

# The automatic construction of bootstrap confidence intervals

Bradley Efron<sup>\*†</sup>

Balasubramanian Narasimhan

## Abstract

The standard intervals, e.g.,  $\hat{\theta} \pm 1.96\hat{\sigma}$  for nominal 95% two-sided coverage, are familiar and easy to use, but can be of dubious accuracy in regular practice. Bootstrap confidence intervals offer an order of magnitude improvement—from first order to second order accuracy. This paper introduces a new set of algorithms that automate the construction of bootstrap intervals, substituting computer power for the need to individually program particular applications. The algorithms are described in terms of the underlying theory that motivates them, along with examples of their application.

*Keywords:* bca method, nonparametric intervals, exponential families, second-order accuracy

## 1 Introduction

Among the most useful, and most used, of statistical constructions are the *standard intervals*

$$\hat{\theta} \pm z^{(\alpha)}\hat{\sigma}, \tag{1.1}$$

giving approximate confidence statements for a parameter  $\theta$  of interest. Here  $\hat{\theta}$  is a point estimate,  $\hat{\sigma}$  an estimate of its standard error, and  $z^{(\alpha)}$  the  $\alpha$ th quantile of a standard normal distribution. If  $\alpha = 0.975$ , for instance, interval (1.1) has approximate 95% coverage.

A prime virtue of the standard intervals is their ease of application. In principle, a single program can be written that automatically produces intervals (1.1), say with  $\hat{\theta}$  a maximum likelihood estimate (MLE) and  $\hat{\sigma}$  its bootstrap standard error. Accuracy, however, can be a concern: if  $X \sim \text{Poisson}(\theta)$  is observed to be 16, then (1.1) gives 95% interval (8.16, 23.84) for  $\theta$ , compared to the exact interval (9.47, 25.41) (obtained via the usual Neyman construction, splitting the atom of probability at 16). Modern theory and computer power now provide an opportunity for the routine use of more accurate intervals, this being the theme of what follows.

Exact intervals for  $\theta$  are almost never available in the presence of nuisance parameters, which is to say, in most applied settings. This accounts for the popularity of the standard intervals. Bootstrap confidence intervals (DiCiccio and Efron, 1996; Efron, 1987) were proposed as general-purpose improvements over (1.1). They are *second-order accurate*, having the error in their claimed coverage probabilities going to zero at rate  $O(1/n)$  in sample size compared to the first-order rate  $O(1/\sqrt{n})$  for standard intervals. This is often a substantial improvement, yielding bootstrap limits (9.42, 25.53) compared to the exact limits (9.47, 25.41) in the Poisson problem.<sup>1</sup>

Can the construction of bootstrap confidence intervals be made fully automatic? The answer is yes for nonparametric intervals: `bcanon` in the R package `bootstrap` carries out the bca (“bias

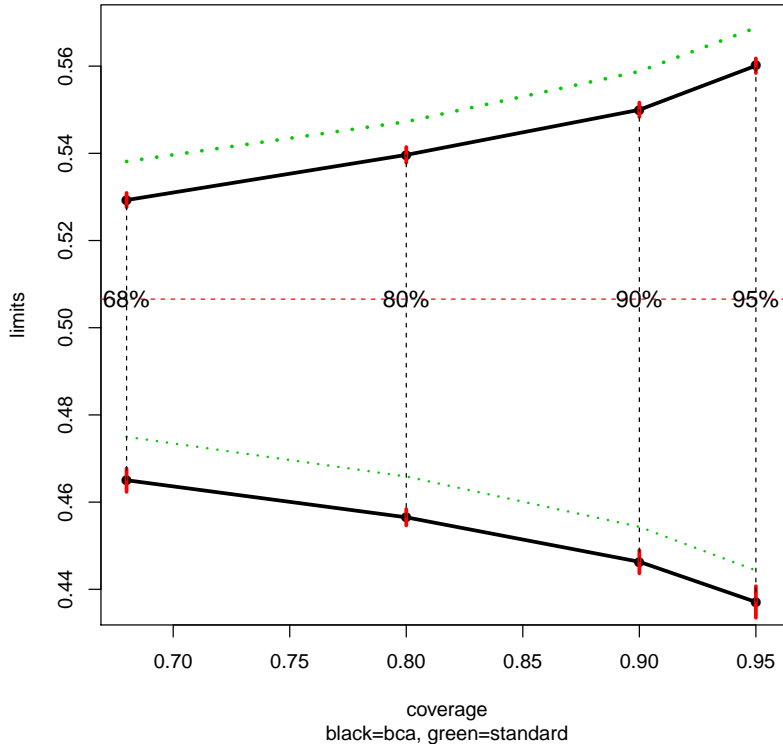
---

<sup>\*</sup>Research supported in part by National Science Foundation award DMS 1608182.

<sup>†</sup>Sequoia Hall, 390 Serra Mall, Stanford University, Stanford, CA 94305-4065; `brad@stat.stanford.edu`

<sup>1</sup>Using the bca algorithm (2.2).

corrected and accelerated”, Efron, 1987) calculations, as does `boot.ci` from the package `boot`. Both require only the relevant data set and the R function for the statistic of interest from the user. The program `bcajack`, Section 3, is an augmented version of `bcanon` that can handle very large data sets, while guiding the choice of the bootstrap sample size  $B$ . A second program, `bcajack2`, is designed to accommodate computationally intensive situations where, for instance, individual bootstrap replications are collected from a distributed network.



**Figure 1:** Two-sided nonparametric confidence limits for adjusted  $R^2$ , diabetes data, plotted vertically versus nominal coverage level. Bootstrap (solid curves); standard intervals (dotted curves). Small red vertical bars indicate Monte Carlo error in the bootstrap curves, from program `bcajack`, Section 3.

Figure 1 concerns an application of `bcajack`. The diabetes data, Table 1 of Efron (2004), comprises  $n = 442$  vectors in  $\mathcal{R}^{11}$ ,

$$v_i = (x_i, y_i), \quad i = 1, 2, \dots, 442; \quad (1.2)$$

$x_i$  is a 10-vector of baseline predictors — age, sex, BMI, etc. — for patient  $i$ , while  $y_i$  is a real-valued measure of disease progression one year later. The data consists of

$$\mathbf{X} \quad \text{and} \quad \mathbf{y}, \quad (1.3)$$

where  $\mathbf{X}$  is the  $n \times 10$  matrix with  $i$ th row  $x_i$ , and  $\mathbf{y}$  the  $n$ -vector of  $y_i$  values. The full data set can be expressed as an  $n \times 11$  matrix  $\mathbf{v}$ , having  $v_i$  as the  $i$ th row.

Suppose we are interested in the adjusted  $R^2$  for ordinary linear regression of  $\mathbf{y}$  on  $\mathbf{X}$ , with point estimate

$$\hat{\theta} = \text{lm}(\mathbf{y} \sim \mathbf{X})\$ \text{adj.r.squared} \quad (1.4)$$

in the R language;  $\hat{\theta} = 0.507$  for the diabetes data. How accurate is this estimate?

The heavy curves in Figure 1 describe the lower and upper limits of nonparametric bootstrap confidence intervals for the true  $\theta$ , with nominal two-sided coverages ranging from 68% at the left of the figure to 95% at the right. At 68%, the bootstrap limits are

$$(0.462, 0.529), \tag{1.5}$$

compared to the standard limits (dotted curves)

$$(0.475, 0.538), \tag{1.6}$$

from (1.1) with  $\alpha = 0.84$ . The bootstrap standard error for  $\hat{\theta}$  was  $\hat{\sigma} = 0.032$ , so the bootstrap limits are shifted downward from the standard intervals by somewhat less than half a standard error. Figure 1 says that  $\hat{\theta}$  is substantially biased upward for the adjusted  $R^2$ , requiring downward corrections for second-order accuracy.

