Les forêts, dans la nature, sont sensibles à différents paramètres et procèdent par sélection. Et Mathématiquement?

C. Tuleau-Malot

Université de Nice - Sophia Antipolis

11 janvier 2018 - Paris - Journée de statistique Mathématique

Plan

Forêts aléatoires

Les forêts aléatoires : Léo Breiman et Adèle Cutler (2001)

$\Rightarrow$ amélioration de l'algorithme CART avec l'introduction de deux aléas

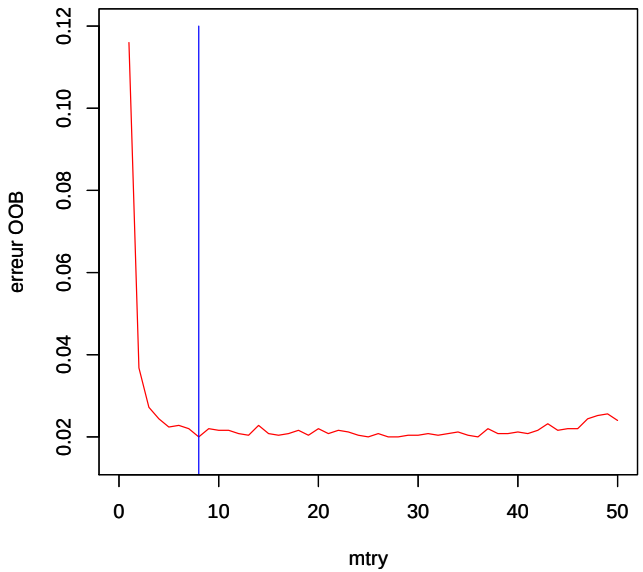
pratiquement :

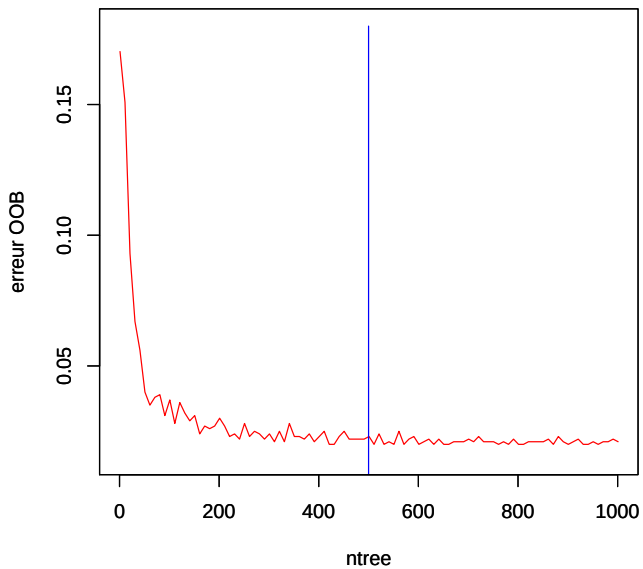
- ntree : nombre d'arbres dans la forêts
- mtry : nombre de variables sélectionnées à chaque noeud dans la construction d'un arbre

$\Rightarrow$ Comment choisir ces deux paramètres?

Influence de mtry et ntree

La question du choix de mtry et ntree est importante :





erreur out-of-bag : $\frac{1}{n}\sum_{i=1}^{n} 1_{Y_i \neq \hat{Y}_i}$

avec $\hat{Y}_i$ la prédiction de Y pour l'individu en agrégeant uniquement les prédictions liées aux arbres n'ayant pas utilisés l'individu i pour leur construction

Évolution des données

Recommandations de Breiman

- Régression : mtry$=\frac{p}{3}$
- Classification : mtry$=\sqrt{p}$

Oui mais évolution des données au cours des années avec

- Émergence des données en grande dimension ($p > n$ et $p >> n$)
- Big Data

$\Rightarrow$ est ce que les recommandations sont toujours les mêmes?
$\Rightarrow$ comment les forêts aléatoires passent à l'échelle des big data?

