



J. R. Statist. Soc. B (2014)
76, Part 4, pp. 795–816

A scalable bootstrap for massive data

Ariel Kleiner, Ameet Talwalkar, Purnamrita Sarkar and Michael I. Jordan

University of California, Berkeley, USA

[Received June 2012. Revised July 2013]

Summary. The bootstrap provides a simple and powerful means of assessing the quality of estimators. However, in settings involving large data sets—which are increasingly prevalent—the calculation of bootstrap-based quantities can be prohibitively demanding computationally. Although variants such as subsampling and the m out of n bootstrap can be used in principle to reduce the cost of bootstrap computations, these methods are generally not robust to specification of tuning parameters (such as the number of subsampled data points), and they often require knowledge of the estimator's convergence rate, in contrast with the bootstrap. As an alternative, we introduce the ‘bag of little bootstraps’ (BLB), which is a new procedure which incorporates features of both the bootstrap and subsampling to yield a robust, computationally efficient means of assessing the quality of estimators. The BLB is well suited to modern parallel and distributed computing architectures and furthermore retains the generic applicability and statistical efficiency of the bootstrap. We demonstrate the BLB's favourable statistical performance via a theoretical analysis elucidating the procedure's properties, as well as a simulation study comparing the BLB with the bootstrap, the m out of n bootstrap and subsampling. In addition, we present results from a large-scale distributed implementation of the BLB demonstrating its computational superiority on massive data, a method for adaptively selecting the BLB's tuning parameters, an empirical study applying the BLB to several real data sets and an extension of the BLB to time series data.

Keywords: Bootstrap; Computational efficiency; Estimator quality assessment; Massive data; Resampling

1. Introduction

The development of the bootstrap and related resampling-based methods in the 1960s and 1970s heralded an era in statistics in which inference and computation became increasingly intertwined (Efron, 1979; Diaconis and Efron, 1983). By exploiting the basic abilities of the von Neumann computer to simulate and iterate, the bootstrap made it possible to use computers not only to compute estimates but also to assess the quality of estimators, yielding results that are generally consistent (Bickel and Freedman, 1981; Giné and Zinn, 1990; Putter and van Zwet, 1996; van der Vaart and Wellner, 1996) and often more accurate than those based on asymptotic approximations (Hall, 1992). Moreover, the bootstrap aligned statistics to computing technology, such that advances in speed and storage capacity of computers could immediately allow statistical methods to scale to larger data sets.

Two recent trends are worthy of attention in this regard. First, the growth in size of data sets is accelerating, with ‘massive’ data sets becoming increasingly prevalent. Second, computational resources are shifting towards parallel and distributed architectures, with multicore and cloud computing platforms providing access to hundreds or thousands of processors. The second

Address for correspondence: Ariel Kleiner, Department of Electrical Engineering and Computer Science, 387 Soda Hall, University of California at Berkeley, Berkeley, CA 94720-1776, USA.
E-mail: akleiner@eecs.berkeley.edu

trend is seen as a mitigating factor with respect to the first, in that parallel and distributed architectures present new capabilities for storage and manipulation of data. However, from an inferential point of view, it is not yet clear how statistical methodology will transport to a world involving massive data on parallel and distributed computing platforms.

Although massive data bring many statistical issues to the fore, there remains the core inferential need to assess the quality of estimators. Indeed, massive data may often motivate the consideration of a wide range of models and estimators, enhancing the need for control over biases and variance. Furthermore, the ability to assess estimator quality efficiently remains essential to allow efficient use of available resources by processing only as much data as is necessary to achieve a desired accuracy or confidence.

The bootstrap brings to bear various desirable features in the massive data setting, notably its relatively automatic nature and its applicability to a wide variety of inferential problems. Indeed, at first glance, the bootstrap would seem ideally suited to exploiting the trend towards parallel and distributed computing: one might imagine using different processors or compute nodes to process different bootstrap resamples independently in parallel. One must recall, however, that bootstrap-based quantities typically must be computed via a form of Monte Carlo approximation in which the estimator in question is repeatedly applied to resamples of the entire original observed data set. These resamples have size of the order of that of the original data, with approximately 63% of data points appearing at least once in each resample, and the need to process hundreds of such resamples may overwhelm computational resources for the large data sets that are increasingly encountered in practice. Indeed, in the massive data setting, computation of even a single point estimate on the full data set can be quite computationally demanding, and repeated computation of estimates on comparably sized resamples can be prohibitively costly.

Although the literature does contain some discussion of techniques for improving the computational efficiency of the bootstrap, that work is largely devoted to reducing the number of resamples that are required (Efron, 1988; Efron and Tibshirani, 1993). These techniques in general introduce significant additional complexity of implementation and do not eliminate the crippling need for repeated computation of estimates on resamples having size that is comparable with that of the original data set.

Another landmark in the development of simulation-based inference is subsampling (Politis *et al.*, 1999) and the closely related m out of n bootstrap (Bickel *et al.*, 1997). These methods initially appear to remedy the bootstrap's key computational shortcoming, as they only require repeated computation of estimates on resamples (or subsamples) that can be significantly smaller than the original data set. However, these procedures also have drawbacks. Though they are more generally consistent than the bootstrap, their finite sample behaviour can be worse, and their success is sensitive to the choice of resample (or subsample) size (Samworth, 2003). Additionally, because the variability of an estimator on a subsample differs from its variability on the full data set, these procedures must perform a rescaling of their output, and this rescaling requires knowledge and explicit use of the rate of convergence of the estimator in question; these methods are thus less automatic and easily deployable than the bootstrap. Although schemes have been proposed for data-driven selection of an optimal resample size (Bickel and Sakov, 2008), they require significantly greater computation which may eliminate any computational gains. Also, there has been work on the m out of n bootstrap that has sought to reduce computational costs by using two different values of m in conjunction with extrapolation (Bickel and Yahav, 1988; Bickel and Sakov, 2002). However, these approaches explicitly utilize series expansions of the estimator's sampling distribution and hence are less automatically usable; they also require execution of the m out of n bootstrap for multiple values of m .

Motivated by the need for an automatic, accurate means of assessing estimator quality that is scalable to large data sets, we introduce a new procedure, the ‘bag of little bootstraps’ (BLB), which functions by combining the results of bootstrapping multiple small subsets of a larger original data set. Instead of applying the estimator directly to each small subset, as in the m out of n bootstrap and subsampling, the BLB applies the bootstrap to each small subset. This is done by forming weighted resamples based on the small subset in such a way that the effect is that of sampling from the small subset n times with replacement, but the computational cost is that associated with the size of the small subset. The result is that, despite operating only on subsets of the original data set, the BLB does not require analytical rescaling of its output. Overall, the BLB has a significantly more favourable computational profile than the bootstrap, as it only requires repeated computation of estimates on quantities of data that can be much smaller than the original data set. As a result, the BLB is well suited to implementation on modern distributed and parallel computing architectures which are often used to process large data sets. Our procedure also maintains the bootstrap’s generic applicability, favourable statistical properties and simplicity of implementation. Moreover, as we show in experiments, the BLB is consistently more robust than alternatives such as the m out of n bootstrap and subsampling.

Of course, none of these procedures should be viewed as a black box solution to inferential problems, particularly in the setting of massive data, where data will often be heterogeneous, non-stationary and subject to non-random errors. Nonetheless, we expect resampling and subsampling methods to play an increasingly important role in inferential activities as data sets grow in size, and we view it as essential to consider jointly the computational and statistical aspects of this growth.

The remainder of our presentation is organized as follows. In Section 2, we formalize our statistical setting and notation, present the BLB in detail and discuss the procedure’s computational characteristics. Subsequently, we elucidate the BLB’s statistical properties via a theoretical analysis (Section 3) showing that it shares the bootstrap’s consistency and higher order correctness, as well as a simulation study (Section 4) which compares the BLB with the bootstrap, the m out of n bootstrap, and subsampling. Section 5 discusses a large-scale implementation of the BLB on a distributed computing system and presents results illustrating the procedure’s superior computational performance in the massive data setting. We present a method for adaptively selecting the BLB’s tuning parameters in Section 6. Finally, we apply the BLB (as well as the bootstrap and the m out of n bootstrap, for comparison) to several real data sets in Section 7, we present an extension of the BLB to time series data in Section 8 and we conclude with a discussion of the BLB’s limitations in Section 9.

2. Bag of little bootstraps

2.1. Setting and notation

We assume that we observe a sample X_1, \dots, X_n drawn independently and identically distributed (IID) from some unknown distribution P ; we denote by $\mathbb{P}_n = n^{-1} \sum_{i=1}^n \delta_{X_i}$ the corresponding empirical distribution. On the basis of the observed data, we form an estimate $\hat{\theta}_n = \hat{\theta}_n(\mathbb{P}_n)$ of some (unknown) population value $\theta(P)$. Our end goal is to obtain an assessment $\xi\{Q_n(P)\}$ of the quality of the estimate $\hat{\theta}_n(\mathbb{P}_n)$, which consists of a summary of the distribution $Q_n(P)$ of some quantity $u(\mathbb{P}_n, P)$, where ξ ranges over a vector space Ξ . The choice of u depends on one’s inferential goals: for instance, $\xi\{Q_n(P)\}$ might be the variance of $u(\mathbb{P}_n, P) = \hat{\theta}_n(\mathbb{P}_n)$, the expectation of $u(\mathbb{P}_n, P) = \hat{\theta}_n(\mathbb{P}_n) - \theta(P)$ (i.e. the bias) or a confidence region based on the distribution of $u(\mathbb{P}_n, P) = n^{1/2}\{\hat{\theta}_n(\mathbb{P}_n) - \theta(P)\}$. In practice, we cannot compute $\xi\{Q_n(P)\}$ directly because

P and $Q_n(P)$ are unknown, and so we must estimate $\xi\{Q_n(P)\}$ on the basis of a single observed data set.

Under this notation, the bootstrap simply computes the plug-in approximation $\xi\{Q_n(\mathbb{P}_n)\} \approx \xi\{Q_n(P)\}$. Although $\xi\{Q_n(\mathbb{P}_n)\}$ cannot be computed exactly in most cases, it is generally amenable to straightforward Monte Carlo approximation via the following algorithm (Efron and Tibshirani, 1993): repeatedly resample n points IID from \mathbb{P}_n , denote the empirical distribution of the resampled points as \mathbb{P}_n^* , compute $u(\mathbb{P}_n^*, \mathbb{P}_n)$ for each resample, let \mathbb{Q}_n^* denote the empirical distribution of the computed values of u , and finally compute $\xi(\mathbb{Q}_n^*) \approx \xi\{Q_n(P)\}$. Note that the vast majority of the bootstrap's computational cost lies in the repeated computation of values of u , which in turn requires costly repeated computation of estimates $\hat{\theta}_n(\mathbb{P}_n^*)$ on resamples.

