NAG Toolbox for MATLAB

g12ba

1 Purpose

g12ba returns parameter estimates and other statistics that are associated with the Cox proportional hazards model for fixed covariates.

2 Syntax

```
[dev, b, se, sc, cov, res, nd, tp, sur, ifail] = g12ba(offset, ns, z,
isz, t, ic, omega, isi, b, ndmax, tol, maxit, iprint, 'n', n, 'm', m,
'ip', ip)
```

3 Description

The proportional hazard model relates the time to an event, usually death or failure, to a number of explanatory variables known as covariates. Some of the observations may be right-censored, that is the exact time to failure is not known, only that it is greater than a known time.

Let t_i, $i = 1, \ldots, n$ be the failure time or censored time for the ith observation with the vector of p covariates z_i. It is assumed that censoring and failure mechanisms are independent. The hazard function, $\lambda(t, z)$, is the probability that an individual with covariates z fails at time t given that the individual survived up to time t. In the Cox proportional hazards model (see Cox 1972b) $\lambda(t, z)$ is of the form:

$$\lambda(t, z) = \lambda_0(t) \exp\left(z^{\mathrm{T}}\beta + \omega\right)$$

where λ_0 is the base-line hazard function, an unspecified function of time, β is a vector of unknown parameters and ω is a known offset.

Assuming there are ties in the failure times giving $n_d < n$ distinct failure times, $t_{(1)} < \cdots < t_{(n_d)}$ such that d_i individuals fail at $t_{(i)}$, it follows that the marginal likelihood for β is well approximated (see Kalbfleisch and Prentice 1980) by:

$$L = \prod_{i=1}^{n_d} \frac{\exp\left(s_i^{\mathrm{T}}\beta + \omega_i\right)}{\left[\sum_{l \in R\left(t_{(i)}\right)} \exp\left(z_l^{\mathrm{T}}\beta + \omega_l\right)\right]^{d_i}} \tag{1}$$

where s_i is the sum of the covariates of individuals observed to fail at $t_{(i)}$ and $R\left(t_{(i)}\right)$ is the set of individuals at risk just prior to $t_{(i)}$, that is it is all individuals that fail or are censored at time $t_{(i)}$ along with all individuals that survive beyond time $t_{(i)}$. The maximum likelihood estimates (MLEs) of β, given by $\hat{\beta}$, are obtained by maximizing (1) using a Newton–Raphson iteration technique that includes step halving and utilizes the first and second partial derivatives of (1) which are given by equations (2) and (3) below:

$$U_j(\beta) = \frac{\partial \ln L}{\partial \beta_j} = \sum_{i=1}^{n_d} \left[s_{ji} - d_i \alpha_{ji}(\beta)\right] = 0 \tag{2}$$

for $j = 1, \ldots, p$, where s_{ji} is the jth element in the vector s_i and

$$\alpha_{ji}(\beta) = \frac{\sum_{l \in R\left(t_{(i)}\right)} z_{jl} \exp\left(z_l^{\mathrm{T}}\beta + \omega_l\right)}{\sum_{l \in R\left(t_{(i)}\right)} \exp\left(z_l^{\mathrm{T}}\beta + \omega_l\right)}.$$

Similarly,

$$I_{hj}(\beta) = -\frac{\partial^2 \ln L}{\partial \beta_h \partial \beta_j} = \sum_{i=1}^{n_d} d_i \gamma_{\mathrm{hji}} \tag{3}$$

where

$$\gamma_{\mathrm{hji}} = \frac{\sum_{l \in R\left(t_{(i)}\right)} z_{hl} z_{jl} \exp\left(z_l^{\mathrm{T}} \beta + \omega_l\right)}{\sum_{l \in R\left(t_{(i)}\right)} \exp\left(z_l^{\mathrm{T}} \beta + \omega_l\right)} - \alpha_{\mathrm{hi}}(\beta)\alpha_{ji}(\beta), \qquad h,j = 1,\ldots,p.$$

$U_j(\beta)$ is the jth component of a score vector and $I_{hj}(\beta)$ is the (h,j) element of the observed information matrix $I(\beta)$ whose inverse $I(\beta)^{-1} = \left[I_{hj}(\beta)\right]^{-1}$ gives the variance-covariance matrix of β.

