Time Series Analysis

ARMA Models

Nicoleta Serban, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Parameter Estimation: Simulation Example

Georgia Tech

About This Lesson

AR Model: Linear Regression

```
# AR(2) process simulation
w2 = rnorm(1500)
b = c(1.2,-0.5)
ar2 = filter(w2,filter=b,method='recursive')
ar2 = ar2[1001:1500]
## Fit Linear Regression to AR(2)
data2 = data.frame(cbind(x1=ar2[1:498],x2=ar2[2:499],y=ar2[3:500]))
model2 = lm(y~x1+x2,data=data2)
summary(model2)
```

AR(2): $X_t = 1.2\, X_{t-1} - 0.5\, X_{t-2} + Z_t$

Georgia Tech

AR Model: Linear Regression (cont'd)

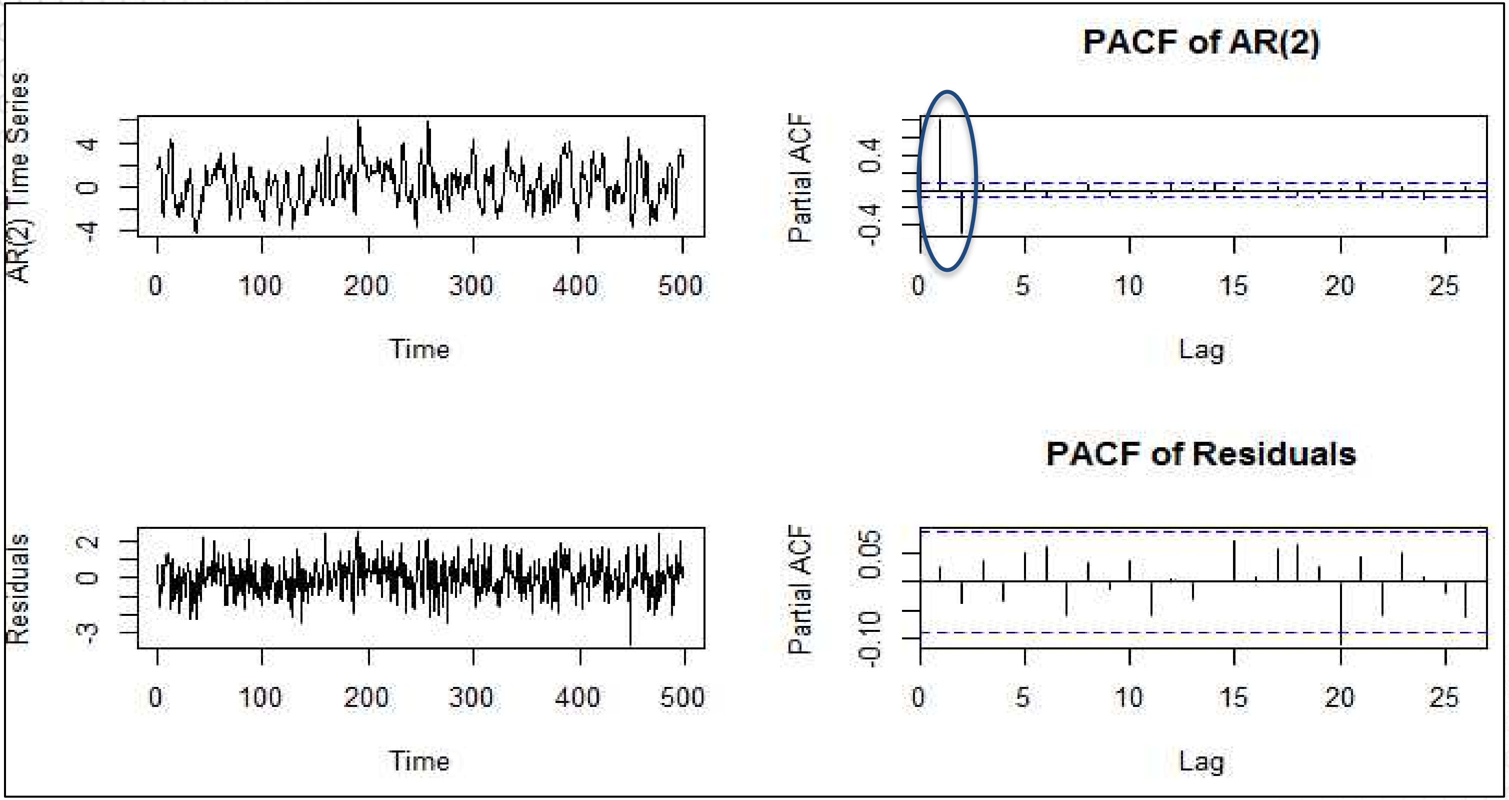
```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.05645    0.04391   1.286    0.199
x1          -0.48149    0.03940 -12.220   <2e-16 ***
x2           1.17750    0.03936  29.913   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.974 on 495 degrees of freedom
Multiple R-squared:  0.7177,    Adjusted R-squared:  0.7166
F-statistic: 629.3 on 2 and 495 DF,  p-value: < 2.2e-16
```

AR(2): $X_t = 1.2\, X_{t-1} - 0.5\, X_{t-2} + Z_t$

$H_0: \beta_1 = -0.5 \ vs \ H_{A:}: \beta_1 \neq -0.5$

```
> t.value = (-0.48149+0.5)/0.0394
> p.value = 2*(1-pnorm(t.value))
> p.value
[1] 0.6385001
```

