Ch. 4: ARMA model

Contents

ARMA(p,q) model

[ToC]

1.1 ARMA(p,q)

[ToC]

ARMA(1,1) model is defined as

$$X_t - \phi_1 X_{t-1} = e_t + \theta_1 e_{t-1}$$

where $e_t \sim WN(0, \sigma^2)$. Using the backward operator, we write this as

$$\begin{aligned} (1 - \phi_1 B) X_t &= (1 + \theta_1 B) e_t \\ \Phi(B) X_t &= \Theta(B) e_t \end{aligned}$$

ARMA(p,q)

$$(1 - \phi B - \phi_2 B^2 - \cdots - \phi_p B^p) X_t = (1 + \theta B + \theta_2 B^2 + \cdots + \theta_p B^p) e_t$$

$$\Phi(B) X_t = \Theta(B) e_t$$

Causal Representation of ARMA(p,q)

If the AR characteristic polinomial $\Phi(z)$ has all the root outside of the unit circle, we can represent ARMA(p, q) process as causal,

$$X_t = \sum_{i=0}^{\infty} \psi_i \, e_{t-i}$$

with absolutely summable sequence ψ_i. The process is stationary.

Invertible Representation of ARMA(p,q)

If the MA characteristic polinomial $\Theta(z)$ has all the root outside of the unit circle, we can represent ARMA(p, q) process as invertible,

$$e_t = \sum_{i=0}^{\infty} \pi_i \, X_{t-i}$$

with absolutely summable sequence π_i.

We assume that all ARMA(p, q) are causal, and invetible. Again, nothing is lost in this assumption.

ACVF of ARMA(p,q)

We have ARMA equation

$$\Phi(B)X_t = \Theta(B)e_t.$$

Assuming that it is causal, we can write it as

$$X_t = \Psi(B)e_t.$$

Then we know the ACVF is

$$\gamma(h) = \sigma^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+h}.$$

ACVF of ARMA

We again do this by comparing terms. We have two equations,

$$\Phi(B)X_t = \Theta(B)e_t,$$

$$X_t = \Psi(B)e_t.$$

This means, that

$$\frac{\Theta(z)}{\Phi(z)} = \Psi(z),$$

or in other words,

$$\Theta(z) = \Phi(z)\Psi(z).$$

plugging the second equation to the first, we get

$$\Theta(z) = \Phi(z)\Psi(z)$$

$$\begin{aligned}\left(1+\theta_1 z+\theta_2 z^2+\cdots+\theta_q z^q\right) \\ = \left(1-\phi_1 z-\phi_2 z-\cdots-\phi_p z\right)\left(\psi_0+\psi_1 z+\psi_2 z^2+\cdots\right)\end{aligned}$$

That means, comparing coeficcients of each term of z,

$$\begin{aligned}
1 &= \psi_0 \\
\theta_1 &= \psi_1-\psi_0\phi_1 && \text{for } z \\
\theta_2 &= \psi_2-\psi_1\phi_1-\psi_0\phi_2 && \text{for } z^2 \\
\theta_3 &= \psi_3-\psi_2\phi_1-\psi_1\phi_2-\psi_0\phi_3 && \text{for } z^3 \\
&\vdots && \vdots
\end{aligned}$$

We turn this around and write

$$\begin{aligned}
\psi_0 &= 1 \\
\psi_1 &= \theta_1 + \psi_0\phi_1 \\
\psi_2 &= \theta_2 + \psi_1\phi_1 + \psi_0\phi_2 \\
\psi_3 &= \theta_3 + \psi_2\phi_1 + \psi_1\phi_2 + \psi_0\phi_3 \\
&\vdots
\end{aligned}$$

This leads to the recursive formula,

$$\begin{aligned}
\psi_0 &= 1 \\
\psi_i &= \theta_i + \sum_{k=1}^{p} \phi_k \psi_{i-k} \qquad i > 0
\end{aligned}$$

now we can numerically calculate the ACVF of ARMA(p, q). In some cases, we can write down the closed formua of ACVF.

Example: ARMA(1,1)

Causal representation is

$$Y_t = e_t + (\phi + \theta) \sum_{j=1}^{\infty} \phi^{j-1} e_{t-j},$$

Now we know the ψ_j and can write ACVF

$$\begin{aligned} \gamma(0) &= \sigma^2 \Big[1 + \frac{(\theta + \phi)^2}{1 - \phi^2} \Big] \\ \gamma(1) &= \sigma^2 \Big[\theta + \phi + \frac{(\theta + \phi)^2 \phi}{1 - \phi^2} \Big] \end{aligned}$$

Theoretical ACF and ACVF of ARMA

```
#--- Theoretical ACF of ARMA ---
# sign of MA parameter is like [Brockwell]:  Y_t = (1+theta B) e_t

  phis   <- c(.4, .2)
  thetas <- c(.6, .2)

  rho <- ARMAacf(ar=phis, ma=thetas, lag.max=20)
  a   <- ARMAacf(ar=phis, ma=thetas, lag.max=20, pacf=T)
  rho
  a

  layout(matrix(1:2, 1,2))
  plot(rho); lines(rho, type='h'); abline(h=0)
  plot(a);   lines(a, type='h');   abline(h=0)

#--- Basic Simulation with ARMA(p,q) ---
  mu <- 5
  x  <- arima.sim(n = 250, list(ar =phis,   ma = thetas )) + mu

  layout(matrix(c(1,1,2,3), 2,2, byrow=T))
  plot(x, type="o"); lines(rho, type='p'); abline(h=0)
  acf(x);  lines(0:20, rho, type='p', col="red")
  pacf(x); lines(a, type='p', col="red")
```

