

Simple bootstrap statistical inference using the SAS system

Stephen R. Cole *

*Division of Preventive Medicine, Brigham and Women's Hospital, Harvard Medical School, 900 Commonwealth Avenue East,
Boston, MA 02215, USA*

Received 4 August 1998; received in revised form 9 February 1999; accepted 25 February 1999

Abstract

Nonparametric bootstrap statistical inference is a robust computer intensive method for generating estimates of statistical variability for which formulae are not known or asymptotic assumptions are not met. A SAS macro that implements simple nonparametric bootstrap statistical inference is presented with an example. The program code is easily generalized to any SAS procedure which includes a BY statement, and to cases of clustered data. © 1999 Elsevier Science Ireland Ltd. All rights reserved.

Keywords: Bootstrap; Resampling statistics; SAS macro

1. Introduction

Parametric statistical inference is limited to those statistics that have a known estimate of variation. Such asymptotic formulae may be increasingly faulty as sample size decreases, and provide biased variance estimates when observations are not independent. The nonparametric bootstrap is an alternative computer intensive method to derive robust estimates of sample variation [1]. Bootstrap statistical inference using SAS has been limited by the lack of a formal procedure, or widely distributed program code. Herein,

I provide a SAS program to implement simple nonparametric bootstrap statistical inference, accompanied by an example.

2. Computational method and theory

Following the notation of Mooney and Duval [2], the process to perform simple nonparametric bootstrap statistical inference can be stated in four steps:

1. Generate a resample of the observed data (with replacement), x_b^*
2. Estimate the desired point estimate, $\hat{\theta}_b^*$, and save it to a vector.
3. Repeat steps 1 and 2 B times,

* Tel.: +1-617-2780872; fax: +1-617-7313843.

E-mail address: scole@rics.bwh.harvard.edu (S.R. Cole)

4. Estimate a measure of variation using the vector of resampled point estimates, which is the bootstrapped estimate of the sampling distribution.

In theory, the observed sample is used to create an empirical distribution function, which is repeatedly sampled from to generate B estimates of the desired statistic (e.g. odds ratio, correlation coefficient, etc.). The standard deviation (SD) of these resampled statistics is the empirical standard error (SE) of the statistic generated by the original sample.

$$SE_{\text{BOOT}} = \left[\sum_{b=1}^B (\hat{\theta}_b^* - \theta') / (B - 1) \right]^{1/2} \quad (2.1)$$

where:

$$\theta' = \sum_{b=1}^B \hat{\theta}_b^* / B$$

Efron suggests that in most cases, $B = 200$ – 500 for a SE and $B \geq 1000$ for a confidence interval (CI) will be appropriate [1]. As B increases to infinity, the random error due to the bootstrap process itself decreases towards zero. Indeed, the greatest number of replicates (largest B) that is reasonable is the desirable B and Efron's limits may be viewed as a lower boundary.

To increase the precision of bootstrap estimates, resampling should attempt to replicate conditions under which the data were generated. Further, simple bootstrap resampling may be inadequate for complicated regression equations where the covariates are better treated as fixed, and model residuals are resampled instead of subjects [3]. In addition to replacing the asymptotic SE with the robust bootstrap empirical SE in traditional formulae for CI's, one can generate bootstrap percentile CI's by taking the appropriate percentiles of the sorted vector of resampled estimates. For example with $B = 1000$, a 95% CI would be represented by the 25th and 975th resamples from the vector sorted by increasing size. Further developments, includ-

ing bias corrected CI's [1], are beyond the scope of this paper.

3. Program description

A SAS program has been written to implement this procedure. The program is nested in a macro %DO loop, and may be easily altered for various applications other than the present example.

Step 1. PROC MULTTEST is invoked with the BOOTSTRAP option to generate random replicates with replacement from the data, the number of replicates is controlled by the NSAMPLE = option and each replicate sample is denoted in the output dataset by the variable _sample_. A CLASS statement is required but is simply a nuisance for the current application, so a subject ID is placed on the CLASS statement. A TEST statement is also required and is used to retain variables needed in Step 2. The STRATA option allows one to resample within strata, which may better mimic the data structure.

Step 2. Any SAS procedure that has the BY option may be invoked to compute the desired statistic by _sample_. The resulting point estimates are then output to a dataset.

Step 3. The macro %DO loop repeats adding the point estimates to the dataset using PROC APPEND on each iteration. The total number of replicate samples is the product of the number of macro %DO loop iterations and the number of NSAMPLEs per loop. The optimal combination of %DO and NSAMPLE is application specific. A single iteration of the %DO loop with NSAMPLE very large (say 100 000) is not often feasible as the temporary dataset created by PROC MULTTEST would be contain 10 000 000 observations with only 100 subjects in the original sample.

