NAG Cライブラリを用いた

季節ARIMAモデルのフィッティング

2009年5月11日

The Numerical Algorithms Group Ltd

Wilkinson House

Jordan Hill

Oxford

OX2 8DR

UK

Web サイト: www.nag-j.co.jp

問い合わせ先: sales@nag-j.co.jp

テクニカルサポート: naghelp@nag-j.co.jp

時系列モデルでよく使用されるモデルの一つに季節自己回帰和分移動平均(ARIMA) モデルがあります。ARIMA モデルでは、時系列 $y_1, y_2,\ldots, y_n$（$t = 1, 2,\ldots, n$ ）は以下の式に従うと考えられています：

$$\Delta^d \Delta_s^D y_t - c = w_t$$

ここで $\Delta^d \Delta_s^D y_t$ は次数 d の非季節階差と季節性s で次数 D の季節階差を時系列y_t へ適用した結果とします。 階差系列の長さは$N = n - d'$ になります。ここでd' は一般化次数を示し $d' = d + (D \times s)$ で表されます。
スカラー c は階差系列の期待値であり、系列 $w_1,w_2,\ldots,w_N$ は回帰方程式のペアで定義されるゼロ平均の定常自己回帰移動平均（ARMA）モデルに従います。これらは中間系列 e_t を通じて無相関系列a_t によりw_t を表しています。最初の方程式は季節性構造を表しています：

$$w_t = \Psi_1 w_{t-s} + \Psi_2 w_{t-2s} + \ldots + \Psi_P w_{t-Ps} + e_t - \Theta_1 e_{t-s} - \Theta_2 e_{t-2s} - \ldots - \Theta_Q e_{t-Qs}$$

二番目の方程式は非季節性構造を表しています。もしモデルが純粋に非季節性である場合、最初の方程式は重複しており、上記のe_t は w_t と等しくなります：

$$e_t = \psi_1 e_{t-1} + \psi_2 e_{t-2} + \ldots + \psi_p e_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \ldots - \theta_q a_{t-q}$$

NAG Fortran ライブラリには季節ARIMA モデルからのフィッティング（G13AEF)と予測(G13AJF)専用のルーチンがありますが、 C ライブラリにはありません。

Fortranライブラリ と C ライブラリには共に一つの出力系列を一つ以上の入力系列に関連づける、多入力モデルをフィッティングするルーチンがあります（それぞれG13BEF と g13bec ）。多入力系列では、出力系列y_t（$t = 1, 2,\ldots,n$）は、（未観測）成分 $z_{i;t}$ の合計であると考えられており、それはm個の入力系列 $x_{i,t}(i = 1, 2,\ldots,m)$ になるはずです。典型的な成分 z_t は以下のいずれかになります。

- 単回帰成分 $z_t = wx_t$ もしくは
- 以下で表される x_t に関連した、変数のラグによる影響を考慮した伝達関数モデル成分

$$z_t = \delta_1 z_{t-1} + \delta_2 z_{t-2} + \ldots + \delta_p z_{t-p} + w_0 x_{t-b} - w_1 x_{t-b-1} - \ldots - w_q x_{t-b-q}$$

出力系列 y_t は以下で定義されます。

$$y_t = z_{1,t} + \ldots + z_{m,t} + n_t$$

ここではn_t は誤差、もしくは出力ノイズ成分であり、（おそらく季節）ARIMA model に従うと考えられます。そのため、もし入力系列がなければ、それは $m = 0$ となり、多入力モデルは季節ARIMA モデルと等しくなります。

この文章はC ライブラリルーチン g13bec (nag_tsa_multi_inp_model_estim)を用いてどのように季節ARIMAモデルをフィッティングするか、そしてC ライブラリルーチン g13bjc (nag_tsa_multi_inp_model_forecast)を用いてどのようにこのようなモデルから予測をするかを簡単に説明しています。それはこれら二つのルーチンのドキュメントと併せて理解することが必要です。完全なサンプルソースコード、データ、期待値はNAG ウェブサイトでご覧いただけます。

私たちはARIMA モデルをフィッティングしていますので、入力系列はありません。 これは単一系列(y, ルーチンドキュメントでは出力系列と呼ばれています)しかないことを意味しており、以下で表されます。

```
nseries = 1;
```

私たちはまた動的にメモリを割り当てようとしています。そのため、ストライドパラメータ tdxxy は系列の数、この場合は 1 に設定することができます。

```
tdxxy = 1;
```

入力系列がない場合でも、メモリを以下の伝達関数の構造体に割り当てる必要があります。

```
nag_tsa_transf_orders(nseries, &transfv, NAGERR_DEFAULT);
```

しかし初期化する必要はありません。フィッティングされた ARIMA モデルを表しているパラメータをもつ arimav 構造体 により、残りのパラメータはご覧いただけばすぐおわかりいただけます。そしてパラメータの数、npara は以下のように設定されます。

```
npara = arimav.p + arimav.q + arimav.bigp + arimav.bigq + nseries;
```

そしてメモリを以下のように割り当てることができます。

```
para = NAG_ALLOC(npara, double);
sd = NAG_ALLOC(npara, double);
xxy = NAG_ALLOC((nxxy+nfv)*tdxxy, double);
```

上記のコードの抜粋では、nfv はg13bjcを用いて予測したい値の数を表しています。そして para、sd、xxy、npara、nxxy 及び tdxxy は g13becからのパラメータを表しています。 配列 xxy は長さが(nxxy+nfv)*tdxxy であると定義されていますが、上記で述べたように、季節 ARIMA モデルをフィッティングする場合、パラメータtdxxy は値 1 となり、式には表われる必要がありません。ここではxxy の大きさがどのようにルーチンドキュメントで定義されているかを表しており、わかりやすくするために式に含まれています。私たちはARIMA モデルから nfv 個の値を予測しているので、xxy 配列は元の系列(nxxy*tdxxy)と予測値(nfv*tdxxy)を保つのに十分な大きさが必要となります。
一旦、残りの入力パラメータが取り込まれると、モデルパラメータの推定は以下を使用して行うことができます。

```
nag_tsa_multi_inp_model_estim(&arimav, nseries, &transfv, para, npara,
                              nxxy, xxy, tdxxy, sd, &rss, &objf, &df,
                              G13_DEFAULT, NAGERR_DEFAULT);
```

ARIMA モデルパラメータが推定されると、モデルは g13bjc (nag_tsa_multi_inp_model_forecast)を呼び出すことによって予測値の生成に使用することができます。パラメータnev は出力系列の測定値の数になり、以下で表されます。

```
nev = nxxy;
```

入力系列はありません。したがって、rmsxy は初期化する必要がありません。mrx と parx はnullに設定することができ、tdmrx、 ldparx 及び tdparx は 1 に設定することができます。

```
mrx = 0;
parx = 0;
tdmrx = ldparx = tdparx = 1;
```

他の入力パラメータが初期化されると、予測は以下を使用して行うことができます。

```
nag_tsa_multi_inp_model_forecast(&arimav, nseries, &transfv, para,
                                 npara, nev, nfv, xxy, tdxxy, rmsxy,
                                 mrx, tdmrx, parx, ldparx, tdparx,
                                 fva, fsd, G13_DEFAULT, NAGERR_DEFAULT);
```