Simplifying Random Forests using Diversity

SOUAD TALEB ZOUGGAR[1] AND ABDELKADER ADLA[2]
[1]Department of Economics, Oran 2 University, ALGERIA
[2]Department of Computer Science, Oran 1 University, ALGERIA
souad.taleb@gmail.com, adla.abdelkader@univ-oran1.dz

Abstract: - In this paper, we propose a diversity measure for random forests simplification using both SFS and SBE paths. This is performed in two stages: 1) we use first an overproduce method which generates a large number of trees; 2) We use SFS and SBE paths combined with diversity measurement to reduce the initial ensemble of trees. The proposed method is applied to UCI Repository data sets. A comparative study of the two types of paths with a performance-based pruning method is given. The results are encouraging and allow obtaining ensembles of reduced sizes exceeding, in some cases, the performances of the initial forest and the method used for comparison.

Key-Words: - Classification, CART Trees, Random Forests, Pruning, Diversity, Accuracy, Forward Selection, Backward Elimination.

1 Introduction

Random forests [1] use bagging [2] to generate CART trees [3]. Bagging allows random selection of a subset of training data (bootstrap) for generating each tree in the forest. Bootstraps are built using random draws with delivery to the original learning set.

A random selection of variables (or Random feature selection) is added to the bagging. This selection allows choosing a subset of variables for the partition at each node; a fixed number of K characteristics is chosen randomly and from which are chosen those which optimize the partitioning.

To be efficient, a forest must be composed of trees very different from each other. In [1] the author introduces the correlation between two trees and shows that the reduction of this correlation induces a decrease in the prediction error. The author also brings in the strength of a tree (a quality measure of an individual tree) and shows that the prediction error decreases as this power increases. The Breiman's prediction error increase seems rather rough, but the ideas it conveys are very important to understand random forests: a forest of individual efficient and different trees is an efficient forest.

A large number of trees forming the forest have also the effect to reduce the variability of the global predictor. In his paper [1] Breiman has shown that beyond a certain number of trees the error in generalization tends to its maximum, which shows that a large number of trees in a forest does not make it more efficient.

In this direction, several studies try to limit the number of trees in a random forest by trying to find the optimal ensemble of the forest. This process is called "Random Forest Pruning".

The pruning of a random forest is an additional step which aims to reduce the number of constitutive trees. This allows saving the storage space and reducing the prediction time while aggregating or combining all the generated trees. In a regression case, aggregating the predictions of q predictors consists in averaging them: given q models, each of them provides a response $\hat{y}_l$ and, then the final prediction is $\frac{1}{q}\sum_{l=1}^{q} \hat{y}_l$. In the case of classification, aggregation consists in making a majority vote among the classes provided by each predictor.

The goal of this research is to simplify a random forest using a diversity measure and two research strategies: Sequential Forward Selection (SFS) and Sequential Backward Elimination (SBE). The measure maximizes the chances of choosing trees that disagree with the classifying of an instance. SFS begins with an empty ensemble and adds the trees one by one to optimize the measure, while SBE starts with the entire forest and eliminates, one by one, the trees affecting the ensemble negatively.

The experiments are performed out on benchmarks of the UCI Repository [4]. A comparative study is carried out between the initial forest composed of all the trees, the pruning method based on the ensemble performance and that proposed by Bernard and his colleagues [5] also using the SFS and SBE paths, based on two

criteria: performance and size of the obtained ensembles.

The rest of the paper is organized as follows: In section 2, we give some preliminaries on random forests, sequential forward selection, and sequential backward elimination. Section 3 presents related work that led to the discovery of forests as well as the areas of application in which the approach was applied. In section 4 we describe our proposal for random forest pruning. In section 6 the experiments results are presented. Finally in the last section we conclude and give some future work.

2 Preliminaries

2.1 Random Forests

In [1] the author gives the following definition of a random forest: Let $\{\hat{h}(.,\Theta_1), ..., \hat{h}(., \Theta_q)\}$ be a collection of predictors using trees, where $(\Theta_1, ..., \Theta_q)$ is a sequence of random variables, independent of the learning sample Ω_n. The predictor of random forests is obtained by aggregating this collection of predictors.

