

TIMSAC for R package

統計数理研究所

November 1, 2006

1 はじめに

TIMSAC(TIME Series Analysis and Control program package) は, 統計数理研究所で開発された時系列データの解析, 予測, 制御のための総合的プログラムパッケージです. オリジナル TIMSAC(TIMASAC-72) は 1972 年に発表され, その後, TIMSAC シリーズとして TIMSAC-74, TIMSAC-78, TIMSAC-84 が Computer Science Monograph¹ に発表されました. 工業プロセスの最適制御, 経済変動の分析等広い分野で実際に利用されています. TIMSAC の特徴としては, 情報量規準の考え方をういた時系列解析プログラムであることが挙げられます. TIMSAC-72 では FPE(Final Prediction Error), TIMSAC-74 以降では AIC(Akaike Information Criterion), TIMSAC-78 のベイズ型モデルでは ABIC(Akaike Bayesian Information Criterion) も用いられています.

TIMSAC は FORTRAN で書かれたプログラムですが, 最近その一部のプログラムの計算処理機能をライブラリ化した Windows のための DLL(Dynamic Link Library) と Linux のための Shared Library を作成しました. ユーザーが作成した FORTRAN, C, Java のプログラムにこのライブラリをリンクすることにより, より扱い易い環境が実現されました.

一方, R はフリーな統計処理言語かつ環境です. R は配列演算を基本に設計されているため多次元配列の扱いに適している上, グラフィック関数も豊富であり, かつ FORTRAN や C のサブルーチンを簡単に呼び出せるインタフェースを備えています. TIMSAC の R パッケージは, R 言語を用いて前述の TIMSAC のサブルーチンを呼び出し, 必要であればその解析結果等をグラフィック表示する関数群です. 実際に R の中で呼び出されるライブラリは, R の拡張としてパッケージを構築する際前述のライブラリと同じソースから g77, gcc を使って作成されたものです.

¹H. Akaike, E. Arahata, T. Ozaki: TIMSAC-74, A Time series analysis and control program package (1) & (2), *Computer Science Monographs*, No.5 & 6, The Institute of Statistical Mathematics, Tokyo, 1975-1976

H. Akaike, G. Kitagawa, E. Arahata, F. Tada: TIMSAC-78, *Computer Science Monographs*, No.11, The Institute of Statistical Mathematics, Tokyo, 1979

H. Akaike, T. Ozaki, M. Ishiguro, Y. Ogata, G. Kitagawa, Y.-H. Tamura, E. Arahata, K. Katsura, Y. Tamura: TIMSAC-84 Part 1 & Part 2, *Computer Science Monographs*, No.22 & 23, The Institute of Statistical Mathematics, Tokyo, 1985

なお，本研究の一部は，情報・システム研究機構 新領域融合研究センターの融合研究プロジェクト「機能と帰納：情報化時代にめざす科学的推論の形」の一環として実施されています．

2 本パッケージの関数について

ここでは，発表された年次別にパッケージの関数とそこで仮定される時系列モデルおよび情報量規準について簡単に説明します．詳しくは，R Help をご覧ください．

2-1 TIMSAC-72 を利用する R の関数

autcor() : 直接法による自己共分散関数の計算
mulcor() : 直接法による相互共分散関数の計算
fftcor() : FFT による自己・相互共分散関数の計算
auspec() : BT 法によるパワースペクトル推定
mulspe() : BT 法によるクロススペクトル推定
sglfre() : 周波数応答関数の推定 (一入力)
mulfre() : 周波数応答関数の推定 (多入力)
fpeaut() : 一変量 AR モデルのあてはめ (Yule-Walker 法)
fpec() : 多変量 AR モデルのあてはめ (Yule-Walker 法)
mulnos() : ノイズ寄与率の計算
raspec() : 有理型スペクトル推定 (一変量)
mulrsp() : 有理型スペクトル推定 (多変量)
optdes() : 最適制御系の設計
optsim() : 最適制御系のシミュレーション
wnoise() : 白色ガウス雑音の生成