Données

Toys data de Weston et al. (2003) : jeu de données comprenant 100 observations de (X, Y)

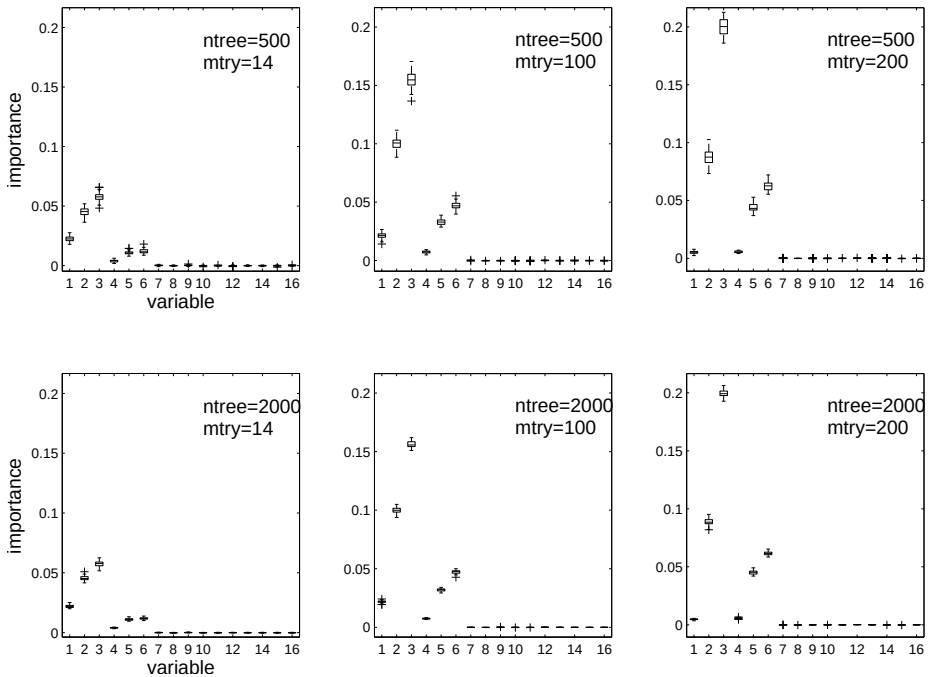
- $Y \in \{-1, 1\}$
- X : vecteur de p variables explicatives comprenant 6 vraies variables $(X^{(1)}, \ldots, X^{(6)})$, les autres étant des variables de bruit

Définition des $X^{(i)}$ conditionnellement à $Y = y$:

- pour 70% des données, $X^{(i)} \sim y\mathcal{N}(i, 1)$ pour $i \in \{1, 2, 3\}$ et $X^{(i)} \sim y\mathcal{N}(0, 1)$ pour $i \in \{4, 5, 6\}$
- pour les 30% restant, $X^{(i)} \sim y\mathcal{N}(0, 1)$ pour $i \in \{1, 2, 3\}$ et $X^{(i)} \sim y\mathcal{N}(i - 3, 1)$ pour $i \in \{4, 5, 6\}$
- pour $i \in \{7, \ldots, p\}$ des gaussiennes indépendantes

Influence of mtry and ntree on variable importance

($n = 100$ et $p = 200$)



$$I(X^{(j)}) = \frac{1}{ntree} \sum_{t=1}^{ntree} (errÕOB_t - errOOB_t)$$

où $errOOB_t$ est l'erreur out-of-bag de l'arbre numéro t et $errÕOB_t$ est l'erreur out-of-bag de l'arbre numéro t, mais avec permutation des observations de la variable $X^{(j)}$.

Influence of mtry and ntree on variable importance (2)

- l'influence d'une grande valeur pour mtry est évidente : l'ordre de grandeur de l'importance des variables est presque doublée suite au passage de mtry=14 à mtry=100.
- l'effet de ntree : avec ntree=2000, il y a une meilleure stabilité

D'autres paramètres peuvent influencer notamment l'importance des variables, comme les corrélations entre les variables explicatives.

