# De la sélection d'arbres de décision dans les forêts aléatoires

Laurent Heutte, Simon Bernard, Sébastien Adam, Émilie Oliveira

# De la Selection d'Arbres de Décision dans les Forêts Aléatoires

Laurent Heutte – Simon Bernard – Sébastien Adam – Emilie Oliveira

Université de Rouen, LITIS EA 4108
BP 12 - 76801 Saint-Etienne du Rouvray, France.

{laurent.heutte,simon.bernard,sebastien.adam}@univ-rouen.fr

**Résumé** : *Dans cet article nous présentons une étude sur une nouvelle famille de méthodes d'Ensembles de Classifieurs, appelée Forêts Aléatoires (RF pour Random Forest). Dans un processus d'induction de forêts aléatoires "traditionnel", un nombre préalablement fixé d'arbres de décision est généré, à l'aide notamment de principes d'apprentissage partiellement aléatoires. Ce type de processus présente deux principaux inconvénients : i) le nombre d'arbres doit être fixé a priori ii) l'interprétabilité et les capacités d'analyse offertes par les classifieurs de type arbres de décisions sont perdues, du fait de l'utilisation de principes de "randomisation" au cours de leur induction. Ces constatations soulèvent alors deux questions : ce type de forêts aléatoires contient-elle des arbres de décision qui détériorent les performances de l'ensemble ? Si oui, ces arbres présentent-ils des propriétés particulières qui pourraient expliquer cette perte de performances ? Pour répondre à ces questions, nous abordons cette problématique comme un problème de sélection de classifieurs, et montrons que de meilleurs sous-ensembles d'arbres de décision peuvent être obtenus en utilisant des méthodes sous-optimales de sélection de classifieurs. Les résultats prouvent notamment qu'un algorithme d'induction de forêts aléatoires "classique" n'est pas la meilleure approche pour produire des classifieurs de type forêts aléatoires qui soient performants.*

**Mots-clés** : Ensemble de Classifieurs, Sélection de Classifieurs, Forêts Aléatoires, Arbres de Décision.

## 1 Introduction

Un des principaux enjeux de l'apprentissage automatique consiste à concevoir des systèmes de classification performants à partir d'un ensemble d'exemples représentatifs d'une population de données. Parmi les différentes approches permettant d'aborder ce type de problématique, combiner un ensemble de classifieurs individuels faibles pour former un unique système de classification — appelé Ensemble de Classifieurs — a suscité un intérêt grandissant de la communauté scientifique. Cet intérêt a généré de récents travaux de recherche qui ont montré que certains principes de combinaison de classifieurs sont particulièrement efficaces, tel que le Boosting [FRE 96] (ou Arcing [BRE 98]), le Bagging [BRE 96], le Random Subspaces [HO 98], ou plus récemment les Random Forests [BRE 01]. L'efficacité des combinaisons de classifieurs repose principalement sur leur capacité à tirer parti des complémentarités des classifieurs individuels, dans le but d'améliorer autant que possible les performances en généralisation de l'ensemble. Une explication de ce lien entre complémentarité et performances est donnée par la notion de diversité. Bien qu'il n'y ait pas dans la littérature de définition de la propriété de diversité sur laquelle tout le monde s'accorde [KUN 03], ce concept est usuellement reconnu comme étant l'une des plus importantes caractéristiques pour l'amélioration des performances en généralisation d'un ensemble de classifieurs [KUN 04]. On peut définir la diversité comme la capacité des classifieurs individuels d'un ensemble à être en accord sur les bonnes prédictions et en désaccord sur les erreurs de prédiction.

Parmi les différentes approches de construction d'ensembles de classifieurs, celles qui s'appuient sur l'aléatoire pour produire de la diversité se sont montrées particulièrement efficaces à l'image des méthodes de Bagging [BRE 96] ou de Random Subspaces [HO 98]. Ces deux méthodes introduisent l'aléatoire dans le processus d'induction, dans le but de construire des classifieurs de base différents les uns des autres, et ainsi de produire de la diversité dans l'ensemble. Récemment Leo Breiman a proposé une nouvelle famille de méthodes d'ensembles appelée Forêts Aléatoires (RF pour Random Forest) [BRE 01], basée sur ce concept de "randomisation". Les RF peuvent être définies comme un principe générique de combinaison de classifieurs, composée de $L$ classifieurs élémentaires de type arbres de décision et notée $\{h(x, \Theta_k),\ k = 1, ...L\}$, où $\{\Theta_k\}$ est une famille de vecteurs aléatoires, indépendants et identiquement distribués, et où $x$ représente une donnée d'entrée. La particularité de ce type de combinaison est que chaque arbre de décision est construit à partir d'une réalisation d'un vecteur aléatoire de paramètres. Une RF peut par exemple être construite en générant des sous-ensembles aléatoires de caractéristiques pour chaque arbre (comme dans la méthode Random Subspaces), et/ou en générant des sous-ensembles aléatoires de données d'apprentissage pour chaque arbre (comme dans la méthode de Bagging).

