信号解析　第８回講義録

日時：２０１５年6月２２日
講義内容：ＡＲモデルの推定
担当者：情報知能工学科　小島史男

1　はじめに

いままで信号の統計処理方法および統計モデルについて学習してきました。今回からはモデルを使って信号を“読む”ことをおこないましょう。ARMA モデルのなかでも AR モデルは取り扱いが比較的容易です。本日の講義では時系列信号をＡＲモデルに当てはめる計算法について説明します。教科書第７章の内容です。

2　AR モデルの復習

AR モデルは ARMA モデルの特別の場合で、ガウス型白色雑音系列 $v[n]$ を用いて

$$y[n] = \sum_{i=1}^{m} a_i y[n-i] + v[n]$$

で与えられることはすでに学習しました。$v[n]$ は平均 0 分散 σ^2 の正規分布に従う白色雑音で、時系列の過去の信号 $y[n-i]$ とは独立と仮定していました。今回の目的は任意の時系列信号 $Y_N = \{y[n]\}_{n=1}^N$ が与えられたとき、このデータを AR モデルによって当てはめを行おうということです。そのためにするべきことは何でしょうか？　まず AR モデルの次元 m を決めなければなりません。これは赤池法で AIC の値を求めれば良いことになります。適当な次元がわかれば、あと決めるべきことは AR モデルの m 次元の係数ベクトル $\{a_i\}_{i=1}^m$ を推定することになります。もちろん白色雑音の分散 σ^2 も推定する必要があります。AR モデルではこの一連の作業が非常に見通しよくできるのです。

3　AR モデルの尤度関数

まず AR モデルの統計的性質を調べてみましょう。次数 m のときの時系列 $\{y[n]\}_{n=1}^N$ の各要素の自己共分散関数 C_k は

$$\begin{aligned} C_0 &= \sum_{i=1}^{m} a_i C_i + \sigma^2 \\ C_k &= \sum_{i=1}^{m} a_i C_{k-i} \quad (k = 1, 2, \cdots) \end{aligned}$$

であたえられることは前回の講義で学習しました。このことから AR モデルの尤度関数をもとめていきましょう。このモデルは線形であり、かつ入力 $v[n]$ は正規型確率密度関数に従います。このことはその出力である $y[n]$ も正規型となることを意味します。入力時系列信号 $\{v[n]\}_{n=1}^N$ の統計モデルは白色雑音が完全に独立な時系列であるので、その確率密度関数は

$$g(v[1], v[2], \cdots, v[N]) = g(v[1])g(v[2]) \cdots g(v[N]) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N \prod_{n=1}^{N} \exp\left(-\frac{v[n]^2}{2\sigma^2}\right)$$

のように直積で書くことができます。しかし、出力時系列信号 $\{y[n]\}_{n=1}^N$ の統計モデルはどうでしょうか。この分布は正規分布にはちがいがありませんが、それぞれは独立ではありません。この場合 次の正規型確率密度関数が統計モデルとなります。すなわち

$$g(y[1], y[2], \cdots, y[N]) = (2\pi)^{-\frac{N}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} Y_N^T \Sigma^{-1} Y_N\right)$$

$$\Sigma = \begin{bmatrix} C_0 & C_1 & \cdots & C_{N-1} \\ C_1 & C_0 & \cdots & C_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N-1} & C_{N-2} & \cdots & C_0 \end{bmatrix}$$

で与えられます。推定すべきは $\theta = \{a_1, a_2, \cdots, a_m, \sigma^2\}$ の $(m+1)$ 次元のパラメータベクトルです。したがってその赤池情報量基準は

$$\mathrm{AIC} = -2\log\left\{g(y[1], y[2], \cdots, y[N] \mid \hat{\theta})\right\} + 2(m+1)$$

で与えられます。復習になりますが、適当な最大次元 $M(m \leq M)$ までの AR モデルの AIC を計算し、そのなかで最小値となる次数を最適な次数としてモデルを決定することになります。上の式の第 1 項は最大対数尤度関数であり、$\hat{\theta}$ はそれぞれに次元において最大尤度となる最適パラメータを意味します。このパラメータの決定方法について以下考察をすすめましょう。

