Universidade Federal de São Carlos
Centro de Ciências Exatas e de tecnologia
Departamento de Estatística

UM ESTUDO SOBRE *WAVESTRAP*

Kae da Silva Gremes

Trabalho de Conclusão de Curso

Kae da Silva Gremes

Um estudo sobre *wavestrap*

Este exemplar corresponde à redação final do trabalho de conclusão de curso devidamente corrigido e defendido por "Kae da Silva Gremes" e aprovado pela banca examinadora.

São Carlos,
27 de Junho de 2021

Banca Examinadora

- Profª Drª Maria Silvia de Assis Moura (Orientador)
- Profª Drª Michel Helcias Montoril
- Profª Drª Renato Jacob Gava

Agradecimentos

Agradeço a todos aqueles que acreditaram em mim, mesmo quando eu quis desistir.

Resumo

Ondaletas são bases de funções que podem ser utilizadas para descrever ou aproximar sinais contínuos (funções) ou sinais discretos (sequências); o estudo de ondaletas ganhou destaque considerável após o desenvolvimento das ondaletas de Daubechies (DAUBECHIES, 1988). Também na segunda metade do século XX, o avanço na tecnologia e capacidade de processamento possibilitou a introdução do *bootstrap* (EFRON, 1979), uma técnica computacionalmente intensiva baseada em reamostragem.

Uma das principais suposições para a aplicação do *bootstrap* é que os elementos sejam não correlacionados, o que em geral não ocorre na análise de séries temporais. Este estudo apresenta o *wavestrap*, técnica de aplicação do *bootstrap* aos coeficientes de uma Transformada Ondaleta Discreta (DWT). Essa técnica é comparada com outras adaptações de *bootstrap* para séries temporais. São analisados métodos de desenvolvimento de intervalos de confiança não paramétricos para a primeira autocorrelação de processos autorregressivos de primeira ordem.

Palavras-chave: *Bootstrap, Séries Temporais, Wavelets, Wavestrap.*

Abstract

Wavelets are basis of function spaces that can be used to represent both continuous (functions) and discrete (sequences) signals; wavelets study gained great notoriety after the work of Daubechies, who developed a wavelet family with compact support (DAUBECHIES, 1988). Also in the second half of twentiest century the great advances in computer processing allowed the emergence of various computation intensive methods, such as bootstrap (EFRON, 1979).

One of the key assumptions to use bootstrap is that the sample elements are not correlated, generally that is not a characteristic found in time series analysis. This study presents a review on wavestrap: a technique that joins both wavelet analysis and bootstrap resampling. By applying bootstrap to the wavelet transform coefficients we can generate samples that retain roughly the same characteristics of the original signal. We also analyze other nonparametric confidence intervals based on bootstrap for estimating the first autocorrelation of first order autorregressive processes.

Keywords: *Bootstrap, Timeseries, Wavelets, Wavestrap.*

Sumário

Lista de Figuras

Convenções e notações

- FT: Transformada de Fourier (*Fourier transform*);
- WT: Transformada Ondaleta (*Wavelet transform*);
- DFT, DWT: Transformada Discreta de Fourier e Transformada Ondaleta Discreta (respectivamente);
- CFT, CWT: Transformada Contínua de Fourier e Transformada Ondaleta Contínua (respectivamente);
- L_2: Espaço das funções quadrado-integráveis;
- T: Conjunto totalmente ordenado que indexa um processo estocástico;
- $\mathbf{t}$: Subconjunto dos instantes de T que foram observados;
- t: Instante no tempo ($t \in \mathbf{t}$);
- $Z(t),\ Z_t$: Variável aleatória no instante $t \in \mathbf{t}$;
- z_t: Observação de $Z(t)$;
- $\gamma(t_1, t_2)$: Função de autocovariância de um processo estocástico;
- $\gamma(h)$: Função de autocovariância de um processo estacionário;
- $\rho(t_1, t_2)$: Função de autocorrelação de um processo estocástico;
- $\rho(h)$: Função de autocorrelação de um processo estacionário;
- $\langle a, b \rangle$: Produto interno entre a e b (quando definido);
- $\varphi(x)$: Função ondaleta pai;
- $\psi(x)$: Função ondaleta mãe;

- $\varphi_{0k}(x),\ \psi_{jk}(x)$: $\varphi(x-k)$ e $2^{j/2}\psi(2^j x-k)$ respectivamente;
- $\alpha_{0k},\ \beta_{jk}$: Coeficientes associados a WT;
- $\hat{\alpha}$: Quando α for algum parâmetro de interesse, é a estimativa do parâmetro;
- $\hat{f}$: Quando f for uma função, é a Transformada Contínua de Fourier de f;
- $f^{(p)}$: p-ésima derivada de f;
- $\mathbf{W}$: Vetor de coeficientes DWT de um uma série temporal;
- $\mathbf{W}_j$: Subvetor dos coeficientes DWT de uma série temporal associados ao nível de frequência j;
- E_j: Energia associada ao nível de frequência j;

Capítulo 1

Introdução

O estudo de séries temporais é um campo da estatística que tem avançado constantemente ao longo das últimas décadas. A popularização dos microcomputadores e da internet impulsionou o desenvolvimento de novas técnicas nessa área (assim como em muitas outras) ao longo da segunda metade do século XX; como exemplos, podemos citar a modelagem ARIMA (BOX e JENKINS, 1970) ou os métodos de alisamento exponencial (GARDNER JR, 1985).

Em geral, as séries temporais são observadas no domínio do tempo (isto é, são coletadas observações adjacentes no tempo). Através da Transformada de Fourier (que abreviaremos como FT) é possível visualizar uma série temporal no domínio da frequência. As ondaletas são interessantes pois nos permitem visualizar a informação presente na série temporal tanto no domínio da frequência como no domínio do tempo, ou seja, enquanto a FT apresenta uma série temporal como uma soma de frequências, a Transformada Wavelet (WT) apresenta essa mesma série como várias somas de frequências adjacentes no tempo.

Outra técnica que ganhou destaque durante a segunda metade do século XX é o *bootstrap*; o *bootstrap* é uma técnica baseada em reamostragem computacionalmente intensiva, sendo utilizada tanto para estimação quanto para cálculo de previsões. A aplicação do *bootstrap* em séries temporais exige uma série de adaptações.

Uma das (poucas) suposições do *bootstrap* é que os elementos da população (que serão reamostrados) sejam não correlacionados. Isto não ocorre naturalmente na análise de séries temporais e diversas adaptações foram desenvolvidas para contornar este problema. Uma das possíveis formas de se aplicar o *bootstrap* a séries temporais é através do *wavestrap*, técnica na qual se reamostram os coeficientes de ondaleta de uma série temporal. Este estudo consiste em uma revisão desta técnica.

O próximo capítulo deste trabalho será dedicado ao estudo das ondaletas, contendo também uma revisão de toda a teoria necessária para o entendimento dessas técnicas e um exemplo de aplicação das mesmas. Após o estudo das ondaletas, o capítulo 3 descreve brevemente o *bootstrap* e algumas das adaptações do método para séries temporais. O capítulo 4 apresenta o *wavestrap*, união das duas técnicas citadas anteriormente.

O capítulo 5 apresenta comparações entre três adaptações de *bootstrap* para séries temporais: o *bootstrap* em blocos, o *bootstrap* em janelas e o *wavestrap*. As comparações se dão por meio do cálculo de intervalos de confiança para a primeira autocorrelação de processos autorregressivos de ordem um.

Capítulo 2

Ondaletas

A palavra "Ondaleta" possui sua origem no termo francês *ondelette*, uma forma diminutiva de onda, diferente das "grandes ondas" (como o seno ou o cosseno), as ondaletas oscilam apenas em um pequeno intervalo ao redor do 0. A figura abaixo 2.1 apresenta dois exemplos de ondaletas:

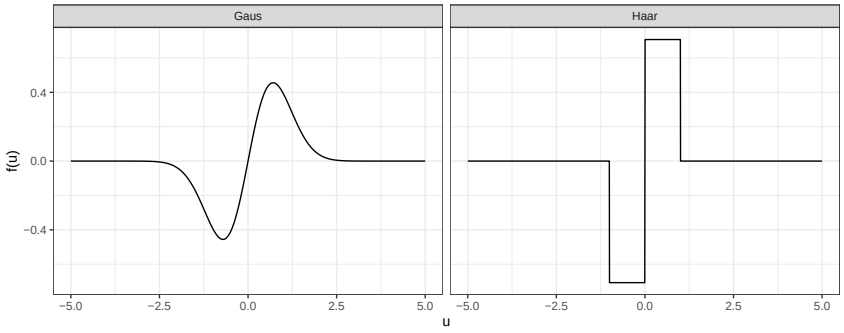


Figura 2.1: Exemplos de ondaletas: a esquerda a ondaleta de Gauss, a direita a ondaleta de Haar

Essas funções são utilizadas para gerar bases ortonormais em L_2 com propriedades interessantes. L_2 é o espaço de todas as funções quadrado-integráveis, ou seja:

$$L_2 = \left\{ f : \mathbb{R} \mapsto \mathbb{R} : \int_{-\infty}^{\infty} f(x)^2 dx < +\infty \right\}.$$

Neste trabalho serão consideradas apenas funções reais, no entanto, os resultados desenvolvidos podem ser estendidos para funções complexas.

Antes de entrar em detalhes na construção das ondaletas, é conveniente que o leitor esteja familiarizado com os principais conceitos de análise de séries temporais.

2.1 Séries temporais

Série temporal é uma realização de um processo estocástico. Existem várias séries temporais utilizadas no cotidiano, como a temperatura medida em uma cidade diariamente, ou o preço de uma determinada ação na bolsa de valores. MORETTIN e TOLOI (2018) define processo estocástico:

Definição 2.1 (Processo estocástico) *Seja T um conjunto arbitrário e totalmente ordenado e seja S um conjunto de estados possíveis. Um processo estocástico é uma família de variáveis aleatórias $Z = \{Z(t) : t \in T\}$.*

Um exemplo de processo estocástico é o passeio aleatório simples:

Exemplo 2.2 (Passeio Aleatório Simples) *Seja um objeto em repouso no ponto zero. Suponha que a cada minuto esse objeto pode tomar duas ações: ele pode mover-se uma unidade para direita ou uma unidade para a esquerda. Considere a variável aleatória $Z(t)$: o ponto em que o objeto se encontra no instante t, neste caso:*

- *S é o conjunto dos inteiros.*
- *T é o conjunto dos números naturais (medimos a posição do objeto a partir do minuto zero);*

E $Z(t)$ é um processo estocástico.

