NAG Toolbox

nag_tsa_multi_inputmod_forecast_state (g13bh)

1 Purpose

nag_tsa_multi_inputmod_forecast_state (g13bh) produces forecasts of a time series (the output series) which depends on one or more other (input) series via a multi-input model which will usually have been fitted using nag_tsa_multi_inputmod_estim (g13be). The future values of the input series must be supplied. The original observations are not required. nag_tsa_multi_inputmod_forecast_state (g13bh) uses as input either the original state set obtained from nag_tsa_multi_inputmod_estim (g13be), or the state set updated by a series of new observations from nag_tsa_multi_inputmod_update (g13bg). Standard errors of the forecasts are produced. If future values of some of the input series have been obtained as forecasts using ARIMA models for those series, this may be allowed for in the calculation of the standard errors.

2 Syntax

```
[xxyn, mrx, fva, fsd, ifail] = nag_tsa_multi_inputmod_forecast_state(sttf, mr,
mt, para, xxyn, mrx, parx, rmsxy, kzef, 'nsttf', nsttf, 'nser', nser, 'npara',
npara, 'nfv', nfv)
```

```
[xxyn, mrx, fva, fsd, ifail] = g13bh(sttf, mr, mt, para, xxyn, mrx, parx, rmsxy,
kzef, 'nsttf', nsttf, 'nser', nser, 'npara', npara, 'nfv', nfv)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 22: **nfv** was made optional.

3 Description

The forecasts of the output series y_t are calculated, for $t = n+1, \ldots, n+L$, where n is the latest time point of the observations used to produce the state set and L is the maximum lead time of the forecasts.

First the new input series values x_t are used to form the input components z_t, for $t = n+1, \ldots, n+L$, using the transfer function models:

(a) $z_t = \delta_1 z_{t-1} + \delta_2 z_{t-2} + \cdots + \delta_p z_{t-p} + \omega_0 x_{t-b} - \omega_1 x_{t-b-1} - \cdots - \omega_q x_{t-b-q}$.

The output noise component n_t is then forecast by setting $a_t = 0$, for $t = n+1, \ldots, n+L$, and using the ARIMA model equations:

(b) $e_t = \phi_1 e_{t-1} + \phi_2 e_{t-2} + \cdots + \phi_p e_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_1 a_{t-q}$

(c) $w_t = \Phi_1 w_{t-s} + \Phi_2 w_{t-2\times s} + \cdots + \Phi_P w_{t-P\times s} + e_t - \Theta_1 e_{t-s} - \Theta_2 e_{t-2\times s} - \cdots - \Theta_Q e_{t-Q\times s}$

(d) $n_t = \left(\nabla^d \nabla_s^D\right)^{-1}(w_t + c)$.

This last step of 'integration' reverses the process of differencing. Finally the output forecasts are calculated as

$$y_t = z_{1,t} + z_{2,t} + \cdots + z_{m,t} + n_t.$$

The forecast error variance of y_{t+l} (i.e., at lead time l) is S_l^2, which is the sum of parts which arise from the various input series, and the output noise component. That part due to the output noise is

$$sn_l^2 = V_n \times \left(\psi_0^2 + \psi_1^2 + \cdots + \psi_{l-1}^2\right),$$

where V_n is the estimated residual variance of the output noise ARIMA model, and $\psi_0, \psi_1, \ldots$ are the 'psi-weights' of this model as defined in Box and Jenkins (1976). They are calculated by applying the equations (b), (c) and (d) above, for $t = 0, 1, \ldots, L$, but with artificial values for the various series and with the constant c set to 0. Thus all values of a_t, e_t, w_t and n_t are taken as zero, for $t < 0$; a_t is taken to be 1, for $t = 0$ and 0, for $t > 0$. The resulting values of n_t, for $t = 0, 1, \ldots, L$, are precisely $\psi_0, \psi_1, \ldots, \psi_L$ as required.

Further contributions to S_l^2 come only from those input series, for which future values are forecasts which have been obtained by applying input series ARIMA models. For such a series the contribution is

$$sz_l^2 = V_x \times \left(\nu_0^2 + \nu_1^2 + \cdots + \nu_{l-1}^2\right),$$

where V_x is the estimated residual variance of the input series ARIMA model. The coefficients $\nu_0, \nu_1, \ldots$ are calculated by applying the transfer function model equation (a) above, for $t = 0, 1, \ldots, L$, but again with artificial values of the series. Thus all values of z_t and x_t, for $t < 0$, are taken to be zero, and $x_0, x_1, \ldots$ are taken to be the psi-weight sequence $\psi_0, \psi_1, \ldots$ for the **input series** ARIMA model. The resulting values of z_t, for $t = 0, 1, \ldots, L$, are precisely $\nu_0, \nu_1, \ldots, \nu_L$ as required.

In adding such contributions sz_l^2 to sn_l^2 to make up the total forecast error variance S_l^2, it is assumed that the various input series with which these contributions are associated are statistically independent of each other.

When using the function in practice an ARIMA model is required for all the input series. In the case of those inputs for which no such ARIMA model is available (or its effects are to be excluded), the corresponding orders and parameters and the estimated residual variance should be set to zero.

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

1: **stff**(**nsttf**) – REAL (KIND=nag_wp) array

The **nsttf** values in the state set as returned by nag_tsa_multi_inputmod_estim (g13be) or nag_tsa_multi_inputmod_update (g13bg).

2: **mr**(**7**) – INTEGER array

The orders vector (p, d, q, P, D, Q, s) of the ARIMA model for the output noise component.

p, q, P and Q give respectively the number of autoregressive (ϕ), moving average (θ), seasonal autoregressive (Φ) and seasonal moving average (Θ) parameters.

d, D and s refer respectively to the order of non-seasonal differencing, the order of seasonal differencing, and the seasonal period.

Constraints:

p, d, q, P, D, Q, $s \geq 0$;
$p + q + P + Q > 0$;
$s \neq 1$;
if $s = 0$, $P + D + Q = 0$;
if $s > 1$, $P + D + Q > 0$.

3: **mt**(**4**, **nser**) – INTEGER array

The transfer function orders b, p and q of each of the input series. The data for input series i are held in column i. Row 1 holds the value b_i, row 2 holds the value q_i and row 3 holds the value p_i. For a simple input, $b_i = q_i = p_i = 0$.

Row 4 holds the value r_i, where $r_i = 1$ for a simple input, $r_i = 2$ or 3 for a transfer function input. When $r_i = 1$, any nonzero contents of rows 1, 2 and 3 of column i are ignored. The choice of $r_i = 2$ or $r_i = 3$ is an option for use in model estimation and does not affect the operation of nag_tsa_multi_inputmod_forecast_state (g13bh).

