NAG Toolbox

nag_tsa_multi_varma_forecast (g13dj)

1 Purpose

nag_tsa_multi_varma_forecast (g13dj) computes forecasts of a multivariate time series. It is assumed that a vector ARMA model has already been fitted to the appropriately differenced/transformed time series using nag_tsa_multi_varma_estimate (g13dd). The standard deviations of the forecast errors are also returned. A reference vector is set up so that, should future series values become available, the forecasts and their standard errors may be updated by calling nag_tsa_multi_varma_update (g13dk).

2 Syntax

```
[qq, predz, sefz, ref, ifail] = nag_tsa_multi_varma_forecast(z, tr, id, delta,
ip, iq, mean, par, qq, v, lmax, lref, 'k', k, 'n', n, 'lpar', lpar)
```

```
[qq, predz, sefz, ref, ifail] = g13dj(z, tr, id, delta, ip, iq, mean, par, qq,
v, lmax, lref, 'k', k, 'n', n, 'lpar', lpar)
```

3 Description

Let the vector $Z_t = (z_{1t}, z_{2t}, \ldots, z_{kt})^{\mathrm{T}}$, for $t = 1, 2, \ldots, n$, denote a k-dimensional time series for which forecasts of $Z_{n+1}, Z_{n+2}, \ldots, Z_{n+l_{\max}}$ are required. Let $W_t = (w_{1t}, w_{2t}, \ldots, w_{kt})^{\mathrm{T}}$ be defined as follows:

$$w_{it} = \delta_i(B) z_{it}^*, \quad i = 1, 2, \ldots, k,$$

where $\delta_i(B)$ is the differencing operator applied to the ith series and where z_{it}^* is equal to either z_{it}, $\sqrt{z_{it}}$ or $\log_e(z_{it})$ depending on whether or not a transformation was required to stabilize the variance before fitting the model.

If the order of differencing required for the ith series is d_i, then the differencing operator for the ith series is defined by $\delta_i(B) = 1 - \delta_{i1}B - \delta_{i2}B^2 - \cdots - \delta_{id_i}B^{d_i}$ where B is the backward shift operator; that is, $BZ_t = Z_{t-1}$. The differencing parameters δ_{ij}, for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, d_i$, must be supplied by you. If the ith series does not require differencing, then $d_i = 0$.

W_t is assumed to follow a multivariate ARMA model of the form:

$$W_t - \mu = \phi_1(W_{t-1} - \mu) + \phi_2(W_{t-2} - \mu) + \cdots + \phi_p(W_{t-p} - \mu) + \epsilon_t - \theta_1\epsilon_{t-1} - \cdots - \theta_q\epsilon_{t-q}, \quad (1)$$

where $\epsilon_t = (\epsilon_{1t}, \epsilon_{2t}, \ldots, \epsilon_{kt})^{\mathrm{T}}$, for $t = 1, 2, \ldots, n$, is a vector of k residual series assumed to be Normally distributed with zero mean and positive definite covariance matrix Σ. The components of ϵ_t are assumed to be uncorrelated at non-simultaneous lags. The ϕ_i and θ_j are k by k matrices of parameters. The matrices ϕ_i, for $i = 1, 2, \ldots, p$, are the autoregressive (AR) parameter matrices, and the matrices θ_i, for $i = 1, 2, \ldots, q$, the moving average (MA) parameter matrices. The parameters in the model are thus the p (k by k) ϕ-matrices, the q (k by k) θ-matrices, the mean vector μ and the residual error covariance matrix Σ. The ARMA model (1) must be both stationary and invertible; see nag_tsa_uni_arma_roots (g13dx) for a method of checking these conditions.

The ARMA model (1) may be rewritten as

$$\phi(B)\big(\delta(B)Z_t^* - \mu\big) = \theta(B)\epsilon_t,$$

where $\phi(B)$ and $\theta(B)$ are the autoregressive and moving average polynomials and $\delta(B)$ denotes the k by k diagonal matrix whose ith diagonal elements is $\delta_i(B)$ and $Z_t^* = \left(z_{1t}^*, z_{2t}^* \ldots z_{kt}^*\right)^{\mathrm{T}}$.