一変量 AR (AutoRegressive, 自己回帰) モデル :

定常な時系列 $y(t)$ が与えられたとき, この時系列の AR モデルは

$$y(t) = a(1)y(t-1) + \cdots + a(p)y(t-p) + u(t)$$

で表現されます．ただし, $u(t)$ はその平均は 0 で分散が σ^2 のホワイトノイズです．AR モデルをあてはめるとは, 次数 p を決定し, AR 係数 (自己回帰係数) $a(1), \dots, a(p)$ と分散 σ^2 を推定することになります．

多変量 AR (AutoRegressive, 自己回帰) モデル :

一変量の場合と同様, k 次元の定常時系列についても

$$y(t) = \sum_{m=1}^p A(m)y(t-m) + u(t)$$

の形に書くことができます。ただし、このとき $A(m)$ は $k \times k$ 行列であり、 $u(t) = (\epsilon_1(t), \epsilon_2(t), \dots, \epsilon_k(t))'$ はその各要素の平均が 0 であり、また共分散行列 $[\sigma_{ij}]$ を持つ k 次元のホワイトノイズです。

FPE(Final Prediction Error, 最終予測誤差) :

TIMSAC-72 で用いられる情報量規準 FPE とは期待される予測誤差の 2 乗の値であり、一変量 p 次の AR モデルの FPE は、データ数 N 、イノベーション分散の推定値を $\hat{\sigma}^2$ とすると

$$FPE = \frac{N + p + 1}{N - p - 1} \hat{\sigma}^2$$

で求められます。FPE の値が小さいほどよいモデルとみなされます。(fpeaut)
多変量の場合も同様の定義が用いられます。(fpec)

2-2 TIMSAC-74 を利用する R の関数

armafit() : ARMA モデルのあてはめ (一変量)
 autoarmafit() : ARMA モデルのあてはめ (一変量)
 canarm() : 正準相関解析 (一変量)
 covgen() : 与えられたスペクトルに対応する共分散関数の作成
 canoca() : 正準相関解析 (多変量)
 markov() : ARMA モデルのあてはめ (多変量)
 prdctr() : ARMA モデルによる予測
 simcon() : ARMA モデルによる最適制御系の設計, シミュレーション
 nonst() : 局所定常 AR モデルのあてはめ (一変量)
 thirmo() : 三次モーメントの計算
 bispec() : バイスpekトルの計算

一変量 ARMA(AutoRegressive Moving Average, 自己回帰移動平均) モデル :

ARMA モデルは、時系列 $y(t)$ を過去の観測値とホワイトノイズの現在および過去の値の線形和で

$$y(t) - \sum_{l=1}^p a(l)y(t-l) = u(t) - \sum_{m=1}^q b(m)u(t-m)$$

のように表現されます。ここで、 $y(t)$ は時系列であり、 $u(t)$ は平均が 0、分散 σ^2 のホワイトノイズです。(armafit, autoarmafit)

多変量 ARMA (AutoRegressive Moving Average, 自己回帰移動平均) モデル :

多変量の場合, i 番目の時系列 $y_i(t)$ は

$$y_i(t) - \sum_{l=1}^p \sum_{j=1}^k A_{ij}(l)y_j(t-l) = u_i(t) - \sum_{m=1}^q B_i(m)u_i(t-m) \quad (i = 1, 2, \dots, k)$$

のように表現されます. ただし, $(u_1(t), \dots, u_k(t))$ は k 次元のホワイトノイズです. (markov)

AIC (Akaike Information Criterion) :

AIC (赤池情報量基準) は

$$\begin{aligned} AIC &= -2l(\hat{\theta}) + 2k \\ &= -2(\text{最大対数尤度}) + 2(\text{パラメータ数}) \end{aligned}$$

で定義されます. したがって Yule-Walker 法の場合, データ数 N で p 次の一変量 AR モデルでは

$$AIC = N \log(2\pi \hat{\sigma}_p^2) + N + 2(p+1)$$

となり, 前述の FPE とは次の関係があります.

$$N \log FPE(p) = AIC - N \log(2\pi) - N - 2$$

ただし, 最小二乗法を用いた場合は, 解析に使用するデータの数が異なっているので上式の AIC とは異なります.