4 ユールウオーカーの方法

第 7 回の講義で説明しましたが、次の AR モデルの自己共分散関数はユールウオーカーの方程式で与えられます。ところで時系列信号 $\{y[n]\}_{n=1}^{N}$ から標本自己共分散関数が

$$\hat{C}_k = \frac{1}{N}\sum_{n=k+1}^{N}(y[n]-\hat{\mu})(y[n-k]-\hat{\mu}) \quad \left(\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} y[n]\right)$$

で計算できることを 2 回目の講義で学習しました。もしこれが自己共分散の十分良い近似となっているならば、さきほどのユールウオーカーの方程式にこれを代入して

$$\begin{aligned} \hat{C}_0 &= \sum_{i=1}^{m} a_i \hat{C}_i + \sigma^2 \\ \hat{C}_k &= \sum_{i=1}^{m} a_i \hat{C}_{k-i} \quad (k = 1, 2, \cdots) \end{aligned}$$

を得ることができます。2 番目の式はよく見ると AR 係数 $\{a_i\}_{i=1}^{m}$ を未知数とする以下の連立方程式に帰着されます。

$$\begin{bmatrix} \hat{C}_0 & \hat{C}_1 & \cdots & \hat{C}_{m-1} \\ \hat{C}_1 & \hat{C}_0 & \cdots & \hat{C}_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}_{m-1} & \hat{C}_{m-2} & \cdots & \hat{C}_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \hat{C}_1 \\ \hat{C}_2 \\ \vdots \\ \hat{C}_m \end{bmatrix}$$

この方程式を解けば容易に AR 係数を求めることができます。また 1 番目の式に求めた AR 係数 $\{\hat{a}_i\}_{i=1}^{m}$ の推定値を代入すると正規型白色雑音の分散の推定値

$$\hat{\sigma}^2 = \hat{C}_0 - \sum_{i=1}^{m} \hat{a}_i \hat{C}_i$$

も同時に計算できることになります。こうして求めたパラメータのことをユールウオーカーの推定値と呼びます。ところで $\hat{\theta} = \{\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_m, \hat{\sigma}^2\}$ をさきほどの対数尤度関数に代入したときにはたして最大値をとるのでしょうか。これは以下のようにして確認することができます。AR モデルによる予測誤差分散は

$$\begin{aligned} E(v[n]^2) &= E\left(y[n] - \sum_{i=1}^{m} a_i y[n-i]\right)^2 \\ &= C_0 - 2\sum_{i=1}^{m} a_i C_i + \sum_{i=1}^{m}\sum_{j=1}^{m} a_i a_j C_{i-j} \end{aligned}$$

となります。そこで係数パラメータで偏微分すると、停留点は

$$\frac{\partial E(v[n]^2)}{\partial a_i} = -2C_i + 2\sum_{j=1}^{m} a_j C_{i-j} = 0$$

を満足するはずです。つまり共分散関数 $\{C_0, C_2, \cdots C_m\}$ にその標本共分散 $\{\hat{C}_0, \hat{C}_2, \cdots \hat{C}_m\}$ を代入して、先ほどの連立方程式を求めること自体が予測誤差分散を近似的に最小にしていることになるわけです。

5　レビンソンのアルゴリズム

これまでの手順をまとめると $m = 1$ から $m = M$ までの次数ごとに、連立方程式を解くことで AIC(m) を求め、最適な次数決定とパラメータ $\hat{\theta} = \{\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_m, \hat{\sigma}^2\}$ を決めればよいことになりますが、次数が大きくなるにつれて方程式をいちいち求めていくのはめんどうです。レビンソンのアルゴリズム (Levinson's Algorithm) ではこれまでの計算を逐次的に効率的に解く方法です。このアルゴリズムを以下に示します。

Step 1: 初期値の設定をおこなう。

$$\hat{\sigma}_0^2 = \hat{C}_0, \quad \mathrm{AIC}(0) = N(\log 2\pi\hat{\sigma}_0^2 + 1) + 2$$