This may be rewritten as

$$\phi(B)\delta(B)Z_t^* = \phi(B)\mu + \theta(B)\epsilon_t$$

or

$$Z_t^* = \tau + \psi(B)\epsilon_t = \tau + \epsilon_t + \psi_1\epsilon_{t-1} + \psi_2\epsilon_{t-2} + \cdots$$

where $\psi(B) = \delta^{-1}(B)\phi^{-1}(B)\theta(B)$ and $\tau = \delta^{-1}(B)\mu$ is a vector of length k.

Forecasts are computed using a multivariate version of the procedure described in Box and Jenkins (1976). If $\hat{Z}_n^*(l)$ denotes the forecast of Z_{n+l}^*, then $\hat{Z}_n^*(l)$ is taken to be that linear function of $Z_n^*, Z_{n-1}^*, \ldots$ which minimizes the elements of $E\{e_n(l)e_n'(l)\}$ where $e_n(l) = Z_{n+l}^* - \hat{Z}_n^*(l)$ is the forecast error. $\hat{Z}_n^*(l)$ is referred to as the linear minimum mean square error forecast of Z_{n+l}^*.

The linear predictor which minimizes the mean square error may be expressed as

$$\hat{Z}_n^*(l) = \tau + \psi_l\epsilon_n + \psi_{l+1}\epsilon_{n-1} + \psi_{l+2}\epsilon_{n-2} + \cdots.$$

The forecast error at t for lead l is then

$$e_n(l) = Z_{n+l}^* - \hat{Z}_n^*(l) = \epsilon_{n+l} + \psi_1\epsilon_{n+l-1} + \psi_2\epsilon_{n+l-2} + \cdots + \psi_{l-1}\epsilon_{n+1}.$$

Let $d = \max(d_i)$, for $i = 1, 2, \ldots, k$. Unless $q = 0$ the function requires estimates of ϵ_t, for $t = d + 1, \ldots, n$, which are obtainable from nag_tsa_multi_varma_estimate (g13dd). The terms ϵ_t are assumed to be zero, for $t = n + 1, \ldots, n + l_{\max}$. You may use nag_tsa_multi_varma_update (g13dk) to update these $l_{\max}$ forecasts should further observations, $Z_{n+1}, Z_{n+2}, \ldots$, become available. Note that when $l_{\max}$ or more further observations are available then nag_tsa_multi_varma_forecast (g13dj) must be used to produce new forecasts for $Z_{n+l_{\max}+1}, Z_{n+l_{\max}+2}, \ldots$, should they be required.

When a transformation has been used the forecasts and their standard errors are suitably modified to give results in terms of the original series, Z_t; see Granger and Newbold (1976).

4 References

Box G E P and Jenkins G M (1976) *Time Series Analysis: Forecasting and Control* (Revised Edition) Holden–Day

Granger C W J and Newbold P (1976) Forecasting transformed series *J. Roy. Statist. Soc. Ser. B* **38** 189–203

Wei W W S (1990) *Time Series Analysis: Univariate and Multivariate Methods* Addison–Wesley

5 Parameters

The quantities **k**, **n**, **kmax**, **ip**, **iq**, **par**, **npar**, **qq** and **v** from nag_tsa_multi_varma_estimate (g13dd) are suitable for input to nag_tsa_multi_varma_forecast (g13dj).

5.1 Compulsory Input Parameters

1: **z**($kmax$, **n**) – REAL (KIND=nag_wp) array

$kmax$, the first dimension of the array, must satisfy the constraint $kmax \geq$ **k**.

z(i, t) must contain, z_{it}, the ith component of Z_t, for $i = 1, 2, \ldots, k$ and $t = 1, 2, \ldots, n$.