The calculations for Figure 1 came from the nonparametric bootstrap confidence interval program `bcajack` discussed in Section 3. The call was

$$\text{bcajack}(\mathbf{v}, 2000, \text{rfun}), \tag{1.7}$$

with  $\mathbf{v}$  the diabetes data matrix,  $B = 2000$  the number of bootstrap replications, and `rfun` an R program for  $\hat{\theta}$ : for any matrix  $\mathbf{v}^*$  having 11 columns `rfun`( $\mathbf{v}^*$ ) linearly regresses the 11th column on the first 10 and returns the adjusted  $R^2$ .

There is nothing special about adjusted  $R^2$ . For any parameter of interest, say  $\hat{\phi} = \text{tfun}(\mathbf{v})$ , the call `bcajack(v, 2000, tfun)` produces the equivalent of Figure 1. All effort has been transferred from the statistician to the computer. It is in this sense that `bcajack` deserves to be called “automatic”.

Was the number of bootstrap replications,  $B = 2000$ , too many or perhaps too few? `bcajack` also provides a measure of *internal error*, an estimate of standard deviation of the Monte Carlo bootstrap confidence limits due to stopping at  $B$  replications. The vertical red bars in Figure 1 indicate  $\pm$  one such internal standard deviation. In this example,  $B = 2000$  was sufficient and not greatly wasteful. Stopping at  $B = 1000$  would magnify the Monte Carlo error by about  $\sqrt{2}$ , perhaps becoming uncomfortably large at the lower bootstrap confidence limits.

*Parametric* bootstrap confidence intervals are inherently more challenging to automate. All nonparametric problems have the same basic structure, whereas parametric models differ from each other in their choice of sufficient statistics. Working within this limitation, the program `bcapar`, Section 4, minimizes the required input from the statistician: bootstrap replications of the statistic of interest along with their bootstrap sufficient statistics. This is particularly convenient for generalized linear models, where the sufficient vectors are immediately available.

One can ask whether second-order corrections are worthwhile in ordinary situations. The answer depends on the importance attached to the value of the parameter of interest  $\theta$ . As a point of comparison, the usual student- $t$  intervals  $\bar{x} \pm t_n^{(\alpha)} \hat{\sigma}$  make corrections of order only  $O(\hat{\sigma}/n)$  to (1.1), while the bootstrap corrections are  $O(\hat{\sigma}/\sqrt{n})$ . An obvious comment, but a crucial one, is that modern computing power makes  $B = 2000$  bootstrap replications routinely feasible in a great range of real applications. (The calculations for Figure 1 took three seconds.)

Bootstrap methods are not the only approach to more accurate confidence limits. Barndorff-Nielsen (1983), Skovgaard (1985), and several others have developed likelihood-based theories for higher-order inference. The theory is more formulaic than computer-driven. However, Pierce and Bellio (2017), in the spirit of this paper, have developed an R package for likelihood-based confidence statements.

We proceed as follows: Section 2 gives a brief review of bootstrap confidence intervals. Section 3 discusses nonparametric intervals and their implementation in `bcajack`. Parametric bootstrap intervals, and their implementation by the program `bcapar`, are discussed in Section 4. Comments on the programs appear in the Appendix.

## 2 Bootstrap confidence intervals

We present a brief review of the bca theory for bootstrap confidence intervals, which underlies algorithms `bcajack` and `bcapar`. More thorough expositions appear in DiCiccio and Efron (1996) and Chapter 11 of Efron and Hastie (2016).

The standard method level- $\alpha$  endpoint

$$\hat{\theta}_{\text{stan}}(\alpha) = \hat{\theta} \pm z^{(\alpha)} \hat{\sigma} \quad (2.1)$$

depends on two summary statistics,  $\hat{\theta}$  and  $\hat{\sigma}$ . The bca level- $\alpha$  endpoints (Efron, 1987),

$$\hat{\theta}_{\text{bca}}(\alpha) = G^{-1} \Phi \left( z_0 + \frac{z_0 + z^{(\alpha)}}{1 - a(z_0 + z^{(\alpha)})} \right) \quad (2.2)$$

as plotted in Figure 1 ( $\Phi$  the standard normal cdf), require three other quantities:  $G(\cdot)$ , the cdf of the bootstrap distribution of  $\hat{\theta}$ , and two auxiliary constants discussed below: the *bias corrector*  $z_0$  and the *acceleration*  $a$ . In almost all cases,  $G$ ,  $z_0$ , and  $a$  must be obtained from Monte Carlo simulations. Doing the bca calculations in fully automatic fashion is the goal of the programs presented in this paper.

The standard intervals (1.1) take literally the first-order approximation

$$\hat{\theta} \sim \mathcal{N}(\theta, \sigma^2), \quad (2.3)$$

which can be quite inaccurate in practice. A classical tactic is to make a monotone transformation  $m(\theta)$  to a more favorable scale,

$$\hat{\phi} = m(\hat{\theta}) \quad \text{and} \quad \phi = m(\theta), \quad (2.4)$$

compute the standard interval on the  $\phi$  scale, and map the endpoints back to the  $\theta$  scale by the inverse transformation  $m^{-1}(\cdot)$ . In the classical example,  $m(\cdot)$  is Fisher's  $z$ -transformation for the normal correlation coefficient. For the Poisson family  $\hat{\theta} \sim \text{Poisson}(\theta)$ , the square root transformation  $m(\theta) = \theta^{1/2}$  is often suggested.

This argument assumes the existence of a transformation (2.4) to a normal translation family,

$$\hat{\phi} \sim \mathcal{N}(\phi, 1). \quad (2.5)$$

(If the variance of  $\hat{\phi}$  is any constant it can be reduced to 1 by rescaling.) The bca formula (2.2) is motivated by a less restrictive assumption: that there exists a monotone transformation (2.4) such that

$$\hat{\phi} \sim \mathcal{N}(\phi - z_0 \sigma_\phi, \sigma_\phi^2), \quad \sigma_\phi = 1 + a\phi. \quad (2.6)$$

Here the *bias corrector*  $z_0$  allows for a bias in  $\hat{\phi}$  as an estimate of  $\phi$ , while the *acceleration*  $a$  allows for non-constant variance. Lemma 1 of Efron (1987) shows that (2.4)–(2.6) implies that the bca formula (2.2) gives correct confidence interval endpoints—in a strong sense of correctness.

Crucially, *knowledge of the transformation (2.4) is not required*; only its existence is assumed, serving as motivation for the bca intervals. Formula (2.2) is *transformation invariant*: for any monotone transformation (2.4), the bca endpoints transform correctly,

$$\hat{\phi}_{\text{bca}}(\alpha) = m\left(\hat{\theta}_{\text{bca}}(\alpha)\right). \quad (2.7)$$

This isn't true for the standard intervals, making them vulnerable to poor performance if applied on the "wrong" scale.

Under reasonably general conditions, Hall (1988) and others showed that formula (2.2) produces second-order accurate confidence intervals. See also the derivation in Section 8 of DiCiccio and Efron (1996).

Formula (2.2) makes three corrections to (2.1):

1. for non-normality (through the bootstrap cdf  $G$ );
2. for bias (through  $z_0$ );
3. for nonconstant standard error (through  $a$ ).

In various circumstances, all three can have substantial effects.

As a particularly simple parametric example, where all the calculations can be done theoretically rather than by simulation, suppose we observe

$$\hat{\theta} \sim \theta \cdot \text{Gamma}_{10}/10, \quad (2.8)$$

$\text{Gamma}_{10}$  a gamma variable with 10 degrees of freedom, so  $\theta$  is the expectation of  $\hat{\theta}$ .

**Table 1:** Lower and upper confidence limits for  $\theta$  having observed  $\hat{\theta} = 1$  from the model  $\hat{\theta} \sim \theta \cdot \text{Gamma}_{10}/10$ .

