

nag_tsa_auto_corr_part (g13acc)

1. Purpose

nag_tsa_auto_corr_part (g13acc) calculates partial autocorrelation coefficients given a set of autocorrelation coefficients. It also calculates the predictor error variance ratios for increasing order of finite lag autoregressive predictor, and the autoregressive parameters associated with the predictor of maximum order.

2. Specification

```
#include <nag.h>
#include <nagg13.h>

void nag_tsa_auto_corr_part(double r[], Integer nk, Integer nl, double p[],
                           double v[], double ar[], Integer *nvl, NagError *fail)
```

3. Description

The data consist of values of autocorrelation coefficients $r_1, r_2, \dots, r_K$, relating to lags $1, 2, \dots, K$. These will generally (but not necessarily) be sample values such as may be obtained from a time series x_t using **nag_tsa_auto_corr (g13abc)**.

The partial autocorrelation coefficient at lag l may be identified with the parameter $p_{l,l}$ in the autoregression

$$x_t = c_l + p_{l,1}x_{t-1} + p_{l,2}x_{t-2} + \dots + p_{l,l}x_{t-l} + e_{l,t}$$

where $e_{l,t}$ is the predictor error.

The first subscript l of $p_{l,l}$ and $e_{l,t}$ emphasises the fact that the parameters will in general alter as further terms are introduced into the equation (i.e., as l is increased).

The parameters are determined from the autocorrelation coefficients by the Yule-Walker equations

$$r_i = p_{i,1}r_{i-1} + p_{i,2}r_{i-2} + \dots + p_{i,i}r_{i-i}, \quad i = 1, 2, \dots, l$$

taking $r_j = r_{|j|}$ when $j < 0$, and $r_0 = 1$.

The predictor error variance ratio $v_l = \text{var}(e_{l,t})/\text{var}(x_t)$ is defined by

$$v_l = 1 - p_{l,1}r_1 - p_{l,2}r_2 - \dots - p_{l,l}r_l.$$

The above sets of equations are solved by a recursive method (the Durbin-Levinson algorithm). The recursive cycle applied for $l = 1, 2, \dots, (L-1)$, where L is the number of partial autocorrelation coefficients required, is initialised by setting $p_{1,1} = r_1$ and $v_1 = 1 - r_1^2$.

Then

$$\begin{aligned} p_{l+1,l+1} &= (r_{l+1} - p_{l,1}r_l - p_{l,2}r_{l-1} - \dots - p_{l,l}r_1)/v_l \\ p_{l+1,j} &= p_{l,j} - p_{l+1,l+1}p_{l,l+1-j}, \quad j = 1, 2, \dots, l \\ v_{l+1} &= v_l \times (1 - p_{l+1,l+1})(1 + p_{l+1,l+1}). \end{aligned}$$

If the condition $|p_{l,l}| \geq 1$ occurs, say when $l = l_0$, it indicates that the supplied autocorrelation coefficients do not form a positive-definite sequence (see Hannan (1960)), and the recursion is not continued. The autoregressive parameters are overwritten at each recursive step, so that upon completion the only available values are $p_{L,j}$, for $j = 1, 2, \dots, L$ or $p_{l_0-1,j}$ if the recursion has been prematurely halted.

4. Parameters

r[nk]

Input: **r**[$k-1$] contains the autocorrelation coefficient relating to lag k , for $k = 1, 2, \dots, K$.

nk

Input: the number of lags, K . The lags range from 1 to K and do not include zero.
Constraint: **nk** > 0.

nl

Input: the number of partial autocorrelation coefficients required, L .
Constraint: $0 < \mathbf{nl} \leq \mathbf{nk}$.

p[nl]

Output: **p**[$l-1$] contains the partial autocorrelation coefficient at lag l , $p_{l,l}$, for $l = 1, 2, \dots, \mathbf{nl}$.

v[nl]

Output: **v**[$l-1$] contains the predictor error variance ratio v_l , for $l = 1, 2, \dots, \mathbf{nl}$.

ar[nl]

Output: the autoregressive parameters of maximum order, i.e., $p_{L,j}$ if **fail.code** = **NE_NOERROR**, or $p_{l_0-1,j}$ if **fail.code** = **NE_CORR_NOT_POS_DEF**, for $j = 1, 2, \dots, \mathbf{nl}$.

nlv

Output: the number of valid values in each of **p**, **v** and **ar**. Thus in the case of premature termination at iteration l_0 (see Section 3), **nlv** is returned as $l_0 - 1$.

fail

The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_INT_ARG_LE

On entry, **nk** must not be less than or equal to 0: **nk** = $\langle \text{value} \rangle$.

