Amélioration des Forêts Aléatoires pour une Meilleure Prédiction

Mostafa El HABIB DAHO
Biomedical Engineering Laboratory
Tlemcen University, Algeria
Email: mostafa.elhabibdaho@gmail.com

Nesma SETTOUTI
Biomedical Engineering Laboratory
Tlemcen University, Algeria
nesma.settouti@univ-tlemcen.dz

Mohammed El Amine LAZOUNI
Biomedical Engineering Laboratory
Tlemcen University, Algeria
aminelazouni@gmail.com

Mohammed Amine CHIKH
Biomedical Engineering Laboratory
Tlemcen University, Algeria
mea_chikh@univ-tlemcen.dz

Abstract

Les Forêts Aléatoires RF (Random Forest) sont largement utilisées dans les problèmes de classification. Dans cet article, une version modifiée de RF appelée Sub_RF (pour Subspaces Random Forests) est proposée. Notre procédure de génération des arbres est basée sur l'algorithme des forêts aléatoires appliqué à un modèle RSM (Random subspaces). Cette approche a été testée sur cinq bases de données de l'UCI Machine Learning Repository. Les résultats montrent que notre méthode proposée Sub_RF améliore les performances de chaque ensemble de données comparant au PERT, SubBag et l'algorithme RF.

Mots clés: *Forêt Aléatoire, Méthodes d'ensembles, Sous-espaces aléatoires, Sub_RF, indice de Gini, classification.*

1 Introduction

Le principe de la méthode d'ensemble (voir, par exemple [1]) est de construire une collection des prédicteurs, et puis agréger l'ensemble de leurs prédictions. Dans la classification, l'agrégation revient, par exemple, a un vote majoritaire parmi les classes fournies par les prédicteurs.

Au lieu d'essayer d'optimiser un modèle qui contient "une seule hypothèse", les méthodes ensemble génèrent plusieurs règles de prédiction et ensuite, mettent en commun leurs différentes réponses.

L'heuristique de ces méthodes est qu'en générant beaucoup de prédicteurs, on explore grandement l'espace des solutions, et qu'en agrégeant toutes les prédictions, on récupère un prédicteur qui prend en considération toute cette exploration.

L'objectif visé est que le prédicteur final soit meilleur que chacun des prédicteurs individuels : même si les classifieurs individuels commettent des erreurs, il est peu probable qu'ils commettent les mêmes erreurs pour les mêmes entrées. Ici, surgit l'idée que les prédicteurs individuels doivent être différents les uns des autres : la majorité ne doit pas se tromper pour un même x.

Pour que cela soit possible, il faut également que les prédicteurs individuels soient relativement bons et différents les uns des autres (là où un prédicteur se trompe, les autres doivent prendre le relais en ne se trompant pas). Le premier point est nécessaire, car agréger des prédicteurs mauvais ne pourra vraisemblablement pas donner un bon prédicteur. Le deuxième point est également naturel, car agréger

des prédicteurs qui sont quasiment pareils donnera encore un prédicteur semblable et n'améliorera pas les prédictions.

Dans ce travail, l'une des méthodes d'ensemble les plus répondus appelée RF (Random Forest) [2] est utilisé. Une Forêt Aléatoire (Random Forest) est constituée d'un ensemble d'arbres simples de prévision, chacun étant capable de produire une réponse lorsqu'on lui présente un sous-ensemble de prédicteurs.

Pour les problématiques de classification, la réponse prend la forme d'une classe qui associe un ensemble (classe) de valeurs indépendantes (prédicteur) à une des catégories présente dans la variable indépendante.

En utilisant les ensembles d'arbres on obtient une amélioration significative de la prévision (c'est-à-dire une meilleure tendance à prévoir sur les nouvelles données), par rapport aux techniques classiques (par exemple CART)[3]. La réponse de chaque arbre dépend du sous-ensemble de prédicteurs choisis indépendamment (avec remplacement) et avec la même distribution pour tous les arbres de la forêt.

Dans cet article, l'intérêt principal est donc d'étudier les performances d'une version modifiée des forêts aléatoires que nous appelons Sub_RF (Subspaces Random Forests). Notre méthode proposée pour l'induction des arbres tente d'améliorer la précision. Pour cela, ce papier a été formulé comme suit: dans la section 2, nous présentons le principe des méthodes d'ensemble utilisées dans cet article. Dans la section 4, nous détaillons notre approche d'induction d'arbres proposée. Nos résultats sont présentés et discutés dans la section suivante. Enfin, une synthèse générale qui met en évidence les principales propriétés de notre technique ainsi que quelques perspectives sont proposés.

