Chapter 5 – Modeling and Forecasting with ARMA Processes.

5.1 Preliminary Estimation

The Yule–Walker and Burg procedures apply to the fitting of pure AR model (Although the former can be adapted to models with $q>0$, it will be less efficient.)

The innovation and Hannan–Rissanen algorithms are used to provide preliminary estimates of the ARMA when $q>0$.

Yule–Walker

The Yule–Walker coefficients $\hat{\phi}_{p1}, \cdots \hat{\phi}_{pp}$ are precisely the coefficients of the best linear predictor of X_{p+1} in terms of $\{X_p, \cdots X_1\}$ under the assumption the ACF of $\{X_t\}$ coincides with the sample ACF at lags $1, \cdots n$.

For a pure AR model, $\Phi(B) X_t = Z_t$ (*)

for $h>0$, $E(X_t X_{t-h}) = E[(Z_t + \sum_{r=1}^{p} \phi_r X_{t-r}) X_{t-h}]$

$= 0 + \sum_{r=1}^{p} \phi_r E(X_{t-r} \cdot X_{t-h})$

$\Rightarrow \gamma(h) = \sum_{r=1}^{p} \phi_r \gamma(|h-r|)$

for $h=0$, $\gamma(0) = \alpha^2 + \sum_{r=1}^{p} \phi_r \gamma(r)$

Then, we obtain the Yule–Walker equations

$$\Gamma_p \phi = \gamma_p \quad \text{and} \quad \alpha^2 = \gamma(0) - \phi^T \gamma_p$$

where Γ_p is the covariance matrix $[\gamma(i-j)]_{i,j=1}^{p}$ and $\gamma_p = (\gamma(1) \cdots \gamma(p))^T$.

If we replace covariances $\gamma(j)$ by the corresponding sample covariances $\hat{\gamma}(j)$, we obtain a set of equations for the so-called Yule–Walker estimators $\hat{\phi}$ and $\hat{\alpha}^2$ of ϕ and α^2.

$$\hat{\Gamma}_p \hat{\phi} = \hat{\gamma}_p \quad \text{and} \quad \hat{\alpha}^2 = \hat{\gamma}(0) - \hat{\phi}^T \hat{\gamma}_p$$

where $\hat{\Gamma}_p = [\hat{\gamma}(i-j)]_{i,j=1}^{p}$ and $\hat{\gamma}_p = (\hat{\gamma}(1) \cdots \hat{\gamma}(p))^T$.

If $\hat{\gamma}(0) > 0$, then $\hat{\Gamma}_m$ is nonsingular for every $m = 1, 2, \cdots$

Sample Yule–Walker Equations:

$$\hat{\phi} = \left(\hat{\phi}_1, \ldots, \hat{\phi}_p\right)' = \hat{R}_p^{-1}\hat{\rho}_p \tag{5.1.7}$$

and

$$\hat{\sigma}^2 = \hat{\gamma}(0)\left[1 - \hat{\rho}_p'\hat{R}_p^{-1}\hat{\rho}_p\right], \tag{5.1.8}$$

where $\hat{\rho}_p = \left(\hat{\rho}(1), \ldots, \hat{\rho}(p)\right)' = \hat{\gamma}_p/\hat{\gamma}(0)$.

It is often the case that moment estimators, i.e. estimators that (like $\hat{\phi}$) are obtained by equating theoretical and sample moments, have much higher variances than estimators obtained by alternative methods such as maximum likelihood. However, the Yule-Walker estimators of the coefficients $\phi_1, \cdots, \phi_p$ of an AR(p) process have approximately the same distribution for large samples as the corresponding maximum likelihood estimators.

Large-Sample Distribution of Yule-Walker Estimators

For large n and true AR(p) model, $\hat{\underset{\sim}{\phi}}_p \dot{\sim} N\left(\underset{\sim}{\phi}, \frac{1}{n}\sigma^2\Gamma_p^{-1}\right)$

- If $m < p$, missing ϕ's!
  fitting ↓ truth
  $\Rightarrow \hat{\underset{\sim}{\phi}}_m$ will not be consistent for the truth.