Similarly, using our notation, the m out of n bootstrap (and subsampling) functions as follows, for $m < n$ (Bickel *et al.*, 1997; Politis *et al.*, 1999), repeatedly resample m points IID from \mathbb{P}_n (subsample m points without replacement from X_1, \dots, X_n), form the empirical distribution \mathbb{P}_m^* and compute $u(\mathbb{P}_m^*, \mathbb{P}_n)$ for each resample (subsample), form the empirical distribution \mathbb{Q}_m^* of the computed u -values, compute $\xi(\mathbb{Q}_m^*) \approx \xi\{Q_m(P)\}$ and apply an analytical correction in turn to approximate $\xi\{Q_n(P)\}$.

We use $\mathbf{1}_d$ to denote the d -dimensional vector of 1s, and we let I_d denote the $d \times d$ identity matrix.

2.2. Bag of little bootstraps method

The BLB functions by averaging the results of bootstrapping multiple small subsets of X_1, \dots, X_n . More formally, given a subset size $b < n$, the BLB samples s subsets of size b without replacement from the original n data points, uniformly at random (one can also impose the constraint that the subsets are disjoint). Let $\mathcal{I}_1, \dots, \mathcal{I}_s \subset \{1, \dots, n\}$ be the corresponding index sets (note that $|\mathcal{I}_j| = b, \forall j$), and let $\mathbb{P}_{n,b}^{(j)} = b^{-1} \sum_{i \in \mathcal{I}_j} \delta_{X_i}$ denote the empirical distribution corresponding to subset j . The BLB's estimate of $\xi\{Q_n(P)\}$ is then given by

$$s^{-1} \sum_{j=1}^s \xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\}, \quad (1)$$

where the key is the use of Q_n even though the object $\mathbb{P}_{n,b}^{(j)}$ has its support on only b points. Although the terms $\xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\}$ in expression (1) cannot be computed analytically in general, they can be computed numerically via straightforward Monte Carlo approximation in the manner of the bootstrap: for each term j , repeatedly resample n points IID from $\mathbb{P}_{n,b}^{(j)}$, form the empirical distribution $\mathbb{P}_{n,b}^*$ and compute $u(\mathbb{P}_{n,b}^*, \mathbb{P}_{n,b}^{(j)})$ for each resample, form the empirical distribution $\mathbb{Q}_{n,j}^*$ of the computed u -values, and compute $\xi(\mathbb{Q}_{n,j}^*) \approx \xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\}$.

Now, to realize the substantial computational benefits that are afforded by the BLB, we utilize the following crucial fact: each BLB resample, despite having nominal size n , contains at most b distinct data points. In particular, to generate each resample, it suffices to draw a vector of counts from an n -trial uniform multinomial distribution over b objects. We can then represent each resample by simply maintaining its empirical distribution, consisting of the at most b distinct points within it, accompanied by corresponding sampled counts. Therefore, each resample requires only storage space in $O(b)$. In turn, if the estimator (and u) can work directly with this weighted data representation, then the computational requirements of the estimator (and u)—with respect to both time and storage space—scale only in b , rather than n . Fortunately, this property does indeed hold for many if not most commonly used estimators, such as general M -estimators. The resulting BLB algorithm, including Monte Carlo resampling (algorithm 1), is shown in Table 1.

Table 1. Algorithm 1: the BLB

```

Input: data  $X_1, \dots, X_n$ ;  $b$ , subset size;  $s$ , number of sampled subsets;  $r$ ,
      number of Monte Carlo iterations;  $\xi, u$ , estimator quality assessment  $\xi$ 
      summarizing the distribution of quantity  $u$ 
Output: an estimate of  $\xi\{Q_n(P)\}$ 
For  $j \leftarrow 1$  to  $s$  do
  // subsample the data
  randomly sample a set  $\mathcal{I} = \{i_1, \dots, i_b\}$  of  $b$  indices from  $\{1, \dots, n\}$  without
  replacement (or, choose  $\mathcal{I}$  to be a disjoint subset of size  $b$  from a predefined
  random partition of  $\{1, \dots, n\}$ )
   $\mathbb{P}_{nb}^{(j)} \leftarrow b^{-1} \sum_{i \in \mathcal{I}} \delta_{X_i}$ 
  // approximate  $\xi\{Q_n(\mathbb{P}_{nb}^{(j)})\}$ 
  for  $k \leftarrow 1$  to  $r$  do
    sample  $(n_1, \dots, n_b) \sim \text{multinomial}(n, \mathbf{1}_b/b)$ 
     $\mathbb{P}_{nk}^* \leftarrow n^{-1} \sum_{a=1}^b n_a \delta_{X_{i_a}}$ 
     $u_{nk}^* \leftarrow u(\mathbb{P}_{nk}^*, \mathbb{P}_{nb}^{(j)})$ 
  end
   $\mathbb{Q}_{n,j}^* \leftarrow r^{-1} \sum_{k=1}^r \delta_{u_{nk}^*}$ 
   $\xi_{n,j}^* \leftarrow \xi(\mathbb{Q}_{n,j}^*)$ 
end
// average values of  $\xi\{Q_n(\mathbb{P}_{nb}^{(j)})\}$  computed for different data subsets
return  $s^{-1} \sum_{j=1}^s \xi_{n,j}^*$ 

```

The BLB thus avoids the bootstrap's problematic need for repeated computation on resamples having size comparable with that of the original data set. As noted earlier, a simple and standard calculation (Efron and Tibshirani, 1993) shows that each bootstrap resample contains approximately $0.632n$ distinct points, which is large if n is large. In contrast, as discussed above, each BLB resample contains at most b distinct points, and b can be chosen to be much smaller than n or $0.632n$. For example, we might take $b = n^\gamma$ where $\gamma \in [0.5, 1]$. More concretely, if $n = 1$ million, then each bootstrap resample would contain approximately 632000 distinct points, whereas with $b = n^{0.6}$ each BLB subsample and resample would contain at most 3981 distinct points. If each data point occupies 1 Mbyte of storage space, then the original data set would occupy 1 Tbyte, a bootstrap resample would occupy approximately 632 Gbytes, and each BLB subsample or resample would occupy at most 4 Gbytes. As a result, the cost of computing u (which entails computing the estimate) on each BLB resample is generally substantially lower than the cost of computing u on each bootstrap resample. Furthermore, as we show in our simulation study and scalability experiments below, the BLB typically requires less total computation (across multiple data subsets and resamples) than the bootstrap to reach comparably high accuracy; fairly modest values of s and r suffice.

Owing to its much smaller subsample and resample sizes, the BLB is also significantly more amenable than the bootstrap to the distributed processing of different subsamples and resamples and their associated computations on independent compute nodes; therefore, the BLB allows for simple distributed and parallel implementations, enabling additional large computational gains. In the large data setting, computing a single full data point estimate often requires simultaneous distributed computation across multiple compute nodes, among which the observed data set is partitioned. Given the large size of each bootstrap resample, computing the estimate on even a single such resample in turn also requires the use of a comparably large cluster of compute nodes; the bootstrap requires repetition of this computation for multiple resamples. Each computation

of the estimate (and hence of u) is thus quite costly, and the aggregate computational costs of this repeated distributed computation are quite high (indeed, the computation for each bootstrap resample requires use of an entire cluster of compute nodes and incurs the associated overhead).

In contrast, the BLB straightforwardly permits computation on multiple (or even all) subsamples and resamples simultaneously in parallel: because BLB subsamples and resamples can be significantly smaller than the original data set, they can be transferred to, stored by and processed on individual (or very small sets of) compute nodes. For example, we could naturally leverage modern hierarchical distributed architectures by distributing subsamples to different compute nodes and subsequently using intranode parallelism to compute across different resamples generated from the same subsample. Thus, relative to the bootstrap, the BLB both decreases the total computational cost of assessing estimator quality and allows more natural use of parallel and distributed computational resources. Moreover, even if only a single compute node is available, the BLB allows the following somewhat counterintuitive possibility: even if it is prohibitive to compute a point estimate for the full observed data using a single compute node actually (because the full data set is large), it may still be possible to assess efficiently such a point estimate's quality using only a single compute node by processing one subsample (and the associated resamples) at a time.

3. Consistency and higher order correctness

We now show that the BLB has statistical properties—in particular, consistency and higher order correctness—which are identical to those of the bootstrap, under the same conditions that have been used in prior analysis of the bootstrap. As in standard analyses of the bootstrap, we do not explicitly account here for error that is introduced by use of Monte Carlo approximation to compute the individual plug-in approximations $\xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\}$.

The following theorem states that (under standard assumptions) as $b, n \rightarrow \infty$, the estimates $s^{-1} \sum_{j=1}^s \xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\}$ returned by the BLB approach the population value $\xi\{Q_n(P)\}$ in probability. Interestingly, the only assumption about b required for this result is that $b \rightarrow \infty$, though in practice we would generally take b to be a slowly growing function of n .

Theorem 1. Suppose that $\hat{\theta}_n(\mathbb{P}_n) = \phi(\mathbb{P}_n)$ and $\theta(P) = \phi(P)$, where ϕ is Hadamard differentiable at P tangentially to some subspace, with P , \mathbb{P}_n and $\mathbb{P}_{n,b}^{(j)}$ viewed as maps from some Donsker class \mathcal{F} to \mathbb{R} such that \mathcal{F}_δ is measurable for every $\delta > 0$, where $\mathcal{F}_\delta = \{f - g : f, g \in \mathcal{F}, \rho_P(f - g) < \delta\}$ and $\rho_P(f) = \{P(f - Pf)^2\}^{1/2}$. Additionally, assume that $\xi\{Q_n(P)\}$ is a function of the distribution of $u(\mathbb{P}_n, P) = n^{1/2}\{\phi(\mathbb{P}_n) - \phi(P)\}$ which is continuous in the space of such distributions with respect to a metric that metrizes weak convergence. Then,

$$s^{-1} \sum_{j=1}^s \xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\} - \xi\{Q_n(P)\} \xrightarrow{P} 0$$

as $n \rightarrow \infty$, for any sequence $b \rightarrow \infty$ and for any fixed s .

