Applied Time-Series Analysis

Arun K. Tangirala

Department of Chemical Engineering, IIT Madras

Estimation of Time-Series Models

Contents of Lecture

In this lecture, we shall:

- Learn the different techniques for estimating AR models
- Briefly discuss methods for estimating MA models
- Learn how to estimate ARMA and ARIMA models

Estimation of AR models

AR models result in *linear* predictors - therefore a linear OLS method suffices. The linear nature of the AR predictors also attracts a few other specialized methods.

The historical nature and the applicability of this topic is such that numerous texts and survey/tutorial articles (references) dedicated to this topic have been written. We shall only discuss four popular estimators, namely

i. Yule-Walker method
ii. LS / Covariance method
iii. Modified covariance method
iv. Burg's estimator

We shall also briefly discuss the maximum likelihood estimator.

Estimation of auto-regressive models

The AR estimation problem is stated as follows. Given N observations of a stationary process $\{v[k]\}$, $k = 0, \cdots, N-1$, fit an AR(P) model.

$$\boxed{v[k] = \sum_{j=1}^{P} (-d_j) v[k-j] + e[k]} \tag{1}$$

One of the first methods used to estimate AR models was the Yule-Walker method. This method belongs to the class of MoM estimators. It is also one of the simplest to use.

However, the Y-W method is known to produce poor estimates when the true poles are close to the unit circle.

Yule-Walker method

Idea: The second-order moments of the bivariate p.d.f. $f(v[k], v[k-l])$, *i.e.*, the ACVFs of an AR(P) process are related to the parameters of the model as,

$$\underbrace{\begin{bmatrix} \sigma_{vv}[0] & \sigma_{vv}[1] & \cdots & \sigma_{vv}[P-1] \\ \sigma_{vv}[1] & \sigma_{vv}[0] & \cdots & \sigma_{vv}[P-2] \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{vv}[P-1] & \sigma_{vv}[P-2] & \cdots & \sigma_{vv}[0] \end{bmatrix}}_{\Sigma_P} \underbrace{\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_P \end{bmatrix}}_{\boldsymbol{\theta}_P} = - \underbrace{\begin{bmatrix} \sigma_{vv}[1] \\ \sigma_{vv}[2] \\ \vdots \\ \sigma_{vv}[P] \end{bmatrix}}_{\boldsymbol{\sigma}_P}$$

$$\sigma_v^2 + \boldsymbol{\sigma}_P^T \boldsymbol{\theta}_P = \sigma_e^2$$

Y-W method . . . contd.

Thus, the Y-W estimates of the AR(P) model and the innovations variance σ_e^2 are

$$\boxed{\begin{aligned} \hat{\boldsymbol{\theta}} &= -\hat{\Sigma}_P^{-1}\hat{\boldsymbol{\sigma}}_P && \text{(2a)} \\ \hat{\sigma}_e^2 &= \hat{\sigma}_v^2 + \hat{\boldsymbol{\sigma}}_P^T\hat{\boldsymbol{\theta}} = \hat{\sigma}_v^2 - \hat{\boldsymbol{\sigma}}_P^T\hat{\Sigma}_P^{-1}\hat{\boldsymbol{\sigma}}_P && \text{(2b)} \end{aligned}}$$

provided $\hat{\Sigma}_P$ is invertible, which is guaranteed so long as $\sigma[0] > 0$.

Y-W method

The matrix $\hat{\Sigma}_P$ is constructed using the biased estimator of the ACVF

$$\hat{\sigma}[l] = \frac{1}{N} \sum_{k=l}^{N-1} (v[k] - \bar{v})(v[k-l] - \bar{v}) \tag{3}$$

The Y-W estimates can be shown as the solution to the OLS minimization

$$\hat{\boldsymbol{\theta}}_{\mathsf{YW}} = \arg \min_{\boldsymbol{\theta}} \sum_{k=0}^{N+P-1} \varepsilon^2[k] \tag{4}$$

where $\varepsilon[k] = v[k] - \hat{v}[k|k-1] = v[k] - \sum_{i=1}^{P} (-d_i) v[k-i]$

Y-W Method: Remarks

The summation in (4) starts from $k = 0$ and runs up to $k = N + P - 1$. In order to compute the prediction errors from $k = 0, \cdots, P - 1$ and $k = N, \cdots, N + P - 1$, the method pads p zeros to both ends of the series.

This approach is frequently referred to as *pre*- and *post*-windowing of data.

Properties of Y-W estimator

The Y-W estimates, in general, enjoy good asymptotic properties:

1. **For a model of order P, if the process $\{v[k]\}$ is also AR(P),** the parameter estimates asymptotically follow a multivariate Gaussian distribution

$$\sqrt{N}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) \xrightarrow{d} \mathcal{N}\left(0, \sigma_e^2 \Gamma_P^{-1}\right) \tag{5}$$

   In practice, the theoretical variance and covariance matrix are replaced by their respective estimates.

Properties of Y-W estimator . . . contd.