Comumente, a análise de séries temporais consiste na modelagem do processo estocástico $Z(t)$ gerador da série (lembre que a série consiste apenas na observação de $Z(t)$ para algum conjunto $\mathbf{t} \subset T$).

A principal peculiaridade na análise dos processos estocásticos (e das séries temporais) é que as variáveis aleatórias $Z(t)$ em geral são correlacionadas, muitas das técnicas utilizadas para analisar este tipo de fenômeno consistem na filtragem dessa correlação.

Para quantificar as correlações entre as variáveis aleatórias, são utilizadas a autocovariância e a autocorrelação:

Definição 2.3 (Autocovariância) *Seja $Z(t)$ um processo estocástico e sejam t_1, $t_2 \in T$. A função de autocovariância $\gamma(t_1, t_2)$ de Z é dada por:*

$$\gamma(t_1, t_2) = E[Z(t_1)Z(t_2)] - E[Z(t_1)]E[Z(t_2)].$$

Definição 2.4 (Autocorrelação) *Seja $Z(t)$ um processo estocástico e sejam t_1, $t_2 \in T$. A função de autocorrelação $\rho(t_1, t_2)$ de Z é dada por:*

$$\rho(t_1, t_2) = \frac{\gamma(t_1, t_2)}{\sqrt{\gamma(t_1, t_1)\gamma(t_2, t_2)}}.$$

Para uma série temporal qualquer, as funções de autocorrelação e autocovariância podem se tornar muito complexas, por isso, em geral restringimos nossos estudos a processos estacionários:

Definição 2.5 (Processo estacionário) *Um processo $Z(t)$ é dito estacionário se suas características são invariantes no tempo, ou seja:*

$$F_Z(t) = F_Z(t + h),$$

em que $F_Z(T)$ é a função de distribuição acumulada do processo $Z(t)$.

Assumir que um processo estocástico é estacionário é uma suposição forte que em geral não é atendida, para relaxar um pouco essa suposição, introduzimos a noção de estacionariedade fraca:

Definição 2.6 (Processo fracamente estacionário) *Um processo $Z(t)$ é dito fracamente estacionário se:*

- $E[Z(t)] = \mu \quad \forall \quad t \in T$ *e*
- $Var[Z(t)] = \sigma^2 \quad \forall \quad t \in T$.

É interessante notar que no caso de séries fracamente estacionárias, $\gamma(t_1, t_2)$ e $\rho(t_1, t_2)$ dependem de t_1 e t_2 apenas pela sua diferença $h = t_2 - t_1$. Quando for este o caso, escreveremos apenas $\gamma(h)$ e $\rho(h)$. Além disso, as autocovariâncias e autocorrelações de processos estacionários possuem as seguintes propriedades:

- $\gamma(h) > 0$, além disso $\rho(0) = 1$;
- $\gamma(-h) = \gamma(h)$;
- $|\rho(h)| \leq 1$;

2.1.1 Transformações de séries temporais

Os processos estocásticos (e as séries observadas) estacionários possuem várias propriedades interessantes que facilitam seu estudo, no entanto, não é incomum encontrar processos não estacionários, para contornar estes problemas, existem algumas técnicas capazes de transformar as séries temporais em séries (fracamente) estacionárias.

A primeira técnica é conhecida como diferenciação, quando $E[Z(t)]$ é da forma $E[Z(t)] = \mu_0 + \mu t$, então o processo $Z^*(t) = Z(t) - Z(t-1)$ possui esperança μ que não depende de t. A figura 2.2 abaixo apresenta o efeito da diferenciação em uma série temporal simulada:

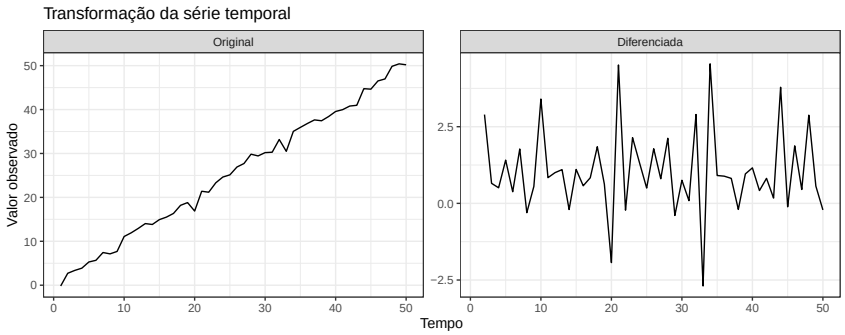


Figura 2.2: Exemplo de série temporal estabilizada pela diferenciação: a esquerda a série original e a direita a série diferenciada

Note como no primeiro gráfico a série temporal apresenta uma tendência crescente, enquanto no segundo a série oscila em torno de um valor constante.

Também é comum encontrar séries em que a variância muda ao longo do tempo, nesses casos, utilizar $Z^*(t) = log(Z(t))$ pode estabilizar a variância, a figura 2.3 abaixo apresenta a transformação logarítmica aplicada na série temporal "AirPassengers", esta série mede a quantidade de passageiros internacionais de avião por mês entre 1949 e 1960 (WOODWARD *et al.*, 2017).

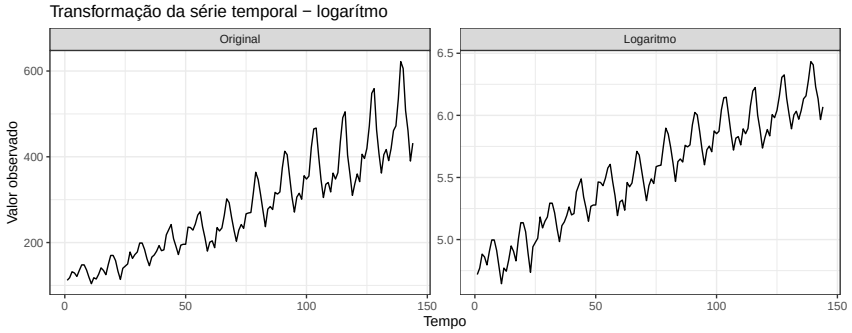


Figura 2.3: Exemplo de série temporal estabilizada pelo logaritmo

Uma vez que o logaritmo é uma função estritamente crescente, tomar o logaritmo da

série não faz com que a variância seja exatamente igual ao longo do tempo, no entanto, faz com que as diferenças nas variâncias sejam mais sutis, nos permitindo supor que a série é estacionária.

Existem muitas outras técnicas que podem ser utilizadas para obter os mesmos resultados (ex: transformação de Box-Cox). O leitor pode conferir algumas dessas técnicas no terceiro capítulo de HYNDMAN e ATHANASOPOULOS (2018)

2.2 Ondaletas

As ondaletas são uma ferramenta matemática com aplicações em várias áreas, como a física e computação, por exemplo. Uma possível aplicação para ondaletas é a análise de séries temporais.

Essas técnicas foram introduzidas no início do século XX. A primeira ondaleta foi proposta por Haar, por volta de 1910. Apesar disso, esse campo só viu sua popularidade crescer na década de 80, com o desenvolvimento das ondaletas de suporte compacto em DAUBECHIES (1988).

Em suma, as ondaletas são caracterizadas por duas funções ($\varphi(\cdot)$ e $\psi(\cdot)$) que quanto transladadas e dilatadas formam uma base ortonormal para L_2. Utilizando essa base, qualquer função $f \in L_2$ pode ser escrita como:

$$f(\cdot) = \sum_k \alpha_{0k}\varphi_{0k}(\cdot) + \sum_{j=0}^{\infty}\sum_k \beta_{jk}\psi_{jk}(\cdot).$$

Na prática, nosso interesse na análise de uma função está em calcular os coeficientes α_{0k} e β_{jk} e interpretar seus significados. A seguir introduziremos mais formalmente o conceito de ondaletas utilizando a ondaleta de Haar.

2.2.1 Ondaleta de Haar

Esta subseção foi desenvolvida com base em HÄRDLE *et al.* (2012). Antes de introduzir a ondaleta de Haar, é necessário definir alguns conceitos que serão utilizados ao longo deste estudo:

Definição 2.7 (Produto interno) *Sejam f e g duas funções quaisquer, o produto in-*

terno $\langle f, g\rangle$ *é definido por:*

$$\langle f, g\rangle = \int_{-\infty}^{+\infty} f(x)g(x)dx.$$

Se $\langle f, g\rangle = 0$, dizemos que f e g são ortogonais.

Definição 2.8 (Sistema ortonormal) *Seja um sistema de funções* $\{f : \mathbb{R} \mapsto \mathbb{R}\}$*, este sistema é dito ortornomal se, e somente se:*

$$\langle f_k, f_{k'}\rangle = \begin{cases} 1, & se\ k = k', \\ 0, & c.c. \end{cases}$$

Definição 2.9 (Base) *Seja* V *um subespaço de* L_2 *e seja* $\{f_k\}$ *um sistema de funções, se:*

1. *Para todo* $g \in V$*, existe um conjunto* $\{c_k : c_k \in \mathbb{R}\}$ *tal que:*

$$g(\cdot) = \sum_{k=-\infty}^{\infty} c_k f_k(\cdot).$$

2. $\{f_k\}$ *é linearmente independente, ou seja:* $\langle f_k, f_{k'}\rangle = 0$ *se* $k \neq k'$

Então $\{f_k\}$ *é base de* V*. Dizemos também que* V *é gerado por* $\{f_k\}$*.*

Considere agora a função:

$$\varphi(x) = \begin{cases} 1, & se\ x \in (0, 1], \\ 0, & c.c. \end{cases}$$

Considere ainda o sistema de funções $\{\varphi_{0k} : \varphi_{0k}(x) = \varphi(x - k)\}$, $k \in \mathbb{Z}$. Note que este é um sistema ortonormal de funções ($\int_{-\infty}^{\infty} \varphi_{0k}(x)\varphi_{0k'}(x)dx = 1$ somente se $k = k'$). O espaço:

$$V_0 = \{f \in L_2 : f \text{ é constante em } (k, k+1], k \in \mathbb{Z}\}.$$

é gerado por este sistema. Considere agora o espaço:

$$V_1 = \left\{f \in L_2 : f \text{ é constante em } \left(\frac{k}{2}, \frac{k+1}{2}\right], k \in \mathbb{Z}\right\}.$$

Outra maneira de se escrever esse conjunto seria:

$$V_1 = \{f \in L_2 : f(x) = h(2x), h \in V_0\}.$$

Uma base para V_1 é dada por: $\left\{\varphi_{1k} : \varphi_{1k}(x) = 2^{1/2}\varphi(2^1 x - k)\right\}$. Também é fácil notar que $V_0 \subset V_1$ (toda função que for constante em $(k, k+1]$ é constante em: $(2k, 2k+1/2]$).