Step 2: AR 次数 $m = 1, 2, \cdots, M$ について下記の反復演算をおこなう。

$$\begin{aligned}
\hat{a}_m^m &= \left(\hat{C}_m - \sum_{j=1}^{m-1} \hat{a}_j^{m-1}\hat{C}_{m-j}\right) \Big/ \hat{\sigma}_{m-1}^2 \\
\hat{a}_i^m &= \hat{a}_i^{m-1} - \hat{a}_m^m \hat{a}_{m-i}^{m-1} \\
\hat{\sigma}_m^2 &= \hat{\sigma}_{m-1}^2 \left\{1 - (\hat{a}_m^m)^2\right\} \\
\mathrm{AIC}(m) &= N(\log 2\pi\hat{\sigma}_m^2 + 1) + 2(m+1)
\end{aligned}$$

上記のアルゴリズムでの AR 係数 $\hat{a}_i^m$ は AR 次数が m のときの i 番目の AR 係数の推定値を意味しています。

6　R をつかっておさらい

ここでは、適当な次元の AR モデル (たとえば 77 ページの例) で求めたデータを観測された信号 $\{y[n]\}_{n=1}^N$ から AR モデルを逆に作ってみることにします。次の図は人工的に作成された時系列信号とその標本自己相関関数およびペリオドグラムです。教科書 84 ページの図 6.4 のスペクトルとペリオドグラムとの比較からこの時系列信号は AR モデルで近似できそうです。そこでレビンソンの計算アルゴリズムを使ってモデリングを試みます。時系列信号の長さは $n = 256$、AR モデルの最高次数は $M = 20$ としておきます。以下は標本自己相関関数の計算およびレビンソンの計算アルゴリズムのプログラム例です。

