Time Series: Homework 2

Martín Prado

September 5, 2024

Universidad de los Andes – Bogotá Colombia

Exercise 1.11.

If $\{X_t\}$ and $\{Y_t\}$ are uncorrelated stationary sequences, i.e. if X_s and Y_t are uncorrelated for every s and t, show that $\{X_t + Y_t\}$ is stationary with autocovariance function equal to the sum of the autocovariance functions of $\{X_t\}$ and $\{Y_t\}$.

Solution:

$$\mathbf{E}\,|X_t + Y_t|^2 \leq \mathbf{E}\,|X_t|^2 + \mathbf{E}\,|Y_t|^2 < \infty$$
$$\mathbf{E}\,(X_t + Y_t) = \mathbf{E}\,X_0 + \mathbf{E}\,Y_0$$

Define $X_t' = X_t - \mathbf{E}\,X_0$ and $Y_t' = Y_t - \mathbf{E}\,Y_0$

$$\begin{aligned}
\mathbf{cov}\,(X_t + Y_t, X_s + Y_s) &= \mathbf{E}\,[(X_t' + Y_t')(X_s' + Y_s')] \\
&= \mathbf{E}\,X_t'X_s' + \mathbf{E}\,X_t'Y_s' + \mathbf{E}\,X_s'Y_t' + \mathbf{E}\,Y_t'Y_s' \\
&= \mathbf{E}\,X_t'X_s' + \mathbf{E}\,Y_t'Y_s' \\
&= \gamma_X(t-s) + \gamma_Y(t-s).
\end{aligned}$$

Exercise 1.12

Which, if any, of the following functions defined on the integers is the autocovariance function of a stationary time series?

(b) $f(h) = (-1)^{|h|}$,

(d) $f(h) = 1 + \cos\frac{\pi h}{2} - \cos\frac{\pi h}{4}$,

(f) $f(h) = \begin{cases} 1 & \text{if } h = 0, \\ .6 & \text{if } h = \pm 1, \\ 0 & \text{otherwise.} \end{cases}$

Solution Item (b)

For every $n \in \mathbb{N}$, the matrix

$$M_n = [f(i-j)]_{i,j=1}^n = \begin{bmatrix} 1 & -1 & \cdots \\ -1 & 1 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix},$$

has eigenvalue 0 with multiplicity $n-1$ (the first column is repeated n times) and eigenvalue n with multiplicity 1:

$$M \begin{bmatrix} 1 \\ -1 \\ \vdots \end{bmatrix} = \begin{bmatrix} n \\ -n \\ \vdots \end{bmatrix}.$$

Therefore, since all eigenvalues are non-negative, the function is positive semi-definite, so it corresponds to an autocovariance function of an stationary process.

Solution Item (d)

With the following code I found numerically that the matrix $[f(i-j)]_{i,j=1}^n$ is not semidefinite positive when $n = 4$.

```
import numpy as np
cos = np.cos
sin = np.sin
pi = np.pi
n = 4
matrix = np.matrix([[
    1 + cos(pi*(i-j)/2) - cos(pi*(i-j)/4)
    for i in range(1,n+1)] for j in range(1,n+1)])
print(np.linalg.eigvals(matrix).round(10).astype(float))
```

This last line returns `array([-0.76536686,  2.76536686,  0.76536686,  1.23463314])`, which has a negative eigenvalue. The numerical error of the algorithm is low enough to consider the negative eigenvalue close enough to the real eigenvalue.

Solution Item (f)

Let M_n be a $n \times n$ tridiagonal (Toeplitz) matrix with the following form

$$M_n = \begin{bmatrix} a & c & & & \\ b & a & c & & \\ & \ddots & \ddots & \ddots & \\ & & b & a & c \\ & & & b & a \end{bmatrix}.$$

The eigenvalues of this matrix are

$$\lambda_k = a + 2\sqrt{bc}\cos\frac{k\pi}{n+1}, \quad k = 1, \dots, n.$$

For our case, $a = f(0) = 1$, $b = c = f(\pm 1) = 0.6$,

$$\lambda_k = 1 + 1.2\cos\frac{k\pi}{n+1},$$

so $\lambda_n < 1$ for a big enough n. In fact, by taking $n = 5$, we have that the smallest eigenvalue is approximately $\lambda_n = -0.0392$ (this was done numerically as the previous exercise)

Solution Item (c)

Note: I solved by mistake this exercise, so I'm leaving the solution here.

