Journal of the Korean Data & Information Science Society
2022, **33**(2), 325–335

http://dx.doi.org/10.7465/jkdi.2022.33.2.325
한국데이터정보과학회지

How many trees in a random forest?[†]

Cheolyong Park[1] · Fred W. Huffer[2]

[1]Major in Statistics, Keimyung University
[2]Department of Statistics, Florida State Univrsity

Received 25 January 2022, revised 27 February 2022, accepted 27 February 2022

Abstract

We propose diagnostic statistics which might assist in choosing the size of a random forest for classification. We use these statistics sequentially as we construct the forest. The statistics are computed from out-of-bag or test set votes and give an estimate of expected disagreement between the current and infinite forests. Simulation studies are provided to illustrate the performance of these statistics and to compare them with other methods for choosing the size of a random forest.

Keywords: Binary classification, diagnostic statistics, measure of disagreement, number of trees, random forest.

1. Introduction

In this paper, we are mainly concerned about choosing the proper size of a random forest (Breiman, 2001) for binary classification. This work is a continuation of Huffer and Park (2020)'s which is related to Hernández-Lobato *et al.* (2011, 2013). Their studies start from the observation that it is reasonable to choose the size of the random forest large enough so that its prediction is quite close to that of the infinite forest with infinitely many trees.

For this purpose, Huffer and Park (2020) proposed several statistics, generally denoted by $\widehat{\Delta}_n$, where the current forest with n trees is generated sequentially. These statistics $\widehat{\Delta}_n$ predict measures of disagreement between the current and the infinite forests. Their stopping rule is when $\widehat{\Delta}_n$ falls below some threshold value.

Huffer and Park (2020) focused mainly on monitoring the out-of-bag error rate of the random forest. This idea can be easily extended to the test set version of $\widehat{\Delta}_n$, but the statistics are just intended for the disagreement prediction of a fixed data set. Similar studies in this regard can be found in Park (2016, 2017). In this paper, we further develop diagnostic statistics for the prediction of future cases sampled from the same population from which the training cases were drawn.

[†] This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2019R1F1A1058723).
[1] Corresponding author: Professor, Major in Statistics, Keimyung University, Daegu 42601, Korea. E-mail: cypark1@kmu.ac.kr
[2] Professor, Department of Statistics, Florida State University, FL 32304, U.S.A.

The statistics $\widehat{\Delta}_n$ are easily computed and can be used with other ensemble methods based on a majority vote of individual classifiers which are generated in independent and identically distributed fashion.

This paper is organized as follows. In Section 2, we give a general form of our statistics and then provide mathematical derivations of our statistics. In Section 3, we provide simulation studies to illustrate the performance of our statistics and to compare our statistics with other methods for choosing the size of a random forest for classification.

2. Our methods

We will briefly introduce some notations; these are mostly the same as in Huffer and Park (2020). Let $(x_1, y_1), \ldots, (x_J, y_J)$ be the training sample of size J with $y_i \in \{1, 2, \ldots, r\}$. We want to classify a set $\{x_1, \ldots, x_L\}$ of L unlabeled cases.

Consider a fixed case x and a current random forest with n trees. Let $\hat{\pi}_j$ $(1 \leq j \leq r)$ be the proportion of trees in the current forest which classify the case x as class j. This will be the natural estimate of the true proportion π_j in the infinite forest with infinitely many trees. The case x may be either one of the training or test cases. If x is from training cases, $\hat{\pi}_j$ and π_j are calculated from the out-of-bag votes. If x is from the test cases, all training votes are used.

For case x, let j and k be the two classes with the most votes in the current forest. Define

$$\hat{\delta} = \frac{\hat{\pi}_j - \hat{\pi}_k}{\hat{\pi}_j + \hat{\pi}_k} \quad \text{and} \quad \delta = \frac{\pi_j - \pi_k}{\pi_j + \pi_k}\,. \tag{2.1}$$

Here δ may be thought as the margin of victory in the infinite forest. Note that the current and infinite forests disagree for the case x when $\hat{\delta} \cdot \delta < 0$, and, in this case, it is argued in Huffer and Park (2020) that the value of $|\delta|$ is a reasonable measure of the magnitude of the disagreement. This suggests using

$$D(\delta, \hat{\delta}) = |\delta| I(\hat{\delta} \cdot \delta < 0) \tag{2.2}$$

as an overall measure of the disagreement between the current and infinite forests for the case x.

For each case x_i in a test set $\{x_1, x_2, \ldots, x_L\}$ define $\hat{\delta}_i$ and δ_i as in (2.1). A reasonable strategy for choosing the number of trees is to make the following measure small:

$$\Delta_n = E\left(\frac{1}{L}\sum_{i=1}^{L} D(\delta_i, \hat{\delta}_i)\right), \tag{2.3}$$

where the subscript n of Δ_n in (2.3) denotes the size of the current forest.

