DIPLOMARBEIT

Four Robust Filtering Algorithms and Their Application to Heart-rate Variability in Diabetes

ausgeführt am Institut für
Statistik und Wahrscheinlichkeitstheorie
der Technischen Universität Wien

unter der Anleitung von
O.Univ.-Prof. Dipl.-Ing. Dr. techn. Rudolf DUTTER

durch
Charles-Edouard CADY
1140 Wien, Breitenseerstrasse 76-80/6/3

Wien, 1. Mai 2005

Unterschrift

Thanks!

The author would like to thank his supervisor Professor R. Dutter for his constant support and advice during this work, Dipl. Ing. Berhard Spangl for his patience and clear explanations, his mother for spotting out a lot of the spelling mistakes in this report, the creators of Vim and LaTeX for producing efficient typesetting tools and the Eristoff and Ottakringer companies for their moral support. This report is dedicated to Dr. Eric Chojnacki (IRSN, Cadarache (France)) for introducing me to the wonders of statistics.

Contents

List of Figures

Chapter 1

Introduction

– " *Si parmi ces erreurs il s'en trouve que l'on juge trop grandes pour être admises, alors on rejettera les équations qui ont produit ces erreurs, comme venant d'expériences trop défectueuses, et on déterminera les inconnues par le moyen des équations restantes, qui alors donneront des erreurs beaucoup moindres.* "

in A. M. Legendre, Nouvelles méthodes pour la détermination des orbites des comètes (p. 74), 1806

Translation of the above:
– "*If among these errors there are some which we consider too large to be admissible, then we will reject the equations which produced them as coming from too faulty experiments, and we will determine the unknowns with the remaining equations, which will then give much smaller errors.*"

This report deals with various methods to remove outliers from time series and their application to the measurement of heart rate variability in diabetes. Four algorithms will be presented successively: (1) the repeated median filter, (2) the biweight filter, (3) the biweight filter cleaner and (4) the repeated median cleaner. The purpose of this introduction is to give an idea of the motivations behind the development of such algorithms. In fact, only the last two algorithms are of practical interest but they are derived from the first two which is why we devoted two chapters to their analysis.

There are three main conceptual problems that we faced when dealing with outliers: (1) the actual definition of an outlier, (2) what one should do with outliers and (3) when an outlier should lead to a change in the model.

1. The definition of an outlier. Every statistician can give *his* explanation of the concept, but not only does this explanation vary from one person to another but there is also no commonly accepted rigorous definition. The Wikipedia encyclopedia defines an outlier to be "*a single observation far away from the rest of the data.*" The problem is, what does "far away" mean? This definition suggests that the rest of the data complies to a

certain pattern - a model. It is therefore meaningless to speak of outliers without having defined a model (at least a rough one) first. For example, if we are given a random variable $\mathsf{X} \sim \mathcal{N}(\mu,\ \sigma^2)$, then an observation of this variable can be said to be an α-outlier if it lies outside the $1-\alpha$-quantile of $\mathcal{N}(\mu,\ \sigma^2)$. However, such a definition supposes that one knows how the variable is distributed, which is quite rare in the study of time series.

2. What one should do with these outliers: should we simply remove the outliers or try to replace them with more *pertinent values*? This report presents two types of algorithms: data filters and data cleaners. The *data filter* just replaces the time series by a robust estimation of that time series. However, it might be that most values in the time series are all right and do not need changing. This motivates the implementation of the *data cleaner*: it uses a data filter to detect possible outliers and then replaces them with a more *likely value*, leaving most of the time series unchanged. A naïve question could be: "Why bother? Why not just remove the problematic value from the data set?". The first consequence of removing an outlier is to reduce the size of the data set and affects the estimation of the distribution parameters. For example, variances will be underestimated. Apart from reducing the size of the the data set, this method has the major drawback of being equivalent to augmenting the model with a 0/1 variable where a 1 is used to denote the time point and 0's elsewhere. This obviously privileges one particular time point which does not make any sense in general. However, if a significant event occurs at this time point, then the addition of this 0/1 variable is justified, which leads me to the third problem we encountered.

3. Should the model be changed? An outlier might be precisely *the* interesting value of a data set: for example, if one records seismic vibrations to detect an earthquake, a sudden jerk in the curve is exactly what one is looking for. This is why we spoke earlier of "pertinent value" or "likely value": an outlier is a value which is not pertinent for the model under consideration, but not necessarily wrong. This is another justification for using a data cleaner instead of a data filter: we leave those values which comply to the model and change the others.

Notation

General Symbols

Concept	Symbol	Example
"Defined as"	$:=$	$a := b$
Binomial coefficients		$\binom{n}{p}$
Closed interval	[]	$[a, b]$
Semi-open intervals	]] , [[	$]a, b]$, $[a, b[$
Open intervals	][	$]a, b[$
Infinity	$-\infty, +\infty$	$\forall x \in \mathbb{R}, -\infty < x < +\infty$
Pair of numbers	$(,)$	(a, b)

Sets

Concept	Symbol	Example
Set	capital, roman	A
Indicator function	$\mathbb{1}$	$\mathbb{1}_A : \begin{cases} E & \rightarrow & \{0,1\} \\ x & \mapsto & 1, x \in A, 0 \text{ otherwise} \end{cases}$
Cardinal	$\lvert . \rvert$	$\lvert\{1,2,3\}\rvert = 3$
Set of integers	$\mathbb{N}$	$\mathbb{N} := \{0, 1, 2, \cdots\}$
Set of signed integers	$\mathbb{Z}$	$\mathbb{Z} := \mathbb{N} \cup -\mathbb{N}$
Set without 0	A^*	$\mathbb{N}^* := \{1, 2, \cdots\}$
Set of rational numbers	$\mathbb{Q}$	$\mathbb{Q} := \{p/q : p \in \mathbb{Z} \wedge q \in \mathbb{N}^*\}$
Set of real numbers	$\mathbb{R}$	$\mathbb{R} := \overline{\mathbb{Q}}$
Set of complex numbers	$\mathbb{C}$	$\mathbb{C} := \mathbb{R} + \sqrt{-1}\mathbb{R}$
Probability space	$(\Omega, \mathfrak{A}, P)$	
σ-algebra	$\mathfrak{A}$	
Real Borel σ-algebra	$\mathfrak{B}$	

Linear Algebra

Concept	Symbol	Example
Vector	Bold-face, lower-case	$\boldsymbol{v} = (v_1, \cdots, v_n)$
Matrix	Bold-face, capital	$\mathbf{M}$
Trace of a matrix	$\mathrm{tr}\,()$	$\mathrm{tr}\,(\mathbf{M})$
Determinant of a matrix	$\det()$	$\det(\mathbf{A})$
Transpose of a vector or matrix	$^\intercal$	$\mathbf{A}^\intercal$
Scalar product	$\langle , \rangle$	$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$
Norm	$\lVert \rVert$	$\lVert \boldsymbol{x} \rVert$

Numbers and Functions

Concept	Symbol	Example
Real numbers	Letters between t and y as well as Greek letters	t, x, γ
Integers	Letters between i and p, except o	i, j, k
Complex numbers	z	z
Imaginary number `i`	Lower-case letter, courier	$\mathtt{i} := \sqrt{-1}$
Real part of a complex	Re	Re z
Imaginary part of a complex	Im	Im z
Sequence	$(a_n)_{n\in\mathbb{N}}$	$(a_n)_{n\in\mathbb{N}} := (0, 1, 2, \cdots)$
Integer intervals	$[\![a, b]\!]$	$[\![0, n]\!] := \{0, 1, \cdots, n\}$

Random Variables

Concept	Symbol	Example
Random variable	courier, except `i`	X, t
Realisation of a random variable	not courier	x
Expectation of a random variable	$\mathbb{E}$	$\mathbb{E}\mathsf{X}$
Variance of a random variable	$\mathbb{V}$	$\mathbb{V}\mathsf{Y}$
Covariance matrix	$\mathbf{cov}$	$\mathbf{cov}(\mathsf{X}, \mathsf{Y})$
Bernoulli distribution	$\mathcal{B}(p)$	$\mathsf{x} \sim \mathcal{B}(p) \Rightarrow P(\mathsf{x} = 0) = 1 - P(\mathsf{x} = 1) = 1 - p$
Normal distribution	$\mathcal{N}(\mu,\ \sigma^2)$	$\mathsf{X} \sim \mathcal{N}(\mu,\ \sigma^2) \Rightarrow \mathbb{E}\mathsf{X} = \mu, \mathbb{V}\mathsf{X} = \sigma^2$
Distribution of a random variable	$\mathcal{L}(\ )$	$\mathcal{L}(\mathsf{x}) = \mathcal{N}(0,\ 1)$

Note that the formatting can be combined: for example, $\mathbf{z}$ would denote a vector of complex random numbers.

Chapter 2

Mathematical Context

In this chapter, we present two complementary models used to represent data sets: the autoregressive model, which is meant to produce a time series ressembling a real data set, and the additive outlier model, which adds outliers to a previously defined time series.

2.1 The Additive Outlier Model

In the introduction we said an outlier was a value not pertinent for the model under consideration. This implies that there exists such a model. One model commonly used to describe a time series, and the one which we will use throughout this text, is the additive outlier model. We now need to define what a time series is, but as it is a particular stochastic process, we will define what a stochastic process is first.

Definition 1 *(Stochastic process)*

- $(\Omega, \mathfrak{A}, P)$ is a probability space
- I is a non-empty ordered set called *index set*
- $(\gamma, \mathfrak{G})$ is a measure space called *state space*
- $(\mathbb{R}, \mathfrak{B})$ is the set of real numbers with its Borel σ-algebra
- $\mathsf{X} : \begin{cases} \Omega \times I & \to & \gamma \\ (\omega, t) & \mapsto & \mathsf{X}_t(\omega) \end{cases}$

If

$\forall t \in I, \mathsf{X}_t : (\Omega, \mathfrak{A}) \to (\gamma, \mathfrak{G})$ is a random variable, then $(\Omega, \mathfrak{A}, P, (\mathsf{X}_t)_{t \in I})$ is called *stochastic process*.

$\diamondsuit$

Example: The simplest example of a stochastic process is that of the one-dimensional random walk. At each time-point, one either stays put, goes right one unit or left one unit. In this example, we suppose we have a probability space $(\Omega, \mathfrak{A}, P)$. Let $(p, q) \in ([0, 1])^2$ such that $p + q \leq 1$. Then, in the previous definition, we take:

1. $I = \mathbb{N}$,
2. $(\gamma, \mathfrak{B}) = (\mathbb{Z}, \mathfrak{B}(\mathbb{Z}))$,
3. X such that $P(\mathsf{X} = 1) = p$, $P(\mathsf{X} = -1) = q$ and $P(\mathsf{X} = 0) = 1 - (p + q)$

We can now define the basic object used in all of this report:

Definition 2 *(Time series)*

If $I \subseteq \mathbb{Z}$ in the definition of a stochastic process, then $(\mathsf{X}_t)_{t \in I}$ is called *time series*.

♢

The random variables we consider will usually be normally distributed, which motivates the following definition:

Definition 3 *(Gaussian process)*

- $(\Omega, \mathfrak{A}, P)$ is a probability space
- $I \subseteq \mathbb{R}$ is non-empty
- $(\mathbb{R}, \mathfrak{B})$ is the set of real numbers with its Borel σ-algebra
- $\mathsf{X} : \left\{ \begin{array}{ccc} \Omega \times I & \to & \gamma \\ (\omega, t) & \mapsto & \mathsf{X}_t(\omega) \end{array} \right.$ is a stochastic process

If $\forall n \in \mathbb{N}^*, (t_1, \cdots, t_n) \in I^n, t_1 < \cdots < t_n, \mathsf{X}_{t_1} \otimes \cdots \otimes \mathsf{X}_{t_n}$ is joint normal, then $(\Omega, \mathfrak{A}, P, (\mathsf{X}_t)_{t \in I})$ is called *Gaussian process*.

♢

In the special case of time series, Gaussian processes take on the following form:

Definition 4 *(Gaussian time series)*

If $I \subseteq \mathbb{Z}$ in the definition of a Gaussian process, then $(\mathsf{X}_t)_{t \in I}$ is called *Gaussian time series*.

♢

A case which often arises in practice is the case where the distribution is "constant" with time, i.e. the process is stationary.

Definition 5 *(Stationary process)*

Let $(\Omega, \mathfrak{A}, P, (\mathsf{X}_t)_{t\in I})$ be a stochastic process. If, $\forall n \in \mathbb{N}^*, \forall (t_1, \cdots, t_n) \in I^n$ such that $t_1 < \cdots < t_n$ and $\forall s \in I$, the distribution of $\mathsf{X}_{t_1+s} \otimes \cdots \otimes \mathsf{X}_{t_n+s}$ is the same as the distribution of $\mathsf{X}_{t_1} \otimes \cdots \otimes \mathsf{X}_{t_n}$, then $(\Omega, \mathfrak{A}, P, (\mathsf{X}_t)_{t\in I})$ is said to be a *stationary process*.

♢

Example: A process with trend can be defined by: $\varepsilon_t \sim$ iid $\mathcal{N}(0,\ \sigma^2)$, $\delta > 0$ and $\mathsf{X}_t = \delta t + \varepsilon_t$. As $\mathbb{E}\mathsf{X}_t = \delta t$ which depends on t, it is not a stationary process.

Definition 6 *(Trajectory)*

Given a stochastic process X and using the notation of the definition, $\forall \omega \in \Omega, t \mapsto \mathsf{X}_t(\omega)$ is called *trajectory*.

♢

We will seldom use the somewhat cumbersome notation of the definition of a stochastic process and will often refer to $(\mathsf{X}_t)_{t\in\mathbb{N}}$ as a time series without specifying the underlying σ-algebras. Unless otherwise specified, the state space will always be $\mathbb{R}$ and $\mathfrak{G} = \mathfrak{B}$. This point being clarified, we will now expose the model used throughout these pages.

Definition 7 *(Additive outlier model)*

All stochastic processes in this definition have same index set $(\Omega, \mathfrak{A}, P)$ and state space $(\Gamma, \mathfrak{G})$. Let $\sigma_b \in \mathbb{R}^+$.

- $(\mathsf{a}_t)_{t\in\mathbb{N}}$ is a stochastic process such that $\forall t \in \mathbb{N}, \mathsf{a}_t \sim \mathcal{B}(\gamma)$ where $\gamma \in [0, 1]$ is called *degree of contamination* and the a_t are independent
- $(\mathsf{b}_t)_{t\in\mathbb{N}}$ is a stochastic process such that $\forall t \in \mathbb{N}, \mathsf{b}_t \sim \mathcal{N}(0,\ \sigma_b^2)$ and the b_t are independent
- $(\mathsf{x}_t)_{t\in\mathbb{N}}$ is a stationary time series called *core process*
- $(\mathsf{v}_t)_{t\in\mathbb{N}}$ is a time series such that $\forall t \in \mathbb{N}, \mathsf{v}_t = \mathsf{a}_t\mathsf{b}_t$ called *contaminating process*.

The time series $(\mathsf{y}_t)_{t\in\mathbb{N}}$ defined by $\forall t \in \mathbb{N}, \mathsf{y}_t := \mathsf{x}_t + \mathsf{v}_t$, is said to follow an *additive outlier model* with degree of contamination γ.

♢

The additive outlier model is a stationary core process to which occasional outliers have been added. γ represents the amount of noise that has been added to the core process, $\gamma = 0$ corresponding to no contamination and $\gamma = 1$ to a

contamination at every observation. This model will be used to test the behavior of the various algorithms presented here.

Another aspect of the additive outlier model is that it allows us to test an algorithm: indeed, we can generate what we will call a *core process*, i.e. a process with contamination $\gamma = 0$, and add contamination to this process. The aim of all the algorithms presented in this report is, given the contaminated process, to recover the underlying core process. Therefore, we can test the efficiency of an algorithm by presenting it with a contaminated process and comparing its output with the original core process.

Example: On Figure 2.1 page 19, we generated a core process X_t by means which will be explained in the next section.

Then we created the contaminating process V_t with $\gamma = 0.1$ (Figure 2.2 page 19).

Putting everything together, we get a contaminated process which is the sum of the two previous processes. To get a better idea of the influence of the contaminating process, we plotted on Figure 2.3 page 20 the core process in black and the occasional outliers in red.

Having now solved the problem of how to model outliers, we will now address the question of how to model the core process.

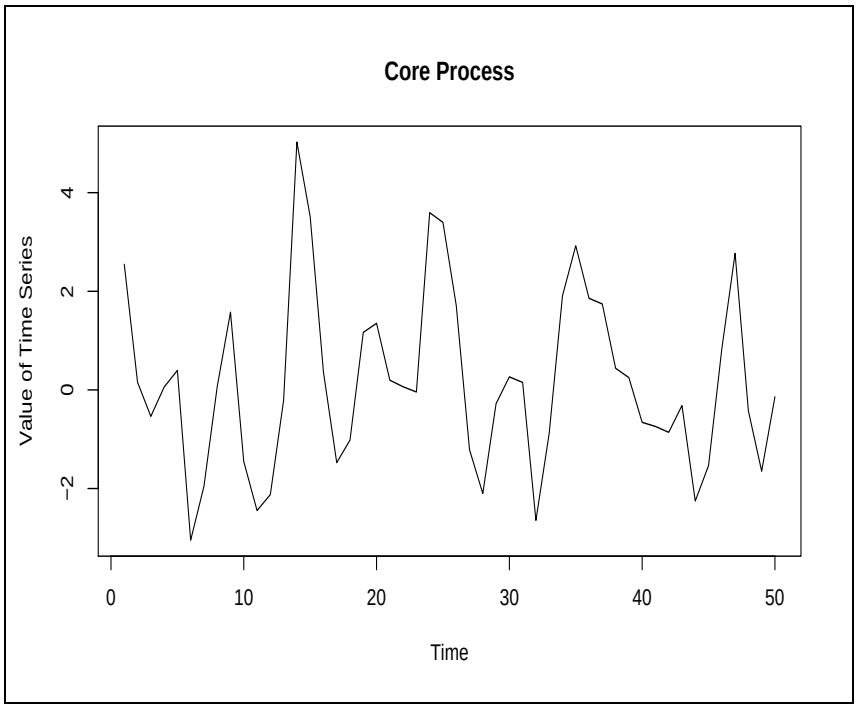

Figure 2.1: Plot of the core process

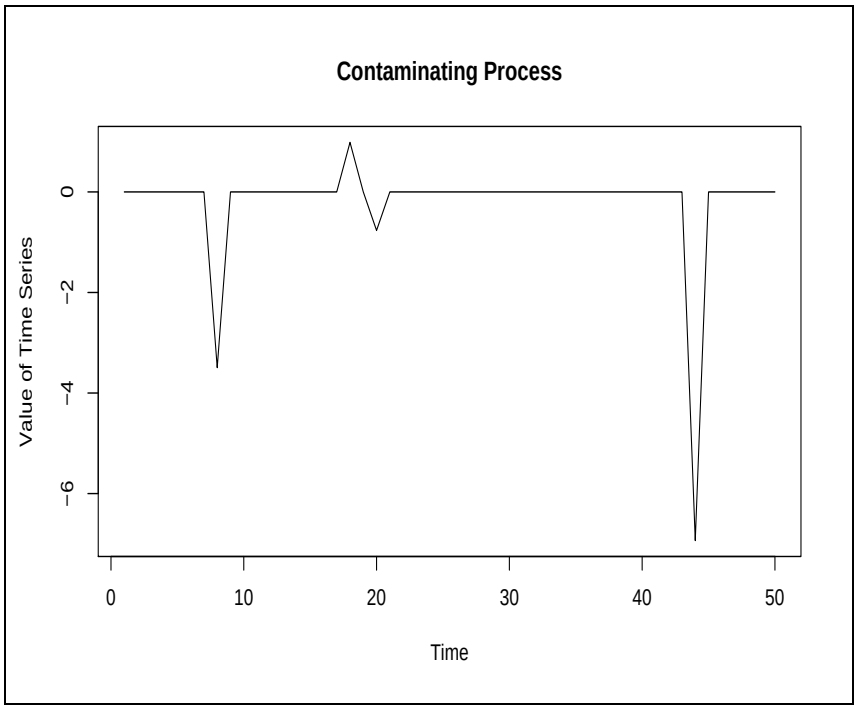

Figure 2.2: Plot of the contaminating process

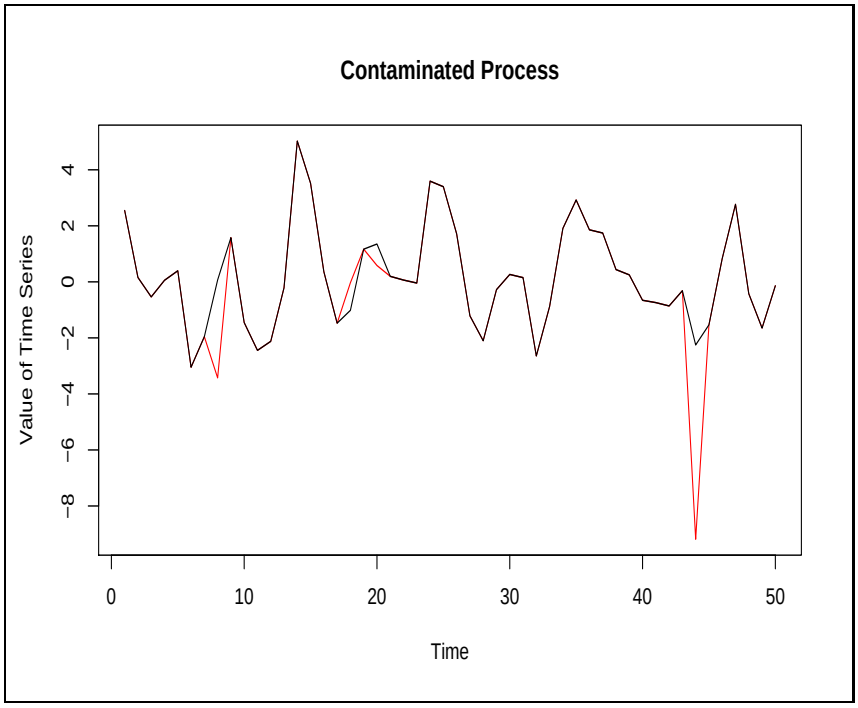

Figure 2.3: Plot of the contaminated process

2.2 Auto-regressive Models

Finding a model for the core process is difficult because the processes under consideration can be of very different nature. Indeed, the problem of removing outliers can arise in very different contexts: medical science, physics, economy, etc. Time series can be modeled in various ways. A model often used is the auto-regressive model. The time series at a given position only depends on a weighted sum of its previous values and a "white noise" error term.

Definition 8 *(Auto-regressive model)*
Given

- $p \in \mathbb{N}^*, \sigma \in \mathbb{R}^+, (\lambda_1, \cdots, \lambda_{p-1}) \in \mathbb{R}^{p-1}, \lambda_p \in \mathbb{R}^*, \mu \in \mathbb{R}$
- $\mathsf{x}_0, \cdots, \mathsf{x}_{p-1}$ are constants
- $(\varepsilon_t)_{t\in\mathbb{N}}$ is a time series such that $\forall t \in \mathbb{N}, \varepsilon_t \sim \mathcal{N}(0,\ \sigma^2)$, the ε_t being independent
- $(\mathsf{x}_t)_{t\in\mathbb{N}}$ is a time series such that $\forall t \in \mathbb{N}, \mathsf{x}_t - \mu = \sum_{i=1}^{p} \lambda_i(\mathsf{x}_{t-i} - \mu) + \varepsilon_t$

$(\mathsf{x}_t)_{t\in\mathbb{N}}$ follows an auto-regressive model of order p with mean μ. ♢

An auto-regressive model of order p is often denoted by AR(p).

Example: Figure 2.4 page 23 is that of an AR(2) process with $\lambda_1 = 0.1$ and $\lambda_2 = -0.2$.

We only generated samples from AR(2) processes. For such processes, there exists an explicit formulation of the time series if we suppose that $\forall t \in \mathbb{N}, \varepsilon_t = 0$.

Indeed, consider for $(a, b) \in \mathbb{R}^* \times \mathbb{R}$ the set E of all sequences of real numbers $(u_n)_{n\in\mathbb{N}}$ such as:

$$u_{n+2} = au_{n+1} + bu_n$$

E is a 2-dimensional vector space. To describe it, we need a basis. Lets try with $(u_n)_{n\in\mathbb{N}} := (r^n)_{n\in\mathbb{N}}$ with $r \neq 0$. This leads to solving the equation

$$r^{n+2} = ar^{n+1} + br^n$$

Dividing by $r^n \neq 0$ yields

$$r^2 = ar + b$$

Let $\Delta := a^2 + 4b$.

- For $\Delta > 0$, $r_1 := \frac{a-\sqrt{a^2+4b}}{2}$ and $r_2 := \frac{a+\sqrt{a^2+4b}}{2}$

  $$u_n = \frac{1}{\sqrt{a^2+4b}}\left[(r_2u_0 - u_1)r_1^n + (u_1 - r_1u_0)r_2^n\right]$$

  with $(u_0, u_1) \in \mathbb{R}^2$.

- For $\Delta = 0$, let us first notice that if $(r^n)_{n\in\mathbb{N}}$ is a solution, so is $(nr^n)_{n\in\mathbb{N}}$.

$$u_n = (1-n)\left(\frac{a}{2}\right)^n u_0 + n\left(\frac{a}{2}\right)^{n-1} u_1$$

  with $(u_0, u_1) \in \mathbb{R}^2$.

- For $\Delta < 0$, let $\theta := \arctan \frac{\sqrt{-(a^2+4b)}}{a}$.

$$u_n = \sqrt{-b}^n \left(u_0 \cos(n\theta) + \left(\frac{\sqrt{1+\theta^2}}{\theta\sqrt{-b}} u_1 - \frac{u_0}{\theta} \right) \sin(n\theta) \right)$$

  with $(u_0, u_1) \in \mathbb{R}^2$.

Therefore, the choice of the coefficients of the AR(2) model is of some importance. One usually wants the model to oscillate to some degree and not diverge exponentially so one usually chooses the coefficients so that $a^2 + 4b \leq 0$.

To test the algorithms, we will often use a Gaussian AR(2) process. To generate it, we will simulate two random values x_1 and x_2 by a random number generation algorithm (see appendix C) and a vector of "noise" $\boldsymbol{\varepsilon}$, also by a random number generator. Then we build the time series recursively by:

$$\mathsf{x}_t = a\mathsf{x}_{t-1} + b\mathsf{x}_{t-2} + \varepsilon_t$$

We now have enough background to review the first algorithm: the repeated median algorithm.

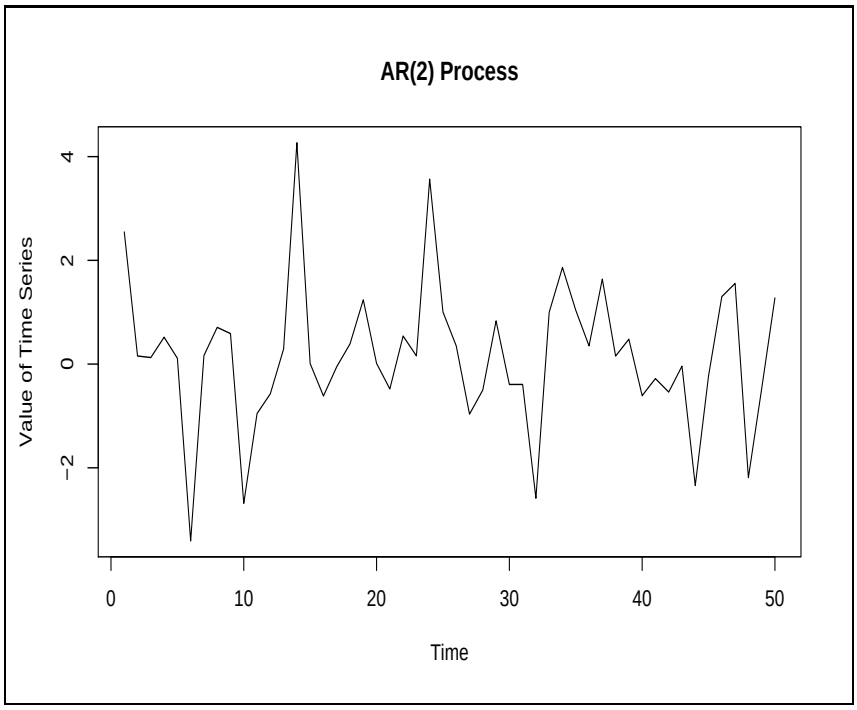

Figure 2.4: Plot of the process $x_t = 0.1x_{t-1} - 0.2x_{t-2} + \varepsilon_t$

Chapter 3

Repeated Median

The first algorithm that we will expose is the repeated median algorithm. We will first motivate and explain the fitting of a sinusoid through a time-series and then generalize this approach to build a robust Fourier transformation.

3.1 Motivation

We previously said that the simulations used to qualify the algorithms would be based on an autoregressive model $\mathsf{x}_{p+2} = \alpha\mathsf{x}_{p+1} + \beta\mathsf{x}_p$ with $(\alpha, \beta) \in \mathbb{R}^2$ such as $\alpha^2 + 4\beta \leq 0$ and $p \in I \subseteq \mathbb{N}$, index set. The general solution for an AR(2) model with $\alpha^2 + 4\beta < 0$ and $\forall p \in I, \varepsilon_p = 0$ being

$$\exists\theta \in \mathbb{R}_+^*, \forall p \in I, : \mathsf{x}_p = \sqrt{-\beta}^p \left[\mathsf{x}_0 \cos(p\theta) + \left(\frac{\sqrt{1+\theta^2}}{\theta\sqrt{-\beta}}\mathsf{x}_1 - \frac{\mathsf{x}_0}{\theta} \right) \sin(p\theta) \right]$$

it makes sense to investigate a procedure which will give an exact fit for a time series $(\mathsf{x}_p)_{p\in I}$ such as $x_p = a\cos(\omega p) + b\sin(\omega p)$. If $\beta = -1$ and $-2 < \alpha < 2$, we will expect to find

$$\mathsf{a} = \mathsf{x}_0, \mathsf{b} = \left(\frac{\sqrt{1+\theta^2}}{\theta}\mathsf{x}_1 - \frac{\mathsf{x}_0}{\theta} \right), \omega = \theta = \arctan\left(\frac{\sqrt{4-a^2}}{a} \right)$$

We now suppose the observed process $(\mathsf{y}_p)_{p\in I}$ follows an additive outlier model $\forall p \in I, \mathsf{y}_p = \mathsf{x}_p + \mathsf{v}_p$ with $P(\mathsf{v}_p = 0) = 1 - \gamma$. If $\omega \neq 0$ is known, then the coefficients a and b can be found using two different observations y_i and y_j:

$$\begin{bmatrix} y_i \\ y_j \end{bmatrix} = \begin{bmatrix} \cos(\omega i) & \sin(\omega i) \\ \cos(\omega j) & \sin(\omega j) \end{bmatrix} \begin{bmatrix} a_{ij} \\ b_{ij} \end{bmatrix}$$

If $\gamma = 0$, we will have $\forall (i,j) \in I^2, a_{ij} = a, b_{ij} = b$. Otherwise, let

$$\mathbf{M} := \begin{bmatrix} \cos(\omega i) & \sin(\omega i) \\ \cos(\omega j) & \sin(\omega j) \end{bmatrix}$$

$\det(\mathbf{M}) = \cos(\omega i)\sin(\omega j) - \sin(\omega i)\cos(\omega j) = \sin[\omega(i-j)]$. The system therefore has a solution if, and only if, i and j are such as $i - j \neq \frac{k\pi}{\omega}, k \in \mathbb{Z}$. If this condition holds, then i and j will be said to be in *normal position*.

We now have two sets of coefficients a_{ij} and $\mathsf{b}_{ij}, (i,j) \in I^2$. To get an estimate of a and b out of them, we will build the *repeated median*:

$$\widehat{\mathsf{a}} := MED_{j\in I} MED_{i\in I}(\mathsf{a}_{ij})$$

$$\widehat{\mathsf{b}} := MED_{j\in I} MED_{i\in I}(\mathsf{b}_{ij})$$

The robust estimate of the signal $(\mathsf{x}_p)_{p\in I}$ will therefore be:

$$\forall p \in I, \widehat{\mathsf{x}_p} := \widehat{\mathsf{a}}\cos(\omega p) + \widehat{\mathsf{b}}\sin(\omega p)$$

Although this method works extremely well if $\forall p \in I, x_p = a\cos(\omega p) + b\sin(\omega p)$ (for $\gamma = 0$, the signal is recovered without error), it needs to be generalized to a wider class of time series.

In order to apply the same kind of method as above, we will write a time series as a sum of trigonometric functions. From now on, we will be dealing with time series indexed by a finite, non-empty set I with $n := |I| \geq 1$. Without loss of generality we will suppose that $I = [\![0, n-1]\!]$. All functions will be defined on I and $\mathcal{F}(I,\mathbb{R})$ will denote the set of all functions $f : I \to \mathbb{R}$. First we will define a scalar product

$$\langle , \rangle : \left\{ \begin{array}{ccc} (\mathcal{F}(I,\mathbb{R}))^2 & \to & \mathbb{R} \\ ((x_p)_{p\in I}, (y_p)_{p\in I}) & \mapsto & \sum_{p=0}^{n-1} x_p y_p \end{array} \right.$$

Let $\|.\|$ be the norm induced by $\langle ., . \rangle$. $E := \mathcal{F}(I,\mathbb{R})$ is an n-dimensional $\mathbb{R}$-vector-space. We will therefore need n independent functions to get a basis for E. $\forall k \in [\![0, n-1]\!], \omega_k := \frac{2k\pi}{n}$.

We want to build a basis of E out of $(\cos(\omega_k p))_{p\in I}$ and $(\sin(\omega_k p))_{p\in I}$. Let $(k,k') \in I^2$ and $SP(k,k') := \langle (\cos(\omega_k p))_{p\in I}, (\cos(\omega_{k'} p))_{p\in I} \rangle$.

$$\begin{aligned} SP(k,k') &= \sum_{p=0}^{n-1} \cos(\omega_k p)\cos(\omega_{k'} p) \\ &= \frac{1}{2}\sum_{p=0}^{n-1} [\cos((\omega_k + \omega_{k'})p) + \cos((\omega_k - \omega_{k'})p)] \end{aligned}$$

Let

$$r_1 := \mathrm{e}^{\frac{2\mathrm{i}(k+k')\pi}{n}}, S_1 := \sum_{p=0}^{n-1} r_1^p$$

and

$$r_2 := \mathrm{e}^{\frac{2\mathrm{i}(k-k')\pi}{n}}, S_2 := \sum_{p=0}^{n-1} r_2^p$$

$$SP(k,k') \quad = \quad \Re(S_1 + S_2)$$

To calculate S_1 and S_2, we first need to investigate when r_1 and r_2 (respectively) are equal to 1.

$$(r_1 = 1) \Leftrightarrow \left(\frac{k+k'}{n} = \ell \in \mathbb{Z}\right)$$

As $k, k' \geq 0$, this reduces to:

$$(r_1 = 1) \Leftrightarrow \left(\frac{k+k'}{n} = \ell \in \mathbb{N}^*\right)$$

In other words, $k + k'$ must not be a multiple of n. A necessary and sufficient condition to have $r_1 \neq 1$ is therefore $0 < k, k' < \frac{n}{2}$. This condition is also sufficient to prevent $r_2 = 1$.

We find that

$$SP = \begin{cases} \frac{n}{2}, & k = k' \\ 0, & k \neq k' \end{cases}$$

Hence the set of vectors $e_k := \sqrt{\frac{n}{2}}(\cos(\omega_k p))_{p\in I}$, for $0 < k < \frac{n}{2}$ is a set of orthonormal vectors: they are therefore independent. We can also add the vector $e_0 := \sqrt{\frac{n}{2}}$ because it is orthogonal to all the others and its norm is 1. If n is even, we can add an extra vector corresponding to $k = \frac{n}{2}$: its $e_{\frac{n}{2}} := \sqrt{n}(\cos(\omega_{\frac{n}{2}} p))_{p\in I} = (\cos(\pi p))_{p\in I} = ((-1)^p)_{p\in I}$.

For $(\sin(\omega_k p))_{p\in I}$, by a similar calculation we find the set of vectors: $f_k := \sqrt{\frac{n}{2}}(\sin(\omega_k p))_{p\in I}$ for $0 < k < \frac{n}{2}$.

Moreover, $\forall (k,k') \in I^2, \langle (\cos(\omega_k p))_{p\in I}, (\sin(\omega'_k p))_{p\in I} \rangle = 0$.

All in all, we have:

- If n is even:

$$\left.\begin{array}{ll} e_0, \cdots, e_{\frac{n}{2}} & \Rightarrow \frac{n}{2}+1 \text{ vectors} \\ f_1, \cdots, f_{\frac{n}{2}-1} & \Rightarrow \frac{n}{2}-1 \text{vectors} \end{array}\right\} \Rightarrow n \text{ vectors.}$$

- If n is odd:

$$\left.\begin{array}{ll} e_0, \cdots, e_{\frac{n-1}{2}} & \Rightarrow \frac{n+1}{2} \text{vectors} \\ f_1, \cdots, f_{\frac{n-1}{2}} & \Rightarrow \frac{n-1}{2} \text{ vectors} \end{array}\right\} \Rightarrow n \text{ vectors.}$$

We can now write any time series in E on the orthonormal basis we have just built. The result is called *Fourier representation of a finite time series.*

$$\forall (x_p)_{p\in I} \in E, \forall p \in I, x_p = a_0 + \sum_{0<k<\frac{n}{2}} \left(a_k \cos(\omega_k p) + b_k \sin(\omega_k p)\right) + (-1)^p a_{\frac{n}{2}}$$

where the last term is included only if n is even, and for $0 < k < \frac{n}{2}$,

$$a_k := \frac{2}{n} \sum_{p=0}^{n-1} x_p \cos(\omega_k p)$$

$$b_k := \frac{2}{n} \sum_{p=0}^{n-1} x_p \sin(\omega_k p)$$

$$a_0 := \frac{1}{n} \sum_{p=0}^{n-1} x_p$$

$$a_{\frac{n}{2}} := \frac{1}{n} \sum_{p=0}^{n-1} x_p (-1)^p$$

3.2 Repeated Median Algorithm

This algorithm is presented by Tatum and Hurvich in [19]. The Fourier representation of a function is just another way of writing the same function. Hence if we calculate the Fourier coefficients by the method explained above and we build the sum of the trigonometric functions weighted with the Fourier coefficients, we will end up with exactly the same time series as we started with. If we want to remove outliers from a time series, the Fourier representation is useless as we will just be writing the same time series in a different form (and thus getting all the outliers back). Fortunately, we do not need to throw away everything we just did: instead of calculating the Fourier coefficients directly, we will use a "robustified" version of these coefficients with the repeated median. We will do this exactly the same way as we did it for one frequency: we will successively fit a sinusoid at each frequency and then substract that frequency from the rest of the data to get a more precise result.

At each frequency ω_k with $0 < k < n/2$, we will build as before the sets of coefficients

$$a_{kij} := \frac{x_i \sin(\omega_k j) - x_j \sin(\omega_k i)}{\sin(\omega_k (j-i))}, (i,j) \in I^2, i \neq j$$

and

$$b_{kij} := \frac{x_j \cos(\omega_k i) - x_i \cos(\omega_k j)}{\sin(\omega_k (j-i))}, (i,j) \in I^2, i \neq j$$

The estimate for a_0 will be the median of the time series. In order to have one coefficient less to calculate and get all observations in normal position, we will take a time series of prime length. This can always be done by extracting two overlapping sequences of prime length (the greatest prime $n' < n$) from the original time series. The first sequence consists of the first n' observations and the second sequence of the last n' observations. We will apply the algorithm separately to each subsequence, then recombine them by averaging.

The following theorem can be found in the article by Tatum and Hurvich ([19]).

Theorem 1 *(Breakdown bound of the repeated median filter)*
The repeated median filter has a breakdown point of

$$\frac{(n'-1)/2-1}{n}$$

where n is the length of the time-series to filter and n' is the greatest prime less than or equal to n.

□

To get a more precise evaluation of the coefficients, we will sweep each frequency out after having calculated the corresponding coefficient and the whole algorithm will be run M times (of course, we will not reinitialize the time-series between each run). Intuitively, we can guess that the bigger the coefficient, the more likely it is to contaminate the other coefficients and the less likely it is to be contaminated by the other coefficients. We therefore need to know what coefficients are the biggest in order to treat them first. To do this, we do not need a very precise estimate of the spectrum as we just want to know in what order we will calculate the coefficients. We take the natural estimator as defined in appendix B (page 129) and we smooth it to reduce its variance, the amount of smoothing being determined by AIC_C (as presented page 149).

The first few steps of the algorithm will be:

Separate the time-series into two overlapping subsequences of prime length
for each subsequence **do**
  Build natural periodogram
  Smooth the periodogram
  Order the frequencies from the strongest to the weakest
  $y_t \leftarrow y_t - MED(y_t)$
end for

For each frequency, we will then calculate the two sets of coefficients a_{kij} and b_{kij}:

for $i = 0$ to $n' - 1$ **do**
  for $j = 0$ to $n' - 1$, $j \neq i$ **do**
    $a_{kij} = \frac{x_i \sin(\omega_k j) - x_j \sin(\omega_k i)}{\sin(\omega_k (j-i))}$
    $b_{kij} = \frac{x_j \cos(\omega_k i) - x_i \cos(\omega_k j)}{\sin(\omega_k (j-i))}$
  end for
end for
$a'_k \leftarrow MED_j MED_i a_{ijk}$
$b'_k \leftarrow MED_j MED_i b_{ijk}$

$y_t = y_t - (a'_k \cos(\omega_k t) + b'_k \sin(\omega_k t))$
$a_k^{RM} \leftarrow a_k^{RM} + a'_k$
$b_k^{RM} \leftarrow b_k^{RM} + b'_k$

In the sweeping phase, we remove the residuals from the previous step and update the robust estimate of the coefficients a_k^{RM} and b_k^{RM} ("update" because unless $M = 1$, we already have an estimate from the previous run):

The complete algorithm is presented page 31.

