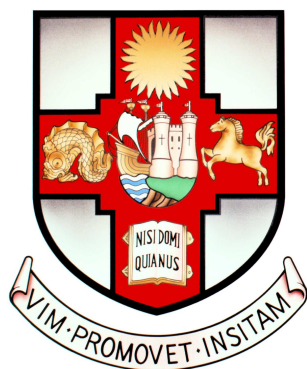

NEW METHODS IN TIME SERIES ANALYSIS: UNIVARIATE TESTING AND NETWORK AUTOREGRESSION MODELLING

KATHRYN ALICE LEEMING



School of Mathematics
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in
accordance with the requirements of the degree of Doctor
of Philosophy in the Faculty of Science.

MARCH 2019

Word count: twenty-six thousand

Abstract

This thesis presents new methods in time series analysis focusing on three areas: stationarity testing, network autoregression modelling, and local white noise testing.

We begin by describing a bespoke stationarity test for use when univariate data has missing observations. This test is based upon a second-generation wavelet method known as the non-decimated lifting transform, which allows for the analysis of irregularly spaced data. The variance of a spectral estimate linked to the non-decimated lifting transform is used in our test statistic, and compared to the same quantity calculated on simulated stationary samples to assess significance.

The second section provides a model for multivariate time series observed on nodes of a network. Our model allows such data to be modelled with few parameters, and is shown to be a useful modelling tool for predicting multivariate time series. A stationarity condition and consistency results for this model are described, and results are generated using our software package for fitting this model.

A local white noise test is motivated in the third section, which can be applied at many different positions in a time series. The smoothed wavelet periodogram forms the basis of our test, and relevant distributional results for its implementation are described, including new results of Haar and Shannon wavelet quantities. Different test statistics are investigated, each based upon testing equality of the periodogram at different scales.

Acknowledgements

I would like to thank my supervisor, Professor Guy Nason, for his patient direction, enthusiasm, and encouragement, and for knowing what to say to keep me motivated, especially towards the end.

To those I've shared this PhD roller coaster with, in particular Sophie, Patrick, Jenny, and Joe, thanks for the mood boosting gym classes, wine, and so much coffee.

Thanks to Tom, for making both the good and bad days better, and to friends and family who have encouraged me along the way. Most importantly, I thank my parents for their love and support through this, and all chapters of my life.

And finally to Andrew, for always reminding me that life's too short to not enjoy what you do.

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: DATE:

Contents

1	Introduction	1
2	Literature Review	3
2.1	Time Series Definitions	3
2.1.1	Fourier analysis	4
2.2	Wavelets and locally stationary wavelet processes	11
2.2.1	Non-decimated wavelets	17
2.2.2	Wavelet packets	21
2.2.3	Second-generation wavelets	23
2.3	Bristol Traffic Data	25
3	Stationarity testing with missing observations	28
3.1	Introduction and additional literature review	28
3.1.1	Tests of second-order stationarity	28
3.1.2	Statistical tools for missing data	30
3.2	Description of our method	31
3.2.1	Estimating the wavelet spectrum	32
3.2.2	LiftToS	35
3.2.3	Constructing bootstrap test statistics using circulant embedding	37
3.2.4	Summary of LiftToS	38
3.2.5	Comparisons to stationarity tests for complete data	38
3.3	Simulations	39
3.4	Examples	41
3.4.1	Air Quality	41
3.4.2	Bristol Traffic	43
3.5	Conclusion	47
4	Generalised Network Autoregressive Processes	48
4.1	Introduction and additional literature review	48
4.1.1	Multivariate time series modelling with autoregression	48
4.1.2	Network models	50

4.1.3	Network regression models	51
4.2	Description of the GNAR Model	56
4.2.1	Network neighbourhood structure	56
4.2.2	Generalised Network Autoregressive Model	57
4.2.3	Comparisons to other models	58
4.3	Theoretical results	60
4.3.1	Stationarity condition for GNAR processes with fixed network . . .	60
4.3.2	Parameter consistency	63
4.4	Implementation	65
4.4.1	Order selection	65
4.4.2	Handling missing observations	66
4.5	Simulations	67
4.5.1	Comparison to theoretical results and model selection	67
4.5.2	Comparison to other models	71
4.6	Examples	74
4.6.1	GDP and network construction	74
4.6.2	Bristol Traffic	79
4.7	Conclusion	83
5	Local White Noise Testing	86
5.1	Introduction and additional literature review	86
5.1.1	White noise tests	86
5.2	Background	89
5.2.1	Shannon wavelets	89
5.2.2	The β -spectrum	91
5.3	Theoretical covariance results	93
5.3.1	Covariance of $I_{j,k}$	93
5.3.2	Covariance of $I_{j,k}$ under smoothing	96
5.3.3	Haar wavelet quantities	96
5.3.4	Shannon wavelet quantities	98
5.4	Local white noise test method	105
5.4.1	Definition of local white noise	105
5.4.2	Estimating the β -spectrum	106

5.4.3	Test statistics	107
5.5	Simulations	109
5.5.1	Multiple test correction	109
5.5.2	Variance and kurtosis estimation	110
5.5.3	Size and power properties as a global test	112
5.5.4	Size and power properties as a local test	117
5.6	Examples	119
5.6.1	GDP example	119
5.6.2	Bristol Traffic	121
5.7	Conclusion	123
6	Discussion	125
6.1	Stationarity testing with missing observations	125
6.2	Generalised Network Autoregressive Processes	126
6.3	Local White Noise Testing	126
	Appendices	127
A	LiftToS example documentation	128
B	GNAR documentation	129
C	LWN documentation	156

List of Tables

1	Bristol Traffic locations and their corresponding descriptions from the Open Data Bristol database.	27
2	Empirical size and power (%) calculated using 100 simulations and 200 bootstrap simulations, $T = 256$ and 5% nominal size. JWW column is from Cho (2016a), Tables 1 and 4.	42
3	Empirical size and power (%) calculated using 100 simulations and 200 bootstrap simulations, $T = 512$ and 5% nominal size. JWW and Cho columns are from Cho (2016a), Tables 1 and 4.	42
4	Empirical size and power (%) calculated using 100 simulations, $T = 256$ and 5% nominal size, where the missing values in the series are replaced with their mean, or removed before using the test by Cho (2016a). LiftToS results are replicated from Table 2.	43
5	Properties of GNAR[1, (1)] global- α estimators of Model 1, using 1000 realisations. * results have been multiplied by 1000 for readability, and \dagger results have been multiplied by 10.	71
6	Number of times each model minimised the BIC criterion, over 1000 simulations of Model 2. The bold numbers highlight the true order of Model 2.	72
7	Properties of GNAR[1,(1)] global- α estimators of Model 1 featuring missingness, using 1000 realisations of series of length $T = 32, 64,$ and 128 . * results have been multiplied by 1000 for readability. 10% of observations at each node are removed.	72
8	Mean number of parameters and prediction error over 1000 simulations of each model. The results from fitting the generating model are in bold. . . .	74
9	Prediction error of differenced GDP change at $t = 52$	79
10	Parameter estimates of the GNAR model with $p = 11, s_1 = 2, s_2 = s_3 = 1,$ and $s_4 = \dots = s_{11} = 0$. Significance codes indicate the size of the p -value associated with each parameter, with * indicating a value under 0.05, ** indicating a value under 0.01, and *** indicating a value under 0.001. . . .	85

11	Bias and MSE ($\times 100$) of different estimators of σ^2 , calculated using 1000 simulations of length $T = 64$ of each distribution.	111
12	Performance of the fourth moment estimator $\hat{\mu}_{4,32}$ over 1000 simulations of white noise of length $T = 64$. All distributions have parameters set to achieve variance 1. * the bias values have been multiplied by 100 for readability.	113
13	Simulated power and size using 1000 simulations of series with $T = 32$, $s = 15$, $J^* = 3$	114
14	Simulated power and size using 1000 simulations of series with $T = 64$, $s = 31$, $J^* = 4$	114
15	Simulated power and size using 1000 simulations of series with $T = 128$, $s = 63$, $J^* = 5$	115
16	Simulated power and size using 1000 simulations of series with $T = 128$, $s = 63$, $J^* = 4$	115
17	Simulated power and size using 1000 simulations of series with $T = 128$, $s = 63$, $J^* = 3$	115
18	Simulated power of local white noise tests against HWWN, Ljung-Box and Bartlett results taken from Table III in Nason and Savchev (2014).	116
19	Simulated power and size using 1000 simulations of series with $T = 32$, $s = 15$, $J^* = 3$ and Uniform innovations.	117
20	Simulated power and size using 1000 simulations of series with $T = 32$, $s = 15$, $J^* = 3$ and Laplace innovations.	117
21	Power of tests applied to a TVMA(1) process over 1000 simulations using standard normal innovations. The LWN global test refers to a rejection at any of the three sections tested. LWN tests have $s = 31$ and $J^* = 4$	118
22	Results of Haar Q LWN test with $J^* = 3$ and $s = 15$ applied to residuals of the GNAR model in Section 4.6.1 at $t = 4 - 35$	120
23	Local white noise test results applied to GNAR residuals at times 197-260. The LWN test used was Shannon C_{com} , with running mean bandwidth $s = 15$ and $J^* = 3$	122

List of Figures

1	Mother wavelet functions of Haar (Daubechies Extremal Phase 1), Daubechies Extremal Phase 2, Daubechies Least Asymmetric 4, Daubechies Least Asymmetric 8, Hermitian 2, Hermitian 3.	13
2	The Bristol Traffic dataset: average hourly car speeds from 12am 19/04/2016 to 8am 11/05/2016 at the 29 locations listed in Table 1. White space indicates missing observations.	26
3	Differenced PM-10 series. Red triangles indicate the positions of missing observations.	44
4	Estimated evolutionary wavelet spectrum of the differenced PM-10 time series. The spectrum is normalised by dividing through by the mean at each scale, as in Equation (65). Vertical white spaces indicate missing values.	44
5	Average hourly speeds of cars inbound on the A38 (Monks Park to Stokes Croft) from 12am on 19/04/2016 to 8am on 11/05/2016. Red triangles indicate the location of missing observations.	45
6	Residuals after seasonal decomposition of hourly speeds of cars inbound on the A38 (Monks Park to Stokes Croft) from 12am on 19/04/2016 to 8am on 11/05/2016. Red triangles indicate the location of missing observations.	46
7	Estimated evolutionary wavelet spectrum of the time series in Figure 6. Vertical white spaces indicate missing values.	46
8	Five node network used in simulations.	68
9	A realisation of Model 2 using the network in Figure 8, with $T = 128$	69
10	Difference between estimated and true parameters for 1000 realisations of Model 1.	70
11	First-differenced annual GDP growth rates of OECD countries. White space at the beginning of the series indicates missing values.	76
12	Boxplots of the prediction error at $t = 51$ using 10,000 random networks a range of GNAR models.	77
13	GDP network minimising the prediction error for the global- α GNAR(2,[2,2]) model.	78
14	Bristol Traffic network, with relative geographic distances preserved.	80

15	Residual values of Bristol Traffic time series after trend and daily seasonality is removed. The residuals at each location are standardised to have variance 1. White space indicates missing values.	81
16	Autocorrelation, partial autocorrelation and cross-correlation plots for series at nodes 1-3. Missing values were replaced with zeros to calculate the correlations.	82
17	Autocorrelation, partial autocorrelation and cross-correlation plots for residuals of the GNAR model fit at nodes 1-3. Missing values were replaced with zeros to calculate the correlations.	84
18	Shannon mother wavelet for $j = 1, \dots, 4$	90
19	Shannon autocorrelation wavelets for $j = 1, \dots, 6$	101
20	Shannon (black) and Haar (red dashed) autocorrelation wavelets for $j = 1, \dots, 6$	102
21	Theoretical covariance (black) and simulated covariance (red dashed) of the Shannon wavelet spectrum over 10,000 simulations of standard Normal white noise with $T = 256$	103
22	Values of $\sum_t \psi_{\ell, m-t}^2 \psi_{j, n-t}^2$ for Shannon wavelets with $\ell = j$ ranging from 1 to 4.	104
23	Empirical density of innovation variance estimates using the standard estimator, and three LWN estimators. This plot is produced using the function <code>density</code> with default smoothing arguments.	112
24	MA parameters of the TVMA(1) processes used in simulations.	118
25	Residuals after GNAR model fit. Red triangles indicate the location of missing observations.	121
26	Residuals after GNAR fit with results of two local white noise tests. Arrows indicate the tested segments using Shannon C_{com} , with running mean bandwidth $s = 15$ and $J^* = 3$	122

1 Introduction

Data frequently have features that are challenging to model using standard techniques, such as missing observations, high-dimensionality, and non-Gaussianity. Attributes such as these can mean that classical models and analysis methods require adaption before use. Three bespoke methods for use with data that do not meet classical assumptions are presented in this thesis: stationarity testing with missing observations, generalised network autoregressive modelling, and local white noise testing.

Chapter 2 provides a general literature review covering an overview of time series and wavelet topics, with particular focus on themes relevant to this thesis. Also introduced is the Bristol Traffic dataset, which is used to demonstrate the new methods in each of the later chapters. Additional literature reviews are found within Chapters 3-5 to cover chapter-specific material.

Missing observations are frequent in practical data analysis, and can arise due to random omission, instrument failure, or human error. Chapter 3 details our new stationarity test designed for use on such data that is complicated by missing observations. This test is based upon the wavelet spectrum estimation method of Knight et al. (2012), which uses second-generation wavelets to handle irregularly-spaced data via the non-decimated lifting transform of Knight and Nason (2009). Our developed test compares the average variance of the spectral estimate over scales, to the same test statistic calculated on bootstrapped time series. When tested on simulated series without missingness our stationarity test performs well in terms of size and power compared with stationarity tests designed for complete data. When up to 20% of observations are missing the test also has good statistical properties, and is a marked improvement on a simple adaption to a test that is designed to work on fully observed data.

High-dimensional time series can be modelled sparsely using the generalised network autoregression model presented in Chapter 4. The multivariate time series is assumed to be observed at nodes of a network, which provides connection information between the nodes. Our network autoregression model combines standard autoregressive behaviour at each node, with network autoregression (dependence on the past values observed at neighbouring nodes). Generalisations including modelling with covariates, changing networks,

and different neighbourhood sets allow this model to be applied in a range of settings. In addition, we demonstrate an application of the changing network framework to model multivariate time series with missing observations. An intuitive stationarity condition is proven for our generalised network autoregression model, and the fitted parameters of our model are shown to be consistent.

Chapter 5 presents a local white noise test that is also designed for general application; using wavelet properties this test is designed for use on Gaussian and non-Gaussian distributed white noise with finite sixth moment. The locally stationary wavelet framework of Nason et al. (2000) is used to motivate test statistics based upon equality of values of the wavelet spectrum at different scales. Distributional results for the wavelet spectrum are calculated to assess the significance of these test statistics across a range of smoothing options for the wavelet spectrum. As a local test, this method is designed to be applied locally at many positions of a time series.

2 Literature Review

In this chapter, an overview of key ideas in time series analysis is presented, with particular focus on material that underpins our new techniques developed in later chapters. Section 2.1 presents general time series concepts and definitions, and Section 2.2 introduces wavelets and locally stationary wavelet processes. Concepts and definitions specific to material in Chapters 3-5 are included within the relevant chapter.

2.1 Time Series Definitions

In this section, useful definitions from the time series literature are stated and discussed. Accessible introductions to time series analysis include Chatfield (2004), Shumway and Stoffer (2000), and Brockwell and Davis (2006), with a more theoretical approach found in Priestley (1981).

We begin with the definition of white noise, a process that forms the basis of many time series models.

Definition 2.1 (White noise). A stochastic process $\{X_t\}_{t \in \mathbb{Z}}$ is called a white noise process with mean zero and variance $\sigma^2 < \infty$ if

1. $\mathbb{E}(X_t) = 0$ and
2. $\mathbb{E}(X_s X_t) = \delta_{s,t} \sigma^2$,

where $\delta_{s,t}$ is the Kronecker delta function, taking the value 1 when $s = t$, and 0 otherwise.

One important question when analysing time series is whether properties of the series change over time. Loosely speaking, if a series has an unchanging generating process it is known as strictly stationary.

Definition 2.2 (Strict stationarity). A process $\{X_t\}_{t \in \mathbb{Z}}$ is strictly stationary if for any admissible collection t_1, \dots, t_n and $k \in \mathbb{Z}$,

$$F_{X_{t_1}, \dots, X_{t_n}}(x_1, \dots, x_n) \equiv F_{X_{t_1+k}, \dots, X_{t_n+k}}(x_1, \dots, x_n), \quad (1)$$

where $F_{\mathcal{M}}(\cdot)$ is the distribution function over the variables in the set \mathcal{M} .

For practical purposes, the less restrictive definition of second-order stationarity is generally used.

Definition 2.3 (Second-order stationarity). A mean-zero stochastic process $\{X_t\}_{t \in \mathbb{Z}}$ is stationary if the following three properties hold:

1. $\mathbb{E}(|X_t|^2) < \infty \quad \forall t \in \mathbb{Z}$
2. $\mathbb{E}(X_t) = 0 \quad \forall t \in \mathbb{Z}$
3. $\gamma(r, s) = \gamma(r + t, s + t) \quad \forall r, s, t \in \mathbb{Z}$,

where $\gamma(r, s) = \text{cov}(X_r, X_s)$, the autocovariance function.

We can also consider higher orders of stationarity, by including further moments in the definition. As the autocovariance is assumed to be time-invariant in the above definition, it is often referred to with a single input; $\gamma(h) = \gamma(r, r + h)$. The autocorrelation is usually defined via the autocovariance, $\rho(h) = \gamma(h)\gamma(0)^{-1}$. In practice, important tools for assessing the second-order properties of a time series are the sample forms of the autocovariance and autocorrelation.

Definition 2.4 (Sample autocovariance). The sample autocovariance of a realisation $\{X_t\}_{t=1}^T$ at lag h is

$$\hat{\gamma}(h) = T^{-1} \sum_{t=1}^T (X_t - \bar{X})(X_{t+h} - \bar{X}), \quad (2)$$

where $\bar{X} = T^{-1} \sum_{t=1}^T X_t$, the sample mean of $\{X_t\}_{t=1}^T$.

Definition 2.5 (Sample autocorrelation). The sample autocorrelation realisation $\{X_t\}_{t=1}^T$ at lag h is

$$\hat{\rho}(h) = \hat{\gamma}(h)\hat{\gamma}^{-1}(0). \quad (3)$$

Definitions 2.4 and 2.5 calculate the second order properties over the entire finite length sample, which means they are global quantities.

2.1.1 Fourier analysis

Fourier methods allow for investigation of the frequency content of a process. In this work, we use the following definition of the Fourier transform pair.

Definition 2.6 (Fourier transform). The Fourier transform of a function $f(x) \in L_2(\mathbb{R})$ is

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx, \quad (4)$$

for $\omega \in \mathbb{R}$, and the inverse Fourier transform of $\hat{f}(\omega)$ is

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega x} d\omega. \quad (5)$$

In the case of a function evaluated at discrete points, we use the discrete Fourier transform.

Definition 2.7 (Fourier transform). The Fourier transform of a discretely sampled function $f(x) \in L_2(\mathbb{R})$, with $x \in \mathbb{Z}$ is

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \sum_{x=-\infty}^{\infty} f(x) e^{-i\omega x}, \quad (6)$$

for $-\pi < \omega \leq \pi$, and the inverse Fourier transform of $\hat{f}(\omega)$ is

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \hat{f}(\omega) e^{i\omega x} d\omega, \quad (7)$$

for $x \in \mathbb{Z}$.

The frequency content of a process is described using the spectral density function, a quantity with links to the Fourier transform.

Definition 2.8 (Spectral density function). The spectral density function of a process $\{X_t\}_{t \in \mathbb{Z}}$ is

$$f(\omega) = (2\pi)^{-1} \sum_{h=-\infty}^{\infty} \gamma(h) e^{-i\omega h}, \quad (8)$$

for $\omega \in [-\pi, \pi]$.

Therefore the spectral density function is the (scaled) Fourier transform of the autocovariance function. We also have the inverse result for the autocovariance function,

$$\gamma(h) = \int_{-\pi}^{\pi} f(\omega) e^{i\omega h} d\omega, \quad (9)$$

for $h \in \mathbb{Z}$, the autocovariance function is the (scaled) inverse Fourier transform of the spectral density function. For a second-order stationary process, the covariance function

is invariant to shifts in time, therefore the spectral density function will also stay fixed throughout a stationary series. The spectral density function is also known as the spectrum of a series, and represents the how the power (variance) of a process is distributed across frequency. The periodogram is used to estimate the spectral density function, presented here for use with real-valued processes.

Definition 2.9 (Periodogram). The periodogram of a realisation $\{X_t\}_{t=1}^T$ of a real-valued process is

$$I_p = \{A(\omega_p)\}^2 + \{B(\omega_p)\}^2, \quad \text{for } p = 0, 1, \dots, \text{floor}(T/2), \quad (10)$$

where $A(\omega_p) = \sqrt{2}T^{-1/2} \sum_{t=1}^T X_t \cos(\omega_p t)$, $B(\omega_p) = \sqrt{2}T^{-1/2} \sum_{t=1}^T X_t \sin(\omega_p t)$, $\{\omega_p\}$ are the Fourier frequencies with $\omega_p = 2\pi p T^{-1}$, and floor is the operator that returns the highest integer equal to or lower than the argument.

When the process $\{X_t\}_{t=1}^T$ is a realisation of a Gaussian white noise process, $X_t \sim N(0, \sigma^2)$ for $t = 1, \dots, T$ with $\sigma^2 < \infty$, each of the components $A(\omega_p)$, $B(\omega_p)$ are normally distributed and independent due to the orthogonality of the trigonometric functions. Therefore the sample periodogram in this case has the following distribution.

Proposition 2.1 (Distribution of the periodogram for Gaussian white noise (adapted from Priestley (1981) Theorem 6.1.1)). For $\{X_t\}_{t=1}^T$, with $X_t \sim N(0, \sigma^2)$ and T even,

$$I_p \sim \begin{cases} \sigma^2 \chi_2^2 & p = 1, 2, \dots, T/2 - 1, \\ 2\sigma^2 \chi_1^2 & p = 0, T/2, \end{cases} \quad (11)$$

where I_p are independently distributed. Therefore the mean and variance of the Gaussian white noise periodogram are

$$\begin{aligned} \mathbb{E}(I_p) &= 2\sigma^2 \quad p = 0, 1, \dots, T/2, \\ \text{var}(I_p) &= \begin{cases} 4\sigma^4 & p = 1, 2, \dots, T/2 - 1, \\ 8\sigma^4 & p = 0, T/2. \end{cases} \end{aligned} \quad (12)$$

The periodogram is formed using orthogonal components, $\cos(\omega_p t)$ and $\sin(\omega_p t)$, and we next define a process with orthogonality. These processes form the building blocks of more complex processes defined through this section.

Definition 2.10 (Orthogonal increment process (adapted from Brockwell and Davis (2006), Definition 4.6.1)). A complex-valued stochastic process $Z(\omega)$ for $\omega \in (-\pi, \pi)$ is called an orthogonal increment process if the following conditions hold:

- (i) $\text{var} \{Z(\omega)\} < \infty, \forall \omega \in (-\pi, \pi)$,
- (ii) $\mathbb{E} \{dZ(\omega)\} = 0, \forall \omega \in (-\pi, \pi)$,
- (iii) $\text{cov} \{dZ(\omega), dZ(\omega')\} = 0, \forall \omega, \omega' \in (-\pi, \pi)$ such that $\omega \neq \omega'$.

Using orthogonal increments, certain processes can be represented according to the following proposition.

Proposition 2.2 (Orthogonal representation of a process (adapted from Priestley (1981), Theorem 4.11.2)). For a zero mean process $\{X_t\}_{t=1}^T$ with covariance function $\gamma(s, t) = \mathbb{E}(X_s X_t)$, let $\{\varphi_t(\omega)\}$ be a collection of functions defined for $t \in \{1, \dots, T\}$ and $\omega \in \mathbb{R}$ such that

$$\gamma(s, t) = \int_{-\pi}^{\pi} \varphi_s^*(\omega) \varphi_t(\omega) d\mu(\omega) \quad \forall s, t \in \{1, \dots, T\} \quad (13)$$

where $*$ denotes the complex conjugate, and

$$\int_{-\infty}^{\infty} |\varphi_t(\omega)|^2 d\mu(\omega) < \infty \quad \forall t \in \{1, \dots, T\}, \quad (14)$$

where $\mu(\omega)$ is a measure defined on the real line.

Then, the process $\{X_t\}_{t=1}^T$ has the following representation;

$$X_t = \int_{-\pi}^{\pi} \varphi_t(\omega) dZ(\omega), \quad (15)$$

where $\{Z(\omega)\}$ is an orthogonal increment process and $\mathbb{E} \{|dZ(\omega)|^2\} = d\mu(\omega)$.

Next, we introduce a special case of Proposition 2.2, the Cramer representation of a stationary process.

Theorem 2.3 (Cramer representation of a stationary series (adapted from Priestley (1981), page 251)). Let $\{X_t\}_{t \in \mathbb{Z}}$ be a mean zero, real-valued, stationary process. Then, there exists an orthogonal increment process $Z(\omega)$, for $\omega \in (-\pi, \pi)$, such that the following representation holds for all $t \in \mathbb{Z}$,

$$X_t = \int_{-\pi}^{\pi} e^{it\omega} dZ(\omega). \quad (16)$$

This representation can also be rewritten as

$$X_t = \int_{-\pi}^{\pi} A(\omega) e^{it\omega} d\xi(\omega), \quad (17)$$

where $\xi(\omega)$ is an orthonormal increment process, which is an orthogonal increment process with $\mathbb{E} \left\{ |dZ(\omega)|^2 \right\} = 1$.

The Cramer representation involves an integral over $A(\omega)e^{it\omega}$, which is a particular case of an oscillatory function, defined next.

Definition 2.11 (Oscillatory functions (adapted from Priestley (1981), Definition 11.2.1)).

The function $\varphi_t(\omega)$ for $t = 1, \dots, T$ and $\omega \in [-\pi, \pi]$, considered as a function of t by fixing ω , is called an oscillatory function if it can be written in the form

$$\varphi_t(\omega) = A_t(\omega) e^{i\theta(\omega)t}, \quad (18)$$

for some frequency $\theta(\omega)$ and

$$A_t(\omega) = \int_{-\pi}^{\pi} e^{it\eta} dK_{\omega}(\eta), \quad (19)$$

with $|dK_{\omega}(\eta)|$ having absolute maximum at $\eta = 0$.

Therefore the oscillatory function has frequency $\theta(\omega)$, with changing amplitude $A_t(\omega)$. As an extension of the stationary representation to a more general class, Priestley used oscillatory functions as building blocks to the oscillatory process model.

Definition 2.12 (Oscillatory processes (adapted from Priestley (1981), Definition 11.2.2)).

A process $\{X_t\}_{t=1}^T$ is called an oscillatory process if there exists a family of oscillatory functions, $\{\varphi_t(\omega)\}$ such that

$$X_t = \int_{-\pi}^{\pi} \varphi_t(\omega) dZ(\omega), \quad (20)$$

where $Z(\omega)$ is an orthogonal process.

As a stationary process can be written in the form of Equation (17), which by inspection is a special case of Definition 2.12, the class of stationary processes is contained within the set of oscillatory processes. As oscillatory processes are not necessarily stationary, the power, or variation, of the process is not necessarily constant over time. These processes motivate use of a time-varying version of the spectrum in Definition 2.8.

Definition 2.13 (Evolutionary power spectrum (adapted from Priestley (1981), Definition 11.2.3)). Let $\{X_t\}_{t=1}^T$ be an oscillatory process with respect to the family of oscillatory functions, \mathcal{F} , where $\mathcal{F} = \{\varphi_t(\omega)\} = \{A_t(\omega)e^{i\omega t}\}$. The evolutionary power spectrum of $\{X_t\}_{t=1}^T$ at time t with respect to \mathcal{F} is

$$dH_t(\omega) = |A_t(\omega)|^2 d\mu(\omega), \quad (21)$$

where $d\mu(\omega) = \mathbb{E} \left\{ |dZ(\omega)|^2 \right\}$ for $dZ(\omega)$ as in Equation (20).

The evolutionary power spectrum describes the distribution of power over frequency, in a time neighbourhood of t . By applying more restrictions to an oscillatory process, Priestley defined a semi-stationary process that has slowly-changing characteristics.

Definition 2.14 (Semi-stationary process (adapted from Priestley (1981), Definition 11.2.4)). A process $\{X_t\}_{t=1}^T$ is called semi-stationary if it is an oscillatory process with respect to a family of oscillatory functions $\mathcal{F} = \{\varphi_t(\omega)\} = \{A_t(\omega)e^{i\theta(\omega)t}\}$ for which $B_{\mathcal{F}}(\omega) = \int_{-\pi}^{\pi} |\theta| |dK_{\omega}(\theta)|$ is bounded $\forall \omega \in [-\pi, \pi]$, and dK_{ω} is as in Equation (19). The family \mathcal{F} has characteristic width $B_{\mathcal{F}} = [\sup_{\omega} \{B_{\mathcal{F}}(\omega)\}]^{-1}$.

The region of approximate stationarity of a semi-stationary process is related to the supremum of the characteristic widths over all oscillatory function families associated with the process with bounded $B_{\mathcal{F}}(\omega)$. The semi-stationary process model can be applied to series with steady time evolution in a generalisation of the class of stationary processes, however a drawback of this model is that sudden changes in the generating process cannot be captured. Both the oscillatory process and semi-stationary process models feature amplitude functions that are defined for each t and ω , which allows for flexibility, but gives an evolutionary spectrum that cannot be found with high resolution in both the time and frequency domain simultaneously.

Another extension from stationary to locally stationary processes was described by Dahlhaus (1997), with a model allowing for both the mean and covariance properties to evolve steadily over time. An important aspect of their work is redefining the process to be doubly indexed, so for each observation length T , the process is rescaled to be considered on $tT^{-1} \in (0, 1]$. Then as T increases, more data is available for approximating local structures. This overcomes a difficulty with the non-stationary models above, as more data corresponds to more elements to estimate in the single-indexed framework.

Definition 2.15 (Locally stationary processes (adapted from Dahlhaus (1997))). A doubly indexed sequence of stochastic processes, $\{X_{t,T}\}_{t=1}^T$, is called locally stationary if there exists a representation

$$X_{t,T} = \mu(tT^{-1}) + \int_{-\pi}^{\pi} A_{t,T}^0(\omega) e^{i\omega t} d\xi(\omega), \quad (22)$$

for $t = 1, \dots, T$, where $A_{t,T}^0(\omega)$ is the transfer function, and μ is the trend, such that properties (i)-(iii) hold.

- (i) $\mu(u)$ is continuous for $u \in [0, 1]$.
- (ii) $\xi(\omega)$ is a stochastic process on $[-\pi, \pi]$ with $\overline{\xi(\omega)} = \xi(-\omega)$ and

$$\kappa \{d\xi(\omega_1), \dots, d\xi(\omega_k)\} = \sum_{m=-\infty}^{\infty} \delta \left(\sum_{j=1}^k \omega_j + 2\pi m \right) g_k(\omega_1, \dots, \omega_{k-1}) d\omega_1 \dots d\omega_k, \quad (23)$$

where κ denotes the k th order cumulant, δ is the Diric delta function, and g_k is a function defined for $[-\pi, \pi]^{k-1}$ such that $g_1 = 0$, $g_2(\omega) = 1$ and $|g_k(\omega_1, \dots, \omega_{k-1})| \leq C_k$, where C_k is a constant for all $k \in \mathbb{N}$.

- (iii) There exists a function $A : [0, 1] \times \mathbb{R} \rightarrow \mathbb{C}$, continuous in its first argument and 2π periodic, with $\overline{A(u, \omega)} = A(u, -\omega)$, such that

$$\sup_{t, \omega} |A_{t,T}^0(\omega) - A(tT^{-1}, \omega)| \leq KT^{-1}, \quad \forall T. \quad (24)$$

Equation (22) reduces to the Cramer representation of a stationary process as written in Equation (17) when the transfer function $A_{t,T}^0(\omega)$ does not depend on the double index (t, T) , and the mean is constantly zero, $\mu(tT^{-1}) \equiv 0$.

The time-varying spectral density of the locally stationary process of Dahlhaus (1997) is $f(u, \omega) = |A(u, \omega)|^2$. Under smoothness conditions on the function A , this time-varying spectral density can be shown to be asymptotically equal to the Fourier transform of the covariance evaluated at uT , a locally stationary process equivalent of Equation (8), see Dahlhaus (1996) for further details. Although this model has many benefits, particularly for theoretical development when T is increasing, it does not allow for step changes in the process generation. As an extension of this process, Adak (1998) defined a piecewise locally stationary process to be one that admits a locally stationary representation as in

Equation (22) for all $u \in [0, 1]$ except for finitely many jump locations.

Another non-stationary process model is the SLEX model of Ombao et al. (2002). This model uses smooth localised complex exponential (SLEX) functions which are localised in both time and frequency, as described in Chapter 4 of Wickerhauser (1994). These functions form a basis of $L^2(\mathbb{R})$, allowing exact representation of signals within the space. The SLEX model contrasts with Dahlhaus' model above as the Fourier components of the locally stationary model above are not localised in time. The SLEX basis has blocks S_i with associated basis vector elements $\{\varphi_{S_i, \omega_{k_i}}(t)\}$ for $\omega_{k_i} = k_i |S_i|^{-1}$ and $k_i = -|S_i|/2 + 1, \dots, |S_i|/2$, and the union of vectors over blocks S_i forms a basis of $L^2(\mathbb{R})$. The SLEX representation of a double-indexed random process $\{X_{t,T}\}_{t=1}^T$ with zero mean is of the form

$$X_{t,T} = \sum_i M_i^{-1/2} \sum_{k_i=-M_i/2+1}^{M_i/2} A_{S_i, k_i, T} \overline{\varphi_{S_i, \omega_{k_i}}}(t) dZ(2^i k_i / T), \quad (25)$$

where $A_{S_i, k_i, T}$ are complex-valued coefficients with $|A_{S_i, k_i, T}|^2 > 0$, $M_i = 2^{-i}T$, $\{\varphi_{S_i, \omega_{k_i}}\}_i$ form a basis of $L^2(\mathbb{R})$, and $Z(\omega)$ is an orthonormal increment process for $\omega \in [-1/2, 1/2]$.

