1 Purpose
nag_binary_factor (g11sac) fits a latent variable model (with a single factor) to data consisting of a set of measurements on individuals in the form of binary-valued sequences (generally referred to as score patterns). Various measures of goodness-of-fit are calculated along with the factor (theta) scores.

2 Specification

```c
void nag_binary_factor (Nag_OrderType order, Integer p, Integer n, Boolean gprob, Integer ns, Boolean x[], Integer pdx[], Integer irl[], double a[], double c[], Integer *niter, const char *outfile, double cgetol, Integer maxit, Boolean chisqr, Integer *niter, double alpha[], double pigam[], double cm[], Integer pdcm, double g[], double expp[], Integer pde, double obs[], double exf[], double y[], Integer iob[], double *rlogl, double *chi, Integer *idf, double *siglev, NagError *fail)
```

3 Description

Given a set of \( p \) dichotomous variables \( \mathbf{x} = (x_1, x_2, \ldots, x_p)' \), where \( ' \) denotes vector or matrix transpose, the objective is to investigate whether the association between them can be adequately explained by a latent variable model of the form (see Bartholomew (1980) and Bartholomew (1987))

\[
G\{\pi_i(\theta)\} = \alpha_{i0} + \alpha_{i1}\theta.
\] (1)

The \( x_i \) are called item responses and take the value 0 or 1. \( \theta \) denotes the latent variable assumed to have a standard Normal distribution over a population of individuals to be tested on \( p \) items. Call \( \pi_i(\theta) = P(x_i = 1|\theta) \) the item response function: it represents the probability that an individual with latent ability \( \theta \) will produce a positive response (1) to item \( i \). \( \alpha_{i0} \) and \( \alpha_{i1} \) are item parameters which can assume any real values. The set of parameters, \( \alpha_{ij} \), for \( i = 1, 2, \ldots, p \), being coefficients of the unobserved variable \( \theta \), can be interpreted as ‘factor loadings’.

\( G \) is a function selected by the user as either \( \Phi^{-1} \) or logit, mapping the interval (0,1) onto the whole real line. Data from a random sample of \( n \) individuals takes the form of the matrices \( X \) and \( R \) defined below:

\[
X_{s \times p} = \begin{bmatrix}
  x_{11} & x_{12} & \ldots & x_{1p} \\
  x_{21} & x_{22} & \ldots & x_{2p} \\
  \vdots & \vdots & \ddots & \vdots \\
  x_{s1} & x_{s2} & \ldots & x_{sp}
\end{bmatrix},
\]

\[
R_{s \times 1} = \begin{bmatrix}
  r_1 \\
  r_2 \\
  \vdots \\
  r_s
\end{bmatrix}
\]

where \( x_{il} = (x_{i1}, x_{i2}, \ldots, x_{ip})' \) denotes the \( l \)th score pattern in the sample, \( r_l \) the frequency with which \( \mathbf{x}_l \) occurs and \( s \) the number of different score patterns observed. (Thus \( \sum_{l=1}^{s} r_l = n \)). It can be shown that the log likelihood function is proportional to

\[
\sum_{l=1}^{s} r_l \log P_l,
\]

where

\[
P_l = P(x = \mathbf{x}_l) = \int_{-\infty}^{\infty} P(\mathbf{x} = \mathbf{x}_l|\theta)\phi(\theta)\,d\theta
\]

(\( \phi(\theta) \) being the probability density function of a standard Normal random variable).

[NP3645/7] g11sac.1
The unconditional probability of observing score pattern \( \bar{x}_i \). The integral in (2) is approximated using Gauss–Hermite quadrature. If we take \( G(z) = \text{logit } z = \log\left(\frac{z}{1-z}\right) \) in (1) and reparametrise as follows,

\[
\alpha_i = \alpha_{i1}, \\
\pi_i = \pi_{i0},
\]

then (1) reduces to the logit model (see Bartholomew (1980))

\[
\pi_i(\theta) = \frac{\pi_i}{\pi_i + (1 - \pi_i) \exp(-\alpha_i \theta)}.
\]

If we take \( G(z) = \Phi^{-1}(z) \) (where \( \Phi \) is the cumulative distribution function of a standard Normal random variable) and reparametrise as follows,

\[
\alpha_i = \frac{\alpha_{i1}}{\sqrt{1 + \alpha_{i1}^2}} \\
\gamma_i = \frac{-\alpha_{i0}}{\sqrt{1 + \alpha_{i1}^2}}
\]

then (1) reduces to the probit model (see Bock and Aitkin (1981))

\[
\pi_i(\theta) = \phi\left(\frac{\alpha_i \theta - \gamma_i}{\sqrt{1 + \gamma_i^2}}\right).
\]

An E-M algorithm (see Bock and Aitkin (1981)) is used to maximize the log likelihood function. The number of quadrature points used is set initially to 10 and once convergence is attained increased to 20. The theta score of an individual responding in score pattern \( \bar{x}_i \) is computed as the posterior mean, i.e.,

\[ E(\theta | \bar{x}_i) \].