It should be noted that if a covariate or a linear combination of covariates is monotonically increasing or decreasing with time then one or more of the β_j's will be infinite.

If $\lambda_0(t)$ varies across ν strata, where the number of individuals in the kth stratum is n_k, for $k = 1,\ldots,\nu$ with $n = \sum_{k=1}^{\nu} n_k$, then rather than maximizing (1) to obtain $\hat{\beta}$, the following marginal likelihood is maximized:

$$L = \prod_{k=1}^{\nu} L_k, \tag{4}$$

where L_k is the contribution to likelihood for the n_k observations in the kth stratum treated as a single sample in (1). When strata are included the covariate coefficients are constant across strata but there is a different base-line hazard function λ_0.

The base-line survivor function associated with a failure time $t_{(i)}$, is estimated as $\exp\left(-\hat{H}\left(t_{(i)}\right)\right)$, where

$$\hat{H}\left(t_{(i)}\right) = \sum_{t_{(j)} \le t_{(i)}} \left(\frac{d_i}{\sum_{l \in R\left(t_{(j)}\right)} \exp\left(z_l^{\mathrm{T}} \hat{\beta} + \omega_l\right)} \right), \tag{5}$$

where d_i is the number of failures at time $t_{(i)}$. The residual for the lth observation is computed as:

$$r(t_l) = \hat{H}(t_l) \exp\left(-z_l^{\mathrm{T}} \hat{\beta} + \omega_l\right)$$

where $\hat{H}(t_l) = \hat{H}\left(t_{(i)}\right), t_{(i)} \le t_l < t_{(i+1)}$. The deviance is defined as $-2 \times$ (logarithm of marginal likelihood). There are two ways to test whether individual covariates are significant: the differences between the deviances of nested models can be compared with the appropriate χ^2-distribution; or, the asymptotic normality of the parameter estimates can be used to form z tests by dividing the estimates by their standard errors or the score function for the model under the null hypothesis can be used to form z tests.

4 References

Cox D R 1972b Regression models in life tables (with discussion) *J. Roy. Statist. Soc. Ser. B* **34** 187–220

Gross A J and Clark V A 1975 *Survival Distributions: Reliability Applications in the Biomedical Sciences* Wiley

Kalbfleisch J D and Prentice R L 1980 *The Statistical Analysis of Failure Time Data* Wiley

5 Parameters

5.1 Compulsory Input Parameters

1: **offset – string**

Indicates if an offset is to be used.

If **offset** = 'Y', an offset must be included in **omega**.

If **offset** = 'N', no offset is included in the model.

Constraint: **offset** = 'Y' or 'N'.

2: **ns – int32 scalar**

The number of strata. If **ns** > 0 then the stratum for each observation must be supplied in **isi**.

Constraint: **ns** ≥ 0.

3: **z(ldz,m) – double array**

ldz, the first dimension of the array, must be at least **n**.

The ith row must contain the covariates which are associated with the ith failure time given in **t**.

4: **isz(m) – int32 array**

Indicates which subset of covariates is to be included in the model.

If **isz**$(j) \geq 1$, the jth covariate is included in the model.

If **isz**$(j) = 0$, the jth covariate is excluded from the model and not referenced.

Constraint: **isz**$(j) \geq 0$ and at least one and at most $n_0 - 1$ elements of **isz** must be nonzero where n_0 is the number of observations excluding any with zero value of **isi**..

5: **t(n) – double array**

The vector of n failure censoring times.

6: **ic(n) – int32 array**

The status of the individual at time t given in **t**.

If **ic**$(i) = 0$, the ith individual has failed at time **t**(i).

If **ic**$(i) = 1$, the ith individual has been censored at time **t**(i).

Constraint: **ic**$(i) = 0$ or 1, for $i = 1, 2, \ldots,$ **n**.

7: **omega(*) – double array**

Note: the dimension of the array **omega** must be at least **n** if **offset** = 'Y', and at least 1 otherwise.

If **offset** = 'Y', the offset, ω_i, for $i = 1, 2, \ldots,$ **n**. Otherwise **omega** is not referenced.

8: **isi(*) – int32 array**

Note: the dimension of the array **isi** must be at least **n** if **ns** > 0, and at least 1 otherwise.

If **ns** > 0, the stratum indicators which also allow data points to be excluded from the analysis.

