ENSEMBLES D'ARBRES

Théorie et application au scoring

Guillaume Ausset

5 novembre 2015

M2 MASEF - Crédit Agricole GRO

Introduction

PLAN

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

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* :

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

et (si possible) *bien calibré* :

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

MESURE DE PERFORMANCE POUR LE SCORING : AUC

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

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

et de la même façon

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

La courbe ROC est alors la courbe paramétrée :

$$\mathrm{ROC}(t) = \begin{bmatrix} \mathbb{P}(T > t \mid Y = 0) \\ \mathbb{P}(T > t \mid Y = 1) \end{bmatrix}$$

Sous cette forme l'AUC se calcule comme :

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

MESURES DE PERFORMANCES : AUC

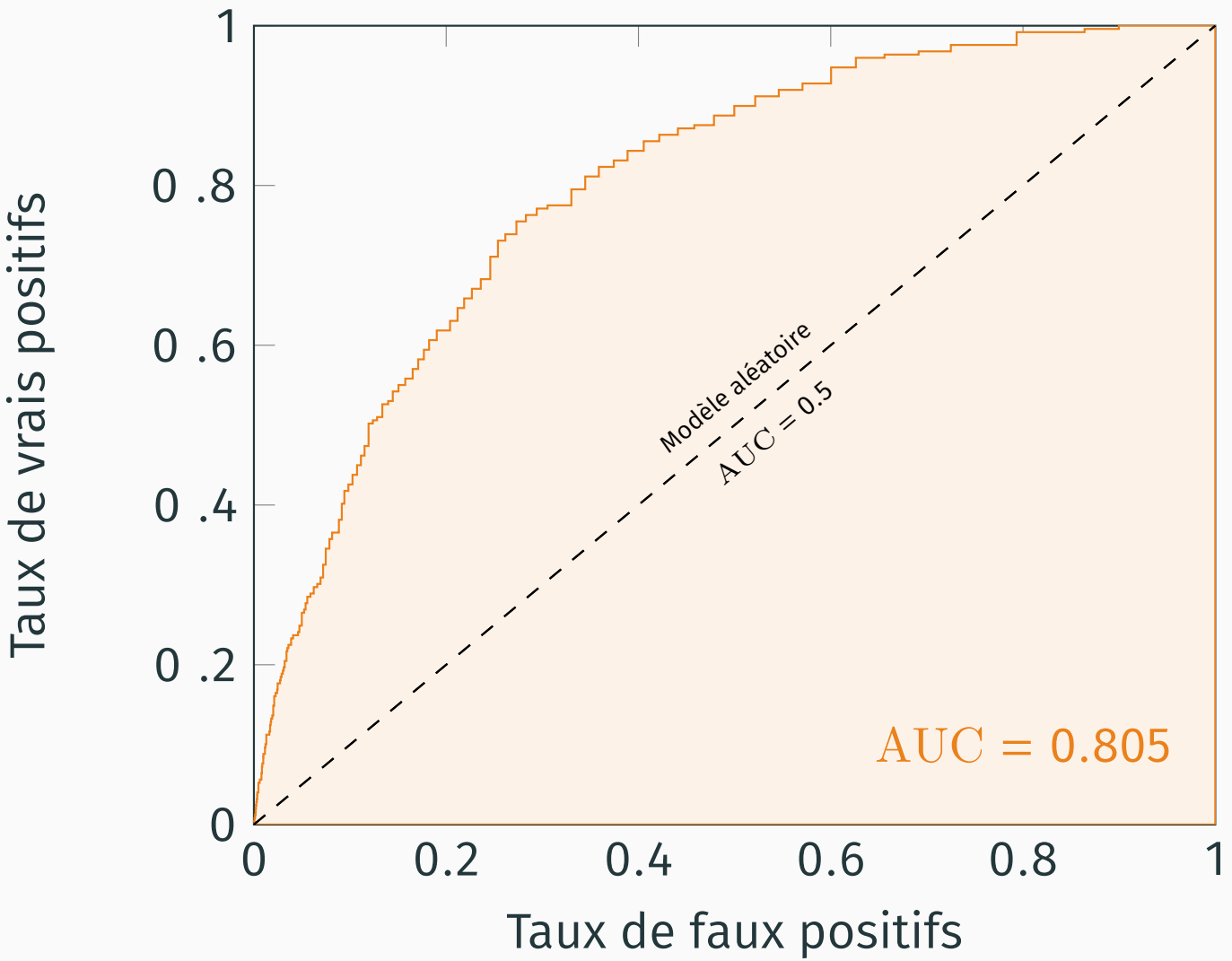


FIG. 1 : Courbe ROC et AUC.

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

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

Definition (Régle de scoring propre)

Une règle de scoring

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

est dite propre si et seulement si

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

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.

MINIMISATION DU RISQUE EMPIRIQUE

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

$$\begin{aligned}
\mathrm{Err}(\varphi) &= \mathbb{E}_{X,Y}\left[L(\varphi_{\mathcal{L}}(Y, X))\right] \\
\varphi^{\star} &= \arg\min_{\varphi_{\mathcal{L}} \in \Phi} \mathbb{E}_{X,Y}\left[L\left(Y, \varphi_{\mathcal{L}}(X)\right)\right] \\
&\simeq \arg\min_{\varphi_{\mathcal{L}} \in \Phi} \frac{1}{N} \sum_{i=1}^{N} L\left(y_i, \varphi_{\mathcal{L}}(\S_i)\right)
\end{aligned}$$

$$\mathrm{Err}_N(\varphi_{\mathcal{L}}) = \frac{1}{N} \sum_{i=1}^{N} L\left(y_i, \varphi_{\mathcal{L}}(\S_i)\right)$$