- If $m > p$, NOT missing ϕ's!
  $X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + 0 \cdot X_{t-p-1} + \cdots 0 \cdot X_{t-m}$
  "True" model with m ϕs just needs $m-p$ 0-valued ϕ's for the remaining terms.

Burg's algorithm

Burg's algorithm estimates the PACF $\{\phi_{11}, \phi_{22}, \cdots\}$ by successively minimizing sum of squares of forward and backward one-step prediction errors with respect to the coefficients ϕ_{ii}.

Let $u_i(t) = x_{n+1-t} - \hat{x}^{(i)}_{n+1-t}$ for $t = i+1, \cdots, n$ $(0 \le i < n)$ where $\hat{x}^{(i)}_{n+1-t}$ is the BLP of observation x_{n+1-t} given the preceding i observations.

e.g. $n=15$, $i=3$, $t=5$

$u_3(5) = x_{14} - \hat{x}^{(3)}_{14}$

• BLP for x_{14} using x_{13}, x_{12}, x_{11}

Let $v_i(t) = x_{n+1-t} - \tilde{x}^{(i)}_{n+1-t}$ for $t = i+1, \cdots, n$ where $\tilde{x}^{(i)}_{n+1-t}$ is BLP of x_{n+1-t} using i subsequent (immediately following) observations.

e.g. $v_3(5) = x_1 - \tilde{x}_1^{(3)}$

• BLP for x_1 using x_2, x_3, x_4

$u_i(t)$ and $v_i(t)$ satisfy

$$u_0(t) = v_0(t) = x_{n+1-t}$$
$$u_i(t) = u_{i-1}(t-1) - \phi_{ii} v_{i-1}(t)$$
$$v_i(t) = v_{i-1}(t) - \phi_{ii} u_{i-1}(t-1)$$

equal (PACF we want)

Note $u_i(t)$ and $v_i(t)$ are functions of the data!

Burg's estimate is computed by (innovation)

$$\phi^{(B)}_{11} = \min_{\phi_{11}} \frac{1}{2(n-1)} \sum_{t=2}^{n} (u_1(t)^2 + v_1(t)^2)$$

$$\phi^{(B)}_{22} = \min_{\phi_{22}} \frac{1}{2(n-2)} \sum_{t=3}^{n} (u_2(t)^2 + v_2(t)^2)$$ use $\phi^{(B)}_{11}$ in $u_2(t)$ and $v_2(t)$

⋮ ⋮

until we get $\phi^{(B)}_{pp}$.

The above recursion is equivalent to solving the recursion

$$d(1) = \sum_{t=2}^{n} \left(u_0^2(t-1) + v_0^2(t) \right)$$

$$\phi_{11}^{(B)} = \frac{2}{d(1)} \sum_{t=1+1}^{n} v_{1-1}(t)\, u_{1-1}(t-1)$$

$$d(1+1) = (1 - \phi_{11}^{(B)2})\, d(1) - v_1^2(1+1) - u_1^2(n)$$

$$\sigma_1^2(B) = \frac{(1 - \phi_{11}^{(B)2})\, d(1)}{2(n-1)}$$

PACF coefficients: $\hat{\phi}_{11}^{(B)}, \hat{\phi}_{22}^{(B)} \cdots \hat{\phi}_{pp}^{(B)}$

The coefficients for AR(p) are $\phi_{p1}, \cdots \phi_{pp}$

$\Rightarrow$ Durbin-Levison algorithm

① Compute ϕ_{mm}.

② Compute $\phi_{m+1} \cdots \phi_{m1}$ using ϕ_{mm} and $\phi_{m-1\,m} \cdots \phi_{m+1}$

Replace the step where we compute ϕ_{mm} with $\hat{\phi}_{mm}$ from Burg's algorithm.

