Time Series Exercise Sheet 4

Sofia Olhede

March 13, 2025

Exercise 4.1

Determine the sample autocovariances for the Wolfer sunspot numbers posted on Moodle for lags $0, 1, 2, 3$. Determine the Yule-Walker estimators of ϕ_1, ϕ_2 and σ^2 in the model

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \varepsilon_t$$

where Y_t is the mean corrected series, and ε_t is assumed to be mean-zero white noise.

Solution 4.1

We compute the autocovariances with the `acf` function in R (you will need to set the working directory to the directory of the data to run this):

```
sunspot <- read.table("sunspot.txt", header = TRUE)
acvs_sunspot <- acf(x=sunspot$sunspot, type = "covariance", plot = FALSE)
cat("the ACVs from lags zero to three is: ", acvs_sunspot$acf[1:4], "\n")
```

Running this will output:

```
the ACVs from lags zero to three is:  1382.185 1114.378 591.7208 96.21545
```

So we obtain: $\hat{\gamma}(0) = 1382.18510$, $\hat{\gamma}_1 = 1114.37835$, $\hat{\gamma}_2 = 591.72080$, and $\hat{\gamma}_3 = 96.21545$.

Using the estimator

$$\hat{\boldsymbol{\phi}} = \hat{\boldsymbol{\Gamma}}^{-1}\hat{\boldsymbol{\gamma}}$$

where the matrix $\hat{\boldsymbol{\Gamma}} = \begin{pmatrix} \hat{\gamma}_0 & \hat{\gamma}_1 \\ \hat{\gamma}_1 & \hat{\gamma}_0 \end{pmatrix}$, and $\hat{\boldsymbol{\gamma}} = (\hat{\gamma}_1, \hat{\gamma}_2)$ we have

$$\begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \end{bmatrix} = \begin{bmatrix} 1382.2 & 1114.4 \\ 1114.4 & 1382.2 \end{bmatrix}^{-1} \begin{bmatrix} 1114.4 \\ 591.72 \end{bmatrix} \approx \begin{bmatrix} 1.3175 \\ -0.6341 \end{bmatrix}$$

Additionally,

$$\hat{\sigma}_\varepsilon^2 = \hat{\gamma}_0 - \hat{\phi}_1\hat{\gamma}_1 - \hat{\phi}_2\hat{\gamma}_2 = 289.2139.$$

Exercise 4.2

Let $\{Y_t\}$ be a mean-zero stationary process with autocovariance γ. Show that the values $\phi_1, \ldots, \phi_k$ which minimise

$$\mathbb{E}\left[(Y_t - \phi_1 Y_{t-1} - \cdots - \phi_k Y_{t-k})^2\right]$$

satisfy the Yule-Walker equations

$$\gamma_j = \phi_1 \gamma_{j-1} + \cdots + \phi_k \gamma_{j-k}$$

for $j = 1, \ldots, k$.

Solution 4.2

Expanding terms, we compute:

$$\begin{aligned}
\mathbb{E}\left[\left(Y_t - \sum_{i=1}^{k}\phi_i Y_{t-i}\right)^2\right] &= \mathrm{Var}\left(Y_t - \sum_{i=1}^{k}\phi_i Y_{t-i}\right) + \mathbb{E}\left[Y_t - \sum_{i=1}^{k}\phi_i Y_{t-i}\right]^2 \\
&= \mathrm{Var}\left(Y_t - \sum_{i=1}^{k}\phi_i Y_{t-i}\right) + \left(1 - \sum_{i=1}^{k}\phi_i\right)^2 \mathbb{E}\left[Y_t\right]^2 && (Y_t \text{ stationary}) \\
&= \mathrm{Var}\left(Y_t - \sum_{i=1}^{k}\phi_i Y_{t-i}\right) && (Y_t \text{ mean-zero}) \\
&= \mathrm{Cov}\left(Y_t - \sum_{i=1}^{k}\phi_i Y_{t-i}, Y_t - \sum_{j=1}^{k}\phi_j Y_{t-j}\right) \\
&= \mathrm{Cov}\left(Y_t, Y_t\right) - 2\sum_{i=1}^{k}\phi_i \,\mathrm{Cov}\left(Y_t, Y_{t-i}\right) + \sum_{i=1}^{k}\sum_{j=1}^{k}\phi_i\phi_j \,\mathrm{Cov}\left(Y_{t-i}, Y_{t-j}\right) \\
&= \gamma_0 - 2\sum_{i=1}^{k}\phi_i\gamma_i + \sum_{i=1}^{k}\sum_{j=1}^{k}\phi_i\phi_j\gamma_{j-i}. && (Y_t \text{ stationary})
\end{aligned}$$

Summing up we end up with:

$$\mathbb{E}\left[\left(Y_t - \sum_{i=1}^{k}\phi_i Y_{t-i}\right)^2\right] = \gamma_0 - 2\sum_{i=1}^{k}\phi_i\gamma_i + \sum_{i=1}^{k}\sum_{j=1}^{k}\phi_i\phi_j\gamma_{i-j}.$$

Being a positive quadratic equation, we can only find the minimum. Hence, differentiating with respect to the i^{th} index and equating to zero we obtain:

$$-2\gamma_i + 2\sum_{j=1}^{k}\phi_j\gamma_{i-j}{=}0.$$

Rearranging

$$\gamma_i = \sum_{j=1}^{k}\phi_j\gamma_{i-j}.$$

Exercise 4.3

We assume that we have observations from a mean-zero MA(1). We observe $X_1 = 0$, $X_2 = 1$ and $X_3 = 0.5$. Estimate θ.