If **ns** $= 0$, **isi** is not referenced.

isi$(i) = k$

The ith data point is in the kth stratum, where $k = 1, 2, \ldots,$ **ns**.

isi$(i) = 0$

The ith data point is omitted from the analysis.

Constraint: if **ns** > 0, $0 \leq$ **isi**$(i) \leq$ **ns**, for $i = 1, 2, \ldots,$ **n**, and more than **ip** values of **isi**$(i) > 0$.

9: **b(ip) – double array**

Initial estimates of the covariate coefficient parameters β. **b**(j) must contain the initial estimate of the coefficient of the covariate in **z** corresponding to the jth nonzero value of **isz**.

Suggested value: In many cases an initial value of zero for **b**(j) may be used. For other suggestions see Section 8.

10: **ndmax – int32 scalar**

Constraint: **ndmax** $\geq$ the number of distinct failure times. This is returned in **nd**.

11: **tol – double scalar**

Indicates the accuracy required for the estimation. Convergence is assumed when the decrease in deviance is less than **tol** $\times$ $(1.0 + \text{CurrentDeviance})$. This corresponds approximately to an absolute precision if the deviance is small and a relative precision if the deviance is large.

Constraint: **tol** $\geq 10 \times$ ***machine precision***.

12: **maxit – int32 scalar**

The maximum number of iterations to be used for computing the estimates. If **maxit** is set to 0 then the standard errors, score functions, variance-covariance matrix and the survival function are computed for the input value of β in **b** but β is not updated.

Constraint: **maxit** ≥ 0.

13: **iprint – int32 scalar**

Indicates if the printing of information on the iterations is required.

iprint ≤ 0

No printing.

iprint ≥ 1

The deviance and the current estimates are printed every **iprint** iterations. When printing occurs the output is directed to the current advisory message unit (see x04ab).

5.2 Optional Input Parameters

1: **n – int32 scalar**

n, the number of data points.

Constraint: **n** ≥ 2.

2: **m – int32 scalar**

the number of covariates in array **z**.

Constraint: **m** ≥ 1.

3: **ip – int32 scalar**

Default: The dimension of the arrays **b**, **se**, **sc**. (An error is raised if these dimensions are not equal.)

the number of covariates included in the model as indicated by **isz**.

Constraint: **ip** $=$ number of nonzero values of **isz**.

5.3 Input Parameters Omitted from the MATLAB Interface

ldz, wk, iwk

5.4 Output Parameters

1: **dev – double scalar**

The deviance, that is $-2 \times$ (maximized log marginal likelihood).

2: **b(ip) – double array**

Suggested value: In many cases an initial value of zero for $\mathbf{b}(j)$ may be used. For other suggestions see Section 8.

$\mathbf{b}(j)$ contains the estimate $\hat{\beta}_i$, the coefficient of the covariate stored in the ith column of **z** where i is the jth nonzero value in the array **isz**.

3: **se(ip) – double array**

$\mathbf{se}(j)$ is the asymptotic standard error of the estimate contained in $\mathbf{b}(j)$ and score function in $\mathbf{sc}(j)$, for $j = 1, 2, \ldots, \mathbf{ip}$.

4: **sc(ip) – double array**

$\mathbf{sc}(j)$ is the value of the score function, $U_j(\beta)$, for the estimate contained in $\mathbf{b}(j)$.

5: **cov(ip × (ip + 1)/2) – double array**

The variance-covariance matrix of the parameter estimates in **b** stored in packed form by column, i.e., the covariance between the parameter estimates given in $\mathbf{b}(i)$ and $\mathbf{b}(j)$, $j \geq i$, is stored in $\mathbf{cov}(j(j-1)/2 + i)$.

6: **res(n) – double array**

The residuals, $r(t_l)$, for $l = 1, 2, \ldots, \mathbf{n}$.

7: **nd – int32 scalar**

The number of distinct failure times.

8: **tp(ndmax) – double array**

$\mathbf{tp}(i)$ contains the ith distinct failure time, for $i = 1, 2, \ldots, \mathbf{nd}$.

9: **sur(ndmax,∗) – double array**

The first dimension of the array **sur** must be at least the number of distinct failure times. This is returned in **nd**

The second dimension of the array must be at least $\max(\mathbf{ns}, 1)$

If $\mathbf{ns} = 0$, $\mathbf{sur}(i, 1)$ contains the estimated survival function for the ith distinct failure time.

