanced RF	0.843	0.151	0.717	0.145
Weighted RF	0.801	0.029	0.702	0.023
Roughly Balanced RF	0.843	0.154	0.721	0.146
Nearly Balanced RF	0.843	0.133	0.724	0.122
Very Roughly Balanced RF	0.842	0.130	0.721	0.121
RF + Metacost	0.822	0.294	0.698	0.189
RF + Feuilles SVMs	0.766	0.114	0.590	0.025
RF Uniformes	0.842	0.028	0.705	0.024
RF Uniformes Équilibrées	0.843	0.215	0.731	0.208
ExtRa-Trees	0.779	0.029	0.675	0.025
BART	0.837	0.027	0.732	0.023

**TAB. 1 :** Performances des différentes méthodes sur 2 bases de validation

## CONCLUSION

- Méthodes ensemblistes : façon simple d'améliorer les performances.
- Rééquilibrage crucial.
- Très bonnes performances prédictives des méthodes bayésiennes, très mauvaises performances computationnelles : perspectives d'amélioration intéressantes.
- Il semble judicieux de se concentrer sur les méthodes de *ranking* et d'optimiser l'AUC directement au lieu de réadapter les méthodes d'apprentissage classiques.

MERCI



## ANNEXE

---

Condition suffisante (non nécessaire) pour assurer que  $p(\theta)$  est la distribution invariante est la condition de réversibilité :

$$p(\theta_i)g(\theta_{i-1} | \theta_i) = p(\theta_{i-1})g(\theta_i | \theta_{i-1}) \quad (12)$$

L'algorithme de Metropolis-Hastings utilise alors une densité de proposition  $q(\theta' | \theta)$ , qui joue le rôle de chaîne de Markov continue. À partir de  $\theta$ , une valeur  $\theta'$  est proposée puis acceptée comme nouvel état avec probabilité :

$$A(\theta, \theta') = \min \left\{ 1, \frac{p(\theta')q(\theta | \theta')}{p(\theta)q(\theta' | \theta)} \right\}$$

## Théorème (Théorème Ergodique)

Soient  $\hat{\theta}_1, \dots, \hat{\theta}_m$  une suite de valeurs d'une chaîne de Markov de noyaux de transition  $g$ , et matrice de transition  $P$ , telle que la chaîne soit :

$$\exists i, \forall j > i, g(\hat{\theta}_j | \hat{\theta}_i) > 0$$

$$\forall i, j \ g(\hat{\theta}_j | \hat{\theta}_i) > 0$$

$$\forall i, \mathbb{E}[T_i] < \infty$$

$$T_i = \inf \{j \geq 1 : \hat{\theta}_j = i \mid \hat{\theta}_0 = i\}$$

Alors si  $\mathbb{E}[f(\theta)] < \infty$  alors

$$\mathbb{P} \left( \frac{1}{n} \sum_{i=1}^n f(\hat{\theta}_i) \rightarrow \int f(\theta) \pi(\theta) d\theta \right)$$

où  $\pi$  est la distribution stationnaire de la chaîne de Markov c'est-à-dire  $\pi = \pi P$

---

## Algorithme 4 Metropolis-Hastings

---

```
procedure METROPOLIS-HASTINGS( $q, \theta_0, n$ )  
  pour  $i \in [0, n - 1]$  faire  
    Tire  $u \sim \mathcal{U}_{[0,1]}$   
    Tire  $\theta' \sim q(\theta' \mid \theta_i)$   
    si  $u < \min \left\{ 1, \frac{p(\theta')q(\theta_i \mid \theta')}{p(\theta_i)q(\theta' \mid \theta_i)} \right\}$  alors  
       $\theta_{i+1} = \theta'$   
    sinon  
       $\theta_{i+1} = \theta_i$   
    fin si  
  fin pour  
  renvoyer  $(\theta_0, \dots, \theta_{n-1})$   
fin procedure
```

---

On a ici que

$$g(\theta_i, \theta_{i+1}) = q(\theta_{i+1} \mid \theta_i)A(\theta_i, \theta_{i+1}) + \delta_{\theta_i}(\theta_{i+1})R(\theta_i)$$

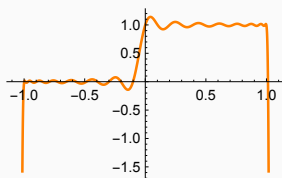
Avec  $R\theta_i$  le terme de rejet :

$$R(\theta_i) = \int q(\theta \mid \theta_i)(1 - A(\theta_i, \theta)) \, d\theta$$

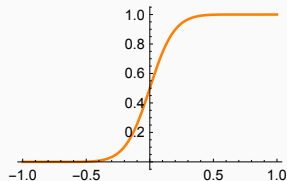
Par construction  $g$  remplit la propriété 12. De plus, le terme de rejet garantit l'apériodicité et il suffit que le support de  $q$  inclue le support de  $p$  pour garantir l'irréductibilité.

La présence de  $p$  non seulement au numérateur, mais aussi au dénominateur permet de se débarrasser de la constante de renormalisation, souvent l'un des éléments les plus contraignants.

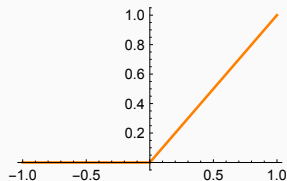
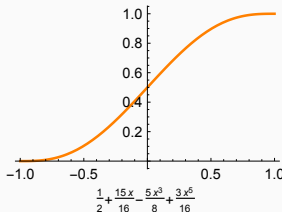
# PROBLÈMES AVEC L'APPROXIMATION DE L'AUC



Approx. polynomiale



Approx. rationnelle



Éventuel substitut