The spectrum of a SLEX model for a frequency $\omega_{k_{i^*}}$ corresponding to block S_{i^*} is

$f_T(u, \omega_{k_{i^*}}) = \sum_i |A_{S_i, k_i, T}|^2 \mathbb{1}_{S_i} \{\text{floor}(uT)\}$, for $k_{i^*} = -|S_{i^*}|/2 + 1, \dots, |S_{i^*}|/2$, where $S_{i^*} \in \cup_i S_i$. As in previous stochastic process representations, the SLEX representation features oscillatory functions, amplitudes, and a random increment process. By using the SLEX functions, the SLEX model is able to offer a time-frequency decomposition of a process. This contrasts the previous models within this section as they feature non-localised exponential functions.

2.2 Wavelets and locally stationary wavelet processes

Another useful way to represent time series uses orthogonal wavelet components.

Daubechies (1992) and Meyer (1993) are comprehensive sources for the development of wavelets, whilst Vidakovic (2009) and Nason (2008) provide accessible introductions to the subject focusing on the use of wavelets in statistics, and Percival and Walden (2000) give a time series approach. The material in this section adapted from the above sources, and further references herein.

Wavelets are constructed based upon two functions, a father wavelet φ , and a mother wavelet, ψ . The father wavelet is a function that integrates to a non-zero value,

$\int \varphi(t)dt \neq 0$, and the mother wavelet is an oscillatory function that integrates to 0, $\int \psi(t)dt = 0$. For many commonly used wavelet families the father wavelet integrates to 1, as is the case with the example given below, the Haar wavelet.

Definition 2.16 (Haar wavelet). The (continuous) Haar wavelet family has father wavelet

$$\varphi(t) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & \text{otherwise,} \end{cases}$$

and mother wavelet

$$\psi(t) = \begin{cases} 1 & 0 \leq t \leq 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & \text{otherwise.} \end{cases}$$

The Haar wavelet has a simple analytical form for both the father and mother wavelets, however this is not the case for all wavelets. Daubechies wavelets are constructed according to satisfying multi-resolution properties introduced later in Definition 2.17, and apart from the special case of the Haar wavelet, the mother and father Daubechies wavelets do not have closed analytic expressions. Instead, filters are used to construct these wavelets, and as the filters have a finite number of non-zero elements the constructed wavelets are compactly supported. See Daubechies (1992), Section 6 for values of these filter coefficients and construction details. Daubechies wavelets are indexed by the number of vanishing moments of the mother wavelet, where ψ is said to have p vanishing moments if

$$\int_{-\infty}^{\infty} x^n \psi(x) dx = 0 \quad \text{for } n = 0, \dots, p-1. \quad (26)$$

The number of vanishing moments corresponds to the order of a polynomial that a linear combination of wavelet functions could reproduce. Daubechies wavelets are asymmetric, with exception of the Haar wavelet, and are characterised as either ‘extremal phase’ or ‘least asymmetric’, with the latter indicating wavelets that are designed to be closer to symmetry.

Another family of wavelets are called the Hermitian wavelets, the Hermitian n wavelet has mother wavelet that is a scaled n th derivative of the Gaussian distribution. A particular member of this wavelet family is the Mexican hat wavelet, which is related to the second

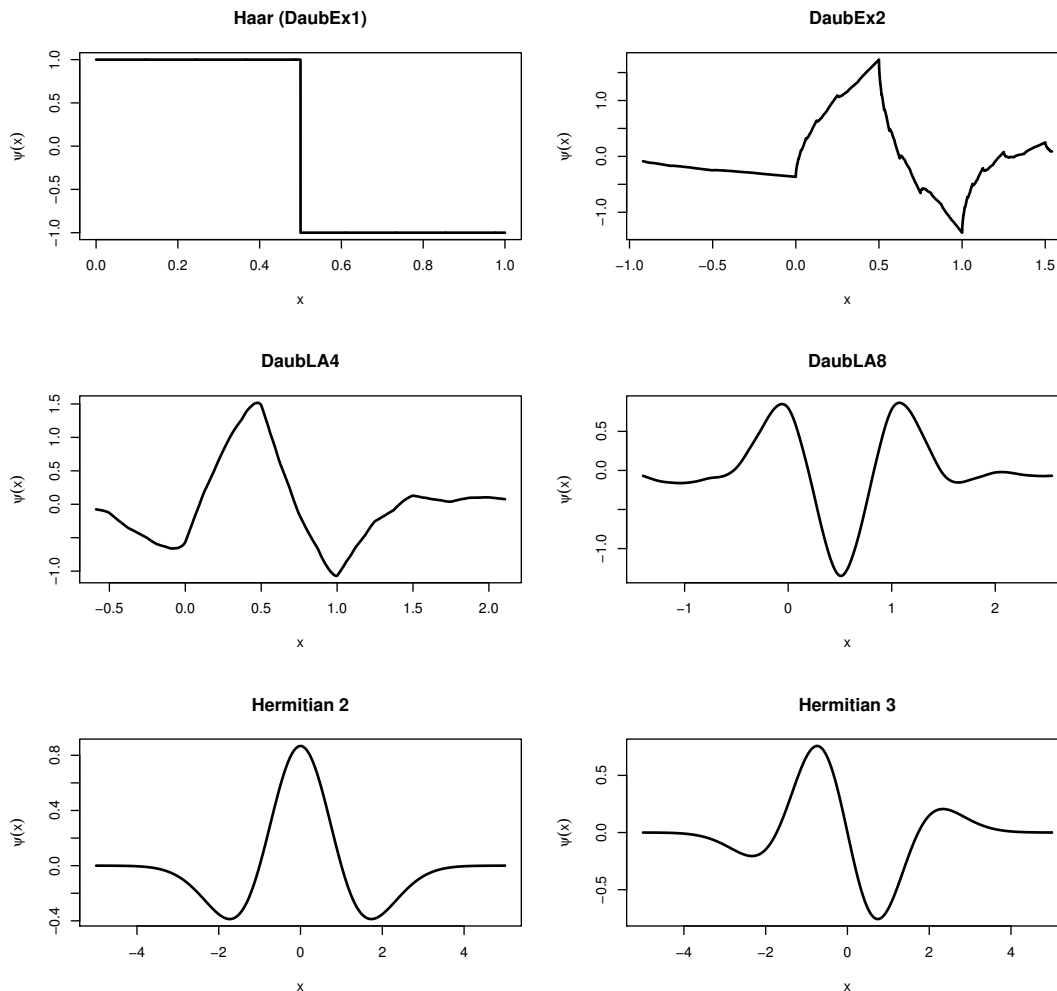


Figure 1: Mother wavelet functions of Haar (Daubechies Extremal Phase 1), Daubechies Extremal Phase 2, Daubechies Least Asymmetric 4, Daubechies Least Asymmetric 8, Hermitian 2, Hermitian 3.

derivative of the Gaussian distribution,

$$\psi(x) = \frac{2}{\sqrt{3\pi^{1/4}}} (1 - x^2) e^{-x^2/2}. \quad (27)$$

Unlike Daubechies wavelets, Hermitian mother wavelets do not decay to zero on a finite interval so have infinite support, have a closed form expression, and are symmetric. Figure 1 shows examples of the Daubechies and Hermitian wavelets and demonstrates that wavelets are not necessarily smooth functions, nor are they necessarily centred at the origin. Many other wavelets could also be chosen, each with different benefits that will suit different applications.

From the building blocks of a mother and father wavelet, double-indexed wavelets are

constructed as translations and dilations of the original wavelet functions. These translations and dilations may be chosen so that these wavelet functions form an orthonormal basis over $L^2(\mathbb{R})$, by setting $\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k)$ for $j, k \in \mathbb{N}$, along with satisfying other conditions. This forms a set of decimated wavelets, where the scale, j , controls the dilation of the wavelet, and k indicates the position. As j increases, the wavelet gets wider but still has norm one. With this orthonormal set, any function can be represented as the sum over decimated wavelets, as they fit into a multi-resolution analysis, as defined by Daubechies (1992), and used here with the indexing of Mallat (1989a).

Definition 2.17 (Multi-resolution analysis (adapted from Daubechies (1992) Section 5.1)). A multi-resolution analysis is a sequence of closed subspaces $V_j \in L_2(\mathbb{R})$ that satisfy the following conditions:

1. $V_j \subset V_{j+1} \forall j \in \mathbb{Z}$,
2. $\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2(\mathbb{R})$, where \overline{W} denotes the closure of set W ,
3. $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$,
4. $f(x) \in V_j \iff f(2^{-j}x) \in V_0$,
5. $f(x) \in V_0 \implies f(x - n) \in V_0 \forall n \in \mathbb{Z}$,
6. $\exists \varphi \in V_0$ such that $\{\varphi_{0,n}; n \in \mathbb{Z}\}$ is an orthonormal basis in V_0 , where $\varphi_{j,n}(x) = 2^{j/2}\varphi(2^jx - n), \forall j, n \in \mathbb{Z}$.

As in Jawerth and Sweldens (1994), we denote the orthogonal complement of subset V_j in V_{j+1} as W_j , and the mutually orthogonal subspaces in the set $\{W_j\}_{j \in \mathbb{Z}}$ are called wavelet spaces.

The construction of wavelets can be defined in terms of filters, such as is used in the wavelet scaling equation.

Definition 2.18 (Wavelet scaling equation). The scaling equation of a father wavelet $\varphi(x) \in V_0$ is

$$\varphi(x) = \sum_{k \in \mathbb{Z}} h_k \sqrt{2} \varphi(2x - k), \tag{28}$$

where h_k are filter coefficients, and $\mathbf{h} = \{h_k, k \in \mathbb{Z}\}$ is a vector (which may be of infinite length) called a low-pass quadrature mirror filter.

The wavelet scaling equation exists because $V_0 \subset V_1$, so the function $\varphi(x) \in V_0$ can be expressed as the sum of the basis functions of V_1 . If a sequence of subspaces satisfies the properties of Definition 2.17, then an orthonormal basis of $L_2(\mathbb{R})$ exists, $\{\psi_{j,k}(x) = 2^{j/2}\psi(2^jx - k)\}_{j,k \in \mathbb{Z}}$. Additionally, $\{\psi_{j,k}(x)\}_{k \in \mathbb{Z}}$ forms an orthonormal basis of $W_j = V_{j+1} \setminus V_j$ for $j \in \mathbb{Z}$. Similarly to the wavelet scaling equation above, the (mother) wavelet function $\psi(x) = \psi_{0,0}(x)$ can be constructed from the (father) scaling function as follows;

$$\psi(x) = \sum_{k \in \mathbb{Z}} g_k \sqrt{2} \varphi(2x - k), \tag{29}$$

for filter coefficients g_k , where $\mathbf{g} = \{g_k, k \in \mathbb{Z}\}$ is a high-pass quadrature mirror filter. The usual relationship between \mathbf{h} and \mathbf{g} is

$$g_k = (-1)^k h_{1-k}, \tag{30}$$

for $k \in \mathbb{Z}$.

Definition 2.19 (Wavelet scaling coefficients). The wavelet scaling coefficients of a function $f(x) \in L_2(\mathbb{R})$ are

$$c_{j,k} = \int_{-\infty}^{\infty} f(x) \varphi_{j,k}(x) dx, \tag{31}$$

for $j, k \in \mathbb{Z}$.

Definition 2.20 (Wavelet detail coefficients). The wavelet detail coefficients, or wavelet coefficients of a function $f(x) \in L_2(\mathbb{R})$ are

$$d_{j,k} = \int_{-\infty}^{\infty} f(x) \psi_{j,k}(x) dx, \tag{32}$$

where $j, k \in \mathbb{Z}$.

As the set of wavelets, $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$, form an orthonormal basis of $L_2(\mathbb{R})$, a function in the space can be written as

$$f(x) = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} d_{j,k} \psi_{j,k}(x), \tag{33}$$

where $d_{j,k}$ are the detail coefficients of Equation (32).

Due to the multi-resolution framework, we also have the following representation of a

function in $L_2(\mathbb{R})$;

$$f(x) = \sum_{k \in \mathbb{Z}} c_{\ell,k} \varphi_{\ell,k}(x) + \sum_{j \geq \ell} \sum_{k \in \mathbb{Z}} d_{j,k} \psi_{j,k}(x), \quad (34)$$

for a fixed $\ell \in \mathbb{Z}$.

In practice, we have a discretely-sampled realisation of a process rather than a function observed on a continuum. For a process $\mathbf{X} = \{X_t\}_{t=1}^T$ with $T = 2^J$ observations, the discrete wavelet transform produces the scaling and wavelet coefficients, similar to those in Definitions 2.19-2.20. This discrete transform can also be written step-wise using the quadrature mirror filters \mathbf{h} and \mathbf{g} , also known as Mallat's pyramid algorithm Mallat (1989b). The scaling coefficients at scale $\ell - 1$ can be constructed from those at scale $\ell = 2, \dots, J$ using the relationship

$$c_{\ell-1,k} = \sum_n h_{n-2k} c_{\ell,n}, \quad (35)$$

for $k \in \mathbb{Z}$, where the original data is $\mathbf{c}_J = \mathbf{X}$, so the original data are considered as being subspace V_J . The equivalent wavelet coefficient equation is

$$d_{\ell-1,k} = \sum_n g_{n-2k} c_{\ell,n}. \quad (36)$$

This pyramid algorithm can also be used in reverse to reconstruct scaling coefficients at scale ℓ from those at $\ell - 1$ as

$$c_{\ell,k} = \sum_n h_{n-2k} c_{\ell-1,n} + \sum_n g_{n-2k} d_{\ell-1,n}, \quad (37)$$

for $\ell = 2, \dots, J$.

The discrete decimated wavelet transform produces half as many coefficients at each progressive scale due to the decimation, and in total, the discrete decimated wavelet transform of a process of length T produces T coefficients.

As the wavelet framework allows for both deconstruction and reconstruction of a signal, it can be applied to many numerical problems and application areas, and a small selection of examples are detailed here. A natural use of wavelets is for investigating vibration data, such as the use of harmonic wavelets demonstrated by Newland (2000) on examples such as experimental pressure wave fluctuations and ground vibrations caused by passing trains. An example of wavelet use in a less obvious area, numerical analysis, is provided by

Alpert (2002), who used wavelet basis representations to improve the computation time of integral equations, also discussed in Chui (1997), Section 7.3. Wavelet use in turbulence problems is detailed in Jaffard et al. (2001), Chapter 9, with wavelets providing space-scale representations ideal for structure detection.

Wavelets are also used in medical applications, such as constructing diagnostic techniques based upon heart sounds as shown by Akay (1996). Field (2000) demonstrated the link between wavelet transforms and visual systems of mammals, with a focus on structure extraction rather than data compression. In economic data settings wavelets have many advantages, some of those noted by Ramsey (2000) included locating discontinuities or isolated events, modelling the different market behaviour at different scales, and forecasting in a non-stationary setting. In the statistical setting, wavelet use includes non-parametric function estimation by providing a sparse representation of a noisy signal, see eg Johnstone (2000). Thresholding wavelet coefficients and analysis of unequally-spaced data by interpolating to a regularly-spaced grid was discussed by Kovac and Silverman (2000). Recently, wavelets have been used in non-parametric regression applications such as estimating random fields where spatial dependence is present by Krebs (2018).

2.2.1 Non-decimated wavelets

Rather than using the translation and dilation operations to form an orthonormal set of wavelet functions, they can also be used to construct a non-decimated set as described by Nason and Silverman (1995). A set of non-decimated wavelets is constructed from the mother wavelet as $\psi_{j,k}(t) = 2^{-j/2}\psi\{2^{-j}(t-k)\}$ for $j, k \in \mathbb{N}$. Whilst these non-decimated wavelets still individually have mean-square norm of 1, they no longer form an orthogonal set, but form an over-complete basis set. A benefit of using non-decimated wavelets is that the transform is translation invariant. This means that the transform generates the same coefficients (although labels are switched) when a different time origin is chosen and data undergo circulant shifting. Translation invariance is not a feature of the decimated wavelet transform due to the dyadic sampling positions. The non-decimated wavelet transform of a process of length $T = 2^J$ results T coefficients at every scale, and a total of $JT = T \log_2(T)$ coefficients when evaluated at J scales.

Non-decimated wavelets were used by Nason et al. (2000) in construction of locally stationary wavelet processes, and the following definitions in this section are adapted from

that work. A locally stationary wavelet process is formed of innovations or increment sequence ξ_{jk} , non-decimated wavelets $\psi_{j,k}$, and constants $w_{j,k;T}$. Here we denote the scales, j , with positive integers. Together, the three elements above define a locally stationary wavelet process of length T , where T is of power of two length.

Definition 2.21 (Locally Stationary Wavelet process). Locally stationary wavelet processes, $\{X_{t,T}\}_{t=1,\dots,T}$, are a sequence of doubly indexed stochastic processes such that

$$X_{t,T} = \sum_{j=1}^J \sum_k w_{j,k;T} \psi_{j,k}(t) \xi_{jk}, \quad (38)$$

where ξ_{jk} is a random orthogonal increment sequence, $\psi_{j,k}$ is a discrete non-decimated family of wavelets, and $T = 2^J$. The orthogonal increment process is a white noise process with zero mean and covariance array $\text{cov}(\xi_{jk}, \xi_{\ell m}) = \delta_{j\ell} \delta_{km}$. The constants $w_{j,k;T}$ have the following smoothness constraint; there exists a sequence of constants C_j and a Lipschitz continuous function $W_j(z)$ such that for each T ,

$$\sup_k |w_{j,k;T} - W_j(k/T)| \leq C_j/T \quad (39)$$

where $\sum_{j=1}^{\infty} C_j < \infty$, and the function $W_j(z)$ for each j and $z \in (0, 1)$ satisfies $\sum_{j=1}^{\infty} |W_j(z)|^2 < \infty$ uniformly in $z \in (0, 1)$. In addition, the Lipschitz constants, L_j are uniformly bounded in j , and $\sum_{j=1}^{\infty} 2^{-j} L_j < \infty$.

Similar to the time series spectral density function in the Fourier domain, the evolutionary wavelet spectrum describes the power of a process, with a key difference being that the evolutionary wavelet spectrum is localised in time and scale, rather than frequency.

Definition 2.22 (Evolutionary Wavelet Spectrum). The evolutionary wavelet spectrum of a sequence $\{X_{t,T}\}_{t=1,\dots,T}$ for the infinite sequence $T \geq 1$ is

$$S_j(z) := |W_j(z)|^2, \quad \text{for } j = 1, \dots, J(T) = \log_2(T), z \in (0, 1), \quad (40)$$

with respect to the generation wavelet $\psi_{j,k}$.

Rather than considering the power of the process over frequency like in Fourier analysis, the wavelet spectrum represents the power over scales. The raw wavelet periodogram is the first step to estimating the evolutionary wavelet spectrum.

Definition 2.23 (Wavelet periodogram). The (raw) wavelet periodogram of $\{X_{t,T}\}_{t=1,\dots,T}$ is formed by squaring the detail coefficients of its wavelet transform,

$$I_{j,k} = |d_{j,k}|^2 = \left| \sum_{t=1}^T X_t \psi_{j,k-t} \right|^2, \quad (41)$$

for $j \in 1, \dots, J$ and $k \in 1, \dots, T$, where $\psi_{j,k-t} = \psi_{j,k}(t)$.

The raw wavelet periodogram is corrected using the inverse of the autocorrelation wavelet matrix.

Definition 2.24 (Autocorrelation wavelet). An autocorrelation wavelet, Ψ_j , is defined for $j, \tau \in \mathbb{Z}$ as

$$\Psi_j(\tau) = \sum_k \psi_{j,k} \psi_{j,k-\tau}. \quad (42)$$

Definition 2.25 (Autocorrelation wavelet matrix). The autocorrelation wavelet matrix, A_J , is a $J \times J$ matrix with entries

$$A_{i,j} = \sum_{\tau} \Psi_i(\tau) \Psi_j(\tau) = \sum_{\tau} \left(\sum_k \psi_{i,k} \psi_{i,k-\tau} \right) \left(\sum_l \psi_{j,l} \psi_{j,l-\tau} \right). \quad (43)$$

The previous quantities are combined to form the corrected wavelet periodogram as follows.

Definition 2.26 (Corrected wavelet periodogram). The corrected wavelet periodogram of a process $\{X_{t,T}\}_{t=1,\dots,T}$ where $T = 2^J$ is a matrix L of dimension $J \times T$ such that

$$L = A_J^{-1} I, \quad (44)$$

where I is the $J \times T$ raw wavelet periodogram matrix.

As shown by Nason et al. (2000), this corrected wavelet periodogram is an asymptotically unbiased but not consistent estimator of the evolutionary wavelet spectrum and requires smoothing for consistency.

Compared to oscillatory processes, the locally stationary wavelet model benefits from a time-scale representation, but also has the drawback of not allowing for sudden changes in the generating process due to the smoothness constraints. Van Bellegem and von Sachs (2008) presented an extension of the locally stationary wavelet model that replaces the Lipschitz continuity assumption for $W_j(z)$ with the following condition bounding the total

variation of $W_j^2(z)$,

$$\sup \left\{ \sum_{i=1}^I |W_j^2(a_i) - W_j^2(a_{i-1})|, 0 < a_0 < \dots < a_I < 1 \right\} \leq M_j, \quad (45)$$

where the supremum is over all possible partitions a_0, \dots, a_I of $(0, 1)$ with $I \in \mathbb{N}$, and $\sum_j M_j < \infty$. This allows for discontinuous behaviour in the process, for example concatenation of two processes to create a process featuring a break-point rather than smooth transition. Their work describes estimating the wavelet spectrum using an adaptive averaging of the corrected wavelet periodogram of Definition 2.26, by averaging using as large a segment as possible for which the spectrum estimate appears homogeneous.

The univariate locally stationary wavelet model and extensions have been used in many applications, including forecasting time series by Fryzlewicz et al. (2003), where forecasts are created for series which may have variance and autocorrelation changes over time. An alternative forecasting approach was proposed by Xie et al. (2009) and demonstrated on financial time series, this forecasting technique was designed to avoid outliers by imposing restrictions on the predictors. Consistent classification of time series using the empirical wavelet spectrum was demonstrated by Fryzlewicz and Ombao (2009), including results on earthquake and explosion recordings. A binary segmentation approach was used on the wavelet periodogram by Cho and Fryzlewicz (2012) to split a non-stationary time series into stationary blocks, and the uncertainty associated with wavelet periodogram-based changepoint identification was quantified by Nam et al. (2015) using a hidden Markov modelling approach.

Tests of stationarity using the locally stationary wavelet framework include Nason (2013) and Cho (2016a). Costationarity, where a linear combination of two non-stationary time series is stationary, was also tested using this framework by Cardinali and Nason (2010). A test of local white noise and aliasing (sampling at a rate lower than the highest frequency content of the series) was proposed by Eckley and Nason (2018) and demonstrated on a wind speed application. Due to the time and scale localisation of the wavelet framework, it is natural to apply it to assess questions of stationarity and evolving structure.

Locally stationary wavelet random fields were introduced by Eckley et al. (2010), which are a two-dimensional extension to the locally stationary wavelet process, shown to be useful for texture analysis of images. A multivariate generalisation of the locally stationary

wavelet model was proposed by Park et al. (2014) and used to investigate linear dependencies between univariate components of multivariate series. This multivariate model was applied to a time-varying class identification problem by Park et al. (2018).

2.2.2 Wavelet packets

Another type of wavelet analysis uses wavelet packets, a subset of which are the wavelets discussed above.

Definition 2.27 (Wavelet packets (adapted from Vidakovic (2009), Section 5.3.1)). A wavelet packet library is a set of functions

$$W_{j,n,k}(x) = 2^{j/2}W_n(2^jx - k) \quad \text{for } (j, n, k) \in \mathbb{Z} \times \mathbb{N} \times \mathbb{Z}, \quad (46)$$

where the functions W_n are related by the following sequence

$$\begin{aligned} W_{2n}(x) &= \sum_k h_k \sqrt{2}W_n(2x - k), \\ W_{2n+1}(x) &= \sum_k g_k \sqrt{2}W_n(2x - k), \quad \text{for } n \in \mathbb{N}, \end{aligned} \quad (47)$$

and $W_0(x)$ integrates to 1. $W_0(x)$ is the scaling function, and $W_1(x)$ is the wavelet function corresponding to quadrature mirror filters \mathbf{h} and \mathbf{g} . The indices j, k are scaling and translation parameters as in the wavelet framework, and n represents an oscillation parameter.

Rather than being linked by just the \mathbf{h} filter as in the father wavelet case of Definition 2.18, wavelet packets use both \mathbf{h} and \mathbf{g} in construction. The standard wavelet basis is contained within the wavelet packet library, as it is constructed using the \mathbf{h} and \mathbf{g} filters in a particular order. The wavelet packets can be selected from the library to form a basis by choosing particular indices, as described by the following Theorem.

Theorem 2.4 (Wavelet packet basis (adapted from Vidakovic (2009))). Define a collection of indices $\mathbb{P} \subseteq \mathbb{Z} \times \mathbb{N}$ such that the intervals $R_{j,n} = \{[2^j n, 2^j(n+1)], (j, n) \in \mathbb{P}\}$ are disjoint and form a countable covering of $\{0\} \cup \mathbb{R}^+$. Then a complete, orthonormal basis of $L_2(\mathbb{R})$ is

$$\{W_{j,n,k}(x), (j, n, k) \in \mathbb{P} \times \mathbb{Z}\}. \quad (48)$$

This orthonormal basis can be used to give another representation of any $f(x) \in L_2(\mathbb{R})$, where \mathbb{P} is as in Theorem 2.4,

$$f(x) = \sum_{(j,n) \in \mathbb{P}} \sum_{k \in \mathbb{Z}} \langle f, W_{j,n,k} \rangle W_{j,n,k}, \quad (49)$$

where $\langle f, W_{j,n,k} \rangle$ are called the wavelet packet coefficients.

Elements of a wavelet packet library can be selected depending on the situation, and a ‘best’ basis chosen as discussed by Percival and Walden (2000), Section 6.3. Examples of a ‘best’ basis include those that minimise the number of wavelet packet coefficients with absolute value over a threshold, or minimising norms of the coefficients.

In an analogous framework to locally stationary wavelet processes, Cardinali and Nason (2017) define the locally stationary wavelet packet process. These processes have an associated wavelet packet basis which will be indexed by b , such that $\{W_{j_p, n_p, k}(x), p \in b, k \in \mathbb{Z}\}$ forms the orthonormal basis. The locally stationary wavelet packet process of Cardinali and Nason (2017) allows representation of a doubly indexed process $\{X_{t,T}\}_{t=1, \dots, T}$ with $T = 2^J$ as

$$X_{t,T} = \sum_{p \in b} \sum_k \omega_{j_p, n_p, k; T} W_{j_p, n_p, k}(t) \xi_{j_p, n_p, k}, \quad (50)$$

where $\{\omega_{j_p, n_p, k; T}\}_{k \in 1, \dots, T}^{p \in b}$ are amplitude constants, and $\{\xi_{j_p, n_p, k}\}_{k \in 1, \dots, T}^{p \in b}$ are orthonormal random variables. For $p \in b$ and $z \in (0, 1)$, there exist functions $V_{j_p, n_p, k}(z)$ such that for each $p \in b$ a constant C_p exists, such that for each T ,

$$\sup_k |\omega_{j_p, n_p, k; T} - V_{j_p, n_p, k}(k/T)| \leq C_p/T, \quad (51)$$

where $\sum_{p \in b} C_p < \infty$. In addition, in a similar constraint to Equation (45), for each $p \in b$ the total variation norm of $V_{j_p, n_p}^2(z)$ is bounded above by M_p , where $\sum_{p \in b} M_p < \infty$. The locally stationary wavelet process is a particular form of the wavelet packet process, so the packet process allows for representation of a greater range of signals.

Similarly to the locally stationary wavelet process framework, the packet process also has a spectrum, in this case the evolutionary wavelet packet spectrum of a sequence $\{X_{t,T}\}_{t=1, \dots, T}$ is

$$S_p(z) = |V_{j_p, n_p}(z)|^2, \quad \text{for } p \in b, z \in (0, 1), \quad (52)$$

with respect to the wavelet packet basis b .

With this model Cardinali and Nason (2017) suggest cost functions such as penalised least squares as a means to choose a ‘best’ basis for an individual process. Such a basis may be selected to provide the most sparse representation of a series.

Wavelet packets and their two-dimensional extension have been used in a variety of applications, such as digital watermarking of images by Paquet et al. (2003), who make alterations of certain wavelet packet coefficients to embed a secret identifying key. Baseline drift of electrocardiograph signals were estimated and removed using a wavelet packet decomposition in Tinati and Mozaffary (2006), allowing for precise measurement of features of interest. Image textures were classified by Huang and Aviyente (2008), utilising the range of different features a packet analysis identifies, and combining them based on dependencies between coefficients at different levels.

2.2.3 Second-generation wavelets

Second-generation wavelets allow for dual multi-resolution analysis of data. This is more flexible than a standard multi-resolution analysis, as in the dual case there are two mutually orthogonal bases, that are not orthogonal themselves. Second-generation wavelets allow for analysis of more general data, such as data that is not be sampled on a regular grid. The dual analysis involves a pair of scaling functions, $(\varphi, \tilde{\varphi})$, and a pair of wavelet functions, $(\psi, \tilde{\psi})$. These functions satisfy the following conditions of Jawerth and Sweldens (1994):

$$(i) \int \tilde{\varphi}(x)\psi(x-n)dx = 0,$$

$$(ii) \int \varphi(x-n)\tilde{\psi}(x)dx = 0,$$

$$(iii) \int \tilde{\varphi}(x)\varphi(x-n)dx = \delta_{0,n},$$

$$(iv) \int \tilde{\psi}(x)\psi(x-n)dx = \delta_{0,n}.$$

The subspaces that these functions correspond to are denoted V_j and W_j for the scaling and wavelet functions and \tilde{V}_j and \tilde{W}_j for their duals. These subspaces satisfy the following biorthogonality criteria for $j, j' \in \mathbb{Z}$:

$$(i) \tilde{V}_j \perp W_j,$$

- (ii) $V_j \perp \widetilde{W}_j$,
- (iii) $\widetilde{W}_j \perp W_{j'}$ for $j \neq j'$.

As in the first-generation wavelet case of Definition 2.18, there is a scaling equation for second-generation wavelets.

Definition 2.28 (Dual wavelet scaling equation). The scaling equation of a dual wavelet function $\widetilde{\varphi}(x)$ is

$$\widetilde{\varphi}(x) = \sum_{k \in \mathbb{Z}} \widetilde{h}_k \sqrt{2} \widetilde{\varphi}(2x - k), \tag{53}$$

where $\widetilde{\mathbf{h}} = \{\widetilde{h}_k, k \in \mathbb{Z}\}$ is a vector of filter coefficients.

The dual wavelet function can also be represented as $\widetilde{\psi}(x) = \sum_{k \in \mathbb{Z}} \widetilde{g}_k \sqrt{2} \widetilde{\varphi}(2x - k)$, for filter $\widetilde{\mathbf{g}}$. Therefore the construction equations for the wavelet and dual wavelets are similar. The filters also satisfy similar equations to Equation (30) as

$$\begin{aligned} g_k &= (-1)^k \widetilde{h}_{1-k} \\ \widetilde{g}_k &= (-1)^k h_{1-k}. \end{aligned} \tag{54}$$

With a biorthogonal basis, two sets of scaling and wavelet coefficients are formed using the inner product, and a function $f(x) \in L_2(\mathbb{R})$ can be represented in two ways;

$$\begin{aligned} f(x) &= \sum_j \sum_k \langle f, \widetilde{\psi}_{j,k} \rangle \psi_{j,k} \\ f(x) &= \sum_j \sum_k \langle f, \psi_{j,k} \rangle \widetilde{\psi}_{j,k}. \end{aligned} \tag{55}$$

These representations highlight that the labelling of the wavelet and scaling functions and dual wavelet and scaling functions can be interchanged.

Similarly to first-generation wavelets, the second-generation wavelets can be used to form a spectral estimate, such as in the method of Knight et al. (2012) which is described in Section 3.2. As second-generation wavelets are no longer simply scaled and translated versions of the same function there are more options for constructing a spectral estimate. Second-generation wavelets are generally considered for analysis of series rather than construction, due to their connection to the (potentially) irregular sampling locations.

Example applications of second-generation wavelets include data compression of terrain

data via triangulated irregular networks by Pradhan et al. (2007), a three-dimensional dataset. Vasilyev and Bowman (2000) demonstrate the use of second-generation wavelets in an efficient algorithm for calculating spatial derivatives to solve partial differential equations. Second-generation wavelets were used by Bose et al. (2004) for image sequence superresolution, constructing a higher resolution image from a set of noisy lower resolution frames.

2.3 Bristol Traffic Data

To demonstrate the work in this thesis, a traffic example is used throughout. The Open Bristol Historic Journey Times data contains the speeds of cars on stretches of roads in Bristol, accessed from the Open Data Bristol website¹. At each of the locations, the speeds were recorded at irregular time intervals of between 5 and 30 minutes, so to turn this into a regularly spaced hourly time series, the mean of observations from each calendar hour was calculated. The cleaning of this data was completed for the Jean Golding Institute Bristol Traffic Data Competition, but the analysis is new for this thesis.

For this work, the hourly time series of average car speeds at 29 locations is selected, from 12am on 19/04/2016 to 8am on 11/05/2016, giving $T = 512$. Due to issues with recording, some observations are missing from this dataset, as shown in Figure 2. The 29 locations are described in Table 1.

