LEAST SQUARES
WITH NONNEGATIVE PREDICTED VALUES

JAN DE LEEUW

1. PROBLEM

The problem is to minimize

$$\sigma(b) = \frac{1}{2}(y - Xb)'(y - Xb)$$

over b, under the condition that $Xb \geq 0$.

An important special case is to fit polynomials that are non-negative at the data points (which is different from fitting non-negative polynomials).

2. LAGRANGIAN

The Lagrangian for this problem is

$$\mathscr{L}(b, \lambda) = \frac{1}{2}(y - Xb)'(y - Xb) - \lambda' Xb.$$

The primal problem is

$$\min_b \max_{\lambda \geq 0} \mathscr{L}(b, \lambda).$$

This is the same as the original problem because

$$\max_{\lambda \geq 0} \mathscr{L}(b, \lambda) = \begin{cases} \sigma(b) & \text{if } Xb \geq 0, \\ -\infty & \text{otherwise.} \end{cases}$$

Date: Wednesday 27th November, 2013 — 13h 38min — Typeset in TIMES ROMAN.
Made with knitr and RStudio.

3. DUAL

The dual problem is

$$\max_{\lambda\geq 0}\min_b \mathscr{L}(b,\lambda),$$

and dual objective function is

$$\rho(\lambda)=\min_b \mathscr{L}(b,\lambda).$$

Now

$$\mathscr{L}(b,\lambda)=\frac{1}{2}((y+\lambda)-Xb)'((y+\lambda)-Xb)-\frac{1}{2}(y+\lambda)'(y+\lambda)+\frac{1}{2}y'y,$$

and thus the minimum over b is attained at $\hat{b}=(X'X)^{-1}X'(y+\lambda)$, and

$$\rho(\lambda)=\mathscr{L}(\hat{b},\lambda)=\frac{1}{2}y'y-\frac{1}{2}(y+\lambda)'P(y+\lambda),$$

where

$$P=X(X'X)^{-1}X'.$$

4. SOLVING

Now let $P=QQ'$, where Q is an orthonormal basis for the column space of X, found most efficiently from the QR decomposition $X=QR$ of X. Maximizing ρ over $\lambda\geq 0$ can be done by minimizing $(Q'y+Q'\lambda)'(Q'y+Q'\lambda)$ over $\lambda\geq 0$, which can be done by `nnls` [Mullen and van Stokkum, 2012]. This solves the dual problem. And given λ we can easily compute $b=(X'X)^{-1}X'(y+\lambda)=R^{-1}Q'(y+\lambda)$, which solves the primal problem.

5. WLS

In the case of weighted least squares, with weights in a positive semi-definite matrix W, we simply apply these computations to $\tilde{y}=L'y$ and $\tilde{X}=L'X$, where $W=LL'$ is a Cholesky decomposition of W. If W is diagonal, then of course so is L.

References

Katharine M. Mullen and Ivo H. M. van Stokkum. *nnls: The Lawson-Hanson algorithm for non-negative least squares (NNLS)*, 2012. URL `http://CRAN.R-project.org/package=nnls`. R package version 1.4.

6. Example

6.1. **Numerical.**

```
set.seed (12345)
library(nnls)
x <- cbind(rep(1,6),rchisq(6,1))
y <- -3:2
z <- qr.Q (qr (x))
u <- - colSums (y * z)
print (l <- nnls (t(z), u) $ x)

## [1] 0.000 3.912 0.000 0.000 0.000 0.000

print (b <- qr.solve (x, y + l))

## [1] -5.164e-07  1.426e-01

x %*% b

##             [,1]
## [1,]  8.330e-02
## [2,] -3.454e-17
## [3,]  8.551e-02
## [4,]  6.476e-02
## [5,]  6.181e-01
## [6,]  6.075e-02
```