ACF and ACVF of ARMA

- Both ACF and ACVF of ARMA(p,q) dacays

1.2 Parameter Estimation

[ToC]

---

1. Yule-Walker for AR parameters
2. Conditional Sum of Squares Estimators
3. Maximum Likelihood Estimators

1.3 Order Selection

[ToC]

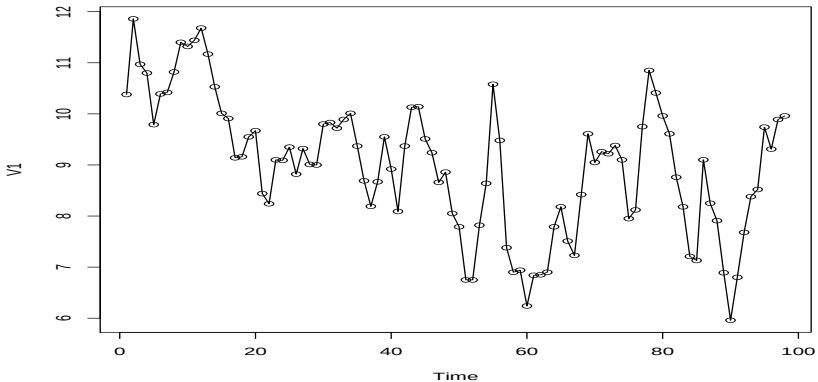
Choose your p, q based on these indicators (smaller is better):

1. AICc
2. AIC
3. BIC
4. $-$ Log Likelihood
5. number of parameters
6. $\hat{\sigma}^2$

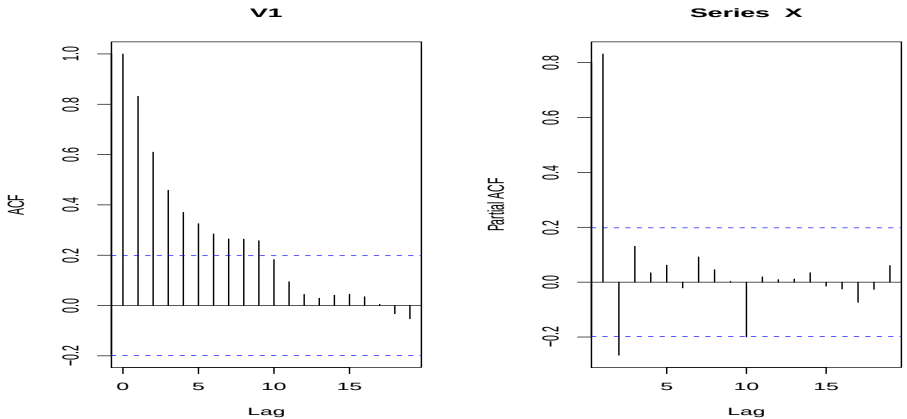
1.4 Example: Lake Huron Data

[ToC]

Level of Lake Huron 1875-1972



Sample ACF and PACF



```
#--- Analysis 1: direct fit

  library(forecast)
  source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")

  X1 <- read.csv("http://gozips.uakron.edu/~nmimoto/pages/datasets/lake.txt")
  X  <- ts(X1, start=1875, freq=1)

  plot(X, type="o")

  layout(matrix(c(1,2), 1, 2)
  acf(X);  pacf(X); layout(1)

  Fit1 <- auto.arima(X, d=0);   Fit1    # find best ARMA(p,q) by AICc

  plot(Fit1$residuals)

  layout(matrix(c(1,2), 1, 2))
  acf(Fit1$residuals);  pacf(Fit1$residuals); layout(1)

  Randomness.tests(Fit1$residuals)
```

Analysis 1 (direct fit)

- `auto.arima()` chooses AR(2) with min AICC.
- AR(2) with constant mean was fit directly to data.

$$\begin{aligned} Y_t &= \mu + X_t \\ X_t &= \phi_1 X_{t-1} + \phi_2 X_{t-2} + e_t \end{aligned}$$

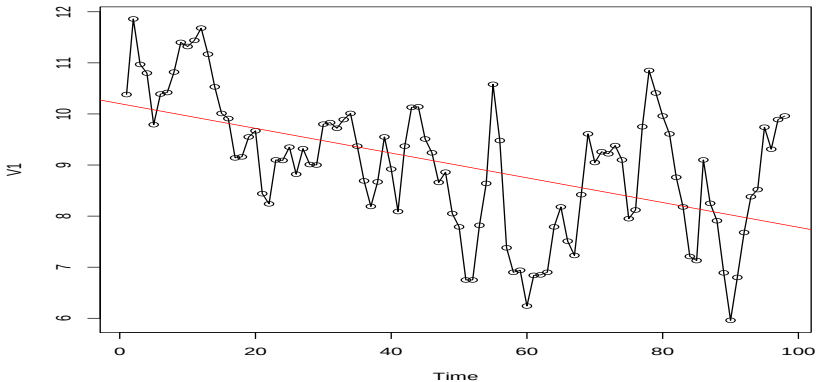
```
#--- Predicting 10 step ahead

  Y <- X

  Fit1 <- Arima(Y, order=c(2,0,0)   )   #- Force to fit MA(1)
  Y.h  <- predict(Fit1, n.ahead=10)
  Yhat <- Y.h$pred
  Yhat.CIu <- Yhat+1.96*Y.h$se
  Yhat.CIl <- Yhat-1.96*Y.h$se

  ts.plot(cbind(Y, Yhat, Yhat.CIu, Yhat.CIl ), type="o", col=c("black","red","blue","blue"))
  abline(h=0)
  abline(h=mean(Y), col="blue")


#--- (Alternatively) Predicting 10 step ahead
  plot(forecast(Fit1, h=10))
```