Random forests RI (Random Input) are an implementation of random forests [6] which the corresponding general algorithm is as follows:

Algorithm RI:
Input:
Ω_n: A learning sample comprising n examples and p variables,
m: a subset of variables to choose at each step;
Output:
T = Number of trees to build;
Begin
To build each tree:
Create a sample Ω_B of Ω_n
Create a non-pruned CART tree;
At each node of the tree, choose randomly m variables from which the test variable will be chosen;
End.

2.2 Sequential Forward Selection

A Sequential Forward Selection (SFS) path is used to find a sub-optimal solution, because the sequential process used by this method makes each iteration dependent on the previous one and, therefore, all the possible solutions are not explored. However, this course method has the advantage of being simple and fast. Its principle is simple: the process starts from an empty set and add progressively the models that maximize the evaluation function (the entropy function here). The models are added if they belong to the neighborhood of the current ensemble of trees. The process stops when there are no trees to add or there is no improvement in the evaluation function.

Given an ensemble of 4 trees {T1, T2, T3, T4}, a function f_E to maximize and a current ensemble S, a SFS selection from this ensemble is shown in Fig. 1:

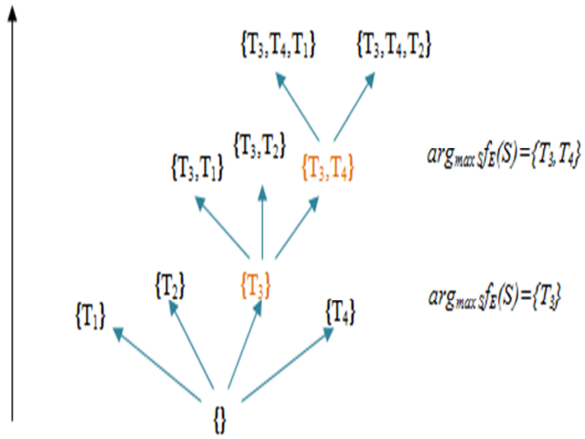

Fig. 1: SFS path for an ensemble composed of 4 trees

The neighborhood of the subset S = $\{T_2\}$ is the subset augmented with one tree among the remaining trees neighborhood $_S$={$\{T_2, T_1\}$, $\{T_2, T_3\}$, $\{T_2, T_4\}$}.

The following algorithm shows the steps to perform a forward selection:

Algorithm SFS ;
Input:
F: Ensemble of classifiers F={t_t, t=1…T}, f_E : Evaluation function, Evaluation set: E={(x_i,y_i), i=1…n} of training examples with their classes;
Output:
S: sub ensemble of F;
Begin
S:=Φ;
While Sub ≠ F do
t_t:= $argmax_{fE}(S,t,E)$;
$t \in F - S$;
$S \coloneqq S \cup \{t_t\}$;
End While;
End.

2.3 Sequential Backward Elimination

A Sequential Backward Elimination (SBE) path has the same properties as a SBS one (search is done locally) but first operates by initializing the search

ensemble by all the trees in the ensemble (Fig. 2). Then, the algorithm iteratively eliminates trees that negatively affect the ensemble (i.e., decrease the evaluation function value). As with SFS, the process stops when there is no improvement in the evaluation function