Forêts aléatoires et Big Data

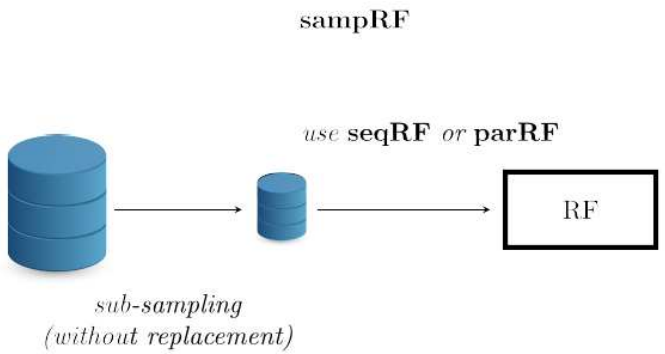
Les forêts aléatoires, de par leur usage fréquent dans différents domaines, ont déjà été transposées au cadre des données massives.
$\Rightarrow$ Comment?

Il a fallu faire des adaptations car problème de temps de calcul notamment et surtout de calcul.

Différentes stratégies ont été proposées.

stratégies

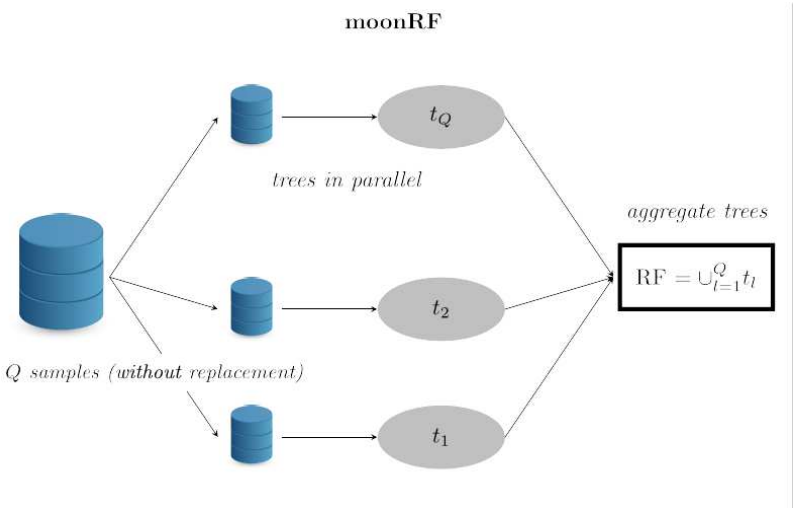
• sous-échantillonnage : idée que pas besoin des n données et donc on en prélève simplement un nombre m



stratégies

• sous-échantillonnage : idée que pas besoin des n données et donc on en prélève simplement un nombre m

• bootstrap m out of n : de nouveau on considère m observations différentes

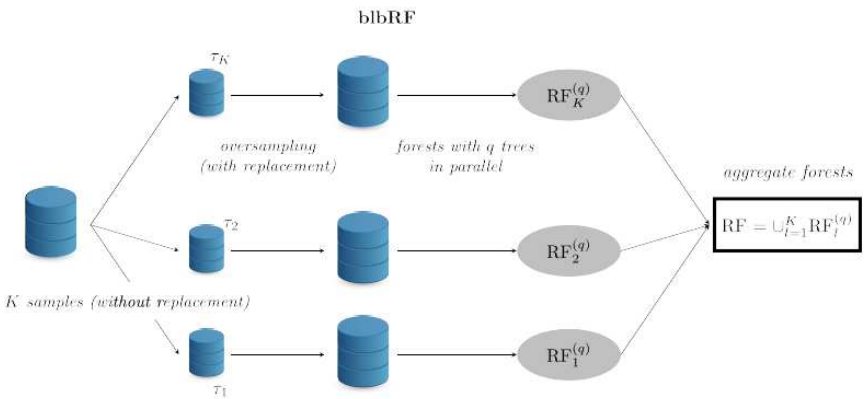


stratégies

• sous-échantillonnage : idée que pas besoin des n données et donc on en prélève simplement un nombre m

• bootstrap m out of n : de nouveau on considère m observations différentes

• bag of little bootstrap : échantillon de taille n mais avec simplement m valeurs différentes



