UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques
i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

ARIMA processes for EEG modeling

Autor: Gabriel Vayá Abad

Director: Dr. Josep Vives i Dr. Ignasi Cos
Realitzat a: Departament de
Matemàtiques i Informàtica

Barcelona, 11 de juny de 2023

Abstract

The main goal of this work is to find a suitable model for EEGs. This model has to be appropriate for the performing of classification of subjects based on the level of social pressure, and has to preserve time variability.

2020 Mathematics Subject Classification. 37M10, 62H12, 62M10, 62P10, 92-10, 92C55

Agraïments

En primer lloc li vull agraïr el suport i comprensió durant els moments de més estrés a la meva família, als meus pares i al meu germà, així com a la meva parella, que han estat al meu costat tot el procés. En segon lloc, li vull agraïr als meus amics Jon i al Miki per l'ajuda brindada en aquest treball, a més del Guillem. També vull mencionar a la Clàudia, amb qui he pogut compartir maldecaps i preocupacions, i ha donat una gran ajuda. Finalment agraïr als dos tutors d'aquest treball; l'Ignasi, que m'ha acompanyat en cada passa del camí i m'ha motivat per extreure el màxim del treball, i el Josep, qui ha aportat un coneixement i expèriencia valuosíssimes.

Moltes gràcies a tots.

Contents

Introduction

1 Preliminaries **1**
- 1.1 Initial concepts . . . 1
- 1.2 Stationary time series . . . 1
- 1.3 AR processes . . . 5
- 1.4 Examples . . . 8

2 ARIMA models **11**
- 2.1 MA processes . . . 11
- 2.2 ARMA(p,q) processes . . . 13
- 2.3 Time series differencing . . . 14
- 2.4 ARIMA(p,d,q) processes . . . 15

3 Model Estimation **17**
- 3.1 AR(1) parameter estimation: Maximum Likelihood Estimation . . . 17
- 3.2 AR(p) Maximum Likelihood Estimation . . . 20
- 3.3 Model comparison: Akaike Information Criterion . . . 22
- 3.4 Source selection: Independent Component Analysis . . . 23

4 EEG fit **25**
- 4.1 Experiment description . . . 25
- 4.2 Preliminary observations . . . 26
- 4.3 Model choice . . . 29
- 4.4 Model evaluation . . . 38
- 4.5 Code . . . 41

5 Discussion **45**

Bibliography **49**

Introduction

Time series analysis is a field that has been around since the 1970's, and has been used to tackle numerous problems, especially in fields like Finances and Econometrics. However, it has not been since recently that has entered the medical domain. In this case, we had a data-set of electroencephalograms (EEG) taken from an experiment performed by Dr. Ignasi Cos and his colleagues, which worked on the effect of social pressure on motor behavior and decision-making. The underlying objective is to find the particularities of the EEG readings in every defined state of social pressure: no social pressure, reaffirming social pressure and discouraging social pressure.

In our work, we explored the modeling of the EEG gathered in the state of no social pressure using time series analysis, hoping to find a process able to capture the time variability of the readings, not yet tried in this particular experiment. Nevertheless, we had a delusive advantage, which is not recurrent in traditional time series analysis: we had more than one sample of the same process, in particular, 432 of them. Hence, the challenge was to, using small twists on known methods, find a model that explains the underlying process, observed 432 times.

For that, we read about time series and deepened into ARIMA models, especially into auto-regressive processes. Since nowadays we count on the support of computers, we explored the classical parameter estimating techniques used by today's software, to understand as detailed as possible the process of fit of our data. Finally, we researched model information criteria, thus we intended to use an exhaustive process of model comparison to find the most suitable one.

Having reached a vast understanding of time series analysis and the data itself, and after a lot of disappointing trials, we found a way to fit the totality of the observations into a single process, and later demonstrated that this process was suitable for modeling all data in this particular state of social pressure. With that, we proved that time series analysis can be used, mixed with some modern data science techniques, in a much more extensive repertoire of situations as it is commonly believed, and gives unique tools to face today's challenges.

Chapter 1

Preliminaries

1.1 Initial concepts

Definition 1.1. A **time series** is a set of observations $\{x_t\}$ recorded at specific times $t = 1, \ldots, T$.

The difference between *time series* and *model* is worth noting.

Definition 1.2. A **time series model** for an observed data $\{x_t\}$ is a specification of the joint distributions of a sequence of random variables $\{X_t\}$ of which $\{x_t\}$ is postulated to be a realization.

Notation: Once specified the difference, we will use *time series* for both time series and models.

1.2 Stationary time series

In order to give the definition for stationary time series, a few preliminary definitions have to be stated.

Definition 1.3. Let $\{X_t\}$ be a time series with $E(X_t)^2 < \infty$. The **mean function** of $\{X_t\}$ is

$$\mu_X(t) = E(X_t).$$

The **covariance function** of $\{X_t\}$ is

$$\gamma_X(r,s) = Cov(X_r, X_s) = E[(X_r - \mu_X(r))(X_s - \mu_X(s))].$$

for all times r and s.

Now we can define what we mean by *stationary time series*.

Definition 1.4. We say that $\{X_t\}$ is **(weakly) stationary** if

(a) $\mu_X(t)$ is independent of t.

(b) $\gamma_X(t+h,t)$ is independent of t for each h.

Definition 1.5. We say that $\{X_t\}$ is **(strictly) stationary** under the condition that $(X_1,...,X_n)$ has the same distribution as $(X_{1+h},...,X_{n+h})$.

Proposition 1.6. *Let $\{X_t\}$ be a strictly stationary time series such that $E(X_t^2) < \infty$. Then $\{X_t\}$ is weakly stationary.*

Proof. First we observe that $E(X_t^2) < \infty$ implies that $E(X_t)$ is constant, it doesn't depend on t, we have (a). Now for the existence of the covariance function, we need Cauchy-Schwarz inequality. Indeed we have

$$Cov(X_r, X_s) = E(X_r X_s) - E(X_r)E(X_s)$$
$$\leq \sqrt{E(X_r^2)E(X_s^2)} - E(X_r)E(X_s) < \infty.$$

Now, using the definition of strictly stationary we have that the variables $(X_1,...,X_n)$ and $(X_{1+h},...,X_{n+h})$ have the same distribution. Then we have

$$Cov(X_1, X_{1+h}) = Cov(X_t, X_{t+h})$$

which also does not depend on t. □

Remark 1: From now on, when we use the term *stationary* we will refer to *weakly stationary*.

Remark 2: As for the *covariance function*, whenever we are referring to a stationary time series, we can look at it as a one variable function defined by

$$\gamma_X(h) := \gamma_X(h,0) = \gamma_X(t+h,t)$$

where we used condition (b) of the definition in the second equality.

Definition 1.7. We refer as **lag h** to the difference between times t and $t+h$.

Definition 1.8. We refer to the previous function as the **autocovariance function** (ACVF) at *lag h*. We can express it as:

$$\gamma_X(h) = Cov(X_{t+h}, X_t).$$

The **autocorrelation function** (ACF) at *lag h* is

$$\rho_X(h) = \frac{\gamma_X(h)}{\gamma_X(0)} = Cor(X_{t+h}, X_t).$$

This are functions for *models*. However, in our case we are not working directly with models, but with *observations at a time t*. For that matter, we will now see how to assess the degree of dependence in our observations.

Definition 1.9. Let $x_1, \ldots, x_n$ be observations of a time series. The **sample mean** of $x_1, \ldots, x_n$ is

$$\overline{x} = \frac{1}{n} \sum_{t=1}^{n} x_t.$$

The **sample autocovariance function** at *lag h* is

$$\widehat{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-h} (x_{t+h} - \overline{x})(x_t - \overline{x}) \qquad -n < h < n.$$

The **sample autocorrelation function** at *lag h* is

$$\widehat{\rho}(h) = \frac{\widehat{\gamma}(h)}{\widehat{\gamma}(0)} \qquad -n < h < n.$$

However, when modeling observed data, whenever we say *autocorrelation function (ACF)* we are referring to the *sample autocorrelation function*

Now for finding the best model for observed data, there exist many indicators that can be evaluated. Apart from the ACF, we can look at the *Partial Autocorrelation Function*.

Definition 1.10. Let $\{X_t\}$ be a stationary time series. We define the **partial autocorrelation function** (PACF) at *lag h* as the following:

$$\alpha(h) = \rho(X_1 - P(X_1|X_2, \ldots, X_h), X_{h+1}|P(X_{h+1}|X_2, \ldots, X_h))$$

where $P(X_1|X_2, \ldots, X_h)$ is the linear projection of X_1 in the subspace generated by $X_2, \ldots, X_h$.

The information that the Partial Autocorrelation Function gathers is about the correlation between two variables, having taken into account the values of other set of variables. For instances, in the model:

$$X_t = \beta_0 + \beta_1 X_{t-1}$$

β_1 can be interpreted as the linear relationship between X_t and X_{t-1}. Now let's have a look at this other model:

$$X_t = \beta_0 + \beta_1 X_{t-1} + \beta_2 X_{t-2}.$$

In this case, β_2 is the linear dependency between X_t and X_{t-2} having taken into account the linear correlation between X_t and X_{t-1}.

Now that we have assessed the dependence in the observations of one time series, we can now give some notions on dependence between multiple time series, which is a crucial factor in multivariate time series analysis.

Definition 1.11. Let $\{X_t\}$ and $\{Y_t\}$ be two time series with finite variance. The **Cross-covariance function** at *lag h* between the two series is given by:

$$\gamma_{XY}(s,t) = Cov(X_s, Y_t) = E((X_s - \mu_{xs})(Y_t - \mu_{yt})).$$

Now we can give the notion for two time series presenting stationarity features together, which as we will now see, it is similar to the condition for one time series.

Definition 1.12. Let $\{X_t\}$ and $\{Y_t\}$ be two time series. We say that they are **jointly stationary** if they are both stationary, and the cross-covariance function can be written as a one variable function as such:

$$\gamma_{XY}(h) = Cov(X_{t+h}, Y_t) = E((X_{t+h} - \mu_x)(Y_t - \mu_y)) \quad \forall t.$$

Definition 1.13. The **Cross-correlation function** (CCF) of two jointly stationary time series is:

$$\rho_{XY}(h) = \frac{\gamma_{XY}(h)}{\sqrt{\gamma_X(0)\gamma_Y(0)}}.$$

In a similar fashion as with the autocovariance and autocorrelation functions, for computing the codependece between two sets of observations instead of two models.

Definition 1.14. The **sample autocovariance function** of two sets of observations $\{x_t\}$ and $\{y_t\}$ is given by:

$$\widehat{\gamma}_{xy}(h) = \frac{1}{n}\sum_{t=1}^{n-h}(x_{t+h} - \overline{x})(y_t - \overline{y}) \qquad -n < h < n.$$

The **sample autocorrelation function** is:

$$\widehat{\rho}_{xy}(h) = \frac{\widehat{\gamma}_{xy}(h)}{\widehat{\gamma}_x(0)\widehat{\gamma}_y(0)} \qquad -n < h < n.$$

The (sample) Cross-correlation function is often examined graphically as a function of *lag h* to search for lagging relations in data. To illustrate the point:

Example 1.15. Let's assume that we have to sets of data x_t and y_t that happen to be outputs of the following model:

$$Y_t = AX_{t-l} + z_t$$

where X_t and Y_t are *jointly stationary* time series, and z_t is random noise uncorrelated with X_t. Then the cross-covariance function would look like so:

$$\gamma_{XY}(h) = Cov(Y_{t+h}, X_t) = Cov(AX_{t+h-l} + z_{t+h}, X_t) = Cov(AX_{t+h-l}, X_t)$$

$$= A\gamma_x(h-l).$$

At the same time it is clear that:

$$\gamma_Y(h) = Cov(Y_{t+h}, Y_t) = Cov(AX_{t+h-l} + z_{t+h}, AX_{t-l} + z_t)$$

$$= Cov(AX_{t+h-l}, AX_{t-l}) = A^2\gamma_X(h)$$

where we used that X_t and z_t are uncorrelated. Now we can compute what would be the cross-correlation function:

$$\rho_{XY}(h) = \frac{\gamma_{XY}(h)}{\sqrt{\gamma_X(0)\gamma_Y(0)}} = \frac{A\gamma_X(h-l)}{\sqrt{\gamma_X(0)A^2\gamma_X(0)}} = \frac{A\gamma_X(h-l)}{A\gamma_X(0)}$$

$$= \rho_X(h-l).$$

This way, if we want to check if two sets of observations are correlated, we can compute the sample cross-correlation function, and if it looks like the correlation function of the first set x_t with a peak on the positive side, then x_t leads y_t, and if the peak is on the negative side, y_t leads x_t.