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

Théorème

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

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

$$\mathbb{E}\left[\sup_{\varphi\in\Phi}|\mathrm{Err}_N(\varphi)-\mathrm{Err}(\varphi)|\right]\leq 2\sqrt{\frac{\log G(\Phi,N)+\log 2}{N}} \tag{3}$$

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

Biais-variance : cas de la régression

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

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

$$\mathbb{E}_{\mathcal{L}}\left[\mathrm{Err}(\varphi_{\mathcal{L}}(x))\right] = \mathrm{bruit}(x) + \mathrm{biais}^2(x) + \mathrm{var}(x) \tag{4}$$

où

$$\mathrm{bruit}(x) = \mathrm{Err}(\varphi_B(x))$$
$$\mathrm{biais}^2(x) = \left(\varphi_B(x) - \mathbb{E}_{\mathcal{L}}\left[\varphi_{\mathcal{L}}(x)\right]\right)^2$$
$$\mathrm{var}(x) = \mathbb{E}_{\mathcal{L}}\left[\left(\varphi_{\mathcal{L}}(x) - \mathbb{E}_{\mathcal{L}}[\varphi_{\mathcal{L}}(x)]\right)^2\right]$$

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

BIAIS-VARIANCE : DILEMME

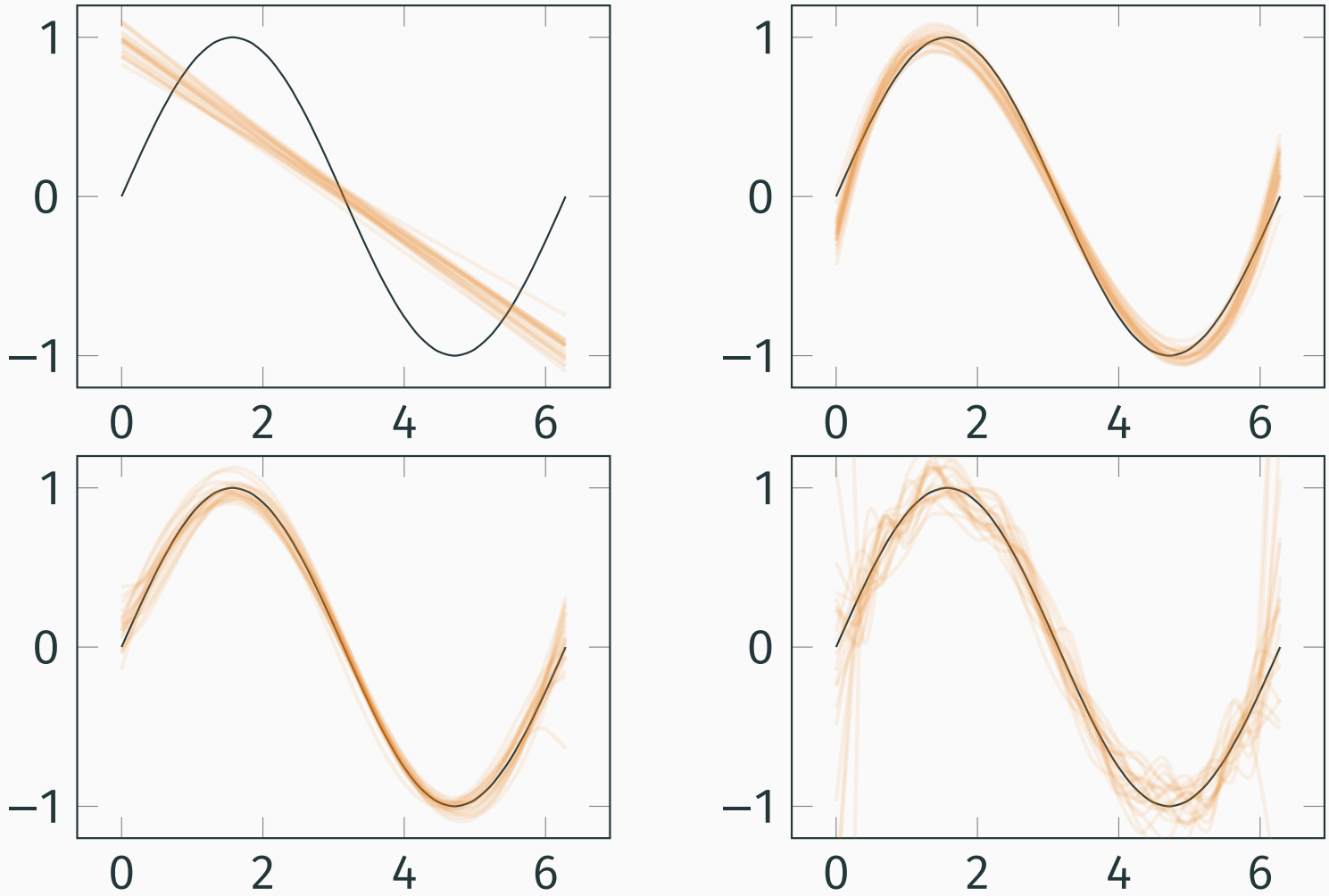


FIG. 2 : Apprentissage de $\sin(x)$ sur $[0\,,2\pi]$ à l'aide de polynômes de degrés 1,3,5 et 20.

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

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

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

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

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

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

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

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

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

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

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

Minimiser une perte différentiable $\Rightarrow$ descente de gradient. On entraine alors les classifieurs successifs sur le gradient.