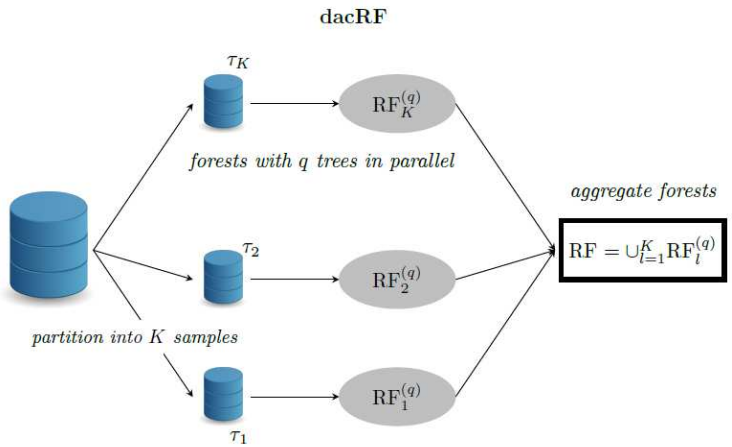
stratégies

• sous-échantillonnage : idée que pas besoin des n données et donc on en prélève simplement un nombre m

• bootstrap m out of n : de nouveau on considère m observations différentes

• bag of little bootstrap : échantillon de taille n mais avec simplement m valeurs différentes

• version MapReduce : division du problème en sous-problèmes de taille plus faible



erreur out-of-bag et importance des variables pour les variantes

Pb : les notions d'erreur out-of-bag et d'importance des variables doivent être transformées!

$$\Rightarrow \text{approximations}$$

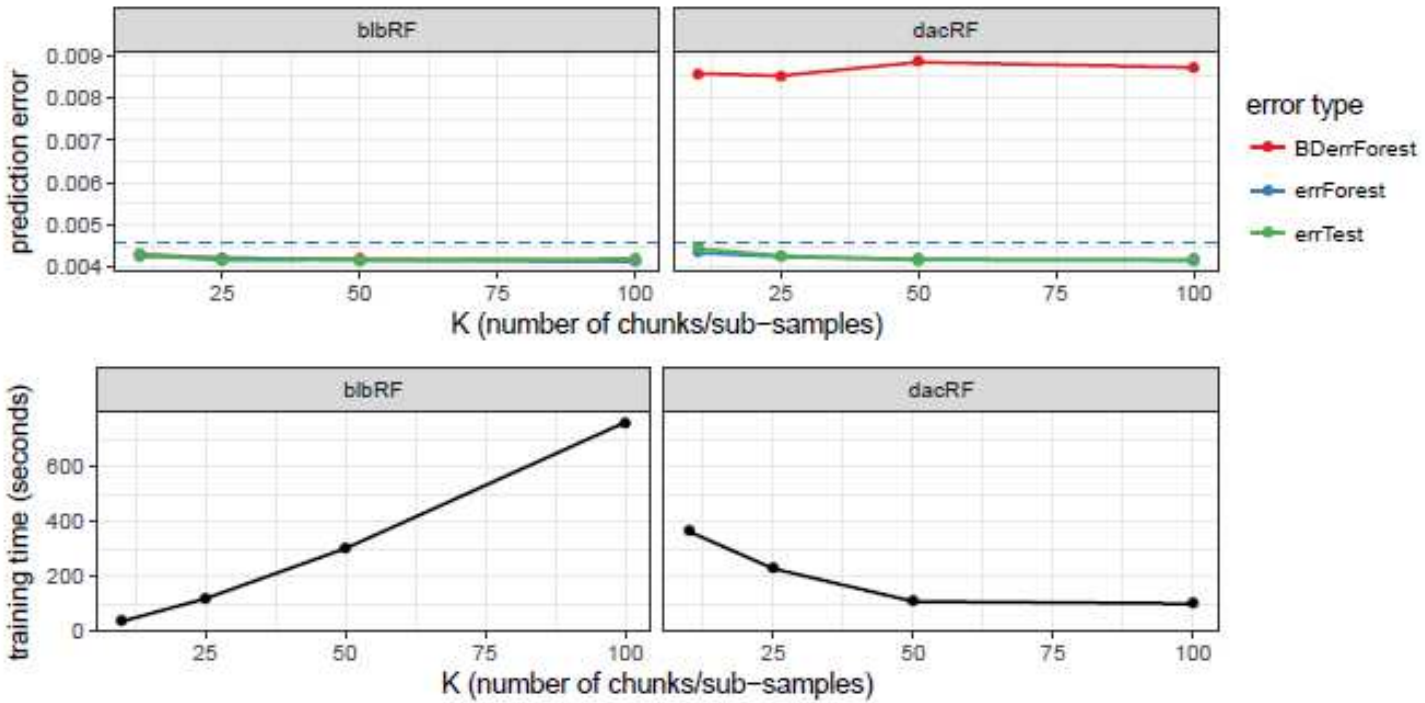
- dacRF : $\text{BDerrForest} = \frac{1}{n}\sum_{l=1}^{K} m.errForest^l$ avec $errForest^l = \frac{1}{m}\sum_{i\in\tau_l} 1_{y_i \neq \hat{y}_i^l}$
- sampRF : $\text{BDerrForest} = errForest^1$
- moonRF : même principe que usuellement mais en se restreignant à une sous-partie des observations
- blbRF : une approximation existe