¹<https://opendata.bristol.gov.uk/explore/dataset/historic-journey-times/information/>

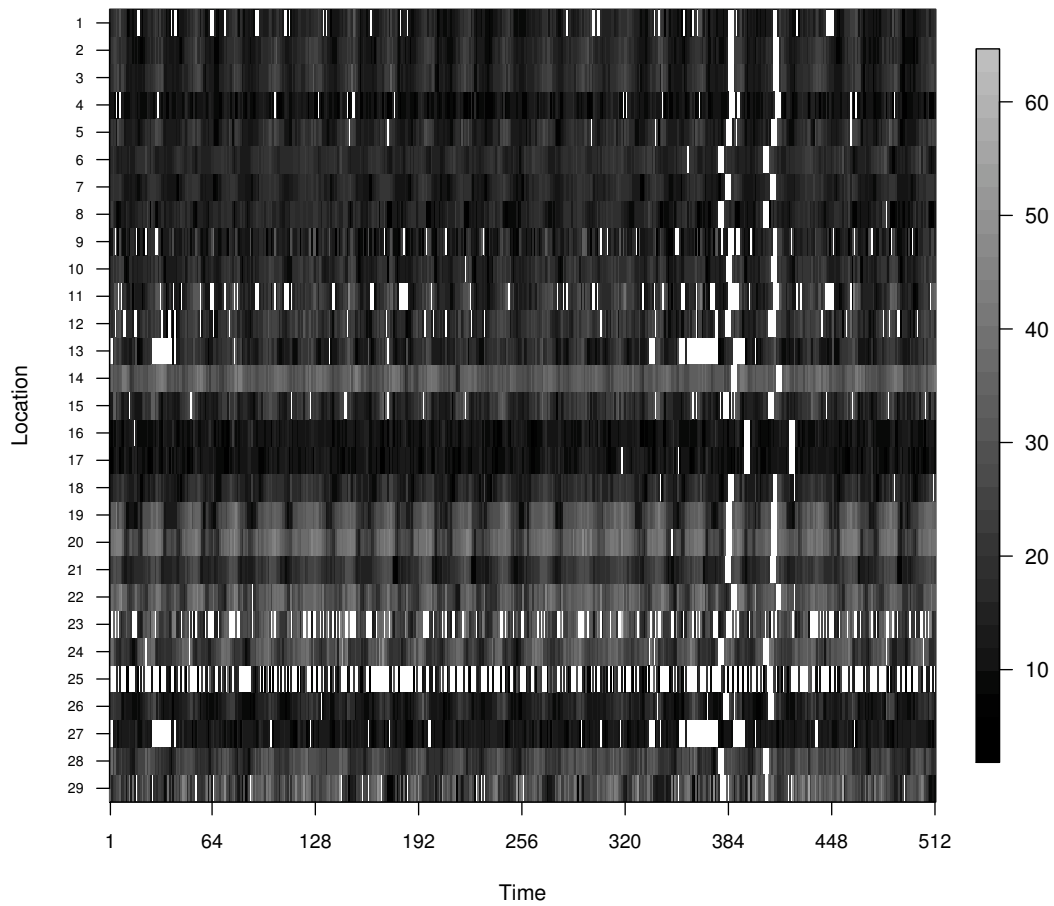


Figure 2: The Bristol Traffic dataset: average hourly car speeds from 12am 19/04/2016 to 8am 11/05/2016 at the 29 locations listed in Table 1. White space indicates missing observations.

Location number	Description
1	A38 Monks Pk IB to Stokes Croft IB
2	A432/Coombe OB to A432/Straits OB
3	A432/Straits IB to A432/Coombe IB
4	Bath Rd P&R IB to Bath Rd/ Three Lamps IB
5	Bath Rd P&R IB to Hartcliffe Way IB
6	Bath Rd/ Three Lamps IB to Victoria St NB
7	Bed Pde OB to Parson St/Bed Dwn
8	Bedminster Parade IB to Redcliffe Way West OB
9	Blackboy Hill IB to St Michaels IB
10	Church Rd IB to Old Mkt East IB
11	Gloucester Rd @ Monks Park Ave IB to M32 J2 IB
12	Hartcliffe Way IB to Parson St/Bed Dwn
13	Hartcliffe Way OB to Bath Rd P&R OB
14	Hotwell Rd OB to Portway OB
15	M32 J2 OB to Gloucester Rd @ Monks Park Ave OB
16	M32 Newfoundland Circus IB to Old Market St West WB
17	Old Market St West EB to M32 Newfoundland Circus OB
18	Old Mkt East OB to Church Rd OB
19	Parson St/Parson SB to Hartcliffe Wy OB
20	Parson St/W'Stoke to Portway OB
21	Parson St/West to Bed Pde IB
22	Portway IB to Hotwell Rd IB
23	Portway IB to Parson St/West
24	Redcliffe Way West IB to Coronation Rd OB
25	St Michaels OB to Blackboy Hill OB
26	Stokes Croft OB to A38 Monks Pk OB
27	Three Lamps OB to Bath Rd P&R OB
28	Victoria St SB to Bath Rd/ Three Lamps OB
29	Victoria St SB to Old Market St East OB

Table 1: Bristol Traffic locations and their corresponding descriptions from the Open Data Bristol database.

3 Stationarity testing with missing observations

In this section, the developed method is joint work with Marina Knight, Guy Nason, and Matthew Nunes. My contribution to this material includes development of the test from a prototype, and evaluating its performance using simulations. The Bristol Traffic data application is new and completely my own work.

3.1 Introduction and additional literature review

In the framework of time series analysis, a common question of interest is whether the underlying structure of a series changes over time. Indeed, it is a question to answer before continuing assessment or forecasting of a series, as a process with a changing generating mechanism requires different methods compared to a stationary process. Testing for second-order stationarity, as defined in Section 2.1, enables us to check for underlying structural changes over time.

In this chapter, the time series is assumed to suffer from missing observations. In practice, this is a common situation, however few time series tools are designed for use with this type of data. The developed test is designed for application on time series with missing observations under the assumption of a stationary mean. A review of available tests of second-order stationarity is presented in Section 3.1.1, and missing data tools are discussed in Section 3.1.2. Section 3.2 contains the method and underlying techniques involved in our stationarity test for missing data, including use of second-generation wavelets and estimating the evolutionary wavelet spectrum using a lifting algorithm. Our new test is compared using simulations in Section 3.3 and exhibited with two real data examples in Section 3.4.

3.1.1 Tests of second-order stationarity

Priestley and Subba Rao (1969) provided an early test of second-order stationarity. For their test an analysis of variance procedure is performed on the log-spectrum evaluated at different time positions, finding differences (if present) in the spectrum at set frequencies over time. Fourier approaches were also taken by Hurd and Gerr (1991) and Dwivedi

and Subba Rao (2011), who compared functions of Fourier ordinates, which should be approximately uncorrelated under the null hypothesis. An advantage of the Dwivedi and Subba Rao (2011) test over the early Priestley and Subba Rao (1969) test is that it has a theoretical basis for non-Gaussian distributed series, as shown by simulations in Nason (2013) the performance of the Priestley and Subba Rao (1969) test is poor for series with heavy tailed innovations.

A time domain approach was provided by Jin et al. (2015), where Walsh functions were used to estimate the covariance on systematic subsamples compared to global estimates of the covariance, giving a powerful test. Their test benefits from being fully non-parametric, however it requires the choice of tuning parameters including the number of covariances to test and the number of subsamples to use.

The locally stationary framework of Dahlhaus (1997) was used by von Sachs and Neumann (2000) to construct a time-varying spectrum, which was then assessed for constancy using Haar wavelets. Paparoditis (2009) also used the concept of local stationarity, and provided a test comparing the deviation of local spectral densities (calculated within a tapered window). The use of local spectra allow these tests to locate areas of non-stationarity, but can require a higher computational cost. The L^2 distance was used by Dette et al. (2011) to compare the estimated density of a locally stationary process, with that of a stationary process, in a method similar to that of Živanovic and Gardner (1991).

Other works using wavelets include Nason (2013) who used the locally stationary wavelet process of Nason et al. (2000), defined in Section 2.2, and tested the constancy of the wavelet periodogram (Definition 2.23) using Haar wavelets. Their test statistic at periodogram scale $j \in \{1, \dots, J\}$, Haar scale $i \in \{1, \dots, J\}$, and location $p \in \{1, \dots, 2^i - 1\}$, is of the form

$$T_{i,p}^{(j)} = v_{i,p}^{(j)} \left\{ \hat{\sigma}_{i,p}^{(j)} \right\}^{-1}, \quad (56)$$

where $v_{i,p}^{(j)}$ is the Haar coefficient and $\hat{\sigma}_{i,p}^{(j)}$ is an estimator of the coefficient variance given by

$$\begin{aligned} v_{i,p}^{(j)} &= 2^{-i/2} \left(\sum_{r=0}^{2^{i-1}-1} I_{j,2^i p-r} - \sum_{q=2^{i-1}}^{2^i-1} I_{j,2^i p-q} \right) \\ \hat{\sigma}_{i,p}^{(j)} &= T^{-1} \left(2 \sum_t I_{j,t}^2 \right)^{1/2}. \end{aligned} \quad (57)$$

This test statistic is then compared to critical values of the normal distribution, and multiple tests are carried out across j, i, p to test for stationarity at different periodogram scales, test scales, and test positions, respectively. The p -values of these tests are then adjusted to account for the multiple testing. In their paper this test was shown to perform well for series with Gaussian and heavy-tailed innovations in simulations, but showed high size properties for a stationary AR model with negative parameter.

Cho (2016a) used unsystematic sampling to compare the constancy of the wavelet periodogram over disjoint intervals. For disjoint intervals $[s_p, e_p], [s_q, e_q]$, with respective lengths n_p and n_q , their statistic at scale $j \in \{1, \dots, J\}$ is given by

$$T_{p,q}^{(j)} = \sqrt{(n_p n_q)(n_p + n_q)^{-1}} \left(n_p^{-1} \sum_{k=s_p}^{e_p} I_{j,k} - n_q^{-1} \sum_{k=s_q}^{e_q} I_{j,k} \right). \quad (58)$$

The overall test statistic is calculated as the maximum of $T_{p,q}^{(j)}$ over pairs of disjoint intervals and scales, divided by estimates of the standard deviation. This test statistic is also compared to the normal distribution to assess significance. In simulations in Cho (2016a) their test was shown to perform well with a range of innovation distributions, but featured lower power than the test of Jin et al. (2015), especially for smaller sample sizes such as $T = 256$.

The method of Nason (2013) was extended by Cardinali and Nason (2018) to include wavelet packets for additional power. The test statistic is calculated on the wavelet packet periodogram,

$$I_{\ell,k} = \left(\sum_{t=1}^T X_t \psi_{\ell,k-t} \right)^2, \quad (59)$$

where $\psi_{\ell,k} = W_{\ell,k}$ in Definition 2.27, and ℓ is a vector of length 2 representing the indices in a wavelet packet basis.

These tests described in this section are designed for many different settings, and have advantages in different scenarios, however none are designed for use with data featuring missing observations.

3.1.2 Statistical tools for missing data

Some methods are available for use on time series data with missing observations, such as autocovariance estimation, Chatfield (2004), page 270. Unit-root tests, such as those by

Toda and McKenzie (1999) and Busetti and Taylor (2005) can be used to indicate deviations from stationarity of the mean in the case of missing observations. However, often the suggestion for analysing data with missing values is to impute, rather than carry out methods with the incomplete data directly. If the data comes from an ARMA model then Ferreiro (1987) offers imputation methods such as using the recorded values to estimate the parameters of an ARMA model and then imputing missing values based on these fitted parameters and observed values. Another method they describe is a pseudo estimation-maximisation method, which iterates between imputing missing values based upon a fitted model as above, and estimating the model parameters via maximum likelihood based upon the observed and imputed values. This iterative process continues until convergence. For non-stationary processes imputation techniques are offered by Hong and Chen (2003) and Zgheib et al. (2006). Hong and Chen (2003) demonstrated non-parametric imputation of a general non-stationary process using radial basis neural networks, whilst the Zgheib et al. (2006) method for non-stationary AR processes uses pseudo linear recursive least squares in an on-line setting.

Imputation is often considered the first step in the analysis of data with missing observations, as after this standard time series methods can be applied. The structure of the data is important in choosing an imputation method, and as shown from the references above, this includes knowledge of whether data is stationary. Therefore choosing the most suitable technique for further analysis may require the ability to test for stationarity before imputation has been applied.

3.2 Description of our method

Our stationarity test involves modelling the process as a locally stationary wavelet process, with observations missing at random. Formally, there exists an underlying locally stationary wavelet process, $\{X_t\}_{t=1}^T$, which is observed at random time points $\{t_k\}_{k=1}^n$. These n observation locations are assumed to be sampled uniformly at random from the total set of observation locations, $\{1, \dots, T\}$. This stationarity test is designed to test whether the observed process, $\{X_{t_k}\}_{k=1}^n$, is second-order stationary, against the alternative hypothesis that it is a locally stationary wavelet process with changing spectrum, without imputing the missing values. For our test we use the method of spectrum estimation described by Knight et al. (2012) as it allows for missing observations. An overview of the techniques

involved is included below, and for full details see the references herein.

3.2.1 Estimating the wavelet spectrum

In this section we set out the method of calculating the wavelet spectrum which mostly follows that of Knight et al. (2012). As the data feature missingness, the standard wavelet transform is unsuitable, and instead the second-generation wavelet transform as introduced by Sweldens (1998) is used.

Second-generation wavelets

We use second-generation wavelets (see Section 2.2.3) constructed using lifting schemes, which are suited for irregularly sampled data, including data with missing observations. Whilst first-generation wavelets are scalings and translations of a mother wavelet, second-generation wavelets are more complicated as they are adapted to the irregularity of the data.

Second-generation wavelets are not just motivated by irregularly sampled data, Sweldens (1998) also cites the applications of their use on solutions of PDEs, manifolds, and use on spaces with weighted or non-translation invariant measures. There are many properties that second-generation wavelets share with first-generation wavelets, included in those set out in Sweldens (1998) are locality in space and frequency, the Riesz basis they form over $L_2(\mathbb{R})$, and their use for multiresolution analysis. The second-generation wavelet transform can be calculated in linear time with recursive application of filters on the wavelet coefficients. A major difference of second-generation wavelets compared to first-generation is that the filter coefficients of the second-generation transform are different for every coefficient, allowing the flexibility for its use on non-standard domains.

The non-decimated lifting transform

A second-generation wavelet transform called the non-decimated lifting transform (NLT) was defined in Knight and Nason (2009). This method lifts one coefficient at a time, similar to Jansen et al. (2006) and Jansen et al. (2009). At every step of the lifting algorithm, three actions are performed: split, predict, and update. The split step refers to selecting an observation to lift. This observation is then predicted by its neighbouring values, and the error in this prediction recorded as the lifted coefficient. This coefficient

has an associated length which arises from the time interval this prediction is performed over. Then the observations at the neighbouring points to the lifted location are updated, so that the mean of the observations stays constant as points are removed.

For data with n observations, the split, predict, update steps are completed $(n - L)$ times, where L is the chosen primary resolution level. Following previous work, we adopt the convention that the procedure is first carried out at stage n , then stage $n - 1$ and so on. Let $\{\tilde{X}_{n+1,t_k}\} = \{X_{t_k}\}$ denote the scaling coefficients before the lifting transform is performed, and \tilde{X}_{r,t_k} denote the scaling coefficient at t_k at stage r , after $(n - r + 1)$ applications of the split, predict and update steps. Let the time positions of scaling coefficients at stage r be denoted \mathcal{S}_r , and the positions of the detail coefficients given by \mathcal{D}_r . To perform the algorithm at stage $r - 1$, a time position, $t_k \in \mathcal{S}_r$ is chosen in the split step, and the neighbouring values of the scaling coefficients are required for prediction. We denote these neighbouring positions as $\mathcal{N}_r(t_k)$, where the subscript of stage highlights that as the algorithm is applied, the number of scaling coefficients reduces so the position of neighbouring scaling coefficients is stage-dependent. Usually, there will be two positions within this set, but when a coefficient at the boundary of the scaling coefficients is chosen there will only be one neighbouring position. The prediction is carried out using regression over values at $\mathcal{N}_r(t_k)$, and the detail coefficient is the difference between the value and prediction;

$$d_{t_k} = \tilde{X}_{r,t_k} - \sum_{t \in \mathcal{N}_r(t_k)} a_t^{(r-1)} \tilde{X}_{r,t_k}, \quad (60)$$

where $a_t^{(r-1)}$ are the regression weights at stage $r - 1$, associated with the neighbourhood of t_k . A lifting length, l_{t_k} , is associated with this operation, recording the interval length that the detail coefficient was calculated over.

The scaling coefficients at indices $\mathcal{N}_r(t_k)$ are then updated in order to keep the mean of the scaling coefficients constant as the scaling coefficient at t_k is removed. This involves weights $b_t^{(r-1)}$;

$$\begin{aligned} \tilde{X}_{r-1,t} &= \tilde{X}_{r,t} + b_t^{(r-1)} d_{t_k} && \text{for } t \in \mathcal{N}_r(t_k), \\ \tilde{X}_{r-1,t} &= \tilde{X}_{r,t} && \text{otherwise.} \end{aligned} \quad (61)$$

The scaling coefficient at t_k is removed, leaving $(r - 1)$ scaling coefficients and $(n - r + 1)$ detail coefficients at stage $r - 1$ after $n - r + 1$ applications of the split, predict and update

steps.

This lifting algorithm can be described using a linear (matrix) transformation or a dual basis scheme. The dual basis at stage r is denoted $\varphi_r^* = \{\psi_k^*, \varphi_{r,m}^*\}_{k \in \mathcal{D}_r}^{m \in \mathcal{S}_r}$. The basis is constructed recursively, with the stage $r - 1$ basis functions given by the following expressions;

$$\begin{aligned} \psi_{t_k}^* &= \varphi_{r,t_k}^* - \sum_{t \in \mathcal{N}_r(t_k)} a_t^r \varphi_{r,t}^*, \\ \varphi_{r-1,t}^* &= \varphi_{r,t}^* + b_t^r \psi_{t_k}^* \quad \text{for } t \in \mathcal{N}_r(t_k), \\ \varphi_{r-1,t}^* &= \varphi_{r,t}^* \quad \text{for } t \in \mathcal{S}_{r-1} \setminus \mathcal{N}_r(t_k). \end{aligned} \tag{62}$$

Using these wavelet and scaling functions we have detail coefficients of the form $d_{t_k} = \sum_{s=1}^n X_{t_s} \psi_{t_k}^*(t_s)$, where $\psi_{t_k}^*(t_s)$ is the second-generation wavelet function associated with position t_k , and evaluated at t_s . As these are second-generation wavelet functions, constructed using a lifting scheme, the wavelet functions are not dilations and translations of each other. The notion of scale is no longer discrete, but instead the interval lengths are used as an analogue of scale. The lengths are a continuous quantity, but are usually mapped to a discrete artificial scale representation. The artificial scales are used to evaluate second-generation wavelet quantities in a similar manner to first-generation wavelet quantities.

Through repeated application of the split step, an ordering of the removal of observations is formed which is described as a trajectory, denoted \mathcal{T}^α . For each trajectory, the second-generation transform can be written as a matrix operation on the observations; the $(n - L)$ details for trajectory α , $\mathbf{D}^\alpha = (d_{t_1}^\alpha, \dots, d_{t_n}^\alpha)'$, can be written as $\mathbf{D}^\alpha = R^\alpha \mathbf{X}$, where R^α is the $(n - L) \times n$ lifting matrix associated with trajectory α , with rows given by the second-generation wavelet functions, and \mathbf{X} denotes the column vector of observations.

In the NLT this algorithm is calculated over p different trajectories, $\{\mathcal{T}^\alpha\}_{\alpha=1}^p$. The NLT therefore results in a collection of scaling coefficients, wavelet coefficients and associated lifted lengths for each time point. The length associated with detail coefficient $d_{t_k}^\alpha$, lifted from point t_k on trajectory α , is denoted $l_{t_k}^\alpha$. Knight and Nason (2009) converted these lengths into artificial dyadic scales by assigning half of the coefficients to the finest level, those with the lowest lifted lengths, the second scale contains the next quarter of coefficients ordered by lifted length, and so on.

The evolutionary wavelet spectrum calculated by Knight et al. (2012) used the NLT over many trajectories to form an estimated spectrum. To convert between the continuum of lengths and discrete artificial scales a linear smoother was applied to the squared coefficients with respect to the \log_2 values of the lengths.

3.2.2 LiftToS

Our lifting test of stationarity (LiftToS) aims to assess the estimated wavelet spectrum for constancy over time. For a mean-zero locally stationary wavelet process $\{X_t\}_{t=1}^T$, observed at points $\{t_k\}_{k=1}^n$, the LiftToS test statistic is constructed as follows, using the notation from the previous section:

1. Apply the non-decimated lifting transform of Knight and Nason (2009) with P trajectories, $\{\mathcal{T}^\alpha\}_{\alpha=1}^P$. Normalise the details, \mathbf{D}^α , to remove the additional variance induced by the transform by dividing by $\sqrt{\text{diag}(R^\alpha R^{\alpha'})}$. This gives normalised details $\mathcal{D}^\alpha = (\tilde{d}_{t_1}^\alpha, \dots, \tilde{d}_{t_n}^\alpha) = \mathbf{D}^\alpha / \sqrt{\text{diag}(R^\alpha R^{\alpha'})}$.
2. For each time position t_k , combine the details from all P trajectories. This is done by first taking the mean of the squared details at each t_k that have the same associated length from the wavelet transform; for each t_k and unique length $l_{t_k}^\beta$,

$$\bar{d}_{t_k, l_{t_k}^\beta} = \left\{ \sum_{\alpha=1}^P \mathbb{I}(l_{t_k}^\alpha = l_{t_k}^\beta) \right\}^{-1} \sum_{\alpha=1}^P \left(\tilde{d}_{t_k}^\alpha \right)^2 \mathbb{I}(l_{t_k}^\alpha = l_{t_k}^\beta). \quad (63)$$

3. Convert these into q equally-spaced artificial scales, using the lengths associated with the lifting transform. Let $\ell_{t_k}^\beta = \log_2(l_{t_k}^\beta)$, for all unique lengths. The lengths are converted to artificial scales, where scale $j \in \{1, \dots, q\}$ is located at length $\ell_j = a + (j-1)(b-a)(q-1)^{-1}$, where $a = \min_{t_k, \beta}(\ell_{t_k}^\beta)$ and $b = \max_{t_k, \beta}(\ell_{t_k}^\beta)$. The spectral estimate is determined using linear interpolation of the \bar{d} values with respect to the q scales. Let l and ℓ be used interchangeably in the indices, so $\bar{d}_{t_k, \ell_{t_k}^\beta} = \bar{d}_{t_k, l_{t_k}^\beta}$. Then for each t_k , the value at scale j is $\tilde{I}_{t_k, j} = \bar{d}_{t_k, \ell_j}$ if $\ell_j \in \{\ell_{t_k}^\beta\}$, the $\bar{d}_{t_k, l_{t_k}^\beta}$ value with closest length to ℓ_j when $j = \{1, q\}$, or linear interpolation for a central scale, as described below. To avoid notational clutter, let $\ell_-, \ell_+ \in \{\ell_{t_k}^\beta\}$ be such that $\ell_- < \ell_j < \ell_+$, $\ell_- = \text{argmin}_\beta(\ell_j - \ell_{t_k}^\beta)$, and $\ell_+ = \text{argmin}_\beta(\ell_{t_k}^\beta - \ell_j)$. The

linear interpolation used to estimate the spectrum at time t_k and artificial scale j is

$$\tilde{I}_{t_k,j} = \bar{d}_{t_k,\ell_-} + (\ell_j - \ell_-) \frac{\bar{d}_{t_k,\ell_+} - \bar{d}_{t_k,\ell_-}}{\ell_+ - \ell_-}, \quad (64)$$

where we note that ℓ_- and ℓ_+ are different for each t_k and j . This gives a spectral estimate $\left\{ \tilde{I}_{t_k,j} \right\}_{t_k=t_1, \dots, t_n}^{j=1, \dots, q}$.

4. Smooth the spectral estimate over time using a running mean smoother with bandwidth M , $\bar{I}_{t_k,j} = |\mathcal{M}_{t_k}|^{-1} \sum_{m \in \mathcal{M}_{t_k}} \tilde{I}_{m,j}$, where $\mathcal{M}_{t_k} = \{t_s : |t_s - t_k| \leq M\}$.
5. Divide each scale of the spectral estimate by its sample mean,

$$\check{I}_{t_k,j} = n \bar{I}_{t_k,j} \left(\sum_{s=1}^n \bar{I}_{t_s,j} \right)^{-1}. \quad (65)$$

This allows for comparison of the variance across the scales, as for a stationary distributed process the variances at each scale should be approximately equal due to the chi-squared nature of the spectrum.

6. At each artificial scale calculate the sample variance, the average of these values forms the test statistic; $Q = q^{-1} \sum_{j=1}^q \text{var}(\check{I}_{\cdot,j})$, where $\check{I}_{\cdot,j}$ denotes the j th column of the matrix \check{I} .

Under the null hypothesis, the spectrum is constant so we expect low values for the variance of the estimated spectrum at each scale. Hence, significant departures of this test statistic from zero will cause us to reject the null hypothesis.

The spectral estimate generated using steps 1-3 above is similar to that of Knight et al. (2012), with normalisation added here in step 1 as the variance of the spectral estimate is key to our test, and linear interpolation chosen as the linear smoother to map from lengths to artificial scales.

Alternative test statistics, such as the maximum variance over the scales could also be used here, as well as alternative methods of smoothing and constructing the wavelet spectrum estimate. The choices of parameter values for P , q and M can be chosen based upon the data. As discussed by Knight et al. (2012), the number of unique trajectories is $n!$, so in general a smaller value of P will be chosen for computational reasons. P should be chosen to be as large as computationally feasible, as more trajectories corresponds to more detail

coefficients to estimate the spectrum with. The number of artificial scales, q , could be chosen to match the first generation wavelet case, $q = \log_2(T)$, or a higher value used if greater scale resolution is desired. The smoothing parameter, M , could be chosen using a data-driven approach, or set based upon the sample length.

3.2.3 Constructing bootstrap test statistics using circulant embedding

A bootstrap approach is chosen to evaluate how extreme an observed test statistic is. The circulant embedding method of Percival and Constantine (2006) is used to generate stationary samples with similar spectral content to the original series. Their method can be described using matrices, and is based upon the estimated covariance of the input time series. For a stationary series of length T , let $\{\gamma_p\}_{p=0}^T$ be the values of the autocovariance function. Define the first row of the $2T \times 2T$ circulant matrix C to be $(\gamma_0, \gamma_1, \dots, \gamma_{T-1}, \gamma_T, \gamma_{T-1}, \dots, \gamma_1, \gamma_0)$, then the matrix is filled with rows such that each has the same values as the previous row with entries shifted to the right by one place. The matrix C can be written as $C = F\Lambda F^H$, where Λ is a diagonal matrix, F is a $2T \times 2T$ matrix with entries $F_{j,k} = \left(\sqrt{2T}\right)^{-1} \exp(-i2\pi jk/2T)$, and $F^H = \overline{F}^T$, the Hermitian transpose of F formed by taking the transpose and element-wise complex conjugation. A bootstrap series is then generated as $W = F\Lambda^{1/2}Z$, where Z is a vector of uncorrelated complex-valued Gaussian random variables. Percival and Constantine (2006) showed that the first T elements of vector W form a Gaussian process with the same mean and (stationary) covariance structure as the input time series.

This method was originally used for complete data, so for data with missing observations it requires adaption. The missing values are first set to zero (the mean of the series) before the circulant embedding step. Setting the values to the mean assumes no additional structure for the original process, which is preferable to using a technique designed for a particular generating model, for example the method of Ferreiro (1987), as the generating model is unknown. The surrogate series generated by the circulant embedding algorithm then have the values removed at locations where there were missing values in the original, resulting in series of the form $\left\{\tilde{X}_{t_k}\right\}_{k=1}^n$, and the test statistic is calculated on each of these surrogate series. For a nominal size $\lambda\%$, the null hypothesis of stationarity is rejected if the original test statistic is greater than $(100 - \lambda)\%$ of the bootstrap test statistics. As with the choice of P , we note that the number of bootstrap simulations may be dictated

by computational power. For each bootstrap series, P trajectories are used to estimate the spectrum, so the computation time is multiplicative in the number of trajectories and bootstrap series.

3.2.4 Summary of LiftToS

The following algorithm presents a summary the steps included in LiftToS, as described in the previous sections.

1. Use the NLT of Knight and Nason (2009) over P trajectories to generate lifting details and lengths
2. Combine the details to form a spectral estimate with q artificial scales, and smooth spectrum over time
3. Calculate variance of the smoothed spectrum at each scale, and average these to form the test statistic
4. Generate B bootstrap time series using circulant embedding of Percival and Constantine (2006) and perform steps 1-3 on each of these series to give B bootstrap test statistics
5. Calculate the significance of the original test statistic by comparing it to the B bootstrap statistics.

3.2.5 Comparisons to stationarity tests for complete data

Whilst our test is designed for use on data with missing observations, it has similarities with methods created for use on complete data, described in Section 3.1.1. The alternative hypothesis of our test is a locally stationary wavelet process with time varying spectrum, the same alternative as the tests of Nason (2013), Cho (2016a), and Cardinali and Nason (2018). However, rather than comparing sections of the wavelet periodogram using sub-samples or Haar wavelets, in our test the variance is calculated over the entire time length of the periodogram. Indeed, our statistic Q is similar to the sample version of the wavelet packet measure of Cardinali and Nason (2018), Equation (22).

Bootstrap samples were also used by Cardinali and Nason (2018) to assess the significance of the test statistic, however their samples were created using phase randomisation rather

than circulant embedding, for details see Percival and Constantine (2006). Cho (2016a) also used a bootstrapping method for their test, generating samples of AR processes with increasing time lags for increasing T in order to estimate the standard deviation of the coefficients.

3.3 Simulations

This section presents simulation results that compare LiftToS with other stationarity tests. As discussed in Section 3.1.1, there is no existing test of stationarity that can handle data with missing observations, hence the tests we compare to are adapted forms of existing tests. For ease of comparison, we use the following models from Nason (2013), also used by Cho (2016a) and Cardinali and Nason (2018). Models S1-7 are stationary processes, used to check size calibration, and P1-4 are non-stationary processes which are used for power comparison; both sets of processes are directly quoted from Nason (2013).

S1 Independent, identically distributed (iid) standard normal.

S2 AR(1) model with AR parameter of 0.9 with standard normal innovations.

S3 As **S2** but with AR parameter of -0.9.

S4 MA(1) model with parameter of 0.8 with standard normal innovations.

S5 As **S4** but with parameter of -0.8.

S6 ARMA(1,2) with AR parameter of -0.4, MA parameters of (-0.8,0.4), with standard normal innovations.

S7 AR(2) with AR parameters of $\alpha_1 = 1.385929$ and $\alpha_2 = -0.9604$, with standard normal innovations. The roots associated with the auxiliary equation, see Chatfield (2004), are $\beta_1 = \overline{\beta_2} = 0.98e^{i\pi/4}$. The process is stationary, but close to the ‘unit root’: a ‘rough’ stochastic process with spectral peak near $\pi/4$.

P1 Time-varying AR model $X_t = \alpha_t X_{t-1} + \epsilon_t$ with iid standard normal innovations and the AR parameter evolving linearly from 0.9 to -0.9 over the length of the series.

P2 A locally stationary wavelet process based on Haar wavelets with spectrum $S_j(z) = 0$ for $j > 1$ and $S_1(z) = \frac{1}{4} - (z - \frac{1}{2})^2$ for $z \in (0, 1)$. This process is a time varying moving average process.

P3 A locally stationary wavelet process based on Haar wavelets with spectrum $S_j(z) = 0$ for $j > 2$ and $S_1(z)$ as for P2 and $S_2(z) = S_1(z + \frac{1}{2})$ using periodic boundaries (for the construction of the spectrum only).

P4 A locally stationary wavelet process based on Haar wavelets with spectrum $S_j(z) = 0$ for $j = 2, j > 4$ and $S_1(z) = \exp\{-64(z - \frac{1}{2})^2\}$, $S_3(z) = S_1(z - \frac{1}{4})$, $S_4(z) = S_1(z + \frac{1}{4})$ again assuming periodic boundaries.

The simulated series were generated using `auto.arima` from the `forecast` package, Hyndman and Khandakar (2008), and `LSWsim` from the `wavethresh` package, Nason (2016b). An example manual entry for the software used to generate our test statistics is included in Appendix A. The code utilises the `fwtnpperm` function from the `nlt` package to carry out the NLT, Knight and Nunes (2012), and `surrogate` from the `fractal` package to generate bootstrap series, Constantine and Percival (2014). It is anticipated that our function will be added to the existing R package `nlt` in due course.

Throughout the simulations $P = 100$ trajectories were used for the non-decimated lifting transform, and 200 bootstrap simulations were used for the p -values of the LiftToS. The other parameters for the test were number of artificial scales $q = \log_2(T)$, smoothing bandwidth $M = \text{floor}(\sqrt{n})$, and primary resolution level $L = 2$. This is the resolution level recommended for applications by Nunes and Nason (2005), and was also used by Knight and Nason (2009).

For each model, 100 complete realisations were sampled and observations were selected at random for removal. For comparison we include results from Cho (2016a), where results of the best performing method in their tests on complete data is reproduced for each sample size. These best performing methods have high power and size close to the nominal 5%, and were the Jin et al. (2015) model, labelled JWW, for $T = 256$, and for $T = 512$ the Cho (2016a) test had the best size properties, with JWW performing best in terms of power.

For the smaller sample size, Table 2 shows that LiftToS performs well for up to 10% data loss for most models. By 20% missing data the size is large for several of the stationary models. However, in the case of no missing observations, LiftToS outperforms the JWW test in terms of overall size proximity to 5% and has better power for model P2, but not P3. Therefore, LiftToS performs well even against tests that are designed for use with equally-spaced observations.

Table 3 shows similar LiftToS performance for the higher sample size of $T = 512$. The size for S7 is consistently high, this may be due to the model being close to unit-root. LiftToS is the only test to achieve 100% power for P1, P2, and P4, this occurs even when 10% of the data are missing. Again, LiftToS has lower power for the model P3, this may be because the variance in the spectrum is split over two neighbouring scales, which may become combined in the estimated spectrum resulting in flatter values at the artificial scales than the original. LiftToS performs better than the JWW test on complete data in terms of size properties but is worse for power, and vice-versa with the Cho test. Hence, this test works comparatively to the best performing tests available on complete data.

The test properties are also compared to using a test with a ‘work around’ to cope with the missing data. The original test is implemented using the `unsystation` package, Cho (2016b). Table 4 shows the performance of the Cho test with naive adaptations to allow for its use in a missing data setting, with LiftToS results for the same sample size (reiterated from Table 2). One adaptation is to set all missing values to the mean of the series, and another is to ignore the unequally-spaced nature of the observations and to treat the data as complete data of length n . The size properties of the adapted tests are poor, particularly for model S3 (an AR model with negative parameter). Removing the missing values results in a lower-powered test than LiftToS for P1, P2, and P4 and a similarly low-powered test for P3. Setting the values to the mean first gives a surprisingly high-powered test for P2 and P3, indeed this has higher power than the original Cho (2016a) test for complete data. Although the power for this adaptation is high, it also has sizes of up to 54%, compared to LiftToS sizes up to 20% for the same sample size. Therefore these naive adaptations have poor size properties, and turn a powerful, well-calibrated test for complete data into unstable tests for data with missing observations.

