How Many Trees in a Random Forest?

Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas

Department of Computer Science and Mathematics
Faculty of Philosophy, Sciences and Languages at Ribeirao Preto
University of Sao Paulo
{thaismayumi,pedrosperez,augusto}@usp.br

Abstract. Random Forest is a computationally efficient technique that can operate quickly over large datasets. It has been used in many recent research projects and real-world applications in diverse domains. However, the associated literature provides almost no directions about how many trees should be used to compose a Random Forest. The research reported here analyzes whether there is an optimal number of trees within a Random Forest, i.e., a threshold from which increasing the number of trees would bring no significant performance gain, and would only increase the computational cost. Our main conclusions are: as the number of trees grows, it does not always mean the performance of the forest is significantly better than previous forests (fewer trees), and doubling the number of trees is worthless. It is also possible to state there is a threshold beyond which there is no significant gain, unless a huge computational environment is available. In addition, it was found an experimental relationship for the AUC gain when doubling the number of trees in any forest. Furthermore, as the number of trees grows, the full set of attributes tend to be used within a Random Forest, which may not be interesting in the biomedical domain. Additionally, datasets' density-based metrics proposed here probably capture some aspects of the VC dimension on decision trees and low-density datasets may require large capacity machines whilst the opposite also seems to be true.

Keywords: Random Forest, VC Dimension, Number of Trees.

1 Introduction

A great interest in the machine learning research concerns ensemble learning — methods that generate many classifiers and combine their results. It is largely accepted that the performance of a set of many weak classifiers is usually better than a single classifier given the same quantity of train information [28]. Ensemble methods widely known are boosting [12], bagging [8], and more recently Random Forests [7,24].

The boosting method creates different base learners by sequentially reweighting the instances in the training set. At the beginning, all instances are initialized with equal weights. Each instance misclassified by the previous base learner will get a larger weight in the next round, in order to try to classify it correctly. The

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 154–168, 2012.

error is computed, the weight of the correctly classified instances is lowered, and the weight of the incorrectly classified instances is increased. The vote of each individual learner is weighted proportionally to its performance [31].

In the bagging method (bootstrap aggregation), different training subsets are randomly drawn with replacement from the entire training set. Each training subset is fed as input to base learners. All extracted learners are combined using a majority vote. While bagging can generate classifiers in parallel, boosting generates them sequentially.

Random Forest is another ensemble method, which constructs many decision trees that will be used to classify a new instance by the majority vote. Each decision tree node uses a subset of attributes randomly selected from the whole original set of attributes. Additionally, each tree uses a different bootstrap sample data in the same manner as bagging.

Normally, bagging is almost always more accurate than a single classifier, but it is sometimes much less accurate than boosting. On the other hand, boosting can create ensembles that are less accurate than a single classifier. In some situations, boosting can overfit noisy datasets, thus decreasing its performance. Random Forests, on the other hand, are more robust than boosting with respect to noise; faster than bagging and boosting; their performance is as good as boosting and sometimes better, and they do not overfit [7].

Nowadays, Random Forest is a method of ensemble learning widely used in the literature and applied fields. But the associate literature provides few or no directions about how many trees should be used to compose a Random Forest. In general, the user sets the number of trees in a *trial and error* basis. Sometimes when s/he increases the number of trees, in fact, only more computational power is spent, for almost no performance gain. In this study, we have analyzed the performance of Random Forests as the number of trees grows (from 2 to 4096 trees, and doubling the number of trees at every iteration), aiming to seek out for a number (or a range of numbers) of trees from which there is no more significant performance gain, unless huge computational resources are available for large datasets. As a complementary contribution, we have also analyzed the number (percentage) of attributes appearing within Random Forests of growing sizes.

The remaining of this paper is organized as follows. Section 2 describes some related work. Section 3 describes what Random Tree and Random Forest are and how they work. Section 4 provides some density-based metrics used to group datasets described in Section 5. Section 6 describes the methodology used, and results of the experiments are shown in Section 7. Section 8 presents some conclusions from this work.

2 Related Work

Since Random Forests are efficient, multi-class, and able to handle large attribute space, they have been widely used in several domains such as real-time face recognition [29], bioinformatics [16], and there are also some recent research

in medical domain, for instance [18,6,21,19] as well as medical image segmentation [33,22,34,15,32].

A tracking algorithm using adaptive random forests for real-time face tracking is proposed by [29], and the approach was equally applicable to tracking any moving object. One of the first illustrations of successfully analyzing genome-wide association (GWA) data with Random Forests is presented in [16]. Random Forests, support vector machines, and artificial neural network models are developed in [18] to diagnose acute appendicitis. Random Forests are used in [6] to detect curvilinear structure in mammograms, and to decide whether it is normal or abnormal. In [21] it is introduced an efficient keyword based medical image retrieval method using image classification with Random Forests. A novel algorithm for the efficient classification of X-ray images to enhance the accuracy and performance using Random Forests with Local Binary Patterns is presented in [19]. An enhancement of the Random Forests to segment 3D objects in different 3D medical imaging modalities is proposed in [33]. Random Forests are evaluated on the problem of automatic myocardial tissue delineation in real-time 3D echocardiography in [22]. In [34] a new algorithm is presented for the automatic segmentation and classification of brain tissue from 3D MR scans. In [15] a new algorithm is presented for the automatic segmentation of Multiple Sclerosis (MS) lesions in 3D MR images. An automatic 3D Random Forests method which is applied to segment the fetal femur in 3D ultrasound and a weighted voting mechanism is proposed to generate the probabilistic class label is developed in [32].