Algorithm 1 Repeated median algorithm

Separate the time-series into two overlapping subsequences of prime length
for each subsequence **do**
  Build natural periodogram
  Smooth the periodogram
  Order the frequencies from the strongest to the weakest
  for $m = 1$ to M **do**
    for k from strongest to weekest frequency **do**
      $y_t \leftarrow y_t - MED(y_t)$
      for $i = 0$ to $n' - 1$ **do**
        for $j = 0$ to $n' - 1$, $j \neq i$ **do**
          $a_{kij} = \frac{x_i \sin(\omega_k j) - x_j \sin(\omega_k i)}{\sin(\omega_k (j-i))}$
          $b_{kij} = \frac{x_j \cos(\omega_k i) - x_i \cos(\omega_k j)}{\sin(\omega_k (j-i))}$
        end for
      end for
      $a'_k \leftarrow MED_j MED_i a_{ijk}$
      $b'_k \leftarrow MED_j MED_i b_{ijk}$
    end for
    $y_t = y_t - (a'_k \cos(\omega_k t) + b'_k \sin(\omega_k t))$
    $a_k^{RM} \leftarrow a_k^{RM} + a'_k$
    $b_k^{RM} \leftarrow b_k^{RM} + b'_k$
  end for
end for

3.3 Results

In this section, we will see the repeated median filter in action. The canvas used for the presentation of the results of all the algorithms will be the same as this one. In each subsection we will detail how the results were obtained and how to interpret them. Four types of results will be analyzed successively: (1) a filtering example, (2) the response of the filter to various contamination levels, (3) a sensitivity curve and (4) the speed of the algorithms.

3.3.1 Filtering Example

Overview

This example is simply the filtering of a contaminated autoregressive time-series. In other words, we generated an AR(2) process and contaminated it with another process. The aim is, given the contaminated process, compare the output of the filter to the core process. A perfect filter would recover the core process exactly. We have to bear in mind that this is just a qualitative way of judging the filter.

Data Set Used

In our C++ implementation of the algorithms, we specify the parameter values of the algorithms in a text file. We will give an example of such a file and comment its various sections.

```
Model parameters

phi_1 0.75
phi_2 -0.5
Seed 4
NbTermes 25
gamma 0.1
variance 20
param_biweight_location 4
param_biweight_regression 6
Max_Nb_Iteration 60
M 2
Scale_param_cleaner 4
Sensitivity_min -40
Sensitivity_max 40
Sensitivity_nb_points 100
Sensitivity_pos 15
Cleaner_lower_bound 2
```

`Cleaner_upper_bound 6`

1. `phi_1 0.75`
   `phi_2 -0.5`

   These are the parameters of the AR(2) process used. In the example file, they correspond to the process defined by: $\mathtt{x}_t = 0.75\mathtt{x}_{t-1} - 0.5\mathtt{x}_{t-2} + \varepsilon_t$ where ε_t is $\mathcal{N}(0,\ 1)$.

2. `NbTermes 25`

   This is just the length of the time series.

3. `gamma 0.1`

   Controls the contamination level (as explained in the definition of the additive outlier model, page 17).

4. `variance 20`

   This is the variance σ_b of the contaminating process (see the definition of the additive outlier model page 17).

5. `param_biweight_location 4`
   `param_biweight_regression 6`

   These are the parameters of the biweight filter algorithm.

6. `Max_Nb_Iteration 60`

   Sets the maximum number of iterations for the recursive procedures.

7. `M 2`

   This is the number of times the algorithm will be performed (corresponds to the M in the algorithm page 31, for example).

8. `Scale_param_cleaner 4`

   Scale parameter used for the flagging procedure in the biweight filter cleaner and repeated median cleaner algorithms.

9. `Sensitivity_min -40`
   `Sensitivity_max 40`
   `Sensitivity_nb_points 100`
   `Sensitivity_pos 15`

   Parameters for the sensitivity curves.

10. `Cleaner_lower_bound 2`
    `Cleaner_upper_bound 6`

    The a and b constants, as defined page 63.

The parameter values we used for this particular result are reproduced on page 35.

Results

The results Figure 3.1 page 35 show us that although the filter is not perfect, it yields reasonably good results. It certainly cannot be used to try and recover a complex core process, but we can use it as reference for future algorithms.

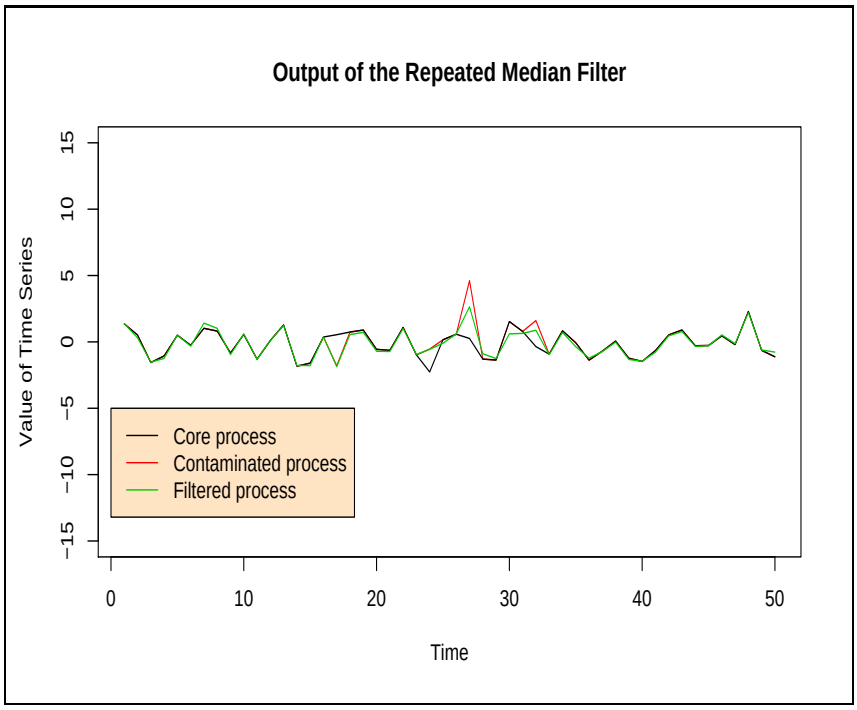

Figure 3.1: Output of the repeated median filter

```
Model parameters

phi_1 0.2
phi_2 -0.2
Seed 2
NbTermes 50
gamma 0.05
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 2
Scale_param_cleaner 4
Sensitivity_min -100
Sensitivity_max 100
Sensitivity_nb_points 100
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

3.3.2 Response to Various Contamination Levels

Overview

Here we will plot a series of boxplots for various contamination levels. At each contamination level, we ran the filter a few times and calculated the logarithm of the residuals. By "residuals" we mean

$$\frac{\|\mathbf{y} - \mathbf{y}^F\|^2}{\|\mathbf{v}\|^2}$$

where $\mathbf{y}$ is the core process, $\mathbf{y}^F$ is the filtered process and $\mathbf{v}$ is the contaminating process. The norm used is the euclidean one. In the case of an ideal filter, the quotient is equal to one. Therefore, the closer the residuals are to one, the better the filter. Of course, the residuals will always be greater than one because the contaminating process is null everywhere except at the contaminated time points, which is not the case of the *observed* residuals $\boldsymbol{y}-\boldsymbol{y}^F$. We ran the repeated median filter fifty times, taking a different seed each time (seeds from 0 to 49), the other parameters being left unchanged.

Data Set Used

Apart from the random number generator seed, we used the parameters on page 37 each time.

Results

The first thing we notice in the results (Figure 3.2 page 37) is that the residuals grow exponentially with the contamination rate. The second is that the variances of these residuals also grow. This is because the variance of $\boldsymbol{v}$ is proportional to γ. However, it seems that up until $\gamma\#0.44$, the residuals do not grow too much, which would tend to confirm that the breakdown bound is indeed $\frac{(47-1)/2-1}{50}$ as predicted by the theorem on the breakdown point of the repeated median filter Theorem 3.2 page 29. In order to get a better view of the response for $\gamma \leq 0.44$, we will do another plot.

In Figure 3.3 page 38, the growth of the residuals does not seem to be exponential, at least. However, from $\gamma\#0.293$ on, the residuals seem a bit large: the filter does not seem to perform quite as well as expected.

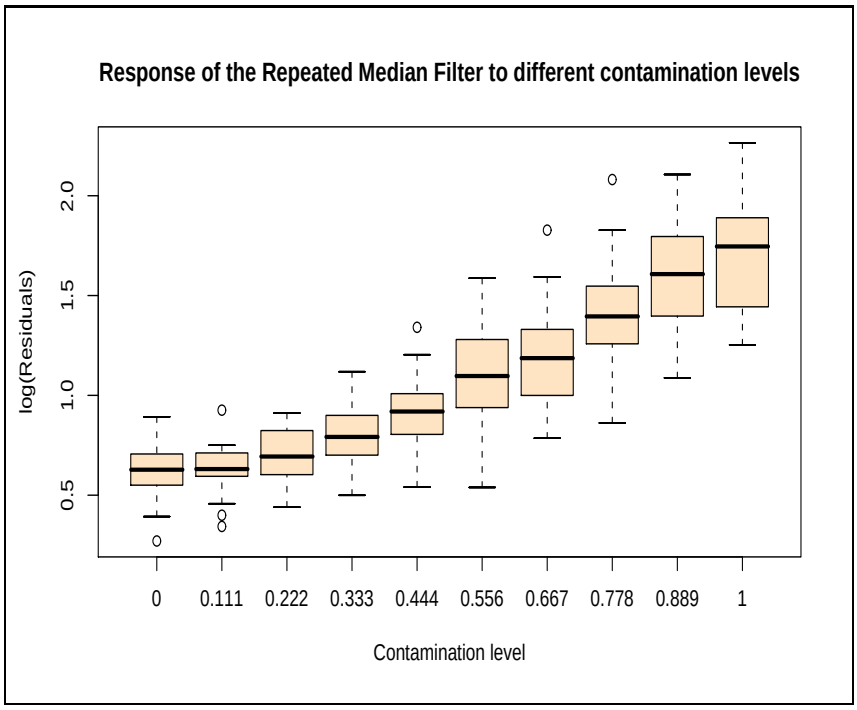

Figure 3.2: Residuals for γ from 0 to 1

```
Model parameters

phi_1 0.75
phi_2 -0.5
Seed 33
NbTermes 50
gamma 0.15
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -15
Sensitivity_max 15
Sensitivity_nb_points 200
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

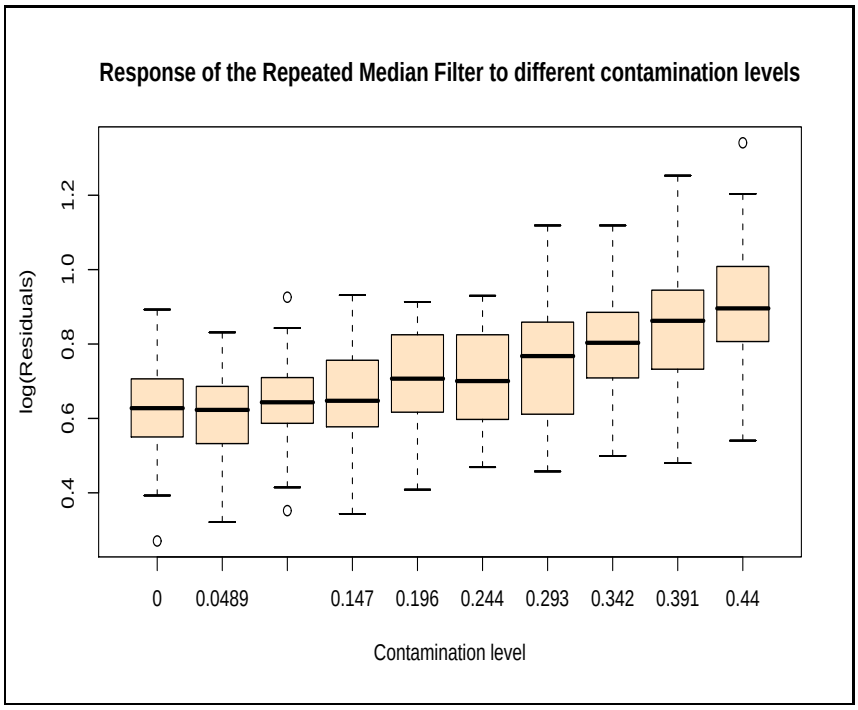

Figure 3.3: Residuals for γ from 0 to 0.44

```
Model parameters

phi_1 0.75
phi_2 -0.5
Seed 33
NbTermes 50
gamma 0.15
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -15
Sensitivity_max 15
Sensitivity_nb_points 200
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

3.3.3 Sensitivity Curve

Overview

A sensitivity curve is constructed as follows: we choose a particular point of the time series, we shift it and then record the output of the filter at this particular time-point. We then plot the output of the filter against the values of the shift. A robust filter will not be too sensitive to extreme values. For a non-robust algorithm, on the contrary, the output will become worse and worse as the shift increases.

Data Set Used

The parameter values for this sensitivity curve can be found on page 40.

Results

Figure 3.4 page 40 shows that although the output of the filter is bounded, the curve is fairly irregular and even if it is vaguely linear in its center, an outlier will continue to exert an influence, no matter how large it is. This is, of course, a major drawback of this algorithm.

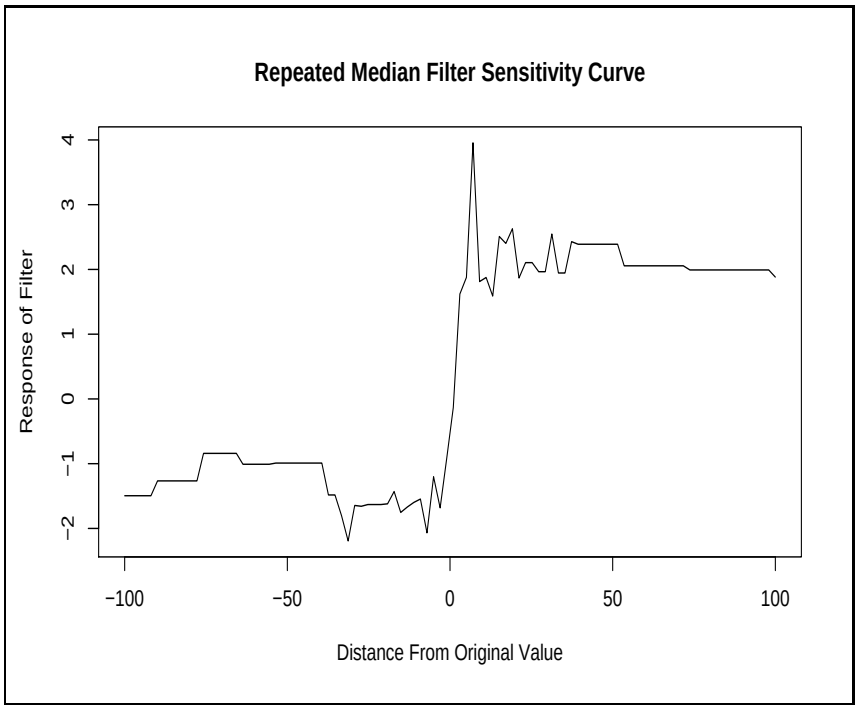

Figure 3.4: Sensitivity curve repeated median filter

```
Model parameters

phi_1 1
phi_2 -0.5
Seed 3141
NbTermes 50
gamma 0
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 2
Scale_param_cleaner 4
Sensitivity_min -100
Sensitivity_max 100
Sensitivity_nb_points 100
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

3.3.4 Speed

How fast is our algorithm running? This subsection presents the tests we performed to assess the speed of the repeated median algorithm.

Overview

As for the residuals, we used the same data set each time, changing only the length of the time series. Using the Linux command "time", we got the user time needed for the execution of our C++ implementation of this algorithm. We then tried to fit a model to the data so as to predict how long the algorithm would run on a given data set. This will be particularly useful when deciding which algorithm to use on a big data set.

Data Set

Apart from the length, the parameters were held constant through the simulation. The parameter file can be found page 42.

Results

The graph we obtained is reproduced page 42. We managed to fit the following model (t is the execution time and n the length of the time series):

$$\log(t) = -9.87 + 0.007n + 2.014\log(n)$$

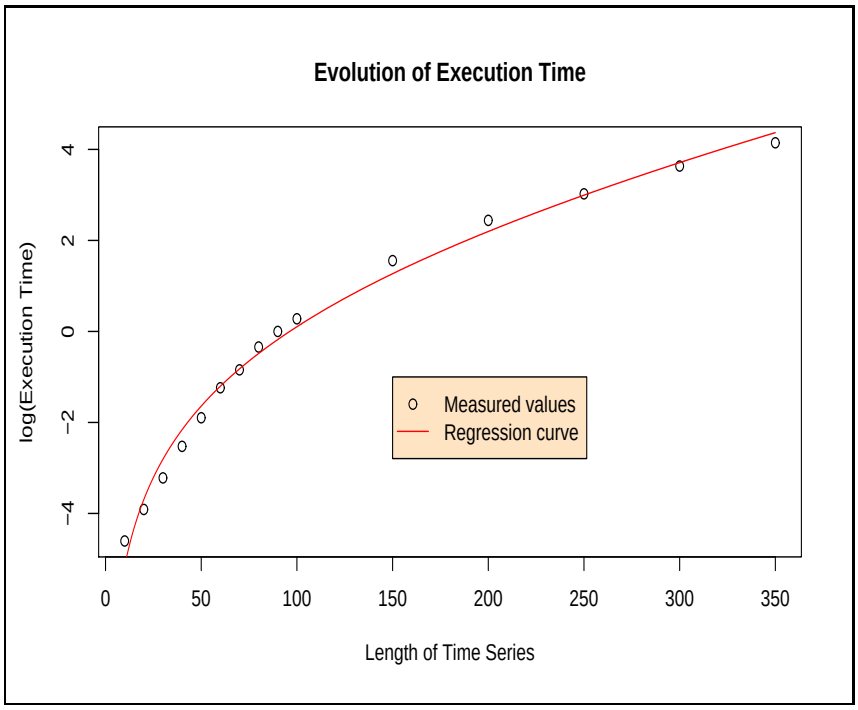

Figure 3.5: Evolution of the execution time of the repeated median algorithm

```
Model parameters

phi_1 0.1
phi_2 -0.2
Seed 4
NbTermes 50
gamma 0.10
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -20
Sensitivity_max 20
Sensitivity_nb_points 50
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

3.3.5 Conclusions

The repeated median filter has advantages and disadvantages which we will summarize here.

Advantages

1. It is fast.
2. It is robust.

Disadvantages

1. The residuals always exert some influence on the results, no matter how extreme they are.
2. The output of the filter is perfectible.

A decisive advantage of the repeated median algorithm is its exact fit property, as explained by Rousseeuw and Leroy in [17] page 60.

Improvements

We would like to reduce the influence of the observations with the highest residuals (downweigh them). Perhaps if we changed the regression procedure used in the algorithm we would be able to improve the results. These improvements will lead us to the biweight filter algorithm which is discussed in the next section.

Chapter 4

Biweight Filter

The biweight filter is the second algorithm we will present in this thesis. It arises as an answer to the limitations of the repeated median algorithm.

4.1 Motivation

The repeated median algorithm has one major drawback: an outlier can have an influence on the result, no matter how large it is. Although the algorithm is robust because its response is bounded, if we have a very large outlier, it will still distort the output, whereas we would like it to be ignored completely, i.e. we would like to give weight 0 to extreme observations. To do this, we have to generalize the repeated median algorithm.

In the repeated median algorithm, we performed a robust regression on a set of trigonometric functions to get a robust estimate of the Fourier coefficients of the time series. The regression we did was not even a weighted regression, which means that we gave the same importance to all the values, even the most extreme ones. In Appendix A, we show that if we have more confidence in certain observations than in others, it is better to use a weighted regression than an ordinary least-squares one. We also present a regression method which calculates the weights "automagically" for us: the recursively reweighted least-squares regression. A theorem of appendix A gives a good justification for using a Huber function. Tatum and Hurvich chose Tukey's biweight function: not only do we get a breakdown point of 0.5 but we know that given regularity assumptions, the estimator sequence converges. The starting estimate of the coefficients is very important, lest we land in a local minimum. We will use the repeated median as a first estimate.

The algorithm is therefore the same as the repeated median algorithm, the only difference being the type of regression.

4.2 Biweight Filter Algorithm

The algorithm is presented by Tatum and Hurvich in [18]. We will review it in detail.

After the following step:

$a'_k \leftarrow MED_j MED_i a_{ijk}$
$b'_k \leftarrow MED_j MED_i b_{ijk}$

we will add:

$a''_k \leftarrow$ biweight regression with a'_k as starting value
$b''_k \leftarrow$ biweight regression with b'_k as starting value

Moreover, we will not use the median as an estimate for the frequency zero: instead, we will perform a biweight regression on the constant function $x \mapsto 1$. Indeed, in non-robust regression, this would correspond to finding the mean. The robust equivalent is called *biweight location estimate*.

The only difference between my implementation and that of Tatum and Hurvich is that we added a convergence acceleration algorithm to increase the precision of the biweight regression, as explained in Appendix C.

In summary, the biweight filter algorithm is:

Algorithm 2 Biweight filter algorithm

Separate the time-series into two overlapping subsequences of prime length
for each subsequence **do**
  Build natural periodogram
  Smooth the periodogram
  Order the frequencies from the strongest to the weakest
  for $m = 1$ to M **do**
    for k from strongest to weekest frequency **do**
      $y_t \leftarrow y_t -$ biweight_location_estimate(y_t)
      for $i = 0$ to $n' - 1$ **do**
        for $j = 0$ to $n' - 1$, $j \neq i$ **do**
          $a_{kij} = \frac{x_i \sin(\omega_k j) - x_j \sin(\omega_k i)}{\sin(\omega_k (j-i))}$
          $b_{kij} = \frac{x_j \cos(\omega_k i) - x_i \cos(\omega_k j)}{\sin(\omega_k (j-i))}$
        end for
      end for
      $a'_k \leftarrow MED_j MED_i a_{ijk}$
      $b'_k \leftarrow MED_j MED_i b_{ijk}$
      $a''_k \leftarrow$ biweight regression with a'_k as starting value
      $b''_k \leftarrow$ biweight regression with b'_k as starting value
    end for
    $y_t = y_t - (a''_k \cos(\omega_k t) + b''_k \sin(\omega_k t))$
    $a_k^{BF} \leftarrow a_k^{BF} + a''_k$
    $b_k^{BF} \leftarrow b_k^{BF} + b''_k$
  end for
end for

4.3 Results

As in the corresponding section in the previous chapter, we will examine the output of the biweight filter.

4.3.1 Filtering Example

Data Set Used

The data set used is reproduced on page 49.

Results

The results of the filter can be seen on Figure 4.1 page 49.
The fit is better than the one achieved for the repeated median.
The core process, in black, is an AR(2) process with

$$\mathsf{x}_t = 0.1\mathsf{x}_{t-1} - 0.5\mathsf{x}_{t-2} + \varepsilon_t$$

The contaminated values appear in red, which means the contaminated process is exactly the same as the core process except for the contaminated values for which it coincides with the red curve instead of the black one.

The filtered process is drawn in green. We can see that it lies very close to the core process, which means the filter is working well.

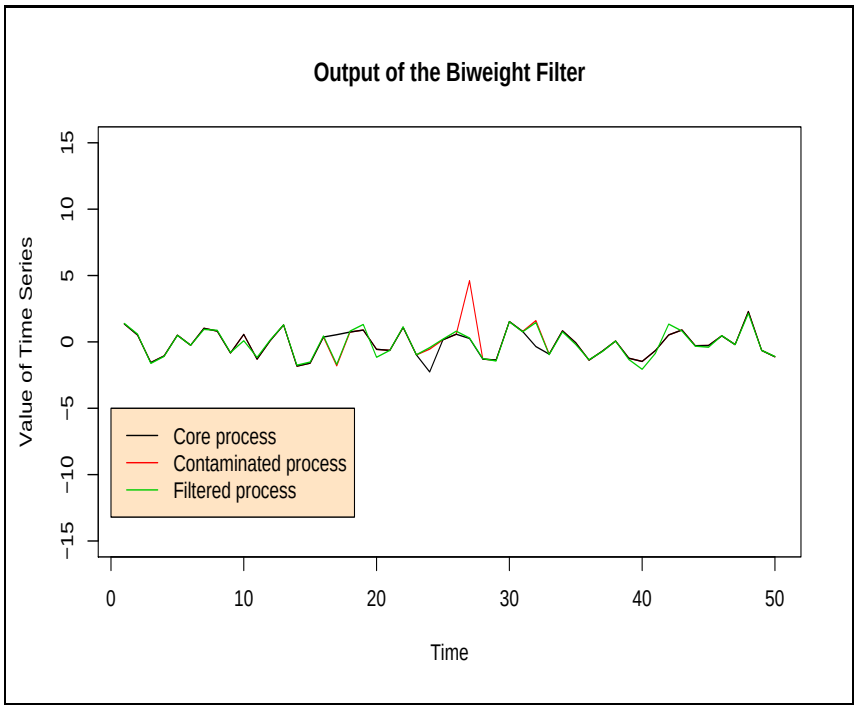

Figure 4.1: Filtering with the biweight filter

```
Model parameters

phi_1 0.2
phi_2 -0.2
Seed 2
NbTermes 50
gamma 0.05
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 2
Scale_param_cleaner 4
Sensitivity_min -100
Sensitivity_max 100
Sensitivity_nb_points 100
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

4.3.2 Distribution of the Residuals

Once again, we plot the values of $\log\left(\frac{\|\mathbf{y}-\mathbf{y}^F\|^2}{\|\mathbf{v}\|^2}\right)$.

Data Set Used

Apart from the random number generator seed, we used the parameter file on page 54 (the same as the one used for the repeated median so we can compare both outputs).

Results

Figure 4.2 page 51 shows us a first set of boxplots for γ from 0 to 1. The biweight filter seems less robust than its predecessor as the boxplots rise quicker. We will now zoom in to the first part of the plot (Figure 4.3 page 52), like we did for the repeated median filter.

In the closer view, things are not as bad as expected: the residuals do not vary much up till $\gamma\#0.233$ so in that range the biweight filter actually seems to perform *better* than its simpler counterpart. Of course, such a comparison is very incomplete and we would need to perform more tests, but our purpose in this report is not to compare the algorithms - just to present them. Therefore we are only interested in seeing whether we addressed the problems of the repeated median algorithm correctly.

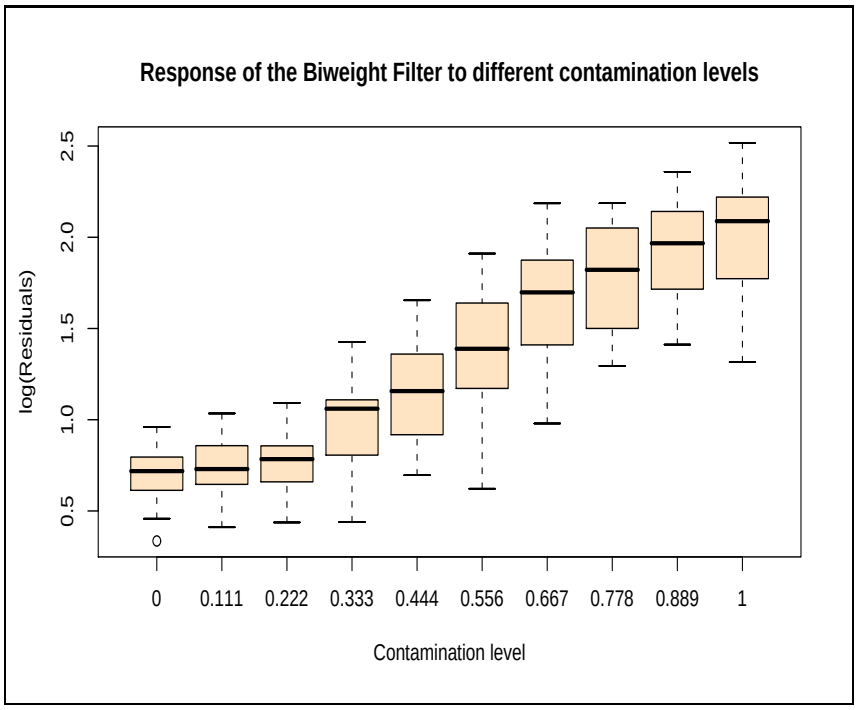

Figure 4.2: Residuals for γ from 0 to 1

```
Model parameters

phi_1 0.75
phi_2 -0.5
Seed 33
NbTermes 50
gamma 0.15
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -15
Sensitivity_max 15
Sensitivity_nb_points 200
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

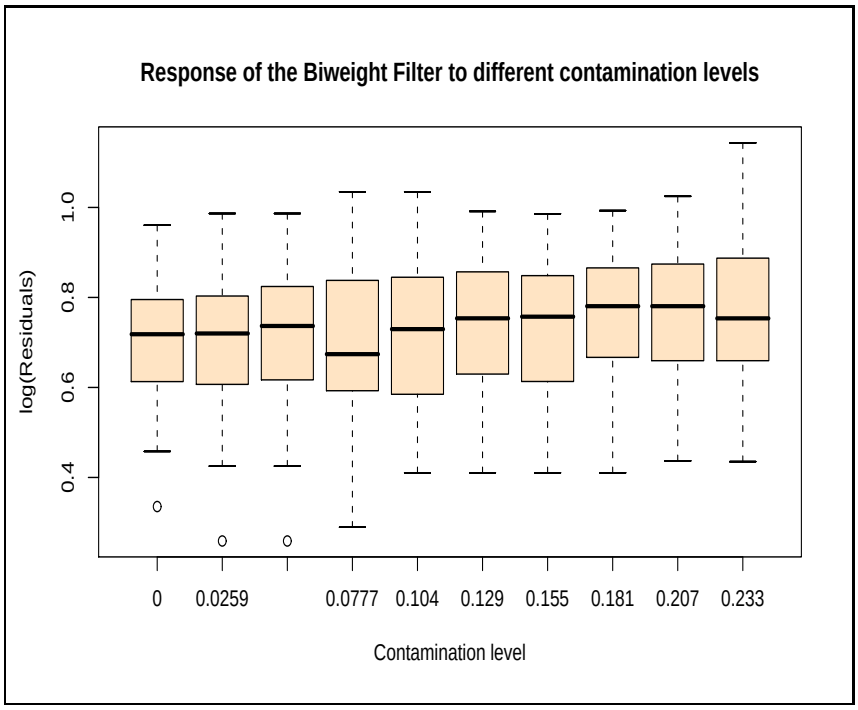

Figure 4.3: Residuals for γ from 0 to 0.35

```
Model parameters

phi_1 0.75
phi_2 -0.5
Seed 33
NbTermes 50
gamma 0.15
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -15
Sensitivity_max 15
Sensitivity_nb_points 200
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

4.3.3 Sensitivity Curve

As before, we will plot a sensitivity curve. We hope it will at least be more regular than that of the repeated median.

Data Set Used

The data set used can be found page 54.

Results

The resulting Figure 4.4 page 54 is much smoother than the previous one. More importantly, big shifts are not taken into consideration and this is exactly what we wanted. All is not perfect, however: the curve is not symmetrical which means that positive shifts are not equivalent to negative ones, and although the weights are in a neighborhood of zero for extreme deviations, they are not *exactly* zero. The curve has no reason to be centered in zero: indeed, the asymptotic value of the sensitivity curve is just the value of the filtered process at this time point without taking the observation at this time point into account.

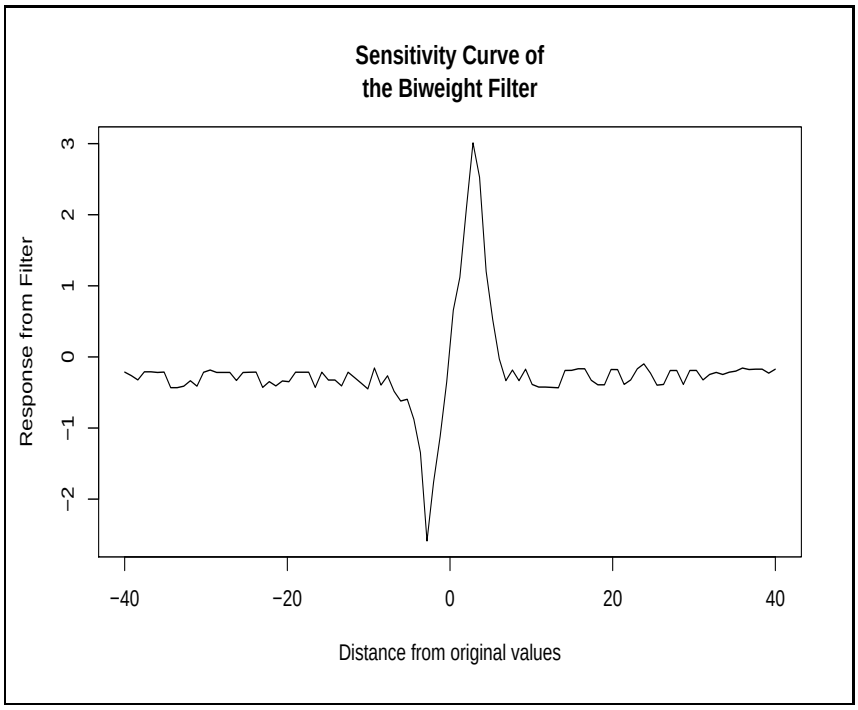

Figure 4.4: Sensitivity curve of the biweight filter

```
Model parameters

phi_1 0.75
phi_2 -0.5
Seed 4
NbTermes 50
gamma 0
variance 20
param_biweight_location 4
param_biweight_regression 6
Max_Nb_Iteration 60
M 2
Scale_param_cleaner 4
Sensitivity_min -40
Sensitivity_max 40
Sensitivity_nb_points 100
Sensitivity_pos 15
Cleaner_lower_bound 2
Cleaner_upper_bound 6
```

4.3.4 Speed

Figure 4.5 page 56 is the graph of the execution time of the biweight filter as a function of the length of the time series. We used the parameter set page 56. The red line is the fitted model. We found the following coefficients:

$$\log(\text{Time}) = -8.8 + 4.2 \times 10^{-3}\text{Length} + 2.1 \log(\text{Length})$$

The biweight filter is therefore approximately twice as slow as the repeated median.