STACKING

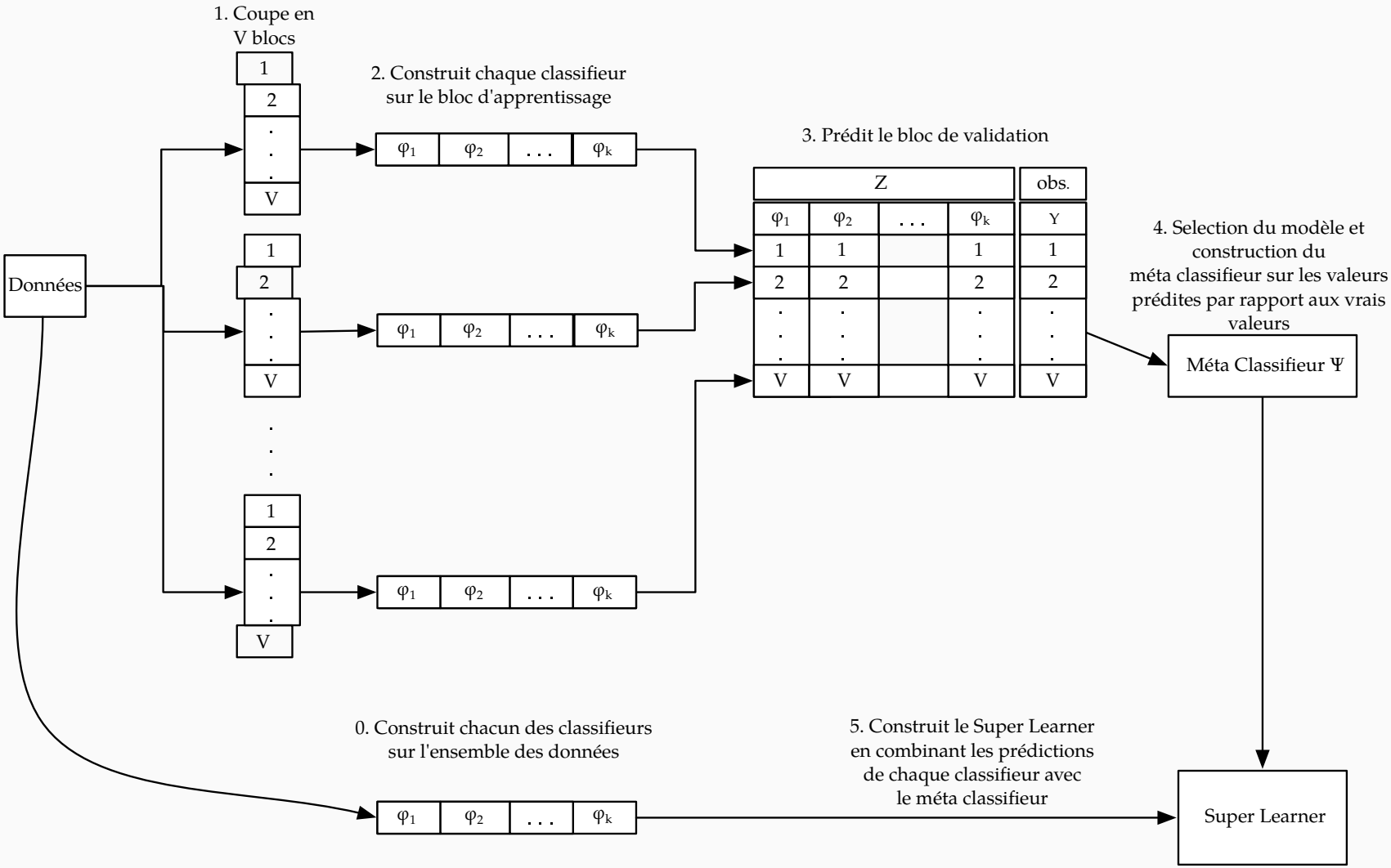


FIG. 3 : Fonctionnement de l'algorithme SuperLearner, adapté de van der LAAN et al. (2007)

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

$$\mathbb{P}(\zeta \mid \mathcal{L}) = \sum_{m=1}^{M} \mathbb{P}(\zeta \mid \varphi_m, \mathcal{L}) \mathbb{P}(\varphi \mid \mathcal{L}) \tag{5}$$

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

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

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

$$\mathbb{P}(\varphi \mid \mathcal{L}) \propto \mathbb{P}(\varphi_m) \mathbb{P}(\mathcal{L} \mid \varphi_m) \tag{7}$$

$$\propto \mathbb{P}(\varphi_m) \int \mathbb{P}(\mathcal{L} \mid \theta_m, \varphi_m) \mathbb{P}(\theta_m \mid \varphi_m) \, \mathrm{d}\theta_m \tag{8}$$

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

COMBINAISONS BAYÉSIENNES : UNE CORRECTION

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 $\mathcal{M}$, $C(\mathcal{M})$.

$$\mathbb{P}(y_i \mid x_i, \mathcal{L}) = \sum_{c \in C(\mathcal{M})} \mathbb{P}(y_i \mid x_i, \varphi_m, \mathcal{L}) \mathbb{P}(c \mid \mathcal{L}) \tag{9}$$

ARBRES ET FORÊTS

Arbres de classification : pourquoi ?

- 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 $\varphi : \mathcal{X} \to \mathcal{Y}$ 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 $t_i, \ldots, t_j$ de t représentent alors une partition disjointe de $\mathcal{X}_t$.

