A Bootstrap Example

Charles J. Geyer

October 23, 2003

1 The Problem

Suppose X_1, X_2, $\ldots$ are independent and identically distributed $\text{Gam}(\alpha, \lambda)$ and we want to estimate the shape parameter α. The usual method of moments estimator is

$$\hat{\alpha}_n = \frac{\bar{x}_n^2}{s_n^2}$$

where $\bar{x}_n$ and s_n^2 are the sample mean and variance, respectively.

The delta method gives the asymptotic distribution

$$\hat{\alpha}_n \approx \mathcal{N}\left(\alpha, \frac{2\alpha(1+\alpha)}{n}\right).$$

The variance stabilizing transformation defined by

$$g(\alpha) = \sqrt{2} \cdot \sinh^{-1}(\sqrt{\alpha}),$$

makes $g(\hat{\alpha}_n)$ asymptotically standard normal.

We can make some test data as follows

```
> set.seed(42)
> alpha.true <- 2.2
> n <- 100
> x <- rgamma(n, shape = alpha.true)
```

2 Asymptotic Theory Intervals

A 95% confidence interval for α using the delta method is given by

```
> conf.level <- 0.95
> zcrit <- qnorm((1 + conf.level)/2)
> est <- function(x) mean(x)^2/var(x)
> sdfun <- function(x) {
+     alpha.hat <- est(x)
+     n <- length(x)
+     return(sqrt(2 * alpha.hat * (1 + alpha.hat)/n))
+ }
> foo <- est(x) + c(-1, 1) * zcrit * sdfun(x)
> foo

[1] 1.683309 3.325652

> bar <- foo
```

This can be improved using variance stabilization as follows

```
> varstab <- function(alpha) sqrt(2) * asinh(sqrt(alpha))
> varstabinv <- function(beta) sinh(beta/sqrt(2))^2
> foo <- varstabinv(varstab(est(x)) + c(-1, 1) * zcrit/sqrt(n))
> foo

[1] 1.788911 3.452364

> bar <- rbind(bar, foo)
```

3 Bootstrap Intervals

Bootstrap intervals are presented in (I think) the order of treatment in Efron and Tibshirani.

3.1 Bootstrap-t Intervals

3.1.1 Asymptotic Variance Supplied

Using the function `boott` with the asymptotic variance of the estimator supplied as a function `sdfun` so it can bootstrap an asymptotically pivotal quantity

```
> library(bootstrap)
> perc <- c(1 - conf.level, 1 + conf.level)/2
> sdfun <- function(x, nbootsd, theta, ...) {
+     alpha.hat <- est(x)
+     n <- length(x)
+     return(sqrt(2 * alpha.hat * (1 + alpha.hat)/n))
+ }
> nboot <- 1000
> out.piv <- boott(x, est, perc = perc, sdfun = sdfun, nboott = nboot)
> out.piv$confpoints

        0.025    0.975
[1,] 1.911426 3.116927

> bar <- rbind(bar, out.piv$confpoints)
```

The reason for the `nboott` argument is given in the on-line help for the `boott` command, which says about this argument

> The number of bootstrap samples used to estimate the distribution of the bootstrap T statistic. 200 is a bare minimum and 1000 or more is needed for reliable $\alpha\%$ confidence points, $\alpha > .95$ say.

The reason for the odd call signature `function(x, nbootsd, theta, ...)` for `sdfun` that adds some arguments we don't use is that's what the on-line help for the `boott` command specifies.

3.1.2 Asymptotic Variance Estimated by Double Bootstrap

Just like the preceding except that when `sdfun` is omitted it is calculated by double bootstrap

```
> out <- boott(x, est, perc = perc, nboott = nboot)
> out$confpoints

        0.025   0.975
[1,] 1.832367 3.24807

> bar <- rbind(bar, out$confpoints)
```

3.1.3 Variance-Stabilizing Transformation by Double Bootstrap

An alternative use of the double bootstrap is to estimate an approximate variance-stabilizing transformation

```
> out.vs <- boott(x, est, perc = perc, VS = TRUE, nboott = nboot)
> out.vs$confpoints

        0.025    0.975
[1,] 1.851543 3.078752

> bar <- rbind(bar, out.vs$confpoints)
```

If we want to see the approximate variance-stabilizing transformation, it's in `out.vs`. The R statements

```
> plot(out.vs$theta, out.vs$g, xlab = expression(alpha), ylab = expression(g(alpha)))
```

plot the graph of the function.

3.2 Percentile Intervals

So-called "percentile intervals" are by far the simplest to use. Their endpoints are just quantiles of the distribution of θ^* the bootstrap analog of $\hat{\theta}$. Unlike all the other bootstrap intervals described here, these are *not* second order accurate.