2. The 95% CI for the individual parameter θ_{i0} are *approximately* constructed as

$$\hat{\theta}_i \pm \frac{1.96\hat{\sigma}_e}{\sqrt{N}}(\hat{\Sigma}_P^{-1})_{ii}^{1/2} \tag{6}$$

3. Further, if $\{v[k]\}$ is an AR(P_0) process and an AR model of order $P > P_0$ is fit to the series, then the coefficients *in excess of the true order* are distributed as

$$\sqrt{N}\theta_l \sim \mathsf{AN}(0,1) \qquad \forall l > P_0 \tag{7}$$

To verify this fact, consider fitting an AR(P) model to a white-noise process, *i.e.*, when $P_0 = 0$. Then $\Sigma_P = \sigma_e^2\mathbf{I}$.

Properties of Y-W estimators . . . contd.

4. Recall that the last coefficient of an AR(P) model is the PACF coefficient ϕ_{PP} of the series. By the present notation,

$$\phi_{ll} = -d_l = -\theta_l \tag{8}$$

It follows from the above property that if the true process is AR(P_0), **the 95% significance levels for PACF estimates at lags** $l > P_0$ are

$$\boxed{-\frac{1.96}{\sqrt{N}} \leq \hat{\phi}_{ll} \leq \frac{1.96}{\sqrt{N}}} \tag{9}$$

Properties of Y-W estimator . . . contd.

5. From (5) it follows the *Y-W estimates of an AR model are consistent*.
6. **The Y-W estimator suffers from a drawback**. It *may produce* poor (high variability) estimates when the generating auto-regressive process has poles close to unit circle. The cause is the poor conditioning of the auto-covariance matrix $\hat{\Sigma}_P$ for such processes combined with the bias in the ACVF estimator. The effects of the latter (bias) always prevail, but are magnified when $\hat{\Sigma}_P$ is poorly conditioned.
7. The D-L algorithm facilitates recursive Y-W estimation without having to invert $\hat{\Sigma}_P$.
8. The Toeplitz structure of $\hat{\Sigma}_P$ and the biased ACVF estimator guarantee that the resulting model is *stable and minimum phase*.

Example

Y-W method

A series consisting of $N = 500$ observations of a random process is given. Fit an AR(2) model using the Y-W method.

Solution: The variance and ACF estimates at lags $l = 1, 2$ are computed to be $\hat{\sigma}[0] = 7.1113$, $\hat{\rho}[1] = 0.9155$, $\hat{\rho}[2] = 0.7776$ respectively. Plugging in these estimates into (2a) produces

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{d}_1 \\ \hat{d}_2 \end{bmatrix} = - \begin{bmatrix} 1 & 0.9155 \\ 0.9155 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0.9155 \\ 0.7776 \end{bmatrix} = \begin{bmatrix} -1.258 \\ 0.374 \end{bmatrix} \tag{10}$$

Example

The estimate of the innovations variance can be computed using (2b)

$$\hat{\sigma}_e^2 = 7.1113 + \begin{bmatrix} 0.9155 & 0.7776 \end{bmatrix} \begin{bmatrix} -1.258 \\ 0.374 \end{bmatrix} = 0.9899 \tag{11}$$

The errors in the estimates can be computed from (5) by replacing the theoretical values with their estimated counterparts.

$$\Sigma_{\hat{\boldsymbol{\theta}}} = 0.9899 \begin{bmatrix} 1 & 0.9155 \\ 0.9155 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 0.0017 & -0.0016 \\ -0.0016 & 0.0017 \end{bmatrix} \tag{12}$$

Example . . . contd.

Consequently, approximate 95% C.I.s for d_1 and d_2 are $[-1.3393, -1.1767]$ and $[0.2928, 0.4554]$ respectively.

Compare the estimates and C.I.s with the respective true values used for simulation

$$d_{1,0} = -1.2; \quad d_{20} = 0.32; \sigma_e^2 = 1 \tag{13}$$

Note: The Y-W estimator is generally used when the data length is large and it is known a priori that the generating process has poles well within the unit circle. In general, it is used to initialize other non-linear estimators.

Least squares / Covariance method

The least squares method obtains the estimate as

$$\hat{\boldsymbol{\theta}}_{\mathsf{LS}} = \arg \min_{\boldsymbol{\theta}} \sum_{k=p}^{N-1} \varepsilon^2[k] \tag{14}$$

Comparing with the standard *linear* regression form, we have

$$\boldsymbol{\varphi}[k] = \begin{bmatrix} -v[k-1] & \cdots & -v[k-P] \end{bmatrix}^T; \qquad \boldsymbol{\theta} = \mathbf{d} = \begin{bmatrix} d_1 & \cdots & d_P \end{bmatrix}^T \tag{15}$$

Least squares / Covariance method ...contd.

Using the LS solution, we have

$$\hat{\boldsymbol{\theta}}_{\text{LS}} = \hat{\mathbf{d}}_{\text{LS}} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{v} = \left(\frac{1}{N-P}\Phi^T\Phi\right)^{-1}\left(\frac{1}{N-P}\Phi^T\mathbf{v}\right) \tag{16a}$$

$$\text{where} \quad \Phi = \begin{bmatrix}\boldsymbol{\varphi}[P] & \boldsymbol{\varphi}[P+1] & \cdots & \boldsymbol{\varphi}[N-1]\end{bmatrix}^T \tag{16b}$$

$$\mathbf{v} = \begin{bmatrix}v[P] & v[P+1] & \cdots & v[N-1]\end{bmatrix}^T \tag{16c}$$

LS / COV method ...contd.