There is one similar work to the one presented here. In [20] is proposed a simple procedure that *a priori* determine the minimum number of classifiers. They applied the procedure to four multiple classifiers systems, among them Random Forests. They used 5 large datasets, and produced forests with a maximum of 200 trees. They concluded that it was possible to limit the number of trees, and this minimum number could vary from one classifier combination method to another. In this study we have evaluated 29 datasets in forests with up to 4096 trees. In addition, we have also evaluated the percentage of attributes used in each forest.

3 Random Trees and Random Forests

Assume a training set T with a attributes, n instances, and define T_k a bootstrap training set sampled from T with replacement, and containing m random attributes ($m \leq a$) with n instances.

A Random Tree is a tree drawn at random from a set of possible trees, with m random attributes at each node. The term "at random" means that each tree has an equal chance of being sampled. Random Trees can be efficiently generated, and the combination of large sets of Random Trees generally lead to accurate models [35,10].

A Random Forest is defined formally as follows [7]: it is a classifier consisting of a collection of tree structured classifiers $\{h_k(\mathbf{x}, T_k)\}$, $k = 1, 2, \ldots, L$, where T_k are independent identically distributed random samples, and each tree casts a unit vote for the most popular class at input $\mathbf{x}$.

As already mentioned, Random Forests employ the same method bagging does to produce random samples of training sets (bootstrap samples) for each Random Tree. Each new training set is built, with replacement, from the original training set. Thus, the tree is built using the new subset, and a random attribute selection. The best split on the random attributes selected is used to split the node. The trees grown are not pruned.

The use of bagging method is justified by two reasons [7]: the use of bagging seems to enhance performance when random attributes are used; and bagging can be used to give ongoing estimates of the generalization error of the combined ensemble of trees, as well as estimates for the strength and correlation. Theses estimates are performed out-of-bag. In a Random Forest, the out-of-bag method works as follows: given a specific training set T, generate bootstrap training sets T_k, construct classifiers $\{h_k(\mathbf{x}, T_k)\}$ and let them vote to create the bagged classifier. For each $(\mathbf{x}, y)$ in the training set, aggregate the votes only over those classifiers for which T_k does not contain $(\mathbf{x}, y)$. This is the out-of-bag classifier. Then the out-of-bag estimate for the generalization error is the error rate of the out-of-bag classifier on the training set.

The error of a forest depends on the strength of the individual trees in the forest, and the correlation between any two trees in the forest. The strength can be interpreted as a measure of performance for each tree. Increasing the correlation increases the forest error rate, and increasing the strength of the individual trees decreases the forest error rate inasmuch as a tree with a low error rate is a strong classifier. Reducing the number of random attributes selected reduces both the correlation and the strength [23].

4 Density-Based Metrics for Datasets

It is well known from the computational learning theory that, given a hypotheses space (in this case, defined by the Random Forest classifier), it is possible to determine the training set complexity (size) for a learner to converge (with high probability) to a successful hypothesis [25, Chap. 7]. This requires knowing the hypotheses space size (i.e., its cardinality) or its capacity provided by the VC dimension [30]. In practice, finding the hypotheses space size or capacity is hard, and only recently an approach has defined the VC dimension for binary decision trees, at least partially, since it was defined in terms of left and right subtrees [4], whereas the gold standard should be defined in terms of the instances space.

On the other hand, datasets (instances space) metrics are much less discussed in the literature. Our concern is, once the hypotheses space is fixed (but its size or its VC dimension are both unknown or infinite), which training sets *seem* to have enough content so that learning could be successful. In a related work we have proposed some class balance metrics [27]. Since in this study we have used

datasets with very different numbers of classes, instances and attributes, they cannot be grouped in some (intuitive) sense using these three dimensions. For this purpose, we suggest here three different metrics, shown in (1), (2), and (3), where each dataset has c classes, a attributes, and n instances.

These metrics have been designed using the following ideas. For a physical object, the density D is its mass divided by its volume. For a dataset, we have considered its mass as the number of instances; its volume given by its attributes. Here we have used the concept the volume of an object (dataset) is understood as its capacity, i.e., the amount of fluid (attributes) that the object could hold, rather than the amount of space the object itself displaces. Under these considerations, we have $D \triangleq \frac{n}{a}$. Since, in general, these numbers vary considerably, a better way to looking at them was using both numbers in the natural logarithmic scale, $D \triangleq \frac{\ln n}{\ln a}$ which lead us to (1). In the next metric we have considered the number of instances (mass) is rarefied by the number of classes, therefore providing (2), and the last one embraces empty datasets (no instances) and datasets without the class label (unsupervised learning).

$$D_1 \triangleq \log_a n \tag{1}$$

$$D_2 \triangleq \log_a \frac{n}{c} \tag{2}$$

$$D_3 \triangleq \log_a \frac{n+1}{c+1} \tag{3}$$

Considering the common assumption in machine learning that $c \leq n$ (in general, $c \ll n$), it is obvious that, for every metric D_i, $D_i \geq 0$, $i = 1, 2, 3$. We considered that if $D_i < 1$, the density is low, and *perhaps* learning from this dataset should be difficult, under the computational point of view. Otherwise, $D_i \geq 1$, the density is high, and learning *may be* easier.