Constraint: $\mathbf{mt}(4, i) = 1$, 2 or 3, for $i = 1, 2, \ldots, \mathbf{nser} - 1$.

4: **para**(**npara**) – REAL (KIND=nag_wp) array

Estimates of the multi-input model parameters as returned by nag_tsa_multi_inputmod_estim (g13be). These are in order, firstly the ARIMA model parameters: p values of ϕ parameters, q values of θ parameters, P values of Φ parameters and Q values of Θ parameters. These are followed by the transfer function model parameter values $\omega_0, \omega_1, \ldots, \omega_{q_1}$, $\delta_1, \delta_2, \ldots, \delta_{p_1}$ for the first of any input series and similar sets of values for any subsequent input series. The final component of **para** is the constant c.

5: **xxyn**($ldxxyn$, **nser**) – REAL (KIND=nag_wp) array

$ldxxyn$, the first dimension of the array, must satisfy the constraint $ldxxyn \geq \mathbf{nfv}$.

The supplied **nfv** values for each of the input series required to produce the **nfv** output series forecasts. Column i contains the values for input series i. Column **nser** need not be supplied.

6: **mrx**(**7**, **nser**) – INTEGER array

The orders array for each of the input series ARIMA models. Thus, column i contains values of p, d, q, P, D, Q, s for input series i. In the case of those inputs for which no ARIMA model is available, the corresponding orders should be set to 0. (The model for any input series only affects the standard errors of the forecast values.)

7: **parx**($ldparx$, **nser**) – REAL (KIND=nag_wp) array

$ldparx$, the first dimension of the array, must satisfy the constraint $ldparx \geq ncd$, where ncd is the maximum number of parameters in any of the input series ARIMA models. If there are no input series, $ldparx \geq 1$.

Values of the parameters (ϕ, θ, Φ and Θ) for each of the input series ARIMA models. Thus column i contains $\mathbf{mrx}(1, i)$ values of ϕ parameters, $\mathbf{mrx}(3, i)$ values of θ parameters, $\mathbf{mrx}(4, i)$ values of Φ parameters and $\mathbf{mrx}(6, i)$ values of Θ parameters – in that order.

Values in the columns relating to those input series for which no ARIMA model is available are ignored. (The model for any input series only affects the standard errors of the forecast values.)

8: **rmsxy**(**nser**) – REAL (KIND=nag_wp) array

The estimated residual variances for each input series ARIMA model followed by that for the output noise ARIMA model. In the case of those inputs for which no ARIMA model is available, or when its effects are to be excluded in the calculation of forecast standard errors, the corresponding entry of **rmsxy** should be set to 0.

9: **kzef** – INTEGER

Must not be set to 0, if the values of the input component series z_t and the values of the output noise component n_t are to overwrite the contents of **xxyn** on exit, and must be set to 0 if **xxyn** is to remain unchanged on exit, apart from the appearance of the forecast values in column **nser**.

5.2 Optional Input Parameters

1: **nsttf** – INTEGER

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

The exact number of values in the state set array **sttf** as returned by nag_tsa_multi_inputmod_estim (g13be) or nag_tsa_multi_inputmod_update (g13bg).

2: **nser** – INTEGER

Default: the dimension of the arrays **mt**, **rmsxy** and the second dimension of the arrays **xxyn**, **parx**. (An error is raised if these dimensions are not equal.)

The total number of input and output series. There may be any number of input series (including none), but only one output series.

3: **npara** – INTEGER

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

The exact number of ϕ, θ, Φ, Θ, ω, δ and c parameters. (c must be included, whether its value was previously estimated or was set fixed).

4: **nfv** – INTEGER

Default: the first dimension of the array **xxyn**.

The number of forecast values required.

5.3 Output Parameters

1: **xxyn**($ldxxyn$, **nser**) – REAL (KIND=nag_wp) array

If **kzef** $= 0$, then column **nser** of **xxyn** contains the output series forecast values (as does **fva**), but **xxyn** is otherwise unchanged.

If **kzef** $\neq 0$, then the columns of **xxyn** hold the corresponding values of the forecast components z_t for each of the input series and the values of the output noise component n_t in that order.

2: **mrx**(**7**, **nser**) – INTEGER array

Unchanged, apart from column **nser** which is used for workspace.

3: **fva**(**nfv**) – REAL (KIND=nag_wp) array

The required forecast values for the output series.

4: **fsd**(**nfv**) – REAL (KIND=nag_wp) array

The standard errors for each of the forecast values.

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, **nsttf** is not consistent with the orders in arrays **mr** and **mt**.

ifail $= 2$

On entry, **npara** is not consistent with the orders in arrays **mr** and **mt**.

ifail $= 3$

On entry, $ldxxyn$ is too small.

ifail $= 4$

On entry, iwa is too small.

ifail $= 5$

On entry, $ldparx$ is too small.

ifail $= 6$

On entry, one of the r_i, stored in **mt**$(4, i)$, for $i = 1, 2, \ldots,$ **nser** $- 1$, does not equal 1, 2 or 3.

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 computations are believed to be stable.

8 Further Comments

The time taken by nag_tsa_multi_inputmod_forecast_state (g13bh) is approximately proportional to **nfv** $\times$ **npara**.

9 Example

This example follows up that described in nag_tsa_multi_inputmod_update (g13bg) and makes use of its data. These consist of output series orders and parameter values, input series transfer function orders and the updated state set.

Four new values of the input series are supplied, as are the orders and parameter values for the single input series ARIMA model (which has 2 values of ϕ, 2 values of θ, 1 value of Θ, single seasonal differencing and a seasonal period of 4), and the estimated residual variances for the input series ARIMA model and the output noise ARIMA model.

Four forecast values and their standard errors are computed and printed; also the values of the components z_t and the output noise component n_t corresponding to the forecasts.

9.1 Program Text