3.4 Examples

3.4.1 Air Quality

The data used in this section are taken from the UK Department for Environment, Food and Rural Affairs UK-AIR Data Archive². The time series is average PM-10 concentrations, measured daily. PM-10 stands for particulate matter that is smaller than 10 microns

²<https://uk-air.defra.gov.uk/>

Model	JWW	LiftTOS			
	0% missing	0%	5%	10%	20%
S1	9	2	4	5	3
S2	4	6	1	5	4
S3	3	7	7	8	17
S4	6	7	1	3	10
S5	4	3	4	6	7
S6	0	6	8	14	8
S7	10	7	9	20	18
P1	100	100	100	99	97
P2	53	94	85	82	72
P3	33	16	15	7	4
P4	100	100	99	100	100

Table 2: Empirical size and power (%) calculated using 100 simulations and 200 bootstrap simulations, $T = 256$ and 5% nominal size. JWW column is from Cho (2016a), Tables 1 and 4.

Model	JWW	Cho	LiftTOS			
	0% missing	0%	0%	5%	10%	20%
S1	7	5	2	3	2	2
S2	5	3	7	5	5	4
S3	4	4	6	7	6	17
S4	4	2	5	5	1	8
S5	9	4	5	6	3	6
S6	4	5	5	7	7	17
S7	11	5	8	16	21	22
P1	100	100	100	100	100	100
P2	96	53	100	100	100	98
P3	81	17	23	29	14	14
P4	100	100	100	100	100	100

Table 3: Empirical size and power (%) calculated using 100 simulations and 200 bootstrap simulations, $T = 512$ and 5% nominal size. JWW and Cho columns are from Cho (2016a), Tables 1 and 4.

in diameter. The series we use is from the Bristol Centre for air monitoring, and features missing observations. In this example and the next some observations are missing in a block, rather than missing at uniformly sampled positions so to carry out the test on these examples it is assumed that the missing positions are independent of the missing data values. As the spectral analysis method holds for irregularly spaced time series it is expected that the test will perform similarly on data with missing observations in blocks, however future work includes checking the performance of the bootstrapping method on such series as the padded series would feature blocks of zero values.

To account for non-stationarities in the mean, the first-order differenced data is used.

Model	Mean, then Cho			Removed, then Cho			LiftToS		
	5% missing	10%	20%	5%	10%	20%	5%	10%	20%
S1	2	7	4	3	3	4	4	5	3
S2	45	51	54	5	4	6	1	5	4
S3	39	54	46	75	84	72	7	8	17
S4	3	7	8	5	2	3	1	3	10
S5	2	3	10	7	11	6	4	6	7
S6	11	14	14	27	38	21	8	14	8
S7	4	9	13	2	6	10	9	20	18
P1	51	52	49	56	42	41	100	99	97
P2	85	81	88	9	5	9	85	82	72
P3	81	80	82	10	7	11	15	7	4
P4	93	95	94	98	96	93	99	100	100

Table 4: Empirical size and power (%) calculated using 100 simulations, $T = 256$ and 5% nominal size, where the missing values in the series are replaced with their mean, or removed before using the test by Cho (2016a). LiftToS results are replicated from Table 2.

This differenced data is plotted in Figure 3, where changes in second-order properties such as the variance can be seen. The data is of length $T = 256$, with approximately 8% missingness. We test the differenced PM-10 series for stationarity using our new method from Section 3.2. The parameters are set to those used in Section 3.3. All 200 bootstrap test statistics were less extreme than the PM-10 test statistic, so the null hypothesis of stationarity was rejected. Figure 4 shows the estimated evolutionary wavelet spectrum for this example. The increased variance at around observation 200, that is noticeable in the time domain plot in Figure 3, can also be seen with lighter-coloured blocks in the wavelet domain. The further detail given by the plot in Figure 4 shows that the departure from stationarity is present at all artificial scales.

3.4.2 Bristol Traffic

The Bristol Traffic dataset, introduced in Section 2.3, is used here in univariate form. The first of the locations has been selected, which describes traffic speed on the A38 between Monks Park and Stokes Croft, heading towards the centre of the city. The selected time series consists of hourly average car speeds from 12am on 19/04/2016, to 8am on 11/05/2016. This data is of length 512, with 47 missing observations as shown in Figure 5. As there is clear daily seasonality in the data, with period 24, this is removed before testing for stationarity. To do this, the function `decompose` is used on the data with the mean values imputed into the gaps, R Core Team (2018). This function splits the time series into the three components trend, seasonal, and residuals. Although we test the first

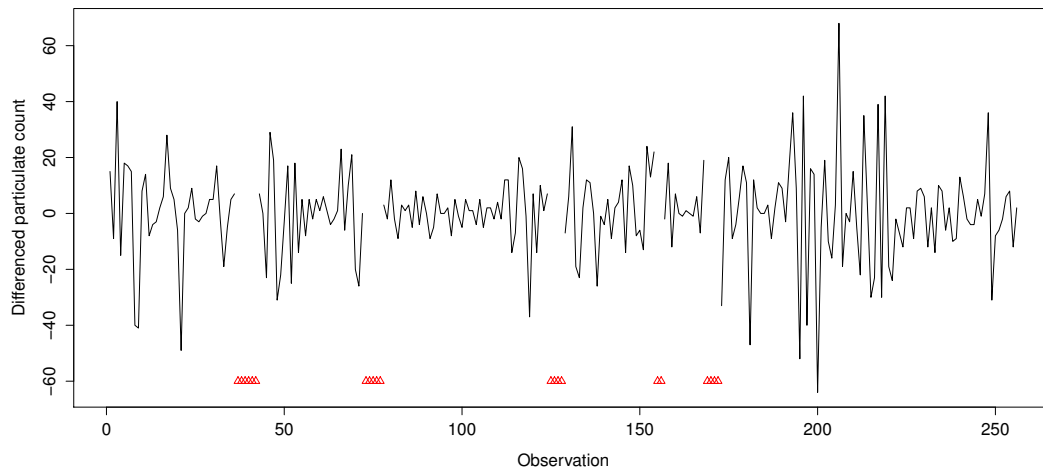


Figure 3: Differenced PM-10 series. Red triangles indicate the positions of missing observations.

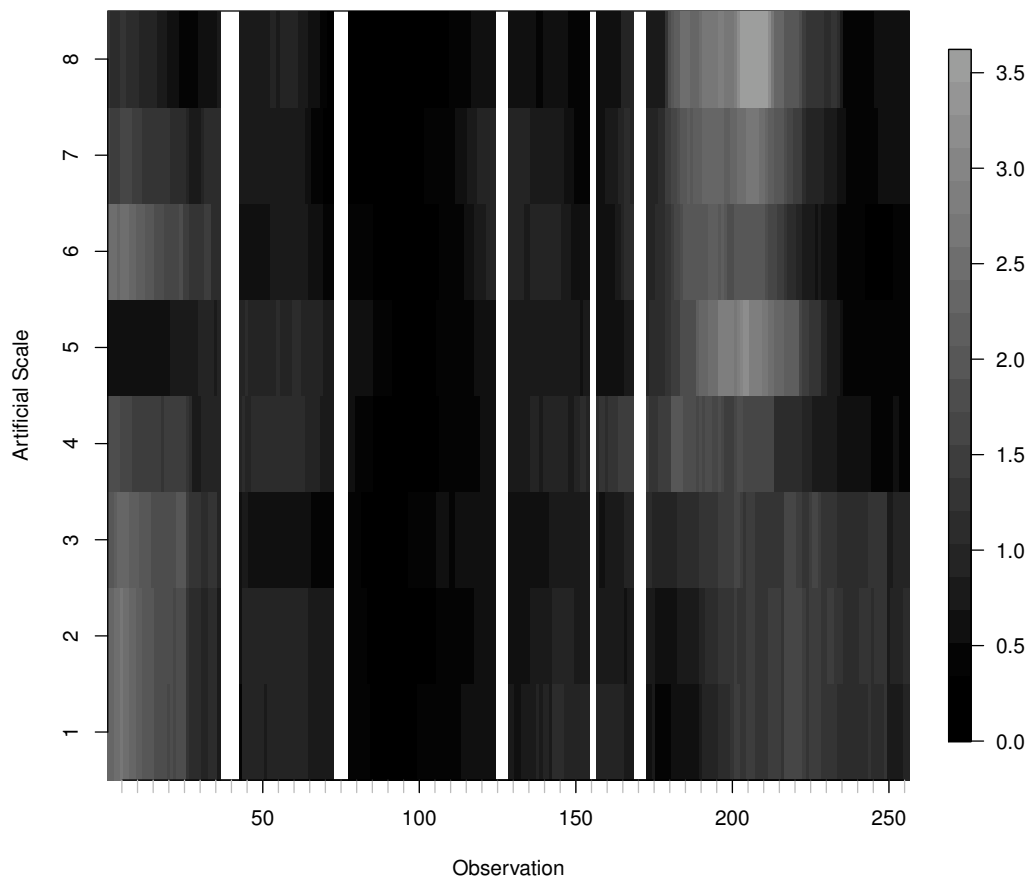


Figure 4: Estimated evolutionary wavelet spectrum of the differenced PM-10 time series. The spectrum is normalised by dividing through by the mean at each scale, as in Equation (65). Vertical white spaces indicate missing values.

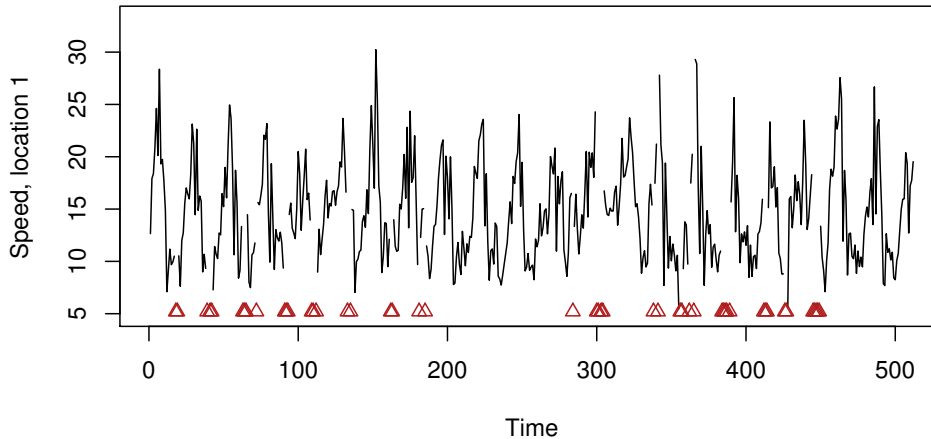


Figure 5: Average hourly speeds of cars inbound on the A38 (Monks Park to Stokes Croft) from 12am on 19/04/2016 to 8am on 11/05/2016. Red triangles indicate the location of missing observations.

location in this section, the decomposition is performed for each of the 29 locations. At the time positions of the mean value imputations, the ‘false’ residuals are removed so that the residual series has missing values at the same time positions as the original. These residuals with missing observations are denoted $\{B_{i,t}\}_{i=1,\dots,29}^{t=1,\dots,512}$. The residuals at location 1, $\mathbf{B}_1 = (B_{1,1}, \dots, B_{1,512})'$, are tested for stationarity using LiftToS. The residuals at location 1 are plotted in Figure 6, with the time positions of missing observations shown. In this example 9.2% of the data are missing.

As in the previous example, the parameters are set to those in Section 3.3. Before testing \mathbf{B}_1 is centred by subtracting the sample mean, as the test assumes the series has mean zero. The resulting p -value of LiftToS is 0.08, close to the nominal threshold. As the test statistic was not more extreme than 95% of the surrogate series’ test statistics there is not enough evidence to reject the null hypothesis of stationarity. Therefore methods designed for stationary data can be used with this series, as shown in Section 4.6.2. Figure 7 shows the estimated spectrum for this example, which features some variation at all scales, but has a smaller range of values compared to the PM-10 example of Figure 4.

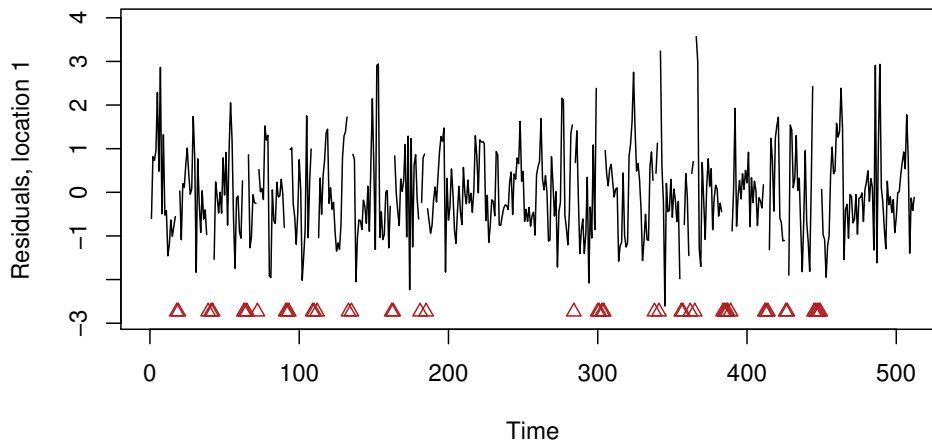


Figure 6: Residuals after seasonal decomposition of hourly speeds of cars inbound on the A38 (Monks Park to Stokes Croft) from 12am on 19/04/2016 to 8am on 11/05/2016. Red triangles indicate the location of missing observations.

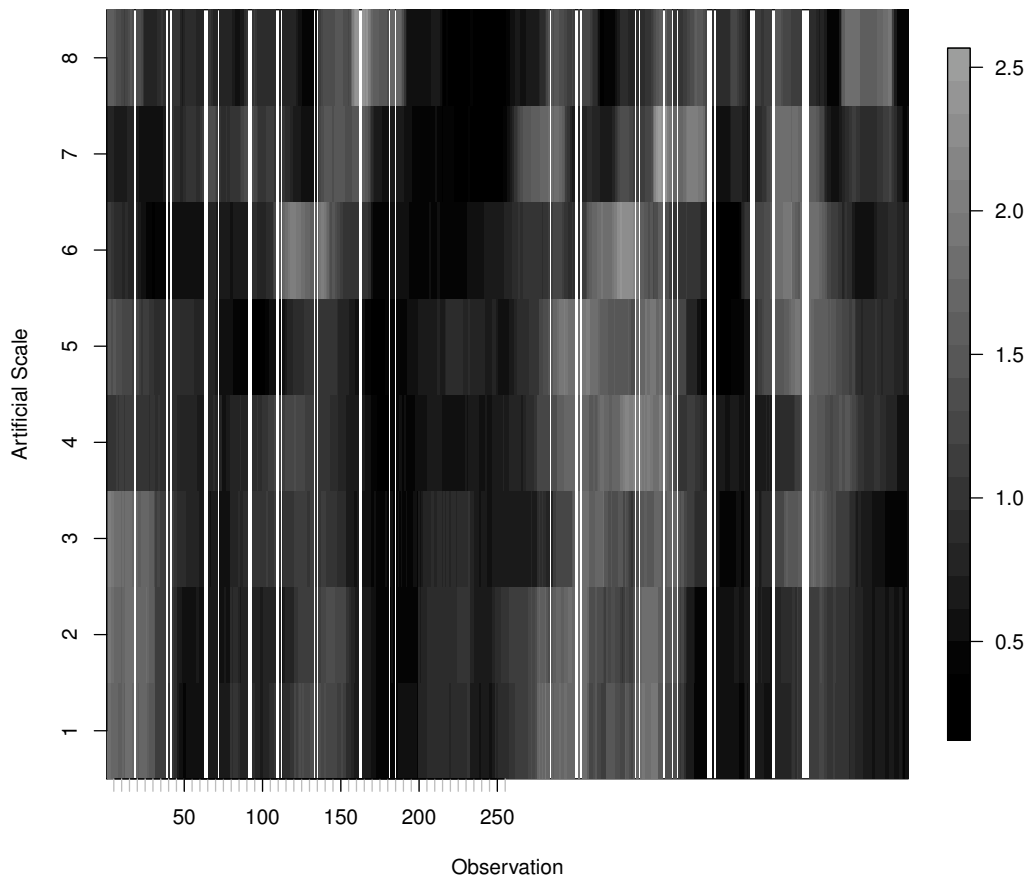


Figure 7: Estimated evolutionary wavelet spectrum of the time series in Figure 6. Vertical white spaces indicate missing values.

3.5 Conclusion

This chapter presented a new second-order stationarity test, designed for use with missing observations. Despite the test being designed for the missing data case, simulations in Section 3.3 have shown that LiftToS has comparable performance to state of the art tests in the complete data case. The examples show the usefulness of this test in real data scenarios where observations are missing from records.

An area of further work would be to adapt this test for use on irregularly-sampled series. The method of estimating the wavelet spectrum is applicable for irregularly-sampled series, but a method of simulating stationary bootstrap series for irregularly-sampled data would have to be developed. The current test is computationally intensive, as many trajectories and bootstrap series are used. Another potential area of further research is investigating which trajectories are ‘best’ in estimating the wavelet spectrum, this may enable fewer trajectories to be used, therefore increasing the speed of the test.

4 Generalised Network Autoregressive Processes

In this chapter, Sections 4.2-4.4 and the example in Section 4.6.1 are joint work with Marina Knight, Guy Nason, and Matthew Nunes. Personal contribution to these sections includes all proofs in Section 4.3 and the order criteria in Section 4.4.1. The simulations in Section 4.5 and the Bristol Traffic data application are new and completely my own work.

4.1 Introduction and additional literature review

Networks are used in many fields to study connections between objects of interest, and in recent years their use has increased rapidly, in part due to an increase in large-scale data collection. In this chapter network time series are considered, which consist of time series recorded at nodes of a network, or graph. Rather than treating the network and associated time series separately, in this chapter the multivariate time series and network are analysed jointly and simultaneously.

In this section, models for multivariate and network time series are described. Section 4.2 sets out our Generalised Network Autoregression model and the relevant network notation. In Section 4.3 results on the stationarity of the model and consistency of fitted parameters are presented. Implementation considerations such as order selection criteria are discussed in Section 4.4, and demonstrated in Section 4.5. A GDP example including construction of a useful network for prediction, and the Bristol Traffic example are shown in Section 4.6.

4.1.1 Multivariate time series modelling with autoregression

When split into its individual parts, a network time series consists of at least one network and a multivariate time series. This multivariate time series has observations of dimension $N \times T$, where N is the maximum number of nodes in the network(s) and T is the number of time observations. Throughout this section, the time series at node i and time t will be denoted $X_{i,t}$.

A model often used for multivariate time series, particularly in economics, is the vector

autoregressive (VAR) model.

Definition 4.1 (Vector AutoRegressive Model (adapted from Lütkepohl (2005), Definition 2.1.1)). A VAR model of order p is given by the following equation;

$$\mathbf{X}_t = \boldsymbol{\mu} + \phi_1 \mathbf{X}_{t-1} + \dots + \phi_p \mathbf{X}_{t-p} + \mathbf{u}_t, \quad t = 0, \pm 1, \pm 2, \dots, \quad (66)$$

where $\mathbf{X}_t = (X_{1,t}, \dots, X_{N,t})'$ is the vector of observations, $\{\phi_j\}_{j=1}^p$ are $N \times N$ matrices of coefficients, $\boldsymbol{\mu} = (\mu_1, \dots, \mu_N)'$ is a vector containing the means of each node, and $\mathbf{u}_t = (u_{1,t}, \dots, u_{N,t})$ is an innovation vector. The innovations are assumed to have zero mean and covariance matrix $\mathbb{E}(\mathbf{u}_t \mathbf{u}_t') = \Sigma_u$.

In matrix form observations of the VAR model with zero mean can be written as

$$X = BZ + U, \quad (67)$$

where $X = [\mathbf{X}_{p+1}, \dots, \mathbf{X}_T]$, $B = [\phi_1, \dots, \phi_p]$, $Z = [\mathbf{Z}_p, \dots, \mathbf{Z}_{T-1}]$, with $\mathbf{Z}_t' = [\mathbf{X}_t, \dots, \mathbf{X}_{t-p+1}]$, and $U = [\mathbf{u}_{p+1}, \dots, \mathbf{u}_T]$. The general mean-zero VAR model has up to pN^2 parameters within the ϕ matrices, so fitting a complete model of order p requires $T \geq pN$ time observations. For an innovation vector without constraints there are a further N^2 entries of Σ_u to be estimated. Therefore, a downside of the VAR model is that when N is large compared to T , only low orders of the full VAR model can be fitted.

A VAR model has the following stationarity condition, which is based upon the entries of the ϕ matrices.

Theorem 4.1 (Stationarity condition for VAR (adapted from Lütkepohl (2005), Section 2.1)). A VAR process \mathbf{X}_t as defined in Definition 4.1 is stationary if

$$\det(I_N - \phi_1 z - \dots - \phi_p z^p) \neq 0, \quad \forall z \in \mathbb{C}, \text{ with } |z| \leq 1, \quad (68)$$

where I_N is the identity matrix of dimension $N \times N$.

This stationarity condition can be assessed in practice by finding eigenvalues, so it is not a condition that can immediately be checked using the parameters of the VAR model without computation.

The VAR model can also be used with linear constraints, which enable control over the

parameters and restricts the number of free parameters in the model. The constraints imposed to form a VAR model can be written using the matrix form of Equation (67) as $\text{vec}(B) = R\gamma + \mathbf{r}$, where R is the known constraint matrix of dimension $pN^2 \times M$, γ is an unrestricted parameter vector of length M , \mathbf{r} is a pN^2 -length vector of known constants, and $\text{vec}(\cdot)$ is the operator that stacks the columns of a matrix into a vector.

In the linear constraint case, a consistent estimator of the innovation covariance matrix can be found as follows.

Proposition 4.2 (Consistent estimator of Σ_u (adapted from Lütkepohl (2005), Proposition 5.4)). For a stationary VAR process X , as in Equation (67), with linear constraints $\text{vec}(B) = R\gamma + \mathbf{r}$ and an innovation process that is independent white noise with bounded fourth moments and non-singular covariance matrix Σ_u , a consistent estimator of Σ_u is given by $\tilde{\Sigma}_u$,

$$\tilde{\Sigma}_u = T^{-1} \left(X - \hat{B}Z \right) \left(X - \hat{B}Z \right)', \quad (69)$$

where $\hat{B} = R \{ R' (ZZ' \otimes I_N) R \}^{-1} R' (Z \otimes I_N) \text{vec}(X) + \mathbf{r}$.

The \hat{B} values used in this estimator are generated by fitting the VAR parameters as if the underlying innovation covariance matrix is the identity. The consistent estimator of Σ_u can be used to generate consistent estimates of the constrained parameters, as shown in the following proposition.

Proposition 4.3 (Consistency of linearly constrained VAR parameters (adapted from Lütkepohl (2005), Proposition 5.2)). Under the conditions of Proposition 4.2, a consistent estimator of the free parameters of the constrained VAR model, γ , is given by

$$\hat{\gamma} = \left\{ R' \left(ZZ' \otimes \tilde{\Sigma}_u \right) R \right\}^{-1} R' \left(Z \otimes \tilde{\Sigma}_u \right) \text{vec}(X). \quad (70)$$

In addition,

$$\sqrt{T}(\hat{\gamma} - \gamma) \rightarrow^d N \left[0, \left\{ R' \left(\Gamma \otimes \Sigma_u^{-1} \right) R \right\}^{-1} \right], \quad (71)$$

where $\Gamma = \mathbb{E}(Z_t Z_t')$, which under stationarity is the limit in probability of $ZZ'T^{-1}$.

4.1.2 Network models

A simple random graph model is the Erdős-Rényi model, which assigns equal probability to each pair of nodes being linked.

Definition 4.2 (Erdős-Rényi random graph (adapted from Grimmett (2010), page 205)).

For a network with nodes $\mathcal{K} = \{1, \dots, N\}$, define a set of independent Bernoulli random variables, $\{Y_{i,j} : 1 \leq i < j \leq N, i, j \in \mathcal{K}\}$, with parameter p . An Erdős-Rényi random graph $G_{N,p}$ has an edge between nodes i and j exactly when $Y_{i,j} = 1$.

A more general model of random networks is the stochastic block model, which allows for different probabilities of connection within the network.

Definition 4.3 (Stochastic block model (adapted from Kolaczyk (2017))). For a network with nodes $\mathcal{K} = \{1, \dots, N\}$, let $\mathbf{Z}_i = (Z_{i,1}, \dots, Z_{i,Q})$ for $i \in \mathcal{K}$ and $Q \in \mathbb{N}$ define a vector encoding the block assignment of node i , where

$$\mathbf{Z}_i \sim \text{Multinomial}(1, \boldsymbol{\alpha}), \quad (72)$$

independently and identically distributed for all $i \in \mathcal{K}$, and where the Q -length parameter vector $\boldsymbol{\alpha}$ sums to one. Therefore the vector \mathbf{Z}_i is a vector of zeros, with exactly one entry being 1, indicating the block-assignment of node i . Then for each $1 \leq i < j \leq N$ with $Z_{i,q} = 1$ and $Z_{j,q'} = 1$, define an independent Bernoulli random variable,

$$Y_{i,j} \sim \text{Bernoulli}(p_{q,q'}), \quad (73)$$

where $p_{q,q'} \in [0, 1]$ is the probability of connection between a node in block q and a node in block q' . A stochastic block model has an edge between nodes i and j exactly when $Y_{i,j} = 1$, and $Y_{i,i}$ is defined as 0 for all $i \in \mathcal{K}$.

The Erdős-Rényi random graph is a special case of the stochastic block model with $Q = 1$. The stochastic block model can be used to encode community information within social networks.

As our work concerns networks that are generally considered to be known, we do not provide further details of stochastic network models here, for a comprehensive review see Salter-Townshend et al. (2012).

4.1.3 Network regression models

A work combining linear equations and network concepts in the social science context is that of Doreian (1981). The model they described has been referred to by many different

names, including the mixed endogenous-exogenous model, the regressive-autoregressive model, and a spatial autoregressive model (Leenders (2002)). This model can be written for a vector of observations \mathbf{X} as

$$\mathbf{X} = \rho W \mathbf{X} + Z \boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad (74)$$

where ρ is an autocovariance parameter, W is a weight matrix, Z is a matrix of observations on exogenous variables, $\boldsymbol{\beta}$ is a vector of parameters, and $\boldsymbol{\epsilon}$ is an innovation vector. The innovations are from a multivariate normal distribution, with $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 I)$.

This model can describe spatial autocorrelations using the weight matrix, W . An important feature of network time series models is the inclusion of network interactions within the model via the W matrix. Leenders (2002) discusses a range of different choices of weighting matrix, including row-normalised adjacency matrices and adjacency matrices subject to thresholding. Whilst this model is flexible in terms of the dependencies between observations over time, it is restricted by the use of a normally distributed innovations.

Analogously to the univariate time series context, a spatial moving average model has also been proposed for data observed with a network, such as by Ord (1975) and Mur (1999). This model can be written as

$$\begin{aligned} \mathbf{X} &= Z \boldsymbol{\beta} + \boldsymbol{\epsilon}, \\ \boldsymbol{\epsilon} &= \rho W \boldsymbol{\epsilon} + \boldsymbol{\nu}, \end{aligned} \quad (75)$$

where $\boldsymbol{\nu} \sim N(\mathbf{0}, \sigma^2 I)$, and other quantities are as Equation (74). In this case the network autocorrelation is present within the innovation vector, $\boldsymbol{\epsilon}$, rather than the observations \mathbf{X} . These two models are not specifically time series models, as there is no restriction or specific ordering on the elements of \mathbf{X} .

A time series model of a similar form to the linear models above is the m-STAR model of Hays et al. (2010). This model contains C sets of spatial weights, and can be written for observation t at node i for a network with N nodes as

$$X_{i,t} = \mathbf{Z}'_i \boldsymbol{\beta} + \alpha X_{i,t-1} + \sum_{j=1}^N \sum_{c=1}^C \rho_c w_{i,j,c}^{(t)} X_{j,t} + \epsilon_{i,t}, \quad (76)$$

where \mathbf{Z}_i is a vector of node-specific covariates, with associated parameter vector $\boldsymbol{\beta}$, α is a first-order autoregression parameter and there are C different (potentially time-varying) network weightings indexed by c with individual weights $w_{i,j,c}^{(t)}$ and influence parameters, ρ_c . The innovations, $\{\epsilon_{i,t}\}_{i=1,\dots,N}^{t=1,\dots,T}$, are considered to be independent and identically distributed normal random variables for the purpose of calculating the likelihood. The m-STAR model offers both lag-1 regression effects and contemporaneous weighted neighbourhood effects, but is limited as it does not allow for higher lag time effects.

Network time series were used to assess interventions in an experimental design context by Spencer et al. (2015). Denoting a measurement taken on variable $j \in \{1, \dots, p\}$ at time $t \in \{1, \dots, T\}$ under experimental condition $c \in \{1, \dots, C\}$ as $X_{j,c,t}$, the non-intervention model considered is

$$X_{j,c,t} = \begin{cases} \alpha_1^{(j)} + \sum_{i \in P(j)} X_{i,c,t-1} \beta_i^{(j)} + \epsilon_{j,c,t}, & t > 1, \\ \alpha_2^{(j)} + \epsilon_{j,c,1}, & t = 1, \end{cases} \quad (77)$$

where $P(j)$ are the parent nodes of node j , the α parameters are intercepts, the β parameters represent the influence of the parent nodes, and the innovations $\epsilon_{j,c,t} \sim N(0, \sigma_j^2)$.

This model assumes an underlying Dynamic Bayesian network model, where edges between nodes indicate influence over time. The network encodes a first-order Markov relation between nodes, where there is no within time-slice interdependence unlike the m-STAR model. For the non-intervention model Spencer et al. (2015) assume network and parameter stationarity. This model is used for inference of the underlying network, and therefore the relationships between variables.

The interventions are then modelled using a Causal Dynamic Bayesian Network, which alters the form of the likelihood compared to the non-intervention case. The intervention is assumed to be known to target a particular node, and have no effect elsewhere within the network. In addition, it is assumed that the effect of an intervention continues throughout the experiment. Interventions can remove edges from or to a node, or change parameters at certain nodes. For example, a perfect-out intervention removes the connection(s) from the intervention parent node(s) to the target node. This is modelled using zeroes in the

design matrix, giving the following change to Equation (77);

$$X_{j,c,t} = \begin{cases} \alpha_1^{(j)} + \sum_{i \in P(j)} \mathbb{I}\{i \notin Q_c(j)\} X_{i,c,t-1} \beta_i^{(j)} + \epsilon_{j,c,t}, & t > 1, \\ \alpha_2^{(j)} + \epsilon_{j,c,1}, & t = 1, \end{cases} \quad (78)$$

where $Q_c(j)$ is the set of parent nodes of node j subject to the intervention in experimental condition c . The model is fitted and inference made about the graph by performing variable selection at each node using Bayesian methods involving priors on the network connections and the marginal likelihood at each node. Whilst this model is useful to examine changes with known effects, the set $Q_c(j)$ is assumed known which is restrictive. Another restriction of this model is the single time-lag used for parent node effects, and the assumption that the innovations are uncorrelated.

Another network time series model is the Dynamic Chain Graph Model (DCGM) of Anacleto and Queen (2017). This is a Bayesian dynamic model, designed for high-dimensional network time series. Their model uses a dynamic chain graph to define conditional independence structures between the nodes. These dependencies are split into intra time-slices and inter time-slices, the first representing associations at a fixed time, and the second representing associations across time. By ordering n partitions of N variables such that the intra time-slice parents of partition i are contained within partitions $1, \dots, i-1$, and regressive associations of partition i are contained within partitions $1, \dots, i$, the model can be defined using the conditional independence structure implied by these partitions. Let $\sigma_1, \dots, \sigma_N$ denote the ordering of the variables such that the above partition properties hold, and $\sigma(i) = \sigma_k$ with $k = 1 + \sum_{j=1}^{i-1} r_j$ denote the location of the first variable in partition i , where r_i is the number of variables in partition i .

The Anacleto and Queen (2017) DCGM is defined for an observation of the r_i variables in partition i , $\mathbf{X}_t(i) = \{X_{\sigma(i)}, \dots, X_{\sigma(i+1)-1}\}'$ has observation, system, and initial equations as follows for non-negative integer t :

$$\begin{aligned} \mathbf{X}_t(i) &= \mathbf{F}_t(i) \boldsymbol{\theta}_t(i) + \mathbf{u}_t(i), & \mathbf{u}_t(i) &\sim \{\mathbf{0}, \Sigma_t(i)\}, & i = 1, \dots, n, \\ \boldsymbol{\theta}_t &= G_t \boldsymbol{\theta}_{t-1} + \mathbf{w}_t, & \mathbf{w}_t &\sim (\mathbf{0}, W_t), \\ \boldsymbol{\theta}_0 | D_0 &\sim (\mathbf{m}_0, C_0). \end{aligned} \quad (79)$$

In the DCGM, $\mathbf{F}_t(i)$ is a matrix containing a known function of the values at t of the parents of the variables in partition i , and values at $t - 1$ of variables in partitions $1, \dots, i$. $\boldsymbol{\theta}$ is the state vector, with block-diagonal evolution matrix G_t , and evolution covariance matrix W_t . The observations in partition i have covariance matrix $\Sigma_t(i)$. Given initial information, D_0 , the initial state vector has mean vector \mathbf{m}_0 and block-diagonal covariance matrix C_0 .

The DCGM can be used in many cases to embed other models in the Bayesian framework, for example certain VAR models can be written as DCGMs. If the connections between variables can be written as a chain graph then the DCGM allows for the partitions of the graph to be treated separately, allowing for high-dimensional problems to be split down into computationally feasible parts. Limitations of this model include that the time dependence modelled is only at lag-1, and the problem has to be separable into the chained conditionally independent partitions for the DCGM to be suitable.

Knight et al. (2016) described network time series models including Network AR (NAR), and Network ARIMA (NARIMA) which can include multiple time lags in the network regression. The NAR model for time t and node i is written as

$$X_{i,t} = \sum_{j=1}^p \left\{ \alpha_j X_{i,t-j} + \sum_{r=1}^{s_j} \sum_{q \in \mathcal{N}^{(r)}(i)} \beta_{j,r,q} X_{q,t-j} \right\} + u_{i,t}, \quad (80)$$

where $\mathcal{N}^{(r)}(i)$ is the set of stage- r neighbours of node i (see Definition 4.5), p is the maximal time lag, $\mathbf{s} = (s_1, \dots, s_p)$ indicates the number of neighbour stages to include at each time lag, and $u_{i,t}$ are mutually uncorrelated innovation variables.