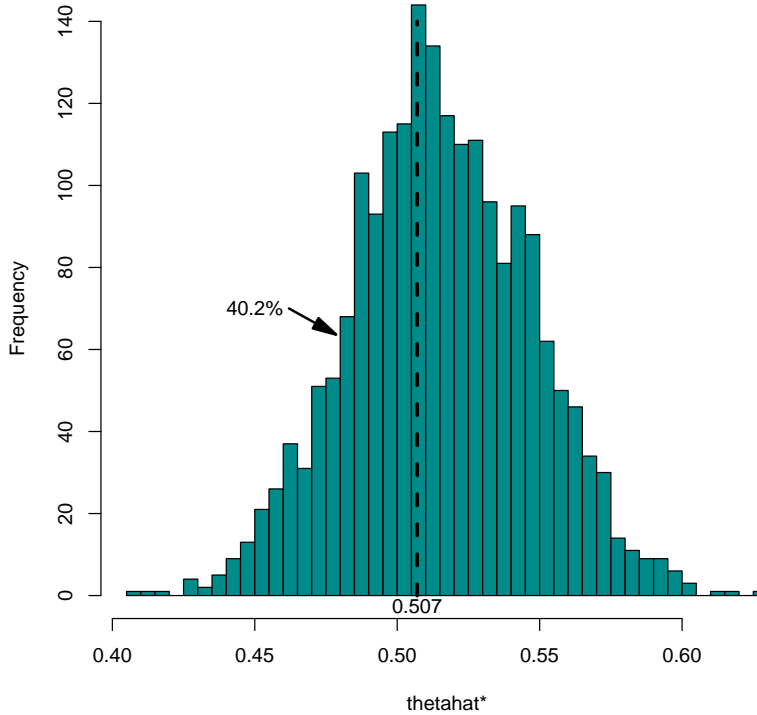
$\alpha$	standard	pctile	bc	bca	exact
.025	.38	.48	.52	.585	.585
.16	.69	.69	.74	.764	.764
.84	1.31	1.31	1.39	1.448	1.448
.975	1.62	1.71	1.80	2.086	2.085

Table 1 shows confidence limits at  $\alpha = 0.025, 0.16, 0.84,$  and  $0.975$ , having observed  $\hat{\theta} = 1$ . (Any other value of  $\hat{\theta}$  simply multiplies the limits.) Five different methods are evaluated: (1) the standard endpoint (2.1); (2) the percentile method  $\hat{\theta}_{\text{pct}}(\alpha) = G^{-1}(\alpha)$ , i.e., taking  $z_0$  and  $a = 0$  in (2.2); (3) the bias-corrected method  $\hat{\theta}_{\text{bc}}(\alpha)$ , setting  $a = 0$  in (2.2); (4) the full bca endpoint (2.2); (5) the exact endpoint. Accuracy of the approximations increases at each step, culminating in near three-digit accuracy for the full bca method.

The three elements of the bca formula —  $G$ ,  $z_0$ , and  $a$  — must usually be estimated by simulation. Figure 2 shows the histogram of 2000 nonparametric bootstrap replications of  $\hat{\theta}$ , the adjusted  $R^2$  statistic for the diabetes data example of Figure 1. Its cdf  $\hat{G}$  estimates  $G$  in (2.2).

We also get an immediate estimate for  $z_0$ ,

$$\hat{z}_0 = \Phi^{-1}\hat{G}\left(\hat{\theta}\right), \quad (2.9)$$



**Figure 2:** 2000 nonparametric bootstrap replications of the adjusted  $R^2$  statistic for the diabetes example of Figure 1. Only 40.2% of the 2000  $\hat{\theta}^*$  values were less than  $\hat{\theta} = 0.507$ , suggesting its strong upward bias.

this following from  $z_0 = \Pr_{\phi}\{\hat{\phi} \leq \phi\}$  in (2.6) (assuming the true value  $\phi = 0$ ). Only 40.2% of the 2000  $\hat{\theta}^*$ 's were less than the original value  $\hat{\theta} = 0.507$ , indicating a substantial upward bias in  $\hat{\theta}$ . The estimated bias corrector

$$\hat{z}_0 = \Phi^{-1}(0.402) = -0.327 \quad (2.10)$$

accounts for most of the downward shift of the bca limits seen in Figure 1.

The acceleration  $a$  cannot in general be read from the bootstrap distribution  $\hat{G}$ , this being an impediment to automatic bootstrap confidence intervals. The next two chapters include discussions of the computation of  $a$  in nonparametric and parametric situations.

### 3 Nonparametric bca intervals

Nonparametric inference begins with an observed data set

$$\mathbf{x} = (x_1, x_2, \dots, x_n), \quad (3.1)$$

where the  $x_i$  are independent and identically distributed (iid) observations from an unknown probability distribution  $F$  on a space  $\mathcal{X}$ ,

$$x_i \stackrel{\text{ind}}{\sim} F, \quad i = 1, 2, \dots, n. \quad (3.2)$$

The space  $\mathcal{X}$  can be anything at all — one-dimensional, multidimensional, functional, discrete — with no parametric form assumed for  $F$ . The assumption that  $F$  is the same for all  $i$  makes (3.2) a *one-sample problem*, the only type we will consider here.

A real-valued statistic  $\hat{\theta}$  has been computed by applying some estimating algorithm  $t(\cdot)$  to  $\mathbf{x}$ ,

$$\hat{\theta} = t(\mathbf{x}); \quad (3.3)$$

we wish to assign a confidence interval to  $\hat{\theta}$ . In the diabetes example of Section 1,  $x_i$  is  $v_i$  (1.2),  $n = 442$ ,  $\mathbf{x}$  is what was called  $\mathbf{v}$ , and  $\hat{\theta} = t(\mathbf{x})$  is the adjusted  $R^2$  statistic (1.3)–(1.4).

A nonparametric bootstrap sample  $\mathbf{x}^*$ ,

$$\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*), \quad (3.4)$$

is composed of  $n$  random draws, *with replacement*, from the set of original observations  $\{x_1, x_2, \dots, x_n\}$ . It produces a bootstrap replication

$$\hat{\theta}^* = t(\mathbf{x}^*) \quad (3.5)$$

of  $\hat{\theta}$ .  $B$  such replications are independently drawn,

$$\hat{\theta}_i^* = t(\mathbf{x}_i^*), \quad i = 1, 2, \dots, B, \quad (3.6)$$

as shown in Figure 2 for the diabetes example,  $B = 2000$ . The bootstrap estimate of standard error is the empirical standard deviation of the  $\hat{\theta}_i^*$  values,

$$\hat{\sigma}_{\text{boot}} = \left[ \sum_{i=1}^B (\hat{\theta}_i^* - \hat{\theta}^*)^2 / (B - 1) \right]^{1/2} \quad (3.7)$$

(where  $\hat{\theta}^* = \sum \hat{\theta}_i^* / B$  equals 0.032 in Figure 2).

The *jackknife* provides another estimate of standard error: let  $\mathbf{x}_{(i)}$  be the data set of size  $n - 1$  having  $x_i$  removed from  $\mathbf{x}$ , and  $\hat{\theta}_{(i)}$  the corresponding estimate

$$\hat{\theta}_{(i)} = t(\mathbf{x}_{(i)}). \quad (3.8)$$

Then

$$\hat{\sigma}_{\text{jack}} = \left[ \frac{n-1}{n} \sum_1^n (\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)})^2 \right]^{1/2}, \quad \left( \hat{\theta}_{(\cdot)} = \sum_1^n \hat{\theta}_{(i)} / n \right), \quad (3.9)$$

which equals 0.033 in Figure 2;  $\hat{\sigma}_{\text{jack}}$  isn't used in the bca algorithm, but jackknife methodology plays an important role in the program `bcajack`.

The vector of bootstrap replications, say

$$\mathbf{t}^* = (\hat{\theta}_1^*, \hat{\theta}_2^*, \dots, \hat{\theta}_B^*), \quad (3.10)$$

provides estimates  $\widehat{G}$  (its cdf) and  $\hat{z}_0$  (2.9). For nonparametric problems, the jackknife differences

$$d_i = \hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)} \quad (3.11)$$

provide an estimate of the acceleration  $a$ ,

$$\hat{a} = \frac{1}{6} \sum_{i=1}^n d_i^3 / \left( \sum_{i=1}^n d_i^2 \right)^{3/2}. \quad (3.12)$$

The call `bcajack(x,B,func)` — where `func` represents  $t(\mathbf{x})$  (3.2) — computes  $\hat{G}$ ,  $\hat{z}_0$ , and  $\hat{a}$ , and the bca limits obtained from (2.2).

