

Cauchy Example for Le Cam Made Simple

Charles J. Geyer

May 30, 2005

1 Introduction

We do a very simple one-parameter model for which everything is trivial, the Cauchy location model. This model is mildly notorious among theoreticians because the likelihood is multimodal and the multimodality does not go away as n goes to infinity (Reeds, 1985). It's not that there is any theory that leads one to expect that multimodality should go away as n goes to infinity, but just that the Cauchy model is simple enough to analyze and in it one can see that indeed what theory doesn't lead one to expect actually doesn't happen.

So let's make up some data. Since the model is translation invariant, it doesn't matter what we choose at the "simulation truth" parameter value

```
> set.seed(42)
> n <- 30
> theta0 <- 0
> x <- theta0 + rcauchy(n)
```

To do maximum likelihood using local optimization we need a good starting point for Newton's method. The obvious starting point for the Cauchy location model is the sample median, easily calculated, highly robust, guaranteed \sqrt{n} -consistent.

Minus the log likelihood (minus because `nlm` does minimization rather than maximization)

```
> mlogl <- function(theta, x) sum(-dcauchy(x, theta, log = TRUE))
```

and the MLE

```
> out <- nlm(mlogl, median(x), typsize = n, hessian = TRUE, x = x)
> print(out)
```

```

$minimum
[1] 77.47065

$estimate
[1] -0.1816497

$gradient
[1] 7.39296e-08

$hessian
      [,1]
[1,] 21.86743

$code
[1] 1

$iterations
[1] 2

```

So our MLE, observed Fisher information (we will pretend we don't know how to calculate expected Fisher information), and asymptotic confidence interval calculated from them are

```

> theta.hat <- out$estimate
> info <- out$hessian
> conf.level <- 0.95
> crit <- qnorm((1 + conf.level)/2)
> theta.hat + crit * c(-1, 1)/sqrt(info)

[1] -0.6007801  0.2374808

```

Couldn't be simpler.

2 Single Bootstrap

By “bootstrap” we mean *parametric bootstrap*, since we are working with maximum likelihood.

```

> nboot <- 200
> save.seed <- .Random.seed
> theta.star <- double(nboot)

```

```

> theta.star.code <- double(nboot)
> info.star <- double(nboot)
> for (iboot in 1:nboot) {
+   x.star <- theta.hat + rcauchy(n)
+   out <- nlm(mlogl, median(x.star), typsize = n, hessian = TRUE,
+     x = x.star)
+   theta.star[iboot] <- out$estimate
+   theta.star.code[iboot] <- out$code
+   info.star[iboot] <- out$hessian
+ }
> all(theta.star.code <= 3)

```

```
[1] TRUE
```

The pivotal quantity used to construct the confidence interval (or, more precisely, its bootstrap approximation) is

```
> z.star <- (theta.star - theta.hat) * sqrt(info.star)
```

Figure 1 shows its Q-Q plot

```

> qqnorm(z.star)
> abline(0, 1)

```

and appears on p. 4. It looks like we are in asymptopia.

Thus there is no point in using the bootstrap distribution of the pivot rather than the normal approximation to calculate the confidence interval.

```

> crit.star <- as.numeric(quantile(z.star, probs = c(0.025, 0.975)))
> theta.hat - rev(crit.star)/sqrt(info)

```

```
[1] -0.5925892  0.2072899
```

```
> theta.hat + crit * c(-1, 1)/sqrt(info)
```

```
[1] -0.6007801  0.2374808
```

Pretty much the same thing either way.

We should note that with this model there is no point to a double bootstrap. Since the model is translation invariant, the single bootstrap sampling distribution is pivotal, and there is no work for the double bootstrap to do.