AR Model: Linear Regression: Residuals



Linear Regression & AR Order

w2 = rnorm(1500)
b = 0.5
ar1 = filter(w2,filter=b,method='recursive')
ar1 = ar1[1001:1500]
 # Fit an AR(2) to an AR(1) process using linear regression
data3 = data.frame(cbind(x1=ar1[1:498],x2=ar1[2:499],y=ar1[3:500]))
model3 = lm(y~x1+x2,data=data3)
summary(model3)

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.05761    0.04394   1.311    0.190
x1          -0.02216    0.04491  -0.493    0.622
x2           0.50486    0.04494  11.234   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '

Residual standard error: 0.974 on 495 degrees of freedom
Multiple R-squared:  0.2444,    Adjusted R-squared:  0.2414
F-statistic: 80.07 on 2 and 495 DF,  p-value: < 2.2e-16
```

AR(1): $X_t = 0.5\, X_{t-1} + Z_t$

Georgia Tech

Yule-Walker & AR Estimation

```
## Fit an AR(3) to an AR(2) model
# Form gamma(1) & Gamma_3 Matrix
covf = acf(ar2,type='covariance',plot=FALSE)
Gammamatrix = matrix(0,3,3)
for(i in 1:3){
 if(i>1){
    Gammamatrix[i,] = c(covf$acf[i:2,,1],covf$acf[1:(3-i+1),,1]) }
 else{
    Gammamatrix[i,] = covf$acf[1:(3-i+1),,1] }
}
Gamma1 = covf$acf[2:4,,1]
## Estimate phi
phi.estim = solve(Gammamatrix,Gamma1)
```

AR(2): X_t=$1.2\ X_{t-1} - 0.5\ X_{t-2} + Z_t$

Yule-Walker & AR Estimation (cont'd)

```
> Gammamatrix
         [,1]     [,2]     [,3]
[1,] 4.021455 3.233452 1.872159
[2,] 3.233452 4.021455 3.233452
[3,] 1.872159 3.233452 4.021455
> Gamma1
[1] 3.2334523 1.8721586 0.5991792
> phi.estim
[1]  1.20513809 -0.48696104 -0.02050628
```

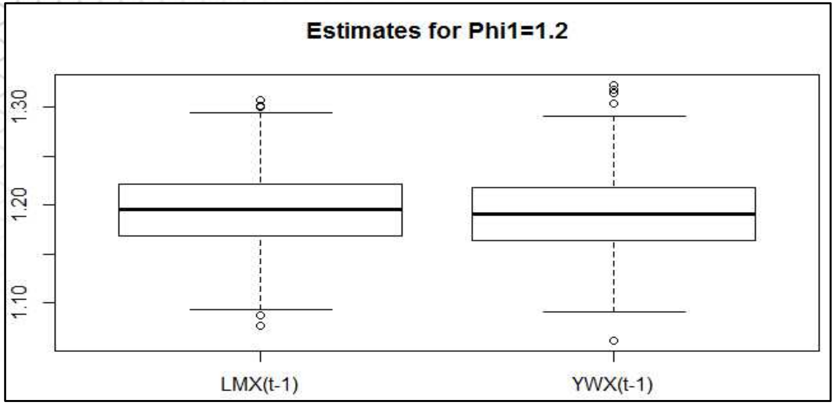
AR(2): $X_t = 1.2\, X_{t-1} - 0.5\, X_{t-2} + Z_t$

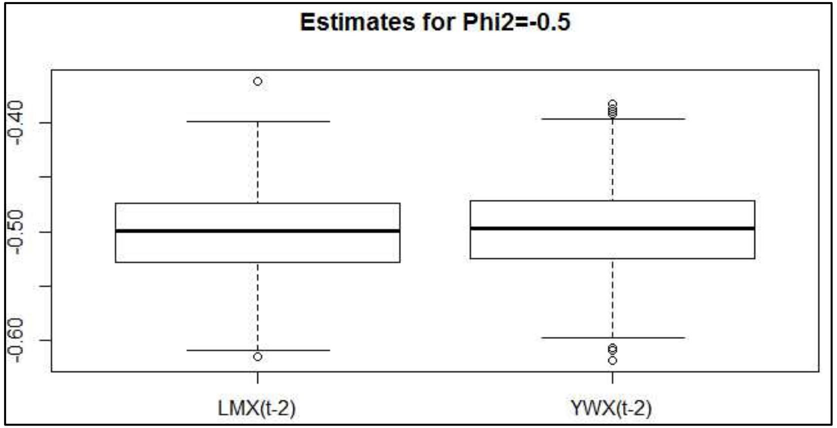
Compare Yule-Walker & AR Estimation

```
phi.estim.yw = NULL
for(s in 1:500){
  w2 = rnorm(1500)
  b = c(1.2,-0.5)
  ar2 = filter(w2,filter=b,method='recursive')
  ar2 = ar2[1001:1500]
  ## Fit Yule-Walker to AR(2)
  covf = acf(ar2,type='covariance',plot=FALSE)
  Gammamatrix = matrix(0,2,2)
  Gammamatrix[2,] = c(covf$acf[2,,1],covf$acf[1,,1])
  Gammamatrix[1,] = covf$acf[1:2,,1]
  Gamma1 = covf$acf[2:3,,1]
  phi.estim = solve(Gammamatrix,Gamma1)
  phi.estim.yw = rbind(phi.estim.yw,phi.estim)
}
```

AR(2): X_t=$1.2\ X_{t-1} - 0.5\ X_{t-2} + Z_t$

Compare Yule-Walker & AR Estimation





AR(2): $X_t = 1.2\, X_{t-1} - 0.5\, X_{t-2} + Z_t$

Summary