On entry, **nl** must not be less than or equal to 0: **nl** = $\langle \text{value} \rangle$.

NE_2_INT_ARG_LT

On entry, **nk** = $\langle \text{value} \rangle$ while **nl** = $\langle \text{value} \rangle$. These parameters must satisfy **nk** $\geq$ **nl**.

NE_INVALID_AUTOCO_COEF

On entry, the autocorrelation coefficient of lag 1 has an absolute value greater than or equal to 1.0; no recursions could be performed.

NE_CORR_NOT_POS_DEF

Recursion has been prematurely terminated; the supplied autocorrelation coefficients do not form a positive-definite sequence.

Parameter **nlv** returns the number of valid values computed.

6. Further Comments

The time taken by the function is proportional to **nlv**².

6.1. Accuracy

The computations are believed to be stable.

6.2. References

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

Durbin J (1960) The Fitting of Time Series Models *Rev. Inst. Internat. Stat.* **28** 233.

Hannan E J (1960) *Time Series Analysis* Methuen, p 11.

7. See Also

nag_tsa_auto_corr (g13abc)

8. Example

In the example below the input series is the set of 10 sample autocorrelation coefficients derived from the original series of sunspot numbers by nag_tsa_auto_corr (g13abc) example program. The results show 5 values of each of the three output arrays – partial autocorrelation coefficients, predictor error variance ratios and autoregressive parameters. All of these were valid.

8.1. Program Text

```
/* nag_tsa_auto_corr_part(g13acc) Example Program
 *
 * Copyright 1991 Numerical Algorithms Group.
 *
 * Mark 2, 1991.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg13.h>

#define NLMAX 5
#define NKMAX 10

main()
{
    Integer i, nk, nl, nvl;
    double ar[NLMAX], p[NLMAX], r[NKMAX], v[NLMAX];
    static NagError fail;

    Vprintf("g13acc Example Program Results\n");
    /* Skip heading in data file */
    Vscanf("%*[^\\n]");
    Vscanf("%ld %ld", &nk, &nl);

    if (nl>0 && nl<=NLMAX && nk>0 && nk<=NKMAX)
    {
        for (i=0; i<nk; ++i)
            Vscanf("%lf", &r[i]);

        g13acc(r, nk, nl, p, v, ar, &nvl, &fail);

        if (fail.code != NE_NOERROR)
        {
            Vprintf(" ***** ERROR *****  \n");
            Vprintf("g13acc fails with failure message %s\n", fail.message);
        }
        if (fail.code == NE_CORR_NOT_POS_DEF)
            Vprintf(" Only %2ld valid sets were generated\n\n", nvl);
        if (fail.code == NE_NOERROR || fail.code == NE_CORR_NOT_POS_DEF)
        {
            Vprintf("\n Lag   Partial   Predictor error  Autoregressive\n");
            Vprintf("      autocorrn  variance ratio   parameter\n\n");
            for (i=0; i<nvl; ++i)
                Vprintf(" %2ld%9.3f%16.3f%14.3f\n", i+1, p[i], v[i], ar[i]);
        }
        if (fail.code != NE_NOERROR)
            exit(EXIT_FAILURE);
    }
    else
    {
        Vfprintf(stderr, "One or both of nl and nk are out of range:\n
nl = %-3ld while nk = %-3ld\n", nl, nk);
        exit(EXIT_FAILURE);
    }
    exit(EXIT_SUCCESS);
}
```

8.2. Program Data

```
g13acc Example Program Data
10 5
    0.8004    0.4355    0.0328   -0.2835   -0.4505
   -0.4242   -0.2419   -0.0550    0.3783    0.5857
```

8.3. Program Results

```
g13acc Example Program Results

Lag  Partial    Predictor error  Autoregressive
     autocorrn  variance ratio  parameter

 1    0.800          0.359          1.108
 2   -0.571          0.242         -0.290
 3   -0.239          0.228         -0.193
 4   -0.049          0.228         -0.014
 5   -0.032          0.228         -0.032
```