1.5 Rolling 1-Step Prediction

```
#--- Rolling 1-step predicton

  Rolling.len = 10   #- Size of out-sample (validation)
  Window.size = 87   #- Size of in-sample  (training)

  p = 2    # ARMA parameter
  d = 0
  q = 0

  Y <- X

  Yhat <- Yhat.CIu <- Yhat.CIl <- Y2<- 0     #- Initialize
  for (i in 1:Rolling.len) {
      window.bgn <- i
      window.end <- i+Window.size-1
      Fit1 <- Arima(Y[window.bgn:window.end], order=c(p,d,q) )    #- Force to fit AR(p)
      Y.h  <- predict(Fit1, n.ahead=1)
      Yhat[i]     <- Y.h$pred
      Yhat.CIu[i] <- Yhat[i]+1.96*Y.h$se
      Yhat.CIl[i] <- Yhat[i]-1.96*Y.h$se
  }
```

```
X  <- window(Y, start=time(Y)[1], end=time(Y)[Window.size])
Y2 <- window(Y, start=time(Y)[Window.size+1], end=time(Y)[Window.size+Rolling.len])

Yhat       <- ts(Yhat,     start=c(floor(time(Y)[Window.size+1]), cycle(Y)[Window.size+1]), freq=frequency(Y)    )
Yhat.CIu   <- ts(Yhat.CIu, start=c(floor(time(Y)[Window.size+1]), cycle(Y)[Window.size+1]), freq=frequency(Y)    )
Yhat.CIl   <- ts(Yhat.CIl, start=c(floor(time(Y)[Window.size+1]), cycle(Y)[Window.size+1]), freq=frequency(Y)    )

Pred.error <- Y2-Yhat

Pred.rMSE = sqrt( mean( (Pred.error)^2 ) ) #- prediction root Mean Squared Error
Pred.rMSE

mean(Pred.error)

layout(matrix(c(1,1,1,2,2,3), 2, 3, byrow=TRUE))
plot(Y, type="o", col="blue", main=paste("Rolling 1-step prediction with window size", Window.size) ) #- Entire dataset
  lines(X, type="o")
  lines(Yhat, type="o", col="red")
  lines(Yhat.CIu, type="l", col="red", lty=2)
  lines(Yhat.CIl, type="l", col="red", lty=2)
plot(Pred.error, type="o", main="Prediction Error (Blue-Red)")
  abline(h=c(-1.96, 1.96), col="blue", lty=2)
  abline(h=0)
acf(Pred.error, main="ACF of prediction error")
layout(1)
```

Analysis 2: Linear Trend?



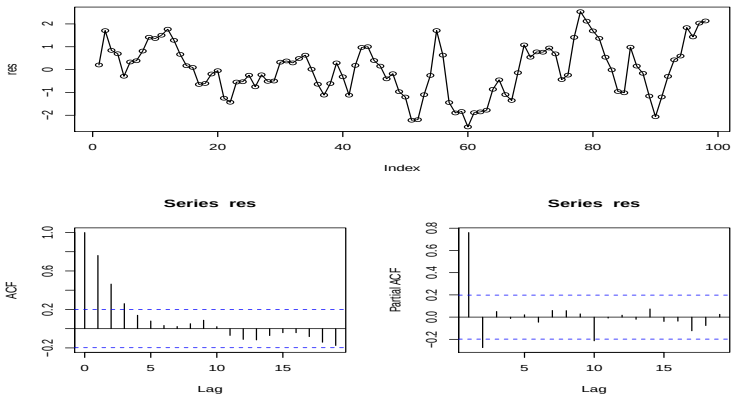
Result of Analysis 2: linear de-trend

ARMA(1,1) was fit to the residuals of linear regression.

$$\begin{aligned} X_t &= \beta_0 + \beta_1 t + Y_t \\ Y_t &= \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t \end{aligned}$$

```
#-----------------------------
#Analysis 2: linear de-trend

  plot(X, type="o")

  Reg <- lm(X~time(X))
  summary(Reg)

  plot(X, type="o"); abline(Reg, col='red')

  res <- Reg$residuals                    #- This is residual after the regression
  Randomness.test(Reg$residuals)

  Fit2 <- auto.arima(res, d=0);   Fit2    #- find best ARMA(p,q) by AICc
  Randomness.tests(Fit2$residuals)
```

Residual from Regression



Parameter Estimation

[ToC]

2.1 MLE

[ToC]

Suppose $e_t \sim N(0, \sigma^2)$. Then causal ARMA(p, q) model X_t has to be normal as well,

$$\begin{aligned} X_t &= \sum_{i=0}^{\infty} \psi_i e_{t-i} \\ X_t &\sim N\Big(0, \gamma(0)\Big) \end{aligned}$$

We know that X_{t+1} has same distribution, and covariance between X_t and X_{t+1} is $\gamma(1)$.

Multivariate Normal

That means, $\{X_1, \ldots, X_n\}$ is n-dimentional multivariate normal vector.