The book suggested to find a time series for which $\gamma(h) = 1 + \cos\frac{\pi h}{2} + \cos\frac{\pi h}{4}$. My idea was to use what I proved in exercise 1.7.(c) in the previous homework:

$$\begin{aligned} A_t &= aZ_1, \\ B_t &= bZ_2\cos\frac{\pi t}{2} + bZ_3\sin\frac{\pi t}{2}, \\ C_t &= cZ_4\cos\frac{\pi t}{4} + cZ_5\sin\frac{\pi t}{4}, \end{aligned}$$

for independent $Z_t \sim N(0,1)$. Therefore,

$$\gamma_A(h) = a^2, \quad \gamma_B(h) = b^2\cos\frac{\pi h}{2}, \quad \gamma_C(h) = c^2\cos\frac{\pi h}{4}$$

Then, since A_t, B_t and C_t are uncorrelated, we can use the previous exercise to assert that if

$$X_t = A_t + B_t + C_t,$$

then the autocovariance function is

$$\begin{aligned} \gamma_X(h) &= \gamma_A(h) + \gamma_B(h) + \gamma_C(h) \\ &= 1 + \cos\frac{\pi h}{2} + \cos\frac{\pi h}{4}. \end{aligned}$$

By choosing $a = b = c = 1$ we'd prove that $\gamma_X = f$ is a semidefinite positive function.

Exercise 2.3.

Show that if $\{X_t, t = 0, \pm 1, \dots\}$ is stationary and $|\theta| < 1$ then for each n, $\sum_{j=1}^{m} \theta^j X_{n+1-j}$ converges in mean square as $m \to \infty$.

Solution: Assume without restriction that $\mu = \mathbf{E}\, X_t = 0$. Otherwise, define $X_t' = X_t - \mu$, and note that the following series

$$Y_m^{(n)} = \sum_{j=1}^{m} \theta^j X_{n+1-j} = \sum_{j=1}^{m} \theta^j X_{n+1-j}' + \mu \sum_{j=1}^{m} \theta^j,$$

converges if and only if $\sum_{j=1}^{m} \theta^j X_{n+1-j}'$ converges.

I don't know where does the series converges for a fixed n. However, I can prove that it converges to something by proving that the sequence $\{Y_m^{(n)}\}_{m\in\mathbb{N}}$ is Cauchy (in the mean square metric $\|\cdot\|$).

Assume without restriction that $M > m \in \mathbb{N}$. Then,

$$\begin{aligned}
\|Y_M^{(n)} - Y_{m-1}^{(n)}\| &= \mathbf{E}\left[\left(\sum_{j=m}^{M} \theta^j X_{n+1-j}\right)^2\right] \\
&= \mathbf{E}\left(\sum_{i=m}^{M}\sum_{j=m}^{M} \theta^{i+j} X_{n+1-i} X_{n+1-j}\right) \\
&\leq \sum_{i=m}^{M}\sum_{j=m}^{M} \theta^{i+j} \left|\mathbf{E}\left[X_{n+1-i} X_{n+1-j}\right]\right|.
\end{aligned}$$

Then, note that by Cauchy-Schwarz inequality

$$\left|\mathbf{E}\left[X_{n+1-i} X_{n+1-j}\right]\right| \leq \sqrt{\mathbf{Var}\left(X_{n+1-i}\right)} \cdot \sqrt{\mathbf{Var}\left(X_{n+1-j}\right)} = \sigma^2.$$

Therefore,

$$\begin{aligned}
\|Y_M^{(n)} - Y_{m-1}^{(n)}\| &\leq \sigma^2 \sum_{i=m}^{M}\sum_{j=m}^{M} \theta^{i+j} \\
&= \sigma^2 \theta^{2m} \sum_{i=0}^{M-m}\sum_{j=0}^{M-m} \theta^{i+j} \\
&\leq \sigma^2 \theta^{2m} \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \theta^{i+j}.
\end{aligned}$$

