Estimating model parameters – AR(2) Simulation

PRACTICAL TIME SERIES ANALYSIS

THISTLETON AND SADIGOV

Objectives

- Estimate variance of a white noise in a simulated AR(2) processes
- Estimate coefficients of a simulated AR(2) process using Yule-Walker equations in matrix form

AR(2) process (with mean zero)

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t$$

where

$$Z_t \sim Normal\ (0, \sigma_Z^2)$$

We simulate this process for

$$\phi_1 = \frac{1}{3}, \phi_2 = \frac{1}{2}, \sigma_Z = 4$$

Yule –Walker equaitons

We estimate coefficients of the model

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t$$

by first finding r_1, r_2 using acf() routine, then solving the system of equations

$$\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 1 & r_1 \\ r_1 & 1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \end{bmatrix}$$

σ_Z Estimation

Since

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + Z_t$$

We have

$$Var(X_t) = \phi_1^2 Var(X_{t-1}) + \phi_2^2 Var(X_{t-2}) + 2\phi_1\phi_2 Cov(X_{t-1}, X_{t-2}) + \sigma_Z^2$$

Thus

$$\sigma_Z^2 = \gamma(0)\left[1 - \phi_1^2 - \phi_2^2 - \frac{2\phi_1\phi_2\gamma(1)}{\gamma(0)}\right] = \gamma(0)[1 - \phi_1^2 - \phi_2^2 - 2\phi_1\phi_2\rho_1]$$

Since

$$\begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} = \begin{bmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}$$

we have

$$\rho_1 = \phi_1 + \rho_1 \phi_2$$

$$\rho_2 = \phi_1 \rho_1 + \phi_2$$

$$1 - \phi_1^2 - \phi_2^2 - 2\phi_1\phi_2\rho_1$$

$$= 1 - \phi_1^2 - \phi_1\phi_2\rho_1 - \phi_2^2 - \phi_1\phi_2\rho_1$$

$$= 1 - \phi_1(\phi_1 + \rho_1\phi_2) - \phi_2(\phi_1\rho_1 + \phi_2)$$

$$= 1 - \phi_1\rho_1 - \phi_2\rho_2$$

σ_Z Estimation cont.

Thus,

$$\sigma_Z^2 = \gamma(0)[1 - \phi_1\rho_1 - \phi_2\rho_2]$$

Yule –Walker estimator

$$\hat{\sigma}_Z^2 = c_0\,[1 - \hat{\phi}_1 r_1 - \hat{\phi}_2 r_2]$$

Simulation

- Number of data points, $n = 10000$

Routines that we use:

- arima.sim() # simulating
- plot() # plotting the series
- acf() # autocorrelation function
- matrix(,m,n) # matrix with dimensions m by n
- solve(R,b) # finds the solution to Rx=b

Code details

- sigma=4
- phi[1:2]=c(1/3,1/2)
- n=10000
- set.seed(2017)
- ar.process=arima.sim(n, model=list(ar=c(1/3,1/2)), sd=4)
- ar.process[1:5]

$$4.087685, 5.598492, 3.019295, 2.442354, 5.398302$$

Code details cont.

- r[1:2]=acf(ar.process, plot=F)$acf[2:3]

$$r[1] = 0.6814103$$
$$r[2] = 0.7255825$$

R=matrix(1,2,2) # matrix of dimension 2 by 2, with entries all 1's.

$$R = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

- R[1,2]=r[1] # only diagonal entries are edited
- R[2,1]=r[1] # only diagonal entries are edited

$$R = \begin{bmatrix} 1 & r_1 \\ r_1 & 1 \end{bmatrix}$$

Code details cont.

- b=matrix(r,2,1)# b- column vector entires from r

$$b = \begin{bmatrix} r[1] \\ r[2] \end{bmatrix} = \begin{bmatrix} 0.6814103 \\ 0.7255825 \end{bmatrix}$$

Code details cont.

- solve(R,b)

$$\begin{bmatrix} 0.3490720 \\ 0.4877212 \end{bmatrix}$$

- phi.hat=matrix(c(solve(R,b)[1,1], solve(R,b)[2,1]),2,1)

$$\hat{\phi}_1 = 0.3490720$$
$$\hat{\phi}_2 = 0.4877212$$

Code details cont.

