NAG Toolbox

nag_tsa_multi_inputmod_forecast (g13bj)

1 Purpose

nag_tsa_multi_inputmod_forecast (g13bj) produces forecasts of a time series (the output series) which depends on one or more other (input) series via a previously estimated multi-input model for which the state set information is not available. The future values of the input series must be supplied. In contrast with nag_tsa_multi_inputmod_forecast_state (g13bh) the original past values of the input and output series are required. 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

```
[para, xxy, rmsxy, mrx, fva, fsd, sttf, nsttf, ifail] =
nag_tsa_multi_inputmod_forecast(mr, mt, para, kfc, nev, nfv, xxy, kzef, rmsxy,
mrx, parx, isttf, 'nser', nser, 'npara', npara)
```

```
[para, xxy, rmsxy, mrx, fva, fsd, sttf, nsttf, ifail] = g13bj(mr, mt, para, kfc,
nev, nfv, xxy, kzef, rmsxy, mrx, parx, isttf, 'nser', nser, 'npara', npara)
```

3 Description

nag_tsa_multi_inputmod_forecast (g13bj) has two stages. The first stage is essentially the same as a call to the model estimation function nag_tsa_multi_inputmod_estim (g13be), with zero iterations. In particular, all the parameters remain unchanged in the supplied input series transfer function models and output noise series ARIMA model. The internal nuisance parameters associated with the pre-observation period effects of the input series are estimated where requested, and so are any backforecasts of the output noise series. The output components z_t and n_t, and residuals a_t are calculated exactly as in Section 3 in nag_tsa_multi_inputmod_estim (g13be), and the state set for forecasting is constituted.

The second stage is essentially the same as a call to the forecasting function nag_tsa_multi_inputmod_forecast_state (g13bh). The same information is required, and the same information is returned.

Use of nag_tsa_multi_inputmod_forecast (g13bj) should be confined to situations in which the state set for forecasting is unknown. Forecasting from the original data is relatively expensive because it requires recalculation of the state set. nag_tsa_multi_inputmod_forecast (g13bj) returns the state set for use in producing further forecasts using nag_tsa_multi_inputmod_forecast_state (g13bh), or for updating the state set using nag_tsa_multi_inputmod_update (g13bg).

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: **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 refer respectively to 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 $s > 1$, $P + D + Q > 0$;
$d + s \times (P + D) \leq n$;
$p + d - q + s \times (P + D - Q) \leq n$.

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

The transfer function model orders b, p and q of each of the input series. The data for input series i is 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, and $r_i = 2$ or 3 for a transfer function input.

The choice $r_i = 3$ leads to estimation of the pre-period input effects as nuisance parameters, and $r_i = 2$ suppresses this estimation. This choice may affect the returned forecasts and the state set.

When $r_i = 1$, any nonzero contents of rows 1, 2 and 3 of column i are ignored.

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

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

Estimates of the multi-input model parameters. These are in order, firstly the ARIMA model parameters: p values of ϕ parameters, q values of θ parameters, P values of Φ parameters, Q values of Θ parameters.

These are followed by the transfer function model parameter values $\omega_0, \omega_1, \ldots, \omega_{q_1}$, $\delta_1, \ldots, \delta_{p_1}$ for the first of any input series and similarly for each subsequent input series. The final component of **para** is the value of the constant c.

4: **kfc** – INTEGER

Must be set to 1 if the constant was estimated when the model was fitted, and 0 if it was held at a fixed value. This only affects the degrees of freedom used in calculating the estimated residual variance.

Constraint: $\textbf{kfc} = 0$ or 1.

5: **nev** – INTEGER

The number of original (undifferenced) values in each of the input and output time series.

6: **nfv** – INTEGER

The number of forecast values of the output series required.

Constraint: $\textbf{nfv} > 0$.