Finally, using the integral test, for $\theta \in (0,1)$

$$\begin{aligned}
\int_0^\infty \int_0^\infty \theta^{x+y} dx dy &= \int_0^\infty \theta^y \int_0^\infty \theta^x dx dy \\
&=^{(*)} \int_0^\infty \theta^y \frac{-1}{\ln(\theta)} dy \\
&= \frac{1}{\ln^2(\theta)} < \infty
\end{aligned}$$

$$\implies \exists K > 0,\ \|Y_M^{(n)} - Y_{m-1}^{(n)}\| \leq K\theta^{2m}$$

Therefore, as m, M go to infinity, $\|Y_M^{(n)} - Y_{m-1}^{(n)}\| \to 0$. The last detail $(*)$ is this limit:

$$\int_0^\infty \theta^x dx =^{(*)} \lim_{x\to\infty} \frac{\theta^x}{\ln(\theta)} - \lim_{x\to 0} \frac{\theta^x}{\ln(\theta)}$$

The first limit goes to 0 when $0 < \theta < 1$.

Exercise 2.5.

If $\mathscr{M}$ is a closed subspace of the Hilbert space $\mathscr{H}$ and $x \in \mathscr{H}$, prove that

$$\min_{y\in\mathscr{M}} \|x - y\| = \max\{|\langle x, z\rangle| \ :\ z \in \mathscr{M}^\perp, \|z\| = 1\}.$$

Solution:

By the projection theorem, we know that there exists a linear function $P : \mathscr{H} \to \mathscr{H}$ such that

$$\min_{y\in\mathscr{M}} \|x - y\| = \|x - Px\|.$$

Now, define the linear functional $T : \mathscr{M}^\perp \to \mathscr{M}^\perp$, $T(z) = \langle x, z\rangle$ and note that

$$\|T\|_{op} = \max\{|\langle x, z\rangle| \ :\ z \in \mathscr{M}^\perp, \|z\| = 1\}.$$

Again, by some corollary of the projection theorem we have $x = Px + (I - P)x$, with $Px \in \mathscr{M}$ and $(I - P)x \in \mathscr{M}^\perp$. So it follows that for $z \in \mathscr{M}^\perp$,

$$\begin{aligned}
T(z) &= \langle Px + (I-P)x, z\rangle \\
&= \cancelto{0}{\langle Px, z\rangle} + \langle (I-P)x, z\rangle \\
&= \langle x - Px, z\rangle .
\end{aligned}$$

If we take $z = (x - Px)/\|x - Px\| \in \mathscr{M}^T$ we have that

$$\begin{aligned} |T(z)| &= \frac{1}{\|x - Px\|} |\langle x - Px, x - Px \rangle| \\ &= \frac{\|x - Px\|^2}{\|x - Px\|} = \|x - Px\|. \\ &\implies \|x - Px\| \leq \|T\|_{op} \end{aligned}$$

On the other hand, by Cauchy-Schwarz inequality,

$$\begin{aligned} |T(z)| &\leq \|x - Px\| \|z\| \\ &\implies \|T\|_{op} \leq \|x - Px\|. \end{aligned}$$

Therefore,

$$\min_{y \in \mathscr{M}} \|x - y\| = \|x - Px\| = \|T\|_{op} = \max\{|\langle x, z \rangle| \ : \ z \in \mathscr{M}^\perp, \|z\| = 1\}$$

Exercise 2.6.

Verify the calculations of ϕ_1 and ϕ_2 in Example 2.3.4. Also check that $X_3 = (2\cos\omega)X_2 - X_1$.

Solution:

We use Python's package sympy for the symbolic calculations of the following verifications of ϕ_1 and ϕ_2. In the example they say that ϕ_n is the solution for the following linear system

$$\Gamma_n \phi_n = \gamma_n, \qquad \Gamma_n = [\gamma(i-j)]_{i,j=1}^n, \quad \gamma_n = (\gamma(1), \dots, \gamma(n))^T.$$

First, we import all the symbolic variables

```
import sympy as sp
from sympy import cos, sin # cosine and sine functions
from sympy.abc import t, h, A, B # variables t, h, A, B
omega, sigma = sp.symbols("omega, sigma") # variables omega and sigma
```

We define $X_t = A\cos(\omega t) + B\sin(\omega t)$ with the following line

```
X = lambda t: A*cos(omega*t) + B*sin(omega*t)
```

Then, the autocovariance function $\gamma(h) = \sigma^2 \cos(\omega h)$

```
gamma = lambda h: sigma**2 * cos(omega*h)
```

In the following two lines, we define Γ_n and γ_n

```
Gamma_n = lambda n: sp.Matrix([[gamma(i-j) for i in range(1,n+1)] for j in
↪  range(1,n+1)])
gamma_n = lambda n: sp.Matrix([gamma(i) for i in range(1,n+1)])
```