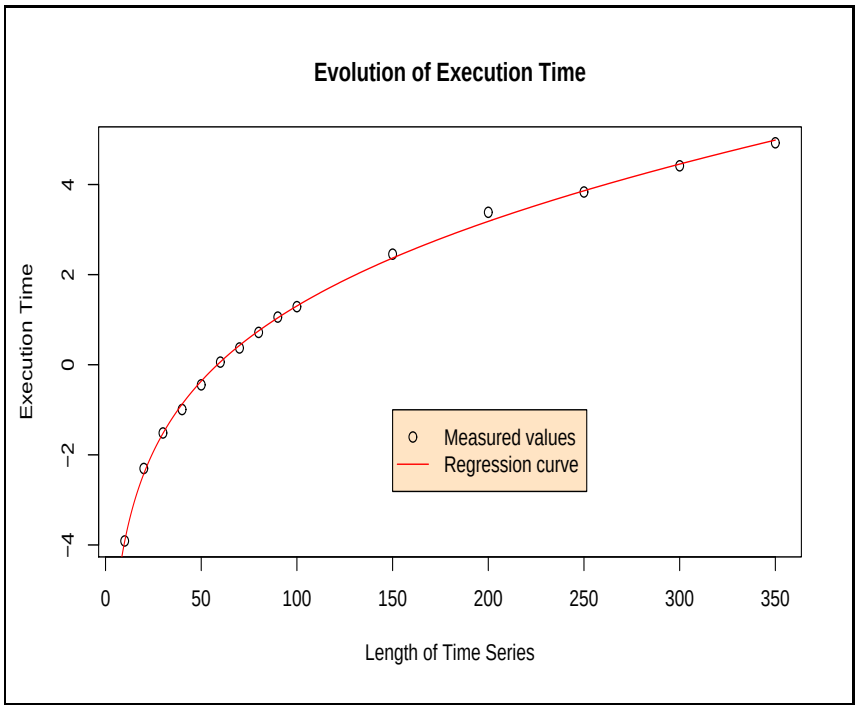

Figure 4.5: Evolution of execution time

```
Model parameters

phi_1 0.1
phi_2 -0.2
Seed 4
NbTermes 50
gamma 0.10
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -20
Sensitivity_max 20
Sensitivity_nb_points 50
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

4.3.5 Conclusions

As before, our new filter has advantages and bad disadvantages.

Advantages

1. More regular for small contamination
2. Large outliers are not taken into consideration

Disadvantages

1. Weights for large outliers are not *exactly* zero
2. Even with no contamination, we cannot recover the original process (we have lost the exact fit property)

Improvements

What we would like to do next is to use the output of the biweight filter as a first estimate for a more powerful algorithm. This drives us to the biweight filter cleaner, explained in the next chapter.

Chapter 5

Biweight Filter Cleaner

5.1 Motivation

One disadvantage of the biweight filter is that no matter how small a contamination we have, the output of the process will reflect this contamination. Indeed, the biweight filter has lost the exact fit property of the repeated median filter. What we would like is an algorithm that leaves most of the process unchanged, modifying only the "problematic" values.

The idea is to use the biweight filter to get a "likely" version of the time series we then use to flag the values which lie too far from the original process. If there is a sudden jolt in the time series, the value of the filtered process will be quite different at this time point from the original value and therefore it will be flagged.

Then, we build the "cleaned" process: we first build a compromise between the filtered version of the time series and the original data without considering the flagged values. We call this compromise *interpolated time series*. Then we leave all the values of the original time series unchanged, except the extreme ones. These values are then replaced by a compromise between the original values and the interpolated ones which lie closer to the rest of the process. We could simply use the filtered version instead of going through the process of building an interpolated time series, but Tatum and Hurvich have found this to be less efficient in simulations.

5.2 Biweight Filter Cleaner Algorithm

The first step in the biweight filter cleaner algorithm is to flag possible outliers. This step is called, quite unsurprisingly, *flagging*. We then build an interpolated time series, i.e. a compromise between the original contaminated series and the filtered one. The last step is putting everything together and computing the cleaned sequence.

5.2.1 Flagging

We first compute the filtered version of the time-series using the biweight filter. We then look at the discrepancy between the original contaminated process and the filtered one to decide whether a time-point should be considered as an outlier. To decide which observation should be flagged, we need to estimate a "typical" deviation between the filtered data and the original observations (a standardization procedure very similar to that of the biweight regression). We use the median absolute deviation (MAD, see Appendix A) as a robust scale estimate of the residuals. Let $n \in \mathbb{N}^*$ and $(y_i)_{1 \leq i \leq n}$ be the original observations. $(y_i^F)_{1 \leq i \leq n}$ will denote the filtered process.

$$r_i := y_i - y_i^F, 1 \leq i \leq n$$
$$s := MED_i \, |r_i - MED_i(r_i)|$$

The scaled residuals are then defined by:

$$u_i := \frac{r_i}{Ks}, 1 \leq i \leq n$$

where K is a tuning constant (similar to that of the biweight regression). Tatum and Hurvich used $K = 4$ in their paper. If $|u_t| > 1$, then y_t is flagged.

5.2.2 Linear Interpolator

The interpolated time-series is simply the best approximation of the original time-series, not taking flagged values into account. This means that we will build a weighted version of the original time-series, i.e. at each time-point i the interpolated value $\hat{y_i}$ will be a weighted average of all the other values of the time-series with weight 0 for flagged observations. More precisely, at each time-point i, we will define a vector of weights $\boldsymbol{w}^{(i)}$ and the interpolated value $\hat{y_i}$ will be

$$\hat{y_i} := \sum_{j=1}^{n} \boldsymbol{w}_j^{(i)} (y_j - \overline{y^F}) + \overline{y^F}$$

where $\overline{y^F}$ is the average of the filtered time-series and $\boldsymbol{w}_{jF}^{(i)} = 0$ if jF is a flagged time-point. How are we going to compute the weights? First, we center the time series by subtracting the mean of the filtered version of the time-series. We then get a process

$$\forall i \in [\![1, n]\!], z_i := y_t - \overline{y^F}$$

Then we build the best approximation of the time-series $(z_i)_{1 \leq i \leq n}$ using all the values except the flagged ones. Thus, we are looking for n vectors of weights minimizing (for all $i \in [\![1, n]\!]$):

$$\left(\sum_{j=1}^{n} (\boldsymbol{w}_j^{(i)} z_j) - z_i \right)^2$$

with the condition $\boldsymbol{w}_{jF}^{(i)} = 0$ if jF is a flagged time-point.

To solve this optimization problem, let us consider the general case of a zero-mean, stationary process $(\mathsf{z}_t)_{1 \leq t \leq n}$. The expression to be minimized can be written:

$$\mathbb{E}\left[\left(\sum_{j=1}^{n}(\boldsymbol{w}_j^{(i)}\mathsf{z}_j) - \mathsf{z}_i\right)^2\right]$$

If $\mathbf{z}^\intercal := (\mathsf{z}_1, \cdots, \mathsf{z}_n)$ and $\mathbf{C} := \mathbb{E}(\mathbf{z}\mathbf{z}^\intercal)$, this is equivalent to minimizing:

$$\mathbb{E}\left[\boldsymbol{w}^{(i)\intercal}\mathbf{z}\mathbf{z}^\intercal\boldsymbol{w}^{(i)}\right] = \boldsymbol{w}^{(i)\intercal}\mathbf{C}\boldsymbol{w}^{(i)}$$

$\mathbf{C}$ is the variance-covariance matrix, i.e. if γ is the autocovariance function,

$$\mathbf{C} = \begin{bmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(n) \\ & \ddots & \ddots & \vdots \\ & \text{Sym} & \ddots & \gamma(1) \\ & & & \gamma(0) \end{bmatrix}$$

We therefore need an estimate $\hat{\gamma}$ of the autocovariance function. Tatum and Hurvich propose:

$$\hat{\gamma}(r) = \frac{1}{n}\sum_{i=1}^{n-r}(y_i^F - \overline{y^F})(y_{i+r}^F - \overline{y^F})$$

The problem of this estimator is that we do not use the same number of observations for each value of r: we will need to smooth it using the method described in Appendix B. Choosing the Bartlett-Priestley window guarantees that $\mathbf{C}$ is positive definite. We will denote this smoothed version by $\tilde{\gamma}$. We determine the amount of smoothing with the AIC_C criterion (also explained in Appendix B).

For the constraints, we do the following reasoning. Let m be the number of flagged values. For each time-point i, we will define a matrix $\mathbf{M}^{(i)}$ and a vector $\boldsymbol{b}^{(i)}$ the following way.

If the time-point i has been flagged, then $\mathbf{M}^{(i)}$ will be an $m \times n$ matrix and $\boldsymbol{b}^{(i)}$ will have dimension m.

If i has not been flagged, then $\mathbf{M}^{(i)}$ will be an $(m+1) \times n$ matrix and $\boldsymbol{b}^{(i)}$ will have dimension $m+1$.

In both cases, each row of $\mathbf{M}^{(i)}$ corresponds to a flagged value. The matrix $\mathbf{M}$ has zeros everywhere except at the flagged time-points: for example, if the i-th flagged value is value number j in the time-series, then we will put a 1 in the (i,j)-entry. If i is flagged, then the vector $\boldsymbol{b}^{(i)}$ will contain zeros everywhere except at the index corresponding to i (i.e. the position of the flagged value i in the list of all flagged values) where it will hold -1. If i is not flagged, then $\boldsymbol{b}^{(i)}$ will have zeros everywhere except on its last entry which will be -1 and the

matrix $\mathbf{M}^{(i)}$ will be defined as above but the last line $(m+1)$ will have a 1 in position i.

For example, if $n = 6$ and the flagged values are $(1, 3, 4)$, then $m = 3$ and the matrix $\mathbf{M}^{(1)}$ looks like:

$$\mathbf{M}^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Vector $\boldsymbol{b}^{(1)}$ will be:

$$\boldsymbol{b}^{(1)^\mathsf{T}} = (-1, 0, 0)$$

For $i = 2$,

$$\mathbf{M}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Vector $\boldsymbol{b}^{(2)}$ will be:

$$\boldsymbol{b}^{(2)^\mathsf{T}} = (0, 0, 0, -1)$$

Thus, $\forall i \in [\![1, n]\!]$, the constraints can be written:

$$\mathbf{M}^{(i)}\boldsymbol{w}^i = \boldsymbol{b}^{(i)}$$

Putting everything together, at each time-point i, we have to solve the following minimization problem for $\boldsymbol{w}^{(i)}$:

$$\begin{gathered} \text{Minimize } \boldsymbol{w}^{(i)^\mathsf{T}}\mathbf{C}\boldsymbol{w}^{(i)}, \\ \text{subject to } \mathbf{M}^{(i)}\boldsymbol{w}^{(i)} = \boldsymbol{b}^{(i)}. \end{gathered}$$

Using Appendix C, we apply the Lagrange multiplicator theorem to the special case of a quadratic form (Theorem 11.1.4 page 156), which yields to:

$$\forall i \in [\![1, n]\!], \boldsymbol{w}^{(i)} = \mathbf{C}^{-1}\mathbf{M}^{(i)^\mathsf{T}}(\mathbf{M}^{(i)}\mathbf{C}^{-1}\mathbf{M}^{(i)^\mathsf{T}})^{-1}\boldsymbol{b}^{(i)}$$

Once we have computed these weights, we can calculate the interpolated values:

$$\forall i \in [\![1, n]\!], \hat{y}_i := \sum_{j=1}^{n} \boldsymbol{w}_j^{(i)}(y_j - \overline{\boldsymbol{y}^F}) + \overline{\boldsymbol{y}^F}$$

To decide whether an observation is "far enough" to be replaced by the interpolated value, we use the studentized residuals d_i with:

$$\forall i \in [\![1, n]\!], d_i := \frac{y_i - \hat{y}_i}{\hat{\sigma}_i}$$

where $\hat{\sigma_i}$ is derived from the estimated interpolation variance:

$$\hat{\sigma_i}^2 := \sum_{j=1}^{n} \sum_{k=1}^{n} \boldsymbol{w}_j^{(i)} \boldsymbol{w}_k^{(i)} \tilde{\gamma}(j-k)$$

The cleaned data set is constructed as follows:

$$\forall i \in [\![1, n]\!], y_i^C := \begin{cases} y_i, & \text{if } |d_i| \leq a \\ \alpha_i y_i + (1-\alpha_i)\hat{y}_i, & \text{if } a < |d_i| \leq b \\ \hat{y}_i, & \text{if } |d_i| > b \end{cases}$$

where a and b are two constants chosen empirically to permit most flagged values to be readmitted when the core process is Gaussian autoregressive and non-contaminated. Tatum and Hurvich chose $a = 3$ and $b = 5$. $\forall i \in [\![1, n]\!], \alpha_i := \frac{b-d_i}{b-a}$.

5.2.3 Complete Algorithm

The complete biweight filter cleaner algorithm, as presentend by Tatum and Hurvich in [18], is detailed on page 64.

Algorithm 3 Biweight filter cleaner algorithm

Compute filtered version of the series $\boldsymbol{y}^F$
for $i = 1$ to n **do**
  $r_i \leftarrow y_i - y_i^F$
end for
Compute scale estimate $s \leftarrow MED_i\,|r_i - MED_j(r_j)|$
Set a value for K, a and b (for example, $K \leftarrow 4, a \leftarrow 2$ and $b \leftarrow 3$)
Initialize an empty vector for the flagged values
for $i = 1$ to n **do**
  $u_i \leftarrow \frac{r_i}{Ks}$
  if $|u_i| > 1$ **then**
    append i to the vector of flagged values
  end if
end for
Compute estimate for the periodogram
Determine the amount of smoothing with AIC_C
Smooth periodogram
Inverse Fourier-transform the periodogram
Compute the covariance matrix $\mathbf{C}$
for $i = 1$ to n **do**
  Compute matrix $\mathbf{M}^{(i)}$ and vector $\boldsymbol{b}^{(i)}$
  $\boldsymbol{w}^{(i)} \leftarrow \mathbf{C}^{-1}\mathbf{M}^{(i)^\intercal}(\mathbf{M}\mathbf{C}^{-1}\mathbf{M}^\intercal)^{-1}\boldsymbol{b}^{(i)}$
end for
for $i = 1$ to n **do**
  Compute interpolated value: $\hat{y}_i \leftarrow \sum_{j=1}^{n} \boldsymbol{w}_j^{(i)}(y_j - \overline{\boldsymbol{y}^F}) + \overline{\boldsymbol{y}^F}$
  Compute estimated interpolation variance: ${\hat{\sigma}_i}^2 \leftarrow \sum_{j=1}^{n}\sum_{k=1}^{n} \boldsymbol{w}_j^{(i)}\boldsymbol{w}_k^{(i)}\tilde{\gamma}(j-k)$
  Calculate the studentized residuals: $\forall i \in [\![1, n]\!], d_i \leftarrow \frac{y_i - \hat{y}_i}{\hat{\sigma}_i}$
  Calculate $\alpha_i \leftarrow \frac{b - d_i}{b - a}$
  Construct the cleaned data set:

$$\forall i \in [\![1, n]\!], y_i^C \leftarrow \begin{cases} y_i, & \text{if } |d_i| \leq a \\ \alpha_i y_i + (1-\alpha_i)\hat{y}_i, & \text{if } a \leq |d_i| \leq b \\ \hat{y}_i, & \text{if } |d_i| \geq b \end{cases}$$

end for

5.3 Results

5.3.1 Filtering Example

We will use the same parameters as in the two previous algorithms. They can be read on page 66. We obtained Figure 5.1 page 66. The graph is not perfect, but not far from it: apart from four time points, we recover the core process exactly. This is an encouraging first result!

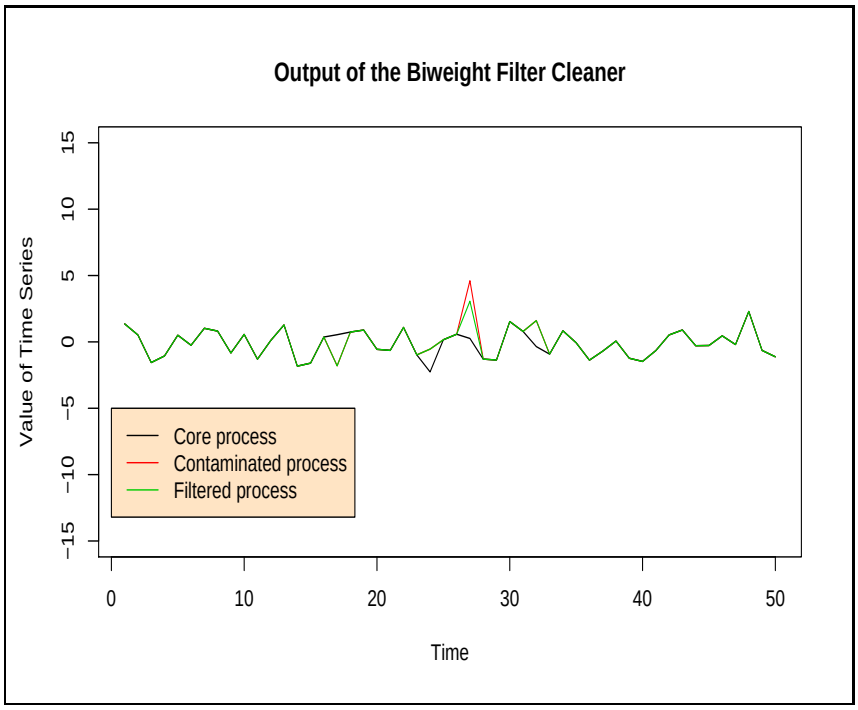

Figure 5.1: Output of the biweight filter cleaner

```
Model parameters

phi_1 0.2
phi_2 -0.2
Seed 2
NbTermes 50
gamma 0.05
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 2
Scale_param_cleaner 4
Sensitivity_min -100
Sensitivity_max 100
Sensitivity_nb_points 100
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

5.3.2 Distribution of the Residuals

As before, we use the parameters on page 68. The breakdown point seems to be even lower than for the biweight filter. To investigate this, let us look at a close-up (Figure 5.3 page 69). From $\gamma = 0.23$ on, the residuals grow almost exponentially. This presumably means that the performance of the cleaner comes with a price: a relatively low breakdown point.

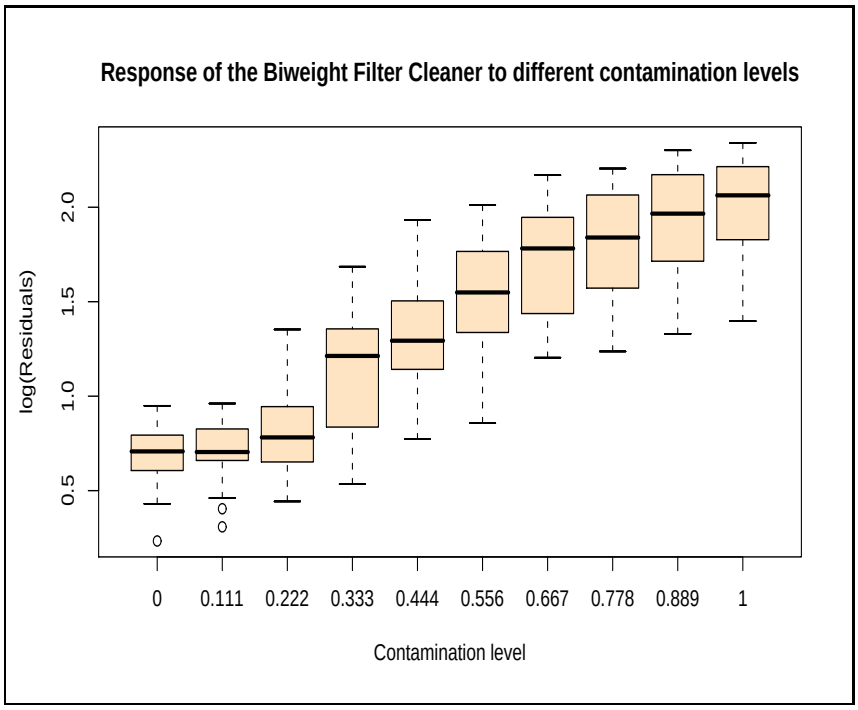

Figure 5.2: Residuals for γ from 0 to 1

```
Model parameters

phi_1 0.75
phi_2 -0.5
Seed 33
NbTermes 50
gamma 0.15
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -15
Sensitivity_max 15
Sensitivity_nb_points 200
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

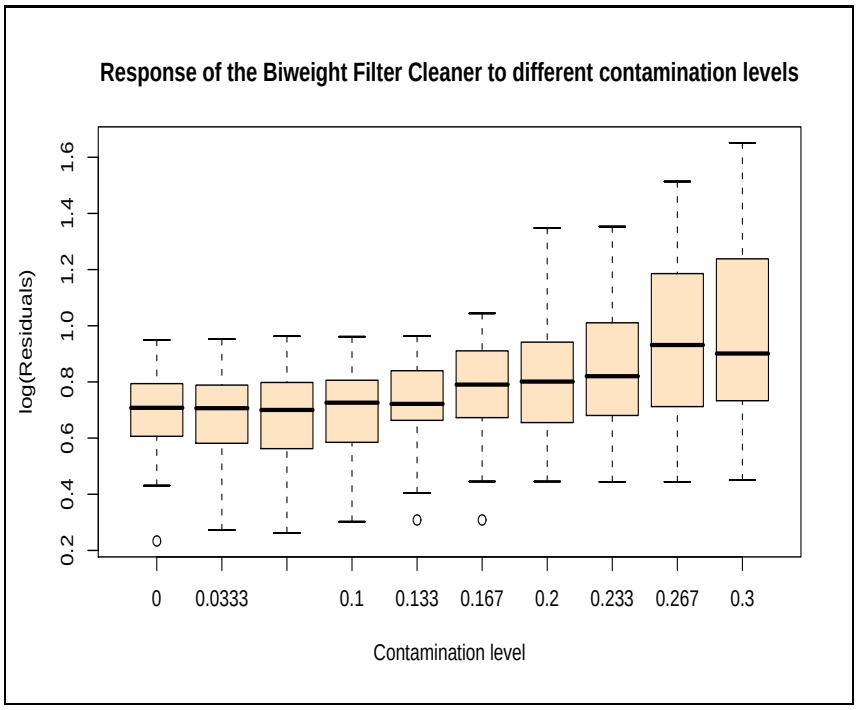

Figure 5.3: Residuals for γ from 0 to 0.3

```
Model parameters

phi_1 0.75
phi_2 -0.5
Seed 33
NbTermes 50
gamma 0.15
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -15
Sensitivity_max 15
Sensitivity_nb_points 200
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

5.3.3 Sensitivity Curve

We used the parameter file page 71. In Figure 5.4 page 71, we can see that this curve is far more regular than the two previous ones. Of course, this is mainly caused by the interpolation at the end of the algorithm and not so much by Tukey's biweight function. Still, it is quite an attractive result.

The robustness of an algorithm is independent from it taking extreme observations into account: indeed, an algorithm can very well have a bounded output and always take outliers into account. The obvious example of this property is the repeated median filter. Conversely, an algorithm which weighs down extreme values might be not very robust, for example, the biweight filter cleaner. Therefore, we always have to find a tradeoff between performance and robustness.

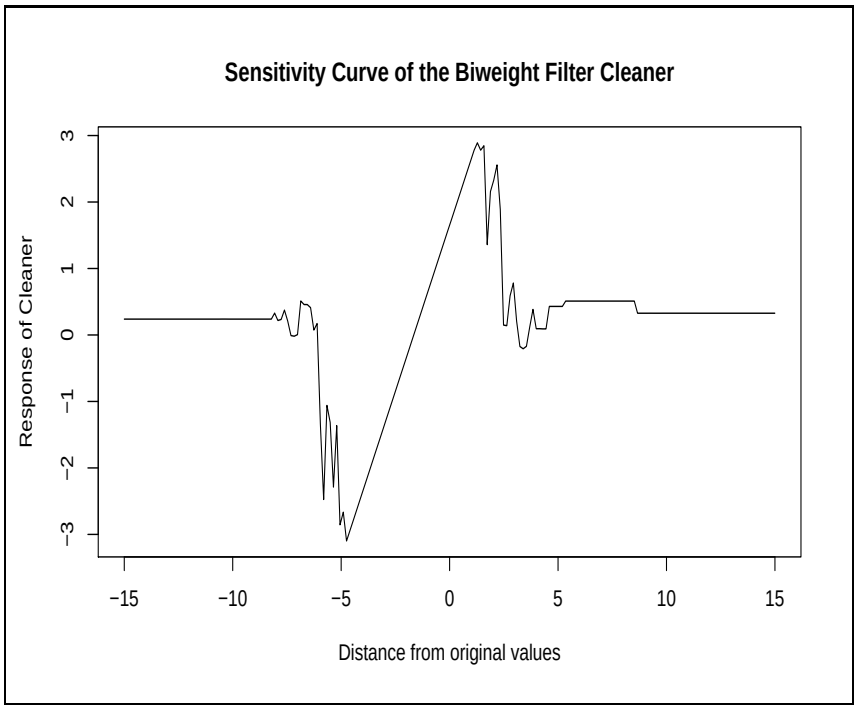

Figure 5.4: Sensitivity Curve for the biweight filter cleaner

```
Model parameters

phi_1 1
phi_2 -0.5
Seed 333
NbTermes 50
gamma 0.15
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 2
Scale_param_cleaner 4
Sensitivity_min -15
Sensitivity_max 15
Sensitivity_nb_points 200
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

5.3.4 Speed

The biweight filter cleaner is by far the slowest of the three algorithms. We used the parameters page 73. The curve we obtained is page 73. The red curve is the linear model we fitted:

$$\log(t) = -9.16 + 0.0074n + 2.22\log(n)$$

This is the same kind of model as for the biweight filter.

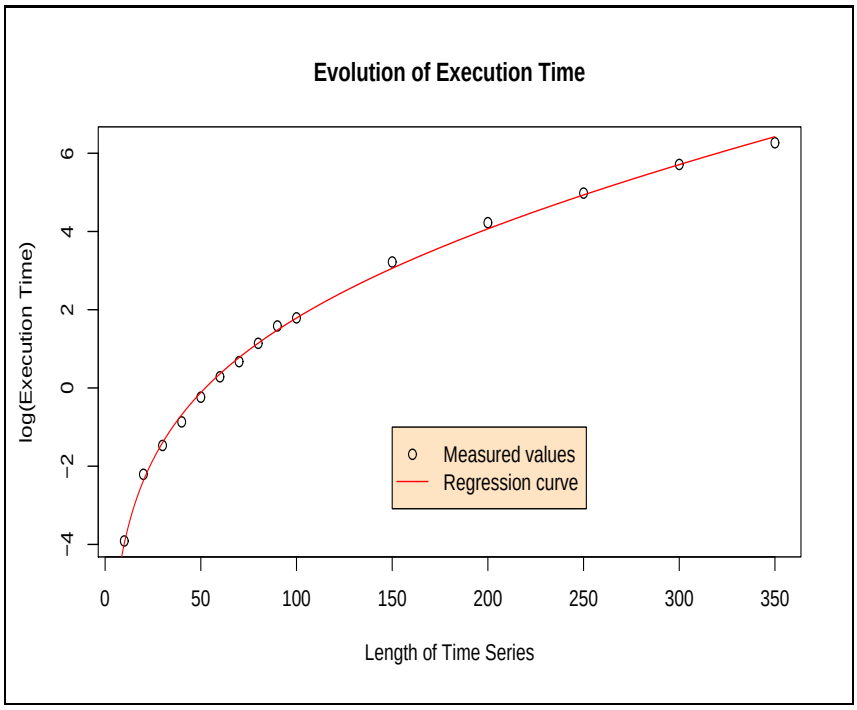

Figure 5.5: Execution time of the cleaner

```
Model parameters

phi_1 0.1
phi_2 -0.2
Seed 4
NbTermes 50
gamma 0.10
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -20
Sensitivity_max 20
Sensitivity_nb_points 50
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

5.3.5 Conclusions

The main advantage of this algorithm is that it gives extremely good results. The main drawback is that it takes ages to do so. The algorithm we will present in the next chapter is meant to combine "the best of both worlds": it is called "repeated median cleaner".

Chapter 6

Repeated Median Cleaner

This algorithm is an attempt to solve the problems of the biweight filter cleaner. The only difference with the previous algorithm is that instead of starting with the biweight-filtered process, we use the repeated median filter. Otherwise, the algorithm is exactly the same as the biweight filter cleaner. As this chapter is almost identical to the previous one, we will only present very briefly the results we obtained.

6.1 Filtering Example

A filtering example is reproduced on Figure 6.1 page 76. The output seems comparable to that of the biweight filter cleaner.

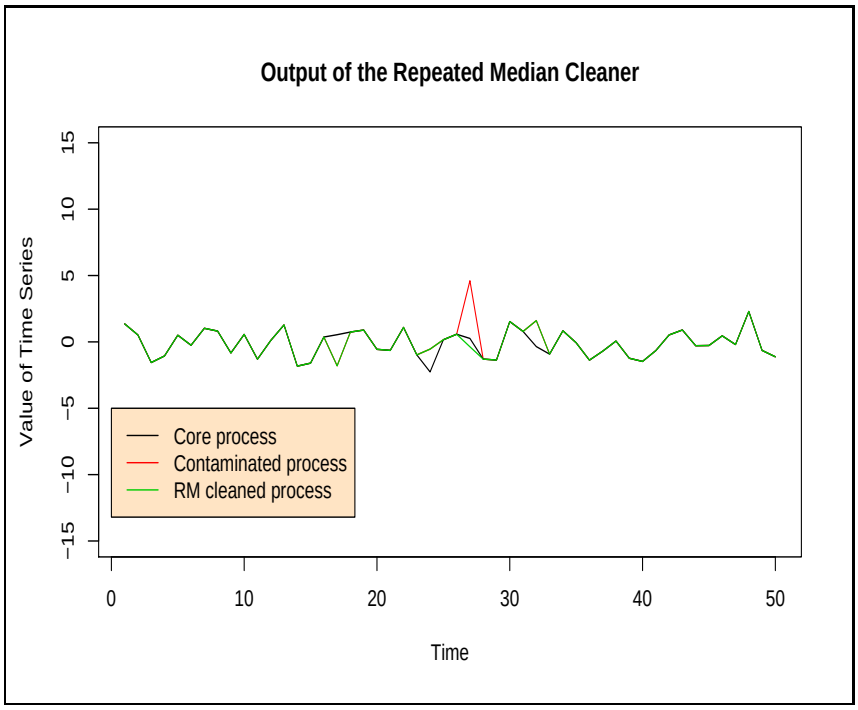

Figure 6.1: Output of the repeated median cleaner

```
Model parameters

phi_1 0.2
phi_2 -0.2
Seed 2
NbTermes 50
gamma 0.05
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 2
Scale_param_cleaner 4
Sensitivity_min -100
Sensitivity_max 100
Sensitivity_nb_points 100
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

6.2 Reponse to Various Contamination Levels

The repeated median cleaner seems much more robust than the other algorithms. Although its breakdown bound is about 0.3, it is much more regular than the biweight filter or the biweight filter cleaner for $\gamma \leq 0.3$.

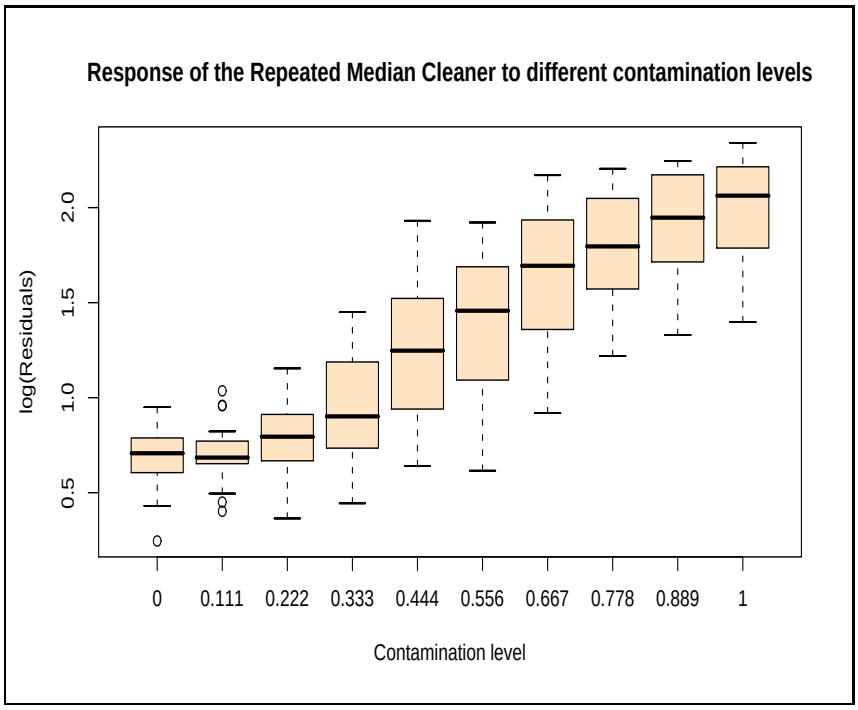

Figure 6.2: Residuals for γ from 0 to 1

```
Model parameters

phi_1 0.75
phi_2 -0.5
Seed 33
NbTermes 50
gamma 0.15
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -15
Sensitivity_max 15
Sensitivity_nb_points 200
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

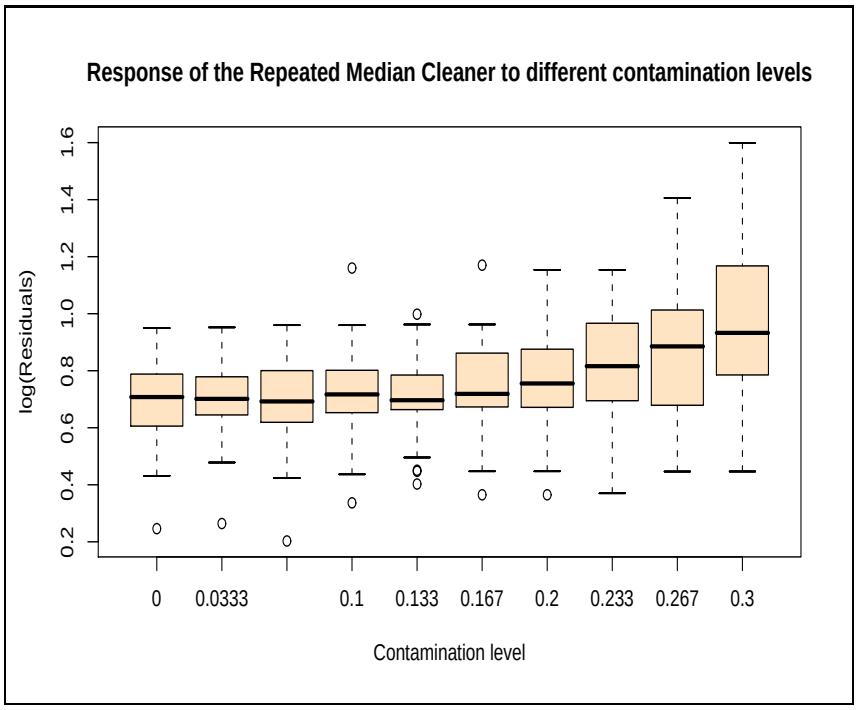

Figure 6.3: Residuals for γ from 0 to 0.3

```
Model parameters

phi_1 0.75
phi_2 -0.5
Seed 33
NbTermes 50
gamma 0.15
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -15
Sensitivity_max 15
Sensitivity_nb_points 200
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

6.3 Sensitivity Curve

Figure 6.4 page 81 is extremely regular, although the curve is quite different from Tukey's biweight function.

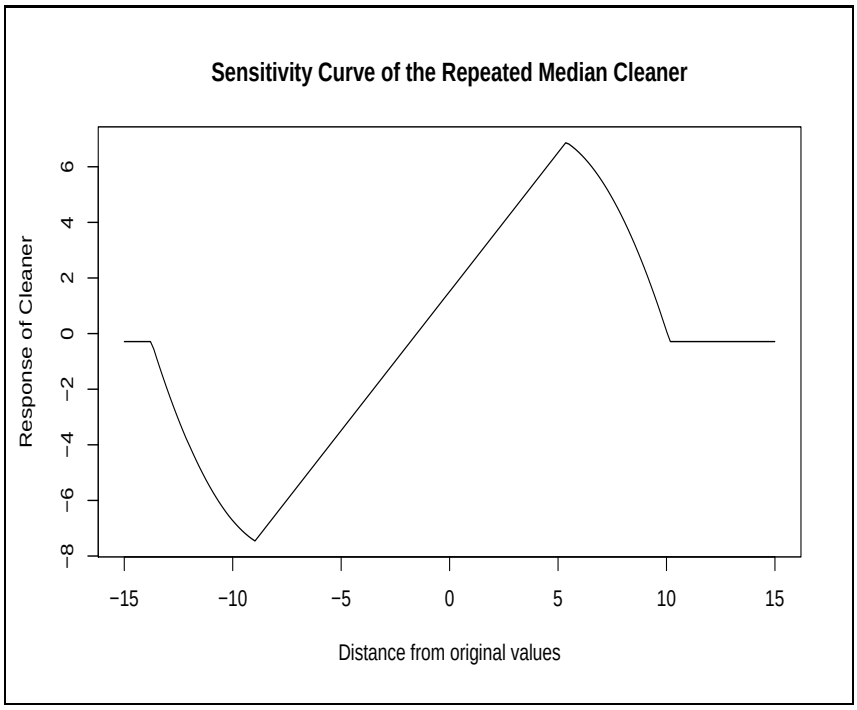

Figure 6.4: Sensitivity curve for the repeated median cleaner

```
Model parameters

phi_1 1
phi_2 -0.5
Seed 333
NbTermes 50
gamma 0.15
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 2
Scale_param_cleaner 4
Sensitivity_min -15
Sensitivity_max 15
Sensitivity_nb_points 200
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

6.4 Speed

In Figure 6.5 page 83, we can see the model we fitted:

$$\log(t) = -10.8 + 9.10^{-3}n + 2.3\log(n)$$

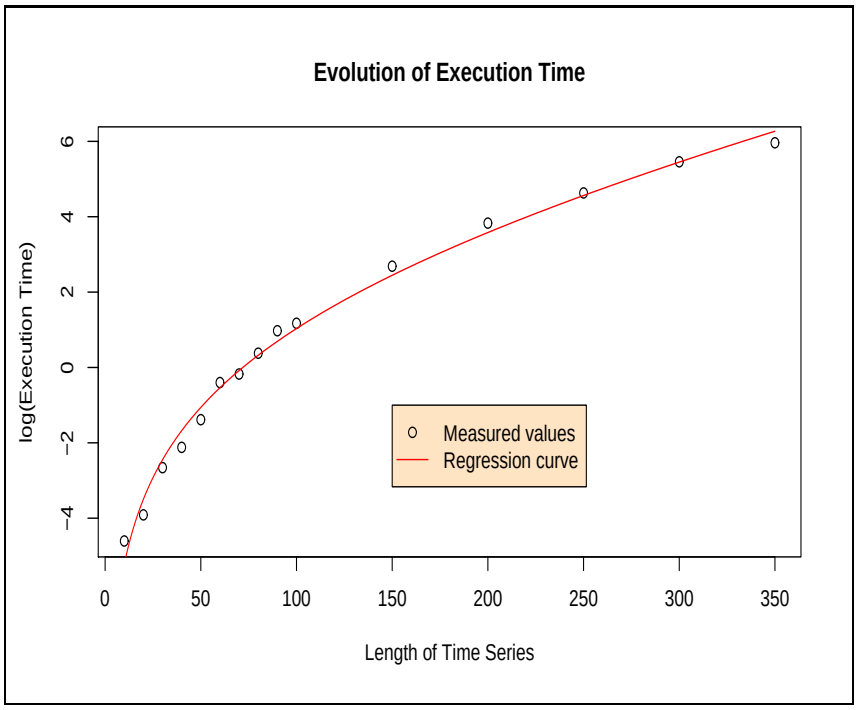

Figure 6.5: Execution time of the repeated median cleaner

```
Model parameters

phi_1 0.1
phi_2 -0.2
Seed 4
NbTermes 50
gamma 0.10
variance 20
param_biweight_location 6
param_biweight_regression 6
Max_Nb_Iteration 60
M 1
Scale_param_cleaner 4
Sensitivity_min -20
Sensitivity_max 20
Sensitivity_nb_points 50
Sensitivity_pos 25
Cleaner_lower_bound 3
Cleaner_upper_bound 5
```

6.5 Conclusions

This algorithm seems to be the most promising of the four. Indeed, it has the robustness of the repeated median without lagging too much behind the biweight filter cleaner in performance.

Chapter 7

Heart Rate Variability in Diabetes

The aim of this work was to filter a time-series in order to remove outliers. This technique was to be applied to the special case of diabetes diagnosis. In this last chapter we will see the biweight filter cleaner in action on a real data set. First, we will present the heart-rate variability diagnosis technique. Then we will review a few facts about diabetes and how heart-rate variability diagnosis is used for this illness. Last, we will present the results we obtained.

7.1 Heart Rate Variability (HRV)

Heart rate variability is the analysis of the fluctuations in heart rate as an early and sensitive indicator of bad health. It is based on the fact that the heart does not beat at the same rate all of the time – it quickens as you run up stairs, for instance, and slows as you nap. As the body experiences these constant changes in rhythm throughout the day, the nervous system must continually function to preserve a smooth, cohesive, overall pattern. Simple analysis of heart rate variability has been used in clinical practice since the 1960s in fetal hypoxia. In this section, we shall first give an overview of how the heart works, then in the second part we will explain what electro-cardiograms are. In the last part, we will give a few indications of how to use heart rate variability in diagnosis.

7.1.1 The Heart

From Wikipedia Encyclopedia, http://www.wikipedia.org.

Structure

A diagram of the heart can be seen on Figure 7.1 page 88. In the human body the heart is normally situated slightly to the left of the middle of the thorax, be-

hind the sternum (breastbone). It is enclosed by a sac known as the pericardium and is surrounded by the lungs. In adults, it weighs about 300-350 g. It consists of four chambers, the two upper atria (singular: atrium) and the two lower ventricles.

A thick muscular wall, the septum, divides the right atrium and ventricle from the left atrium and ventricle, keeping blood from passing between them. Valves between the atria and ventricles (atrioventricular valves) maintain coordinated unidirectional flow of blood from the upper atria to the lower ventricles.

The ventricles are the parts of the heart that pump blood around the body or to the lungs. They are thicker walled than the atria, and the contraction of the ventricle wall is much more important to move blood around.

Oxygen-depleted or deoxygenated blood from the body enters the right atrium through two great veins, the superior vena cava which drains the upper part of the body and the inferior vena cava that drains the lower part. The blood then passes through the tricuspid valve to the right ventricle. The right ventricle pumps the deoxygenated blood to the lungs, through the pulmonary artery. In the lungs gaseous exchange takes places and the blood releases carbon dioxide into the lung cavity and picks up oxygen. The oxygenated blood then flows through pulmonary veins to the left atrium. From the left atrium this newly oxygenated blood passes through the mitral valve to enter the left ventricle. The left ventricle then pumps the blood through the aorta to the entire body except the lungs.

The left ventricle is much more muscular than the right as it has to pump blood around the entire body, which involves exerting a considerable force to overcome the vascular pressure. As the right ventricle needs to pump blood only to the lungs, it requires less muscle.

Even though the ventricles lie below the atria, the two vessels through which the blood exits the heart (the pulmonary artery and the aorta) leave the heart at its top side.

The contractile nature of the heart is due to the presence of cardiac muscle in its wall which can work continuously without fatigue. The heart wall is made of three distinct layers. The first is the outer epicardium which is composed of a layer of flattened epithelial cells and connective tissue. Beneath this is a much thicker myocardium made up of cardiac muscle. The endocardium is a further layer of flattened epithelial cells and connective tissue which lines the chambers of the heart.

The blood supply to the heart itself is supplied by the left and right coronary arteries, which branch off from the aorta.

The Cardiac Cycle

The function of the heart is to pump blood around the body, in cycles. The cycle is explained below.

Every single beat of the heart involves a sequence of events known as the

cardiac cycle consisting of three major stages; atrial systole, ventricular systole and complete cardiac diastole. The atrial systole consists of the contraction of the atria and the corresponding influx of blood in to the ventricles. Once the blood has fully left the atria, the atrioventricular valves, which are situated between the atria and ventricular chambers, close. This prevents any backflow into the atria. It is the sound of the valves closing which produces the familiar beating sounds of the heart.

The ventricular systole consists of the contraction of the ventricles and flow of blood into the circulatory system. Again, once all the blood has left, the pulmonary and aortic semilunar valves close. Finally complete cardiac diastole involves the relaxation of the atria and ventricles in preparation for new blood to enter the heart.

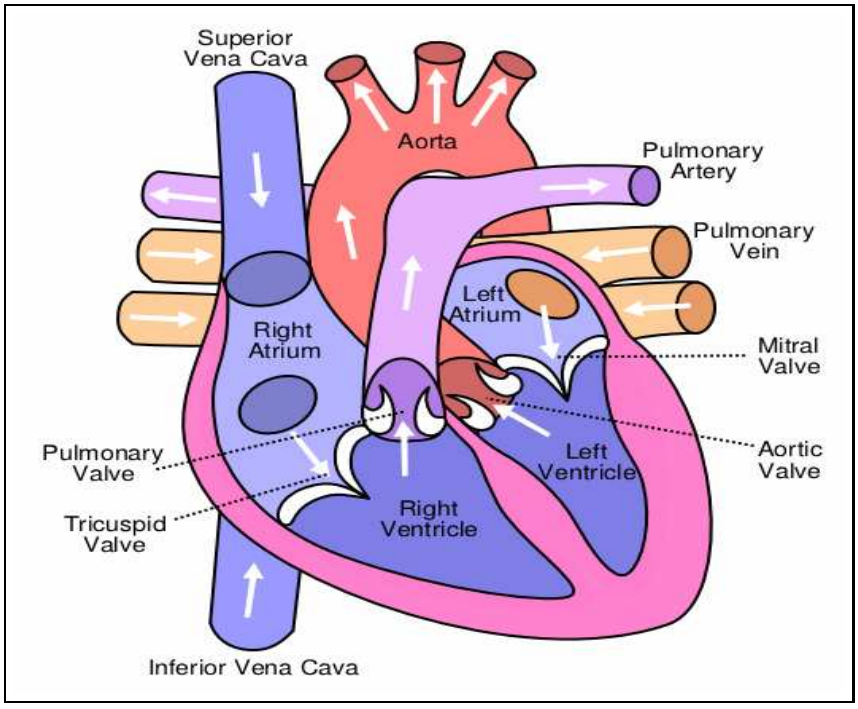

Figure 7.1: Anterior (frontal) view of the opened heart. White arrows indicate normal blood flow.

7.1.2 The Electrocardiogram

An electrocardiogram (ECG or EKG, abbreviated from the German Elektrokardiogramm) is a graphic produced by an electrocardiograph, which records the electrical voltage in the heart in the form of a continuous strip graph. It is the prime tool in cardiac electrophysiology, and has a prime function in screening and diagnosis of cardiovascular diseases. An example of electrocardiogram is given on Figure 7.2 page 91.

Uses

The ECG has a wide array of uses:

* Determine whether the heart is performing normally or suffering from abnormalities (e.g. extra or skipped heartbeats - cardiac arrhythmia).
* May indicate acute or previous damage to heart muscle (heart attack) or ischaemia of heart muscle (angina).
* Can be used for detecting potassium, calcium, magnesium and other electrolyte disturbances.
* Allows the detection of conduction abnormalities (heart blocks and bundle branch blocks).
* As a screening tool for ischaemic heart disease during an exercise tolerance test.
* Can provide information on the physical condition of the heart (eg: left ventricular hypertrophy, mitral stenosis).
* Can suggest non-cardiac disease (e.g. pulmonary embolism, hypothermia)

Intervals

A typical ECG tracing of a normal heartbeat consists of a P wave, a QRS complex and a T wave. A small U wave is not normally visible (see Figure 7.2 page 91).

Axis The axis is the general direction of the electrical impulse through the heart. It is usually directed to the bottom left, although it can deviate to the right in very tall people and to the left in obesity. Extreme deviation is abnormal and indicates a bundle branch block, ventricular hypertrophy or (if to the right) pulmonary embolism. It also can diagnose dextrocardia or a reversal of the direction in which the heart faces, but this condition is very rare and often has already been diagnosed by something else(such as a chest x-ray).

P wave The P wave is the electrical signature of the current that causes atrial contraction. Both the left and right atria contract simultaneously. Irregular or absent P waves may indicate arrhythmia. Its relationship to QRS complexes determines the presence of a heart block.

QRS The QRS complex corresponds to the current that causes contraction of the left and right ventricles, which is much more forceful than that of the atria and involves more muscle mass, thus resulting in a greater ECG deflection.

The Q wave, when present, represents the small horizontal (left to right) current as the action potential travels through the interventricular septum. Very wide and deep Q waves do not have a septal origin, but indicate myocardial infarction.

The R and S waves indicate contraction of the myocardium. Abnormalities in the QRS complex may indicate bundle branch block (when wide), ventricular origin of tachycardia, ventricular hypertrophy or other ventricular abnormalities. The complexes are often small in pericarditis.

T wave The T wave represents the repolarization of the ventricles. The QRS complex usually obscures the atrial repolarization wave so that it is not usually seen. Electrically, the cardiac muscle cells are like loaded springs. A small impulse sets them off, they depolarize and contract. Setting the spring up again is repolarization (more at action potential).

In most leads, the T wave is positive. Negative T waves can be signs of disease, although an inverted T wave is normal in V1 (and V2-3 in black people).

The ST segment connects the QRS complex and the T wave. It can be depressed in ischemia and elevated in myocardial infarction, and downslopes in digoxin use.

T wave abnormalities may indicate electrolyte disturbance, such as hyperkalemia.

