Stat 8053, Fall 2009: $L_1$ and Quantile Regression


**Breakdown and Robustness**

The finite sample breakdown of an estimator/procedure is the smallest fraction $\alpha$ of data points such that if $[n\alpha]$ points $\to \infty$ then the estimator/procedure also becomes infinite.

**Example.** The sample mean of $x_1, \ldots, x_n$ is

$$
\bar{x}_n = \frac{1}{n} \sum_{i=1}^{n} x_i \\
= \frac{1}{n} \left[ \sum_{i=1}^{n-1} x_i + x_n \right] \\
= \frac{n-1}{n} \bar{x}_{n-1} + \frac{1}{n} x_n
$$

and so if $x_n$ is large enough then $\bar{x}_n$ can be made as large as desired regardless of the other $n-1$ values.

**Example.** Unlike the last example, the sample median, as an estimate of a population median, can tolerate at least 50% bad values. In general, breakdown cannot exceed 50%. (Why is that?)

In general (and in my opinion), very high breakdown estimates should be avoided unless you have faith that the model you are fitting is correct, as the very high breakdown estimates do not allow for diagnosis of model misspecification (Cook, R. D., Hawkins, D. M., and Weisberg, S. (1992) Comparison of model misspecification diagnostics using residuals from OLS and high breakdown estimates. *Journal of the American Statistical Association*, 1992, 87, 419-424).

**Example.** These data are counts of the total number of ridge lines on the fingers of the left and right hands of a random sample of six tribal members in Australia (from Sheather and Staudte (1990), *Robust Estimation and Testing*, Wiley). For all Australians, the mean number is 138, and of interest is if these represent a sample with a larger mean.

```r
> total <- c(166, 229, 142, 141, 136, 153)
> sort(total)
[1] 136 141 142 153 166 229
> summary(total)

   Min. 1st Qu.  Median   Mean 3rd Qu.   Max. 
   136    141     148    161    163    229

> t.test(total, alternative = "greater", mu = 138)
```
One Sample t-test

data: total
t = 1.625, df = 5, p-value = 0.08257
alternative hypothesis: true mean is greater than 138
95 percent confidence interval:
  132.4    Inf
sample estimates:
  mean of x
    161.2

In the graph below, the second value of total is replaced with total[2]+x for x between −100 and 50. For each new “sample” both the t-test, and a bootstrap test are computed, and the resulting p-values are displayed. The code for generating the plot is included here.

```r
> offset <- c(0, 1, 0, 0, 0, 0)
> p.vals <- boot.p.values <- NULL
> boot.p <- function(x, B = 399) {
+     mysample <- function(x) {
+         x1 <- sample(x, replace = TRUE)
+         if (sd(x1) > 0)
+             (mean(x1) - 138)/sd(x1)
+         else (mysample(x))
+     }
+     stat <- (mean(x) - 138)/sd(x)
+     stats <- NULL
+     for (j in 1:B) {
+         stats <- c(stats, mysample(x))
+     }
+     sum(stat >= stats)/(B + 1)
+ }
> for (inc in -100:50) {
+     p.vals <- c(p.vals, t.test(total + inc * offset, alternative = "greater",
+         mu = 138)$p.value)
+ }
> for (inc in -100:50) {
+     boot.p.values <- c(boot.p.values, boot.p(total + inc * offset,
+         B = 99))
+ }
> plot(-100:50 + total[2], p.vals, type = "l", xlab = "Replaced value for X[2]",
+         ylab = "p-values", ylim = c(0, 0.65))
> lines(-100:50 + total[2], boot.p.values, lty = 2)
> library(splines)
> lines(-100:50 + total[2], predict(lm(boot.p.values ~ bs(-100:50 +
+    total[2], df = 6))), col = "red")
```
Interestingly enough, if the largest value is decreased from its observed value, the $p$-value of the one-sided $t$-test actually decreases at first before finally increasing. As it increases and becomes more outlying, the $p$-value increases. Can you explain this?

Sample and population quantiles

Given an distribution $F$, for any $0 < \tau < 1$ we define the $\tau$-th quantile to be the solution to

$$\xi_{\tau}(x) = F^{-1}(\tau) = \inf\{x : F(x) \geq \tau\}$$

Sample quantiles $\hat{\xi}_{\tau}(x)$ are similarly defined, with the sample CDF $\hat{F}$ replacing $F$.