According to [4] the VC dimension of a binary tree is $\mathrm{VC} = 0.7(1 + \mathrm{VC}_l + \mathrm{VC}_r - \log a) + 1.2 \log M$, where VC_l and VC_r represent the VC dimension of its left and right subtrees and M is the number of nodes in the tree. Considering this, our density-based metrics may capture important information about the VC dimension: (i) the number a of attributes is directly expressed in this equation; (ii) since having more classes implies the tree must have more leaves, the number c of classes is related to the number of leaves, and more leaves implies larger M, therefore c is related to M, and probably VC_l and VC_r; (iii) the number n of instances does not appear directly in this expression but it is surely related to VC_l, VC_r, a and/or M, once the VC dimension of a hypotheses space is defined over the instances space [25, Section 7.4.2].

Intuitively, decision trees are able to represent the family of boolean functions and, in this case, the number n of required training instances for a boolean attributes is $n = 2^a$, and therefore $a = \log_2 n$; in other words, n is related to a as well as M, since more nodes are necessary for larger a values. For these problems expressed by boolean functions with $a \geq 2$ attributes and $n = 2^a$ instances,

Table 1. Density-based metrics for binary class problems ($c = 2$) expressed by boolean functions with a attributes and $n = 2^a$ instances

a	n	D_1	D_2	D_3
2	4	2.00	1.00	**0.74**
3	8	1.89	1.26	1.00
4	16	2.00	1.50	1.25
5	32	2.15	1.72	1.49

$D_i \geq 1$ (except $D_3 = 0.74$ for $a = 2$), according to Table 1. Nevertheless, the proposed metrics are able to capture the fact binary class problems have high-density, indicating there is, probably, enough content so learning can take place.

5 Datasets

The experiments reported here used 29 datasets, all representing real medical data, and none of which had missing values for the class attribute. The biomedical domain is of particular interest since it allows one to evaluate Random Forests under real and difficult situations often faced by human experts.

Table 2 shows a summary of the datasets, and the corresponding density metrics defined in the previous section. Datasets are ordered according to metric D_2, obtaining 8 low-density and 21 high-density datasets. In the remaining of this section, a brief description of each dataset is provided.

Breast Cancer, Lung Cancer, CNS (Central Nervous System Tumour Outcome), *Lymphoma, GCM* (Global Cancer Map), *Ovarian 61902, Leukemia, Leukemia nom., WBC* (Wisconsin Breast Cancer), *WDBC* (Wisconsin Diagnostic Breast Cancer), *Lymphography* and *H. Survival* (*H.* stands for *Haberman's* are all related to cancer and their attributes consist of clinical, laboratory and gene expression data. *Leukemia* and *Leukemia nom.* represent the same data, but the second one had its attributes discretized [26]. *C. Arrhythmia* (*C.* stands for *Cardiac*), *Heart Statlog, HD Cleveland, HD Hungarian* and *HD Switz.* (*Switz.* stands for *Switzerland*) are related to heart diseases and their attributes represent clinical and laboratory data. *Allhyper, Allhypo, ANN Thyroid, Hypothyroid, Sick* and *Thyroid 0387* are a series of datasets related to thyroid conditions. *Hepatitis* and *Liver Disorders* are related to liver diseases, whereas *C. Method* (*C.* stands for *Contraceptive*), *Dermatology, Pima Diabetes* (Pima Indians Diabetes) and *P. Patient* (*P.* stands for *Postoperative*) are other datasets related to human conditions. *Splice Junction* is related to the task of predicting boundaries between exons and introns. Datasets were obtained from the UCI Repository [11], except *CNS, Lymphoma, GCM* and *ECML* were obtained from [2]; *Ovarian 61902* was obtained from [3]; *Leukemia* and *Leukemia nom.* were obtained from [1].

Table 2. Summary of the datasets used in the experiments, where n indicates the number of instances; c represents the number of classes; a, $a_{\#}$ and a_a indicates the total number of attributes, the number of numerical and the number of nominal attributes, respectively; MISS represents the percentage of attributes with missing values, not considering the class attribute; the last 3 columns are the density metrics D_1, D_2, D_3 of each dataset, respectively. Datasets are in ascending order by D_2.