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \sim \mathcal{N}_n\Big(0, \mathbf{\Sigma}\Big), \qquad \text{where } \mathbf{\Sigma} = \begin{bmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(n-1) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(n-2) \\ \vdots & \vdots & & \vdots \\ \gamma(n-1) & \gamma(n-2) & \cdots & \gamma(0) \end{bmatrix}$$

Mean of each element is 0, and covariance matrix is

$$\mathbf{\Sigma} = \mathbf{\Gamma}_n$$

We know that joint pdf of multivariate Normal vector with covariance matrix $\mathbf{\Gamma}_n$ is

$$L(\phi, \theta, \sigma) = \frac{1}{(2\pi)^{n/2}(\det \mathbf{\Gamma}_n)^{1/2}} \exp\Big(-\frac{1}{2} \boldsymbol{X}'_n \mathbf{\Gamma}_n^{-1} \boldsymbol{X}_n \Big)$$

When MLE is used, the parameters must be searched numerically. Therefore, we need reasonable initial value from preliminary estimators.

It is popular to assume that the errors are Normal, and use the Gaussian Likelihood in MLE. For ARMA model, you will not be penalized so much when the assumption of Normality is violated.

2.2 Large-Sample Property of MLE

[ToC]

- For large sample from an ARMA(p, q) process, MLE has sample distribution,

$$\hat{\beta} \approx \mathcal{N}\Big(\beta, \frac{V(\beta)}{n}\Big)$$

where $\beta = (\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q, \sigma)^T$

Example: AR(p)

$$V(\beta) = \sigma^2 \mathbf{\Gamma}_p^{-1}$$

This is same as asymptotic covariance matrix of Yule-Walker estimators.

$$\begin{aligned} \text{AR(1):} \quad V(\phi_1) &= (1 - \phi_1^2) \\ \text{AR(2):} \quad V(\phi_1, \phi_2) &= \begin{bmatrix} 1 - \phi_2^2 & -\phi_1(1 + \phi_2) \\ -\phi_1(1 + \phi_2) & 1 - \phi_2^2 \end{bmatrix} \end{aligned}$$

Example: MA(q)

$$\text{MA(1):} \quad V(\theta_1) = (1 - \theta_1^2)$$

$$\text{MA(2):} \quad V(\theta_1, \theta_2) = \begin{bmatrix} 1 - \theta_2^2 & -\theta_1(1 + \theta_2) \\ -\theta_1(1 + \theta_2) & 1 - \theta_2^2 \end{bmatrix}$$

Example: ARMA(1,1)

$$V(\phi_1, \theta_1) = \frac{1-\phi_1\theta_1}{(\phi_1-\theta_1)^2} \begin{bmatrix} (1-\phi_1^2)(1-\phi_1\theta_1) & -(1-\theta_1^2)(1-\phi_1^2) \\ -(1-\theta_1^2)(1-\phi_1^2) & (1-\phi_1^2)(1-\phi_1\theta_1) \end{bmatrix}$$

Simulation Study:

Questions:

1. For what n is the asymptotic variance fromula close enough to use?
2. When the normality is violated, how bad is the normal MLE estimator?

Simulation Study

See R code file

2.3 MLE and CSS

[ToC]

We know that joint pdf of multivariate Normal vector with covariance matrix $\mathbf{\Gamma}_n$ is

$$L(\phi, \theta, \sigma) = \frac{1}{(2\pi)^{n/2}(\det \mathbf{\Gamma}_n)^{1/2}} \exp\Big(-\frac{1}{2} \boldsymbol{X}'_n \mathbf{\Gamma}_n^{-1} \boldsymbol{X}_n \Big)$$

Recall from the innovations algorithm,

$$\begin{bmatrix} X_1 - \hat{X}_1(1) \\ X_2 - \hat{X}_2(1) \\ X_3 - \hat{X}_3(1) \\ X_4 - \hat{X}_4(1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_{11} & 1 & 0 & 0 \\ a_{21} & a_{22} & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

$$\boldsymbol{X}_n - \hat{\boldsymbol{X}}_{\boldsymbol{n}} = \boldsymbol{A}_n \boldsymbol{X}_n$$

or

$$\boldsymbol{X}_n = \boldsymbol{A}_n^{-1}\left(\boldsymbol{X}_n - \hat{\boldsymbol{X}}_{\boldsymbol{n}}\right) \quad \text{where} \quad \boldsymbol{A}_n^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \theta_{11} & 1 & 0 & 0 \\ \theta_{21} & \theta_{22} & 1 & 0 \\ \theta_{31} & \theta_{32} & \theta_{33} & 1 \end{bmatrix}.$$

We know that for random vector $\boldsymbol{Y}$ and $\boldsymbol{X}$ having relationship $\boldsymbol{Y} = \boldsymbol{B}\boldsymbol{X}$, we can write the covariance matrix as

$$\boldsymbol{\Sigma}_{YY} \;=\; \boldsymbol{B}\boldsymbol{\Sigma}_{XX}\boldsymbol{B}'$$

That means, if we can write $\boldsymbol{X}_n = \boldsymbol{A}_n^{-1}(\boldsymbol{X}_n - \hat{\boldsymbol{X}}_{\boldsymbol{n}})$ as in last page, then

$$\boldsymbol{\Gamma}_n \;=\; \boldsymbol{A}_n^{-1}\, E\Big[(\boldsymbol{X}_n - \hat{\boldsymbol{X}}_{\boldsymbol{n}})(\boldsymbol{X}_n - \hat{\boldsymbol{X}}_{\boldsymbol{n}})'\Big]\,(\boldsymbol{A}_n^{-1})'.$$

It so happens that covariance matrix of the innovations is

$$E\Big[(\boldsymbol{X}_n - \hat{\boldsymbol{X}}_{\boldsymbol{n}})(\boldsymbol{X}_n - \hat{\boldsymbol{X}}_{\boldsymbol{n}})'\Big] = \mathrm{diag}(\nu_0, \ldots, \nu_{n-1}) = \boldsymbol{D}_n$$

where ν_i are from the innovations algorithm.