$L_1$ Regression

We start by assuming a model like this:

$$y_i = x'_i \beta + e$$

where the $e$ are random variables. We will estimate $\beta$ by solving the minimization problem

$$\tilde{\beta} = \text{arg min} \frac{1}{n} \sum_{i=1}^{n} |y_i - x'_i \beta| = \frac{1}{n} \sum_{i=1}^{n} \rho_{\frac{1}{n}}(y_i - x'_i \beta)$$

(2)
where $\rho_\tau(u)$ is called a check function,

$$\rho_\tau(u) = u \times (\tau - I(u < 0)) \quad (3)$$

where $I$ is the indicator function (more on check functions later in this handout). If the $e$ are iid from a double exponential distribution, then $\hat{\beta}$ will be the corresponding mle for $\beta$. In general, however, we will be estimating the median at $x'_i \beta$, so one can think of this as median regression.

**Example** We begin with a simple simulated example with $n_1$ “good” observations and $n_2$ “bad” ones.

```r
> set.seed(10131986)
> library(MASS)
> library(quantreg)
> l1.data <- function(n1 = 100, n2 = 20) {
+     data <- mvrnorm(n = n1, mu = c(0, 0), Sigma = matrix(c(1,
+         0.9, 0.9, 1), ncol = 2))
+     data <- rbind(data, mvrnorm(n = n2, mu = c(1.5, -1.5), Sigma = 0.2 *
+         diag(c(1, 1))))
+     data <- data.frame(data)
+     names(data) <- c("X", "Y")
+     ind <- c(rep(1, n1), rep(2, n2))
+     plot(Y ~ X, data, pch = c("x", "o")[ind], col = c("black",
+         "red") [ind], main = paste("N1 =", n1, " N2 =", n2))
+     summary(r1 <- rq(Y ~ X, data = data, tau = 0.5))
+     abline(r1)
+     abline(lm(Y ~ X, data), lty = 2, col = "red")
+     abline(lm(Y ~ X, data, subset = 1:n1), lty = 1, col = "blue")
+     legend("topleft", c("L1", "ols", "ols on good"), lty = c(1,
+         2, 1), col = c("black", "red", "blue"), cex = 0.9)
+ }
> par(mfrow = c(2, 2))
> l1.data(100, 20)
> l1.data(100, 30)
> l1.data(100, 75)
> l1.data(100, 100)
```
Comparing $L_1$ and $L_2$

$L_1$ minimizes the sum of the absolute errors while $L_2$ minimizes squared errors. $L_1$ gives much less weight to large deviations:

```r
> curve(abs(x), -2, 2, ylab = "L1 or L2 or Huber M evaluated at x")
> curve(x^2, -3, 3, add = T, col = "red")
> curve(pmin(x^2, 1.345), add = T, col = "red", lty = 2, lwd = 2)
> abline(h = 0)
> abline(v = 0)
```
**L_1 facts**

1. The L_1 estimator is the mle if the errors are independent with a double-exponential distribution.

2. In (1) if \( x \) consists only of a “1”, then the L_1 estimator is the median.

3. Computations are not nearly as easy as for ls, as a linear programming solution is required.

4. If \( X = (x'_1, \ldots, x'_n)' \) the \( n \times p \) design matrix is of full rank \( p \), then if \( h \) is a set at indexes exactly \( p \) of the rows of \( X \), there is always an \( h \) such that the L_1 estimate \( \hat{\beta} \) fits these \( p \) points exactly, so \( \hat{\beta} = (X'_hX_h)^{-1}X'_hy_h = X_h^{-1}y_h \). Of course the number of potential subsets is large, so this may not help much in the computations.

5. L_1 is equivariant, meaning that replacing \( Y \) by \( a + bY \) and \( X \) by \( A + B^{-1}X \) will leave the solution essentially unchanged.

6. The breakdown points of the L_1 estimate can be shown to be \( 1 - 1/\sqrt{2} \approx 0.29 \), so about 29% “bad” data can be tolerated.

7. In general we are estimating the median of \( y|x \), not the mean.

8. Suppose we have (1) with the errors iid from a distribution \( F \) with density \( f \). The population median is \( \xi = F^{-1}(\tau) \) with \( \tau = 0.5 \), and the sample median is \( \hat{\xi}_5 = F^{-1}(\tau) \). We assume a standardized version of \( f \) so \( f(u) = (1/\sigma)f_0(u/\sigma) \).
Write \( Q_n = n^{-1} \sum x_i x'_i \), and suppose that in large samples \( Q_n \to Q_0 \), a fixed matrix. We will then have:

\[
\sqrt{n}(\beta - \beta) \sim N(0, \omega Q_0^{-1})
\]

where \( \omega = \sigma^2 \tau (1 - \tau) / [f_0(F_0^{-1}(\tau))]^2 \) and \( \tau = 0.50 \). For example, if \( f \) is the standard normal density, \( f(F_0^{-1}(\tau)) = 1/\sqrt{2\pi} = .399 \), and \( \sqrt{\omega} = .5\sigma / .399 = 1.26\sigma \), so in the normal case the standard deviations of the \( L_1 \) estimators are 26\% larger than the standard deviations of the ols estimators.