7: **xxy**($ldxxy$, **nser**) – REAL (KIND=nag_wp) array

$ldxxy$, the first dimension of the array, must satisfy the constraint $ldxxy \geq (\textbf{nev} + \textbf{nfv})$.

The columns of **xxy** must contain in the first **nev** places, the past values of each of the input and output series, in that order. In the next **nfv** places, the columns relating to the input series (i.e., columns 1 to **nser** $-$ 1) contain the future values of the input series which are necessary for construction of the forecasts of the output series y.

8: **kzef** – INTEGER

Must be set to 0 if the relevant **nfv** values of the forecasts (**fva**) are to be held in the output series column (**nser**) of **xxy** (which is otherwise unchanged) on exit, and 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 **xxy** on exit.

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

The first (**nser** $-$ 1) elements of **rmsxy** must contain the estimated residual variance of the input series ARIMA models. In the case of those inputs for which no ARIMA model is available or its effects are to be excluded in the calculation of forecast standard errors, the corresponding entry of **rmsxy** should be set to 0.

10: **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.

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

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

Values of the parameters (ϕ, θ, Φ, and Θ) for each of the input series ARIMA models.

Thus column i contains **mrx**$(1, i)$ values of ϕ, **mrx**$(3, i)$ values of θ, **mrx**$(4, i)$ values of Φ and **mrx**$(6, i)$ values of Θ, in that order.

Values in the columns relating to those input series for which no ARIMA model is available are ignored.

12: **isttf** – INTEGER

The dimension of the array **sttf**.

Constraint: **isttf** $\geq (P \times s) + d + (D \times s) + q + \max(p, Q \times s) + ncf$, where $ncf = \sum(b_i + q_i + p_i)$ and the summation is over all input series for which $r_i > 1$.

5.2 Optional Input Parameters

1: **nser** – INTEGER

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

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

2: **npara** – INTEGER

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

The exact number of ϕ, θ, Φ, Θ, ω, δ, c parameters, so that **npara** $= p + q + P + Q +$ **nser** $+ \sum(p + q)$, the summation being over all the input series. (c must be included whether its value was previously estimated or was set fixed.)

5.3 Output Parameters

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

The parameter values may be updated using an additional iteration in the estimation process.

2: **xxy**($ldxxy$, **nser**) – REAL (KIND=nag_wp) array

If **kzef** $= 0$ then **xxy** is unchanged except that the relevant **nfv** values in the column relating to the output series (column **nser**) contain the forecast values (**fva**), but if **kzef** $\neq 0$ then the columns of **xxy** contain the corresponding values of the input component series z_t and the values of the output noise component n_t, in that order.

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

rmsxy(**nser**) contains the estimated residual variance of the output noise ARIMA model which is calculated from the supplied series. Otherwise **rmsxy** is unchanged.

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

Unchanged, except for column **nser** which is used as workspace.

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

The required forecast values for the output series. (Note that these are also output in column **nser** of **xxy** if **kzef** $= 0$.)

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

The standard errors for each of the forecast values.

7: **sttf**(**isttf**) – REAL (KIND=nag_wp) array

The **nsttf** values of the state set based on the first **nev** sets of (past) values of the input and output series.

8: **nsttf** – INTEGER

The number of values in the state set array **sttf**.

9: **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, **kfc** < 0,
or **kfc** > 1,
or $ldxxy < (\mathbf{nev} + \mathbf{nfv})$,
or **nfv** ≤ 0.

ifail $= 2$

On entry, $ldparx$ is too small or **npara** is inconsistent with the orders specified in arrays **mr** and **mt**; or one of the r_i, stored in **mt**$(4, i)$, does not equal 1, 2 or 3.

ifail $= 3$

On entry or during execution, one or more sets of δ parameters do not satisfy the stationarity or invertibility test conditions.

ifail $= 4$

On entry, iwa is too small for the final forecasting calculations. This is a highly unlikely error, and would probably indicate that **nfv** was abnormally large.