A careful examination of (16a) suggests that it can be written as a MoM estimate

$$\hat{\boldsymbol{\theta}} = -\hat{\Sigma}_P^{-1}\hat{\sigma}_P \tag{17}$$

$$\text{where } \hat{\Sigma}_P \triangleq \frac{1}{N-P}\Phi^T\Phi = \begin{bmatrix} \hat{\sigma}_{vv}[1,1] & \hat{\sigma}_{vv}[1,2] & \cdots & \hat{\sigma}_{vv}[1,P] \\ \vdots & \vdots & \cdots & \vdots \\ \hat{\sigma}_{vv}[P,1] & \hat{\sigma}_{vv}[P,2] & \cdots & \hat{\sigma}_P[P,P] \end{bmatrix} \tag{18}$$

$$\hat{\sigma}_P \triangleq \frac{1}{N-P}\Phi^T\mathbf{v} = \begin{bmatrix} \hat{\sigma}_{vv}[1,1] \\ \vdots \\ \hat{\sigma}_{vv}[P,1] \end{bmatrix} \tag{19}$$

LS / COV method ...contd.

where the estimate of the ACVF is given by

$$\hat{\sigma}_{vv}[l_1, l_2] = \frac{1}{N-P} \sum_{n=P}^{N-1} v[n-l_1]v[n-l_2] \tag{20}$$

Observe that $\hat{\Sigma}_P$ is a symmetric matrix by virtue of (20). Due to the equivalence above, the method is also known as the **covariance method**.

Modified covariance method

The modified covariance (MCOV) method stems from a modification of the objective function in the LS approach. It minimizes the sum squares of *both forward and backward* prediction errors, ε_F and ε_B respectively.

$$\hat{\boldsymbol{\theta}}_{\mathsf{LS}} = \arg \min_{\boldsymbol{\theta}} \left(\sum_{k=p}^{N-1} \varepsilon_F^2[k] + \sum_{k=0}^{N-p-1} \varepsilon_B^2[k] \right) \tag{21}$$

By a change of summation index, the objective function can also be written as

$$\sum_{k=p}^{N-1} \varepsilon_F^2[k] + \sum_{k=0}^{N-p-1} \varepsilon_B^2[k] = \sum_{k=p}^{N-1} (\varepsilon_F^2[k] + \varepsilon_B^2[k-P]) \tag{22}$$

MCOV method . . . contd.

The *backward* prediction error is defined in a similar way as the forward version:

$$\varepsilon_B[k] = v[k] - \hat{v}[k|\{v[k+1], \cdots, v[k+P]\}] = v[k] - \sum_{i=1}^{P}(-d_i)v[k+i] \tag{23}$$

Thus, the objective in the MCOV method is to minimize

$$\sum_{k=p}^{N-1}\left[\left(v[k] + \sum_{i=1}^{P} d_i v[k-i]\right)^2 + \left(v[k-P] + \sum_{i=1}^{P} d_i v[k-P+i]\right)^2\right] \tag{24}$$

MCOV method . . . contd.

The solution to this optimization problem is of the same form as from the LS/COV method but by replacing the auto-covariance estimate with the one given below.

$$\hat{\boldsymbol{\theta}}_{\text{MCOV}} = -\hat{\Sigma}_P^{-1}\hat{\sigma}_P \tag{25a}$$

$$\hat{\sigma}_{vv}[l_1, l_2] = \sum_{k=p}^{N-1}(v[k-l_1]v[k-l_2] + v[k-P+l_1]v[k-P+l_2]) \tag{25b}$$

$$\hat{\Sigma}_{P,ij} = \hat{\sigma}[i,j]; \quad \hat{\sigma}_{P,i} = \hat{\sigma}[i,1], \qquad i = 1, \cdots, P; \; j = 1, \cdots, P \tag{25c}$$

Note: The covariance matrix $\hat{\Sigma}_P$ is no longer Toeplitz and therefore a recursion algorithm such as the D-L method cannot be applied.

Properties of covariance estimators

1. In both the LS and MCOV methods, the regressor $\boldsymbol{\varphi}[k]$ and the prediction error are constructed from $k = P$ to $k = N - 1$ unlike in the Y-W method. *Thus, the LS and the MCOV methods do not pad the data.*
2. The asymptotic properties of the covariance (LS) and the MCOV estimators are, however, identical to that of the Y-W estimator.
3. Unfortunately, **stability of the resulting models is not guaranteed while using the covariance-based estimators**. Moreover, the covariance matrix does not possess a Toeplitz structure, which is disadvantageous from a computational viewpoint.

Example

Estimating AR(2) using LS and MCOV

For the series of the example illustrating Y-W method, estimate the parameters using the LS and MCOV methods.