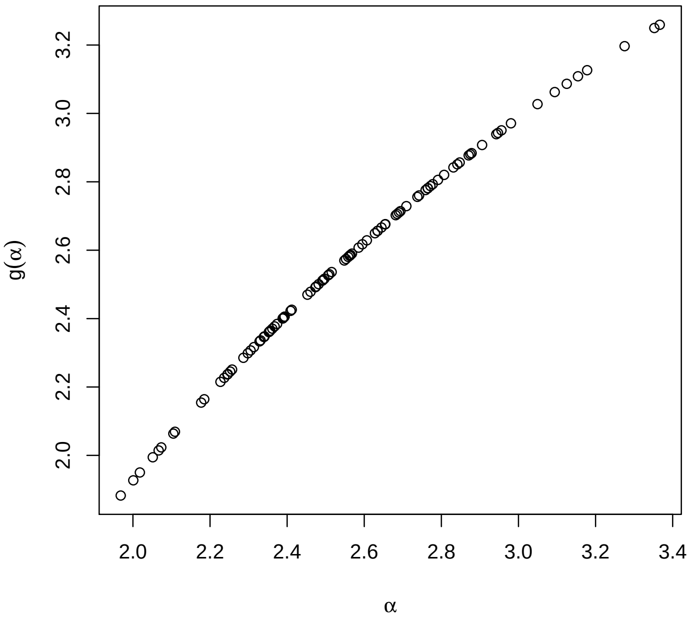

Figure 1: Approximate Variance Stabilizing Transformation Calculated by the Double Bootstrap

```
> out.boot <- bootstrap(x, nboot, est)
> foo <- quantile(out.boot$thetastar, perc)
> foo

    2.5%    97.5%
2.008358 3.245006

> bar <- rbind(bar, foo)
```

3.3 Alphabet Soup Intervals

These are very complicated. See Efron and Tibshirani (1994) for explanations.

3.3.1 BC$_a$

```
> out.bca <- bcanon(x, nboot, est, alpha = perc)
> out.bca$confpoints

     alpha bca point
[1,] 0.025  1.941966
[2,] 0.975  3.192793

> bar <- rbind(bar, out.bca$confpoints[, "bca point"])
```

3.3.2 ABC

This one actually doesn't bootstrap. It calculates an analytic approximation to what the bootstrap should do to be second order accurate. It requires the estimating function be written in "resampling form" as a function with signature `function(p, x)` where `x` is the data and `p` is a probability vector the same length as the data. The idea is that the relationship of a bootstrap sample `x.star` to the original data `x` can be expressed as a probability vector `p.star` such that `p.star[i]` is the fraction of times `x[i]` occurs in `x.star`. In general this is hard, for moments it is straightforward.

```
> estfun <- function(p, x) {
+     mu <- sum(p * x)
+     sigsq <- sum(p * (x - mu)^2) * n/(n - 1)
+     return(mu^2/sigsq)
+ }
> out.abc <- abcnon(x, estfun, alpha = perc)
> out.abc$limits

     alpha      abc     stan
[1,] 0.025 1.956451 1.906997
[2,] 0.975 3.157011 3.101964

> bar <- rbind(bar, out.abc$limits[, "abc"])
```

4 Summary

Having saved all this junk, we can now make a table

```
> dimnames(bar)[[1]] <- c("ordinary asymptotic", "variance-stabilized asymptotic",
+     "bootstrap-$t$ (pivoting)", "bootstrap-$t$ (default)", "bootstrap-$t$ (variance-stabil
+     "percentile", "$\\text{BC}_a$", "ABC")
> bar
```

```
                                       0.025    0.975
ordinary asymptotic                 1.683309 3.325652
variance-stabilized asymptotic      1.788911 3.452364
bootstrap-$t$ (pivoting)            1.911426 3.116927
bootstrap-$t$ (default)             1.832367 3.248070
bootstrap-$t$ (variance-stabilized) 1.851543 3.078752
percentile                          2.008358 3.245006
$\\text{BC}_a$                      1.941966 3.192793
ABC                                 1.956451 3.157011
```

The reason for the funny labels is to use them in LaTeX with the R `xtable` command

	0.025	0.975
ordinary asymptotic	1.68	3.33
variance-stabilized asymptotic	1.79	3.45
bootstrap-t (pivoting)	1.91	3.12
bootstrap-t (default)	1.83	3.25
bootstrap-t (variance-stabilized)	1.85	3.08
percentile	2.01	3.25
BC_a	1.94	3.19
ABC	1.96	3.16