See Appendix A for outlines of the proofs of the results that are presented in this section (and see the on-line supplementary materials for full proofs). Note that the assumptions of theorem 1 are standard in analysis of the bootstrap and in fact hold in many cases of practical interest. For example, M -estimators are generally Hadamard differentiable (under some regularity conditions) (van der Vaart, 1998; van der Vaart and Wellner, 1996), and the assumptions on ξ are satisfied if, for example, ξ is the value of the cumulative distribution function at a fixed point. Theorem 1 can also be generalized to hold for sequences $s \rightarrow \infty$ and more general forms of ξ , but such generalization appears to require stronger assumptions, such as uniform integrability of

the $\xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\}$; the need for stronger assumptions to obtain more general consistency results has also been noted in prior work on the bootstrap (see, for example, Hahn (1995)).

Moving beyond analysis of the BLB's consistency, we now characterize its higher order correctness. Much prior work has been devoted to showing that the bootstrap is higher order correct in many cases (see, for example, Hall (1992)), converging to the true value $\xi\{Q_n(P)\}$ at a rate of $O_P(1/n)$ or faster. As shown by the following theorem, the BLB shares the same degree of higher order correctness as the bootstrap, assuming that s and b are chosen to be sufficiently large. Importantly, sufficiently large values of b here can still be significantly smaller than n , with $b/n \rightarrow 0$ as $n \rightarrow \infty$. Following prior analyses of the bootstrap, we now make the standard assumption that ξ can be represented via an asymptotic series expansion in powers of $1/\sqrt{n}$; prior work provides such expansions in a variety of settings, for example, in the form of Edgeworth or Cornish–Fisher expansions (Hall, 1992).

Theorem 2. Suppose that $\xi\{Q_n(P)\}$ admits an expansion as an asymptotic series

$$\xi\{Q_n(P)\} = z + \frac{p_1}{\sqrt{n}} + \dots + \frac{p_k}{n^{k/2}} + o\left(\frac{1}{n^{k/2}}\right), \quad (2)$$

where z is a constant independent of P and the p_k are polynomials in the moments of P . Additionally, assume that the empirical version of $\xi\{Q_n(P)\}$ for any j admits a similar expansion

$$\xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\} = z + \frac{\hat{p}_1^{(j)}}{\sqrt{n}} + \dots + \frac{\hat{p}_k^{(j)}}{n^{k/2}} + o_P\left(\frac{1}{n^{k/2}}\right), \quad (3)$$

where z is as defined above and the $\hat{p}_k^{(j)}$ are polynomials in the moments of $\mathbb{P}_{n,b}^{(j)}$ obtained by replacing the moments of P in the p_k with those of $\mathbb{P}_{n,b}^{(j)}$. Then, assuming that $b \leq n$ and $E(\hat{p}_k^{(1)})^2 < \infty$ for $k \in \{1, 2\}$,

$$\left| s^{-1} \sum_{j=1}^s \xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\} - \xi\{Q_n(P)\} \right| = \sum_{k=1}^2 O_P \left[\frac{\sqrt{\{\text{var}(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n)\}}}{n^{k/2} \sqrt{s}} \right] + O_P\left(\frac{1}{n}\right) + o\left(\frac{1}{b\sqrt{n}}\right).$$

Therefore, taking $s = \Omega[\max\{n \text{var}(\hat{p}_1^{(1)} - p_1 | \mathbb{P}_n), \text{var}(\hat{p}_2^{(1)} - p_2 | \mathbb{P}_n)\}]$ and $b = \Omega(\sqrt{n})$ yields

$$\left| s^{-1} \sum_{j=1}^s \xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\} - \xi\{Q_n(P)\} \right| = O_P\left(\frac{1}{n}\right),$$

in which case the BLB enjoys the same level of higher order correctness as the bootstrap.

It is natural to assume that $\xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\}$ can be expanded in powers of $1/\sqrt{n}$, rather than $1/\sqrt{b}$, because $Q_n(\mathbb{P}_{n,b}^{(j)})$ is the distribution of values of u computed on the basis of n points sampled from $\mathbb{P}_{n,b}^{(j)}$. The fact that only b points are represented in $\mathbb{P}_{n,b}^{(j)}$ enters via the $\hat{p}_k^{(j)}$, which are polynomials in the sample moments of those b points.

Theorem 2 indicates that, like the bootstrap, the BLB can converge at rate $O_P(1/n)$ (assuming that s and b grow at a sufficient rate). Additionally, because $\text{var}(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n)$ is decreasing in probability as b and n increase, s can grow significantly more slowly than n (indeed, unconditionally, $\hat{p}_k^{(j)} - p_k = O_P(1/\sqrt{b})$). Although $\text{var}(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n)$ can in principle be computed given an observed data set, as it depends only on \mathbb{P}_n and the form of u and the estimator under consideration, we can also obtain a general bound (in probability) on the rate of decrease of this conditional variance.

Remark 1. Assuming that $E(\hat{p}_k^{(1)})^4 < \infty$, $\text{var}(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n) = O_P(1/\sqrt{n}) + O(1/b)$.

The following result, which applies to the alternative variant of the BLB that constrains the s randomly sampled subsets to be disjoint, also highlights the fact that s can grow substantially more slowly than n .

Theorem 3. Under the assumptions of theorem 2, and assuming that the BLB uses disjoint random subsets of the observed data (rather than simple random subsamples), we have

$$\left| s^{-1} \sum_{j=1}^s \xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\} - \xi\{Q_n(P)\} \right| = O_P\left\{ \frac{1}{\sqrt{(nbs)}} \right\} + O\left(\frac{1}{b\sqrt{n}} \right).$$

Therefore, if $s \sim n/b$ and $b = \Omega(\sqrt{n})$, then

$$\left| s^{-1} \sum_{j=1}^s \xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\} - \xi\{Q_n(P)\} \right| = O_P\left(\frac{1}{n} \right),$$

in which case the BLB enjoys the same level of higher order correctness as the bootstrap.

Finally, although the assumptions of the two preceding theorems generally require that u Studentizes the estimator under consideration, similar results hold even if the estimator is not Studentized. In particular, not Studentizing slows the rate of convergence of both the bootstrap and the BLB by the same factor, generally causing the loss of a factor of $O_P(1/\sqrt{n})$ (van der Vaart, 1998).

4. Simulation study

In this section we compare the statistical performance of the BLB with that of existing methods via experiments on simulated data. The use of simulated data is necessary here because it allows knowledge of P and $Q_n(P)$, and hence $\xi\{Q_n(P)\}$; this ground truth is required for evaluation of statistical correctness. For different data sets and estimation tasks, we study the convergence properties of the BLB as well as the bootstrap, the m out of n bootstrap and subsampling.

We consider two different settings: regression and classification. For both settings, the data have the form $X_i = (\tilde{X}_i, Y_i) \sim P$, IID for $i = 1, \dots, n$, where $\tilde{X}_i \in \mathbb{R}^d$; $Y_i \in \mathbb{R}$ for regression, whereas $Y_i \in \{0, 1\}$ for classification. In each case, $\hat{\theta}_n$ estimates a parameter vector in \mathbb{R}^d for a linear or generalized linear model of the mapping between \tilde{X}_i and Y_i . We define ξ as a procedure that computes a set of marginal 95% confidence intervals, one for each element of the estimated parameter vector. In particular, given the distribution $Q_n(P)$ of $u(\mathbb{P}_n, P) = \hat{\theta}_n(\mathbb{P}_n)$ (or an approximation thereof), ξ forms the boundaries of the relevant confidence intervals as the 2.5th and 97.5th percentiles of the marginal componentwise distributions defined by $Q_n(P)$; averaging across these confidence intervals in the averaging step of the BLB simply consists in averaging these percentile estimates.

To evaluate the various quality assessment procedures on a given estimation task and true underlying data distribution P , we first compute the ground truth $\xi\{Q_n(P)\}$ by generating 2000 realizations of data sets of size n from P , computing $\hat{\theta}_n$ on each, and using this collection of $\hat{\theta}_n$ s to form a high fidelity approximation to $Q_n(P)$. Then, for an independent data set realization of size n from the true underlying distribution, we run each quality assessment procedure (without parallelization) until it converges and record the estimate of $\xi\{Q_n(P)\}$ produced after each iteration (e.g. after each bootstrap resample or BLB subsample has been processed), as well as the cumulative processing time required to produce that estimate. Every such estimate is evaluated on the basis of the average (across dimensions) relative deviation of its componentwise confidence intervals' widths from the corresponding true widths; given an estimated

confidence interval width c and a true width c_0 , the relative deviation of c from c_0 is defined as $|c - c_0|/c_0$. We repeat this process on five independent data set realizations of size n and average the resulting relative errors and corresponding processing times across these five data sets to obtain a trajectory of relative error *versus* time for each quality assessment procedure. The relative errors' variances are small relative to the relevant differences between their means, and so these variances are not shown in our plots. Note that we evaluate on the basis of confidence interval widths, rather than coverage probabilities, to control the running times of our experiments: in our experimental setting, even a single run of a quality assessment procedure requires non-trivial time, and computing coverage probabilities would require a large number of such runs. All experiments in this section were implemented and executed using MATLAB (<http://www.mathworks.com/>) on a single processor. To maintain consistency of notation, we refer to the m out of n bootstrap as the b out of n bootstrap throughout the remainder of this section. For the BLB, the b out of n bootstrap, and subsampling, we consider $b = n^\gamma$ with $\gamma \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$; we use $r = 100$ in all runs of the BLB.