We define ϕ_1 and ϕ_2, as follows

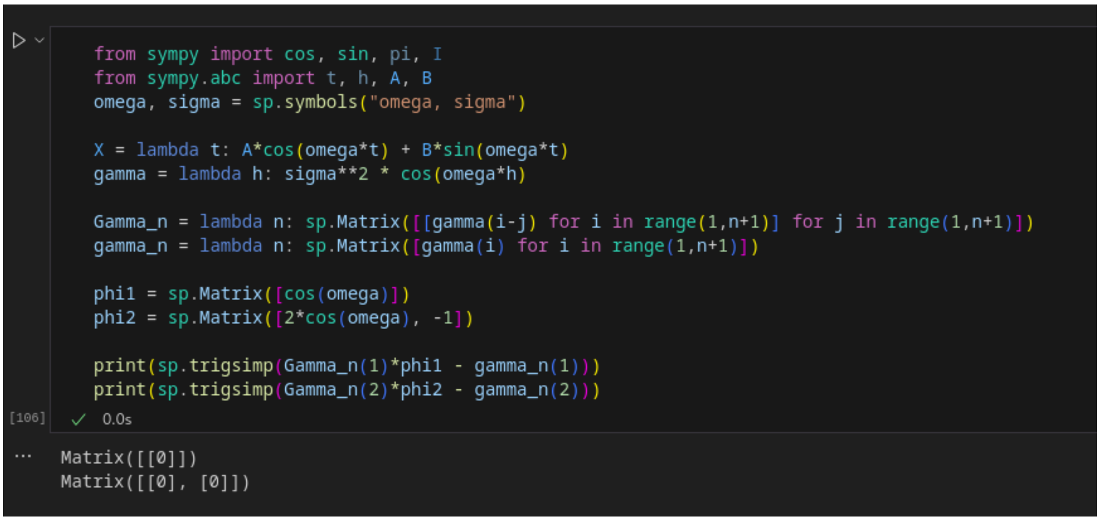
```
phi1 = sp.Matrix([cos(omega)])
phi2 = sp.Matrix([2*cos(omega), -1])
```

Finally, we verify $\Gamma_1\phi_1 - \gamma_1 = [0]$ and $\Gamma_2\phi_2 - \gamma_2 = [0, 0]$ with the following lines

```
print(sp.trigsimp(Gamma_n(1)*phi1 - gamma_n(1)))
print(sp.trigsimp(Gamma_n(2)*phi2 - gamma_n(2)))
```

The function `sp.trigsimp` simplifies the trigonometric expression.

The output of the previous lines are: `Matrix([[0]])` and `Matrix([[0], [0]])` respectively.



```
from sympy import cos, sin, pi, I
from sympy.abc import t, h, A, B
omega, sigma = sp.symbols("omega, sigma")

X = lambda t: A*cos(omega*t) + B*sin(omega*t)
gamma = lambda h: sigma**2 * cos(omega*h)

Gamma_n = lambda n: sp.Matrix([[gamma(i-j) for i in range(1,n+1)] for j in range(1,n+1)])
gamma_n = lambda n: sp.Matrix([gamma(i) for i in range(1,n+1)])

phi1 = sp.Matrix([cos(omega)])
phi2 = sp.Matrix([2*cos(omega), -1])

print(sp.trigsimp(Gamma_n(1)*phi1 - gamma_n(1)))
print(sp.trigsimp(Gamma_n(2)*phi2 - gamma_n(2)))
[106]  ✓ 0.0s

...  Matrix([[0]])
     Matrix([[0], [0]])
```

Therefore, since the calculations made before are symbolic, ϕ_1 and ϕ_2 are the exact solutions to their respective linear systems. Finally, we verify that $X_3 - 2\cos(w)X_2 + X_1 = 0$ with the following line

```
from sympy.simplify.fu import TR8
expr = X(3) - 2*cos(omega)*X(2) + X(1)
print(TR8(expr))
```

What the function `TR8` does according to the documentation in this website is to *"expand products of sin-cos to sums"*. The output of the previous lines is `0` as we intended.

```
from sympy.simplify.fu import TR8
expr = X(3) - 2*cos(omega)*X(2) + X(1)
print(TR8(expr))
```

```
[127] ✓ 0.0s
... 0
```

This concludes the exercise.