Both algorithms are consistent for true $\phi_1 \cdots \phi_p$ and they have the same asymptotic distribution

$$\hat{\underset{\sim}{\phi}}_p^{(B)} \sim N(\underset{\sim}{\phi}_p, \frac{1}{n} \sigma^2 \Gamma_p^{-1})$$

Innovation algorithm

Substitute $\hat{\gamma}(h)$ for $\gamma(h)$ in the innovation algorithm $(\kappa(i,j))$

For $MA(q)$, $X_t = Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q}$ $\quad Z_t \sim WN(0, \sigma^2)$

If we apply innovation algorithm to obtain $\hat{X}_n$, we have that

$(\theta_{n1}, \cdots \theta_{nn})$ at n-step

$\theta_{m1}, \cdots \theta_{mq}$ for sufficiently large m will be approximately equal to $\theta_1, \cdots \theta_q$

Now we can plug in $\hat{\kappa}(i,j) = \hat{\gamma}(i,j)$ and obtain $\hat{\theta}_{m1}, \cdots \hat{\theta}_{mq}$ and then for large enough m

$$(\hat{\theta}_{m1}, \cdots \hat{\theta}_{mq}) \sim N(\underset{\sim}{\theta}, \frac{1}{n} Var(\underset{\sim}{\theta}))$$

Hannan - Rissanen algorithm

Basic idea:

① Use Yule-Walker as if $AR(p)$, then use AR coefficients to estimate $\{Z_t\}$ process. (if invertible)

$$\hat{Z}_t = X_t - \hat{\phi}_1 X_{t-1} - \hat{\phi}_2 X_{t-2} \cdots - \hat{\phi}_p X_{t-p}$$

② $X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \theta_1 \hat{Z}_{t-1} + \cdots + \theta_q \hat{Z}_{t-q} + \hat{Z}_t$

Solve for new values of $\underset{\sim}{\phi}$ and $\underset{\sim}{\theta}$ via least squares

$\Rightarrow$ use multiple linear regression to estimate $\underset{\sim}{\phi}$ and $\underset{\sim}{\theta}$ by minimizing

$$S(\underset{\sim}{\phi}, \underset{\sim}{\theta}) = \sum_{t=1}^{n} (X_t - \phi_1 X_{t-1} \cdots - \phi_p X_{t-p} - \theta_1 \hat{Z}_{t-1} - \cdots \theta_q \hat{Z}_{t-q})^2$$

(?)

Problem Sometimes $\hat{Z}_t$ is not good estimator

Moment estimator (don't need assumption for distribution)

① Yule-Walker for AR(p) (ACF)

② Burg's algorithm for AR(p) (PACF)

③ Innovation algorithm for MA(q) (ACF)

④ Hannan-Rissanen for ARMA(p,q) (ACF/PACF + MLR)

<u>Problem</u> Might not perform well if the model is not correctly specified.

5.2 Maximum Likelihood Estimation

<u>Assumption</u> Assume $\{X_t\}$ is a Gaussian time series with mean zero and $\kappa(i,j) = E(X_i X_j)$

Let $\underline{\gamma}_n = (\gamma_1, \cdots \gamma_n)$. Then the likelihood for a mean-zero Gaussian time series is the multivariate Normal density.

$$L(\Gamma_n) = (2\pi)^{-n/2}(\det \Gamma_n)^{-1/2} \exp\left(-\frac{1}{2}\mathbf{X}_n' \Gamma_n^{-1} \mathbf{X}_n\right)$$

where $\Gamma_n = E(\underline{X}_n \underline{X}_n^T) = \begin{pmatrix} \kappa(1,1) & \cdots & \kappa(1,n) \\ \vdots & & \vdots \\ \kappa(n,1) & \cdots & \kappa(n,n) \end{pmatrix}$

Recall that $\kappa(i,j)$ is a function of $\underline{\phi}$, $\underline{\theta}$ and σ^2 of ARMA(p,q).