For the logit model the component score \( X_l = \sum_{j=1}^{p} \alpha_{ij}x_{ij} \) is also calculated. (Note that in calculating the theta scores and measures of goodness-of-fit nag_binary_factor (g11sac) automatically reverses the coding on item \( j \) if \( \alpha_j < 0 \); it is assumed in the model that a response at the one level is showing a higher measure of latent ability than a response at the zero level.)

The frequency distribution of score patterns is required as input data. If the user’s data is in the form of individual score patterns (uncounted), then nag_binary_factor_service (g11sbc) may be used to calculate the frequency distribution.

### 4 References


### 5 Parameters

1: \( \text{order} - \) **Nag_OrderType**

*Input*

*On entry:* the \text{order} parameter specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by \text{order} = **Nag_RowMajor**. See Section 2.2.1.4 of the Essential Introduction for a more detailed explanation of the use of this parameter.

*Constraint:* \text{order} = **Nag_RowMajor** or **Nag_ColMajor**.
g11 – Contingency Table Analysis

2: \( p \)  – Integer  
   \textit{Input}
   \textit{On entry:} the number of dichotomous variables, \( p \).
   \textit{Constraint:} \( p \geq 3 \).

3: \( n \)  – Integer  
   \textit{Input}
   \textit{On entry:} the number of individuals in the sample, \( n \).
   \textit{Constraint:} \( n \geq 7 \).

4: \( gprob \)  – Boolean  
   \textit{Input}
   \textit{On entry:} \( gprob \) must be set equal to \( \text{TRUE} \) if \( G(z) = \Phi^{-1}(z) \) and \( \text{FALSE} \) if \( G(z) = \logit z \).

5: \( ns \)  – Integer  
   \textit{Input}
   \textit{On entry:} \( ns \) must be set equal to the number of different score patterns in the sample, \( s \).
   \textit{Constraint:} \( 2 \times p < ns \leq \min(2^p, n) \).

6: \( x[dim] \)  – Boolean  
   \textit{Input/Output}
   \textit{Note:} the dimension, \( dim \), of the array \( x \) must be at least \( \max(1, pdx \times p) \) when \( \text{order} = \text{Nag\_ColMajor} \) and at least \( \max(1, pdx \times ns) \) when \( \text{order} = \text{Nag\_RowMajor} \).

Where \( X(i,j) \) appears in this document, it refers to the array element

\[
\begin{align*}
\text{if order = Nag\_ColMajor,} & \quad x[(j-1) \times pdx + i - 1]; \\
\text{if order = Nag\_RowMajor,} & \quad x[(i-1) \times pdx + j - 1].
\end{align*}
\]

\textit{On entry:} the first \( s \) rows of \( x \) must contain the \( s \) different score patterns. The \( l \)th row of \( x \) must contain the \( l \)th score pattern with \( X(l,j) \) set equal to \( \text{TRUE} \) if \( x_{lj} = 1 \) and \( \text{FALSE} \) if \( x_{lj} = 0 \). All rows of \( x \) must be distinct.

\textit{On exit:} given a valid parameter set then the first \( s \) rows of \( x \) still contain the \( s \) different score patterns. However, the following points should be noted:

(i) If the estimated factor loading for the \( j \)th item is negative then that item is re-coded, i.e., 0s and 1s (or \( \text{TRUE} \) and \( \text{FALSE} \)) in the \( j \)th column of \( x \) are interchanged.

(ii) The rows of \( x \) will be re-ordered so that the theta scores corresponding to rows of \( x \) are in increasing order of magnitude.

7: \( pdx \)  – Integer  
   \textit{Input}
   \textit{On entry:} the stride separating matrix row or column elements (depending on the value of \( \text{order} \)) in the array \( x \).

\textit{Constraints:}

\[
\begin{align*}
\text{if order = Nag\_ColMajor,} & \quad pdx \geq ns; \\
\text{if order = Nag\_RowMajor,} & \quad pdx \geq p.
\end{align*}
\]

8: \( \text{irl}[ns] \)  – Integer  
   \textit{Input/Output}
   \textit{On entry:} the \( i \)th component of \( \text{irl} \) must be set equal to the frequency with which the \( i \)th row of \( x \) occurs.