2 Méthodes

Les algorithmes les plus utilisés dans la littérature sont présentés dans cette section.

2.1 Bagging

Le Bagging est une méthode d'ensemble introduite par Breiman en1996[4]. Le mot Bagging est la contraction des mots Bootstrap et Aggregating. Le Bootstrap est un principe de ré-échantillonnage statistique [5] traditionnellement utilisé pour l'estimation de grandeurs ou de propriétés statistiques. L'idée du bootstrap est d'utiliser plusieurs ensembles de données ré-échantillonnées à partir de l'ensemble des données observées et ce à l'aide d'un tirage aléatoire avec remise. Ainsi chaque classifieur élémentaire de l'ensemble sera entrainé sur un des échantillons bootstrap de sorte qu'ils soient tous entrainés sur un ensemble d'apprentissage différent. L'agrégation de ces classifieurs permet d'obtenir un prédicteur plus performant.

2.2 Sous-espaces Aléatoires

La Méthode RSM (pour Random Subspaces Method) a été proposée par Ho [6]. Cette méthode est assez similaire au Bagging, mais il ne s'agit pas de manipulation des données, mais les caractéristiques et sans remise. L'idée de base est de former chaque classificateur sur un sous-espace d'attributs tirés aléatoirement de la base originale. Chaque sous-espace aléatoire a la même dimension P, avec $P < M$, où M est la dimension de l'espace original de descripteurs. Dans [6], Ho a montré, pour le paramètre P, que les meilleurs résultats sont généralement obtenus par $P \approx M/2$ caractéristiques. Ho a également montré que le RSM est applicable à tout type de classificateur.

2.3 Forêts Aléatoires

Une Forêt Aléatoire est un prédicteur constitué d'un ensemble de classifieurs élémentaires de type arbres de décision. Dans les cas spécifiques des modèles CART (arbres binaires), Breiman [2] propose une amélioration du bagging avec un algorithme d'induction de forêts aléatoires (Forest-RI — pour Random Forest - Random Input —) qui utilise le principe de randomisation "Random Feature Selection" proposé par Amit et Geman [7]. L'induction des arbres se fait sans élagage et selon l'algorithme CART, toutefois, au niveau de chaque nœud, la sélection de la meilleure partition, basée sur l'index de Gini, s'effectue uniquement sur un sous-ensemble d'attributs de taille préfixée (généralement égale à la racine carrée du nombre total d'attributs) sélectionné aléatoirement depuis

l'espace original des caractéristiques [8]. La prédiction globale de la forêt aléatoire est calculé en prenant la majorité des votes de chacun de ses arbres. Cet algorithme appartient à la famille la plus large des forets aléatoires définis comme suit par Breiman [2].

Algorithm 1 Pseudo code de l'algorithme Random forest

Entrée: L'ensemble d'apprentissage L, Nombre d'arbres N.
Sortie: *Ensemble d'arbres E*
Processus:
for $i = 1 \rightarrow N$ **do**
$T^i \leftarrow BootstrapSample(T)$
$C^i \leftarrow ConstructTree(T^i)$ où à chaque noeud:

- Sélection aléatoire de $K = \sqrt{M}$ Variables à partir de l'ensemble d'attributs M
- Sélection de la variable la plus informatif K en utilisant l'index de Gini
- Création d'un nœud fils en utilisant cette variable