In general, maximizing $L(\Gamma_n)$ will be impossible due to the presence of Γ_n^{-1} and $\det(\Gamma_n)$. However, the innovation algorithm allows us to simplify things.

Let $\hat{X}_j = E(X_j \mid X_{j-1}, \cdots X_1) = P_{j-1} X_j$
Let $\{\theta_{ij}\}$ be the coefficients from the innovations algorithm and $X_j - \hat{X}_j$ is the prediction error of the jth observation.
We know that $X_n = C_n (X_n - \hat{X}_n)$ where

$$C_n = \begin{pmatrix} 1 & 0 & 0 & \cdots \\ \theta_{11} & 1 & 0 & \cdots \\ \theta_{nn} & \cdots & & \end{pmatrix}$$

Now we also have that

$$E([X_i - \hat{X}_i][X_j - \hat{X}_j]) = 0 \qquad \text{(induction)}$$

i.e. innovation for $i \neq j$ are uncorrelated $\forall i, j,\ i \neq j$

Therefore, $\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n$ has a diagonal covariance matrix

$$D_n = \text{diag}(v_0, \cdots v_{n-1})$$

where v_j comes from the innovation algorithm.

because the variance of the innovations is

$$\text{Var}(X_n - \hat{X}_n) = E[(X_n - \hat{X}_n)^2] \quad \text{b.c } E(X_n - \hat{X}_n) = 0$$
$$= MSE$$

Γ_n

$$\text{Var}(\underset{\sim}{X}_n) = E(\underset{\sim}{X}_n \underset{\sim}{X}_n^T) = E([C_n(\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)][C_n(\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)]^T)$$
$$= E[C_n(\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)(\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)^T C_n^T]$$
$$= C_n E[(\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)(\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)^T] C_n^T$$
$$= C_n \text{Var}(\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n) C_n^T$$
$$= C_n D_n C_n^T$$

So in $L(\Gamma)$, $\underset{\sim}{X}_n^T \Gamma_n^{-1} \underset{\sim}{X}_n = (\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)^T C_n^T \Gamma_n C_n (\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)$

$$= (\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)^T C_n^T (C_n^T)^{-1} D_n^{-1} C_n^{-1} C_n (\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)$$
$$= (\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)^T D_n^{-1} (\underset{\sim}{X}_n - \hat{\underset{\sim}{X}}_n)$$
$$= \sum_{j=1}^{n} \frac{(X_j - \hat{X}_j)^2}{v_{j-1}} \quad \text{since } D_n \text{ diagonal.}$$

and $\det(\Gamma_n) = \det(C_n D_n C_n^T) = \det(C_n)\det(D_n)\det(C_n^T)$

lower diagonal matrix $\Rightarrow \det(d_1 \cdots d_n) = \det(1)$

$$= \det(C_n)^2 \det(D_n)$$
$$= v_0 \cdot v_1 \cdots v_{n-1}$$
$$= \prod_{j=0}^{n} v_j$$

$$\Rightarrow L(\Gamma_n) = \frac{1}{(\sqrt{2\pi})^n \sqrt{v_0 \cdots v_{n-1}}} \exp\left(-\frac{1}{2} \frac{\sum_{j=1}^{n} (X_j - \hat{X}_j)}{v_{j-1}}\right)$$

If Γ_n depends on a finite number of parameters, the MLE maximizes L for any given datasets.

Remember

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

$E[(X_{n+1} - \hat{X}_n)^2] = \sigma^2 r_n = v_n$ where $r_n = E[(W_{n+1} - \hat{W}_{n+1})^2]$ from the recursion in the innovation algorithm.