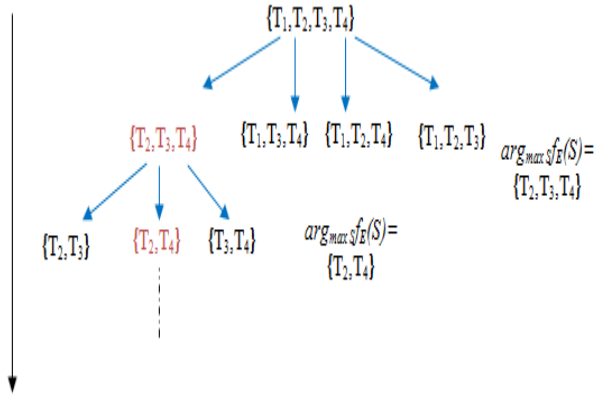

Fig. 2: SBE path for an ensemble composed of 4 trees

Algorithm SBE ;
Input:
F: Ensemble of classifiers F={t_t, t=1…T}, f_E : Evaluation function, Evaluation set: E={(x_i,y_i), i=1…n} of training examples with their classes;
Output:
S: sub ensemble of F;
Begin
S:=F;
While Sub ≠ Φ do
t_t:= $argmax_{fE}$(S,t,E);
$t \in S$
$S \coloneqq S - \{t_t\}$;
End While;
End.

3 Related Work

Several decision tree ensemble methods have emerged. They have been successfully applied to various applications. Early work addressing issues related to the synthesis results of multiple trees [7] [8] shows that a large improvement in accuracy can be achieved by using the same training sample to generate a combination of binary decision trees (generated by selection criteria for different variables) and combining them using the Dempster and Shafer model [9] [10]. The approach is applied in the field of character recognition.

The random draw of variables to cut a node had been used by [1] in image recognition problems for random feature selection or random trees. They introduce a disturbance in the choices of the internal partitions, by preselecting randomly at each node, a subset of variables to choose the optimal partition.

Based on the work presented in [11], Breiman [1] introduces Random Forest (RF). Since their appearance, forests have been used in a wide variety of fields of application, particularly in the medical field.

The number of trees in the forest is increased without over-learning risk [1]. Trees are added arbitrarily to the forest; so there is no guarantee that all the trees will cooperate well. In addition, a large number of trees results in increased learning time, storage resources, and prediction time required for querying all the forest trees. Forest pruning is important for two reasons: 1) efficiency: reduced response time and storage resources; and 2) great performance in prediction.

The pruning methods of random forests can be classified into two categories: static and dynamic. Static methods generate a fix number of trees then select the ones that will be part of the random forest, while dynamic methods generate trees that will be directly included in the forest using a certain criterion.

For the static approach, Latinne [12] proposes to use a direct and non-parametric comparison test. The McNemar test [13] allows deciding whether to include a tree in an ensemble or not. The process systematically determines a minimum number of models to combine for a given database. Knowing the minimum size of the classifier ensemble that gives the best accuracy allows saving time and storage space especially for large data sizes and real-time applications.

In order to reduce the number of trees in the forest while maintaining its precision, Bernard and his colleagues [5] proposed methods of tree selection after the construction of the forest. The authors show that better subsets of decision trees can be obtained by using the sub-optimal methods of selecting SFS (Sequential Forward Selection) and SBS (Sequential Backward Selection) classifiers where adding or removing models is based on the performance measure.

In [14] the authors introduce a pruning algorithm based on margin optimization that can reduce the size and increase the performance of a random forest ensemble. The proposed algorithm takes into account the distribution of the forest margin on the learning ensemble. To this end, four

different metrics based on the margin distribution are used to evaluate the generalization capacity of subsets and the importance of individual classifiers. Once the forest is built, the trees are ordered according to the margin metrics. Finally, ensembles with decreasing sizes are constructed by recursively removing the least important trees one at a time.

In [15] the authors propose to prune a random forest (RF) for limited sources prediction. Initially an RF random forest is constructed then pruned to optimize the cost and accuracy of the expected features. The forest pruning program encompasses linear constraints that favor the reuse of features. The total uni-modularity of the constraints is set to prove that the corresponding LP relaxation solves the original whole program. Connections to combinatorial optimization are finally exploited and an efficient primal-dual algorithm adaptable to large scale data is developed.

In [16] the authors use statistical analyzes of basic classifiers to ensemble pruning without compromising the classification accuracy. Learning the statistics of the entire forest in addition to the information available in the dataset can reveal the optimal thresholds that should be used to prune an ensemble model.