In the regression setting, we generate each data set from a true underlying distribution P consisting of either a linear model $Y_i = \tilde{X}_i^T \mathbf{1}_d + \varepsilon_i$ or a model $Y_i = \tilde{X}_i^T \mathbf{1}_d + \tilde{X}_i^T \tilde{X}_i + \varepsilon_i$ having a quadratic term, with $d = 100$ and $n = 20000$. The \tilde{X}_i and ε_i are drawn independently from one of the following pairs of distributions: $\tilde{X}_i \sim \text{normal}(0, I_d)$ with $\varepsilon_i \sim \text{normal}(0, 10)$, $\tilde{X}_{i,j} \sim \text{StudentT}(3)$ IID for $j = 1, \dots, d$ with $\varepsilon_i \sim \text{normal}(0, 10)$ or $\tilde{X}_{i,j} \sim \text{gamma}\{1 + 5(j - 1)/\max(d - 1, 1), 2\} - 2\{1 + 5(j - 1)/\max(d - 1, 1), 2\}$ independently for $j = 1, \dots, d$ with $\varepsilon_i \sim \text{gamma}(1, 2) - 2$. All these distributions have $E(\tilde{X}_i) = E(\varepsilon_i) = 0$, and the last \tilde{X}_i -distribution has non-zero skewness which varies among the dimensions. In the regression setting under both the linear and the quadratic data-generating distributions, our estimator $\hat{\theta}_n$ consists of a linear least squares regression with a small L_2 -penalty on the parameter vector to encourage numerical stability (we set the weight on this penalty term to 10^{-5}). The true average (across dimensions) marginal confidence interval width for the estimated parameter vector is approximately 0.1 under the linear data-generating distributions (for all \tilde{X}_i -distributions) and approximately 1 under the quadratic data-generating distributions.

Fig. 1 shows results for the regression setting under the linear and quadratic data-generating distributions with the gamma and StudentT \tilde{X}_i -distributions; similar results hold for the normal \tilde{X}_i -distribution. In all cases, the BLB (Figs 1(a)–1(c)) succeeds in converging to low relative error significantly more quickly than the bootstrap, for all values of b considered. In contrast, the b out of n bootstrap (Figs 1(d)–1(f)) fails to converge to low relative error for smaller values of b (below $n^{0.7}$). Additionally, subsampling (Figs 1(g)–1(i)) performs strictly worse than the b out of n bootstrap, as subsampling fails to converge to low relative error for both smaller and larger values of b (e.g. for $b = n^{0.9}$). Note that fairly modest values of s suffice for convergence of the BLB (recall that s -values are implicit in the time axes of our plots), with s at convergence ranging from 1–2 for $b = n^{0.9}$ up to 10–14 for $b = n^{0.5}$ in the experiments shown in Fig. 1; larger values of s are required for smaller values of b , which accords with both intuition and our theoretical analysis. Under the quadratic data-generating distribution with StudentT \tilde{X}_i -distribution (for which plots are not shown), none of the procedures (including the bootstrap) converge to low relative error, which is unsurprising given the StudentT(3) distribution's lack of moments beyond order 2.

For the classification setting, we generate each data set considered from either a linear model $Y_i \sim \text{Bernoulli}[\{1 + \exp(-\tilde{X}_i^T \mathbf{1}_d)\}^{-1}]$ or a model $Y_i \sim \text{Bernoulli}[\{1 + \exp(-\tilde{X}_i^T \mathbf{1}_d - \tilde{X}_i^T \tilde{X}_i)\}^{-1}]$ having a quadratic term, with $d = 10$. We use the same three distributions on \tilde{X}_i that were used in the regression setting. Our estimator, under both the linear and the quadratic data-generating distributions, consists of a linear (in \tilde{X}_i) logistic regression fit via Newton's method, again

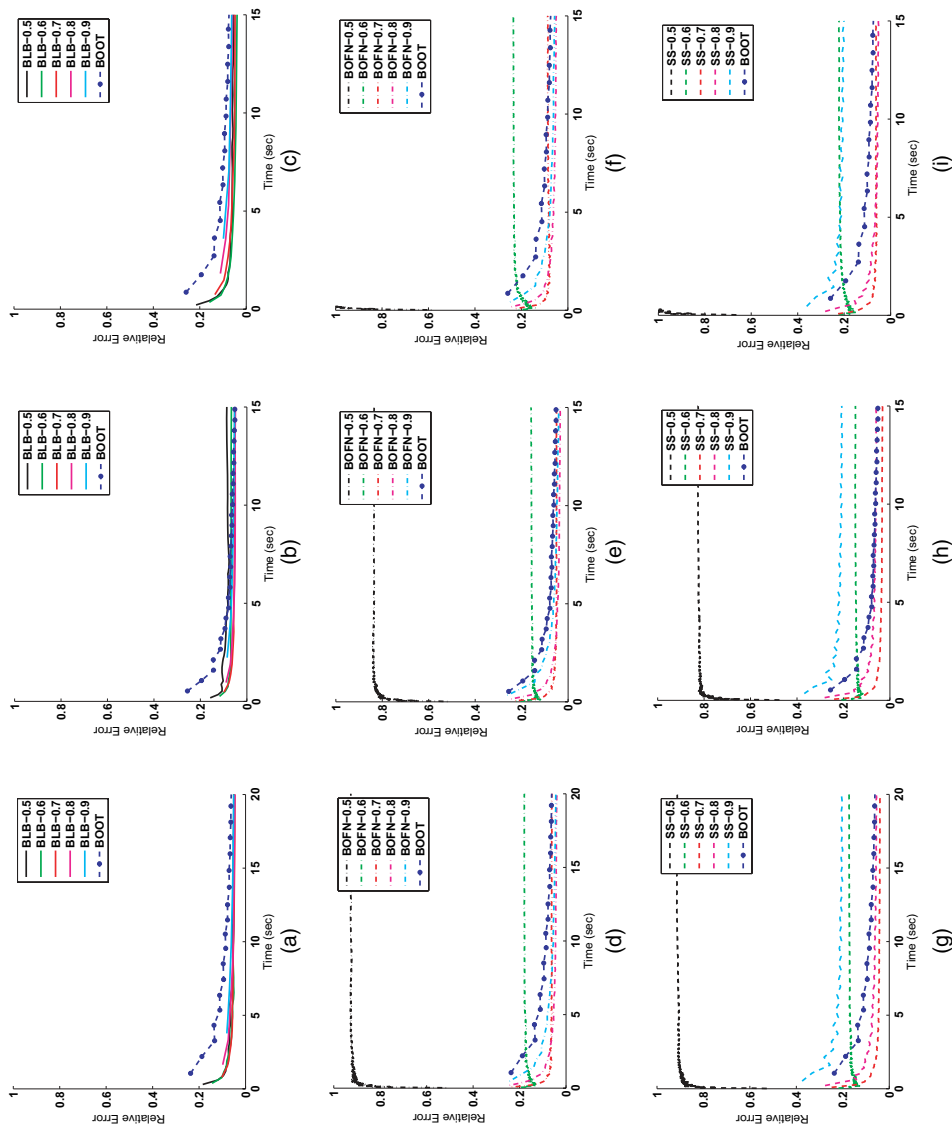


Fig. 1. Relative error versus processing time for the regression setting (for the BLB, b out of n bootstrap (BOFN) and subsampling (SS), $b = n^\gamma$ with the value of γ for each trajectory given in the keys): (a)–(c) the BLB with the bootstrap (BOOT); (d)–(f) the b out of n bootstrap; (g)–(i) subsampling; (a), (d), (g) results for linear regression with a linear data-generating distribution and gamma X_j -distribution; (b), (e), (h) results for linear regression with a quadratic data-generating distribution and gamma X_j -distribution; (c), (f), (i) results for linear regression with a linear data-generating distribution and StudentT X_j -distribution

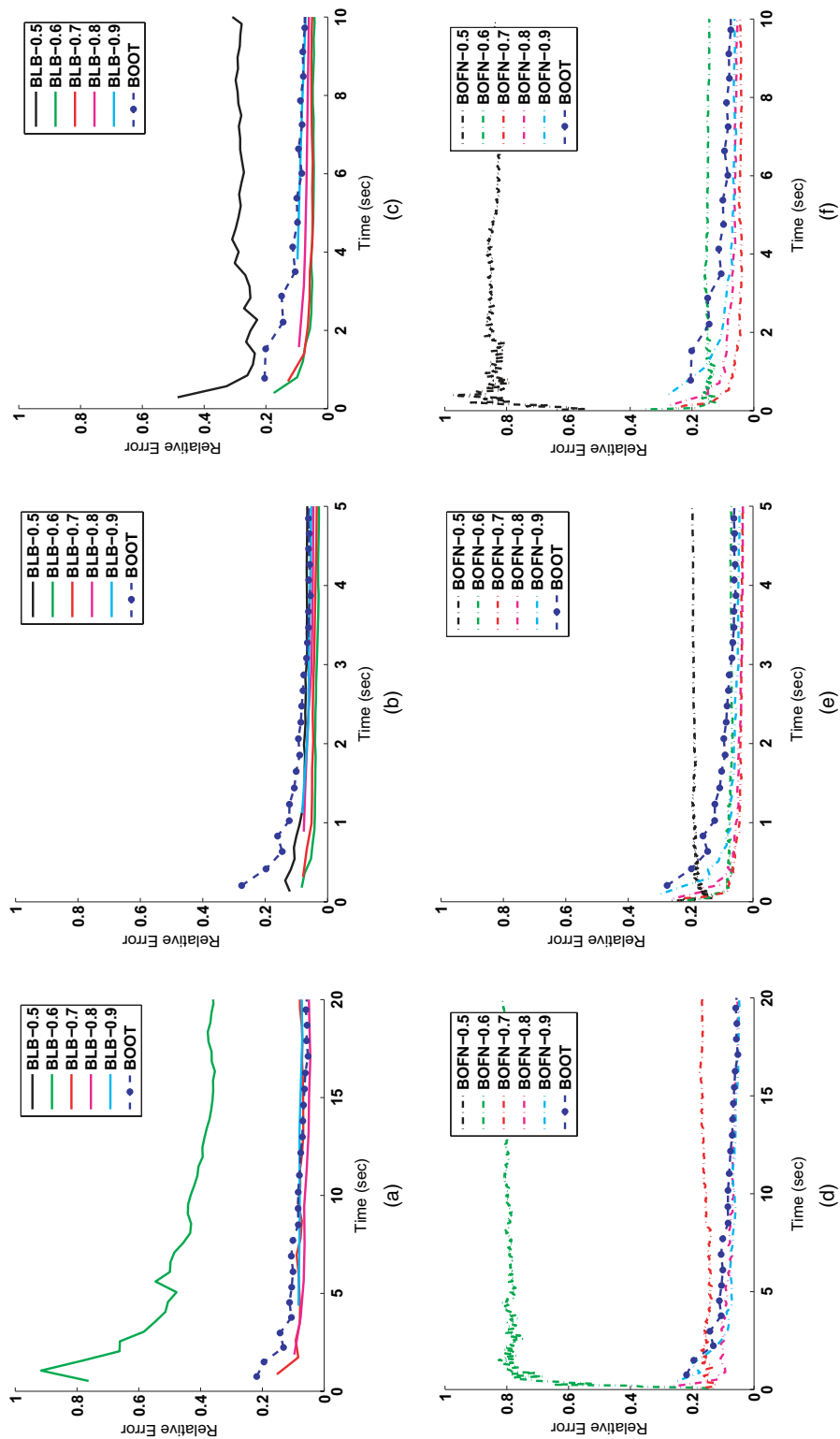


Fig. 2. Relative error *versus* processing time for the classification setting with $n = 20000$ (for both the BLB and the b out of n bootstrap (BOFN), $b = n^\gamma$ with the value of γ for each trajectory given in the keys): (a)–(c) the BLB with the bootstrap (BOOT); (d)–(f) the b out of n bootstrap (BOFN); (a), (d) results for logistic regression with a linear data-generating distribution and gamma \hat{X}_γ -distribution; (b), (e) results for logistic regression with a quadratic data-generating distribution and gamma \hat{X}_γ -distribution; (c), (f) results for logistic regression with a linear data-generating distribution and Student \hat{X}_γ -distribution

using an L_2 -penalty term with weight 10^{-5} to encourage numerical stability. For this estimation task with $n = 20000$, the true average (across dimensions) marginal confidence interval width for the estimated parameter vector is approximately 0.1 under the linear data-generating distributions (for all \tilde{X}_i -distributions) and approximately 0.02 under the quadratic data-generating distributions.