There are several situations which lead to four different closely related statistics $\widehat{\Delta}_n^{(1,\mathrm{oob})}$, $\widehat{\Delta}_n^{(1,\mathrm{test})}$, $\widehat{\Delta}_n^{(2,\mathrm{oob})}$, $\widehat{\Delta}_n^{(2,\mathrm{test})}$ for estimating Δ_n. All of these statistics have a general average form over cases in the training or test data:

$$\widehat{\Delta}_n = \frac{1}{L}\sum_{i=1}^{L} S(V_{in}) \tag{2.4}$$

where $S(V_{in})$ is a score computed from the vector of votes $V_{in} = (v_1, v_2, \ldots, v_r)$ for case i from the n trees in the current forest. If the L cases are from the training data, the vectors V_{in} are calculated from the out-of-bag votes and we use 'oob' to indicate this. If the L cases are from the test data, the vectors V_{in} are calculated from votes of all the trees and we use 'test' to indicate this.

Huffer and Park (2020) consider the first situation where we want to classify the fixed set $\{x_1, \ldots, x_L\}$ with L unlabeled cases; we use the superscript '1' to indicate this. In Huffer and Park (2020) it is derived that

$$\widehat{\Delta}_n^{(1,\mathrm{oob})} = \frac{1}{L}\sum_{i=1}^{L} \Psi(-\hat{\delta}_i\,, (1-\hat{\delta}_i^2)/t_i), \tag{2.5}$$

where

$$\Psi(a,b) = \sqrt{\frac{b}{2\pi}}\,\exp\left(-\frac{a^2}{2b}\right) + a\Phi\left(\frac{a}{\sqrt{b}}\right),$$

$\Phi(\cdot)$ is the cumulative distribution of the $N(0,1)$ distribution, and $t_i = v_j + v_k$ where v_j and v_k (with $v_j \geq v_k$) are the votes for classes j and k, the two classes receiving the most out-of-bag votes for case x_i in the current forest (as in (2.1)). Similarly we can compute $\widehat{\Delta}_n^{(1,\mathrm{test})}$ when the votes are computed from the test data.

In this paper, we consider the second situation where a user is constructing a random forest for the purpose of classifying future test cases to be sampled from the same population from which the training cases were drawn. This is the situation considered by Hernández-Lobato *et al.* (2013). Here we would like the forest to be large enough to give good average agreement with the infinite forest for cases sampled from this population. For this situation we describe two related statistics $\widehat{\Delta}_n^{(2,\mathrm{oob})}$ and $\widehat{\Delta}_n^{(2,\mathrm{test})}$, the first based on the out-of-bag votes accumulated while constructing the forest, and the second based on the votes for a test set of labeled cases which are held out from the training data. These two statistics have the form in equation (2.4) and are analogous to the 'oob' and 'test' versions of the procedure in Hernández-Lobato *et al.* (2013). If one has constructed a random forest with n trees and considers using this to classify a new case to be randomly drawn from the same population, the statistics $\widehat{\Delta}_n^{(2,\mathrm{oob})}$ and $\widehat{\Delta}_n^{(2,\mathrm{test})}$ supply estimates of the average disagreement to be expected between this classification and that of the infinite forest. That is, for D in (2.2), they supply estimates of $ED(\delta, \hat{\delta})$ where δ and $\hat{\delta}$ are the random values arising from this new case.

2.1. Derivation of $\widehat{\Delta}_n^{(2,\mathbf{oob})}$ and $\widehat{\Delta}_n^{(2,\mathbf{test})}$

We now discuss the derivation of $\widehat{\Delta}_n^{(2,\mathrm{oob})}$ and $\widehat{\Delta}_n^{(2,\mathrm{test})}$. These statistics are estimates of $ED(\delta, \hat{\delta})$ for a new case sampled from the same population as the training data. We follow Hernández-Lobato *et al.* (2013) and consider a slightly more general situation than previously discussed: Suppose we have training data which is a random sample from some population, and have used this training data to construct a random forest with n trees. We wish to use the prediction results of this forest with n trees to estimate the expected disagreement $ED(\delta, \hat{\delta})$ we would obtain if we were to use a random forest with T trees (constructed from the same training data) to classify a future case randomly chosen from the same population. In the end we shall set $T = n$ to estimate the performance of the forest

we have already constructed; we use our procedure in a sequential fashion, adding trees to the forest and periodically re-computing $ED(\delta, \hat{\delta})$ until we achieve a pre-set goal. The authors in Hernández-Lobato *et al.* (2013) use their procedure in a non-sequential fashion; after constructing a random forest with n trees, they "solve" to obtain a value of T which approximately attains their goal, and then construct a forest with this many trees and re-estimate the needed value of T, etc., homing in on the desired forest size.

Here we follow Hernández-Lobato *et al.* (2013) and assume there are only two classes. This assures that with T trees, when classifying future cases the number of votes for the top two classes is always T. When there are more than two classes, the number of votes for the top two classes is random, which complicates matters.