In [17] the authors propose an ensemble selection technique that provides a small size and a great accuracy. They use a genetic algorithm for which the initial population is composed of individual trees with high performance to improve the result of the algorithm.

In [18] the authors propose a new forest simplification strategy by assessing the importance of tree branches against the complete ensemble. This importance is evaluated considering the ensemble performance as well as the diversity of the elements composing the whole. The proposed metric is used to evaluate how well the ensemble accuracy can be improved when a branch is pruned.

For the dynamic approach, which consists in generating trees gradually satisfying a certain criterion, several works have also been proposed, namely in [19] which proposes the development of a method which automatically determines the number of trees to include in a forest during the generation process. The method is based on the use of an online adjustment procedure and is evaluated using conventional random forests and its variants as ensemble methods. Initially the ensemble contains ten trees. At each iteration, a new tree is added and tested if it allows a better fit. To select the best fit, eight polynomials are used. The end of the iterative process is based on predefined thresholds for the adjusted value and accuracy.

In [20] the authors propose to add trees independently. A tree is added based on the evaluation of the current sub-forest using adaptive approach. A tree is initially generated, then, to generate the next tree, the weights of the individuals of the learning sample are modified. These weights are incremented for misclassified instances and decremented for those that are well ranked. The trees generated are thus dependent on each other.

In [21] the authors develop three heuristics to improve learning by random forest. The first is to use disjointed data partitions to learn basic trees, then to reduce the depth of trees without using repetitive variables, and finally to select reduced subsets of attributes for cutting at each node of each tree.

4 The Measure for Random Forest Pruning

We propose a static method of random forest pruning which consists in generating the whole forest in what we call “an overproduce phase”. Then we will eliminate trees that negatively influence performance (Fig. 3).

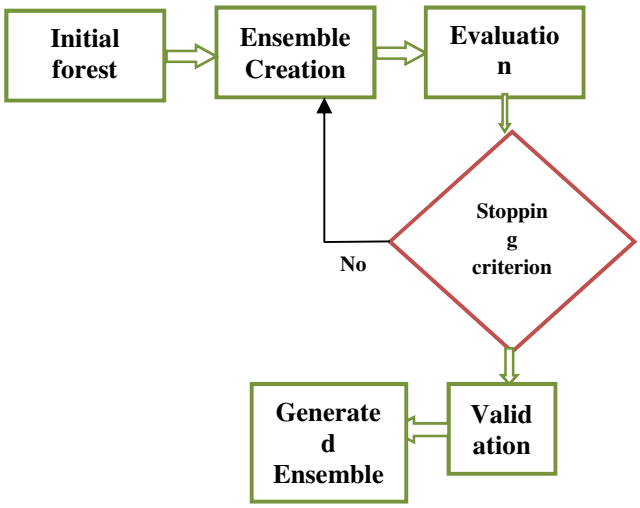

Fig. 3: Generating steps of an ensemble from an initial random forest

The forest ensemble is generated. A first ensemble containing a single tree chosen randomly is created. A diversity-based evaluation function is calculated: if its value is improved, we continue to add trees in the ensemble otherwise we stop and choose the current ensemble.

The key idea in this approach is to generate only trees that have maximum diversity (they

are less correlated with each other). This is based on the principle that the error of generalization of the random forest reduces wile diversity increases among the trees.

Let Ω_V be a sample of individuals with their labels (classes), $|\Omega_V|=n$, $\Omega_V=\{v_1,\ldots,v_n\}$, and $\Omega_V=\{v_1,\ldots,v_n\}$. Each individual v_j is described by m variables denoted $x_{1j}, \ldots, x_{mj}$. Let C_i be a classifier belonging to the classifiers ensemble $\{C_1,\ldots,C_i,\ldots,C_T\}$ represented by a n-dimensional binary vector $y_i=(y_{1i},\ldots,y_{ni})^T$ such that y_{ji}=1if the classifier Ci recognize the individual v_j and 0 otherwise. The entropy function f_E measures the diversity within an ensemble (forest) [22]. Given an individual $x_j \in \Omega_V$, if half of the classifiers T/2 doesn't misclassify x_j then the other half T-T/2 misclassifies it necessarily and vice versa. In this case, we speak of maximum diversity.

