THE UNIVERSITY OF MINNESOTA

Statistics 5401 November 14, 2005

Example of Generalized Least Squares (GLS) Factor Analysis in MacAnova

Here is MacAnova output illustrating GLS factor extraction. Two methods are used, the iterative method described in the handout on factor extraction as implemented in **stepgls()** and direct minimization of the GLS criterion as implemented in **facanal()**.

The data are the chicken bone data in file **cbbones.txt**.

```
Cmd> bonedata <- read("","bonedata") # read cbbones.txt
bonedata      276      6 format labels
) Bone measurements on n = 276 outbred female chickens, all in mm.
) Col. 1:  skull length
) Col. 2:  skull breadth
) Col. 3:  femur length (leg bone)
) Col. 4:  tibia length (leg bone)
) Col. 5:  humerus length (wing bone)
) Col. 6:  ulna length (wing bone)
Read from file "TP1:Stat5401:Data:cbbones.txt"
Cmd> r <- cor(bonedata) # compute correlation matrix
```

Iteration using stepgls()

stepgls() is used similarly to **stepuls()** and **stepml()**:

```
Cmd> result <- stepgls(r, psi, m)
```

or

```
Cmd> result <- stepgls(r, psi, m, print:T)
```

where **r** is a p by p variance matrix or correlation matrix, **psi** is a <u>vector</u> whose value is the current values of ψ_1, ..., ψ_p, the diagonal elements of $\mathbf{\Psi}$, **m** is the number of factors. With **print:T**, the new value of **psi** and a goodness of fit quantity is printed. Argument **psi** can also be a structure whose first component is a vector containing the diagonal elements ψ_1,...ψ_p of $\mathbf{\Psi}$.

result is a structure with three components, **psi**, **loadings**, and **crit**, where **psi** is the vector of updated uniquenesses ψ_j, **loadings** is the updated matrix of loadings, and **crit** is a goodness of fit criterion (see below). **crit** actually measures the goodness of the fit provided by the argument **psi** with non-updated loadings.

Because **stepgls()** accepts a structure as first argument, you can use **result** in place of **psi** in the next iteration. In fact, a generic step of the iteration is

```
Cmd> psihat <- stepgls(s,paihat,m [,print:T])
```

Besides returning a structure as values, **stepgls()** creates variables **PSI**, **LOADINGS** and **CRITERION** as "side effects". These are identical to components **psi**, **loadings** and **crit** of the returned value.

The goodness of fit criterion **crit** is

$$\mathrm{tr}(\mathbf{I}_p - \mathbf{S}^{-1}\mathbf{\Sigma})^2 = \sum_{m+1 \le j \le p}(1 - 1/\gamma_i)^2$$

where $\gamma_1 \ge \ldots \ge \gamma_p$ are the eigenvalues of $\mathbf{S}$ relative to $\mathbf{\Psi}$. This quantity is reduced on each iteration, and its minimum is achieved at the GLS estimate. The value of $\sum_{m+1 \le j \le p}(1 - 1/\gamma_i)^2$ is the same whether $\mathbf{S}$ is the sample covariance or correlation matrix. At convergence, this can be used as a goodness-of-fit statistic for the m-factor model, since, in large samples, when $\mathbf{\Sigma}$ is of factor analytic form,

$$\{f_e - (2p+5)/6 - 2m/3\} \sum_{m+1 \le j \le p}(1 - 1/\gamma_i)^2$$

is approximately distributed as χ_f^2, where $f = \{(p-m)^2 - p - m\}/2$.

Iteration stops with an error message if it happens that $\gamma_m < 1$. In that case you can retrieve the most recent estimates from **PSI** and **loadings**.

