ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

Random Forests Model Selection

Ilenia Orlandi*, Luca Oneto, and Davide Anguita

DIBRIS - University of Genova
Via Opera Pia 13, I-16145 Genova - Italy

Abstract. Random Forests (RF) of tree classifiers are a popular ensemble method for classification. RF have shown to be effective in many different real world classification problems and nowadays are considered as one of the best learning algorithms in this context. In this paper we discuss the effect of the hyperparameters of the RF over the accuracy of the final model, with particular reference to different theoretically grounded weighing strategies of the tree in the forest. In this way we go against the common misconception which considers RF as an hyperparameter-free learning algorithm. Results on a series of benchmark datasets show that performing an accurate Model Selection procedure can greatly improve the accuracy of the final RF classifier.

1 Introduction

It is well known that combining the output of several classifiers results in a much better performance than using any one of them alone [1, 2]. In fact, many state-of-the-art algorithms search for a weighted combination of simpler classifiers [3]: Bagging [1], Boosting [4] and Bayesian approaches [5] or even Neural Networks (NN) [6] and Kernel methods such as Support Vector Machines (SVM) [7]. Optimising the generalisation performance of the final model still represents an unsolved problem. How do we build these simple classifiers? How many simple classifiers do we have to combine? How can we combine them? Is there any theory which can support us in making these choices?

In [1] Breiman tried to give an answer to these questions by proposing the Random Forests (RF) of tree classifiers, one of the state-of-the-art algorithm for classification which has shown to be probably one of the most effective tool in this context [8]. RF combine bagging to random subset feature selection. In bagging, each tree is independently constructed using a bootstrap sample of the dataset [9]. RF add an additional layer of randomness to bagging. In addition to constructing each tree using a different bootstrap sample of the data, RF change how the classification trees are constructed. In standard trees, each node is split using the best division among all variables. In a RF, each node is split using the best among a subset of predictors randomly chosen at that node. Eventually, a simple majority vote is taken for prediction. In [1] it is shown that the accuracy of the final model depends mainly on three different factors: how many trees compose the forest, the accuracy of each tree and the correlation between them.

*This research has been partially supported by MIUR, the Italian Ministry of University and Research, through the Project SMART MANUFACTURING 2020 of the Smart Factory National Technological Cluster. Ilenia Orlandi has also been partly supported under a Ph.D. grant from Infinity Technology Solutions S.r.l.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

The accuracy for RF converges to a limit as the number of trees in the forest increases, while it rises as the accuracy of each tree increases and the correlation between them decreases. RF counterintuitive learning strategy turns out to perform very well compared to many other classifiers, including NN and SVM, and is robust against overfitting [1, 8].

A common misconception about RF is to consider this algorithm as an hyperparameter-free learning algorithm [10, 11]. In fact, there are several hyperparameters which characterise the performance of the final model: the number of trees, the number of samples to extract during the bootstrap procedure, the depth of each tree, the number of predictors exploited in each subset during the growth of each tree, and finally the weights assigned to each tree. For this reason we will show that a Model Selection (MS) procedure is needed in order to select the set of hyperparameters [12] which allow to build a RF model characterised by the best generalisation performances. Results on a series of benchmark datasets show that an accurate MS procedure over these hyperparameters can remarkably improve the accuracy of the final RF model.