The NAR model contains autoregressive parameters, $\{\alpha_j\}$, and network regressive parameters, $\{\beta_{j,r,q}\}$. This model is more flexible than many of the previous models described in this section as it can model effects more than one time point in the past. However, a drawback of this model is that it does not include node-specific covariates, and only allows for one network per time point, unlike the m-STAR model. In Section 4.2 our generalised network autoregression model is presented, an extension of the NAR model of Knight et al. (2016) to additionally allow for different autoregressive parameters at each node, and allow the inclusion of covariates. Key differences between other related models and our GNAR model are described in Section 4.2.3.

4.2 Description of the GNAR Model

The Generalised Network Autoregressive (GNAR) model describes a multivariate time series process, observed at nodes of a network. The process at each node follows an autoregressive model, with additional dependence defined by the network, more specifically the neighbourhood structure of each node.

4.2.1 Network neighbourhood structure

The network(s) used in the GNAR model can be weighted, and directed, or follow a simple binary structure. A graph is denoted $\mathcal{G} = (\mathcal{K}, \mathcal{E})$, where \mathcal{K} is a set of N nodes, and \mathcal{E} is a set of edges. These edges may have weight and direction attributes.

Within this network terminology, a directed edge from node $i \in \mathcal{K}$ to $j \in \mathcal{K}$ is denoted $i \rightsquigarrow j$, with an un-directed edge being described by either two directed edges, $i \rightsquigarrow j$ and $j \rightsquigarrow i$, or using a double arrow, $i \leftrightarrow j$.

Definition 4.4 (Neighbour set). The neighbour set of a subset $A \subset \mathcal{K}$ is

$$\mathcal{N}(A) = \{j \in \mathcal{K} : i \rightsquigarrow j; i \in A\}.$$

Members of a neighbour set of a node are also known as the node's first-stage neighbours.

Definition 4.5 (Stage- r neighbour set). The r th stage neighbour set of a node $i \in \mathcal{K}$, for $r = 2, 3, \dots$ is

$$\mathcal{N}^{(r)}(i) = \mathcal{N} \left\{ \mathcal{N}^{(r-1)}(i) \right\} / \left[\left\{ \bigcup_{s=1}^{r-1} \mathcal{N}^{(s)}(i) \right\} \cup \{i\} \right],$$

where $\mathcal{N}^{(1)}(i) = \mathcal{N}(\{i\})$.

When the network changes over time, a subscript t is added to the neighbour set notation.

Nodes of a neighbour set may have different levels of influence due to distance or strength of connection to the original node. This is captured using connection weights, which take into account the weightings or distances if present in the network. For ease of notation, each edge in an un-weighted network has weight (or distance) 1. Distances between stage- r neighbours are constructed by summing the distances along r -length paths between the two nodes, and taking the minimum of these distances. Formally, let $\sigma = (\sigma_1, \dots, \sigma_{r+1})$

be an ordering of nodes denoting a path between stage- r neighbours i and j , with $\sigma_1 = i$, $\sigma_{r+1} = j$, and $\sigma_k \rightsquigarrow \sigma_{k+1}$, $\forall k \in 1, \dots, r$. The (stage- r) distance between i and j is $d_{i,j} = \min_{\sigma} \sum_{k=1}^r d_{\sigma_k, \sigma_{k+1}}$. In our notation the stage is not denoted on the distances $d_{i,j}$, as for each directed pair i, j at any time point, there is at most one $r \in \mathbb{N}$ for which $j \in \mathcal{N}^{(r)}(i)$.

Weightings and distances are treated as duals of each other, where a distance from node i to node j is denoted $d_{i,j} \in \mathbb{R}_+$ and the weight of the edge from i to node j is denoted $\mu_{i,j} \in \mathbb{R}_+$, we set $d_{i,j} = \mu_{i,j}^{-1}$.

Definition 4.6 (Connection weights). The connection weight of a node $i \in \mathcal{K}$ and its stage- r neighbour $j \in \mathcal{N}^{(r)}(i)$ is

$$\omega_{i,j} = d_{i,j}^{-1} \left\{ \sum_{k \in \mathcal{N}^{(r)}(i)} d_{i,k}^{-1} \right\}^{-1}.$$

Therefore the connection weights will sum to one whenever the neighbour set is non-empty. The connection weights are similar to the inverse distance prediction weights used by Jansen et al. (2009). These connection weights will be time-varying whenever the network is. In an un-weighted network, all neighbours in a set will have equal influence on the node. Note that the connection weights are not necessarily symmetric, even in an un-directed network.

The GNAR model also allows for covariates on edges or nodes, which splits them into different types. Here the covariate may take $C \in \mathbb{N}$ values and the definition of the neighbour sets and connection weights are unaltered by these covariates. An example of where covariates could be used is where edges in the network define different relationships, such as family or friendship links in a social network. The covariates could also be used to encode the block-assignment of the stochastic block model (Definition 4.3), so each block can have a different covariate and network parameters.

4.2.2 Generalised Network Autoregressive Model

For a network of N nodes, \mathcal{G} , and associated vector of time series $\mathbf{X}_t = (X_{1,t}, \dots, X_{N,t})'$, the Generalised Network Autoregressive model describes dependence at a node on its values at previous time points, and also across neighbouring nodes at previous time points.

Definition 4.7 (Generalised Network Autoregressive Model). The GNAR($p, [\mathbf{s}]$) model with $p \in \mathbb{N}$ and $\mathbf{s} \in \mathbb{N}_0^p$ is defined for each node $i \in \{1, \dots, N\}$ and $t \in \{1, \dots, T\}$ as

$$X_{i,t} = \sum_{j=1}^p \left\{ \alpha_{i,j} X_{i,t-j} + \sum_{c=1}^C \sum_{r=1}^{s_j} \beta_{j,r,c} \sum_{q \in \mathcal{N}_t^{(r)}(i)} \omega_{i,q,c}^{(t)} X_{q,t-j} \right\} + u_{i,t}, \quad (81)$$

where $u_{i,t}$ is an identically distributed white noise process such that \mathbf{u}_t is independent with covariance matrix Σ_u . A sum from one to zero is assumed to be zero in this definition; $\sum_{r=1}^0(\cdot) := 0$, and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.

The GNAR model has two sets of parameters, the autoregression parameters, α , and the network regression parameters, β . The order of the model, $(p, [\mathbf{s}])$ defines the number of each type of parameter, p is the maximum time lag of autoregression in the model, and \mathbf{s} is a p -length vector defining the number of neighbour sets included at each time lag. The GNAR model describes potentially high-dimensional data situation with just this small collection of parameters and the network. There are Np of the α parameters and $C \sum_{j=1}^p s_j$ of the β parameters in the GNAR($p, [\mathbf{s}]$) model, giving $M = Np + C \sum_{j=1}^p s_j$ parameters in total. To reduce the number of model parameters further, the α parameters can be shared throughout the network so that for all $j \in \{1, \dots, p\}$, $\alpha_{i,j} \equiv \alpha_j \forall i \in \mathcal{K}$. This reduced-parameter model is called the global- α GNAR model, and has $M = p + C \sum_{j=1}^p s_j$ parameters. The network regression parameters, β are always ‘shared’ by all nodes of the network; GNAR is always a global- β model.

As well as the sparse representation, a key benefit of our model is the flexibility of the network structure that can be included. Although the network may change over time via the connection weights $\{\omega\}$, the model parameters stay fixed. This allows the model to be used in both changing network, and missing data situations.

4.2.3 Comparisons to other models

When the network in the GNAR model is fixed in time, the GNAR model can be written as a restricted VAR model, where the restrictions are defined by the network. The column form of the static-network GNAR($p, [\mathbf{s}]$) model, where $\mathbf{X}_t = (X_{1,t}, \dots, X_{N,t})'$, is written in the VAR framework as

$$\mathbf{X}_t = \phi_1 \mathbf{X}_{t-1} + \dots + \phi_p \mathbf{X}_{t-p} + \mathbf{u}_t, \quad (82)$$

where ϕ_k are $N \times N$ matrices containing the autoregressive and network autoregressive information such that $\phi_k = \text{diag}\{\alpha_{i,k}\} + \sum_{c=1}^C \sum_{r=1}^{s_k} \beta_{k,r,c} W^{(r,c)}$, matrices $W^{(r,c)}$ have entries $[W^{(r,c)}]_{\ell,m} = \omega_{\ell,m,c} \mathbb{I}\{m \in \mathcal{N}^{(r)}(\ell)\}$ and \mathbf{u}_t is the vector of errors at time t .

In matrix form the GNAR model is

$$X = BZ + U, \quad (83)$$

where $X = [\mathbf{X}_{p+1}, \dots, \mathbf{X}_T]$, $B = [\phi_1, \dots, \phi_p]$, $Z = [\mathbf{Z}_p, \dots, \mathbf{Z}_{T-1}]$, with $\mathbf{Z}'_t = [\mathbf{X}_t, \dots, \mathbf{X}_{t-p+1}]$, and $U = [\mathbf{u}_{p+1}, \dots, \mathbf{u}_T]$. The constraints imposed to form a GNAR model are $\text{vec}(B) = R\boldsymbol{\gamma}$, where R is the constraint matrix embedding the network structure of dimension $pN^2 \times M$, $\boldsymbol{\gamma}$ is an unrestricted parameter vector of length M , where M is the number of free parameters, and vec is the operator that stacks the columns of a matrix into a vector. Although the GNAR model with static network can be written in restricted VAR form, the network concept allows for greater interpretability than a standard reduced VAR, and additionally the GNAR model gives a simple form for the stationarity condition in Theorem 4.4.

The VAR model in its standard form for an N -dimensional time series has $\mathcal{O}(N^2)$ parameters. In comparison, the GNAR model has far fewer parameters, usually $M = \mathcal{O}(N)$ for a model with known network, or $M = \mathcal{O}(1)$ for the global- α model. As the VAR model has more free parameters, the model can very closely fit data, however this does not always create a model that works well in prediction, as shown in Section 4.6.1.

In work addressing the issue of many parameters for the VAR model, Davis et al. (2016) propose a two-stage method to fit sparse VAR models. Their method uses partial spectral coherence to assess which pairs of univariate series within the multivariate series feature partial correlation, then a t -statistic to assess the significance of each fitted parameter, and the BIC at each step to select a sparse VAR representation. This method, and other similar approaches, can be used to construct an influence network based upon the parameters connecting univariate series in the sparse representation. Although related to the network underpinning our GNAR model, the network based upon VAR modelling would blur the differences between stage neighbours in our model, for example a node would be connected to all its stage-1 to stage- s_1 neighbours using the sparse VAR approach to network reconstruction.

A model that is similar to the GNAR model is the network autoregression model proposed by Zhu et al. (2017) in the context of social networks. There are some key differences between the two models, to show these we write the Zhu et al. (2017) model in the GNAR notation as

$$X_{i,t} = \beta_0 + Z_i' \gamma + \sum_{j=1}^p \left\{ \alpha_j X_{i,t-j} + \beta_j \sum_{q \in \mathcal{N}^{(1)}(i)} \omega_i X_{q,t-j} \right\} + u_{i,t}, \quad (84)$$

where β_0 is a global intercept term, Z_i is a vector containing node-specific covariates with corresponding parameters γ , ω_i is the reciprocal of the number of edges leaving node i . The innovations, $u_{i,t}$ are independent and identically distributed white noise. Comparing (84) to Definition 4.7, the Zhu et al. (2017) model has more node-specific covariates, which could be added to the GNAR model. However, if these covariates are removed from (84), the model is reduced to a special case of the GNAR model, with constant network, only stage-1 neighbour sets, and binary valued edges. Therefore the GNAR model can describe dependence on higher-order neighbour sets, and is more flexible by allowing for a range of input networks.

4.3 Theoretical results

In this section, a stationarity condition for our GNAR model is presented, along with consistency results for fitted parameters.

4.3.1 Stationarity condition for GNAR processes with fixed network

Theorem 4.4. Given a fixed network, \mathcal{G} , a sufficient condition for the GNAR model in Definition 4.7 to be stationary is

$$\sum_{j=1}^p \left(|\alpha_{i,j}| + \sum_{c=1}^C \sum_{r=1}^{s_j} |\beta_{j,r,c}| \right) < 1 \quad \forall i \in 1, \dots, N. \quad (85)$$

Proof. First Gerschgorin's theorem and a corollary are presented without proof, both quoted from Varga (1962).

Theorem Varga (1962) Theorem 1.5 Let $A = (a_{i,j})$ be an arbitrary $N \times N$ complex matrix, and let $\Lambda_i \equiv \sum_{j=1, j \neq i}^N |a_{i,j}|$, $1 \leq i \leq N$. Then, all of the eigenvalues λ of A lie in the

union of the disks $|z - a_{i,i}| \leq \Lambda_i$, $1 \leq i \leq N$.

Since the disk $|z - a_{i,i}| \leq \Lambda_i$ is a subset of the disk $|z| \leq |a_{i,i}| + \Lambda_i$, we have the immediate result of

Corollary *Varga (1962) Theorem 1.5, Corollary 1* If $A = (a_{i,j})$ is an arbitrary $N \times N$ complex matrix with eigenvalues λ_i , $1 \leq i \leq N$, and $\nu \equiv \max_{1 \leq i \leq N} \sum_{j=1}^N |a_{i,j}|$, then $\max_{1 \leq i \leq N} |\lambda_i| \leq \nu$.

As described in Equation 82, the static-network GNAR process can be written as a VAR process, by defining $\mathbf{X}_t = (X_{1,t}, \dots, X_{N,t})'$, and writing $\mathbf{X}_t = \phi_1 \mathbf{X}_{t-1} + \dots + \phi_p \mathbf{X}_{t-p} + \mathbf{u}_t$, where ϕ_k are $N \times N$ matrices such that $\phi_k = \text{diag}\{\alpha_{i,k}\} + \sum_{c=1}^C \sum_{r=1}^{s_k} \beta_{k,r,c} W^{(r,c)}$, where matrices $W^{(r,c)}$ have entries $[W^{(r,c)}]_{\ell,m} = \omega_{\ell,m,c} \mathbb{I}\{m \in \mathcal{N}^{(r)}(\ell)\}$ and \mathbf{u}_t is the vector of errors at time t . The notation $[\cdot]_{\ell,m}$ is used to denote the ℓ, m entry of a matrix.

From (eg) Brockwell and Davis (2006), the VAR model has exactly one stationary solution if $\det(I_N - \phi_1 z - \dots - \phi_p z^p) \neq 0$ for all $z \in \mathbb{C}$ such that $|z| \leq 1$. Using Lemma 2.1 from Tsay (2014), $\det(I_N - \phi_1 z - \dots - \phi_p z^p) = \det(I_{Np} - \Phi z)$, where Φ is the $Np \times Np$ companion matrix defined as

$$\Phi = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_{p-1} & \phi_p \\ I_N & 0_N & \dots & 0_N & 0_N \\ 0_N & I_N & \dots & 0_N & 0_N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_N & 0_N & \dots & I_N & 0_N \end{bmatrix}$$

where I_N and 0_N are the $N \times N$ identity and zero matrices, respectively.³ Thus the roots of $\det(I_{Np} - \Phi z)$ must be outside of the unit-circle for stationarity of the GNAR process, or equivalently, the eigenvalues of Φ must lie inside the unit-circle.

Next the eigenvalues of Φ are investigated using Corollary 1.

For rows $N + 1, \dots, Np$, $\max_{N+1 \leq \ell \leq Np} \sum_{m=1}^{Np} |\Phi_{\ell,m}| = 1$.

³Note that Φ is defined differently in the two books, this is the Tsay (2014) version.

For rows $1, \dots, N$,

$$\begin{aligned}
\max_{1 \leq \ell \leq N} \sum_{m=1}^{Np} |\Phi_{\ell,m}| &= \max_{1 \leq \ell \leq N} \sum_{s=1}^N \sum_{k=1}^p |[\phi_k]_{\ell,s}| \\
&= \max_{1 \leq \ell \leq N} \sum_{s=1}^N \sum_{k=1}^p \left| \text{diag}\{\alpha_{i,k}\} + \sum_{c=1}^C \sum_{r=1}^{s_k} \beta_{k,r,c} W^{(r,c)} \right|_{\ell,s} \\
&\leq \max_{1 \leq \ell \leq N} \sum_{s=1}^N \sum_{k=1}^p \left| \text{diag}\{|\alpha_{i,k}|\} + \sum_{c=1}^C \sum_{r=1}^{s_k} |\beta_{k,r,c}| W^{(r,c)} \right|_{\ell,s} \\
&= \max_{1 \leq \ell \leq N} \sum_{s=1}^N \sum_{k=1}^p \left(|\alpha_{\ell,k}| \mathbb{I}\{\ell = s\} \right. \\
&\quad \left. + \sum_{c=1}^C \sum_{r=1}^{s_k} |\beta_{k,r,c}| \omega_{\ell,s,c} \mathbb{I}\{s \in \mathcal{N}^{(r)}(\ell)\} \right) \\
&= \max_{1 \leq \ell \leq N} \sum_{k=1}^p \left(|\alpha_{\ell,k}| \sum_{s=1}^N \mathbb{I}\{\ell = s\} \right. \\
&\quad \left. + \sum_{c=1}^C \sum_{r=1}^{s_k} |\beta_{k,r,c}| \sum_{s=1}^N \omega_{\ell,s,c} \mathbb{I}\{s \in \mathcal{N}^{(r)}(\ell)\} \right) \\
&\leq \max_{1 \leq \ell \leq N} \sum_{k=1}^p \left(|\alpha_{\ell,k}| + \sum_{c=1}^C \sum_{r=1}^{s_k} |\beta_{k,r,c}| \right),
\end{aligned}$$

as at each node $\ell \in \mathcal{K}$ and each covariate $c \in \{1, \dots, C\}$, $\sum_{s \in \mathcal{N}^{(r)}(\ell)} \omega_{\ell,s,c} \leq 1$. Under condition (85), $\max_{1 \leq \ell \leq N} \sum_{m=1}^{Np} |\Phi_{\ell,m}| < 1$. Therefore, $\max_{1 \leq \ell \leq Np} \sum_{m=1}^{Np} |\Phi_{\ell,m}| \leq 1$, and using Corollary 1, we have that the spectral radius of Φ is at most 1.

Whether an eigenvalue with modulus 1 is possible is now investigated.

Assume that there exists an eigenvalue, λ of Φ such that $|\lambda| = 1$. By definition, there exists an eigenvector $\mathbf{v} \in \mathbb{C}^{Np}$ such that $\Phi \mathbf{v} = \lambda \mathbf{v}$. By writing $\mathbf{v} = (\mathbf{v}'_1, \dots, \mathbf{v}'_p)'$, where each \mathbf{v}_k is a column vector of length N , the eigenequation can be rewritten as the following simultaneous equations:

$$\begin{aligned}
\text{(i)} \quad & \sum_{k=1}^p \phi_k \mathbf{v}_k = \lambda \mathbf{v}_1 \\
\text{(ii)} \quad & \mathbf{v}_k = \lambda \mathbf{v}_{k+1} \quad \forall k \in \{1, \dots, p-1\}.
\end{aligned}$$

Therefore $\mathbf{v}_k = \lambda^{p-k} \mathbf{v}_p \quad \forall k \in \{1, \dots, p\}$ and replacing this on both sides of (i) gives $\sum_{k=1}^p \phi_k \lambda^{p-k} \mathbf{v}_p = \lambda^p \mathbf{v}_p$. This results in the equation $\sum_{k=1}^p \phi_k \lambda^{-k} \mathbf{v}_p = \mathbf{v}_p$, which can be written in matrix form as $\Psi \mathbf{v}_p = \mathbf{v}_p$, where Ψ is the $N \times N$ matrix $\Psi = \sum_{k=1}^p \phi_k \lambda^{-k}$.

Hence if Φ has an eigenvalue of modulus 1, then Ψ must have 1 as an eigenvalue.

Corollary 1 is used again for the eigenvalues of Ψ , under the assumption $|\lambda| = 1$.

$$\begin{aligned}
\max_{1 \leq \ell \leq N} \sum_{m=1}^N |\Psi_{\ell,m}| &= \max_{1 \leq \ell \leq N} \sum_{m=1}^N \left| \left[\sum_{k=1}^p \phi_k \lambda^{k-2} \right]_{\ell,m} \right| \\
&= \max_{1 \leq \ell \leq N} \sum_{m=1}^N \left| \sum_{k=1}^p \lambda^{k-2} \left[\text{diag}\{\alpha_{i,k}\} + \sum_{c=1}^C \sum_{r=1}^{s_k} \beta_{k,r,c} W^{(r,c)} \right]_{\ell,m} \right| \\
&\leq \max_{1 \leq \ell \leq N} \sum_{m=1}^N \sum_{k=1}^p |\lambda^{k-2}| \left(|\alpha_{\ell,k}| \mathbb{I}\{\ell = m\} \right. \\
&\quad \left. + \sum_{c=1}^C \sum_{r=1}^{s_k} |\beta_{k,r,c}| \omega_{\ell,m,c} \mathbb{I}\{m \in \mathcal{N}^{(r)}(\ell)\} \right) \\
&= \max_{1 \leq \ell \leq N} \sum_{k=1}^p \left(|\alpha_{\ell,k}| + \sum_{c=1}^C \sum_{r=1}^{s_k} |\beta_{k,r,c}| \sum_{m=1}^N \omega_{\ell,m,s} \mathbb{I}\{m \in \mathcal{N}^{(r)}(\ell)\} \right) \\
&\leq \max_{1 \leq \ell \leq N} \sum_{c=1}^C \sum_{k=1}^p \left(|\alpha_{\ell,k}| + \sum_{c=1}^C \sum_{r=1}^{s_k} |\beta_{k,r,c}| \right)
\end{aligned}$$

Under condition (85) this is smaller than 1, so by Corollary 1 no eigenvalues of Ψ have modulus 1 or greater. This contradicts the assumption that an eigenvalue of Φ , λ , exists such that $|\lambda| = 1$. Hence the eigenvalues of Φ are inside the unit circle under condition (85), so the GNAR model is stationary. \square

4.3.2 Parameter consistency

Least squares estimation is employed for the GNAR model parameters and their consistency is established using results from Lütkepohl (2005), Section 5.2. The GNAR model can be written in the least squares framework under the assumption of independent and identically distributed innovations at each node, or the generalised least squares framework if the multivariate innovations are correlated across nodes. By using these frameworks many theoretical results can be applied to the GNAR model, and in addition, implementation of the model fitting can be done using adaptations to standard software packages. Methods such as maximum likelihood estimation, or Bayesian frameworks could be applied if distributional assumptions are made.

As shown in Equation 82, the column form of the static-network GNAR($p, [\mathbf{s}]$) model can

be written in a VAR framework as

$$\mathbf{X}_t = \phi_1 \mathbf{X}_{t-1} + \dots + \phi_p \mathbf{X}_{t-p} + \mathbf{u}_t,$$

where the matrices ϕ_i contain the network information. As described in Equation 83, in matrix form the GNAR model is $X = BZ + U$, where $X = [\mathbf{X}_{p+1}, \dots, \mathbf{X}_T]$, $B = [\phi_1, \dots, \phi_p]$, $Z = [\mathbf{Z}_p, \dots, \mathbf{Z}_{T-1}]$, with $\mathbf{Z}'_t = [\mathbf{X}_t, \dots, \mathbf{X}_{t-p+1}]$, and $U = [\mathbf{u}_{p+1}, \dots, \mathbf{u}_T]$. The constraints imposed to form a GNAR model can be written linearly as $\text{vec}(B) = R\gamma$, where R is the constraint matrix embedding the network structure of dimension $pN^2 \times M$, γ is an unrestricted parameter vector of length M , where M is defined as in Section 4.4.1, and vec is the operator that stacks the columns of a matrix into a vector.

Using the estimated generalised least squares estimator, the results of Lütkepohl (2005), Propositions 4.2 and 4.3 in this document, are applied to obtain consistency for the GNAR parameters for fixed network size. Let \otimes denote the Kronecker product and plim denote limit in probability.

Proposition 4.5 (adapted from Lütkepohl (2005) Proposition 5.2). Suppose $\{\mathbf{X}_t\}$ is an N -dimensional, stationary GNAR(p) process with a static network, whose innovations $\{\mathbf{u}_t\}$ are independent white noise with finite fourth moment, and covariance matrix Σ_u . Then, given an estimator of the innovation covariance matrix $\tilde{\Sigma}_u$, such that $\text{plim} \tilde{\Sigma}_u = \Sigma_u$, the estimated generalised least squares estimator of the unrestricted parameters,

$$\tilde{\gamma} = \left\{ R' \left(ZZ' \otimes \tilde{\Sigma}_u^{-1} \right) R \right\}^{-1} R \left(Z \otimes \tilde{\Sigma}_u^{-1} \right) \text{vec}(X),$$

is consistent; $\text{plim} \tilde{\gamma} = \gamma$ and $\sqrt{T}(\tilde{\gamma} - \gamma) \rightarrow^d N \left[0, \left\{ R' \left(\Gamma \otimes \tilde{\Sigma}_u^{-1} \right) R \right\}^{-1} \right]$ where $\Gamma = \text{plim} T^{-1} ZZ'$.

Again, using Lütkepohl (2005), the following result holds for a consistent estimator of the innovation covariance matrix in the GNAR setting.

Proposition 4.6 (adapted from Lütkepohl (2005) Proposition 5.4). A consistent estimator of Σ_u is given by

$$\tilde{\Sigma}_u = T^{-1} \left(X - \hat{B}Z \right) \left(X - \hat{B}Z \right)',$$

where $\hat{B}Z$ are the fitted values from estimating the parameters using the least squares estimator $\hat{\gamma} = \left\{ R' \left(ZZ' \otimes I_N \right) R \right\}^{-1} R' \left(Z \otimes I_N \right) \text{vec}(X)$.

Estimating the parameters with $\hat{\gamma}$ involves using the linear constraints, but assumes independent and identically distributed innovations across nodes.

These results are for GNAR processes with fixed network size, but for cases where N is increasing with T additional assumptions would be required for estimation of the covariance matrix. One example is an assumption of sparsity which was used by Fan et al. (2013) to construct a covariance estimation method for large data based upon thresholding.

4.4 Implementation

Our R package, `GNAR`, to fit, predict, and simulate GNAR processes is available on CRAN (Leeming et al. (2018)). The manual for this package is included in Appendix B. Currently, this package works using the assumption that the innovations are independent across both time and nodes, so Σ_u is a multiple of the identity matrix.

4.4.1 Order selection

The GNAR model has two types of order to select; both the maximal autoregression lag, and the maximal neighbour set for each lag. These orders can be selected using an order criterion, such as the BIC proposed by Schwarz (1978). The BIC is consistent for the VAR model (Lütkepohl, 2005, Corollary 4.2.2), and assuming a constant network and that the innovations are independent and identically distributed with bounded fourth moments, this consistency also holds for the GNAR model. All orders can be selected simultaneously by selecting the model with smallest BIC from a range of candidate models.

For the $\text{GNAR}(p, [\mathbf{s}])$ model in Definition 4.7 with N nodes and length T , the corresponding BIC criterion is

$$\text{BIC}(p, [\mathbf{s}]) = \ln |\hat{\Sigma}_{p, \mathbf{s}}| + M \ln(T)T^{-1}, \quad (86)$$

where $\hat{\Sigma}_{p, \mathbf{s}} = T^{-1} \hat{U}' \hat{U}$, with \hat{U} the residual matrix from the $\text{GNAR}(p, [\mathbf{s}])$ fit, and M the total number of parameters. For the general GNAR case $M = Np + C \sum_{j=1}^p s_j$, and for the global- α model this is reduced to $M = p + C \sum_{j=1}^p s_j$.

Similarly, the AIC criterion is

$$\text{AIC}(p, [\mathbf{s}]) = \ln |\hat{\Sigma}_{p, \mathbf{s}}| + 2MT^{-1}, \quad (87)$$

with the details as in Equation (86). However, the AIC is not a consistent estimator of the order of a VAR model (Lütkepohl, 2005, Corollary 4.2.1), therefore the BIC will be used for GNAR model selection in the remaining sections.

4.4.2 Handling missing observations

As the GNAR model allows for time-changing networks, this feature can be exploited to allow modelling of data with missing observations. When a node does not have an observation recorded at a particular time point, the connection weights at its neighbours are recalculated, so that the missing observation is not included in the analysis.

Formally, let $Z_{i,t}$ denote an indicator corresponding to variable $X_{i,t}$, such that $Z_{i,t} = 1$ if $X_{i,t}$ is observed, and $Z_{i,t} = 0$ if $X_{i,t}$ is missing from the observations. The stage- r neighbour set of Definition 4.5 is altered to deal with missing observations defining it for every time, t , and time lag, j as follows;

$$\mathcal{N}_{t,j}^{(r)}(i) = \{q : q \in \mathcal{N}^{(r)}(i), Z_{q,t-j} = 1\}. \quad (88)$$

This stage- r neighbour set only contains nodes for which an observation is recorded at the relevant time position. The connection weights are calculated as in Definition 4.6, except that the sum is over nodes in the set $\mathcal{N}_{t,j}^{(r)}(i)$. The connection weight of node i and its stage- r neighbour $q \in \mathcal{N}^{(r)}(i)$ at time t and lag j can also be written as

$$\omega_{i,q,j}^{(t)} = Z_{q,t-j} d_{i,q}^{-1} \left\{ \sum_{k \in \mathcal{N}^{(r)}(i)} Z_{k,t-j} d_{i,k}^{-1} \right\}. \quad (89)$$

Therefore the connection weight for a missing observation is zero, and the connection weights still sum to one at every node, time, and lag, for which an observation is observed within the stage- r neighbour set. The only change to the GNAR model in Definition 4.7 is the additional subscript of lag, j , on the connection weights and neighbour set.

By adjusting the connection weights in this way all of the relationships between observed variables are preserved. For example, consider the network of three nodes, A, B, C , with $A \rightsquigarrow B \rightsquigarrow C$. Here $C \in \mathcal{N}^{(2)}(A)$, and in the event of missing observations at node B , the stage-2 relationship between nodes A and C is preserved. This modelling assumes that the underlying network is fixed, and observations are unobserved using the missing

at random framework.

There is no imputation of the missing values, and gaps will be found in the fitted and residual series corresponding to gaps in the original series, at the node with missing observations. However these gaps will only be present within the time series of the node with missingness, and other nodes will have complete series.

4.5 Simulations

Simulations in this section demonstrate the GNAR model and results from Sections 4.2-4.4. In addition, the GNAR model is compared to univariate AR and multivariate VAR models fitted to generated GNAR series.

4.5.1 Comparison to theoretical results and model selection

Two GNAR models are used in this section to demonstrate the theoretical results from earlier in the chapter. Both models use an un-weighted, un-directed network with five nodes, as plotted in Figure 8. Model 1 is a global- α GNAR(1, [1]) model with parameters $\alpha = 0.2$ and $\beta = 0.5$. Model 2 is an individual- α GNAR(1, [1]) model with parameters $\alpha_A = 0.4$, $\alpha_B = \alpha_D = \alpha_E = 0.2$, $\alpha_C = -0.4$ and $\beta = 0.5$. The models have independent and identically distributed standard normal innovations at each node, so the covariance matrix is $\Sigma_u = I$.

For both models, the stationarity condition of Theorem 4.4 holds. A realisation of Model 2 using the network in Figure 8 is plotted in Figure 9. The observations at each node fluctuate around the mean at 0, but the mean stays constant over time. The second-order properties are also stationary, with the difference in α parameters shown in the observations as nodes B, D, and E show similar lag-1 autocorrelation structure, whereas node A has stronger positive autocorrelation, and node C has negative autocorrelation at lag-1.

Using Model 1, a simulation is run to check the parameter consistency result. For this simulation, 1000 realisations of length $T = 128$ were generated. Then the GNAR model of order (1, [1]) was fitted three times to each of the 1000 realisations, the first fit involving just the first $T = 32$ observations, the second using the first $T = 64$, and the third using all $T = 128$. The parameter estimates for these fits are denoted $\hat{\alpha}_T, \hat{\beta}_T$. Figure 10 shows the error in estimating the two parameters for $T = 32$ and $T = 128$. As expected from Propo-

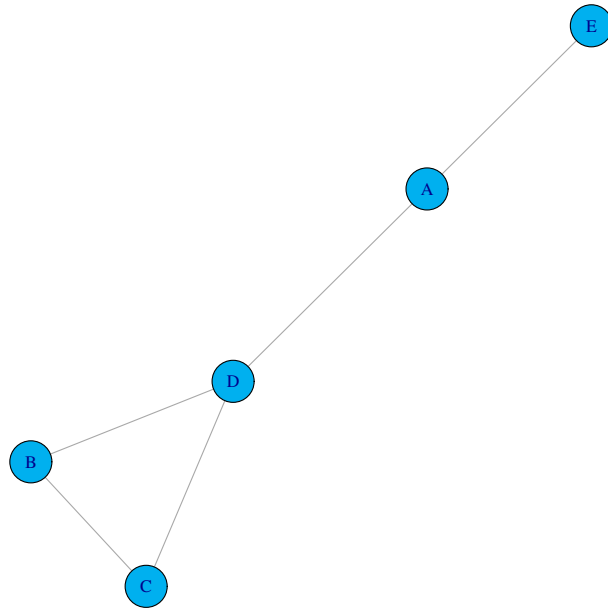


Figure 8: Five node network used in simulations.

sition 4.5, the estimates using a longer time series are more dense around the point $(0, 0)$. To check the proposition further, the bias, variance, and T multiplied by the variance are included in Table 5. As shown by Figure 10, the variance is lower for the estimates using the longer time observation of the series. Indeed, this reduction in variance is proportional to T^{-1} , shown by the values of $T * \text{var}$ in Table 5 being approximately the same for the estimates of the same parameter. This matches the result in Proposition 4.5.