If the number of trees T is even, then ties can occur (i.e., $\hat{\delta} = 0$). The issue is avoided in Hernández-Lobato *et al.* (2013) by always taking T to be odd. In our paper, we handle ties by slightly modifying the definition of $D(\delta, \hat{\delta})$ in (2.2) as follows:

$$D(\delta, \hat{\delta}) = |\delta| \left\{ I(\hat{\delta} \cdot \delta < 0) + \frac{1}{2} I(\hat{\delta} = 0) \right\} .$$

This corresponds to increasing D by its expectation if the tie were broken at random.

Let x be a new case randomly sampled from the same population the training data were drawn from. For this case x, let π be the probability that a randomly generated tree (using the original training data) votes for class 1, and V be the total number of votes for class 1. With T trees, $V \sim \text{Binomial}(T, \pi)$.

With two classes and T trees, the quantities δ and $\hat{\delta}$ in (2.1) become (taking $j = 1$ and $k = 2$) $\delta = 2\pi - 1$ and $\hat{\delta} = (2V - T)/T$. Conditional on δ, the distribution of $\hat{\delta}$ is determined by $V \,|\, \pi \sim \text{Binomial}(T, \pi)$. This leads to

$$\begin{aligned}
&ED(\delta, \hat{\delta}) \\
&= E\delta I(\delta > 0)\{P(\hat{\delta} < 0 \;\,|\, \delta) + 0.5P(\hat{\delta} = 0 \,|\, \delta)\} \\
&\quad + E(-\delta) I(\delta < 0)\{P(\hat{\delta} > 0 \,|\, \delta) + 0.5P(\hat{\delta} = 0 \,|\, \delta)\} \\
&= E(2\pi - 1) I(\pi > 1/2)\{P(V < T/2 \,|\, \pi) + 0.5P(V = T/2 \,|\, \pi)\} \\
&\quad - E(2\pi - 1) I(\pi < 1/2)\{P(V > T/2 \,|\, \pi) + 0.5P(V = T/2 \,|\, \pi)\} \\
&= E(2\pi - 1)[P(V < T/2 \,|\, \pi) + 0.5P(V = T/2 \,|\, \pi) - I(\pi < 1/2)] \\
&= \int_0^1 (2\pi - 1)[P(V < T/2 \,|\, \pi) + 0.5P(V = T/2 \,|\, \pi) - I(\pi < 1/2)] f(\pi)\, d\pi \,. \qquad (2.6)
\end{aligned}$$

We have introduced $f(\pi)$ to denote the density of π in the population of cases we are sampling from. Equation (2.6) is analogous to the integral in equation (15) of Hernández-Lobato *et al.* (2013).

In order to estimate $ED(\delta, \hat{\delta})$ we must first come up with a reasonable estimate of $f(\pi)$ based on the random forest with n trees we have already constructed. The simplest approach is to follow Hernández-Lobato *et al.* (2013) and use the empirical distribution of the observed fractions $\hat{\pi}^{(i)}$, $i = 1, \ldots, L$, of votes for class 1 for the cases in a test set (or OOB votes for class 1 for the cases in the training data) and estimate (2.6) by

$$\frac{1}{L} \sum_{i=1}^{L} (2\hat{\pi}^{(i)} - 1)[P(V < T/2 \,|\, \hat{\pi}^{(i)}) + 0.5P(V = T/2 \,|\, \hat{\pi}^{(i)}) - I(\hat{\pi}^{(i)} < 1/2)] \,.$$

This is analogous to equation (17) in Hernández-Lobato *et al.* (2013). Instead of this approach, we estimate the density $f(\pi)$ by an equally weighted mixture of the posterior densities of $\pi^{(i)}$, $i = 1, \ldots, L$, developed in equation (19) of Huffer and Park (2020). This leads to an estimate for (2.6) given by

$$\frac{1}{L}\sum_{i=1}^{L}\int_0^1 (2\pi-1)[P(V < T/2 \,|\, \pi) + 0.5P(V = T/2 \,|\, \pi) - I(\pi < 1/2)]f_i(\pi)\, d\pi, \tag{2.7}$$

where $f_i(\cdot)$ denotes the posterior density of $\pi^{(i)}$. If we assume a Beta(α_1, α_2) prior, then the posterior $f_i(\cdot)$ is a Beta(c_{1i}, c_{2i}) density with $c_{1i} = \alpha_1 + V_{1i}$ and $c_{2i} = \alpha_2 + V_{2i}$ density, where V_{1i} and V_{2i} are the number of votes for classes 1 and 2, respectively, for case i among the n trees in the current forest. The integral inside the summation in (2.7) can be evaluated explicitly and has the closed form