2 Hyperparameters in Random Forests

Let us recall the multi-class classification problem [6] where a set of labeled samples $\mathcal{D}_n = \{(X_1, Y_1), \cdots, (X_n, Y_n)\}$ drawn i.i.d. according to an unknown probability distribution μ over $\mathcal{X} \times \mathcal{Y}$ are available and where $X \in \mathcal{X}^d$ and $Y \in \mathcal{Y} = \{1, 2, \cdots, c\}$. A learning algorithm $\mathscr{A}$ maps $\mathcal{D}_n$ into a function belonging to a possibly unknown set of functions $f \in \mathcal{F}$ according to some criteria $\mathscr{A} : \mathcal{D}_n \to \mathcal{F}$. The error of f in approximating μ is measured with reference to a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Since we are dealing with classification problems we choose the loss function which counts the number of misclassified samples $\ell(f(X), Y) = [f(X) \neq Y]$. The expected error of f in representing μ is called generalization error [7] and it is defined as $L(f) = \mathbb{E}_{(X,Y)}\ell(f(X), Y)$. Since μ is unknown $L(f)$ cannot be computed, but we can compute its empirical estimator, the empirical error, defined as $\widehat{L}(f) = {}^1\!/_n \sum_{i=1}^{n} \ell(f(X_i), Y_i)$. The RF learning and classification phases are reported in Algorithm 1. The learning phase of each of the n_t trees composing the RF is quite simple. From $\mathcal{D}_n$, $\lfloor bn \rfloor$ samples are sampled with replacement and $\mathcal{D}'_{\lfloor bn \rfloor}$ is built. A tree is constructed with $\mathcal{D}'_{\lfloor bn \rfloor}$ but the best split is chosen among a subset of n_v predictors over the possible d predictors randomly chosen at each node. The tree is grown until the node contains a maximum of n_l samples. During the classification phase of a previously unseen X, each tree classifies X in a class $Y_{i \in \{1, \cdots, n_t\}}$, and then the final classification is the $\{p_1, \cdots, p_{n_t}\}$-weighted combination of all the answers of each tree of the RF. If $b = 1$, $n_v = \sqrt{n}$, $n_l = 1$ and $p_{i \in \{1, \cdots, n_t\}} = 1$ we get the original RF formulation [1] where n_t is usually chosen to tradeoff accuracy and efficiency [13] or based on the out-of-bag estimate [1] or according to some consistency result [13].

In this paper we argue that performing a MS procedure over b, n_v and n_l and choosing a different weighting procedure can remarkably improve the accuracy

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

of the final model. In particular we exploit the Bootstrap (BOO) MS procedure [12] in order to select the best values for b, n_v and n_l. The smaller b, n_v and n_l are, the more independent are the trees in the RF, but also the lower will be the accuracy of each one of the trees in the RF. Obviously, there is a tradeoff which produces an optimal RF classifier. Besides b, n_v and n_l, the weights $p_{\{i\in 1,\cdots,n_t\}}$ are of paramount importance for the accuracy of an ensemble classifier [3, 2] and for this reason we will compare the original choice of [1] (W_1) with other two state of the art alternatives. One (W_2) is due to [14] and recently studied in [15], while the other one (W_3) has been proposed in [16] and recently further developed in [3]. For what concerns W_2, the proposal is to set $p_i = \ln\left({}^{(1-L(T_i))}/_{L(T_i)}\right)$, where $L(T_i)$ is the accuracy of the tree T_i. Since $L(T_i)$ is unknown, we substitute the empirical error $\widehat{L}(T_i)$ over the out of bag estimate (which is an unbiased estimator of $L(T_i)$). Instead, for what concerns W_3, we have that $p_i = e^{-\gamma\widehat{L}(T_i)}$ where γ is an hyperparameter which must be set as b, n_v and n_l according to the BOO MS procedure. Note that the study of n_t has been deeply investigated in [13], so in this paper we consider it fixed and we study the effect of the other hyperparameters for a given value of n_t.

Algorithm 1: RF learning and classification phases.

```
   /* Learning phase                                                    */
   Input: D_n, n_t, b, n_v and n_l
   Output: A set of tree {T_1, ··· , T_{n_t}}
1  for i ← 1 to n_t do
2  |  D'_{⌊bn⌋} sample with replacement ⌊bn⌋ sample from D_n;
3  |  T_i = DT(D'_{⌊bn⌋}, n_v, n_l);
   /* Classification phase                                              */
   Input: X, n_t, {p_1, ··· , p_{n_t}}
   Output: Y
4  for i ← 1 to n_t do
5  |  Y_i = T_i(X);
6  Y = arg max_{j∈{1,···,c}} Σ_{i∈{1,···,n_t}:Y_i=j} p_i ;
   /* Functions                                                         */
7  function T = DT(D_n, n_v, n_l);
8  if n ≤ n_l then
9  |  T.l = mode({Y ∈ D_n}) ;
10 else
11 |  Split D_n in D'_{n'} and D''_{n''} based on the best predictor s over the n_v ones sampled
   |    from the whole d predictors ;
12 |  T.s = s;  T.T' = DT(D'_{n'}, n_v, n_l);  T.T'' = DT(D''_{n''}, n_v, n_l);
```