When fitting the GNAR models above, the order was selected to be the same as the generating process. Next, the BIC criterion of Section 4.4.1 is demonstrated using Model 2. Table 6 shows how many realisations of Model 2 had each order as the model minimising the BIC. For $T = 32$, minimising the BIC selects the global- α GNAR(2, [1, 0]) most, and the (true) individual- α GNAR(1, [1]) was the next most selected. The correct order was selected for more than a third of simulations for $T = 32$. For $T = 64$ the correct order was selected most, with global- α GNAR(2, [1, 0]) the second most selected. For the longest tested time length $T = 128$, the simulations show the BIC selecting the true model order in 97.1% of realisations. For $T = 128$ a model that contains fewer parameters than the

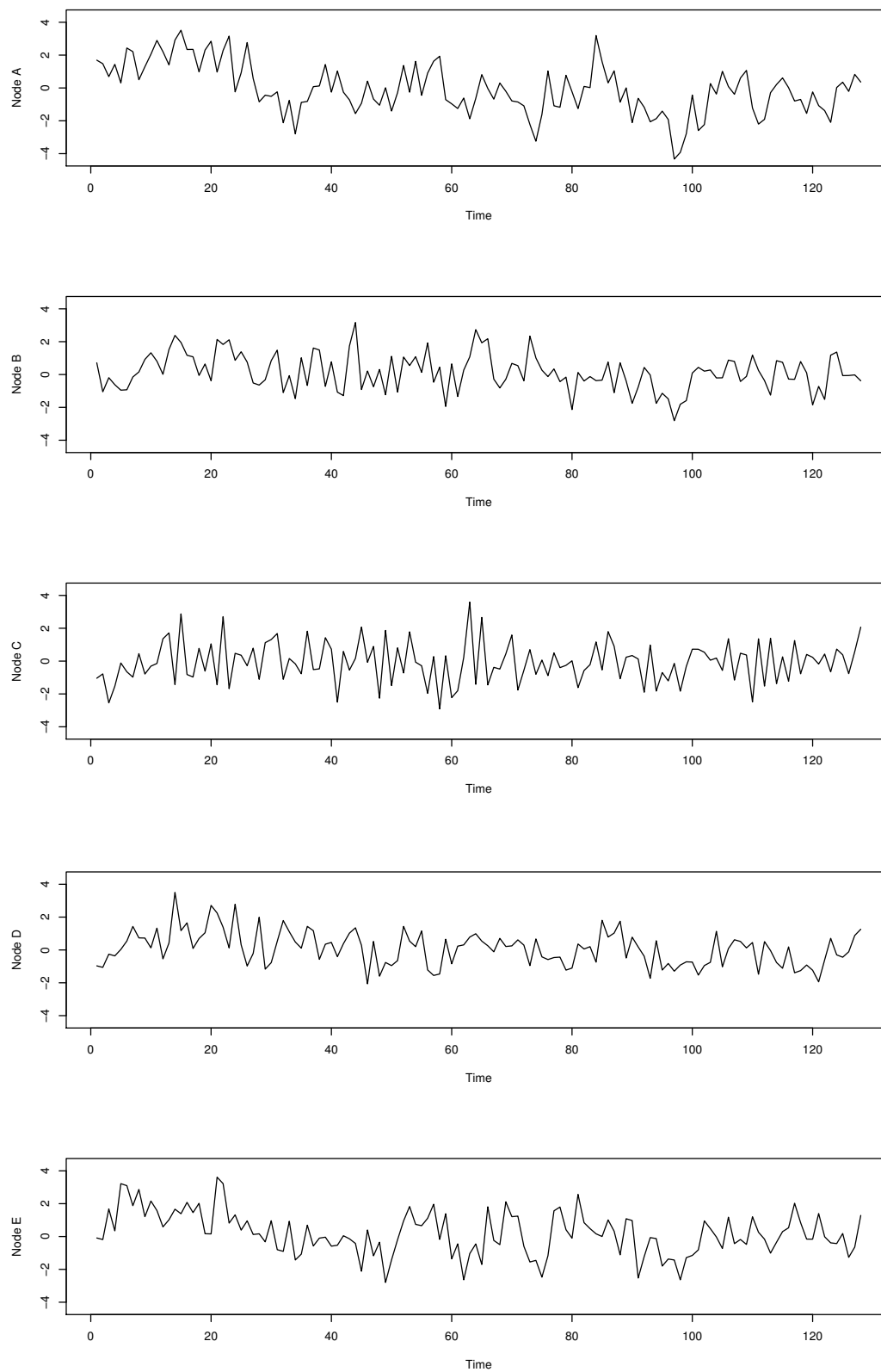


Figure 9: A realisation of Model 2 using the network in Figure 8, with $T = 128$.

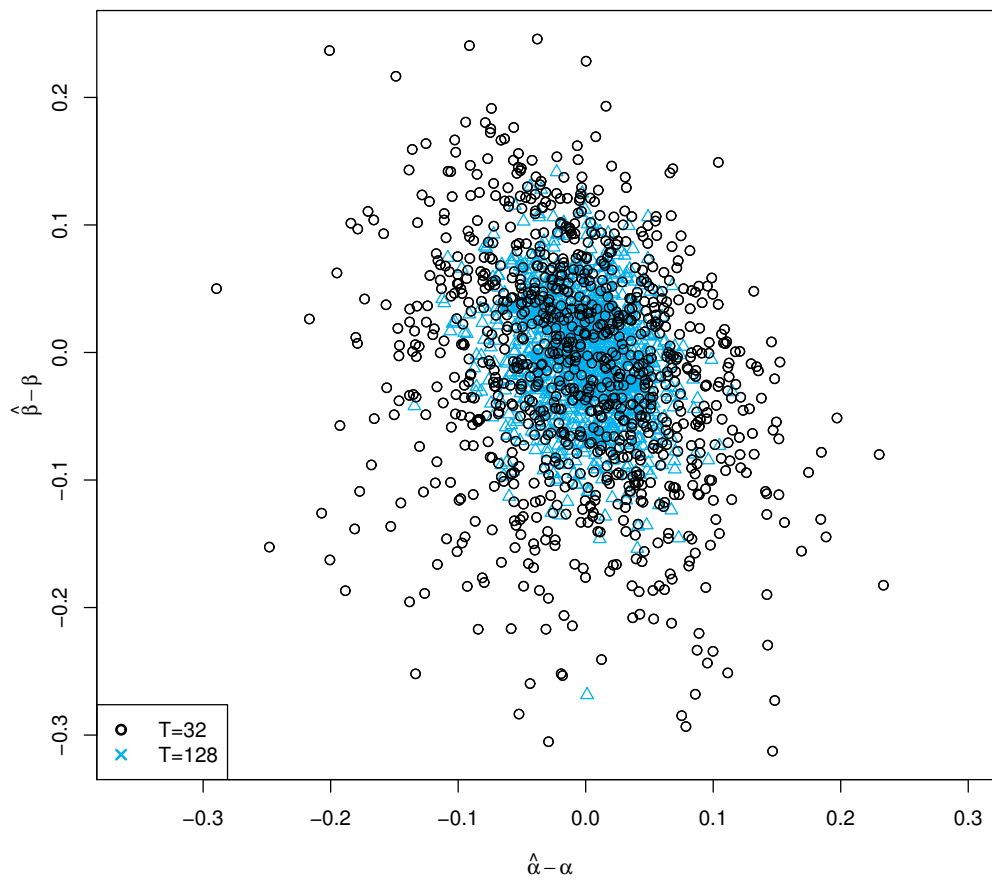


Figure 10: Difference between estimated and true parameters for 1000 realisations of Model 1.

Estimator	Bias*	Variance*	$T \times \text{var}^\dagger$
$\hat{\alpha}_{32}$	-5.39	5.26	1.68
$\hat{\alpha}_{64}$	-3.31	2.54	1.63
$\hat{\alpha}_{128}$	-2.10	1.28	1.64
$\hat{\beta}_{32}$	-13.18	8.24	2.64
$\hat{\beta}_{64}$	-7.29	4.30	2.75
$\hat{\beta}_{128}$	-3.32	2.30	2.94

Table 5: Properties of GNAR[1, (1)] global- α estimators of Model 1, using 1000 realisations. * results have been multiplied by 1000 for readability, and † results have been multiplied by 10.

true model is only chosen in 0.3% of realisations, whereas for $T = 64$ this happened in 14.2% of the realisations, and for $T = 32$ over half of the simulations had GNAR orders chosen via BIC with fewer parameters than Model 2.

Table 7 presents results involving incomplete observations to assess the properties of the GNAR model using the adaption for missing data described in Section 4.4.2. The simulations were performed over 1000 simulations with observations missing at random (MAR), or missing in a randomly placed segment, each at different positions per node and missingness at the same time positions at all nodes. For each T , 10% of the observations are removed. Table 7 demonstrates that the consistency of parameter estimation still holds in the case of missing observations as the variance decreases with sample size. The missing data simulations show bias particularly in the neighbourhood regression parameters where the missingness is at a random position per node. More time observations are affected by the missingness in this case, which results in more re-weightings than the case with all nodes featuring the missingness at the same time positions. At most this bias is 11.4% in these simulations, with the missing data results underestimating the neighbourhood regression effect, and slightly over estimating the own-node regression effect. This potential for bias in the GNAR parameter estimation should be noted when interpreting and using parameters from fitted GNAR models featuring missing observations.

4.5.2 Comparison to other models

In this section, the prediction accuracy of the GNAR model is compared to predictions made using AR and VAR models. The simulations are from Model 1 and 2 as in the previous section. The class of AR models is nested within the class of GNAR models, which in turn is nested within the VAR model class when GNAR has a static network.

T	GNAR Model order											
	Global- α						Individual- α					
	(1,[0])	(1,[1])	(1,[2])	(2,[0,0])	(2,[1,0])	(2,[1,1])	(1,[0])	(1,[1])	(1,[2])	(2,[0,0])	(2,[1,0])	(2,[1,1])
32	0	60	11	5	398	45	1	362	47	0	61	10
64	0	19	1	0	113	9	0	784	45	0	24	5
128	0	0	0	0	3	0	0	971	20	0	5	1

Table 6: Number of times each model minimised the BIC criterion, over 1000 simulations of Model 2. The bold numbers highlight the true order of Model 2.

Missing data pattern	Estimator	Bias*	Variance*
MAR per node	$\hat{\alpha}_{32}$	3.44	6.98
	$\hat{\alpha}_{64}$	8.65	3.35
	$\hat{\alpha}_{128}$	9.50	1.66
	$\hat{\beta}_{32}$	-56.47	10.98
	$\hat{\beta}_{64}$	-52.37	5.56
	$\hat{\beta}_{128}$	-46.16	2.53
MAR all nodes	$\hat{\alpha}_{32}$	-4.94	5.90
	$\hat{\alpha}_{64}$	-4.42	3.16
	$\hat{\alpha}_{128}$	-1.76	1.53
	$\hat{\beta}_{32}$	-17.17	10.20
	$\hat{\beta}_{64}$	-8.91	5.16
	$\hat{\beta}_{128}$	-4.89	2.49
Segment per node	$\hat{\alpha}_{32}$	2.85	6.42
	$\hat{\alpha}_{64}$	9.36	3.14
	$\hat{\alpha}_{128}$	9.78	1.43
	$\hat{\beta}_{32}$	-56.87	9.81
	$\hat{\beta}_{64}$	-51.20	4.94
	$\hat{\beta}_{128}$	-49.32	2.51
Segment all nodes	$\hat{\alpha}_{32}$	-8.71	6.22
	$\hat{\alpha}_{64}$	-5.09	2.95
	$\hat{\alpha}_{128}$	-1.87	1.38
	$\hat{\beta}_{32}$	-18.13	9.23
	$\hat{\beta}_{64}$	-8.38	4.57
	$\hat{\beta}_{128}$	-4.94	2.37

Table 7: Properties of GNAR[1,(1)] global- α estimators of Model 1 featuring missingness, using 1000 realisations of series of length $T = 32, 64,$ and 128 . * results have been multiplied by 1000 for readability. 10% of observations at each node are removed.

The AR model is unable to capture the network autoregression of the GNAR model, whereas the VAR model can exactly model the simulated data. Model 1 has 2 parameters in the GNAR global- α representation, 6 parameters as an individual- α GNAR, and 15 parameters when represented as a VAR model (the sum of the number of nodes and the number of stage-1 neighbours at each node). Model 2 has 6 parameters as an individual- α GNAR model, cannot be represented as a global- α model, and requires 15 parameters as a VAR model.

As the generating model is of order $p = 1$, the one-step ahead predictions of each model are compared. These tests were done using 1000 simulations of each model, with the models each fitted to $T - 1$ observations so the prediction at T could be evaluated. The AR comparisons are calculated by fitting an AR model with maximum order 1 to the univariate series at each node. This is computed using `auto.arima` from the `forecast` package Hyndman and Khandakar (2008), with S3 method `predict` used for prediction of the fitted model at each node. The AR comparison may have between 0 and 5 parameters.

The VAR comparison is calculated using the `VAR`, `restrict`, and `predict` functions from the `vars` package Pfaff (2008). This combination of functions fits a VAR model of maximum order 1, then removes any parameters with t -statistic with absolute value less than 2. The `predict` function does not work when all parameters at a node are insignificant, so when this occurs the prediction results are not calculated. Therefore the VAR models used for prediction in this simulation study will have between 5 and 25 parameters. The fitted GNAR models are of order $(1, [1])$, and have coefficients with p -value smaller than 0.05 set to zero before prediction is performed. The global- α GNAR model can be fitted with 0 to 2 parameters, and the individual- α model has from 0 to 6 parameters.

The mean prediction error is calculated as the mean over $S = 1000$ simulations of the sum of squared differences of the prediction from the observations at T ; letting $\hat{X}_{i,t}^s$ denote the prediction of $X_{i,t}$ in simulation s , this is given by $S^{-1} \sum_{s=1}^S \sum_{i=1}^N \left(X_{i,T} - \hat{X}_{i,T}^s \right)^2$. Table 8 shows the results of fitting and predicting from these models. As the VAR model could not be predicted in all cases, 660, 579, 19, and 17 simulations from each input, respectively, are not included in the VAR results.

Reassuringly, fitting the generating model results in lowest mean prediction error for each input. The AR model has highest mean prediction error in each case, showing the importance of capturing the network autoregression in modelling. Model 1 has best prediction

Input	Fitted model	Mean # parameters	Mean prediction error
Model 1, $T = 32$	AR	2.60	6.53
	VAR	7.09	6.17
	GNAR g- α	1.72	5.23
	GNAR	2.05	5.44
Model 2, $T = 32$	AR	3.16	6.66
	VAR	7.77	6.40
	GNAR g- α	1.37	5.65
	GNAR	2.97	5.43
Model 1, $T = 128$	AR	4.71	6.39
	VAR	11.66	5.58
	GNAR g- α	2.00	5.26
	GNAR	4.42	5.38
Model 2, $T = 128$	AR	4.76	5.98
	VAR	12.38	5.12
	GNAR g- α	1.84	5.42
	GNAR	5.16	4.89

Table 8: Mean number of parameters and prediction error over 1000 simulations of each model. The results from fitting the generating model are in bold.

results using its generating model, with the individual- α model increasing the mean prediction error by around 4% for $T = 32$, and 2% for $T = 128$. Although the individual- α GNAR model uses more parameters, the prediction performance is similar to that of the generating model. For $T = 32$ using the VAR model increases the mean prediction error by 18%, and for $T = 128$ is increases by 6%, so despite Model 1 being in the class of VAR models, fitting a VAR model performs poorly when the length of time observation is short.

Model 2 cannot be written as a global- α GNAR model, but this model has lower prediction error than the VAR model for $T = 32$. However for the longer series, VAR outperforms the global- α GNAR model. In both models the mean number of parameters used increases to be closer to the true number of parameters as T increases. Indeed, for Model 1 with $T = 128$ all of the 1000 simulations had exactly two parameters in the GNAR global- α model fit, the number in the generating model.

4.6 Examples

4.6.1 GDP and network construction

This section demonstrates using the GNAR model in a missing data setting as described in Section 4.4.2, in this case where the network is also unknown. The data are annual

GDP growth rates for 35 countries, obtained from the OECD website⁴. The annual values are available for the years 1961-2013, but not all countries are included in the data for every time point. We denote these observations $\{G_{i,t}\}_{i=1,\dots,35}^{j=1,\dots,53}$. To remove trend, the univariate series of each country is first-differenced, giving values $\{D_{i,t}\}_{i=1,\dots,35}^{j=1,\dots,52}$ where $D_{i,t} = G_{i,t+1} - G_{i,t}$. We associate an indicator series, $\{Z_{i,t}\}$, with the series $\{D_{i,t}\}$, such that $Z_{i,t} = \mathbb{I}(D_{i,t} \text{ is observed})$.

The data are shown graphically in Figure 11, where the missing values at the beginning of some series are evident. $T = 52$ time points are used, and there are $N = 35$ countries which form the nodes of the accompanying network. The missing data are modelled by changing the network weights over time, as described in Section 4.4.2. This data has structured missingness similar to the missing data segment simulations in Section 4.5, however it is noted that a greater proportion (21%) of the data is missing in this example.

Constructing a network to aid prediction

As the GDP data do not come with an associated network of countries, an exploratory method for constructing a network that works well in terms of prediction is examined for this example. It is noted that this is not the true underlying network, merely a useful network that aids prediction. This is found by constructing Erdős-Rényi random graphs (Definition 4.2), fitting and predicting from a GNAR model, and selecting a network that minimises the prediction error.

The Erdős-Rényi random graphs that were constructed have 35 nodes, and each pair of nodes has a network connection with fixed probability 0.15. The networks that are produced are un-weighted and un-directed. For this example 10,000 of these random networks were generated.

Not only is the network unknown, but also the order of the GNAR model. Initial analysis of the autocorrelation structure of the individual series suggested a second-order autoregressive component would be sufficient. Because of this we fit and predict from a range of GNAR model orders, with up to two autoregressive lags and parameters involving stage-1 and stage-2 neighbour sets, for both the GNAR and global- α GNAR models. Boxplots of the prediction error (the sum of squared differences between the predictions and true values) at $t = 51$ are shown in Figure 12. The prediction error for the global- α GNAR

⁴OECD (2018), Quarterly GDP (indicator). doi:10.1781/b86d1fc8-en (Accessed on 29 January 2018)

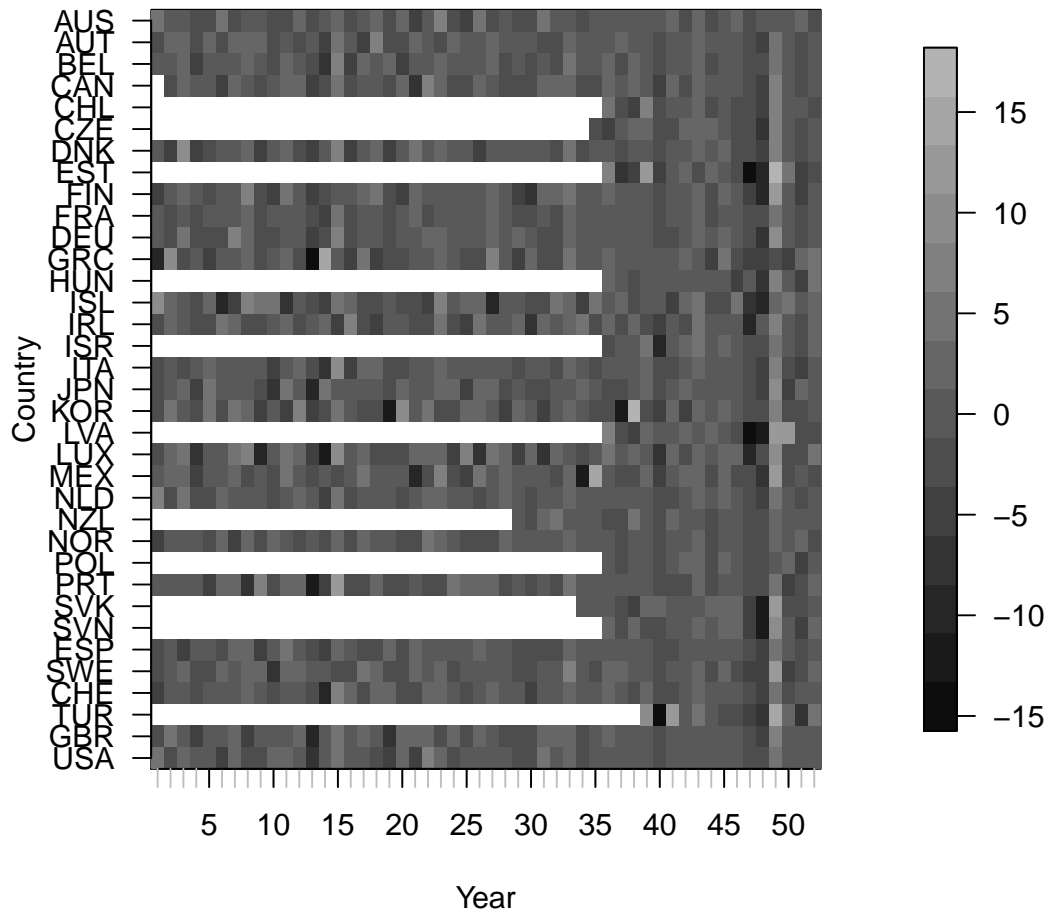


Figure 11: First-differenced annual GDP growth rates of OECD countries. White space at the beginning of the series indicates missing values.

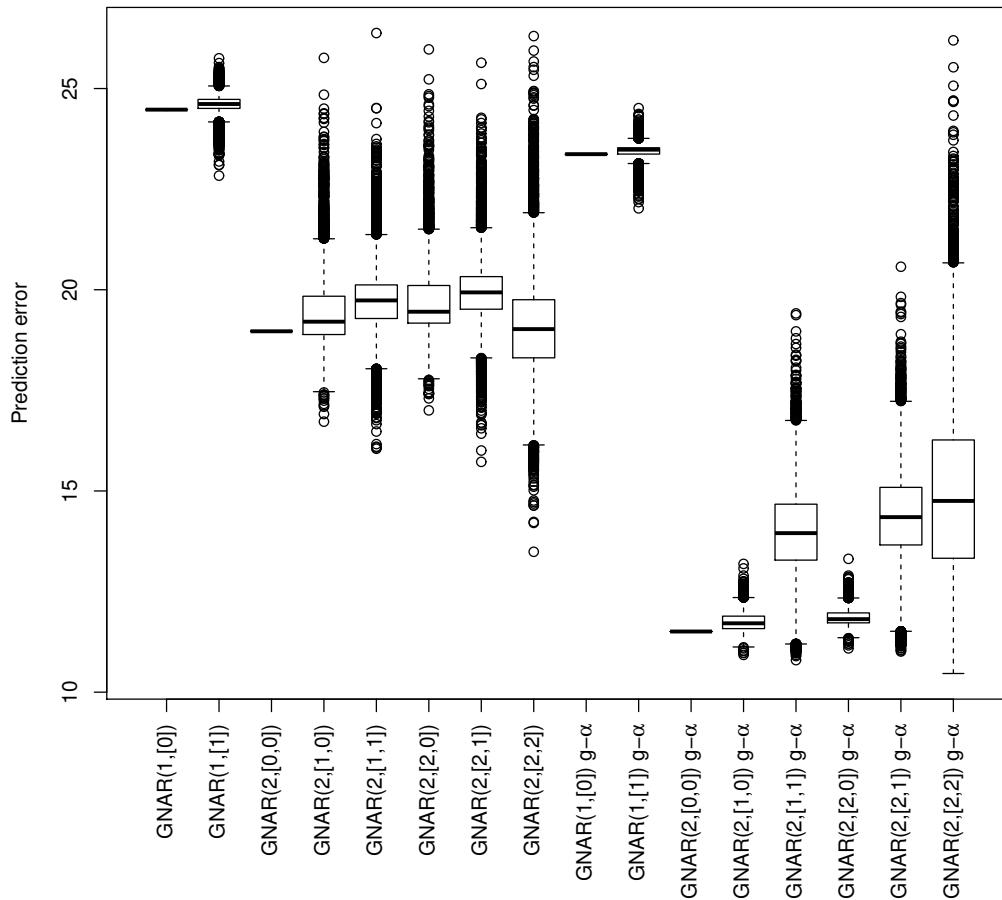


Figure 12: Boxplots of the prediction error at $t = 51$ using 10,000 random networks a range of GNAR models.

models was much lower in general, so this version of the model was used for further analysis. Of the global- α models, the lower order models are nested within the GNAR(2,[2,2]) model, so the network that minimised the prediction error for this model was chosen. This network, chosen from 10,000 randomly generated graphs, is plotted in Figure 13. The chosen network has 97 edges in total, and all countries are connected to at least two others. This network could be further tested, including dropping subgraphs or edges, to find a better performing network or find the more ‘important’ connections for predictive performance. With this chosen network the BIC from Section 4.4.1 is used to select the model with lowest value. In this example GNAR(2,[2,0]) with global- α s was the model that minimised the BIC, a model with two autoregressive parameters, and stage-1 and -2 neighbour regression parameters at the first time lag.

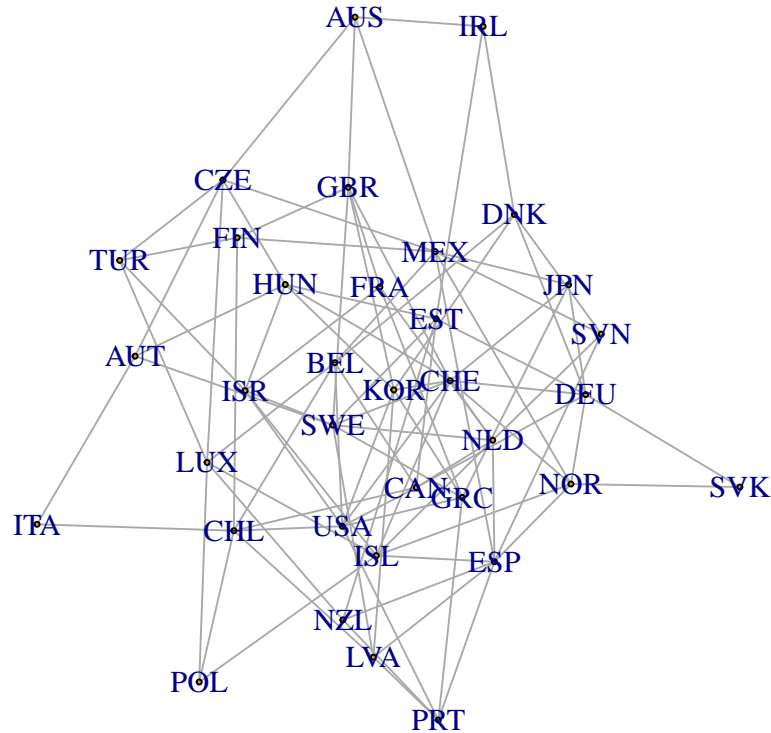


Figure 13: GDP network minimising the prediction error for the global- α GNAR(2,[2,2]) model.

Results

The model and network found above were chosen using prediction of the time series at $t = 51$, this ‘best predictive model’ is now tested on the final time point, $t = 52$. The GNAR model is compared to AR and VAR models’ prediction performance at $t = 52$.

We use the AR model individually at each node of the series. Therefore with this model no network dependence can be captured. As with the GNAR model, up to two autoregressive lags were used at each node. The order of up to two was selected at each node using the BIC. As each country was treated separately, the missing values at the beginning of each series could not be modelled, so each series for the AR model started at the first non-missing value. These comparisons were calculated using `forecast.ar()` and `auto.arima()` functions from the `forecast` package Hyndman and Khandakar (2008).

The VAR model with current software cannot handle the missing values at the beginning of the series, so these values were set to zero. For a zero-mean VAR(p) model there are pN^2 parameters, and in this example $T < 2N$ therefore there is only enough data to fit a first-order VAR model. This comparison is calculated using the `VAR` and `restrict` func-

Model	Number of Parameters	Prediction error
GNAR(2,[2,0])	4	5.74
Individual AR(2)	38	8.07
VAR(1)	199	26.20

Table 9: Prediction error of differenced GDP change at $t = 52$.

tions of the `vars` package Pfaff (2008). The restriction is a thresholding on the coefficients, setting those with a t -statistic of absolute value below two to zero.

Table 9 shows the improvement in prediction error that the GNAR model gives compared to the AR and VAR models. It is also notable that this is achieved with far fewer parameters than the other models. Even when thresholding the VAR coefficients to reduce the number included in the prediction the model has poor predictive performance. This example shows that the GNAR model with a network chosen based upon within-sample predictive performance works well compared to other models, despite not having knowledge of an underlying network.

4.6.2 Bristol Traffic

To apply the GNAR model to the Bristol Traffic data, $\{B_{i,t}\}_{i=1,\dots,29}^{t=1,\dots,512}$ of Section 3.4.2, a network is needed. To construct this network, the geographical start and end points of each section of road are identified, then any nodes that share a location are connected with an un-directed edge. For this example, edges do not have any covariates or weights associated with them, so all edges act equally in the network. The created network is plotted in Figure 14, where each node is plotted using its geographical coordinates. The numbered locations refer to the journey sections described in Table 1.

The standardised residual values after removing trend and daily seasonality at the 29 locations are plotted in Figure 15. As in the GDP example, there are many missing values in this series, with the missingness pattern in this data being similar to simulations in Section 4.5. Node 25 appears particularly sparse, with frequent gaps throughout the time series.

Initial exploratory analysis of the second-order properties of the individual series suggested that a high order of autoregression may be required for the GNAR model fit. For example, Figure 16 shows significant values in the partial autocorrelation at lags 10 and 11 for Node 3. There is clear structure both within and between the series at each node.

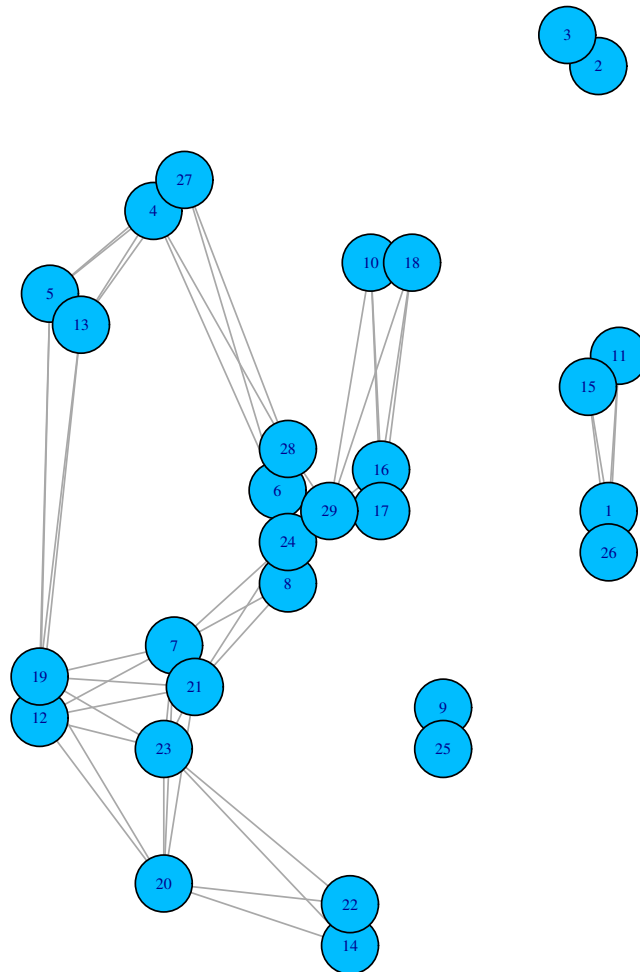


Figure 14: Bristol Traffic network, with relative geographic distances preserved.

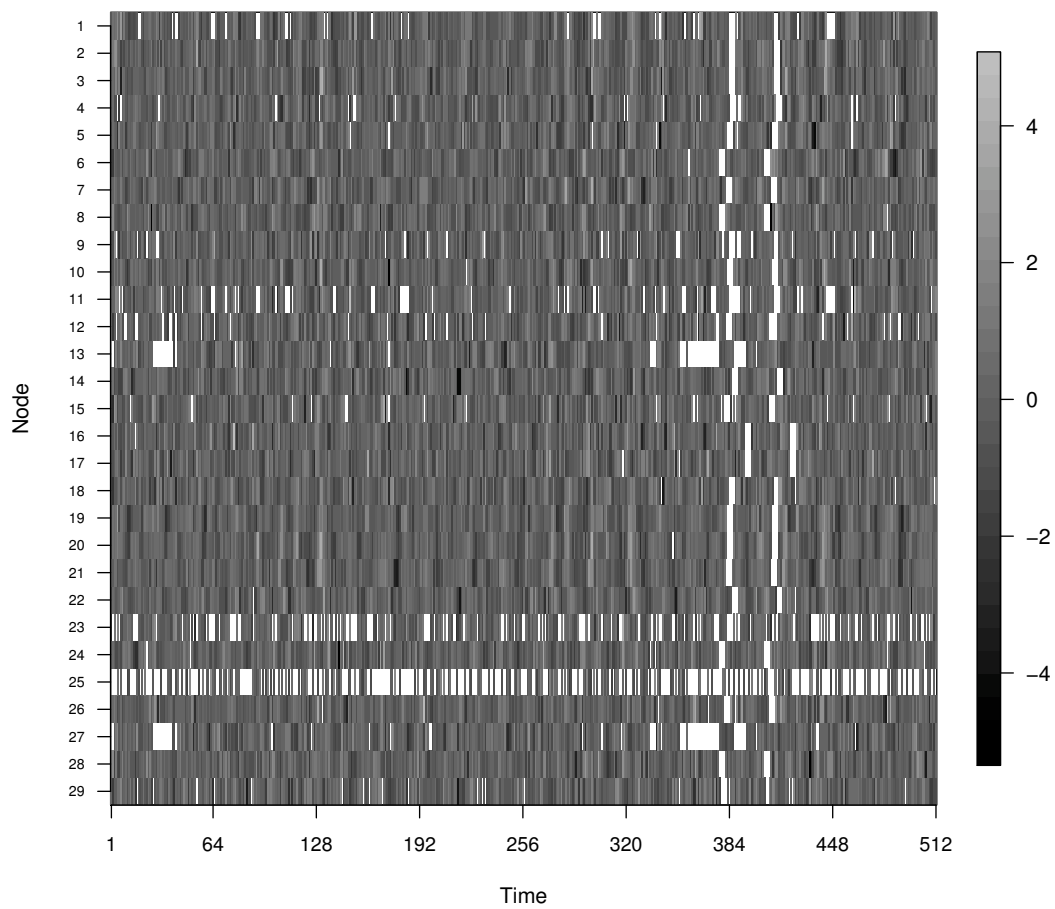


Figure 15: Residual values of Bristol Traffic time series after trend and daily seasonality is removed. The residuals at each location are standardised to have variance 1. White space indicates missing values.

