NAG Toolbox for MATLAB

g13ad

1 Purpose

g13ad calculates preliminary estimates of the parameters of an autoregressive integrated moving average (ARIMA) model from the autocorrelation function of the appropriately differenced times series.

2 Syntax

```
[par, rv, isf, ifail] = g13ad(mr, r, xv, npar, 'nk', nk)
```

3 Description

Preliminary estimates of the p non-seasonal autoregressive parameters $\phi_1, \phi_2, \ldots, \phi_p$ and the q non-seasonal moving average parameters $\theta_1, \theta_2, \ldots, \theta_q$ may be obtained from the sample autocorrelations relating to lags 1 to $p+q$, i.e., $r_1, \ldots, r_{p+q}$, of the differenced $\nabla^d \nabla_s^D x_t$, where x_t is assumed to follow a (possibly) seasonal ARIMA model (see Section 3 of the document for g13ae for the specification of an ARIMA model).

Taking $r_0 = 1$ and $r_{-k} = r_k$, the ϕ_i, for $i = 1, 2, \ldots, p$ are the solutions to the equations

$$r_{q+i-1}\phi_1 + r_{q+i+2}\phi_2 + \cdots + r_{q+i-p}\phi_p = r_{q+i}, \qquad i = 1, 2, \ldots, p.$$

The θ_j, for $j = 1, 2, \ldots, q$, are obtained from the solutions to the equations

$$c_j = \tau_0\tau_j + \tau_1\tau_{j+1} + \cdots + \tau_{q+j}\tau_q, \qquad j = 0, 1, \ldots, q$$

(Cramer Wold-factorization), by setting

$$\theta_j = -\frac{\tau_j}{\tau_0},$$

where c_j are the ‘covariances’ modified in a two stage process by the autoregressive parameters.

Stage 1:

$$\begin{array}{ll} d_j = r_j - \phi_1 r_{j-1} - \cdots - \phi_p r_{j-p}, & j = 0, 1, \ldots, q; \\ d_j = 0, & j = q+1, q+2, \ldots, p+q. \end{array}$$

Stage 2:

$$c_j = d_j - \phi_1 d_{j+1} - \phi_2 d_{j+2} - \cdots - \phi_p d_{j+p}, \qquad j = 0, 1, \ldots, q.$$

The P seasonal autoregressive parameters $\Phi_1, \Phi_2, \ldots, \Phi_P$ and the Q seasonal moving average parameters $\Theta_1, \Theta_2, \ldots, \Theta_Q$ are estimated in the same way as the non-seasonal parameters, but each r_j is replaced in the calculation by $r_{s\times j}$, where s is the seasonal period.

An estimate of the residual variance is obtained by successively reducing the sample variance, first for non-seasonal, and then for seasonal, parameter estimates. If moving average parameters are estimated, the variance is reduced by a multiplying factor of τ_0^2, but otherwise by c_0.

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) – **int32 array**

The orders vector (p, d, q, P, D, Q, s) of the ARIMA model whose parameters are to be estimated. 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 $s > 1$, $P + D + Q > 0$.

2: **r**(**nk**) – **double array**

The autocorrelations (starting at lag 1), which must have been calculated after the time series has been appropriately differenced.

Constraint: $-1.0 \leq \mathbf{r}(i) \leq 1.0$, for $i = 1, 2, \ldots, \mathbf{nk}$.

3: **xv** – **double scalar**

The series sample variance, calculated after appropriate differencing has been applied to the series.

Constraint: $\mathbf{xv} > 0.0$.

4: **npar** – **int32 scalar**

The exact number of parameters specified in the model by array **mr**.

Constraint: $\mathbf{npar} = p + q + P + Q$.

5.2 Optional Input Parameters

1: **nk** – **int32 scalar**

Default: The dimension of the array **r**.

the maximum lag of the autocorrelations in array **r**.

Constraint: $\mathbf{nk} \geq \max(p + q, s \times (P + Q))$.

5.3 Input Parameters Omitted from the MATLAB Interface

wa, nwa

5.4 Output Parameters

1: **par**(**npar**) – **double array**

The first **npar** elements of **par** contain the preliminary estimates of the ARIMA model parameters, in standard order.

2: **rv** – **double scalar**

An estimate of the residual variance of the preliminarily estimated model.

3: **isf(4) – int32 array**

Contains success/failure indicators, one for each of the four types of parameter (autoregressive, moving average, seasonal autoregressive, seasonal moving average).

The indicator has the interpretation:

0 No parameter of this type is in the model.

1 Parameters of this type appear in the model and satisfactory preliminary estimates of this type were obtained.

-1 Parameters of this type appear in the model but satisfactory preliminary estimates of this type were not obtainable. The estimates of this type of parameter were set to 0.0 in array **par**.

4: **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, the orders vector **mr** is invalid. One of the constraints in Section 5 has been violated.

ifail $= 2$

On entry, **nk** $< \max(p + q, s \times (P + Q))$. There are not enough autocorrelations to enable the required model to be estimated.

ifail $= 3$

On entry, at least one element of **r** lies outside the range $[-1.0, 1.0]$.

ifail $= 4$

On entry, **xv** ≤ 0.0.

ifail $= 5$

On entry, **npar** $\neq p + q + P + Q$.

ifail $= 6$

On entry, the workspace array **wa** is too small. See Section 5 for the minimum size formula.

ifail $= 7$

Satisfactory parameter estimates could not be obtained for all parameter types in the model. Inspect array **isf** for indicators of the parameter type(s) which could not be estimated.

7 Accuracy

The performance of the algorithm is conditioned by the roots of the autoregressive and moving average operators. If these are not close to unity in modulus, the errors, e, should satisfy $e < 100\epsilon$ where ϵ is ***machine precision***.

8 Further Comments

The time taken by g13ad is approximately proportional to $\left(p^3 + q^2 + P^3 + Q^2\right)$.

9 Example

```
mr = [int32(0);
      int32(1);
      int32(1);
      int32(0);
      int32(1);
      int32(1);
      int32(12)];
r = [-0.32804;
     0.0985;
     -0.21854;
     0.05585;
     0.04679;
     0.04135;
     -0.07989;
     0.00335;
     0.13973;
     -0.04022;
     0.07618;
     -0.40583;
     0.18239;
     -0.05057;
     0.16094;
     -0.159;
     0.09152;
     -0.03474;
     0.05195;
     -0.14417;
     0.04264;
     -0.08169999999999999;
     0.23389;
     -0.02828;
     -0.09001000000000001;
     0.0305;
     -0.02046;
     0.05522;
     -0.02048;
     -0.06651;
     -0.0294;
     0.20204;
     -0.13953;
     0.10098;
     -0.20849;
     0.03338;
     0.00829;
     0.07081999999999999;
     -0.04457;
     -0.01216];
xv = 0.00213;
npar = int32(2);
[par, rv, isf, ifail] = g13ad(mr, r, xv, npar)
```

```
par =
    0.3739
    0.5124
rv =
    0.0015
isf =
           0
           1
           0
           1
ifail =
           0
```