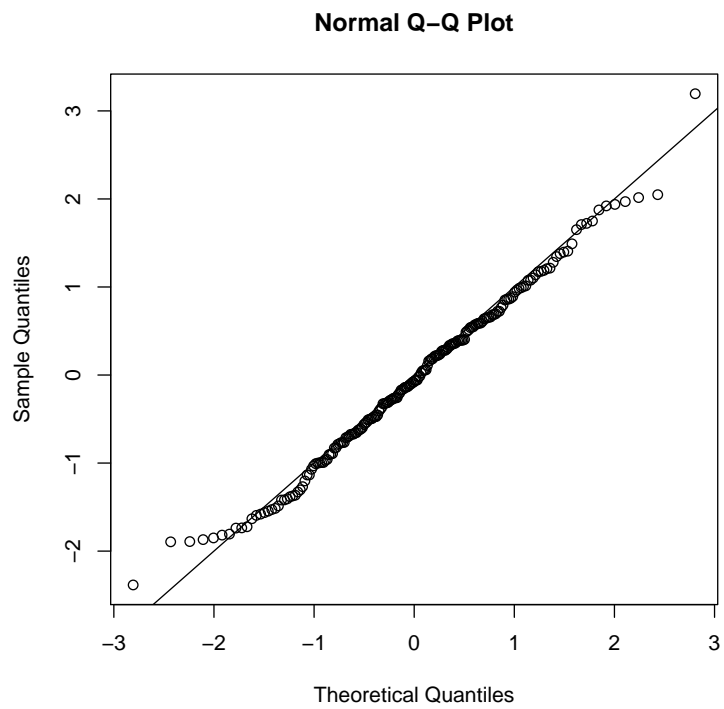


Figure 1: Q-Q Plot of Bootstrap Approximation of Pivotal Quantity. Line has intercept zero and slope one.

3 One-Step Newton

For this section, we'll assume we do know the formula for the Cauchy log likelihood and derivatives

```
> logl <- expression(-log(1 + (x - theta)^2))
> scor <- D(logl, "theta")
> hess <- D(scor, "theta")
> print(scor)

2 * (x - theta)/(1 + (x - theta)^2)

> print(hess)

-(2/(1 + (x - theta)^2) - 2 * (x - theta) * (2 * (x - theta))/(1 +
  (x - theta)^2)^2)

> mloglfun <- function(theta, x) sum(-eval(logl))
> gradientfun <- function(theta, x) sum(-eval(scor))
> hessianfun <- function(theta, x) sum(-eval(hess))
```

So we redo the bootstrap and this time also do one-step Newton

```
> objfun <- function(theta, x) {
+   result <- mloglfun(theta, x)
+   attr(result, "gradient") <- gradientfun(theta, x)
+   return(result)
+ }
> .Random.seed <- save.seed
> theta.star2 <- double(nboot)
> theta.star2.code <- double(nboot)
> theta.start <- double(nboot)
> theta.onestep <- double(nboot)
> for (iboot in 1:nboot) {
+   x.star <- theta.hat + rcauchy(n)
+   theta.start[iboot] <- median(x.star)
+   out <- nlm(objfun, theta.start[iboot], typsize = n, x = x.star)
+   theta.star2[iboot] <- out$estimate
+   theta.star2.code[iboot] <- out$code
+   theta.onestep[iboot] <- theta.start[iboot] - gradientfun(theta.start[iboot],
+     x.star)/hessianfun(theta.start[iboot], x.star)
+ }
> max(abs(theta.star2 - theta.star))
```

```
[1] 1.518016e-05

> all(theta.star2.code <= 3)

[1] TRUE
```

Figure 2 compares the two estimators

```
> err.start <- theta.start - theta.star
> err.onestep <- theta.onestep - theta.star
> plot(err.start, err.onestep, xlab = "error of starting point (sample median)",
+       ylab = "error of one-step Newton estimator")
```

and appears on p. 7. It looks like we are not yet in asymptopia, though perhaps pretty close. For most of the bootstrap samples Newton's method nearly converges in one step (those in the black clump along the horizontal axis). A sizeable fraction are not in the clump, however, even for points not in the clump the one-step error is less than a tenth the starting error (except for 5 points out of 200).

4 Quadraticity

In this section we directly investigate quadraticity over the region containing a certain fraction of the MLE and starting points for Newton's method

```
> alpha <- 0.05
> foo <- quantile(theta.start, probs = c(alpha/4, 1 - alpha/4))
> bar <- quantile(theta.star2, probs = c(alpha/4, 1 - alpha/4))
> baz <- pretty(c(foo, bar))
> print(baz)

[1] -0.8 -0.6 -0.4 -0.2  0.0  0.2  0.4

> .Random.seed <- save.seed
> hessians <- matrix(NA, nboot, length(baz))
> for (iboot in 1:nboot) {
+   x.star <- theta.hat + rcauchy(n)
+   for (j in seq(along = baz)) hessians[iboot, j] <- hessianfun(baz[j],
+     x.star)
+ }
```