1.3 AR processes

Now we will give some inside on Autoregressive Models (AR), which represent a major role on Time Series Analysis, not only because of their simplicity, but also for a making a very sensible and logical assumption, which is that the observation at time t is related in some way to the previous observations at times $t-1, t-2, \ldots, t-p$.

First of all we will present the notation to refer to these previous observations $t-h$.

Definition 1.16. We define the **backwards shift operator** B as

$$BY_j := Y_{j-1}.$$

We can also talk about the powers of the backwards shift operator B. Being k an integer, the kth power of B would be

$$B^k(Y_j) = Y_{j-k}.$$

This is specially useful in order to simplify the notation around *auto-regressive* and *differentiating* models as well.

Definition 1.17. Let $\{X_t\}$ be a time series. $\{X_t\}$ is an **auto-regressive process of order p** (AR(p)) if it is a stationary process which complies with the expression:

$$X_t = \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} + Z_t$$

where Z_t is a centered (0 mean) Gaussian white noise with variance $\sigma^2 > 0$, so let us write it:

$$Z \sim GWN(0, \sigma^2).$$

Using the notation for the backwards shift operator we can define a polynomial of degree p: $\Phi(x) = 1 - \phi_1 z - ... - \phi_p z^p$ with $\phi_i \in \mathbb{R}$. Now we can rewrite an *AR(p)* as following

$$\Phi(B)X_t = Z_t.$$

Definition 1.18. Let $\Phi(x)$ be a polynomial of degree d with $x \in \mathbb{C}$. We say that $\Phi(x)$ is invertible if it exists a square integrable series $\sum_{i=0}^{\infty} \psi_i x^i$ such that

$$\Phi(x) \sum_{i=0}^{\infty} \psi_i x^i = 1.$$

Now if we express the model in the proper manner

$$X_t = \Phi^{-1}(B)Z_t$$

we see that a necessary condition for a degree p polynomial to generate an AR(p) model, is for it to be invertible. A sufficient condition for this is given in the following result.

Theorem 1.19. *Let $\Phi(B)$ be a lag polynomial of degree p, and let $\Phi(z)$ be its associated polynomial with $z \in \mathbb{C}$. Then it is invertible if and only if all its roots $\{z : \Phi(z) = 0\} \subseteq \{z : |z| > 1\}$.*

Proof. To start with, we need to define the space in which time series live, stationary time series in particular. Let's consider the space **X** of sequences of random variables $\{X_t\}$ for $t \in \mathbb{Z}$ defined in some probability space $(\Omega, \mathcal{F}, P)$, that has the property that $\sup_t E|X_t| < \infty$. Then we can see that **X** is a normed space with the norm $||X||_\infty = \sup_t E|X_t|$. We can see that any stationary time series as previously defined belongs to this space **X**. Now let us define what a Banach space is.

Definition 1.20. *We say that $(X, ||\cdot||)$ is a **Banach space** if is a complete normed space. Equivalently, it is a normed space where every Cauchy sequence $\{X_n\}$,*

$$\lim_{n\leftarrow\infty} X_n = x$$

with $x \in (X, ||\cdot||)$.

With the following lemma, we ensure that **X** is in fact a Banach space.

Lemma 1.21. *The space* **X** *with the norm previously defined is a Banach space.*

Next, we will prove the invertibility of order one lag polynomials. For that, the following result will be used later on, which is proved in [8] on page 111.

Proposition 1.22. *Let* **B** *be a Banach space, and let T be an operator $T : \mathbf{B} \to \mathbf{B}$. If it is true that $||I - T|| < 1$ (where I is the identity), then T is invertible with inverse*

$$T^{-1} = \sum_{k=0}^{\infty} (I - T)^k.$$

Now, we attempt to determine what would be the *lag operator norm*. As the lag operator B is a linear operator in a normed space, we have that $||B|| = \sup_{||X||\leq 1} ||BX||$, and with the fact that $\sup_t E|X_t| = \sup_t E|X_{t-1}|$, we can write

$$||B|| = \sup_{||X||\leq 1} ||BX|| = \sup_{\{X_t\}\in\mathbf{X}:\sup_t ||X||\leq 1} (\sup_t E|X_{t-1}|) = 1.$$

Now that we have that $||B|| = 1$ (and conversely $||B^{-1}|| = 1$), we can prove the condition of invertibility for first order lag polynomials. Consider the polynomial $\phi(B) = I - \phi_1 B$. We have

$$||I - \phi(B)|| = ||\phi_1 B|| = |\phi_1|||B|| = |\phi_1|,$$

where we used the previous equality. Now we can apply proposition 1.22 and we get that $\phi(B)$ is invertible if only if $|\phi_1| < 1$ with inverse:

$$\phi^{-1}(B) = \sum_{k=0}^{\infty} (I - \phi_1)^k.$$

Before finishing the proof, we are going to make use of the following result, proved in [4] in page 5.

Lemma 1.23. *Let* **B** *be a Banach space and suppose that the operators T_i for $i \in (1, \ldots, p)$ commute. Let $T = T_1 T_2 \cdots T_p$. Then T is invertible if only if all of T_i are invertible.*

Next, we can now consider the order p lag polynomial:

$$\Phi(B) = I - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p,$$

which we can decompose in the following manner

$$\Phi(B) = (I - \frac{1}{\rho_1}B)(I - \frac{1}{\rho_2}B)\cdots(I - \frac{1}{\rho_p}B),$$

where $\{\rho_i : i = 1, \ldots, p\}$ are the roots of the equivalent complex polynomial $\Phi(z)$ for $z \in \mathbb{C}$. Having written that, we can consider the operator $\phi_i(B) = I - \frac{1}{\rho_i}B$ inside of the space $\mathbf{X}$. By proposition 1.22 this operator is invertible for $|\rho| > 1$ and has inverse

$$\phi_i^{-1}(B) = \sum_{k=0}^{\infty} (\frac{1}{\rho_i})^k B^k.$$

Finally, because the operators $\phi_i(B)$ commute, therefore, using lemma 1.23 we can say that the lag polynomial $\Phi(B)$ is indeed invertible as we wanted. □

1.4 Examples

Example 1.24. The AR(1) model. Let's consider the following model:

$$X_t = \phi X_{t-1} + Z_t \qquad \phi \in (-1, 1).$$

What we want to see is that the model is a well defined second order stationary process. Notice that we can write X_{t-1} as:

$$X_{t-1} = \phi X_{t-2} + Z_t$$

which, if we iterate, gives us a recursive expression for the AR(1) model:

$$X_t = \phi X_{t-1} + Z_t = \phi(\phi X_{t-2} + Z_t) + Z_t$$

$$= \phi^2 X_{t-2} + \phi Z_{t-1} + Z_t$$

$$= \ldots$$

$$= \sum_{i=0}^{k} \phi^i Z_{t-i} + \phi^{k+1} X_{t-k-1}.$$

With this expression we have written the model as a two-part. As far as the noise part is concerned, we can see that while $\{Z_i\}$ for $i \in I$ are all centered, pairwise uncorrelated random variables, $\phi^i Z_{t-i}$ are as well, and we now that the expression

$\sum_{i=0}^{k} \phi^i Z_{t-i}$ doesn't diverge if only if the sum of it's variance doesn't as well, for which we write:

$$\sum_{i=0}^{k} Var(\phi^i Z_{t-i}) = \sum_{i=0}^{k} \phi^{2i}\sigma^2 \leq \frac{\sigma^2}{1-\phi^2} < \infty$$

as $k \uparrow \infty$ given that $|\phi| < 1$ which has been stated as a hypothesis. On the other hand, we have that X_{t-k-1} is a stationary process by definition, which means that $E(X_{t-k-1})$ is a constant expression non-dependent of t. Now we can write:

$$E|X_t - \sum_{i=0}^{k} \phi^{2i}\sigma^2|^2 = \phi^{2k+2}E(X_{t-k-1}^2)$$

and we can see that, using one more time the fact that $|\phi| < 1$, this expression converges to 0 as $k \uparrow \infty$, which means that we can state that $E(X_t)$ is non-dependent of t. Let us compute the auto-covariance function of our AR(1) model:

$$\begin{aligned}
\gamma(t, t+h) &= Cov(Y_t, Y_{t+h}) \\
&= Cov(\sum_{i=0}^{\infty} \phi^i Z_{t-i}, \sum_{j=0}^{\infty} \phi^j Z_{t+h-j}) \\
&= \sum_{i,j=0}^{\infty} \phi^i \phi^j Cov(Z_{t-i}, Z_{t+h-j}) \\
&= \sum_{i,j=0}^{\infty} \phi^{i+j}\sigma^2 \mathbb{1}_{\{i=j-h\}} \\
&= \sum_{i=0}^{\infty} \phi^{2i+h}\sigma^2 = \frac{\sigma^2\phi^h}{1-\phi^2}
\end{aligned}$$

which we can see that doesn't depend on t. With all of these we can say that the AR(1) model is indeed a well defined, second order stationary process as we wanted.

Finally, we can compute the auto-correlation function which we will use further along this work:

$$\begin{aligned}
\rho_X(h) &= \frac{\gamma_X(h)}{\gamma_X(0)} \\
&= \frac{\frac{\sigma^2\phi^h}{1-\phi^2}}{\frac{\sigma^2\phi^0}{1-\phi^2}} = \phi^h,
\end{aligned}$$

and state that whenever we want to see if a set of observation can be adjusted into an AR(1), we will look for the ACF function to be a decreasing function which converges to 0 as the lags tend to ∞.

Example 1.25. The AR(p) model. As stated previously, the AR(p) process can be written as follows:

$$\Phi(B)X_t = Z_t.$$

First of all, as we explained previously, it is necessary for the polynomial $\Phi(B)$ to be invertible in order for it to define an AR(p) process. Let's suppose that $\Phi(B)$ is in fact invertible, and following Definition 1.18, we will write

$$\Phi(B)^{-1} = \sum_{i=0}^{\infty} \psi_i B^i,$$

and express the AR(p) model as such

$$X_t = \sum_{i=0}^{\infty} \psi_i B^i Z_t.$$

Looking at this expression, and given the invertibility condition in Theorem 1.19, it is clear that the AR(p) model is stationary and well defined. Now we can give it's auto-covariance function, which is immediately given by

$$\gamma(h) = Cov(X_t, X_{t+h}) = \sigma^2 \sum_{i=0}^{\infty} \psi_i \psi_{i+h}.$$

Finally, for the auto-correlation function we can write:

$$Cov(X_t, X_{t+h} = Cov(X_t, \phi_1 X_{t+h-1} + \cdots + \phi_p X_{t+h-p} + Z_{t+h})$$

$$= \phi_1 \gamma(h-1) + \phi_2 \gamma(h-2) + \cdots + \phi_p \gamma(h-p) = \gamma(h),$$

and if we divide the whole expression by $\gamma(0)$, we get:

$$\rho(h) = \phi_1 \rho(h-1) + \phi_2 \rho(h-2) + \cdots + \phi_p \rho(h-p) \qquad h > 0.$$

With this expression in mind, and using that $\rho(0) = 1$ and the symmetry of the auto-correlation function we get the following system of equations

$$\rho(h) = \phi_1 \rho(h-1) + \phi_2 \rho(h-2) + \cdots + \phi_p \rho(h-p)$$

$$\rho(0) = 1$$

$$\rho(-h) = \rho(h)$$

which is solvable, and its solution is the expression for the auto-correlation function of the AR(p) process.

Chapter 2

ARIMA models

Most of the data in time series study is non-stationary, which means that differencing is needed. If that was the case, ARIMA models would be used to fit our data. However, in our case, the data generated by EEGs is supposed to be stationary, which means that we can use ARMA models instead. However, before talking about ARMA models, we should give some notions about moving-average models.

2.1 MA processes

Whenever we construct a time series, we can think of it as a filtering of white noise. What we mean by that is by taking a linear combination of white noise variables, we necessarily obtain as a result a time series that is in fact stationary. Let Z_t be a white noise variable, and let $(Z_t, Z_{t-1}, ..., Z_{t-q})$ a random vector of white noise variables. Then, if we take a linear real-valued function $g = (\cdot, \dots, \cdot)$, and consider

$$X_t = g(Z_t, Z_{t-1}, ..., Z_{t-q}),$$

we would get a stationary time series. We also can infer from the equation, that X_t and X_s are independent if only if $|t - s| > q$. We refer to this dependence as being *q-dependent*. Additionally, a stationary time series is *q-correlated* if $\gamma(h) = 0$ for $|h| > q$.

Definition 2.1. Let $\{X_t\}$ be a time series. $\{X_t\}$ is a **moving-average process of order q** (MA(q)) if

$$X_t = Z_t + \theta_1 Z_{t-1} + ... + \theta_q Z_{t-q}$$

where $\{Z_t\} \sim WN(0, \sigma^2)$ and $\theta_1, ..., \theta_q$ are constants.

