DUAL CYCLIC COORDINATE DESCENT ALGORITHMS FOR INEQUALITY CONSTRAINED LEAST SQUARES

JAN DE LEEUW

1. Problem

The problem we solve in this note is

$$\min_{\beta}\{\frac{1}{2}(y - X\beta)'W(y - X\beta) \mid AX\beta \geq c\}. \tag{1}$$

We assume that W is positive definite and diagonal, and that X has full column rank. It is convenient to make a change of variables first. Suppose $X = QR$, with $Q'WQ = I$ and R upper triangular. Define $\gamma = R\beta$, $z = Q'Wy$, and $D = AQ$. Then

$$(y - X\beta)'W(y - X\beta) = y'Wy - z'z + (\gamma - z)'(\gamma - z),$$

and Problem (1) can be solved by solving

$$\min_{\gamma}\{\frac{1}{2}(z - \gamma)'(z - \gamma) \mid D\gamma \geq c\}, \tag{2}$$

If $\hat{\gamma}$ is the solution of (2), then $\hat{\beta} = R^{-1}\hat{\gamma}$ solves (1), and $X\hat{\beta} = Q\hat{\gamma}$.

Problem (2) is the *primal problem*. It can also be written as

$$\min_{\gamma}\max_{\lambda\geq 0}\frac{1}{2}(z - \gamma)'(z - \gamma) - \lambda'(D\gamma - c). \tag{3}$$

The *dual problem* is obtained by interchanging the min and the max in (3). Some simple computation gives

$$\max_{\lambda\geq 0}\min_{\gamma}\frac{1}{2}(z - \gamma)'(z - \gamma) - \lambda'(D\gamma - c) = -\min_{\lambda\geq 0}\frac{1}{2}\lambda'DD'\lambda + \lambda'r,$$

Date: February 17, 2007.

2000 *Mathematics Subject Classification.* 62H25.

Key words and phrases. Multivariate Analysis, Correspondence Analysis.

where $r = Dz - c$. By the Fenchel Duality Theorem [Rockafellar, 1970] if $\hat{\lambda} \geq 0$ is a solution of the dual problem, then $\hat{y} = z + D'\hat{\lambda}$ is a solution of the primal problem (2) and thus of the primal problem (1)

2. Algorithm

We give an iterative algorithm of the coordinate descent type [Hildreth, 1957; D'Esopo, 1959; Tseng, 1993]. It solves the dual problem, i.e. minimizes

$$\sigma(\lambda) = \frac{1}{2}\lambda' DD'\lambda + \lambda' r$$

over non-negative λ. It is convenient to define a vector δ with the diagonal elements of DD' and a vector $\tau = D'\lambda$. For row i of D we write d_i.

The algorithm starts with $\lambda = 0$ and then cycles through the coordinates in order. Each coordinate step replaces λ by $\lambda + \theta e_i$, where e_i is one of the unit vectors (with all elements zero, except element i which is one). Now

$$\sigma(\lambda + \theta e_i) = \sigma(\lambda) + \frac{1}{2}\theta^2\delta_i + \theta(d_i'\tau + r_i),$$

which is minimized by choosing

$$\hat{\theta} = \max(-\lambda_i, -\frac{d_i'\tau + r_i}{\delta_i}).$$

Update

$$\hat{\lambda} = \lambda + \hat{\theta}e_i,$$

$$\hat{\tau} = \tau + \hat{\theta}d_i,$$

$$\hat{y} = z + \hat{\tau}.$$

Then go the next coordinate, unless we have completed a cycle, in which case we start with the first coordinate again. And so on, until convergence.

Observe that this is a dual algorithm, which means that only at convergence will we actually have $AQy \geq c$. All the intermediate solutions will be infeasible. If feasibility is critical we may have to iterate to rather high precision.

3. Applications

In the Appendix we give an **R** function for the algorithm in this note. By default the monRegCCA() function does an unwieghted monotone regression, i.e. it chooses $X = W = I$ and $c = 0$, while A takes successive differences. By choosing $c \neq 0$ we can do bounded monotone regression. By choosing only certain rows of the difference matrix we can fit partial orders.

Using the same finite difference matrix A but choosing monomials for X means fitting a monotone polynomial. In the same way choosing a B-spline basis for X, for instance by using the bs() function from the splines package, will fit a monotone spline.

References

D. A. D'Esopo. A Convex Programming Procedure. *Naval Research Logistic Quarterly*, 6:33–42, 1959.

C. Hildreth. A Quadratic Programming Procedure. *Naval Research Logistic Quarterly*, 14(79–85), 1957.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

P. Tseng. Dual Coordinate Ascent Methods for Non-strictly Convex Minimization. *Mathematical Programming*, 59:231–249, 1993.

Appendix A. Code

```
monRegCCA<-function(y,x=diag(length(y)),a=diff(diag(length(y)))
    ,c=rep(0,nrow(a)),w=rep(1,length(y)),eps=1e-15,itmax=100,
    verbose=FALSE,itout=TRUE){
m<-nrow(a); n<-ncol(x)
qr<-qr(sqrt(w)*x); q<-qr.Q(qr)/sqrt(w); r<-qr.R(qr)
z<-colSums(y*w*q); d<-a%*%q; lbd<-rep(0,m); gam<-z
ss<-(sum(w*y^2)-sum(z^2))/2; tau<-rep(0,n); r<-drop(d%*%z)-c
del<-rowSums(d^2); itel<-1; sold<-0
repeat {
    snew<-sold
        for (i in 1:m) {
                di<-d[i,]; deli<-del[i]; lbdi<-lbd[i]
                ri<-r[i]; taudi<-sum(tau*di)
                topt<-max(-lbdi,-(taudi+ri)/deli)
                lbd[i]<-lbdi+topt; tau<-tau+topt*di; gam<-z+tau
                snew<-snew+(deli*topt^2)/2+topt*(taudi+ri)
                if (verbose)
                        cat("Cycle: ",formatC(itel,digits=3,
                            width=3),
                                "Coordinate: ",formatC(i,digits
                                    =3,width=3),
                                "Loss: ",formatC(ss-snew,digits
                                    =6,width=10,format="f"),
                                "\n")
                }
        if (itout)
                cat("Cycle: ",formatC(itel,digits=3,width=3),
                        "***************",
                        "Previous Loss: ",formatC(ss-sold,
                            digits=6,width=10,format="f"),
                        "Current Loss: ",formatC(ss-snew,digits
                            =6,width=10,format="f"),
                        "\n")
        if (((sold-snew) < eps) || (itel == itmax)) break()
        sold<-snew; itel<-itel+1
```

```
        }
return(list(yhat=q%*%gam, s=ss-snew, k=itel))
}
```

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