3 Results and Discussion

Let us consider a series of biclass and multiclass problems from [17]: BanknoteAuth (D1), Anneal (D2), Parkinson (D3), Wine (D4), Seed (D5), Tic-tac-toe (D6), Car (D7), LSVT (D8), Fertility (D9), Horse (D10), Blogger (D11), Nursey (D12), Segment (D13), HAR (D14), MiceProteins (D15), Audiology

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

(D16), CNAE (D17), Glass (D18), SensorlessDrive (D19), Optdigits (D20), BreastTissue (D21), MovementsLibras (D22), PittsburgBridges (D23), Bach (D24), Cmc (D25), and Yeast (D26). These datasets are commonly used as a benchmark for learning algorithms. For each dataset, analogously to [2], up to a maximum of 500 samples are randomly chosen to be in the training set $\mathcal{D}_n$, and the remaining examples are kept as test set if it is not already available. We set $n_t = 100$ and for the BOO MS procedure we set the number of bootstrap resamples $n_b = 100$ [12]. We compare the following RF models:

- RF with the weighting strategy W_1 ($p_{i\in\{1,\cdots,n_t\}} = 1$ which is the majority vote [1]) in the following cases:
  - STD: we set b, n_v, and n_l at the standard values $b = 1$, $n_v = \sqrt{n}$, and $n_l = 1$ [1];
  - O(b): we set $n_v = \sqrt{n}$ and $n_l = 1$ while $b \in \{0.20, 0.22, \cdots, 1.20\}$ is optimised based on BOO MS procedure;
  - O(n_v): we set $b = 1$ and $n_l = 1$ while $n_v \in d^{\{0.00,0.02,\cdots,1.00\}}$ is optimised based on BOO MS procedure ;
  - O(n_l): we set $b = 1$ and $n_v = \sqrt{n}$ while $n_l \in n\cdot\{0.00, 0.01, \cdots, 0.50\}+1$ is optimised based on BOO MS procedure;
  - ALL: $b \in \{0.20, 0.22, \cdots, 1.20\}$, $n_v \in d^{\{0.00,0.02,\cdots,1.00\}}$, $n_l \in n\cdot\{0.00, 0.01, \cdots, 0.50\}+1$ are optimised based on BOO MS procedure;
- RF with the weighting strategy W_2 ($p_i = \ln\left((1-\widehat{L}(T_i))/\widehat{L}(T_i)\right)$ [14, 15]) in the same sub-configuration depicted from W_1: STD, O(b), O(n_v), O(n_l), and ALL;
- RF with the weighting strategy W_3 ($p_i = e^{-\gamma\widehat{L}(T_i)}$ [16, 3]) where $\gamma \in 10^{\{-6.0,-5.8,\cdots,4\}}$ is optimised based on BOO MS procedure, in the same sub-configuration depicted from W_1: STD, O(b), O(n_v), O(n_l), and ALL;

Table 1 reports on the error on the test set for all the datasets and for all the experimental settings that we have just described while, Table 2 reports on the corresponding value of the optimised hyperparameters which have been selected during the BOO MS procedure.

From Tables 1 and 2 we can draw the following observations:

- optimising the hyperparameters (even just one of them) mostly leads to model characterised by higher accuracy;
- the hyperparameter that has the more noticeable impact on the accuracy of the model is the weighting strategy, followed by the n_v, and then b. Instead, the less important hyperparameter resulted to be n_l;
- the usual choice of W_1, $b = 1$, $n_v = \sqrt{n}$, and $n_l = 1$, which is obviously the most computational inexpensive choice, resulted to be a competitive one. However if one has to choose the best tradeoff, we would suggest W_3, $b = 1$, $n_v = \sqrt{n}$, and $n_l = 1$;
- the weighting strategy W_2 resulted to be the more inaccurate way of weighting the different classifiers even if this weighting strategy has very strong theoretical properties [15];
- the weighting strategy W_3 resulted to be really effective in this context and this supports all its theoretical properties studied in [16, 3];