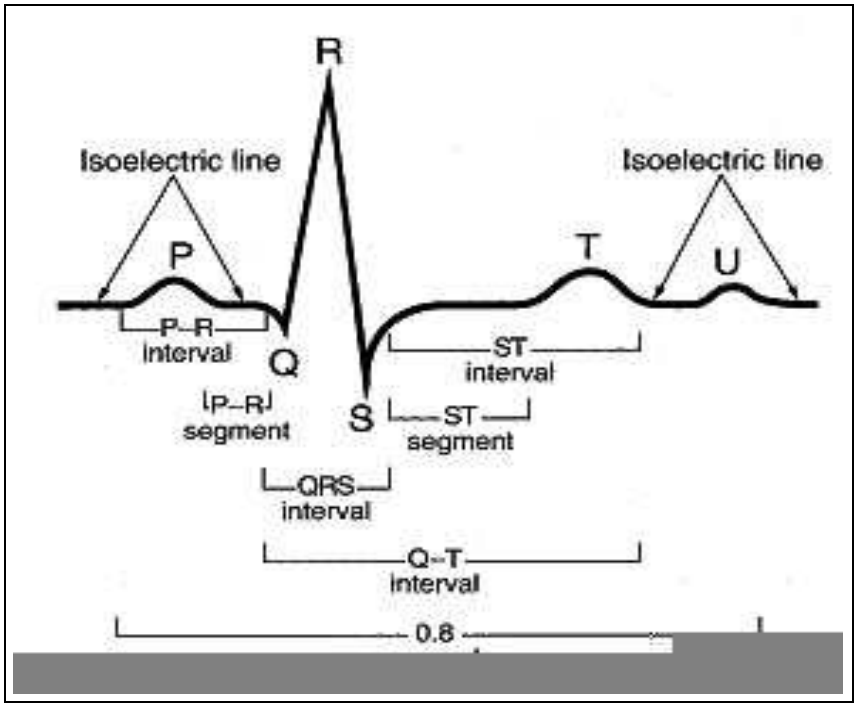

Figure 7.2: Intervals in the ECG

7.1.3 Use of HRV in Diagnosis

Jiri Pumprla et al. present in [14] the latest developments in the use of HRV in diagnosis. Here, we will only give a very brief review.

How to Measure HRV

Early techniques for analysis of autonomic activity were based on a set of physiological tests proposed by Ewing et al., quoted by Jiri Pumprla et al. in [14]. These tests included the measurement in heart rate induced by a certain stimulus, such as deep breathing, hand-grip or the Valsa-manoeuvre. All of these tests suffer the same drawback: it is difficult to standardize the stimulus which means that the variability of the results will be very high.

What HRV Can Detect

A high variability in heart rate is a sign of good adaptability, implying a healthy individual with good autonomic control mechanisms. A low variability is often the sign that the autonomic nervous system is not sufficiently adaptable and further investigation is required to give a more precise diagnosis.

More than half of all patients with end stage renal disease have detectable autonomic neuropathy and measurement HRV is superior to classical reflex testing for its detection. Metabolic derangements in chronic liver disease and/or hypoxia in chronic respiratory disease may also induce autonomic abnormalities leading to reduced HRV. Usually, an improvement in metabolic or neurological functions is associated with a return to a normal HRV pattern.

7.2 Diabetes

In this section, we will give background information on diabetes. First, what it is, then how to treat it and afterwards how we can use HRV to treat it.

7.2.1 What is Diabetes?

There are currently more than 194 million people with diabetes worldwide and if nothing is done to slow the epidemic, the number will exceed 333 million by 2025. We will first give a definition of diabetes, then list the symptoms associated with this condition and last review the main causes of this disease.

Definition

Diabetes mellitus is a condition in which the amount of glucose (sugar) in the blood is too high because the body cannot use it properly. Glucose comes

from the digestion of starchy foods such as bread, rice, potatoes, chapatis, yams and plantain, from sugar and other sweet foods, and from the liver which makes glucose.

When sugar and starchy foods have been digested, they turn into glucose. If somebody has diabetes, the glucose in their body is not turned into energy, either because there is not enough insulin in their body, or because the insulin that the body produces is not working properly. This causes the liver to make more glucose than usual but the body still cannot turn the glucose into energy. The body then breaks down its stores of fat and protein to try to release more glucose but still this glucose cannot be turned into energy. This is why people with untreated diabetes often feel tired and lose weight. The unused glucose passes into the urine, which is why people with untreated diabetes pass large amounts of urine and are extremely thirsty.

There are four types of diabetes in all, two of which account for 90% of all diagnosed cases of diabetes.

1. Type 1 diabetes is most commonly diagnosed in children and adolescents, but can occur in adults as well. It is an autoimmune disorder, in which the body's own immune system attacks the beta cells in the Islets of Langerhans of the pancreas, destroying them or damaging them sufficiently to reduce insulin production. The autoimmune attack may be triggered by reaction to an infection, for example by one of the viruses of the Coxsackie virus family. A subtype of Type 1 (identifiable by the presence of antibodies against beta cells) develops slowly and so is often confused with Type 2. In addition, a small proportion of Type 1 cases has the hereditary condition maturity onset diabetes of the young (MODY). Some poisons (e.g. certain rat poisons) work by selectively destroying certain types of cells, including pancreatic beta cells, thus producing "artificial" Type 1 diabetes. Other pancreatic problems including trauma, pancreatitis or tumors (either malignant or benign) can also lead to loss of insulin production.

2. Type 2 diabetes is characterized by "insulin resistance" as body cells do not respond appropriately when insulin is present. This is a more complex problem than Type 1, but is sometimes easier to treat, since insulin is still produced, especially in the initial years. Type 2 may go unnoticed for years in a patient before diagnosis, since the symptoms are typically milder (no ketoacidosis) and can be sporadic. However, severe complications can result from unnoticed Type 2 diabetes, including renal failure, and coronary artery disease.

3. All other specific forms of diabetes, accounting for up to 5% of all diagnosed cases of diabetes, are termed Type 3:

    - Type 3A: genetic defect in beta cells.

- Type 3B: genetically related insulin resistance.
- Type 3C: diseases of the pancreas.
- Type 3D: caused by hormonal defects.
- Type 3E: caused by chemicals or drugs.

4. Type 4 or gestational diabetes mellitus appears in about 2-5% of all pregnancies. It is temporary and fully treatable, but if untreated it may cause problems with the pregnancy, including macrosomia (high birth weight) of the child. It requires careful medical supervision during the pregnancy. In addition, about 20-50% of these women go on to develop Type 2 diabetes.

This section used the following web sites as a primary source of information.

`http://www.idf.org/home/index.cfm?node=37`
`http://www.diabetes.org.uk/diabetes/under.htm`

What are the Symptoms of Diabetes?

The main symptoms of diabetes are:

- increased thirst
- urinating frequently, especially at night
- extreme tiredness
- weight loss
- genital itching or regular episodes of thrush
- blurred vision.

Type 2 diabetes develops slowly and the symptoms are usually less severe. Some people may not notice any symptoms at all and their diabetes is only picked up in a routine medical check up. Some people may put the symptoms down to 'getting older' or 'overwork'.

Type 1 diabetes develops much more quickly, usually over a few weeks, and symptoms are normally very obvious.

In both types of diabetes, the symptoms are quickly relieved once the diabetes is treated. Early treatment will also reduce the chances of developing serious health problems. In 2003, the five countries with the highest diabetes prevalence in the adult population were Nauru (30.2 %), The United Arab Emirates (20.1 %), Qatar (16%), Bahrain(14.9%), and Kuwait (12.8%). At least 50% of all people with diabetes are unaware of their condition and in some countries this figure may rise to 80%, which makes diagnostic very important. In addition, diabetes is the fourth main cause of death in most developed countries and is the leading cause of blindness and visual impairment in adults in developed countries. Diabetes is the most common cause of amputation which is not the result of an accident.

How Does One Get Diabetes?

Although the condition can occur at any age, it is rare in infants and becomes more common as people get older.

Type 1 diabetes develops when the insulin-producing cells in the pancreas have been destroyed. Nobody knows for sure why these cells have been damaged but the most likely cause is an abnormal reaction of the body to the cells. This may be triggered by a viral or other infection. This type of diabetes generally affects younger people. Both sexes are affected equally.

Type 2 diabetes used to be called "maturity onset" diabetes because it usually appears in middle-aged or elderly people, although it does occasionally appear in younger people. The main causes are that the body no longer responds normally to its own insulin, and/or that the body does not produce enough insulin.

Both type 1 and type 2 diabetes are at least partly inherited. Type 1 diabetes appears to be triggered by infection, stress, or environmental factors (e.g. exposure to a causative agent). There is a genetic element in the susceptibility of individuals to some of these triggers which has been traced to particular HLA genotypes (i.e. genetic "self" identifiers used by the immune system). However, even in those who have inherited the susceptibility, type 1 diabetes mellitus seems to require an environmental trigger. A small proportion of type 1 diabetics carry a mutation that causes maturity onset diabetes of the young (MODY).

There is an even stronger inheritance pattern for Type 2 diabetes; those with type 2 ancestors or relatives have very much higher chances of developing Type 2. It is also often connected to obesity, which is found in approximately 85% of (North American) patients diagnosed with that form of the disease, so inheriting a tendency toward obesity seems also to contribute. Age is also thought to be a contributing factor, as most type 2 patients in the past were older. The exact reasons for these connections are unknown.

7.2.2 How is Diabetes Treated?

Although diabetes cannot be cured, it can be treated very successfully.

Type 1 diabetes is treated by injections of insulin and a healthy diet. Type 2 diabetes is treated by a healthy diet or by a combination of a healthy diet, sport and tablets. Sometimes people with Type 2 diabetes also have insulin injections, although they are not totally 'dependent' on the insulin.

Treatments for Type 1 diabetes

People with Type 1 diabetes need injections of insulin for the rest of their lives and also need to eat a healthy diet that contains the right balance of foods. Insulin cannot be taken by mouth because it is destroyed by the digestive juices

in the stomach. People with this type of diabetes commonly take either two or four injections of insulin each day.

Treatments for Type 2 diabetes

People with Type 2 diabetes need to eat a healthy diet that contains the right balance of foods and may also need to take tablets. Sometimes the combination of a healthy diet and exercise is sufficient and the tablets are not necessary.

There are several kinds of tablets for people with Type 2 diabetes. Some kinds help the pancreas to produce more insulin. Other kinds help the body to make better use of the insulin that the pancreas does produce. Another type of tablet slows down the speed at which the body absorbs glucose from the intestine.

7.2.3 How Can We Use HRV to Treat Diabetes?

Cardiovascular complications are the main cause of death in people with diabetes. Early, asymptomatic changes are due to autonomic nervous system dysfunction, which if identified can lead to improved health. Indeed, diabetes can cause severe autonomic dysfunction that can be responsible for several disabling symptoms, including sudden cardiac death. Although traditional measures of autonomic function are able to document the presence of neuropathy, in general they are only abnormal when there is severe symptomatology, i.e. when the pathology is already obvious. Thus by the time changes in function are evident, the natural course of autonomic neuropathy is well established. HRV and sudden cardiac death Ventricular tachyarrhythmias represent a leading cause of sudden cardiac death(SCD) in the community.

The pathophysiology of SCD is probably an interaction between an abnormal anatomical substrate such as coronary artery disease with associated myocardial scarring, left ventricular hypertrophy or cardiomyopathy, and transient functional disturbances which trigger the terminal dysrhythmia. This may include factors such as ischaemia, premature beats, electrolyte disturbance and fluctuations in autonomic balance. A recent study reported that decreased HRV was more predictive of subsequent arrhythmic events than the presence of late potentials (recorded from areas of conduction delay in the ventricles which provide one of the substrates for re-entrant arrhythmias), Holter-derived arrhythmias, treadmill exercise test results or left ventricular ejection fraction. In multivariate analysis of combinations of risk factors, the combination of late potentials recorded by the signal averaged ECG and reduced HRV was more predictive than any other combination.

The application of heart rate variability to the diagnosis of diabetes is still fairly new to the public: according to the results of a survey in the United States released on April 20th 2005, 83% of the estimated 16 million Americans with diabetes have never heard of heart rate variability testing.

7.3 Results

In this section we will present the filtering and cleaning of a real data set given by the Michael Beuern Allgemeines Krankenhaus (general hospital) in Vienna, Austria.

7.3.1 Data

The data set under consideration is the tachogram of a patient with diabetes who was asked to sit down and stand up at certain time points. The data set also contains a manually cleaned version of the tachogram, i.e. a tachogram that a doctor has looked at and corrected based on his experience. Until now, this is the way the filtering has been done, which is why the new filtering algorithms may come as something of a relief to the doctors' strained eyes. The data set itself is quite big (more than 2,000 values): filtering the whole data set would take literally hours (even days) and it is very difficult to see anything on the graph on such a huge scale. Therefore we have only filtered the first 200 values. In practical cases, we conjecture that filtering the whole time series at once is not necessary and that cutting it in big enough chunks and plate the filtered chunks together might be sufficient: a research with medical doctors would be necessary to validate or not this conjecture.

7.3.2 Filtering and Cleaning

Manually Cleaned Data Set

Figure 7.3 page 99 shows the original data set and the manually cleaned one. The big jolts of the original process, which are due to a change in the patient position (from sitting down to standing up or converse) are removed.

Repeated Median Filter

This filter, the output of which can be seen on Figure 7.4 page 99, performs surprisingly well in that its output is not too far from the manually cleaned tachogram.

Biweight Filter

The biweight filter (see Figure 7.5 page 100) seems to have a bit more trouble than the repeated median: it has jolts and lies far from the manually cleaned process.

Biweight Filter Cleaner

As the output of the biweight filter is bad, it is not surprising that the biweight filter cleaner also gives poor results (on Figure 7.6 page 100), which is a bit unfortunate as it is the most time-consuming algorithm.

Repeated Median Cleaner

This algorithm performs better than the biweight filter cleaner because the output of the filter is better. See Figure 7.7 page 101.

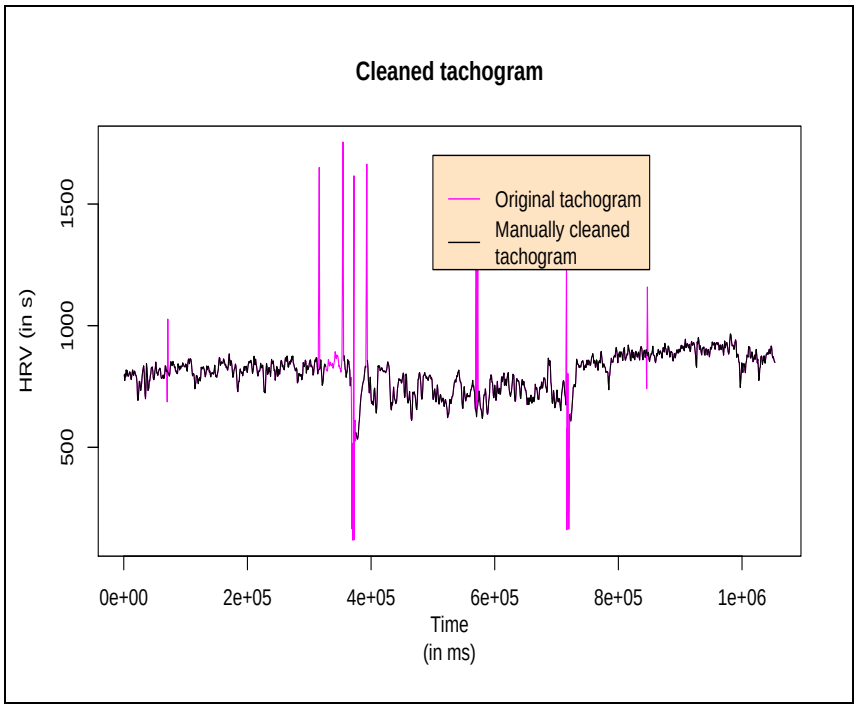

Figure 7.3: Manually cleaned tachogram

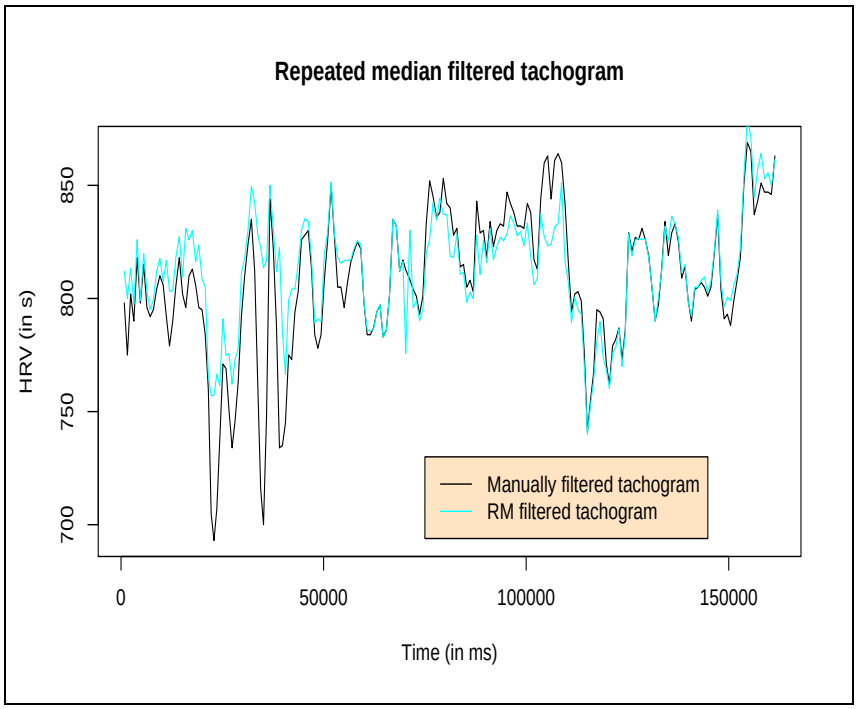

Figure 7.4: Repeated Median filtered tachogram

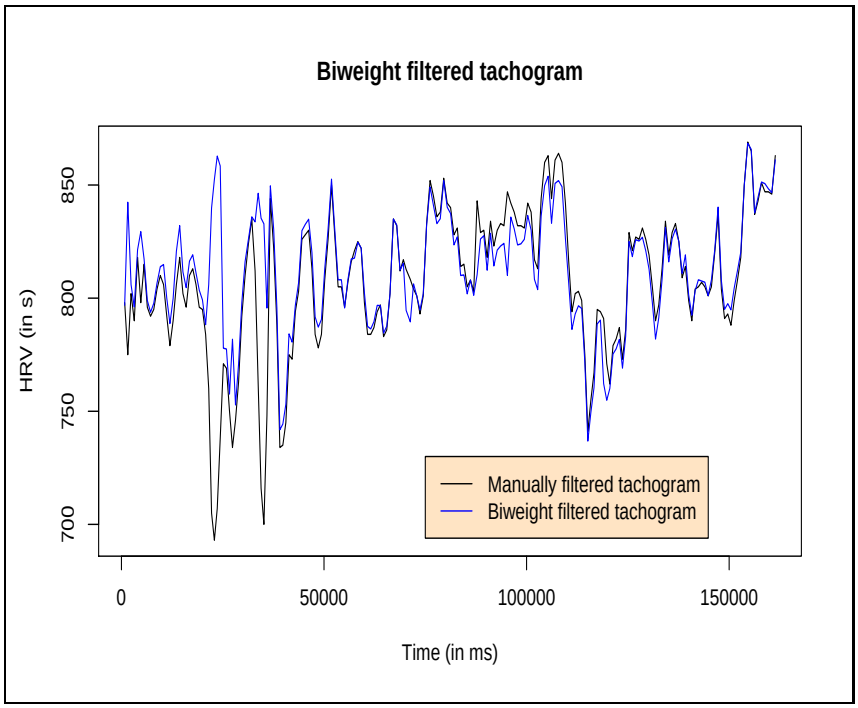

Figure 7.5: Biweight filtered tachogram

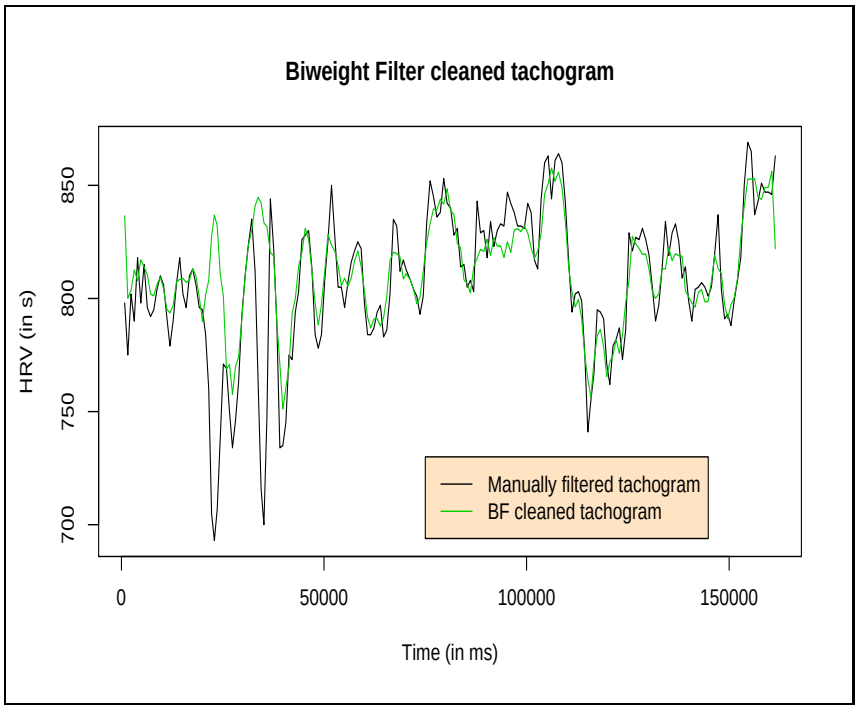

Figure 7.6: BF cleaned tachogram

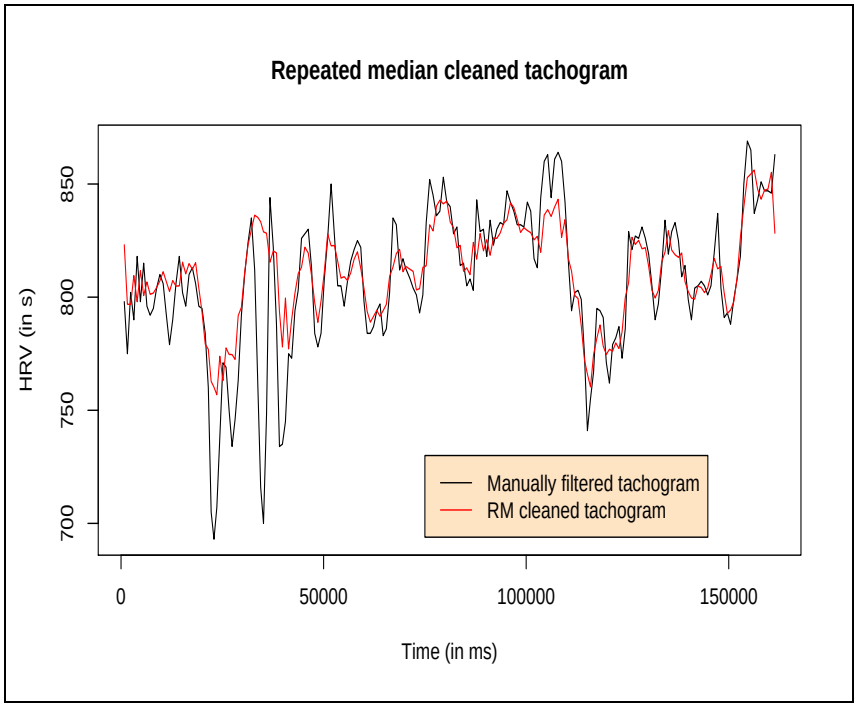

Figure 7.7: Repeated Median cleaned tachogram

7.3.3 Conclusion

The main problem with these algorithms are their speed: doctors would want to have a result in a few seconds whereas the version of the biweight filter cleaner we implemented in C++ took several hours to produce the cleaned version of the complete time series! Tatum and Hurvich propose the use of the least-squares regression as starting values for the biweight regression so as to reduce the computational cost. We did not investigate this, nor did we try to implement other ideas we had. One of them is to divide the times series into several chunks. Indeed, what is really costly is the repeated matrix inversions we perform. As we know that it takes $o(n^2)$ steps to calculate this inverse, if we divide the time series into two chunks of length n_1 and n_2 such that $n = n_1 + n_2$, we would need $o(n^2)$ steps to invert each matrix of the full data set, but only $o(n_1^2)$ and $o(n_2^2)$ operations to invert each of the submatrices. We therefore have a difference in time of $2n_1n_2$. Of course, this gain comes at a price, as we lose a lot of information by cutting the time series in two (all the "cross terms" are lost, which means we do not take into account the interaction between the beginning and the end of the time series).

More generally, if we cut the series into p chunks such that

$$n = \sum_{i=1}^{p} n_i,$$

the difference in computation time is

$$o\left(\sum_{i=1}^{p}\sum_{j<i} n_i n_j\right)$$

If we suppose that the chunks are of equal length, this reduces to $(n/p)^2$. Already, several questions arise:

1. How much information is lost during the process?

2. How should we divide the time series? Empirically or with a certain measure (such as AIC_C or a measure of the "variability" of the time series over a certain period of time if it is not homoscedastic) and if so, which measure should we use? How many chunks should we make?

3. How do we put the information of the different chunks back together? Do we run the full algorithm on each chunk separately and then recombine by averaging or do we put the chunks together at some point in the algorithm? If we use averaging, should we do a weighted average by downweighting the high variability chunks?

This idea is not the only one to increase speed. Tatum and Hurvich conjecture that instead of using the repeated median as starting values for the biweight regression, we could use the ordinary least squares and still have a (strictly) positive breakdown bound and this would reduce the computational burden dramatically. Moreover, our implementation of the algorithm is probably not optimal.

Chapter 8

Conclusion

In this report, we presented successively four algorithms to remove outliers from a time series and we applied the last one to the practical problem of the use of heart rate variability for the diagnosis of diabetes. Of course, such an algorithm can be used in a variety of different contexts other than the medical one: for example, Tatum and Hurvich use the biweight filter cleaner to clean the weekly Dutch auction divided rates for Citibank.

The first algorithm was the simple, yet efficient, repeated median algorithm. The second, the biweight filter algorithm, was an attempt to "robustify" it, the third one, the biweight filter cleaner, arose as an attempt to improve the quality of the output and the last one, the repeated median cleaner, tried to reconcile the "best of both worlds" in that it had the exact fit property of the repeated median and the goodness-of-fit of the biweight filter cleaner. The repeated median is rough but fairly fast, whereas the biweight filter cleaner is very slow but yields extremely good results. The repeated median cleaner is a compromise between the two.

The practical implementation of these algorithms was done in two programming languages: C++ and the statistical programming language R. We tried to use a convergence accelerator in our implementation because the algorithms are still extremely slow (when it is confronted to a real data set, they can take hours to compute the result). The convergence accelerator we chose (ε-aglorithm) unfortunately failed to converge to the values of the parameter in the biweight regression. This means that the type of sequences in the biweight regression algorithm are not in the kernel of the ε-algorithm. This does not necessarily mean that we have to reject convergence acceleration alltogether, but we need to design an algorithm more carefully if we choose to use this method. As we removed the ε-algorithm from our implementation, we presented a few ideas to speed up the program in the last chapter of this report.

This report is far from being exhaustive on the study of these algorithms. First, we did not try to optimize the values of the parameters of the various algorithms because we failed to find an answer to the simple yet embarrassing

question "optimal for what?". By this, we mean that we suspect that the optimal values of the parameters depend strongly on the model coefficients. We also conjecture that the variance of the contaminating process plays a decisive role in the quality of the output of the various filters and cleaners. Moreover, we did not implement any of the suggestions we made in the last chapter and doing so would probably require a substantial amount of work, at least from the theoretical point of view. In the following appendices, we give the theoretical background which we (implicitly or not) referred to constantly in the text.

Chapter 9

Appendix A: Robustness Concepts

In this appendix, we will present notions such as M-estimators, robust regression and influence functions. They are implicitly referred to in this thesis.

9.1 M-estimators

The name "M-estimators" comes from "generalized Maximum likelihood estimators". Indeed, a motivation behind M-estimators can be to generalize maximum likelihood estimators. When using a maximum likelihood estimator, the aim is to maximize $\prod_{i=1}^{n} f(x_i)$ or, equivalently, minimize $\sum_{i=1}^{n} -\log f(x_i)$. In 1964, Huber proposed generalising this to the minimization of $\sum_{i=1}^{n} \rho(x_i)$, where ρ is some function. Maximum likelihood estimators are therefore a special case of M-estimators

9.1.1 Estimators

Mathematical Context

In this section, we will assume the following:

- $(\Omega, \mathfrak{A}, P)$ is a probability space,
- $(\mathcal{X}, \Sigma)$ is a measure space called *state space*,
- Δ_x is the Dirac mass with mass 1 in x and 0 elsewhere,
- (Θ, S) is a measure space called *parameter space*,
- $F := \{\mathsf{x} : (\Omega, \mathfrak{A}) \to (\mathcal{X}, \Sigma)\}$ is the set of all random variables on $\mathcal{X}$,
- $\mathcal{F}(\mathcal{X})$ is the set of all probability distibutions on $\mathcal{X}$.

Definitions

An estimator is just a statistic.

Definition 9 *(Estimator)*

For $n \in \mathbb{N}^*$, an *estimator* is any measurable function

$$T_n : \left\{ \begin{array}{ccc} F^n & \to & \Theta \\ (\mathsf{x}_1, \cdots, \mathsf{x}_n) & \mapsto & T_n(\mathsf{x}_1, \cdots, \mathsf{x}_n) \end{array} \right.$$

♢

Definition 10 *(Estimating sequence)*

If $\forall n \in \mathbb{N}^*$, T_n is an estimator, then $(T_n)_{n\in\mathbb{N}^*}$ is called an *estimating sequence*.

♢

Definition 11 *(Functional)*

For $n \in \mathbb{N}^*$, an estimator T_n is a *functional* if:

$$\exists T : \left\{ \begin{array}{ccc} \mathcal{F}(\mathcal{X}) & \to & \Theta \\ G & \mapsto & T(G) \end{array} \right. , \forall (\mathsf{x}_1, \cdots, \mathsf{x}_n) \in F^n, T_n(\mathsf{x}_1, \cdots, \mathsf{x}_n) = T(G_n)$$

with

$$G_n := \frac{1}{n} \sum_{i=1}^{n} \Delta_{\mathsf{x}_i}$$

♢

Definition 12 *(Asymptotic value)*

Let $n \in \mathbb{N}^*$ and $(\mathsf{x}_1, \cdots, \mathsf{x}_n) \in F^n$ be independant and identically distributed according to $G \in \mathcal{F}(\mathcal{X})$. Furthermore, let $(T_n)_{n\in\mathbb{N}^*}$ be an estimating sequence.

If $\exists T : \mathcal{F}(\mathcal{X}) \to \Theta$ such that $T_n(\mathsf{x}_1, \cdots, \mathsf{x}_n) \xrightarrow[n\infty]{} T(G)$, then T is called *asymptotic value* of the estimating sequence $(T_n)_{n\in\mathbb{N}^*}$ and we say T_n can be asymptotically replaced by the functional T at distribution G.

♢

Desirable Properties

In the whole of this report, we consider estimators which are functionals or can asymptotically be replaced by functionals. We will also assume the following property to be fullfiled:

Definition 13 *(Asymptotic normality)*

Let $(T_n)_{n\in\mathbb{N}^*}$ be an estimating sequence which can be asymptotically replaced by a functional T at distribution G. $(T_n)_{n\in\mathbb{N}^*}$ is said to be *asymptotically normal* if:

$$\mathcal{L}\left(\sqrt{n}[T_n - T(G)]\right) \xrightarrow[n\infty]{} \mathcal{N}(0,\ V(T,G)) \text{ weakly}$$

$V(T,G)$ is called *asymptotic variance*

♢

Definition 14 *(Fisher Consistency)*

A functional $T : \mathcal{F}(\mathcal{X}) \to \Theta$ is said to be *Fisher consistent* if, for any distribution $F_\theta \in \mathcal{F}(\mathcal{X})$ depending on a parameter $\theta \in \Theta$,

$$T(F_\theta) = \theta$$

♢

9.1.2 Types of M-estimators

Minimizing $\sum_{i=1}^n \rho(x_i)$ can often be done by solving $\sum_{i=1}^n \rho'(x_i) = 0$, but not always. We will therefore distinguish these two problems, although in most practical cases, we will only try to solve $\sum_{i=1}^n \rho'(x_i) = 0$.

Definition 15 *(M-estimators of ρ-type)*

An M-estimator of ρ-type is a functional $T : F \in \mathcal{F}(\Sigma) \mapsto T(F) \in \Theta$ defined through a measurable function $\rho : (\Theta, S) \times (\mathcal{X}, \Sigma) \to (\Theta, S)$ (usually $\Theta = \mathbb{R}$) which maps a probability distribution to the value of $\hat{\theta}$ (if it exists) that minimizes $\int_{\mathcal{X}} \rho(\theta, x) dF(x)$.

♢

ρ is not always differentiable, which can complicate the calculation of the estimator. For most practical cases, we use the following type of estimators:

Definition 16 *(M-estimators of ψ-type)*

An M-estimator of ψ-type is a functional $T : F \in \mathcal{F}(\Sigma) \mapsto T(F) \in \Theta$ defined through a measurable function $\psi : (\Theta, S) \times (\mathcal{X}, \Sigma) \to (\Theta, S)$ which maps a probability distribution to the solution $\hat{\theta}$ (if it exists) of the vector equation $\int_{\mathcal{X}} \psi(\theta, x) dF(x) = 0$.

♢

Such an estimator is not necessarily an M-estimator of ρ-type, but if ρ has a continuous first derivative with respect to θ, then a necessary condition for the corresponding M-estimator of ψ-type to be an M-estimator of ρ-type is $\psi(\theta, x) = \nabla_\theta \rho(\theta, x)$. The previous definitions can easily be extended to finite samples:

Definition 17 *(Finite-sample M-estimators)*

For $n \in \mathbb{N}^*$, an estimator T_n is said to be an M-estimator of ρ-type (or of ψ-type) if there exists an M-estimator of ρ-type (or of ψ-type) such that $\forall(\mathsf{x}_1, \cdots, \mathsf{x}_n) \in F^n, T_n(\mathsf{x}_1, \cdots, \mathsf{x}_n) = T(G_n)$, with $G_n := \frac{1}{n}\sum_{i=1}^n \Delta_{\mathsf{x}_i}$.

♢

9.2 Robustness Concepts

9.2.1 Breakdown Point

General Definition

This definition is built using probability distributions. It describes what happens to an estimator when we move away from the distribution for which it was designed. We therefore need a metric. There are several variants of the breakdown point, depending on the metric used.

Definition 18 *(Kolmogorov distance)*

- $P, Q : \mathbb{R} \to [0, 1]$ are two distribution functions

$$d_{ko}(P, Q) := \sup_{x \in \mathbb{R}} |P(]-\infty, x]) - Q(]-\infty, x])|$$

♢

Example: Figure 9.1 page 113 is an illustration of the Kolmogorov distance. We have plotted the distribution functions of $\mathcal{N}(0,\ 0.3)$ and $\mathcal{N}(0,\ 3)$. The red line corresponds to the maximum distance between the two distributions and by definition its length is the Kolmogorov distance between the two distributions.

Definition 19 *(Prohorov metric)*

Let (S, d) be a separable metric space and P, Q any two laws (probability measures on the Borel σ-algebra $\mathfrak{B}(S)$) on S. Then the *Prokhorov metric* is defined by

$$d_0(P, Q) := \inf_{\varepsilon \in \mathbb{R}_+^*} \{\varepsilon : \forall A \in \mathfrak{B}(S), P(A) \leq Q(A^\varepsilon) + \varepsilon\}$$

with

$$\forall \varepsilon \in \mathbb{R}_+^*, A^\varepsilon := \{y \in S : \exists x \in A, d(x, y) < \varepsilon\}$$

♢

Definition 20 *(Lipschitz function)*

Let (S, d) be a separable metric space and $f : S \to \mathbb{R}$. f is Lipschitz if:

$$\|f\|_L := \sup_{x \neq y, (x,y) \in S^2} \left\{ \frac{|f(x) - f(y)|}{d(x, y)} \right\} < +\infty$$

♢

Definition 21 *(Dual-bounded Lipschitz metric)*
Let (S, d) be a separable metric space and $f : S \to \mathbb{R}$. Using

$$\|f\|_\infty := \sup_{x \in S} \{|f(x)|\}$$

the *dual-bounded Lipschitz metric* is defined as:

$$\|f\|_{BL} := \|f\|_L + \|f\|_\infty$$

♢

Both the Prohorov metric and the dual-bounded Lipschitz metric define metrics, for the same topology, on the set of all laws on S and metricize weak convergence.

Definition 22 *(Breakdown point)*
The *breakdown point* ε^* of the sequence of estimators $(T_n)_{n \in \mathbb{N}^*}$ at F is defined by:

$$\varepsilon^* := \sup \{\varepsilon \leq 1 : \text{there is a compact set } K_\varepsilon \subset \Theta \text{ such that}$$

$$(e(F, G) < \varepsilon) \Rightarrow G(\{T_n \in K_\varepsilon\}) \xrightarrow[n\infty]{} 1\}$$

where $e = d_0, d_{BL}$ or d_{ko}.

♢

Example: The mean has a breakdown point of 0. Indeed, given a set of values, the mean can be made as large as desired by changing only one point, as illustrated on Figure 9.2 page 113.

Finite Sample Version

Definition 23 *(Finite-sample breakdown point)*
The *finite-sample breakdown point* ε_n^* of the functional T_n at the sample $(\mathsf{x}_1, \cdots, \mathsf{x}_n)$ is given by:

$$\varepsilon_n^*(\mathsf{x}_1, \cdots, \mathsf{x}_n) := \frac{1}{n} \max \left\{ m : \max_{i_1, \cdots, i_n} \sup_{y_1, \cdots, y_n} |T_n(z_1, \cdots, z_n)| < +\infty \right\}$$

where $(z_1, \cdots, z_n)$ is obtained by replacing the m data points $\mathsf{x}_{i_1}, \cdots, \mathsf{x}_{i_m}$ by arbitrary values $y_1, \cdots, y_n$.

♢

In many cases, $\varepsilon_n^* \xrightarrow[n\infty]{} \varepsilon^*$.

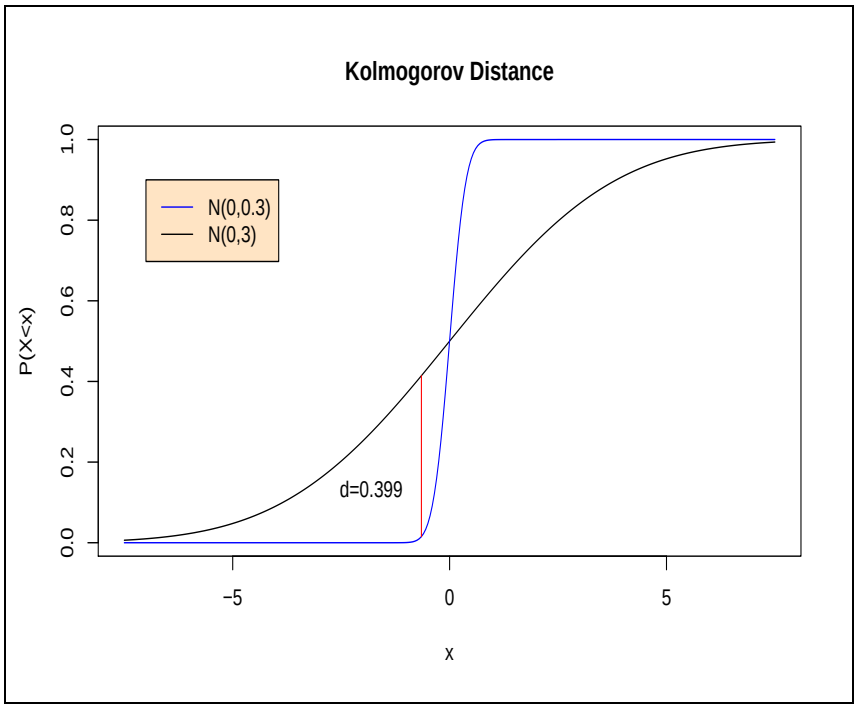

Figure 9.1: Kolmogorov distance between two normal distributions

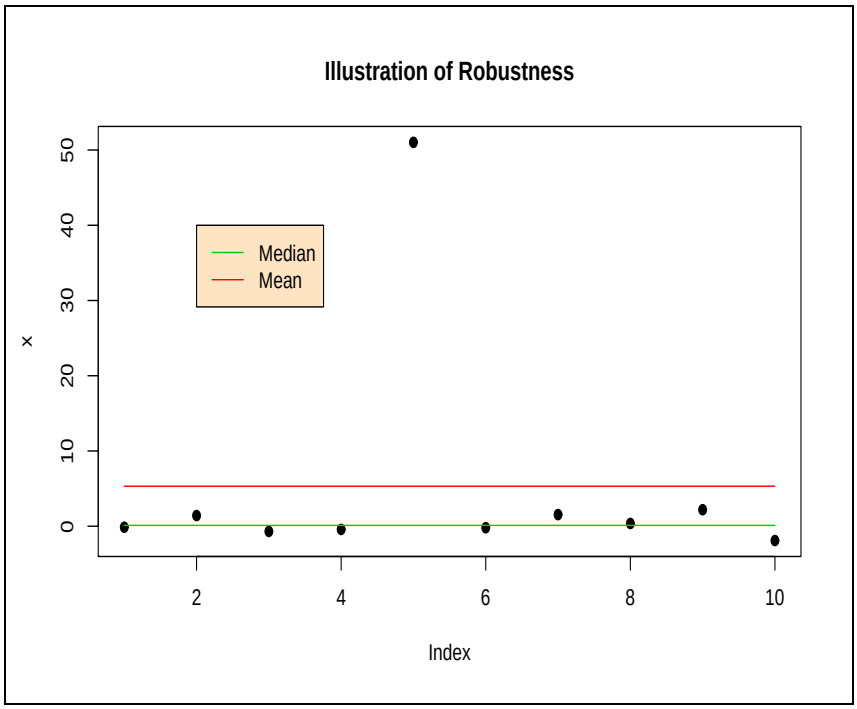

Figure 9.2: Mean is not a robust estimator

Maximum Breadown Point

A breakdown point is usually lower than 1/2, intuitively because if we replace more than half of the x_i's by arbitrary values, we will have no way of knowing which values are authentic. More precisely, we have the following theorem:

Theorem 2 *(Maximum breakdown point)*
Let $\Theta \subseteq \mathbb{R}$ be a parameter space. Let $\mathcal{P}'$ be a closed set under affine transformations of probability measures of $\mathfrak{B}(\mathbb{R})$ and e be a metric on $\mathcal{P}'$. $\forall n \in \mathbb{N}, T_n : \mathcal{P}' \to \Theta$ is a sequence of equivariant real-valued statistics and $dP_0 : \Theta \to [0,1]$ is a law on $\mathbb{R}$. $\forall \theta \in \Theta$, we define $dP_\theta : x \mapsto dP_0(x-\theta)$. Let e be d_0 or d_1. Then if $(T_n)_{n\in\mathbb{N}}$ has a breakdown point $> 1/2$ at P_θ for all $\theta \in \Theta$, it cannot be a consistent sequence of estimators of θ for any $\theta \in \Theta$.