Step 4. PROC UNIVARIATE is applied to the output dataset to produce the bootstrap empirical SE (2.1), and can be used to produce the bootstrap percentile 95% CI (based on the 2.5 and 97.5% replicates of the statistic).

Table 1
Data from Cole et al. ($N = 76$)

Oligoclonal bands	MS ^a -No. (%)	No. patients				
Normal	6 (16)	38				
Abnormal	16 (42)	38				
Measure of association	Sample estimate	Asymptotic SE	Asymptotic 95% CI	Bootstrap SE ^b	Bootstrap 95% CI ^c	Exact 95% CI ^d
Odds ratio	3.88	1.739	1.31, 11.47	1.834	1.18, 12.74	1.18, 13.85
Risk ratio	2.67	1.522	1.17, 6.08	1.605	1.06, 6.75	NA

^a Multiple sclerosis.

^b 100 000 replicates.

^c Bootstrap 95% CI calculated by replacing naïve asymptotic SE with robust bootstrap SE.

^d Permutation-based exact 95% CI for the odds ratio, not available for the risk ratio.

To automate the procedure, the dataset name, number of macro %DO loop iterations, and NSAMPLE parameters are passed to the macro via the macro call statement. To generalize this SAS code to cases of nonindependent data, one would resample clusters in step 1 and add a DATA step before step 2 to split the data from clusters to single observations.

3.1. Example

Data from Cole et al. [4] are used in the following example (Table 1). Oligoclonal band status, an immunological indicator treated as the exposure variable, was determined to be normal in 38 and abnormal in 38 patients who were followed for a maximum of 5 years or until onset of multiple sclerosis, the outcome. The odds ratio and risk ratio are measures of association for 2×2 tables, defined as:

$$\text{odds ratio} = p_1/(1 - p_1)/p_0/(1 - p_0)$$

$$\text{risk ratio} = p_1/p_0$$

where:

p_i = probability of outcome in group i .

Appendix A provides the SAS code to perform simple nonparametric bootstrap inference on the logarithm of the odds ratio. 100 000 replicates are generated by 20 macro %DO loop iterations with NSAMPLE = 5000. PROC FREQ is used to gen-

erate the OR for each replicate. If any cell frequency in a resample was zero, half was added to all four cells before computation of the odds ratio [5].

The Table provides results from traditional asymptotic, simple nonparametric bootstrap and exact permutation-based statistical inference for the odds ratio and risk ratio. The program provided in the Appendix took 3:41 min using SAS version 6.12 on a SUN ULTRA SPARCstation, and ~ 14 min using the same version of SAS on a 200 MHz Pentium PC with 32 Mb of RAM.

Acknowledgements

The Optic Neuritis Study Group kindly provided these data.

Appendix A. SAS code for bootstrap odds ratio

```
%MACRO resample(data,loop,size);
%DO I = 1%to &loop;
/* Step 1: Generate resamples from data, by
strata */
proc multtest data = &data bootstrap nsample = &size noprint
outsamp = out1; class id; test ca(ms); strata
bands;
/* Step 2: Generate OR by sample, saving to
vector as log(OR) */
```

```

proc freq noprint data = out1; output out =
out1 lgor;
  tables _strata_ *ms /cmh; by _sample_;
data out1 (KEEP = est); set out1; by _sample_
; est = log(_lgor_);
proc append base = out data = out1;
/* Step 3: Repeat macro loop */
%END;
/* Step 4: Generate & Print out SD of resampled
log(OR)s */
proc univariate data = out; var est; title "Boot-
strap S.E. of log(OR)"; run;
%MEND resample;
%resample(one,20,5000);

```

References

- [1] B. Efron, R. Tibshirani, *An introduction to the bootstrap*, Chapman & Hall, New York, 1993.
- [2] C. Mooney, R. Duval, *Bootstrapping: A nonparametric approach to statistical inference*, Sage, Newbury Park, 1993.
- [3] L. Moulton, S. Zeger, *Bootstrapping generalized linear models*, *Comput. Stat. Data Anal.* 11 (1991) 53–63.
- [4] S.R. Cole, R.W. Beck, P. Moke, D. Kaufman, et al., *The value of CSF analysis for predicting the development of multiple sclerosis after optic neuritis: experience of the optic neuritis treatment trial*, *Neurology* 51 (1998) 885–887.
- [5] SAS Institute, *SAS/STAT User's Guide*, SAS Institute Inc., Cary, 1990.