- c0= acf(ar.process, type='covariance', plot=F)$acf[1]
- var.hat= c0*(1-sum(phi.hat*r))
- par(mfrow=c(3,1))
- plot(ar.process, main='Simulated AR(2)')
- acf(ar.process, main='ACF')
- pacf(ar.process, main='PACF')

Results

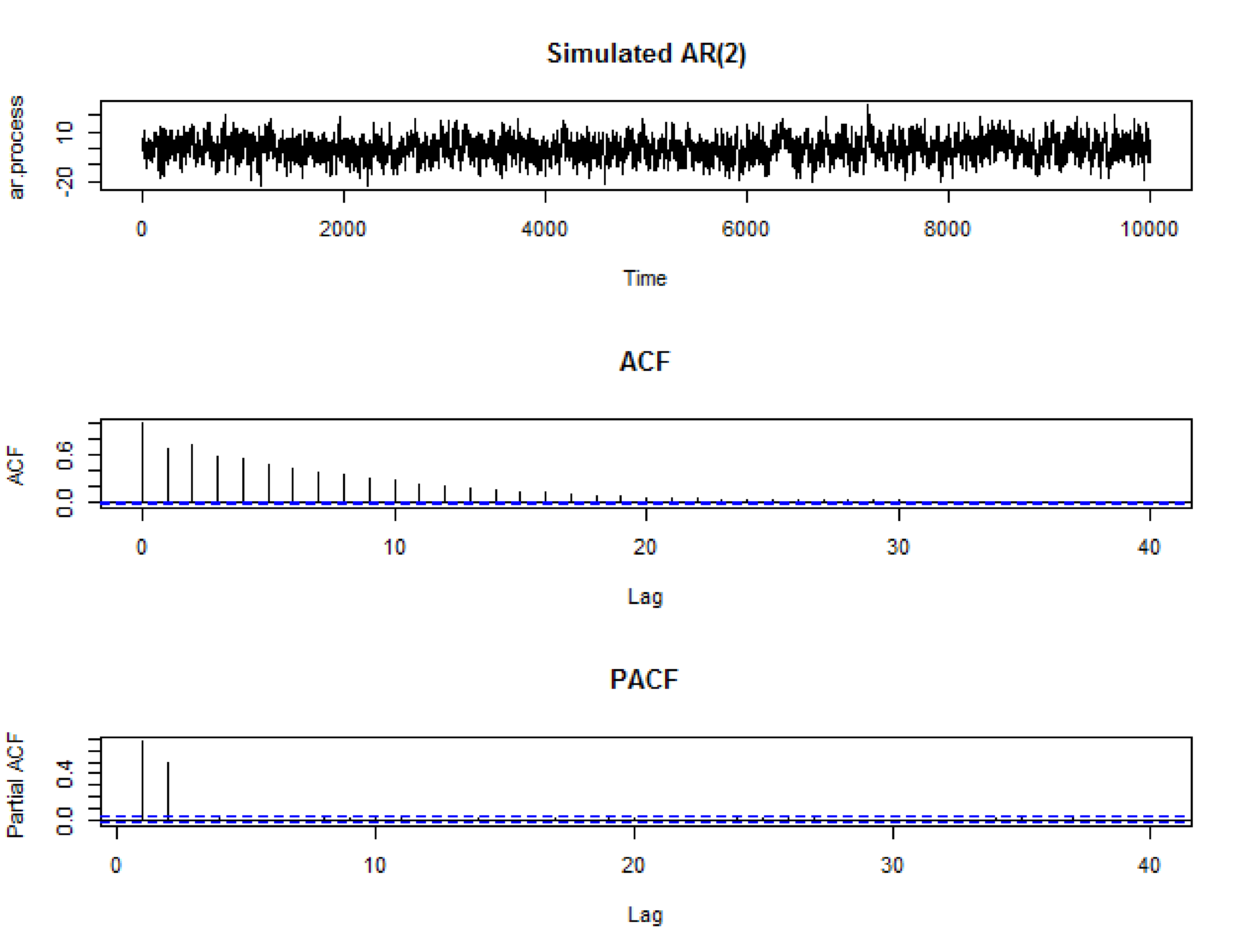
- $\phi_1 \approx \hat{\phi}_1 = 0.3490720$
- $\phi_2 \approx \hat{\phi}_2 = 0.4877212$
- $\sigma_Z^2 \approx \widehat{\sigma_Z^2} = 16.37169$

Actual Model

$$X_t = 0.\overline{3}\, X_{t-1} + 0.5X_{t-2} + Z_t, \qquad Z_t \sim N(0,16)$$

Fitted model

$$X_t = 0.3490720\, X_{t-1} + 0.4877212\, X_{t-2} + Z_t, \qquad Z_t \sim N(0,16.37169)$$



Simulated AR(2)
ar.process
-20
10
0
2000
4000
6000
8000
10000
Time
ACF
ACF
0.0
0.6
0
10
20
30
40
Lag
PACF
Partial ACF
0.0
0.4
0
10
20
30
40
Lag

What We've Learned

- Estimating model parameters of a simulated AR(2) process using Yule-Walker equations in a matrix form