\textit{Constraint:}

\[
\begin{align*}
\text{irl}[i] & \geq 0 \quad \text{for } i = 0, 1, \ldots, s - 1; \\
\sum_{i=0}^{s-1} \text{irl}[i] & = n.
\end{align*}
\]

\textit{On exit:} given a valid parameter set then the first \( s \) components of \( \text{irl} \) are re-ordered as are the rows of \( x \).
9: \( \mathbf{a} \) – double

*Input/Output*

On entry: \( \mathbf{a}[j-1] \) must be set equal to an initial estimate of \( \alpha_{j1} \). **In order to avoid divergence problems with the E-M algorithm the user is strongly advised to set all the \( \mathbf{a}[j-1] \) to 0.5.**

On exit: \( \mathbf{a}[j-1] \) contains the latest estimate of \( \alpha_{j1} \), for \( j = 1, 2, \ldots, p \). (Because of possible re-encoding all elements of \( \mathbf{a} \) will be positive.)

10: \( \mathbf{c} \) – double

*Input/Output*

On entry: \( \mathbf{c}[j-1] \) must be set equal to an initial estimate of \( \alpha_{j0} \). **In order to avoid divergence problems with the E-M algorithm the user is strongly advised to set all the \( \mathbf{c}[j-1] \) to 0.0.**

On exit: \( \mathbf{c}[j-1] \) contains the latest estimate of \( \alpha_{j0} \), for \( j = 1, 2, \ldots, p \).

11: \textit{iprint} – Integer

*Input*

On entry: the frequency with which the maximum likelihood search routine is to be monitored.

If \( \textit{iprint} > 0 \), the search is monitored once every \( \textit{iprint} \) iterations, and when the number of quadrature points is increased, and again at the final solution point.

If \( \textit{iprint} = 0 \), the search is monitored once at the final point.

If \( \textit{iprint} < 0 \), the search is not monitored at all.

\textit{iprint} should normally be set to a small positive number.

*Suggested value: \( \textit{iprint} = 1 \).*

12: \textit{outfile} – char *

*Input*

On entry: the name of a file to which diagnostic output will be directed. If \( \textit{outfile} \) is NULL the diagnostic output will be directed to standard output.

13: \textit{cgetol} – double

*Input*

On entry: the accuracy to which the solution is required. If \( \textit{cgetol} \) is set to \( 10^{-4} \) and on exit fail.code = \texttt{NE_NOERROR} or \texttt{NEZERO_DF}, then all elements of the gradient vector will be smaller than \( 10^{-4} \) in absolute value. For most practical purposes the value \( 10^{-4} \) should suffice. The user should be wary of setting \( \textit{cgetol} \) too small since the convergence criterion may then have become too strict for the machine to handle. If \( \textit{cgetol} \) has been set to a value which is less than the square root of the *machine precision*, \( \epsilon \), then \texttt{nag_binary_factor} (g11sac) will use the value \( \sqrt{\epsilon} \) instead.

14: \textit{maxit} – Integer

*Input*

On entry: the maximum number of iterations to be made in the maximum likelihood search. There will be an error exit (see Section 6) if the search routine has not converged in \( \textit{maxit} \) iterations.

*Constraint: \( \textit{maxit} \geq 1 \).*

*Suggested value: \( \textit{maxit} = 1000 \).*

15: \textit{chisqr} – Boolean

*Input*

On entry: if \( \textit{chisqr} \) is set equal to \texttt{TRUE}, then a likelihood ratio statistic will be calculated (see \texttt{chi}).

If \( \textit{chisqr} \) is set equal to \texttt{FALSE}, no such statistic will be calculated.

16: \textit{niter} – Integer *

*Output*

On exit: given a valid parameter set then \( \textit{niter} \) contains the number of iterations performed by the maximum likelihood search routine.
17: \texttt{alpha[p]} – double

Output

\textit{On exit:} given a valid parameter set then \texttt{alpha[j - 1]} contains the latest estimate of \( \alpha_j \). (Because of possible recoding all elements of \texttt{alpha} will be positive.)

18: \texttt{pigam[p]} – double

Output

\textit{On exit:} given a valid parameter set then \texttt{pigam[j - 1]} contains the latest estimate of either \( \pi_j \) if \texttt{gprob = FALSE} (logit model) or \( \gamma_j \) if \texttt{gprob = TRUE} (probit model).

19: \texttt{cm[dim]} – double

Output

\textit{Note:} the dimension, \texttt{dim}, of the array \texttt{cm} must be at least \texttt{pdcn} \( \times 2 \times \texttt{p} \).