Let's now look at a few properties of *MA(q)* processes:

(a) It is a second order process.

(b) It is a centered process because

$$E(X_t) = E(\sum_{i=0}^{q} \theta_i Z_{t-i})$$

$$= \sum_{i=0}^{q} \theta_i E(Z_{t-i})$$

$$= 0,$$

where we used that $E(Z_{t-i}) = 0 \quad \forall i \in 0, ..., q$.

(c) The **autocovariance function** at *lag h* is

$$\gamma_X(h) = Cov(X_t, X_{t+h} = \sigma^2 \sum_{i=0}^{q-|h|} \theta_i \theta_{i+|h|}, \quad |h| \leq q.$$

(d) The **autocorrelation function** at *lag h* is

$$\rho_X(h) = \frac{\sum_{i=0}^{q-h} \theta_i \theta_{i+h}}{\sum_{i=0}^{q} \theta_r^2} \quad 0 \leq h \leq q.$$

Now we can express it in terms of the previously defined backwards shift operator as such

$$X_t = Z_j + \theta_1 B Z_j + \theta_2 B^2 Z_j + ... + \theta_q B^q Z_j.$$

If we define the polynomial $\Theta_q(z) = 1 + \theta_1 z + ... + \theta_q z^q$ as we did with AR(p), we can simplify the notation and write

$$X_t = \Theta_q(B) Z_j.$$

Notice that this time, no invertibilty condition is required. Additionally, we should mention that the importance of MA(q) processes is that it describes all *q-correlated* processes. We state this in the following proposition:

Proposition 2.2. *Let $\{X_t\}$ be a stationary q-correlated time series with 0 mean. Then it can be represented as the previous MA(q) process.*

Proof. See [2], page 50. □

Example 2.3. The MA(1) process. Let's write the expression for the MA(1) model, which is given by

$$X_t = Z_t + \theta Z_{t-1},$$

where $Z_t \sim WN(0, \sigma^2)$. First of all we want to compute the variance of the process, which is

$$Var(X_t) = Var(Z_t + \theta Z_{t-1})$$
$$= Var(Z_t) + \theta^2 Var(Z_{t-1}) = \sigma^2(1 + \theta^2).$$

Now, we can give the auto-covariance function. For lags 0 and 1 we have:

$$\gamma(0) = Var(X_t) = \sigma^2(1 + \theta^2)$$
$$\gamma(1) = Cov(X_t, X_t + 1) = \sigma^2\theta$$

and we can see that for any other lag the auto-covariance function is 0 (indeed, just by looking at the expression of the MA(1) process, $Cov(X_t, X_{t+c} = 0)$ for $c > 1$. Finally, we can compute the auto-correlation function for lags 0 and 1:

$$\rho(0) = 1$$
$$\rho(1) = \frac{\theta}{1 + \theta^2}.$$

We can deduce that for lags bigger than one, the correlation is 0.

2.2 ARMA(p,q) processes

Once stated what moving-average (MA(q)) and auto-regressive (AR(p)) processes are, we can now define what an auto-regressive moving-average model is.

Definition 2.4. Let $\{X_t\}$ be a time series. X_t is an **ARMA(p,q)** if it is stationary and satisfies

$$X_t - \phi_1 X_{t-1} - ... - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + ... + \theta_q Z_{t-q}$$

where $Z_t \sim WN(0, \sigma^2)$ and $(1 + \theta_1 x + ... + \theta_q x^q)$ and $(1 - \phi_1 x - ... - \phi_q x^p)$ are polynomials with no common factors, being the second invertible.

It is worth noting that a crucial condition for the definition is that the series has to be stationary

Notation: Using the backward shift operator we can express an ARMA(p,q) process as following:

$$\phi(B)X_t = \theta(B)Z_t,$$

and using the invertibility of $\phi(B)$ we can write

$$X_t = \phi(B)^{-1}\theta(B)Z_t.$$

2.3 Time series differencing

Let $\{X_t\}$ be a random walk

$$X_t = X_{t-1} + Z_j$$

with $Z \sim WN(0, \sigma^2)$ and $X_0 = Z_0 = 0$. We can see that this is not a stationary process as defined early on. To see that we can compute the *covariance function* of $\{X_t\}$. We have

$$\gamma_X(h) = Cov(X_t, X_{t+h}).$$

Since $X_t = \sum_{i=1}^{t} Z_i$ where Z_i are iid random variables, we can express

$$X_{t+h} = X_t + \sum_{i=t+1}^{t+h} Z_i = X_t + W.$$

Now we let us write

$$\begin{aligned} Cov(X_t, X_{t+h}) &= Cov(X_t, X_t + W) \\ &= Cov(X_t, X_t) + Cov(X_t, W) \\ &= Var(X_t) + 0 \\ &= Var(\sum_{i=1}^{t} Z_i) \\ &= \sum_{i=1}^{t} Var(Z_i) = t\sigma^2. \end{aligned}$$

Because the covariance function depends of t, we can not call the random walk a stationary process. In order to "make" it stationary, we can *differentiate* it. Looking at the previous example:

$$Y_t := X_t - X_{t-1} = Z_t.$$

We can clearly see that the differentiated series Y_t is now stationary since Z_t is. Let's generalize a definition for every differentiable time series.

Definition 2.5. Let X_t and Y_t be two time series. Then, if we have

$$Y_t = (Id - B)X_t,$$

we say that Y_t is process X_t **differentiated**.

We also can talk about multiple times differentiating, which is the result of the consecutive powers of the previous expression. This way, we say that process Y_t is process X_t *d times differentiated* if we have

$$Y_t = (Id - B)^d X_t.$$

2.4 ARIMA(p,d,q) processes

Once defined what differentiating means for time series, we can now define auto-regressive integrated moving-average (ARIMA) models.

Definition 2.6. Let $\{X_t\}$ be a time series. We say that $\{X_t\}$ is **ARIMA(p,d,q)** if $Y_t := (1-B)^d X_t$ is an ARMA(p,q) process.

Using the previous notations for both AR(p) and MA(q) processes, we can write our ARIMA model using the operator B as such:

$$\Phi_p(B)(Id - B)^d X_t = \Theta_q(B)Z_t \qquad t \in \mathbb{Z}.$$

Following the definition, is obvious that an ARIMA(p,0,q) is a *0 times integrated* ARMA(p,q) process, which is the same as an ARMA(p,q) process itself. It is also worth noting that the previously discussed *random walk*

$$X_t = X_{t-1} + Z_t$$

is in fact an ARIMA(0,1,0) process.

Finally, ARIMA processes can have *trend*, in which case we would write

$$\Phi_p(B)(Id - B)^d X_t = \delta + \Theta_q(B)Z_t$$

where δ is the trend, and can also have seasonality factors in them, in which case are called **SARIMAX** processes.

Chapter 3

Model Estimation

In this chapter we will give practical methods to preform a fit of an observed data, as well as briefly define the proper metrics to evaluate the fit. Finally, since it is implicitly used in the initial data-set treatment used in this work, we will make a quick mention of what Independent Component Analysis is and in what contexts can be used.

3.1 AR(1) parameter estimation: Maximum Likelihood Estimation

Let us consider the *auto-regressive model of order p* AR(p) as previously defined:

$$X_t = \phi_1 X_{t-1} - ... - \phi_p X_{t-p} + Z_t \qquad Z_t \sim WN(0, \sigma^2).$$

Given a set of observations $\{\mathbf{x}_t\}$ for $t = 1, 2, ..., T$, our objective is to estimate the vector of parameters $\phi = (\phi_1, ..., \phi_p)$ as well as the parameter σ^2 of the Gaussian White Noise, that makes the model as true as possible to $\{x_t\}$. There are numerous ways to tackle this problem, however, in our work, since it is the method used by the vast majority of the software, we will explain in depth *Maximum Likelihood Estimation*.

Definition 3.1. Let $\{x_t\}$ for $t = 1, ..., T$ be a set of observations, and let $p(x, \theta)$ be the density function of the random variable X_t. Then, we define the **Likelihood function** of the set of observations $\{x_t\}$ as:

$$L(x, \theta) = \prod_{i=1}^{T} p(x_i, \theta).$$

To start with, let's try to compute the Likelihood Function of an AR(1) model. Given our set of observations $\{\mathbf{x}_t\}$, we first need the density function of the data $\{\mathbf{x}_t\}$, which can be written as $p(x_1, x_2, ..., x_T | \phi, \sigma^2)$.

Definition 3.2. We say that a random variable X_n is a **Markov Chain** if

$$p(X_{n+1}|X_0, ..., X_n) = p(X_{n+1}|X_n).$$

Now, since the observation are co-independent, let's factor the previous expression into the product of the marginal densities:

$$p(x_1, x_2, ..., x_T|\phi, \sigma^2) = p(x_T|x_{T-1}, ..., x_1, \phi, \sigma^2)p(x_1, ..., x_{T-1}, \phi, \sigma^2)$$

$$= p(x_T|x_{T-1}, ..., x_1, \phi, \sigma^2)p(x_{T-1}|x_{T-2}, ..., x_1, \phi, \sigma^2)p(x_1, ..., x_{T-2}, \phi, \sigma^2)$$

$$= \cdots$$

$$= \prod_{t=1}^{T} p(x_t|x_{t-1}, ..., x_1, \phi, \sigma^2)p(x_1|\phi, \sigma^2).$$

But since we know that the AR(1) process is in fact a Markovian Chain, for every t we can write

$$p(x_t|x_{t-1}, ..., x_1, \phi, \sigma^2) = p(x_t|x_{t-1}, \phi, \sigma^2),$$

and our expression becomes

$$p(x_1, x_2, ..., x_T|\phi, \sigma^2) = \prod_{t=1}^{T} p(x_t|x_{t-1}, \phi, \sigma^2)p(x_1|\phi, \sigma^2).$$

From this point, let us again write the expression of an AR(1) process:

$$X_t = \phi X_{t-1} + Z_t \qquad Z_t \sim WN(0, \sigma^2).$$

It is clear by looking at the expression of the process that, given an observation of the process x_t, we have

$$x_t|x_{t-1} \sim N(E(x_t|x_{t-1}), Var(x_t|x_{t-1})),$$

moreover, we have

$$E(x_t|x_{t-1}) = \phi x_{t-1} \qquad Var(x_t|x_{t-1}) = \sigma^2,$$

so we have now an explicit expression for the previous marginal densities

$$p(x_t|x_{t-1}, \phi, \sigma^2) = \frac{1}{\sqrt{2\pi Var(x_t|x_{t-1})^2}} \exp(-\frac{(x_t - E(x_t|x_{t-1})^2}{2Var(x_t|x_{t-1})^2})$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x_t - \phi x_{t-1})^2}{2\sigma^2}).$$

From this point, there are multiple paths to be taken. If we come back to the expression of the Likelihood Function of the AR(1) process we have

$$L(\phi, \sigma^2) = p(x_1, x_2, ..., x_T|\phi, \sigma^2) = \prod_{t=1}^{T} p(x_t|x_{t-1}, \phi, \sigma^2) p(x_1|\phi, \sigma^2).$$

In Maximum Likelihood Estimation, the objective is to find the values of parameters ϕ and σ^2 that maximizes $L(\phi, \sigma^2)$. However, a usual practice is to maximize the so-called log Likelihood Function, which means to maximize the expression:

$$\mathcal{L}(\phi, \sigma^2) = \log p(x_1|\phi, \sigma^2) + \sum_{t=1}^{T} \log p(x_t|x_{t-1}, \phi, \sigma^2),$$

in which case we would be dealing with sums instead of products, which can be less difficult.

Now we want to differentiate between two ways to proceed to maximize the Likelihood Function. The first one would be to maximize the whole previous expression, which is called *Exact Maximum Likelihood Estimation*. It is what is broadly used in today's time series software, in particular by *statsmodels* python library, which is the one used in this particular work. Firstly we have to note that the first term $p(x_1|\phi, \sigma^2)$ is known. Indeed we have:

$$E(x_1) = \frac{1}{1-\phi}$$

$$Var(x_1) = \frac{\sigma^2}{1-\phi^2}$$

and we can write

$$p(x_1|\phi, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x_1 - (1/1-\phi)^2)}{2\sigma^2/(1-\phi^2)}).$$

Finally, we can give the explicit expression of the log Likelihood Function which has to be maximized.

$$\mathcal{L}(\phi, \sigma^2) = -\frac{1}{2}\log 2\pi - \frac{1}{2}\log(\sigma^2/(1-\phi^2)) - \frac{(x_1 - [1/(1-\phi)])^2}{2\sigma^2/(1-\phi^2)}$$

$$-((T-1)/2)\log 2\pi - ((T-1)/2)\log\sigma^2 - \sum_{t=2}^{T} \frac{(x_t - \phi x_{t-1})^2}{2\sigma^2}.$$

In *Exact Maximum Likelihood Estimation* the values $\{\hat{\phi}, \hat{\sigma}^2\}$ that maximize the previous function are usually found with help of iterative or numerical procedures, which is why it is mostly exclusively used by computer software.