$$\begin{aligned}
&\int_0^1 (2\pi-1)[P(V < T/2 \,|\, \pi) + 0.5P(V = T/2 \,|\, \pi) - I(\pi < 1/2)]f_i(\pi)\, d\pi \\
&= \sum_{0\le x\le T/2} \left(\frac{1}{2}\right)^{I(x=T/2)} [(a+x) - (b+T-x)] \binom{T}{x} \frac{a^{\overline{x}} b^{\overline{T-x}}}{(a+b)^{\overline{T+1}}} \\
&\quad - \left[2\frac{a}{a+b}P(W^* < 1/2) - P(W < 1/2)\right] \\
&\quad \text{where } a = c_{1i},\ b = c_{2i},\ W \sim \text{Beta}(a,b),\ W^* \sim \text{Beta}(a+1,b)
\end{aligned} \tag{2.8}$$

and $a^{\overline{x}}$ denotes the rising factorial $a^{\overline{x}} = a(a+1)\cdots(a+x-1)$ with $a^{\overline{0}} = 1$. Note that when T is even the term for $x = T/2$ is multiplied by $1/2$.

For our work in Section 3 we use an approximation to (2.7) which can be computed more quickly for larger values of T. This approximation is based on the normal approximations in equations (13) and (12) of Huffer and Park (2020). The approximation to (2.7) may be restated in terms of $\delta = 2\pi - 1$ as

$$ED(\delta, \hat{\delta}) \approx \frac{1}{L}\sum_{i=1}^{L} E_i D(\delta, \hat{\delta}),$$

where E_i denotes the expectation when δ is sampled from the posterior distribution for δ_i, in which case we have

$$\delta \approx N(\hat{\delta}_i, (1-\hat{\delta}_i^2)/t_i) \quad \text{and} \quad \hat{\delta} \,|\, \delta \approx N(\delta, (1-\delta^2)/T)\,.$$

Using this approximation to the joint distribution of $(\delta, \hat{\delta})$, we may calculate $E_i D(\delta, \hat{\delta})$ as

Table 2.1 R code for computing an approximation to H in (2.10)

```
H.approx<-function(a,b,c){
pi<-acos(-1) # pi = 3.1415926...
m <- pmax(0,(a*pi^(1/2)*c-2^(1/2)*b^2)*pi^(1/2)*c/(pi*c^2+2*b^2))
u<-sqrt(1-m^2)*c
v<- -m/u
w<- -u^2*(4*m^2+1)/4
d<- pnorm(v,log.p=TRUE)+dnorm((m-a)/b,log=TRUE)-log(b)
L1<- dnorm(v)/pnorm(v)
L2<- -v*L1-L1^2
L3<- (v^2-1)*L1+3*v*L1^2+2*L1^3
e <- a/b^2-m/b^2-L1/u^3*c^2
f<- -(-1/b^2-(3*L1*m*u-L2)/u^6*c^4)
g<- 12*c^6/u^9*(w*L1+3/4*u*m*L2-L3/12)
(1/6)*exp(-e*m+d)*(sqrt(2*pi/f)*pnorm((f*m+e)/sqrt(f))*
((e^3*f*m+e^4+3*e*f^2*m+6*e^2*f+3*f^2)*g+6*f^4*m+6*e*f^3)*
exp(e*(2*f*m+e)/2/f)+exp(-f*m^2/2)*(e^3*g+5*e*g*f-m*f^2*g+6*f^3))/f^4
}
```

follows:

$$\begin{aligned} E_i D(\delta,\hat{\delta})) &= E_i \delta I(\delta>0,\hat{\delta}<0) - E_i \delta I(\delta<0,\hat{\delta}>0) \\ &= E_i \delta I(\delta>0) P(\hat{\delta}<0 \,|\, \delta) - E_i \delta I(\delta<0) P(\hat{\delta}>0 \,|\, \delta) \\ &= E_i \delta I(\delta>0) \Phi\left(\frac{-\delta\sqrt{T}}{\sqrt{1-\delta^2}}\right) - E_i \delta I(\delta<0) \Phi\left(\frac{\delta\sqrt{T}}{\sqrt{1-\delta^2}}\right) \\ &= H\left(\hat{\delta}_i, \sqrt{\frac{(1-\hat{\delta}_i^2}{t_i}}, \frac{1}{\sqrt{T}}\right) + H\left(-\hat{\delta}_i, \sqrt{\frac{(1-\hat{\delta}_i^2}{t_i}}, \frac{1}{\sqrt{T}}\right), \end{aligned} \tag{2.9}$$

where we define

$$H(a,b,c) = \int_0^\infty \frac{x}{b}\,\phi\left(\frac{x-a}{b}\right)\Phi\left(\frac{-x}{c\sqrt{1-x^2}}\right)\,dx\,. \tag{2.10}$$

The function $H(a,b,c)$ does not have a convenient closed-form expression, but can be approximated using a Laplace-type approximation. This was obtained with the assistance of Maple and is rather complicated, but still quick to compute. It is perhaps most easily described by giving the R code displayed in Table 2.1.