So we have

$$
\begin{aligned}
\boldsymbol{\Gamma}_n &= A_n^{-1}\boldsymbol{D}_n(A_n^{-1})', \\
\boldsymbol{\Gamma}_n^{-1} &= \left(A_n^{-1}\boldsymbol{D}_n(A_n^{-1})'\right)^{-1} \\
\boldsymbol{X}'_n\boldsymbol{\Gamma}_n^{-1}\boldsymbol{X}_n &= \boldsymbol{X}'_n\left(A_n^{-1}\boldsymbol{D}_n(A_n^{-1})'\right)^{-1}\boldsymbol{X}_n \\
\boldsymbol{X}'_n\boldsymbol{\Gamma}_n^{-1}\boldsymbol{X}_n &= \boldsymbol{X}'_n\left(A'_n\boldsymbol{D}_n^{-1}A_n\right)\boldsymbol{X}_n \\
\boldsymbol{X}'_n\boldsymbol{\Gamma}_n^{-1}\boldsymbol{X}_n &= (\boldsymbol{X}_n - \hat{\boldsymbol{X}}_{\boldsymbol{n}})'\boldsymbol{D}_n^{-1}(\boldsymbol{X}_n - \hat{\boldsymbol{X}}_{\boldsymbol{n}}) \\
&= \sum_{i=1}^{n}\left(X_j - \hat{X}_j\right)^2/\nu_{j-1}.
\end{aligned}
$$

$$
\det\Gamma_n = (\det A_n^{-1})(\det D_n)(\det A_n^{-1}) = \nu_0\nu_1\cdots\nu_{n-1}.
$$

Rewriting the LH

$$
\begin{aligned}
L(\phi,\theta,\sigma) &= \frac{1}{(2\pi)^{n/2}(\det\boldsymbol{\Gamma}_n)^{1/2}}\exp\Big(-\frac{1}{2}\boldsymbol{X}'_n\boldsymbol{\Gamma}_n^{-1}\boldsymbol{X}_n\Big)\\
&= \frac{1}{\sqrt{(2\pi)^n\nu_0\cdots\nu_{n-1}}}\exp\Big\{-\frac{1}{2}\sum_{i=1}^{n}(X_j-\hat{X}_j)^2/\nu_{j-1}\Big\}\\
&= \frac{1}{\sqrt{(2\pi\sigma^2)^n\, r_0\cdots r_{n-1}}}\exp\Big\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(X_j-\hat{X}_j)^2/r_{j-1}\Big\}
\end{aligned}
$$

by letting $r_i=\sigma^2\nu_i$.

MLE

$$L(\phi, \theta, \sigma) \;=\; \frac{1}{\sqrt{(2\pi\sigma^2)^n\, r_0 \cdots r_{n-1}}} \exp\Big\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (X_j - \hat{X}_j)^2 / r_{j-1} \Big\}$$

Letting $m = max(p, q)$,

$$\hat{X}_j = \begin{cases} \sum_{i=1}^{n} \theta_{nj} \Big(X_{n+1-j} - \hat{X}_{n+1-j} \Big) & \text{if } 1 \le n < m \\ \phi_1 X_n + \cdots + \phi_p X_{n+1-p} + \sum_{j=1}^{q} \theta_{nj} \Big(X_{n+1-j} - \hat{X}_{n+1-j} \Big) & \text{if } n \ge m \end{cases}$$

using θ_{nj} and r_n determined by the innovations algorithm.

With

$$S(\phi, \theta) = \sum_{i=1}^{n} (X_j - \hat{X}_j)^2 / r_{j-1},$$

MLE will minimize the log-likelihood function

$$\ell(\phi, \theta) = \ln(S(\phi, \theta)/n) + \frac{1}{n} \sum_{i=1}^{n} \ln r_{j-1}$$

and let $\hat{\sigma}^2 = S(\phi, \theta)/n$.

CSS

Conditional Sum of Squares Estimator minimizes $S(\phi, \theta)$ only.

Good starting point for MLE

Example: MLE and initial values

```
X1 <- read.csv("http://gozips.uakron.edu/~nmimoto/477/TS-hw4_data01.csv")
X  <- ts(X1[,2], start=796)    #- extract only second column as time series
plot(X,type="o")


?arima

library(forecast)

#- Default is CSS-ML
Arima(X, order=c(2,0,2))

#- See what CSS gave
Arima(X, order=c(2,0,2), method="CSS")

#- perform MLE with different initial value
Arima(X, order=c(2,0,2), method="ML", init=c(.6, -.1, -.3, .2, 100))
```

2.4 Summary 2

[ToC]

---

1. We assume the normality of the error, e_t, and compute MLE of parameters $\hat{\phi}_i$ and $\hat{\theta}_i$ and $\hat{\sigma}$.
2. Since it is MLE, large-sample sample distributions is normal, and large-sample variance of the estimates are known.
3. That means when n is small, Standard Error from R output may not be accurate.
4. Even though normality was assumed in calculation of MLE algorithm, the estimation is still consistent when e_t is not normal.
5. MLE is computed numerically, and numerical optimization needs good starting point. If MLE gives you an computation error, try different starting point.
6. CSS is used as starting point in `Arima()` and `auto.arima()` by default.

7. Calculating AICc, `auto.arima()` uses CSS value to make the computation faster. Use `approximation=FALSE` option to turn it off.