The other path consists of considering the Likelihood of $p(x_1|\phi,\sigma^2)$ as deterministic, and maximizing the likelihood conditioned on the first observation:

$$L(\phi,\sigma^2) = p(x_2,...,x_T|x_1,\phi,\sigma^2)\prod_{t=1}^{T} p(x_t|x_{t-1},\phi,\sigma^2),$$

which is referred to by the name of *Conditional Maximum Likelihood Estimation*. We can also give explicit expression for $\hat{\phi}$ and $\hat{\sigma}$:

$$\hat{\phi} = \frac{\frac{1}{T}\sum_{t=1}^{T} x_{t-1}x_t}{\frac{1}{T}\sum_{t=1}^{T} x_{t-1}^2}$$

$$\hat{\sigma}^2 = \frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{\phi}x_{t-1})$$

which, as we can easily see, they both coincide with the Ordinary Least Squares (OLS) estimator.

3.2 AR(p) Maximum Likelihood Estimation

In this section we will generalize what was explained in the previous section, as well as give explicit values for the Maximum Likelihood estimators for AR(p) models.

Let us write the AR(p) model once again

$$X_t = \phi_1 X_{t-1} - ... - \phi_p X_{t-p} + Z_t \qquad Z_t \sim GWN(0,\sigma^2),$$

where we write $\Phi = \{\phi_1,\dots,\phi_p\}$ and let $\{\mathbf{x}_p\} = \{x_1, x_2, ..., x_p\}$ our vector with the first p-observation, which we consider it to have mean vector $\{\boldsymbol{\mu}_p\} = \{\mu_1, \mu_2, ..., \mu_p\}$ with

$$\mu_i = \frac{1}{1-\phi_1-\cdots-\phi_i} \qquad i \in \{1,\dots,p\}$$

and variance-covariance matrix as following

$$\sigma^2 Var_p = \begin{pmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \cdots & \gamma_{p-1} \\ \gamma_1 & \gamma_0 & \gamma_1 & \cdots & \gamma_{p-2} \\ \gamma_2 & \gamma_1 & \gamma_0 & \cdots & \gamma_{p-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_{p-1} & \gamma_{p-2} & \gamma_{p-3} & \cdots & \gamma_0 \end{pmatrix},$$

where γ_i is the sample auto-covariance function at lag i for $i \in \{0,\dots,p-1\}$. Let us compute in this case the density function of the first p observations:

$$p(x_p,\dots,x_1|\Phi,\sigma^2) = \frac{1}{\sqrt{(2\pi)^p}}\sqrt{|\sigma^{-2}Var_p^{-1}|}\exp\left(-\frac{1}{2\sigma^2}(\mathbf{x}_p-\boldsymbol{\mu}_p)^t Var_p^{-1}(\mathbf{x}_p-\boldsymbol{\mu}_p)\right).$$

Now let us consider the rest of the observations at times $p+1,\dots,T$. In this case we would have that the observation at time t, by definition of the AR(p) process, conditioned to the first $t-1$ observations would have a Gaussian distribution with mean $\phi_i x_{t-1}+\cdots+\phi_p x_{t-p}$ and variance σ^2. Now let us write the joint density of the observations with $t>p$, which is decomposed into the product of the marginal densities in the same manner as for the AR(1) process (all observations in the AR(1) process comply with $t>p$, except for the first one):

$$p(x_T,\dots,x_{p+1}|\Phi,\sigma^2) = \prod_{t=p+1}^{T} p(x_t|x_{t-1},\dots,x_{t-p},\Phi,\sigma^2),$$

where every marginal density is given by the expression

$$p(x_t|x_{t-1},\dots,x_{t-p},\Phi,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-(x_t-\phi_1 x_{t-1}-\cdots-\phi_p x_{t-p})^2}{2\sigma^2}\right).$$

At this stage, we can compute the log Likelihood Function of the whole sample of an AR(p) model

$$\mathcal{L}(\Phi,\sigma^2) = \log p(x_T,\dots,x_1|\Phi,\sigma^2)$$

$$= \log p(x_p,\dots,x_1|\Phi,\sigma^2)\sum_{t=p+1}^{T}\log p(x_t|x_{t-1},\dots,x_{t-p},\Phi,\sigma^2)$$

$$= -\frac{p}{2}\log(2\pi)-\frac{p}{2}\log(\sigma^2)+\frac{1}{2}log|Var_p^{-1}|-\frac{1}{2\sigma^2}(\mathbf{x}_p-\boldsymbol{\mu}_p)^t Var_p^{-1}(\mathbf{x}_p-\boldsymbol{\mu}_p)$$

$$-\frac{T-p}{2}\log(2\pi)-\frac{T-p}{2}\log(\sigma^2)-\sum_{t=p+1}^{T}\frac{(x_t-\phi_1 x_{t-1}-\cdots-\phi_p x_{t-p})^2}{2\sigma^2}.$$

As far as *Conditional Maximum Likelihood Estimation* goes, we can rewrite the previous expression but, similarly to the AR(1) model, considering the first p observations as deterministic and we would get the log Likelihood Function conditional those p observations

$$\mathcal{L}^*(\Phi,\sigma^2) = \log p(x_T,\dots,x_{p+1}|x_p,\dots,x_1,\Phi,\sigma^2)$$

$$= -\frac{T-p}{2}\log(2\pi)-\frac{T-p}{2}\log(\sigma^2)-\sum_{t=p+1}^{T}\frac{(x_t-\phi_1 x_{t-1}-\cdots-\phi_p x_{t-p})^2}{2\sigma^2}.$$

We can see that the value of $\{\Phi, \sigma^2\}$ that maximizes this expression is the same that minimize

$$\sum_{t=p+1}^{T} (x_t - \phi_1 x_{t-1} - \cdots - \phi_p x_{t-p})^2,$$

so once again we get that the CMLE estimator for Φ is in fact the same as the OLS estimator. Finally, we see that the MLE estimator for σ^2 is

$$\hat{\sigma}^2 = \frac{1}{T-p} \sum_{t=p+1}^{T} (x_t - \hat{\phi}_1 x_{t-1} - \cdots - \hat{\phi}_p x_{t-p})^2$$

which is the residual from the previous regression.

The case for *Exact Maximum Likelihood Estimation* requires numerical methods and is most common on computer software as well as mentioned in the AR(1) section.

3.3 Model comparison: Akaike Information Criterion

In cases involving "imperfect" data, meaning data that comes with weird noises or values from real experiments, it can be sometimes difficult to choose a model only by looking at its sample auto-correlation or sample partial auto-correlation functions. In this case, and counting on today's software support, a path forward is to simply estimate the parameters of more than one model, and compare the goodness of fit for every model.

However, there are times were goodness of fit is not the only factor that makes one model better than the other. Usually simpler models are easier to understand, which also makes it easier to understand the process behind it, even if they not adjust to the data as well as more complex models do. To solve this problem in comparing multiple models, the *Akaike Information Criterion (AIC)* estimates the relative amount of information lost by a model.

Definition 3.3. Let X_t be a model, with a number K of estimated parameters. Let $\hat{L}$ be the maximized value of the *Likelihood Function* during MLE. Then the **Akaike Information Criterion** is given by

$$AIC = 2K - 2\ln \hat{L}.$$

Remark 1: The objective is to minimize the AIC between the pool of models.

Remark 2: Let n be the size of the sample. For small n, it is standard to use the *second-order AIC*, given by

$$AIC_c = 2K + \frac{2K(K+1)}{n-K-1} - 2\ln \hat{L}.$$

Minimizing the AIC at the end of the day means optimizing the relationship between the goodness of fit of the model and the simplicity of it.

3.4 Source selection: Independent Component Analysis

Here we attempt to give a brief explanation of what Independent Component Analysis (ICA) is, as it is used implicitly in this work, in a quite simplified manner.

Let $(s_{1,t}, \ldots, s_{m,t})$ be m independent time series or *source signals*, and now let $(x_{1,t}, \ldots, x_{n,t})$ be n different linear mixtures of our source signals $(s_{1,t}, \ldots, s_{m,t})$ which are observed in a determined environment. Here we have an instance of the Blind Source Problem, so we have to find the source signals $(s_{1,t}, \ldots, s_{m,t})$ only given the mixture $(x_{1,t}, \ldots, x_{n,t})$. Hence, the objective of ICA is to find A and $\mathbf{x}$ such as the following expression is verified:

$$\mathbf{x} = A\mathbf{s},$$

where a is $A \in \mathcal{M}_{\mathbb{R}}(n, m)$, $\mathbf{x} = (x_{1,t}, \ldots, x_{n,t})$ and $\mathbf{s} = (s_{1,t}, \ldots, s_{m,t})$.

The first we want to do is to *center* and *sphere* our vector of correlated observations $\mathbf{x}$, which has mean $\boldsymbol{\mu} = E(X)$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{xx}} = Cov(\mathbf{x}, \mathbf{x})$. The main objective while *centering* is to make the process have mean 0, and the only step needed is to subtract $\boldsymbol{\mu}$ from $\mathbf{x}$. *Sphering*, on the other hand, aims to transform the correlated data into a vector of uncorrelated observations. For that, we have that we can decompose our covariance matrix as $\boldsymbol{\Sigma}_{\mathbf{xx}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^t$ where $\mathbf{U}$ is an orthogonal matrix with the eigenvectors of $\boldsymbol{\Sigma}_{\mathbf{xx}}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix with its eigenvalues. In case $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_{\mathbf{xx}}$ are known, the sphered version of $\mathbf{x}$ would be given by

$$\mathbf{x} \leftarrow \boldsymbol{\Lambda}^{-1/2}\mathbf{U}^t(\mathbf{x} - \boldsymbol{\mu}).$$

However, in the case of the general ICA problem, $\boldsymbol{\mu} = \overline{\mathbf{x}}$ and $\boldsymbol{\Sigma}_{\mathbf{xx}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^t$, so to center and sphere the data we will use the following transformation

$$\mathbf{x}_i \leftarrow \boldsymbol{\Lambda}^{-1/2}\mathbf{U}^t(\mathbf{x}_i - \overline{\mathbf{x}}), \qquad i = 1, \ldots, n.$$

Coming back to our original problem, we have $\mathbf{x} = A\mathbf{s}$ and our objective is to recover A and $\mathbf{s}$. For a given A with full rank, there exists what it is called a *separating* matrix W with which the sources are extracted from the original mix, hence $\mathbf{s} = W\mathbf{x}$, where in general we can write $W = (A^tA)^{-1}A^t$. In particular, if the number of observations on the mix is the same as the number of sources ($n = m$, not the case in this work), we would have $W = A^{-1}$. Moreover, if the data has been centered and sphered, A is orthogonal and $W = A^t$. In practice, we attempt to get an approximation of $\widehat{W}$, the separating matrix to give an approximation $\mathbf{y}$ of the source given by

$$\mathbf{y} = \widehat{W}\mathbf{x}.$$

Now that the problem has been stated, we can talk about the algorithms used to solve it numerically. Particularly we will briefly explain the most broadly used by software (and more efficient), the *FastICA Algorithm*.

Let Y be a projection of $\mathbf{x}$ such that $Y = \mathbf{w}^t\mathbf{x}$. The goal would be to find a vector $\mathbf{w}$ that optimizes a given objective function. For instances, if the objective function were to be the variance of the projection, we would have $Var(Y) = \mathbf{w}^t\mathbf{\Sigma_{xx}}\mathbf{w}$ where we suppose $||\mathbf{w}|| = 1$. Then, maximizing this would give us the first principal component of X, which would be the eigenvector corresponding to the largest eigenvalue of $\mathbf{\Sigma_{xx}}$. Then, by maximizing the variance of the projection of the same space within the orthogonal compliment of the subspace generated by the first principal component of X, we would get the next principal component and so on. However, while the objective function to be maximized can be chosen, the procedure by which the independent components arise will always be similar.

Chapter 4

EEG fit

4.1 Experiment description

The experiment was held by Dr. Ignasi Cos at the Center for Brain and Cognition of the Pompeu Fabra University of Barcelona (Cos et al. 2021). The main purpose of the experiment was to characterize the influence of social pressure on the participant's movements and choices, as well as its associated neuromodulation.