Dataset	n	c	$a(a_{\#},a_a)$	MISS	D_1	D_2	D_3
GCM	190	14	16063 (16063, 0)	0.00%	0.54	0.27	0.26
Lymphoma	96	9	4026 (4026, 0)	5.09%	0.55	0.28	0.27
CNS	60	2	7129 (7129, 0)	0.00%	0.46	0.38	0.34
Leukemia	72	2	7129 (7129, 0)	0.00%	0.48	0.40	0.36
Leukemia nom.	72	2	7129 (7129, 0)	0.00%	0.48	0.40	0.36
Ovarian 61902	253	2	15154 (15154, 0)	0.00%	0.57	0.50	0.46
Lung Cancer	32	3	56 (0, 56)	0.28%	0.86	0.59	0.52
C. Arrhythmia	452	16	279 (206, 73)	0.32%	1.08	0.59	0.58
Dermatology	366	6	34 (1, 33)	0.06%	1.67	1.17	1.12
HD Switz.	123	5	13 (6, 7)	17.07%	1.88	1.25	1.18
Lymphography	148	4	18 (3, 15)	0.00%	1.73	1.25	1.17
Hepatitis	155	2	19 (6, 13)	5.67%	1.71	1.48	1.34
HD Hungarian	294	5	13 (6, 7)	20.46%	2.21	1.59	1.52
HD Cleveland	303	5	13 (6, 7)	0.18%	2.22	1.60	1.53
P. Patient	90	3	8 (0, 8)	0.42%	2.16	1.63	1.50
WDBC	569	2	30 (30, 0)	0.00%	1.86	1.66	1.54
Splice Junction	3190	3	60 (0, 60)	0.00%	1.97	1.70	1.63
Heart Statlog	270	2	13 (13, 0)	0.00%	2.18	1.91	1.75
Allhyper	3772	5	29 (7, 22)	5.54%	2.44	1.97	1.91
Allhypo	3772	4	29 (7, 22)	5.54%	2.44	2.03	1.97
Sick	3772	2	29 (7, 22)	5.54%	2.44	2.24	2.12
Breast Cancer	286	2	9 (0, 9)	0.35%	2.57	2.26	2.07
Hypothyroid	3163	2	25 (7, 18)	6.74%	2.50	2.29	2.16
ANN Thyroid	7200	3	21 (6, 15)	0.00%	2.92	2.56	2.46
WBC	699	2	9 (9, 0)	0.25%	2.98	2.66	2.48
C. Method	1473	3	9 (2, 7)	0.00%	3.32	2.82	2.69
Pima Diabetes	768	2	8 (8, 0)	0.00%	3.19	2.86	2.67
Liver Disorders	345	2	6 (6, 0)	0.00%	3.26	2.87	2.65
H. Survival	306	2	3 (2, 1)	0.00%	5.21	4.58	4.21

6 Experimental Methodology

Using the open source machine learning Weka [17], experiments were conducted building Random Forests with varying number of trees in exponential rates using base two, i.e., $L = 2^j$, $j = 1, 2, \ldots, 12$. Two measures to analyze the results were chosen: the weighted average area under the ROC curve (AUC), and the percentage of attributes used in each Random Forest. To assess performance, the experiments used ten repetitions of 10-fold cross-validation. The average of all repetitions for a given forest on a certain dataset was taken as the value of performance (AUC and percentage) for the pair.

In order to analyze if the results were significantly different, we applied the Friedman test [13], considering a significance level of 5%. If the Friedman test rejects the null hypothesis, a *post-hoc* test is necessary to check in which classifier pairs the differences actually are significant [9]. The *post-hoc* test used was the Benjamini-Hochberg [5], and we performed an all versus all comparison, making

all possible comparisons among the twelve forests. The tests were performed using the R software for statistical computing (`http://www.r-project.org/`).

7 Results and Discussion

The AUC values obtained for each dataset, and each number of trees within a Random Forest are presented in Table 3. We also provide in this table mean and median figures as well as the average rank obtained in the Friedman test. Mean, median and average rank are presented for the following groups: all datasets; only the 8 low-density datasets; and only the 21 high-density ones.

As can be seen, in all groups (all/8 low-density/21 high-density) the forest with 4096 trees has the smallest (best) rank of all. Besides, in the 21 high-density group, we can observe the forests with 2048 and 4096 trees have the same rank. Analyzing the group using all datasets and the 8 low-density datasets, we can notice that the forest with 512 trees has a better rank than the forest with 1024 trees, contrary to what would be expected. Another interesting result concerns mean and median values of high-density datasets for each one of the first three iterations, $L = 2, 4, 8$, are larger than low-density ones; the contrary is true for $L = 16, \ldots, 4096$. This may suggest low-density datasets, in fact, require more expressiveness power (larger forests) than high-density ones. This expressiveness power, of course, can be expressed as the Random Forests (hypotheses) space size or its VC dimension, as explained in Section 4.

In order to get a better understanding, AUC values are also presented in boxplots in Figures 1, 2 and 3 considering all datasets, only the 8 low-density datasets and only the 21 high-density datasets, respectively. As can be seen, in Figures 1 and 2, both mean and median increase as the number of trees grows, but from 64 trees and beyond these figures do not present major changes. In Figure 3, mean and median do not present major changes from 32 trees and 16 trees, respectively.

With these results we can notice an asymptotical behavior, where increases in the AUC values are harder to obtain, even doubling the number of trees within a forest. One way to comprehend this asymptotical behavior is computing the AUC difference from one iteration to the next (for instance, from 2 to 4, 4 to 8, etc.). These results are presented in Figures 4, 5 and 6 for all, 8 low-density and 21 high-density datasets, respectively. For this analysis, we have excluded AUC differences from datasets reaching AUC value equal to 99.99% before 4096 trees (boldface figures in Table 3). Analyzing them, we can notice that using all datasets and the 8 low-density datasets AUC differences (mean and median) between 32 and 64 trees in the forest are below 1%. Considering the 21 high-density datasets, these differences are below 1% between 16 and 32 trees in the forest, and below 0.3% between 32 and 64 trees.

