

# The Yule Walker Equations for the AR Coefficients

Gidon Eshel

If you assume a given zero-mean discrete timeseries  $\{x_i\}_1^N$  is an AR process, you will naturally want to estimate the appropriate order  $p$  of the AR( $p$ ),

$$x_{i+1} = \phi_1 x_i + \phi_2 x_{i-1} + \cdots + \phi_p x_{i-p+1} + \xi_{i+1} \quad (1)$$

and the corresponding coefficients  $\{\phi_j\}$ . There are (at least) 2 methods, and those are described in this section.

## 1 Direct Inversion

The first possibility is to form a set of direct inversions,

### 1.1 $p = 1$

With

$$x_{i+1} = \phi_1 x_i + \xi_{i+1},$$

one can form the over-determined system

$$\underbrace{\begin{pmatrix} x_2 \\ x_3 \\ \vdots \\ x_N \end{pmatrix}}_{\mathbf{b}} = \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{pmatrix}}_{\mathbf{A}} \phi_1$$

which can be readily solve using the usual least-squares estimator

$$\hat{\phi}_1 = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} = \frac{\sum_{i=1}^{N-1} x_i x_{i+1}}{\sum_{i=1}^{N-1} x_i^2} = \frac{c_1}{c_0} = r_1$$

where  $c_i$  and  $r_i$  are the  $i$ th autocovariance and autocorrelation coefficients, respectively.

## 1.2 $p = 2$

With

$$x_{i+1} = \phi_1 x_i + \phi_2 x_{i-1} + \xi_{i+1},$$

start by forming the over-determined system

$$\underbrace{\begin{pmatrix} x_3 \\ x_4 \\ \vdots \\ x_N \end{pmatrix}}_{\mathbf{b}} = \underbrace{\begin{pmatrix} x_2 & x_1 \\ x_3 & x_2 \\ \vdots & \vdots \\ x_{N-1} & x_{N-2} \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}}_{\mathbf{\Phi}}.$$

Unlike the previous  $p = 1$  case, trying to express the solution

$$\hat{\mathbf{\Phi}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

analytically is not trivial. We start with

$$\begin{aligned} (\mathbf{A}^T \mathbf{A})^{-1} &= \left[ \begin{pmatrix} x_2 & x_3 & \cdots & x_{N-1} \\ x_1 & x_2 & \cdots & x_{N-2} \end{pmatrix} \begin{pmatrix} x_2 & x_1 \\ x_3 & x_2 \\ x_{N-1} & x_{N-2} \end{pmatrix} \right]^{-1} \\ &= \begin{pmatrix} \sum_{i=2}^{N-1} x_i^2 & \sum_{i=2}^{N-1} x_i x_{i-1} \\ \sum_{i=2}^{N-1} x_i x_{i-1} & \sum_{i=1}^{N-2} x_i^2 \end{pmatrix}^{-1} \\ &= \frac{1}{\sum_{i=2}^{N-1} x_i^2 \sum_{i=1}^{N-2} x_i^2 - \sum_{i=2}^{N-1} x_i x_{i-1} \sum_{i=2}^{N-1} x_i x_{i-1}} \begin{pmatrix} \sum_{i=1}^{N-2} x_i^2 & -\sum_{i=2}^{N-1} x_i x_{i-1} \\ -\sum_{i=2}^{N-1} x_i x_{i-1} & \sum_{i=2}^{N-1} x_i^2 \end{pmatrix}. \end{aligned}$$

Next, let's use the fact that the timeseries is stationary, so that autocovariance elements are a function of the lag only, not the exact time limits. In this case,

$$\begin{aligned} (\mathbf{A}^T \mathbf{A})^{-1} &= \frac{1}{c_o^2 - c_1^2} \begin{pmatrix} c_o & -c_1 \\ -c_1 & c_o \end{pmatrix}, \\ (\mathbf{A}^T \mathbf{A})^{-1} &= \frac{1}{c_o^2(1 - r_1^2)} \begin{pmatrix} c_o & -c_1 \\ -c_1 & c_o \end{pmatrix}, \\ (\mathbf{A}^T \mathbf{A})^{-1} &= \frac{1}{c_o(1 - r_1^2)} \begin{pmatrix} r_o & -r_1 \\ -r_1 & r_o \end{pmatrix}. \end{aligned}$$

Similarly,

$$\mathbf{A}^T \mathbf{b} = \begin{pmatrix} x_2 & x_3 & \cdots & x_{N-1} \\ x_1 & x_2 & \cdots & x_{N-2} \end{pmatrix} \begin{pmatrix} x_3 \\ x_4 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} \sum_{i=3}^N x_i x_{i-1} \\ \sum_{i=3}^N x_i x_{i-2} \end{pmatrix}$$

which, exploiting again the stationarity of the timeseries, becomes

$$\mathbf{A}^T \mathbf{b} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}.$$

Combining the 2 expressions, we have

$$\begin{aligned} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} &= \frac{1}{c_o(1-r_1^2)} \begin{pmatrix} r_o & -r_1 \\ -r_1 & r_o \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \\ &= \frac{1}{1-r_1^2} \begin{pmatrix} 1 & -r_1 \\ -r_1 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}. \end{aligned}$$