Fig. 2 shows results for the classification setting under the linear and quadratic data-generating distributions with the gamma and StudentT \tilde{X}_i -distributions, and $n = 20000$ (as in Fig. 1); results for the normal \tilde{X}_i -distribution are qualitatively similar. Here, the performance of the various procedures is more varied than in the regression setting. The case of the linear data-generating distribution with gamma \tilde{X}_i -distribution (Figs 2(a) and 2(d)) appears to be the most challenging. In this setting, the BLB converges to relative error comparable with that of the bootstrap for $b > n^{0.6}$ and converges to higher relative errors for the smallest values of b that were considered. For the larger values of b , which are still significantly smaller than n , we again converge to low relative error faster than the bootstrap. We are also once again more robust than the b out of n bootstrap, which fails to converge to low relative error for $b \leq n^{0.7}$. In fact, even for $b \leq n^{0.6}$, the BLB's performance is superior to that of the b out of n bootstrap. Qualitatively similar results hold for the other data-generating distributions, but with the BLB and the b out of n bootstrap both performing better relatively to the bootstrap. In the experiments that are shown in Fig. 2, the values of s (which are implicit in the time axes of our plots) that are required for convergence of the BLB range from 1–2 for $b = n^{0.9}$ up to 10–20 for $b \leq n^{0.6}$ (for cases in which the BLB converges to low relative error). As in the regression setting, subsampling (for which plots are not shown) has a performance that is strictly worse than that of the b out of n bootstrap in all cases.

To examine further the cases in which the BLB (when using small values of b) does not converge to relative error comparable with that of the bootstrap, we explore how the various procedures' relative errors vary with n . In particular, for various values of n (and b), we run each procedure as described above and report the relative error that it achieves after it converges (i.e. after it has processed sufficiently many subsets, in the case of the BLB, or resamples, in the case

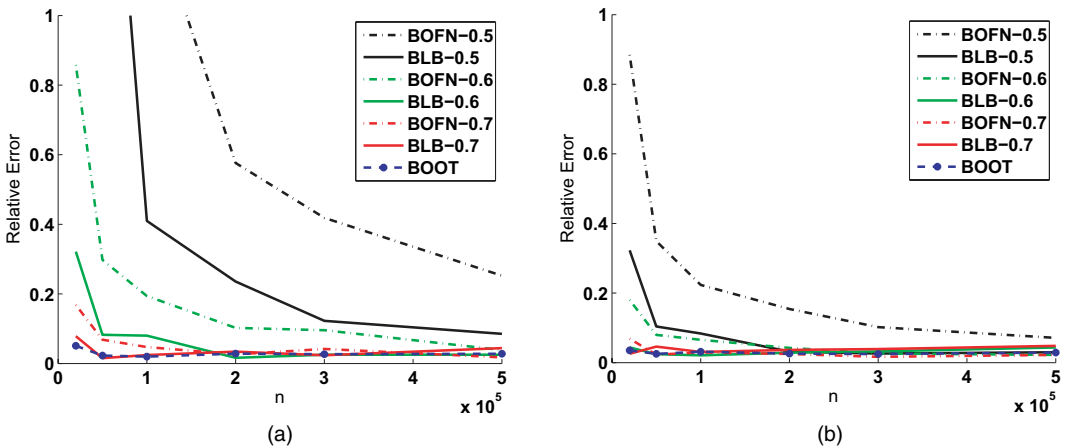


Fig. 3. Relative error (after convergence) versus n for the BLB, the b out of n bootstrap (BOFN) and the bootstrap (BOOT) in the classification setting (for both the BLB and the b out of n bootstrap, $b = n^\gamma$ with the relevant values of γ given in the keys): (a) results for logistic regression with a linear data-generating distribution and gamma \tilde{X}_i -distribution; (b) results for logistic regression with a linear data-generating distribution and StudentT \tilde{X}_i -distribution

of the b out of n bootstrap and the bootstrap, to allow its output to stabilize). Fig. 3 shows results for the classification setting under the linear data-generating distribution with the gamma and StudentT \tilde{X}_i -distributions; qualitatively similar results hold for the normal \tilde{X}_i -distribution. As expected on the basis of our previous results for fixed n , the BLB's relative error here is higher than that of the bootstrap for the smallest values of b and n that were considered. Nonetheless, the BLB's relative error decreases to that of the bootstrap as n increases—for all values of γ considered, with $b = n^\gamma$ —in accordance with our theoretical analysis; indeed, as n increases, we can set b to progressively more slowly growing functions of n while still achieving low relative error. Furthermore, the BLB's relative error is consistently substantially lower than that of the b out of n bootstrap and decreases more quickly to the low relative error of the bootstrap as n increases.

5. Computational scalability

The experiments of the preceding section, though primarily intended to investigate statistical performance, also provide some insight into computational performance: as seen in Figs 1 and 2, when computing on a single processor, the BLB generally requires less time, and hence less total computation, than the bootstrap to attain comparably high accuracy. Those results only hint at the BLB's superior ability to scale computationally to large data sets, which we now demonstrate via large-scale experiments on a distributed computing platform.

As discussed in Section 2, modern massive data sets often exceed both the processing and the storage capabilities of individual processors or compute nodes, thus necessitating the use of parallel and distributed computing architectures. As a result, the scalability of a quality assessment method is closely tied to its ability to utilize such computing resources effectively.

Recall from our exposition in preceding sections that, owing to the large size of bootstrap resamples, the following is the most natural avenue for applying the bootstrap to large-scale data by using distributed computing: given data partitioned across a cluster of compute nodes, parallelize the computation of the estimate (and hence u) on each resample across the cluster, and compute on one resample at a time. This approach, although at least potentially feasible, remains quite problematic. Each computation of the estimate will require the use of an entire cluster of compute nodes, and the bootstrap repeatedly incurs the associated overhead, such as the cost of repeatedly communicating intermediate data between nodes. Additionally, many cluster computing systems that are currently in widespread use (e.g. Hadoop MapReduce (<http://hadoop.apache.org>)) store data only on disc, rather than in memory, owing to physical size constraints (if the data set size exceeds the amount of available memory) or architectural constraints (e.g. the need for fault tolerance). In that case, the bootstrap incurs the extreme costs that are associated with repeatedly reading a very large data set from disc—reads from disc are orders of magnitude slower than reads from memory. Though disc read costs may be acceptable when (slowly) computing only a single full data point estimate, they easily become prohibitive when computing many estimates on 100 or more resamples. Furthermore, as we have seen, executing the bootstrap at scale requires implementing the estimator such that it can be run on data distributed over a cluster of compute nodes.

In contrast, the BLB permits computation on multiple (or even all) subsamples and resamples simultaneously in parallel, allowing for straightforward distributed and parallel implementations which enable effective scalability and large computational gains. For instance, given the relatively small size of BLB subsamples and resamples, we can distribute subsamples to different compute nodes and subsequently use intranode parallelism to compute across different resamples generated from the same subsample. Note that generation and distribution of the

subsamples require only a single pass over the full data set (i.e. only a single read of the full data set from disc, if it is stored only on disc), after which all required data (i.e. the subsamples) can potentially be stored in memory. Beyond this significant architectural benefit, we also achieve implementation and algorithmic benefits: we do not need to parallelize the estimator internally to take advantage of the available parallelism, as the BLB uses this available parallelism to compute on multiple resamples simultaneously, and exposing the estimator to only b rather than n distinct points significantly reduces the computational cost of estimation, particularly if the computation of the estimator scales superlinearly.

Given the shortcomings of the m out of n bootstrap and subsampling as illustrated in the preceding section, we do not include these methods in the scalability experiments of this section. However, it is worth noting that these procedures have a significant computational shortcoming in the setting of large-scale data: the m out of n bootstrap and subsampling require repeated access to many different random subsets of the original data set (in contrast with the relatively few, potentially disjoint, subsamples that are required by the BLB), and this access can be quite costly when the data are distributed across a cluster of compute nodes.

We now detail our large-scale experiments on a distributed computing platform. For this empirical study, we use the experimental set-up of Section 4, with some modification to accommodate larger scale and distributed computation. First, we now use $d = 3000$ and $n = 6$ million so the size of a full observed data set is approximately 150 Gbytes. The full data set is partitioned across a number of compute nodes. We again use simulated data to allow knowledge of ground truth; because of the substantially larger data size and attendant higher running times, we now use 200 independent realizations of data sets of size n to compute the ground truth numerically. As our focus is now computational (rather than statistical) performance, we present results here for a single data-generating distribution which yields representative statistical performance based on the results of the previous section; for a given size of data set, changing the underlying data-generating distribution does not alter the computational resources that are required for storage and processing. For the experiments in this section, we consider the classification setting with StudentT \tilde{X}_i -distribution. The mapping between \tilde{X}_i and Y_i remains similar to that of the linear data-generating distribution in Section 4, but with the addition of a normalization factor to prevent degeneracy when using larger d : $Y_i \sim \text{Bernoulli}[\{1 + \exp(-\tilde{X}_i^T \mathbf{1}_d / \sqrt{d})\}^{-1}]$. We implement the logistic regression by using L-BFGS (Nocedal and Wright, 2006) owing to the significantly larger value of d .