ifail $= 5$

On entry, iwa is too small by a very considerable margin. No information is supplied about the requisite minimum size.

ifail $= 6$

On entry, iwa is too small, but the requisite minimum size is returned in $mwa(1)$.

ifail $= 7$

On entry, $imwa$ is too small, but the requisite minimum size is returned in $mwa(1)$.

ifail $= 8$

This indicates a failure in nag_linsys_real_posdef_solve_1rhs (f04as) which is used to solve the equations giving the latest estimates of the parameters.

ifail $= 9$

This indicates a failure in the inversion of the second derivative matrix associated with parameter estimation.

ifail $= 10$

On entry or during execution, one or more sets of the ARIMA (ϕ, θ, Φ or Θ) parameters do not satisfy the stationarity or invertibility test conditions.

ifail $= 11$

On entry, **isttf** is too small.

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 (g13bj) is approximately proportional to the product of the length of each series and the square of the number of parameters in the multi-input model.

9 Example

The data in this example relates to 40 observations of an output time series and 5 input time series. The output series has one autoregressive (ϕ) parameter and one seasonal moving average (Θ) parameter, with initial values $\phi = 0.495$, $\Theta = 0.238$ and $c = -82.858$. The seasonal period is 4. This example differs from the example in nag_tsa_multi_inputmod_estim (g13be) in that four of the input series are simple series and the fifth is defined by a transfer function with orders $b_5 = 1$, $q_5 = 0$, $p_5 = 1$, $r_5 = 3$, which allows for pre-observation period effects. The initial values for the transfer model are:

$$\omega_1 = -0.367, \quad \omega_2 = -3.876, \quad \omega_3 = 4.516, \quad \omega_4 = 2.474 \quad \omega_5 = 8.629, \quad \delta_1 = 0.688.$$

A further 8 values of the input series are supplied, and it is assumed that the values for the fifth series have themselves been forecast from an ARIMA model with orders $2 \quad 0 \quad 2 \quad 0 \quad 1 \quad 1 \quad 4$, in which $\phi_1 = 1.6743$, $\phi_2 = -0.9505$, $\theta_1 = 1.4605$, $\theta_2 = -0.4862$ and $\Theta_1 = 0.8993$, and for which the residual mean square is 0.1720.

The following are computed and printed out: the state set after initial processing of the original 40 sets of values, the estimated residual variance for the output noise series, the 8 forecast values and their standard errors, and the values of the components z_t and the output noise component n_t.