Contemporary sample sizes  $n$  can be large, or even *very* large, making the calculation of  $n$  jackknife values impractically slow. Instead, the observations can be collected into  $m$  groups each of size<sup>2</sup>  $g = n/m$ , with  $X_k$  the collection of  $x_i$ 's in group  $k$ . The statistic  $\hat{\theta} = t(\mathbf{x})$  can just as well be thought of as a function of  $\mathbf{X} = (X_1, X_2, \dots, X_m)$ , say

$$\hat{\theta} = T(\mathbf{X}), \tag{3.13}$$

with  $\mathbf{X}$  an iid sample taking values in the product space  $\mathcal{X}^g$ . Now only  $m$ , instead of  $n$ , recomputations are needed to evaluate  $\hat{a}$  from (3.8)–(3.12). (*Note:* `bcajack` does not require separate specification of  $T(\cdot)$ .) For the diabetes example, Figure 1 and Figure 2 were recomputed using  $m = 40$ , with only minor changes from  $m = 442$ .

There are two sources of error in the confidence limits  $\hat{\theta}_{\text{bca}}(\alpha)$ : *sampling error*, due to the random selection of  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  from  $F$ , and *internal error*, due to the Monte Carlo selection of  $B$  bootstrap samples  $\mathbf{x}^*(1), \mathbf{x}^*(2), \dots, \mathbf{x}^*(B)$ . The latter error determines how large  $B$  must be.

**Table 2:** Partial output of nonparametric bootstrap confidence interval program `bcajack` for adjusted  $R^2$  statistic, diabetes data, Figure 1 and Figure 2.

		\$lims					
		bcalims	jacksd	standard	pct		
		.025	.437	.004	.444	.006	
		.05	.446	.003	.454	.013	
		.1	.457	.002	.466	.032	
		.16	.465	.003	.475	.060	
		.5	.498	.001	.507	.293	
		.84	.529	.002	.538	.672	
		.9	.540	.002	.547	.768	
		.95	.550	.002	.559	.862	
		.975	.560	.002	.569	.918	
\$stats						\$ustats	
	$\hat{\theta}$	$\hat{\sigma}_{\text{boot}}$	$\hat{z}_0$	$\hat{a}$	$\hat{\sigma}_{\text{jack}}$	ustat	sdu
est	.507	.032	-.327	-.007	.033	.496	.038
jsd	.000	.001	.028	.000	.000		

A different jackknife calculation provides estimates of internal error, based just on the original  $B$  bootstrap replications. The  $B$ -vector  $\mathbf{t}^*$  (3.10) is randomly partitioned into  $J$  groups of size  $B/J$  each, say  $J = 10$ . Each group is deleted in turn, and  $\hat{\theta}_{\text{bca}}(\alpha)$  recomputed. Finally, the jackknife estimate of standard error (3.9) is calculated from the  $J$  values of  $\hat{\theta}_{\text{bca}}(\alpha)$ . The red vertical bars in Figure 1 indicate  $\pm$  one such jackknife standard error.

Table 2 shows part of the `bcajack` output for the adjusted  $R^2$  statistic, diabetes data, as in Figure 2. The internal standard error estimates for the bca limits  $\hat{\theta}_{\text{bca}}(\alpha)$  are labeled “jacksd”. The column “pct” gives the percentiles of the bootstrap histogram in Figure 2 corresponding to each  $\hat{\theta}_{\text{bca}}(\alpha)$ ; for example,  $\hat{\theta}_{\text{bca}}(0.975)$  corresponded to the 90.2 percentile of the 2000  $\hat{\theta}^*$  values, i.e., the 1804th largest value. Having pct very near 0 or 1 for some  $\alpha$  suggests instability in  $\hat{\theta}_{\text{bca}}(\alpha)$ .

<sup>2</sup>See the Appendix for how `bcajack` proceeds if  $m$  does not exactly divide  $n$ .



Also shown are the estimates  $\hat{\theta}$ ,  $\hat{\sigma}_{\text{boot}}$ ,  $\hat{z}_0$ ,  $\hat{a}$ , and  $\hat{\sigma}_{\text{jack}}$ , as well as their internal standard error estimates “jsd”. ( $\hat{\theta}$ ,  $\hat{a}$ , and  $\hat{\sigma}_{\text{jack}}$  do not involve bootstrap calculations, and so have jsd zero.) Acceleration  $\hat{a}$  is nearly zero, while the bias corrector  $\hat{z}_0$  is large, with a substantial jsd.

The argument  $B$  in `bcajack` can be replaced by  $\mathbf{t}^*$ , the vector of  $B$  bootstrap replications (3.10). Calculating the  $\hat{\theta}^*$  values separately is sometimes advantageous:

- In cases where one wishes to compare `bcajack` output in different settings, for instance, for different choices of the grouping parameter  $m$ .
- If there is interest in each of the components of a  $p$ -dimensional parameter  $\Theta = T(\mathbf{x})$ , a  $B \times p$  matrix  $\mathbf{T}^*$  of bootstrap replications can be calculated from a single run of nonparametric samples, and `bcajack` executed separately for each column of  $\mathbf{T}^*$ .
- The “hybrid method,” discussed in Section 4, allows  $\mathbf{t}^*$  to be drawn from parametric probability models, and then analyzed with `bcajack`.
- For large data sets and complicated statistics  $\hat{\theta}$ , distributed computation of the  $\hat{\theta}^*$  values may be necessary.

This last point brings up a limitation of `bcajack`. Even if the vector of bootstrap replications  $\mathbf{t}^*$  is provided to `bcajack`, the program still needs to execute  $t(\cdot)$  for the jackknife calculations (3.8)–(3.12); an alternative algorithm `bcajack2` entirely avoids step (3.8). It *is* able to provide nonparametric bca intervals from just the replication vector  $\mathbf{t}^*$ . (It also provides a slightly better estimate of internal error: `bcajack` cheats on the calculations of `jacksd` by keeping  $a$  fixed instead of recalculating it fold to fold, while `bcajack2` lets  $a$  vary.) `Bcajack2` is designed for situations where the function  $\hat{\theta} = \text{tfun}(\mathbf{x})$  is awkward to implement within the bca algorithm.

Both `bcajack` and `bcajack2` also return “ustat”, a bias-corrected version of  $\hat{\theta}$ ,

$$\text{ustat} = 2\hat{\theta} - \text{mean}(\hat{\theta}^*), \quad (3.14)$$

so  $\text{ustat} = \hat{\theta} - (\text{mean}(\hat{\theta}^*) - \hat{\theta})$ . Also provided is “sdu”, an estimate of sampling error for `ustat`, based on the theory in Efron (2014). For the adjusted  $R^2$  statistic of Table 2, `ustat` = 0.496 with standard error 0.038, compared to  $\hat{\theta}$  = 0.507 with standard error 0.038, a downward adjustment of about half a standard error.

## 4 Bootstrap confidence intervals for parametric problems

The bca method was originally designed for parametric estimation problems. We assume that the observed data  $\mathbf{y}$  has come from a parametric density function

$$f_{\alpha}(\mathbf{y}), \quad (4.1)$$

where  $\alpha$  is the unknown parameter vector. We wish to set confidence intervals for a one-dimensional parameter of interest

$$\theta = s(\alpha). \quad (4.2)$$

An estimate  $\hat{\alpha} = A(\mathbf{y})$ , perhaps its MLE, provides

$$\hat{\theta} = s(\hat{\alpha}), \quad (4.3)$$

a point estimate of  $\theta$ . By sampling from  $f_{\hat{\alpha}}(\cdot)$  we obtain parametric bootstrap resamples  $\mathbf{y}^*$ ,

$$\mathbf{y}^* \sim f_{\hat{\alpha}}(\cdot). \quad (4.4)$$

Each  $\mathbf{y}^*$  provides an  $\hat{\alpha}^* = A(\mathbf{y}^*)$  and then a bootstrap replication of  $\hat{\theta}$ ,

$$\hat{\theta}^* = s(\hat{\alpha}^*). \quad (4.5)$$

Carrying out (4.4)–(4.5) some large number  $B$  of times yields a bootstrap data vector  $\mathbf{t}^* = (\hat{\theta}_1^*, \hat{\theta}_2^*, \dots, \hat{\theta}_B^*)$ . Bca confidence limits (2.2) can be estimated beginning from  $\mathbf{t}^*$ , but we will see that there is one complication not present in nonparametric settings.