Solution: The LS and MCOV methods yield, respectively,

$$\hat{d}_1 = -1.269; \;\; \hat{d}_2 = 0.3833 \qquad\qquad \hat{d}_1 = -1.268; \;\; \hat{d}_2 = 0.3827$$

which only slightly differ among each other and the Y-W estimates.
The standard errors in both estimates are identical to those computed in the Y-W case by virtue of the properties discussed above.

Burg's estimator

Burg's method (Burg's reference) minimizes the same objective as the MCOV method except that it aims at incorporating two desirable features:

i. Stability of the estimated AR model

ii. A D-L like recursion algorithm for parameter estimation.

The key idea is to **employ the reflection coefficient (negative PACF coefficient)-based AR representation**. Therefore, the reflection coefficients $\kappa_p,\ p = 1, \cdots, P$ are estimated instead of the model parameters. Stability of the model is guaranteed by *requiring the magnitudes of the estimated reflection coefficients to be each less than unity*.

Burg's method . . . contd.

The optimization problem remains the same as in the MCOV method.

$$\hat{\boldsymbol{\theta}}_{\text{Burg}} = \arg \min_{\kappa_p} \sum_{k=p}^{N-1} (\varepsilon_F^2[k] + \varepsilon_B^2[k-P]) \tag{26}$$

In order to force a D-L like recursive solution, the forward and backward prediction errors associated with a model of order p are re-written as follows:

$$\varepsilon_F^{(p)}[k] = v[k] + \sum_{i=1}^{p} d_i v[k-i] = \begin{bmatrix} v[k] & \cdots & v[k-p] \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{\theta}^{(p)} \end{bmatrix} \tag{27}$$

$$\varepsilon_B^{(p)}[k-p] = v[k-p] + \sum_{i=1}^{p} d_i v[k-p+i] = \begin{bmatrix} v[k] & \cdots & v[k-p] \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{\theta}}^{(p)} \\ 1 \end{bmatrix} \tag{28}$$

Burg's method . . . contd.

Then, using D-L algorithm, we have

$$\boldsymbol{\theta}^{(p)} = \begin{bmatrix} \boldsymbol{\theta}^{(p-1)} + \kappa_p \bar{\boldsymbol{\theta}}^{(p-1)} \\ \kappa_p \end{bmatrix} \tag{29}$$

from where the following recursive relations can be obtained:

$$\varepsilon_F^{(p)}[k] = \varepsilon_F^{(p-1)}[k] + \kappa_p \varepsilon_B^{(p-1)}[k-p] \tag{30}$$

$$\varepsilon_B^{(p)}[k-p] = \varepsilon_B^{(p-1)}[k-p] + \kappa_p \varepsilon_F^{(p-1)}[k] \tag{31}$$

Burg's method . . . contd.

Inserting the recursive relations into the objective function and solving

$$\hat{\kappa}_p^{\star} = -2\frac{\sum_{n=p}^{N-1} \varepsilon_F^{(p-1)}[n]\varepsilon_B^{(p-1)}[n-p]}{\sum_{n=p}^{N-1}\left((\varepsilon_F^{(p-1)}[n])^2 + (\varepsilon_B^{(p-1)}[n-p])^2\right)} \tag{32}$$

Stability of the estimated model can be verified by showing that the optimal reflection coefficient in (32) satisfies $|\kappa_p| \leq 1,\ \forall p$.

Burg's method . . . contd.

The estimates of the innovations variance are also recursively updated as:

$$\boxed{\hat{\sigma}_e^{2(p)} = \hat{\sigma}_e^{2(p-1)}(1 - \hat{\kappa}_p^2)} \tag{33}$$

Given that the reflection coefficients are always less than unity in magnitude, the innovations variance is guaranteed to decrease with increase in order.

Burg's estimation procedure

A basic procedure for Burg's algorithm thus follows:

Burg's method

1. Set $p = 0$ and $\boldsymbol{\theta}^{(0)} = 0$ so that the forward and backward prediction errors are initialized to $\varepsilon_F^{(0)}[k] = v[k] = \varepsilon_F^{(0)}[k]$.
2. Increment the order p by one and compute κ_{p+1} using (32).
3. Update the parameter vector $\boldsymbol{\theta}^{(p+1)}$ using (29).
4. Update the prediction errors for the incremented order using (27) and (28)
5. Repeat steps 2-4 until a desired order $p = P$.

Properties of Burg's estimator

Asymptotic properties of optimal estimates of κ_p are not trivial to derive. The following is a summary of facts from extensive studies by several researchers:

1. The bias of Burg's estimates are as large as those of the LS estimates, but lower than those of the Yule-Walker, especially when the underlying process is auto-regressive with roots near the unit circle.
2. The variance of $\hat{\kappa}_p$ for models with orders $p \geq P_0$ is given by

$$\mathsf{var}(\hat{\kappa}_p) = \begin{cases} \dfrac{1-\kappa_p^2}{N}, & p = P_0 \\ \dfrac{1}{N}, & p > P_0 \end{cases} \tag{34}$$

Properties of Burg's estimator ...contd.

Note that the case of $p > P_0$ is consistent with the result for the variance of the PACF coefficient estimates at lags $l > P_0$ given by (7).