Depuis qu'elles ont été introduites en 2001, les RF ont beaucoup été étudiées, d'un point de vue théorique comme d'un point de vue expérimental [BER 07, BOI 05, BRE 01, BRE 04, CUT 01, GEU 06, LAT 01, ROD 06, ROB 04]. Dans la plupart de ces travaux, il a été montré que ces méthodes étaient particulièrement compétitives avec l'un des principes d'apprentissage les plus efficaces, *i.e.* le boosting [BRE 01, CUT 01, ROD 06]. Cependant les mécanismes qui expliquent ces bonnes performances n'ont pas encore été clairement identifiés. Par exemple, il a été mathématiquement prouvé dans [BRE 01] et expérimentalement confirmé

dans [LAT 01], qu'au delà d'un certain nombre d'arbres de décision, il n'était plus utile d'en ajouter à la forêt pour en améliorer les performances en généralisation. Cette affirmation concerne les processus d'induction qui utilisent l'aléatoire pour produire des arbres sans connaissance a priori sur leur caractéristiques intrinséques. Pourtant aucun travail de recherche à notre connaissance ne s'est intéressé aux mécanismes qui font qu'un ensemble d'arbres est plus ou moins performant qu'un autre utilisant plus ou moins d'arbres.

Dans cet article nous proposons d'apporter quelques élements pour aider à mieux comprendre ces mécanismes. Le but est de déterminer s'il est possible ou non de sélectionner un sous-ensemble d'arbres à partir d'une forêt, meilleur en terme de performances que l'ensemble initial. Notre but n'est pas ici de trouver le sous-ensemble optimal parmi un plus large ensemble d'arbres, mais plutôt d'étudier les propriétés de différents sous-ensembles en fonction de leur performances. De cette façon, nous souhaitons apporter des premiers éléments permettant d'identifier les propriétés remarquables partagées par les sous-forêts les plus performantes, obtenues au cours du processus de sélection. C'est la raison pour laquelle, comme nous l'expliquons dans la section 3, il n'est pas nécessaire ici de mettre en œuvre des techniques de sélection de classifieurs optimales. Nous avons donc utilisé deux méthodes de sélection simples *i.e.* SFS (Sequential Forward Selection) et SBS (Sequential Backward Selection) [HAO 03], et étudié ensuite les taux d'erreur en classification des sous-ensembles obtenus au cours de l'expérience. Nos résultats expérimentaux montrent que l'algorithme d'induction de RF "classique" n'est pas la meilleure approche pour produire des forêts performantes.

Cet article est donc organisé de la façon suivante : nous rappelons dans la section 2 le principe de l'algorithme d'induction de RF Forest-RI ; dans la section 3, nous commençons par expliquer notre approche de la sélection de classifieurs appliquée aux RF, et décrivons ensuite notre protocole expérimental, les bases de données utilisées, ainsi que les résultats obtenus avec les deux méthodes de sélection utilisées. Nous dressons finalement quelques conclusions et perspectives dans une dernière section.

## 2 L'algorithme Forest-RI

Le terme Forêts Aléatoires désigne une famille de méthodes de classification, composée de différents algorithmes d'induction d'ensemble d'abres de décision, tels que l'algorithme Forest-RI présenté par Breiman dans [BRE 01] et souvent cité dans la littérature comme la méthode d'induction de référence. Dans cet algorithme deux principes de "randomisation" sont utilisés : le Bagging et le Random Feature Selection. L'étape d'apprentissage consiste donc à construire un ensemble d'arbres de décision, chacun entraîné à partir d'un sous-ensemble "bootstrap" issu de l'ensemble d'apprentissage original — *i.e.* en utilisant le principe de Bagging — et à l'aide d'une méthode d'induction d'arbres appelée Random Tree. Cet algorithme d'induction, habituellement basé sur l'algorithme CART [BRE 84], modifie la procédure de partitionnement des nœuds de l'arbre, de sorte que la sélection de la caractéristique utilisée comme critère de partitionnement soit partiellement aléatoire. C'est-à-dire que pour chaque nœud de l'arbre, un sous-ensemble de caractéristiques est généré aléatoirement, à partir duquel le meilleur partitionnement est réalisé.