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

	W_1					W_2					W_3				
$\mathcal{D}_n$	STD	O(b)	O(n_v)	O(n_l)	ALL	STD	O(b)	O(n_v)	O(n_l)	ALL	STD	O(b)	O(n_v)	O(n_l)	ALL
D1	0.33	0.33	0.33	0.33	**0**	0.33	0.33	0.33	0.33	**0**	0.33	0.33	0.33	0.33	**0**
D2	3	2	1	4	**0**	3	2	1	4	**0**	2	1	**0**	2	**0**
D3	3.39	1.69	1.69	3.39	1.69	3.39	1.69	1.69	3.39	1.69	3.39	1.69	1.69	3.39	**0**
D4	3.77	3.77	1.89	5.66	1.89	3.77	3.77	1.89	5.66	**0**	3.77	1.89	1.89	1.89	**0**
D5	4.76	3.17	3.17	4.76	3.17	7.94	3.17	3.17	4.76	3.17	4.76	**1.59**	3.17	3.17	**1.59**
D6	8	8	5.67	7.67	5.33	8	7.67	5.33	8.33	5	7.67	7	4.33	7.67	**3.67**
D7	10.67	9.33	5.33	9.67	5.67	10.67	9.33	5.33	9.33	5.33	10	7.33	4.67	8.33	**4**
D8	15.79	13.16	7.89	13.16	5.26	15.79	10.53	10.53	13.16	5.26	15.79	7.89	7.89	7.89	**2.63**
D9	13.33	10	13.33	13.33	10	13.33	10	13.33	13.33	10	13.33	10	10	10	**3.33**
D10	22.06	17.65	14.71	22.06	11.76	22.06	16.18	14.71	20.59	13.24	16.18	14.71	13.24	11.76	**10.29**
D11	20	16.67	16.67	20	10	20	20	20	23.33	10	20	13.33	16.67	16.67	**6.67**
D12	10.67	10.67	11.33	11.67	10	93.67	10.67	12	74	9.33	10.67	9	9.67	10.33	**8.67**
D13	10.33	7.33	5.67	8.67	4.33	100	8.67	100	97	6.33	10	6.33	5.67	8	**3.67**
D14	12.33	11	11.33	12	**9.67**	100	11.67	100	99	11	11.67	10.33	10.67	12	**9.67**
D15	11.67	13	**0**	29	**0**	100	100	87.33	87.33	**0**	6	**0**	**0**	2.67	**0**
D16	23.08	11.54	11.54	15.38	**3.85**	100	100	100	92.31	7.69	23.08	7.69	11.54	15.38	**3.85**
D17	15	14	14	16.67	13.33	99.67	98.33	89.33	86	16.33	14.67	13.33	13	14.33	**12**
D18	12.5	7.81	7.81	9.38	**6.25**	100	100	100	92.19	62.5	9.38	7.81	7.81	9.38	**6.25**
D19	15	14	13.67	15.67	12.67	100	100	99	94	19.67	14	12.67	12.67	14.33	**11.33**
D20	11.33	11	10	11.67	9.67	100	100	99.67	95.33	88.33	11.33	10.33	10.33	11.67	**9.33**
D21	28.13	28.13	25	31.25	**15.63**	100	96.88	96.88	96.88	75	25	25	21.88	25	**15.63**
D22	26.85	26.85	25	27.78	24.07	100	100	100	97.22	87.96	26.85	26.85	25.93	27.78	**23.15**
D23	34.38	34.38	34.38	34.38	31.25	100	90.63	93.75	81.25	68.75	34.38	31.25	31.25	34.38	**28.13**
D24	29.33	27.67	27.33	29.67	27.33	100	99.67	99.67	97	88.67	29.33	27	27.33	29.67	**26.67**
D25	44	42.33	41.67	44.67	40.67	80.67	77.33	78	76	59	43.67	42.67	41	43.33	**39.33**
D26	43.67	41.33	41.67	42.33	39.67	99.67	99.33	99.33	95.33	72.67	43.33	40	41.67	42	**38.67**

Table 1: Errors (in percentage) on the test set of the different RF models.

- by observing Table 2, it is possible to note that even if the value chosen as hyperparameter in the MS phase differs a lot from the standard one ($b = 1$, $n_v = \sqrt{n}$, and $n_l = 1$), the corresponding accuracy in Table 1 may not vary so much; this means that there are a lot of combinations of the hyperparameters which are able to give good results. This may be useful when it comes to perform a MS phase, since the task of selecting good hyperparameters results simplified;
- for what concerns the choice of the hyperparameters, b results to be the one that is more often different from the conventional choice of $b = 1$, while n_l is mostly near to $n_l = 0$, which is the conventional choice. Note that having $b < 1$ or $n_l > 0$ increases also the speed of the training and classification phase; hence, in the future it will be interesting to check how a computational budget can influence the accuracy of the model.