3. The innovations variance estimate is asymptotically unbiased, again when the postulated order is at least equal to the true order

$$E(\hat{\sigma}_e^2) = \sigma_e^2 \left(1 - \frac{p}{N}\right), p \geq P_0 \qquad \Longrightarrow \lim_{N\to\infty} E(\hat{\sigma}_e^2) = \sigma_e^2 \tag{35}$$

4. All reflection coefficients for orders $p \geq P_0$ are independent of the lower order estimates.

Properties of Burg's estimator . . . contd.

5. By the asymptotic equivalence of Burg's method with the Y-W estimator, the distribution and covariance of resulting parameter estimates are identical to that given in (5). The difference is in the point estimate of $\boldsymbol{\theta}$ and the estimate of the innovations variance.
6. Finally, a distinct property of Burg's estimator is that **it guarantees stability of AR models**.

Example

Simulated AR(2) series

For the simulated series considered in the previous examples, obtain Burg's estimates of the model parameters.

Solution:

$$\hat{d}_1 = -1.267; \quad \hat{d}_2 = 0.3827$$

which are almost identical to the MCOV estimates.

Once again given the large sample size, the asymptotic properties can be expected to be identical to those of previous methods.

Estimation of AR models using MLE

We shall illustrate now the estimation of AR models using MLE through an example.

Estimating parameters of an AR(1) model

Given N observations $\mathbf{y}_N$ of a random process, fit a first-order AR model.

$$v[k] = -d_1 v[k-1] + e[k], \qquad e[k] \sim \mathcal{N}(0, \sigma_e^2)$$

Thus, the parameters to be estimated are $\boldsymbol{\theta} = \begin{bmatrix} d_1 & \sigma_e^2 \end{bmatrix}^T$

Example: MLE for AR models ...contd.

Solution:

1. **Density function:** The joint p.d.f. of $\mathbf{v}_N$ is **not** the product of the marginals since $\{v[0], v[1], \cdots, v[N-1]\}$ forms a correlated series.
   Fortunately, the conditioned series $y[k]|y[k-1]$ is uncorrelated. Why?

$$v[k]|v[k-1] = -d_1 v[k-1] + e[k]|v[k-1] \qquad (v[k-1] \text{ is fixed})$$
$$v[k-1]|v[k-2] = -d_1 v[k-2] + e[k-1]|v[k-2] \qquad (v[k-2] \text{ is fixed})$$

   Therefore, $\text{corr}(v[k|v[k-1]], v[k-1|v[k-2]]) = \text{corr}(e[k], e[k-1]) = 0$.
   Subsequently, applying $k = 1$ onwards and invoking Bayes rule, we have

$$f(\mathbf{v}_N|\boldsymbol{\theta}) = f(v[0])f(v[1]|v[0]) \cdots f(v[N-1]|v[N-2]) = f(v[0])\Pi_{k=1}^{N-1} f(v[k]|v[k-1])$$

Example: MLE for AR models ...contd.

Noting that $e[k]$ is a Gaussian, $v[k]$ is also a Gaussian. Further,

$$E(v[0]) = 0; \quad \text{var}(v[0]) = \frac{\sigma_e^2}{1-d_1^2} \;\; \forall \; k \leq 0$$

$$E(v[k]|v[k-1]) = -d_1 v[k-1] = \hat{v}[k|k-1]$$

$$\text{var}(v[k]|v[k-1]) = \sigma_e^2$$

The corresponding density functions are therefore,

$$f(v[0]) = \frac{\sqrt{1-d_1^2}}{\sqrt{2\pi\sigma_e^2}} \exp\left(-\frac{1}{2}\frac{v^2[0](1-d_1^2)}{\sigma_e^2}\right)$$

$$f(v[k]|v[k-1]) = \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp\left(-\frac{1}{2}\frac{(v[k]-\hat{v}[k|k-1])^2}{\sigma_e^2}\right)$$

Example: MLE for AR models ...contd.

Putting together the foregoing expressions, we finally have the log-likelihood function

$$L(\boldsymbol{\theta}|\mathbf{v}_N) = \text{const.} + \frac{1}{2}\ln(1-d_1^2) - \frac{N}{2}\ln\sigma_e^2 - \frac{1}{2}\frac{v^2[0](1-d_1^2)}{\sigma_e^2} - \frac{1}{2}\sum_{k=1}^{N-1}\frac{(v[k]-\hat{v}[k])^2}{\sigma_e^2}$$

$$= \text{const.} + \frac{1}{2}\ln(1-d_1^2) - \frac{N}{2}\ln\sigma_e^2 - \frac{1}{2}\frac{v^2[0](1-d_1^2)}{\sigma_e^2} - \frac{1}{2\sigma_e^2}\underbrace{\sum_{k=1}^{N-1}\epsilon^2[k]}_{\text{LS obj. fun.}} \quad (36a)$$

Example: MLE for AR models ...contd.