Pour résumer, dans la méthode Forest-RI, un arbre de décision est construit selon la procédure suivante :

- Pour $N$ données de l'ensemble d'apprentissage, tirer aléatoirement $N$ individus avec remise. L'ensemble résultant sera celui utilisé pour l'induction de l'arbre en question.
- Pour $M$ caractéristiques, un nombre $K << M$ est spécifié de sorte qu'à chaque nœud de l'arbre, un sous-ensemble de $K$ caractéristiques soit tiré aléatoirement, parmi lesquelles la meilleure est ensuite sélectionnée pour le partitionnement.
- L'arbre est ainsi construit jusqu'à atteindre sa taille maximale. Aucun élagage n'est réalisé.

Dans ce processus, l'induction de l'arbre est principalement dirigée par un hyperparamètre, *i.e.* le nombre $K$. Ce nombre permet d'introduire plus ou moins d'aléatoire dans l'induction. De cette façon, excepté quand $K = M$, auquel cas l'induction de l'arbre n'est pas du tout "randomisée", chaque arbre de la forêt présente une structure et des propriétés qui ne peuvent être appréhendées *a priori*. Avec l'introduction de l'aléatoire dans l'induction des RF, on espère tirer parti de la complémentarité des arbres, mais rien ne garantit qu'ajouter un arbre à la forêt permettra effectivement d'améliorer les performances de l'ensemble. On peut même imaginer que certains arbres détérioreront les performances en généralisation de l'ensemble. Cette idée nous a amenés à étudier la façon d'améliorer les performances d'une RF en ne sélectionnant qu'un sous-ensemble particulier de ses arbres.

Dans la littérature, quelques travaux de recherche seulement se sont intéressés au nombre d'arbres de décision à construire au sein d'une forêt. Quand Breiman a introduit le formalisme des RF dans [BRE 01], il démontra également qu'au delà d'un certain nombre d'arbres, en ajouter d'autres ne permettait pas systématiquement d'améliorer les performances de l'ensemble. Précisement, il établit que pour un nombre croissant d'arbres dans la forêt, l'erreur en généralisation converge vers une borne inférieure. Ce résultat indique que le nombre d'arbres d'une RF ne doit pas nécessairement être le plus grand possible pour produire un classifieur performant. Les travaux de Latinne et al. dans [LAT 01], ainsi que nos travaux dans [BER 07], ont expérimentalement confirmé cette affirmation. Cependant, admettre qu'au delà d'un certain nombre d'arbres les performances en généralisation se stabilisent ne signifie bien évidemment pas que les performances optimales ont été atteintes. Donc l'idée de nos expérimentations est d'établir si il est possible ou non d'obtenir un sous-ensemble d'arbres capable de surpasser la forêt initiale.

A noter que dans la suite de cet article, le terme forêt aléatoire (ou RF) designera toujours une forêt induite à l'aide de l'algorithme Forest-RI.

# 3 Sélection de Classifieurs et Forêts Aléatoires

Le principe de ce travail expérimental est d'appliquer des techniques de sélection de classifieurs à une RF d'un grand nombre d'arbres. Pour ce faire il nous faut choisir i) un critère de sélection et ii) une méthode de sélection.

En ce qui concerne le critère de sélection, deux principales approches sont proposées dans la littérature : l'approche "filter" et l'approche "wrapper" [KOH 97]. L'approche "filter" consiste à sélectionner un sous-ensemble de classifieurs à l'aide d'un critère d'évaluation *a priori* qui ne prend pas en compte les performances de l'ensemble. L'approche "wrapper" en revanche réalise une sélection de sous-ensembles de classifieurs en optimisant *a posteriori* les performances de l'ensemble. Notre but étant d'établir s'il est possible ou non de trouver un sous-ensemble d'arbres de décision plus performant que la forêt initiale, c'est l'approche "wrapper" qui a été adoptée pour ces expérimentations. La sélection de classifieurs a donc été réalisée en tentant d'optimiser les performances des sous-ensembles d'arbres résultants.

