Bootstrap Methods in R and C

Ella Kaye Xenia Miscouridou

December 3, 2015

1 Introduction

The bootstrap (Efron 1979), is a statistical tool which gives measures of accuracy of estimators, and can therefore be used to draw inference about the parameters of the sampling distribution. It is a powerful and widely applicable tool. The bootstrap is at the interface between statistical inference and computation, and it was developed at a time when advances in computing technology allowed this computationally intensive method to be used in practice. Recently, in this era of 'big data', statistical methodology is again being developed alongside developments in computing technology. Parallel and distributed computing architectures now make it possible to extend the bootstrap so that it can be applied to massive datasets, the so-called Bag of Little Bootstraps (Kleiner et al. 2014).

We have developed a package, `BLB`, with functions to implement the bag of little bootstraps. The package can be obtained from GitHub: `https://www.github.com/EllaKaye/BLB`.

2 The Bootstrap

Suppose we observe a sample of n iid realisations $x_1, \dots, x_n \sim P$, for some probability measure P. Let θ be a parameter of the distribution, and $\hat{\theta}_n$ be an estimator of θ. The goal of the bootstrap is to obtain an assessment, ξ, of the quality of the estimator. For example, θ could be the median, and ξ the standard error. To obtain a bootstrap estimate, we proceed as follows:

1. Repeatedly (B times) sample n points with replacement from the original dataset, giving bootstrap replications (resamples) $(x_1^{*(i)}, \dots, x_n^{*(i)}),\ i = 1, \dots, B$
2. Compute $\hat{\theta}_n^{*(i)}$ on each of the B resamples.
3. Compute $\xi^* = \xi(\hat{\theta}_n^{*(1)}, \dots, \hat{\theta}_n^{*(B)})$ as our bootstrap estimate of ξ.

3 Bag of Little Bootstraps

Kleiner et al. (2014) have developed a scalable bootstrap for massive data, known as the Bag of Little Bootstraps (BLB). The scalability comes from breaking the problem down into subsamples that are much smaller than the original dataset, then running a bootstrap algorithm on each. The BLB breaks down the process as follows:

1. Repeatedly (s times) subsample $b(n) < n$ data points *without replacement* from the original dataset of size n.
2. For each of the s subsamples, do the following:
   (a) Repeatedly (r times) resample n data points *with replacement* from the subsample.
   (b) Compute $\hat{\theta}_n^*$ on each resample.
   (c) Compute an estimate of ξ based on these r realisations of $\hat{\theta}_n^*$.
3. We now have one estimate of ξ per subsample, $\xi_1^*, \ldots, \xi_s^*$. Output their average, ξ^*, as the final estimate of ξ for $\hat{\theta}_n$.

Kleiner et al. (2014) recommend taking $b(n) = n^\gamma$, where $\gamma \in [0.5, 1]$. This procedure dramatically reduces the size of each resample. The BLB lends itself well to parallel and distributed computing in that there are essentially two nested `for` loops; the outer one of the s subsamples and an inner one over the r resamples.

The `BLB` package contains a family of functions that make use of different levels of parallelisation, as well as computer clusters. Use `?BLB` in `R` to see details and examples of these functions.

4 Further work

There is still much work to be done. With the exception of the function `BLB_R`, the functions only take θ, the parameter to be estimated, as the mean. Of course, for these functions to be of any practical use, they need to be able to accept as an argument any function to calculate $\hat{\theta}_n$ for any θ of interest. Similarly, the functions all take the quality assessment, ξ, to be the standard error, but it would be advantageous for the user to be able to specify other choices for ξ, for example as a $(1 - \alpha)$ confidence interval for any chosen $\alpha \in (0, 1)$.

We are aware that there is an obvious optimisation that we could make to our code. Currently at each resampling stage, we resample a vector of length n, the full length of the data set. This is unnecessary when θ is the mean. For each resample, we only need to sample a vector of length b of how many times each of the b elements in the subsample is selected, and then calculate the weighted mean of the subsample with that vector as weights. We are some way towards implementing this improvement.

Moreover, the `BLB` family of functions only accept 1-dimensional data (i.e. a vector) for their input. We are currently working an a multi-dimensional version for running the BLB with θ being the regression coefficients from a d-dimensional linear model. The package currently includes a function, `bootstrap_multidim`, that is designed as a helper function for a multi-dimensional BLB, performing Step 2 in the BLB algorithm. We are still working on incorporating it into a full BLB function.

We hope to have sucessfully implemented at least some of this further work by the time we present tomorrow!

References

Efron, Bradley (1979). “Bootstrap methods: another look at the jackknife”. In: *Annals of Statistics* 7, pp. 1–26.

Kleiner, Ariel et al. (2014). “A scalable bootstrap for massive data”. In: *Journal of the Royal Statistical Society (Series B)* 76.4, pp. 795–816.

Rubin, Donald B (1981). “The Bayesian Bootstrap”. In: *The Annals of Statistics* 9.1, pp. 130–134.