We compare the performance of the BLB and the bootstrap, both implemented as described above. That is, our implementation of the BLB processes all subsamples simultaneously in parallel on independent compute nodes; we use $r = 50$, $s = 5$ and $b = n^{0.7}$. Our implementation of the bootstrap uses all available processors to compute on one resample at a time, with computation of the logistic regression parameter estimates parallelized across the available compute nodes by simply distributing the relevant gradient computations among the different nodes on which the data are partitioned. We utilize Poisson resampling (van der Vaart and Wellner, 1996) to generate bootstrap resamples, thereby avoiding the complexity of generating a random multinomial vector of length n in a distributed fashion. Owing to high running times, we show results for a single trial of each method, though we have observed little variability in qualitative outcomes during development of these experiments. All experiments in this section were run on Amazon EC2 and implemented in the Scala programming language using the Spark cluster computing framework (Zaharia *et al.*, 2012), which provides the ability either to read data from disc (in which case the performance is similar to that of Hadoop MapReduce) or to cache it in memory across a cluster of compute nodes (provided that sufficient memory is available) for faster repeated access.

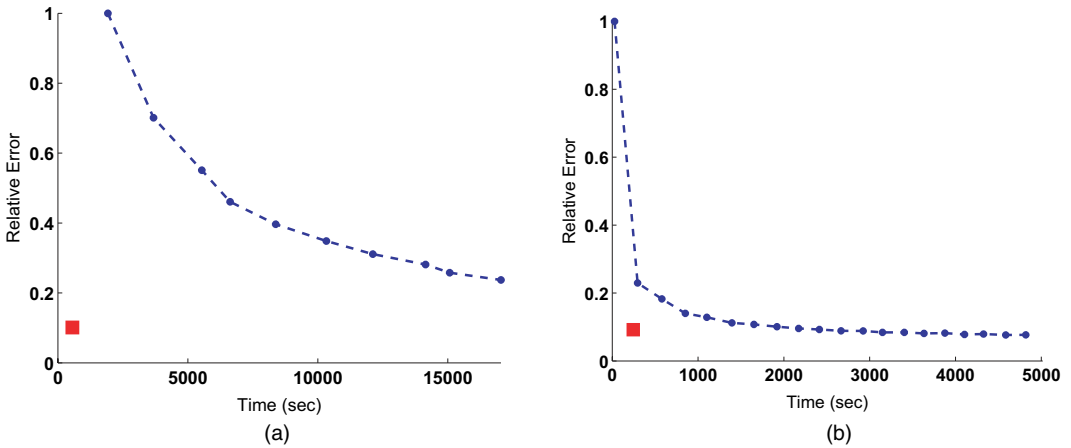


Fig. 4. Relative error *versus* processing time for the BLB with $b = n^{0.7}$ (■) and the bootstrap (●) on 150 Gbytes of data in the classification setting (because the BLB's computation is fully parallelized across all subsamples, we show only the processing time and relative error of the BLB's final output): (a) results with the full data set stored only on disc; (b) results with the full data set cached in memory

In Fig. 4(a), we show results obtained by using a cluster of 10 worker nodes, each having 6 Gbytes of memory and eight compute cores; thus, the total memory of the cluster is 60 Gbytes, and the full data set (150 Gbytes) can only be stored on disc (the available disc space is ample and far exceeds the size of the data set). As expected, the time that is required by the bootstrap to produce even a low accuracy output is prohibitively high, whereas the BLB provides a high accuracy output quite quickly, in less than the time that is required to process even a single bootstrap resample. In Fig. 4(b), we show results that were obtained by using a cluster of 20 worker nodes, each having 12 Gbytes of memory and four compute cores; thus, the total memory of the cluster is 240 Gbytes, and we cache the full data set in memory for faster repeated access. Unsurprisingly, the bootstrap's performance improves significantly with respect to the previous disc-bound experiment. However, the performance of the BLB (which also improves) remains substantially better than that of the bootstrap.

6. Tuning parameter selection

Like existing resampling-based procedures such as the bootstrap, the BLB requires the specification of tuning parameters controlling the number of subsamples and resamples processed. Setting such tuning parameters to be sufficiently large is necessary to ensure good statistical performance; however, setting them to be unnecessarily large results in wasted computation. Prior work on the bootstrap and related procedures—which largely does not address computational issues—generally assumes that a procedure's user will simply select *a priori* a large, constant number of resamples to be processed (an exception is Tibshirani (1985), who discussed the issue but did not provide a general solution). However, this approach reduces the level of automation of these methods and can be quite inefficient in the large data setting, in which each subsample or resample can require a substantial amount of computation.

Thus, we now examine the dependence of the BLB's performance on the choice of r and s , with the goal of better understanding their influence and providing guidance towards achieving adaptivity in their selection. For any particular application of the BLB, we seek to select the minimal values of r and s which are sufficiently large to yield good statistical performance.

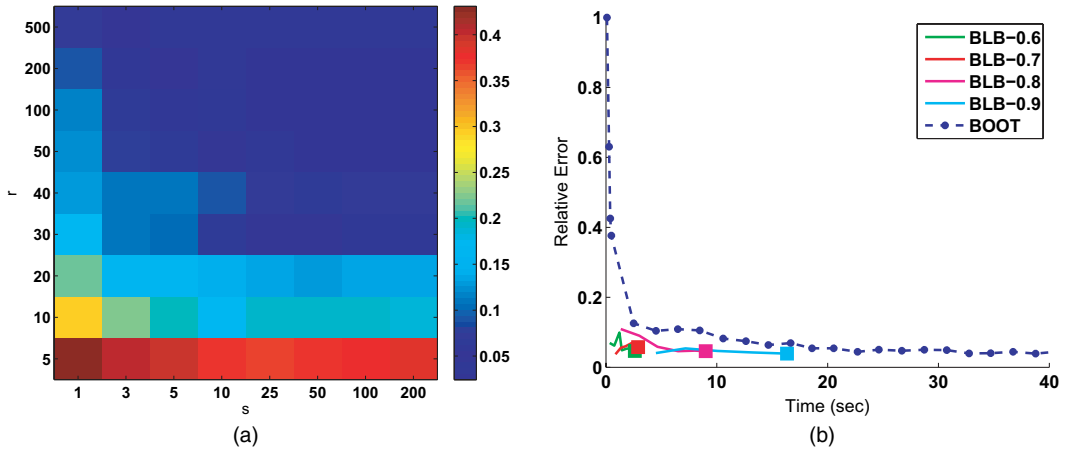


Fig. 5. Results for BLB tuning parameter selection: (a) relative error achieved by the BLB for various values of r and s , with $b = n^{0.7}$; (b) relative error versus processing time (without parallelization) for the BLB with adaptive selection of r and s (■), resulting stopping times of the BLB trajectories) and the bootstrap (BOOT); for the BLB, $b = n^\gamma$ with the value of γ for each trajectory given in the key

Recall that in the simulation study of Section 4, across all of the settings considered, fairly modest values of r (100 for confidence intervals) and s (from 1–2 for $b = n^{0.9}$ up to 10–20 for $b = n^{0.6}$) were sufficient. Fig. 5(a) provides further insight into the influence of r and s , giving the relative errors that are achieved by the BLB with $b = n^{0.7}$ for various r, s pairs in the classification setting with a linear data-generating distribution and StudentT \bar{X}_i -distribution. In particular, note that, except for the smallest values of r and s , it is possible to choose these values independently such that the BLB achieves low relative error; in this case, selecting $s \geq 3$ and $r \geq 50$ is sufficient.

Although these results provide some guidance for tuning parameter selection, we expect the sufficient values of r and s to change on the basis of the identity of u and ξ (for example, we expect a confidence interval to be more difficult to compute and hence to require larger r than a standard error) and the properties of the underlying data. Thus, to help to avoid the need to set r and s to be conservatively and inefficiently large, we now provide a means for adaptive tuning parameter selection, which we validate empirically.

Concretely, to select r adaptively in the inner loop of the BLB algorithm (see Table 1), we propose an iterative scheme whereby, for any given subsample j , we continue to process resamples and to update $\xi_{n,j}^*$ until it has ceased to change significantly. Noting that the values $u_{n,k}^*$ that are used to compute $\xi_{n,j}^*$ are conditionally IID given $\mathbb{P}_{n,b}^{(j)}$, for most forms of ξ the series of computed $\xi_{n,j}^*$ -values will be well behaved and will converge (in many cases at rate $O(1/\sqrt{r})$, though with unknown constant) to a constant target value as more resamples are processed. Therefore, it suffices to process resamples (i.e. to increase r) until we are satisfied that $\xi_{n,j}^*$ has ceased to fluctuate significantly; we propose to use algorithm 2 (detailed in Table 2) to assess this convergence. The same scheme can be used to select s adaptively by processing more subsamples (i.e. increasing s) until the BLB's output value $s^{-1} \sum_{j=1}^s \xi_{n,j}^*$ has stabilized; in this case, one can simultaneously also choose r adaptively and independently for each subsample. When parallelizing across subsamples and resamples, one can simply process batches of subsamples and resamples (with batch size determined by the available parallelism) until the output stabilizes.

Fig. 5(b) shows the results of applying such adaptive tuning parameter selection in a representative empirical setting from our earlier simulation study (without parallelization). For selection

Table 2. Algorithm 2: convergence assessment

<i>Input:</i> a series $z^{(1)}, z^{(2)}, \dots, z^{(t)} \in \mathbb{R}^d$; $w \in \mathbb{N}$, window size ($< t$); $\varepsilon \in \mathbb{R}$, target relative error (> 0)
<i>Output:</i> true if and only if the input series is deemed to have ceased to fluctuate beyond the target relative error
If $\forall j \in [1, w], (1/d) \sum_{i=1}^d z_i^{(t-j)} - z_i^{(t)} / z_i^{(t)} \leq \varepsilon$ then
return <i>true</i>
otherwise
return <i>false</i>
end

Table 3. Statistics of the various values of r selected by the BLB's adaptive tuning selection (across multiple subsamples, with $b = n^{0.7}$) when ξ is either our confidence interval width based measure or a componentwise standard error†

	<i>Confidence interval</i>	<i>Standard error</i>
Mean	89.6	67.7
Minimum	50	40
Maximum	150	110

†The relative errors achieved by the BLB and the bootstrap are comparable in both cases.

of r we use $\varepsilon = 0.05$ and $w = 20$, and for selection of s we use $\varepsilon = 0.05$ and $w = 3$. As illustrated in Fig. 5(b) the adaptive tuning parameter selection allows the BLB to cease computing shortly after it has converged (to low relative error), limiting the amount of unnecessary computation that is performed without degradation of statistical performance. Though selected *a priori*, ε and w are more intuitively interpretable and less dependent on the details of u and ξ and the underlying data-generating distribution than r and s . Indeed, the aforementioned specific values of ε and w yield results of comparably good quality when also used for the other data generation settings that were considered in Section 4, when applied to a variety of real data sets in Section 7 below, and when used in conjunction with different forms of ξ (see Table 3, which shows that smaller values of r are selected when ξ is easier to compute). Thus, our scheme significantly helps to alleviate the burden of *a priori* tuning parameter selection.