2-3 TIMSAC-78 を利用する R の関数

unimar() : AR モデルのあてはめ (一変量, 最小二乗法)

unibar() : AR モデルのあてはめ (一変量, ベイズ法)

bsubst() : 非線形モデルのあてはめ (一変量, ベイズ法)

mulmar() : AR モデルのあてはめ (多変量, 最小二乗法)

mulbar() : AR モデルのあてはめ (多変量, ベイズ法)

perars() : 周期的 AR モデルのあてはめ

mlocar() : 局所定常 AR モデルのあてはめ (一変量, 最小二乗法)

blocar() : 局所定常 AR モデルのあてはめ (一変量, ベイズ法)

mlomar() : 局所定常 AR モデルのあてはめ (多変量, 最小二乗法)

blomar() : 局所定常 AR モデルのあてはめ (多変量, ベイズ法)

exsar() : AR モデルのあてはめ (一変量, 最尤法)

xsarma() : ARMA モデルのあてはめ (一変量, 最尤法)

このパッケージでは, 最小二乗法 (least squares method) の計算に主としてハウスホルダー変換を利用します. 最小二乗法では, 予測値と実際の観測値との誤差の二乗和を最小にするようにパラメータを定めます. 一変量 AR モデル

$$y(t) = a(1)y(t-1) + \dots + a(p)y(t-p) + u(t)$$

について考えるとき，デザイン行列 Z およびベクトル y と a を次のように定義します．

$$Z = \begin{bmatrix} y(p) & y(p-1) & \dots & y(1) \\ y(p+1) & y(p) & \dots & y(2) \\ \vdots & \vdots & & \vdots \\ y(N-1) & y(N-2) & \dots & y(N-p) \end{bmatrix}, y = \begin{bmatrix} y(p+1) \\ y(p+2) \\ \dots \\ y(N) \end{bmatrix}, a = \begin{bmatrix} a(1) \\ \vdots \\ a(p) \end{bmatrix} \quad (1)$$

まず $(n-p) \times (p+1)$ 行列

$$X = [Z|y]$$

を作り，ハウスホルダー変換 U を適用すると次のような上三角行列 S に変形できます．

$$UX = S = \begin{bmatrix} s_{11} & \dots & s_{1p} & s_{1,p+1} \\ & \ddots & \vdots & \vdots \\ & & s_{pp} & s_{p,p+1} \\ & & & s_{p+1,p+1} \end{bmatrix} \quad (2)$$

ハウスホルダー変換ではユークリッドノルムは変化しないので

$$\|Za - y\|^2 = \|UZa - Uy\|^2 \quad (3)$$

であることから，AR 係数 $a_p(1), \dots, a_p(p)$ は，次式の解として求められます．

$$\begin{bmatrix} s_{11} & \dots & s_{1p} \\ & \ddots & \vdots \\ 0 & & s_{pp} \end{bmatrix} \begin{bmatrix} a_p(1) \\ \vdots \\ a_p(p) \end{bmatrix} = \begin{bmatrix} s_{1,p+1} \\ \vdots \\ s_{p,p+1} \end{bmatrix} \quad (4)$$