Notice that once again the LS objective function is contained in the MLE formulation. *The main difference is that MLE takes into account the randomness of the first observation while the LSE takes it to be fixed.*

Introduce as in Shumway and Stoffer, 2006, two quantities

$$\mathfrak{S}_c(d_1) = \sum_{k=1}^{N-1}(v[k] - \hat{v}[k])^2 \qquad \text{(conditional sum squares)} \qquad (37)$$

$$\mathfrak{S}_u(d_1) = v^2[0](1-d_1^2) + \sum_{k=1}^{N-1}(v[k] - \hat{v}[k])^2 \qquad \text{(unconditional sum squares)} \qquad (38)$$

Example: MLE for AR models . . . contd.

so that (36) can be written as

$$L(d_1, \sigma_e^2) = \text{const.} + \frac{1}{2}\ln(1 - d_1^2) - \frac{N}{2}\ln\sigma_e^2 - \mathfrak{S}_u(d_1, \sigma_e^2) \tag{39}$$

The *conditional sum* is minimized by the *ordinary least squares* estimator, whereas the *unconditional sum* is minimized by the *weighted least squares* estimator.

Example: MLE for AR models ...contd.

The log-likelihood is obviously non-linear in the unknowns. No closed-form expression is available for the optimal estimate of d_1. However, the optimal estimate of innovations variance can be computed explicitly from (39) as follows:

$$\frac{\partial L(d_1, \sigma_e^2)}{\partial \sigma_e^2} = 0 \Longrightarrow \qquad \boxed{\hat{\sigma}^2_{e,\mathsf{ML}} = \frac{\mathfrak{S}_u(\hat{d}_{1,ML})}{N}} \tag{40}$$

where $\hat{d}_{1,ML}$ and $\hat{\sigma}^2_{e,\mathsf{ML}}$ have their obvious meanings.

Numerical example

For a given process data, let us estimate the parameters using MLE. The process used for simulation is also an AR(1) generating $N = 500$ observations.

$$v[k] - 0.7v[k-1] = e[k] \qquad e[k] \sim \mathcal{N}(0,1)$$

Setting up the log-likelihood function and solving for the resulting optimization problem yields,

$$\hat{d}_{1,\mathsf{ML}} = -0.701; \qquad \hat{\sigma}^2_{e,\mathsf{ML}} = 1.002 \tag{41}$$

Note that the ML estimates are local optima whereas the LS estimates are unique.

R code for computing the ML estimates

```
# Generate data
vk <- arima.sim(model=list(order=c(1,0,0),ar=0.7),n=500)

# Set up the log-likelihood function
neglogl <- function(dpar,sigmae) {
  N = length(vk)
  pred_err = vk[2:N] - (-dpar)*vk[1:N-1]
  logl = 0.5*(log(1 - dpar^2) - N*log(sigmae^2) - vk[1]^2*(1 - dpar^2)/sigmae^2 -
  sum(pred_err^2)/sigmae^2)
  return(-logl)
}
# Estimate using the 'mle2' routine of the 'bbmle' package
library(bbmle)
thetahat <- mle2(neglogl,start=list(dpar=-0.4,sigmae=0.5))
summary(thetahat)
# Compare with results from 'ar.mle'
theta_armle <- ar.mle(vk,order.max=1,aic=F)
```

Remarks

- The ML approach to estimating a general AR(P) model is now a straightforward extension of the illustrated AR(1) example.
- The main difference is that the joint p.d.f. would be conditioned on the first P observations.
- Importantly, the log-likelihood would still contain the sum-square prediction errors (from $k = P$ to $k = N - 1$).
- Similar notions of CSS and UCSS (or CSS-ML) apply.

Estimation of MA models

The problem of estimating an MA model is more involved than that of the AR parameters primarily because the predictor is non-linear in the unknowns.

With an MA(M) model the predictor is

$$\hat{v}[k|k-1] = c_1 e[k-1] + \cdots + c_M e[k-M], \quad k \geq M \tag{42}$$

wherein both the parameters and the past innovations are unknown. The past innovations are a non-linear function of the model parameters.

Estimation of MA models . . . contd.

Thus, the non-linear least squares (NLS) estimation method and the MLE are popularly used for estimating MA models. Both these methods require a proper initialization in order to get out of local minima.

Four popular methods for obtaining preliminary estimates are briefly discussed. For details, read (Brockwell, 2002).

Preliminary estimates of MA models

1. **Method of moments:** Same as Y-W method, but now the equations are non-linear. Only invertible solutions are accepted.
2. **Durbin's estimator:** Idea is to first generate the innovation sequence by a high-order AR model. Subsequently, re-write the MA(M) model as

$$v[k] - \hat{e}[k] = \sum_{i=1}^{M} c_i \hat{e}[k-i] \tag{43}$$

where $\hat{e}[k] = \hat{D}(q^{-1})v[k]$ is the estimate obtained from the AR model.

Note: The order of the AR model used for this purpose can be selected in different ways, for *e.g.*, using AIC or BIC. A simple guideline recommends $P = 2M$.

Preliminary estimates of MA models . . . contd.

3. **Innovations algorithm:** It is similar to the D-L algorithm for AR models. The key idea is to use the innovations representation of the MA model by recalling that the white-noise sequences are also theoretically the one-step ahead predictions.
   Defining $c_0 \equiv 1$

$$v[k] = \sum_{i=0}^{M} c_i e[k-i] = \sum_{i=0}^{M} c_i (v[k] - \hat{v}[k|k-1]) \tag{44}$$

Preliminary estimates of MA models ...contd.

