# Regression with Linear Inequality Restrictions on Predicted Values

*Jan de Leeuw*

*May 2, 2015*

## Note: the code in this note is now incorporated in the CRAN package isotone.

In this note we study minimization of

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

over all $b$ for which $A'Xb \geq 0$. For instance $X$ can be a spline basis, and $Xb$ is the spline transformation. In situtations like this we are not really interested in the $b$ themselves. We can then use $A$ to require the transformation to be non-negative, or increasing, or satisfying any partial order.

We proceed by formulating and solving the dual problem, and then use the solution of the dual problem to compute the solution to our original problem.

The Lagrangian is

$$\mathcal{L}(b, \lambda) = f(b) - \lambda' A' Xb.$$

The associate primal objective is

$$p(b) := \max_{\lambda \geq 0} \mathcal{L}(b, \lambda) = \begin{cases} f(b) & \text{if } A'Xb \geq 0, \\ +\infty & \text{otherwise.} \end{cases}$$

The dual objective is

$$d(\lambda) := \min_b \mathcal{L}(b, \lambda).$$

The primal problem is to minimize $p$ over $b$, which is the same as minimizing $f$ over $A'Xb \geq 0$, our original problem. The dual problem is to maximize $d$ over $\lambda \geq 0$.

The classical duality theorem for non-degenerate quadratic programs says that

$$\min_b p(b) = \min_b \max_{\lambda \geq 0} \mathcal{L}(b, \lambda) = \max_{\lambda \geq 0} \min_b \mathcal{L}(b, \lambda) = \max_{\lambda \geq 0} d(\lambda).$$

Moreover if $\hat{\lambda}$ solves the dual problem, then $\hat{b} := \operatorname{argmin} \mathcal{L}(b, \hat{\lambda})$ solves the primal problem.

Now analyze the dual problem in more detail.

$$d(\lambda) = \mathcal{L}(X^+(A\lambda + y), \lambda) = \frac{1}{2}y'y - \frac{1}{2}(y + A\lambda)'P_X(y + A\lambda)$$

where $X^+$$ is the Moore-Penrose inverse of $X$, and $P_X := XX^+$.

Find an orthonormal $K$, using singular value or QR decomposition, such that $P_X = KK'$. Define $u := K'y$ and $V := -K'A$. Then maximizing the dual objective is equivalent to minimizing $(u - V\lambda)'(u - V\lambda)$ over $\lambda \geq 0$. If $\hat{\lambda}$ solves the dual problem, then $\hat{b} := X^+(A\hat{\lambda} + y)$ solves the primal problem, so that $X\hat{b} = K(u - V\hat{\lambda})$.

For computation we use the QR decomposition, and solve the dual problem using the `nnls` package [Mullen and Van Stokkum, 2012]. A simple general function is

```
library(nnls)

mregnn <- function (x, y, a) {
  k <- qr.Q(qr(x))
  u <- drop (crossprod(k, y))
  v <- -crossprod (k, t(a))
  lb <- nnls(v, u)$x
  xb <- drop(k %*% (u - v %*% lb))
  return (list(xb = xb,lb = lb,f = sum((y - xb) ^ 2)))
}
```

If we require monotonicity of $z = Xb$, i.e. $z_1 \leq \cdots \leq z_n$, then we can eliminate the matrix multiplication $V = K'A$ and replace it with taking forward differences of the rows of $K$.