To summarize, our computations of $\widehat{\Delta}_n^{(2,\mathrm{oob})}$ and $\widehat{\Delta}_n^{(2,\mathrm{test})}$ in Section 3 use

$$\widehat{\Delta}_n^{(2,\mathrm{test})} = \frac{1}{L}\sum_{i=1}^{L} E_i D(\delta,\hat{\delta}), \tag{2.11}$$

where $E_i D(\delta,\hat{\delta})$ is given by (2.9) with H approximated using the R code in Table 2.1. $\widehat{\Delta}_n^{(2,\mathrm{oob})}$ is given by a similar formula, summing over the training data and using the OOB votes.

3. Performance of our methods

In this section we illustrate the performance of our statistics and then compare them with the procedure proposed by Hernández-Lobato *et al.* (2013). For these simulations, we use data sets from the UCI repository (Frank and Asuncion, 2010) and the R package mlbench (Leisch and Dimitriadou, 2010).

We first conduct a simulation study to check whether our statistics $\widehat{\Delta}_n^{(2,\mathrm{oob})}$ and $\widehat{\Delta}_n^{(2,\mathrm{test})}$ are close to corresponding $\Delta_n^{(2,oob)}$ and $\Delta_n^{(2,test)}$, respectively. In this simulation, we need data sets with large sample size, so that we choose real data sets like `Banana`, `Magic` and `Phoneme`, and synthetic data sets like `Spirals`, `Twonorm`, `Ringnorm` and `Threenorm`. The difference between out-of-bag and test settings of the simulation is that the training and test data are fixed or not in each repetition of the simulation. In other words, in the out-of-bag setting, each data set is divided into training data and test data for each repetition, but in the test setting, each data is first divided into training and test data and the same training and test data are used for each repetition.

We will explain the out-of-bag simulation in more detail. We use 200 repetitions for each data set. In each repetition: The data is divided into training and test data. The training data is used to construct forests of size 100, 200, 400, and 100000. The OOB votes from the forests of size $n = 100, 200, 400$ are used to compute the estimates $\widehat{\Delta}_n^{(2,\mathrm{oob})}$ for $n = 100, 200, 400$. The test data is classified using the forests of size 100, 200, 400, and 100000, and the classifications results are used to calculate estimates of $\Delta_n^{(2,oob)}$ for $n = 100, 200, 400$. The training and test data are selected as follows: For `Banana`, `Phoneme`, and `Magic` data, 1000 cases (2000 cases for `Magic`) are randomly selected as training data and the remaining cases are used as test data. For the synthetic data `Spirals`, `Twonorm`, `Ringnorm`, and `Threenorm`, 1000 and 10000 cases are randomly sampled from each distribution as training and test data, respectively. For the Spiral data, we set sd=0.15.

We explain the test simulation in detail. Here we use 500 repetitions for each data set. Each data set is divided into training and test data, and the training data is used to construct an 'infinite' forest of size 100000. These are fixed and used in all repetitions. For the real data sets, 1000 cases (2000 cases for `Magic`) are randomly selected as training data and the remaining cases serve as test data. For the synthetic data, 1000 and 10000 cases are randomly sampled from each synthetic distribution as training and test data, respectively. For `Spirals` data, we set sd=0.15. In each repetition: Independent forests of size 100, 200, 400 are constructed from the training data. The test data is classified using the forests of size 100, 200, 400, and 100000, and the classifications results are used to calculate $\widehat{\Delta}_n^{(2,\mathrm{test})}$ and $\Delta_n^{(2,test)}$ for $n = 100, 200, 400$.

The simulation results of out-of-bag and test settings are shown in Table 3.1 and Table 3.2, respectively. From these tables, we can see that $\widehat{\Delta}_n^{(2,\mathrm{oob})}$ and $\widehat{\Delta}_n^{(2,\mathrm{test})}$ are not quite different from $\Delta_n^{(2,oob)}$ and $\Delta_n^{(2,test)}$ respectively, and that the difference between $\widehat{\Delta}_n^{(2,\mathrm{oob})}$ and $\Delta_n^{(2,oob)}$ (or $\widehat{\Delta}_n^{(2,\mathrm{test})}$ and $\Delta_n^{(2,test)}$) is not a decreasing function of n but an decreasing function of n on average.

We next conduct a simulation study to compare our statistics with the procedure by Hernández-Lobato *et al.* (2013). Here we include $\widehat{\Delta}_n^{(1,\mathrm{oob})}$ and $\widehat{\Delta}_n^{(1,\mathrm{test})}$ in Huffer and Park (2020) as well as $\widehat{\Delta}_n^{(2,\mathrm{oob})}$ and $\widehat{\Delta}_n^{(2,\mathrm{test})}$ because they are all suggested by us. For a variety of well-known real and synthetic data sets, we study the sizes of random forests needed to

Table 3.1 Average and standard deviation of $\hat{\Delta}_n^{(2,oob)}$ and $\Delta_n^{(2,oob)}$ with 100, 200, 400 trees; 200 repetitions are used.