□

This means that all consistent estimators have a breakdown point lower than 1/2.

9.2.2 Influence Function

Derivation

The influence function represents the behavior of a functional T at a distribution F, contaminated infinitesimally by a distribution Δ_x with all its weight in a point x. We would like to see what happens to an estimator when we move from F towards Δ_x, therefore the influence function will be the directional derivative of T at F, in the direction of Δ_x.

First, we recall that the directional derivative of T at p in the direction v is given by:

$$dT_p(v) := \left[\frac{dT(c(t))}{dt}\right]_{t=0}$$

with $c(t)$ such that $c(0) = p$ and $c'(0) = v$.
Then, we replace p by F and v by $\Delta_x - F$.

Definition 24 *(Influence function)*
$(\Omega, \mathfrak{A}, P)$ is a probability space, $(\mathcal{X}, \Sigma)$ is a measure space, $\mathcal{F}(\mathcal{X})$ is the set of all probability distributions on $\mathcal{X}$ and $T : \mathcal{F}(\mathcal{X}) \to \mathcal{X}$ is a functional.

$$\forall x \in \mathcal{X}, IF(x; T, F) := dT_F(G - F) = \lim_{t\to 0+} \frac{F + t(\Delta_x - F)}{t}$$

♢

Often we cannot compute the influence function algebrically but we can compute successive values of a functional T_n. We can then use an *empirical influence function*.

Definition 25 *(Empirical influence function)*
Given a functional T_n, the empirical influence function is defined by:

$$x \mapsto T_n(\mathsf{x}_1, \cdots, \mathsf{x}_n - 1, x)$$

♢

The following theorem is of crucial importance, as it states that the influence function of an M-estimator of ψ type is proportional to its defining ψ function. This explains why the function ψ is somtimes (abusively) called "influence function".

Theorem 3 *(Influence function of an M-estimator)*
Let T be an M-estimator of ψ type. Let G be a probability distribution for which $T(G)$ is defined and let $x \in \mathcal{X}$.

$$IF(x; T, G) = -\frac{\psi(x, T(G))}{\int \left[\frac{\partial \psi(y,\theta)}{\partial \theta}\right]_{y=T(F)} dF(y)}$$

□

Proof:
By definition, $\forall G \in \text{dom}(T), \int \psi(x, T(G))dG(x) = 0$. Let $c(0) = G$ and $c'(0) = \Delta_x - G$, for example $c(t) := G + t(\Delta_x - G)$.Then

$$\forall t \in \mathcal{X}, \int \psi(y, T(c(t)))d(c(t)(y)) = 0$$

Differentiating yields

$$\forall t \in \mathcal{X}, \frac{\partial}{\partial t} \int \psi(y, T(c(t)))d(c(t)(y)) = 0$$

We know that $dc(t) = td(\Delta_x - G) + dF$. Therefore,

$$\forall t \in \mathcal{X}, \frac{\partial}{\partial t} \int \psi(y, T(c(t)))td(\Delta_x - G)(y) + \frac{\partial}{\partial t} \int \psi(x, T(c(t)))dG(y) = 0$$

Supposing differentiation and integration can be interchanged,

$$\forall t \in \mathcal{X}, t \int \frac{\partial \psi(y, T(c(t)))}{\partial t} d(\Delta_x - G)(y) + \int \psi(y, T(c(t))) d(\Delta_x - G)(y)$$

$$+ \int \frac{\partial \psi(x, T(c(t)))}{\partial t} dG(x) = 0$$

As

$$\int \psi(y, T(c(t))) d(\Delta_x - G)(y)$$

$$= \int \psi(y, T(c(t))) d(\Delta_x)(y) - \int \psi(y, T(c(t))) dG(y) = \psi(x, T(c(t))) - 0$$

$$\forall t \in \mathcal{X}, \psi(x, c(t)) + t \int \frac{\partial \psi(y, T(c(t)))}{\partial t} d(\Delta_x - G)(y) + \int \frac{\partial \psi(x, T(c(t)))}{\partial t} dG(x) = 0$$

Now, $\frac{\partial \psi(y, T(c(t)))}{\partial t} = \left[\frac{\partial \psi(x,\theta)}{\partial \theta}\right]_{T(c(t))} \frac{\partial T(c(t))}{\partial t}$.

Therefore,

$$\forall t \in \mathcal{X}, \psi(x, c(t)) + t \int \frac{\partial \psi(y, T(c(t)))}{\partial t} d(\Delta_x - G)(y) +$$

$$\int \left[\frac{\partial \psi(x, \theta)}{\partial \theta}\right]_{T(c(t))} dG(x) \frac{\partial T(c(t))}{\partial t} = 0$$

As this equation is valid for all t in $\mathcal{X}$, we can take $t = 0$:

$$\psi(x, T(G)) + \int \left[\frac{\partial \psi(x, \theta)}{\partial \theta}\right]_{T(G)} dG(x) \left[\frac{\partial T(c(t))}{\partial t}\right]_{t=0} = 0$$

By definition, $\left[\frac{\partial T(c(t))}{\partial t}\right]_{t=0} = d_G T(\Delta_x - G) = IF(x; T, G)$, hence

$$IF(x; T, G) = -\frac{\psi(x, T(G))}{\int \left[\frac{\partial \psi(y,\theta)}{\partial \theta}\right]_{y=T(G)} dG(y)}$$

♦

9.3 Robust Regression

9.3.1 Regression Analysis

Let $p \in \mathbb{N}^*$. $(\Omega, \mathcal{A}, P)$ will denote a probability space and, $\forall j \in [\![1, p]\!]$, (Γ_j, S_j) will be a measure space. $\Theta \subseteq \mathbb{R}^p$ is a set of coefficients. The *response variable* is

a random variable $\mathsf{y} : (\Omega, \mathcal{A}) \to (\mathbb{R}, \mathfrak{B})$. This variable will be "explained" using other random variables called factors, i.e. we will write y as a function of factors.

$\forall j \in \{1, \cdots, p\}, \mathsf{x}_j : (\Omega, \mathcal{A}) \to (\Gamma_j, S_j)$, is called a *factor*.

Let $\eta : \left\{ \begin{array}{ccc} \Gamma_1 \times \cdots \times \Gamma_p \times \Theta & \to & \mathbb{R} \\ (\mathsf{x}_1, \cdots, \mathsf{x}_p; \boldsymbol{\theta}) & \mapsto & \eta(\mathsf{x}_1, \cdots, \mathsf{x}_p, \boldsymbol{\theta}) \end{array} \right.$

We then define $\varepsilon := \mathsf{y} - \eta(\mathsf{x}_1, \cdots, \mathsf{x}_p; \overline{\boldsymbol{\theta}})$, which means that

$$\mathsf{y} = \eta(\mathbf{x}; \boldsymbol{\theta}) + \varepsilon$$

where $\mathbf{x} = (\mathsf{x}_1, \cdots, \mathsf{x}_p)$

We suppose that there exists a vector of (unknown) parameters $\overline{\boldsymbol{\theta}} \in \Theta$ called *true parameters*, such that:

$$\mathbb{E}[\mathsf{y}|\mathbf{x}] = \eta(\mathbf{x}, \overline{\boldsymbol{\theta}})$$

The general form of the function η is known. In fact, the only unknown in the equation is $\overline{\boldsymbol{\theta}}$. The aim of regression is, given a set of data, to find an estimate $\widehat{\boldsymbol{\theta}}$ of $\overline{\boldsymbol{\theta}}$ optimizing some criterion.

9.3.2 Least Squares and its Limitations

Linear Regression

Linear regression is the most common case in practice because it is the easiest to compute and gives good results. Indeed, by restraining the variations of the factors to a "small enough" domain, the response variable can be approximated locally by a linear function of $\boldsymbol{\theta}$. When we do a linear regression, we are implicitly supposing that given a set of factors $\mathsf{x}_1, \cdots, \mathsf{x}_p$, the best approximation of the response variable y we can find is a linear combination of theses factors $\mathsf{x}_1, \cdots, \mathsf{x}_p$. We are therefore also supposing that $\forall j \in [\![1, p]\!], (\Gamma_j, S_j) = (\mathbb{R}, \mathfrak{B}(\mathbb{R}))$. The aim of linear regression is to find a good estimator of the right coefficients $\overline{\boldsymbol{\theta}}$ of this linear combination.

We choose η the following way:

$$\eta(\mathbf{x}, \boldsymbol{\theta}) = \sum_{j=1}^{p} \theta^j \mathsf{x}_j$$

Geometrical Interpretation

Let F be the L^2-space of all real-valued random variables whose square has a finite Lebesgue integral. We suppose that $\mathsf{y} \in F$ and that $\forall j \in [\![1, p]\!], \mathsf{x}_j \in F$. Let G be the subspace generated by $(\mathsf{x}_1, \cdots, \mathsf{x}_p)$. We can build a scalar product in F with $\langle \mathsf{u}, \mathsf{v} \rangle_2 := \mathbb{E}[\mathsf{uv}]$ (it is indeed a scalar product because if $\|\mathsf{u}\|_2^2 = 0$, then $\mathsf{u} = 0$ almost surely).

It is easy to show that ε is orthogonal to any x_j and hence to the whole of the subspace G, which means that η is the projection of y on G, orthogonal with respect to the scalar product we have just defined. We have therefore shown:

$$\|\mathsf{y} - \eta(\mathbf{x}; \overline{\boldsymbol{\theta}})\|_2^2 = \min_{\mathsf{f} \in G} \|\mathsf{y} - \mathsf{f}\|_2^2$$

An illustration of this can be seen on Figure 9.3 page 120.

Estimating the Projection

We now suppose that, for each factor $\mathsf{x}_j, j \in [\![1, p]\!]$, we have a sample of size $n \in \mathbb{N}^*$, $n > p : \mathbf{X}_j := (\mathsf{x}_j^1, \cdots, \mathsf{x}_j^n)$ and that we have the corresponding sample of y: $\mathbf{y} = (\mathsf{y}^1, \cdots, \mathsf{y}^n)$. Then we can build a matrix X where each line represents an experiment:

$$\mathsf{X} = \begin{bmatrix} \mathsf{x}_1^1 & \cdots & \mathsf{x}_p^1 \\ \vdots & & \vdots \\ \mathsf{x}_1^n & \cdots & \mathsf{x}_p^n \end{bmatrix}$$

This is a matrix of random variables often called design matrix (for experimental designs). Each column represents a factor. As we have n trials and p factors, it is a $n \times p$ matrix. We also have a corresponding error vector (of size n):

$$\boldsymbol{\varepsilon} = \mathbf{y} - \eta(\mathsf{X}; \overline{\boldsymbol{\theta}})$$

Based on the observations $\mathbf{y} = (\mathsf{y}^1, \cdots, \mathsf{y}^n)$ and on the design matrix X, we would like to estimate the unkown parameters $\overline{\boldsymbol{\theta}} = (\theta^1, \cdots, \theta^n)$ (one per factor).

Under assumptions which are met relatively often, there exists an optimal solution to the linear regression problem. These assumptions are called Gauß-Markov assumptions.

Gauss-Markov Assumptions

We suppose that $\mathbb{E}\boldsymbol{\varepsilon} = \mathbf{0}$ and that $\mathbb{V}\boldsymbol{\varepsilon} = \sigma^2 \mathbf{I}_n$ (uncorrelated, but not necessarily independent) where $\sigma^2 < +\infty$ and $\mathbf{I}_n$ is the $n \times n$ identity matrix.

Least-squares Estimator

We can build an estimation of the coefficients of the orthogonal projection. To do this, we can use an estimation of the scalar product defined earlier.

For all couples of samples of size n, $\mathbf{u}, \mathbf{v} \in F^n$ of random variables u and v, we define $\langle \mathbf{u}, \mathbf{v} \rangle := \frac{1}{n} \mathbf{u}^\intercal \mathbf{v}$, and the corresponding norm is: $\|\cdot\| := \sqrt{\langle \cdot, \cdot \rangle}$. Note that the scalar product $\langle \cdot, \cdot \rangle$ is defined in F^n and no longer in F. $\langle \cdot, \cdot \rangle$ is an unbiased estimator of the scalar product $\langle \cdot, \cdot \rangle_2$.

We want to find the value $\widehat{\boldsymbol{\theta}}_n^{LS} \in \Theta$, if it exists, which minimizes $\|\mathbf{y} - \eta(\mathbf{X}, \boldsymbol{\theta})\|^2$.

Computation of the Estimator $\widehat{\boldsymbol{\theta}}_n^{LS}$

We are using an orthogonal projection to estimate $\overline{\boldsymbol{\theta}}$, therefore:

$$\forall j \in [\![1, p]\!], \langle \mathbf{X}_j, \boldsymbol{\varepsilon} \rangle = 0,$$

hence $\mathbf{X}^T(\eta(\mathbf{X}; \widehat{\boldsymbol{\theta}}_n^{LS}) - \mathbf{y}) = \mathbf{0}$. As $\eta(\mathbf{X}; \widehat{\boldsymbol{\theta}}_n^{LS}) = \mathbf{X}\widehat{\boldsymbol{\theta}}_n^{LS}$, this equation yields $\mathbf{X}^T\mathbf{X}\widehat{\boldsymbol{\theta}}_n^{LS} = \mathbf{X}^T\mathbf{y}$. If $\mathbf{X}$ is of full rank, then so is $\mathbf{X}^T\mathbf{X}$. In that case:

$$\widehat{\boldsymbol{\theta}}_n^{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

9.3.3 Quality and Limitations of the Least-squares Estimator

Apart from being very easy to compute, the least-squares estimator is, under the Gauß-Markov assumptions, the most efficient estimator for $\overline{\boldsymbol{\theta}}$, as shown by the following theorem.

Theorem 4 *(Gauss-Markov)*

Under the Gauß-Markov hypothesis, of all unbiased estimators $\widehat{\boldsymbol{\theta}}$ of $\overline{\boldsymbol{\theta}}$ which depend linearly on $\mathbf{y}$, the least-squares estimators are the most efficient ones. □

Unfortunately, the Gauß-Markov hypothesis is fairly stringent and is often not fulfilled in practice: if the $\mathbf{x}_j$ are correlated, which is often the case in the study of time series, the results can be quite significantly corrupted. What is more, the least-squares estimator is very sensitive to outliers: a rather naïve example of this is given in Figure 9.4 page 120: all points lie on the same line, except one, and this is sufficient to completely ruin the least-squares estimation of the coefficients.

Lack of Robustness of the Least-squares Estimators

To compute the influence function of the least-squares estimator, we must first notice that it is an M-estimator. Indeed, given the choice of η, the regression problem can be written:

$$\forall i \in [\![1, n]\!], \mathbf{y}^i = \mathbf{X}^i\overline{\boldsymbol{\theta}}^T + \boldsymbol{\varepsilon}^i$$

where

$\mathbf{y}^i$ is the i-th observation

$\mathbf{X}^i$ is the i-th row of the design matrix $\mathbf{X}$

$\overline{\boldsymbol{\theta}}$ is the p-vector of true parameters

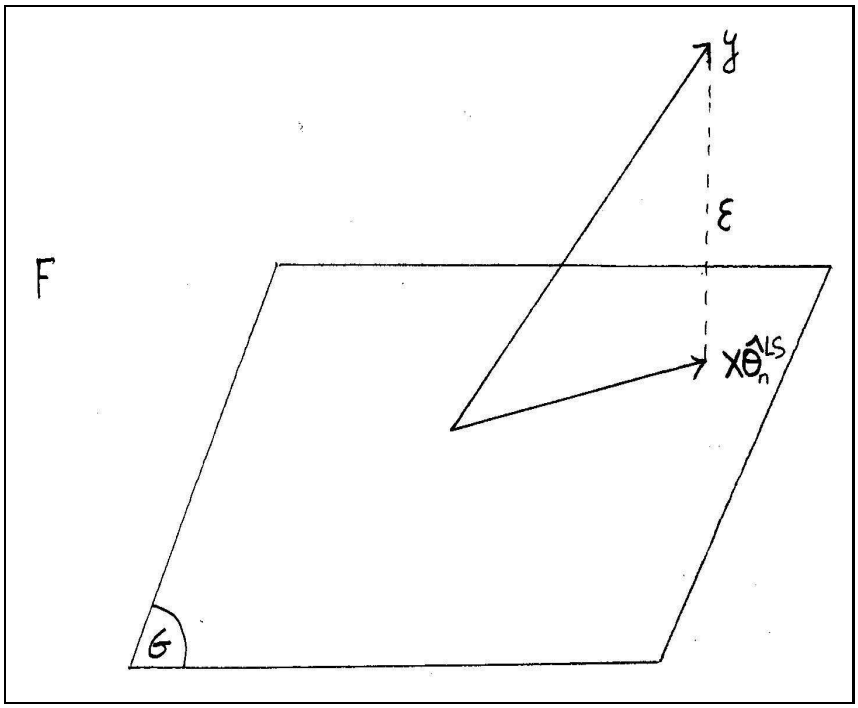

Figure 9.3: Least-squares is a projection

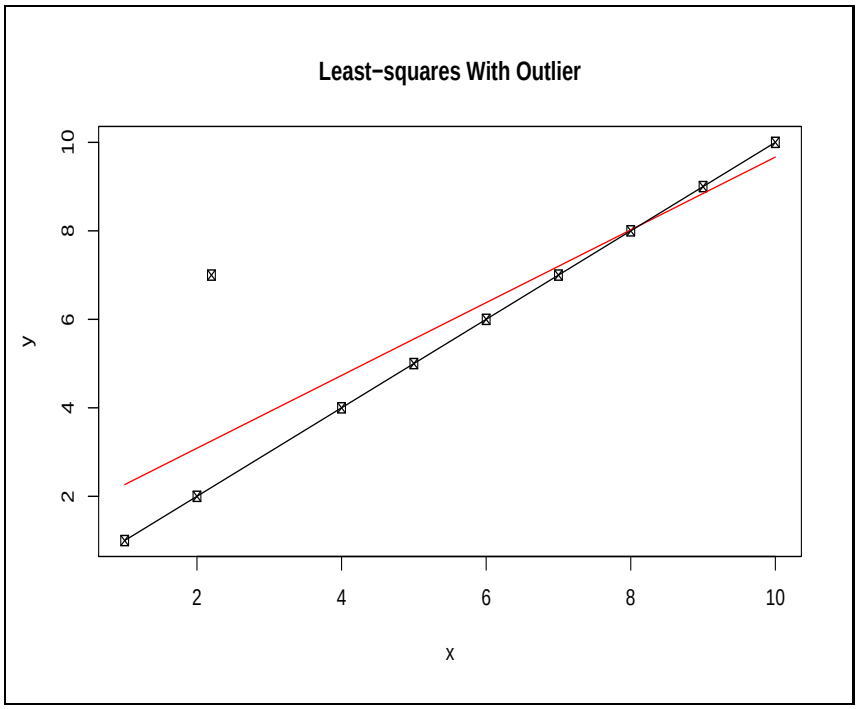

Figure 9.4: Least-squares is not robust

$\boldsymbol{\varepsilon}^i$ is the i-th error

If we define $\rho^{LS} : \left\{ \begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^+ \\ r & \mapsto & \frac{r^2}{2} \end{array} \right.$,

$$\widehat{\boldsymbol{\theta}}_n^{LS} = \arg\min_{\boldsymbol{\theta}\in\Theta} \sum_{i=1}^{n} \rho^{LS}(\mathbf{y}^i - \mathbf{X}^i\boldsymbol{\theta}^\intercal)$$

The division by 2, which obviously has no impact on minimization, is an artefact to ensure that $\psi^{LS}(r) := \frac{\partial \rho^{LS}(r)}{\partial r} = r$.

Let K_n be the empirical distribution of the *residuals* $\mathbf{r}^i(\boldsymbol{\theta}) := \mathbf{y}^i - \mathbf{X}^i\boldsymbol{\theta}^\intercal$: $K_n := \frac{1}{n}\sum_{i=1}^{n} \Delta_{\mathbf{r}^i}$, where $\Delta_{\mathbf{r}^i}$ is the Dirac mass in $\mathbf{r}^i$. Then, if, for all probability distributions F,

$$\widehat{\boldsymbol{\theta}}^{LS}(F) := \arg\min_{\boldsymbol{\theta}\in\Theta} \int \rho^{LS}(r)dF,$$

we have $\widehat{\boldsymbol{\theta}}_n^{LS} = \widehat{\boldsymbol{\theta}}^{LS}(K_n)$.

$\widehat{\boldsymbol{\theta}}^{LS}$ is an M-estimator of ρ-type for the function ρ^{LS} we have just defined.

We will now see that the main reason the Gauß-Markov least-squares estimate is not robust is because its influence function is unbound.

Theorem 5 *(Influence function of the least-squares estimator)*
Under the Gauß-Markov hypothesis, the influence function of the least-squares estimator is:

$$IF(\mathbf{x}, \mathbf{x}^\intercal\widehat{\boldsymbol{\theta}}^{LS} + r; \widehat{\boldsymbol{\theta}}^{LS}, F) = r\mathbb{E}[(\mathbf{x}\mathbf{x}^\intercal)^{-1}]\mathbf{x}$$

where:

$\mathbf{x} = (\mathsf{x}_1, \cdots, \mathsf{x}_p) \in F^p$,

$r \in \mathbb{R}$,

F is the joint distribution of the residuals $r(\boldsymbol{\theta}) := \mathsf{y} - \mathbf{x}^\intercal\boldsymbol{\theta}$ and of $\mathbf{x}$.

□

Proof:
The definition of an influence function is (see [4] page 230):

$$IF(\mathbf{x}, y; T, F) := M(\psi, F)^{-1}\psi(\mathbf{x}, y; T(F))$$

with

$\mathbf{x} \in F^p, y \in \mathbb{R}, r \in \mathbb{R}$,

$T = \widehat{\boldsymbol{\theta}}^{LS}$,

$\psi(\mathbf{x}, r) = \psi^{LS}(r)\mathbf{x} = r\mathbf{x}$,

$M(\psi, F) := -\int \left[\frac{\partial\psi(\boldsymbol{x},\boldsymbol{\theta})}{\partial\theta}\right]_{\hat{\boldsymbol{\theta}}^{LS}(F)} dF(\mathbf{x}, r)$

We find:

$$\begin{aligned} \frac{\partial\psi}{\partial\boldsymbol{\theta}} &= \frac{\partial\psi(\mathbf{x}, r)}{\partial r}\frac{\partial r(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \\ &= \mathbf{x}(-\mathbf{x}^{\intercal}) \\ M &= \mathbb{E}[\mathbf{x}\mathbf{x}^{\intercal}] \end{aligned}$$

hence

$$IF(\mathbf{x}, \mathbf{x}^{\intercal}\widehat{\boldsymbol{\theta}}^{LS} + r; \widehat{\boldsymbol{\theta}}^{LS}, F) = r\mathbb{E}[(\mathbf{x}\mathbf{x}^{\intercal})]^{-1}\mathbf{x}$$

♦

The influence function being proportional to the identity function, it is clear that the bigger an outlier, the bigger its influence on the Gauß-Markov estimator.

9.3.4 Biweight Regression

As before,

$$\mathsf{y} = \eta(\mathbf{x}; \boldsymbol{\theta}) + \varepsilon$$

and

$$r(\boldsymbol{\theta}) := \mathsf{y} - \mathbf{x}^{\intercal}\boldsymbol{\theta}$$

We are computing an estimator $\widehat{\boldsymbol{\theta}}$ satisfying the p-vector equation:

$$\int \psi(r(\hat{\theta}))dF = \mathbf{0}$$

where F is the distribution of the residuals.
We choose:

$$\eta : \left\{ \begin{array}{ccc} \Gamma_1 \times \cdots \times \Gamma_p \times \Theta & \rightarrow & \mathbb{R} \\ (\mathsf{x}_1, \cdots, \mathsf{x}_p; \boldsymbol{\theta}) & \mapsto & \eta(\mathsf{x}_1, \cdots, \mathsf{x}_p, \boldsymbol{\theta}) \end{array} \right.$$

As the influence function of an M-estimator of ψ-type is proportional to its ψ function, we will choose ψ carefully so as to avoid the drawbacks of the least-squares estimator.

A possible solution is to choose Tukey's *biweight function* defined as follows:

$$\psi_{bi}(x) := x(c^2 - x^2)^2 \mathbb{1}_{[-c,c]}(x), c > 0$$

A plot of this function can be seen in Figure 9.5 page 126. The advantage of this function is that it will give the estimator a small gross-error sensitivity, a small local-shift sensitivity and a finite rejection point.

Practical Computation

One of the prices to pay for a robust estimator is a higher computation cost and a more complex algorithm.

The idea behind this algorithm is to see biweight regression as a special case of iteratively reweighted least-squares. Weighted least-squares regression is just a generalization of ordinary least squares: instead of projecting using the scalar product $\forall(\mathbf{u}, \mathbf{v}) \in (F^n)^2, \langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T\mathbf{v}$, we use a metric matrix $\mathbf{W}$ (i.e. positive, symmetric, definite), which is usually just a diagonal matrix: $\mathbf{W} = \text{diag}(w_1, \cdots, w_n)$ and we calculate $\mathbf{u}^T\mathbf{W}\mathbf{v}$. Hence instead of minimizing

$$\sum_{i=1}^{n} (\mathbf{y}^i - \mathbf{x}^{iT}\boldsymbol{\theta})^2$$

we choose $\widehat{\boldsymbol{\theta}}_n^{WLS}$ so that

$$\widehat{\boldsymbol{\theta}}_n^{WLS} = \arg\min_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^{n} w_i (\mathbf{y}^i - \mathbf{x}^{iT}\boldsymbol{\theta})^2$$

Using the same reasoning as for the least-squares estimator, we find:

$$\widehat{\boldsymbol{\theta}}_n^{WLS} = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{y}$$

Recursively reweighted least squares is an iterative algorithm where the weight matrix at step k, $\mathbf{W}^{(k)}$, is defined using the residuals from the previous step.

We define the *weight function*:

$$w(r) := \frac{\psi(r)}{r}$$

$\widehat{\boldsymbol{\theta}}_n^{bi}$ is then a solution of the p-vector equation:

$$\sum_{i=1}^{n} w\left(\mathbf{r}^i(\widehat{\boldsymbol{\theta}}_n^{bi})\right) \mathbf{r}^i(\widehat{\boldsymbol{\theta}}_n^{bi}) \frac{\partial \mathbf{r}^i}{\partial \theta}(\widehat{\boldsymbol{\theta}}_n^{bi}) = \mathbf{0}$$

which is the same as solving

$$\sum_{i=1}^{n} w\left(\mathbf{r}^i(\widehat{\boldsymbol{\theta}}_n^{bi})\right) \mathbf{r}^i(\boldsymbol{\theta}) \frac{\partial \mathbf{r}^i}{\partial \theta}(\boldsymbol{\theta}) = \mathbf{0}$$

This is the gradient, with respect to $\boldsymbol{\theta}$, of

$$\sum_{i=1}^{n} w(\mathbf{r}^i(\widehat{\boldsymbol{\theta}}_n^{bi}))\mathbf{r}^i(\boldsymbol{\theta})^2 = \mathbf{0}$$

which is a weighted least-squares problem. As we do not know $w(\mathbf{r}^i(\widehat{\boldsymbol{\theta}}_n^{bi}))$, we solve this equation iteratively: we start with weights at 1 and compute $\hat{\boldsymbol{\theta}}$. Based on the residuals, we compute new weights, and so on.

We would like to give weight 0 to the observations which, given the model assumptions, have a probability of $\alpha \in [0, 1]$ or less of occuring. We know that the residuals are normally distributed:

$$\mathbf{r} := \mathbf{y} - \eta(\mathbf{X}; \boldsymbol{\theta}) \sim \mathcal{N}\big(0,\ \sigma^2\mathbf{I}_n\big)$$

$\forall i \in [\![1, n]\!], P\left(\left|\frac{\mathbf{r}^i}{\sigma^2}\right| > q(p)\right) = 2(1-p)$, where q is the quantile function of $\mathcal{N}(0,\ 1)$. Therefore, $\forall i \in [\![1, n]\!], P\left(\left|\frac{\mathbf{r}^i}{\sigma^2 q(1-\frac{\alpha}{2})}\right| > 1\right) = \alpha$ and the algorithm then runs as follows:

Algorithm 4 Biweight regression algorithm

1. Choose a value $c = q(1-\frac{\alpha}{2})$ by which the residuals will be divided (to select the cut-off point after which the residuals will be considered large enough to identify the corresponding observation as an outlier).
2. Choose a value $\varepsilon > 0$ for the precision and a maximum number of iterations.
3. Set $k = 1$ and initialize the weights by $\mathbf{W}^{(1)} = \mathbf{I}_n$.
4. Compute $\boldsymbol{\theta}_n^{(k)} = (\mathbf{X}^\intercal \mathbf{W}^{(k)} \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{W}^{(k)} \boldsymbol{y}$.
5. Calculate the residuals:
$$\boldsymbol{r}^{(k)} := \boldsymbol{y} - \mathbf{X}\boldsymbol{\theta}_n^{(k)}$$
6. If the maximum number of iterations has been reached or $\|\boldsymbol{r}\| \leq \varepsilon$, stop (returning "fail" if $\|\boldsymbol{r}\| > \varepsilon$ and "success" otherwise).
7. Calculate $S_k := \text{median}\left\{\left|\boldsymbol{r}_1^{(k)}\right|, \cdots, \left|\boldsymbol{r}_n^{(k)}\right|\right\}$ for $i \in [\![1, n]\!]$.
8. Compute the new weights: $\mathbf{W} := \text{diag}(w(\frac{\boldsymbol{r}_1}{cS_k}), \cdots, w(\frac{\boldsymbol{r}_n}{cS_k}))$, where
$$w(r) := \frac{\psi^{bi}(r)}{r}$$
9. Increase k by 1 and start again from step 4.

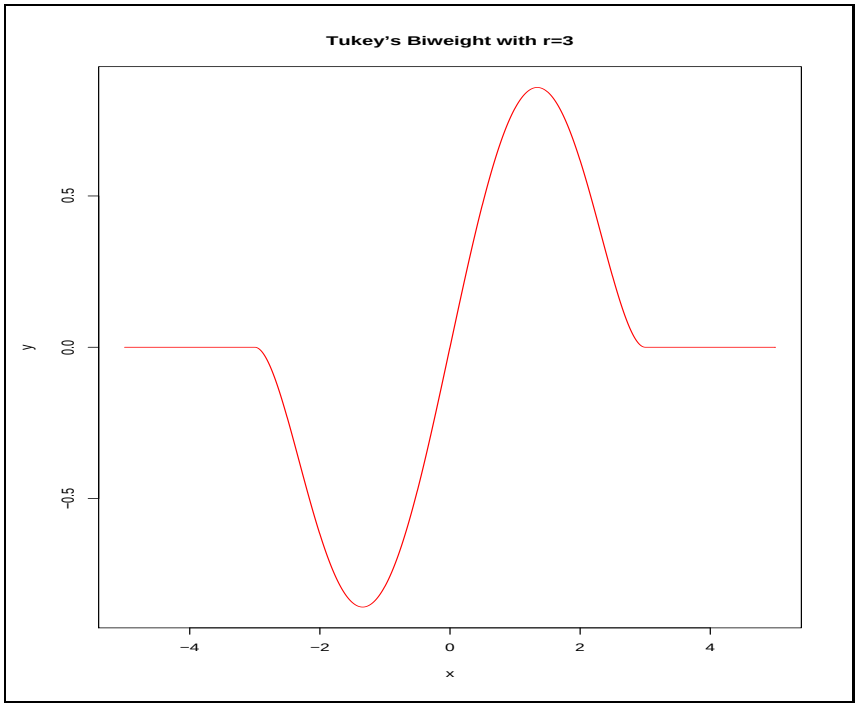

Figure 9.5: Tukey's biweight

Chapter 10

Appendix B: Window Smoothing

10.1 Window Smoothing of Time Series

To evaluate the importance of each frequency in the repeated median filter, we need to get an estimation of the spectrum. We will cover some of the ways to do this in this section. We will first give two estimators of the spectrum and motivate the use of smoothing by examining the variance of these estimators. A more complete coverage can be found in the book by Donald B. Percival and Andrew T. Walden ([11]), and in that of M. B. Priestley ([12]).

10.1.1 Motivation

Graphical Illustration of Smoothing

To have a better idea of what we will be doing, we will give an example of smoothing.

Suppose we have a time series which looks like Figure 10.1 page 128.

We can then build its spectrum. On Figure 10.2 page 128, we have plotted the spectrum itself and the smoothed version.

In the following we will give a more rigorous motivation for smoothing time series.

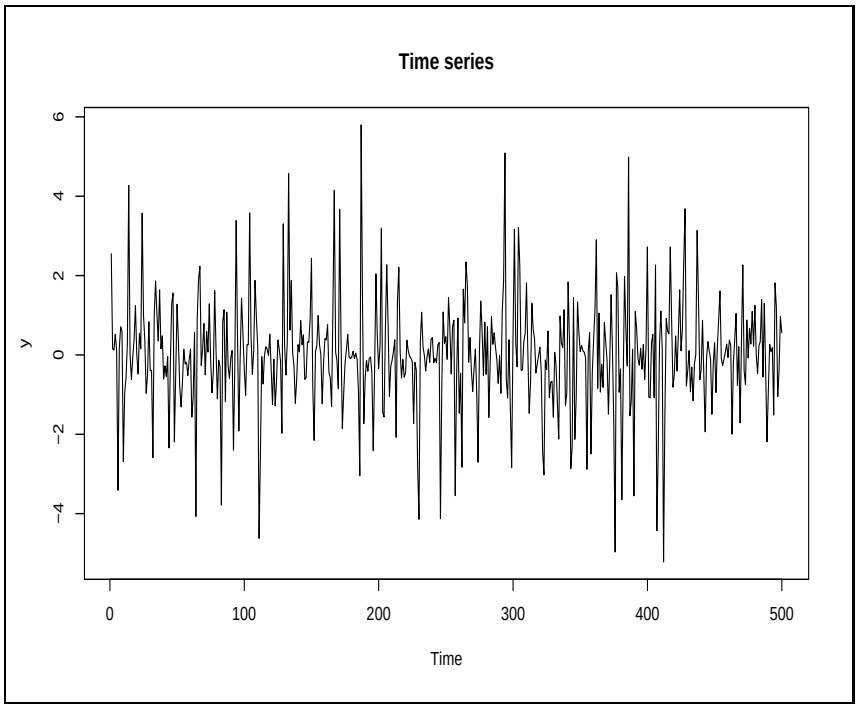

Figure 10.1: Time Series

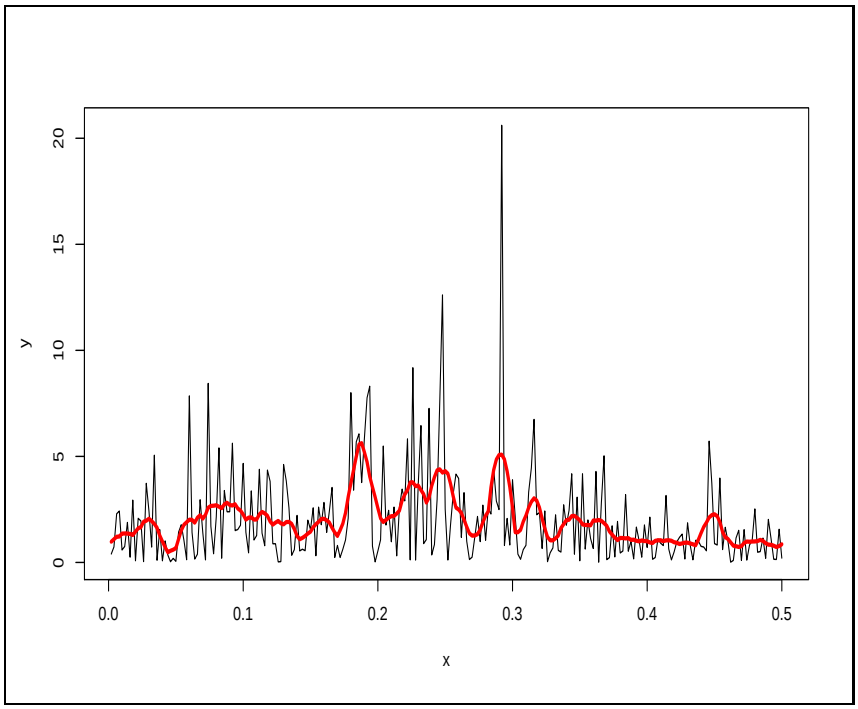

Figure 10.2: Spectrum and Smoothed Spectrum

A Classical Estimator

First, let us review a definition of the power spectrum.

Definition 26 *(Power spectrum)*

Let $(\mathsf{X}_t)_{t\in\mathbb{N}}$ be a time series. The *power spectrum* Φ of $(\mathsf{X}_t)_{t\in\mathbb{N}}$ can be defined by:

$$\Phi : \left\{ \begin{array}{ccc} [-\pi,\pi] & \to & \mathbb{R}^+ \\ \omega & \mapsto & \Phi(\omega) \end{array} \right.$$

with

$$\Phi(\omega) := \lim_{n\to+\infty} \mathbb{E}\left[\frac{1}{n}\left|\sum_{k=1}^{n} \mathsf{X}_k e^{-\mathtt{i}\omega k}\right|^2\right]$$

♢

In all practical cases, we only have access to a few realizations of the time series at certain times. From this incomplete information, we would like to get an estimate of the power spectrum. This can be done the following way:

Definition 27 *(Natural estimation of the power spectrum)*

Let $(\mathsf{X}_t)_{t\in\mathbb{N}}$ be a time series, $n \in \mathbb{N}^*$ and $(x_1,\cdots,x_n)$ a realization of the random variables $(\mathsf{X}_1,\cdots,\mathsf{X}_n)$. We call *natural estimate of the power spectrum* the function

$$\hat{\Phi} : \left\{ \begin{array}{ccc} [-\pi,\pi] & \to & \mathbb{R}^+ \\ \omega & \mapsto & \hat{\Phi}(\omega) \end{array} \right.$$

with

$$\hat{\Phi}(\omega) := \frac{1}{n}\left|\sum_{k=1}^{n} x_k e^{-\mathtt{i}\omega k}\right|^2$$

♢

Another formulation is equivalent to that one and uses the auto-covariance function.

Definition 28 *(Autocovariance function)*

Let $I \subseteq \mathbb{R}$ and $(\mathsf{X}_t)_{t\in I}$ be a stationary Gaussian process. Then the function

$$\Gamma : \left\{ \begin{array}{ccc} I\times I & \to & \mathbb{R} \\ (s,t) & \mapsto & \mathbb{E}\left[(\mathsf{X}_s - \mathbb{E}\mathsf{X}_s)(\mathsf{X}_t - \mathbb{E}\mathsf{X}_t)\right] \end{array} \right.$$

is shift-invariant, i.e. $\forall (s,t) \in I^2$, if $(0, t-s) \in I^2$, $\Gamma(s,t) = \Gamma(0,t-s)$. We call the function

$$\gamma : \left\{ \begin{array}{ccc} \mathbb{R} & \to & \mathbb{R} \\ u & \mapsto & \gamma(u) := \Gamma(0,u) \end{array} \right.$$

autocovariance function of the process $(\mathsf{X}_t)_{t\in I}$.

♢

Definition 29 *(Estimate of power spectrum with autocovariance function)*
Let $(\mathsf{X}_t)_{t\in\mathbb{N}}$ be a time series, $n \in \mathbb{N}^*$ and $(x_1, \cdots, x_n)$ a realization of the random variables $(\mathsf{X}_1, \cdots, \mathsf{X}_n)$. Let γ be the auto-covariance function of the process $(\mathsf{X}_t)_{t\in\mathbb{N}}$. Another estimate of the power spectrum is the function

$$\hat{s} : \left\{ \begin{array}{ccc} [-\pi, \pi] & \to & \mathbb{R}^+ \\ \omega & \mapsto & \hat{s}(\omega) \end{array} \right.$$

with

$$\hat{s}(\omega) := \frac{1}{2\pi} \sum_{j=-n+1}^{n-1} \gamma(j) e^{-\mathbf{i}\omega j}$$

♢

Drawback of the Classical Estimator

Let $(\mathsf{X}_i)_{n\in\mathbb{N}}$ be a stationary Gaussian process and $\forall n \in \mathbb{N}^*, \forall \omega \in \mathbb{R}, \mathsf{X}_n(\omega) := \frac{1}{\sqrt{n}} \sum_{k=1}^{n} \mathsf{X}_k e^{-\mathbf{i}\omega k}$. Let $\hat{\Phi}$ be the natural spectrum estimator of the process. If $\forall k \in [\![0, n]\!], \mathsf{X}_k \sim \mathcal{N}(\mu, \sigma^2(\omega_k))$, then Re $\mathsf{X}_n(\omega) \sim \mathcal{N}(0, \sigma^2(\omega))$ and Im $\mathsf{X}_n(\omega) \sim \mathcal{N}(0, \sigma^2(\omega))$ with $\sigma^2(\omega) := \frac{1}{n} \sum_{k=1}^{n} \sigma_k^2 \cos^2(\omega k)$. Therefore, Re $\frac{\mathsf{X}_n(\omega)}{\sigma(\omega)} \sim \mathcal{N}(0, 1)$ and Im $\frac{\mathsf{X}_n(\omega)}{\sigma(\omega)} \sim \mathcal{N}(0, 1)$. Hence, $\frac{\hat{\Phi}(\omega)}{\sigma^2(\omega)} \sim \chi_2^2$. But we also have: $\mathbb{E}\Phi(\omega) = 2\sigma^2(\omega)$, therefore

$$\frac{2\hat{\Phi}(\omega)}{\Phi(\omega)} \sim \chi_2^2$$

This means two things: first, that the natural estimator is asymptotically unbiased, and second, that its variance does not depend on n. As the quartiles of a χ_2^2 are quite large , the variance of the estimator is big, no matter how large n is, which means that the estimator is not consistent. Günther (in [3]) shows similar results for the second estimator we reviewed.

