NAG Toolbox for MATLAB

g03ca

1 Purpose

g03ca computes the maximum likelihood estimates of the parameters of a factor analysis model. Either the data matrix or a correlation/covariance matrix may be input. Factor loadings, communalities and residual correlations are returned.

2 Syntax

```
[e, stat, com, psi, res, fl, ifail] = g03ca(matrix, weight, n, x, nvar,
isx, nfac, wt, iop, 'm', m)
```

3 Description

Let p variables, $x_1, x_2, \ldots, x_p$, with variance-covariance matrix Σ be observed. The aim of factor analysis is to account for the covariances in these p variables in terms of a smaller number, k, of hypothetical variables, or factors, $f_1, f_2, \ldots, f_k$. These are assumed to be independent and to have unit variance. The relationship between the observed variables and the factors is given by the model:

$$x_i = \sum_{j=1}^{k} \lambda_{ij} f_j + e_i, \qquad i = 1, 2, \ldots, p$$

where λ_{ij}, for $i = 1, 2, \ldots, p$; $j = 1, 2, \ldots, k$, are the factor loadings and e_i, for $i = 1, 2, \ldots, p$, are independent random variables with variances ψ_i, for $i = 1, 2, \ldots, p$. The ψ_i represent the unique component of the variation of each observed variable. The proportion of variation for each variable accounted for by the factors is known as the communality. For this function it is assumed that both the k factors and the e_i's follow independent Normal distributions.

The model for the variance-covariance matrix, Σ, can be written as:

$$\Sigma = \Lambda \Lambda^{\mathrm{T}} + \Psi \tag{1}$$

where Λ is the matrix of the factor loadings, λ_{ij}, and Ψ is a diagonal matrix of unique variances, ψ_i, for $i = 1, 2, \ldots, p$.

The estimation of the parameters of the model, Λ and Ψ, by maximum likelihood is described by Lawley and Maxwell 1971. The log-likelihood is:

$$-\tfrac{1}{2}(n-1)\log(|\Sigma|) - \tfrac{1}{2}(n-1)\mathrm{trace}\left(S, \Sigma^{-1}\right) + \text{constant},$$

where n is the number of observations, S is the sample variance-covariance matrix or, if weights are used, S is the weighted sample variance-covariance matrix and n is the effective number of observations, that is, the sum of the weights. The constant is independent of the parameters of the model. A two stage maximization is employed. It makes use of the function $F(\Psi)$, which is, up to a constant, $-2/(n-1)$ times the log-likelihood maximized over Λ. This is then minimized with respect to Ψ to give the estimates, $\hat{\Psi}$, of Ψ. The function $F(\Psi)$ can be written as:

$$F(\Psi) = \sum_{j=k+1}^{p} \left(\theta_j - \log\theta_j\right) - (p-k)$$

where values θ_j, for $j = 1, 2, \ldots, p$ are the eigenvalues of the matrix:

$$S^* = \Psi^{-1/2} S \Psi^{-1/2}.$$

The estimates $\hat{\Lambda}$, of Λ, are then given by scaling the eigenvectors of S^*, which are denoted by V:

$$\hat{\Lambda} = \Psi^{1/2} V (\Theta - I)^{1/2}.$$

where Θ is the diagonal matrix with elements θ_i, and I is the identity matrix.

The minimization of $F(\Psi)$ is performed using e04lb which uses a modified Newton algorithm. The computation of the Hessian matrix is described by Clark 1970. However, instead of using the eigenvalue decomposition of the matrix S^* as described above, the singular value decomposition of the matrix $R\Psi^{-1/2}$ is used, where R is obtained either from the QR decomposition of the (scaled) mean centred data matrix or from the Cholesky decomposition of the correlation/covariance matrix. The function e04lb ensures that the values of ψ_i are greater than a given small positive quantity, δ, so that the communality is always less than one. This avoids the so called Heywood cases.

In addition to the values of Λ, Ψ and the communalities, g03ca returns the residual correlations, i.e., the off-diagonal elements of $C - \left(\Lambda\Lambda^{\mathrm{T}} + \Psi\right)$ where C is the sample correlation matrix. g03ca also returns the test statistic:

$$\chi^2 = [n - 1 - (2p + 5)/6 - 2k/3] F\left(\hat{\Psi}\right)$$

which can be used to test the goodness-of-fit of the model (1), see Lawley and Maxwell 1971 and Morrison 1967.

