NAG Fortran Library Routine Document

G03CAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G03CAF 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 Specification

```
 SUBROUTINE G03CAF(MATRIX, WEIGHT, N, M, X, LDX, NVAR, ISX, NFAC, WT, E,
1                  STAT, COM, PSI, RES, FL, LDFL, IOP, IWK, WK, LWK,
2                  IFAIL)
 INTEGER           N, M, LDX, NVAR, ISX(M), NFAC, LDFL, IOP(5),
1                  IWK(4*NVAR+2), LWK, IFAIL
 real              X(LDX,M), WT(*), WT(*), E(NVAR), STAT(4), COM(NVAR),
1                  PSI(NVAR), RES(NVAR*(NVAR-1)/2), FL(LDFL,NFAC),
2                  WK(LWK)
 CHARACTER*1       MATRIX, WEIGHT
```

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:

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

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 routine 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:

$$\Sigma = \Lambda\Lambda^{\mathrm{T}} + \Psi \qquad (1)$$

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:

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

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:

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

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 E04LBF 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 routine E04LBF 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, G03CAF returns the residual correlations, i.e., the off-diagonal elements of $C - (\Lambda\Lambda^{\mathrm{T}} + \Psi)$ where C is the sample correlation matrix. G03CAF also returns the test statistic:

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

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.*

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

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

Morrison D F (1967) *Multivariate Statistical Methods* McGraw-Hill

5 Parameters

1: MATRIX – CHARACTER*1 *Input*

On entry: selects the type of matrix on which factor analysis is to be performed.

If MATRIX = 'D' (Data input), then the data matrix will be input in X and factor analysis will be computed for the correlation matrix.

If MATRIX = 'S', then 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.

If MATRIX = 'C', then the correlation/variance-covariance matrix will be input in X and factor analysis computed for this matrix.

See Section 8 for further comments.

Constraint: MATRIX = 'D', 'S' or 'C'.

2: WEIGHT – CHARACTER*1 *Input*

On entry: if MATRIX = 'D' or 'S', WEIGHT indicates if weights are to be used.

If WEIGHT = 'U', then no weights are used.

If WEIGHT = 'W', then weights are used and must be supplied in WT.

Note: if MATRIX = 'C', WEIGHT is not referenced.

Constraint: if MATRIX = 'D' or 'S', WEIGHT = 'U' or 'W'.

3: N – INTEGER *Input*

On entry: 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: M – INTEGER *Input*

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

Constraint: M ≥ NVAR.

5: X(LDX,M) – ***real*** array *Input*

On entry: the input matrix.

If MATRIX = 'D' or 'S', then X must contain the data matrix, i.e., X(i,j) must contain the ith observation for the jth variable, for $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots,$ M.

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

6: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G03CAF is called.

Constraints:

if MATRIX = 'D' or 'S', then LDX ≥ N,
if MATRIX = 'C', then LDX ≥ M.

7: NVAR – INTEGER *Input*

On entry: the number of variables in the factor analysis, p.

Constraint: NVAR ≥ 2.

8: ISX(M) – INTEGER array *Input*

On entry: ISX(j) indicates whether or not the jth variable is included in the factor analysis. If ISX$(j) \geq 1$, then 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.

9: NFAC – INTEGER *Input*

On entry: the number of factors, k.

Constraint: 1 ≤ NFAC ≤ NVAR.

10: WT(*) – ***real*** array *Input*

On entry: 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, then 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', then WT(i) ≥ 0.0, for $i = 1, 2, \ldots, n$, and the sum of weights > NVAR.

11: E(NVAR) – ***real*** array *Output*

On exit: the eigenvalues θ_i, for $i = 1, 2, \ldots, p$.

12: STAT(4) – ***real*** array *Output*

On exit: the test statistics.

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

STAT(2) contains the test statistic, χ^2.

STAT(3) contains the degrees of freedom associated with the test statistic.

STAT(4) contains the significance level.

13: COM(NVAR) – ***real*** array *Output*

On exit: the communalities.

14: PSI(NVAR) – ***real*** array *Output*

On exit: the estimates of ψ_i, for $i = 1, 2, \ldots, p$.

15: RES(NVAR*(NVAR−1)/2) – ***real*** array *Output*

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

16: FL(LDFL,NFAC) – ***real*** array *Output*

On exit: the factor loadings. FL(i,j) contains λ_{ij}, for $i = 1, 2, \ldots, p$; $j = 1, 2, \ldots, k$.