Where \texttt{CM(i, j)} appears in this document, it refers to the array element

\begin{align*}
\text{if} \quad \texttt{order = Nag\_ColMajor}, & \quad \texttt{cm}[(j - 1) \times \texttt{pdcn} + i - 1]; \\
\text{if} \quad \texttt{order = Nag\_RowMajor}, & \quad \texttt{cm}[(i - 1) \times \texttt{pdcn} + j - 1].
\end{align*}

\textit{On exit:} given a valid parameter set then the strict lower triangle of \texttt{cm} contains the correlation matrix of the parameter estimates held in \texttt{alpha} and \texttt{pigam} on exit. The diagonal elements of \texttt{cm} contain the standard errors. Thus:

\begin{align*}
\text{CM}(2 \times i - 1, 2 \times i - 1) & = \text{standard error} \left( \texttt{alpha}[i - 1] \right) \\
\text{CM}(2 \times i, 2 \times i) & = \text{standard error} \left( \texttt{pigam}[i - 1] \right) \\
\text{CM}(2 \times i, 2 \times i - 1) & = \text{correlation} \left( \texttt{pigam}[i - 1], \texttt{alpha}[i - 1] \right),
\end{align*}

for \( i = 1, 2, \ldots, \texttt{p} \).

\begin{align*}
\text{CM}(2 \times i - 1, 2 \times j - 1) & = \text{correlation} \left( \texttt{alpha}[i - 1], \texttt{alpha}[j - 1] \right) \\
\text{CM}(2 \times i, 2 \times j) & = \text{correlation} \left( \texttt{pigam}[i - 1], \texttt{pigam}[j - 1] \right) \\
\text{CM}(2 \times i - 1, 2 \times j) & = \text{correlation} \left( \texttt{alpha}[i - 1], \texttt{pigam}[j - 1] \right) \\
\text{CM}(2 \times i, 2 \times j - 1) & = \text{correlation} \left( \texttt{alpha}[j - 1], \texttt{pigam}[i - 1] \right),
\end{align*}

for \( j = 1, 2, \ldots, i - 1 \).

If the second derivative matrix cannot be computed then all the elements of \texttt{cm} are returned as zero.

20: \texttt{pdcn} – Integer

Input

\textit{On entry:} the stride separating row or column elements (depending on the value of \texttt{order}) of the matrix \textit{M} in the array \texttt{cm}.

\textit{Constraint:} \texttt{pdcn} \( \geq 2 \times \texttt{p} \).

21: \texttt{g[dim]} – double

Output

\textit{Note:} the dimension, \texttt{dim}, of the array \texttt{g} must be at least \( 2 \times \texttt{p} \).

\textit{On exit:} given a valid parameter set then \texttt{g} contains the estimated gradient vector corresponding to the final point held in the arrays \texttt{alpha} and \texttt{pigam}. \texttt{g[2 \times j - 2]} contains the derivative of the log likelihood with respect to \texttt{alpha}[j - 1], for \( j = 1, 2, \ldots, \texttt{p} \). \texttt{g[2 \times j - 1]} contains the derivative of the log likelihood with respect to \texttt{pigam}[j - 1], for \( j = 1, 2, \ldots, \texttt{p} \).

22: \texttt{expp[dim]} – double

Output

\textit{Note:} the dimension, \texttt{dim}, of the array \texttt{expp} must be at least \( \texttt{pde} \times \texttt{p} \).

Where \texttt{EXPP(i, j)} appears in this document, it refers to the array element

\begin{align*}
\text{if} \quad \texttt{order = Nag\_ColMajor}, & \quad \texttt{expp}[(j - 1) \times \texttt{pde} + i - 1]; \\
\text{if} \quad \texttt{order = Nag\_RowMajor}, & \quad \texttt{expp}[(i - 1) \times \texttt{pde} + j - 1].
\end{align*}

\textit{On exit:} given a valid parameter set then \texttt{EXPP(i, j)} contains the expected percentage of individuals in the sample who respond positively to items \( i \) and \( j \) (\( j \leq i \)), corresponding to the final point held in the arrays \texttt{alpha} and \texttt{pigam}. 

\[ \text{NP3645/7} \]
pde – Integer  

On entry: the stride separating row or column elements (depending on the value of order) of the matrix $E$ in the array expp.

Constraint: $\text{pde} \geq p$.

24: obs[dim] – double

Output  
Note: the dimension, dim, of the array obs must be at least $\text{pde}/p$.