$E \leftarrow E \cup \{C^i\}$
end for
Retourner E

3 Etat de l'art

Le principe des méthodes d'ensembles est de construire une collection de prédicteurs et ensuite d'agréger leurs décisions. L'objectif est d'être en mesure de trouver un ensemble d'hypothèses qui sont différents dans leurs prises de décision afin qu'ils puissent se compléter mutuellement.
Il existe différentes possibilités pour la génération de ces ensembles. La première possibilité consiste à manipuler les données comme dans le Bagging [4] avec l'utilisation du principe de Bootstrap ou dans le RSM [6] où nous utilisons uniquement un sous-ensemble aléatoire de l'espace des caractéristiques.
Une deuxième possibilité d'induire des ensembles de classificateurs est d'utiliser différents algorithmes de classification (ou un algorithme avec différents paramètres) et former chacun sur le même ensemble de données (ici nous parlons des méthodes d'ensembles hétérogènes) [9][10][11][12].
Une troisième possibilité consiste combiner les deux premières méthodes en randomisant les données (Bootstraping par exemple) et l'algorithme d'apprentissage (en utilisant différents algorithmes ou bien un seul avec différents paramètres).
Dans cette optique, Breiman a proposé les Forêts Aléatoires (Random Forests)[2]. L'algorithme des Forêts Aléatoires est l'un des réalisations les plus populaires de la recherche consacrée à l'agrégation d'arbres aléatoires. Depuis leur proposition, plusieurs chercheurs ont tenté d'améliorer les forêts aléatoires par la modification du mécanisme de vote ou la technique d'induction des arbres. Pour exemple, l'algorithme PERT (pour les arbres aléatoires parfaites) qui a été proposé par Cutler et Guohua (2001) [13]. Dans cet algorithme, pour la division de chaque nœud non-terminal, deux exemples de différentes classes sont d'abord tirés au hasard parmi l'ensemble d'apprentissage. Ensuite, un attribut aléatoire est sélectionné et le point de découpage est aléatoirement et uniformément établi entre les valeurs de cet attribut pour les deux exemples pris au hasard. Geurts et al (2006) [14] ont proposé un autre algorithme très aléatoire connu sous le nom d'Extra-trees (Pour Extremely Randomized trees) qui est similaire à PERT et qui combine la randomisation des attributs de la méthode RSM avec une sélection totalement aléatoire du point de découpage. Dans un autre travail, Panov et al [15] ont proposé de combiner les Sous-espaces aléatoire et le Bagging pour construire de meilleurs ensembles. L'idée du SubBag (pour Subspaces Bagging) consiste à sélectionner des échantillons aléatoires avec remplacement à partir de l'ensemble original (comme dans le Bagging) et pour chaque échantillon, seulement 75% des attributs sont choisis au hasard (en utilisant le RSM). Ainsi, les sous-bases obtenues sont plus randomisées par rapport à ceux générées par le Bootsrping. Dans leur papier [15], Panov et Dzeroski ont prouvé que cette approche donne des performances comparables à celles des forêts aléatoires, avec l'avantage supplémentaire d'être applicable à n'importe quel algorithme d'apprentissage sans la nécessité de le randomiser.
Louppes et Geurts ont proposé dans [16] un travail similaire au SubBag appelé Random Patches où chaque modèle de l'ensemble est construit à partir d'un sous ensemble de données et de variables obtenues par tirage aléatoires de l'ensemble original des données. Leur méthode est très efficace dans le cas de l'utilisation de grandes bases de données, car elle réduit considérablement l'espace mémoire et le temps

d'exécution. Une méthode qui utilise le même principe de SubBag avec l'utilisation de l'algorithme de forêt aléatoire pour générer des classificateurs (des arbres de type CART) est proposée dans ce travail.

4 Méthode proposée

La méthode proposée permet la création d'un ensemble de classificateurs en utilisant le procédé du SubBag [15] pour la génération des échantillons d'apprentissage tant dit que les classificateurs sont des arbres de décisions générés en utilisant l'algorithme Forest-RI [2]. Cette méthode d'induction d'arbres a été nommée Sub_RF (pour Subspaces Random Forest). Le pseudo-code illustre notre algorithme:

Algorithm 2 Pseudo code de l'algorithme Sub_Rf

Entrée: L'ensemble d'apprentissage L, Nombre d'arbres N, Taille du sous-espace S.
Sortie: *Ensemble d'arbres*
Processus:
for $i = 1 \rightarrow N$ **do**
  $T^i \leftarrow BootstrapSample(T)$
  $T^i \leftarrow SelectRandomSubSpaces(T^i, S)$
  $C^i \leftarrow ConstructRF\_tree(T^i)$
  $E \leftarrow E \cup \{C^i\}$
end for
Retourner E

Cet algorithme ajoute un autre niveau de randomisation à la méthode du SubBag grâce à la fonction $ConstructRF\_tree()$ qui permet de créer des arbres à l'aide du principe des forêts aléatoires. Pendant le processus de l'apprentissage des arbres, à chaque nœud, la sélection de la meilleure répartition sur la base de l'indice de Gini est effectuée non pas sur l'ensemble des attributs M mais sur un sous-ensemble sélectionné de façon aléatoire (racine de M selon l'algorithme Forest-RI). Dans la section suivante, les résultats obtenus avec Sub_RF seront discutés et comparés au PERT, SubBag et aux Forêts aléatoires.