In conclusion, we can state that we have empirically proved that the different hyperparameters characterising the RF learning algorithm have a remarkable impact on the accuracy of the final model, even the ones that are commonly not considered as hyperparameters (b and n_l). Most of all, choosing a good weighting strategy (in particular W_3) results in being one of the most effective ways to improve the generalisation performance of the final model. This work is a step forward in understanding the learning properties of the RF and more work is needed in this direction. In particular, a deeper empirical analysis is needed and a theoretical characterisation of the different phenomena should be provided. In any case, from the results of this paper, we can definitely state that a MS phase should always be performed when RF is adopted and this MS is always beneficial with just a computational overhead on the learning phase and no overhead over the classification phase.

References

[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

	W_1						W_2						W_3									
$\mathcal{D}_n$	b	n_v	n_l	b	n_v	n_l	b	n_v	n_l	b	n_v	n_l	γ	γ	b	γ	n_v	γ	n_l	γ	b	n_v	n_l
D1	0.28	0.04	0	0.34	0.24	0	0.22	0.04	0	0.34	0.24	0	-6	0.72	0.22	1.72	0	-6	0	-6	0.34	0.24	0
D2	0.64	0.62	0	0.78	0.76	0	0.64	0.62	0	0.78	0.76	0	0.92	2.34	0.7	1.94	0.66	1.52	0	2.14	0.58	0.62	0
D3	0.56	0.44	0.01	0.4	1	0	0.56	0.44	0.01	0.4	1	0	-6	1.32	0.4	-6	0.44	-6	0.01	1.94	0.88	0.14	0.02
D4	0.54	0	0	0.2	0.14	0	0.54	0	0	0.7	0.1	0.05	-6	2.14	0.98	-6	0	1.52	0	1.72	0.36	0.58	0
D5	0.5	0	0	0.2	0.1	0	0.5	0.52	0	0.22	0.44	0	-6	2.34	0.8	-6	0	1.94	0.19	2.34	0.4	0.06	0.02
D6	0.76	0.8	0	0.76	0.72	0	0.76	0.96	0	1	0.94	0	1.12	1.52	0.78	2.14	0.96	1.32	0	1.72	0.88	1	0
D7	0.64	1	0	0.8	1	0	0.64	1	0	0.8	1	0	1.94	1.94	1	1.72	1	1.52	0	2.14	0.94	0.96	0
D8	0.42	0.62	0	0.32	0	0.01	0.94	0.82	0	0.84	1	0.2	-6	1.72	0.5	-6	0.62	1.72	0.14	1.72	0.2	0.8	0
D9	0.44	0	0	0.32	1	0	0.66	0	0	0.34	0.44	0	-6	2.34	0.34	1.32	0.24	2.34	0.03	2.54	0.44	0.28	0.05
D10	0.54	0.62	0.01	0.7	0.8	0.02	0.54	0.62	0.01	0.7	0.8	0.02	1.94	1.72	0.44	1.94	0.82	2.54	0	1.94	0.2	0.62	0
D11	0.42	0.58	0	0.2	0.82	0.01	0.4	0.18	0	0.34	0.8	0.02	-6	1.94	0.88	1.94	0.24	2.14	0.17	-6	0.4	1	0.03
D12	0.86	0.32	0.01	0.7	0.52	0	0.42	0.76	0.3	0.62	0.48	0	0.92	1.94	0.58	1.72	0.44	1.52	0	2.14	0.84	0.68	0.01
D13	0.44	0.18	0	0.54	0.14	0	0.28	0	0.29	0.32	0.32	0	1.52	1.94	0.92	-6	0.18	1.72	0	1.12	0.54	0.14	0
D14	0.88	0.52	0	0.84	0.56	0	0.54	0	0.08	0.4	0.56	0	0.72	1.12	0.54	1.32	0.52	-6	0	-6	0.84	0.56	0
D15	0.22	0.58	0	0.2	0.58	0	0.2	0.04	0.23	0.2	0.62	0	0.92	0.92	0.2	1.32	0.48	1.52	0	-6	0.2	0.58	0
D16	0.34	0.82	0	0.2	0.82	0	0.2	0	0.2	0.2	0.96	0	-6	1.52	0.48	-6	0.82	-6	0	-6	0.2	0.82	0
D17	0.84	0.58	0	0.4	0.56	0	0.36	0	0.17	0.36	0.82	0	0.72	1.12	0.94	1.72	0.62	1.52	0	1.94	0.84	0.62	0
D18	0.98	0.34	0	0.94	0.44	0	0.2	0	0.19	0.2	0	0.1	1.32	-6	0.98	1.12	0.04	-6	0	1.32	0.84	0.48	0
D19	0.64	0.56	0	0.86	0.62	0	0.2	0.06	0.2	0.28	0.86	0	1.52	1.32	0.78	1.72	0.72	1.52	0.01	0.92	1	0.52	0
D20	0.78	0.38	0	0.94	0.38	0	0.2	0	0.25	0.2	0.28	0.22	-6	0.92	0.72	-6	0.38	0.5	0	0.72	0.94	0.38	0
D21	0.62	0.8	0.05	0.86	0.76	0	0.2	0.04	0.02	0.92	0	0.3	0.72	1.72	0.8	1.32	0.68	1.52	0	1.72	0.8	0.38	0.02
D22	0.72	0.38	0	0.62	0.68	0	0.2	0	0.27	0.2	0.34	0.2	-6	-6	0.72	-6	0.04	-6	0	-6	0.94	0.14	0
D23	0.28	0.38	0	0.26	0.52	0.01	0.72	0.76	0.29	0.5	0.28	0.16	-6	1.52	0.58	1.52	0.44	-6	0	2.14	0.42	0.8	0.01
D24	0.54	0.18	0	0.44	0.42	0	0.88	0.04	0.13	0.28	0.34	0.14	-6	1.52	0.54	1.32	0.38	-6	0	1.72	0.64	0.2	0
D25	0.32	0.8	0.01	0.36	1	0	0.5	0.04	0.21	0.2	0.1	0.28	-6	-6	0.32	0.72	0.8	1.94	0.08	1.94	0.94	0.62	0.04
D26	0.5	0.86	0.01	0.32	0.94	0	0.2	0.24	0.29	0.2	0	0.1	0.92	1.94	0.5	1.12	0.52	1.32	0	1.72	1	1	0