The Gaussian Likelihood for an ARMA Process:

$$L(\boldsymbol{\phi}, \boldsymbol{\theta}, \sigma^2) = \frac{1}{\sqrt{(2\pi\sigma^2)^n r_0 \cdots r_{n-1}}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{j=1}^{n} \frac{(X_j - \hat{X}_j)^2}{r_{j-1}} \right\}. \quad (5.2.9)$$

$$\Rightarrow l(\underline{\phi}, \underline{\theta}, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2}\sum_{j=0}^{n-1}\log r_j - \frac{1}{2\sigma^2}\sum_{j=1}^{n}\frac{(X_j - \hat{X}_j)^2}{r_{j-1}}$$

Note that $\hat{X}_j$ and r_{j-1} are both function of model parameters $\underline{\phi}, \underline{\theta}$ but not of σ^2.

Maximum Likelihood Estimators:

$$\hat{\sigma}^2 = n^{-1} S(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}}), \quad \Rightarrow \text{by setting } \frac{\partial l}{\partial \sigma^2} = 0 \quad (5.2.10)$$

where

$$S(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}}) = \sum_{j=1}^{n} (X_j - \hat{X}_j)^2 / r_{j-1}, \quad (5.2.11)$$

and $\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}}$ are the values of $\boldsymbol{\phi}, \boldsymbol{\theta}$ that minimize

$$\ell(\boldsymbol{\phi}, \boldsymbol{\theta}) = \ln\left(n^{-1} S(\boldsymbol{\phi}, \boldsymbol{\theta})\right) + n^{-1} \sum_{j=1}^{n} \ln r_{j-1}. \quad (5.2.12)$$

↑ No closed form
use numerical optimization algorithm.

Most optimizer methods require some initial values for $\underline{\phi}, \underline{\theta}$ (Other estimators can be used as starting point e.g. Burg's)

< Lazy maximation >

Try to minimize $S(\underline{\phi}, \underline{\theta}) = \sum_{j=1}^{n} \frac{(X_j - \hat{X}_j)^2}{r_{j-1}}$ (least square approach)

But can lead to noncausal or noninvertible solutions without additional constraints.

And the estimate of σ^2 would be

$$\sigma^2 = \frac{S(\hat{\underline{\phi}}^{ls}, \hat{\underline{\theta}}^{ls})}{n - (p+q)}$$

The MLE for large n will be approximately normal.

Let $\underset{\sim}{\beta} = (\phi_1, \dots \phi_p, \theta_1 \dots \theta_q)$

$\hat{\underset{\sim}{\beta}} = (\hat{\phi}_1, \dots \hat{\phi}_p, \hat{\theta}_1, \dots \hat{\theta}_q)$ MLE from above

For n large, $\hat{\underset{\sim}{\beta}} \sim MVN(\underset{\sim}{\beta}, V(\beta))$

where $V(\beta) = \sigma^2 \left(\begin{array}{c|c} E(\underset{\sim}{u}_t \underset{\sim}{u}_t^T) & E(\underset{\sim}{u}_t \underset{\sim}{v}_t^T) \\ \hline E(\underset{\sim}{v}_t \underset{\sim}{u}_t^T) & E(\underset{\sim}{v}_t \underset{\sim}{v}_t^T) \end{array} \right)^{-1}$

($E(\underset{\sim}{u}_t \underset{\sim}{u}_t^T)$: $p \times p$; $E(\underset{\sim}{u}_t \underset{\sim}{v}_t^T)$: $p \times q$; $E(\underset{\sim}{v}_t \underset{\sim}{u}_t^T)$: $q \times p$; $E(\underset{\sim}{v}_t \underset{\sim}{v}_t^T)$: $q \times q$)

where $\Phi(B) u_t = Z_t$ and $\Theta(B) v_t = Z_t$

($\Phi(B)$: AR(p) with $\phi_1 \dots \phi_p$; $\Theta(B)$: AR(q) with $\theta_1 \dots \theta_q$)

In practice, we can approximate $V(\hat{\underset{\sim}{\beta}})$ with $H^{-1}(\beta)$ where

$$H = \left(\frac{\partial L(\beta)}{\partial \beta_i \partial \beta_j} \right)_{i,j=1}^{p \times q}$$

Note H is estimated during many optimization algorithms.