Where $\text{OBS}(i, j)$ appears in this document, it refers to the array element

- if order = Nag_ColMajor, $\text{obs}[(j-1) \times \text{pde} + i - 1]$;
- if order = Nag_RowMajor, $\text{obs}[(i-1) \times \text{pde} + j - 1]$.

On exit: given a valid parameter set then $\text{OBS}(i, j)$ contains the observed percentage of individuals in the sample who responded positively to items $i$ and $j$ ($j \leq i$).

exf[ns] – double

Output  
On exit: given a valid parameter set then $\text{exf}[l-1]$ contains the expected frequency of the $l$th score pattern ($l$th row of $x$), corresponding to the final point held in the arrays alpha and pigam.

y[ns] – double

Output  
On exit: given a valid parameter set then $y[l-1]$ contains the estimated theta score corresponding to the $l$th row of $x$, for the final point held in the arrays alpha and pigam.

iob[ns] – Integer

Output  
On exit: given a valid parameter set then $iob[l-1]$ contains the number of items in the $l$th row of $x$ for which the response was positive (TRUE).

rlogl – double *

Output  
On exit: given a valid parameter set then $\text{rlogl}$ contains the value of the log likelihood kernel corresponding to the final point held in the arrays alpha and pigam, namely

$$\sum_{l=0}^{s-1} \text{irl}[l] \times \log(\text{exf}[l]/n).$$

chi – double *

Output  
On exit: if chisqr was set equal to TRUE on entry, then given a valid parameter set, $\text{chi}$ will contain the value of the likelihood ratio statistic corresponding to the final parameter estimates held in the arrays alpha and pigam, namely

$$2 \times \sum_{l=0}^{s-1} \text{irl}[l] \times \log(\text{exf}[l]/\text{irl}[l]).$$

The summation is over those elements of $\text{irl}$ which are positive. If $\text{exf}[l-1]$ is less than 5.0, then adjacent score patterns are pooled (the score patterns in $x$ being first put in order of increasing theta score).

If chisqr has been set equal to FALSE, then $\text{chi}$ is not used.

idf – Integer *

Output  
On exit: if chisqr was set equal to TRUE on entry, then given a valid parameter set, $\text{idf}$ will contain the degrees of freedom associated with the likelihood ratio statistic, $\text{chi}$.

$$\text{idf} = s_0 - 2 \times p \quad \text{if } s_0 < 2^p;$$
$$\text{idf} = s_0 - 2 \times p - 1 \quad \text{if } s_0 = 2^p,$$

where $s_0$ denotes the number of terms summed to calculate $\text{chi}$ ($s_0 = s$ only if there is no pooling).
If \texttt{chisqr} has been set equal to \texttt{FALSE}, then \texttt{idf} is not used.

\begin{verbatim}
31: siglev – double *
     Output
     On exit: if \texttt{chisqr} was set equal to \texttt{TRUE} on entry, then given a valid parameter set, \texttt{siglev} will contain the significance level of chi based on \texttt{idf} degrees of freedom. If \texttt{idf} is zero or negative then \texttt{siglev} is set to zero. If \texttt{chisqr} was set equal to \texttt{FALSE}, then \texttt{siglev} is not used.
\end{verbatim}

\begin{verbatim}
32: fail – NagError *
     Input/Output
     The NAG error parameter (see the Essential Introduction).
\end{verbatim}

6 Error Indicators and Warnings

\textbf{NE_INT}

On entry, \texttt{p} = \langle value\rangle.
Constraint: \texttt{p} \geq 3.

On entry, \texttt{pdx} = \langle value\rangle.
Constraint: \texttt{pdx} > 0.

On entry, \texttt{pdcm} = \langle value\rangle.
Constraint: \texttt{pdcm} > 0.

On entry, \texttt{pde} = \langle value\rangle.
Constraint: \texttt{pde} > 0.

On entry, \texttt{n} = \langle value\rangle.
Constraint: \texttt{n} \geq 7.

On entry, \texttt{maxit} = \langle value\rangle.
Constraint: \texttt{maxit} \geq 1.

\textbf{NE_INT_2}

On entry, \texttt{pdx} = \langle value\rangle, \texttt{ns} = \langle value\rangle.
Constraint: \texttt{pdx} \geq \texttt{ns}.

On entry, \texttt{pdx} = \langle value\rangle, \texttt{p} = \langle value\rangle.
Constraint: \texttt{pdx} \geq \texttt{p}.

On entry, \texttt{pdcm} = \langle value\rangle, \texttt{p} = \langle value\rangle.
Constraint: \texttt{pdcm} \geq 2 \times \texttt{p}.

On entry, \texttt{pde} = \langle value\rangle, \texttt{p} = \langle value\rangle.
Constraint: \texttt{pde} \geq \texttt{p}.