17: LDFL – INTEGER *Input*

On entry: the first dimension of the array FL as declared in the (sub)program from which G03CAF is called.

Constraint: LDFL $\geq$ NVAR.

18: IOP(5) – INTEGER array *Input*

On entry: 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 ϵ = ***machine precision*** then if IOP(1) = 0, then the following default values are used:

$iprint = -1$

$maxfun = 100p$

$acc = 10\sqrt{\epsilon}$

$eps = \epsilon$

If IOP(1) $\neq 0$, then

$iprint =$ IOP(2)

$maxfun =$ IOP(3)

$acc = 10^{-l}$ where $l =$ IOP(4)

$eps = 10^{-l}$ where $l =$ IOP(5)

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

19: IWK(4*NVAR+2) – INTEGER array *Workspace*

20: WK(LWK) – ***real*** array *Workspace*

21: LWK – INTEGER *Input*

On entry: the length of the workspace.

Constraints:

if MATRIX = 'D' or 'S', then LWK $\geq$ max((5 × NVAR × NVAR + 33 × NVAR − 4)/2, N × NVAR + 7 × NVAR + NVAR × (NVAR − 1)/2);
if MATRIX = 'C', then LWK $\geq$ (5 × NVAR × NVAR + 33 × NVAR − 4)/2.

22: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value −1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, because for this routine the values of the output parameters may be useful even if IFAIL $\neq$ 0 on exit, the recommended value is −1. **When the value −1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL = 0 or −1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, LDFL < NVAR,
or NVAR < 2,
or N $\leq$ 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 $\neq$ 'D''S' or 'C',
or MATRIX = 'D' or 'S' and WEIGHT $\neq$ 'U' or 'W',

or IOP(1) $\neq$ 0 and IOP(3) is such that $maxfun < 1$,
or IOP(1) $\neq$ 0 and IOP(4) is such that $acc \geq 1.0$,
or IOP(1) $\neq$ 0 and IOP(4) is such that $acc <$ ***machine precision***,
or IOP(1) $\neq$ 0 and IOP(5) is such that $eps \geq 1.0$,
or IOP(1) $\neq$ 0 and IOP(5) is such that $eps <$ ***machine precision***,
or MATRIX = 'C' and LWK < $(5 \times \text{NVAR} \times \text{NVAR} + 33 \times \text{NVAR} - 4)/2$,
or MATRIX = 'D' or 'S' and
LWK < $\max((5 \times \text{NVAR} \times \text{NVAR} + 33 \times \text{NVAR} - 4)/2, \text{N} \times \text{NVAR} + 7 \times \text{NVAR} + \text{NVAR} \times (\text{NVAR} - 1)/2)$.

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 $\leq$ 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 E04LBF for further details. In this case all results are computed.

7 Accuracy

The accuracy achieved is discussed in E04LBF with the value of the parameter XTOL given by acc as described in Section 5.

8 Further Comments

The factor loadings may be orthogonally rotated by using G03BAF and factor score coefficients can be computed using G03CCF. 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 the user inputs the covariance/correlation matrix the input matrix itself is used and so the user is advised to input the correlation matrix rather than the covariance matrix.