Analyzing Figures 4, 5 and 6, we have adjusted mean and median values by least squares fit to the curve $g = aL^b$, where g represents the percentage AUC difference (gain), and L is the number of trees within the forest. We have obtained

Table 3. AUC values, mean, median and average rank obtained in the experiments. Boldface figures represent values excluded from the AUC difference analysis.

Datasets	Number of Trees											
	2	4	8	16	32	64	128	256	512	1024	2048	4096
GCM	0.72	0.77	0.83	0.87	0.89	0.91	0.91	0.92	0.92	0.92	0.93	0.93
Lymphoma	0.85	0.92	0.96	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99
CNS	0.50	0.52	0.56	0.58	0.59	0.59	0.59	0.58	0.60	0.60	0.60	0.60
Leukemia	0.76	0.85	0.93	0.97	0.98	0.98	0.99	0.99	0.99	0.99	0.99	1.00
Leukemia nom.	0.72	0.81	0.91	0.96	0.99	1.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Ovarian 61902	0.90	0.96	0.98	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
Lung Cancer	0.58	0.64	0.66	0.65	0.65	0.66	0.66	0.68	0.69	0.68	0.68	0.69
C. Arrhythmia	0.71	0.77	0.82	0.85	0.87	0.88	0.89	0.89	0.89	0.89	0.89	0.89
Dermatology	0.97	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
HD Switz.	0.55	0.55	0.58	0.58	0.60	0.61	0.60	0.60	0.60	0.61	0.61	0.61
Lymphography	0.82	0.87	0.90	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
Hepatitis	0.76	0.80	0.83	0.84	0.85	0.85	0.85	0.85	0.86	0.85	0.86	0.86
HD Hungarian	0.80	0.84	0.86	0.87	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
HD Cleveland	0.80	0.84	0.87	0.88	0.89	0.89	0.90	0.89	0.89	0.89	0.90	0.90
P. Patient	0.45	0.45	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
WDBC	0.96	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Splice Junction	0.87	0.93	0.97	0.99	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00
Heart Statlog	0.80	0.84	0.87	0.89	0.89	0.89	0.90	0.90	0.90	0.90	0.90	0.90
Allhyper	0.89	0.95	0.98	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Allhypo	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Sick	0.92	0.97	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Breast Cancer	0.60	0.63	0.64	0.65	0.65	0.66	0.66	0.67	0.66	0.66	0.66	0.66
Hypothyroid	0.95	0.97	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
ANN Thyroid	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
WBC	0.97	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
C. Method	0.62	0.64	0.66	0.66	0.67	0.67	0.67	0.68	0.68	0.68	0.68	0.68
Pima Diabetes	0.72	0.76	0.79	0.81	0.81	0.82	0.82	0.82	0.82	0.82	0.83	0.83
Liver Disorders	0.66	0.70	0.72	0.74	0.75	0.76	0.76	0.77	0.77	0.77	0.77	0.77
H. Survival	0.58	0.60	0.61	0.62	0.63	0.63	0.64	0.64	0.64	0.64	0.64	0.64
All												
Mean	0.77	0.81	0.84	0.85	0.86	0.86	0.86	0.87	0.87	0.87	0.87	0.87
Median	0.80	0.84	0.87	0.89	0.89	0.91	0.91	0.92	0.92	0.92	0.93	0.93
Average rank	11.83	10.55	8.79	8.05	6.88	5.81	5.12	4.62	4.31	4.39	3.91	3.72
8 low-density												
Mean	0.72	0.78	0.83	0.85	0.87	0.88	0.88	0.88	0.88	0.88	0.89	0.89
Median	0.72	0.79	0.87	0.91	0.93	0.94	0.95	0.96	0.96	0.96	0.96	0.96
Average rank	12.00	11.00	9.62	8.81	7.94	6.25	4.81	4.44	3.37	3.69	3.37	2.69
21 high-density												
Mean	0.79	0.82	0.84	0.85	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86
Median	0.80	0.84	0.87	0.89	0.89	0.89	0.90	0.90	0.90	0.90	0.90	0.90
Average rank	11.76	10.38	8.47	7.76	6.47	5.64	5.24	4.69	4.66	4.66	4.12	4.12

(for all datasets) using the median AUC difference $a = 6.42$ and $b = -0.83$ with correlation coefficient $R^2 = 0.99$, and using the mean AUC difference $a = 6.06$ and $b = -0.65$ with correlation coefficient $R^2 = 0.98$. For practical purposes, it is possible to approximate to $g \backsimeq \frac{7}{L}\%$ with correlation coefficient $R^2 = 0.99$, which indicates that this is a good fit as well. For instance, having $L = 8$ trees with AUC equals 0.90 its possible to estimate the AUC for 16 trees (doubling L), therefore $g \backsimeq \frac{7}{8}\%$ and the expected AUC value for 16 trees is $0.90 \times (1 + \frac{7/8}{100}) \backsimeq 0.91$. Of course, this formula may be used with any positive number of trees, for example, having an forest of 100 trees, the expected gain in AUC for a forest with 200 trees is 0.07%.