If $\mathbf{ns} > 0$, $\mathbf{sur}(i, k)$ contains the estimated survival function for the ith distinct failure time in the kth stratum.

10: **ifail – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **offset** ≠ 'Y' or 'N',
or **m** < 1,
or **n** < 2,
or **ns** < 0,
or **ldz** < **n**,
or **tol** < 10 × ***machine precision***,
or **maxit** < 0.

ifail $= 2$

On entry, **isz**$(i) < 0$ for some i,
or the value of **ip** is incompatible with **isz**,
or **ic**$(i) \neq 1$ or 0.
or **isi**$(i) < 0$ or **isi**$(i) >$ **ns**,
or number of values of **isz**$(i) > 0$ is greater than or equal to n_0, the number of observations excluding any with **isi**$(i) = 0$,
or all observations are censored, i.e., **ic**$(i) = 1$ for all i,
or **ndmax** is too small.

ifail $= 3$

The matrix of second partial derivatives is singular. Try different starting values or include fewer covariates.

ifail $= 4$

Overflow has been detected. Try using different starting values.

ifail $= 5$

Convergence has not been achieved in **maxit** iterations. The progress toward convergence can be examined by using a nonzero value of **iprint**. Any non-convergence may be due to a linear combination of covariates being monotonic with time.

Full results are returned.

ifail $= 6$

In the current iteration 10 step halvings have been performed without decreasing the deviance from the previous iteration. Convergence is assumed.

7 Accuracy

The accuracy is specified by **tol**.

8 Further Comments

g12ba uses mean centering which involves subtracting the means from the covariables prior to computation of any statistics. This helps to minimize the effect of outlying observations and accelerates convergence.

If the initial estimates are poor then there may be a problem with overflow in calculating $\exp\left(\beta^{\mathrm{T}} z_i\right)$ or there may be non-convergence. Reasonable estimates can often be obtained by fitting an exponential model using g02gc.

9 Example

```
offset = 'No-offset';
ns = int32(0);
z = [0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
```

```
      0;
      0;
      0;
      0;
      0;
      0;
      0;
      0;
      0;
      0;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1;
      1];
isz = [int32(1)];
t = [1;
      1;
      2;
      2;
      3;
      4;
      4;
      5;
      5;
      8;
      8;
      8;
      8;
      11;
      11;
      12;
      12;
      15;
      17;
      22;
      23;
      6;
      6;
      6;
      7;
      10;
      13;
      16;
      22;
      23;
      6;
      9;
      10;
      11;
      17;
      19;
      20;
      25;
```

```
     32;
     32;
     34;
     35];
ic = [int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1)];
omega = [0];
isi = [int32(0)];
b = [-1.526606964258046];
ndmax = int32(42);
tol = 5e-05;
maxit = int32(20);
iprint = int32(0);
[dev, bOut, se, sc, cov, res, nd, tp, sur, ifail] = ...
     g12ba(offset, ns, z, isz, t, ic, omega, isi, b, ndmax, tol, maxit,
iprint)
```

```
dev =
  172.7592
bOut =
   -1.5091
se =
    0.4096
sc =
   3.3775e-04
cov =
    0.1677
res =
    0.0780
    0.0780
```

```
    0.1626
    0.1626
    0.2088
    0.3057
    0.3057
    0.4130
    0.4130
    0.9141
    0.9141
    0.9141
    0.9141
    1.1864
    1.1864
    1.4175
    1.4175
    1.7233
    2.0993
    2.6630
    3.5226
    0.1312
    0.1312
    0.1312
    0.1452
    0.2216
    0.3466
    0.4217
    0.5888
    0.7789
    0.1312
    0.2021
    0.2216
    0.2623
    0.4642
    0.4642
    0.4642
    0.7789
    0.7789
    0.7789
    0.7789
    0.7789
nd =
          17
tp =
     1
     2
     3
     4
     5
     6
     7
     8
    10
    11
    12
    13
    15
    16
    17
    22
    23
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
```

```
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
sur =
    0.9640
    0.9264
    0.9065
    0.8661
    0.8235
    0.7566
    0.7343
    0.6506
    0.6241
    0.5724
    0.5135
    0.4784
    0.4447
    0.4078
    0.3727
    0.2859
    0.1908
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
ifail =
          0
```