```
     function g13bh_example

fprintf('g13bh example results\n\n');

% orders and transfer function
mr = [nag_int(1);0;0;0;1;1;4];
mt = [nag_int(1),0;
                 0, 0;
                 1, 0;
                 3, 0];

% data
sttf = [  6.7160; 158.3022; -80.3352; -74.8937; -80.7694;
        -70.3022;   0.8476;  -2.0234;  -5.8080;  10.2943];
para = [0.5158; 0.9994; 8.6343; 0.6726; -0.3172];
xxyn = [6.923, 0; 6.939, 0; 6.705, 0; 6.914, 0];
mrx  = [nag_int(2),0;
                   0, 0;
                   2, 0;
                   0, 0;
```

```
                1, 0;
                1, 0;
                4, 0];
parx = [1.6743, 0;
       -0.9505, 0;
        1.4605, 0;
       -0.4862, 0;
        0.8993, 0];
rmsxy = [0.172; 22.9256];

kzef = nag_int(1);

% Produce forecasts
[xxyn, mrx, fva, fsd, ifail] = ...
  g13bh( ...
         sttf, mr, mt, para, xxyn, mrx, parx, rmsxy, kzef);

%   Display results
fprintf('The forecast values and their standard errors\n\n');
fprintf('   i     fva       fsd\n\n');
fprintf('%4d%10.4f%10.4f\n',[[1:numel(fva)]' fva fsd]');
fprintf('\n    z(t)      n(t)\n');
disp(xxyn);
```

9.2 Program Results

```
     g13bh example results

The forecast values and their standard errors

   i     fva       fsd

   1   88.2723    4.7881
   2   99.9425    6.4690
   3  100.6499    7.3175
   4   95.0958    7.5534

    z(t)      n(t)
  164.4620  -76.1897
  170.3924  -70.4499
  174.5193  -73.8694
  175.2747  -80.1789
```