In Table 4 are presented the results of the *post-hoc* test after the Friedman's test, and the rejection of the null hypothesis. It shows the results using all datasets, the 8 low-density datasets, and the 21 high-density datasets. In this table $\triangle$ ($\blacktriangle$) indicates the Random Forest at the specified row is better (significantly) than the Random Forest at the specified column; $\triangledown$ ($\blacktriangledown$) the Random Forest at the specified row is worse (significantly) than the Random Forest at the specified column; $\circ$ indicates no difference whatsoever.

Some important observations can be made from Table 4. First, we can observe that there is no significant difference between a given number of trees (2^j) and its double (2^{j+1}), in all cases. When there is a significant difference, it only appears when we compare the number of trees (2^j) with at least four times this number (2^{j+2}). Second, from $64 = 2^6$ a significant difference was found only at $4096 = 2^{12}$, only when the Random Forest grew sixty four times. Third, it can be seen that from $128 = 2^7$ trees, there is no more significant difference between the forests until 4096 trees.

In order to analyze the percentage of attributes used, boxplots of these experiments are presented in Figures 7, 8 and 9 for all datasets, the 8 low-density datasets, and the 21 high-density datasets, respectively. Considering Figure 7, the mean and median values from 128 trees corresponds to 80.91% and 99.64% of the attributes, respectively. When we analyze the 8 low-density datasets in Figure 8, it is possible to notice that even with 4096 trees in the forest, not all attributes were used. However, as can be seen, this curve has a different shape (sigmoidal) than those in Figures 7 and 9 (exponential). Also, the sigmoidal seems to grow up to its maximum at 100%.

Even Random Forests do not overtrain, this appear to be a unwanted side effect of them, for instance, datasets of gene expression have thousands genes, and in that case a large forest will use all the genes, even if not all are important to learn the biological/biomedical concept. In [26], trees have only 2 genes among 7129 genes expression values; and in [14] the aim of their work was to build classifiers composed by rules with few conditions, and when they use the same dataset with 7129 genes they only use 2 genes in their subgroup discovery strategy. Considering the 21 high-density datasets in Figure 9, from 8 trees the mean and median already corresponds to 96.18% and 100% of attributes, respectively.

Table 4. Friedman's test results for AUC values using all datasets/8 low-density datasets/21 high-density datasets

Number of Trees	2	4	8	16	32	64	128	256	512	1024	2048	4096
2	○	▽ / ▽ / ▽	▼ / ▽ / ▼ ▼	/ ▽ / ▼ ▼	/ ▽ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼
4		○	▽ / ▽ / ▽	▼ / ▽ / ▼ ▼	/ ▽ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼
8			○	▽ / ▽ / ▽ ▽	/ ▽ / ▽ ▼	/ ▽ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼
16				○	▽ / ▽ / ▽	▼ / ▽ / ▽	▼ / ▽ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼
32					○	▽ / ▽ / ▽	▽ / ▽ / ▼	/ ▽ / ▽ ▼	/ ▼ / ▽ ▼	/ ▼ / ▽ ▼	/ ▼ / ▽	▼ / ▼ / ▽
64						○	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽	▼ / ▽ / ▽	▽ / ▽ / ▽
128							○	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽
256								○	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽
512									○	△ / △ / ○	▽ / ○ / ▽	▽ / ▽ / ▽
1024										○	▽ / ▽ / ▽	▽ / ▽ / ▽
2048											○	▽ / ▽ / ○
4096												○