Forecasting ARMA model

[ToC]

3.1 Innovations Algorithm

[ToC]

Brockwell p71

- Recall that to find the best $h-$step linear predictor,

$$\hat{Y}_n(h) = a_0 + a_1 Y_n + \cdots + a_n Y_1,$$

- we need to minimize prediction MSE,

$$\text{MSE} = E\Big[\big(\hat{Y}_n(h) - Y_{n+h}\big)^2\Big].$$

- To find such $a_0, a_1, \ldots, a_n$, we ended up with matrix equation,

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} \gamma(0) & \gamma(1) & \gamma(2) & \gamma(3) \\ \gamma(1) & \gamma(0) & \gamma(1) & \gamma(2) \\ \gamma(2) & \gamma(1) & \gamma(0) & \gamma(1) \\ \gamma(3) & \gamma(2) & \gamma(1) & \gamma(0) \end{bmatrix}^{-1} \begin{bmatrix} \gamma(h) \\ \gamma(h+1) \\ \gamma(h+2) \\ \gamma(h+3) \end{bmatrix}.$$

- What happens when n is large? Can we get inverse of 1000×1000 matrix?

- Recursive formula is needed.

Innovations Algorithm

innovations: one-step prediction errors using $n-1$ observations: $Y_n - \hat{Y}_n(1)$

$$
\begin{array}{lllll}
\text{Use 0 obs,} & \text{innov: } Y_1 - \hat{Y}_1(1), & \hat{Y}_1(1) & = & 0 \\
\text{Use 1 obs,} & \text{innov: } Y_2 - \hat{Y}_2(1), & \hat{Y}_1(1) & = & a_{11}Y_1 \\
\text{Use 2 obs,} & \text{innov: } Y_3 - \hat{Y}_3(1), & \hat{Y}_1(1) & = & a_{21}Y_2 + a_{22}Y_1 \\
\text{Use 3 obs,} & \text{innov: } Y_4 - \hat{Y}_4(1), & \hat{Y}_1(1) & = & a_{31}Y_3 + a_{32}Y_3 + a_{33}Y_1 \\
 & & & \vdots &
\end{array}
$$

$$
\begin{aligned}
Y_1 - \hat{Y}_1(1) &= Y_1 - 0 \\
Y_2 - \hat{Y}_2(1) &= Y_2 - a_{11}Y_1 \\
Y_3 - \hat{Y}_3(1) &= Y_3 - a_{21}Y_2 - a_{22}Y_1 \\
Y_4 - \hat{Y}_4(1) &= Y_4 - a_{31}Y_3 - a_{32}Y_3 - a_{33}Y_1 \\
&\vdots
\end{aligned}
$$

$$
\begin{aligned}
Y_1 - \hat{Y}_1(1) &= Y_1 - 0 \\
Y_2 - \hat{Y}_2(1) &= Y_2 - a_{11}Y_1 \\
Y_3 - \hat{Y}_3(1) &= Y_3 - a_{21}Y_2 - a_{22}Y_1 \\
Y_4 - \hat{Y}_4(1) &= Y_4 - a_{31}Y_3 - a_{32}Y_3 - a_{33}Y_1
\end{aligned}
$$

$$
\begin{bmatrix} Y_1 - \hat{Y}_1(1) \\ Y_2 - \hat{Y}_2(1) \\ Y_3 - \hat{Y}_3(1) \\ Y_4 - \hat{Y}_4(1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_{11} & 1 & 0 & 0 \\ a_{21} & a_{22} & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 1 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix}
$$

$$
\boldsymbol{U}_4 = \boldsymbol{A}_4 \boldsymbol{Y}_4
$$

Note that $\boldsymbol{A}_n$ is a lower triangular matrix.

We have another way to write the left hand side,

$$\begin{bmatrix} Y_1 - \hat{Y}_1(1) \\ Y_2 - \hat{Y}_2(1) \\ Y_3 - \hat{Y}_3(1) \\ Y_4 - \hat{Y}_4(1) \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} - \begin{bmatrix} \hat{Y}_1(1) \\ \hat{Y}_2(1) \\ \hat{Y}_3(1) \\ \hat{Y}_4(1) \end{bmatrix}$$

$$\boldsymbol{U}_n = \boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_{\boldsymbol{n}}$$

So we have two equations,

$$\begin{aligned} \boldsymbol{U}_n &= \boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_{\boldsymbol{n}} \\ \boldsymbol{U}_n &= \boldsymbol{A}_4 \boldsymbol{Y}_4 \end{aligned}$$

From the two equations,

$$\begin{aligned} \boldsymbol{U}_n &= \boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_{\boldsymbol{n}} \\ \boldsymbol{U}_n &= \boldsymbol{A}_n \boldsymbol{Y}_n \end{aligned}$$

We can get

$$\begin{aligned} \hat{\boldsymbol{Y}}_{\boldsymbol{n}} &= \boldsymbol{Y}_n - \boldsymbol{U}_n \\ \boldsymbol{A}_n^{-1} \boldsymbol{U}_n &= \boldsymbol{Y}_n \end{aligned}$$

Substituting, we can write the predictors as

$$\begin{aligned} \hat{\boldsymbol{Y}}_{\boldsymbol{n}} &= \boldsymbol{A}_n^{-1} \boldsymbol{U}_n - \boldsymbol{U}_n \\ &= \Big(\boldsymbol{A}_n^{-1} - \boldsymbol{I} \Big) \boldsymbol{U}_n. \end{aligned}$$

Inverse of a lower triangular matrix is a lower triangular matrix. Let us call the elements of $\boldsymbol{A}_n^{-1}$ as θ_{ij}.

$$\boldsymbol{A}_4^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \theta_{11} & 1 & 0 & 0 \\ \theta_{21} & \theta_{22} & 1 & 0 \\ \theta_{31} & \theta_{32} & \theta_{33} & 1 \end{bmatrix}$$