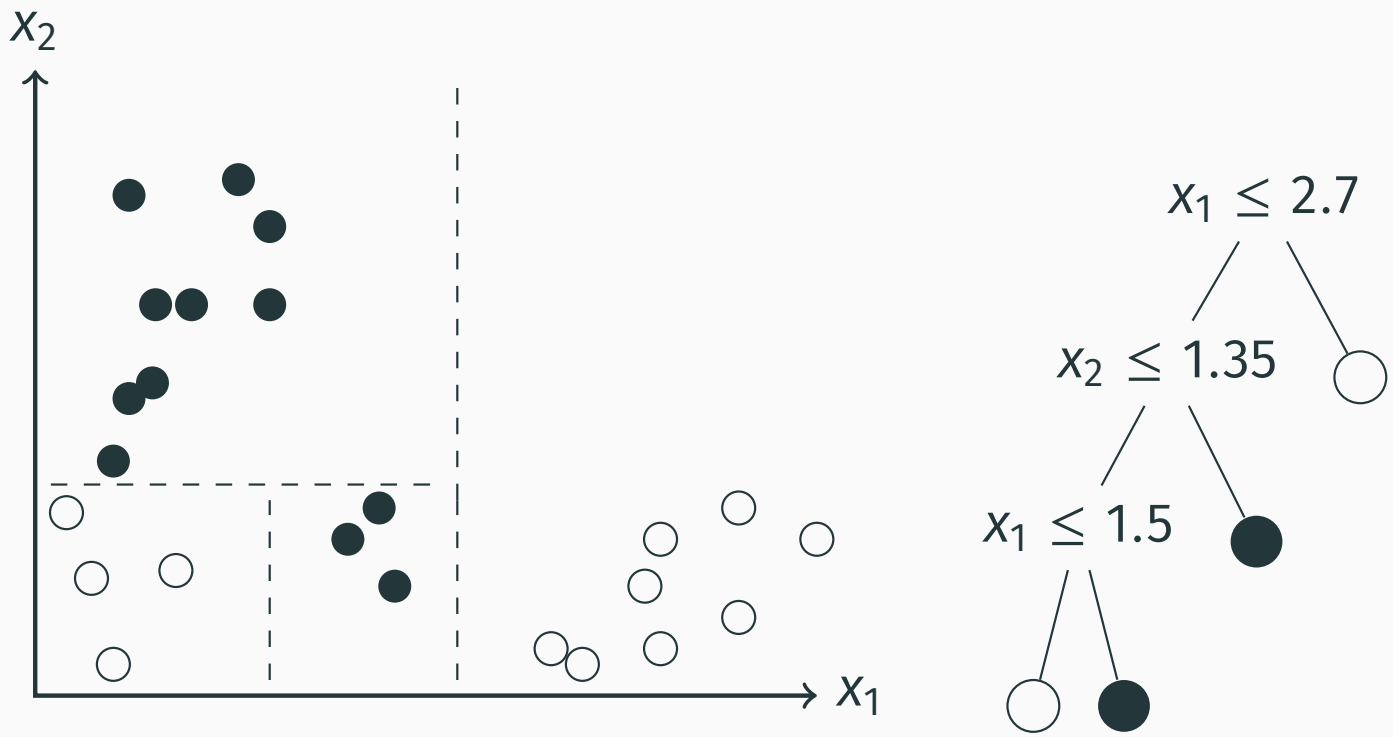


Fig. 4 : Équivalence entre la partition du plan et un arbre de décision binaire

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 :

$$\begin{cases} s^* = \underset{\substack{s_j^* \\ j=1,\dots,p}}{\operatorname{argmax}} \Delta i(s_j^*, t) \\ s_j^* = \underset{\substack{s \in Q(X_j) \\ \mathcal{L}_{t_L}, \mathcal{L}_{t_R} \neq \emptyset}}{\operatorname{argmax}} \Delta i(s,t) \end{cases}$$

Arbres de classification : exemples d'impuretés

Classification

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

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

Regression

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

Forêts de Breiman

- 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.