Breaking this into individual components, we get

$$\hat{\phi}_1 = \frac{r_1(1-r_2)}{1-r_1^2}$$

and

$$\hat{\phi}_2 = \frac{r_2-r_1^2}{1-r_1^2}$$

Of course it is possible to continue to explore  $p \geq 3$  cases in this fashion. However, the algebra, while not fundamentally different from the  $p = 2$  case, quickly becomes quite nightmarish. For example, for  $p = 3$ ,

$$\mathbf{A}^T \mathbf{A} = \begin{pmatrix} c_o & c_1 & c_2 \\ c_1 & c_o & c_1 \\ c_2 & c_1 & c_o \end{pmatrix},$$

whose determinant, required for the inversion, is the cumbersome-looking

$$\det(\mathbf{A}^T \mathbf{A}) = c_o \left( c_o^2 - 2c_1^2 + 2\frac{c_1^2 c_2}{c_o} - c_2^2 \right) = c_o [c_o^2 + 2c_1^2(r_2 - 1) - c_2^2],$$

which, on pre-multiplying by the remainder matrix, yields very long expressions.

Fortunately, there is a better, easier way to obtain the AR coefficient for the arbitrary  $p$ , the Yule-Walker Equations.

## 2 The Yule-Walker Equations

Consider the general AR( $p$ )

$$x_{i+1} = \phi_1 x_i + \phi_2 x_{i-1} + \cdots + \phi_p x_{i-p+1} + \xi_{i+1}.$$

### 2.1 Lag 1

- multiply both sides of the model by  $x_i$ ,

$$x_i x_{i+1} = \sum_{j=1}^p (\phi_j x_i x_{i-j+1}) + x_i \xi_{i+1},$$

where  $i$  and  $j$  are the time and term indices, respectively,

- take expectance,

$$\langle x_i x_{i+1} \rangle = \sum_{j=1}^p (\phi_j \langle x_i x_{i-j+1} \rangle) + \langle x_i \xi_{i+1} \rangle$$

where the  $\{\phi_j\}$ s are kept outside the expectance operator because they are deterministic, rather than statistical, quantities.

- note that  $\langle x_i \xi_{i+1} \rangle = 0$  because the shock (or random perturbation)  $\xi$  of the current time is unrelated to—and thus uncorrelated with—previous values of the process,

$$\langle x_i x_{i+1} \rangle = \sum_{j=1}^p (\phi_j \langle x_i x_{i-j+1} \rangle)$$

- divide through by  $(N-1)$ , and use the evenness of the autocovariance,  $c_{-l} = c_l$ ,

$$c_1 = \sum_{j=1}^p \phi_j c_{j-1}$$

- divide through by  $c_0$ ,

$$r_1 = \sum_{j=1}^p \phi_j r_{j-1}.$$

## 2.2 Lag 2

- multiply by  $x_{i-1}$ ,

$$x_{i-1}x_{i+1} = \sum_{j=1}^p (\phi_j x_{i-1}x_{i-j+1}) + x_{i-1}\xi_{i+1},$$

- take expectance,

$$\langle x_{i-1}x_{i+1} \rangle = \sum_{j=1}^p (\phi_j \langle x_{i-1}x_{i-j+1} \rangle) + \langle x_{i-1}\xi_{i+1} \rangle$$

- eliminate the zero correlation forcing term

$$\langle x_{i-1}x_{i+1} \rangle = \sum_{j=1}^p (\phi_j \langle x_{i-1}x_{i-j+1} \rangle)$$

- divide through by  $(N-1)$ , and use  $c_{-l} = c_l$ ,

$$c_2 = \sum_{j=1}^p \phi_j c_{j-2}$$

- divide through by  $c_0$ ,

$$r_2 = \sum_{j=1}^p \phi_j r_{j-2}.$$

## 2.3 Lag k

- multiply by  $x_{i-k-1}$ ,

$$x_{i-k+1}x_{i+1} = \sum_{j=1}^p (\phi_j x_{i-k+1}x_{i-j+1}) + x_{i-k+1}\xi_{i+1},$$

- take expectance,

$$\langle x_{i-k+1}x_{i+1} \rangle = \sum_{j=1}^p (\phi_j \langle x_{i-k+1}x_{i-j+1} \rangle) + \langle x_{i-k+1}\xi_{i+1} \rangle$$

- eliminate the zero correlation forcing term

$$\langle x_{i-k+1}x_{i+1} \rangle = \sum_{j=1}^p (\phi_j \langle x_{i-k+1}x_{i-j+1} \rangle)$$

- divide through by  $(N - 1)$ , and use  $c_{-l} = c_l$ ,

$$c_k = \sum_{j=1}^p \phi_j c_{j-k}$$

- divide through by  $c_o$ ,

$$r_k = \sum_{j=1}^p \phi_j r_{j-k}.$$

## 2.4 Lag p

- multiply by  $x_{i-p+1}$ ,

$$x_{i-p+1}x_{i+1} = \sum_{j=1}^p (\phi_j x_{i-p+1}x_{i-j+1}) + x_{i-p+1}\xi_{i+1},$$