We note $nc(x_j)$ the number of classifiers of T which correctly classify x_j, $nc(x_j) = \sum_{i=1}^{T} y_{ji}$. The entropy measure f_E is written as:

$$f_E = \frac{1}{n}\sum_{j=1}^{n} \frac{1}{T-\frac{T}{2}} \min\{nc(x_j), T - nc(x_j)\}$$

$f_E \in [0,1]$ where 1 indicates a very large diversity and 0 a lack of diversity. Thus, the goal is to maximize the f_E function

5 Experiments and results

In this section, we describe information about the datasets used to carry out our experiments. We test 9 benchmarks of UCI Repository [4][23] presented in Table 1. The value of the parameter k is fixed to $\sqrt{m}$ [24] (m represents the number of descriptors).

Table1: Datasets description and the used values of the parameter k

Dataset	Number of Instances	Features	Classes	k
Gamma	19020	10	2	$\overline{10}=3$
Letter	20000	16	26	$\overline{16}=4$
Pendigits	10992	16	10	$\overline{16}=4$
Segment	2310	19	7	$\overline{19}=4$
Spambase	4610	57	2	$\overline{57}=7$
Vehicle	946	18	4	$\overline{18}=4$
Waveform	5000	40	3	$\overline{40}=6$
Ringnorm	7400	20	2	$\overline{20}=4$
Townorm	60000	84	2	$\overline{84}=9$

The datasets are split into two samples: a sample for learning and pruning denoted Ω_L (80% of the initial sample size), and a test sample to compute the performance in generalization or the rate of success (20% remaining).

An initial set of 300 trees is generated first composing the Random Forest (RF). The pruning methods are then applied to RF ensemble in order to eliminate irrelevant trees based on the f_E function.

The comparison is made between the proposed methods SFS_{fE}, SBE_{fE}, the SFS_P and SBE_P methods, based on performance, proposed in [5] and the initial RF against two criteria: success rate in generalization using the test sample Ω_T and the ensembles size obtained after pruning.

Table 2: Success Rates of ensembles obtained by SFS_{fE}, SBE_{fE}, SFS_P, SBE_P and RF for all the datasets

	Success Rate					Improved Success Rate	
	SFS_{fE}	SBE_{fE}	SFS_P	SBE_P	RF	SFS_{fE} Vs SFS_P	SBE_{fE} Vs SBE_P
Gamma	88.56%	88.42%	**88.93%**	88.83%	87.81%	-0.0037	-0.0041
Letter	**96.98%**	96.91%	96.93%	96.8%	95.91%	0.0005	0.0011
Pendigits	99.42%	99.42%	**99.59%**	99.43%	98.99%	-0.0017	-0.0001
Segment	99.24%	99.18%	**99.34%**	98.43%	97.51%	-0.0010	0.0075
Spambase	**96.88%**	96.43%	96.67%	96.02%	94.78%	0.0021	0.0041
Vehicle	85.62%	85.50%	**85.71%**	80.36%	73.21%	-0.0009	0.05%
Waveform	**89.92%**	89.88%	89.84%	89.54%	86.00%	0.0008	0.0034
Ringnorm	**98.32%**	98.02%	98.1%	97.85%	96.67%	0.0022	0.0017
Townorm	**99.93%**	99.78%	98.18%	99.81%	96.8%	0.02	-0.0003
Average Success Rate	0.9499	0.9484	0.9481	0.9412	0.9196		

From Table 2, we note that the pruning methods SFS_{fE}, SBE_{fE}, SFS_P, and SBE_P have better accuracy performance compared to the RF forest by improving it with 3.02%, 2.87%, 2.85%, and 2.15% respectively. SFS_{fE} is better than all other methods in 5 cases with a minimum improvement of 0.05% and a maximum improvement of 5.26%. SFS_p is the best in 4 cases with a minimum improvement of 0.02% and a maximum improvement of 5.35%. Over all the 9 datasets, SFS_{fE} comes in first place with an average success rate of 0.499, SBE_{fE} in second position with 0.9484 followed by SFS_p and SBE_p with average rates of 0.9412 and 0.9196 respectively.