Seguindo a mesma lógica, é possível construir espaços

$$V_j = \left\{f \in L_2 : f(x) = h(2^j x), h \in V_0\right\}$$

com bases $\{\varphi_{jk}\}$ indefinidamente. Além disso, os espaços são encaixados (isto é: $V_j \subset V_{j+1}$).

Continuando indefinidamente a geração de espaços V_j nos aproximamos de L_2, pois qualquer função pode ser aproximada por somas de indicadoras nos intervalos $\left(\frac{k}{2^j}, \frac{k+1}{2^j}\right]$ (conforme j cresce, os intervalos apresentam amplitude cada vez menor).

Sendo assim, o sistema $\left\{\varphi_{jk} : \varphi_{jk}(x) = 2^{j/2}\varphi(2^j x - k)\right\}$, $j, j \in \mathbb{Z}$ gera L_2, no entanto, este sistema não é ortogonal (note que $\langle\varphi_{00}, \varphi_{10}\rangle = 1/2$);

Denote agora por W_0 o complemento ortogonal de V_0 em V_1, ou seja:

$$W_0 = V_1 \ominus V_0.$$

Se encontrarmos uma base para W_0, então poderemos escrever qualquer elemento em V_1 como uma soma de elementos em V_0 e elementos em W_0. Considere a função:

$$\psi(x) = \begin{cases} -1, & x \in (0, 1/2] \\ 1, & x \in (1/2, 1] \\ 0, & c.c. \end{cases}$$

Proposição 2.10 *O sistema $\{\psi_{0k}\}$ é uma base de W_0.*

Prova.

1. *O sistema é ortogonal: os suportes de ψ_{0k} não se sobrepõem;*
2. *O sistema é ortogonal a $\{\varphi_{0k}\}$: quando os elementos de φ e ψ possuem o mesmo suporte, o produto interno é nulo (ex: $\int_0^1 \varphi_{01}(x)\psi_{01}(x)dx = 0$);*

3. *Toda $f \in V_1$ possui uma representação única no sistema $\{\varphi_{0k}, \psi_{0k}\}$:*

   Se $f = \sum_{k=-\infty}^{\infty} c_k \phi_{1k}$, então basta provar que ϕ_{1k} é uma combinação linear de φ_{0k} e ψ_{0k}. De fato:

$$\varphi_{1k} = \frac{2^{j/2}}{2} \left(\varphi_{0k} - \psi_{0k}\right).$$

■

Com os sistemas $\{\varphi_{0k}\}$ e $\{\psi_{0k}\}$ temos uma base para V_1. O sistema $\{\psi_{1k}\}$ ã uma base para W_1, sendo assim, podemos escrever V_2 como:

$$V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1.$$

Seguindo este processo indefinidamente, podemos ir encaixando os espaços até termos uma base para L_2:

$$L_2 = V_0 \bigoplus_{j=0}^{\infty} W_j$$

Ou seja, cada $f \in L_2$ pode ser escrita como:

$$f(x) = \sum_k \alpha_{0k} \varphi_{0k}(x) + \sum_{j=0}^{\infty} \sum_k \beta_{jk} \psi(x).$$

A função φ é chamada de ondaleta pai, já a função ψ é chamada de ondaleta mãe. Esse é apenas um exemplo de base para L_2 com base na ondaleta de Haar. A seguir veremos como se dá a construção de uma ondaleta.

2.2.2 Construção de uma ondaleta

A ondaleta de Haar apresentat forma analítica relativamente simples (tanto quanto funções descontínuas podem ser) e interpretabilidade física intuitiva (podemos enxergar as integrais de φ e ψ como médias e diferenças de médias que se deslocam e dilatam no tempo). Ao longo do século XX, inúmeras ondaletas foram desenvolvidas para atender a diferentes requisitos.

Como visto na seção anterior, o processo de construção de ondaletas envolve encontrar os espaços V_j gerados pelos sistemas $\{\varphi_{jk}, k \in \mathbb{Z}\}$. Além disso, é necessário encontrar a ondaleta mãe ψ cujos sistemas $\{\psi_{jk}, k \in \mathbb{Z}\}$ gerem os espaços W_j. Para essas duas funções caracterizarem uma base de L_2 e poderem ser chamadas de ondaletas existem as seguintes

condições:

- $V_j \subset V_{j+1}$, ou seja, os espaços são encaixados
- $\bigcup_{i=0}^{\infty} V_j = L_2$, isto é, a união dos espaços V_j é densa em L_2.
- $W_j = V_{j+1} \ominus V_j$, ou seja, W_j é o complemento ortogonal de V_j em V_{j+1}.
- Os sistemas $\{\varphi_{jk}, k \in \mathbb{Z}\}$ e $\{\psi_{jk}, k \in \mathbb{Z}\}$ são ortonormais.

Ainda utilizando como base HÄRDLE *et al.* (2012), introduziremos agora resultados que facilitam a verificação das propriedades descritas acima. A seguir, a Coiflet será introduzida para exemplificar a aplicação dos teoremas.

Os lemas à seguir utilizam resultados da teoria de Fourier, portanto, é conveniente definir a CFT de uma função:

Definição 2.11 (Transformada contínua de Fourier (CFT)) *Seja f uma função qualquer, a função $\hat{f}(\xi)$:*

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} e^{-ix\xi} f(x) dx,$$

é chamada de Transformada contínua de Fourier (CFT) de f. A transformação inversa é dada por:

$$f(x) = \int_{-\infty}^{\infty} e^{ix\xi} \hat{f}(\xi) d\xi.$$

A Transformada contínua de Fourier possui as seguintes propriedades:

- $||f||_2^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} |f(\xi)|^2 d\xi$;
- $\langle f, g \rangle = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\xi)\overline{\hat{g}(\xi)} d\xi$;
- $[\hat{f}(x-k)](\xi) = e^{-ik\xi}\hat{f}(\xi)$;
- $[\hat{f}(ax)](\xi) = \frac{1}{a}\hat{f}(\frac{\xi}{a})$;
- Seja $h = f * g$ a convolução de f e g (ou seja, $h(x) = \int f(x-t)g(t)dt$), então: $\hat{h} = \hat{f}\hat{g}$;

A primeira condição para que uma função φ seja uma ondaleta pai é que o sistema $\{\varphi_{jk}, k \in \mathbb{Z}\}$ seja ortornomal:

Resultado 2.12 *Seja $\varphi \in L_2$ uma função real. O sistema $\{\varphi_{0k}, k \in \mathbb{Z}\}$ é ortonormal se, e apenas se:*

$$\sum_k |\hat{\varphi}(\xi + 2\pi k)|^2 = 1.$$

não provaremos este resultado (assim como outros). A segunda condição para a existência das ondaletas é que os espaços V_j sejam encaixados:

Resultado 2.13 *Os espaços V_j são encaixados ($V_j \subset V_{j+1}$) se, e somente se, existe uma função periódica $m_0(\xi)$ tal que:*

$$\hat{\varphi}(\xi) = m_0\left(\frac{\xi}{2}\right)\hat{\varphi}\left(\frac{\xi}{2}\right).$$

Satisfeitos os resultados 2.12 e 2.13, então $\bigcup_{i=0}^{\infty} V_j = L_2$. A prova deste resultado se encontra no capítulo 8 de HÄRDLE *et al.* (2012). Encontrando uma função φ que atenda a todos estes requisitos, a ondaleta mãe pode ser encontrada utilizando o seguinte resultado:

Resultado 2.14 *Seja φ uma ondaleta pai (isto é, uma função que satisfaça 2.12 e 2.13) e seja m_0 uma solução de 2.13. Seja ainda:*

$$\hat{\psi} = m_1\left(\frac{\xi}{2}\right)\hat{\varphi}\left(\frac{\xi}{2}\right),$$

em que $m_1 = \overline{m_0(\xi + \pi)}e^{-i\xi}$. Então a função ψ dada pela transformação inversa de $\hat{\psi}$ é uma ondaleta mãe.

Com os resultados 2.12 e 2.13 é possível verificar se uma determinada função φ é uma ondaleta pai, caso a função o seja, o resultado 2.14 nos diz como podemos encontrar ondaletas mãe (no plural, pois pode existir mais de uma função m_0 que satisfaça 2.13).

Apesar de apresentadas as condições necessárias para a criação de ondaletas, ainda não está claro qual procedimento pode ser utilizado para essa construção. O algoritmo à seguir apresenta um passo a passo que pode ser seguido para a criação de uma ondaleta:

1. Escolha uma função m_0 tal que:

   - $|m_0(\xi)|^2 + |m_0(\xi + \pi)|^2 = 1$;
   - m_0 possui período 2π;

- $m_0 \in L_2(0, 2\pi)$;
- $m_0(0) = 1$;

2. Conclua através de 2.13 que:

$$\hat{\varphi} = \prod_{j=1}^{\infty} m_0\left(\frac{\xi}{2^j}\right).$$

Na prática as escolhas de m_0 envolvem o polinômio trigonométrico (HÄRDLE *et al.*, 2012), ou seja: $m_0(\xi) = \frac{1}{\sqrt{2}} \sum_{k=N_0}^{N_1} h_k e^{-ik\xi}$. Nessas condições o produtório definido acima é finito e o suporte da ondaleta é compacto. Além disso, a condição $|m_0(\xi)|^2 + |m_0(\xi + \pi)|^2 = 1$ garante que $\hat{\varphi}$ é a transformada de Fourier de uma função $\varphi \in L_2$.

Sendo assim, é possível criar uma (ou mais) ondaletas especificando apenas os coeficientes h_k. Em geral, esses coeficientes são escolhidos tal que:

- A ondaleta pai (e/ou a mãe) possui N de momentos nulos[1].
- A ondaleta pai (e/ou a mãe) possui N derivadas contínuas.