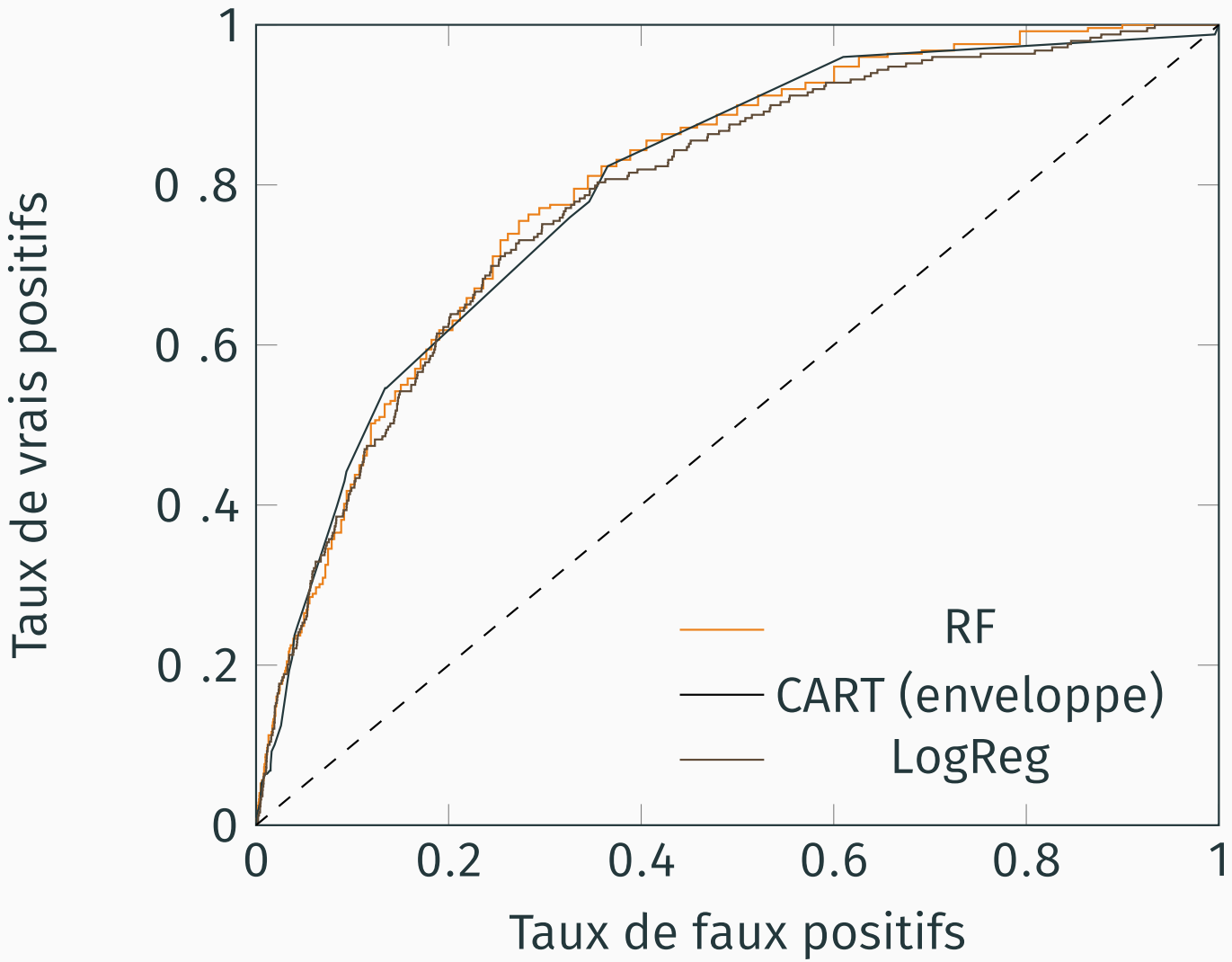


FIG. 5 : Performances de RF ($\mathrm{AUC} = 0\ .805$) par rapport à la régression logistique ($\mathrm{AUC} = 0\ .7889$) et un arbre seul ($\mathrm{AUC} = 0\ .79649$).

FORÊTS BAYÉSIENNES

- 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.

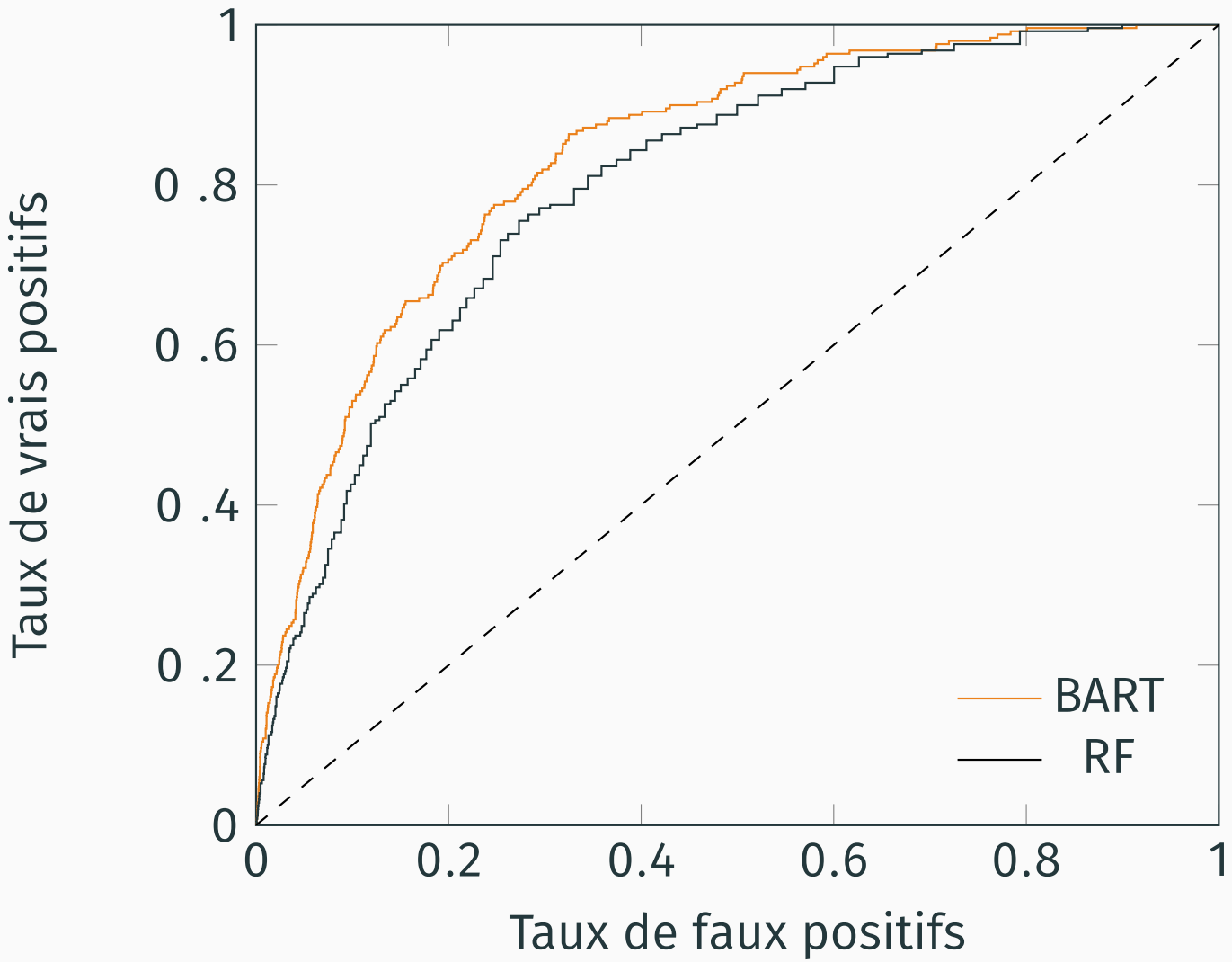


FIG. 6 : Performances de BART ($\mathrm{AUC} = 0\ .8365$).

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 :

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

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.