9.1 Program Text

```
     function g13bj_example

fprintf('g13bj example results\n\n');

% orders and transfer
mr = [nag_int(1); 0; 0; 0; 0; 1; 4];
mt = [nag_int(0), ...
         0, 0, 0, 1, 0;
      0, 0, 0, 0, 0, 0;
      0, 0, 0, 0, 1, 0;
      1, 1, 1, 1, 3, 0];
% model parameters
para = [0.495; 0.238; -0.367; -3.876; 4.516;
        2.474; 8.629; 0.688; -82.858];
% Observed values
xxy = [ 1     1     0     0     8.075 105;
        1     0     1     0     7.819 119;
        1     0     0     1     7.366 119;
        1    -1    -1    -1     8.113 109;
        2     1     0     0     7.380 117;
        2     0     1     0     7.134 135;
        2     0     0     1     7.222 126;
        2    -1    -1    -1     7.768 112;
        3     1     0     0     7.386 116;
        3     0     1     0     6.965 122;
        3     0     0     1     6.478 115;
        3    -1    -1    -1     8.105 115;
        4     1     0     0     8.060 122;
        4     0     1     0     7.684 138;
        4     0     0     1     7.580 135;
        4    -1    -1    -1     7.093 125;
        5     1     0     0     6.129 115;
        5     0     1     0     6.026 108;
        5     0     0     1     6.679 100;
        5    -1    -1    -1     7.414  96;
        6     1     0     0     7.112 107;
        6     0     1     0     7.762 115;
        6     0     0     1     7.645 123;
        6    -1    -1    -1     8.639 122;
        7     1     0     0     7.667 128;
        7     0     1     0     8.080 136;
        7     0     0     1     6.678 140;
        7    -1    -1    -1     6.739 122;
        8     1     0     0     5.569 102;
        8     0     1     0     5.049 103;
        8     0     0     1     5.642  89;
```

```
        8    -1    -1    -1     6.808  77;
        9     1     0     0     6.636  89;
        9     0     1     0     8.241  94;
        9     0     0     1     7.968 104;
        9    -1    -1    -1     8.044 108;
       10     1     0     0     7.791 119;
       10     0     1     0     7.024 126;
       10     0     0     1     6.102 119;
       10    -1    -1    -1     6.053 103];
[nev, nis] = size(xxy);
% Add future values
xyf = [11     1     0     0     5.941   0;
       11     0     1     0     5.386   0;
       11     0     0     1     5.811   0;
       11    -1    -1    -1     6.716   0;
       12     1     0     0     6.923   0;
       12     0     1     0     6.939   0;
       12     0     0     1     6.705   0;
       12    -1    -1    -1     6.914   0];
nfv = size(xyf,1);
xxy = [xxy; xyf];

% Residual variance of input series
rmsxy = [0;   0;    0;    0;    0.172;  0];
% orders for ARIMA (only 5th residua variance is non-zero)
mrx = zeros(7, nis, nag_int_name);
mrx(:,5) = [2;   0;    2;    0;    1;    1;   4];
% Parameters for input series ARIMA
parx = zeros(5,nis);
parx(:,5) = [ 1.6743; -0.9505;  1.4605;  -0.4862;  0.8993];

% Sizes
nev   = nag_int(40);
nfv   = nag_int(8);
isttf = nag_int(20);

% Update forecasts
kfc  = nag_int(1);
kzef = nag_int(1);
[para, xxy, rmsxy, mrx, fva, fsd, sttf, nsttf, ifail] = ...
    g13bj( ...
           mr, mt, para, kfc, nev, nfv, xxy, kzef, rmsxy, mrx, parx, isttf);

%   Display results
fprintf('After processing %4d sets of observations\n', nev);
fprintf('%4d values of the state set are derived\n\n', nsttf);
disp(sttf(1:nsttf)');
fprintf('\nThe residual mean square for the output\n');
fprintf('series is also derived and its value is %10.4f\n\n', rmsxy(nis));
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');
[ifail] = x04ca( ...
                 'General', ' ', xxy, 'The values of z(t) and n(t)');
fprintf('The first %2d columns hold z(t) and the last column holds n(t)\n', ...
       nis-1);
```