Concernant les méthodes de sélection maintenant, comme nous l'avons mentionné dans la section 1, notre but n'est pas de trouver un sous-ensemble optimal de classifieurs parmi un plus large ensemble d'arbres de décision, mais plutôt d'étudier les propriétés des différents sous-ensembles en fonction de leurs performances. Par conséquent l'optimalité des méthodes de sélection n'est pas une priorité ici. C'est la raison pour laquelle les deux algorithmes de sélection de classifieurs SFS (Sequential Forward Selection) et SBS (Sequential Backward Selection) ont été choisis. Ces deux méthodes sont bien connues pour être sous-optimales puisque la séquentialité du processus de sélection rend le résultat de chaque itération dépendant de l'itération précédente, et de cette façon toutes les solutions ne sont pas explorées. Cependant ces méthodes présentent l'avantage d'être simples et rapides. Ces deux techniques de sélection construisent de façon itérative un sous-ensemble de classifieurs sous-optimal à partir d'un ensemble plus important [HAO 03]. A chaque itération de la procédure SFS par exemple, un classifieur individuel est sélectionné parmi les classifieurs restants dans l'ensemble d'origine, de sorte que sa contribution — en termes de gain de performances — au sous-ensemble courant soit maximale. De la même manière, chaque itération de la procédure SBS consiste à éliminer du sous-ensemble courant le classifieur qui contribue le moins à ses performances. Le critère d'arrêt d'un tel processus itératif est généralement basé sur la convergence des performances, mais il peut également être défini par un nombre maximum d'itérations de façon à fixer le nombre de classifieurs contenus dans le sous-ensemble final [ROL 01]. Pour nos expérimentations nous avons décidé de laisser les deux processus de sélection explorer toutes les itérations possibles, *i.e.* pour un nombre $L'$ de classifieurs dans les sous-ensembles obtenus, allant de 1 à $L$, où $L$ représente le nombre d'arbres dans la forêt initiale. De cette façon nous avons la possibilité d'étudier l'évolution des performances des RF en fonction du nombre d'arbres qu'elles contiennent.

## 3.1 Bases de données

Les 10 bases de données qui ont été utilisées dans nos expérimentations sont décrites dans le tableau 1 : les 7 premières de ces bases ont été sélectionnées parmi les bases du dépot de l'UCI [ASU 07] ; Twonorm et Ringnorm sont deux bases de données synthétiques conçues par Leo Breiman [BRE 98] ; et la base de données MNIST [LEC 98] est une base de chiffres manuscrits sur lesquels ont été extraites des caractéristiques basées sur une pyramide multi-résolution des niveaux de gris des images comme expliqué dans [BER 07]. Ces bases de données ont été sélectionnées dans un premier temps parce qu'elles sont représentatives des problématiques d'apprentissage automatique en termes de nombre de classes, de nombre de caractéristiques et de nombres de données. Elles ont également été choisies car elles ne contiennent pas de valeur manquante et que les caractéristiques sont toutes essentiellement numériques. Enfin pendant toutes nos expérimentations, les performances des RF se sont montrées très sensibles à la taille de l'ensemble d'apprentissage. Puisque notre but n'est pas d'approfondir ce point, seules les bases de données avec un nombre suffisant de données ont été utilisées.

TAB. 1 – Description des bases de données

| Bases | Taille | Caract | Classes |
|---|---|---|---|
| Gamma | 19020 | 10 | 2 |
| Letter | 20000 | 16 | 26 |
| Pendigits | 10992 | 16 | 10 |
| Segment | 2310 | 19 | 7 |
| Spambase | 4610 | 57 | 2 |
| Vehicle | 946 | 18 | 4 |
| Waveform | 5000 | 40 | 3 |
| Ringnorm | 7400 | 20 | 2 |
| Twonorm | 7400 | 20 | 2 |
| Mnist | 60000 | 84 | 10 |

A noter que pour les expérimentations décrites dans cette section nous avons séparé aléatoirement les bases de données, avec deux tiers des données destinées à l'apprentissage et le tiers restant au test.

## 3.2 Protocole Expérimental

Nos expérimentations ont donc consisté à mettre en œuvre les deux méthodes de sélection de classifieurs présentées précédemment, et à les appliquer à un large ensemble d'arbres de décision générés par l'algorithme Forest-RI. Le but étant de visualiser et d'étudier l'évolution du taux d'erreur de chaque sous-ensemble obtenu durant les processus de sélection sur les ensembles de test, le protocole expérimental exact est décrit dans cette partie.

