EXPLORATORY FACTOR ANALYSIS IN R

JAN DE LEEUW

ABSTRACT. Meet the abstract. This is the abstract.

1. INTRODUCTION

Suppose R is a positive definite correlation matrix of order n. In Exploratory Factor Analysis (EFA) we want to find a diagonal matrix Φ such that $R - \Phi$ has rank $r < n$. Thus $R - \Phi$ must have $n - r$ eigenvalues equal to zero or, equivalently, $R^{-\frac{1}{2}}\Phi R^{-\frac{1}{2}}$ must have $n - r$ eigenvalues equal to one.

In a more restrictive formulation we want to find a non-negative diagonal matrix Φ such that $R - \Phi$ is positive semi-definite of rank $r < n$. This is equivalent to requiring that $R^{-\frac{1}{2}}\Phi R^{-\frac{1}{2}}$ is positive semi-definite and has its $n - r$ largest eigenvalues equal to one.

In Swain [1975] a family of techniques for exploratory factor analysis is proposed that is asymptotically equivalent to the multinormal maximum likelihood method.

Minimize

$$F(\phi) = \sum_{k=r+1}^{n} f(\lambda_k(\sum_{i=1}^{n} \phi_i u_i u_i')) \tag{1}$$

The λ_k are the eigenvalues its matrix argument, ordered from large to small. The u_i are the columns of the matrix $R^{-\frac{1}{2}}$.

Date: Wednesday 25th January, 2012 — 23h 25min — Typeset in LUCIDA BRIGHT.
2000 *Mathematics Subject Classification.* 00A00.
Key words and phrases. Binomials, Normals, LaTeX.