Sendo assim, na prática o que faremos é escolher os coeficientes h_k de modo que as ondaletas possuam propriedades desejáveis. Nota-se que existem outras maneiras de se construir bases ondaleta, como descrito em MALLAT (1999).

2.2.3 Coiflet

A seguir aplicaremos os resultados da seção anterior para o desenvolvimento das Coiflets. As Coiflets foram desenvolvidas por Daubechies a pedido de Ronald Coifman. Assim como a ondaleta de Haar, as Coiflets também possuem suporte compacto. Outra característica interessante da Coiflet é que tanto a ondaleta pai quanto a ondaleta mãe possuem N momentos nulos.

A construção dessa família de bases pode ser feita utilizando o procedimento descrito anteriormente. Seja $N \in \mathbb{N}$. Considere a função:

$$m_0(\xi) = \left(\frac{1 + e^{-i\xi}}{2}\right)^N \sum_k h_k e^{ik\xi}.$$

[1]Dizer que uma função f possui N momentos nulos equivale a dizer que as $N-1$ primeiras derivadas de sua CFT são iguais a 0 no ponto 0 ($\hat{f}^{(p)}(0) = 0$, para $p \in \{1, \ldots, N-1\}$).

Como descrito anteriormente, as seguintes condições devem ser satisfeitas:

- $\int_{-\infty}^{\infty} \varphi(x)dx = 1$ para que o sistema seja ortonormal;
- $\int_{-\infty}^{\infty} \varphi(x)x^l = 0, 1 \leq l \leq N$ para que o momento l seja nulo;
- $\int_{-\infty}^{\infty} \psi(x)x^l = 0, 1 \leq l \leq N$ para que o momento l da mãe seja nulo.

Essas condições são equivalentes a (HÄRDLE *et al.*, 2012):

- $\hat{\varphi}(0) = 1$;
- $\hat{\varphi}^{(l)}(0) = 0$;
- $\hat{\psi}^{(l)}(0) = 0$;

Escolha agora um inteiro $K \geq 1$ tal que $N = 2K$. As restrições descritas anteriormente impõe que m_0 seja da forma:

$$m_0(\xi) = \left(\frac{1 + e^{-i\xi}}{2}\right)^{2K} \sum_{k=0}^{K-1} C_{K-1+k}^{k} \left(\sin^2 \frac{\xi}{2}\right)^k + \left(\sin^2 \frac{\xi}{2}\right)^K F(\xi),$$

em que $F(\xi)$ é um polinômio trigonométrico.

Encontrar as funções φ e ψ com base nessa função m_0 não é um processo simples e muitas vezes não possui solução analítica. Por conta disso, é comum que as implementações do método utilizem aproximações numéricas para o cálculo dos coeficientes de ondaleta.

2.3 Exemplo de análise com ondaletas

Até agora descrevemos o que são ondaletas, qual é a sua origem e como são construídas. Nesta seção realizaremos a análise de uma série temporal utilizando as duas ondaletas apresentadas anteriormente (ondaleta de Haar e Coiflet). Utilizaremos esse exemplo para introduzir o conceito de energia na análise de ondaletas.

Até o momento a teoria desenvolvida considerou os sinais como funções pertencentes a L_2. A partir de agora consideraremos sequências de observações de alguma Variável Aleatória (ou seja, séries temporais), a teoria e as interpretações para o cálculo de transformadas de ondaletas no mundo discreto são análogas ao que foi visto para o mundo contínuo.

2.3.1 Implementação

Todas as transformadas de ondaleta utilizadas neste trabalho foram calculadas através da biblioteca *PyWavelets* disponível para *python* (LEE *et al.*, 2019). Este pacote utiliza o algoritmo piramidal introduzido por MALLAT (2009), permitindo que as transformações sejam calculadas rapidamente.

No que diz respeito a DWT (*Discrete Wavelet Transform*), o pacote possui as seguintes funcionalidades:

1. Quando o número de observações da série temporal for diferente de 2^J ($J \in \mathbb{N}$), ela será estendido para ter o comprimento necessário.

2. A extensão da série temporal pode se dar das seguintes maneiras:

   (a) Circular: Os últimos elementos da série são inseridos antes do início e os elementos iniciais são inseridos após o final.

   (b) Zero: As observações da série fora de $\mathbf{t}$ (conjunto dos instantes observados) são iguais a 0.

   (c) Constante: Os valores anteriores ao início são iguais ao primeiro valor observado, os valores após o final são iguais ao último.

   (d) Simétrico: Os valores iniciais são inseridos antes do começo da série (em ordem inversa) e os valores finais são inseridos após o término (também em ordem inversa).

   (e) Espelhado: Semelhante ao simétrico, mas o primeiro e o último elementos não são utilizados na extensão da série.

   (f) Anti-simétrico: Semelhante ao simétrico, mas os valores são multiplicados por -1 na extensão.

   (g) Tendência: uma reta é calculada com base nos primeiros (e nos últimos) valores da série.

3. Para as ondaletas sem forma analítica fechada, aproximações são armazenadas nas bases de dados do pacote (como é o caso da Coiflet).

2.3.2 Identificação de trapaça em jogos de corrida

Trackmania™ é uma franquia de jogos competitivos de corrida distribuída por um estúdio francês. No modo de jogo mais comum, o objetivo dos jogadores é chegar à linha de chegada no menor tempo possível, ou seja, trata-se de um jogo de competição contra o tempo (*time attack*).

Cada edição do jogo conta com algumas pistas oficiais (criada pelo estúdio) e mais inúmeras pistas criadas pela comunidade de jogadores. A competição pelos melhores tempos nessas pistas é bastante acirrada, e manter o recorde mundial em uma das pistas oficiais garante ao jogador bastante prestígio na comunidade.

O ranqueamento dos tempos é mediado através do site *Trackmania Exchange*[2]. Após dirigir em uma pista, o jogador tem a possibilidade de salvar o *replay* (automaticamente dentro do jogo) e submeter o arquivo ao site, após aprovação da equipe do site o tempo é contabilizado para o *ranking* e o arquivo é disponibilizado.

Recentemente descobriu-se que alguns jogadores utilizavam de técnicas ilegais para gravar seus *replays*. Por conta de uma falha de segurança, é possível dirigir com o jogo em câmera lenta. Quando a velocidade do jogo é reduzida, a contagem do tempo também o é. Os jogadores em questão estavam se aproveitando dessa brecha para dirigir mais precisamente.

No arquivo de *replay* são armazenadas as entradas dos jogadores, uma vez que a física dentro do jogo é completamente determinística, isso permite a reprodução perfeita da corrida com base apenas nas entradas gravadas.

Essas entradas são constituídas de três sinais:

1. Aceleração: Valor em $\{0, 1\}$ que indica se o carro está acelerando em um determinado instante.

2. Frenagem: Valor em $\{0, 1\}$ que indica se o carro está freando em um determinado instante.

3. Esterço: Valor contínuo em $[-1, 1]$ que indica se o carro está virando; -1 representa o máximo possível de esterço para a esquerda, enquanto 1 representa o máximo possível de esterço para direita (o valor 0 indica que o carro está andando para frente).

[2] `https://www.tm-exchange.com/`

Na nossa análise, consideraremos apenas o esterço. Os sinais são captados pelo jogo a cada $10ms$ (milissegundos) desde o momento inicial da corrida até o momento em que o carro cruza a linha de chegada. Ao invés de considerar a série original, aplicaremos a diferenciação, ou seja, X_t é igual a diferença entre o esterço no instante t e no instante $t - 0,01s$.

Quando um jogador dirige com o jogo em câmera lenta, ele possui mais tempo de reação e consegue dirigir com mais precisão, isso pode ser analisado através de suas entradas, conforme mostrado na figura 2.4 abaixo. Nessa figura, temos as entradas de um *replay* suspeito e outro dirigido de maneira legítima. O recorde mundial (obtido pelo jogador suspeito) é $10ms$ mais rápido que o segundo lugar:

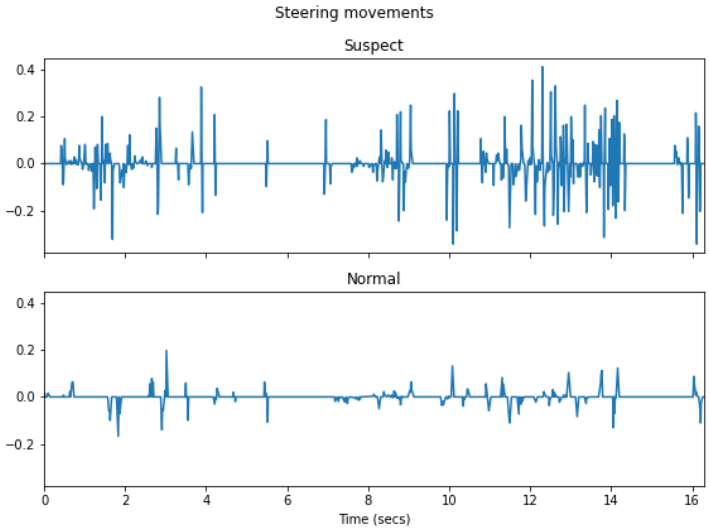
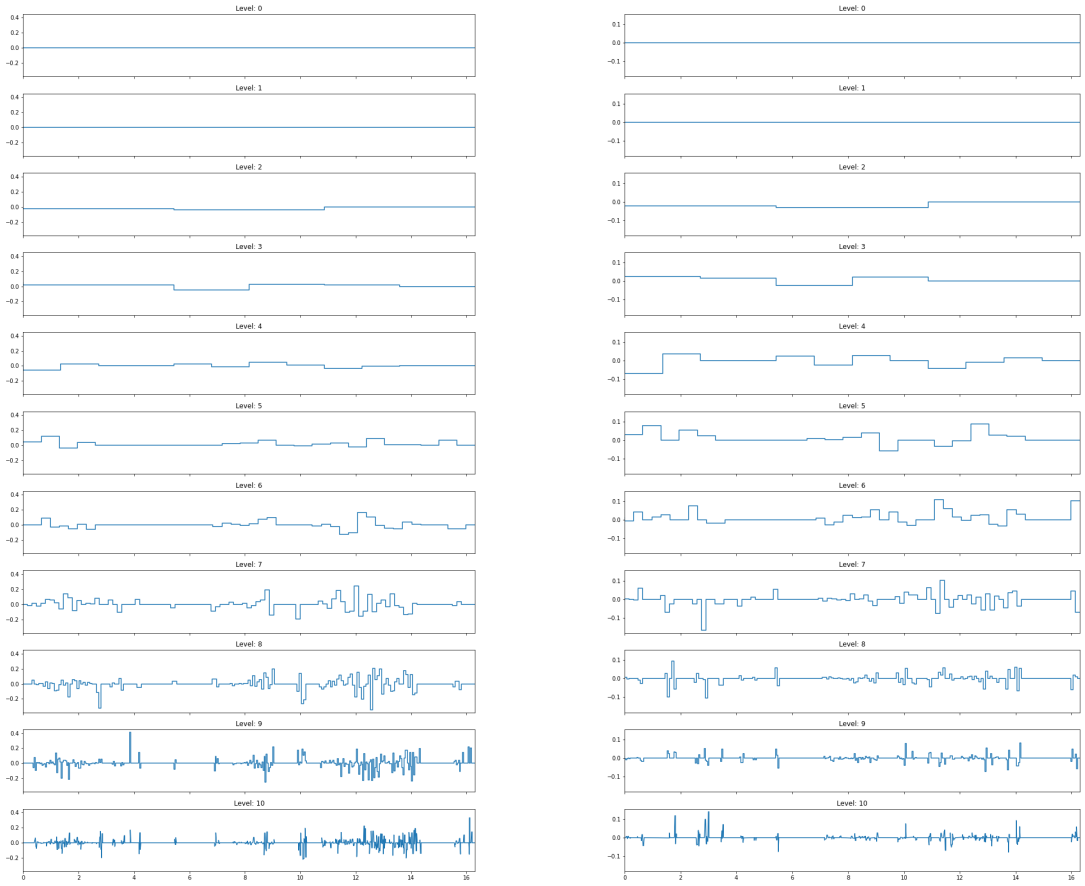