During the experiment, the participant was sat in front of a blank digital tablet, which at its time showed an point of origin and two targets. The aim of the subject was to draw a line from the origin to the nearest point from the center of the target, whichever direction they chose to draw towards. Then a bar plot was displayed, which gave feedback on their accuracy, as well on their opponent's. This opponent was not a real person competing against the subject, hence the bar showing the accuracy of the opponent was conveniently shown to the subject to induce the effects of social pressure. The procedure is graphically represented in figure 4.1.

In order to test the changes in the motor control of the subjects, they were tested while playing multiple trials in 3 states (randomly shuffled): solo, easy and hard. In the first state, no opponent was shown, so they had no notion of competition, which would reflect the state of social pressure 0. In the next state, a weaker opponent was brought up against, which would come to describe a positive kind of social pressure. Finally, in the hard state, which corresponds to a negative or constraining social pressure, the bar displayed the accuracy of a stronger player than the subject. A total number of 60 electrodes were placed in every subject scalp, so, for every trial, 60 EEG samples were generated.

The hypothesis was that with a proper analysis of the EEG generated in each

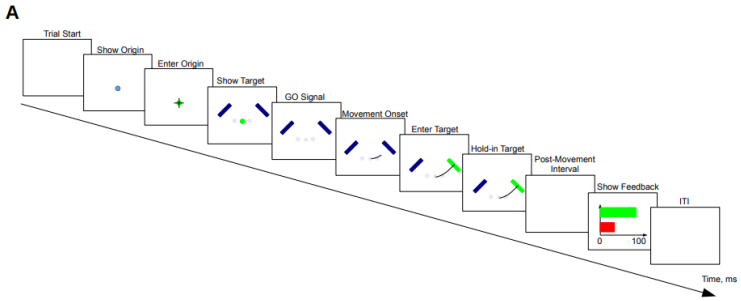


Figure 4.1: Experiment procedure

state, some classification criteria would arise with which the different states of social pressure undergone by a subject could be detected just with the EEG information.

The generated data-set for a single subject can be regarded as a set of time series that can be looked at as a matrix of dimensions (6 x 2 x 60 x 108 x 1200). The data was aggregated into 6 groups, 2 for each state defined previously. For each group, 2 sessions of experiments were made. For each session, the data was gathered by 60 receptors placed in the scalp of the patient, each one of whom was made to repeat the experiment 108 times per session. Finally, the EEG readings were cut out to be exactly 1200ms of time lenght.

This data-set can be divided into 3 subsets of dimensions (60 x 432 x 1200), each one corresponding with the data gathered with each level of social pressure (solo, easy and hard). At this stage an Independent Component Analysis (ICA) was performed by the researchers in order to determine which combination of the 60 channels captured the more information. After the mentioned ICA, 18 of those 60 where ruled out, which left 42 for the analysis of the data, leaving us with a data-set of dimensions (42 x 432 x 1200).

4.2 Preliminary observations

The first thing to do with such a large data-set is to separate the data into channels, so only one of them would be considered at a time. This leaves us with 42 data-sets of dimensions (432 x 1200), in other words, with 432 time series of 1200ms of length. In the case of this work, subject 25 of the study was used, and the first fit was performed in its channel 1 while playing in "Solo" state. The data

at this stage is plotted in figure 4.2.

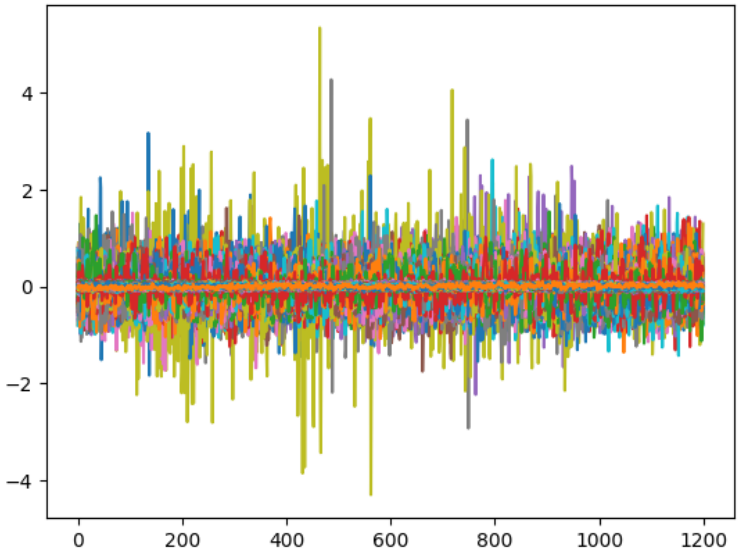


Figure 4.2: Subject 25 Solo

Having such a dense set of observations in each time series, it was convenient to perform some kind of down-sampling, not only to lighten qualitative work, but also to make the fit easier for the computer. The ratio that seemed adequate to not loose to much information in the process was a 1:3 ratio, so the dimension of the data-set became (432 x 400). In figure 4.3 we can see a plot of the data-set after the 1:3 down-sampling.

Now let's have a look at two particular samples in figure 4.4. As it can be easily appreciated, the variability in them is too high. We are not interested in the complete frequency spectrum, we will only focus on the frequencies with the most variability.

In order for us to constrict the sampling frequencies up to 40.000 Hz, we had to make use of a frequency filtering algorithm (Infinite Impulse Response), which would break down the signal into frequency bands, and then rearrange the signal into three separate signals which would correspond to the low frequencies (*alpha*, from 8 to 12 Hz), the mid frequencies (*beta*, from 15 to 30 Hz), and the high frequencies (*gamma*, from 40 to 80 Hz). Figure 4.5 is a plot of the decomposition into band frequencies of *sample 0*.

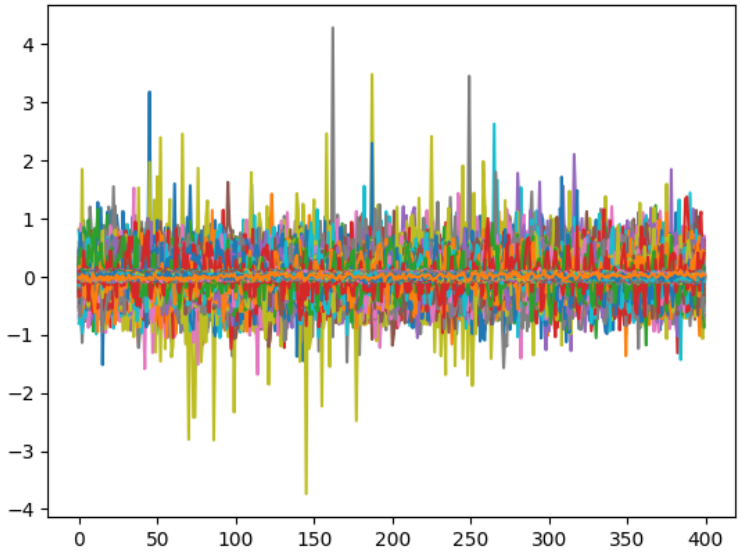
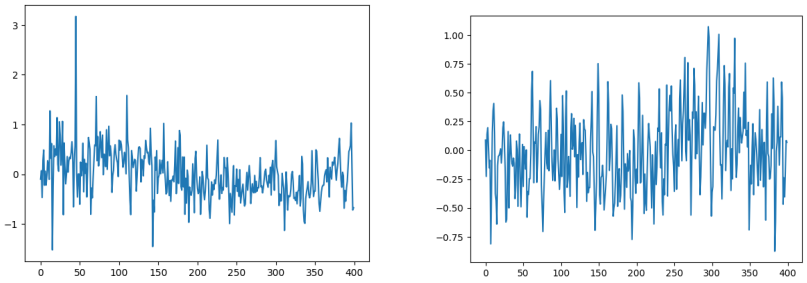
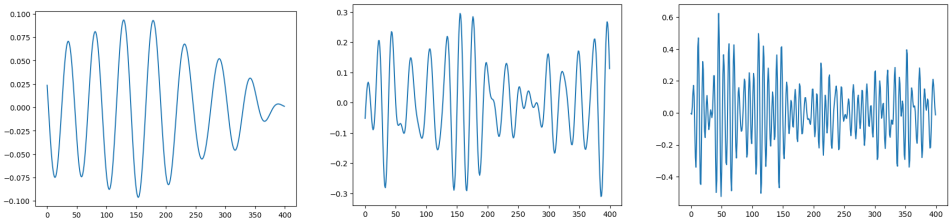


Figure 4.3: Subject 25 after down-sampling



(a) Sample 0 (b) Sample 100

Figure 4.4: Subject 25 individual samples



(a) Alpha (b) Beta (c) Gamma

Figure 4.5: Sample 0 frequency breakdown

Observing the plots, as for the alpha frequencies, not very much variability is appreciated, nor in the beta frequencies. That is the reason that following steps were made with the *gamma* frequencies of the 432 samples of subject 25 on "Solo".

4.3 Model choice

As stated before, the goal of this work is to find a model that captures the time variability of the EEG readings of Dr. Ignasi Cos' experiment. Let us express the structure of our data so far, which can be looked at as a matrix like so:

$$\begin{Bmatrix} x_1^1 & x_2^1 & x_3^1 & \dots & x_{400}^1 \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_{400}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{432} & x_2^{432} & x_3^{432} & \dots & x_{400}^{432} \end{Bmatrix}$$

where each row of the matrix is the *n-th* sample out of the 432 from the dataset, and every column is the observation at time $t \in (0, 400)$.

The way that seems more productive is for us to look at our data as a set of 432 samples of the same process, each one of which has 400 observations. A reasonable first step would be to try to fit each one of the samples separately into independent processes, being a AR(1) model a also reasonable starting point. We would have processes like:

$$X_{1,t} = \phi_1 X_{1,t-1} + Z_{1,t}$$
$$X_{2,t} = \phi_2 X_{2,t-1} + Z_{2,t}$$
$$\dots$$
$$X_{432,t} = \phi_{432} X_{432,t-1} + Z_{432,t}.$$

Once every fit is performed and every ϕ_i and σ_i for $i \in (1, 432)$ is estimated, we could compute the arithmetic mean Φ and Σ of the sets $\{\phi_1, ..., \phi_{432}\}$ and $\{\sigma_1, ..., \sigma_{432}\}$ to finally get a suitable model for all observations:

$$X_t = \Phi X_{t-1} + Z_t \qquad Z_t \sim GWN(0, \Sigma^2).$$

AR(p) models, as previously discussed, seem intuitively appropriated for this kind of fits because, while keeping a relatively simple structure, they are making a strong claim that each observation will depend on a combination of previous ones. Perhaps this assumption is not as strong in this particular case as it would be in Econometrics or other fields were sets of observations are modeled. However, we

are considering them as a starting point in this particular work.

While the previous option was a fair way to compute a model suitable enough for our 432 samples, the ideal scenario would be to perform a single fit for all 432 samples **simultaneously**. A way to tackle the problem is to artificially construct a single time series that contains all the information of the independent samples. Making a time series of length 400 in which every observation is the mean of the 432 observation at the same time is a way forward. This would look like computing the series $\bar{x}_t = \{\bar{x}_1, ..., \bar{x}_{400}\}$, and fitting it into a model that would look like:

$$\bar{X}_t = \hat{\Phi}\bar{X}_{t-1} + \bar{Z}_t.$$

However, an alternative to construct a single time series in a way that the minimum information is lost, is to concatenate all 432 samples one after the other in the following manner:

$$\hat{x}_t = \{x_{1,1}, x_{1,2}, ..., x_{1,400}, x_{2,1}, ..., x_{2,400}, ..., x_{432,1}, ..., x_{432,400}\}.$$

In the figure 4.6 we can see a plot of the previous time series:

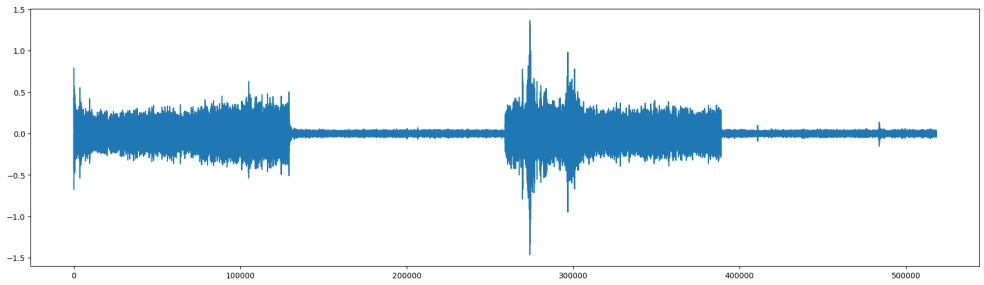
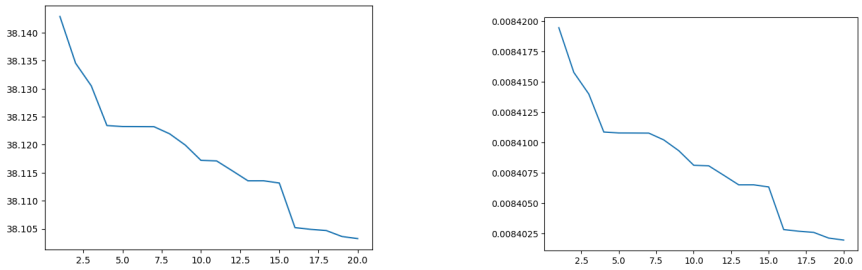


Figure 4.6: 432 samples concatenation

Although it cannot be appreciated in the figure due to the density of the observations, it is true that sudden discontinuities are generated at times $400t$ because of the jump from the sample x_i to x_{i+1}. However, being this every 400 observations we consider the impact of this phenomena to the overall fit to be negligible.