Solution 4.3

We recall for the MA(1) process: $X_t = \varepsilon_t - \theta_1\varepsilon_{t-1}$. We are given X_1, X_2, X_3. We choose $t = 0$ as the starting point in time.

We want the minimise the squared error terms; yet, the ε terms are unknown. Thus we try to recover them via $\varepsilon_t = X_t + \theta_1\varepsilon_{t-1}$ as follows.

Write $\varepsilon_0 = 0$ ($t = 0$ is the starting point), $\varepsilon_1 = X_1 + \theta_1\varepsilon_0 = 0$, $\varepsilon_2 = X_2 + \theta_1\varepsilon_1 = 1$, and finally, $\varepsilon_3 = X_3 + \theta_1\varepsilon_2 = 0.5 + \theta_1$.

The least squares consist of minimising

$$\sum_{t=1}^{3} \varepsilon_t^2 = 0 + 1 + (0.5 + \theta_1)^2,$$

which clearly achieves the minimum for $\theta = -0.5$, hence $\hat{\theta} = -0.5$.

Exercise 4.4

From a series of length 100, we have computed that the ACF at lag 1 is 0.8, at lag 2 is 0.5, and at lag 3 is 0.4. If we assume that an AR(3) model with zero mean is appropriate, obtain estimates of ϕ_1 and ϕ_2 and ϕ_3?

Solution 4.4

Using Yule-Walker, we have that for $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3)^\top$, and $\boldsymbol{\phi} = (\phi_1, \phi_2, \phi_3)^\top$ and

$$\boldsymbol{\Gamma} = \begin{pmatrix} \gamma_0 & \gamma_1 & \gamma_2 \\ \gamma_1 & \gamma_0 & \gamma_1 \\ \gamma_2 & \gamma_1 & \gamma_0 \end{pmatrix},$$

we may write the solution as $\hat{\boldsymbol{\phi}} = \hat{\boldsymbol{\Gamma}}^{-1} \hat{\boldsymbol{\gamma}}$.

As we are given the correlations, we take a new 3×3 diagonal matrix with the quotient of the variance, $1/\gamma_0$, on the diagonal, say V. Let $\bar{\boldsymbol{\Gamma}} = V\boldsymbol{\Gamma}$, $\boldsymbol{\rho} = V\boldsymbol{\gamma}$, and note that similar to the previous matrices, such matrices are filled with the autocorrelation ρ instead of the autocovariance.

Then we can rewrite $\boldsymbol{\gamma} = \boldsymbol{\Gamma}\boldsymbol{\phi}$ as $V\boldsymbol{\gamma} = V\boldsymbol{\Gamma}\boldsymbol{\phi}$, from which we see

$$\boldsymbol{\phi} = (V\boldsymbol{\Gamma})^{-1} V\boldsymbol{\gamma} = \bar{\boldsymbol{\Gamma}}^{-1} \boldsymbol{\rho}.$$

Finally, we can estimate $\hat{\boldsymbol{\phi}} = \hat{\bar{\boldsymbol{\Gamma}}}^{-1} \hat{\boldsymbol{\rho}}$.

Using the given details we obtain

$$\hat{\boldsymbol{\phi}} = (1.30909, -0.954545, 0.509091)^\top.$$

Exercise 4.5

Consider the Wolfer sunspot data. Compute the forwards least squares estimator for an AR(2) model. Do the same for the backwards estimator. Remember you need to remove the mean if it is significantly non-zero. *Hint: do not do this by hand!*

Solution 4.5

The following R code computes both estimators:

```
sunspot <- read.table("sunspot.txt", header = TRUE)

foward_ls_estimator <- function(x, p) {
    x <- x-mean(x)
    n <- length(x)
    Xf <- x[-1:-p]
    compute_x_col <- function(j) {
        x[(p + 1 - j):(n - j)]
    }
    F <- cbind(apply(array(1:p), MARGIN=1, FUN = compute_x_col))
    phi_F <- solve(t(F) %*% F) %*% t(F) %*% Xf
    sigma_F_squared <- sum((Xf - F %*% phi_F)^2) / (n - p - p)
    return(list(phi_F = phi_F, sigma_F_squared = sigma_F_squared))
}
print(foward_ls_estimator(sunspot$sunspot, 2))

backward_ls_estimator <- function(x, p) {
    x <- x-mean(x)
    n <- length(x)
    Xb <- x[1:(n-p)]
```

```r
    compute_x_col <- function(j) {
        x[(1 + j):(n - p + j)]
    }
    B <- cbind(apply(array(1:p), MARGIN=1, FUN = compute_x_col))
    phi_B <- solve(t(B) %*% B) %*% t(B) %*% Xb
    sigma_B_squared <- sum((Xb - B %*% phi_B)^2) / (n - p - p)
    return(list(phi_B = phi_B, sigma_B_squared = sigma_B_squared))
}
print(backward_ls_estimator(sunspot$sunspot, 2))
```

Running this outputs:

```
$phi_F
           [,1]
[1,]  1.4045705
[2,] -0.7113314

$sigma_F_squared
[1] 232.2614

$phi_B
           [,1]
[1,]  1.4061489
[2,] -0.7075649

$sigma_B_squared
[1] 231.0316
```

So we have

$$\begin{aligned}
\phi_F &\approx [1.4046, -0.7113]^\top, \\
\sigma_F &\approx 232.2614, \\
\phi_B &\approx [1.4061, -0.7076]^\top, \\
\sigma_B &\approx 231.0316.
\end{aligned}$$