The parametric bootstrap confidence interval program `bcapar` assumes that  $f_{\alpha}(\mathbf{y})$  (4.1) is a member of a  $p$ -parameter exponential family

$$f_{\alpha}(\mathbf{y}) = e^{\alpha' \hat{\beta} - \psi(\alpha)} f_0(\mathbf{y}), \quad (4.6)$$

having  $\alpha$  as the  $p$ -dimensional *natural parameter* and

$$\hat{\beta} = \hat{\beta}(\mathbf{y}) \quad (4.7)$$

as the  $p$ -dimensional sufficient statistic. Its expected value, the *expectation parameter*

$$\beta = E_{\alpha} \left\{ \hat{\beta}(\mathbf{y}) \right\} \quad (4.8)$$

is a one-to-one function of  $\alpha$ ;  $\hat{\beta}$  and its bootstrap replications  $\hat{\beta}^*$  play a central role in `bcapar`.

The parameter of interest  $\theta = s(\alpha)$  can also be expressed as a function of  $\beta$ , say  $\theta = \tau(\beta)$ , and then  $\hat{\theta}$  written as a function of  $\mathbf{y}$ ,

$$\hat{\theta} = \tau \left( \hat{\beta}(\mathbf{y}) \right) \equiv t(\mathbf{y}). \quad (4.9)$$

In familiar applications, either  $s(\hat{\alpha})$  or  $\tau(\hat{\beta})$  may have simple expressions, or sometimes neither. As mentioned in what follows, a convenient feature of `bcapar` is that it works directly from  $\hat{\theta} = t(\mathbf{y})$ , without requiring specification of  $s(\cdot)$  or  $\tau(\cdot)$ .

As an important example of exponential family applications, consider the logistic regression model

$$\boldsymbol{\eta} = \mathbf{M}\boldsymbol{\alpha}, \quad (4.10)$$

where  $\mathbf{M}$  is a specified  $n \times p$  structure matrix, and  $\boldsymbol{\eta}$  is an  $n$ -vector of logit parameters  $\eta_i = \log \pi_i / (1 - \pi_i)$ . We observe

$$y_i = \begin{cases} 1 & \text{with probability } \pi_i \\ 0 & 1 - \pi_i \end{cases} \quad (4.11)$$

independently for  $i = 1, 2, \dots, n$ . This is a  $p$ -parameter exponential family having natural parameter  $\alpha$ , sufficient statistic

$$\hat{\beta} = \mathbf{M}'\mathbf{y}, \quad (4.12)$$

and expectation parameter  $\beta = \mathbf{M}'\boldsymbol{\pi}$ . In the *neonate* application below,  $\theta = s(\alpha)$  is the first coordinate of  $\alpha$ .

The vector of bootstrap replicates  $\mathbf{t}^* = (\hat{\theta}_1^*, \hat{\theta}_2^*, \dots, \hat{\theta}_B^*)$  provides an estimated bootstrap cdf  $\hat{G}$  for substitution into (2.2), the formula for  $\hat{\theta}_{\text{bca}}(\alpha)$ , and also the estimated bias-corrector  $\hat{z}_0 = \Phi^{-1}\hat{G}(\hat{\theta})$  as at (2.9). This leaves the acceleration  $a$ . In parametric problems  $a$  depends on the choice

of sufficient statistic  $\hat{\beta} = \hat{\beta}(\mathbf{y})$  (4.7); `bcapar` takes as input  $t_0 = t(\mathbf{y}) = \hat{\theta}$ , the  $B$ -vector  $\mathbf{t}^*$  of bootstrap replicates, and also  $\mathbf{b}^*$ , the  $B \times p$  matrix of bootstrap sufficient vectors

$$\mathbf{b}^* = (\hat{\beta}_1^*, \hat{\beta}_2^*, \dots, \hat{\beta}_B^*)'. \quad (4.13)$$

In the logistic regression setting (4.10)–(4.12), the estimate  $\hat{\alpha}$  gives  $\hat{\boldsymbol{\eta}} = \mathbf{M}\hat{\alpha}$  and the vector of estimated probabilities

$$\hat{\pi}_i = 1/(1 + e^{-\hat{\eta}_i}), \quad i = 1, 2, \dots, n. \quad (4.14)$$

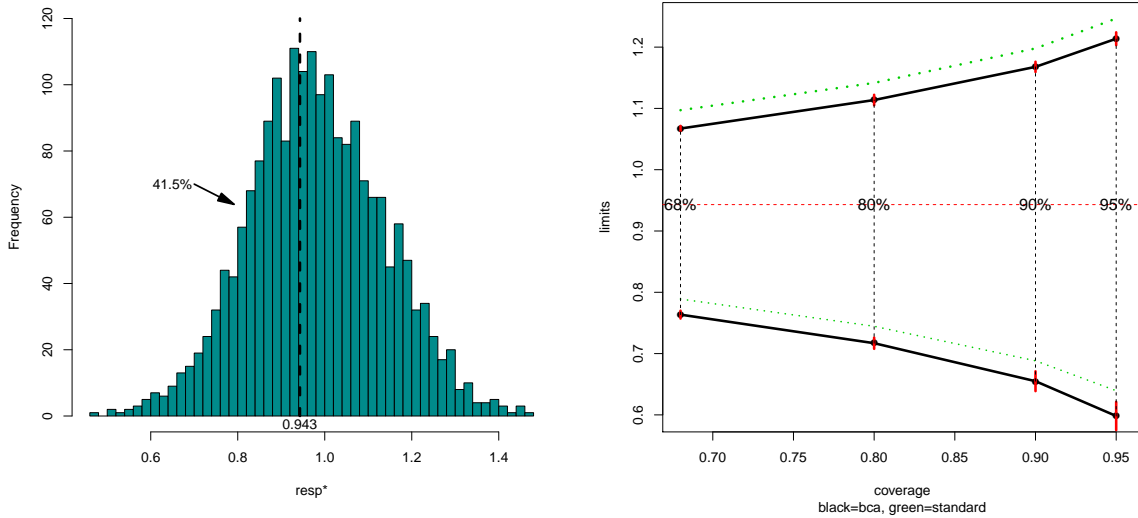
A parametric bootstrap sample  $\mathbf{y}^* = (y_1^*, y_2^*, \dots, y_n^*)$  is generated from

$$y_i^* = \begin{cases} 1 & \text{with probability } \hat{\pi}_i \\ 0 & 1 - \hat{\pi}_i \end{cases} \quad (4.15)$$

independently for  $i = 1, 2, \dots, n$ . Each  $\mathbf{y}^*$  provides a  $\hat{\theta}^* = t(\mathbf{y}^*)$  and also

$$\hat{\beta}^* = \mathbf{M}'\mathbf{y}^*, \quad (4.16)$$

the required inputs for `bcapar`.



**Figure 3:** Neonate data. *Left panel:* 2000 parametric bootstrap replications  $\widehat{\text{resp}}^*$ , MLE. *Right panel:* bca limits (solid) compared to standard limits (dotted), from `bcapar`. Estimates  $\hat{z}_0 = -0.215$ ,  $\hat{\alpha} = -0.019$ .