Figura 2.4: Comparação entre as entradas de dois jogadores para uma determinada pista

Analisando-se a figura, podemos notar que: (1) os sinais não são estacionários (uma vez que a variância do esterço é maior nas curvas do que nas retas); (2) o jogador suspeito de trapaça possui maior variação no esterço ao longo da corrida.

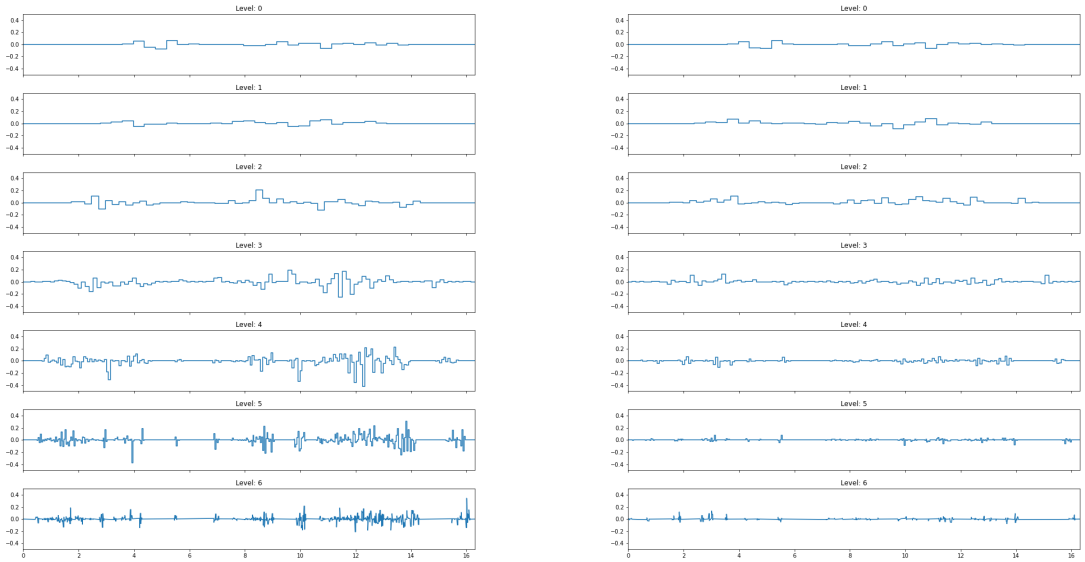
Para comparar os dois sinais, utilizaremos as duas ondaletas introduzidas anteriormente: a ondaleta de Haar e a Coiflet com quatro momentos nulos (abreviada como "Coif4"). A figura 2.5 abaixo contém os coeficientes DWT da ondaleta de Haar para os dois sinais, enquanto a figura 2.6 contém os coeficientes DWT para a Coif4:



(a) Jogador suspeito.

(b) Jogador legítimo.

Figura 2.5: Coeficientes da DWT de Haar dos dois jogadores



(a) Jogador suspeito.

(b) Jogador legítimo.

Figura 2.6: Coeficientes da DWT de Coiflet(4) dos dois jogadores

De fato, analisando as figuras 2.5 e 2.6 é possível identificar que os coeficientes associados a altas frequências (os dois níveis mais altos para cada ondaleta) são maiores (em módulo) para o jogador suspeito. Os coeficientes de baixa frequência assumem valores parecidos, isso pode ser um indicador de que os carros adotam traçados parecidos na pista.

Comparando as duas ondaletas, é possível notar que a Coiflet possui uma representação mais esparsa dos sinais (os coeficientes se distribuem em menos níveis). Isso ocorre por conta do modo de extensão utilizado (os valores da série antes e depois da corrida foram considerados como sendo iguais a zero) e por conta do número momentos nulos escolhido (quatro). Além da esparsidade, a Coif4 parece representar melhor as diferenças entre os sinais nas frequências mais altas.

Uma possível maneira de quantificar a variação dos sinais em cada nível de frequência é através da decomposição de energia da ondaleta (PERCIVAL, 2000):

Definição 2.15 (Decomposição de energia de uma DWT) *Sejam β_{jk} os coeficientes de ondaleta associados a uma transformada de ondaleta discreta (DWT) de uma sequência. Então a quantidade:*

$$E_j = \sum_k \beta_{jk}^2$$

é chamada de energia da DWT associada ao nível j.

Notamos novamente que a distribuição dos níveis se dá de maneira diferente para as duas ondaletas (por isso observamos menos níveis na "Coif4"). A figura 2.7 apresenta a decomposição de energia dos dois sinais:

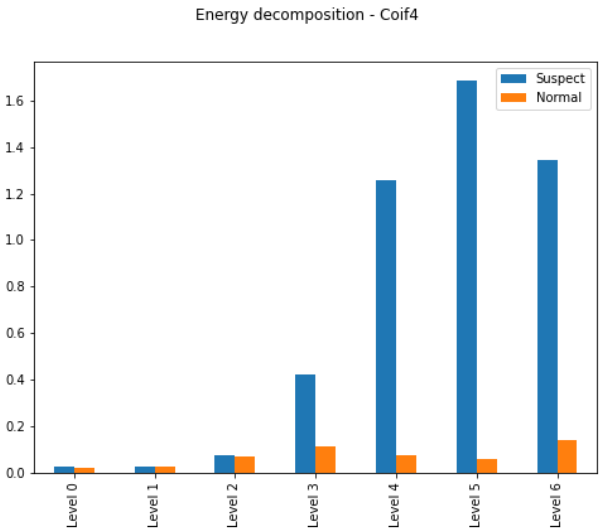


Figura 2.7: Decomposição de energia dos dois sinais utilizando a Coif4

Analisando a figura 2.7 fica ainda mais evidente a diferença entre os dois jogadores. Apesar de ambos os sinais apresentarem energias semelhantes para níveis mais baixos, a medida que aumentamos a frequência, as entradas do jogador suspeito apresentam muito mais energia.

Apesar de não ser possível analisar cada um dos milhares de *replays* submetidos, com base na análise de ondaletas é possível definir um critério de filtragem de *replays* suspeitos. Como vimos anteriormente, jogadores que utilizam técnicas ilegais para diminuir a velocidade do jogo apresentarão energias consideravelmente mais altas do que os competidores legítimos.

Capítulo 3

Bootstrap

É de consenso geral entre os estatísticos que estimativas pontuais são pouco informativas e, quando possível, utilizam-se estimativas intervalares. No entanto, as técnicas para calcular intervalos, seja para a estimação de um parâmetro ou para a previsão de uma observação, costumam ser complexas ou depender de suposições fortes (como a normalidade).

É nesse contexto que surgem, na segunda metade do século XX, técnicas de reamostragem. Uma das primeiras ideias nessa área é atribuída o *jackknife* (QUENOUILLE, 1956). Essa técnica consiste em recalcular estimativas para o parâmetro de interesse retirando-se um item da amostra por vez.

Com a expansão das capacidades computacionais, métodos mais intensos (computacionalmente) puderam ser desenvolvidos. Uma das técnicas mais populares é o *bootstrap*. Trata-se de uma técnica de reamostragem que pode ser utilizada tanto para realizar estimativas intervalares de parâmetros quanto para calcular previsões.

O *bootstrap* foi introduzido por EFRON (1979) e é um método computacional para estimar o erro padrão de um estimador de forma automática e livre de cálculos teóricos. Por conta dessa última característica, o *bootstrap* se faz disponível em cenários onde o estimador é muito complexo, ou ainda quando temos conjuntos de dados de tamanho reduzido.

Para introduzir o procedimento, é conveniente que definamos amostra *bootstrap*:

Definição 3.1 (Amostra *bootstrap* simples) *Seja* $\mathbf{x} = \{x_1, \ldots, x_n\}$ *uma amostra aleatária independente e identicamente distribuída de tamanho* n *de uma variável aleatória* X *(que pode ou não pertencer a um processo estocástico). Uma amostra bootstrap é uma amostra*

aleatória simples com reposição de x:

$$x^* = \{x_1^*, \ldots, x_n^*\},$$

em que o asterisco indica que essa não é a amostra original. Neste procedimento, cada $x_i, i \in \{1, \ldots, n\}$ tem probabilidade igual de ser escolhido como elemento x_i^* da amostra bootstrap.

Esta é uma das formas mais simples e intuitivas de se reamostrar de um conjunto $\mathbf{x}$ de observações. No entanto, não é a única possibilidade, existem inúmeras maneiras diferentes (adaptadas para casos específicos) de retirar as amostras.