5 Résultats et interprétations

Pour tester notre algorithme, cinq bases de données de l'UCI Machine Learning Repository [17] ont été utilisées. Les bases de données qui ont été utilisées dans nos expériences sont décrites dans le tableau suivant :

Table 1: Les bases de données utilisées

Base	Instance	attribut	Classe
Breast	699	9	2
Ecoli	366	7	8
Isolet	7797	617	26
Liver	345	6	2
Pima	768	8	2

Dans nos expériences, quatre algorithmes différents sont mis en œuvre à savoir PERT, Random Forest, SubBag et notre méthode proposée Sub_RF. L'objectif est de visualiser et d'étudier l'évolution de l'erreur de chaque technique.

Tout d'abord, chaque base de données a été divisé en deux sous-ensembles de données, une pour l'apprentissage et l'autre pour le test en utilisant la validation croisée (5-fold cross validation). La division de la base a été effectuée par tirage au hasard de la base originale en respectant la distribution des classes. Comme il l'a été déjà expliqué, notre méthode utilise le Bootstraping pour générer les sous-bases, ainsi nous aurons en moyenne, pour chaque échantillon bootstrap 63,2% des exemples uniques à partir de l'ensemble d'origine, le reste étant des doubles (première randomisation).

Pour nos expérimentations, les paramètres recommandés pour chaque algorithme seront utilisés. Dans [15], Panov et Dzeroski ont suggéré l'utilisation de 75% de l'espace des attributs (à l'aide méthode RSM) pour chaque Bag (seconde randomisation). Pour le paramètre K de l'algorithme Random Forest, plusieurs travaux de la littérature ont montré qu'un nombre d'attributs égal à $\sqrt{M}$ (M est la taille de tout l'espace des attributs) est un bon compromis pour produire une forêt performante [2] [18].

Pour le choix de la taille de l'ensemble, nous avons développé plusieurs forêts aléatoires avec différents nombre d'arbres: 10, 50, 100, 200, 500 et 1000. Les résultats indiquent que, pour plus de 100 arbres, le taux d'erreur reste plus ou moins stable (voir Figure1).

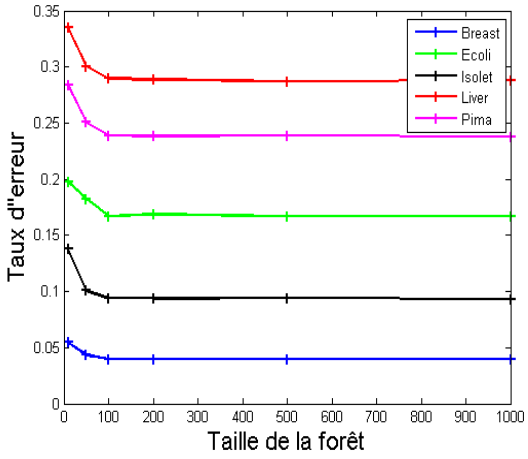

Figure 1: Taux d'erreur de RF avec les différents nombres d'arbres

Dans ce qui suit, la taille de chaque forêt est fixé à100 arbres pour chaque. Dans un second temps, nous avons effectué une comparaison entre RF, SubBag et notre méthode proposée Sub_RF afin d'évaluer la performance de cette nouvelle technique pour générer des classificateurs à base d'arbres.
Les résultats (voir Table 2) montrent que Sub_RF donne de meilleurs résultats par rapport aux autres méthodes, en raison de l'augmentation de sa randomisation. Cela peut être expliqué par le fait que, contrairement au RF, les arbres du Sub_RF sont très différents car ils n'utilisent pas tous les attributs et, contrairement à PERT, ils choisissent la meilleure variable. SubBag donne, dans un cas de nos expériences, de meilleurs résultats que Su_RF, cela est dû à leur ressemblance dans la première et la seconde étape de randomisation. A partir de ces résultats, nous pensons que Sub_RF fournit globalement le meilleur compromis en termes de randomisation dans le cadre de la génération des forêts aléatoires.