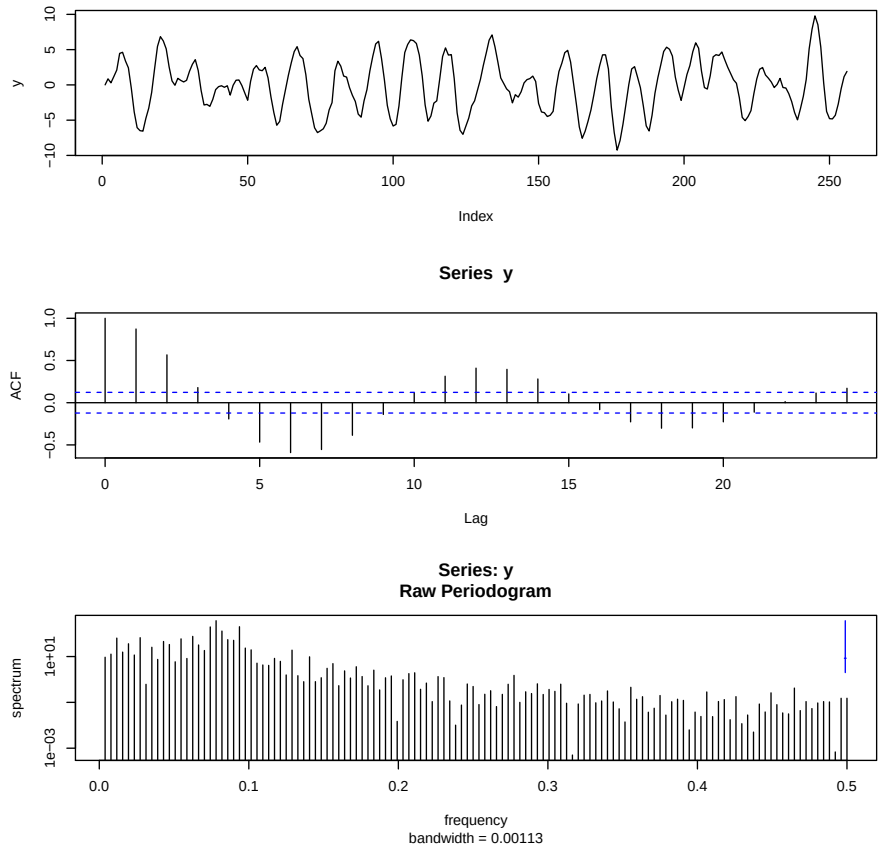

Figure 1: 実験で用いた時系列信号とその標本自己相関関数およびペリオドグラム

標本自己共分散関数の計算

```
acfv <- function(n, mmax, y) {
    chat <- numeric(mmax+1);
    sum = 0;
    for (i in 1:n) {
        sum <- sum + y[i]
    }
    muhat <- sum / n;
    for(k in 0:mmax) {
        sum <- 0;
        for(j in k:(n-1)) {
            sum <- sum + (y[j+1] - muhat) * (y[j+1-k] - muhat)
        }
        chat[k+1] <- sum / n;
    }
    return(chat);
}
```

レビンソンの計算アルゴリズム

```
levinson <- function(n, mmax, chat) {
    theta <- matrix(0, mmax+1, mmax+2);
    theta[1,mmax+1] <- chat[1];
    theta[1,mmax+2] <- n * (log(2*pi*theta[1,mmax+1]) + 1) + 2;
    theta[2,1] <- chat[2] / theta[1,mmax+1];
    theta[2,mmax+1] <- theta[1,mmax+1] * (1 - theta[2,1]^2);
    theta[2,mmax+2] <- n * (log(2*pi*theta[2,mmax+1]) + 1) + 4;
    for (m in 2:mmax) {
        sum <- chat[m+1];
        listj = 1:(m-1);
        for (j in listj) {
            sum <- sum - theta[m,j] * chat[m+1-j];
        }
        theta[m+1,m] <- sum / theta[m,mmax+1];
        listi = 1:m-1;
        for (i in listi) {
            theta[m+1,i] <- theta[m,i] - theta[m+1,m] * theta[m,m-i];
        }
        theta[m+1,mmax+1] <- theta[m,mmax+1] * (1 - theta[m+1,m]^2);
        theta[m+1,mmax+2] <- n * (log(2*pi*theta[m+1,mmax+1]) + 1) + 2 * (m + 1);
    }
    return(theta);
}
```

これを用いた計算結果を以下に示します。θ は AR の最高次数 $+1$ の行と $+2$ の列を有する（ここでは 21×22）行列で、それぞれの行が AR モデルの次数に相当します。ただし第 1 行は 0 次元で AR 係数は含まれません。たとえば m 行目は $(m-1)$ 次の AR モデルの係数が 1 列目から $(m-1)$ 列までに格納されており、残りの係数を 0 に設定、右端から 2 列前が白色雑音の分散推定値、最後の列が AIC(m) の値を示しています。

計算結果

```
> n <- 256
> mmax <- 20
> source("acfv.R")
> chat <- acfv(n,mmax,y)
> source("levinson.R")
> theta <- levinson(n,mmax,chat)
```

この結果を用いて AIC(m) の値をグラフ化してみました。AIC の値は $m = 2$ を境にして上昇に転じていることがわかります。プログラム AIC.R は最適な次数とそのときの AR 係数および白色雑音の分散推定値が表示できるようになっています。

AIC

```
aic <- function(mmax,theta) {
    aicv <- numeric(mmax+1);
    for(i in 1:(mmax+1)) {
        aicv[i] <- theta[i,mmax+2];
    }
    aicm <- which.min(aicv)-1;
    cat("Minimum of AIC : ", aicm, "\n");
    for(i in 1:aicm) {
        cat("a[",i,"]=", theta[aicm+1,i], "\n");
    }
    cat("sigma2=", theta[aicm+1,mmax+1], "\n");
    return(aicv);
}
```

実行画面（その 1）

```
> source("AIC.R")
> eval <- aic(mmax,theta)
Minimum of AIC :  2
a[ 1 ]= 1.591856
a[ 2 ]= -0.82323
sigma2= 1.042901
> x <- c(0:mmax)
> plot(x,eval, type="h")
> dev.copy2eps(file="aic.eps",width=10,height=6)
```