The question is: do we have to resign to an inconsistent estimator of the spectrum? The answer is no: if we obtain the periodogram at a given frequency by averaging the neighboring frequencies, we may reduce the variance. What we do, is that we calculate an estimation of the spectrum and give ourselves a set of weights. Then at each frequency, we build the weighted averaged of the neighboring points to end up with a smoothed version of the spectrum.

10.1.2 Smoothing Methods

Smoothing can be done two equivalent ways: either one smoothes in the frequency domain (as suggested in the previous paragraph) or one smooths in the time domain.

Smoothing in the Frequency Domain

Let $(\mathsf{X}_t)_{t\in\mathbb{N}}$ be a time series, $n \in \mathbb{N}^*$ and $(x_1, \cdots, x_n)$ a realization of the random variables $(\mathsf{X}_1, \cdots, \mathsf{X}_n)$. Let $\hat{\Phi}$ be the natural spectrum estimator.

$$\forall k \in [\![1, n]\!], \omega_k := \frac{2k\pi}{n} - \pi$$

The following function aims at rendering the spectrum "circular":

$$\forall (j,k) \in ([\![1,n]\!])^2, \sigma(j-k) := \begin{cases} j-k & \text{if } 1 \leq j-k \leq n \\ n-(j-k) & \text{otherwise} \end{cases}$$

Let $M \in \mathbb{N}$ such as $M \leq n$ (called *truncation point*) and $(W_n(k))_{-M\leq k\leq M}$ a set of weights satisfying:

1. For $-M \leq k \leq M, W_n(k) = W_n(-k)$,
2. $\sum_{k=-M}^{M} W_n(k) = 1$

The smoothed version of the estimator is:

$$\forall j \in [\![1, n]\!], \hat{\Phi}_S(\omega_j) := \sum_{k=-M}^{M} W_n(k)\hat{\Phi}(\omega_{\sigma(j-k)} - \pi)$$

Smoothing in the Time Domain

Another approach is to smooth the time series in the time domain. As the Fourier transform is bijective, we know that this will be exactly equivalent to the smoothing in the frequency domain.

Let $(\mathsf{X}_t)_{t\in\mathbb{N}}$ be a time series, $n \in \mathbb{N}^*$ and $(x_1, \cdots, x_n)$ a realization of the random variables $(\mathsf{X}_1, \cdots, \mathsf{X}_n)$. Let γ be the autocovariance function of the time series.

Let $M \in \mathbb{N}$ such as $M \leq n$ and $(\lambda_n(k))_{-M\leq k\leq M}$ be a *lag window* satisfying:

1. For $-M \leq k \leq M, \lambda_n(k) = \lambda_n(-k)$,
2. $\lambda_n(0) = 1$,
3. For $-M \leq k \leq M, |\lambda_n(k)| \leq 1$

The smoothed version of the estimator is:

$$\forall \omega \in [-\pi, \pi], \hat{f}_L(\omega) := \frac{1}{2\pi} \sum_{k=-M}^{M} \lambda_n(k)\gamma(k)e^{-\mathtt{i}k\omega}$$

Link Between Time and Frequency Domain

While they appear to be quite different, the two approaches are in fact related. If we define a spectral window with a continuous weight function $W(\omega)$ as the Fourier transform of the lag window $\lambda(k)$, it is easy to show that

$$W(\theta) = \frac{1}{2\pi} \sum_{n=-M}^{M} \lambda\left(\frac{n}{M}\right) e^{-\mathtt{i}n\theta} \text{ and } \frac{-\pi}{M} \leq \theta \leq \frac{\pi}{M}$$

and

$$\lambda\left(\frac{k}{M}\right) = \int_{-\pi}^{\pi} W(\theta) e^{\mathtt{i}k\theta} d\theta$$

10.1.3 Various Windows

Here is a non-exhaustive review of the most important window shapes

$$\theta_k := \frac{k\pi}{M^2} \text{ and } x_k := k/M, \text{ with } -M \leq k \leq MM$$

Rectangular Window

- Definition in the frequency domain

$$W(\theta_k) = \frac{1}{2\pi}\left(\sum_{n=-M}^{M} \cos(n\theta_k)\right) = D_M(\theta_k)$$

- Definition in the time domain

$$\lambda(x) = \begin{cases} 1, |x_k| \leq 1 \\ 0, |x_k| > 1 \end{cases}$$

- Plot on page 135.

Bartlett Window

- Definition in the frequency domain

$$W(\theta_k) = \frac{1}{2M\pi}\left(\frac{\sin\left(\frac{M\theta_k}{2}\right)}{\sin\left(\frac{\theta_k}{2}\right)}\right)^2 = F_M(\theta_k)$$

- Definition in the time domain

$$\lambda(x_k) = \begin{cases} 1 - |x_k|, |x_k| \leq 1 \\ 0, |x_k| > 1 \end{cases}$$

- Plot on page 136.

Parzen Window

- Definition in the frequency domain

$$W(\theta_k) = \frac{6\pi}{M}\left(F_{M/2}(\theta_k)\right)^2\left[1 - \frac{2}{3}\sin^2\left(\frac{\theta_k}{2}\right)\right]$$

- Definition in the time domain

$$\lambda(x_k) = \begin{cases} 1 - 6x_k^2 + 6\left|x_k\right|^3, |x_k| \leq 1/2 \\ 2(1 - |x_k|)^3, 1/2 < |x_k| \leq 1 \\ 0, |x_k| > 1 \end{cases}$$

- Plot on page 137.

Tukey-Hamming Window

- Definition in the frequency domain

$$W(\theta_k) = 0.23D_M\left(\theta_k - \frac{\pi}{M}\right) + 0.54D_M(\theta_k) + 0.23D_M\left(\theta_k + \frac{\pi}{M}\right)$$

- Definition in the time domain

$$\lambda(x_k) = \begin{cases} 1 - 0.46 + 0.46\cos(\pi x_k), |x_k| \leq 1 \\ 0, |x_k| > 1 \end{cases}$$

- Plot on page 138.

Tukey Hanning Window

- Definition in the frequency domain

$$W(\theta_k) = W(\theta_k) = 0.25D_M\left(\theta_k - \frac{\pi}{M}\right) + 0.5D_M(\theta_k) + 0.25D_M\left(\theta_k + \frac{\pi}{M}\right)$$

- Definition in the time domain

$$\lambda(x_k) = \begin{cases} 1 - 0.5 + 0.5\cos(\pi x_k), |x_k| \leq 1 \\ 0, |x_k| > 1 \end{cases}$$

- Plot on page 139.

Daniell Window

- Definition in the frequency domain

$$W(\theta_k) = \frac{1}{2M+1}$$

- Definition in the time domain

$$\lambda(x_k) = \frac{\sin(\pi x_k)}{\pi x_k}$$

- Plot on page 140.

Bartlett-Priestley Window

- Definition in the frequency domain

$$W(\theta_k) = \frac{3M}{4\pi}\left[1 - \left(\frac{M\theta_k}{\pi}\right)^2\right]$$

- Definition in the time domain

$$\lambda(x_k) = \frac{3}{(\pi x_k)^2}\left(\frac{\sin(\pi x_k)}{\pi x_k} - \cos(\pi x_k)\right)$$

- Plot on page 141.

This last window is the one used by Tatum and Hurvich in their paper.

Figure 10.17 page 142 features all the windows scaled so that the sum of the weights is equal to one.

We have not found that the type of window chosen has a measurable influence on the output of the filter. As the spectrum is only use to determine the order in which to treat the frequencies, the main thing is to smooth it, but how has little or no importance. For the biweight filter cleaner however, the use of the Bartlett-Priestley window is necessary to have a positive-definite matrix $\mathbf{C}$. Therefore, it makes sense to use the same window for all the algorithms.

After having chosen a window shape, we have to evaluate the window length. To do this, we will need a measure of the goodness of a smoother. Tatum and Hurvich propose one way to do this and call it "AIC_C": it is the object of the next paragraph.

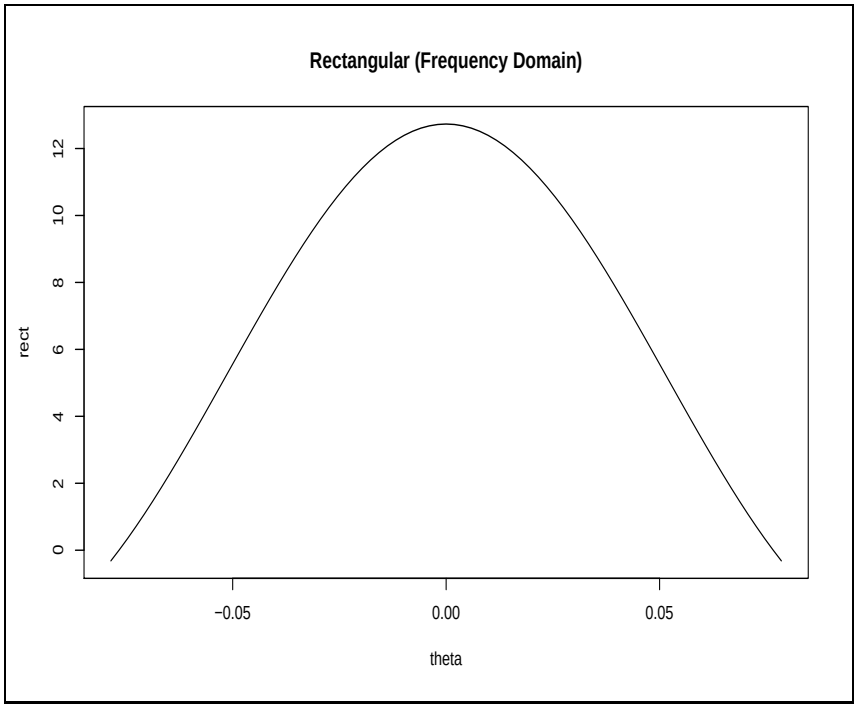

Figure 10.3: Rectangular Window (Frequency Domain)

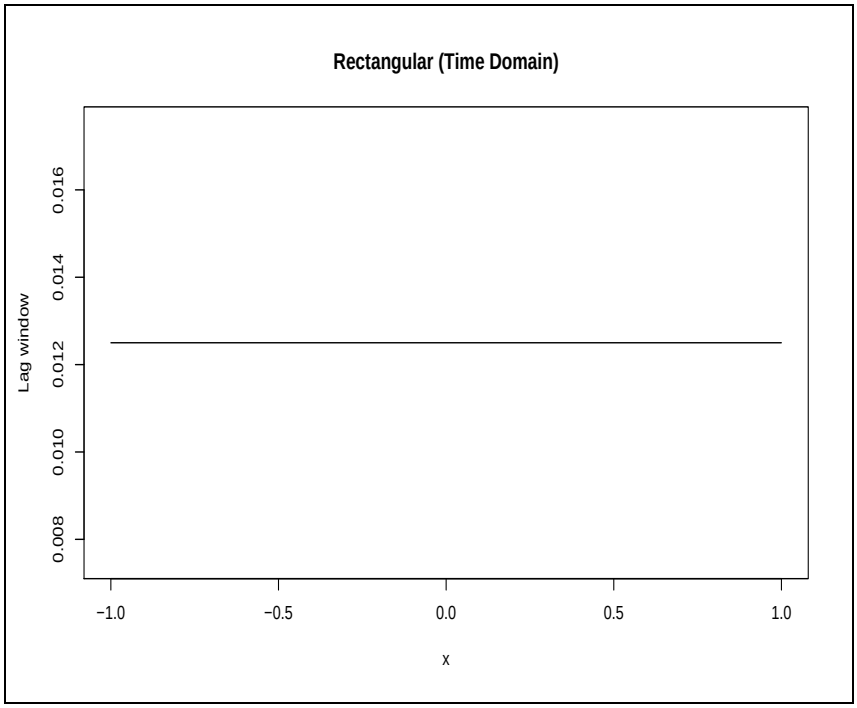

Figure 10.4: Rectangular Window (Time Domain)

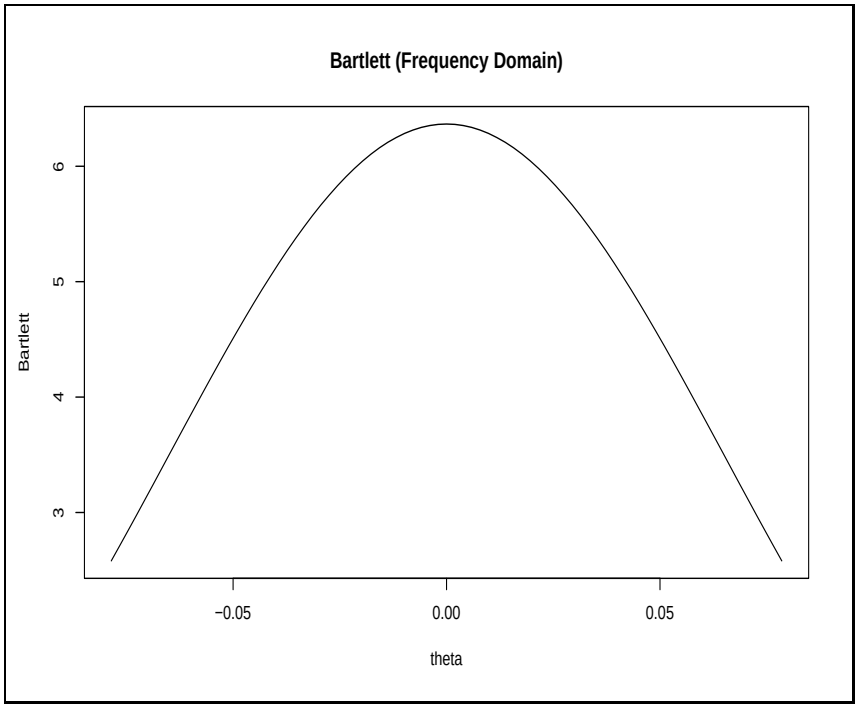

Figure 10.5: Bartlett Window (Frequency Domain)

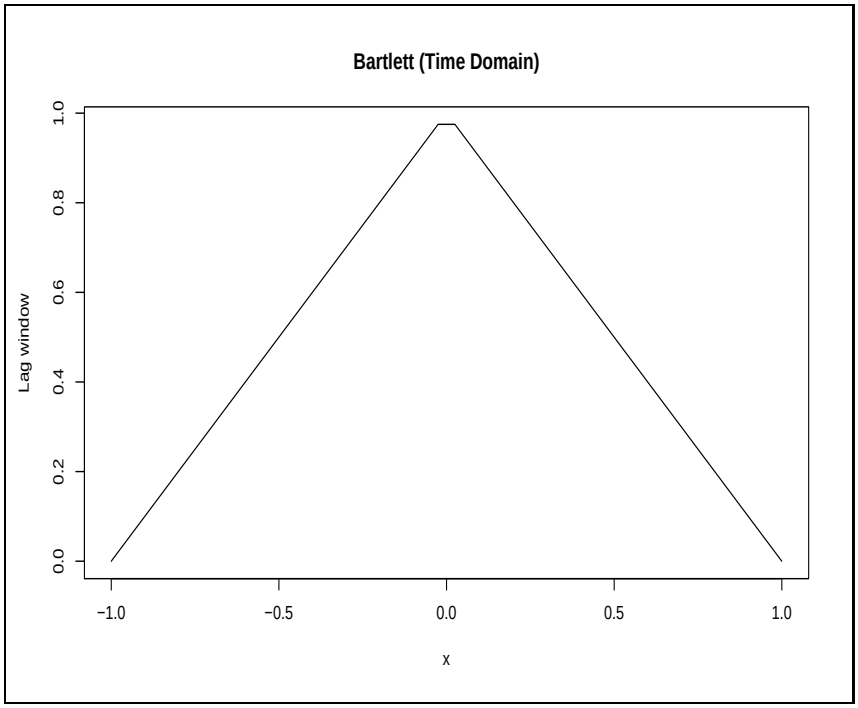

Figure 10.6: Bartlett Window (Time Domain)

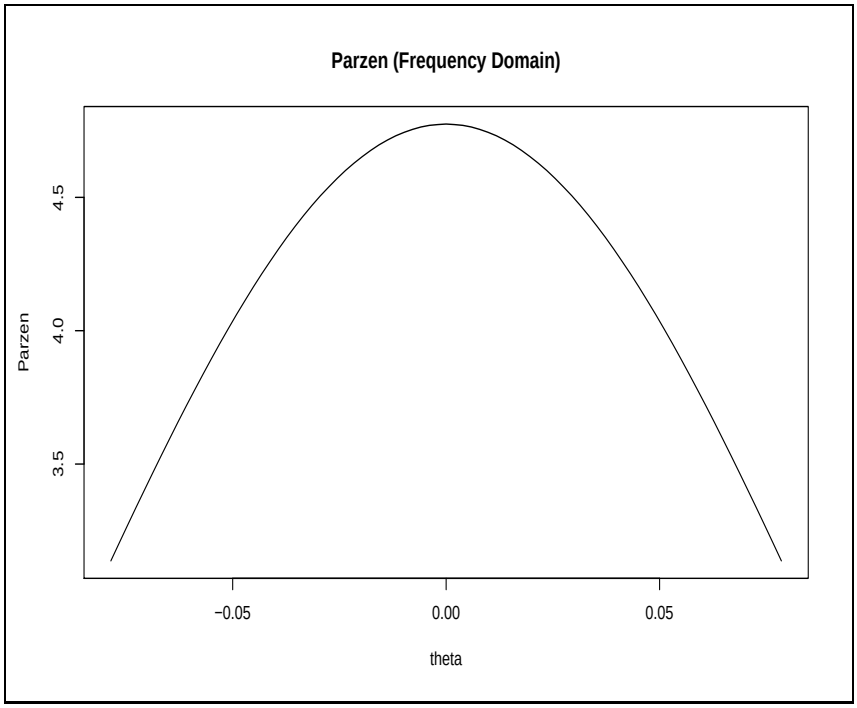

Figure 10.7: Parzen Window (Frequency Domain)

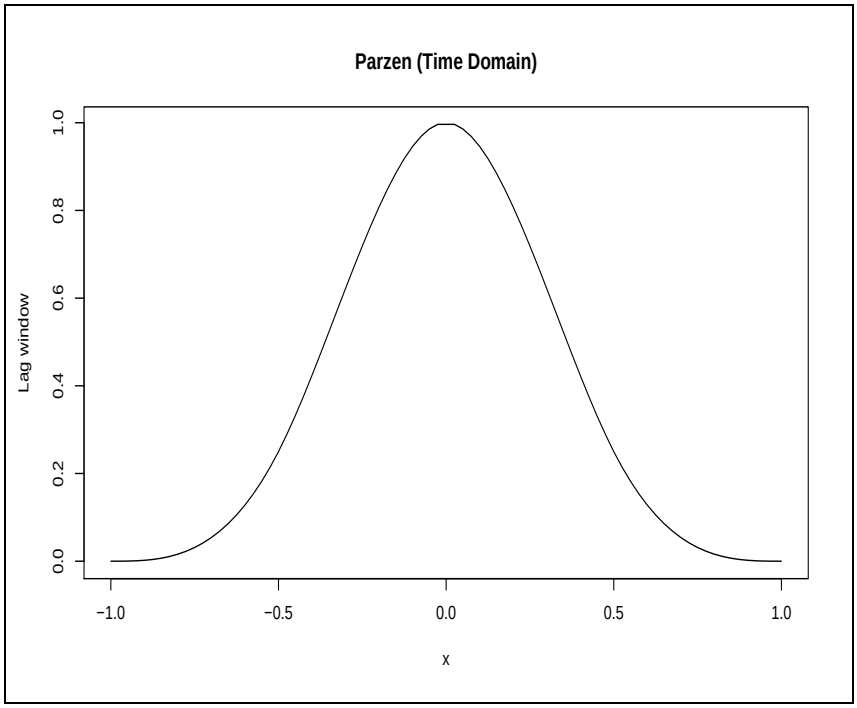

Figure 10.8: Parzen Window (Time Domain)

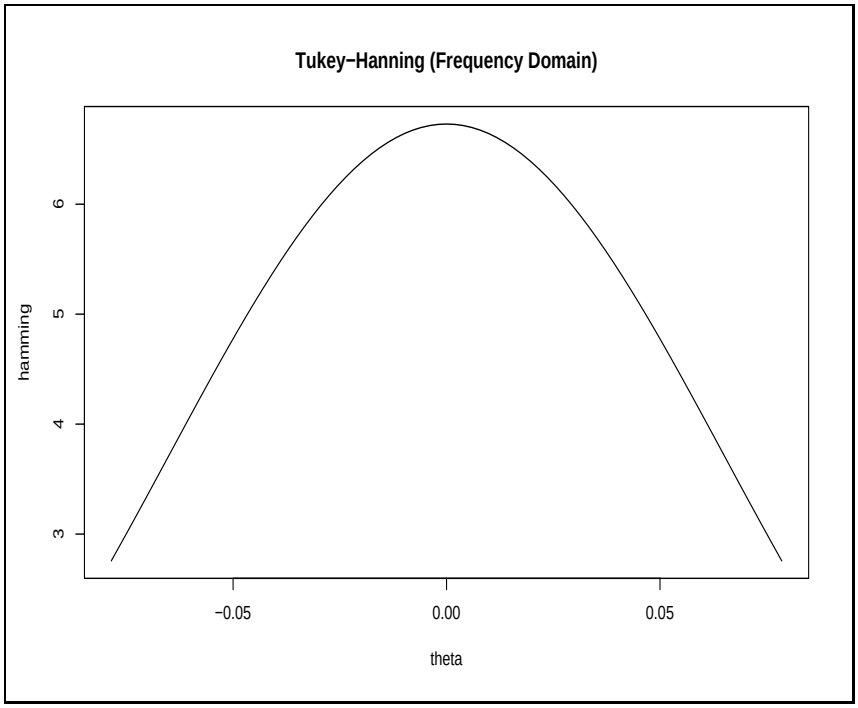

Figure 10.9: Tukey-Hamming Window (Frequency Domain)

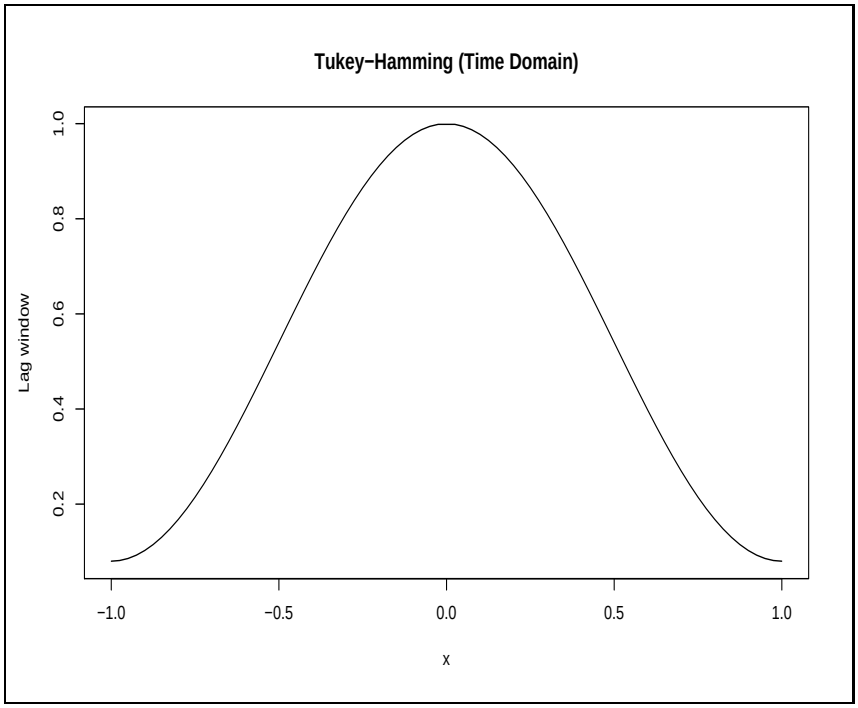

Figure 10.10: Tukey-Hamming Window (Time Domain)

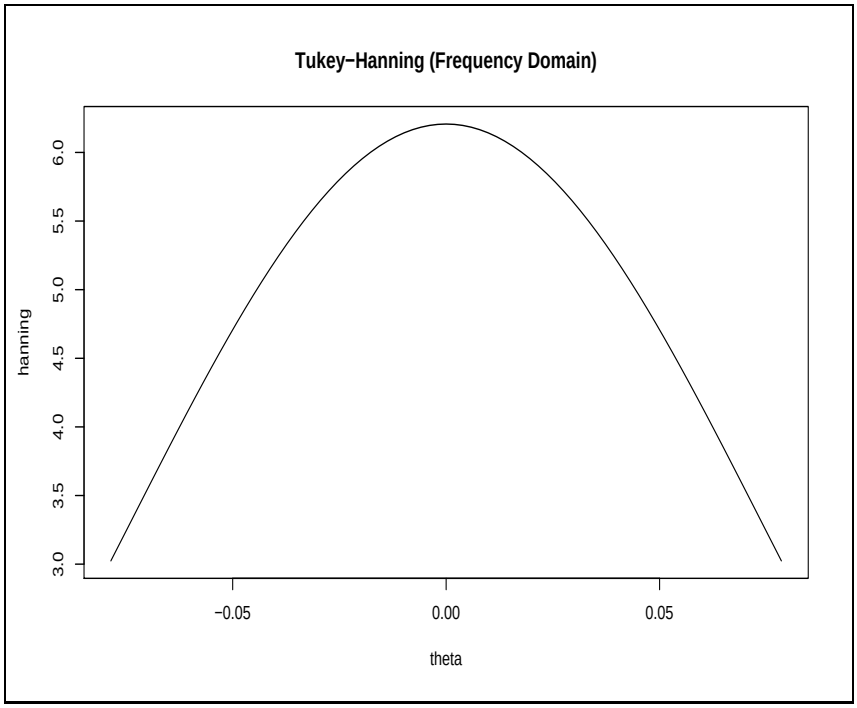

Figure 10.11: Tukey-Hanning Window (Frequency Domain)

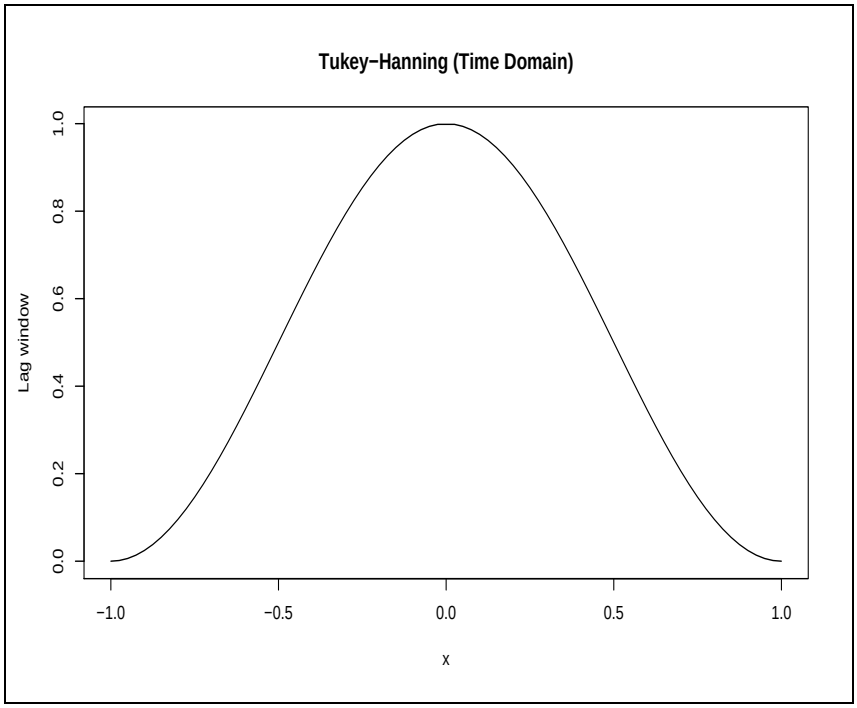

Figure 10.12: Tukey-Hanning Window (Time Domain)

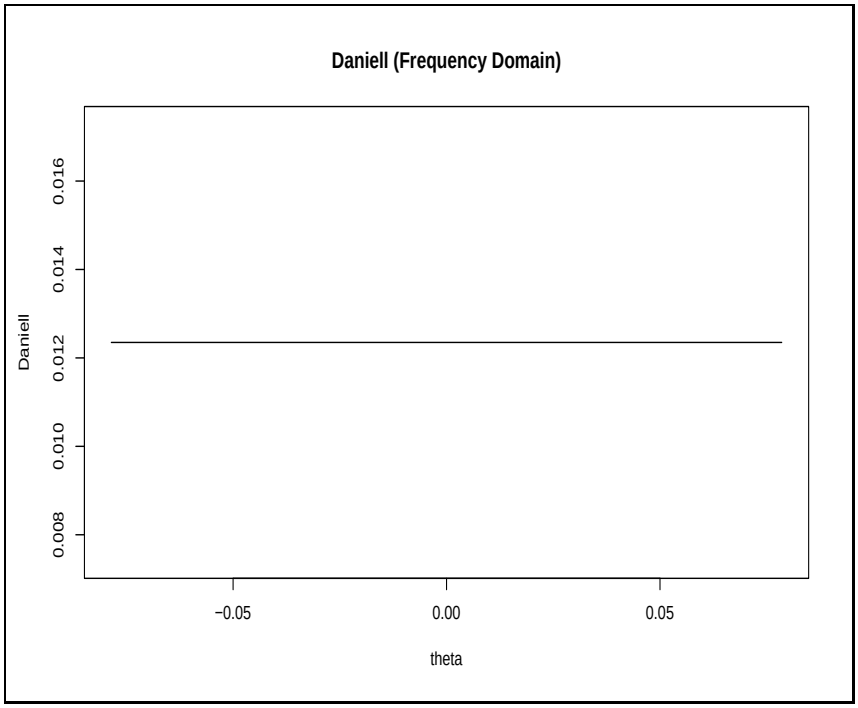

Figure 10.13: Daniell Window (Frequency Domain)

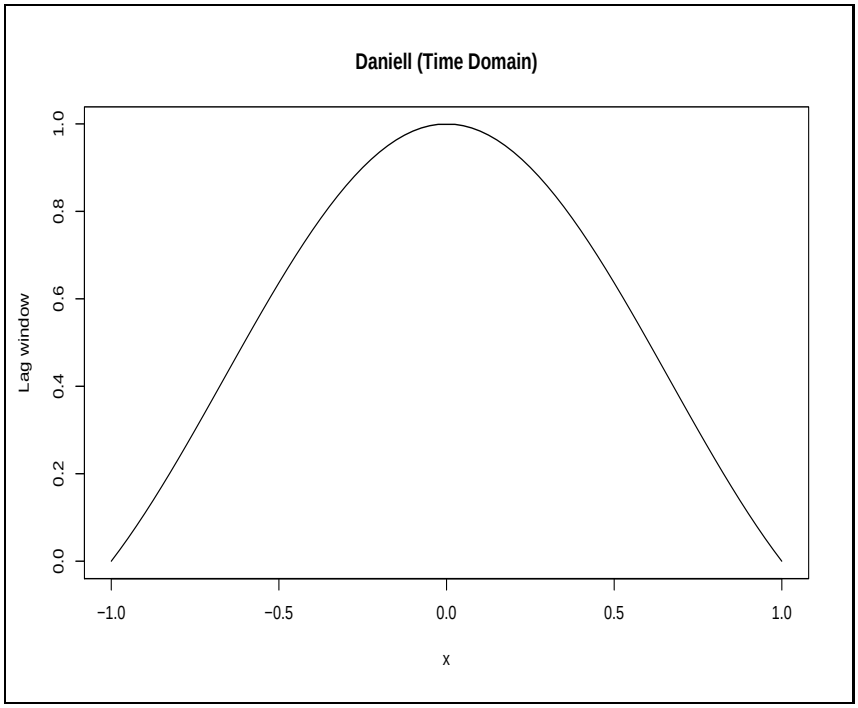

Figure 10.14: Daniell Window (Time Domain)

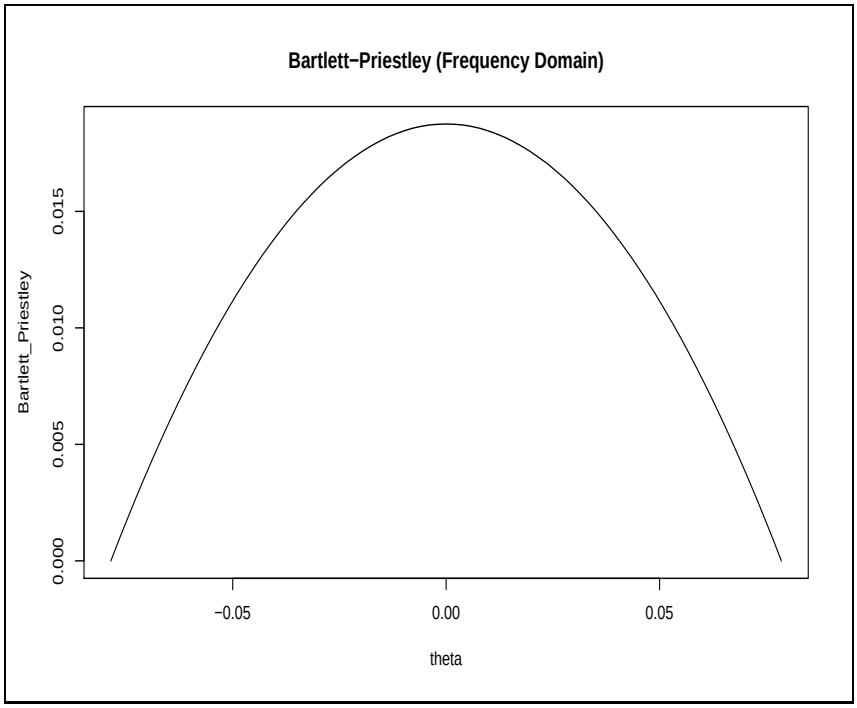

Figure 10.15: Bartlett-Priestley Window (Frequency Domain)

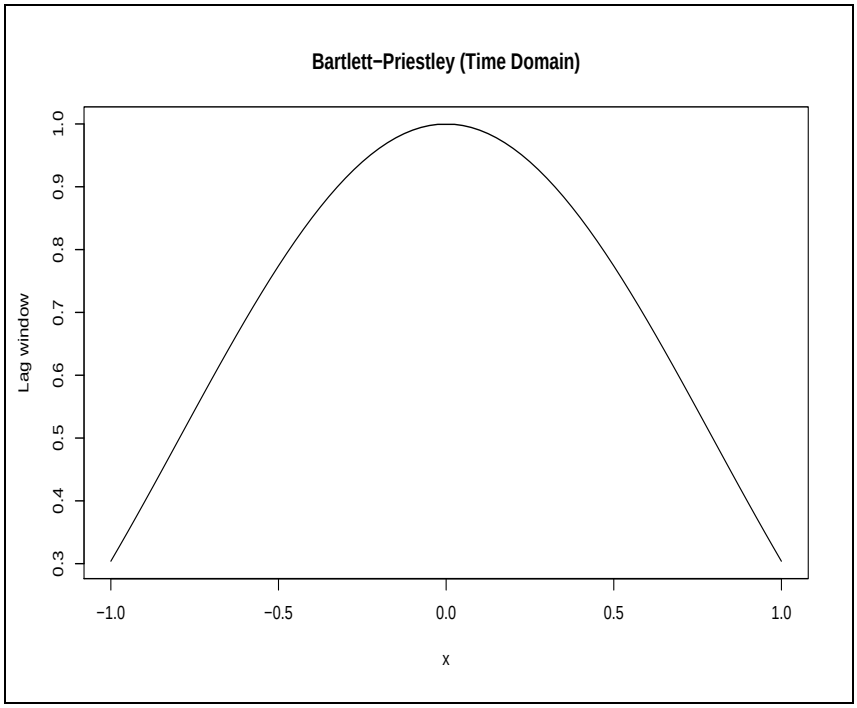

Figure 10.16: Bartlett-Priestley Window (Time Domain)

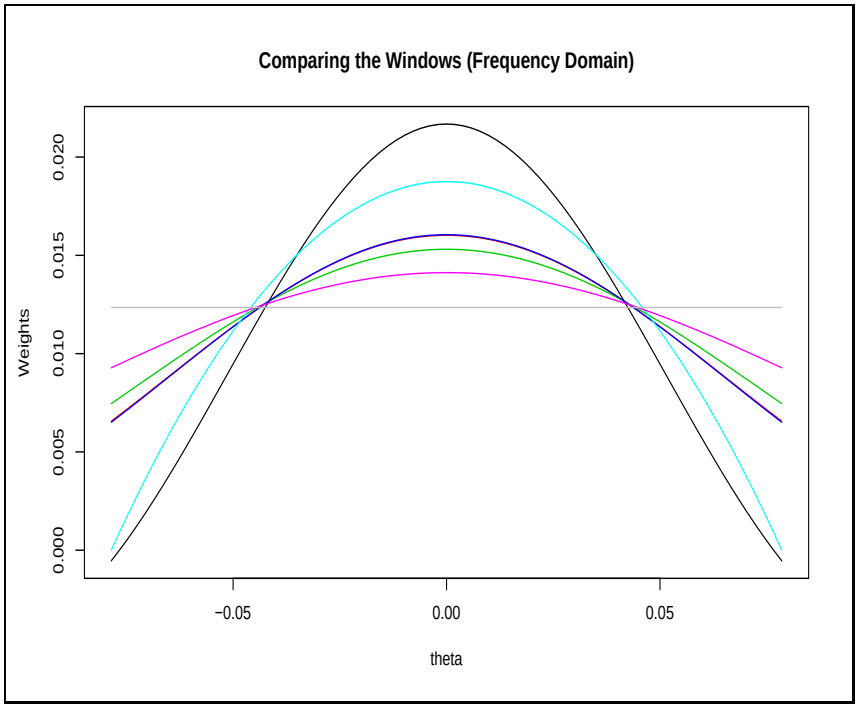

Figure 10.17: Comparing All Windows (Frequency Domain)

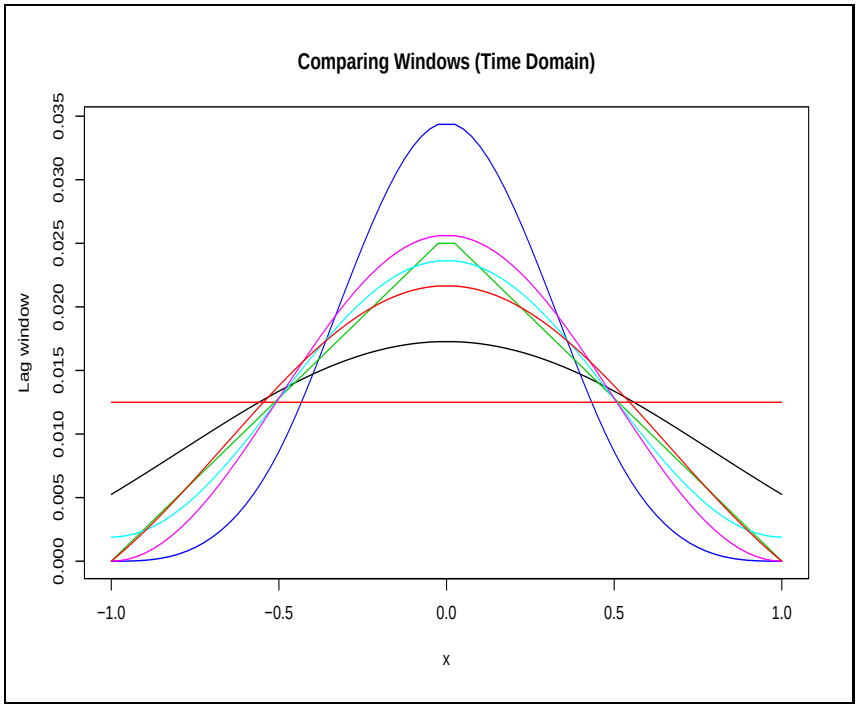

Figure 10.18: Comparing All Windows (Time Domain)

10.2 Akaike's 'An Information Criterion'

We would like to have a criterion to chose the window length *objectively*. The criterion used by Tatum and Hurvich is called *modified AIC* (AIC_C). Akaike's "An Information Criterion" (AIC) is an asymptotically unbiased estimator of the Kullback-Leibler information (relative entropy) which, in turn, is a measure of the information lost when a model is used to approximate full reality. As this AIC criterion derives from the Kullback-Leibler information, which itself is motivated by the Shanon entropy, we shall present these two concepts first. This section owes a lot to the report by Xenia Beate Rendtel ([16]).

10.2.1 Shanon Entropy

The Shanon entropy is a measure of how much "information" is contained in a "message", i.e. the number of bits needed to transmit it. A more complete treatment can be found in the paper by Rendtel ([16]).

Motivating Example

Let A be some finite or countable set. We use the notation A° to denote the set of finite "strings", i.e. vectors of finite length whose elements are in A. For example,

$$\{0,1\}^\circ = \{`', 0, 1, 00, 01, 10, 11, 000, \cdots\}$$

with $`'$ denoting the empty word, i.e. a vector of length 0. We can identify $\mathbb{N}$ with $\{0,1\}^\circ$ according to the following correspondence:

$$0 \mapsto `'$$

$$1 \mapsto 0$$

$$2 \mapsto 1$$

$$3 \mapsto 00$$

The length $l(x)$ of a word x in $\{0,1\}^\circ$ is the number of bits in it. For example, $l(010) = 3$ and $l(`') = 0$. We have:

$$l(x) = \text{floor } [\log_2[(1x)_d]]$$

where floor x is the largest integer smaller than x, $\log_2$ is the logarithm base 2, that is, $\log_2 x = ln(x)/ln(2)$, and $(1x)_d$ is the number you get by concatenating the string 1 and the string x and converting the result to decimal base.