4 References

Clark M R B 1970 A rapidly convergent method for maximum likelihood factor analysis *British J. Math. Statist. Psych.*

Hammarling S 1985 The singular value decomposition in multivariate statistics *SIGNUM Newsl.* **20 (3)** 2–25

Lawley D N and Maxwell A E 1971 *Factor Analysis as a Statistical Method* (2nd Edition) Butterworths

Morrison D F 1967 *Multivariate Statistical Methods* McGraw–Hill

5 Parameters

5.1 Compulsory Input Parameters

1: **matrix – string**

Selects the type of matrix on which factor analysis is to be performed.

matrix = 'D'

The data matrix will be input in **x** and factor analysis will be computed for the correlation matrix.

matrix = 'S'

The data matrix will be input in **x** and factor analysis will be computed for the covariance matrix, i.e., the results are scaled as described in Section 8.

matrix = 'C'

The correlation/variance-covariance matrix will be input in **x** and factor analysis computed for this matrix.

See Section 8.

Constraint: **matrix** = 'D', 'S' or 'C'.

2: **weight – string**

If **matrix** = 'D' or 'S', **weight** indicates if weights are to be used.

weight = 'U'

No weights are used.

weight = 'W'

Weights are used and must be supplied in **wt**.

Note: if **matrix** = 'C', **weight** is not referenced.

Constraint: if **weight** = 'U' or 'W', **matrix** = 'D' or 'S'.

3: **n – int32 scalar**

If **matrix** = 'D' or 'S' the number of observations in the data array **x**.

If **matrix** = 'C' the (effective) number of observations used in computing the (possibly weighted) correlation/variance-covariance matrix input in **x**.

Constraint: **n** > **nvar**.

4: **x(ldx,m) – double array**

ldx, the first dimension of the array, must be at least

if **matrix** = 'D' or 'S', **ldx** ≥ **n**;
if **matrix** = 'C', **ldx** ≥ **m**.

.

The input matrix.

If **matrix** = 'D' or 'S', **x** must contain the data matrix, i.e., **x**(i,j) must contain the ith observation for the jth variable, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots,$ **m**.

If **matrix** = 'C', **x** must contain the correlation or variance-covariance matrix. Only the upper triangular part is required.

5: **nvar – int32 scalar**

p, the number of variables in the factor analysis.

Constraint: **nvar** ≥ 2.

6: **isx(m) – int32 array**

isx(j) indicates whether or not the jth variable is included in the factor analysis. If **isx**$(j) \geq 1$, the variable represented by the jth column of **x** is included in the analysis; otherwise it is excluded, for $j = 1, 2, \ldots,$ **m**.

Constraint: **isx**$(j) > 0$ for **nvar** values of j.

7: **nfac – int32 scalar**

k, the number of factors.

Constraint: 1 ≤ **nfac** ≤ **nvar**.

8: **wt(*) – double array**

Note: the dimension of the array **wt** must be at least **n** if **weight** = 'W' and **matrix** = 'D' or 'S', and at least 1 otherwise.

If **weight** = 'W' and **matrix** = 'D' or 'S', **wt** must contain the weights to be used in the factor analysis. The effective number of observations in the analysis will then be the sum of weights. If **wt**$(i) = 0.0$, the ith observation is not included in the analysis.

If **weight** = 'U' or **matrix** = 'C', **wt** is not referenced and the effective number of observations is n.

Constraint: if **weight** = 'W', the sum of weights > **nvar**, **wt**$(i) \geq 0.0$, for $i = 1, 2, \ldots, n$.

9: **iop(5) – int32 array**

Options for the optimization. There are four options to be set:

iprint controls iteration monitoring;
if $iprint \leq 0$, then there is no printing of information else if $iprint > 0$, then information is printed at every *iprint* iterations. The information printed consists of the value of $F(\Psi)$ at that iteration, the number of evaluations of $F(\Psi)$, the current estimates of the communalities and an indication of whether or not they are at the boundary.

maxfun the maximum number of function evaluations.

acc the required accuracy for the estimates of ψ_i.

eps a lower bound for the values of ψ, see Section 3.