この時系列信号は $AR(2)$ モデルがもっとも良い近似であるといえます。$m = 2$ のときの AR 係数は

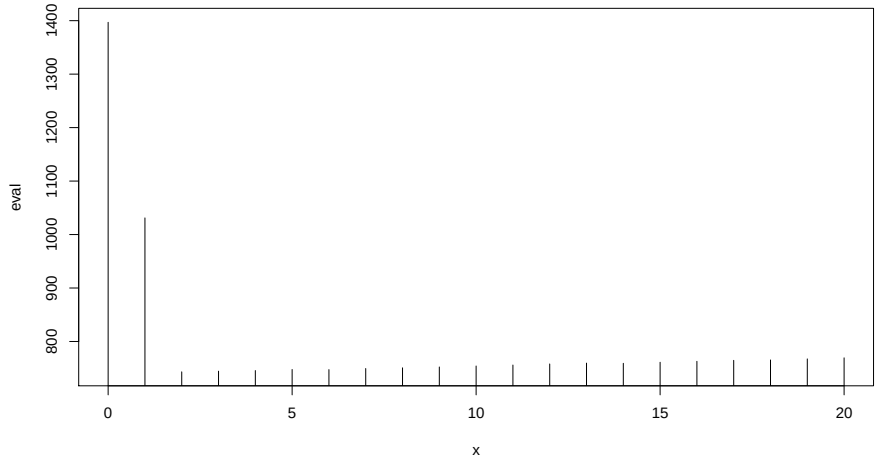

Figure 2: AIC の値の変化

$$\hat{a}_1 = 1.591856 \quad \hat{a}_2 = -0.82323 \qquad \hat{\sigma}^2 = 1.042901$$

となっていることがわかります。よって時系列信号のモデルは下記で与えられます。

$$y[n] = 1.591856y[n-1] - 0.82323y[n-2] + v[n]$$

このモデルに関して、平均 0、分散 1.042901 の疑似白色雑音を生成してシミュレーションを行いました。次の図では実線が元の時系列信号、点線が求めた AR モデルの標本過程です。もちろん両者は一致しませんが、そのプロファイルをよく捉えていることがわかります。最後にここで用いた時系列信号ですが、これは前回講義で示した教科書７７ページの AR(2) モデルから作った信号です。モデルの AR 係数および白色雑音の分散は

$$a_1 = 0.9\sqrt{3} = 1.558846 \quad a_2 = -0.81 \qquad \sigma^2 = 1$$

でしたので、推定値はかなり近いものだということがわかります。もちろん実際の時系列信号はこんなことはわかりません。

実行画面（その２）

```
> a <- c(1.591856,  - 0.82323)
> sigma2<- 1.042901
> ym <- armodel(n,2,a,sigma2)
> plot(y,type="l",lty=1)
> par(new=T)
> plot(ym,type="l",lty=2)
> dev.copy2eps(file="comparison.eps",width=10,height=6)
```

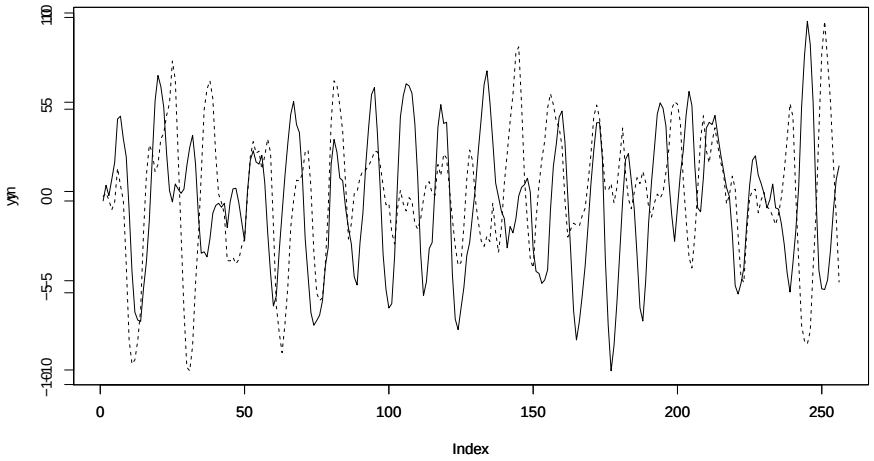

Figure 3: 時系列信号と AR(3) モデルの比較