We now have formula

$$\hat{\boldsymbol{Y}}_{\boldsymbol{n}} = \Big(\boldsymbol{A}_n^{-1} - \boldsymbol{I}\Big)\boldsymbol{U}_n.$$

$$\hat{\boldsymbol{Y}}_{\boldsymbol{n}} = \Big(\boldsymbol{A}_n^{-1} - \boldsymbol{I}\Big)\boldsymbol{U}_n$$

$$\begin{bmatrix} \hat{Y}_1(1) \\ \hat{Y}_2(1) \\ \hat{Y}_3(1) \\ \hat{Y}_4(1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \theta_{11} & 0 & 0 & 0 \\ \theta_{21} & \theta_{22} & 0 & 0 \\ \theta_{31} & \theta_{32} & \theta_{33} & 0 \end{bmatrix} \begin{bmatrix} Y_1 - \hat{Y}_1(1) \\ Y_2 - \hat{Y}_2(1) \\ Y_3 - \hat{Y}_3(1) \\ Y_4 - \hat{Y}_4(1) \end{bmatrix}$$

This will let us compute $\hat{X}_n$ recursively, since row n of $\boldsymbol{\Theta}_n$ only uses up to $\hat{X}_{n-1}$.

Innovations Algorithm

$$\hat{Y}_n(1) = \begin{cases} 0 & \text{if } n = 0 \\ \sum_{j=1}^{n} \theta_{nj}\left(Y_{n+1-j} - \hat{Y}_{n+1-j}\right) & \text{if } n > 0 \end{cases}$$

1. Recursive Formula
2. Can be used for non-stationary series
3. If Y_t is invertible, $\theta_{nj} \to \theta_j$

The coefficients θ_{ij} can be calculated as letting

$$\nu_0 = \gamma(0)$$

and for $k = 0, 1, \ldots, n-1$,

$$\begin{aligned} \theta_{n,n-k} &= \Big(\gamma(n-k) - \sum_{j=0}^{k-1} \theta_{k,k-j}\, \theta_{n,n-j}\, \nu_j\Big)/\nu_k \\ \nu_n &= \gamma(0) - \sum_{j=0}^{n-1} \theta_{n,n-j}^2 \nu_j. \end{aligned}$$

You compute successively, ν_0; θ_{11}, ν_1; $\theta_{22}, \theta_{21}, \nu_2$; $\theta_{33}, \theta_{32}, \theta_{31}, \nu_3$; and so on. It looks like:

$$
\begin{aligned}
\nu_0 &= \hat{\gamma}(0) \\
\theta_{11} &= \hat{\gamma}(1)/\nu_0 \\
\\
\nu_1 &= \hat{\gamma}(0) - \theta_{11}^2 \nu_0 \\
\theta_{22} &= \hat{\gamma}(2)/\nu_0 \\
\theta_{21} &= [\hat{\gamma}(1) - \theta_{11}\theta_{22}\nu_0]/\nu_1 \\
\\
\nu_2 &= \gamma(0) - \theta_{22}^2 \nu_0 - \theta_{21}^2 \nu_1 \\
\theta_{33} &= \hat{\gamma}(3)/\nu_0 \\
\theta_{32} &= \hat{\gamma}(2) - \theta_{11}\theta_{33}\nu_0]/\nu_1 \\
\theta_{31} &= [\gamma(1) - \theta_{22}\theta_{33}\nu_0 - \theta_{21}\theta_{32}\nu_1]/\nu_2 \\
&\vdots
\end{aligned}
$$

```
#- Lake Data Analysis continued


  plot(X, type="o"); abline(Reg, col='red')



  #- Prediction from Analysis 1

  Y.h  <- predict(Fit1, n.ahead=10)

  ts.plot(cbind(X,Y.h$pred, Y.h$pred+1.96*Y.h$se, Y.h$pred-1.96*Y.h$se),
                  type="o", col=c("black","red","blue","blue"))
  abline(h=mean(Y), col="blue")
```

```
#- Prediction from Analysis 2

Y.h2  <- predict(Fit2, n.ahead=10)
ts.plot(cbind(res,Y.h2$pred, Y.h2$pred+Y.h2$se, Y.h2$pred-Y.h2$se),
                type="o", col=c("black","red","blue","blue"), main="residual")
abline(h=0, col="blue")

RegLine <- ts(Reg$coef[1] + Reg$coef[2]* c(index(res), index(Y.h2$pred)), start=start(X) )

ts.plot(cbind(X, RegLine+Y.h2$pred, RegLine+Y.h2$pred+1.96*Y.h2$se, RegLine+Y.h2$pred-1.96*Y.h2$se),
                type="o", col=c("black","red","blue","blue"), main="residual")
lines(RegLine, col="blue")
abline(h=0, col="blue")
```

4

Additional Topics

[ToC]

4.1 Durbin-Levinson Algorithm

[ToC]

If $E(Y_t) = \mu$, then let $X_t = Y_t - \mu$, and $P_n Y_{n+h} = \mu + P_n X_{n+h}$ will be the best linear predictor. So in principle, $\Gamma_n \boldsymbol{a}_n = \boldsymbol{\gamma}_n(h)$ gives the best linear predictor in general, but solving n linear equation may take time.