Após definir um plano de reamostragem, o procedimento *bootstrap* consiste em retirar-se um número arbitrário B de amostras do conjunto de dados original e refazer os cálculos de interesse nessas novas amostras, analisando os resultados em cada caso.

No caso de estimação de um parâmetro, por exemplo, podemos estimar esse parâmetro B vezes e calcular o desvio padrão dessas B estimativas, com isso podemos ter uma ideia de como o estimador se comporta nessa população. Na seção seguinte abordaremos alguns procedimentos diferentes de reamostragem possíveis.

3.1 Procedimentos para reamostragem *bootstrap*

Muitas vezes, a amostra *bootstrap* simples não atende aos objetivos propostos. Em séries temporais, por exemplo, não podemos realizar este procedimento, uma vez que os elementos da amostra observada $\mathbf{z} = \{z_t, t \in \mathbf{t}\}$ em geral não são independentes.

Para contornar este problema podemos utilizar diferentes métodos de reamostragem, que serão enumerados nessa seção:

1. Amostra *bootstrap* simples: Já definido anteriormente.

2. Amostra *bootstrap* paramétrica: Ao invés de retirarmos uma amostra de $\mathbf{x}$, retiramos amostras seguindo uma função de distribuição acumulada F especificada previamente.

3. Amostra em blocos (k): Dividimos a amostra temporal em blocos de tamanho k ($(z_t, z_{t+1}, \ldots, z_{t+k}), t \in \mathbf{t}_{-(k-1)}$[1]) e compomos uma nova amostra com base nesses

[1] $\mathbf{t}_{-k}$ é o conjunto $\mathbf{t}$ sem as últimas k observações.

blocos. Para séries temporais, a amostra simples é igual a amostra em blocos com $k = 1$.

4. Amostra em janelas: semelhante a amostra em blocos, mas os blocos não se sobrepõem.

5. Amostra sobre resíduos: Primeiro realizamos algum procedimento de modelagem sobre os dados, depois retiramos amostras sobre os resíduos calculados.

Considerando essas (e outras) possibilidades, é importante especificar corretamente qual é o procedimento que está sendo utilizado. Além disso, é importante também identificar qual é o objetivo da reamostragem, pois adaptações no procedimento podem se fazer necessárias.

Por conta dessa necessidade, definiremos no próximo capítulo o *wavestrap*.

Capítulo 4

Wavestrap

Vimos até agora como se dá a construção das ondaletas. Vimos também como podemos conduzir alguns procedimentos *bootstrap* para gerar estimativas (ou previsões) intervalares. Uma suposição para realizar o *bootstrap* é que a amostra seja não correlacionada, o que em geral não acontece quando trabalhamos com séries temporais.

As séries temporais podem apresentar dois tipos de correlação (FENG *et al.*, 2005):

1. Dependência Curta: A autocorrelação da série decai rapidamente para zero (a uma taxa exponencial).

2. Dependência Longa: A autocorrelação da série diminui devagar para zero (a uma taxa hiperbólica).

No primeiro caso, observações distantes no tempo[1] são não correlacionadas, quando isso ocorrer, diremos que a série possui memória curta. Já no segundo caso, o decrescimento devagar das autocorrelações faz com que haja uma relação de dependência entre observações distantes no tempo, neste caso, diremos que a série possui memória longa.

Quando montamos um esquema de reamostragem, queremos que os elementos da amostra sejam não correlacionados. Por isso foram desenvolvidos os métodos *bootstrap* adaptados para séries temporais. No entanto, para aplicar um *bootstrap* em janelas de tamanho k devemos assumir que todas correlações de ordem maior que k são nulas, o que não ocorre quando a(s) série(s) apresentam memória longa.

Uma possível alternativa para contornar este problema é o *wavestrap*, este procedimento pode ser considerado uma espécie de *bootstrap* sobre resíduos, uma vez que apli-

[1]Em geral, consideraremos um par de observações como distantes se a diferença de seus índices na amostra seja maior ou igual a 10, mas isso depende da aplicação.

caremos um procedimento anterior à reamostragem. A diferença é que aplicaremos a reamostragem nos coeficientes de ondaleta calculados separadamente em cada nível.

Esse procedimento pode ser realizado pois sob algumas condições os coeficientes de ondaleta são não correlacionados (EVARISTO, 2010). As condições necessárias e suficientes para que os coeficientes sejam não correlacionados ainda são estudadas atualmente e mudam dependendo da base escolhida.

Isso implica que o praticante deve verificar se a suposição de autocorrelação nula dos coeficientes de ondaleta é válida manualmente. Isso comumente pode ser feito através da função de autocorrelação (apresentada no Capítulo 2).

Mesmo quando os coeficientes são de fato não correlacionados, a técnica não é livre de imperfeições; TANG *et al.* (2008) mostra que intervalos de confiança os coeficientes de uma regressão linear (temporal) possuem tamanho reduzido. Uma das possíveis causas é que nos níveis menores (para os quais temos poucos coeficientes DWT) não é possível gerar amostras representativas (mesmo utilizando o *bootstrap* paramétrico).

Capítulo 5

Simulações

Para ilustrar uma possível aplicação das técnicas de *bootstrap* para séries temporais, utilizaremos simulações teóricas de processos autorregressivos de ordem um. Trata-se de um caso particular dos modelos ARIMA (BOX e JENKINS, 1970).

Definição 5.1 (Processo autorregressivo de ordem um) *Um processo autorregressivo de ordem um (AR ou AR(1)) é descrito pela seguinte equação:*

$$Z(t) = \phi Z(t-1) + \varepsilon_t,$$

em que ε_t são variáveis aleatórias independentes e identicamente distribuídas.

Neste trabalho, consideraremos que $\varepsilon_t \sim N(0,1)$, apesar de ser comum assumir que ε_t tenha distribuição normal, outras distribuições podem ser utilizadas.

Para os processos desse tipo, a função de autocorrelação é igual a:

$$\rho(h) = \phi^h.$$

(Fato que deixaremos sem prova), por conta de seu decaimento exponencial essa função de autocorrelação caracteriza um processo de memória curta.

5.1 Procedimentos de simulação

Nosso interesse será em estimar a primeira autocorrelação de processos autorregressivos ($\rho(1)$). Mesmo para modelos relativamente simples (como é o caso do AR(1)), determinar a distribuição teórica de estimadores é um trabalho árduo (e algumas vezes impossível).

Por conta disso, não consideraremos métodos teóricos de se calcular intervalos de confiança para a primeira autocorrelação. Em geral, utiliza-se uma aproximação assintótica desenvolvida por BARTLETT (1946). Seja $\rho(1) = 0$ então $\hat{\rho}(1) \sim N\left(0, \frac{1}{n}\right)$ a medida que o número de observações da série temporal cresce.

Além disso, para os casos em que existe essa correlação ($\rho(1) \neq 0$) não há resultados disponíveis, uma vez que a distribuição do estimador depende também do modelo gerador do processo. Essa dificuldade em se calcular a distribuição de $\hat{\rho}(1)$ nos motiva a desenvolver intervalos de confiança não paramétricos.

Com o intuito de estudar diferentes maneiras de se aplicar o *bootstrap* em séries temporais, compararemos três técnicas distintas. Em cada caso, considere uma série temporal $Z(t),\ t \in T = \{1, \ldots, n\}$:

1. *Bootstrap* em blocos (k):

   - Escolha um valor de t entre 1 e $n - k$;
   - Adicione a série $z_t, \ldots, z_{t+k-1}$ à amostra;
   - Repita os processos anteriores até que sua amostra tenha tamanho igual (ou maior) que n;
   - Considere os n primeiros valores da sua amostra;

2. *Bootstrap* em janelas (k):

   - Escolha um valor de t tal que $t = hk + 1,\ h \in \{0, \ldots, \lfloor n/k \rfloor\}$;
   - Adicione a série $z_t, \ldots, z_{t+k-1}$ à amostra;
   - Repita os processos anteriores até que sua amostra tenha tamanho igual (ou maior) que n;
   - Considere os n primeiros valores da sua amostra;

3. *Wavestrap*:

   - Calcule os coeficientes DWT da série;
   - Para cada nível, realize uma amostra aleatória com reposição dos coeficientes DWT (cada amostra deve ter o mesmo tamanho que o número de coeficientes de seu nível)[1];

[1] É recomendado que se confira as correlações dos coeficientes DWT antes de realizar a reamostragem, mas neste caso suporemos que estas correlações são nulas

- Considere a transformada inversa das séries amostradas (em cada nível);

Para comparar os três esquemas amostrais, realizaremos o seguinte procedimento:

1. Defina um valor de ϕ e n;
2. Gere uma série temporal do modelo AR(1) com os parâmetros definidos anteriormente;
3. Gere amostras *bootstrap* da série simulada (para cada um dos esquemas de reamostragem);
4. Em cada uma das amostras *bootstrap*, calcule a primeira autocorrelação;
5. Utilize a amostra de autocorrelações calculadas para gerar intervalos de confiança *bootstrap*;

Compararemos também dois possíveis métodos para gerar intervalos de confiança (com nível de confiança α): O método *plug-in* consiste em pegar os quantis $\frac{1-\alpha}{2}$ e $\frac{3-\alpha}{2}$ da amostra de estimadores. Enquanto o método normal é baseado na distribuição gaussiana:

$$IC(\rho(1), \alpha) = (\hat{\rho}(1) - \widehat{vies}) \pm z_{\frac{1-\alpha}{2}} \widehat{desv},$$

em que $\widehat{vies} = \frac{1}{B}\sum_{i=1}^{B} \hat{\rho}_i^*(1) - \hat{\rho}(1)$ é uma estimativa *bootstrap* para o viés do estimador; $\widehat{desv} = \frac{1}{B-1}\sum_{i=1}^{B}(\hat{\rho}_i^* - \hat{\rho}(1))^2$ é uma estimativa para o desvio padrão do estimador e $z_{\frac{1-\alpha}{2}}$ é o quantil $\frac{1-\alpha}{2}$ da distribuição normal padrão.

Para os testes, utilizaremos $\phi \in \{0.3, 0.5, 0.7\}$ e $n \in \{64, 128, 512, 1024\}$. Para cada combinação de parâmetros geraremos mil séries temporais. Será utilizado $k = 4$ para os *bootstraps* em blocos e em janelas. Para o *wavestrap*, consideraremos a ondaleta "Coif4".