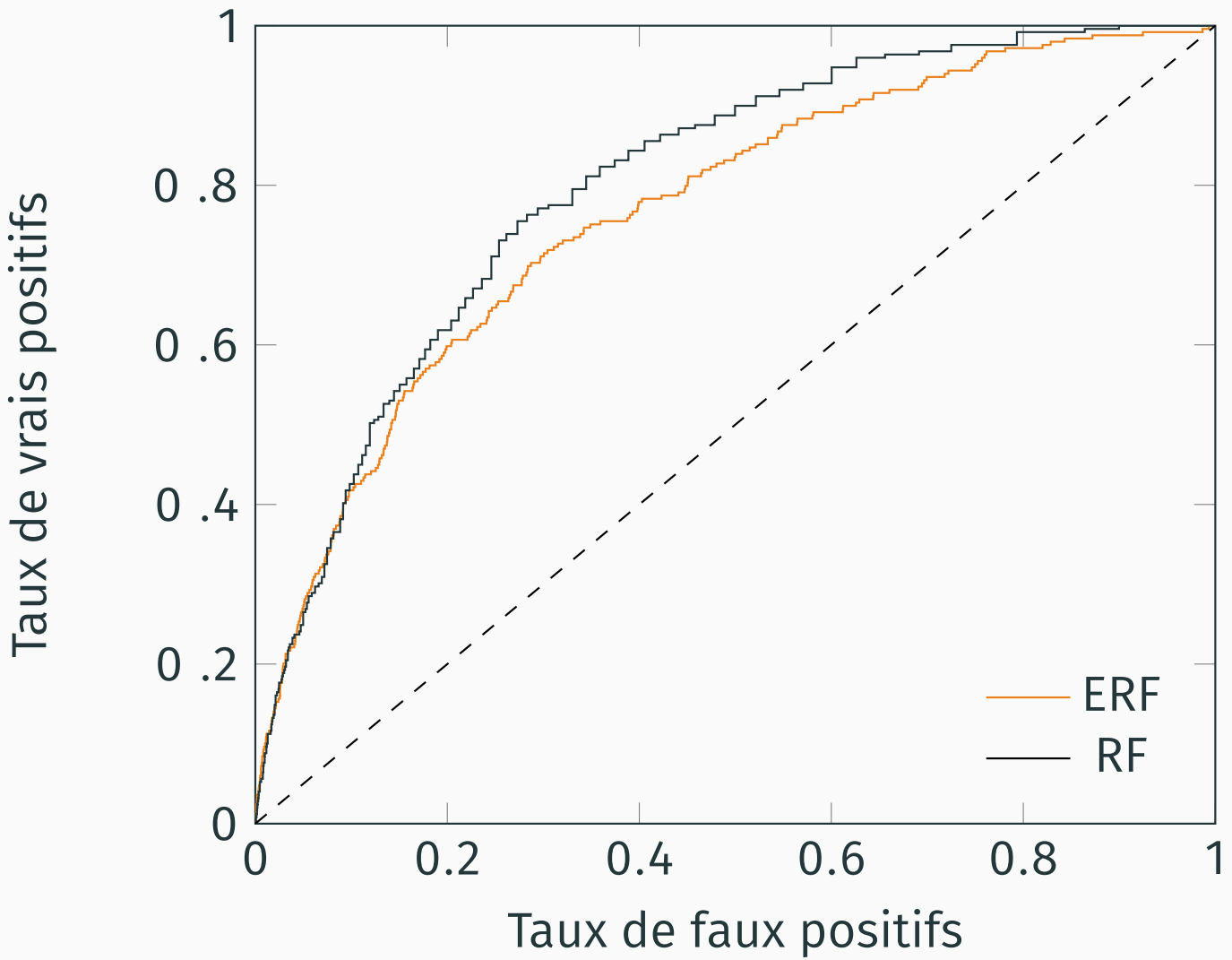


Fig. 7 : Performances de ERF ($\mathrm{AUC} = 0\ .7664$).

Apprentissage des feuilles : optimisation de l'AUC

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

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

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

$$\mathrm{AUC}(\varphi, \mathcal{L}) = \frac{\sum_{k_0 \in \mathcal{L}_\cdot} \sum_{k_1 \in \mathcal{L}_\infty} f\left(\varphi(x_1) - \varphi(x_0)\right)}{|\mathcal{L}_0||\mathcal{L}_1|}$$

avec f dérivable.

DONNÉES DÉSÉQUILIBRÉES

Forêts aléatoires équilibrées

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.