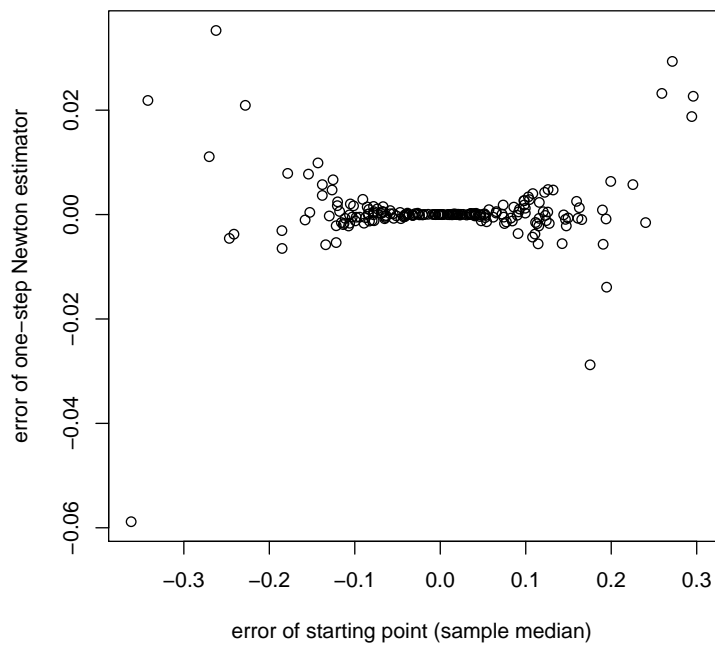


Figure 2: Comparison of error of starting point for Newton’s method (sample median) and error of one-step Newton estimator, where “error” means difference from MLE.

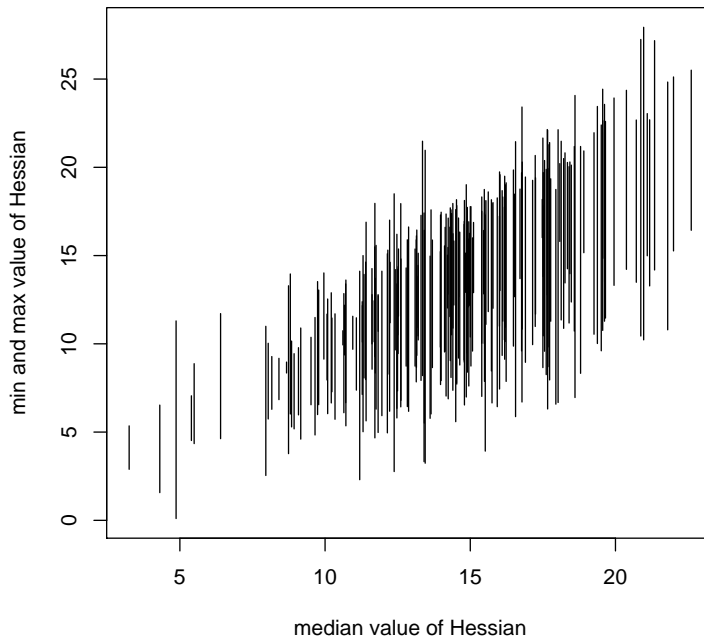


Figure 3: Constancy of the Hessian as a function of the parameter. Lines connect the minimum and maximum value of the Hessian for a particular bootstrap data set at the set of points $-0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4$. The abscissa of the line is the median value of the Hessian at these points.

```
> hess.max <- apply(hessians, 1, max)
> hess.min <- apply(hessians, 1, min)
> hess.med <- apply(hessians, 1, median)
```

Figure 3 compares the two estimators

```
> plot(c(hess.med, hess.med), c(hess.max, hess.min), type = "n",
+      xlab = "median value of Hessian", ylab = "min and max value of Hessian")
> segments(hess.med, hess.min, hess.med, hess.max)
```

and appears on p. 8. This plot looks like we are really not yet in asymptopia. There is a lot of variability of the Hessian over the range.

5 Discussion

One can experiment with this simulation in various ways.

- One can increase the sample size n set in the first code chunk and rerun. We have done this and indeed Figures 2 and Figure 3 do indeed look more “in asymptopia” as n increases.
- One can increase the dimension of the parameter space, going to a Cauchy location-scale model. (Exercise: What is a good starting point for Newton’s method in this case? The median is still a good estimate of location. What scale estimate does one use?)
- One can move outside the class of location-scale families to a family for which the single bootstrap is not exact. Then it is possible to investigate
 - How close the distribution of the Hessian is to *invariant in law*.
 - How necessary (or unnecessary) double bootstrap correction of the single bootstrap is.

This we see that this example is too simple to illustrate all aspects of the “Le Cam Made Simple” view of asymptotics, but the aspects it does illustrate are instructive.

References

REEDS, J. A. (1985). Asymptotic number of roots of Cauchy location likelihood equations. *Ann. Statist.* **13** 775–784.