Figure 3 refers to a logistic regression application: 812 neonates at a large clinic in a developing country were judged to be at serious risk; 205 died within 10 days, while 607 recovered. It was desired to predict death versus survival on the basis of 11 baseline variables, of which “resp”, a measure of respiratory distress, was of particular concern; resp was the first coordinate of  $\alpha$  so  $\hat{\theta} = s(\hat{\alpha}) = \hat{\alpha}_1$ .

A logistic regression analysis having  $\mathbf{M}$   $812 \times 11$  (with the columns of  $\mathbf{M}$  standardized), yielded MLE  $t_0 = \hat{\alpha}_1$  and bootstrap standard error for resp,

$$\widehat{\text{resp}} = 0.943 \pm 0.155. \quad (4.17)$$

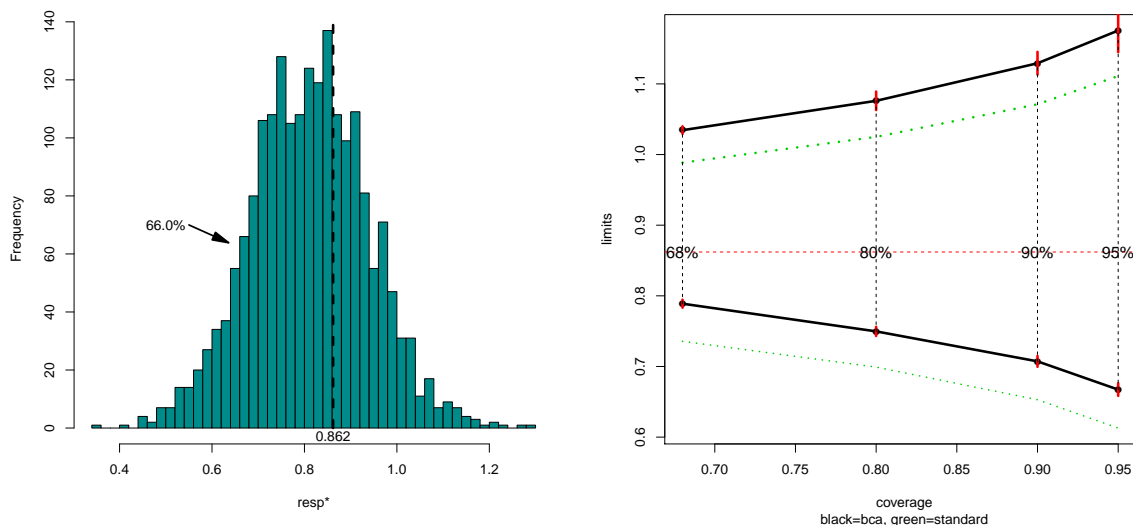
Program `bcapar`, using  $B = 2000$  bootstrap samples, provided the results pictured in Figure 3. The standard confidence limits for resp look reasonably accurate in this case, having only a small

upward bias, about one-fifth of a standard error, compared to the bca limits, with almost all of the difference coming from the bias-corrector  $\hat{z}_0 = -0.215$  (with internal standard error 0.024).

Bootstrap methods are particularly useful for nonstandard estimation procedures. The neonate data was reanalyzed employing model (4.10)–(4.11) as before, but now estimating  $\alpha$  by the *glmnet* regression algorithm, a regularizing procedure that shrinks the components of  $\hat{\alpha}$ , some of them all the way to zero. The R program `cv.glmnet` fits an increasing sequence of  $\alpha$  estimates, and selects a “best” one by cross-validation. Applied to the neonate data, it gave best  $\hat{\alpha}$  having

$$\widehat{\text{resp}} = 0.862 \pm 0.127. \quad (4.18)$$

Regularization reduced the standard error, compared with the MLE (4.15), but at the expense of a possible downward bias.



**Figure 4:** As in Figure 3, but now using *glmnet* estimation for *resp*, rather than logistic regression MLE.

An application of `bcapar` made the downward bias evident. The left panel of Figure 4 shows that 66% of the  $B = 2000$  parametric bootstrap replicates  $\widehat{\text{resp}}^*$  were less than  $\widehat{\text{resp}} = 0.862$ , making the bias corrector large:  $\hat{z}_0 = 0.411$ . The right panel plots the bca intervals, now shifted substantially upward from the standard intervals. Internal errors, the red bars, are considerable for the upper limits. (Increasing  $B$  to 4000 as a check gave almost the same limits.)

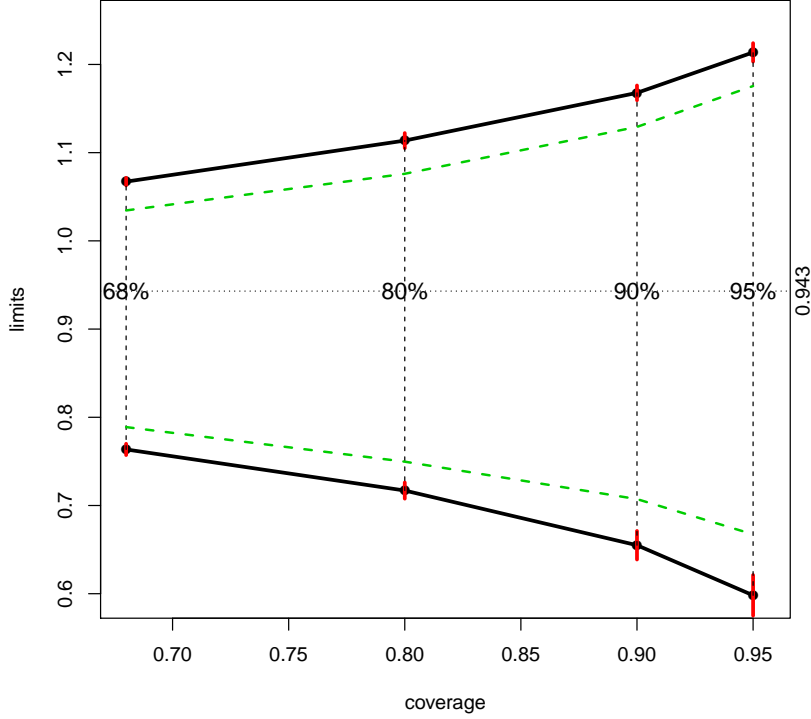
The two sets of bca confidence limits, logistic regression and *glmnet*, are compared in Figure 5. The centering of their two-sided intervals is about the same, *glmnet*’s large bias correction having compensated for its shrunken  $\widehat{\text{resp}}$  estimate (4.17), while its intervals are about 10% shorter.

A convenient feature of `bcapar` that adds to the goal of automatic application is that both  $\hat{\theta}^*$  and  $\hat{\beta}^*$  can be directly evaluated as functions of  $\mathbf{y}^*$ , without requiring a specific function from  $\hat{\beta}^*$  to  $\hat{\theta}^*$  (which would be awkward in a logistic regression setting, for example). If, however, a specific function *is* available, say

$$\hat{\theta}^* = \tau(\hat{\beta}^*), \quad (4.19)$$

then `bcapar` provides supplementary confidence limits based on the *abc* method (“approximate bootstrap confidence” intervals, as distinct from the more recent “Approximate Bayesian Calculation” algorithm).

The *abc* method (DiCiccio and Efron, 1992) substitutes local Taylor series approximations for bootstrap simulations in the calculation of bca-like intervals. It is very fast and, if  $s(\cdot)$  is smoothly



**Figure 5:** Comparison of bca confidence limits for neonate coefficient resp, logistic regression MLE (solid) and glmnet (dashed).

defined, usually quite accurate. In a rough sense, it resembles taking  $B = \infty$  bootstrap replications. It does, however, require function (4.19) as well as other exponential family specifications for each application, and is decidedly not automatic.

Program `abcpar`, discussed in the Appendix, uses the `bcapar` inputs  $\mathbf{t}^*$  and  $\mathbf{b}^*$  (4.13) to automate abc calculations. If function  $\tau(\cdot)$  is supplied to `bcapar`, it also returns the abc confidence limits as well as abc estimates of  $z_0$  and  $a$ . This is illustrated in the next example.