9 Example

The example is taken from Lawley and Maxwell (1971). The correlation matrix for nine variables is input and the parameters of a factor analysis model with three factors are estimated and printed.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     G03CAF Example Program Text
*     Mark 15 Release. NAG Copyright 1991.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, MMAX, LWK
      PARAMETER        (NMAX=9,MMAX=9,LWK=349)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, L, M, N, NFAC, NVAR
      CHARACTER        MATRIX, WEIGHT
*     .. Local Arrays ..
      real             COM(MMAX), E(MMAX), FL(MMAX,MMAX), PSI(MMAX),
     +                 RES(MMAX*(MMAX-1)/2), STAT(4), WK(LWK), WT(NMAX),
     +                 X(NMAX,MMAX)
      INTEGER          IOP(5), ISX(MMAX), IWK(4*MMAX+2)
*     .. External Subroutines ..
      EXTERNAL         G03CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G03CAF Example Program Results'
*     Skip headings in data file
      READ (NIN,*)
      READ (NIN,*) MATRIX, WEIGHT, N, M, NVAR, NFAC
      IF (M.LE.MMAX .AND. (MATRIX.EQ.'C' .OR. MATRIX.EQ.'c' .OR. N.LE.
     +    NMAX)) THEN
         IF (MATRIX.EQ.'C' .OR. MATRIX.EQ.'c') THEN
            DO 20 I = 1, M
               READ (NIN,*) (X(I,J),J=1,M)
   20       CONTINUE
         ELSE
            IF (WEIGHT.EQ.'W' .OR. WEIGHT.EQ.'w') THEN
               DO 40 I = 1, N
                  READ (NIN,*) (X(I,J),J=1,M), WT(I)
   40          CONTINUE
            ELSE
               DO 60 I = 1, N
                  READ (NIN,*) (X(I,J),J=1,M)
   60          CONTINUE
            END IF
         END IF
         READ (NIN,*) (ISX(J),J=1,M)
         READ (NIN,*) (IOP(J),J=1,5)
         IFAIL = -1
*
         CALL G03CAF(MATRIX,WEIGHT,N,M,X,NMAX,NVAR,ISX,NFAC,WT,E,STAT,
     +               COM,PSI,RES,FL,MMAX,IOP,IWK,WK,LWK,IFAIL)
*
         IF (IFAIL.EQ.0 .OR. IFAIL.GT.4) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,*) ' Eigenvalues'
            WRITE (NOUT,*)
            WRITE (NOUT,99998) (E(J),J=1,M)
            WRITE (NOUT,*)
            WRITE (NOUT,99997) '     Test Statistic = ', STAT(2)
            WRITE (NOUT,99997) '                 df = ', STAT(3)
            WRITE (NOUT,99997) ' Significance level = ', STAT(4)
            WRITE (NOUT,*)
            WRITE (NOUT,*) ' Residuals'
            WRITE (NOUT,*)
            L = 1
            DO 80 I = 1, NVAR - 1
               WRITE (NOUT,99999) (RES(J),J=L,L+I-1)
               L = L + I
   80       CONTINUE
            WRITE (NOUT,*)
            WRITE (NOUT,*) ' Loadings, Communalities and PSI'
```

```
            WRITE (NOUT,*)
            DO 100 I = 1, NVAR
               WRITE (NOUT,99999) (FL(I,J),J=1,NFAC), COM(I), PSI(I)
  100       CONTINUE
         END IF
      END IF
      STOP
*
99999 FORMAT (2X,9F8.3)
99998 FORMAT (2X,6e12.4)
99997 FORMAT (A,F6.3)
      END
```

9.2 Program Data

```
G03CAF Example Program Data
'C' 'U' 211 9  9 3
 1.000 0.523 0.395 0.471 0.346 0.426 0.576 0.434 0.639
 0.523 1.000 0.479 0.506 0.418 0.462 0.547 0.283 0.645
 0.395 0.479 1.000 0.355 0.270 0.254 0.452 0.219 0.504
 0.471 0.506 0.355 1.000 0.691 0.791 0.443 0.285 0.505
 0.346 0.418 0.270 0.691 1.000 0.679 0.383 0.149 0.409
 0.426 0.462 0.254 0.791 0.679 1.000 0.372 0.314 0.472
 0.576 0.547 0.452 0.443 0.383 0.372 1.000 0.385 0.680
 0.434 0.283 0.219 0.285 0.149 0.314 0.385 1.000 0.470
 0.639 0.645 0.504 0.505 0.409 0.472 0.680 0.470 1.000
  1    1    1   1    1   1   1   1   1
1 -1 500 2 5
```

9.3 Program Results

```
 G03CAF Example Program Results

  Eigenvalues

    0.1597E+02  0.4358E+01  0.1847E+01  0.1156E+01  0.1119E+01  0.1027E+01
    0.9257E+00  0.8951E+00  0.8771E+00

     Test Statistic =  7.149
                 df = 12.000
 Significance level =  0.848

  Residuals

     0.000
    -0.013   0.022
     0.011  -0.005   0.023
    -0.010  -0.019  -0.016   0.003
    -0.005   0.011  -0.012  -0.001  -0.001
     0.015  -0.022  -0.011   0.002   0.029  -0.012
    -0.001  -0.011   0.013   0.005  -0.006  -0.001   0.003
    -0.006   0.010  -0.005  -0.011   0.002   0.007   0.003  -0.001

  Loadings, Communalities and PSI

     0.664  -0.321   0.074   0.550   0.450
     0.689  -0.247  -0.193   0.573   0.427
     0.493  -0.302  -0.222   0.383   0.617
     0.837   0.292  -0.035   0.788   0.212
     0.705   0.315  -0.153   0.619   0.381
     0.819   0.377   0.105   0.823   0.177
     0.661  -0.396  -0.078   0.600   0.400
     0.458  -0.296   0.491   0.538   0.462
     0.766  -0.427  -0.012   0.769   0.231
```