Let $\epsilon =$ ***machine precision*** then if $\mathbf{iop}(1) = 0$, then the following default values are used:

$$iprint = -1$$

$$maxfun = 100p$$

$$acc = 10\sqrt{\epsilon}$$

$$eps = \epsilon$$

If $\mathbf{iop}(1) \neq 0$, then

$$iprint = \mathbf{iop}(2)$$

$$maxfun = \mathbf{iop}(3)$$

$$acc = 10^{-l} \text{ where } l = \mathbf{iop}(4)$$

$$eps = 10^{-l} \text{ where } l = \mathbf{iop}(5)$$

Constraint: if $\mathbf{iop}(1) \neq 0$, $\mathbf{iop}(i)$ must be such that $maxfun \geq 1$, $\epsilon \leq acc < 1.0$ and $\epsilon \leq eps < 1.0$, for $i = 3, 4, 5$.

5.2 Optional Input Parameters

1: **m – int32 scalar**

Default: The dimension of the array **x**.

the number of variables in the data/correlation/variance-covariance matrix.

Constraint: $\mathbf{m} \geq \mathbf{nvar}$.

5.3 Input Parameters Omitted from the MATLAB Interface

ldx, ldfl, iwk, wk, lwk

5.4 Output Parameters

1: **e(nvar) – double array**

The eigenvalues θ_i, for $i = 1, 2, \ldots, p$.

2: **stat(4) – double array**

the test statistics.

$\mathbf{stat}(1)$ contains the value $F(\hat{\Psi})$.

$\mathbf{stat}(2)$ contains the test statistic, χ^2.

$\mathbf{stat}(3)$ contains the degrees of freedom associated with the test statistic.

$\mathbf{stat}(4)$ contains the significance level.

3: **com**(**nvar**) – **double array**

The communalities.

4: **psi**(**nvar**) – **double array**

The estimates of ψ_i, for $i = 1, 2, \ldots, p$.

5: **res**(**nvar** × (**nvar** − **1**)/**2**) – **double array**

The residual correlations. The residual correlation for the ith and jth variables is stored in **res**$((j-1)(j-2)/2 + i)$, $i < j$.

6: **fl**(**ldfl**,**nfac**) – **double array**

The factor loadings. **fl**(i,j) contains λ_{ij}, for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, k$.

7: **ifail** – **int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Note: g03ca may return useful information for one or more of the following detected errors or warnings.

ifail = 1

On entry, **ldfl** < **nvar**,
or **nvar** < 2,
or **n** ≤ **nvar**,
or **nfac** < 1,
or **nvar** < **nfac**,
or **m** < **nvar**,
or **matrix** = 'D' or 'S' and **ldx** < **n**,
or **matrix** = 'C' and **ldx** < **m**,
or **matrix** ≠ 'D', 'S' or 'C',
or **matrix** = 'D' or 'S' and **weight** ≠ 'U' or 'W',
or **iop**(1) ≠ 0 and **iop**(3) is such that $maxfun < 1$,
or **iop**(1) ≠ 0 and **iop**(4) is such that $acc \geq 1.0$,
or **iop**(1) ≠ 0 and **iop**(4) is such that $acc <$ ***machine precision***,
or **iop**(1) ≠ 0 and **iop**(5) is such that $eps \geq 1.0$,
or **iop**(1) ≠ 0 and **iop**(5) is such that $eps <$ ***machine precision***,
or **matrix** = 'C' and **lwk** < (5 × **nvar** × **nvar** + 33 × **nvar** − 4)/2,
or **matrix** = 'D' or 'S' and
lwk < max((5 × **nvar** × **nvar** + 33 × **nvar** − 4)/2, **n** × **nvar** + 7 × **nvar** + **nvar** × (**nvar** − 1)/2).

ifail = 2

On entry, **weight** = 'W' and a value of **wt** < 0.0.

ifail = 3

On entry, there are not exactly **nvar** elements of **isx** > 0, or the effective number of observations ≤ **nvar**.

ifail = 4

On entry, **matrix** = 'D' or 'S' and the data matrix is not of full column rank, or **matrix** = 'C' and the input correlation/variance-covariance matrix is not positive-definite.