We suppose that two independent variance estimates have been observed in a normal theory setting,

$$\begin{aligned}\hat{\sigma}_1^2 &\sim \sigma_1^2 \chi_{n_1}^2 / n_1, \\ \hat{\sigma}_2^2 &\sim \sigma_2^2 \chi_{n_2}^2 / n_2,\end{aligned}\tag{4.20}$$

and that the parameter of interest is their ratio

$$\theta = \sigma_1^2 / \sigma_2^2.\tag{4.21}$$

In this case the MLE  $\hat{\theta} = \hat{\sigma}_1^2 / \hat{\sigma}_2^2$  has a scaled  $F$  distribution,

$$\hat{\theta} \sim \theta F_{n_1, n_2}.\tag{4.22}$$

Model (4.20) is a two-parameter exponential family having sufficient statistic

$$\hat{\beta} = (\hat{\sigma}_1^2, \hat{\sigma}_2^2).\tag{4.23}$$

Because (4.22) is a one-parameter scale family, we can compute *exact* level  $\alpha$  confidence limits for  $\theta$ ,

$$\hat{\theta}_{\text{exact}}(\alpha) = \hat{\theta} / F_{n_1, n_2}^{(1-\alpha)},\tag{4.24}$$

the notation indicating the  $1 - \alpha$  quantile of a  $F_{n_1, n_2}$  random variable.

`Bcapar` was applied to model (4.20)–(4.23), with

$$n_1 = 10 \quad \text{and} \quad n_2 = 42. \quad (4.25)$$

That is, parametric resamples were obtained according to

$$\begin{aligned} \hat{\sigma}_1^{2*} &\sim \hat{\sigma}_1^2 \chi_{n_1}^2 / n_1, \\ \hat{\sigma}_2^{2*} &\sim \hat{\sigma}_2^2 \chi_{n_2}^2 / n_2, \end{aligned} \quad (4.26)$$

giving

$$\hat{\theta}^* = \hat{\sigma}_1^{2*} / \hat{\sigma}_2^{2*} \quad \text{and} \quad \hat{\beta}^* = (\hat{\sigma}_1^{2*}, \hat{\sigma}_2^{2*}). \quad (4.27)$$

The observed value  $\hat{\theta}$  simply scales the exact endpoints  $\hat{\theta}_{\text{exact}}(\alpha)$  and likewise  $\hat{\theta}_{\text{bca}}(\alpha)$  — correct behavior under transformations is a key property of the bca algorithm — so for comparison purposes the value of  $\hat{\theta}$  is irrelevant. It is taken to be  $\hat{\theta} = 1$  in what follows.

**Table 3:** `Bcapar` output for ratio of normal theory variance estimates of  $\theta = \sigma_1^2 / \sigma_2^2$  (4.20)–(4.21); bca limits are a close match to exact limits (4.24) given observed point estimate  $\hat{\theta} = 1$ . Including `func=funcF` (4.28) in the call added abc limits and statistics.

\$lims						
	bcalims	jacksd	pctiles	stand	abclims	exact.lims
.025	.416	.007	.067	-.054	.396	.422
.05	.477	.006	.103	.116	.465	.484
.1	.562	.005	.164	.311	.557	.570
.16	.639	.008	.230	.465	.642	.650
.5	1.051	.007	.574	1.000	1.052	1.053
.84	1.776	.021	.903	1.535	1.790	1.800
.9	2.093	.033	.952	1.689	2.118	2.128
.95	2.572	.053	.985	1.884	2.659	2.655
.975	3.095	.091	.996	2.054	3.302	3.247

\$stats						ustats				\$abcstats	
	$\theta$	$\hat{\sigma}_{\text{boot}}$	$a$	$z_0$	sd.delta		ustat	sdu	$B$	$a$	$z_0$
est	1	.542	.099	.114	.513	est	.948	.504	16000	.1	.1
jsd	0	.004	.004	.010	.004	jsd	.004	.005	0		

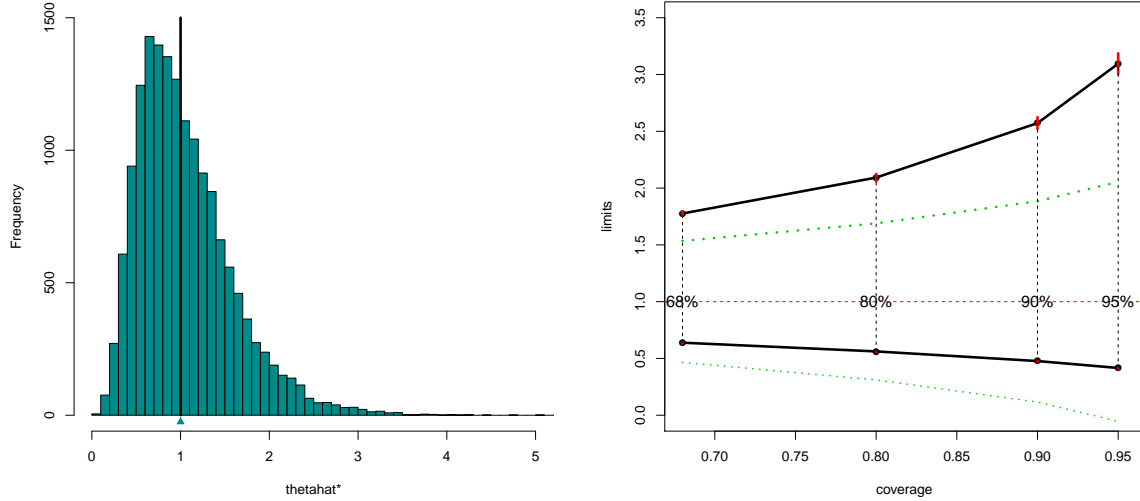
Table 3 shows the output of `bcapar` based on  $B = 16,000$  replications of (4.27).  $B$  was chosen much larger than necessary in order to pinpoint the performance of `bcapar` vis-à-vis  $\hat{\theta}_{\text{exact}}(\alpha)$ . The match was excellent.

Abc limits were also generated by including `func=funcF` in the call,

$$\text{funcF}(\hat{\beta}^*) = \hat{\beta}_1^* / \hat{\beta}_2^*, \quad (4.28)$$

this being  $\tau(\hat{\beta}^*)$  in (4.19). The abc limits were less accurate for the more extreme values of  $\alpha$ . On the other hand, there are theoretical reasons for preferring the abc estimates of  $a$  and  $z_0$  in this case.

The standard, bca, and exact limits are graphed in the right panel of Figure 6. Two results stand out: `bcapar` has done an excellent job of matching the exact limits, and both methods suggest



**Figure 6:** *Left panel:*  $B = 16000$  bootstrap replications of  $\hat{\theta}^*$  (4.27). *Right panel:* confidence limits for  $\theta$  having observed  $\hat{\theta} = 1$ ; green standard, black bca, red dashed exact. The corrections to the standard limits are enormous in this case.

very large upward corrections to the standard limits. In this case the three bca corrections all point in the same direction:  $\hat{z}_0$  is positive,  $\hat{a}$  is positive, and  $\hat{G}$ , the bootstrap cdf, is long-tailed to the right, as seen in the left panel of Figure 6. (This last point relates to a question raised in the early bootstrap literature, as in Hall (1988): if  $\hat{G}$  is long-tailed to the *right* shouldn't the confidence limits be skewed to the *left*? The answer is no, at least in exponential family contexts.)

Even with  $B = 16,000$  replications there is still a moderate amount of internal error in  $\hat{\theta}_{\text{bca}}(0.975)$ , as indicated by its red bar. The “pctiles” column of Table 3 suggests why:  $\hat{\theta}_{\text{bca}}(0.975)$  occurs at the 0.996 quantile of the  $B$  replications, i.e., at the 64th largest  $\hat{\theta}^*$ , where there is a limited amount of data for estimating  $G(\cdot)$ . Bca endpoints are limited to the range of the observed  $\hat{\theta}^*$  values, and can't be trusted when recipe (2.2) calls for extreme percentiles.