Premièrement, chaque base de données a été divisée en deux sous-ensembles de données comme mentionné dans la section précédente ; un pour l'apprentissage et l'autre pour le test. La séparation des données a été réalisée par tirage aléatoire, avec respectivement deux tiers des données pour l'apprentissage et le tiers restant pour le test. Comme nous l'avons déjà expliqué, notre but est d'étudier l'évolution

des performances des forêts en fonction du nombre d'arbres qu'elles contiennent. Donc seule une séparation a été réalisée pour chaque base de données. On note les sous-ensembles résultants par $T = (T_r, T_s)$ où $T_r$ et $T_s$ représentent respectivement les ensembles d'apprentissage et de test.

Une RF a ensuite été induite à partir des données de $T_r$, avec un nombre $L$ d'arbres fixé à 300. La valeur de l'hyperparamètre $K$ a été fixée à $\sqrt{M}$, qui est une valeur par défaut communément utilisée dans la littérature. Un précédent travail sur la paramétrisation des RF, présenté dans [BER 08] a montré que cette valeur de $K$ est un bon compromis pour produire une forêt performante. Les méthodes SFS et SBS ont alors été appliquées sur cette forêt de 300 arbres, de sorte qu'à chaque itération un arbre est ajouté (SFS) ou retiré (SBS) au sous-ensemble courant si sa contribution — en termes de taux d'erreur — à celui-ci est minimale (SFS) ou maximale (SBS). Une troisième méthode de sélection a été mise en œuvre. Elle ajoute itérativement un arbre au sous-ensemble courant en le sélectionnant aléatoirement parmi les arbres restants dans la forêt initiale. Ce processus de sélection, que nous notons SRS (pour Sequential Random Selection), permet de simuler des inductions répétitives de RF avec l'algorithme Forest-RI, pour un nombre croissant d'arbres allant de 1 à $L$. Ainsi, trois tableaux de $L$ valeurs de taux d'erreur ont été obtenus avec cette démarche expérimentale, et ce pour chaque base de données.

**Algorithm 1** Protocole Expérimental

**ENTRÉES:** $N$ le nombre de données disponibles. $M$ la dimension de l'espace de description.

Tirer aléatoirement sans remise $\frac{2}{3} \times N$ des données de la base pour former l'ensemble d'apprentissage $T_r$. Les données restantes forment alors l'ensemble de test $T_s$.

$h \leftarrow$ Forest-RI($L = 300$,$K = \sqrt{M}$,$T_r$).

$h^{(0)}_{SFS} \leftarrow \emptyset$.

$h^{(0)}_{SBS} \leftarrow h$.

$h^{(0)}_{SRS} \leftarrow \emptyset$.

**pour** $i = 1$ *to* $L$ **faire**

$h^{(i)}_{SFS} \leftarrow h^{(i-1)}_{SFS} \cup h(k)$ où $k = argmin_{h(j) \notin h^{(i-1)}_{SFS}}$ {error($h^{(i-1)}_{SFS} \cup h(j)$,$T_s$) }.

$h^{(i)}_{SBS} \leftarrow h^{(i-1)}_{SBS} \setminus h(k)$ où $k = argmin_{h(j) \in h^{(i-1)}_{SBS}}$ {error($h^{(i-1)}_{SBS} \setminus h(j)$,$T_s$) }.

$h^{(i)}_{SRS} \leftarrow h^{(i-1)}_{SRS} \cup h(k)$ où $k =$ random$(j), h(j) \notin h^{(i-1)}_{SRS}$.

Enregistrer les taux d'erreur de $h^{(i)}_{SFS}$, $h^{(i)}_{SBS}$ et $h^{(i)}_{SRS}$.

**fin pour**

L'algorithme 1 résume le protocole expérimental complet appliqué à chaque base de données. Cette procédure fournit en sortie un tableau de valeurs de taille $L \times 3$ (un tableau pour chaque méthode de sélection), qui contient les différents taux d'erreur obtenus. Ces résultats sont détaillés et analysés dans la section suivante.