Constraints:

if **tr**(i) = 'L', **z**$(i, t) > 0.0$;
if **tr**(i) = 'S', **z**$(i, t) \geq 0.0$, for $i = 1, 2, \ldots, k$ and $t = 1, 2, \ldots, n$.

2: **tr**(**k**) – CHARACTER(1) array

tr(i) indicates whether the ith time series is to be transformed, for $i = 1, 2, \ldots, k$.

tr(i) = 'N'
No transformation is used.

tr(i) = 'L'
A log transformation is used.

tr(i) = 'S'
A square root transformation is used.

Constraint: **tr**(i) = 'N', 'L' or 'S', for $i = 1, 2, \ldots, k$.

3: **id**(**k**) – INTEGER array

id(i) must specify, d_i, the order of differencing required for the ith series.

Constraint: $0 \leq$ **id**$(i) <$ **n** $-$ max(**ip**, **iq**), for $i = 1, 2, \ldots, k$.

4: **delta**$(kmax, :)$ – REAL (KIND=nag_wp) array

The first dimension of the array **delta** must be at least **k**.

The second dimension of the array **delta** must be at least $\max(1, d)$, where $d = \max($**id**$(i))$.

If **id**$(i) > 0$, then **delta**(i, j) must be set equal to δ_{ij}, for $j = 1, 2, \ldots, d_i$ and $i = 1, 2, \ldots, k$.

If $d = 0$, **delta** is not referenced.

5: **ip** – INTEGER

p, the number of AR parameter matrices.

Constraint: **ip** ≥ 0.

6: **iq** – INTEGER

q, the number of MA parameter matrices.

Constraint: **iq** ≥ 0.

7: **mean_p** – CHARACTER(1)

mean = 'M', if components of μ have been estimated and **mean** = 'Z', if all elements of μ are to be taken as zero.

Constraint: **mean** = 'M' or 'Z'.

8: **par**(**lpar**) – REAL (KIND=nag_wp) array

Must contain the parameter estimates read in row by row in the order $\phi_1, \phi_2, \ldots, \phi_p$, $\theta_1, \theta_2, \ldots, \theta_q$, μ.

Thus,

if **ip** > 0, **par**$((l-1) \times k \times k + (i-1) \times k + j)$ must be set equal to an estimate of the (i, j)th element of ϕ_l, for $l = 1, 2, \ldots, p$, $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, k$;

if **iq** > 0, **par**$(p \times k \times k + (l-1) \times k \times k + (i-1) \times k + j)$ must be set equal to an estimate of the (i, j)th element of θ_l, for $l = 1, 2, \ldots, q$, $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, k$;

if **mean** = 'M', **par**$((p+q) \times k \times k + i)$ must be set equal to an estimate of the ith component of μ, for $i = 1, 2, \ldots, k$.

Constraint: the first **ip** $\times$ **k** $\times$ **k** elements of **par** must satisfy the stationarity condition and the next **iq** $\times$ **k** $\times$ **k** elements of **par** must satisfy the invertibility condition.

9: **qq**$(kmax,$ **k**$)$ – REAL (KIND=nag_wp) array

$kmax$, the first dimension of the array, must satisfy the constraint $kmax \geq$ **k**.

qq(i, j) must contain an estimate of the (i, j)th element of Σ. The lower triangle only is needed.

Constraint: **qq** must be positive definite.

10: **v**$(kmax, :)$ – REAL (KIND=nag_wp) array

The first dimension of the array **v** must be at least **k**.

The second dimension of the array **v** must be at least $\max(1, \mathbf{n} - d)$, where $d = \max(\mathbf{id}(i))$.

$\mathbf{v}(i, t)$ must contain an estimate of the ith component of ϵ_{t+d}, for $i = 1, 2, \ldots, k$ and $t = 1, 2, \ldots, n - d$.

If $q = 0$, **v** is not used.

11: **lmax** – INTEGER

The number, $l_{\max}$, of forecasts required.

Constraint: $\mathbf{lmax} \geq 1$.