A recursive algorithm can be then constructed.

i. Set $m = 0$ and $\hat{\sigma}^2_{e,0} = \hat{\sigma}^2_v$.

ii. Compute

$$\hat{c}_{m,m-j} = (\hat{\sigma}^2_{e,m})^{-1} \left(\sigma_{vv}[m-j] - \sum_{i=0}^{j-1} \hat{c}_{j,j-i} \hat{c}_{m,m-i} \hat{\sigma}^2_{e,i} \right), \qquad 0 \leq j < m \tag{45}$$

iii. Update the innovations variance $\hat{\sigma}^2_{e,m} = \hat{\sigma}^2_v - \sum_{j=0}^{M-1} \hat{c}^2_{m,m-j} \hat{\sigma}^2_{e,j}$

iv. Repeat steps (ii) and (iii) until a desired order $m = M$.

Preliminary estimates of MA models ...contd.

4. **Hannan-Rissanen's method:** The approach is similar to that of Durbin's estimator. However, the difference is that the parameters are estimated from a linear least-squares regression of $v[k]$ on estimated past innovations:

$$\hat{v}[k] = \sum_{i=1}^{M} c_i \hat{e}[k-i], \quad k \geq M \tag{46}$$

The past terms of $\hat{e}[k]$ are obtained as the residuals of a sufficiently high AR (p) model. The parameter estimates can be further updated using an additional step, but it can be usually avoided.

Estimation of ARMA models

Given a set of N observations $\{v[0], v[1], \cdots, v[N-1]\}$ of a process, estimate the $P' = P + M$ parameters $\boldsymbol{\theta} = \begin{bmatrix} d_1 & \cdots & d_P & c_1 & \cdots & c_M \end{bmatrix}^T$ of the ARMA(P, M) model

$$v[k] + \sum_{j=1}^{P} d_j v[k-j] = \sum_{i=1}^{M} c_i e[k-i] + e[k] \tag{47}$$

and the innovations variance σ_e^2. It is assumed without loss of generality that the generating process is zero-mean.

Estimation of ARMA models . . . contd.

- With the nonlinear LS method, typically a Gauss-Newton method is used. Analytical expressions are used to compute the gradients (of the predictor) at each iteration.
- In the MLE approach, the likelihood function is set up using the prediction error (innovations) approach and a nonlinear optimization solver such as the G-N method is used.
- Any one of the four methods discussed earlier for MA models can be used to initialize the algorithms. The Y-W method is the standard choice.

See Shumway and Stoffer, 2006; Tangirala, 2014 for a theoretical discussion of the NLS and MLE algorithms, *i.e.*, how to evaluate the gradients for the former or set up the likelihood functions for the latter.

NLS and ML estimators of ARMA models

The parameter estimates of an ARMA(P, M) model obtained from the unconditional, conditional least squares and the ML estimators initialized with the MoM are asymptotically consistent. Further,

$$\sqrt{N}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) \sim \text{AN}\left(\mathbf{0}, \sigma_e^2 \mathbb{S}(\boldsymbol{\theta}_0)^{-1}\right) \tag{48}$$

The $(P+M) \times (P+M)$ covariance matrix $\mathbb{S}$ is given by

$$\mathbb{S} = \begin{bmatrix} E(\mathbf{x}_P \mathbf{x}_P^T) & E(\mathbf{x}_P \mathbf{w}_M^T) \\ E(\mathbf{w}_M \mathbf{x}_P^T) & E(\mathbf{w}_M \mathbf{w}_M^T) \end{bmatrix} \tag{49}$$

where $\mathbf{x}_P$ and $\mathbf{w}_M$ are constructed from two auto-regressive processes

$$\mathbf{x}_P = \begin{bmatrix} x[k-1] & x[k-2] & \cdots & x[k-P] \end{bmatrix}^T; \qquad x[k] = \frac{1}{D(q^{-1})} e[k] \tag{50}$$

$$\mathbf{w}_M = \begin{bmatrix} w[k-1] & w[k-2] & \cdots & w[k-M] \end{bmatrix}^T; \qquad w[k] = \frac{1}{C(q^{-1})} e[k] \tag{51}$$

Remarks

- The block diagonals $\mathbb{S}_{11}$ $(P \times P)$ and $\mathbb{S}_{22}$ $(M \times M)$ are essentially the auto-covariance matrices of $x[k]$ and $w[k]$ respectively, while the off-diagonals are the matrices of cross-covariance functions between $x[k]$ and $w[k]$.
- A few special cases are discussed
  1. **AR(1):** For this case, $\mathbb{S}$ is a scalar. Using (49),

$$\mathbb{S} = E(x[k-1]x[k-1]) = \sigma_e^2/(1-d_1^2) \Longrightarrow \mathsf{var}(\hat{d}_1) = (1-d_1^2) \tag{52}$$

  2. **MA(1):** Using (49),

$$\mathbb{S} = E(w[k-1]w[k-1]) = \sigma^2/(1-c_1^2) \Longrightarrow \mathsf{var}(\hat{c}_1) = (1-c_1^2) \tag{53}$$

Procedure to fit an ARMA model

1. Carry out a visual examination of the series. Inspect the data for "outliers", drifts, significantly differing variances, *etc*.
2. Perform the necessary pre-processing of data (e.g., removal of trends, transformation) to obtain a stationary series.
3. For pure AR models, use PACF and likewise for pure MA models, use ACF for estimating the orders. For ARMA models, a good start is an ARMA(1,1) model.