Automatic selection of a value of b in a computationally efficient manner would also be desirable but is more difficult because of the inability to reuse computations that are performed for different values of b easily. One could consider similarly increasing b from some small value until the output of the BLB stabilizes (an approach that is reminiscent of the method that was proposed in Bickel and Sakov (2008) for the m out of n bootstrap); devising a means of doing so efficiently is the subject of future work. Nonetheless, on the basis of our fairly extensive empirical investigation, it seems that $b = n^{0.7}$ is a reasonable and effective choice in many situations.

7. Real data

We now present the results of applying the BLB to several real data sets. In this context, given the absence of ground truth (i.e. the fact that the true value of $\xi\{Q_n(P)\}$ is unknown for a real data

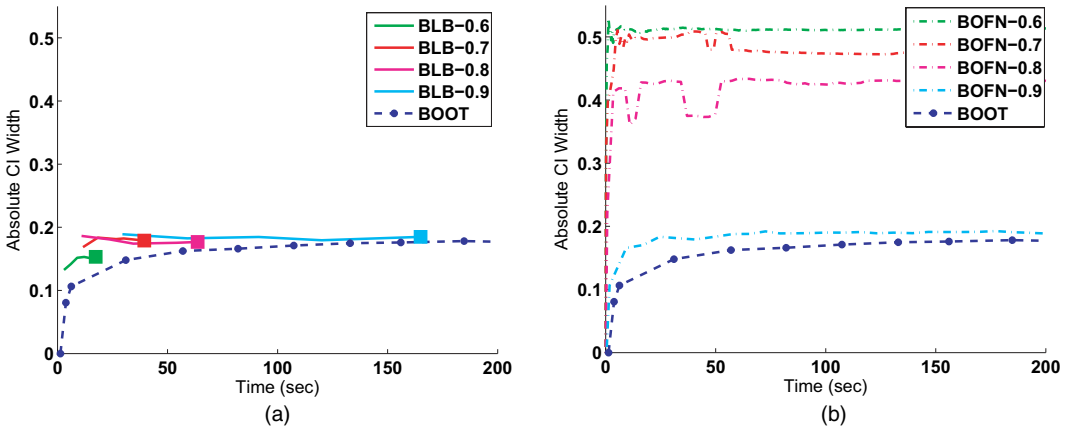


Fig. 6. Average (across dimensions) absolute confidence interval width *versus* processing time on the University of California at Irvine connect4 data set (logistic regression; $d = 42$; $n = 67557$) (for both the BLB and the b out of n bootstrap, $b = n^\gamma$ with the value of γ for each trajectory given in the keys): (a) results for the BLB (using adaptive tuning parameter selection; ■, output at convergence) and the bootstrap (BOOT); (b) results for the b out of n bootstrap (BOFN)

set), it is not possible to evaluate the statistical correctness of any particular estimator quality assessment method objectively; rather, we are reduced to comparing the outputs of various methods (in this case, the BLB, the bootstrap and the b out of n bootstrap) with each other. Because we cannot determine the relative error of each procedure's output without knowledge of the ground truth, we now instead report the average (across dimensions) absolute confidence interval width that is yielded by each procedure.

Fig. 6 shows results for the BLB, the bootstrap and the b out of n bootstrap on the University of California at Irvine connect4 data set (Frank and Asuncion, 2010), where the model is logistic regression (as in the classification setting of our simulation study above), $d = 42$ and $n = 67557$. We select the BLB tuning parameters r and s by using the adaptive method that was described in the preceding section. Notably, the outputs of the BLB for all values of b considered, and the output of the bootstrap, are tightly clustered around the same value; additionally, as expected, the BLB converges more quickly than the bootstrap. However, the values that are produced by the b out of n bootstrap vary significantly as b changes, thus further highlighting this procedure's lack of robustness. We have obtained qualitatively similar results on six additional data sets from the University of California of Irvine data set repository (ct-slice, magic, millionsong, parkinsons, poker and shuttle) (Frank and Asuncion, 2010) with different estimators (linear regression and logistic regression) and a range of values of n and d (see the on-line supplementary materials for plots of these results).

8. Time series

Although we have focused thus far on the setting of IID data, variants of the bootstrap—such as the moving block bootstrap and the stationary bootstrap—have been proposed to handle other data analysis settings such as that of time series (Efron and Tibshirani, 1993; Hall and Mammen, 1994; Kunsch, 1989; Liu and Singh, 1992; Politis and Romano, 1994). These bootstrap variants can be used within the BLB, in computing the requisite plug-in approximations $\xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\}$, to obtain variants of our procedure which are applicable in non-IID data settings. The advantages (e.g. with respect to scalability) of such BLB variants over variants of the bootstrap (and its

Table 4. Comparison of the standard and stationary bootstrap and the BLB on stationary time series data with $n = 5000^\dagger$

<i>Method</i>	<i>Results for standard method</i>	<i>Results for stationary method</i>
BLB-0.6	2.2 ± 0.1	4.2 ± 0.1
BLB-0.7	2.2 ± 0.04	4.5 ± 0.1
BLB-0.8	2.2 ± 0.1	4.6 ± 0.2
BLB-0.9	2.2 ± 0.1	4.6 ± 0.1
Bootstrap	2.2 ± 0.1	4.6 ± 0.2

† We report the average and standard deviation of estimates (after convergence) of the standard deviation of the rescaled mean aggregated over 10 trials. The true population value of the standard deviation of the rescaled mean is approximately 5.

relatives) remain identical to the advantages that were discussed above in the context of large-scale IID data. We briefly demonstrate the extensibility of the BLB by combining our procedure with the stationary bootstrap (Politis and Romano, 1994) to obtain a ‘stationary BLB’ which is suitable for assessing the quality of estimators applied to large-scale stationary time series data.

To extend the BLB in this manner, we must simply alter both the subsample selection mechanism and the resample generation mechanism such that both of these processes respect the underlying data-generating process. In particular, for stationary time series data it suffices to select each subsample as a (uniformly) randomly positioned block of length b within the observed time series of length n . Given a subsample of size b , we generate each resample by applying the stationary bootstrap to the subsample to obtain a series of length n . That is, given $p \in [0, 1]$ (a tuning parameter of the stationary bootstrap), we first select uniformly at random a data point in the subsample series and then repeat the following process until we have amassed a new series of length n : with probability $1 - p$ we append to our resample the next point in the subsample series (wrapping around to the beginning if we reach the end of the subsample series), and with probability p we (uniformly at random) select and append a new point in the subsample series. Given subsamples and resamples that are generated in this manner, we execute the remainder of the BLB procedure as described in algorithm 1.

We now present simulation results comparing the performance of the bootstrap, the BLB, the stationary bootstrap and the stationary BLB. In this experiment, which was initially introduced by Politis and Romano (1994), we generate observed data consisting of a stationary time series $X_1, \dots, X_n \in \mathbb{R}$ where $X_t = Z_t + Z_{t-1} + Z_{t-2} + Z_{t-3} + Z_{t-4}$ and the Z_t are drawn independently from a normal(0, 1) distribution. We consider the task of estimating the standard deviation of the rescaled mean $\sum_{t=1}^n X_t / \sqrt{n}$, which is approximately 5; we set $p = 0.1$ for the stationary bootstrap and the stationary BLB. The results in Table 4 (for $n = 5000$) show the improvement of the stationary bootstrap over the bootstrap, the similar improvement of the stationary BLB over the BLB, and the fact that the statistical performance of the stationary BLB is comparable with that of the stationary bootstrap for $b \geq n^{0.7}$. Note that this exploration of the stationary BLB is intended as a proof of concept, and additional investigation would help to elucidate further and perhaps to improve the performance characteristics of this extension of the BLB.

9. Conclusions

Our results have suggested that the BLB can provide an automatic, accurate means of assessing estimator quality that is well aligned with modern parallel and distributed computing architectures and is scalable to very large data sets. Nonetheless, the BLB has some limitations, and in this section we review some of these limitations, present potential remedies and discuss some avenues of possible future work.

First, recall that the full computational benefits of the BLB are only realized when applying the procedure to an estimator which scales computationally in the number of distinct data points that are presented to it (i.e. in the number of distinct atoms in the empirical distribution of the observed data). However, even when the estimator under consideration does not satisfy this criterion, the BLB continues to provide substantial benefits with respect to storage and network transmission of resamples; its space requirements are of order $O(b)$ rather than $O(n)$. This storage benefit in turn allows the BLB to continue to yield some computational advantage relative to the bootstrap due to the properties of typical computer architectures: smaller BLB resamples can potentially be stored in main memory, which can be accessed far more quickly than hard discs (on which larger bootstrap resamples exceeding the capacity of the main memory would be stored).

We have focused our development of the BLB on the setting of IID data, noting that the procedure can also be applied, with some modification, to data exhibiting appropriate stationarity (e.g. stationary time series). However, particularly when working with large data sets, assumptions regarding independence and stationarity of the observed data may not be valid. In such cases, the BLB as presented above would require that the observed data be preprocessed (e.g. via detrending) to yield a data set which is, for example, stationary. Another potential alternative would be to alter the BLB to obtain a variant which is directly applicable to non-stationary data. For instance, one might assume that BLB subsets can be chosen so that the data within them are stationary. However, after bootstrapping each subset as prescribed by the BLB, the results could not be combined via simple averaging due to intersubset non-stationarity; rather, a more complex procedure would be required, as discussed in the related work of Lahiri *et al.* (2012).

Note also that, although the BLB allows bootstrap methodology to be scaled to much larger data sets than the classical bootstrap, it is important to emphasize that for data sets in the petabyte or exabyte range the ‘small subsets’ that are used by the BLB may themselves be overly large, and extensions of the BLB approach will be required for scalability to such regimes.