The only thing left to be discussed before starting the fit is the optimal number of parameters for the model to have, which in the auto-regressive model case would correspond to the *order (p)* of the model AR(p).

For that, no conclusive number can be extracted from the current neurological knowledge, so there is no obvious starting point. The only reasonable procedure is to, by the use brute force, perform a fit of or set $\{\hat{x}_t\}$ using an increasing number of parameters iteratively, and compare the goodness of each fit with the others (up to a reasonable number of parameters, say 20). This way, the optimal autoregressive model should arise. In figure 4.7 we can see a plot of the errors in each AR(i) model $i \in \{1, ..., 20\}$.



(a) $\sqrt{SSE}$ AR(i) fit (b) Sample 100

Figure 4.7: MSE AR(i) fit

In (a) we are computing, for every number of parameters, the squared root of the sum of the squared errors, or what is to say:

$$\sqrt{\sum_{i=1}^{432}(\hat{x}_i - X_i)^2},$$

where X_i is the value predicted by the model. In (b) we can see the *Mean Squared Error* of the fit, which can be written as

$$\frac{1}{432}\sum_{i=1}^{432}(\hat{x}_i - X_i)^2.$$

From looking at the errors, we can see that the difference in goodness of fit between all AR(i) models is not substantial. Indeed, it would be fair to say that to accumulate an error of little over 38 in a set $\{\hat{x}_t\}$ of 172.800 observations with $|\hat{x}_i| \leq 1$ for every i is not a considerable error. Furthermore, the MSE is of the order of 10^{-3}, which for the described set is not a very high measure. What that means for us is that we have reasons to choose whatever model we like, and the standard procedure in this case is to choose the simplest one, hence the one with the least parameters. In our case would be to choose the AR(1) model.

Having a reduced measure of error means that the model is well adjusted to the data, which means that is good for "explaining" our data-set. However, for this work, this is not the only criterion to be followed in order to determine a suitable model for the data. Once the model is chosen, it will be used in a classifier, which has to be able to differentiate between EEG sampled during the "Solo" state, the "Easy" state and the "Hard" state. Capturing the behavior of the samples only explains us the past, what we need is a tool that helps us to classify future data. In order to test the model in this context, it is necessary to evaluate its forecasting capabilities and, for that, acceptable performance in in-sample and out-of-sample predictions are required from it.

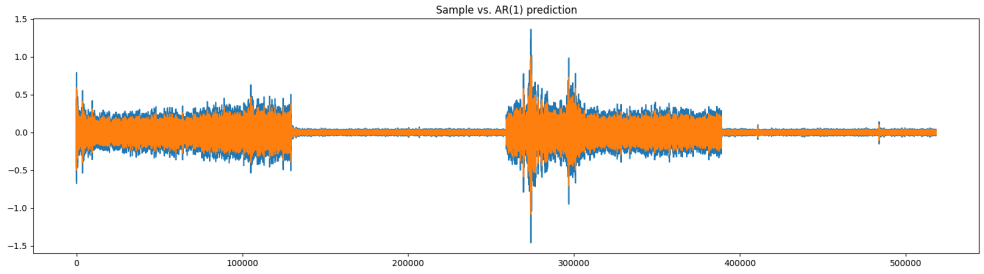


Figure 4.8: Sample (blue) vs. Prediction (orange) in AR(1) fit

In figure 4.8 we can see the in-sample prediction performance of our model. As it can be easily appreciated, it is not as good as expected. In figure 4.9, 100 steps of out-of-sample prediction are plotted, and the performance is as bad, if not worse, that in-sample prediction. What you would expect, is that the forecast presents something with a similar behavior than the sample, but in our case (not only in the case of the AR(1) model, as the same procedure was followed with AR(p) models, as well as with ARMA(p,1) models) the forecasting had a tendency of going to zero in the 20th step more or less instead of oscillating around the $x = 0$ axis as the sample does.

At this point, the characteristics of the data-set had to be reassessed in order to find the reason why the fits are that poor. Although it has been already regarded as a erratic procedure, individual fits for every one of the 432 samples were made, point being that in this manner we could be able to analyze the behavior of every sample on its own, to see if the data-set had to be altered or polished in order to get a good fit.

In figure 4.10 we can see the plots for the 432 individual fits, using an AR(1) model and a AR(3) model. The deviation in the parameters is not considerable (0.1 in the worst of cases), as for all 432 samples it would seem that the model that suits them has similar parameters (a). However, we can appreciate severe

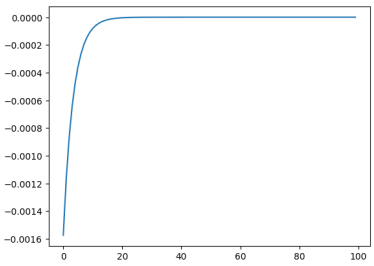
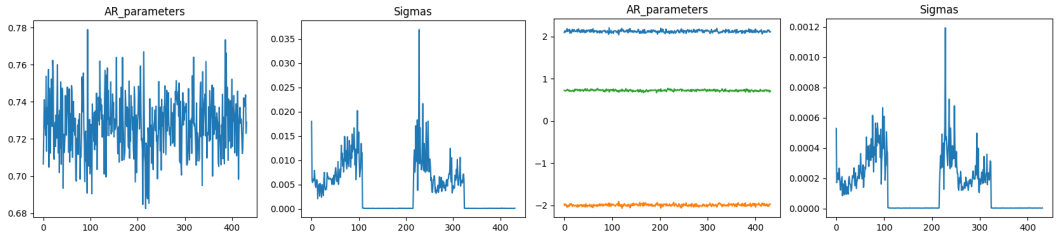


Figure 4.9: 100-step forecast in AR(1) model fit



(a) AR(1) fit of 432 samples

(b) AR(3) fit of 432 samples

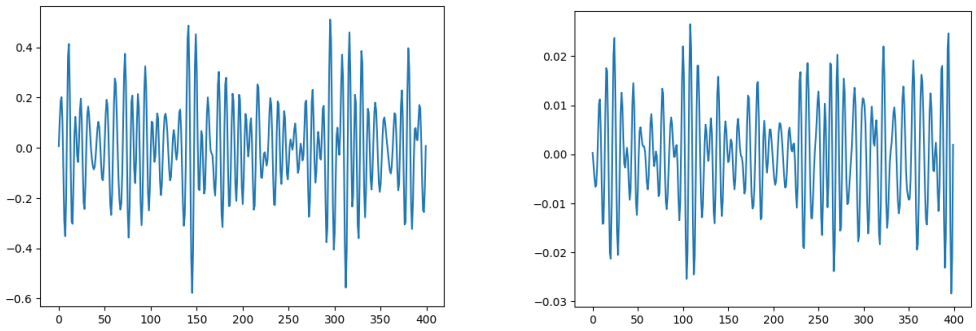
Figure 4.10: Individual fits

discontinuities in the sigmas plot (b), which it is not supposed to be the case. The conclusion drawn from this exercise, is that **the modeling of the data is not adequate because of the high variability in the samples**. Hence, the series resulting of the concatenation of the samples is too volatile to be fitted into one single process.

This point can be proved if we look at each sample individually. If we observe the two samples in figure 4.11, we see that the first one is oscillating between 0.6 and -0.6, while the second one is doing so between 0.03 and -0.03. The data-set is indeed to variate.

That phenomena is not only happening in channel 1 of the data, but in every of the 42. The data of this experiment, as previously mentioned, is separated into 2 sessions. Discussing this point with Dr. Cos, the reason behind the variance in the amplitude of the waves arose, which is that between the 2 sessions, the electrical impedance of the electrode of every channel was different, which made the readings live in two completely different scales.

This is a recurrent problem while working with data gathered with real exper-



(a) Sample 100 (b) Sample 200

Figure 4.11: Sample comparison

iments, which are often full of imprecision and measuring errors, that have to be taken into account while modeling. In fact, the inconsistency in the variance of the samples, even within the same session, can be due to the simple fact that some more gel was applied to the equipment between trials.

In light of the above, a similar procedure as before was applied. The idea from this point forward is to still concatenate the samples to make a simultaneous fit, but this time two concatenations were made, one for each session. This means that for every channel, **two fits will be made**, so two models will be generated from two sets of observations $\hat{x}_t^1$ and $\hat{x}_t^2$. One of them will be containing samples 1 to 108 and 217 to 324, and the second one samples 109 to 216 and 325 to 432, so we can write

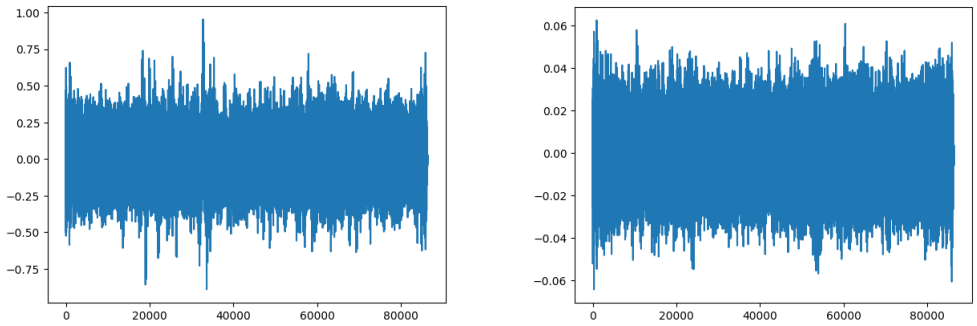
$$\hat{x}_t^1 = \{x_{1,t}, x_{2,t}, \ldots, x_{108,t}, x_{217,t}, \ldots, x_{324,t}\},$$

$$\hat{x}_t^2 = \{x_{109,t}, x_{110,t}, \ldots, x_{216,t}, x_{325,t}, \ldots, x_{432,t}\}.$$

The two sets of observations $\hat{x}_t^1$ and $\hat{x}_t^2$ for channel 1 are plotted in figure 4.12, where the difference in the amplitude between the two can be clearly appreciated.

Now, we want to find a suitable model for each session. For that, being confident that we were on the right track, we computed an estimation of parameters for every ARIMA(p,i,q) process with $p, q \in (0, \ldots, 9)$ and $i = 0, 1$ iteratively using computer software. After that, the best model is regarded to be the one that minimizes the *Akaike Information Criterion* previously explained. This procedure is known in the Data Science community as "grid search".

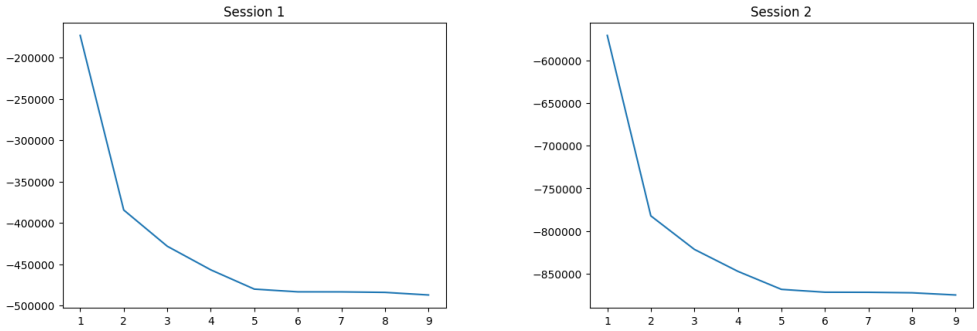
To start with, we wanted to see how the modeling using AR(p) for $p \in (1, \ldots, 9)$ processes behaved. In both cases, we observed that the model that was better ad-



(a) Sample concatenation for session 1 (b) Sample concatenation for session 2

Figure 4.12: Sample concatenations by sessions, channel 1

justed to the data was the one with more parameters. We can see that in figure 4.13, where we plotted the respective AICs for every model tested for the two sessions. A very similar (not equal, as the differentiated data is slightly different) tendency downwards is appreciated, which suggests that the best model will be the one with the most parameters, at least inside the bounds where we aim to analyze. Hence, if we wanted to choose an AR(p) model for our data from our pool, we would choose an AR(9) process. However, given the shy difference between the relative information loss in models AR(5) to AR(9), it would be fair to choose the AR(5) model, which is much more simple. In this case, the respective AICs would be -480035.17488654243 and -868454.1562824219.