Problem	$100\hat{\Delta}_{100}^{(2,oob)}$	$100\Delta_{100}^{(2,oob)}$	$100\hat{\Delta}_{200}^{(2,oob)}$	$100\Delta_{200}^{(2,oob)}$	$100\hat{\Delta}_{400}^{(2,oob)}$	$100\Delta_{400}^{(2,oob)}$
Banana	$.071 \pm .007$	$.074 \pm .018$	$.036 \pm .005$	$.037 \pm .009$	$.018 \pm .003$	$.019 \pm .006$
Magic	$.124 \pm .008$	$.125 \pm .012$	$.061 \pm .005$	$.063 \pm .007$	$.030 \pm .003$	$.030 \pm .004$
Phoneme	$.135 \pm .010$	$.148 \pm .026$	$.069 \pm .006$	$.074 \pm .014$	$.034 \pm .004$	$.036 \pm .008$
Spirals	$.096 \pm .009$	$.102 \pm .019$	$.048 \pm .005$	$.050 \pm .010$	$.024 \pm .003$	$.026 \pm .006$
Twonorm	$.120 \pm .007$	$.098 \pm .010$	$.052 \pm .004$	$.047 \pm .006$	$.024 \pm .003$	$.023 \pm .004$
Ringnorm	$.120 \pm .009$	$.111 \pm .020$	$.055 \pm .005$	$.054 \pm .009$	$.026 \pm .003$	$.027 \pm .005$
Threenorm	$.294 \pm .013$	$.320 \pm .020$	$.151 \pm .008$	$.159 \pm .012$	$.077 \pm .005$	$.079 \pm .006$

Table 3.2 Average and standard deviation of $\hat{\Delta}_n^{(2,test)}$ and $\Delta_n^{(2,test)}$ with $n = 100, 200, 400$ trees; 500 repetitions are used.

Problem	$100\hat{\Delta}_{100}^{(2,test)}$	$100\Delta_{100}^{(2,test)}$	$100\hat{\Delta}_{200}^{(2,test)}$	$100\Delta_{200}^{(2,test)}$	$100\hat{\Delta}_{400}^{(2,test)}$	$100\Delta_{400}^{(2,test)}$
Banana	$.082 \pm .003$	$.086 \pm .018$	$.042 \pm .001$	$.044 \pm .010$	$.021 \pm .001$	$.021 \pm .006$
Magic	$.127 \pm .002$	$.128 \pm .011$	$.063 \pm .001$	$.063 \pm .006$	$.031 \pm .001$	$.032 \pm .003$
Phoneme	$.130 \pm .003$	$.135 \pm .022$	$.065 \pm .002$	$.064 \pm .011$	$.032 \pm .001$	$.030 \pm .006$
Spirals	$.100 \pm .002$	$.100 \pm .015$	$.050 \pm .001$	$.050 \pm .008$	$.025 \pm .001$	$.023 \pm .004$
Twonorm	$.104 \pm .002$	$.096 \pm .010$	$.047 \pm .001$	$.044 \pm .006$	$.022 \pm .001$	$.021 \pm .003$
Ringnorm	$.113 \pm .003$	$.111 \pm .017$	$.056 \pm .001$	$.057 \pm .010$	$.028 \pm .001$	$.029 \pm .006$
Threenorm	$.293 \pm .003$	$.303 \pm .018$	$.149 \pm .001$	$.153 \pm .011$	$.075 \pm .001$	$.077 \pm .007$

achieve specific critical values for our statistics, and the accuracies of the resulting forests. We follow the approach used by Hernández-Lobato *et al.* (2013) in constructing their Table 3, and we display our results in a similar format.

For each real data set, we randomly partition the data 500 times into a training set and a test set using 2/3 and 1/3 of the available data. For each random partition and each statistic, we construct a random forest of the size needed to achieve the chosen critical value, and we evaluate the test set error of this forest. To reduce the variability in our comparisons, we use the same sequence of trees when constructing all the various random forests for a given random partition of the data. For the synthetic classification problems (i.e. `Ringnorm`, `Threenorm`, `Twonorm`) the procedure is similar except that we construct 500 independent realizations of the problem consisting of a training set of 300 cases and a test set of 1000 cases; except for the `Spiral` problem where 5,000 cases are used for training and 10000 for testing.