また，イノベーション分散 $\hat{\sigma}_p^2$ は

$$\hat{\sigma}_p^2 = \frac{s_{p+1,p+1}^2}{N-p}$$

で求められます．したがって， k 次 ($k < p$) の AR モデルのイノベーション分散は次の式で求めることができます．

$$\hat{\sigma}_k^2 = \frac{1}{N-p} \sum_{i=k+1}^{p+1} s_{i,p+1}^2$$

最大対数尤度は

$$-\frac{N-p}{2} \log(2\pi\hat{\sigma}_k^2) - \frac{N-p}{2}$$

であることから，AIC は次式で与えられます．

$$AIC(k) = (N-p)\log(2\pi\hat{\sigma}_k^2) + N - p + 2(k+1)$$

すでに述べた TIMSAC-72 の Yule-Walker 法と最小二乗法とは，同じ一変量 AR モデルのあてはめを行っていますが AIC が異なります．これは，最小二乗法では最初の p 個のデータを用いていませんが，Yule-Walker 法では全てのデータを用いているからです．(unimar)

多変量時系列の場合も同様の議論ができます．(mulmar)

ABIC

ベイズ法では，AIC が対数尤度の推定量であることを用いているので，次数 k のモデルの尤度は

$$f(y|k) = \exp\left(-\frac{1}{2}AIC(k)\right)$$

で与えられます．また， k 次のモデルが得られる事後確率は

$$\pi(k|y) = \frac{f(y|k)\pi(k)}{\int f(y|k)\pi(k)dk}$$

で与えられます．ここで， $\pi(k)$ は事前確率です．AR モデルのベイズ推定は，事後確率を重みとして次数の異なるモデルを平均することによって得られます．ベイズモデルのイノベーションの推定値を $\hat{\sigma}_B^2$ とすれば，情報基準量 ABIC は

$$ABIC = (N - p)\log\hat{\sigma}_B^2 + 2\left(\sum_{i=1}^p d(i)^2 + 1\right)$$

で定義され，この値が小さいほどよいベイズモデルということになります．ただし， $d(i)$ は

$$d(i) = \sum_{k=i}^p \pi(k|y)$$

であり， $\pi(k|y)$ は前述の事後確率です．(unibar, mulbar)

局所定常

一変量局所定常 AR モデルは， l 番めの小区間では時系列 $y(t)$ が

$$y(t) = a^l(1)y(t-1) + \cdots + a^l(p)y(t-p) + \epsilon^l(t) \quad s_{l-1} < s \leq s_l$$

のように表されます． N 個のデータをいくつかのブロックに分け，ブロックごとに AR モデルをあてはめようとするのが局所定常の考え方です．連続するブロックで異なる AR モデルをあてはめた方がよいか，二つのブロックに共通の AR モデルをあてはめた方がよいかを判断するために次の二つの AIC を比較します．

$$AIC_1 = (n_1 - p)\log(2\pi\hat{\sigma}_{p_0}^2) + n_2\log(2\pi\hat{\sigma}_{p_1}^2) + n_1 + n_2 - p + 2(p_0 + p_1 + 2)$$

$$AIC_2 = (n_1 + n_2 - p)\log(2\pi\hat{\sigma}_{p_2}^2) + n_1 + n_2 - p + 2(p_2 + 1)$$

ここで n_1 と n_2 は各ブロックのデータ長です． p_0, p_1, p_2 はそれぞれ $\{y(1), \dots, y(n_1)\}, \{y(n_1+1), \dots, y(n_1+n_2)\}, \{y(1), \dots, y(n_1+n_2)\}$ にあてはめた AR モデルの次数です．前者の方が小さい場合二つのブロックに異なる AR モデルをあてはめ，後者の方が小さい場合共通の AR モデルをあてはめます．(mlocar)

多変量時系列の場合も同様の議論ができます．(mlomar)

2-4 TIMSAC-84 を利用する R の関数

decomp() : 時系列データの分解

decomp は、状態空間表現を利用した時系列データを複数の成分に分解するためのプログラムです。すなわち、時系列を $y(t)$ を

$$y(t) = T(t) + AR(t) + S(t) + TD(t) + W(t)$$

ここで $T(s)$: トレンド成分
 $AR(t)$: AR 成分
 $S(t)$: 季節変動成分
 $TD(t)$: 曜日効果
 $W(t)$: 白色雑音

という和の形に分解します。