On entry, \texttt{ns} > 2^{\texttt{p}}: \texttt{ns} = \langle value\rangle, \texttt{p} = \langle value\rangle.

On entry, \texttt{irl[0]} + \cdots + \texttt{irl[ns-1]} is not equal to \texttt{n}: \texttt{irl[0]} + \cdots + \texttt{irl[ns-1]} = \langle value\rangle, \texttt{n} = \langle value\rangle.

On entry, \texttt{ns} > \texttt{n}: \texttt{ns} = \langle value\rangle, \texttt{n} = \langle value\rangle.

On entry, \texttt{irl[i-1]} < 0: \texttt{i} = \langle value\rangle, \texttt{irl[i-1]} = \langle value\rangle.

On entry, rows \texttt{i} and \texttt{j} of \texttt{x} are identical: \texttt{i} = \langle value\rangle, \texttt{j} = \langle value\rangle.

On entry, \texttt{ns} \leq 2 \times \texttt{p}: \texttt{ns} = \langle value\rangle, \texttt{p} = \langle value\rangle.

\textbf{NE_INT_3}

On entry, \texttt{p} = \langle value\rangle, \texttt{n} = \langle value\rangle, \texttt{ns} = \langle value\rangle.
Constraint: 2 \times \texttt{p} < \texttt{ns} \leq \min(2^\texttt{p}, \texttt{n}).
Failure to invert Hessian matrix plus Heywood case encountered.

Failure to invert Hessian matrix and `maxit` iterations made: `maxit = \langle value \rangle`.

One of the elements of `a` has exceeded 10 in absolute value (Heywood case).

For at least one of the `p` items the responses are all at the same level.

`maxit` iterations have been performed: `maxit = \langle value \rangle`.

Chi-squared statistic has `idf` degrees of freedom: `idf = \langle value \rangle`.

Memory allocation failed.

On entry, parameter `\langle value \rangle` had an illegal value.

Cannot open file `\langle value \rangle` for writing.

Cannot close file `\langle value \rangle`.

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

On exit from `nag_binary_factor (g11sac)` if `fail.code = NE_NOERROR` or `NE_ZERO_DF` then the following condition will be satisfied:

\[
\max_{0 \leq i \leq 2^{p-1}} |g[i]| < \text{cgetol}.
\]

If `fail.code = NE_TOO_MANY_ITER` or `NE_MAT_INV` on exit (i.e., `maxit` iterations have been performed but the above condition does not hold), then the elements in `a`, `c`, `alpha` and `pigam` may still be good approximations to the maximum likelihood estimates. The user is advised to inspect the elements of `g` to see whether this is confirmed.

The number of iterations required in the maximum likelihood search depends upon the number of observed variables, `p`, and the distance of the user-supplied starting point from the solution. The number of multiplications and divisions performed in an iteration is proportional to `p`. 
8.2 Initial Estimates

The user is strongly advised to use the recommended starting values for the elements of $a$ and $c$. Divergence may result from user-supplied values even if they are very close to the solution. Divergence may also occur when an item has nearly all its responses at one level.

8.3 Heywood Cases

As in normal factor analysis, Heywood cases can often occur, particularly when $p$ is small and $n$ not very big. To overcome this difficulty the maximum likelihood search routine is terminated when the absolute value of one of the $a_{ij}$ exceeds 10.0. The user has the option of deciding whether to exit from nag_binary_factor (g11sac) (by setting fail = NAGERR_DEFAULT on entry) or to permit nag_binary_factor (g11sac) to proceed onwards as if it had exited normally from the maximum likelihood search routine (setting fail.print = TRUE or FALSE on entry). The elements in $a$, $c$, alpha and pigam may still be good approximations to the maximum likelihood estimates. The user is advised to inspect the elements $g$ to see whether this is confirmed.

8.4 Goodness of Fit Statistic

When $n$ is not very large compared to $s$ a goodness-of-fit statistic should not be calculated as many of the expected frequencies will then be less than 5.

8.5 First and Second Order Margins

The observed and expected percentages of sample members responding to individual and pairs of items held in the arrays obs and exp on exit can be converted to observed and expected numbers by multiplying all elements of these two arrays by $n/100.0$.