In (1) f is any twice. differentiable function with

$$f(1) = 0,$$
$$f'(1) = 0,$$
$$f''(1) = 1,$$

and

$$f'(x) < 0 \text{ if } x < 1,$$
$$f'(x) > 0 \text{ if } x > 1.$$

Or, equivalently,

$$f(x) = \frac{1}{2}(x-1)^2 + o((x-1)^2).$$

Appendix A. Code

```

swain<-function(s,p,fgh,tra=ident,ph=lsFac(s,p),eps=1e-6,
    itmax=30,verbose=TRUE)
{
n<-nrow(s); es<-eigen(s); lb<-es$values; kk<-es$vectors
id<-1:(n-p); nn<-1:n; itel<-1
ss<-kk%*%((1/sqrt(lb))*t(kk))
f<-fgh$f; g<-fgh$g; h<-fgh$h
ft<-tra$f; fi<-tra$i; gt<-tra$g; ht<-tra$h; th<-fi(ph)
repeat{
    vv<-eigen(ss%*%(ph*ss))
    vl<-vv$values; vk<-vv$vectors; vd<-vl[id]
    gh<-gt(th); gk<-ht(th)
    ff<-sum(f(vd)); uu<-ss%*%vk
    gg<-drop((uu[,id]^2)%*%g(vd))*gh
    gm<-max(abs(gg))
    hh<-matrix(0,n,n)
    for (i in id) {
        vli<-vl[i]; ui<-uu[,i]; uw<-outer(ui,ui)
        for (j in nn) {
            vlj<-vl[j]; uj<-uu[,j]
            bij<-ifelse(i==j,h(vli),2*g(vli)/(vli-vlj))
            hh<-hh+bij*uw*outer(uj,uj)
            }
        }
    hh<-(hh*outer(gh,gh))+diag(gh*gk)
    dr<-solve(hh,gg); ch<-max(abs(dr)); dc<-sum(gg*dr)
    hv<-(eigen(hh,only.values=TRUE)$values)[n]
    if (verbose)
           cat("itel",formatC(itel,format="d",width=4),
               " function",formatC(ff,format="f",digits=8,
                 width=12),
               " maxgrad",formatC(gm,format="f",digits=8,
                 width=12),
```

```
                    " change",formatC(ch,format="f",digits=8,width=12),
                    " descend",formatC(dc,format="f",digits=8,width=12),
                    " minval",formatC(hv,format="f",digits=8,width=12),
                    "\n")
    if ((ch < eps) || (itel == itmax)) break()
    th<-th-dr; ph<-ft(th); itel<-itel+1
    }
return(list(ph=ph,ff=ff,hv-hv))
}


guttmanUpper<-function(x) 1/diag(solve(x))

lsFac<-function(s,p,eps=1e-6,itmax=10) {
itel<-1; pp<-1:p; n<-nrow(s); uold<-guttmanUpper(s)
repeat{
    eig<-eigen(s-diag(uold)); evl<-eig$values; evc<-eig$vectors
    unew<-diag(s-evc[,pp]%*%(evl[pp]*t(evc[,pp])))
    if ((max(abs(unew-uold)) < eps) || (itel == itmax)) break()
    uold<-unew; itel<-itel+1
    }
return(unew)
}

# loss function specifics

maxlik<-list(f=function(x) log(x)+(1/x)-1,
             g=function(x) (x-1)/(x^2),
             h=function(x) (2-x)/(x^3))

gls<-list(f=function(x) 0.5*(x-1)^2,
```

```
          g=function(x) (x-1),
          h=function(x) 1)

james<-list(f=function(x) 0.5*log(x)^2,
            g=function(x) log(x)/x,
            h=function(x) (1-log(x))/(x^2))

# parameter transformations

logit<-list(f=function(x) 1/(1+exp(-x)),
            i=function(y) log(y/(1-y)),
            g=function(x) {p<-1/(1+exp(-x)); p*(1-p)},
            h=function(x) {p<-1/(1+exp(-x)); p*(1-p)*(1-2*p)})

ident<-list(f=function(x) x,
            i=function(y) y,
            g=function(x) rep(1,length(x)),
            h=function(x) rep(0,length(x)))

square<-list(f=function(x) x^2,
             i=function(y) sqrt(y),
             g=function(x) 2*x,
             h=function(x) rep(2,length(x)))

expo<-list(f=function(x) exp(x),
           i=function(y) log(y),
           g=function(x) exp(x),
           h=function(x) exp(x))



swainNLM<-function(s,p,fgh,info=0,verbose=1) {
n<-nrow(s); es<-eigen(s); lb<-es$values; kk<-es$vectors
id<-1:(n-p); nn<-1:n; itel<-1
ss<-kk%*%((1/sqrt(lb))*t(kk))
```

```
nlm(f=fSwain,p=lsFac(s,p),ss,id,fgh,info,print.level=
    verbose,hessian=TRUE)
}

fSwain <- function(ph, ss, id, fgh, info) {
f<-fgh$f; g<-fgh$g; h<-fgh$h; n<-length(ph); nn<-1:n
vv<-eigen(ss%*%(ph*ss))
vl<-vv$values; vk<-vv$vectors; vd<-vl[id]
ff<-sum(f(vd)); uu<-ss%*%vk
gg<-drop((uu[,id]^2)%*%g(vd))
hh<-matrix(0,n,n)
for (i in id) {
    vli<-vl[i]; ui<-uu[,i]; uw<-outer(ui,ui)
    for (j in nn) {
        vlj<-vl[j]; uj<-uu[,j]
        bij<-ifelse(i==j,h(vli),2*g(vli)/(vli-vlj))
        hh<-hh+bij*uw*outer(uj,uj)
        }
    }
if (info > 0)
        attr(ff, "gradient") <- gg
if (info > 1)
        attr(ff, "hessian") <- hh
return(ff)
}

swain2<-matrix(
        c(1.000,0.624,0.626,0.271,0.400,0.340,0.319,0.496,
          0.624,1.000,0.573,0.285,0.263,0.185,0.340,0.396,
          0.626,0.573,1.000,0.120,0.301,0.296,0.249,0.380,
          0.271,0.285,0.120,1.000,0.157,0.239,0.270,0.253,
          0.400,0.263,0.301,0.157,1.000,0.524,0.582,0.560,
          0.340,0.185,0.296,0.239,0.524,1.000,0.563,0.553,
          0.319,0.340,0.249,0.270,0.582,0.563,1.000,0.651,
          0.496,0.396,0.380,0.253,0.560,0.553,0.651,1.000)
              ,8,8)
```

```
harman23<-Harman23.cor$cov

harman74<-Harman74.cor$cov
```

References

A.J. Swain. A Class of Factor Analysis Estimation Procedures with Common Asymptotic Sampling Properties. *Psychometrika*, 40:315–335, 1975.

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