On peut de même en déduire des variantes pour l'importance des variables.

comparaison des variantes

Toys data avec $n = 15000000$

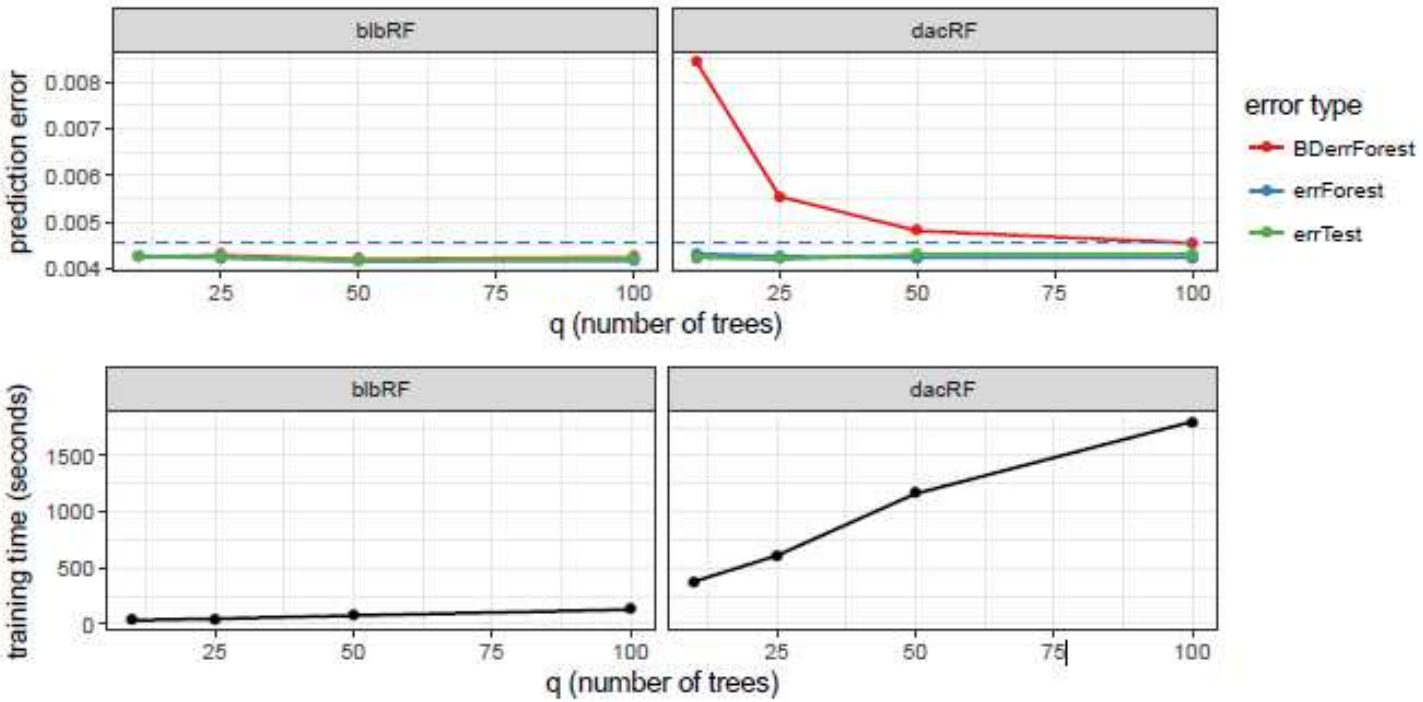
La comparaison des stratégies va se faire selon deux critères : erreur et le temps