9. If \( f \) were known, asymptotic Wald inference/confidence statements can be based on percentiles of the normal distribution. In practice, \( f(F^{-1}(\tau)) \) must be estimated. One standard method due to Siddiqui is to estimate

\[
f(F^{-1}(\tau)) = \left[ \hat{F}^{-1}(\tau + h) - \hat{F}^{-1}(\tau - h) \right] / 2h
\]

for some bandwidth parameter \( h \). This is closely related to density estimation, and so the value of \( h \) used in practice is selected using a method appropriate for density estimation.

Alternatively, \( f(F^{-1}(\tau)) \) can be estimated using a bootstrap procedure.

10. For non-iid errors, suppose that \( \xi_i(\tau) \) is the \( \tau \)-quantile for the distribution of the \( i \)-th error. One can show that

\[
\sqrt{n}(\beta - \beta) \sim N(0, \tau (1 - \tau) H^{-1} Q_0 H^{-1})
\]

where the matrix \( H \) is given by

\[
H = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} x_i x'_i f_i \xi_i(\tau)
\]

a sandwich type estimator is used for estimating the variance of \( \beta \). The \texttt{rq} function in \texttt{quantreg} uses a sandwich formula by default.

**Quantile regression**

\( L_1 \) is a special case of quantile regression in which we minimize the \( \tau = .50 \)-quantile, but a similar calculation can be done for any \( 0 < \tau < 1 \). Here is what the check function (??) looks like for \( \tau \in \{.25, .5, .9\} \).

```r
> rho <- function(u) {
+ u * (tau - ifelse(u < 0, 1, 0))
+ }
> tau <- 0.25
> curve(rho, -2, 2, lty = 1)
> tau <- 0.5
> curve(rho, -2, 2, lty = 2, col = "blue", add = T, lwd = 2)
> tau <- 0.9
> curve(rho, -2, 2, lty = 3, col = "red", add = T, lwd = 3)
```
Quantile regression is just like $L_1$ regression with $\rho_\tau$ replacing $\rho_0.5$ in (2), and with $\tau$ replacing 0.5 in the asymptotics.

**Example.** This example shows expenditures on food as a function of income for nineteenth-century Belgian households.

```r
> data(engel)
> plot(foodexp ~ income, engel, cex = 0.5, xlab = "Household Income",
+     ylab = "Food Expenditure")
> abline(rq(foodexp ~ income, data = engel, tau = 0.5), col = "blue")
> taus <- c(0.1, 0.25, 0.75, 0.9)
> for (i in 1:length(taus)) {
+     abline(rq(foodexp ~ income, data = engel, tau = taus[i]),
+            col = "gray")
+ }
```
> plot(summary(rq(foodexp ~ income, data = engel, tau = 2:98/100)))

(The horizontal line is the ols estimate, with the dashed lines for confidence interval for it.)
Second Example This example examines salary as a function of job difficulty for job classes in a large governmental unit. Points are marked according to whether or not the fraction of female employees in the class exceeds 80%.

```R
> library(alr3)
> par(mfrow = c(1, 2))
> mdom <- with(salarygov, NW/NE < 0.8)
> taus <- c(0.1, 0.5, 0.9)
> cols <- c("blue", "red", "blue")
> x <- 100:900
> plot(MaxSalary ~ Score, salarygov, xlim = c(100, 1000), ylim = c(1000, 10000), cex = 0.75, pch = mdom + 1)
> for (i in 1:length(taus)) {
+   lines(x, predict(rq(MaxSalary ~ bs(Score, 5), data = salarygov[mdom, ], tau = taus[i]), newdata = data.frame(Score = x)),
+         col = cols[i], lwd = 2)
+ }
> legend("topleft", paste("Quantile", taus), lty = 1, col = cols,
+         inset = 0.01, cex = 0.8)
> legend("bottomright", c("Female", "Male"), pch = c(1, 2), inset = 0.01,
+        col = cols[i], lwd = 2)
> plot(MaxSalary ~ Score, salarygov[!mdom, ], xlim = c(100, 1000),
+         ylim = c(1000, 10000), cex = 0.75, pch = 1)
> for (i in 1:length(taus)) {
+   lines(x, predict(rq(MaxSalary ~ bs(Score, 5), data = salarygov[mdom, ],
+                      tau = taus[i]), newdata = data.frame(Score = x)),
+         col = cols[i], lwd = 2)
+ }
> legend("topleft", paste("Quantile", taus), lty = 1, col = cols,
+         inset = 0.01, cex = 0.8)
> legend("bottomright", c("Female"), pch = c(1), inset = 0.01,
+        cex = 0.8)
```