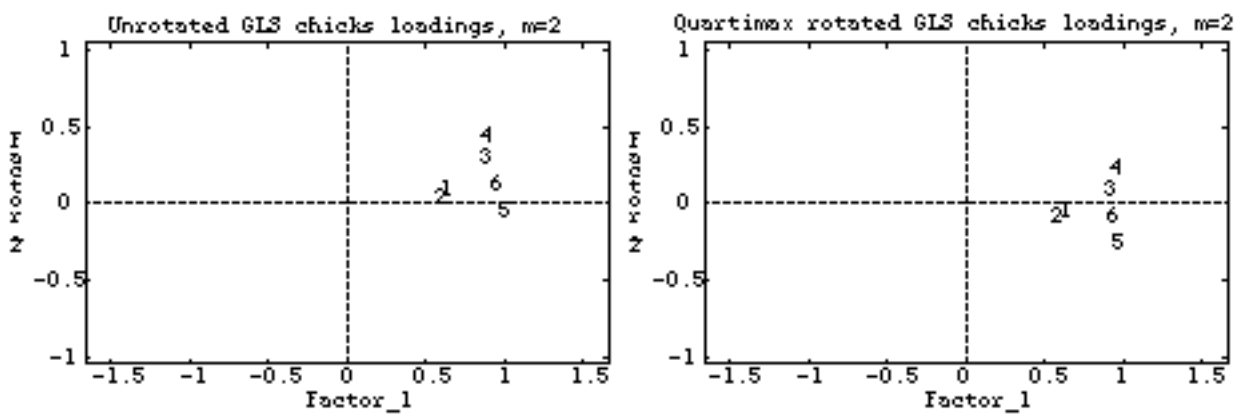
```
Cmd> setoptions(format:"10.6f") # all output with 6 decimals

Cmd> psi0 <- 1/diag(solve(r)); psi0 # compute starting values
(1)   0.528691   0.565077   0.121657   0.109907   0.099535   0.096204

Cmd> psihat <- stepgls(r,psi0,2,print:T) # take one step
psi:                 Printed because of print:T
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.448682   0.446961   0.079338   0.066505   0.056627   0.061822
criterion:
(1)   1.410360     Criterion at starting value

Cmd> iter <- 1 # initialize iteration count

Cmd> n <- 30; for(@i,1,n){ # print on every 5-th iteration
        if(iter %% 5 == 0){ # print every 5th iteration
            print(iter_crit:vector(iter,psihat$crit),psi:psihat$psi)
        }
        iter <- iter+1; psihat <- stepgls(r,psihat,2);;}
iter_crit:     Iter &  crit
(1)   5.000000   0.292092
psi:            Watch psi[5]
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.471961   0.485071   0.100096   0.034898   0.025108   0.078945
iter_crit:     Iter &  crit
(1)  10.000000   0.258046
psi:
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.472880   0.491559   0.113581   0.014548   0.006706   0.089782
iter_crit:     Iter &  crit
(1)  15.000000   0.256316
```

```
psi:
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.473004   0.492874   0.115787   0.010769   0.002529   0.091985
iter_crit:
(1)  20.000000   0.256241
psi:
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.473006   0.493203   0.116013   0.010480   0.001349   0.092525
iter_crit:
(1)  25.000000   0.256224
psi:
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.472988   0.493353   0.115891   0.010814   0.000721   0.092764
iter_crit:
(1)  30.000000   0.256209
psi:
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.472966   0.493470   0.115712   0.011244   0.000202   0.092944

Cmd> # do a few more iterations; looks like psi[5] may hit 0

Cmd> n <- 5;for(@i,1,n){;# print on every iteration
        print(vector(iter,psihat$crit),psihat$psi)
        iter <- iter + 1; psihat <- stepgls(r,psihat,2);;}
iter_crit:                 Iter & crit
(1) 31.000000  0.256206
psi:                       psihat
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.472961   0.493492   0.115675   0.011331   0.000102   0.092978
iter_crit:                 Iter & crit
(1) 32.000000  0.256203
psi:                       psihat
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.472956   0.493514   0.115638   0.011418   0.000003   0.093012
ERROR: non-positive psi; cannot continue in macro stepgls

Cmd> #stepgls aborts as psi[5] goes through 0.

Cmd> print(PSI,LOADINGS,CRITERION)
PSI:           psi at start of last step
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.472956   0.493514   0.115638   0.011418   0.000003   0.093012
LOADINGS:      loadings at start of last step
               (1)       (2)
SklLngth  0.621696  0.123570
SklBrdth  0.583901  0.074885
FemLngth  0.878654  0.332685
TibLngth  0.875638  0.471085
HumLngth  0.999940 -0.004155
UlnLngth  0.937578  0.154590
CRITERION:
(1)  0.256203
```

```
Cmd> # Do varimax rotation of loadings
Cmd> loadings_rot <- rotation(LOADINGS,reorder:F,kaiser:T)
Cmd> loadings_rot
                 (1)        (2)
SklLngth   0.473473   0.421424
SklBrdth   0.465542   0.360304
FemLngth   0.589123   0.731879
TibLngth   0.516425   0.849687
HumLngth   0.864305   0.502866
UlnLngth   0.730132   0.608159
```