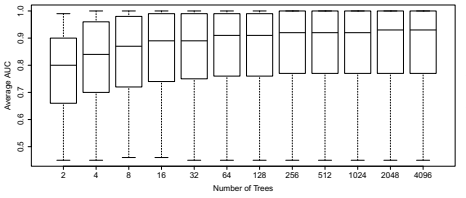

Fig. 1. AUC in all datasets

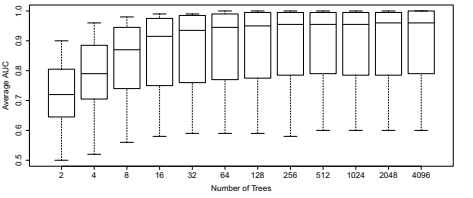

Fig. 2. AUC in the 8 low-density datasets

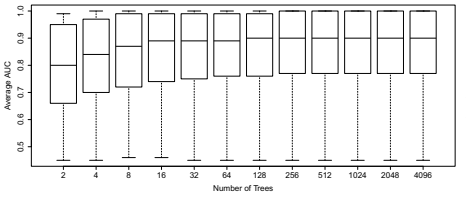

Fig. 3. AUC in the 21 high-density datasets

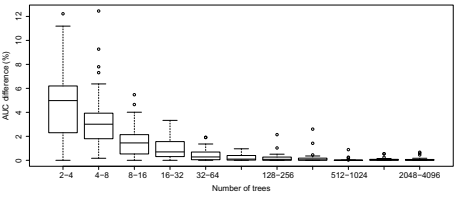

Fig. 4. AUC differences in all datasets

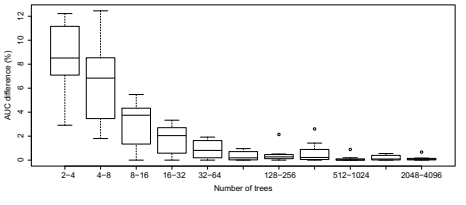

Fig. 5. AUC differences in the 8 low-density datasets

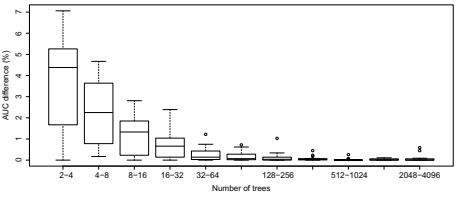

Fig. 6. AUC differences in the 21 high-density datasets

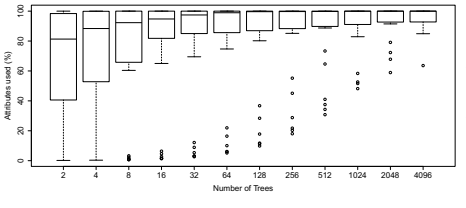

Fig. 7. Percentage of attributes used in all datasets

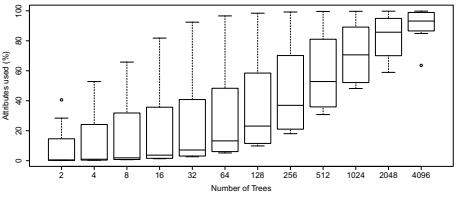

Fig. 8. Percentage of attributes used in the 8 low-density datasets

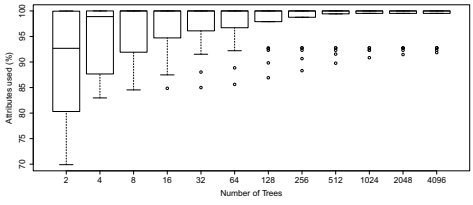

Fig. 9. Percentage of attributes used in the 21 high-density datasets

8 Conclusion

The results obtained show that, sometimes, a larger number of trees in a forest only increases its computational cost, and has no significant performance gain. They also indicate that mean and median AUC tend to converge asymptotically. Another observation is that there is no significant difference between using a number of trees within a Random Forest or its double. The analysis of 29 datasets shows that from 128 trees there is no more significant difference between the forests using 256, 512, 1024, 2048 and 4096 trees. The mean and the median AUC values do not present major changes from 64 trees. Therefore, it is possible to suggest, based on the experiments, a range between 64 and 128 trees in a forest. With these numbers of trees it is possible to obtain a good balance between AUC, processing time, and memory usage. We have also found an experimental relationship (inversely proportional) for AUC gain when doubling the number of trees in any forest.

Analyzing the percentage of attributes used, we can notice that using all datasets, the median reaches the full attribute set with 128 trees in the forest. If the total number of attributes is small, the median reaches the 100% with fewer trees (from 8 trees or more). If this number is larger, it reaches 100% with more trees, in some cases with more than 4096 trees. Thus, asymptotically the tendency indicates the Random Forest will use all attributes, and it is not interesting in some cases, for example in datasets with many attributes (i.e., gene expression datasets), since not all are important for learning the concept [26,14].

We have also proposed density-based metrics for datasets that probably capture some aspects of the VC dimension of decision trees. Under this assumption, low-density datasets may require large capacity learning machines composed by large Random Forests. The opposite also seems to be true.

Acknowledgments. This work was funded by FAPESP (São Paulo Research Foundation) as well as a joint grant between the National Research Council of Brazil (CNPq), and the Amazon State Research Foundation (FAPEAM) through the Program National Institutes of Science and Technology, INCT ADAPTA Project (Centre for Studies of Adaptations of Aquatic Biota of the Amazon).