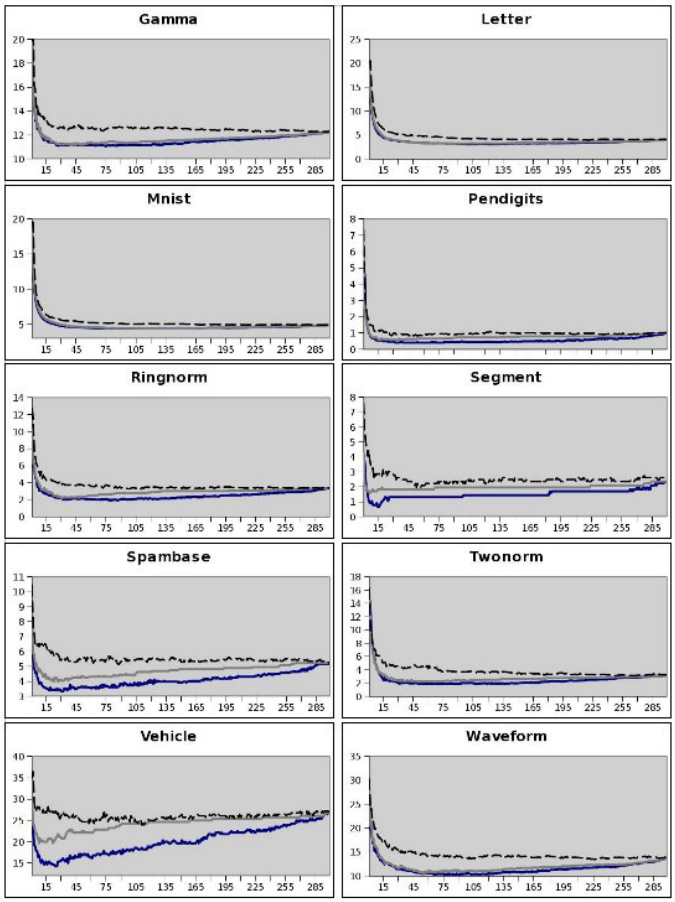
## 3.3 Résultats

La figure 1 présente 10 diagrammes correspondant aux 10 bases de données étudiées. Pour chacun d'eux trois courbes ont été dessinées, représentant les taux d'erreur obtenus avec les trois méthodes de sélection décrites précédemment. Le tableau 2 résume les meilleurs taux d'erreurs obtenus pour chaque processus de sélection appliqué à chaque base de données, ainsi que le nombre d'arbres du sous-ensemble correspondant.

Lorsque l'on examine le tableau 2 on peut tout d'abord observer qu'en dépit de la sous-optimalité des méthodes SFS et SBS, ces algorithmes permettent dans la totalité des cas de trouver un sous-ensemble d'arbres meilleur en terme de performances que la forêt initiale, induite avec Forest-RI. Cette observation met en évidence l'interêt d'étudier la sélection de sous-ensembles d'arbres pour une RF, dans le but d'en améliorer les performances. On peut supposer par conséquent qu'il doit être possible d'améliorer encore plus les performances en cherchant le sous-ensemble d'arbres optimal, en utilisant par exemple des méthodes de sélection optimales (comme la méthode Branch and Bound par exemple [SOM 04]) ou approchant l'optimalité (comme les Algorithmes Génétiques [HAO 03]).

Une deuxième observation qui peut être faite à partir des diagrammes de la figure 1 est que le taux d'erreur minimum obtenu pour chaque base de données, est atteint pour un sous-ensemble d'un nombre d'arbres très petit en comparaison avec l'ensemble d'origine, *i.e.* presque à chaque fois inférieur à 100 arbres. Cela correspond à moins d'un tiers du nombre total d'arbres dans la forêt de départ. En d'autres termes pour chaque RF induite au cours de nos expérimentations, au moins deux tiers des arbres ont été retirés de l'ensemble pour réussir à atteindre les meilleures performances. Ce nombre est même parfois beaucoup plus important puisque les meilleures performances ont été atteintes pour certaines bases de données avec moins de 30 arbres (Segment et Vehicle), ce qui correspond à seulement $10\%$ du nombre total d'arbres induits dans la forêt initiale. Cela montre que parmi tous les arbres de la forêt, seuls quelques-uns peuvent être combinés pour obtenir un classifieur performant. En outre ces résultats mettent en évidence que quand une RF est induite avec un algorithme d'induction "classique" tel que Forest-RI, tous les arbres ne permettent pas systématiquement d'améliorer les performances de l'ensemble, et que l'ajout de certains d'entre eux à l'ensemble a même pour conséquence de faire augmenter le nombre d'erreurs de prédiction. De plus, le fait que le processus de recherche en avant (SFS) soit systématiquement l'approche la plus efficace pour trouver un sous-ensemble d'arbres sous-optimal nous laisse penser qu'il serait intéressant d'étudier la possibilité d'induire une RF dynamiquement en n'ajoutant à l'ensemble que les arbres de décision dont on est certain qu'ils permettraient d'améliorer les performances en généralisation de l'ensemble. Un tel processus d'induction dynamique serait intéressant à la fois en termes de coût de traitement et de gain de performances.