12: **lref** – INTEGER

The dimension of the array **ref**.

Constraint: $\mathbf{lref} \geq (\mathbf{lmax} - 1) \times \mathbf{k} \times \mathbf{k} + 2 \times \mathbf{k} \times \mathbf{lmax} + \mathbf{k}$.

5.2 Optional Input Parameters

1: **k** – INTEGER

Default: the dimension of the arrays **tr**, **id** and the first dimension of the arrays **z**, **delta**, **qq**, **v** and the second dimension of the array **qq**. (An error is raised if these dimensions are not equal.)

k, the dimension of the multivariate time series.

Constraint: $\mathbf{k} \geq 1$.

2: **n** – INTEGER

Default: the second dimension of the array **z**.

n, the number of observations in the series, Z_t, prior to differencing.

Constraint: $\mathbf{n} \geq 3$.

The total number of observations must exceed the total number of parameters in the model; that is

if **mean** = 'Z', $\mathbf{n} \times \mathbf{k} > (\mathbf{ip} + \mathbf{iq}) \times \mathbf{k} \times \mathbf{k} + \mathbf{k} \times (\mathbf{k} + 1)/2$;

if **mean** = 'M', $\mathbf{n} \times \mathbf{k} > (\mathbf{ip} + \mathbf{iq}) \times \mathbf{k} \times \mathbf{k} + \mathbf{k} + \mathbf{k} \times (\mathbf{k} + 1)/2$,

(see the arguments **ip**, **iq** and **mean**).

3: **lpar** – INTEGER

Default: the dimension of the array **par**.

The dimension of the array **par**.

Constraints:

if **mean** = 'Z', $\mathbf{lpar} \geq \max(1, (\mathbf{ip} + \mathbf{iq}) \times \mathbf{k} \times \mathbf{k})$;
if **mean** = 'M', $\mathbf{lpar} \geq (\mathbf{ip} + \mathbf{iq}) \times \mathbf{k} \times \mathbf{k} + \mathbf{k}$.

5.3 Output Parameters

1: **qq**$(kmax, \mathbf{k})$ – REAL (KIND=nag_wp) array

If $\mathbf{ifail} \neq 1$, then the upper triangle is set equal to the lower triangle.

2: **predz**($kmax$, **lmax**) – REAL (KIND=nag_wp) array

predz(i, l) contains the forecast of $z_{i,n+l}$, for $i = 1, 2, \ldots, k$ and $l = 1, 2, \ldots, l_{\max}$.

3: **sefz**($kmax$, **lmax**) – REAL (KIND=nag_wp) array

sefz(i, l) contains an estimate of the standard error of the forecast of $z_{i,n+l}$, for $i = 1, 2, \ldots, k$ and $l = 1, 2, \ldots, l_{\max}$.

4: **ref**(**lref**) – REAL (KIND=nag_wp) array

The reference vector which may be used to update forecasts using nag_tsa_multi_varma_update (g13dk). The first $(\mathbf{lmax} - 1) \times \mathbf{k} \times \mathbf{k}$ elements contain the ψ weight matrices, $\psi_1, \psi_2, \ldots, \psi_{l_{\max}-1}$. The next $\mathbf{k} \times \mathbf{lmax}$ elements contain the forecasts of the transformed series $\hat{Z}^*_{n+1}, \hat{Z}^*_{n+2}, \ldots, \hat{Z}^*_{n+l_{\max}}$ and the next $\mathbf{k} \times \mathbf{lmax}$ contain the variances of the forecasts of the transformed variables. The last **k** elements are used to store the transformations for the series.