Table 2: Value of the hyperpararameters selected with the BOO MS procedure which have produced the model characterized by the error of Table 1.

[2] P. Germain, A. Lacasse, M. Laviolette, A. ahd Marchand, and Roy J. F. Risk bounds for the majority vote: From a pac-bayesian analysis to a learning algorithm. *JMLR*, 16(4):787–860, 2015.

[3] G. Lever, F. Laviolette, and F. Shawe-Taylor. Tighter pac-bayes bounds through distribution-dependent priors. *Theoretical Computer Science*, 473:4–28, 2013.

[4] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[5] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Taylor & Francis, 2014.

[6] B. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[7] V. N. Vapnik. *Statistical learning theory*. Wiley New York, 1998.

[8] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *JMLR*, 15(1):3133–3181, 2014.

[9] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

[10] G. Biau. Analysis of a random forests model. *JMLR*, 13(1):1063–1095, 2012.

[11] S. Bernard, L. Heutte, and S. Adam. Influence of hyperparameters on random forest accuracy. In *International Workshop on Multiple Classifier Systems*, 2009.

[12] D. Anguita, A. Ghio, L. Oneto, and S. Ridella. In-sample and out-of-sample model selection and error estimation for support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, 23(9):1390–1406, 2012.

[13] D. Hernández-Lobato, G. Martínez-Muñoz, and A. Suárez. How large should ensembles of classifiers be? *Pattern Recognition*, 46(5):1323–1336, 2013.

[14] S. Nitzan and J. Paroush. Optimal decision rules in uncertain dichotomous choice situations. *International Economic Review*, 23(2):289–97, 1982.

[15] D. Berend and A. Kontorovitch. Consistency of weighted majority votes. In *NIPS*, 2014.

[16] O. Catoni. *Pac-Bayesian Supervised Classification*. Institute of Mathematical Statistics, 2007.

[17] M. Lichman. UCI machine learning repository, 2013.