Forêts aléatoires équilibrées : performances

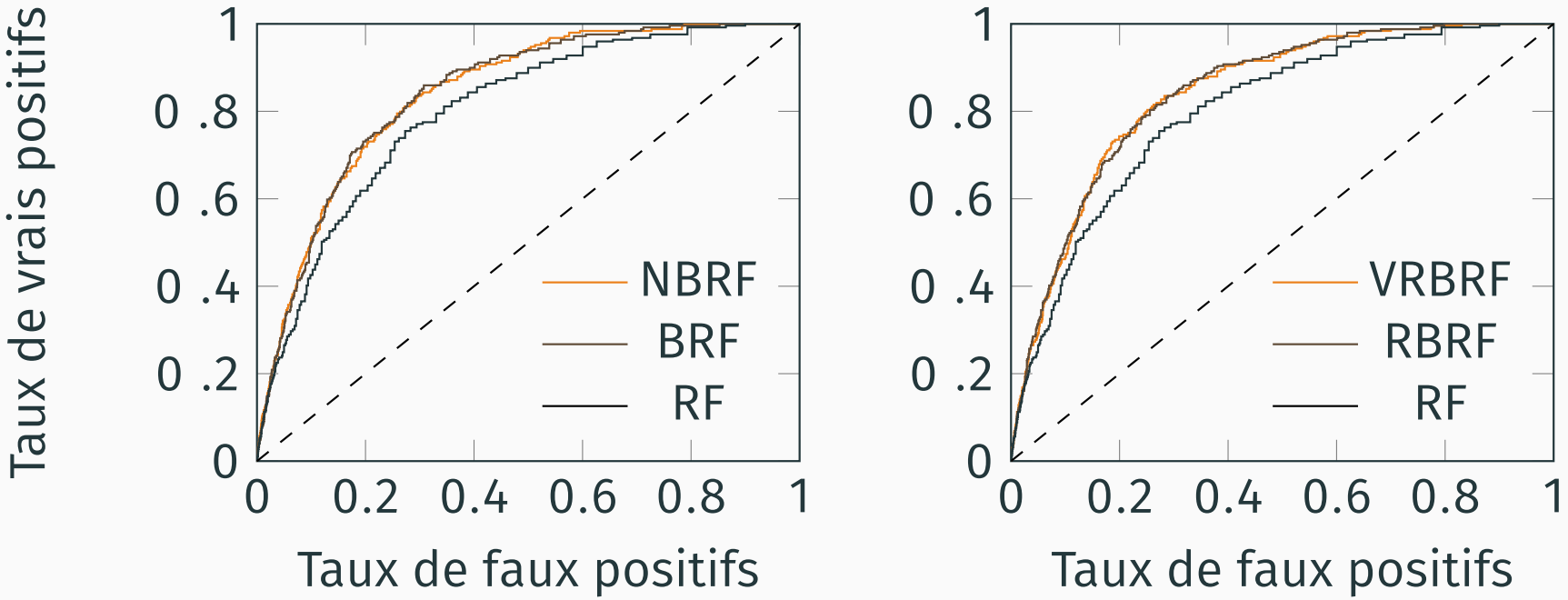


Fig. 8 : Performances de BRF ($\mathrm{AUC} = 0\ .8432$), NBRF ($\mathrm{AUC} = 0\ .8427$), RBRF ($\mathrm{AUC} = 0\ .8432$) et VRBRF ($\mathrm{AUC} = 0\ .8421$)

META-COST

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

$$R(i \mid x) = \sum_j \mathbb{P}(j \mid x) C(i, j) \tag{10}$$

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 10
3. Remplacer dans l'échantillon les vraies étiquettes par les étiquettes « idéales » trouvées précédemment.
4. Apprendre notre classifieur final sur ce nouvel échantillon.

META-COST : PERFORMANCES

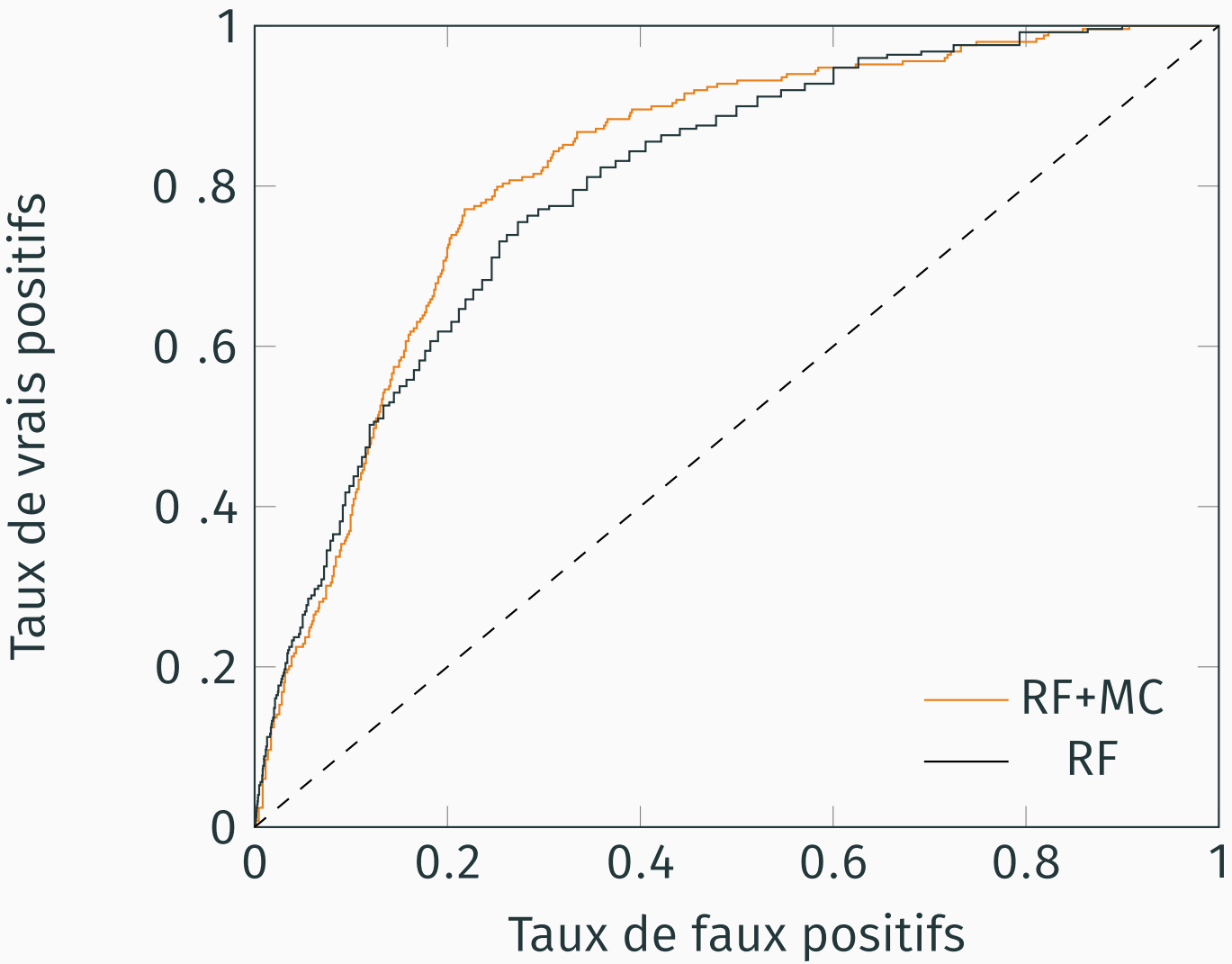


Fig. 9 : Performances de MetaCost + RF ($\mathrm{AUC} = 0\ .8217$).