(a) AIC for session 1 (b) AIC for session 2

Figure 4.13: AIC for models AR(p) for $p \in (1, \ldots, 9)$

Next, as we advanced into more complex models, we looked at ARMA(p,q) for $p, q \in (0, \ldots, 9)$, where we obviously will revisit AR(p) models, nevertheless it is considered worth the effort. In figure 4.14 it represented in a color-map the AIC

values of all ARMA processes as previously mentioned. Along the y-axis are the autoregressive terms p, and along the x-axis the moving average terms q. Whilst it is true that the computed minimum corresponds to the ARMA(9,9) process, it can be apreciated from the plot that the ARMA(4,3) is relatively low in terms of relative information loss, in both cases. In session 1, the computed minimum is -491493.71172786, and the value for $(p,q) = (4,3)$ is -487040.21370961, and for session 2 is -878602.57134095 as optimal versus -874654.65132153. With a similar argument as in the AR process, we could consider as the best ARMA model the ARMA(4,3) process.

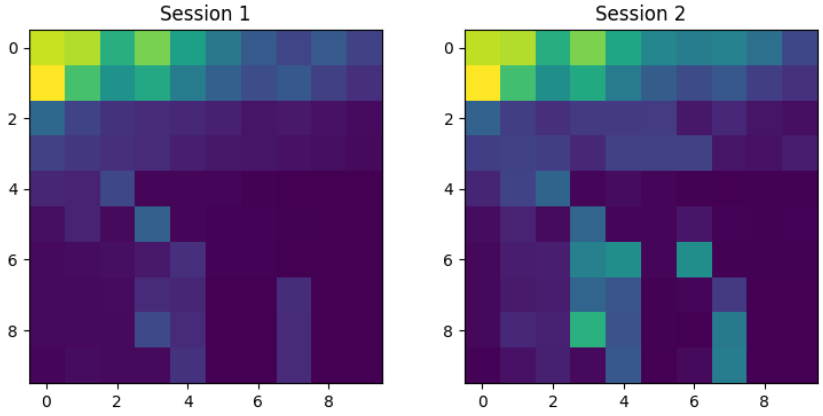


Figure 4.14: AIC for ARMA(p,q) with $p,q \in (0,\ldots,9)$

Finally, we considered the most complex model in our work, ARIMA models. Although the data is supposed to be stationary by hypothesis, we wanted to see if by any chance we could bring some better fits for our data. We fitted the data using models ARIMA(p,1,q) for $p,q \in (0,\ldots,9)$, and, as in the previous cases, the AICs are plotted in figure 4.15. In this particular case, the results are not as conclusive as with AR and ARMA models. The process with the least relative information loss in session 1 is the ARIMA(7,1,7) process, and in session 2 is the ARIMA(6,1,9) process, which, for starters, are surprisingly not the same.

No apparent reason why this discrepancy when it comes to integrated models immediately arises. Not only the best model for the two sessions is not the same, but also the AICs of the models with different parameters are also quite far apart, much more that in the other cases. There is no clear candidate for both sessions with similar relative information loss, and for this reason, no ARIMA models have been considered in this work.

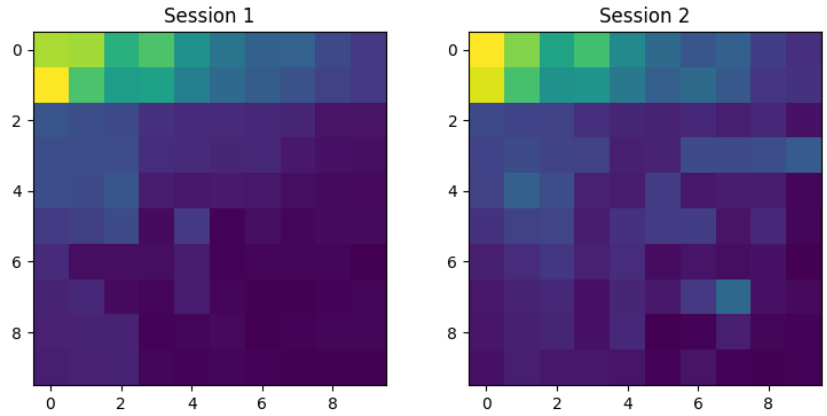


Figure 4.15: AIC for ARIMA(p,1,q) with $p, q \in (0, \ldots, 9)$

Additionally, it was though to be worth analysing not only channel 1, but to see if the same models that seem appropriate for this channel are also suitable for other channels. In figures 4.16 and 4.17 we can see the AIC of the models in the same manner as we recently explained for the data of channels 2 and 11.

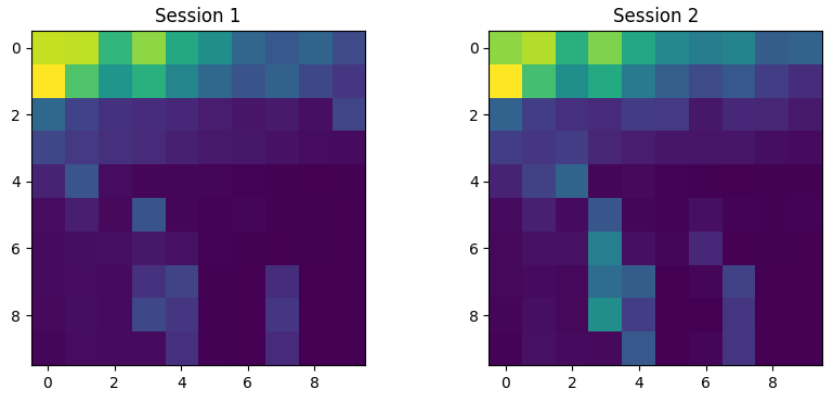


Figure 4.16: AIC for ARMA(p,q) with $p, q \in (0, \ldots, 9)$ in channel 2

If we look at the first column of both plots (except for the (0,0) coordinate), we will see that it corresponds to the models ARMA(p,0), i.e. the AR(p), and we can identify the same behavior as in channel 1. The best model to choose would be the AR(9) process, but we can see that the AR(5) has almost the same AIC, and is much simpler. Furthermore, looking at the whole plot, the coordinate (4,3) has a relatively low information loss in its surroundings in both session of both

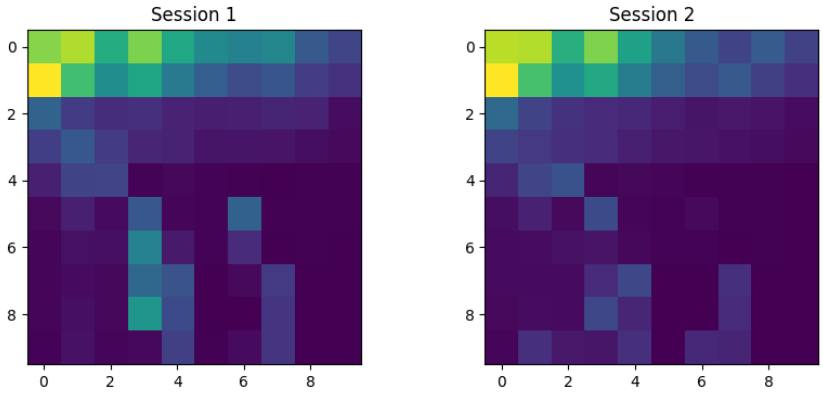


Figure 4.17: AIC for ARMA(p,q) with $p, q \in (0, \ldots, 9)$ in channel 11

channels, and the corresponding value is close to the AIC of the best model, in all cases the ARMA(9,9) process. We have seen that the AR(5) and ARMA(4,3) processes are acceptable choices to fit the data in channel 2 and channel 11. Not only that, **if we follow the same reasoning as we did previously in channel 1, we would end up choosing the same two models**.

4.4 Model evaluation

In this section we will give the specifics of the chosen models for channel 1, as well as evaluate them using various previously mentioned metrics and some standard practices in time series analysis.

To begin with, let's have a look at the fit using the **AR(5)** process. In the table below we can see the estimated parameters of the model, as well as the corresponding values for the *Mean Squared Error* and the squared root of the *Sum of Squared Errors*, which have been both previously defined.

First thing that stands out is the similarity between the estimated parameters in both of the session, which brings up reasons for regarding the both sessions two as samples of the same process. The values for both of the in-sample prediction error metrics are very positive, bearing in mind the characteristics of both data-sets. In session 1 we have a total of 86400 observations oscillating between 1 and -1. Taking this into account, accumulating an error of roughly 4.4 with a mean error per-observation of 0.0002 is believed to be an acceptable goodness of fit. In case of the second session, the we have the same 86400 observations, this time oscillating between 0.06 and -0.06, and we are accumulating a total error of 0.4

AR(5)	Session 1	Session 2
ϕ_1	2.09827819e+00	2.07595737e+00
ϕ_2	-2.00439529e+00	-1.98658612e+00
ϕ_3	3.26459451e-01	3.45353728e-01
ϕ_4	6.14852964e-01	5.69482153e-01
ϕ_5	-4.86164323e-01	-4.65846264e-01
σ^2	2.26192001e-04	2.52335608e-06
MSE	0.00022621754595701047	2.5384481812761144e-06
$\sqrt{SSE}$	4.420994907335419	0.46831818549171916

distributed with a mean of $2.5 * 10^{-6}$. Furthermore, in figure 4.16 we can see that in both cases, the predicted in-sample values and the observed values overlap at almost all times.

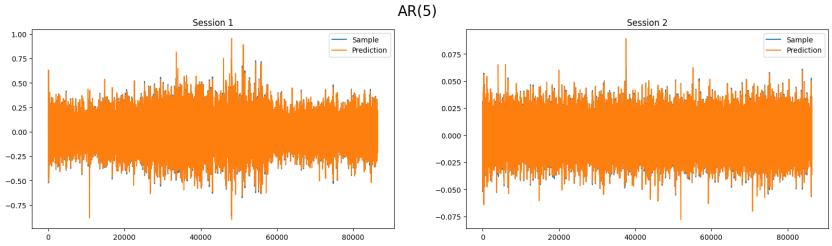


Figure 4.18: In-sample prediction using AR(5) process

Now, as far as out-of-sample prediction goes (or forecasting), we performed a fit for the first 86300 observations, and plotted a 100 step forecast over the last 100 values of the data-set. We can see the results in figure 4.18. The forecasting of the first 15 steps is pretty accurate in both cases, considering the amount of observations of the sample, but it can be appreciated more accuracy in the first session. Afterwards, we can see that the model predicts towards the mean of the sample (0) as expected from this type of fits.

Next, let's analyze in the same manner the fit with the **ARMA(4,3)** process. In the table below we can see the estimated parameters and the values for the respective *MSE* and *SSE*.

Looking at the auto-regressive estimated parameters of the two sessions, we can see the same phenomena as in the previous model. However, there is a slight discrepancy between the moving-average parameters. In terms of goodness of fit compared to the AR(5) model, we can see a shy improvement looking at the com-

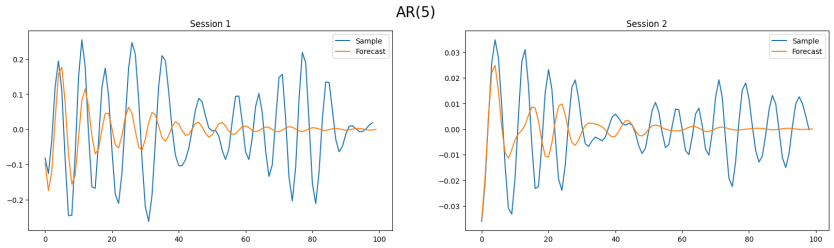


Figure 4.19: Out-of-sample prediction using AR(5) process

ARMA(4,3)	Session 1	Session 2
ϕ_1	2.74608509e+00	2.71910915e+00
ϕ_2	-3.67224744e+00	-3.60760427e+00
ϕ_3	2.50365423e+00	2.43897829e+00
ϕ_4	-8.29655358e-01	-8.05111152e-01
θ_1	-8.02708706e-01	-7.44607674e-01
θ_2	3.15417712e-01	2.42185565e-01
θ_3	-2.24779511e-01	-1.85984067e-01
σ^2	2.08490538e-04	2.34945091e-06
MSE	0.00020859290382794764	2.363592298323305e-06
$\sqrt{SSE}$	4.245282898787155	0.4519008459553197

puted error values. This is not as easily appreciated when you plot the predicted values over the sample, as we can see in figure 4.20.