References

1. Cancer program data sets. Broad Institute (2010), http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi
2. Dataset repository in arff (weka). BioInformatics Group Seville (2010), http://www.upo.es/eps/bigs/datasets.html
3. Datasets. Cilab (2010), http://cilab.ujn.edu.cn/datasets.htm
4. Aslan, O., Yildiz, O.T., Alpaydin, E.: Calculating the VC-dimension of decision trees. In: International Symposium on Computer and Information Sciences 2009, pp. 193–198 (2009)
5. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B 57, 289–300 (1995)
6. Berks, M., Chen, Z., Astley, S., Taylor, C.: Detecting and Classifying Linear Structures in Mammograms Using Random Forests. In: Székely, G., Hahn, H.K. (eds.) IPMI 2011. LNCS, vol. 6801, pp. 510–524. Springer, Heidelberg (2011)
7. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
8. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
9. Demšar, J.: Statistical comparison of classifiers over multiple data sets. Journal of Machine Learning Research 7(1), 1–30 (2006)
10. Dubath, P., Rimoldini, L., Süveges, M., Blomme, J., López, M., Sarro, L.M., De Ridder, J., Cuypers, J., Guy, L., Lecoeur, I., Nienartowicz, K., Jan, A., Beck, M., Mowlavi, N., De Cat, P., Lebzelter, T., Eyer, L.: Random forest automated supervised classification of hipparcos periodic variable stars. Monthly Notices of the Royal Astronomical Society 414(3), 2602–2617 (2011), http://dx.doi.org/10.1111/j.1365-2966.2011.18575.x
11. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
12. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning, pp. 123–140. Morgan Kaufmann, Lake Tahoe (1996)
13. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics 11(1), 86–92 (1940)
14. Gamberger, D., Lavrač, N., Zelezny, F., Tolar, J.: Induction of comprehensible models for gene expression datasets by subgroup discovery methodology. Journal of Biomedical Informatics 37, 269–284 (2004)
15. Geremia, E., Menze, B.H., Clatz, O., Konukoglu, E., Criminisi, A., Ayache, N.: Spatial Decision Forests for MS Lesion Segmentation in Multi-Channel MR Images. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010. LNCS, vol. 6361, pp. 111–118. Springer, Heidelberg (2010)
16. Goldstein, B., Hubbard, A., Cutler, A., Barcellos, L.: An application of random forests to a genome-wide association dataset: Methodological considerations and new findings. BMC Genetics 11(1), 49 (2010), http://www.biomedcentral.com/1471-2156/11/49
17. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining Explor. Newsl. 11(1), 10–18 (2009)
18. Hsieh, C., Lu, R., Lee, N., Chiu, W., Hsu, M., Li, Y.J.: Novel solutions for an old disease: diagnosis of acute appendicitis with random forest, support vector machines, and artificial neural networks. Surgery 149(1), 87–93 (2011)

19. Kim, S.-H., Lee, J.-H., Ko, B., Nam, J.-Y.: X-ray image classification using random forests with local binary patterns. In: International Conference on Machine Learning and Cybernetics 2010, pp. 3190–3194 (2010)
20. Latinne, P., Debeir, O., Decaestecker, C.: Limiting the Number of Trees in Random Forests. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 178–187. Springer, Heidelberg (2001)
21. Lee, J.H., Kim, D.Y., Ko, B.C., Nam, J.Y.: Keyword annotation of medical image with random forest classifier and confidence assigning. In: International Conference on Computer Graphics, Imaging and Visualization, pp. 156–159 (2011)
22. Lempitsky, V., Verhoek, M., Noble, J.A., Blake, A.: Random Forest Classification for Automatic Delineation of Myocardium in Real-Time 3D Echocardiography. In: Ayache, N., Delingette, H., Sermesant, M. (eds.) FIMH 2009. LNCS, vol. 5528, pp. 447–456. Springer, Heidelberg (2009)
23. Leshem, G.: Improvement of adaboost algorithm by using random forests as weak learner and using this algorithm as statistics machine learning for traffic flow prediction. Research proposal for a Ph.D. Thesis (2005)
24. Liaw, A., Wiener, M.: Classification and regression by randomforest. R News 2/3, 1–5 (2002)
25. Mitchell, T.M.: Machine Learning. McGraw-Hill (1997)
26. Netto, O.P., Nozawa, S.R., Mitrowsky, R.A.R., Macedo, A.A., Baranauskas, J.A.: Applying decision trees to gene expression data from dna microarrays: A leukemia case study. In: XXX Congress of the Brazilian Computer Society, X Workshop on Medical Informatics, p. 10. Belo Horizonte, MG (2010)
27. Perez, P.S., Baranauskas, J.A.: Analysis of decision tree pruning using windowing in medical datasets with different class distributions. In: Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM), Athens, Greece, pp. 28–39 (2011)
28. Sirikulviriya, N., Sinthupinyo, S.: Integration of rules from a random forest. In: International Conference on Information and Electronics Engineering, vol. 6, pp. 194–198 (2011)
29. Tang, Y.: Real-Time Automatic Face Tracking Using Adaptive Random Forests. Master's thesis, Department of Electrical and Computer Engineering McGill University, Montreal, Canada (June 2010)
30. Vapnik, V., Levin, E., Cun, Y.L.: Measuring the vc-dimension of a learning machine. Neural Computation 6, 851–876 (1994)
31. Wang, G., Hao, J., Ma, J., Jiang, H.: A comparative assessment of ensemble learning for credit scoring. Expert Systems with Applications 38, 223–230 (2011)
32. Yaqub, M., Mahon, P., Javaid, M.K., Cooper, C., Noble, J.A.: Weighted voting in 3d random forest segmentation. Medical Image Understanding and Analysis (2010)
33. Yaqub, M., Javaid, M.K., Cooper, C., Noble, J.A.: Improving the Classification Accuracy of the Classic RF Method by Intelligent Feature Selection and Weighted Voting of Trees with Application to Medical Image Segmentation. In: Suzuki, K., Wang, F., Shen, D., Yan, P. (eds.) MLMI 2011. LNCS, vol. 7009, pp. 184–192. Springer, Heidelberg (2011)
34. Yi, Z., Criminisi, A., Shotton, J., Blake, A.: Discriminative, Semantic Segmentation of Brain Tissue in MR Images. In: Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (eds.) MICCAI 2009. LNCS, vol. 5762, pp. 558–565. Springer, Heidelberg (2009)
35. Zhao, Y., Zhang, Y.: Comparison of decision tree methods for finding active objects. Advances in Space Research 41, 1955–1959 (2008)