5.2 Resultados

5.2.1 Simulação única

Para ilustrar o comportamento dos estimadores *bootstrap* considerados, faremos uma breve análise dos resultados de uma simulação. A figura 5.1 contém a função de autocorrelação observada:

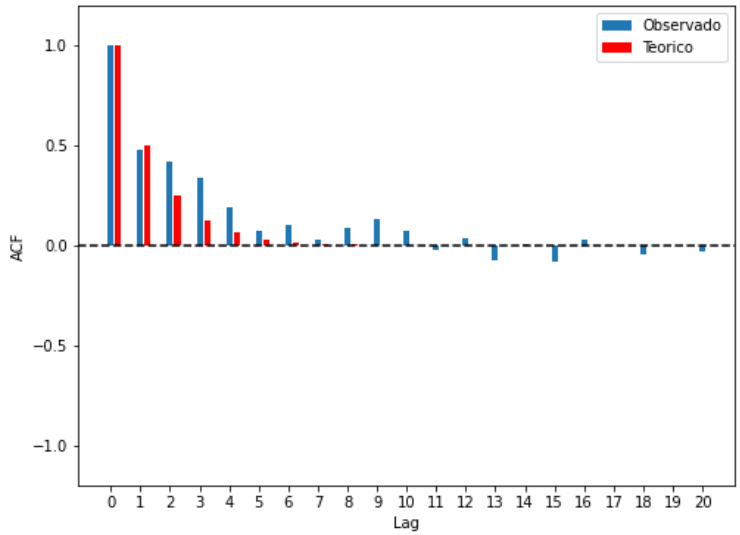


Figura 5.1: Autocorrelações observadas em uma série simulada ($n = 64$, $\phi = 0.5$)

Neste caso $\hat{\rho}(1) = 0.48$, a distância entre as correlações teóricas e observadas diminui a medida que a correlação se aproxima de 0 (o que pode ser observado para os maiores valores de h). A figura 5.2 apresenta as distribuições calculadas pelas diferentes técnicas de *bootstrap*:

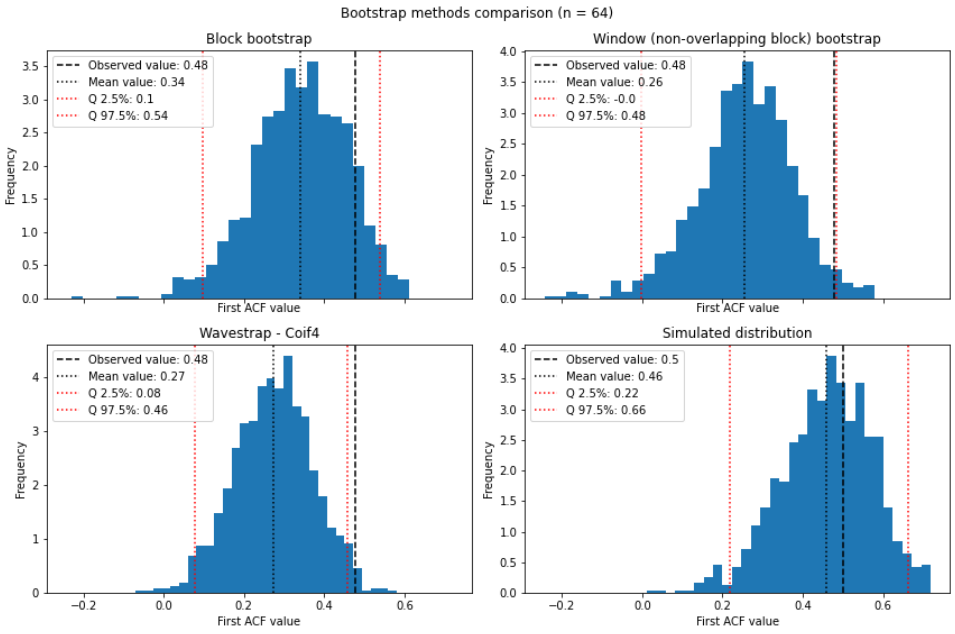


Figura 5.2: Nos três primeiros painéis: Distribuições dos estimadores *bootstrap* para a série simulada. No painel inferior direito: Distribuição das simulações de Monte Carlo ($n = 64$, $\phi = 0.5$).

Na figura 5.2 o painel inferior direito apresenta a distribuição de $\hat{\rho}(1)$ para mil si-

mulações diferentes do processo (diferente dos outros três painéis que utilizam apenas uma amostra para as técnicas *bootstrap*). No que diz respeito a distância entre os quantis 2.5% e 97.5% todos os painéis apresentam resultados parecidos. As técnicas também parecem capturar (até um certo ponto) a assimetria observada na distribuição simulada.

No entanto, é possível notar que os viéses estimados de todas as três técnicas de *bootstrap* consideradas é muito superior ao que se encontra nas simulações (ao invés de um viés de -0.02 unidades, as técnicas estimam viéses de até -0.2 unidades). A figura 5.3 apresenta as mesmas simulações mas com $n = 1024$:

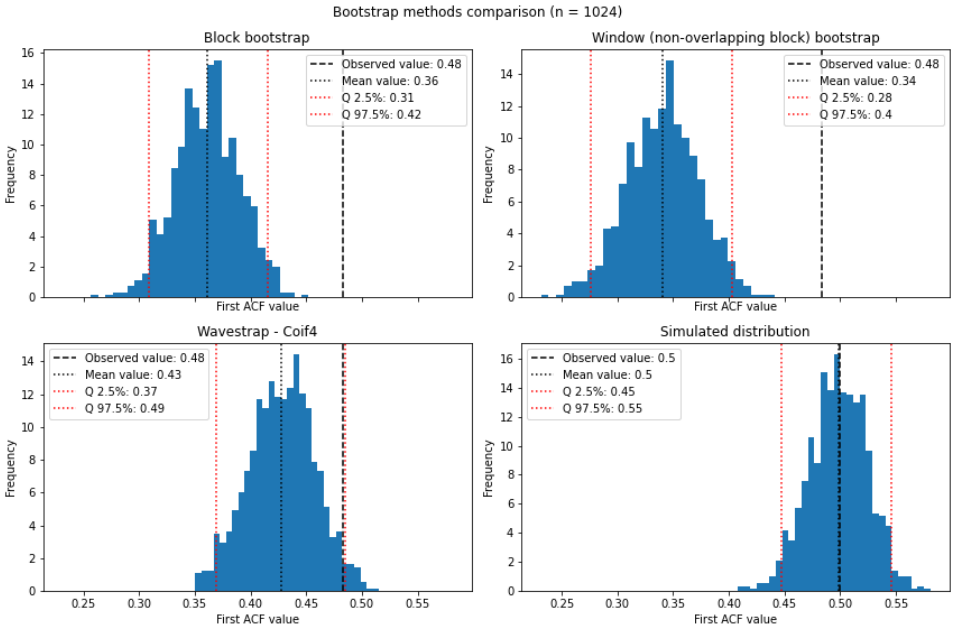


Figura 5.3: Distribuições dos estimadores *bootstrap* para a série simulada ($n = 1024$, $\phi = 0.5$)

Novamente, as distâncias entre os quantis é comparável nos quatro painéis. No entanto, observa-se que os viéses estimados pelas técnicas *bootstrap* continuam grandes quando comparados ao viés da distribuição simulada. A distribuição simulada indica que o estimador $\hat{\rho}(1)$ é consistente (o viés e a variância diminuem a medida que o tamanho da série aumenta).

5.2.2 Simulações gerais

Analisando os resultados de uma série simulada, observamos o seguinte:

- A variância do estimador $\hat{\rho}(1)$ diminui conforme aumentamos o tamanho amostral;

- Para processos do tipo AR(1), a variância do estimador $\hat{\rho}(1)$ é semelhante quando utilizamos as diferentes técnicas de *bootstrap*. Essa variância é também semelhante ao que se encontrou na distribuição simulada.

- Todas as técnicas parecem captar o viés negativo encontrado na distribuição simulada.

- as três formas de *bootstrap* consideradas tendem a superestimar o viés do estimador.

- Essa superestimativa tende a ser mais severa conforme o tamanho amostral aumenta (e o viés real diminui).

No entanto, tomar conclusões após apenas uma rodada de simulações em geral não é uma boa prática. Por conta disso, repetiremos as simulações 1000 vezes, considerando tamanhos amostrais intermediários e outros valores para ϕ (como descrito anteriormente).

A figura 5.4 abaixo apresenta a proporção de vezes que os intervalos de confiança baseados na distribuição normal calculados com cada técnica de *bootstrap* continham o real valor do parâmetro:

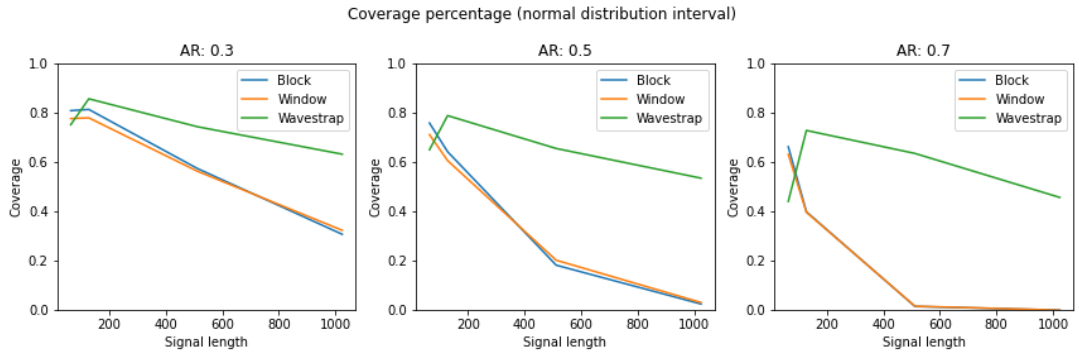


Figura 5.4: Porcentagem de cobertura do real parâmetro conforme aumentamos o tamanho da série.