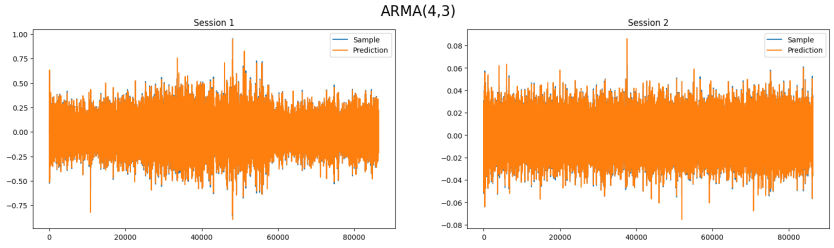


Figure 4.20: In-sample prediction using ARMA(4,3) process

Moving on towards the forecasting performance of the fit, we followed the same procedure as before and plotted the results in figure 4.21. The immediate conclusion is that the forecasting is very similar in the AR(5) and ARMA(4,3) models, for both session 1 and session 2.

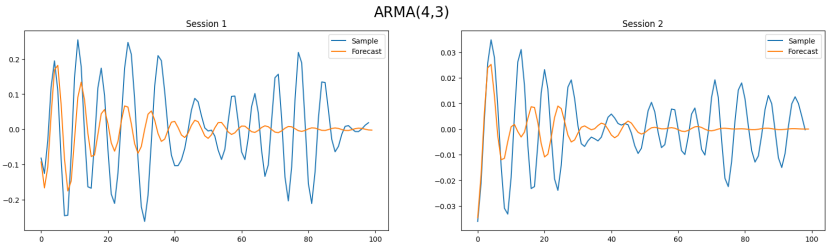
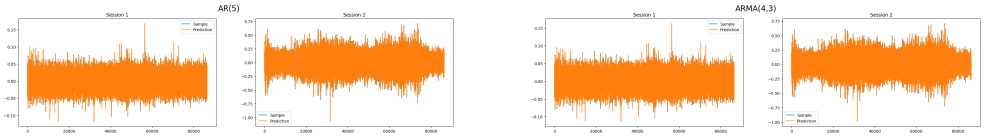


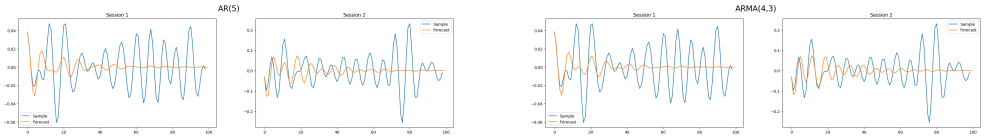
Figure 4.21: Out-of-sample prediction using ARMA(4,3) process

Finally, we wanted to see if the models' behavior was the same in other channels, so in figures 4.22 and 4.23 we add plots for in-sample and out-of-sample performance of the same models while fitting data of channel 11. We can see that the performance is quite similar between sessions, as well as between processes. In-sample performance is as good as expected, and we still maintain about 10 steps of accurate out-of-sample forecasting. Now, we can say that the fit as well suited to channel 11 as it is to channel 1.



(a) AR(5) in-sample performance (b) ARMA(4,3) in-sample performance

Figure 4.22: In-sample performance in channel 11



(a) AR(5) out-of-sample performance (b) ARMA(4,3) out-of-sample performance

Figure 4.23: Out-of-sample performance in channel 11

4.5 Code

Here we leave the specifics of the code that was written in order to get the fit for the model previously discussed, as well as the metrics that were used to test its goodness of fit. The import of libraries used is the following.

```
import numpy as np
import scipy.io
import pandas as pd
import matplotlib.pyplot as plt
import scipy.signal as spsg
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

To start with, we give the code used for the initial treatment of the raw data-set provided by Dr. Ignasi Cos, which as stated before, used the data of subject 25 of the study.

```
mat = scipy.io.loadmat('dataClean-ICA3-25-T1.mat')  #Load data
mat = mat['ic_data3']
dades_dz = mat[:,:,:,:]

dades_solo = dades_dz[:,:,:,[0,1,6,7]]  #Separate data by state
dades_easy = dades_dz[:,:,:,[2,3,8,9]]
dades_hard = dades_dz[:,:,:,[4,5,10,11]]


ch1_solo = dades_solo[0,:,:,:]  #channel 1 -> (400, 108, 4)

#agregate data from 3D to 2D separaing by session -> 2 x
(216,1200)
ag_data_solo_ch_1_s1 = np.array(ch1_solo[:,:,0])
ag_data_solo_ch_1_s2 = np.array(ch1_solo[:,:,1])
ag_data_solo_ch_1_s1 = np.append(ag_data_solo_ch_1_s1, ch1_solo
[:,:,2], axis=1)
agrupated_data_solo_ch_1_s2 = np.append(ag_data_solo_ch_1_s2,
ch1_solo[:,:,3], axis=1)

#down-sampling -> 2 x (216,400)
agrupated_data_solo_ch_1_s1 = agrupated_data_solo_ch_1_s1[::3,:]
agrupated_data_solo_ch_1_s2 = agrupated_data_solo_ch_1_s2[::3,:]
```

Then the frequency filter was applied as follows, and, after that, the concatenation of the samples by session, which will be fitted afterwards.

```
n_order = 3
sampling_freq = 500. #Sampling rate
nyquist_freq = sampling_freq / 2.
freq_band = 'gamma' #In our case we work with gamma frequencies

if freq_band=='alpha':
    low_f = 8./nyquist_freq
    high_f = 12./nyquist_freq
elif freq_band=='beta':
    low_f = 15./nyquist_freq
```

```
        high_f = 30./nyquist_freq
    elif freq_band=='gamma':
        low_f = 40./nyquist_freq
        high_f = 80./nyquist_freq
    else:
        raise NameError('unknown filter')

    #Apply filter
    b,a = spsg.iirfilter(n_order, [low_f,high_f], btype='bandpass',
    ftype='butter')
    filtered_ch1_s1 = spsg.filtfilt(b, a, agrupated_data_solo_ch_1_s1,
     axis=0)
    filtered_ch1_s2 = spsg.filtfilt(b, a, agrupated_data_solo_ch_1_s2,
     axis=0)

    filtered_ch1_400_s1 = filtered_ch1_s1[:,0]
    filtered_ch1_400_s2 = filtered_ch1_s2[:,0]

    #Concatenation
    for i in range(1,216):
        filtered_ch1_400_s1 = np.append(filtered_ch1_400_s1,
    filtered_ch1_s1[:,i])
        filtered_ch1_400_s2 = np.append(filtered_ch1_400_s2,
    filtered_ch1_s2[:,i])
```

Now we will give the code for one of the multiple model test, in particular, the one performed to compare ARMA models.

```
    aic_s1 = np.empty((10,10))
    aic_s2 = np.empty((10,10))
    for i in range(0,10):
        for j in range(0,10):
            if (i+j == 0):  #Case ARMA(0,0) process
                aic_s1[i][j] = 0
                continue
            model_s1 = ARIMA(filtered_ch1_400_s1,order=[i,0,j])
            model_s2 = ARIMA(filtered_ch1_400_s2,order=[i,0,j])
            model_fit_s1 = model_s1.fit()
            model_fit_s2 = model_s2.fit()
            aic_s1[i][j] = model_fit_s1.aic
            aic_s1[i][j] = model_fit_s1.aic
    print(np.argmin(aic_s1))
    print(np.argmin(aic_s2))
```

The mean execution time for codes like the previous one, which was performed over data from multiple channels was **4 hours**, which is considered to be inside what is expected for this kind of procedures. Besides, one way to optimize the code would be to, as we have already seen that the estimated AR parameters for both sessions are very similar, use the parameters estimated for session 1 as the

starting parameters in the fit for session 2, which is believed to save us some execution time.

Chapter 5

Discussion

As stated in the abstract, the main goal of this work was to find a suitable time model for the EEG data gathered during the experiment on social pressure. More specifically, though, the challenge presented to do so was to find a way to fit more than one sample of the same process using time series. In time series studies, the standard methods described in the relevant literature are for finding a model for one single sample of T-observations of one given process, such as wine sales in Australia or Lake Huron's water level. For that, you can analyze the plot of the ACF or PACF previously presented in this work, and qualitatively estimate what model would suit best the data before fitting it to the model and estimating its parameters.

However, the application of these resources was not that straightforward. In our data-set we had 432 repetitions of the same process, because it was gathered from the same subject performing the same action under the same level of social pressure. Theoretically, the fact of having multiple repetitions means having more information and, subsequently, having more tools to do a better fit, which can be regarded as an advantage. Nevertheless, we had to find a way to use this information, which could not be accommodated using standard time series methods.

At first, multiple approaches were considered. Some of them were supposed to give us good fits, but were unable to capture the totality of the information that was available to us. Later on, once we thought that we found a way to take advantage of the data-set using the concatenation of the 432, we were disappointed with the results. But as oftentimes working with real data, it was not the methodology that failed us, it was us that failed to treat the data correctly for the methodology to work properly. This is a lesson that we take, when working with data gathered from real experiments; the importance of understanding the physical process behind it in order to provide intelligence to the modeling of the data, which is

impossible to get from the plain application of standard methods.

Once a fitting strategy was figured out, we had to find the best process to model our data. In most situations in time series analysis, plotting the ACF and PACF you can get a pool of two or three models to fit the data, estimate the corresponding parameters, and finally compare them using multiple Information Criteria, such as Akaike or Bayes. Once again, our case was different. Having a process with so many observations and high variability (result of human and measuring errors) no useful information can be gathered from the previous method. In our case, making use of the massive impact that software resources have on Data Science, by use of the so-called "grid search", which at the end of the day means using brute force computing to fit and compare multiple models, we were able to choose our model from a much bigger pool of processes.

The most similar procedure that was followed by this work compared to standard time series methods was the testing of the chosen models, the AR(5) and the ARMA(4,3) processes. Thus, in-sample and out-of-sample prediction was tested. Although both were satisfactory enough, it has to be said that we were much happier with the former. If we had to choose between one of the two models, it would be fair to choose the AR(5) process, because, while it is true that the ARMA(4,3) has a smaller relative information loss as the AIC suggests, the improvement is not substantial. Moreover, in-sample prediction is very similar (obviously being a little better in the ARMA(4,3) process), and the same can be stated as far as out-of-sample prediction goes.

With all of that in mind, it can be stated that the improvement that the ARMA(4,3) process brings to the table is not enough to compensate the complexity added by the model over the **AR(5)** process. Moreover, with the tests conducted on multiple channels, we extend this choice to the totality of the data in "Solo" state.

To conclude this work, we consider that the objective that was set at its beginning has been accomplished. We found a suitable model for our data, able to capture time variability, as we aimed to do. In the process, we have been able to, by making little twists on known methods, perform a successful fit of multiple samples of the same process using time series.

On a personal level, I have learned more than I expected about time series analysis, both in abstract terms and its applications. In order to perform the fit of the data, I believe that a deep understanding of time series models and parameter estimation was needed, and I had to push myself to be able to tackle the problems presented during the process. Furthermore, I have gathered an immensely valuable experience in managing data-sets and the tools available to do so, as well as

facing the troubles that come with data gathered from real experiments. Some of the programs used in the "grid search" took more than 3 hours to run, which has made me develop the notion of resource optimization, a very concurrent limitation in this type of work.

Finally, this work has taught me a very important lesson about not being contented with disappointing results, and always trusting in one's capabilities to extract the best out of the situation.

Bibliography

[1] Paul S.P. Cowpertwait, Andrew V. Mercalfe, *Introductory Time Series with R*, Springer, (2008), 1–42, 67–87.

[2] Peter J. Brockwell, Richard A. Davis, *Introduction to Time Series and Forecasting, Second Edition*, Springer, (2002), 1–108, 179–219.

[3] Robert H. Shumway, David S. Stoffer, *Time Series Analysis and its Aplications*, Springer, (2011), 1–162.

[4] Ioannis Kasparis, *A simple proof for the invertibility of the lag polynomial operator*, Research Institute for Econometrics, (2016).

[5] Alan Julian Izenman, *Modern Multivariate Statistical Techniques*, Springer, (2013), 553–575.

[6] James D. Hamilton, *Time Series Analysis*, Princeton Univ. Press, (1994), 1–72.

[7] Peter J. Brockwell, Richard A. Davis, *Time Series: Theory and Methods*, Second Edition, Springer, (1991), 1–39, 77–110.

[8] Bryan P. Rynne, Martin A. Youngson, *Linear Funtional Analysis*, Second Edition, Springer, (2000), 104–111.

[9] Pennstate University *STAT 510: Applied Time Series Analysis* `https://online.stat.psu.edu/stat510/`

[10] Josep Vives i Santa Eulalia *Notes of the course in Time Series*, (2023).