The results of the comparative study based on the ensembles size obtained by the four pruning methods are given in Table 3:

Table 3: Ensembles sizes obtained by SFS_{fE}, SBE_{fE}, SFS_P and SBE_P for all datasets

	Ensemble Size				SFS_{fE} Vs SFS_P	SBE_{fE} Vs SBE_P
	SFS_{fE}	SBE_{fE}	SFS_P	SBE_P		
Gamma	65	**45**	79	50	-14	-5
Letter	98	81	98	**70**	0	11
Pendigits	54	30	32	**28**	22	2
Segment	12	**6**	15	8	-3	-2
Spambase	26	**19**	31	24	-5	-5
Vehicle	17	11	25	**9**	-8	2
Waveform	72	**42**	86	56	-14	-14
Ringnorm	32	**22**	34	31	-2	-9
Townorm	63	54	75	**51**	-12	3
Average Size	48.78	34.44	52.78	36.33		

Of the 9 data sets, SBE_{fE} and SBE_p reduce the size of the forest significantly compared to SFS_{fE} and SFS_p. SBE_{fE} gives better size reductions of the ensembles obtained in 5 out of 9 cases with a maximum reduction of 294 trees for Segment and a minimum of 246 trees for townorm. SBep is in second position with 4 cases with a maximum reduction of 291 trees for Vehicle and a minimum of 230 trees for Letter. Finally, compared to the initial forest RF over 9 data sets, the four methods reduce the ensembles size obtained in all the datasets with average reductions of 88.52%, 87.89%, 83.75%, and 82.41% for SBE_{fE}, SBEp , SFSfE, and SFSp respectively.

To better analyze the data, we use the comparison approach proposed by [25] which consists to assign a rank to each compared method. We make 4 comparisons based on the rank of each method for each dataset. The comparative studies will focus on the change impact of the evaluation measure for a given path (Table 4 and Table 5) and that of the path method (Table 6 and Table 7).

Table 4: Ranking SFS_{fE} and SFS_P

	Success Rate		Size	
	SFS_{fE}	SFS_P	SFS_{fE}	SFS_P
Gamma	2	1	1	2
Letter	1	2	1	1
Pendigits	2	1	2	1
Segment	2	1	1	2
Spambase	1	2	1	2
Vehicle	2	1	1	2
Waveform	1	2	1	2
Ringnorm	1	2	1	2
Townorm	1	2	1	2
Average	1.44	1.56	1.11	1.78

Table 5: Ranking SBE_{fE} and SBE_P

	Success Rate		Size	
	SBE_{fE}	SBE_P	SBE_{fE}	SBE_P
Gamma	2	1	1	2
Letter	1	2	2	1
Pendigits	2	1	2	1
Segment	1	2	1	2
Spambase	1	2	1	2
Vehicle	1	2	2	1
Waveform	1	2	1	2
Ringnorm	1	2	1	2
Townorm	2	1	2	1
Average	1.33	1.67	1.44	1.56

Comparing the diversity and performance measures, we find that:

- with SFS path (Table 4), SFS_{fE} is better than SFSp un relation to the success rate (or size of the generated subsets) with an average rank of 1.44 (resp., 1.11), i.e. a difference of 1.56 (resp.0.67);
- with SB_E (Table 5), SBE_{fE} is better than SBEp in relation to success rate with an average rank of 1.33 against 1.44, i.e. a difference of 0.11, and jn relation to the generated ensembles size with a mean rank of 1.44 against 1.56, i.e. a difference of 0.12.

The use of a diversity-based measure allows obtaining smaller ensembles with better performance compared to a performance-based measure.

