FITTING SINGLE-PARAMETER CORRELATION STRUCTURES

JAN DE LEEUW

ABSTRACT. Meet the abstract. This is the abstract.

1. PROBLEM

Suppose R is a function that maps the open interval (x_-, x_+) into the set of correlation matrices of order n.

The three examples are the *regular simplex*

$$r_{ij}(x) = x^{|i-j|}, \tag{1a}$$

with $-1 < x < +1$, the *equi-correlation structure*

$$r_{ij}(x) = (1-x)I + xee', \tag{1b}$$

where $-\frac{1}{n-1} < x < +1$, and the *regular circumplex*

$$r_{ij}(x) = x^{f(i,j)}, \tag{1c}$$

where $f(i,j) = \min(|i-j|, n-|i-j|)$ and $-1 < x < +1$.

The loss function we minimize over θ is

$$f(\theta) = \log \mathbf{det}(R(\theta)) + \mathbf{tr}\, R^{-1}(\theta)S, \tag{2}$$

where S is a given positive matrix.

Date: Friday 10th July, 2009 — 8h 15min — Typeset in TIMES ROMAN.
2000 *Mathematics Subject Classification.* 00A00.
Key words and phrases. Binomials, Normals, LaTeX.

1.1. **Scaling the Problem.** As a further elaboration, we will also look at minimizing

$$f(\theta,\Delta) = \log \mathbf{det}(\Delta R(\theta)\Delta) + \mathbf{tr}\ (\Delta R(\theta)\Delta)^{-1}S \tag{3}$$

over θ and Δ, where Δ is either to be diagonal, or scalar, or equal to the identity (in which case we recover the single-parameter problem (2)).

2. Algorithm

2.1. **Block Relaxation.**

2.2. **Fitting.** Using $S(\Delta) = \Delta^{-1}S\Delta^{-1}$ this can also be written as

$$f(\theta,\Delta) = \log \mathbf{det}(R(\theta)) + 2\sum_{i-1}^{n} \log \delta_i + \mathbf{tr}\ R^{-1}(\theta)S(\Delta) \tag{4a}$$

or as

$$f(\theta,\Delta) = \log \mathbf{det}(R(\theta)) + 2\sum_{i-1}^{n} \log \delta_i + \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{r_{ij}(\theta)s_{ij}}{\delta_i\delta_j}. \tag{4b}$$

2.3. **Coordinate Relaxation for Δ.** Replace Δ by $\Delta_i(\varepsilon)$, which is equal to Δ except for element (i,i), which is equal to $\varepsilon\delta_{ii}$. Then

$$f(\theta,\Delta_i(\varepsilon)) = 2\log\varepsilon + \frac{1}{\varepsilon^2}s_{ii}(\Delta) + 2\frac{1}{\varepsilon}\sum_{j\neq i}^{n} r_{ij}(\theta)s_{ij}(\Delta) + g(\Delta,\theta), \tag{5}$$

where g does not depend on ε. Setting the derivative equal to zero gives the quadratic equation

$$\varepsilon^2 - \varepsilon\sum_{j\neq i}^{n} r_{ij}(\theta)s_{ij}(\Delta) - s_{ii}(\Delta) = 0. \tag{6}$$

There are two real roots, and the one we want is

$$\hat{\varepsilon} = \frac{\sum_{j\neq i}^{n} r_{ij}(\theta)s_{ij}(\Delta) + \sqrt{[\sum_{j\neq i}^{n} r_{ij}(\theta)s_{ij}(\Delta)]^2 + 4s_{ii}(\Delta)}}{2}$$

APPENDIX A. CODE

```
simplex<-function(r,n) {
return(r^outer(1:n,1:n,ldist))
}

equiCor<-function(r,n) {
return((1-r)*diag(n)+r)
}

circumplex<-function(r,n) {
nn<-outer(1:n,1:n,function(i,j) abs(i-j))
return(r^pmin(nn,n-nn))
}

fitRegToeplitz<-function(s,func="S") {
n<-nrow(s)
if (func=="S") ffit<-simplex
if (func=="C") ffit<-circumplex
if (func=="E") ffit<-equiCor
ff<-function(r) {
    cf<-ffit(r,n)
    return(det(cf)+sum(diag((solve(cf,s)))))
    }
fop<-optimize(ff,c(-1,1))
return(list(min=fop$minimum,obj=fop$objective,fit=ffit(fop$minimum,n)))
}

scalRegToeplitz<-function(s,itmax=100,eps=1e-6,scale="N",func="S",verbose=TRUE) {
itel<-1; n<-nrow(s); fmin<-log(det(s))+n; fold<-Inf
```

```
if (scale=="D") sold<-rep(1,n)
    else sold<-1
repeat {
    if (scale=="D") ss<-s/outer(sold,sold)
        else ss<-s/(sold^2)
    rr<-fitRegToeplitz(ss,func)$fit
    if (scale=="D") sfit<-rr*outer(sold,sold)
        else sfit<-rr*(sold^2)
    fone<-2*(log(det(sfit))+sum(diag(solve(sfit,s)))-fmin)
    if (scale=="N") {
        snew<-sold
        sfit<-rr
        }
    if (scale=="D") {
        snew<-dScaleCycle(rr,s)
        sfit<-rr*outer(snew,snew)
        }
    if (scale=="S") {
        snew<-sqrt(sum(diag(solve(rr,s)))/n)
        sfit<-rr*(snew^2)
        }
    fnew<-2*(log(det(sfit))+sum(diag(solve(sfit,s)))-fmin)
    if (verbose)
        cat(formatC(fold,digits=6,width=10,format="f"),formatC(fone,digits=6,width=10,format="f"),formatC(fnew,digits=6,width=10,format="f"),"\n")
    if ((max(abs(snew-sold)) < eps) || (itel == itmax)) break()
    itel<-itel+1; sold<-snew; fold<-fnew
    }
return(list(sig=snew,rr=rr,sfit=sfit,fmin=fnew))
```

```
}

dScaleCycle<-function(r,s) {
n<-nrow(r); d<-rep(1,n)
for (i in 1:n) {
	cc<-s[i,i]
	bc<-sum(r[i,]*s[i,])-cc
	sg<-(bc+sqrt((bc^2)+(4*cc)))/2
	s[i,]<-s[i,]/sg
	s[,i]<-s[,i]/sg
	d[i]<-sg*d[i]
	}
return(d)
}
```

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