A necessary but “un-automatic” aspect of `bcapar` is the need to calculate the matrix of sufficient vectors  $\mathbf{b}^*$  (4.13). This can be avoided in one-sample situations by a *hybrid* application of `bcapar`–`bcajack`, at the risk of possible bias: the vector  $\mathbf{t}^*$  of  $B$   $\hat{\theta}^*$  replications is drawn parametrically, as in the left panel of Figure 6. Then the call

$$\text{bcajack}(t_0 = \hat{\theta}, B = \mathbf{t}^*, \text{func}) \quad (4.29)$$

uses  $\mathbf{t}^*$  to compute  $\hat{G}$  and  $\hat{z}_0$ , as in `bcapar`. In the neonate example, `func` would take a 12-column matrix of predictors and responses, do a logistic regression or `glmnet` analysis of the last column on the first 11, and return the first coordinate of the estimated regression vector.

The bca analysis from (4.29) would differ only in the choice of acceleration  $a$  from a full use of `bcapar`. This made no difference in the neonate example,  $a$  being nearly 0 in both analyses.

## A Appendix

What follows are some additional comments on the new bca programs. The programs are available in the package `bcaboot` from CRAN.

## bcapar

As presented in Efron (1987), the acceleration estimate  $\hat{a}$  depends on the gradient vector of  $\theta = \tau(\beta)$  evaluated at  $\beta = \hat{\beta}$ . If the function  $\tau(\cdot)$  were available, the gradient could be obtained by direct numerical differentiation, say

$$\dot{\mathbf{t}} \equiv \left( \dots \frac{\partial \tau(\beta)}{\partial \beta_j} \dots \right)'_{\beta=\hat{\beta}}. \quad (\text{A.1})$$

However, *not* requiring  $\tau(\cdot)$  is an important practical feature of **bcapar**.

In place of (A.1), **bcapar** uses local linear regression to estimate  $\dot{\mathbf{t}}$ :

- The  $B \times p$  matrix  $\mathbf{b}^*$  (4.13) is standardized to, say,  $\mathbf{c}^*$ , having columns with mean 0 and variance 1.
- The rows  $c_i^*$  of  $\mathbf{c}^*$  are ranked in terms of their length.
- The index set  $\mathbf{I}^*$  of the lowest “Pct” quantile of lengths is identified; Pct = 0.333 by default.
- Finally,  $\dot{\mathbf{t}}$  is set equal to the vector of ordinary linear regression coefficients of  $t_i^* = \hat{\theta}_i^*$  on  $c_i^*$ , within the set  $\mathbf{I}^*$ .

Theoretically, this estimate will approach (A.1) as Pct  $\rightarrow$  0. Empirically the calculation is not very sensitive to the choice of Pct.

Highly biased situations, indicated by very large values of the bias corrector  $\hat{z}_0$ , destabilize the bca confidence limits. (Maximum likelihood estimation in high-dimensional models is susceptible to large biases.) The bias-corrected estimate `ustat` (3.14) can still be useful in such cases. **Bcapar**, like **bcajack**, returns `ustat` and its sampling error estimate “`sdu`”, and also estimates “`jsd`”, their internal errors. These errors are seen to be quite small in the example of Table 3.

**Bcapar** includes an option for also returning the bca *confidence density*, a form of posterior inference for  $\theta$  given the data based on ideas of both Fisher and Neyman; see Section 11.6 of Efron and Hastie (2016) and, for a broader picture, Xie and Singh (2013).

## bcajack and bcajack2

In the nonparametric setting,  $\dot{\mathbf{t}}$  can be closely approximated in terms of the jackknife differences  $d_i = \hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)}$  (3.11) by taking the  $i$ th component to be

$$\dot{t}_i = d_i \sqrt{n \cdot (n-1)} \quad (\text{A.2})$$

(Efron and Hastie, 2016, formula (10.12)).

As discussed near the end of Section 3, using (A.2) in the calculation of the acceleration  $a$  makes **bcajack** require inputting the function for  $t(\mathbf{x})$  (3.8), even if the resamples  $\mathbf{t}^* = (\hat{\theta}_1^*, \hat{\theta}_2^*, \dots, \hat{\theta}_B^*)$  have been previously calculated. **Bcajack2** instead uses a regression estimate for  $\dot{\mathbf{t}}$  and  $\hat{a}$ , analogous to that in **bcapar**.

Nonparametric resamples  $\hat{\theta}^* = t(\mathbf{x}^*)$  (3.5) can also be expressed as

$$\hat{\theta}^* = T(Y), \quad (\text{A.3})$$

$Y$  the *count vector*  $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)$ , where, if  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ ,

$$Y_i = \#\{x_j^* = x_i\}. \quad (\text{A.4})$$



The  $B \times n$  matrix  $\hat{\mathbf{Y}}$  of count vectors for the  $B$  bootstrap samples plays the same role as  $\mathbf{b}^*$  (4.13) in parametric applications. `bcajack2` regresses the  $t_i^*$  values on  $\mathbf{Y}_i$  for a subset of the  $B$  cases having  $\mathbf{Y}_i$  near  $(1, 1, \dots, 1)$ , obtaining an estimate of  $\mathbf{t}$  as in `bcapar`. A previously calculated list of the vector  $\mathbf{t}^*$  and the corresponding matrix  $\hat{\mathbf{Y}}$  can be entered into `bcajack2`, without requiring further specification of  $t(\mathbf{x})$ .

### abcpar

The original abc algorithm (DiCiccio and Efron, 1992) required, in addition to the function  $\theta = \tau(\beta)$  (4.19), a function mapping the expectation parameter  $\beta$  (4.8) back to the natural parameter  $\alpha$  (4.6), say

$$\alpha = \mu(\beta). \quad (\text{A.5})$$

If  $\tau(\cdot)$  is provided to `bcapar`, as at (4.28), it calculates an approximate formula  $\alpha = \hat{\mu}(\beta)$  using the “empirical exponential family” (Efron, 2014, Sect. 6) and then returns abc estimates, as in Table 3.

The abc method is *local*, in the sense of only using resamples  $\hat{\beta}^*$  nearby to  $\hat{\beta}$ , which can be an advantage if  $\hat{\beta}$  falls into an unstable neighborhood of  $\hat{\theta} = \tau(\hat{\beta})$ . In any case, its results depend less on the resample size  $B$ , and provide a check on `bcapar`.

## References

- Barndorff-Nielsen, O. (1983). On a formula for the distribution of the maximum likelihood estimator. *Biometrika* 70: 343–365.
- DiCiccio, T. and Efron, B. (1992). More accurate confidence intervals in exponential families. *Biometrika* 79: 231–245.
- DiCiccio, T. and Efron, B. (1996). Bootstrap confidence intervals. *Statist. Sci.* 11: 189–228.
- Efron, B. (1987). Better bootstrap confidence intervals. *J. Amer. Statist. Assoc.* 82: 171–200, with comments and a rejoinder by the author.
- Efron, B. (2004). The estimation of prediction error: Covariance penalties and cross-validation. *J. Amer. Statist. Assoc.* 99: 619–642, with comments and a rejoinder by the author.
- Efron, B. (2014). Estimation and accuracy after model selection. *J. Amer. Statist. Assoc.* 109: 991–1007.
- Efron, B. and Hastie, T. (2016). *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*. Cambridge: Cambridge University Press, Institute of Mathematical Statistics Monographs (Book 5).
- Hall, P. (1988). Theoretical comparison of bootstrap confidence intervals. *Ann. Statist.* 16: 927–985, with a discussion and a reply by the author.
- Pierce, D. A. and Bellio, R. (2017). Modern likelihood-frequentist inference. *Int. Stat. Rev.* 85: 519–541.
- Skovgaard, I. M. (1985). Large deviation approximations for maximum likelihood estimators. *Probab. Math. Statist.* 6: 89–107.
- Xie, M. and Singh, K. (2013). Confidence distribution, the frequentist distribution estimator of a parameter: A review. *Int. Stat. Rev.* 81: 3–39, with discussion.