Table 2: Taux d'erreurs des différents algorithmes

	PERT	RF	SubBag	Sub_RF
Breast	0,0478	0,04	**0,03**	0,0403
Ecoli	0,2555	0,1672	0,1551	**0,1499**
Isolet	0,1982	0,0899	0,0808	**0,0781**
Liver	0,2816	0,2812	0,2603	**0,2501**
Pima	0,2541	0,2387	0,2311	**0,2283**

6 Conclusion

Dans ce papier, une nouvelle méthode de génération d'arbres appelé Subspaces Random Forest (Sub_RF) qui utilise le Bagging, la méthode des Sous-espaces aléatoires et les Forêts Aléatoires a été essentiellement proposée. Cette méthode s'est, en fait, avéré efficace par rapport à la forêt aléatoire classique. Pour cette raison, nous avons testé expérimentalement notre approche sur dix bases de données du répertoire de l'UCI. Les résultats montrent que notre approche suggérée est compétitif aux méthodes existantes dans l'état de l'art. Il reste plusieurs questions et limites de notre approche auxquelles nous souhaitons répondre à l'avenir. Tout d'abord, nous voulons renforcer ces résultats avec une analyse théorique ainsi que tester notre algorithme sur de grandes bases de données. Nous aimerions également tester certaines techniques de sélection ensemble afin de garder que les meilleurs arbres de la forêt vu que les arbres générés sont très différent.

References

[1] T. Dietterich, "Ensemble methods in machine learning," *Lecture Notes in Computer Science*, vol. 1857, pp. 1–15, 2000.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[3] Nesma Settouti, Mostafa El Habib Daho, Mohammed El Amine Lazouni, and Mohammed Amine Chikh, "Conditional inference forest for variables selection of medical data," in *Biomedical Engineering International Conference (BIOMEIC'12)*, 2012, vol. 1, pp. 84–90.

[4] Leo Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[5] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, Chapman & Hall, New York, 1993.

[6] Tin Kam Ho, "The random subspace method for constructing decision forests," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 8, pp. 832–844, 1998.

[7] Yali Amit and Donald Geman, "Shape quantization and recognition with randomized trees," *Neural Computation*, vol. 9, no. 7, pp. 1545–1588, 1997.

[8] N. Sirikulviriya and S. Sinthupinyo, "Integration of rules from a random forest.," in *International Conference on Information and Electronics Engineering IPCSIT vol.6 (2011) IACSIT Press, Singapore*, 2011.

[9] Kevin Woods, W. Philip Kegelmeyer, Jr., and Kevin Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 405–410, Apr. 1997.

[10] *Hybrid ensembles and coincident-failure diversity*, vol. 4, 2001.

[11] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas, "Effective voting of heterogeneous classifiers," in *In Proceedings of the 15th European Conference on Machine Learning*, 2004, pp. 465–476.

[12] Shun Bian and Wenjia Wang, "On diversity and accuracy of homogeneous and heterogeneous ensembles," *Int. J. Hybrid Intell. Syst.*, vol. 4, no. 2, pp. 103–128, Apr. 2007.

[13] Adele Cutler and Guohua Zhao, "Pert - perfect random tree ensembles," *Computing Science and Statistics*, p. 497, 2001.

[14] Pierre Geurts, Damien Ernst, and Louis Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[15] Panče Panov and Sašo Džeroski, "Combining bagging and random subspaces to create better ensembles," in *Proceedings of the 7th international conference on Intelligent data analysis*, Berlin, Heidelberg, 2007, IDA'07, pp. 118–129, Springer-Verlag.

[16] Gilles Louppe and Pierre Geurts, "Ensembles on random patches," in *Machine Learning and Knowledge Discovery in Databases*, PeterA. Flach, Tijl Bie, and Nello Cristianini, Eds., vol. 7523 of *Lecture Notes in Computer Science*, pp. 346–361. Springer Berlin Heidelberg, 2012.

[17] K. Bache and M. Lichman, "UCI machine learning repository," http://archive.ics.uci.edu/ml, 2013.

[18] Simon Bernard, Laurent Heutte, and Sébastien Adam, "Influence of hyperparameters on random forest accuracy," in *Multiple Classifier Systems*, JónAtli Benediktsson, Josef Kittler, and Fabio Roli, Eds., vol. 5519 of *Lecture Notes in Computer Science*, pp. 171–180. Springer Berlin Heidelberg, 2009.