Table 6: Ranking SFS_{fE} and SBE_{fE}

	Success Rate		Size	
	SFS_{fE}	SBE_{fE}	SFS_{fE}	SBE_{fE}
Gamma	1	2	2	1
Letter	1	2	2	1
Pendigits	1	1	2	1
Segment	1	2	2	1
Spambase	1	2	2	1
Vehicle	1	2	2	1
Waveform	1	2	2	1
Ringnorm	1	2	2	1
Townorm	1	2	2	1
Average	1	1.89	2	1

Table 7: Ranking SFS_P and SBE_P

	Success Rate		Size	
	SFS_P	SBE_P	SFS_P	SBE_P
Gamma	1	2	2	1
Letter	1	2	2	1
Pendigits	1	2	2	1
Segment	1	2	2	1
Spambase	1	2	2	1
Vehicle	1	2	2	1
Waveform	1	2	2	1
Ringnorm	1	2	2	1
Twonorm	2	1	2	1
Average	1.11	1.89	2	1

Comparing path strategies, we find that:

- with diversity measure (Table 6), SFS method allows obtaining ensembles with better performances compared to SBE with an average rank of 1 against 1.89, i.e. a difference of 0.89. In contrast, SBE obtains smaller size ensembles with an average rank of 1 (the smallest average rank).
- With performance-based measure (Table 7), SFS has better success rate than SBE with a difference of 0.78 and SBE better obtained ensembles size compared to SFS.

Both SFS and SBE methods allow suboptimal searches and reduce the ensembles size but SFS the ensembles obtained with SFS path have smaller sizes and are more efficient than SBE. The time complexity of the SFS and SBE methods to cross the ensembles space is $O(t2g(T, N))$: $g(T, N)$ corresponds to the evaluation function complexity, T is the ensemble or forest size, N is the sample size used for pruning and t is the set of trees added or removed during the selection process.

6 Conclusion

To be efficient a forest must be composed of trees very different from each other. (Breiman, 2001) introduced the correlation between two trees and showed that reducing this correlation involves prediction error reducing. He had also introduced the power of a tree (a quality measure of an individual tree) and showed that prediction error decreases as this power increases.

The increase of the number of trees composing the forest decreases the variability of the global predictor; which involves performance degradation. Forest pruning methods have been proposed to address this problem.

In this paper, we used a diversity measure to simplify a random forest ensemble. The measurement employs two search strategies: SFS path (Sequential Forward Selection) and SBE path (Sequential Backward Elimination). The method is compared with that proposed by Bernard and his colleagues [5] which is based on the performance and uses the two types of paths. These search strategies are compared to show which one explores better the search space.

The experimental results show the measure effectiveness to search ensembles of reduced size and performances equal or sometimes exceeding those of the initial forest ensemble as well as the methods proposed in [5] using accuracy as measurement.

Diversity allows obtaining reduced sizes and more efficient ensembles compared to the method using performance. Regarding the path strategies, SFS allows obtaining less reduced sizes and more powerful ensembles compared to SBE.

In future work, in order to improve the performances, we propose to search optimal ensembles of trees using for example optimal methods such as Branch and Bound method [26] or near optimal methods such as Genetic Algorithms. We will also apply the measure (based on diversity and accuracy simultaneously) in bagging ensembles [27][28] for forest pruning. Finally, we plan to experiment our approach in the field of digestive diseases detection.

References:

[1] L. Breiman, Random Forests, *Machine Learning*, vol. 45, 2001, pp. 5-32.

[2] L. Breiman, Bagging Predictors, *Machine Learning*, vol. 26, no. 2, 1996, pp. 123-140.

[3] L. Breiman, J.H. Friedman, R.A. Olshen and C.J. Stone, *Classification and Regression Trees*, Chapman & Hall, New York, 1984.

[4] A. Asuncion and D. Newman, UCI machine learning repository, URL :http://www.ics.uci.edu/»mlearn/MLRepository.html, 2007

[5] S. Bernard, L. Heutte and S. Adam, On the Selection of Decision Trees in Random Forest. IJCNN, 2009, pp. 302-307.