FIG. 1 – Taux d'erreur obtenus au cours du processus de sélection sur les 10 base de données. La courbe noire représente les taux d'erreur obtenus avec SFS, la courbe grise les taux d'erreur obtenus avec SBS et la courbe en pointillé les taux obtenus avec SRS.




TAB. 2 – Récapitulatif des taux d'erreur minimum obtenus et des nombres d'arbres des sous-ensembles correspondants.

| Bases | SFS | | SBS | | Forest-RI |
|---|---|---|---|---|---|
| | taux d'erreur | # arbres | taux d'erreur | # arbres | 300 arbres |
| Gamma | 11.07 | 79 | 11.17 | 50 | 12.19 |
| Letter | 3.07 | 98 | 3.20 | 70 | 4.09 |
| Pendigits | 0.41 | 32 | 0.57 | 28 | 1,01 |
| Segment | 0.66 | 15 | 1.57 | 8 | 2.49 |
| Spambase | 3.33 | 31 | 3.98 | 24 | 5.22 |
| Vehicle | 14.29 | 25 | 19.64 | 9 | 26.79 |
| Waveform | 10.16 | 86 | 10.46 | 56 | 14 |
| Ringnorm | 1.9 | 34 | 2.15 | 31 | 3.33 |
| Twonorm | 1.82 | 75 | 2.19 | 51 | 3.2 |
| MNIST | 4.41 | 97 | 4.4 | 119 | 4.93 |

# 4 Conclusion

Dans cet article, une étude sur la sélection d'arbres de décision pour les RF a été présentée. Le but était de mettre en évidence que certains sous-ensembles d'arbres de décision peuvent présenter de meilleures performances que la forêt initiale. Deux méthodes de sélection de classifieurs ont été utilisées : SFS (Sequential Forward Selection) et SBS (Sequential Backward Selection). En dépit de la sous-optimalité de ces deux méthodes, ce travail a montré qu'il est toujours possible de trouver un sous-ensemble d'arbres plus performant qu'une forêt induite avec un algorithme "traditionnel" tel que Forest-RI, pour peu que ces arbres puissent être soigneusement sélectionnés. Il serait par ailleurs intéressant d'appliquer d'autres méthodes de sélection plus efficaces que SFS et SBS, telles que par exemple la méthode Branch and Bound [SOM 04] ou les Algorithmes Génétiques [HAO 03] pour notamment mieux percevoir dans quelle mesure ce ou ces sous-ensembles peuvent surpasser la forêt initiale.

Ces expérimentations ont également mis en évidence que la meilleure sous-forêt parmi celles que nous avons pu trouver à l'aide de nos processus de sélection, contient toujours très peu d'arbres en comparaison avec la forêt initiale. Pour toutes les bases que nous avons étudiées, au moins deux tiers des arbres ont dû être retirés de l'ensemble initial pour atteindre le taux d'erreur le plus faible. Pour certains même, ce ratio s'est élevé à 90% du nombre total d'arbres. Cela signifie qu'au cours de l'induction d'une RF, tous les arbres ne permettent pas nécessairement d'améliorer l'erreur en généralisation et que seulement un nombre réduit d'entre eux est en réalité nécessaire à l'ensemble pour obtenir un classifieur performant. Par conséquent nous pensons qu'étudier la possibilité de contrôler l'induction d'une forêt, dans le but de n'ajouter à l'ensemble que les arbres qui en améliorent les performances, serait une bonne perspective à ce travail. Il serait notamment intéressant de pouvoir caractériser ces arbres dans le but de pouvoir diriger l'induction de la forêt, ce qui présenterait un intérêt à la fois en termes de complexité algorithmique et en termes de gain de performances.