5: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, $\mathbf{k} < 1$,
or $\mathbf{n} < 3$,
or $kmax < \mathbf{k}$,
or $\mathbf{id}(i) < 0$ for some $i = 1, 2, \ldots, k$,
or $\mathbf{id}(i) \geq \mathbf{n} - \max(\mathbf{ip}, \mathbf{iq})$ for some $i = 1, 2, \ldots, k$,
or $\mathbf{ip} < 0$,
or $\mathbf{iq} < 0$,
or **mean** $\neq$ 'M' or 'Z',
or $\mathbf{lpar} < (\mathbf{ip} + \mathbf{iq}) \times \mathbf{k} \times \mathbf{k} + \mathbf{k}$, and **mean** $=$ 'M',
or $\mathbf{lpar} < (\mathbf{ip} + \mathbf{iq}) \times \mathbf{k} \times \mathbf{k}$ and **mean** $=$ 'Z',
or $\mathbf{n} \times \mathbf{k} \leq (\mathbf{ip} + \mathbf{iq}) \times \mathbf{k} \times \mathbf{k} + \mathbf{k} + \mathbf{k}(\mathbf{k} + 1)/2$, and **mean** $=$ 'M',
or $\mathbf{n} \times \mathbf{k} \leq (\mathbf{ip} + \mathbf{iq}) \times \mathbf{k} \times \mathbf{k} + \mathbf{k}(\mathbf{k} + 1)/2$ and **mean** $=$ 'Z',
or $\mathbf{lmax} < 1$,
or $\mathbf{lref} < (\mathbf{lmax} - 1) \times \mathbf{k} \times \mathbf{k} + 2 \times \mathbf{k} \times \mathbf{lmax} + \mathbf{k}$,
or $lwork$ is too small,
or $liwork$ is too small.

ifail $= 2$

On entry, at least one of the first k elements of **tr** is not equal to `N', `L' or `S'.

ifail $= 3$

On entry, one or more of the transformations requested cannot be computed; that is, you may be trying to log or square-root a series, some of whose values are negative.

ifail $= 4$

On entry, either **qq** is not positive definite or the autoregressive parameter matrices are extremely close to or outside the stationarity region, or the moving average parameter matrices are extremely close to or outside the invertibility region. To proceed, you must supply different parameter estimates in the arrays **par** and **qq**.

ifail $= 5$

This is an unlikely exit brought about by an excessive number of iterations being needed to evaluate the eigenvalues of the matrices required to check for stationarity and invertibility; see nag_tsa_uni_arma_roots (g13dx). All output arguments are undefined.

ifail $= 6$

This is an unlikely exit which could occur if **qq** is nearly non positive definite. In this case the standard deviations of the forecast errors may be non-positive. To proceed, you must supply different parameter estimates in the array **qq**.

ifail $= 7$

This is an unlikely exit. For one of the series, overflow will occur if the forecasts are computed. You should check whether the transformations requested in the array **tr** are sensible. All output arguments are undefined.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

The matrix computations are believed to be stable.

8 Further Comments

The same differencing operator does not have to be applied to all the series. For example, suppose we have $k = 2$, and wish to apply the second order differencing operator ∇^2 to the first series and the first-order differencing operator ∇ to the second series:

$$w_{1t} = \nabla^2 z_{1t} = (1-B)^2 z_{1t} = \left(1 - 2B + B^2\right) Z_{1t}, \qquad \text{and}$$
$$w_{2t} = \nabla z_{2t} = (1-B) z_{2t}.$$

Then $d_1 = 2, d_2 = 1,\ d = \max(d_1, d_2) = 2$, and

$$\textbf{delta} = \begin{bmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 1 & \end{bmatrix}.$$

Note: although differencing may already have been applied prior to the model fitting stage, the differencing parameters supplied in **delta** are part of the model definition and are still required by this function to produce the forecasts.

nag_tsa_multi_varma_forecast (g13dj) should not be used when the moving average parameters lie close to the boundary of the invertibility region. The function does test for both invertibility and stationarity but if in doubt, you may use nag_tsa_uni_arma_roots (g13dx), before calling this function, to check that the VARMA model being used is invertible.

On a successful exit, the quantities **k**, **lmax**, $kmax$, **ref** and **lref** will be suitable for input to nag_tsa_multi_varma_update (g13dk).