It doesn't seem to have simpler structure than unrotated loadings. Try quartimax rotation:

```
Cmd> loadings_rot <- rotation(LOADINGS,kaiser:T,method:"quartimax",\
        reorder:F)
Cmd> loadings_rot
                 (1)        (2)
SklLngth   0.633705  -0.013909
SklBrdth   0.586269  -0.053260
FemLngth   0.929831   0.134638
TibLngth   0.956839   0.270410
HumLngth   0.975342  -0.220469
UlnLngth   0.948813  -0.051989

Cmd> # Now do scatter plots of each set of loadings.
Cmd> plot(Factor_1:LOADINGS[,1],symbols:run(6),\
    Factor_2:LOADINGS[,2],xmin:-1.6,xmax:1.6,ymin:-1,ymax:1,\
    title:"Unrotated GLS chicks loadings, m=2")
Cmd> plot(Factor_1:loadings_rot[,1],symbols:run(6),\
    Factor_2:loadings_rot[,2],xmin:-1.6,xmax:1.6,ymin:-1,ymax:1,\
    title:"Quartimax rotated GLS chicks loadings, m=2")
```




```
Cmd> # The values of xmin and xmax were chosen so that the
Cmd> # horizontal and vertical scales would be about the same
```

Direct minimization of GLS criterion using facanal()

You can use **facanal()** to estimate $\boldsymbol{\Psi}$ and $\mathbf{L}$ by direct minimization of the GLS criterion. If desired, you can specify a rotation method keyword **rotate**, with Kaiser normalization the default.

```
Cmd> result <- facanal(r, 2, method:"gls",rotate:"quartimax")
WARNING: no convergence in 30 iterations
estimated uniquenesses:
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.474700   0.493621   0.116940   0.008253   0.004088   0.091438
quartimax rotated estimated loadings:
            Factor 1   Factor 2
SklLngth    0.634505  -0.016902
SklBrdth    0.586946  -0.053583
FemLngth    0.929232   0.134078
TibLngth    0.956782   0.276245
HumLngth    0.974797  -0.213729
UlnLngth    0.949463  -0.053139
minimized gls criterion:
(1)   0.128172
```

facanal() did not fully converge in the default maximum number of iterations. Redo, allowing for 100 iterations.

```
Cmd> result <- facanal(r, 2, method:"gls",rotate:"quartimax",\
        maxit:100)
Convergence in 47 iterations by criterion 2
estimated uniquenesses:
   SklLngth   SklBrdth   FemLngth   TibLngth   HumLngth   UlnLngth
   0.472930   0.493486   0.115341   0.012007   0.000000   0.092953
quartimax rotated estimated loadings:
            Factor 1   Factor 2
SklLngth    0.633722  -0.013733
SklBrdth    0.586237  -0.053349
FemLngth    0.929958   0.134970
TibLngth    0.956731   0.269540
HumLngth    0.975330  -0.220751
UlnLngth    0.948838  -0.051861
minimized gls criterion:
(1)   0.128098
```

The GLS criterion computed by **facanal()** is $.5\times\mathrm{tr}(\mathbf{I}_p - \mathbf{S}^{-1}\boldsymbol{\Sigma})^2 = .5\times\sum_{m+1\le j\le p}(1 - 1/\gamma_i)^2$, half the criterion used by **stepgls()**.

```
Cmd> 2*result$criterion # make comparable with stepgls() criterion
(1)   0.256196
```

This is slightly smaller than 0.256203, the value reached by **stepgls()** before it quit. The estimated uniquenesses and rotated loadings are very close to those found by **stepgls()**.