Finally, it is worth noting that, although the BLB shares the statistical strengths of the bootstrap, we conversely do not expect our procedure to be applicable in cases in which the bootstrap fails (Bickel *et al.*, 1997). It would presumably be possible to extend the applicability of the BLB to such edge cases by modifying it to apply the m out of n bootstrap or subsampling, rather than the bootstrap, to each subsample (e.g. thus yielding the ‘bag of little m out of n bootstraps’). Although the computational advantages of such a BLB variant would be less pronounced than in settings in which the bootstrap is consistent, this approach could still provide useful computational gains, particularly on very large data sets.

Acknowledgements

This material is based on work supported in part by the US Army Research Laboratory and the US Army Research Office under contract (grant) W911NF-11-1-0391, as well as by National Science Foundation award 1122732.

Appendix A

We provide here the key elements of the proofs of the theoretical results that are included in Section 3. See the on-line supplementary materials for full proofs of all the results.

A.1. Proof sketch for theorem 1

Given that \mathcal{F} is a Donsker class, the empirical process $\mathbb{G}_n = n^{1/2}(\mathbb{P}_n - P)$ converges in distribution to the P -Brownian bridge process \mathbb{G}_P as $n \rightarrow \infty$. Additionally, defining $\mathbb{P}_{n,b}^*$ as the empirical distribution of a size n resample from $\mathbb{P}_{n,b}^{(j)}$, theorem 3.6.3 of van der Vaart and Wellner (1996) implies that, conditionally on the sequence $\mathbb{P}_{n,b}^{(j)}$, the sequence of processes $\mathbb{G}_{n,b}^* = n^{1/2}(\mathbb{P}_{n,b}^* - \mathbb{P}_{n,b}^{(j)})$ also converges in distribution to \mathbb{G}_P , in probability, as $b, n \rightarrow \infty$. Now, let R be the random element to which $n^{1/2}\{\phi(\mathbb{P}_n) - \phi(P)\}$ converges in distribution, as given by the functional delta method (van der Vaart, 1998). The delta method for the bootstrap (see theorem 23.9 of van der Vaart (1998)) then implies that $n^{1/2}\{\phi(\mathbb{P}_{n,b}^*) - \phi(\mathbb{P}_{n,b}^{(j)})\}$ also converges conditionally in distribution to R , given $\mathbb{P}_{n,b}^{(j)}$, in probability. As a result, given the assumed continuity of ξ , it follows that $\xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\}$ and $\xi\{Q_n(P)\}$ have the same asymptotic limit, in probability, for any j . The continuous mapping theorem (van der Vaart, 1998) then immediately yields the final result desired. \square

To prove theorem 2, remark 1 and theorem 3, we use the following two supporting lemmas, which follow straightforwardly from the definitions of the p_k and \hat{p}_k in theorem 2 in conjunction with standard properties of V-statistics and U-statistics (see the on-line supplementary materials for proofs of the lemmas).

Lemma 1. Assume that $X_1, \dots, X_b \sim P$ are IID and let $\hat{p}_k(X_1, \dots, X_b)$ be the sample version of p_k based on X_1, \dots, X_b , as defined in theorem 2. Then, assuming that $E\{\hat{p}_k(X_1, \dots, X_b)^2\} < \infty$, $\text{var}\{\hat{p}_k(X_1, \dots, X_b) - p_k\} = \text{var}\{\hat{p}_k(X_1, \dots, X_b)\} = O(1/b)$.

Lemma 2. Assume that $X_1, \dots, X_b \sim P$ are IID and let $\hat{p}_k(X_1, \dots, X_b)$ be the sample version of p_k based on X_1, \dots, X_b , as defined in theorem 2. Then, assuming that $E|\hat{p}_k(X_1, \dots, X_b)| < \infty$, $|E\{\hat{p}_k(X_1, \dots, X_b)\} - p_k| = O(1/b)$.

A.2. Proof of theorem 2

Summing expansion (3) over j and subtracting expansion (2), we find that

$$\left| s^{-1} \sum_{j=1}^s \xi\{Q_n(\mathbb{P}_{n,b}^{(j)})\} - \xi\{Q_n(P)\} \right| \leq n^{-1/2} \left| s^{-1} \sum_{j=1}^s \hat{p}_1^{(j)} - p_1 \right| + n^{-1} \left| s^{-1} \sum_{j=1}^s \hat{p}_2^{(j)} - p_2 \right| + o_P\left(\frac{1}{n}\right). \quad (4)$$

We now further analyse the first two terms on the right-hand side of expression (4); for the remainder of this proof, we assume that $k \in \{1, 2\}$. Observe that, for fixed k , the $\hat{p}_k^{(j)}$ are conditionally IID given X_1, \dots, X_n for all j , and so

$$\text{var}\left\{s^{-1} \sum_{j=1}^s (\hat{p}_k^{(j)} - p_k) - E(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n) | \mathbb{P}_n\right\} = \frac{\text{var}(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n)}{s},$$

where we denote by $E(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n)$ and $\text{var}(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n)$ the expectation and variance of $\hat{p}_k^{(1)} - p_k$ over realizations of $\mathbb{P}_{n,b}^{(1)}$ conditionally on X_1, \dots, X_n . Now, given that $\hat{p}_k^{(j)}$ is a permutation symmetric function of size b subsets of X_1, \dots, X_n , $E(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n)$ is a U-statistic of order b . Hence, we can apply corollary 3.2(i) of Shao (2003) in conjunction with lemma 1 to find that

$$\text{var}\{E(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n) - E(\hat{p}_k^{(1)} - p_k)\} = \text{var}\{E(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n)\} \leq \frac{b}{n} \text{var}(\hat{p}_k^{(1)} - p_k) = O\left(\frac{1}{n}\right).$$

From the result of lemma 2, we have

$$|E(\hat{p}_k^{(1)} - p_k)| = O\left(\frac{1}{b}\right).$$

Combining the previous three expressions, we find that

$$\left| s^{-1} \sum_{j=1}^s \hat{p}_k^{(j)} - p_k \right| = O_P\left[\frac{\sqrt{\{\text{var}(\hat{p}_k^{(1)} - p_k | \mathbb{P}_n)\}}}{\sqrt{s}}\right] + O_P\left(\frac{1}{\sqrt{n}}\right) + O\left(\frac{1}{b}\right).$$

Finally, plugging into equation (4) with $k = 1$ and $k = 2$, we obtain the desired result. \square

The proofs of remark 1 and theorem 3 follow via similar arguments.

References

- Bickel, P. J. and Freedman, D. A. (1981) Some asymptotic theory for the bootstrap. *Ann. Statist.*, **9**, 1196–1217.
- Bickel, P. J., Gotze, F. and van Zwet, W. (1997) Resampling fewer than n observations: gains, losses, and remedies for losses. *Statist. Sin.*, **7**, 1–31.
- Bickel, P. J. and Sakov, A. (2002) Extrapolation and the bootstrap. *Sankhya A*, **64**, 640–652.
- Bickel, P. J. and Sakov, A. (2008) On the choice of m in the m out of n bootstrap and confidence bounds for extrema. *Statist. Sin.*, **18**, 967–985.
- Bickel, P. J. and Yahav, J. A. (1988) Richardson extrapolation and the bootstrap. *J. Am. Statist. Ass.*, **83**, 387–393.
- Diaconis, P. and Efron, B. (1983) Computer-intensive methods in statistics. *Scient. Am.*, **248**, 96–108.
- Efron, B. (1979) Bootstrap methods: another look at the jackknife. *Ann. Statist.*, **7**, 1–26.
- Efron, B. (1988) More efficient bootstrap computations. *J. Am. Statist. Ass.*, **85**, 79–89.
- Efron, B. and Tibshirani, R. (1993) *An Introduction to the Bootstrap*. New York: Chapman and Hall.
- Frank, A. and Asuncion, A. (2010) UCI machine learning repository. University of California, Irvine. (Available from <http://archive.ics.uci.edu/ml>.)
- Giné, E. and Zinn, J. (1990) Bootstrapping general empirical measures. *Ann. Probab.*, **18**, 851–869.
- Hahn, J. (1995) Bootstrapping quantile regression estimators. *Econometr. Theor.*, **11**, 105–121.
- Hall, P. (1992) *The Bootstrap and Edgeworth Expansion*. New York: Springer.
- Hall, P. and Mammen, E. (1994) On general resampling algorithms and their performance in distribution estimation. *Ann. Statist.*, **22**, 2011–2030.
- Kunsch, H. R. (1989) The jackknife and the bootstrap for general stationary observations. *Ann. Statist.*, **17**, 1217–1241.
- Lahiri, S. N., Spiegelman, C., Appiah, J. and Rilett, L. (2012) Gap bootstrap methods for massive data sets with an application to transportation engineering. *Ann. Appl. Statist.*, **6**, 1552–1587.
- Liu, R. Y. and Singh, K. (1992) Moving blocks jackknife and bootstrap capture weak dependence. In *Exploring the Limits of the Bootstrap* (eds R. LePage and L. Billard), pp. 225–248. New York: Wiley.
- Nocedal, J. and Wright, S. J. (2006) *Numerical Optimization*. New York: Springer.
- Politis, D. N. and Romano, J. P. (1994) The stationary bootstrap. *J. Am. Statist. Ass.*, **89**, 1303–1313.
- Politis, D., Romano, J. and Wolf, M. (1999) *Subsampling*. New York: Springer.
- Putter, H. and van Zwet, W. R. (1996) Resampling: consistency of substitution estimators. *Ann. Statist.*, **24**, 2297–2318.
- Samworth, R. (2003) A note on methods of restoring consistency to the bootstrap. *Biometrika*, **90**, 985–990.
- Shao, J. (2003) *Mathematical Statistics*, 2nd edn. New York: Springer.
- Tibshirani, R. (1985) How many bootstraps? *Technical Report*. Department of Statistics, Stanford University, Stanford.
- van der Vaart, A. W. (1998) *Asymptotic Statistics*. Cambridge: Cambridge University Press.
- van der Vaart, A. W. and Wellner, J. A. (1996) *Weak Convergence and Empirical Processes*. New York: Springer.
- Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M. J., Shenker, S. and Stoica, I. (2012) Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. *USENIX Symp. Networked Systems Design and Implementation*.

Supporting information

Additional ‘supporting information’ may be found in the on-line version of this article:

‘A scalable bootstrap for massive data: supplementary material’.