Coding Interpretation

Definition 30 *(Encoding)*

Let A be a countable set. An *encoding* is any function

$$D : \left\{ \begin{array}{ccc} \{0,1\}^\circ & \to & A \\ x & \mapsto & D(x) \end{array} \right.$$

♢

It may well be that the encoding is not bijective and that a same word has several possible encodings. What we are interested in is saving as much energy as possible in the coding, i.e. using the shortest code possible.

Definition 31 *(Shortest encoding)*

Let A be a countable set and D an encoding on A. The *shortest encoding* is a function defined by:

$$L_D : \left\{ \begin{array}{ccc} A & \to & \mathbb{N} \\ x & \mapsto & \min_{y \in A} \{l(y) : D(y) = x\} \end{array} \right.$$

♢

In general, we cannot recover x and y from $D^{-1}(xy)$. Indeed, if D is the identity mapping, we have $D^{-1}(0000) = 0000 = D^{-1}(00)D^{-1}(00)$. This problem yields to the definition of prefixes, which make this ambiguity impossible.

Definition 32 *(Proper-prefix)*

Let $E \subseteq \{0,1\}^\circ, (a,b) \in E^2$. a is *proper-prefix* of b if $\exists c \in E, c \neq \text{`'} : b = ac$.

♢

Definition 33 *(Prefix-free)*

Let $E \subseteq \{0,1\}^\circ$. E is *prefix-free* if $\forall (a,b) \in E^2$, a is not proper-prefix of b and b is not proper prefix of a.

♢

Definition 34 *(Prefix-code)*

An encoding D is a *prefix-code* if its domain is prefix-free.

♢

Theorem 6 *(Kraft's inequality)*

Let A be a countable set and D be a prefix-code. Then we have the following inequality:

$$\sum_{x \in A} \frac{1}{2^{L_D(x)}} \leq 1$$

□

Proof:
We draw a binary tree where each layer represents a word length (the point on the top being the empty word) and each line represents either a 0 or a 1.

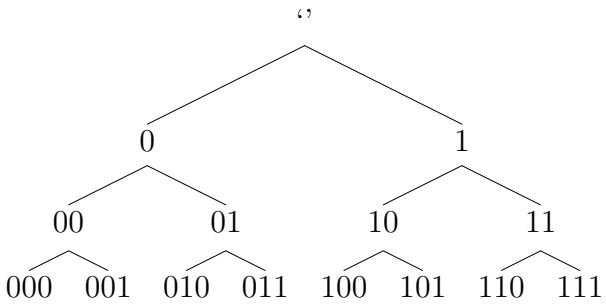

Figure 10.19: Binary tree

For example, in the second layer from the top, the left point is the word 0 and the right point the word 1. In the third layer, we have the words 00, 01, 10 and 11 in that order. Let L be the length of the biggest word. At the k-th layer, each point is above $2^{k-l_D(x)}$ words, which means that if $k = L$, then $\sum_{x\in A} 2^{L-l_D(x)} < 2^L$ because 2^L is the total number of words of length L. Dividing by 2^L gives the inequality.

♦

This means that a prefix-code cannot have too many short words. We now have the following theorem:

Theorem 7 *(Noiseless coding)*
Let A be a countable set, D a prefix-code on A and $(\Omega, \mathfrak{A}, P)$ a probability space. Let X be a random variable on Ω with values in A. $\forall x \in A, n_x := L_D(x)$ and $p_x := P(\mathsf{X} = x)$. Then,

$$\mathbb{E}L_D(\mathsf{X}) = \sum_{x\in A} n_x p_x \geq H(\mathsf{X})$$

with

$$H(\mathsf{X}) := \sum_{x\in A} p_x \log_2\left(\frac{1}{p_x}\right)$$

called *entropy* of X.

□

Proof:
We need to show that $H(\mathsf{X}) - \mathbb{E}L_D(\mathsf{X}) \leq 0$.

$$\begin{aligned} H(\mathsf{X}) - \mathbb{E}L_D(\mathsf{X}) &= \sum_{x \in A} p_x \left(\log_2 \left(\frac{1}{p_x} \right) - L_D(x) \right) \\ &= \sum_{x \in A} p_x \log_2 \left(\frac{1}{p_x} \right) + \log_2 2^{-L_D(x)} \\ &= \sum_{x \in A} p_x \log_2 \left(\frac{2^{-L_D(x)}}{p_x} \right) \end{aligned}$$

as $\log_2$ is concave, we can apply Jensen's inequality:

$$H(\mathsf{X}) - \mathbb{E}L_D(\mathsf{X}) \leq \log_2 \left(\sum_{x \in A} 2^{-L_D(x)} \right)$$

Kraft's inequality yields to:

$$\sum_{x \in A} \frac{1}{2^{L_D(x)}} \leq 1$$

Therefore,

$$H(\mathsf{X}) - \mathbb{E}L_D(\mathsf{X}) \leq \log_2(1) = 0$$

♦

This means that for any prefix-code, the average number of bits we send is at least the entropy of A. The entropy of a message is therefore the minimum average number of bits needed to code it. This is an empirical way to derive the entropy. A more axiomatic approach follows.

Axiomatic approach

This is an outline of the approach for *finite* samples from which the more general case is motivated, but as in this thesis we are only dealing with finite-length time series, we will only expose the finite sample case.

The idea is to build a function for every random variable that will give us how much "information" that the observation of a specific outcome gives us. For example, if we know that $\mathsf{x} = a$ with probability 1, then observing $x = a$ will not give us any information so we will set $I(a) = 0$.

Definition 35 *(Information of a random variable)*

Let $(\Omega, \mathfrak{A}, P)$ be a probability space and X a real valued random variable on Ω. A function

$$I_P : \left\{ \begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R} \\ x & \mapsto & I_P(x) \end{array} \right.$$

will be called *information function of* X *with respect to* P if there exists a function

$$S_{\mathsf{X}} : \left\{ \begin{array}{ccc} [0,1] & \rightarrow & \mathbb{R} \\ x & \mapsto & S_{\mathsf{X}}(x) \end{array} \right.$$

that satisfies the following properties:

1. $\forall x \in \mathbb{R}, I_P(x) = S_{\mathsf{X}}(P(\mathsf{X} = x))$,
2. $S_{\mathsf{X}}(1) = 0$,
3. S_{X} is continuous,
4. S_{X} is strictly decreasing,
5. $\forall (p,q) \in [0,1]^2, S_{\mathsf{X}}(pq) = S_{\mathsf{X}}(p) + S_{\mathsf{X}}(q)$

♢

Theorem 8 *(Expression of an information function)*

Let $(\Omega, \mathfrak{A}, P)$ be a probability space, and X a real-valued random variable on Ω. If I_P is an information function of X with respect to P and S_{X} is defined as above, then $\exists c > 0, \forall x \in \mathbb{R}, S_{\mathsf{X}}(x) = -c \log_2(x)$ and therefore

$$\forall x \in \mathbb{R}, I_P(x) = -c \log_2(P(\mathsf{X} = x))$$

□

Proof:

In all this proof, $(p,q) \in [0,1]$ and $(m,n) \in \mathbb{N}^2$. First, $S_{\mathsf{X}}(0) = 2S_{\mathsf{X}}(0)$ so $S_{\mathsf{X}}(0) = 0$. Then, $S_{\mathsf{X}}(p^2) = 2S_{\mathsf{X}}(p)$. We can show easily by induction that $S_{\mathsf{X}}(p^n) = nS_{\mathsf{X}}(p)$. Now if $n \neq 0$, $S_{\mathsf{X}}\left(\left(p^{\frac{m}{n}}\right)^n\right) = nS_{\mathsf{X}}\left(p^{\frac{m}{n}}\right) = mS_{\mathsf{X}}(p)$, therefore, $S_{\mathsf{X}}\left(\left(p^{\frac{m}{n}}\right)^n\right) = \frac{m}{n} S_{\mathsf{X}}(p)$. As $\mathbb{Q}$ is dense in $\mathbb{R}$, $\forall u \in \mathbb{R}, \exists (u_n)_{n \in \mathbb{N}}, \forall n \in \mathbb{N}, u_n \in \mathbb{Q}$ and $u_n \xrightarrow[n\infty]{} u$. By continuity of S_{X} we have:

$$S_{\mathsf{X}}(p^u) = \lim_{n \to \infty} S_{\mathsf{X}}(p^{u_n})$$

which means that

$$\forall u \in \mathbb{R}, S_{\mathsf{X}}(p^u) = uS_{\mathsf{X}}(p)$$

Let $c \in \mathbb{R}$ and define $S_{\mathsf{X}}(2) := -c$. $\forall x \in \mathbb{R}_+^*, x = 2^{\log_2 x}$. We then have

$$S_{\mathsf{X}}(x) = -c \log_2(x)$$

$c > 0$ because S_{X} has to be strictly decreasing.

⧫

The constant c corresponds to the basis in which the logarithm is defined. We choose that basis according to the number of elements we write our code with (in the case of binary coding, 2).

Definition 36 *(Shanon entropy)*

Let $(\Omega, \mathfrak{A}, P)$ be a probability space, and X a real-valued random variable on Ω. I_P is the information function of X with respect to P, with $c = 1$. Then the *Shanon entropy* of X is defined as:

$$H(\mathsf{X}) := \mathbb{E}_P I_P(\mathsf{X}) := \sum_{x \in \mathbb{R}} P(\mathsf{X} = x) I_P(x) = \sum_{x \in \mathbb{R}} P(\mathsf{X} = x) \log_2 \left(\frac{1}{P(\mathsf{X} = x)} \right)$$

♢

We can see Shanon's entropy as the average amount of information (expressing that information in a number of bits) that one receives when learning the value of a random variable: if $P(\mathsf{X} = x_0) = 1$ and we learn that $X = x_0$, we have not really gained any information because the event $\{\omega \in \Omega : \mathsf{X}(\omega) = x_0\}$ was certain. On the other hand, the smaller the probability, the more unlikely the event and therefore the more information we gain by learning it occurred.

10.2.2 Kullback-Leibler Discrepancy

If a random variable X has a distribution P_{X} that we approximate by another law Q_{X}, the Kullback-Leibler discrepancy is the average information lost when approximating P_{X} by Q_{X}, i.e. the average number of bits lost when performing this approximation. More formally,

Definition 37 *(Kullback-Leibler relative entropy)*

Let $(\Omega, \mathfrak{A}, P)$ be a probability space, Q a probability measure on $\mathfrak{A}$ and X a real-valued random variable on Ω with distribution P_{X}. I_P is the information function of X with respect to P, with $c = 1$ and I_Q with respect to Q. Then the Kullback-Leibler relative entropy of P and Q is defined by:

$$KL(P, Q) := \mathbb{E}_P(I_P(\mathsf{X}) - I_Q(\mathsf{X})) = \sum_{x \in \mathbb{R}} P(\mathsf{X} = x) \log_2 \left(\frac{Q(\mathsf{X} = x)}{P(\mathsf{X} = x)} \right)$$

♢

10.2.3 AIC and AIC_C

If we want to use the Kullback-Leibler relative entropy, we need to estimate it. Suppose we have two distributions P and Q, X being distributed according to P unknown, Q being known. We have:

$$KL(P,Q) = \mathbb{E}_P(I_P(\mathsf{X})) - \mathbb{E}_P(I_Q(\mathsf{X}))$$

P does not vary, as it is the theoretical distribution we are trying to approximate, so minimizing $KL(P,Q)$ is the same as maximizing $\mathbb{E}_P(I_Q(\mathsf{X}))$.

If we have a set of realizations of X say $(x_1, \cdots, x_n)$, we can try to estimate $\mathbb{E}_P(I_Q(\mathsf{X})) = \sum_{x\in\mathbb{R}} P(\mathsf{X}=x)\log_2(Q(\mathsf{X}=x))$.

Akaike showed in 1973 that the maximized likelihood can be used as a biased estimator of $\mathbb{E}_P(I_Q(\mathsf{X}))$ and that the bias, under some important but technical regularity assumptions, is:

$$\text{bias} \underset{n\infty}{=} \dim\Theta + o(1)$$

where $\dim\Theta$ is the dimension of the parameter space (i.e. the number of parameters in the model).

We then define Akaike's AIC by

Definition 38 *(AIC)*

Let K be the dimension of the parameter space and $\log L$ the log-likelihood. Then:

$$AIC = -2\log L + 2K$$

♢

AIC is therefore a first-order approximation of the Kullback-Leibler relative entropy and minimizing AIC is approximately equivalent to minimizing the Kullback-Leibler entropy. If the ratio sample size/number of parameters is small (usually $n/K < 40$) then the use of a second-order AIC is recommended (see Hurvich and Beltráo in [7]). This second-order AIC is the AIC_C.

Definition 39 *(AIC_C)*

$$AIC_C = -2\log L + 2K + \frac{2K(K+1)}{n-K-1}$$

♢

Chapter 11

Appendix C: Numerical Algorithms

11.1 Lagrangian Multipliers

Suppose we have the following problem. Let $n \in \mathbb{N}^*, f : \mathbb{R}^n \to \mathbb{R}, U \subseteq \mathbb{R}^n$: we are looking for $\boldsymbol{x}_0 \in U, \forall \boldsymbol{x} \in U, f(\boldsymbol{x}_0) \leq f(\boldsymbol{x})$. We will call this problem a *constrained minimization problem* and any point solving the constraint will be called a *feasible point*. f is sometimes called *cost function*. If $U = \ker g$ for some linear application $g : \mathbb{R}^n \to \mathbb{R}^m, (m, n) \in (\mathbb{N}^*)^2$, then the problem will be called *linear constrained minimization problem*. The aim of this section is to solve these linear constrained minimization problems.

11.1.1 Context

Definition 40 *(Level sets)*

Let $n \in \mathbb{N}^*$ and suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable function. Let $k \in f(\mathbb{R}^n)$. Then $f^{-1}(k)$ is called *k-level set*.

♢

Definition 41 *(Critical point)*

Let $n \in \mathbb{N}^*$ and suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable function. $\boldsymbol{x}_0 \in \mathbb{R}^n$ is said to be a *critical point* for f if $(\nabla f)(\boldsymbol{x}_0) = 0$.

♢

Definition 42 *(Regular value)*

Let $n \in \mathbb{N}^*$ and suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable function. A constant $k \in f(\mathbb{R}^n)$ is a *regular value* of f if the k-level set has no critical points in it.

♢

Definition 43 *(Topological manifold, regular surface)*
Let $n \in \mathbb{N}^*$ and suppose $(M, \mathcal{T})$ is a Hausdorff topological space, E^n being a Euclidean space of dimension n. M is said to be a *topological manifold* of E^n if every point of M has an open neighborhood homeomorphic to either an open subset of E^n or an open subset of the closed half of E^n. If $E = \mathbb{R}$, we call M an n-dimensional regular surface.

♢

Lemma 1
Let U be an open subset of $\mathbb{R}^n$. If $f : U \to \mathbb{R}$ is a differentiable function and $k \in f(U)$ is a regular value, then the k-level set is an $(n-1)$-dimensional regular surface.

□

The following theorem, which we will not prove (see Serge Lang in [8]), states that we can locally parametrize level sets.

Theorem 9 *(Inverse function)*
Let $n \in \mathbb{N}^*$. If $f : U \subseteq \mathbb{R}^n \to \mathbb{R}^n$ has a nonzero jacobian and if f is continuously differentiable on a neighborhood of $p \in U$, then there exists an open neighborhood V of p such as $f : V \to f(V)$ is a diffeomorphism.

□

Therefore, if k is a regular value, we can give a coordinate system to a neighborhood of all points of the k-level set.

11.1.2 Preliminary results

Lemma 2
Let $n \in \mathbb{N}^*$, U be an open set of $\mathbb{R}^n$, denote by $f : U \to \mathbb{R}$ a differentiable cost function and let $k \in \mathbb{R}$ be a regular value of f. Then:

$$\forall \boldsymbol{x} \in f^{-1}(k), \nabla f(\boldsymbol{x}) \perp f^{-1}(k)$$

where $\perp$ denotes orthogonality in the sense of the euclidean scalar product.

□

Proof:
Let $\boldsymbol{x} \in f^{-1}(k)$. In the previous section, we stated that $f^{-1}(k)$ is a $(n-1)$-dimensional regular surface. Therefore, *via* the inverse function theorem, there exists a parametrization $g : \mathbb{R}^{n-1} \to \mathbb{R}^n$ of a neighborhood of $f^{-1}(k)$ at $\boldsymbol{x}$. We can choose this parametrization g to fulfill $g(\mathbf{0}) = \boldsymbol{x}$.

For any unit vector $\boldsymbol{d} \in \mathbb{R}^{n-1}$, the directional derivative of $f(g(\boldsymbol{x}))$ in the direction $\boldsymbol{d}$ at $\mathbf{0}$ is given by:

$$\frac{\partial f(g(\kappa \boldsymbol{d}))}{\partial \kappa}|_{\kappa=0} = \langle (\nabla f)(\boldsymbol{x}), Dg(\mathbf{0})\boldsymbol{d} \rangle$$

Since f is constant on the image of g, this last quantity is 0 and therefore $\nabla f(\boldsymbol{x})$ is orthogonal to the tangent to the surface in the direction $\boldsymbol{d}$. Since $\boldsymbol{d}$ was arbitrary, $\nabla f(\boldsymbol{x})$ must be perpendicular to the entire surface.

♦

This lemma simply means that the gradient of the cost function is orthogonal to the constraint manifold.

Theorem 10 *(Lagrange multiplier with one constraint)*
Let $n \in \mathbb{N}^*$ and U be an open set of $\mathbb{R}^n$. Suppose that $f : U \to \mathbb{R}$ and $g : U \to \mathbb{R}$ are differentiable functions having a minimum in $\boldsymbol{x}_0$. Then

$$\exists \lambda \in \mathbb{R}, (\nabla f)(\boldsymbol{x}_0) = \lambda (\nabla g)(\boldsymbol{x}_0)$$

and λ is called *Lagrange multiplier*.

□

Proof:
We only need to show that both ∇f and ∇g must be orthogonal to $\ker g$ at $\boldsymbol{x}_0$. The case has already been argued for ∇h.

Now, suppose that $(\nabla f)(\boldsymbol{x}_0)$ is not orthogonal to $\ker g$ at $\boldsymbol{x}_0$. It follows that $\langle (\nabla f)(\boldsymbol{x}_0), \boldsymbol{d} \rangle < 0$ in at least one direction $\boldsymbol{d}$ tangent to $\ker g$.

But this means that if one moved along the surface $\ker g$ a little in that direction, one would obtain a smaller value of f, i.e. for $\kappa \in \mathbb{R}_+^*$ small,

$$f(\boldsymbol{x}_0 + \kappa \boldsymbol{d}) < f(\boldsymbol{x}_0)$$

This contradicts that $\boldsymbol{x}_0$ is a minimum cost. Therefore, both $(\nabla f)(\boldsymbol{x}_0)$ and $(\nabla g)(\boldsymbol{x}_0)$ must be orthogonal to $\ker g$. Since $\ker g$ is a hyperplane in $\mathbb{R}^n$, the two gradients must point in the same direction and are therefore multiples of one another:

$$(\nabla f)(\boldsymbol{x}_0) = \lambda (\nabla h)(\boldsymbol{x}_0)$$

♦

The Lagrange multiplier therefore represents how much a small change in the constraint will change the minimum.

11.1.3 Lagrange Multiplier Theorem

In the previous subsection, we solved the Lagrange multiplier problem with one constraint, i.e. the constraint manifold was reduced to one point. More generally, we might consider higher dimensional regular surfaces.

Theorem 11 *(Lagrange multiplier)*
Under the hypothesis:

- $(m, n) \in (\mathbb{N}^*)^2$,
- U is an open, non-empty set of $\mathbb{R}^n$,
- $f : U \to \mathbb{R}$ is a differentiable cost function,
- $\forall i \in [\![1, m]\!], g_i : U \to \mathbb{R}$ are differentiable functions and define $g := (g_1, \cdots, g_m)$,
- $\boldsymbol{x}_0 \in \ker g$ is a minimal cost of f,
- the vectors $((\nabla g_1)(\boldsymbol{x}_0), \cdots, (\nabla g_m)(\boldsymbol{x}_0))$ are independent,

the following result holds:

$$\exists \lambda = (\lambda_1, \cdots, \lambda_m) \in \mathbb{R}^m, (\lambda f)(\boldsymbol{x}_0) + \sum_{i=1}^{m} \lambda_i (\nabla g_i)(\boldsymbol{x}_0) = 0$$

The constants $(\lambda_1, \cdots, \lambda_m)$ are called *Lagrange multipliers*.

□

Proof:
By hypothesis, the Jacobi matrix $Dg(\boldsymbol{x}_0) = ((\nabla g_1)(\boldsymbol{x}_0), \cdots, (\nabla g_m)(\boldsymbol{x}_0))^\intercal$ has rank m. As $\forall i \in [\![1, m]\!], (\nabla g_j)(\boldsymbol{x}_0) \in \mathbb{R}^n$, $m \leq n$. If $m = n$, the system has a unique solution $\lambda \in \mathbb{R}^n$. We will therefore suppose that $m < n$. We can also suppose that the last m columns of the matrix $Dg(\boldsymbol{x}_0)$ are independent (if not, rename the g_i's).

We now decompose all vectors of $\mathbb{R}^n$ in two:

$$\boldsymbol{x} = (\boldsymbol{x}', \boldsymbol{x}'')$$

with

$$\boldsymbol{x}' := (x_1, \cdots, x_{n-m})$$

and

$$\boldsymbol{x}'' := (x_{n-m+1}, \cdots, x_n)$$

The Jacobi matrix can now be written:

$$\forall \boldsymbol{x} \in \mathbb{R}^n, Dg(\boldsymbol{x}) = (D_{\boldsymbol{x}'} g(\boldsymbol{x}), D_{\boldsymbol{x}''} g(\boldsymbol{x}))$$

By hypothesis, $D_{\boldsymbol{x}''} g(\boldsymbol{x}_0)$ is non-singular. The implicit function theorem then assures us that:

(i) $\exists \varepsilon > 0, \exists \delta > 0, \forall \boldsymbol{v} \in V := B^o(\boldsymbol{x}'_0, \varepsilon), \exists ! \boldsymbol{w} \in W := B^o(\boldsymbol{x}''_0, \varepsilon) : g(\boldsymbol{v}, \boldsymbol{w}) = 0.$

(ii) We can define a function $h : \left\{ \begin{array}{ccc} V & \rightarrow & W \\ u & \mapsto & g(u) = v \end{array} \right.$, h is continuously differentiable and $\boldsymbol{x}''_0 = h(\boldsymbol{x}'_0)$. If we define $J_{\boldsymbol{x}'} = D_{\boldsymbol{x}'} g(\boldsymbol{x}_0)$ and $J_{\boldsymbol{x}''} = D_{\boldsymbol{x}''} g(\boldsymbol{x}_0)$, we have:

$$Dh(\boldsymbol{x}'_0) = -(J_{\boldsymbol{x}''})^{-1} J_{\boldsymbol{x}'}$$

The cost function f now verifies:

$$\forall \boldsymbol{x} = (\boldsymbol{v}, \boldsymbol{w}) \in (V \times W) \cap \ker\, g, f(v, w) = f(v, h(v))$$

As $\boldsymbol{x}_0$ is a local minimum of f, $\exists r > 0, \forall \boldsymbol{x} \in B^o(\boldsymbol{x}_0, r), f(\boldsymbol{x}_0) \leq f(\boldsymbol{x})$. As all norms in $\mathbb{R}^n$ are equivalent, we can suppose that $B^o(\boldsymbol{x}_0, r)$ is defined with the maximum norm $\|\boldsymbol{x}\|_\infty := \max_{1 \leq i \leq n} |x_i|$. We then have $B^o(\boldsymbol{x}_0, r) = B^o(\boldsymbol{x}'_0, r) \times B^o(\boldsymbol{x}''_0, r)$. We can choose r so that $B^o(\boldsymbol{x}_0, r) \subset V \times W$. We then have

$$\forall \boldsymbol{x} \in B^o(\boldsymbol{x}_0, r) \cap \ker\, g, f(\boldsymbol{x}'_0, h(\boldsymbol{x}'_0)) = f(\boldsymbol{x}'_0, \boldsymbol{x}''_0) \leq f(\boldsymbol{x}) = f(\boldsymbol{x}', g(\boldsymbol{x}'))$$

We can now transform the constrained minimization problem into an unconstrained one by defining the function

$$\phi : \left\{ \begin{array}{ccc} \mathbb{R}^{n-m} & \rightarrow & \mathbb{R} \\ v & \mapsto & f(v, h(v)) \end{array} \right.$$

and $\forall v \in B^o(\boldsymbol{x}'_0, r), \phi(\boldsymbol{x}'_0) \leq \phi(v)$. Therefore,

$$(\nabla \phi)(\boldsymbol{x}'_0) = \boldsymbol{0},$$

but this last equality can also be written

$$(\nabla \phi)(\boldsymbol{x}'_0)^\mathsf{T} = (\nabla_{\boldsymbol{x}'_0} f)(\boldsymbol{x}_0)^\mathsf{T} + (\nabla_{\boldsymbol{x}''_0} f)(\boldsymbol{x}_0)^\mathsf{T} Dh(\boldsymbol{x}'_0) = \boldsymbol{0}$$

hence

$$(\nabla \phi)(\boldsymbol{x}'_0)^\mathsf{T} = (\nabla_{\boldsymbol{x}'_0} f)(\boldsymbol{x}_0)^\mathsf{T} - (\nabla_{\boldsymbol{x}''_0} f)(\boldsymbol{x}_0)^\mathsf{T} J_{\boldsymbol{x}''_0})^{-1} J_{\boldsymbol{x}'_0})$$

Let $\lambda := -(\nabla_{\boldsymbol{x}''_0} f)(\boldsymbol{x}_0)^\mathsf{T} J_{\boldsymbol{x}''_0})^{-1}$
which gives us, by right-multiplying by $J_{\boldsymbol{x}''_0}$,

$$(\nabla_{\boldsymbol{x}_0'} f)(\boldsymbol{x}_0)^\mathsf{T} + \lambda^\mathsf{T} J_{\boldsymbol{x}_0'} = 0$$

and by replacing the value of λ in a previous equation,

$$(\nabla_{\boldsymbol{x}_0''} f)(\boldsymbol{x}_0)^\mathsf{T} + \lambda^\mathsf{T} J_{\boldsymbol{x}_0''} = 0$$

which completes the proof.

⧫

We can now define what a "Lagrangian" is.

Definition 44 *(Lagrangian)*

- $(m, n) \in (\mathbb{N}^*)^2$,
- U is an open, non-empty set of $\mathbb{R}^n$,
- $f : U \to \mathbb{R}$,
- $\forall i \in [\![1, m]\!], g_i : U \to \mathbb{R}$,

then the *Lagrangian* of this system of equations is:

$$\forall \boldsymbol{x} \in U, \forall \lambda \in \mathbb{R}^m, L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x})$$

♢

The Lagrange multiplier conditions become:

$$(\nabla_{\boldsymbol{x}} L)(\boldsymbol{x}_0, \lambda^*) = \mathbf{0}$$
$$(\nabla_{\lambda} L)(\boldsymbol{x}_0, \lambda^*) = \mathbf{0}$$

11.1.4 Minimization of a Quadratic Form

This very general formulation of a linear constraint minimization problem can easily be adapted to the minimization of a quadratic form.

Theorem 12 *(Minimization of a quadratic form)*
Under the following assumptions:

- $(m, n) \in \mathbb{N}^*$,
- $\mathbf{A}$ is a symmetric, positive and definite $n \times n$ matrix,

- $f : \left\{ \begin{array}{ccc} \mathbb{R}^n & \rightarrow & \mathbb{R} \\ \boldsymbol{c} & \mapsto & \boldsymbol{x}^\intercal A \boldsymbol{x} \end{array} \right.$,
- $\mathbf{B}$ is an $m \times n$ matrix of rank m,
- $\boldsymbol{c} \in \mathbb{R}^m$,

the minimum value $\boldsymbol{x}_0$ of f verifying the constraint $\mathbf{B}\boldsymbol{x}_0 = \boldsymbol{c}$ is given by:

$$\boldsymbol{x}_0 = \mathbf{A}^{-1} B^\intercal (\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^\intercal)^{-1} \boldsymbol{c}$$

□

Proof:
If we define

$$\forall \boldsymbol{x} \in \mathbb{R}^n, g(\boldsymbol{x}) = \mathbf{B}\boldsymbol{x} - \boldsymbol{c}$$

the Lagrangian can be written:

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \forall \lambda \in \mathbb{R}^m, L(\boldsymbol{x}, \lambda) = f(x) + \lambda^\intercal g(\boldsymbol{x})$$

The first Lagrange multiplier condition is:

$$(\nabla_{\boldsymbol{x}} L)(\boldsymbol{x}_0, \lambda^*) = \mathbf{0}$$

which yields

$$2\mathbf{A}\boldsymbol{x}_0 + \mathbf{B}^\intercal \lambda^* = 0$$

and

$$\boldsymbol{x}_0 = -\frac{1}{2}\mathbf{A}^{-1}\mathbf{B}^\intercal \lambda^*$$

The second Lagrange multiplier condition is:

$$(\nabla_\lambda L)(\boldsymbol{x}_0, \lambda^*) = \mathbf{0}$$

This gives:

$$\mathbf{B}\boldsymbol{x}_0 = \boldsymbol{c}$$

Replacing $\boldsymbol{x}_0$ by the expression in the previous equation gives:

$$-\frac{1}{2}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^\intercal \lambda^* = \boldsymbol{c}$$

Replacing λ^* by this expression in the first Lagrange multiplier condition gives the result.

♦

11.2 Convergence Acceleration Algorithms

When using recursive methods, we often face the problem that the algorithm used converges too slowly to a solution and acquiring a single extra digit can prove to be ridiculously long. For example, to calculate $\ln 2$, we can use the series $\sum \frac{(-1)^{n+1}}{n}$ but as it is an alternate series, one has to calculate one million terms to get a correct result to the 6^{th} decimal place! It can therefore be necessary to accelerate the convergence, i.e. transform the initial sequence $(S_n)_{n\in\mathbb{N}}$ which converges to S, in a sequence $(t_n)_{n\in\mathbb{N}}$ such as $t_n - S \underset{n\infty}{=} o(S_n - S)$. A more complete treatment can be found in the books by André Hautot ([5]) and Claude Brezinski ([1]).

11.2.1 Richardson Algorithm

This is an extrapolation method, which means that we interpolate the sequence to accelerate with a polynomial and extrapolate the limit with a value of that polynomial.

Motivation

When graphing sequences, one often plots a point at each integer value on the x-axis. This means that one cannot possibly see all the terms of the sequence because as n grows larger and larger, so does the graph. Richardson solved this problem by plotting the sequence against another sequence tending to 0. For example, the sequence of the partial sums of the series $\sum \frac{1}{n^2}$ can be plotted against the sequence $\left(\frac{1}{n}\right)_{n\in\mathbb{N}^*}$ producing the result featured on Figure 11.1 page 161.

If we could pass a polynomial through these points, its value in zero might give us a reasonable estimation of the limit of the series, $\frac{\pi^2}{6}$ here. This is the basic idea of Richardson's algorithm.

Neville Extrapolation

As we only want to know the value of the polynomial in zero, there is no need to calculate all its coefficients. Indeed, we can use Neville's method: it is recursive and has the major advantage of not requiring the whole calculation to be done again if we add an extra point.

Let $M_i(x_i, y_i), 0 \leq i \leq n \in \mathbb{N}$ be the points to interpolate. Let A and B be two distinct points amongst the M_i's. Let P be the polynomial of degree n which interpolates the $n+1$ M_i's, P_A interpolating all the points except A and P_B all the points except B. We would like to calculate P from P_A and P_B. We are therefore looking for two first-degree polynomials U and V such as:

$$P(X) = U(X)P_A(X) + V(X)P_B(X)$$

in particular

$$\begin{cases} P(A) = U(A)P_A(A) + V(A)P_B(A) \\ P(B) = U(B)P_A(B) + V(B)P_B(B) \end{cases}$$

therefore

$$\begin{cases} U(A) = 0 \\ V(A) = 1 \end{cases}$$

and

$$\begin{cases} U(B) = 1 \\ V(B) = 0 \end{cases}$$

hence

$$P(X) = \frac{(X - X_B)P_B - (X - X_A)P_A}{X_A - X_B}$$

Richardson Algorithm

Let $(S_n)_{n\in\mathbb{N}}$ be a converging sequence of real numbers. Let $(x_n)_{n\in\mathbb{N}}$ be a decreasing sequence of real numbers converging to 0, for example, $x_n = \frac{1}{n}$. Let $(n_0, p) \in \mathbb{N}^2$. If we consider the $p+1$ points $(x_n, S_n), \cdots, (x_{n+p}, S_{n+p})$, the extrapolated limit we wish to obtain is the value in 0 of the polynomial of degree p passing through these $p+1$ points and we will denote it by $S_n^{(p)}$, identifying $(S_n^{(0)})_{n\in\mathbb{N}}$ with the original sequence. From the above extrapolation method we find that:

$$\forall n \in \mathbb{N}, S_n^{(0)} := S_n$$

$$\forall n \in \mathbb{N}, \forall p \in \mathbb{N}^*, S_n^{(p)} = \frac{x_n S_{n+1}^{(p-1)} - x_{n+p} S_n^{(p-1)}}{x_n - x_{n+p}}$$

This motivates the following definition:

Definition 45 *(Sequence transformation)*

Let E be a vector space of real sequences. We define a sequence transformation e on E by

$$\forall p \in \mathbb{N}, e_p : \begin{cases} E & \to & E \\ (S_n)_{n\in\mathbb{N}} & \mapsto & (e_p^{(n)}(S_n))_{n\in\mathbb{N}} \end{cases}$$

♢

Now a sequence transformation is only of any interest if it accelerates the convergence of a given sequence. For this to happen, it first needs to converge to the same limit.

Definition 46 *(Regular sequence transformation)*

Let E be a vector space of real sequences and e a sequence transformation. Let $I \subseteq \mathbb{N}$ and $(S_n)_{n\in\mathbb{N}}$ be a real sequence converging to $S \in \mathbb{R}$. e is said to be *regular on* I for $(S_n)_{n\in\mathbb{N}}$ if $\forall p \in I, (e_p^{(n)}(S_n))_{n\in\mathbb{N}} \overrightarrow{n\infty} S$. If $I = \mathbb{N}$, it is said to be *regular*.

♢

Which sequences can be accelerated by a convergence acceleration algorithm? The set of all such sequences will be called *kernel*.

Definition 47 *(Kernel)*
Let E be the vector space of all real converging sequences. Let e be a sequence transformation. The *kernel* of e is denoted by $\ker e$ and defined as:

$$\ker e := \{(S_n)_{n\in\mathbb{N}} \in E : \exists p \in \mathbb{N}, \exists S \in \mathbb{R}, \forall n \in \mathbb{N}, e_p^{(n)}(S_n) = S\}$$

♢

The kernel is therefore the set of sequences transformed at some point in the algorithm into constant sequences. The kernel is of particular interest for regular transformations (or at least regular on a big enough subset of $\mathbb{N}$), because in that case the acceleration algorithm outputs the limit of the accelerated sequence, which means it works perfectly. We will hence be looking for regular transformations with a kernel as big as possible.

The kernel of the Richardson transformation depends on the auxiliary sequence used. The problem is now to chose the "right" auxiliary sequence $(x_n)_{n\in\mathbb{N}}$. A special choice of the auxiliary sequence leads to Aitken's Δ^2-algorithm.

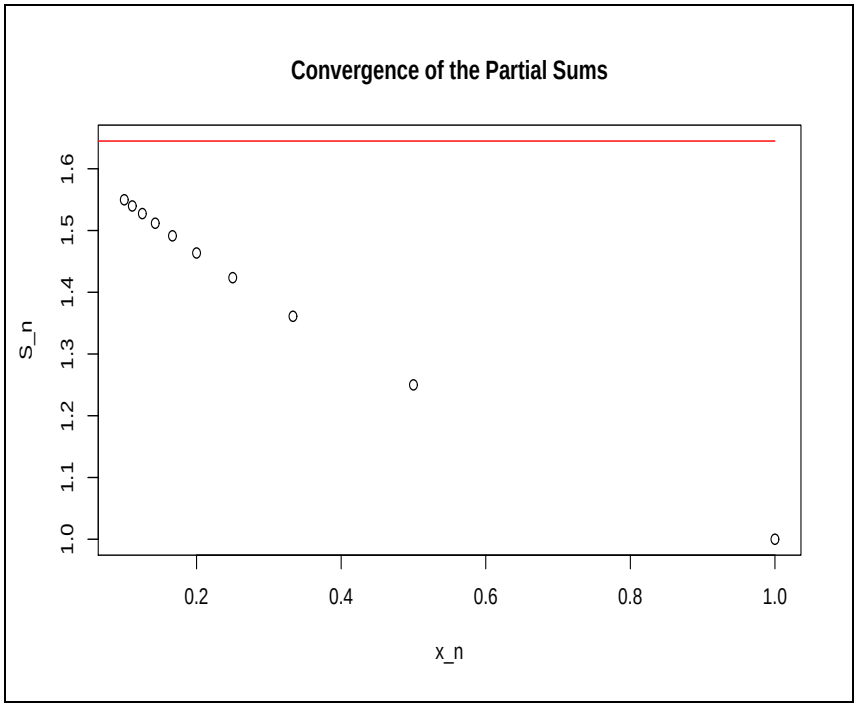

Figure 11.1: Illustration of Richardson's Algorithm

11.2.2 Aitken's Δ^2-algorithm

Construction

Intuitively, if we space out the points according to the difference between two terms of the sequence, we might expect to have a more "linear looking" polynomial. Therefore, Aitken proposed in 1926 the special choice $x_n = \Delta_n S_n := S_{n+1} - S_n$. Let $\Delta_n^2 S_n := (\Delta_n \circ \Delta_n)(S_n) = S_{n+2} - 2S_{n+1} + S_n$. Reporting in the definition of Richardson's algorithm, we get (for $p = 1, n \in \mathbb{N}$):

$$\begin{aligned}
T_n &= e_n^{(1)} = \frac{(\Delta_n S_n)S_{n+1} - (\Delta_n S_{n+1})S_n}{\Delta_n S_n - \Delta_n S_{n+1}} \\
&= \frac{(S_{n+1} - S_n)S_{n+1} - (S_{n+2} - S_{n+1})S_n}{-\Delta_n^2 S_n} \\
&= \frac{S_{n+1} - S_n S_{n+2}}{-\Delta_n^2 S_n} \\
&= \frac{S_n(S_{n+2} - 2S_{n+1} + S_n) + 2S_n S_{n+1} - S_n^2 - S_{n+1}^2}{\Delta_n^2 S_n} \\
&= \frac{S_n \Delta_n^2 S_n - (\Delta_n S_n)^2}{\Delta_n^2 S_n} \\
T_n &= S_n - \frac{(\Delta_n S_n)^2}{\Delta_n^2 S_n}
\end{aligned}$$

Hence the name "Δ^2-algorithm". using the notations of the previous paragraph, we can see that we stopped at $p = 1$. $p = 2$ corresponds to Steffenson's algorithm and the more general case of $p \in \mathbb{N}^*$ is called Overholt's algorithm..

Kernel of the Δ^2-transformation

As $p = 1$, we interpolate only two consecutive terms of the original sequence $(S_n)_{n\in\mathbb{N}}$ by a straight line (polynomial of degree $p = 1$) to get an approximation of the limit S of $(S_n)_{n\in\mathbb{N}}$. The algorithm interpolates two points $(\Delta_n S_n, S_n)$ and $(\Delta_n S_{n+1}, S_{n+1})$ by an affine function $y = ax + b$ where b is the value of interest as it is the value of the polynomial in 0. $T_n = e_n^{(1)}$ is therefore solution of the system:

$$\left\{ \begin{array}{rcl} a\Delta_n S_n + T_n &=& S_n \\ a\Delta_n S_{n+1} + T_n &=& S_{n+1} \end{array} \right.$$

Cramer's formula leads to:

$$T_n = \frac{\begin{vmatrix} \Delta_n S_n & S_n \\ \Delta_n S_{n+1} & S_{n+1} \end{vmatrix}}{\begin{vmatrix} \Delta_n S_n & 1 \\ \Delta_n S_{n+1} & 1 \end{vmatrix}}$$

As $p = 1$, the kernel will just be the set of all sequences such that,

$$(S_n \underset{n\infty}{\longrightarrow} S) \Rightarrow (e_n^1(S_n) = S)$$

Therefore, if $(S_n)_{n\in\mathbb{N}} \in \ker\, e$,

$$\begin{aligned}
\frac{\begin{vmatrix} \Delta_n S_n & S_n \\ \Delta_n S_{n+1} & S_{n+1} \end{vmatrix}}{\begin{vmatrix} \Delta_n S_n & 1 \\ \Delta_n S_{n+1} & 1 \end{vmatrix}} &= S \\
\begin{vmatrix} \Delta_n S_n & S_n - S \\ \Delta_n S_{n+1} & S_{n+1} - S \end{vmatrix} &= 0 \\
\begin{vmatrix} S_{n+1} - S_n & S_n - S \\ S_{n+2} - S_{n+1} & S_{n+1} - S \end{vmatrix} &= 0 \\
\begin{vmatrix} S_{n+1} - S & S_n - S \\ S_{n+2} - S & S_{n+1} - S \end{vmatrix} &= 0
\end{aligned}$$

This leads to: $\exists (a_0, a_1) \in \mathbb{R}^2, a_0(S_n - S) + a_1(S_{n+1} - S) = 0, a_0 + a_1 \neq 0$.
The kernel is therefore:

$$\ker\, e = \{(S_n)_{n\in\mathbb{N}} : \exists (\lambda, \mu) \in \mathbb{R}^2, S_n = S + \lambda\mu^n\}$$

The Δ^2-algorithm is therefore the same as extrapolating the sequence by an exponential one.

11.2.3 ε-algorithm

Generalization of the Δ^2-algorithm

The kernel of the Δ^2-algorithm is fairly small: one may wonder if it is possible to construct a sequence transformation with a kernel such as:

$$\ker\, e = \left\{ (S_n)_{n\in\mathbb{N}} : \exists (a_n)_{n\in\mathbb{N}} \in \mathbb{R}, \sum_{p=0}^{+\infty} a_p (S_{n+p} - S) = 0 \right\}$$

This transformation is known as "Shanks' transformation".