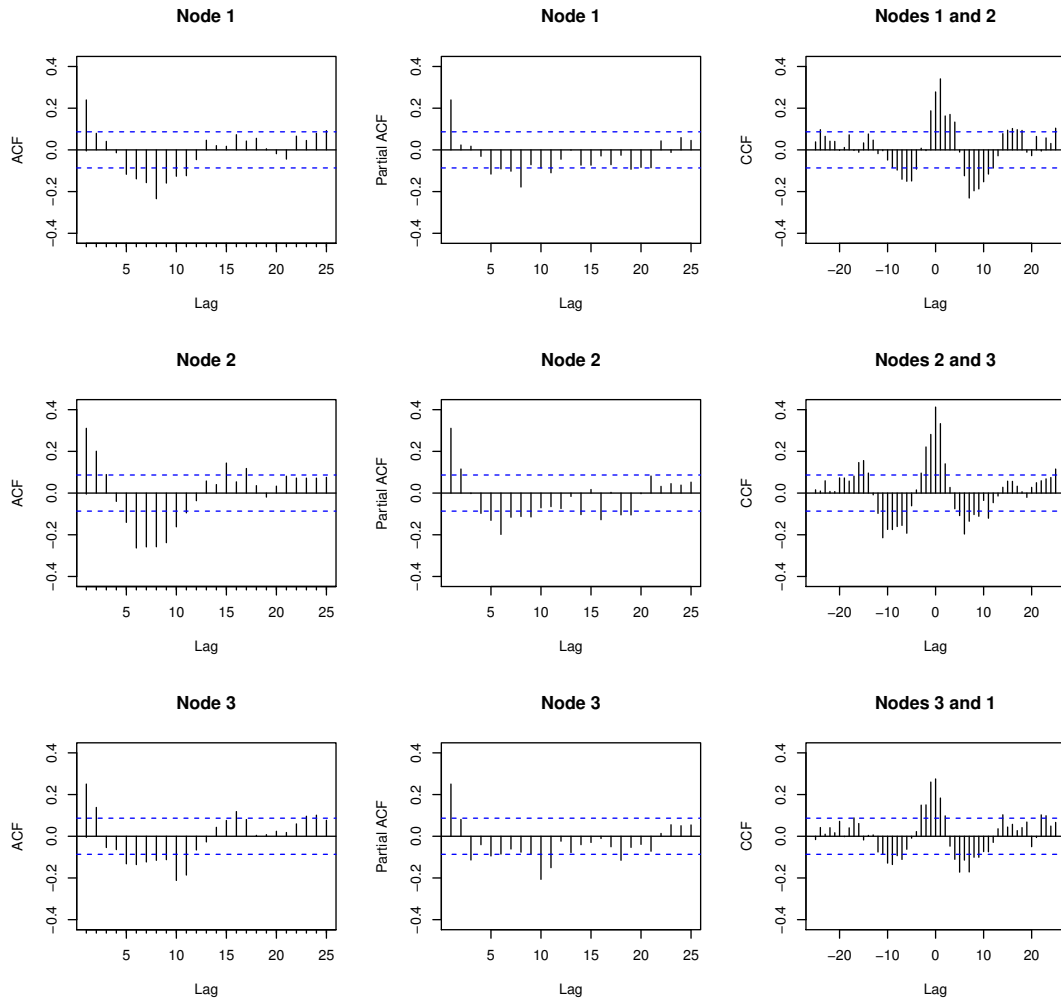


Figure 16: Autocorrelation, partial autocorrelation and cross-correlation plots for series at nodes 1-3. Missing values were replaced with zeros to calculate the correlations.

Firstly, a GNAR(11, $\mathbf{1}_{11}$) model is fitted to the standardised residuals, where $\mathbf{1}_k$ denotes a k -length vector of 1s. As the series at node 25 has many missing segments, there is not enough data to produce fitted and residual values at this node for models when $p > 4$. Node 25 is left in the model for fitting as it contributes to neighbouring series, but is removed from the residual matrix to calculate the BIC for all time lags to enable comparison between model orders. The BIC for the global- α model is -18.33 and the BIC for the individual- α GNAR model is -17.28, showing a narrow improvement for the global- α model for $p = 11$ model, which has far fewer parameters.

The BIC can also be used to compare orders of the β parameters. By investigating which parameters are deemed significant, the BIC can be lowered slightly to -18.38 by choosing $s_1 = 2$, $s_2 = s_3 = 1$, and $s_4 = \dots = s_{11} = 0$. Table 10 shows a summary of this fitted model. All parameters apart from α_2 are significant at the 5% level, including $\beta_{1,2}$ which is a second-stage neighbour parameter. All α parameters except the first are negative, which matches the general shape of the partial autocorrelation plots in Figure 16.

The second-order properties of the GNAR residuals are investigated to identify whether using the GNAR model helped to describe the structure present in the series. The autocorrelation plots in Figure 17 show far fewer autocorrelations outside of the significance bands compared with the input time series in Figure 16. Although structure remains in the multivariate time series, using the GNAR model with 15 parameters has removed a lot of the correlations present in the original. The GNAR model seems to be beneficial for this complex data set. For further analysis, different networks could also be tested, perhaps incorporating the true distances between nodes.

In Section 5.6.2, these residuals are tested for local white noise, to assess how well the GNAR model fits the data.

4.7 Conclusion

This chapter presented a generalised model for network time series and new theoretical results, illustrated by simulations. In the simple case of the GNAR model having fixed network, it can be written as a VAR model, but has many advantages over VAR such as a clearly interpretable stationarity condition, fewer parameters, and robustness to missing observations. The examples demonstrate these features and show the applicability of the

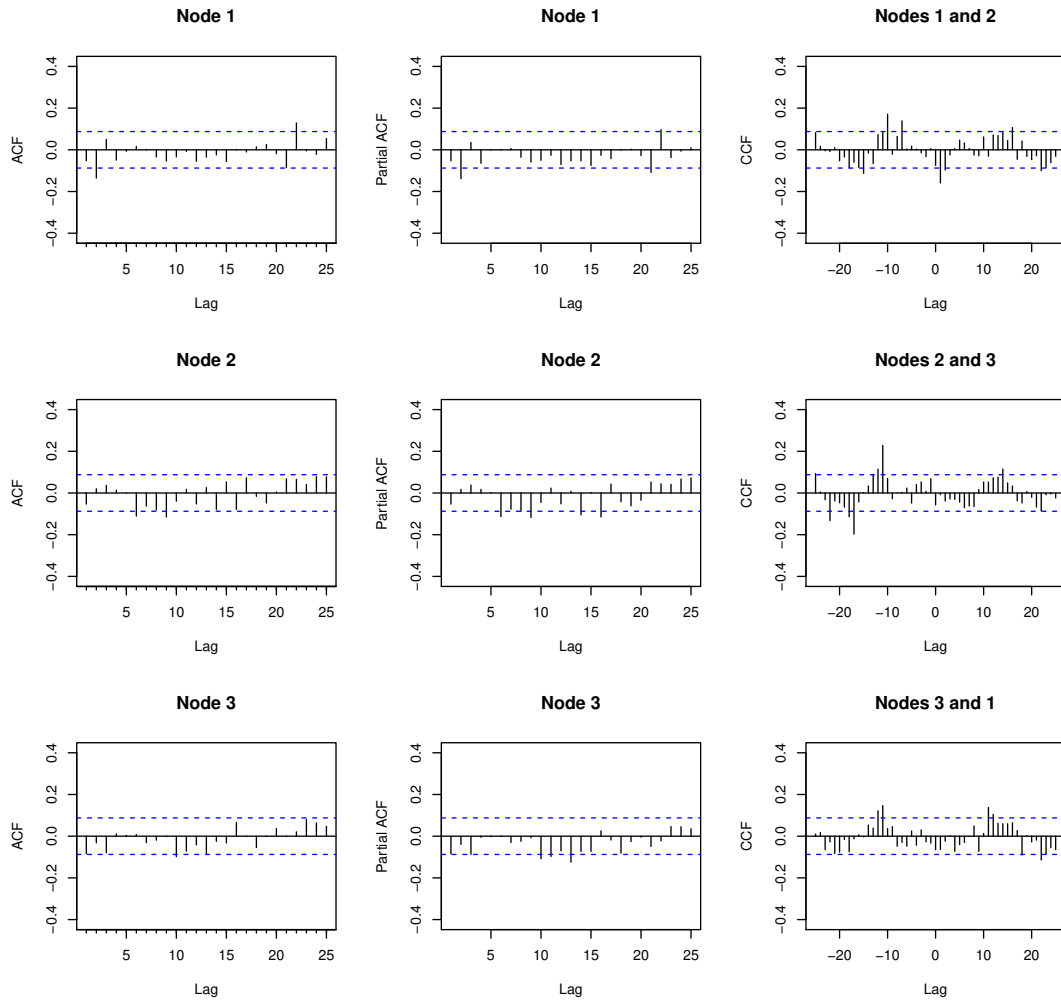


Figure 17: Autocorrelation, partial autocorrelation and cross-correlation plots for residuals of the GNAR model fit at nodes 1-3. Missing values were replaced with zeros to calculate the correlations.

Parameter	Estimate	Significance code
α_1	0.155	***
$\beta_{1,1}$	0.159	***
$\beta_{1,2}$	-0.037	*
α_2	-0.004	
$\beta_{2,1}$	0.060	***
α_3	-0.088	***
$\beta_{3,1}$	0.037	*
α_4	-0.092	***
α_5	-0.111	***
α_6	-0.114	***
α_7	-0.072	***
α_8	-0.066	***
α_9	-0.060	***
α_{10}	-0.091	***
α_{11}	-0.115	***

Table 10: Parameter estimates of the GNAR model with $p = 11$, $s_1 = 2$, $s_2 = s_3 = 1$, and $s_4 = \dots = s_{11} = 0$. Significance codes indicate the size of the p -value associated with each parameter, with * indicating a value under 0.05, ** indicating a value under 0.01, and *** indicating a value under 0.001.

GNAR model.

Further work in GNAR modelling would include additional theoretical development of including time-varying networks and establishing parameter consistency results in this case. Whilst the model allows a general covariance matrix to be used for the innovations, the current version of the GNAR package relies on the GNAR model having independent and identically distributed innovations at each node. This is more restrictive than the GNAR model described in Definition 4.7, so altering the software to allow for this would enable wider practical application. Another area of interest for further study is the second-order quantities of a GNAR process. In Section 4.6, the sample autocorrelation and partial autocorrelation at each node were calculated so that an appropriate order of the GNAR autoregression could be chosen, whereas bespoke functions could be constructed such as network autocovariance, or network cross-covariance for use in analysis and model selection of a network time series.

5 Local White Noise Testing

5.1 Introduction and additional literature review

Testing observations for white noise is commonly carried out during data analyses to identify whether further modelling of a series is needed. This usually takes the form of tests that are applied globally to the series, which may be a series of residuals from a fitted time series or regression model. However, global tests may not detect departures from white noise that occur on a section of the time series, or changes in the autocorrelation structure that average out over time.

In this chapter, a test for local white noise is presented, with our definition of local white noise described in Definition 5.2. A brief review of currently available global white noise tests follows next in this section. Section 5.2 gives definitions and key properties of the wavelet spectrum, motivating its use in our test. Relevant wavelet properties are derived in Section 5.3 and our local white noise test statistics are described in Section 5.4. Properties of our local white noise test are shown in simulations in Section 5.5, and its use is demonstrated in examples in Section 5.6.

5.1.1 White noise tests

There are many classical white noise tests, which can be split into two main categories; those that work in the time domain, and those that work in the frequency domain.

Tests in the time domain often assess whether the sample autocorrelations, $\{\hat{\rho}(h)\}_{h=1}^{\infty}$, of a signal are close to zero. The test developed by Box and Pierce (1970), and the modified version by Ljung and Box (1978) are some of the most frequently used white noise tests. These focus on testing goodness of fit of an ARMA model, and are based upon sums of the squared sample autocorrelations of the residuals. The Ljung-Box test has test statistic

$$Q_M = N(N+2) \sum_{h=1}^M \hat{\rho}^2(h)(N-h)^{-1}, \quad (90)$$

where M is the maximum time lag included in the test. The test statistic Q_M has the χ_M^2 distribution asymptotically under the null hypothesis. A key drawback of the Ljung-Box test is that the sum of squared autocorrelations is calculated up to a maximal lag, M ,

which must be chosen by the user. This choice is a trade off between an M small enough for the test to have power at small departures and low lags, and M high enough to detect departures from white noise at high lags.

A generalised test was proposed by Baragona et al. (2018) which uses a quadratic form of sample autocorrelations,

$$T_M(B) = N \sum_{h=1}^M \sum_{k=1}^M \hat{\rho}(h) \hat{\rho}(k) B_{h,k}, \quad (91)$$

where B is a symmetric, positive definite matrix of dimension $M \times M$. Tests such as the Ljung-Box test fall into this quadratic form framework. By deriving the asymptotic distribution of $T_M(B)$, the matrix B providing the maximum power for a local alternative can be chosen. This test can be utilised effectively when knowledge about alternative hypotheses is available, such as testing for underfitting of an ARMA model.

Frequency domain tests investigate whether the periodogram (Definition 2.9) is flat. An important white noise test is that due to Bartlett (1955), where the periodogram is tested for flatness using the Kolmogorov-Smirnov test, comparing the cumulative sum of normalised periodogram ordinates to the cumulative density function of a Uniform random variable. The values

$$T_q = \sum_{p=1}^q I_p \left(\sum_{p=1}^m I_p \right)^{-1}, \quad (92)$$

with $m = \text{floor}(T/2)$ can be compared to the line $U_q = qm^{-1}$ using a Kolmogorov-Smirnov statistic. This test has high power against many alternative models.

Shao (2011) used a sum of sample autocovariances, $\{\hat{\gamma}(h)\}_{h=1}^{\infty}$, a quantity related to the spectrum, and a blockwise wild bootstrap to test for white noise. The sample summation process is

$$S_n(\lambda) = \sum_{h=1}^{n-1} \sqrt{n} \hat{\gamma}(h) \left\{ (h\pi)^{-1} \sin(h\lambda) \mathbb{I}(h \neq 0) + (2\pi)^{-1} \lambda \mathbb{I}(h = 0) \right\}. \quad (93)$$

The test statistic is a Cramer-von-Mises statistic that compares the sample summation process to the Gaussian process that $S_n(\lambda)$ weakly converges to under the null hypothesis. This test is designed to hold when conditional heteroscedacity or other dependence is present.

Nason and Savchev (2014) used the Haar wavelet transform on the normalised periodogram to assess its flatness. The test used quantities

$$V_{j,k} = 2^{-j/2-1} (\pi \hat{\sigma}^2 T)^{-1} \left(\sum_{r=0}^{2^j-1} I_{2^j k-r} - \sum_{q=2^{j-1}}^{2^j-1} I_{2^j k-q} \right), \quad (94)$$

where $k = 0, \dots, 2^j - 1$ for each $j = 0, \dots, \log_2(T)$. Under the null hypothesis, $V_{j,k}$, the wavelet coefficients of the periodogram will be zero, and the null distribution when using the Haar wavelet transform was derived by modelling the periodogram ordinates as independent exponential random variables. This test benefits from not requiring any tuning parameters.

A test suitable for use against locally stationary alternatives was proposed by Goerg (2012). They defined a spurious white noise process to be one whose autocorrelations are time-varying but average to zero. Tests working on the data as a whole will generally fail to reject the null hypothesis when spurious white noise is present. The proposed modification for standard tests involves splitting a time series into K blocks of length $\text{floor}(T/K)$, calculating a local test statistic such as the Ljung-Box statistic per block $\{Q_k(M)\}_{k=1}^K$, and then summing to give a global test statistic. This global test statistic,

$$Q(M, K) = \sum_{k=1}^K Q_k(M), \quad (95)$$

has a χ_{MK}^2 distribution for the Ljung-Box test with M lags, asymptotically for increasing block length. Whilst K can increase with T , it is required that $\lim_{T \rightarrow \infty} KT^{-1} = 0$ to achieve increasing block length. The original test, such as the Ljung-Box test, fits into this framework by considering the data as a single block. The null hypothesis for this test is global white noise.

Recently, Eckley and Nason (2018) provided a test to investigate the presence of aliasing or local white noise in a signal. Using the locally stationary wavelet framework their dual test assesses whether the values of the (corrected) estimated wavelet spectrum at a particular time point are all greater than zero. This is achieved using a Student's t -statistic on the

values $\left\{ \hat{S}_{j,t_0+r} \right\}_{j=1,\dots,J^*, r=-b,\dots,b}$,

$$T_{j,t_0} = (2b)^{-1/2} \hat{\sigma}_{j,t_0}^{-1} \sum_{r=-b}^b \hat{S}_{j,t_0+r}, \quad (96)$$

where $\hat{S}_{j,\ell} = \sum_{i=1}^{J^*} A_{j,i}^{-1} I_{i,\ell}$ is the estimated wavelet spectrum, b is a chosen window width, $\hat{\sigma}_{j,t_0}^2$ is the sample variance of $\left\{ \hat{S}_{j,t_0+r} \right\}_{r=-b,\dots,b}$ and t_0 is the time position the test is centred on. For b chosen such that $b \rightarrow \infty$ as $t \rightarrow \infty$, the test statistic is shown to converge in distribution to a normal random variable. This test does not distinguish between the presence of aliasing or local white noise as both feature non-zero values at all scales of the spectrum.

5.2 Background

Throughout this section, the focus will be on Haar and Shannon wavelets. Haar wavelets were described in Section 2.2, and Shannon wavelets are introduced next.

5.2.1 Shannon wavelets

Rather than being compactly supported in the time domain, Shannon wavelets are compactly supported in the Fourier domain. We use the definition of the Shannon wavelet given in Chui (1997), Equation 4.2.4, with mother wavelet given by

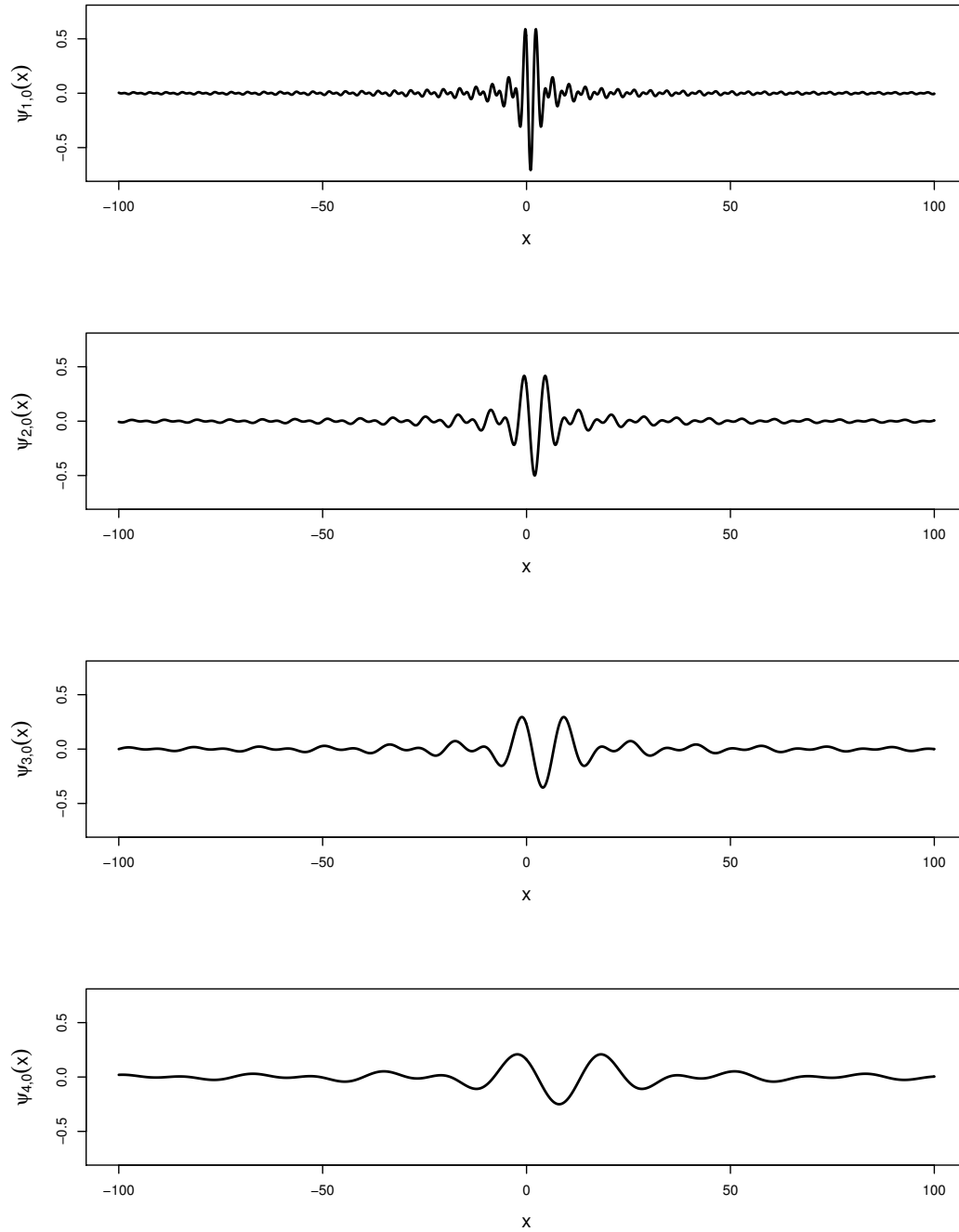
$$\psi(x) = \frac{\sin(2\pi x) - \cos(\pi x)}{\pi(x - 0.5)}, \quad (97)$$

and Fourier transformed mother wavelet (Equation (4.2.6) of Chui (1997))

$$\hat{\psi}(\omega) = -e^{-i\omega/2} \mathbb{I}[\omega \in \{[-2\pi, -\pi) \cup (\pi, 2\pi]\}]. \quad (98)$$

We assign the value at the singularity; $\psi(0.5) := -1$, the value of the limit at this point.

Figure 18 plots the first four Shannon wavelets, from these it is clear to see that despite not being compactly supported, the wavelets rapidly taper to low values at the edges.

Figure 18: Shannon mother wavelet for $j = 1, \dots, 4$.

5.2.2 The β -spectrum

In the following section we motivate the statistic chosen for our local white noise test. For this, we focus on the expected wavelet spectrum, β .

Definition 5.1 (β -spectrum (adapted from Fryzlewicz and Nason (2006), Equation 3)). For scale $j = 1, \dots, J$ and rescaled time $z \in (0, 1)$, the β -spectrum is

$$\beta_j(z) = \sum_{i=1}^{\infty} A_{i,j} S_i(z), \quad (99)$$

where $S_i(z)$ is the evolutionary wavelet spectrum at scale i and position z , and $A_{i,j}$ is the inner product of autocorrelation wavelets as Definition 2.25. The discrete wavelets, $\psi_{i,k}$, are constructed using filters as described in Equation (29).

A benefit of using the estimated β -spectrum directly rather than using it to estimate the evolutionary wavelet spectrum, S , is that it can be estimated without requiring inverse matrix calculations, so its theoretical analysis and practical calculation are simpler. As in Nason et al. (2000), we use the wavelet periodogram, $I_{j,k}$ (Definition 2.23), to estimate the β -spectrum.

Expectation of $I_{j,k}$

The key motivation behind using the β -spectrum for investigating local white noise is that the under the null hypothesis of white noise, the β -spectrum is constant over scales. The expectation of the raw wavelet periodogram for a process $\{X_t\}_{t=1}^T$ for $k \in \{1, \dots, T\}$ and $j \in \{1, \dots, J\}$ is

$$\begin{aligned} \mathbb{E}(I_{j,k}) &= \mathbb{E} \left\{ \left(\sum_{t=1}^T \psi_{j,t} X_{k-t} \right)^2 \right\} \\ &= \sum_{t=1}^T \sum_{s=1}^T \psi_{j,t} \psi_{j,s} \mathbb{E}(X_{k-t} X_{k-s}) \\ &=^{H_0} \sigma^2 \sum_{t=1}^T \psi_{j,t}^2 \\ &\begin{cases} = \sigma^2 & \text{for compactly supported wavelets} \\ =^A \sigma^2 & \text{otherwise,} \end{cases} \end{aligned} \quad (100)$$

where $=^A$ denotes the limit as $T \rightarrow \infty$, using $\mathbb{E}(X_{k-t}X_{k-s}) = \sigma^2\delta_{t,s}$ under white noise, and standard properties of wavelets.

The property in Equation (100) holds for all wavelet bases, and distributions of white noise satisfying Definition 2.1.

The wavelet periodogram may be smoothed using a kernel with bandwidth s , to give the smoothed spectrum

$$\tilde{I}_{\ell,m} = \sum_{k=m-s}^{m+s} K\{s^{-1}(m-k)\} I_{\ell,k}, \quad (101)$$

where K is a (normalised) kernel function. For example, a running mean or rectangular kernel of bandwidth s would take values $K(y) \equiv (2s+1)^{-1}$ for $y \in [0,1]$. Under the null hypothesis the smoothed values have covariance

$$\text{cov}(\tilde{I}_{\ell,m}, \tilde{I}_{j,n}) = \sum_{u=m-s}^{m+s} \sum_{v=n-s}^{n+s} K\{s^{-1}(m-u)\} K\{s^{-1}(n-v)\} \text{cov}(I_{\ell,u}, I_{j,v}), \quad (102)$$

and expectation σ^2 for compactly supported wavelets, or asymptotic expectation σ^2 otherwise. In the case of a rectangular kernel with bandwidth s , the smoothed spectrum values are

$$\tilde{I}_{\ell,m} = (2s+1)^{-1} \sum_{k=m-s}^{m+s} I_{\ell,k}, \quad (103)$$

and the covariance simplifies to

$$\text{cov}(\tilde{I}_{\ell,m}, \tilde{I}_{j,n}) = (2s+1)^{-2} \sum_{u=m-s}^{m+s} \sum_{v=n-s}^{n+s} \text{cov}(I_{\ell,u}, I_{j,v}). \quad (104)$$

Distribution of $\tilde{I}_{\ell,m}$

Given sufficient rectangular kernel smoothing of the estimated β -spectrum, we have the following distributional result, which requires that the white noise process has finite sixth moment.

Proposition 5.1. For a white noise process, $\{X_t\}_{t=1}^T$ with finite sixth moment, the rectangular kernel smoothed spectrum $\tilde{I}_{\ell,m} = (2s+1)^{-1} \sum_{k=m-s}^{m+s} I_{\ell,k}$ converges in distribution as $s \rightarrow \infty$ to the normal distribution, for $\ell \in \{1, \dots, J^*\}$ with J^* chosen to avoid boundary effects, and $m \in \{1, \dots, T\}$.

Proof. In Eckley and Nason (2018) Supplementary material 2.4, the statistic

$\text{var}(Q_{\ell,N})^{-1/2}Q_{\ell,N}$, where $Q_{\ell,N} = \sum_{k=1}^N \{I_{\ell,k} - \mathbb{E}(I_{\ell,k})\}$, is shown to converge in distribution (as $N \rightarrow \infty$) to the standard normal distribution, with rate proportional to $N^{1/2}$. This result holds for any N -length interval in $\{1, \dots, T\}$, therefore by relabelling and setting $N = 2s + 1$, it follows that the smoothed spectrum values, $N^{-1} \sum_{k=m-s}^{m+s} I_{\ell,k}$ also converge in distribution to the normal distribution. \square

Under the null hypothesis of white noise, as $s \rightarrow \infty$, the rectangular kernel smoothed estimate of the β -spectrum converges in distribution to a normal distribution with equal mean at each scale, and covariance which is derived for Haar and Shannon wavelets in Section 5.3.2. Whilst this chapter is focused on local white noise testing, the convergence rate of this result indicates the need to keep the smoothing parameter, s , high. If the test is being performed on non-overlapping intervals this will limit the number of local tests that can be performed as the theory requires $s \rightarrow \infty$.

It is likely that the normal distribution result of Proposition 5.1 also holds when using other types of kernel, under the same conditions. This is left for further work, but to allow for this, general kernels are used in the following results.

5.3 Theoretical covariance results

In this section the formula for the covariance of $I_{j,k}$ is derived, as well as quantities needed for its evaluation for Haar and Shannon wavelets.

5.3.1 Covariance of $I_{j,k}$

To use tests based upon the β -spectrum estimator, the covariance of the estimator is required. Nason (2013) provided the following formula for the asymptotic covariance of the wavelet periodogram in the case of Gaussian locally stationary wavelet processes:

$$\text{cov}(I_{\ell,m}, I_{j,n}) \simeq \begin{cases} 2 \{\mathbb{E}(I_{\ell,m})\}^2 & \text{for } \ell = j, m = n \\ 2 \left\{ \sum_k S_k \left(\frac{m+n}{2T} \right) \sum_{\tau} \Psi_k(\tau) \Psi_{\ell,j}(n - m - \tau) \right\}^2 & \text{otherwise.} \end{cases} \quad (105)$$

From Nason et al. (2000) we have that

$$c(z, \tau) = \sum_j S_j(z) \Psi_j(\tau), \quad (106)$$

so we can rewrite equation (105) as

$$\text{cov}(I_{\ell,m}, I_{j,n}) \simeq \begin{cases} 2 \{\mathbb{E}(I_{\ell,m})\}^2 & \text{for } \ell = j, m = n \\ 2 \left\{ \sum_{\tau} \Psi_{\ell,j}(n-m-\tau) c\left(\frac{m+n}{2T}, \tau\right) \right\}^2 & \text{otherwise.} \end{cases} \quad (107)$$

In addition, under the null hypothesis of white noise where $c(\tau) = \sigma^2 \delta(\tau)$ we have:

$$\begin{aligned} \text{cov}(I_{\ell,m}, I_{j,n}) &\simeq 2 \left\{ \sum_{\tau} \Psi_{\ell,j}(n-m-\tau) \sigma^2 \delta(\tau) \right\}^2 \\ &= 2\sigma^4 \Psi_{\ell,j}^2(n-m). \end{aligned} \quad (108)$$

Equation (108) will be used as a comparison to the results calculated below, noting that Gaussianity is an assumption we want to relax for our test of white noise.

Proposition 5.2. For a white noise process, $\{X_t\}_{t=1}^T$, with variance σ^2 , the covariance of $I_{\ell,m}$ is

$$\text{cov}(I_{\ell,m}, I_{j,n}) = 2\sigma^4 \Psi_{\ell,j}^2(n-m) + \{\mathbb{E}(X_t^4) - 3\sigma^4\} \sum_t \psi_{\ell,m-t}^2 \psi_{j,n-t}^2,$$

where $\ell \in \{1, \dots, J\}$, $m \in \{1, \dots, T\}$, ψ is the analysis wavelet used, and Ψ is the corresponding cross-correlation wavelet.

Proof.

$$\begin{aligned} \text{cov}(I_{\ell,m}, I_{j,n}) &= \mathbb{E}(I_{\ell,m} I_{j,n}) - \mathbb{E}(I_{\ell,m}) \mathbb{E}(I_{j,n}) \\ &= \sum_{t,s,p,q} \psi_{\ell,m-t} \psi_{\ell,m-s} \psi_{j,n-p} \psi_{j,n-q} \mathbb{E}(X_t X_s X_p X_q) \\ &\quad - \left\{ \sum_{t,s} \psi_{\ell,m-t} \psi_{\ell,m-s} \mathbb{E}(X_t X_s) \right\} \left\{ \sum_{p,q} \psi_{j,n-p} \psi_{j,n-q} \mathbb{E}(X_p X_q) \right\} \\ &= \sum_{t,s,p,q} \psi_{\ell,m-t} \psi_{\ell,m-s} \psi_{j,n-p} \psi_{j,n-q} \mathbb{E}(X_t X_s X_p X_q) \\ &\quad - \left\{ \sigma^2 \sum_t \psi_{\ell,m-t}^2 \right\} \left\{ \sigma^2 \sum_p \psi_{j,n-p}^2 \right\} \\ &= \sum_{t,s,p,q} \psi_{\ell,m-t} \psi_{\ell,m-s} \psi_{j,n-p} \psi_{j,n-q} \mathbb{E}(X_t X_s X_p X_q) - \sigma^4 \end{aligned} \quad (109)$$

Where we use that $\text{cov}(X_t X_s) = 0$ for $t \neq s$ as the process is white noise, $\mathbb{E}(X_t^2) = \sigma^2$, for all t , and discrete wavelets have L_2 norm of 1.

Considering the fourth order expectation, under the assumption that we have a mean-zero

white noise process, the odd moments will be zero. Hence we only need to deal with cases where all indices match, or we have two pairs of indices.

$$\begin{aligned} \mathbb{E}(X_t X_s X_p X_q) &= \delta_{t,s} \delta_{p,q} \delta_{t,p} \mathbb{E}(X_t^4) \\ &\quad + (1 - \delta_{t,s} \delta_{p,q} \delta_{t,p}) (\delta_{t,s} \delta_{p,q} + \delta_{t,p} \delta_{s,q} + \delta_{t,q} \delta_{s,p}) \{\mathbb{E}(X_t^2)\}^2 \end{aligned} \quad (110)$$

The first term in the expression occurs when all of the indices are the same, and the second occurs when the indices match in pairs but are not all the same. Replacing this into the previous expression gives:

$$\begin{aligned} \text{cov}(I_{\ell,m}, I_{j,n}) &= \sum_{t,s,p,q} \psi_{\ell,m-t} \psi_{\ell,m-s} \psi_{j,n-p} \psi_{j,n-q} \{ \delta_{t,s} \delta_{p,q} \delta_{t,p} \mathbb{E}(X_t^4) \\ &\quad + (1 - \delta_{t,s} \delta_{p,q} \delta_{t,p}) (\delta_{t,s} \delta_{p,q} + \delta_{t,p} \delta_{s,q} + \delta_{t,q} \delta_{s,p}) \sigma^4 \} - \sigma^4 \\ &= \mathbb{E}(X_t^4) \sum_t \psi_{\ell,m-t}^2 \psi_{j,n-t}^2 \\ &\quad + \sigma^4 \sum_{t,s,p,q} \psi_{\ell,m-t} \psi_{\ell,m-s} \psi_{j,n-p} \psi_{j,n-q} (\delta_{t,s} \delta_{p,q} + \delta_{t,p} \delta_{s,q} + \delta_{t,q} \delta_{s,p}) \\ &\quad - \sigma^4 \sum_{t,s,p,q} \delta_{t,s} \delta_{p,q} \delta_{t,p} \psi_{\ell,m-t} \psi_{\ell,m-s} \psi_{j,n-p} \psi_{j,n-q} (\delta_{t,s} \delta_{p,q} + \delta_{t,p} \delta_{s,q} + \delta_{t,q} \delta_{s,p}) \\ &\quad - \sigma^4 \\ &= \mathbb{E}(X_t^4) \sum_t \psi_{\ell,m-t}^2 \psi_{j,n-t}^2 + \sigma^4 \{1 + 2\Psi_{\ell,j}^2(n-m)\} \\ &\quad - 3\sigma^4 \sum_t \psi_{\ell,m-t}^2 \psi_{j,n-t}^2 - \sigma^4 \\ &= 2\sigma^4 \Psi_{\ell,j}^2(n-m) + \{\mathbb{E}(X_t^4) - 3\sigma^4\} \sum_t \psi_{\ell,m-t}^2 \psi_{j,n-t}^2. \end{aligned} \quad (111)$$

□

In the Gaussian case, $\mathbb{E}(X_t^4) = 3\sigma^4$, and the covariance is reduced to $2\sigma^4 \Psi_{\ell,j}^2(n-m)$ which matches Equation (108). For the Gaussian case, the theorem of Isserlis (1918) can be applied to Equation (109) for a shorter derivation of the result, as in Appendix A of Nason (2013).