Ce travail apporte donc une réponse à la première question évoquée en introduction : une RF induite à l'aide d'un algorithme d'induction "classique" contient-elle des arbres de décision qui en détériorent les performances ? Cette réponse est bien évidemment oui. Cependant une problématique reste ouverte, à savoir comment peut-on identifier a priori les arbres de décision à ajouter ou à retirer de la forêt initiale, pour trouver la meilleure sous-forêt possible. Nous pensons que cette problématique devrait être abordée à travers l'étude de propriétés telles que le compromis force/corrélation tel que Breiman le définit dans [BRE 01], la propriété de diversité, ou les spécificités intrinséques aux arbres de décision comme les critères de partitionnement utilisés à chaque nœud, ou les données d'apprentissage utilisées via le principe de bagging.

## Références


[ASU 07] ASUNCION A., NEWMAN D., UCI Machine Learning Repository, 2007.

[BER 07] BERNARD S., HEUTTE L., ADAM S., Using Random Forests for Handwritten Digit Recognition, *International Conference on Document Analysis and Recognition*, pp. 1043–1047, 2007.

[BER 08] BERNARD S., HEUTTE L., ADAM S., Influence of Hyperparameters on Random Forest Accuracy, *Technical Report, University of Rouen*, , 2008.

[BOI 05] BOINEE P., ANGELIS A. D., FORESTI G., Meta Random Forests, *International Journal of Computational Intelligence*, vol. 2, n° 3, pp. 138–147, 2005.

[BRE 84] BREIMAN L., FRIEDMAN J., OLSHEN R., STONE C., *Classification and Regression Trees*, Chapman and Hall (Wadsworth, Inc.) : New York, 1984.

[BRE 96] BREIMAN L., Bagging Predictors, *Machine Learning*, vol. 24, n° 2, pp. 123–140, 1996.

[BRE 98] BREIMAN L., Arcing classifiers, *The Annals of Statistics*, vol. 26, n° 3, pp. 801–849, 1998.

[BRE 01] BREIMAN L., Random Forests, *Machine Learning*, vol. 45, n° 1, pp. 5–32, 2001.

[BRE 04] BREIMAN L., Consistency of random forests and other averaging classifiers, *Technical Report*, , 2004.

[CUT 01] CUTLER A., ZHAO G., PERT - Perfect Random Tree Ensembles, *Computing Science and Statistics*, vol. 33, 2001.

[FRE 96] FREUND Y., SCHAPIRE R., Experiments with a New Boosting Algorithm, *International Conference on Machine Learning*, pp. 148–156, 1996.

[GEU 06] GEURTS P., ERNST D., WEHENKEL L., Extremely Randomized Trees, *Machine Learning*, vol. 36, n° 1, pp. 3–42, 2006.

[HAO 03] HAO H., LIU C., SAKO H., Comparison of genetic algorithm and sequential search methods for classifier subset selection., *Seventh International Conference on Document Analysis and Recognition*, vol. 2, pp. 765–769, 2003.

[HO 98] HO T., The Random Subspace Method for Constructing Decision Forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, n° 8, pp. 832–844, 1998.

[KOH 97] KOHAVI R., JOHN G. H., Wrappers for Feature Subset Selection, *Artificial Intelligence*, vol. 97, n° 1-2, pp. 273-324, 1997.

[KUN 03] KUNCHEVA L., That Elusive Diversity in Classifier Ensembles, *IbPRIA*, pp. 1126–1138, 2003.

[KUN 04] KUNCHEVA L., *Combining Pattern Recognition. Methods and Algorithms*, John Wiley and Sons, 2004.

[LAT 01] LATINNE P., DEBEIR O., DECAESTECKER C., Limiting the Number of Trees in Random Forests, *2nd International Workshop on Multiple Classifier Systems*, pp. 178–187, 2001.

[LEC 98] LECUN Y., BOTTOU L., BENGIO Y., HAFFNER P., Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE*, vol. 86, n° 11, pp. 2278–2324, 1998.

[ROB 04] ROBNIK-SIKONJA M., Improving Random Forests, *European Conference on Machine Learning, LNAI 3210, Springer, Berlin*, pp. 359–370, 2004.

[ROD 06] RODRIGUEZ J., KUNCHEVA L., ALONSO C., Rotation Forest : A New Classifier Ensemble Method, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, n° 10, pp. 1619–1630, 2006.

[ROL 01] ROLI F., GIACINTO G., VERNAZZA G., Methods for Designing Multiple Classifier Systems, *Multiple Classifiers Systems*, pp. 78–87, 2001.

[SOM 04] SOMOL P., PUDIL P., KITTLER J., Fast Branch and Bound Algorithms for Optimal Feature Selection, *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 26, n° 7, pp. 900–912, 2004.