9.2 Program Results

```
     g13bj example results

After processing   40 sets of observations
   6 values of the state set are derived

    6.0530  193.8741    2.0790   -2.8580   -3.5906   -2.5203

The residual mean square for the output
series is also derived and its value is    20.7599
```

```
The forecast values and their standard errors

   i     fva       fsd

   1   93.3977    4.5563
   2   96.9577    6.2172
   3   86.0463    7.0933
   4   77.5887    7.3489
   5   82.1393    7.3941
   6   96.2755    7.5823
   7   98.3451    8.1445
   8   93.5774    8.8536

 The values of z(t) and n(t)
             1          2          3          4          5          6
  1     -0.3391    -3.8886     0.0000     0.0000   188.6028   -79.3751
  2     -0.3391    -0.0000     4.5139     0.0000   199.4379   -84.6127
  3     -0.3391    -0.0000     0.0000     2.4789   204.6834   -87.8232
  4     -0.3391     3.8886    -4.5139    -2.4789   204.3834   -91.9402
  5     -0.6782    -3.8886     0.0000     0.0000   210.6229   -89.0560
  6     -0.6782    -0.0000     4.5139     0.0000   208.5905   -77.4262
  7     -0.6782    -0.0000     0.0000     2.4789   205.0696   -80.8703
  8     -0.6782     3.8886    -4.5139    -2.4789   203.4065   -87.6242
  9     -1.0173    -3.8886     0.0000     0.0000   206.9738   -86.0678
 10     -1.0173    -0.0000     4.5139     0.0000   206.1317   -87.6283
 11     -1.0173    -0.0000     0.0000     2.4789   201.9196   -88.3812
 12     -1.0173     3.8886    -4.5139    -2.4789   194.8194   -75.6979
 13     -1.3564    -3.8886     0.0000     0.0000   203.9738   -76.7287
 14     -1.3564    -0.0000     4.5139     0.0000   209.8837   -75.0412
 15     -1.3564    -0.0000     0.0000     2.4789   210.7052   -76.8277
 16     -1.3564     3.8886    -4.5139    -2.4789   210.3730   -80.9125
 17     -1.6955    -3.8886     0.0000     0.0000   205.9421   -85.3580
 18     -1.6955    -0.0000     4.5139     0.0000   194.5753   -89.3937
 19     -1.6955    -0.0000     0.0000     2.4789   185.8662   -86.6496
 20     -1.6955     3.8886    -4.5139    -2.4789   185.5090   -84.7094
 21     -2.0346    -3.8886     0.0000     0.0000   191.6056   -78.6824
 22     -2.0346    -0.0000     4.5139     0.0000   193.1941   -80.6734
 23     -2.0346    -0.0000     0.0000     2.4789   199.8958   -77.3402
 24     -2.0346     3.8886    -4.5139    -2.4789   203.4970   -76.3583
 25     -2.3737    -3.8886     0.0000     0.0000   214.5519   -80.2896
 26     -2.3737    -0.0000     4.5139     0.0000   213.7702   -79.9104
 27     -2.3737    -0.0000     0.0000     2.4789   216.7963   -76.9015
 28     -2.3737     3.8886    -4.5139    -2.4789   206.7803   -79.3024
 29     -2.7128    -3.8886     0.0000     0.0000   200.4157   -91.8142
 30     -2.7128    -0.0000     4.5139     0.0000   185.9409   -84.7420
 31     -2.7128    -0.0000     0.0000     2.4789   171.4951   -82.2613
 32     -2.7128     3.8886    -4.5139    -2.4789   166.6735   -83.8565
 33     -3.0519    -3.8886     0.0000     0.0000   173.4176   -77.4771
 34     -3.0519    -0.0000     4.5139     0.0000   176.5733   -84.0353
 35     -3.0519    -0.0000     0.0000     2.4789   192.5940   -88.0211
 36     -3.0519     3.8886    -4.5139    -2.4789   201.2606   -87.1045
 37     -3.3910    -3.8886     0.0000     0.0000   207.8790   -81.5993
 38     -3.3910    -0.0000     4.5139     0.0000   210.2493   -85.3721
 39     -3.3910    -0.0000     0.0000     2.4789   205.2616   -85.3495
 40     -3.3910     3.8886    -4.5139    -2.4789   193.8741   -84.3790
 41     -3.7301    -3.8886     0.0000     0.0000   185.6167   -84.6003
 42     -3.7301     0.0000     4.5139     0.0000   178.9692   -82.7953
 43     -3.7301     0.0000     0.0000     2.4789   169.6066   -82.3091
 44     -3.7301     3.8886    -4.5139    -2.4789   166.8325   -82.4095
 45     -4.0692    -3.8886     0.0000     0.0000   172.7331   -82.6360
 46     -4.0692     0.0000     4.5139     0.0000   178.5789   -82.7481
 47     -4.0692     0.0000     0.0000     2.4789   182.7389   -82.8036
 48     -4.0692     3.8886    -4.5139    -2.4789   183.5818   -82.8311
The first  5 columns hold z(t) and the last column holds n(t)
```