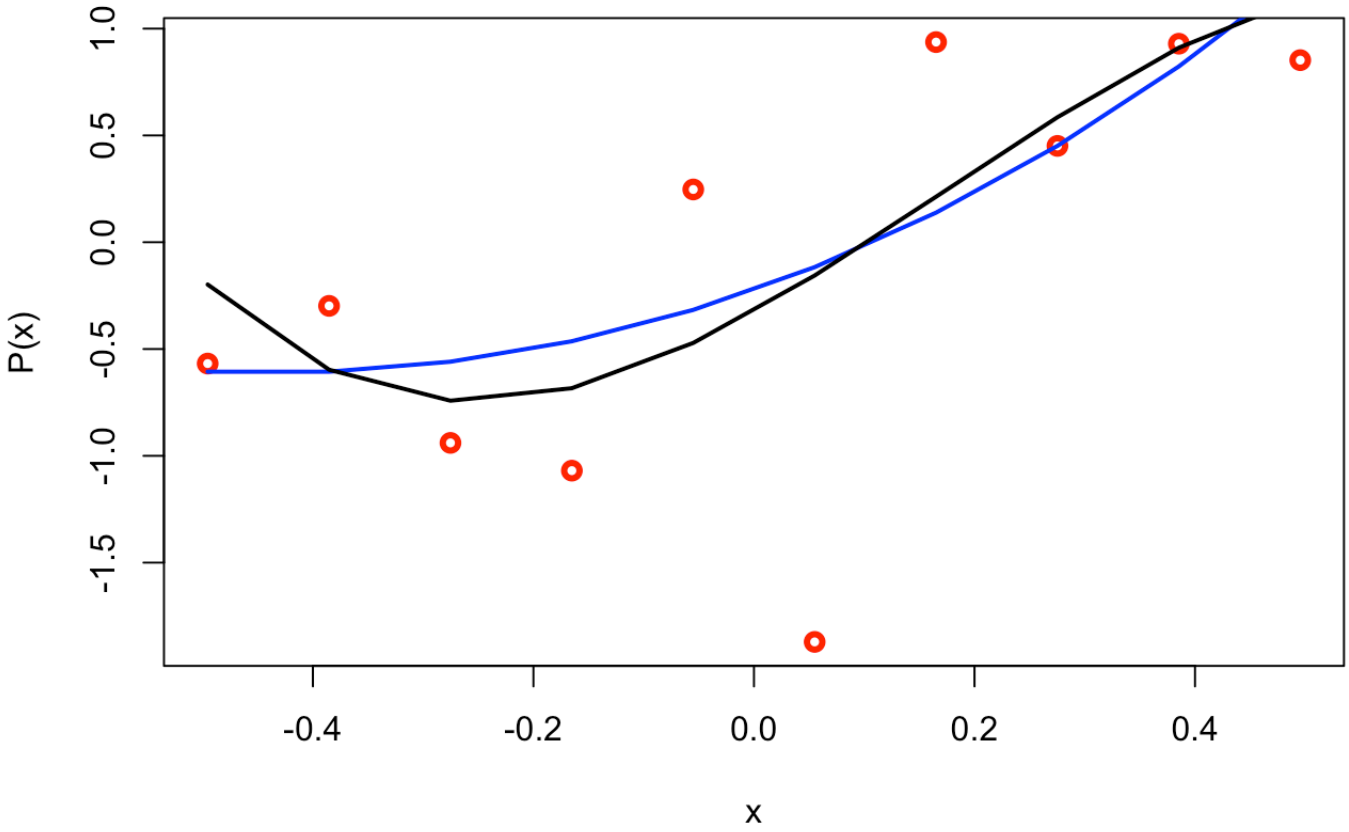
```
mregnnM <- function (x, y) {
  k <- qr.Q(qr(x))
  u <- drop (crossprod(k, y))
  v <- -t(diff(k))
  lb <- nnls(v, u)$x
  xb <- drop(k %*% (u - v%*% lb))
  return (list(xb = xb,lb = lb,f = sum((y - xb) ^ 2)))
}
```

If requiring the elements of $z = Xb$ to be positive, then $A$ is the identity and the function simplifies to

```
mregnnP <- function (x, y) {
  k <- qr.Q(qr(x))
  u <- drop (crossprod(k, y))
  v <- -t(k)
  lb <- nnls(v, u)$x
  xb <- drop(k %*% (u - v%*% lb))
  return (list(xb = xb,lb = lb,f = sum((y - xb) ^ 2)))
}
```