Shanks' Transformation

Let E be the $\mathbb{R}$-vector space of all convergent sequences of real numbers. $\forall p \in \mathbb{N}^*, F_p := \{(S_n)_{n\in\mathbb{N}} : \exists (a_n)_{n\in\mathbb{N}} \in \mathbb{R}, \sum_{k=0}^{p} a_k \neq 0, \sum_{k=0}^{p} a_k (S_{n+k} - S) = 0\}$. $F := F_\infty := \bigcup_{p\in\mathbb{N}} F_p$.

Definition 48 *(Shanks' transformation)*
The transformation e defined for all $p \in \mathbb{N}$ by

$$e^{(2p)} : \begin{cases} E & \to & E \\ (S_n)_{n\in\mathbb{N}} & \mapsto & e_n^{(2p)}(S_n) = \dfrac{\begin{vmatrix} S_n & S_{n+1} & \cdots & S_{n+p} \\ \Delta_n S_n & \Delta_n S_{n+1} & \cdots & \Delta_n S_{n+p} \\ \vdots & \vdots & & \vdots \\ \Delta_n S_{n+p-1} & \Delta_n S_{n+p} & \cdots & \Delta_n S_{n+2p-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta_n S_n & \Delta_n S_{n+1} & \cdots & \Delta_n S_{n+p} \\ \vdots & \vdots & & \vdots \\ \Delta_n S_{n+p-1} & \Delta_n S_{n+p} & \cdots & \Delta_n S_{n+2p-1} \end{vmatrix}} \end{cases}$$

is called *Shanks' transformation.*

♢

Theorem 13 *(Kernel of Shanks' transformation)*
$\ker e = F$

□

Proof:

- Lets show that $\ker e \subseteq F$
  Let $(S_n)_{n\in\mathbb{N}} \in \ker e$ with $S_n \underset{n\infty}{\longrightarrow} S$. By definition,

$$\exists p \in \mathbb{N}^*, \forall n \in \mathbb{N} e_n^{2p}(S_n) = S$$

  Therefore,

$$\forall n \in \mathbb{N}, S = \frac{\begin{vmatrix} S_n & S_{n+1} & \cdots & S_{n+p} \\ \Delta_n S_n & \Delta_n S_{n+1} & \cdots & \Delta_n S_{n+p} \\ \vdots & \vdots & & \vdots \\ \Delta_n S_{n+p-1} & \Delta_n S_{n+p} & \cdots & \Delta_n S_{n+2p-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta_n S_n & \Delta_n S_{n+1} & \cdots & \Delta_n S_{n+p} \\ \vdots & \vdots & & \vdots \\ \Delta_n S_{n+p-1} & \Delta_n S_{n+p} & \cdots & \Delta_n S_{n+2p-1} \end{vmatrix}}$$

  which means that

$$\forall n \in \mathbb{N}, \begin{vmatrix} S_n & S_{n+1} & \cdots & S_{n+p} \\ \Delta_n S_n & \Delta_n S_{n+1} & \cdots & \Delta_n S_{n+p} \\ \vdots & \vdots & & \vdots \\ \Delta_n S_{n+p-1} & \Delta_n S_{n+p} & \cdots & \Delta_n S_{n+2p-1} \end{vmatrix}$$

$$= \begin{vmatrix} S & S & \cdots & S \\ \Delta_n S_n & \Delta_n S_{n+1} & \cdots & \Delta_n S_{n+p} \\ \vdots & \vdots & & \vdots \\ \Delta_n S_{n+p-1} & \Delta_n S_{n+p} & \cdots & \Delta_n S_{n+2p-1} \end{vmatrix}$$

hence

$$\forall n \in \mathbb{N}, \begin{vmatrix} S_n - S & S_{n+1} - S & \cdots & S_{n+p} - S \\ \Delta_n S_n & \Delta_n S_{n+1} & \cdots & \Delta_n S_{n+p} \\ \vdots & \vdots & & \vdots \\ \Delta_n S_{n+p-1} & \Delta_n S_{n+p} & \cdots & \Delta_n S_{n+2p-1} \end{vmatrix} = 0$$

$$\forall n \in \mathbb{N}, \begin{vmatrix} S_n - S & S_{n+1} - S & \cdots & S_{n+p} - S \\ S_{n+1} - S & S_n - S & \cdots & S_{n+p+1} - S \\ \vdots & \vdots & & \vdots \\ S_{n+p} - S & S_{n+p+1} - S & \cdots & S_{n+2p} - S \end{vmatrix} = 0$$

which is equivalent to

$$\exists (a_0, \cdots, a_p) \in \mathbb{R}^{p+1}, \sum_{k=0}^{p} a_k (S_{n+k} - S) = 0, \sum_{k=0}^{p} a_k \neq 0$$

We have shown that $\exists p \in \mathbb{N}, (S_n)_{n\in\mathbb{N}} \in F_p$, so $(S_n)_{n\in\mathbb{N}} \in F$.

- Now lets show that $F \subseteq \ker e$
  Let $(S_n)_{n\in\mathbb{N}} \in F$. By definition, $\exists p \in \mathbb{N}, (S_n)_{n\in\mathbb{N}} \in F_p$. For such a p,

$$\exists (a_0, \cdots, a_p) \in \mathbb{R}^{p+1}, \sum_{k=0}^{p} a_k (S_{n+k} - S) = 0, \sum_{k=0}^{p} a_k \neq 0$$

  The rest of the demonstration is exactly as above in reverse order.

♦

Although Shanks' transformation is appealing, it is impractical because it requires the evaluation of determinants. Indeed, not only is this evaluation costly in terms of computation time, but this evaluation can be subject to roundoff errors, which is a major problem if the aim is to reach a high precision. For these reasons, P. Wynn (quoted by C. Brézinski in [1]) developed an algorithm called "ε-algorithm" which avoids the evaluation of these determinants.

ε-algorithm

Let $(S_n)_{n\in\mathbb{N}}$ be a converging sequence of real numbers.

$$\forall n \in \mathbb{N}, \varepsilon_n^{(-1)}(S_n) := 0, \varepsilon_n^{(-1)}(S_n) := S_n$$

and

$$\forall p \in \mathbb{N}^*, \varepsilon_n^{(p+1)}(S_n) := \varepsilon_{n+1}^{(p-1)}(S_n) + \frac{1}{\varepsilon_{n+1}^{(p)}(S_n) - \varepsilon_n^{(p)}(S_n)}$$

We have the following result, proven by Wynn but the demonstration of which we will not reproduce here:

Theorem 14 *(ε-algorithm and Shanks' transformation)*
Let E denote the set of all converging sequences of real numbers and e be Shanks' transformation.

$$\forall (S_n)_{n\in\mathbb{N}} \in E, \forall p \in \mathbb{N}^*, \forall n \in \mathbb{N}, \varepsilon_n^{(2p)}(S_n) = e_n^{(2p)}(S_n)$$

□

11.2.4 θ-algorithm

Limitations of the ε-algorithm

The kernel of the ε-algorithm may seem quite large at first, but it only contains exponentials and polynomials. It has been shown that, although alternating series are accelerated quite spectacularly by the ε-algorithm, it performs quite poorly on monotonous series. We are therefore looking for a transformation with a bigger kernel.

Generalisation of the ε-algorithm

In the ε-algorithm, only the sequences of even order (i.e. p even) are of interest, the others serving as intermediaries. The first idea is therefore to separate even terms from odd ones:

$$\forall n \in \mathbb{N}, \varepsilon_n^{(-1)}(S_n) := 0, \varepsilon_n^{(-1)}(S_n) := S_n$$

and

$$\forall p \in \mathbb{N}^*, \varepsilon_n^{(2p+1)}(S_n) := \varepsilon_{n+1}^{(2p-1)}(S_n) + \frac{1}{\varepsilon_{n+1}^{(2p)}(S_n) - \varepsilon_n^{(2p)}(S_n)}$$

$$\forall p \in \mathbb{N}^*, \varepsilon_n^{(2p+2)}(S_n) := \varepsilon_{n+1}^{(2p)}(S_n) + \frac{1}{\varepsilon_{n+1}^{(2p+1)}(S_n) - \varepsilon_n^{(2p+1)}(S_n)}$$

We then introduce a parameter ω_p:

$$\forall p \in \mathbb{N}^*, \varepsilon_n^{(2p+2)}(S_n) := \varepsilon_{n+1}^{(2p)}(S_n) + \frac{\omega_p}{\varepsilon_{n+1}^{(2p+1)}(S_n) - \varepsilon_n^{(2p+1)}(S_n)}$$

Now we would like $\varepsilon_n^{(2p+2)}(S_n)$ to converge quicker than $\varepsilon_n^{(2p)}(S_n)$. We have

$$\frac{\Delta_n \varepsilon_n^{(2p+2)}(S_n)}{\Delta_n \varepsilon_n^{(2p)}(S_n)} \xrightarrow[n\infty]{} 0$$

Let

$$D_n^{(2p+1)} := \frac{1}{\varepsilon_{n+1}^{(2p+1)}(S_n) - \varepsilon_n^{(2p+1)}(S_n)}$$

The condition can be written as:

$$\frac{\Delta_n \varepsilon_n^{(2p+2)}(S_n)}{\Delta_n \varepsilon_n^{(2p)}(S_n)} = 1 + \omega_p \frac{\Delta_n D_n^{(2p+1)}(S_n)}{\Delta_n e_{n+1}^{(2p)}(S_n)} \xrightarrow[n\infty]{} 0$$

This condition expresses the fact that the algorithm actually accelerates the convergence of S_n and it is always satisfied if we choose:

$$\omega_p := -\frac{\Delta_n e_{n+1}^{(2p)}}{\Delta_n D_n^{(2p+1)}}$$

hence the expression of the θ-algorithm:

$$\begin{aligned}
\theta_n^{(-1)}(S_n) &:= 0 \\
\theta_n^{(0)}(S_n) &:= S_n \\
\theta_n^{(2p+1)} &= \theta_{n+1}^{(2p-1)} + \frac{1}{\Delta_n \theta_n^{(2p)}} \\
\theta_n^{(2p+2)} &= \frac{\Delta_n(\theta_{n+1}^{(2p)} \Delta_n \theta_n^{(2p+1)})}{\Delta_n^2 \theta_n^{(2p+1)}}
\end{aligned}$$

as before, only the sequences of even indices are of interest.

Remarks

Brezinski discovered this algorithm by intuition: there is therefore no known justification today for this algorithm, except *a posteriori* considering the results it yields. Although it is very easy to implement, very few theoretical results exist: it is indeed very difficult to study. For example, its kernel is not known except for the first step ($p = 1$).

11.2.5 A New Approach

The problem with the θ-algorithm is that there is no justification for it except the experimental fact that it works well. Here, we will give a method to construct a sequence transformation which accelerates sequences from a given kernel.

Formulation of the problem

Suppose that the sequence of real numbers $(S_n)_{n\in\mathbb{N}}$ converges to S. Instead of trying to accelerate S_n directly, we will accelerate the remainder, i.e $S_n - S$. Let $(D_n)_{n\in\mathbb{N}}$ be a known sequence called *remainder estimate* and $(a_n)_{n\in\mathbb{N}}$ an unknown sequence called *correction term* and such as:

$$\forall n \in \mathbb{N}, S_n - S = a_n D_n$$

Let $K := \{(S_n)_{n\in\mathbb{N}} : S_n \underset{n\infty}{\longrightarrow} S, S_n = S + a_n D_n\}$. We will now assume that there exists a linear mapping of the set of sequences into itself L (called *annihilation operator* such as $\forall n \in \mathbb{N}, L(a_n) = 0$. We then have:

$$L\left(\frac{S_n}{D_n}\right) - SL\left(\frac{1}{D_n}\right) = L(a_n) = 0$$

therefore

$$S = \frac{L(S_n/D_n)}{L(1/D_n)}$$

This motivates the following sequence transformation:

Definition 49 *(Versatile transformation)*

Let $(S_n)_{n\in\mathbb{N}}$ be a sequence converging to S, and $(D_n)_{n\in\mathbb{N}}, (a_n)_{n\in\mathbb{N}}$ two other sequences. The *versatile transformation* is defined by:

$$\forall n \in \mathbb{N}, e_n^{(0)} := S_n,$$

and

$$\forall p \in \mathbb{N}^*, e^{(p)} : \left\{ \begin{array}{ccc} E & \rightarrow & E \\ (S_n)_{n\in\mathbb{N}} & \mapsto & e_n^{(p)}(S_n) := \frac{L(e^{(p-1)}(S_n)/D_n)}{L(1/D_n)} \end{array} \right.$$

♢

Theorem 15 *(Kernel of the versatile transformation)*

The kernel of the versatile transformation (using the notations of the definition) is

$$\ker\,(e) = \{(S_n)_{n\in\mathbb{N}} : S_n \underset{n\infty}{\longrightarrow} S, S_n = S + a_n D_n\}$$

□

Proof:
See above.

♦

Example

Simple yet powerful algorithms can be obtained by assuming that the annihilation operator is based on the finite difference operator Δ.

Lemma 3
$\forall n \in \mathbb{N}, \forall p \in \mathbb{N}, \Delta_n^{p+1} n^p = 0$

□

Proof:

$$\begin{aligned}
\Delta_n^{p+1} n^p &= \sum_{k=0}^{p+1} (-1)^{p+k} \binom{p}{k} u_{n+k} \\
&= \sum_{i=0}^{p} \binom{n}{i} n^i \sum_{k=0}^{p+1} (-1]^{p+k} \binom{p+1}{k} k^{p-i}
\end{aligned}$$

By induction on i, we will show that $\forall i \in \mathbb{N}, \sum_{k=0}^{p+1} (-1)^{p+k} \binom{p+1}{k} k^i = 0$. For $i = 1$,

$$\sum_{k=0}^{p+1} (-1)^{p+k} \binom{p+1}{k} = (1-1)^{p+1} = 0$$

Suppose the formula is true up until $i - 1$.

$$\begin{aligned}
\sum_{k=0}^{p+1} (-1)^k \binom{p}{k} k^i &= \sum_{k=1}^{p} (-1)^k \frac{p! k^i}{(p-k)! k!} \\
&= \sum_{k=1}^{p+1} (-1)^k \binom{p}{k} k^{i-1} \\
&= \sum_{k=0}^{p} (-1)^k \frac{p! k^i}{(p-k)!(k-1)!} \\
&= p \sum_{k=0}^{p} (-1)^k \binom{p}{k} k^{i-1} \\
&= 0
\end{aligned}$$

which completes the proof.

♦

A consequence of this lemma is the following theorem.

Theorem 16 *(Annihilation of a polynomial)*
Let $n \in \mathbb{N}$. For all polynomials P of degree $n-1$ in p,

$$\Delta_p^n P(p) = 0$$

□

Proof:
From the previous lemma we have:

$$\forall i < n, \Delta_p^n p^i = 0$$

As Δ_p^n is linear, we can apply this formula at each power of the polynom.

⧫

If we can find a sequence $(w_n)_{n\in\mathbb{N}}$ such that $\forall n \in \mathbb{N}, w_n a_n$ is a polynomial in n of degree $p-1$, $\Delta^p w_n a_n = 0$ and therefore, the weighted difference operator $L := a_n \mapsto \Delta^p w_n a_n$ annihilates $(a_n)_{n\in\mathbb{N}}$.

For example, if

$$a_n := \sum_{i=0}^{p-1} \frac{\alpha_i}{n^i},$$

choosing $w_n := n^{p-1}$ will give us a kernel of sequences such as $S_n = S + a_n D_n$. The choice $w_n := (n+\beta)^{p-1}$ gives an algorithm known as the Durbin-Levinson transformation.

All the algorithms presented above (except the θ-algorithm) can be seen as special cases of the versatile transformation.

11.3 Random Variable Generation

- "*Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.*" *in* John von Neumann, Various techniques used in connection with random digits, 1951.

11.3.1 Introduction

To test the filtering algorithms presented in this report, we need to be able to generate random variables, i.e. a sequence of numbers that appear to have been drawn from a specific distribution. As a random variable is, by definition, an (unknown) measurable function, we cannot really generate such variable: we can only generate sequences of numbers which have certain statistical properties.

These numbers are often called *pseudo random numbers*. The writing of this section was greatly helped by the symposium by G. Marsaglia in [9].

We only need to generate numbers which appear to come from uniform distribution as the other distributions can almost always be obtained by transformation of the uniform distribution. We will call these numbers *uniform deviates*.

Why not simply use a physical process (such as a clock) to generate these uniform deviates? For two reasons: the first is that we are not assured that the physical process used yields to a uniform distribution. Indeed, as we will see in the last part of this section, pseudo random number generators need to pass several statistical tests to be validated. There is no certainty that a sequence of random numbers generated by a physical process will pass these tests or at least not everytime. The second reason is that physically generated sequences will, by definition, not be reproducible. Although this may seem paradoxical, reproducibility is important in pseudo random number generation, especially in Monte-Carlo simulations. Indeed, one would like to be able to reproduce exactly the same results if necessary, e.g. to compare two versions of an algorithm, even though the sequence generated *looks* random.

As said before, the numbers generated are not really random. It is in fact impossible to generate an infinite sequence of random numbers: indeed, the generator program is stored in a computer's memory which has a finite length, which means that the generator can only be in a finite number of states, after which it will keep reproducing the same loop. The only way to change this is by allowing the memory allocated to the program to grow, but this cannot be done indefinitely. This unpleasant situation is not too much of a worry as it is quite simple to create a generator whose period is so big that the fastest computer in the world cannot complete it in less than the expected duration of the universe.

This section is divided into three parts: the first two parts explain various pseudo random number generation algorithms and the last part gives a couple of tests for random number generators.

11.3.2 Simple Generators

Three classes of generators will be presented here: the congruential generators, the shift-register generators and the lagged-Fibonacci generators.

Congruential Generators

The simplest class of generators is the class of congruential generators. They use linear transformations on the ring of reduced residues of some modulus $m \in \mathbb{N}^*$ to produce a sequence of integers:

$$(a, b) \in \mathbb{R}^2, m \in \mathbb{N}^*, (p_n)_{n\in\mathbb{N}}, \forall n \in \mathbb{N}^*, x_n = ax_{n-1} + b \bmod m$$

m is called *modulus*, a is called *multiplier* and b is called *increment*. The recurrence will eventually repeat itself with a period which can be no greater than the modulus. When choosing the coefficients, one wishes to maximize the period, i.e have a period of length m. The linear congruential method has the advantage of working remarkably well for some purposes, being very simple to implement and very fast, hence its extensive use. Nevertheless, it has the major disadvantage of not being free of sequential correlation on successive calls: if one draws k numbers at a time and uses them to plot points in a k-dimensional space (with each coordinate between 0 and 1), the points will tend to lie on "parallel planes" of dimension $k-1$ instead of filling up the hypercube. There will be at most $m^{1/k}$ such planes and if the increment, multiplier and modulus are not carefully chosen, there will be much less than that. For example, in one infamous routine called RANDU, used extensively for twenty years, $a = 65,539$, $m = 2^{31}$, there are eleven planes.

Shift-Register Generators (SRG)

In these generators, one generates a sequence of binary vectors starting from an initial vector $\boldsymbol{i}$ (called *seed vector*) and by multiplying with a binary matrix $\mathbf{P}$. This yields to the sequence $\boldsymbol{i}, \boldsymbol{i}\mathbf{P}, \ldots, \boldsymbol{i}\mathbf{P}^n$ where $\boldsymbol{i}$ is $1 \times n$ and $\mathbf{P}$ is $n \times n$. In the matrix - vector multiplication, all arithmetic is done modulo 2 and the addition is replaced with the exclusive-or operation (denoted by $\oplus$). The maximum possible period for a shift-register generator is $2^n - 1$.

Theorem 17 *(Maximum period condition for SRG)*
Let $\mathbf{P}$ be a non-singular (in the group of binary matrices) $n \times n$ matrix. A necessary and sufficient condition for the sequence $\boldsymbol{i}, \boldsymbol{i}\mathbf{P}, \ldots, \boldsymbol{i}\mathbf{P}^n$ to have period $2^n - 1$ is for the matrix $\mathbf{P}$ to have order $2^n - 1$.

□

Lagged-Fibonacci Generators (LFG)

These generators use two lags $(p_1, p_2) \in \mathbb{N}^*$ such as $p_1 < p_2$. Given an initial sequence $i_1, \ldots, i_{p_2}$, the following elements are generated by $\forall n \in \mathbb{N}, n > p_2, i_n = i_{n-p_1} \circ i_{n-p_2}$ where $\circ$ is a binary operation such as $+, -, *$ or $\oplus$ (exclusive 'OR'). If we denote by $F(p_1, p_2, \circ)$ a Fibonacci generator, examples of generators having maximal period are $F(17, 5, +)$ or $F(17, 5, -)$ on integers mod 2^k. We have the following result, resembling the one for shift-register generators:

Theorem 18 *(Maximum period condition for LFG)*
Let $(n, p) \in \mathbb{N}^*$ and $\mathbf{P}$ be an $n \times n$ matrix of integers with odd determinant. A necessary and sufficient condition for the sequence

$$\boldsymbol{i}, \boldsymbol{i}\mathbf{P}, \ldots, \boldsymbol{i}\mathbf{P}^n \bmod 2^p$$

to have period $(2^r - 1)2^{n-1}$ for every initial vector of integers $i = (m_1, \ldots, m_r)$ not all even and every $n \geq 1$, is for the matrix $\mathbf{P}$ to have order $j := 2^r - 1$ in the group of non-singular matrices for mod 2, order $2j$ for mod 4 and order $4j$ for mod 8.

□

11.3.3 Combination Generators

Having briefly reviewed three kinds of random number generators, one may wonder what happens if one tries to combine two or more generators with an algebraic operation such as $+, -, *$ or $\oplus$: do we get a better sequence? Is there even anything known about such combinations? The answer is given by the following theorem:

Theorem 19 *(Combination generators)*
Let $n \in \mathbb{N}^*$ and $\|.\|$ be any L_p-norm on $\mathbb{R}^n$ with $1 \leq p \leq +\infty$. Let P be a probability measure on $\mathcal{P}(\mathbb{N}_n)$ and

$$\delta : \left\{ \begin{array}{ccc} \mathcal{F}(\mathcal{P}(\mathbb{N}_n), \mathbb{N}_n^*) & \rightarrow & \mathbb{R}^+ \\ \mathsf{x} & \mapsto & \|(P(\mathsf{x}=1), \ldots, P(\mathsf{x}=n)) - (1/n, \ldots, 1/n)\| \end{array} \right.$$

Let $\circ$ be a binary operation such as $+, -, *$ or $\oplus$. Finally, let x and y be to independent random variables with values in $\mathbb{N}_n^*$. Then:

$$\delta(\mathsf{x} \circ \mathsf{y}) \leq \delta(\mathsf{x}) \text{ and } \delta(\mathsf{x} \circ \mathsf{y}) \leq \delta(\mathsf{y})$$

□

Proof:
The operation $\circ$ can be defined in a table. We will denote its elements by $\sigma(i, j)$.

$\circ$	1	2	...	n
1	$\sigma(1,1)$	$\sigma(1,2)$	...	$\sigma(1,n)$
2	$\sigma(2,1)$	$\sigma(2,2)$	...	$\sigma(2,n)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
n	$\sigma(n,1)$	$\sigma(n,2)$	...	$\sigma(n,n)$

Moreover, as $\circ$ is an algebraic operation, the above matrix is a permutation matrix. We can therefore define $\forall (i,j) \in \mathbb{N}_n^*, \tau(i,j) := k, \sigma(j,k) = i$.

Let $\boldsymbol{u} := (P(\mathsf{x}=1), \ldots, P(\mathsf{x}=n))$
and $\boldsymbol{v} := P(\mathsf{x} \circ \mathsf{y} = 1), \ldots, P(\mathsf{x} \circ \mathsf{y} = n))$.

We can write

$$\forall i \in \mathbb{N}_n^*, P(\mathsf{x} \circ \mathsf{y} = i) = \sum_{j=1}^{n} P((\mathsf{x} = i) \wedge (\mathsf{y} = \tau(i,j)))$$

As x and y are independent, we have:

$$\forall i \in \mathbb{N}_n^*, P(\mathsf{x} \circ \mathsf{y} = i) = \sum_{j=1}^{n} P(\mathsf{x} = i) P(\mathsf{y} = \tau(i,j))$$

If we denote by $\mathbf{M}$ the matrix $((P(\mathsf{y} = \tau(i,j))))_{1 \leq i,j \leq n}$, we have:

$$\boldsymbol{v} = \mathbf{M}\boldsymbol{u}$$

We now notice that $\forall i \in \mathbb{N}_n^*, \{\tau(i,j) : j \in \mathbb{N}_n^*\} = \mathbb{N}_n^*$, therefore

$$\forall i \in \mathbb{N}_n^*, \sum_{j=1}^{n} P(\mathsf{y} = \tau(i,j)) = \sum_{j=1}^{n} P(\mathsf{y} = j) = 1$$

Hence, with matrix-vector multiplications,

$$\mathbf{M}(1/n, \ldots, 1/n) = 1/n\mathbf{M}(1, \ldots, 1) = 1/n(1, \ldots, 1)$$

Moreover,

$$\forall p \in \mathbb{N}^* \cup \{+\infty\}, \forall \boldsymbol{x} \in (\mathbb{N}_n^*)^n, \|\boldsymbol{x}\|_p \leq 1 \Rightarrow \|\mathbf{M}\boldsymbol{x}\|_p \leq \|\boldsymbol{x}\|_p$$

Indeed,

$$\forall 1 \leq p < +\infty, \|\mathbf{M}\|_p^p = \sum_{i=1}^{n} \sum_{j=1}^{n} |m_{ij} x_j|^p \leq \sum_{i=1}^{n} \sum_{j=1}^{n} |x_j|^p = \|\boldsymbol{x}\|_p^p$$

For $p = +\infty$,

$$\|M\|_\infty = \max_{1 \leq i \leq n} \left| \sum_{j=1}^{n} m_{ij} x_j \right|$$

but all the m_{ij} and x_j are positive, therefore:

$$\|M\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |m_{ij} x_j| \leq \max_{1 \leq i \leq n} \sum_{j=1}^{n} |x_j| = \|\boldsymbol{x}\|_\infty$$

We have therefore proven that: $\forall p \in \mathbb{N}^* \cup \{+\infty\}, \forall \boldsymbol{x} \in (\mathbb{N}_n^*)^n, \|\boldsymbol{x}\|_p \leq 1 \Rightarrow \|\mathbf{M}\boldsymbol{x}\|_p \leq \|\boldsymbol{x}\|_p$. This means exactly that $\|\mathbf{M}\|_p \leq 1, \forall p \in \mathbb{N}^* \cup \{+\infty\}$. We can now write:

$$\begin{aligned} \delta(\mathsf{x} \circ \mathsf{y}) &= \|\mathbf{M}\boldsymbol{u} - \mathbf{M}(1/n, \ldots, 1/n)\| \\ &= \|\mathbf{M}(\boldsymbol{u} - (1/n, \ldots, 1/n))\| \\ &= \leq \|\boldsymbol{u} - (1/n, \ldots, 1/n)\| \\ &= \delta(\mathsf{x}) \end{aligned}$$

The other inequality is derived in a similar fashion.

♦

This means we may use any binary operation $\circ$ to form a new (pseudo-)random variable from two existing ones and the resulting variable will actually be "closer" to the uniform distribution than either of the starting variables.

This concludes the random generation part. Having generated a sequence, how can we assess its quality?

11.3.4 Tests for Random Number Generators

There exists no absolute method to determine the quality of a generator. The best we can do is to see if a generator passes a series of tests which usually boils down to asking "does the sequence *look* uniform?" in a more or less sophisticated manner. We will shortly review three of these tests. A naïve question would be: why not use the Pearson χ^2-test? The answer is, that such a test, although having a perfectly valid statistical justification, is easily passed. One can easily design tests for random number generation. The difficulty is creating a test "difficult to pass".

Overlapping m-tuple Tests

Suppose we wish to test the sequence $i_1, i_2, \ldots, i_n$. We can build the following overlapping sequence of triples:

$$(i_1, i_2, i_3), (i_2, i_3, i_4), \ldots, (i_{n-3}, i_{n-2}, i_{n-1}), (i_{n-2}, i_{n-1}, i_n)$$

if n is a multiple of three. If not, we can add the first element or the first two elements of the sequence at the end and the result will (asymptotically) be unchanged.

For $\forall (i,j,k) \in \mathbb{N}^3$, let ω_{ijk} be the number of times that the triple (i,j,k) appears in the triple sequence. If the i's all take b different values, there are b^3 different possible triples. If the variables are independent and uniformly distributed, the ω_{ijk} should be joint normal with means $\mu_{ijk} = \frac{n}{b^3}$. We can now build the quantities:

$$Q_3 := \sum_{i=1}^{b}\sum_{j=1}^{b}\sum_{k=1}^{b} \frac{(\omega_{ijk} - \mu_{ijk})^2}{\mu_{ijk}}$$

$$Q_2 := \sum_{i=1}^{b}\sum_{j=1}^{b} \frac{(\omega_{ij} - \mu_{ij})^2}{\mu_{ij}}$$

where ω_{ij} and μ_{ij} are defined in a similar fashion to ωijk and μ_{ijk}.

$Q_3 - Q_2 \sim \chi^2_{b^3-b^2}$. More generally, if each number $1, \ldots, b$ has a probability $(p_1, \ldots, p_b)$,

$$Q_3 := \sum_{i=1}^{b}\sum_{j=1}^{b}\sum_{k=1}^{b}\frac{(\omega_{ijk} - np_ip_jp_k)^2}{np_ip_jp_k}$$

$$Q_2 := \sum_{i=1}^{b}\sum_{j=1}^{b}\frac{(\omega_{ij} - np_ip_j)^2}{np_ip_j}$$

The test is therefore to calculate $Q_3 - Q_2$ and use the $\chi^2_{b^3-b^2}$ table to assess if the variable is uniform.

Overlapping Permutation Tests

As before, we build the overlapping sequence of triples

$$(i_1, i_2, i_3), (i_2, i_3, i_4), \ldots, (i_{n-3}, i_{n-2}, i_{n-1}), (i_{n-2}, i_{n-1}, i_n)$$

Each triple (i_{k-1}, i_k, i_{k+1}) can be in one of the following states:

1. $i_{k-1} < i_k < i_{k+1}$
2. $i_{k+1} < i_{k-1} < i_k$
3. $i_k < i_{k+1} < i_{k-1}$
4. $i_k < i_{k-1} < i_{k+1}$
5. $i_{k+1} < i_{k-1} < i_k$
6. $i_{k-1} < i_{k+1} < i_k$

If we associate each triple with its state, we get a sequence of numbers between 1 and 6 that we call a state sequence. As before, let ω_{ijk} be the number of times that the triple (i, j, k) appears in the state sequence. We then find the means and covariance matrix $\mathbf{C}$ and any weak inverse $\mathbf{C}^-$ of $\mathbf{C}$. We can then build the quantity

$$\sum_{i=1}^{6}\sum_{j=1}^{6}\sum_{k=1}^{6}\sum_{r=1}^{6}\sum_{s=1}^{6}\sum_{t=1}^{6}(\omega_{ijk} - \mu_{ijk})\overline{c_{ijk,rst}}(\omega_{rst} - \mu_{rst})$$

which will asymptotically have a χ^2 distribution. This test is not very stringent and except for the lagged-Fibonacci generators, most generators seem to pass it.

Monkey Tests

"We've all heard that a million monkeys banging on a million typewriters will eventually reproduce the entire works of Shakespeare. Now, thanks to the Internet, we know this is not true."
— Robert Wilensky, University of California

Based on this profound statement, George Marsaglia, professor of statistics at the University of Ohio developed a series of statistical tests known as the "Diehard tests". In his article *Random numbers fall mainly in the planes* he was the first to show that if you plot the output of a pseudo-random number generatora in several dimensions, then the points often tend to lie on a lattice rather than being uniformly distributed.

For example, the Figure 11.2 page 179 is that of a "good" congruential generator, namely $x_i = 133x_{i-1} + 5 \bmod 216$, and the Figure 11.3 page 179 is a "bad" generator ($x_i = 109x_{i-1} + 5 \bmod 216$).

The first step of this test is to convert the random number generated into letters. We can then compare the output of the generator to what a monkey would type on a typewriter. The first test is quite inefficient but many generators fail it. It consists in counting the number of letters "typed" before a certain word appears (for example, "CAT"). There are 26^3 three-letter words and the number of letters needed for the first cat to appear is hypergeometrically distributed so the expected number of strokes is 26^3. Indeed, if we consider the sequence of the first k letters, we can draw three consecutive letters (there are 26^3 different ways to do this). The probability that a drawn sequence spells out "CAT" is $\frac{1}{26^3}$. In order to have exactly one cat in the first k letters, there has to be one subsequence with "CAT", the others being different. The probability of this event is:

$$P(T = k) = \frac{1}{26^3}\left(1 - \frac{1}{26^3}\right)^{k-1}$$

Therefore, the expected number of letters needed to get the first cat is:

$$\mathbb{E}T = \sum_{n=1}^{+\infty} nP(T = n)$$

Let

$$r := 1 - \frac{1}{26^3}, f_n(r) := \sum_{k=1}^{n} r^k$$

and

$$S := \sum_{n=1}^{+\infty} nr^{n-1}$$

We know that

$$S = \lim_{n\to+\infty} f_n'(r) = \left(\lim_{n\to+\infty} f_n(r)\right)'$$

Therefore

$$\mathbb{E}T = \frac{1}{1-r} = 26^3$$

This means that if a random generator produces a cat after a lot more or a lot less than 26^3 letters, the sequence is not uniform. This test might seem a bit silly, but some generators fail it, for example, the shift-generator that produces 31-bits integers by XOR's, left-shift 28 and right-shift 3, suggested to replace congruential generators after the discovery of their lattice structure. It never spells 'CAT', no matter how long it runs. In fact, there are certain words, such as DOG, GOD or SEX which it will never produce. This does not mean that some letters appear more frequently than others: in fact, this generator is quite satisfactory in this respect. This shows that one cannot content oneself with just one statistical test for a random number generator. It also shows that the tests do not need to be very elaborate to show flaws in random number generators. Passing the monkey test does not mean that a generator is good. Failing it however, does mean a generator is bad.

A less naïve approach is, if the total number of letters is a multiple of 6, say, to count the number of occurrences of all possible two-letter words and of all possible three-letter words, denoting these frequencies by ω_i and μ_i respectively. We can then build two quantities (asymptotically) χ^2-distributed:

$$Q_2 := \sum_{i=1}^{26^2} \frac{(\omega_i - N/26^2)^2}{N/26^2}$$

$$Q_3 := \sum_{i=1}^{26^3} \frac{(\omega_i - N/26^3)^2}{N/26^3}$$

The difference between Q_3 and Q_2 is also (asymptotically) χ^2 distributed.

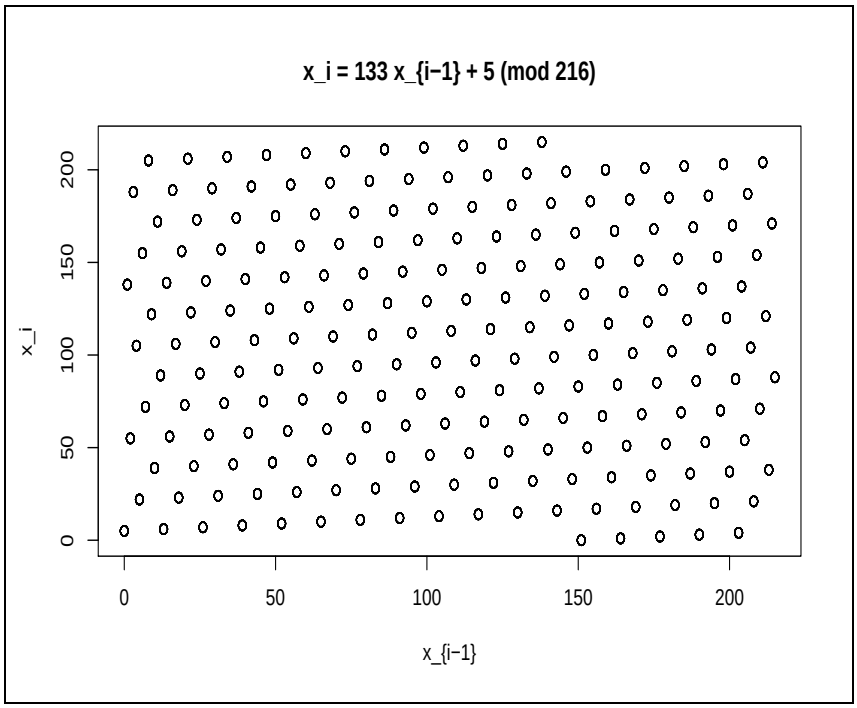

Figure 11.2: Good Distribution

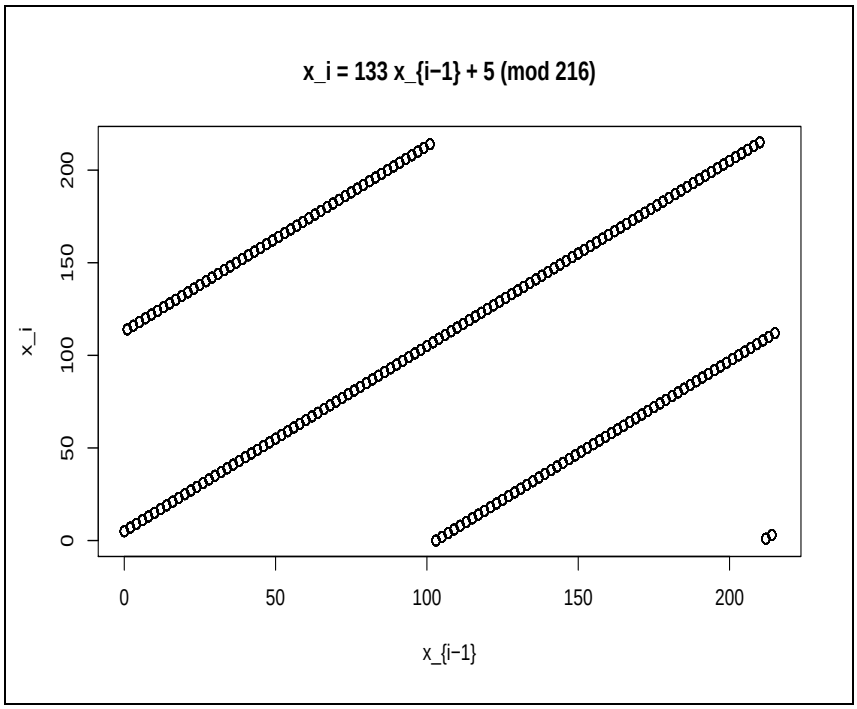

Figure 11.3: Bad Distribution

Chapter 12

GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

12.1 Applicability and Definitions

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **"Document"**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a

publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

12.2 Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

12.3 Copying in Quantity

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

12.4 Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled ”Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled ”Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ”Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

12.5 Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique

number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

12.6 Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

12.7 Aggregation with Independant Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

12.8 Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to

the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

12.9 Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

12.10 Future Revisions of this License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

12.11 Addendum: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliography

[1] Claude Brézinski. Convergence acceleration during the 20th century. *J. Comput. Appl. Math.*, 2:Interpolation and extrapolation:1–21, 2000.

[2] Richard Dudley. Mathematical statistics, spring 2003. http://www.core.org.cn/OcwWeb/Mathematics/18-466Mathematical-StatisticsSpring2003/LectureNotes/index.htm, March 2003.

[3] Armin I. Günther. 2nd international symposium on image and signal processing and analysis. In *Bias and Variance of Average and Smoothed Periodogram-based Log-Amplitude Spectra*, 2001.

[4] F.R. Hampel, W.A. Stahel, E.M. Ronchetti, and P. Rousseeuw. *Robust Statistics: The approach Based on Influence Functions*. John Wiley & Sons, New York, 1986.

[5] André Hautot. Accélération de la convergence en analyse numérique. Cours de l'Université de Liège, March 1993.

[6] P.J. Huber. Robust covariances. In S. Gupta and D. Moore, editors, *Statistical Decision Theory and Related Topics*. Vol. II, Academic Press, New York, 1977.

[7] C.M. Hurvich and Beltrāo. Cross-validatory choice of a spectrum estimate and its connections with AIC. *J. Time Series Analysis*, 11(2):121–137, 1990.

[8] Serge Lang. *Real and Functional Analysis*. Springer Verlag, 1993.

[9] George Marsaglia. A current view of random number generators. In Elsevier Press, editor, *Keynote Address, Computer Science and Statistics: Proc. of the 16th Symposium on the Interface*, 1984.

[10] Makoto Matsumoto and Takuji Nishimura. A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM transactions on Modeling and Computer Simulations*, 8(1):3–30, January 1998.

[11] D.B. Percival and A.T. Walden. *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge University Press, Cambridge, 1993.

[12] M.B. Priestley. *Spectral Analysis and Time Series*, volume 1 of *Probability and Mathematical Statistics*. Academic Press, London, 1981.

[13] Luc Pronzato and Andrej Pázman. Recursively re-weighted least-squares estimation in regression models with parametrized variance. *12th European Signal Processing Conference*, pages 621–624, September 2004.

[14] J. Pumprla, K. Howorka, D. Groves, M. Chester, and J. Nolan. Functional assessment of heart rate variability: physiological basis and practical applications. *Int. J. Cardiology*, 84:1–14, 2002.

[15] Bernard Rapacchi. Une introduction à la notion de robustesse. Centre Interuniversitaire de Calcul de Grenoble, January 2000.

[16] Xenia Beate Rendtel. Codierung und entropie. Stochastikseminar bei Prof. Dr. Rösler, Wintersemester 2000.

[17] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, New York, 1987.

[18] L.G. Tatum and C.M. Hurvich. A frequency domain approach to robust time series analysis. In Morgenthaler, Ronchetti, and Stahel, editors, *New Directions in Statistical Data Analysis and Robustness*. Birkhäuser-Verlag, Basel, 1993.

[19] L.G. Tatum and C.M. Hurvich. High breakdown methods of time series analysis. *J. Royal Statist. Soc. B*, 55(4):881–896, 1993.

Index