comparaison des variantes

- on constate que l'erreur de prédiction n'est pas sensible à la valeur de K avec q=10
- l'approximation proposée est pessimiste dans le cas de la variante dacRF mais par contre bonne pour blbRF
- le temps de calcul augmente avec K pour blbRF et décroit pour dacRF

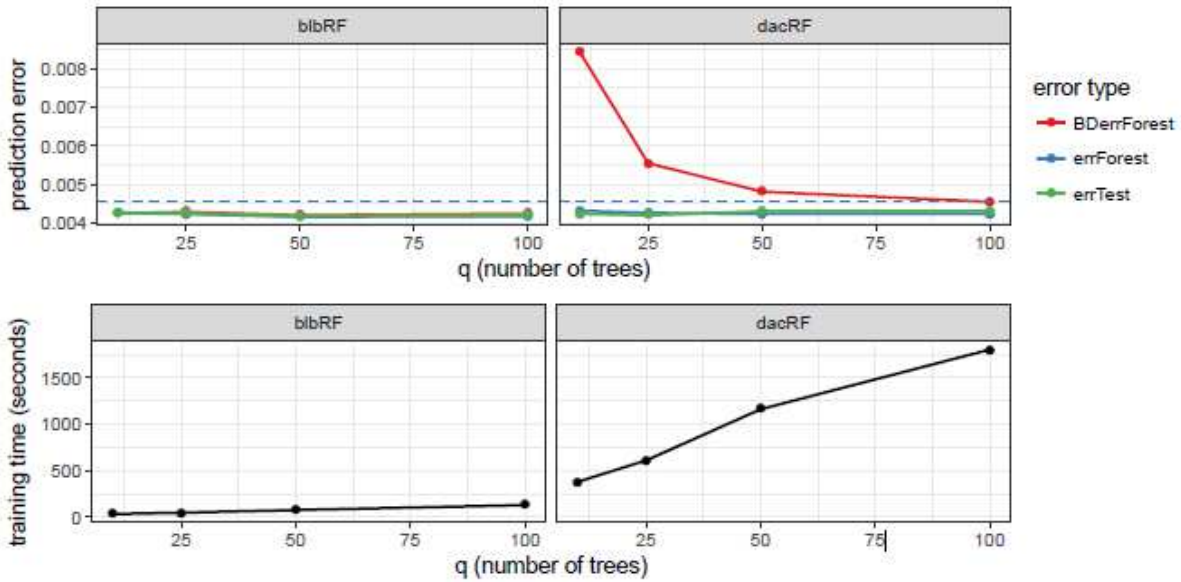
comparaison des variantes (2)



comparaison des variantes (2)

- Encore une fois, blbRF semble plus robuste
- Pour avoir une approximation non biaisée pour dacRF, il faut un grand nombre d'arbres dans chaque sous-forêt, mais avec pour conséquence une augmentation drastique du temps de calcul

comparaison des variantes (3)



comparaison des variantes (3)

- pour sampRF, dès que $\frac{m}{n}$ est suffisamment grand, BDerrorforest semble être non biaisée
- dans le même temps, si $\frac{m}{n}$ devient grand, le temps de calcul augmente fortement pour sampRF, même si ce dernier reste raisonnable si l'on compare à dacRF avec q=10 et K=100
- pour moonRF, DBerrorforest semble être non biaisée

Conclusions

Quelques conclusions :

- le temps d'exécution est réduit avec les variantes par rapport à la méthode séquentielle
- la méthode la plus rapide consiste à extraire un échantillon de petite taille
- la méthode la lus gourmande en temps est la méthode basée sur le MapReduce avec q=10 et K=100
- l'erreur de prédiction est mieux prédite par errforest que par DBerrorforest