We construct random forests whose sizes achieve the critical values $\widehat{\Delta}_n^{(1,\text{oob})} < 0.001$, $\widehat{\Delta}_n^{(2,\text{oob})} < 0.0005$, hms-oob > 0.99, $\widehat{\Delta}_n^{(1,\text{test})} < 0.0005$, $\widehat{\Delta}_n^{(2,\text{test})} < 0.0005$, hms-test > 0.99. Here 'hms-oob' and 'hms-test' denote the procedures designated RF-OOB and RF-test in Table 3 of Hernández-Lobato *et al.* (2013) (except that we implement these procedures in a strictly sequential fashion). The critical values used with $\widehat{\Delta}_n^{(1,\text{oob})}, \ldots, \widehat{\Delta}_n^{(2,\text{test})}$ are chosen here to give random forests of a generally similar size to those produced by hms-oob and hms-test. In Tables 3.3 and 3.4, for each data set we summarize the sizes of the 500 random forests by giving the median and (in parentheses) the quartiles, and summarize the classification accuracy by giving the average and standard deviation of the test set errors. The column headings indicate the statistic and the critical value.

We see from Table 3.3 that there is little difference in the classification accuracy achieved

Table 3.3 Median and interquartile interval (between parentheses) of the number of trees for the estimated RF ensembles; 500 repetitions are used.

Problem	$\Delta_n^{(1,oob)}(.001)$	$\Delta_n^{(2,oob)}(.0005)$	hms-oob	$\Delta_n^{(1,test)}(.0005)$	$\Delta_n^{(2,test)}(.0005)$	hms-test
Abalone	350 (330, 372)	253 (243, 263)	409 (355, 467)	253 (227, 280)	254 (239, 273)	398 (324, 488)
Australian	287 (254, 331)	221 (201, 237)	249 (190, 323)	196 (154, 244)	213 (186, 246)	236 (159, 356)
Banana	183 (174, 194)	130 (126, 134)	113 (102, 127)	133 (119, 148)	132 (124, 141)	113 (93, 137)
Breast	69 (58, 82)	53 (48, 59)	21 (18, 25)	42 (30, 58)	49 (41, 60)	16 (11, 23)
German	739 (684, 819)	535 (502, 568)	1604 (1303, 2039)	514 (442, 606)	539 (492, 595)	1666 (1150, 2423)
Heart	495 (413, 590)	376 (339, 420)	610 (411, 891)	331 (251, 432)	389 (321, 465)	619 (362, 1088)
House	61 (47, 76)	62 (51, 71)	23 (18, 31)	26 (17, 39)	45 (34, 58)	15 (10, 25)
Ionosphere	144 (120, 174)	119 (107, 134)	72 (55, 96)	84 (62, 115)	110 (92, 134)	61 (41, 91)
Liver	931 (801 1084)	692 (628, 750)	2140 (1497, 3204)	631 (496, 795)	695 (597, 803)	2154 (1268, 3918)
Magic	275 (267, 282)	198 (195, 201)	258 (241, 276)	199 (191, 209)	199 (194, 204)	253 (231, 279)
Phoneme	279 (265, 292)	197 (192, 204)	263 (233, 291)	203 (187, 219)	201 (190, 212)	261 (222, 209)
Pima	529 (457, 607)	396 (360, 439)	756 (529, 1058)	353 (280, 465)	400 (334, 470)	732 (457, 1278)
Ringnorm	414 (359, 480)	323 (293, 355)	457 (348, 644)	305 (270, 342)	313 (291, 340)	576 (454, 745)
Sonar	902 (744, 1055)	676 (597, 766)	1826 (1167, 3087)	616 (469, 793)	716 (600, 865)	2107 (1038, 4085)
Spam	135, 128, 144)	103 (99, 106)	67 (61, 75)	97 (87, 107)	100 (94, 106)	63 (53, 74)
Spiral	266 (253, 281)	190 (184, 198)	241 (217, 271)	193 (182, 203)	191 (182, 199)	236 (213, 263)
Threenorm	978 (842, 1114)	720 (663, 783)	2544 (1758, 3611)	725 (663, 794)	730 (693, 769)	3161 (2620, 4007)
Tic-tac-toe	241 (216, 269)	206 (196, 217)	187 (163, 224)	149 (122, 179)	180 (159, 202)	151 (111, 203)
Twonorm	330 (282, 389)	263 (242, 290)	307 (232, 419)	235 (213, 262)	249 (233, 266)	342 (271, 418)
Whitewine	458 (438, 480)	328 (318, 336)	693 (624, 781)	337 (309, 363)	334 (318, 352)	705 (597, 844)

Table 3.4 Average and standard deviation of the test errors (in percent) for the estimated RF ensembles; 500 repetitions are used.