9 Example

This example computes forecasts of the next five values in two series each of length 48. No transformation is to be used and no differencing is to be applied to either of the series. nag_tsa_multi_varma_estimate (g13dd) is first called to fit an AR(1) model to the series. The mean vector μ is to be estimated and $\phi_1(2,1)$ constrained to be zero.

9.1 Program Text

```
     function g13dj_example

fprintf('g13dj example results\n\n');

% Series
z = [-1.490 -1.620  5.200  6.230  6.210  5.860  4.090  3.180  ...
       2.620  1.490  1.170  0.850 -0.350  0.240  2.440  2.580  ...
       2.040  0.400  2.260  3.340  5.090  5.000  4.780  4.110  ...
       3.450  1.650  1.290  4.090  6.320  7.500  3.890  1.580  ...
       5.210  5.250  4.930  7.380  5.870  5.810  9.680  9.070  ...
       7.290  7.840  7.550  7.320  7.970  7.760  7.000  8.350;
       7.340  6.350  6.960  8.540  6.620  4.970  4.550  4.810  ...
       4.750  4.760 10.880 10.010 11.620 10.360  6.400  6.240  ...
       7.930  4.040  3.730  5.600  5.350  6.810  8.270  7.680  ...
       6.650  6.080 10.250  9.140 17.750 13.300  9.630  6.800  ...
       4.080  5.060  4.940  6.650  7.940 10.760 11.890  5.850  ...
       9.010  7.500 10.020 10.380  8.150  8.370 10.730 12.140];
[k,n] = size(z);

% Difference /transform series
tr    = {'N'; 'N'};
id    = [nag_int(0);0];
delta = [0;  0];
[w, nd, ifail] = g13dl( ...
 z, tr, id, delta);

% VARMA info
ip    = nag_int(1);
iq    = nag_int(0);
mean_p = true;

% Initial parameter estimates and free parameter flags
par    = zeros(6, 1);
parhld = [false;  false;  true;  false;  false;  false];

% Exact likelihood
exact  = true;
% control parameters
iprint = nag_int(-1);
cgetol = 0.0001;
ishow  = nag_int(0);

qq = [0, 0; 0, 0];

% Fit VARMA
[par, qq, ~, ~, v, ~, ~, ifail] = ...
  g13dd( ...
  ip, iq, mean_p, par, qq, w, parhld, exact, iprint, cgetol, ...
  ishow, 'n', nd);

% Perform forecast
lmax = nag_int(5);
lref = nag_int(150);
mean_p = 'M';
[qq, predz, sefz, ref, ifail] = ...
  g13dj( ...
  z, tr, id, delta, ip, iq, mean_p, par, qq, v, lmax, lref);

% Display results
fprintf(' Forecast Summary Table\n');
```

```
fprintf(' ----------------------\n\n');
fprintf(' Forecast origin is set at t = %4d\n\n', n);
loop = lmax/5;
if mod(lmax,5)~=0
  loop = loop + 1;
end
for j = 1:loop
  i2 = (j-1)*5;
  l2 = min(i2+5,lmax);
  fprintf('Lead Time %14s',' ');
  fprintf('%7d', [i2+1:l2]);
  fprintf('\n\n');
  for i = 1:k
    fprintf('Series %d : Forecast       ', i);
    fprintf('%7.2f',predz(i,i2+1:l2));
    fprintf('\n%8s : Standard Error ', ' ');
    fprintf('%7.2f',sefz(i,i2+1:l2));
    fprintf('\n');
  end
end
```

9.2 Program Results

```
     g13dj example results

 Forecast Summary Table
 ----------------------

 Forecast origin is set at t =   48

Lead Time                     1      2      3      4      5

Series 1 : Forecast         7.82   7.28   6.77   6.33   5.95
         : Standard Error   1.72   2.23   2.51   2.68   2.79
Series 2 : Forecast        10.31   9.25   8.65   8.30   8.10
         : Standard Error   2.32   2.68   2.78   2.82   2.83
```