Procedure to fit an ARMA model

4.
   - For AR models, use the MCOV or Burg's method with the chosen order. If the purpose is spectral estimation, then prefer the MCOV method.
   - For MA and ARMA models, generate preliminary estimates (typically using the Y-W or the H-R method) with the chosen orders. Use these preliminary estimates with an MLE or NLS algorithm to obtain "optimal" estimates.
5. Subject the model to a quality (diagnostic) check. If the model passes all the checks, then accept this model. Else work towards an appropriate model order until satisfactory results are obtained.

Steps for building an ARIMA model

Procedure

1. Examine/test the series for integrating type non-stationarity using visual inspection of the series and/or the ACF plots and the *unit root tests* (e.g., Dickey-Fuller, Phillips-Perron tests). If the series exhibits strong evidence for unit roots, then an ARIMA model can be fit after following steps 2 and 3 below.
   - Unit root tests have to be performed with care and should be corroborated with visual observations of the series as well as the ACF/PACF plots.
2. If there is a strong evidence (additionally) for trend type non-stationarities, remove them by fitting polynomial functions to the series (using OLS method for example) and work with the residuals of this fit. Denote these by $w[k]$.

Steps for building an ARIMA model . . . contd.

3. If the residuals (or the series in the absence of trends) are additionally known to contain growth effects, then a logarithmic transformation is recommended. Call the resulting series as $\tilde{w}[k]$ or $\tilde{v}[k]$ as the case maybe.
4. Determine the appropriate degree of differencing d (by a visual or statistical testing of the differenced series).
5. Fit an ARMA model to $\nabla^d \tilde{w}[k]$ or $\nabla^d \tilde{v}[k]$ (or to the respective untransformed series if step 3 is skipped).

Seasonal ARIMA models

In general, seasonal effects need not be periodic but can have a stationary process like characteristics operating at the seasonal scale.

SARIMA Model

If d and D are nonnegative integers, then $\{v[k]\}$ is a seasonal ARIMA$(p, d, m) \times (P, D, M)$ process with period s, if the differenced series

$$w[k] = (1 - q^{-1})^d (1 - q^{-s})^D v[k]$$

is a causal ARMA process defined by

$$D(q^{-1}) D_s(q^{-s}) v[k] = C(q^{-1}) C_s(q^{-s}) e[k], \quad e[k] \sim \mathcal{N}(0, \sigma_e^2)$$

Remarks

- A seasonal ARIMA process has two sub-phenomena evolving at two different scales, one at the regular (observed) scale and another at the seasonal (to be usually determined from data).
  - E.g.: sales of clothes / sweets in a city, temperature changes in the atmosphere.
- The SARIMA model essentially is a multiplicative type model (as compared to the classical or Holt-Winters forecasting model).
- It takes into account the "interaction" of the seasonal scale phenomenon with that of the regular scale. In contrast, additive models do not take this "interaction" into account.

R commands

Listing 1: R commands for simulating and fitting SARIMA models

```
# Simulating ARIMA and SARIMA models
arima.sim, simulate.Arima (from forecast)
# Non-parametric analysis
acf, pacf, tsdisplay (forecast), periodogram (TSA), stl, HoltWinters
# Tests for unit roots (integrating effects)
adf.test (tseries), kpss.test (tseries)
# Estimating AR models
ar, ar.yw, ar.ols, ar.burg, ar.ole
# Estimating (S)ARIMA models
arima, Arima (from forecast),
# Predicting and plotting
predict, forecast (forecast), ts.plot
```

Summary

- AR models are much easier to estimate than MA models because they give rise to linear predictors
- A variety of methods are available to estimate AR models - popular ones being the Yule-Walker, LS / COV, modified covariance and Burg's method.
  - Among the four methods, Y-W and Burg's method guarantee stability, but the latter is better for processes with poles close to unit circle.
  - MCOV method is preferred when AR models are used in spectral estimation.
  - ML methods are generally not used for estimating AR models because the improvement achieved is marginal

Summary

- ARMA (and MA) models give rise to non-linear optimization algorithms, which in turn require preliminary estimates.
- Initial guesses are generated by less-efficient methods (e.g., Y-W, H-R estimators).
- NLS and ML estimators both yield asymptotically similar ARMA model estimates.
- The "best" ARMA model is almost always determined iteratively, but in a systematic manner.

Bibliography I

- Brockwell, P. (2002). *Introduction to Time-Series and Forecasting*. New York, USA: Springer-Verlag.
- Shumway, R. and D. Stoffer (2006). *Time Series Analysis and its Applications*. New York, USA: Springer-Verlag.
- Tangirala, A. K. (2014). *Principles of System Identification: Theory and Practice*. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group.