Problem	$\Delta_n^{(1,oob)}(.001)$	$\Delta_n^{(2,oob)}(.0005)$	hms-oob	$\Delta_n^{(1,test)}(.0005)$	$\Delta_n^{(2,test)}(.0005)$	hms-test
Abalone	16.72 ± 0.85	16.73 ± 0.84	16.72 ± 0.84	16.72 ± 0.84	16.72 ± 0.83	16.72 ± 0.83
Australian	13.15 ± 1.92	13.14 ± 1.89	13.15 ± 1.92	1316 ± 1.89	1315 ± 1.90	1317 ± 1.90
Banana	10.73 ± 0.59	10.75 ± 0.59	10.75 ± 0.59	10.75 ± 0.59	10.75 ± 0.59	10.76 ± 0.59
Breast	2.91 ± 0.99	2.97 ± 1.02	3.12 ± 1.0	2.94 ± 0.95	2.95 ± 0.98	3.13 ± 101
German	23.84 ± 2.06	23.85 ± 2.06	23.81 ± 2.08	23.84 ± 2.10	23.86 ± 2.08	23.82 ± 2.08
Heart	17.66 ± 3.63	17.72 ± 3.57	17.65 ± 3.59	17.73 ± 3.51	17.68 ± 3.54	17.73 ± 3.58
House	3.45 ± 170	3.46 ± 166	3.58 ± 179	3.47 ± 170	3.46 ± 167	3.50 ± 175
Ionosphere	6.72 ± 2.05	6.77 ± 2.04	6.81 ± 2.07	6.72 ± 2.00	6.72 ± 1.99	6.72 ± 2.07
Liver	27.63 ± 3.68	27.73 ± 3.66	27.69 ± 3.63	27.70 ± 3.66	27.71 ± 3.64	27.63 ± 3.64
Magic	12.08 ± 0.35	12.09 ± 0.34	12.08 ± 0.35	12.10 ± 0.35	12.09 ± 0.35	12.08 ± 0.35
Phoneme	9.68 ± 0.66	9.70 ± 0.66	9.68 ± 0.65	9.69 ± 0.65	9.70 ± 0.66	9.68 ± 0.65
Pima	22.11 ± 3.16	22.09 ± 3.16	22.08 ± 3.15	22.06 ± 3.20	22.11 ± 3.20	22.12 ± 3.14
Ringnorm	6.54 ± 1.30	6.57 ± 1.30	6.53 ± 1.32	6.57 ± 1.30	6.57 ± 1.29	6.48 ± 1.27
Sonar	18.44 ± 5.16	18.51 ± 5.05	18.30 ± 5.13	18.52 ± 5.02	18.52 ± 5.06	18.28 ± 5.21
Spam	4.99 ± 0.52	5.01 ± 0.52	5.04 ± 0.52	5.00 ± 0.51	5.01 ± 0.52	5.05 ± 0.50
Spiral	16.18 ± 0.41	16.20 ± 0.41	16.19 ± 0.41	16.20 ± 0.41	16.20 ± 0.41	16.19 ± 0.41
Threenorm	16.48 ± 142	16.50 ± 141	16.43 ± 141	16.52 ± 140	16.50 ± 140	16.42 ± 141
Tic-tac-toe	2.25 ± 1.01	2.29 ± 1.04	2.33 ± 1.07	2.35 ± 1.02	2.31 ± 1.01	2.33 ± 0.98
Twonorm	3.93 ± 0.68	3.96 ± 0.67	3.95 ± 0.69	3.98 ± 0.66	3.98 ± 0.66	3.91 ± 0.65
Whitewine	16.72 ± 0.85	16.72 ± 0.85	16.72 ± 0.85	16.72 ± 0.85	16.72 ± 0.85	16.72 ± 0.85

using the different statistics. However, in Table 3.4 we see considerable differences in the sizes of the random forests. In particular, the random forests produced by hms-oob and hms-test for the problems `German`, `Liver`, `Sonar`, and `Threenorm` are much larger than those from the statistics we propose, but with little or no improvement in accuracy. For the problems `Heart`, `Pima`, and `Whitewine`, hms-oob and hms-test produce forests that are moderately larger than our proposed statistics, again with little or no improvement in accuracy.

These results show that our proposed statistics, which are based on the less stringent criterion for agreement between the finite and infinite forests given in (2.3), allow us to construct random forests which achieve good classification accuracy without being unnecessarily large.

References

Breiman, L. (2001). Random forests. *Machine Learning*, **45**, 5–32.

Frank, A. and Asuncion, A. (2010). *UCI machine learning repository* [`http://archive.ics.uci.edu/ml`], Irvine, CA: University of California, School of Information and Computer Science.

Hernández-Lobato, D., Martínez-Muñoz, G. and Suárez, A. (2011). Inference on the prediction of ensembles of infinite size. *Pattern Recognition*, **44**, 1426–1434.

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

Huffer, F. W. and Park, C. (2020). A simple rule for monitoring the error rate of random forest for classification. *Quantitative Bio-Science*, **39**, 1-15.

Leisch, F. and Dimitriadou, E. (2010). *mlbench: Machine learning benchmark problems*, R package version 2.1-0.

Park, C. (2016). A simple diagnostic statistic for determining the size of random forest. *Journal of the Korean Data & Information Science Society*, **27**, 855–863.

Park, C. (2017). A measure of discrepancy based on margin of victory useful for the determination of random forest size. *Journal of the Korean Data & Information Science Society*, **28**, 515–524.