6.2. **Function.**

```
nnlsPred <- function (x, y, w = NULL) {
  if (is.vector (w)) {
    w <- sqrt (unlist (w))
    x <- w * x
    y <- w * y
  }
  if (is.matrix (w)) {
    w <- chol (w, pivot = TRUE)
    w <- w[, order (attr (w, "pivot"))]
    x <- w %*% x
    y <- colSums (y * w)
  }
  z <- qr (x)
  q <- qr.Q (z)
  u <- - colSums (y * q)
  l <- nnls (t(q), u) $ x
  b <- backsolve (qr.R (z), colSums ((y + l) * q))
  h <- drop (x %*% b)
  return (list (coef = b, pred = h,
                ssq = sum ((y-h) & 2)))
}
```

```
nnlsPred(x, y)

## $coef
## [1] -5.164e-07  1.426e-01
##
## $pred
## [1]  8.330e-02 -7.986e-17  8.551e-02  6.476e-02  6.181e-01  6.075
##
## $ssq
## [1] 6
```

6.3. **Plot.**

```
set.seed(12345)
r <- sort(rchisq (100, 1))
x <- cbind (1,r,r^2)
y <- seq (-3, 5, length = 100)
plot (r, y, ylim = c(-3, 7))
lines (cbind(r, drop (x %*% qr.solve (x, y))),
       lwd = 2, col = "RED")
lines (r, nnlsPred (x, y) $ pred,
       lwd = 2, col = "BLUE")
```

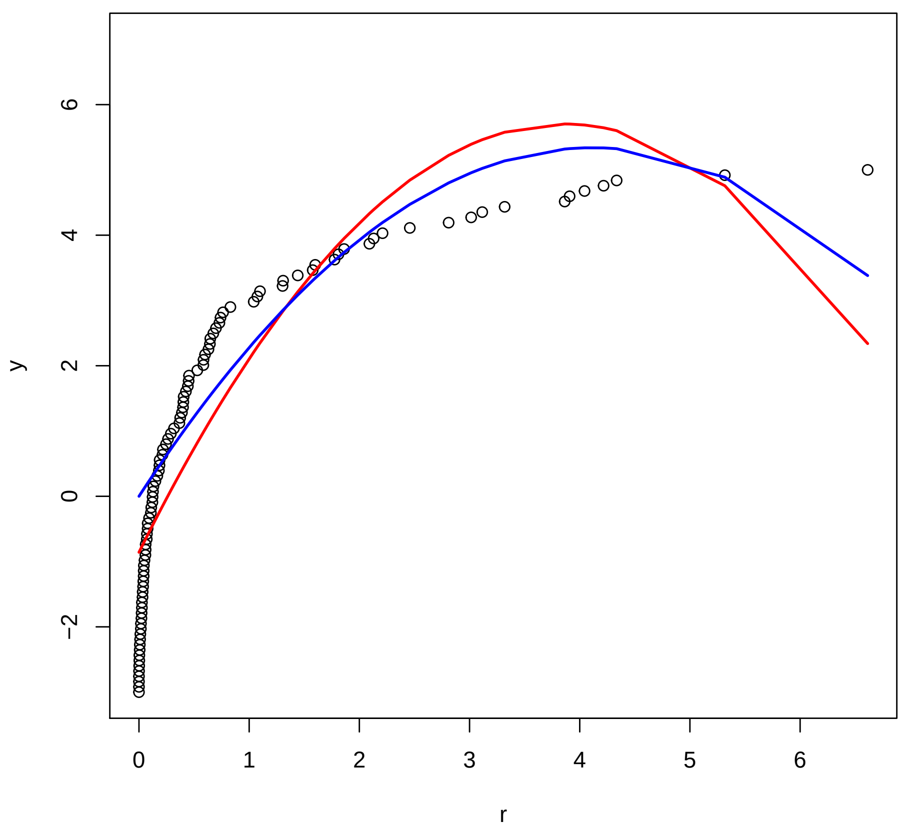

FIGURE 1. LS fit in red, NNLS fit in blue

DEPARTMENT OF STATISTICS, UNIVERSITY OF CALIFORNIA, LOS ANGELES, CA 90095-1554

E-mail address, Jan de Leeuw: deleeuw@stat.ucla.edu

URL, Jan de Leeuw: http://gifi.stat.ucla.edu