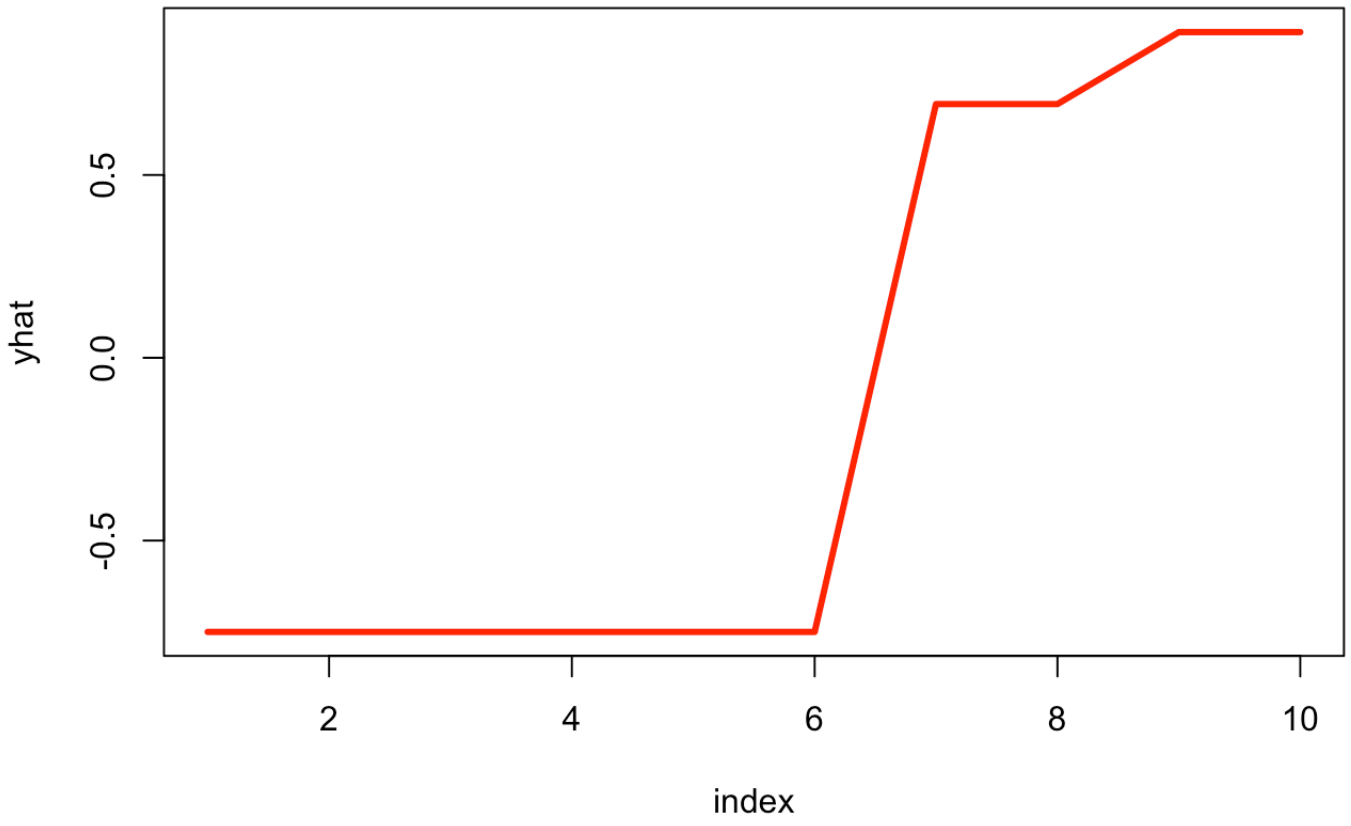
We now generate some points and compute the best fitting quadratic polynomial (in black) and monotone quadratic polynomial (in blue).

```
set.seed(12345)
x <- outer(1:10,1:3,"^")
x <- apply(x,2,function(x)
x - mean(x))
x <- apply (x,2,function(x)
x / sqrt (sum(x ^ 2)))
y <- rowSums(x) + rnorm(10)
plot(x[,1],y,lwd = 3,col = "RED",xlab="x",ylab="P(x)")
o<-mregnnM(x,y)
lines(x[,1],o$xb,col="BLUE", lwd = 2)
xb<-drop(x%*%qr.solve(x,y))
lines(x[,1],xb,col="BLACK", lwd = 2)
```




As a second example, the routines can also be easily used for monotone regression. Just set $X = I$.

```
x<-diag(10)
o<-mregnnM(x,y)
plot(1:10,o$xb,lwd=3,col="RED",type="l",xlab="index",ylab="yhat")
```




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