Todas as técnicas de *bootstrap* apresentam piora conforme o tamanho amostral aumenta. Quando consideramos tamanhos amostrais grandes (maiores que 128), o *wavestrap* apresentou resultados superiores ás duas outras técnicas, que se comportam de maneira semelhante. A figura 5.5 apresenta a proporção de vezes que os intervalos baseados no método *plug-in* continham o real valor do parâmetro:

O intervalo que não utiliza a correção do estimador apresenta cobertura pior do que o baseado na distribuição normal. Novamente o *wavestrap* se destaca dos outros dois

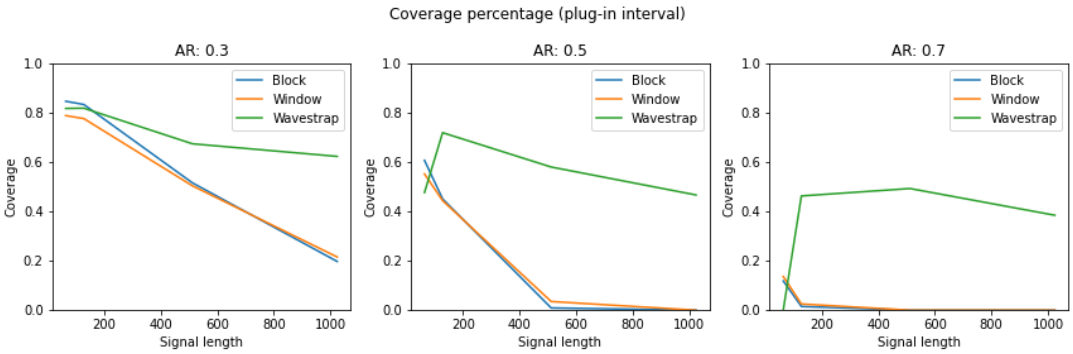


Figura 5.5: Porcentagem de cobertura do real parâmetro conforme aumentamos o tamanho da série.

métodos: apesar de ser pior para tamanhos amostrais pequenos, seu desempenho melhora conforme observamos sequências maiores. Esse comportamento pode ser entendido através da figura 5.6:

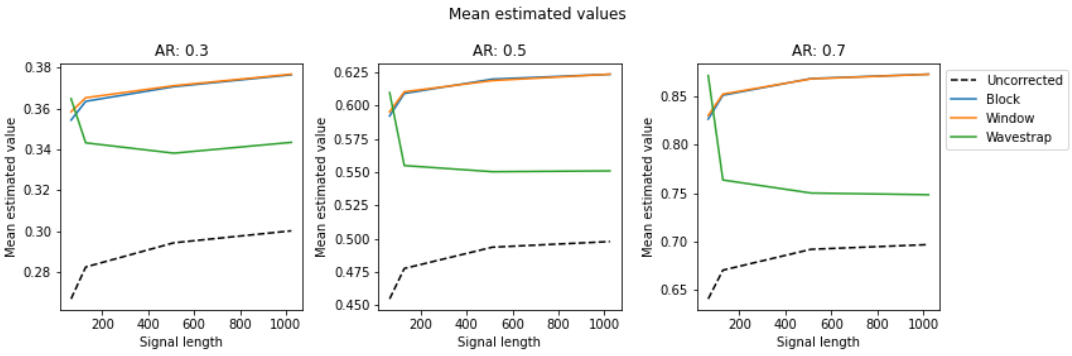


Figura 5.6: Valores médios de $\hat{\rho}(1)$.

Apesar do viés de $\hat{\rho}(1)$ convergir para 0 conforme aumentamos o tamanho da amostra, essa relação não parece ser captada por nenhuma das técnicas de *bootstrap* consideradas. O desempenho dos métodos possui relação direta com a sua capacidade de estimar corretamente o viés. A figura 5.7 contém os intervalos de confiança médios para as 1000 simulações:

Notamos que os intervalos de confiança calculados pelo *bootstrap* em blocos e em janelas são muito próximos (indistinguíveis na figura). Observa-se também que a técnica baseada em ondaletas apresenta intervalos de amplitude semelhantes, mas com centro deslocado (por conta de diferenças na estimativa do viés).

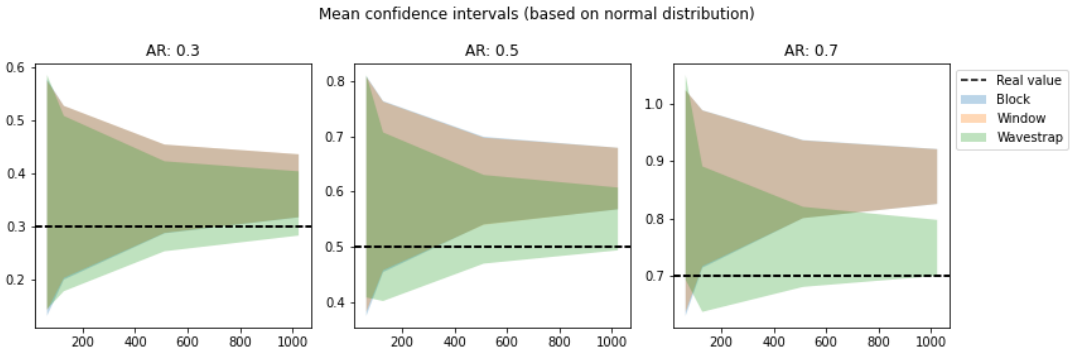


Figura 5.7: Intervalo de confiança médio (baseado no estimador corrigido e na distribuição normal).

Capítulo 6

Considerações finais

As ondaletas constituem uma técnica interessante para a análise de séries temporais, permitindo seu estudo simultaneamente no campo da frequência e do tempo. Nesse sentido, as ondaletas são mais facilmente aplicáveis a séries não estacionárias do que as técnicas mais comuns. Essa característica confere aos métodos baseados na DWT bastante versatilidade.

Outra característica boa das ondaletas é que trata-se de uma técnica não-paramétrica, aumentando sua aplicabilidade, pois muitas vezes as distribuições teóricas associadas aos estimadores ou às previsões são difíceis de se encontrar.

Além das técnicas contidas neste trabalho, as ondaletas possuem outras aplicações, como:

- Suavização de sinais para previsão (KRIECHBAUMER *et al.*, 2014);
- Análise de relações causais entre duas (ou mais) séries temporais (GRINSTED *et al.*, 2004);
- Estimação de densidade (DONOHO *et al.*, 1996);

Isso sem contar aplicações em outras áreas (como compressão de imagens ou análise de sistemas elétricos).

Quando utilizada conjuntamente com o *bootstrap*, as ondaletas novamente se mostram uma técnica versátil e poderosa. No entanto, alguns cuidados são necessários, como a checagem da função de autocorrelação dos coeficientes DWT nos diferentes níveis.

Um aspecto das ondaletas (e do *wavestrap*) que não foi abordado neste estudo é a diversidade de filtros disponíveis, cada um com alguma propriedade desejável e otimizado para aplicações específicas. Essa diversidade pode ser assustadora para iniciantes na área.

O estudo no campo avança a cada dia, já existem (no momento da escrita deste texto) adaptações do *wavestrap* para diversos cenários: como é o caso da técnica *wavestrapping* (EVARISTO, 2010). Baseados na DWPT, estes métodos são análogos às ondaletas, mas com algumas adaptações.

Certamente as técnicas apresentadas neste estudo são úteis para estatísticos e entusiastas em análise de séries temporais. Acredita-se que as análises apresentadas neste trabalho ilustrem algumas das aplicações mais simples desses métodos.

Referências Bibliográficas

BARTLETT, M. S. (1946). On the theoretical specification and sampling properties of autocorrelated time-series. *Supplement to the Journal of the Royal Statistical Society*, **8**(1), 27–41.

BOX, G. E. P. e JENKINS, G. (1970). *Time Series Analysis, forecasting and control*. Holden-Day.

DAUBECHIES, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, **41**(7), 909–996.

DONOHO, D. L., JOHNSTONE, I. M., KERKYACHARIAN, G. e PICARD, D. (1996). Density estimation by wavelet thresholding. *The Annals of statistics*, pages 508–539.

EFRON, B. (1979). Computers and the theory of statistics: Thinking the unthinkable. *Society for Industrial and Applied Mathematics*.

EVARISTO, R. M. (2010). *Métodos de reamostragem de Séries Temporais Baseados em Wavelets*. Master's thesis, Escola Politécnica da Universidade de São Paulo, São Paulo.

FENG, H., WILLEMAIN, T. R. e SHANG, N. (2005). Wavelet-based bootstrap for time series analysis. *Communications in Statistics - Simulation and Computation*, **34**, 393–413.

GARDNER JR, E. S. (1985). Exponential smoothing: The state of the art. *Journal of forecasting*, **4**(1), 1–28.

GRINSTED, A., MOORE, J. C. e JEVREJEVA, S. (2004). Application of the cross wavelet transform and wavelet coherence to geophysical time series. *Nonlinear processes in geophysics*, **11**(5/6), 561–566.

HÄRDLE, W., KERKYACHARIAN, G., PICARD, D. e TSYBAKOV, A. (2012). *Wavelets, approximation, and statistical applications*, volume 129. Springer Science & Business Media.

HYNDMAN, R. J. e ATHANASOPOULOS, G. (2018). *Forecasting: principles and practice*. OTexts.

KRIECHBAUMER, T., ANGUS, A., PARSONS, D. e CASADO, M. R. (2014). An improved wavelet–arima approach for forecasting metal prices. *Resources Policy*, **39**, 32–41.

LEE, G. R., WASILEWSKI, F., WOHLFAHRT, K. e O'LEARY, A. (2019). Pywavelets, a python package for wavelet analysis.

MALLAT, S. G. (2009). A theory for multiresolution signal decomposition: the wavelet representation. In *Fundamental Papers in Wavelet Theory*, pages 494–513. Princeton University Press.

MALLAT, S. G. A. (1999). *A wavelet tour of signal processing*. Elsevier.

MORETTIN, P. A. e TOLOI, C. M. C. (2018). *Análise de séries temporais: modelos lineares univariados*. Editora Blucher.

PERCIVAL, Donald B. e WALDEN, A. T. (2000). *Wavelet Methods for Time Series Analysis*. Cambridge University Press.

QUENOUILLE, M. H. (1956). Notes on bias in estimation. *Biometrika*, **43**.

TANG, L., WOODWARD, W. A. e SCHUCANY, W. R. (2008). Undercoverage of wavelet-based resampling confidence intervals. *Communications in Statistics - Simulation and Computation*, **37**, 1307–1315.

WOODWARD, W. A., GRAY, H. L. e ELLIOTT, A. C. (2017). *Applied time series analysis with R*. CRC press.