This exit may also be caused by two of the eigenvalues of S^* being equal; this is rare (see Lawley and Maxwell 1971), and may be due to the data/correlation matrix being almost singular.

ifail = 5

A singular value decomposition has failed to converge. This is a very unlikely error exit.

ifail = 6

The estimation procedure has failed to converge in the given number of iterations. Change **iop** to either increase number of iterations $maxfun$ or increase the value of acc.

ifail = 7

The convergence is not certain but a lower point could not be found. See e04lb for further details. In this case all results are computed.

7 Accuracy

The accuracy achieved is discussed in e04lb with the value of the parameter **xtol** given by acc as described in parameter **iop**.

8 Further Comments

The factor loadings may be orthogonally rotated by using g03ba and factor score coefficients can be computed using g03cc. The maximum likelihood estimators are invariant to a change in scale. This means that the results obtained will be the same (up to a scaling factor) if either the correlation matrix or the variance-covariance matrix is used. As the correlation matrix ensures that all values of ψ_i are between 0 and 1 it will lead to a more efficient optimization. In the situation when the data matrix is input the results are always computed for the correlation matrix and then scaled if the results for the covariance matrix are required. When you input the covariance/correlation matrix the input matrix itself is used and you are advised to input the correlation matrix rather than the covariance matrix.

9 Example

```
matrix = 'C';
weight = 'U';
n = int32(211);
x = [1, 0.523, 0.395, 0.471, 0.346, 0.426, 0.576, 0.434, 0.639;
     0.523, 1, 0.479, 0.506, 0.418, 0.462, 0.547, 0.283, 0.645;
     0.395, 0.479, 1, 0.355, 0.27, 0.254, 0.452, 0.219, 0.504;
       0.471, 0.506, 0.355, 1, 0.6909999999999999, 0.791, 0.443, 0.285,
0.505;
       0.346, 0.418, 0.27, 0.6909999999999999, 1, 0.679, 0.383, 0.149,
0.409;
     0.426, 0.462, 0.254, 0.791, 0.679, 1, 0.372, 0.314, 0.472;
     0.576, 0.547, 0.452, 0.443, 0.383, 0.372, 1, 0.385, 0.68;
     0.434, 0.283, 0.219, 0.285, 0.149, 0.314, 0.385, 1, 0.47;
     0.639, 0.645, 0.504, 0.505, 0.409, 0.472, 0.68, 0.47, 1];
nvar = int32(9);
isx = [int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1)];
nfac = int32(3);
wt = [0];
iop = [int32(1);
     int32(-1);
     int32(500);
     int32(2);
     int32(5)];
[e, stat, com, psi, res, fl, ifail] = ...
    g03ca(matrix, weight, n, x, nvar, isx, nfac, wt, iop)
```

```
e =
   15.9681
    4.3577
    1.8474
    1.1560
    1.1190
    1.0271
    0.9257
    0.8951
    0.8771
stat =
    0.0350
    7.1494
   12.0000
    0.8476
com =
    0.5495
    0.5729
    0.3835
    0.7877
    0.6195
    0.8231
    0.6005
    0.5384
    0.7691
psi =
    0.4505
    0.4271
    0.6165
    0.2123
    0.3805
    0.1769
    0.3995
    0.4616
    0.2309
res =
    0.0004
   -0.0128
    0.0220
    0.0114
   -0.0053
    0.0231
   -0.0100
   -0.0194
   -0.0162
    0.0033
   -0.0046
    0.0113
   -0.0122
   -0.0009
   -0.0008
    0.0153
   -0.0216
   -0.0108
    0.0023
    0.0294
   -0.0123
   -0.0011
   -0.0105
    0.0134
    0.0054
   -0.0057
   -0.0009
    0.0032
   -0.0059
    0.0097
   -0.0049
   -0.0114
    0.0020
    0.0074
    0.0033
   -0.0012
fl =
    0.6642   -0.3209    0.0735
    0.6888   -0.2471   -0.1933
    0.4926   -0.3022   -0.2224
    0.8372    0.2924   -0.0354
    0.7050    0.3148   -0.1528
    0.8187    0.3767    0.1045
    0.6615   -0.3960   -0.0777
    0.4579   -0.2955    0.4913
    0.7657   -0.4274   -0.0117
ifail =
           0
```