Recursive prediction algorighms use $P_n X_{n+1}$ to obtain $P_{n+1} X_{n+2}$, and so on. We start by, with $\boldsymbol{\gamma}_n(1) = [\gamma(1), \ldots, \gamma(n)]^T$

$$\begin{aligned} \Gamma_n \boldsymbol{a}_n &= \boldsymbol{\gamma}_n(1) \\ \boldsymbol{a}_n &= \Gamma_n^{-1} \boldsymbol{\gamma}_n(1) \\ P_n X_{n+1} &= \boldsymbol{a}_n^T \boldsymbol{X}_n = (\Gamma_n^{-1} \boldsymbol{\gamma}_n(1))^T \boldsymbol{X}_n \\ E(X_{n+1} - P_n X_{n+1})^2 &= \gamma(0) - \boldsymbol{a}_n^T \gamma_n(1) \end{aligned}$$

This leads to the Durbin-Levinson Algorighm. Let $\nu_0 = \gamma(0)$ and $\phi_{11} = \gamma(1)/\gamma(0) = \rho(1)$, and for

$$\begin{aligned}
\nu_n &= \nu_{n-1}[1 - \phi_{nn}^2] \\
\phi_{nn} &= \Big[\gamma(n) - \sum_{j=1}^{n-1} \phi_{n-1,j}\gamma(n-j)\Big]/\nu_{n-1} \\
&= \frac{-1}{\nu_{n-1}}\Big[\phi_{n-1,n-1}, \cdots, \phi_{n-1,1}, -1\Big] \begin{bmatrix} \gamma(1) \\ \vdots \\ \gamma(n) \end{bmatrix}
\end{aligned}$$

$$\begin{bmatrix} \phi_{n1} \\ \phi_{n2} \\ \vdots \\ \phi_{n,n-1} \end{bmatrix} = \begin{bmatrix} \phi_{n-1,1} \\ \phi_{n-1,2} \\ \vdots \\ \phi_{n-1,n-1} \end{bmatrix} - \phi_{nn} \begin{bmatrix} \phi_{n-1,n-1} \\ \phi_{n-1,n-2} \\ \vdots \\ \phi_{n-1,1} \end{bmatrix}$$

So we have one step predictor as

$$P_n X_{n+1} = \Big[\phi_{n1}, \cdots, \phi_{nn}\Big] \begin{bmatrix} X_n \\ \vdots \\ X_1 \end{bmatrix}$$

with MSE

$$\nu_n = E(X_{n+1} - P_n X_{n+1})^2.$$

$$\nu_0 = \gamma(0)$$

$$\phi_{11} = \gamma(1)/\gamma(0) = \rho(1)$$

$$\nu_1 = \nu_0[1 - \phi_{11}^2]$$

$$\begin{aligned}
\phi_{22} &= \Big[\gamma(2) - \phi_{11}\gamma(1)\Big]/\nu_1 \\
\phi_{21} &= \phi_{11} - \phi_{22}\phi_{11} \\
\nu_2 &= \nu_1[1 - \phi_{22}^2]
\end{aligned}$$

$$\begin{aligned}
\phi_{33} &= \Big[\gamma(3) - \phi_{21}\gamma(2) - \phi_{22}\gamma(1)\Big]/\nu_2 \\
\phi_{31} &= \phi_{21} - \phi_{33}\phi_{22} \\
\phi_{32} &= \phi_{22} - \phi_{33}\phi_{21} \\
\nu_3 &= \nu_2[1 - \phi_{33}^2]
\end{aligned}$$

$$\begin{aligned}
\phi_{44} &= \Big[\gamma(4) - \phi_{31}\gamma(3) - \phi_{32}\gamma(2) - \phi_{33}\gamma(1)\Big]/\nu_3 \\
\phi_{41} &= \phi_{31} - \phi_{44}\phi_{33} \\
\phi_{42} &= \phi_{32} - \phi_{44}\phi_{32} \\
\phi_{43} &= \phi_{33} - \phi_{44}\phi_{31} \\
\nu_4 &= \nu_3[1 - \phi_{44}^2]
\end{aligned}$$

ϕ_{nn} is called partial ACF.

D-L algorithm expands $\hat{X}_{n+1}$ in terms of linear coefficients of X_t. Innov algorithm expands $\hat{X}_{n+1}$ in terms of innovations $X_t - \hat{X}_t$. This has some advantage since innovations are uncorrelated. Also, it will be greatly simplified for ARMA(p, q) processes.

For h step prediction with innov algo ,

$$P_n X_{n+h} = \sum_{j=h}^{n+h-1} \theta_{n+h-1,j}\Big(X_{n+h-j} - \hat{X}_{n+h-j}\Big),$$

So if $n = 10$, and $h = 5$,

$$P_{10} X_{15} = \theta_{14,5}(X_{10} - \hat{X}_{10}) + \theta_{14,6}(X_9 - \hat{X}_9) \cdots + \theta_{14,14}(X_1 - \hat{X}_1)$$

4.2 Forecasting ARMA

[Back to Section 2.5] The innovation algorithm can be drastically simplified for ARMA(p,q) process. For ARMA process X_t, transform it by

$$\begin{cases} W_t = X_t/\sigma & \text{if } t = 1, \ldots, m \\ W_t = \phi(B)X_t/\sigma & \text{if } t > m \end{cases}$$

where $m = max(p, q)$. Let $\theta_0 = 1$, $\theta_j = 0$ for $j > q$, and assume $p, q > 1$. Then we can write $E(W_i W_j)$ as

$$\begin{cases} \gamma_X(i-j)/\sigma^2 & \text{if } 1 \le i, j \le m \\ \left\{\gamma_X(i-j) - \sum_{i=1}^{p} \phi_r \gamma_X(r - |i-j|)\right\}/\sigma^2 & \text{if } min(i,j) \le m < max(i,j) \le 2m \\ \sum_{r=0}^{q} \theta_r \theta_{r+|i-j|} & \text{if } \min(i,j) > m \\ 0 & \text{if } otherwise \end{cases}$$