9 Example

A program to fit the logit latent variable model to the following data:

<table>
<thead>
<tr>
<th>Index</th>
<th>Score Pattern</th>
<th>Observed Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0000</td>
<td>154</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>0001</td>
<td>42</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>49</td>
</tr>
<tr>
<td>5</td>
<td>1001</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1100</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>0101</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>0010</td>
<td>84</td>
</tr>
<tr>
<td>9</td>
<td>1101</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>25</td>
</tr>
<tr>
<td>11</td>
<td>0011</td>
<td>75</td>
</tr>
<tr>
<td>12</td>
<td>0110</td>
<td>129</td>
</tr>
<tr>
<td>13</td>
<td>1011</td>
<td>30</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>50</td>
</tr>
<tr>
<td>15</td>
<td>0111</td>
<td>181</td>
</tr>
<tr>
<td>16</td>
<td>1111</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total 1000</td>
</tr>
</tbody>
</table>

9.1 Program Text

/* nag_binary_factor (g11sac) Example Program. */
/* Copyright 2002 Numerical Algorithms Group. */
/* Mark 7, 2002. */
```c
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg11.h>

int main(void)
{
    /* Scalars */
    double cgetol, chi, rlogl, siglev;
    Integer exit_status, i, pdcm, idf, p, iprint, is,
            j, maxit, n, niter, nrx, lw, pdx;
    NagError fail;
    Nag_OrderType order;
    Boolean chisqr, gprob;
    char flag;

    /* Arrays */
    double *a = 0, *alpha = 0, *c = 0, *cm = 0, *exf = 0, *expp = 0,
            *g = 0, *obs = 0, *pigam = 0, *xI = 0, *y = 0;
    Integer *iob = 0, *irl = 0;
    Boolean *x = 0;

    #ifdef NAG_COLUMN_MAJOR
    #define X(I,J) x[(J-1)*pdx + I - 1]
    #define CM(I,J) cm[(J-1)*pdcm + I - 1]
    order = Nag_ColMajor;
    #else
    #define X(I,J) x[(I-1)*pdx + J - 1]
    #define CM(I,J) cm[(I-1)*pdcm + J - 1]
    order = Nag_RowMajor;
    #endif

    INIT_FAIL(fail);
    exit_status = 0;
    Vprintf("g11sac Example Program Results\n");
    /* Skip heading in data file */
    Vscanf("%*[\n]");
    Vscanf("%ld%ld%ld%*[\n] ", &p, &n, &is);
    if (p > 0 && is >= 0)
    {
        /* Allocate arrays */
        pdcm = 2*p;
        nrx = is;
        lw = 4 * p * (p + 16);
        if ( !(a = NAG_ALLOC(p, double)) ||
            !(alpha = NAG_ALLOC(p, double)) ||
            !(c = NAG_ALLOC(p, double)) ||
            !(cm = NAG_ALLOC(pdcm * 2*p, double)) ||
            !(exf = NAG_ALLOC(is, double)) ||
            !(expp = NAG_ALLOC(p * p, double)) ||
            !(g = NAG_ALLOC(2*p, double)) ||
            !(obs = NAG_ALLOC(p * p, double)) ||
            !(pigam = NAG_ALLOC(p, double)) ||
            !(xI = NAG_ALLOC(is, double)) ||
            !(y = NAG_ALLOC(is, double)) ||
            !(iob = NAG_ALLOC(is, Integer)) ||
            !(irl = NAG_ALLOC(is, Integer)) ||
            !(x = NAG_ALLOC(nrx * p, Boolean)) )
            Vprintf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }
    if (order == Nag_ColMajor)
        pdx = nrx;
    else
        pdx = p;
}
```

for (i = 1; i <= is; ++i)
{
    Vscanf("%ld", &irl[i-1]);
    for (j = 1; j <= p; ++j)
    {
        Vscanf(" %c", &flag);
        X(i,j) = (flag == 'T');
    }
    Vscanf("%*[\n] ");
}
gprob = FALSE;
for (i = 1; i <= p; ++i)
{
    a[i-1] = 0.5;
    c[i-1] = 0.0;
}