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 $\sim$ 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

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

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 $k = 1, \ldots, K$ **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

Lemme (Johnson-Lindenstrauss en distribution)

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

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

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**
      $\tilde{X}_{i,n} \leftarrow A_{i,n}^{\mathsf{T}} X_n$ avec $A_{i,n} \sim \mu(\mathcal{L}_{i,n})$
      $s_{i,n}^{*} \leftarrow \mathrm{MeilleureCoupure}(\tilde{X}_{i,n})$
      Coupe $\mathcal{L}_{i,n}$ en $\mathcal{L}_{l_i,n}$ et $\mathcal{L}_{r_i,n}$ selon la règle de coupure $s_{i,n}^{*}$
    fin pour
  fin pour
  renvoyer $\left(\mathrm{Arbres_n}, (A_{i,n})\right)$
fin procedure

Algorithme 3 Algorithme Metacost pour les forêts aléatoires

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

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

$$p(\theta_i)g(\theta_{i-1} \mid \theta_i) = p(\theta_{i-1})g(\theta_i \mid \theta_{i-1}) \tag{11}$$

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

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

Théorème (Théorème Ergodique)

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

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

$$\forall i, j \; g\left(\hat{\theta}_j \mid \hat{\theta}_j\right) > 0$$

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

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

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

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

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

Algorithme 4 Metropolis-Hastings

procedure METROPOLIS-HASTINGS(q , θ_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 $u < \min\left\{1, \frac{p(\theta')q(\theta_i|\theta')}{p(\theta_i)q(\theta'|\theta_i)}\right\}$ **alors**
      $\theta_{i+1} = \theta'$
    sinon
      $\theta_{i+1} = \theta_i$
    fin si
  fin pour
  renvoyer $(\theta_0\,, \cdots\,, \theta_{n-1})$
fin procedure

On a ici que

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

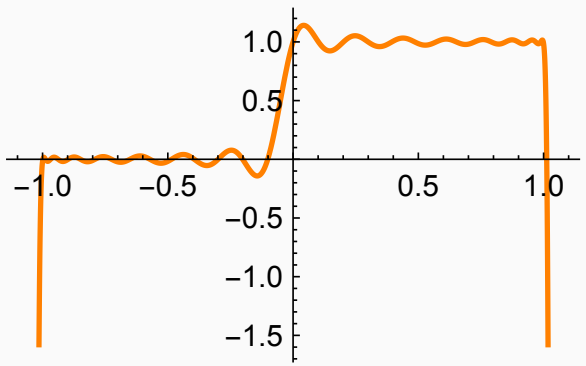
Avec $R\theta_i$ le terme de rejet :

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

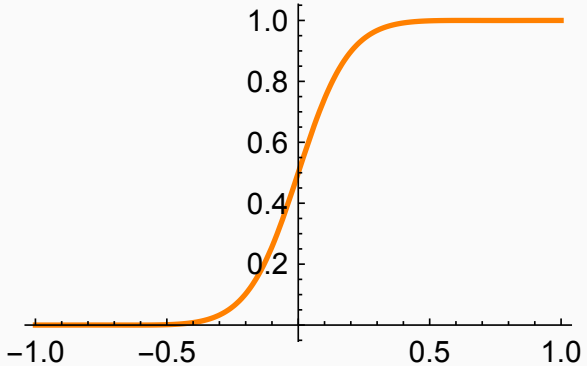
Par construction g remplis la propriété 11. De plus, le terme de rejet garantit l'apériodicité et il suffit que le support de q inclue le support de p pour garantir l'irréductibilité.

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.

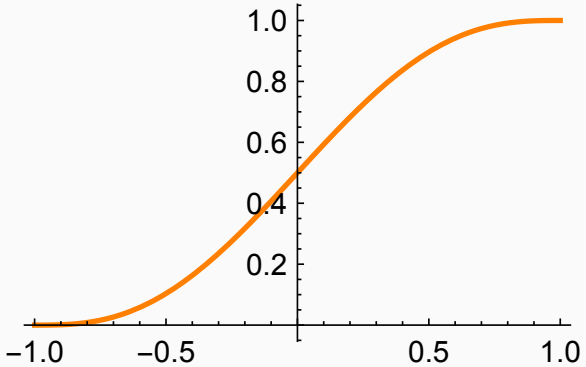
Problèmes avec l'approximation de l'AUC



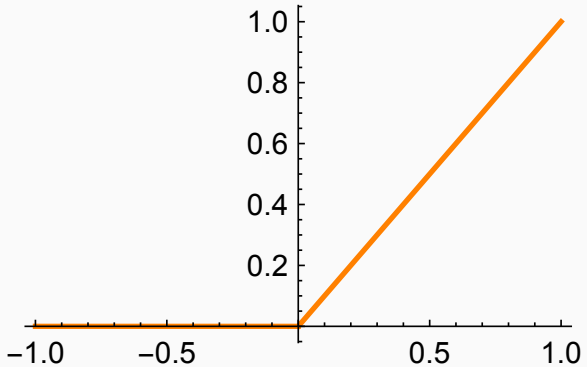
Approx. polynomiale



Approx. rationelle



$\frac{1}{2}+\frac{15x}{16}-\frac{5x^3}{8}+\frac{3x^5}{16}$



Éventuel substitut