- take expectance,

$$\langle x_{i-p+1}x_{i+1} \rangle = \sum_{j=1}^p (\phi_j \langle x_{i-p+1}x_{i-j+1} \rangle) + \langle x_{i-p+1}\xi_{i+1} \rangle$$

- eliminate the zero correlation forcing term

$$\langle x_{i-p+1}x_{i+1} \rangle = \sum_{j=1}^p (\phi_j \langle x_{i-p+1}x_{i-j+1} \rangle)$$

- divide through by  $(N - 1)$ , and use  $c_{-l} = c_l$ ,

$$c_p = \sum_{j=1}^p \phi_j c_{j-p}$$

- divide through by  $c_o$ ,

$$r_p = \sum_{j=1}^p \phi_j r_{j-p}.$$

## 2.5 Putting it All Together

Rewriting all the equations together yields

$$\begin{aligned}
r_1 &= \phi_1 r_o + \phi_2 r_1 + \phi_3 r_2 + \cdots + \phi_{p-1} r_{p-2} + \phi_p r_{p-1} \\
r_2 &= \phi_1 r_1 + \phi_2 r_o + \phi_3 r_1 + \cdots + \phi_{p-1} r_{p-3} + \phi_p r_{p-2} \\
&\vdots \\
r_{p-1} &= \phi_1 r_{p-2} + \phi_2 r_{p-3} + \phi_3 r_{p-4} + \cdots + \phi_{p-1} r_o + \phi_p r_1 \\
r_p &= \phi_1 r_{p-1} + \phi_2 r_{p-2} + \phi_3 r_{p-3} + \cdots + \phi_{p-1} r_1 + \phi_p r_o
\end{aligned}$$

which can also be written as

$$\begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{p-1} \\ r_p \end{pmatrix} = \begin{pmatrix} r_o & r_1 & r_2 & \cdots & r_{p-2} & r_{p-1} \\ r_1 & r_o & r_1 & \cdots & r_{p-3} & r_{p-2} \\ & \vdots & & & \vdots & \\ r_{p-2} & r_{p-3} & r_{p-4} & \cdots & r_o & r_1 \\ r_{p-1} & r_{p-2} & r_{p-3} & \cdots & r_1 & r_o \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{p-1} \\ \phi_p \end{pmatrix}.$$

Recalling that  $r_o = 1$ , the above equation is also

$$\underbrace{\begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{p-1} \\ r_p \end{pmatrix}}_{\mathbf{r}} = \underbrace{\begin{pmatrix} 1 & r_1 & r_2 & \cdots & r_{p-2} & r_{p-1} \\ r_1 & 1 & r_1 & \cdots & r_{p-3} & r_{p-2} \\ & \vdots & & & \vdots & \\ r_{p-2} & r_{p-3} & r_{p-4} & \cdots & 1 & r_1 \\ r_{p-1} & r_{p-2} & r_{p-3} & \cdots & r_1 & 1 \end{pmatrix}}_{\mathbf{R}} \underbrace{\begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{p-1} \\ \phi_p \end{pmatrix}}_{\mathbf{\Phi}}$$

or succinctly

$$\mathbf{R}\mathbf{\Phi} = \mathbf{r}. \tag{2}$$

Note that this is a well-posed system (with a square coefficients matrix  $\mathbf{R}$ ), i.e., with the same number of constraints (equations,  $\mathbf{R}$ 's rows) as unknowns (the elements  $\phi_j$  of the unknown vector  $\mathbf{\Phi}$ ). Further,  $\mathbf{R}$  is full-rank and symmetric, so that invertability is guaranteed,

$$\hat{\mathbf{\Phi}} = \mathbf{R}^{-1}\mathbf{r}.$$

## 3 The Yule-Walker Equations and the Partial Autocorrelation Function

Equation 2 provides a convenient recursion for computing the pacf. The first step is to compute the acf up to a reasonable cutoff, say  $p \simeq N/4$ . Next, let  $\mathbf{r}^{(i)}$  denote

Equation 2's rhs for the  $p = i$  case. Similarly, let  $\mathbf{R}^{(i)}$  denote the coefficient matrix for the same case. Then

- loop on  $i$ ,  $1 \leq i \leq p$ 
  - compute  $\mathbf{R}^{(i)}$  and  $\mathbf{r}^{(i)}$
  - invert for  $\hat{\Phi}^{(i)}$ ,

$$\hat{\Phi}^{(i)} = (\mathbf{R}^{(i)})^{-1} \mathbf{r}^{(i)} = \begin{pmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \vdots \\ \hat{\phi}_i \end{pmatrix}$$

- discard all  $\hat{\phi}_j$  for  $1 \leq j \leq i - 1$
  - retain  $\hat{\phi}_i$ ,

$$\text{pacf}(i) = \hat{\phi}_i$$

- end loop on  $i$
- plot  $\text{pacf}(i)$  as a function of  $i$ .