/* Set iprint > 0 to obtain intermediate output */
iprint = -1;
cgetol = 1e-4;
maxit = 1000;
chisqr = TRUE;
g11sac(order, p, n, gprob, is, x, pdx, irl, a, c, iprint, 0,
cgetol, maxit, chisqr, &niter, alpha, pigam, cm,
pdcm, g, exp, p, obs, exf, y, iob, &rlogl, &chi,
&idf, &siglev, &fail);
if (fail.code != NE_NOERROR)
{
    Vprintf("Error from g11sac.\n%s\n", fail.message);
    exit_status = 1;
goto END;
}
Vprintf("\n");
Vprintf("Item Alpha (s.e.) Pi (s.e.)\n");
for (i=1; i<=p; i++)
    Vprintf(" %ld %g (%10g) %g (%10g)\n", i, alpha[i-1], CM(2*i-1,2*i-1),
pigam[i-1], CM(2*i,2*i));
Vprintf("\n");
Vprintf("Index Observed Expected Theta Pattern\n");
Vprintf(" Frequency Frequency Score\n");
for (i=1; i<=is; i++)
{
    Vprintf(" %2ld %3ld %7g %10g ", i, irl[i-1], exf[i-1],
y[i-1]);
    for (j=1; j<=p; j++)
        Vprintf("%s",X(i,j)==1?"T":"F");
    Vprintf("\n");
}
Vprintf("Chi-squared test statistic = %g\n", chi);
Vprintf("Degrees of freedom = %ld\n", idf);
Vprintf("Significance = %g\n", siglev);
}
END:
if (a) NAG_FREE(a);
if (alpha) NAG_FREE(alpha);
if (c) NAG_FREE(c);
if (cm) NAG_FREE(cm);
if (exf) NAG_FREE(exf);
if (expp) NAG_FREE(expp);
if (g) NAG_FREE(g);
if (obs) NAG_FREE(obs);
if (pigam) NAG_FREE(pigam);
if (xl) NAG_FREE(xl);
if (y) NAG_FREE(y);
if (iob) NAG_FREE(iob);
if (irl) NAG_FREE(irl);
if (x) NAG_FREE(x);
return exit_status;
}

9.2 Program Data

g11sac Example Program Data
4 1000 16
154 F F F F
11 T F F F
42 F F F T
49 F T F F
2 T F F T
10 T T F F
27 F T F T
84 F F T F
10 T T F T
25 F T F F
75 P F T T
129 F T T F
30 T F T T
50 T T T F
181 F T T T
121 T T T T

9.3 Program Results

g11sac Example Program Results

<table>
<thead>
<tr>
<th>Item</th>
<th>Alpha (s.e.)</th>
<th>Pi (s.e.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.04546 (0.148087)</td>
<td>0.218165 (0.0173623)</td>
</tr>
<tr>
<td>2</td>
<td>1.40938 (0.178937)</td>
<td>0.604378 (0.0216392)</td>
</tr>
<tr>
<td>3</td>
<td>2.65916 (0.524787)</td>
<td>0.834117 (0.0357898)</td>
</tr>
<tr>
<td>4</td>
<td>1.12169 (0.139581)</td>
<td>0.484569 (0.0198529)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
<th>Observed Frequency</th>
<th>Expected Frequency</th>
<th>Theta Score</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>154</td>
<td>147.061</td>
<td>-1.27348</td>
<td>FFFF</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>13.4437</td>
<td>-0.873074</td>
<td>TFFF</td>
</tr>
<tr>
<td>3</td>
<td>42</td>
<td>42.4201</td>
<td>-0.846239</td>
<td>FFFT</td>
</tr>
<tr>
<td>4</td>
<td>49</td>
<td>54.818</td>
<td>-0.746856</td>
<td>FFFT</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>5.88558</td>
<td>-0.494146</td>
<td>TFFT</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>8.41022</td>
<td>-0.399461</td>
<td>TFFF</td>
</tr>
<tr>
<td>7</td>
<td>27</td>
<td>27.5115</td>
<td>-0.374319</td>
<td>FFTT</td>
</tr>
<tr>
<td>8</td>
<td>84</td>
<td>92.0619</td>
<td>-0.33196</td>
<td>FFFT</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>6.23651</td>
<td>-0.0186861</td>
<td>TFFT</td>
</tr>
<tr>
<td>10</td>
<td>25</td>
<td>21.8468</td>
<td>0.0272335</td>
<td>TFFT</td>
</tr>
<tr>
<td>11</td>
<td>75</td>
<td>73.8352</td>
<td>0.0549022</td>
<td>FTTT</td>
</tr>
<tr>
<td>12</td>
<td>129</td>
<td>123.766</td>
<td>0.161802</td>
<td>FTTT</td>
</tr>
<tr>
<td>13</td>
<td>30</td>
<td>26.8989</td>
<td>0.465873</td>
<td>TPTT</td>
</tr>
<tr>
<td>14</td>
<td>50</td>
<td>50.8813</td>
<td>0.591349</td>
<td>TTTT</td>
</tr>
<tr>
<td>15</td>
<td>181</td>
<td>179.564</td>
<td>0.625634</td>
<td>FTTT</td>
</tr>
<tr>
<td>16</td>
<td>121</td>
<td>125.36</td>
<td>1.14441</td>
<td>TTTT</td>
</tr>
</tbody>
</table>

Chi-squared test statistic = 9.02731
Degrees of freedom = 7
Significance = 0.250701