[6] L. Breiman, and A. Cutler, Random Forests, Berkeley. Available from http://www.stat.berkeley.edu/users/breiman/RandomForests/, 2005

[7] S. Shlien, Multiple binary decision tree classifiers, *Pattern Recognition*, vol. 23, no. 7, pp. 757-763, 1990.

[8] S. Shlien, Nonparametric classification using matched binary decision trees, *Pattern Recognition Letters*, vol. 13, 1992, pp. 83-87.

[9] B.G. Buchanan, E.H. Shortliffe, *Rule Based Expert Systems*, Addison-Wesley, Reading, Massachusetts, 1984.

[10] P.L. Bogler, Shafer-Dempster reasoning with applications to multisensor target identification, *IEEE Trans. Sys. Man. Cyb.*, vol. 17, 1987, pp. 968-977.

[11] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees" *Neural Computation*, vol. 9, 1997, pp. 1545–1588.

[12] P. Latinne, O. Debeir and C. Decaestecker, Limiting the number of trees in random forests, 2nd International Workshop on Multiple Classifier Systems, 2001, pp. 178-187.

[13] S. Salzberg, S., "On comparing classifiers: Pitfalls to avoid and a recommended approach", *Data Mining and knowledge discovery*, vol. 1, 1997, pp. 317–327.

[14] F. Yang, W.H. Lu, L.K. Luo and T. Li, "Margin optimization based pruning for random forest", Neurocomputing, vol. 94, 2012, pp. 54-63.

[15] F. Nan, J. Wang and V. Saligrama, *Pruning random forests for prediction on a budget*, NIPS, 2016.

[16] K. Dheenadayalan, G. Srinivasaraghavan and V.N. Muralidhara, Machine Learning and Data Mining, *Pattern Recognition*, 2016, pp. 516-529,.

[17] M.N. Adnan and M.Z. Islam, Optimizing the number of trees in a decision forest to discover a sub-forest with high ensemble accuracy using a genetic algorithm, *Knowledge-Based Systems*, vol. 110, 2016, pp. 86-97.

[18] X. Jian, C. Wu and H. Guo, *Forest Pruning Based on Branch Importance*, Computational Intelligence and Neuroscience, 2017.

[19] S. Bernard, S. Adam and L. Heutte, Dynamic random forests, *Pattern Recognition Letters*, vol. 33, no 12, 2012, pp. 1580–1586.

[20] E. Tripoliti, D. Fotiadis and G. Manis, *Dynamic Construction of Random Forests: Evaluation using Biomedical Engineering Problems*, IEEE Society, 2010.

[21] V. Kulkarni, P. Sinha and A. Singh, Heuristic Based Improvements for Effective Random Forest Classifier, Proceedings of International Conference on Computational Intelligence, Chennai, India, 2012.

[22] L.I. Kuncheva and C. J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, *Machine Learning*, vol. 51, 2003, pp. 181–207.

[23] L. Breiman, Arcing classifiers, *The Annals of Statistics*, vol. 26, no. 3, 1998, pp. 801–849.

[24] P. Geurts, D., Ernst and L. Wehenkel, Extremely randomized trees, *Machine Learning*, vol. 63, no. 1, 2006, pp. 3-42.

[25] J. Demsar, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research*, vol. 7, 2006, pp. 1–30.

[26] P. Somol, P. Pudil and J. Kittler, Fast branch and bound algorithms for optimal feature selection, *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, 2004, pp. 900–912.

[27] S. Taleb Zouggar and A. Adla, A New *Function for Ensemble Pruning*, In F. Dargam et al. (Eds): ICDSST 2018, LNBIP 313, Springer International Publishing AG, 2018, pp. 181-190.

[28] Taleb Zouggar, S., Adla, A., 2018.A Diversity-Accuracy Measure for Homogenous Ensemble Selection, *International Journal of Interactive Multimedia and Artificial Intelligence*, 2018.