Equation (111) shows that, unlike the expectation, the covariance of the estimated β -spectrum depends both on the choice of wavelet, and the distribution of the white noise process. The dependence on the wavelet occurs via the cross-correlation wavelet, Ψ , and

the dependence on the white noise is via the excess kurtosis multiplied by σ^4 ; $\mathbb{E}(X_t^4) - 3\sigma^4$.

5.3.2 Covariance of $I_{j,k}$ under smoothing

As is the case with other spectra, the wavelet periodogram requires smoothing for consistent estimation of the β -spectrum. Therefore we also require the values of the covariance of the kernel smoothed estimate.

We consider a general kernel smoothed β -spectrum as this will allow for a range of smoothing to be carried out. Letting the kernel smoothed spectrum be denoted by $\tilde{I}_{\ell,m} = \sum_{u=m-s}^{m+s} K\left(\frac{m-u}{s}\right) I_{\ell,u}$ for a normalised kernel function K , with bandwidth s , we can calculate the covariance as

$$\begin{aligned} \text{cov}(\tilde{I}_{\ell,m}, \tilde{I}_{j,n}) &= \text{cov} \left\{ \sum_{u=m-s}^{m+s} K\left(\frac{m-u}{s}\right) I_{\ell,u}, \sum_{v=n-s}^{n+s} K\left(\frac{n-v}{s}\right) I_{j,v} \right\} \\ &= \sum_{u=m-s}^{m+s} \sum_{v=n-s}^{n+s} K\left(\frac{m-u}{s}\right) K\left(\frac{n-v}{s}\right) \text{cov}(I_{\ell,u}, I_{j,v}). \end{aligned} \quad (112)$$

5.3.3 Haar wavelet quantities

For the covariance, the cross-correlation wavelet and sum of squared wavelets is required.

Cross-correlation wavelet

Proposition 5.3 (Killick et al. (2019)). For $\ell < j$, the cross-correlation Haar wavelet is

$$\Psi_{j,\ell}(\tau) = 2^{-(j-\ell)/2} \begin{cases} 0 & \text{for } \tau < -2^\ell, \\ -(2^{-\ell}\tau + 1) & \text{for } -2^\ell \leq \tau < -2^{\ell-1}, \\ 2^{-\ell}\tau & \text{for } -2^{\ell-1} \leq \tau < 0, \\ 0 & \text{for } 0 \leq \tau < 2^{j-1} - 2^\ell, \\ 2^{-\ell}(2\tau - 2^j + 2^{\ell+1}) & \text{for } 2^{j-1} - 2^\ell \leq \tau < 2^{j-1} - 2^{\ell-1}, \\ 2^{-\ell}(2^j - 2\tau) & \text{for } 2^{j-1} - 2^{\ell-1} \leq \tau < 2^{j-1}, \\ 0 & \text{for } 2^{j-1} \leq \tau < 2^j - 2^\ell, \\ 2^{-\ell}(2^j - \tau - 2^\ell) & \text{for } 2^j - 2^\ell \leq \tau < 2^j - 2^{\ell-1}, \\ 2^{-\ell}(\tau - 2^j) & \text{for } 2^j - 2^{\ell-1} \leq \tau < 2^j, \\ 0 & \text{for } 2^j \leq \tau. \end{cases} \quad (113)$$

and $\Psi_{j,\ell}(\tau) = \Psi_{\ell,j}(-\tau)$ for $\ell > j$, and $\Psi_{j,j}(\tau) = \Psi_j(\tau)$, the regular autocorrelation wavelet.

Sum of squared wavelets

Proposition 5.4. Letting $a = \max(m, n)$, and $b = \min(2^\ell + m, 2^j + n)$, the value of the sum of non-decimated Haar squared wavelets is given by:

$$\sum_t \psi_{\ell,m-t}^2 \psi_{j,n-t}^2 = \begin{cases} 2^{-(\ell+j)}(b-a) & \text{for } a < b \\ 0 & \text{otherwise} \end{cases} \quad (114)$$

Proof.

$$\begin{aligned} \sum_t \psi_{\ell,m-t}^2 \psi_{j,n-t}^2 &= \int_t \psi_{\ell,m-t}^2 \psi_{j,n-t}^2 dt \\ &= 2^{-(j+\ell)} \int_t \mathbb{I}[t \in \{n, 2^j + n\}] \mathbb{I}[t \in \{m, 2^\ell + m\}] dt \end{aligned} \quad (115)$$

Hence only when the two intervals overlap is there any contribution, and the result follows. \square

5.3.4 Shannon wavelet quantities

Cross-correlation wavelet

Proposition 5.5. The cross-correlation Shannon wavelet is

$$\Psi_{j,\ell}(k) = \begin{cases} 0 & \text{for } j \neq \ell \\ 1 & \text{for } j = \ell \text{ and } k = 0 \\ u_{jk} \cos(3u_{jk}^{-1}) \sin(u_{jk}^{-1}) & \text{for } j = \ell \text{ and } k \neq 0 \end{cases}$$

where $u_{jk} = 2^{j+1}(k\pi)^{-1}$.

Proof. Because the Fourier transform of Shannon wavelets do not overlap when $j \neq \ell$, the Shannon cross-correlation wavelet is zero for $j \neq \ell$. Therefore we only have to consider the cross-correlation where scales are equal, which is the autocorrelation wavelet. That is $\hat{\Psi}_{j,j}(\omega) = \hat{\Psi}_j(\omega)$. Using the inverse Fourier transform and following a similar method to the proof of Theorem 2 in Supplementary material of Eckley and Nason (2018) we can calculate the Shannon correlation wavelet at $k = 0$.

$$\begin{aligned}
\Psi_j(0) &= \sum_t \psi_{j,t} \psi_{j,t} \\
&= (2\pi)^{-1} \sum_t \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \hat{\psi}_j(\omega) e^{it\omega} \hat{\psi}_j(\nu) e^{it\nu} d\omega d\nu \\
&= (2\pi)^{-1} \sum_t \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \hat{\psi}_j(\omega) e^{it\omega} \hat{\psi}_j(-\mu) e^{-it\mu} d\omega d\mu \\
&= (2\pi)^{-1} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \hat{\psi}_j(\omega) \overline{\hat{\psi}_j(\mu)} \sum_t e^{it(\omega-\mu)} d\omega d\mu \\
&= (2\pi)^{-1} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \hat{\psi}_j(\omega) \overline{\hat{\psi}_j(\mu)} \sum_n \delta(\omega - \mu + 2\pi n) d\omega d\mu \\
&= (2\pi)^{-1} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \hat{\psi}_j(\omega) \overline{\hat{\psi}_j(\mu)} \delta(\omega - \mu) d\omega d\mu \\
&= (2\pi)^{-1} \int_{-\pi}^{\pi} \hat{\psi}_j(\omega) \overline{\hat{\psi}_j(\omega)} d\omega \\
&= (2\pi)^{-1} \int_{-\pi}^{\pi} 2^j \chi_{C_j}(\omega) d\omega \\
&= (2\pi)^{-1} 2^j \{-\pi 2^{-j} + \pi 2^{-(j-1)} + \pi 2^{-(j-1)} - \pi 2^{-j}\} \\
&= (2\pi)^{-1} 2^j \{\pi 2^{-(j-1)}\} \\
&= 1
\end{aligned} \tag{116}$$

Where line 2 to 3 is a change of variables; $\mu = -\nu$, line 3 to 4 holds as for wavelets we have the property $\overline{\hat{\psi}_j(\omega)} = \hat{\psi}_j(-\omega)$, line 4 to 5 holds by the Poisson sampling theorem. Line 5 to 6 is due to the limits of the integral, 6 to 7 the application of the Kronecker delta function.

Using the proof of Theorem 2 in the Supplementary material of Eckley and Nason (2018) we can determine the value of the Shannon autocorrelation wavelet for $k \neq 0$. We note that the quantity $P(j, k, \ell, m, r) = \left(\sum_t \psi_{j,k-2^r t} \psi_{\ell, m-t} \right)^2$ is the squared autocorrelation wavelet when $\ell = j$, $m = 0$ and $r = 0$; $P(j, k, j, 0, 0) = \left(\sum_t \psi_{j,k-t} \psi_{j,-t} \right)^2 = \Psi_j^2(k)$.

Using equation (5) from Eckley and Nason (2018) Supplementary material with the above values for the variables we have

$$\Psi_j^2(k) = 2^{2(j+1)} (k\pi)^{-2} \cos^2 \left\{ 2^{-(j+1)} 3\pi k \right\} \sin^2 \left\{ 2^{-(j+1)} \pi k \right\}. \tag{117}$$

□

The first six Shannon autocorrelation wavelets are plotted in Figure 19, and with Haar autocorrelation wavelets overlaid in Figure 20. The autocorrelation wavelets for Haar and Shannon wavelets are similar in shape, but the Shannon autocorrelation wavelets take longer to flatten out as the magnitude of k increases.

The autocorrelation result of Proposition 5.2 using the Shannon autocorrelation wavelet of Proposition 5.5 is confirmed using a small simulation study. Standard normal samples of length 256 were generated, and over 10,000 simulations of a white noise series, the covariance of the estimated Shannon β -spectra was calculated. Using the results calculated in previous sections the covariance in this case should follow $\text{cov}(I_{\ell,m}, I_{j,n}) = 2\sigma^4 \mathbb{I}(\ell = j) \Psi_j^2(n - m)$. The simulations used the central 16 observations, with $\text{cov}(I_{\ell,129}, I_{j,129+k})$ for $k = 0, \dots, 15$. The first three scales plotted in Figure 21 show agreement between the theoretical results and simulations.

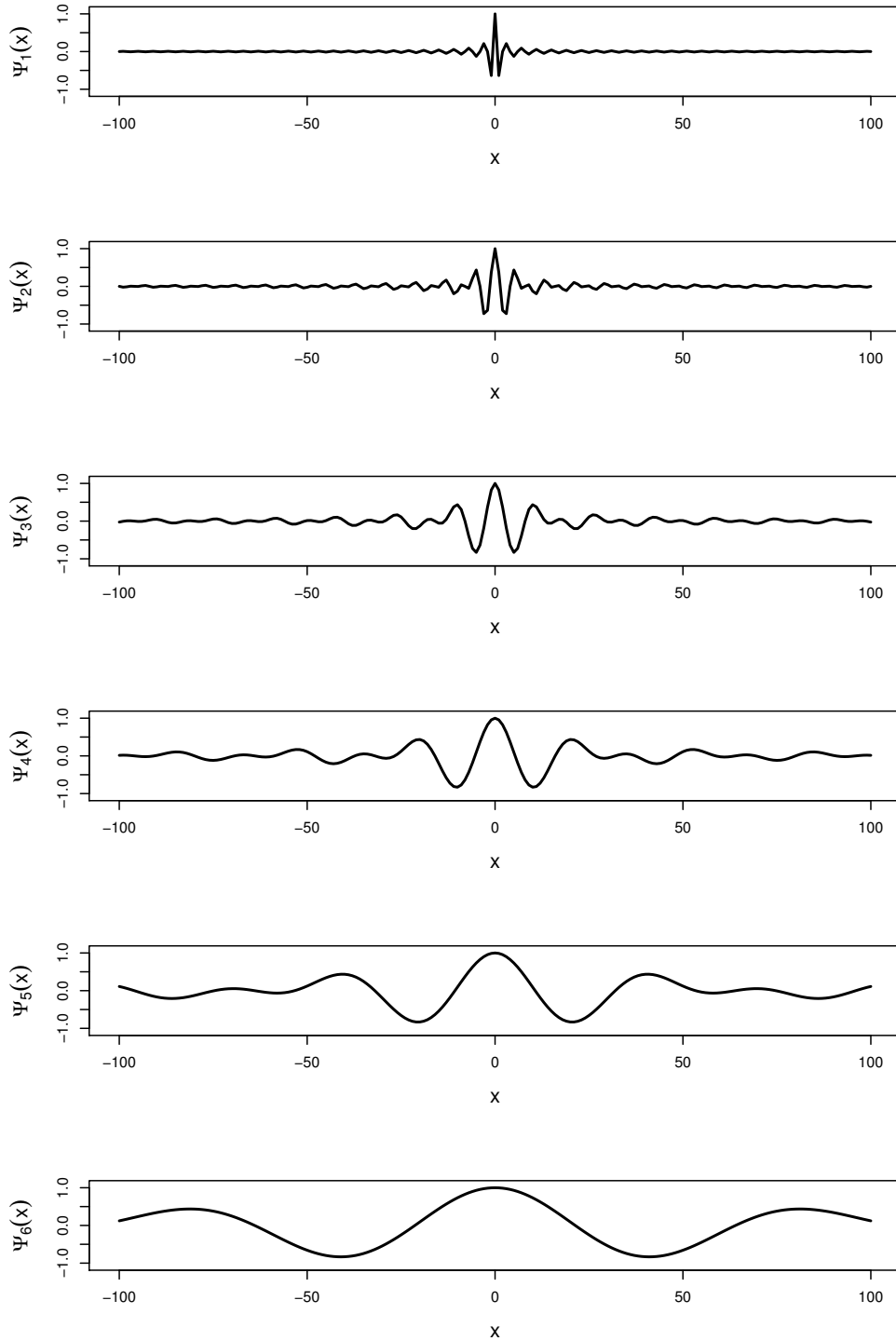
Sum of squared wavelets

We use similar steps to those in the previous section to show that the sum of squared values in the kurtosis term is zero for unequal scales. Replacing the wavelets with the inverse Fourier transform of their Fourier duals gives

$$\sum_t \psi_{\ell, m-t}^2 \psi_{j, n-t}^2 = (2\pi)^2 \sum_t \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \hat{\psi}_{\ell}(\omega) \hat{\psi}_{\ell}(\nu) \hat{\psi}_j(\mu) \hat{\psi}_j(\xi) e^{i\omega(m-t)} e^{i\nu(m-t)} e^{i\mu(n-t)} e^{i\xi(n-t)} d\omega d\nu d\mu d\xi, \quad (118)$$

which evaluates to zero in the case that $\ell \neq j$ because the supports of the Fourier transform of Shannon wavelets do not overlap, as shown in Nason et al. (2000).

Although an analytic form has yet to be found for this quantity, we can calculate an approximation to the summation computationally in the time domain. This is shown in Figure 22. The values in Figure 22 decay quickly as the locations m and n get further apart. The values of the peaks at $m = n$ are $1/3, 1/8, 1/16, 1/32$ for the scales $j = 1, \dots, 4$.

Figure 19: Shannon autocorrelation wavelets for $j = 1, \dots, 6$.

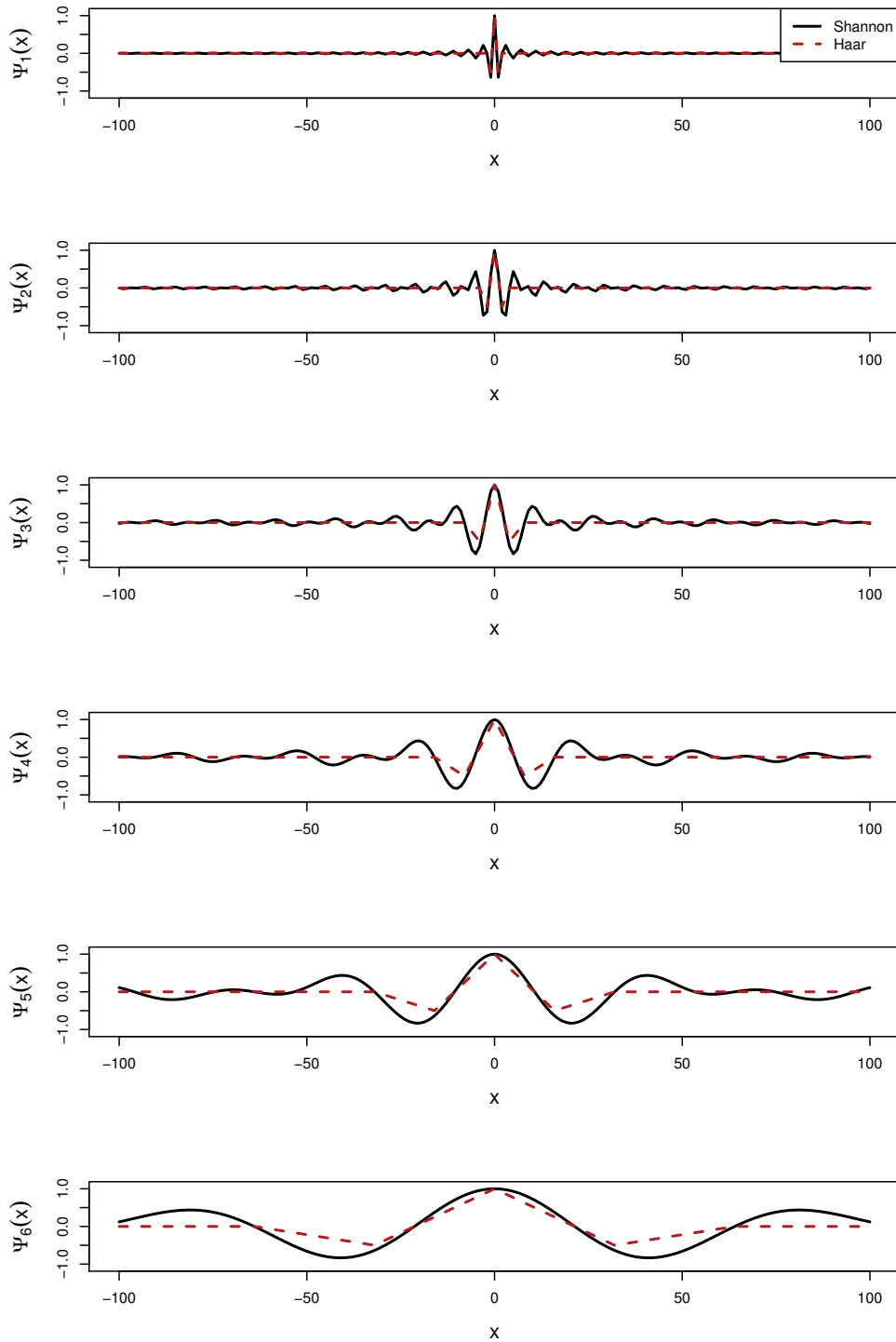


Figure 20: Shannon (black) and Haar (red dashed) autocorrelation wavelets for $j = 1, \dots, 6$.

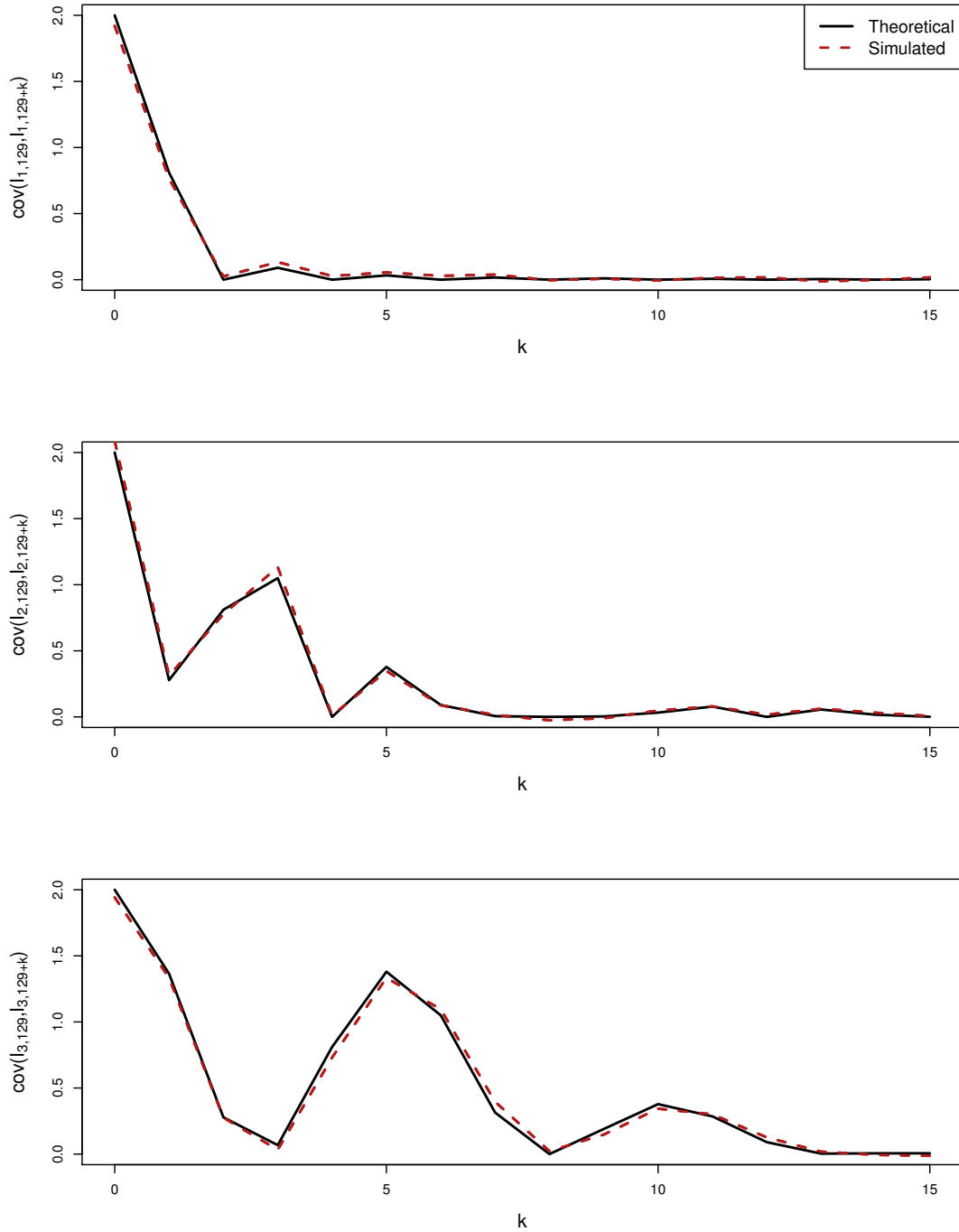


Figure 21: Theoretical covariance (black) and simulated covariance (red dashed) of the Shannon wavelet spectrum over 10,000 simulations of standard Normal white noise with $T = 256$.

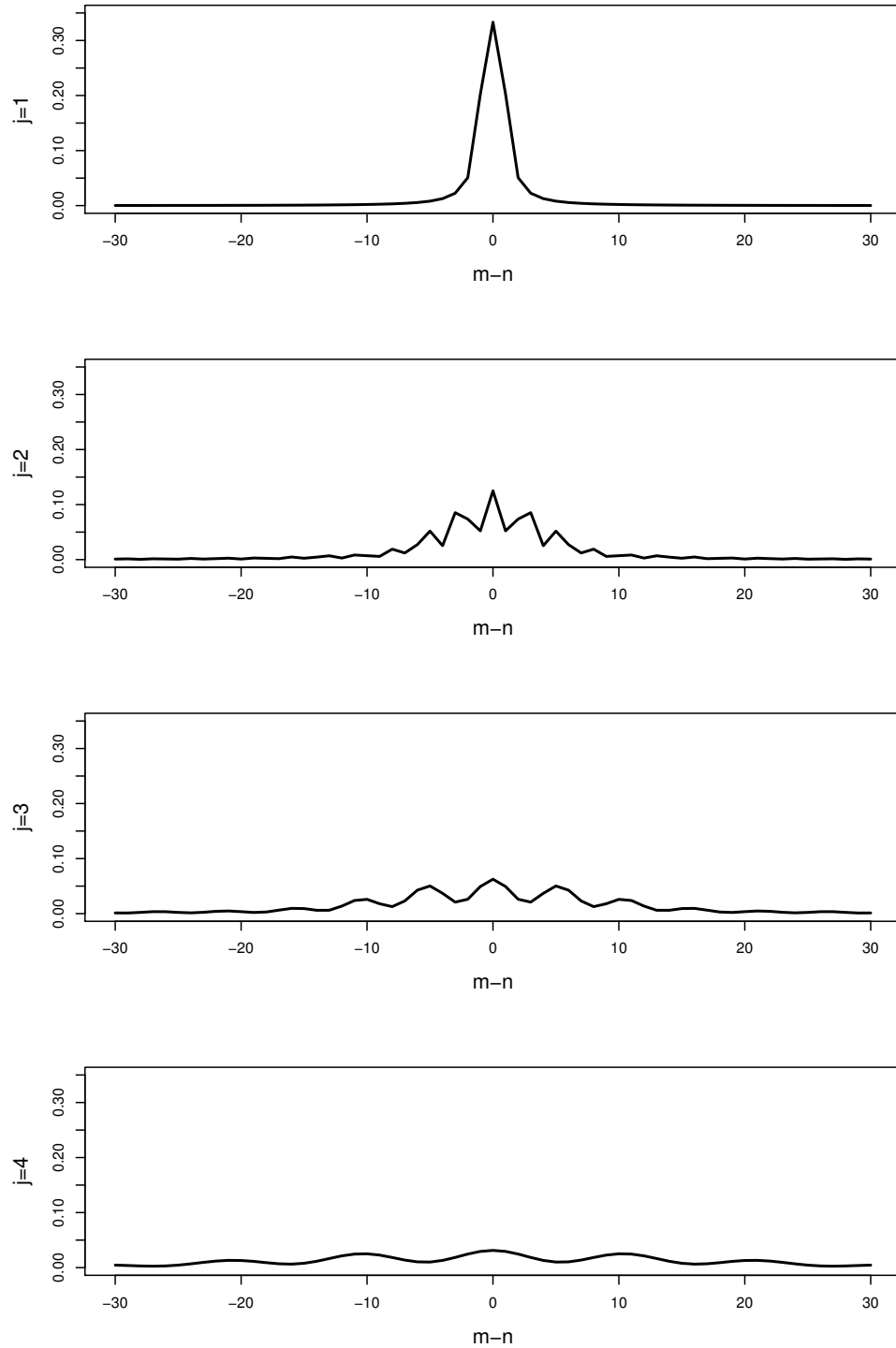


Figure 22: Values of $\sum_t \psi_{\ell, m-t}^2 \psi_{j, n-t}^2$ for Shannon wavelets with $\ell = j$ ranging from 1 to 4.

Covariance of the kernel smoothed Shannon spectrum

As the Shannon cross-correlation wavelet is only non-zero when the scales are the same, the kernel smoothed spectrum has a more sparse covariance structure than when the Haar wavelet is used.

The following result holds when the white noise has zero excess kurtosis, such as the Gaussian distribution;

$$\begin{aligned} \text{cov}(\tilde{I}_{\ell,m}, \tilde{I}_{j,n}) &= \sum_{u=m-s}^{m+s} \sum_{v=n-s}^{n+s} K\left(\frac{m-u}{s}\right) K\left(\frac{n-v}{s}\right) \text{cov}(I_{\ell,u}, I_{j,v}) \\ &= \begin{cases} \sum_{u=m-s}^{m+s} \sum_{v=n-s}^{n+s} K\left(\frac{m-u}{s}\right) K\left(\frac{n-v}{s}\right) 2\sigma^4 \Psi_j^2(u-v) & \text{for } \ell = j \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (119)$$

Results in Section 5.3.3 and 5.3.4 are specific to Haar and Shannon wavelets, however other wavelets could be used to test for local white noise. A benefit of the Shannon wavelet is that off-diagonal entries of the covariance matrix under the null hypothesis are zero, a feature that will not hold for wavelets in general. To use different wavelets, such as non-Haar Daubechies wavelets, their cross-correlation wavelet and sum of squared wavelets are needed, which could be found computationally. Other wavelets may provide test power in alternative directions due to the differences in estimated wavelet spectra when using different wavelets, and could be tested using simulations.

5.4 Local white noise test method

This section describes methods of constructing test statistics for our local white noise test, as well as our definition of local white noise.

5.4.1 Definition of local white noise

There are several ways local white noise could be defined. One option is to define it as instantaneous white noise at an observation, and another is to define it locally on a window or segment.

Instantaneous white noise could be characterised using a particular time position of the β spectrum. At a particular $z = t/T \in [0, 1]$, if $\beta_j(z) \equiv c$ for some $c \in \mathbb{R}^+$ and $j = 1, 2, \dots$,

then there is instantaneous white noise at z . However, in practice, the spectrum values are estimated using more than the observation that the spectrum is centred on via the wavelet transform, so the instantaneous definition of white noise may not be particularly intuitive.

Local white noise could also be defined as equal β -spectrum values over a window, or small interval. This still has the issue discussed above, as the values within the interval rely on those outside it. However a test based on this definition is likely to be more stable and have more power as it will be based on more observations. To ensure locality, the coarsest wavelet scales should be discounted from a test.

Here we chose to test the hypothesis that white noise is present locally on an interval, against the alternative hypothesis of a locally stationary wavelet process with greater structure in the interval. Our aim is not to test for Gaussianity; we wish to test for white noise more generally. Our definition of local white noise is formalised in Definition 5.2.

Definition 5.2 (Local white noise). A section X_{t_1}, \dots, X_{t_2} of a stochastic process, $\{X_t\}_{t=1}^T$, with $1 \leq t_1 < t_2 \leq T$ is said to be local white noise on the interval $[t_1, t_2]$ if the following properties hold for all $s, t = t_1, \dots, t_2$:

- (i) $\mathbb{E}(X_s) = 0$
- (ii) $\mathbb{E}(X_s X_t) = \delta_{s,t} \sigma^2$,

where $\sigma^2 < \infty$ and $\delta_{s,t} = \mathbb{I}(s = t)$ is the Kronecker delta function. If $t_1 = 1$, and $t_2 = T$, then the process is global white noise.

All of the previous results in this section hold for local white noise, but require $t_2 - t_1 \rightarrow \infty$ in place of $T \rightarrow \infty$.

5.4.2 Estimating the β -spectrum

As stated in Section 5.2.2, the wavelet periodogram is used to estimate the β -spectrum. For Haar wavelets, the wavelet periodogram can be estimated from a series using `ewspec3` from the `locits` package (Nason (2016a)). For Shannon wavelets, the function `ewspecShannon` from the `AutoSpec` package can be used (Nason et al. (2012)). Both of these functions require the input series to be of power of two length. Although these functions provide smoothing, we use unsmoothed periodograms, $I_{\ell,m}$, and then apply the kernel smoothing of

Equation (101), with bandwidth s . The dependence of the smoothed β -spectrum estimate, $\tilde{I}_{\ell,m}$ on the smoothing method and bandwidth will not be explicit in our notation.

5.4.3 Test statistics

As we test for *local* white noise, the estimated β -spectrum is tested for equality of values over scale at the central point of the interval, which equates to considering $\tilde{I}_{1:J^*,t_k}$ where t_k is equidistant from t_1 and t_2 . The number of scales used, J^* , is such that $J^* < J = \log(T)$, with the effect of this choice discussed in Section 5.5. For notational purposes, we will drop the explicit time labelling in what follows, and define the J^* -length vector $\mathbf{B} = \tilde{I}_{1:J^*,t_k}$. As shown in Proposition 5.1 and Equation (100), under the null hypothesis the rectangular-kernel smoothed β -spectrum estimate at t_k has an asymptotic normal distribution; $\mathbf{B} \sim^A N(\sigma^2 \mathbf{1}, \Sigma_B)$, where $[\Sigma_B]_{i,j} = \text{cov}(\tilde{I}_{i,t_k}, \tilde{I}_{j,t_k})$, given by Equation (102).

Contrast test statistic

For a fixed $t_k \in \{1, \dots, T\}$, the β -spectrum has equal values at each scale under the null hypothesis. Using the equality over scale as motivation, we define a set of test statistics based on contrasts. Let R define a $c \times J^*$ contrast matrix where the row sums are all zero. Noting that \mathbf{B} is a consistent estimator of $\sigma^2 \mathbf{1}$, the Wald statistic is given by

$$U_r = (R\mathbf{B})' \{R(\Sigma_B)R'\}^{-1} (R\mathbf{B}), \quad (120)$$

which tends in distribution to χ_c^2 . To test for white noise of a particular known model, the exact form of Σ_B can be used, but for general testing the covariance matrix will be replaced by $\hat{\Sigma}_B$ containing estimators of the process variance and excess kurtosis.

Different types of contrast are described in the following three sections, setting out potential forms of the contrast matrix, R .

Single contrast: alternate

This test consists of a single contrast, where the R matrix has values that alternate between 1 and -1 for even numbers of scales. When the number of scales is odd, R is first filled with the alternating 1 and -1 values, then the central value is replaced with 0 or -2 to ensure the sum is always zero. For example, when $J^* = 3$ the contrast matrix is $R = (1 \ -2 \ 1)$,

and when $J^* = 5$ the contrast matrix is $R = (1 \ -1 \ 0 \ -1 \ 1)$. This contrast will work best against an alternative hypothesis where consecutive scales of the wavelet periodogram have different values. Certain processes with power in narrow bands would be examples of this, the difference between scales with power and their low-valued neighbours would be captured using an alternate contrast.

Single contrast: opposite

This contrast compares the coarse scales with fine scales: the first half of this R vector is set to 1, and the second half is -1. If J^* is odd, the central value is zero. This contrast will capture spectrum differences for processes with monotonically increasing or decreasing power over scales, such as MA(1) processes.

Complete contrast

The complete contrast contains $J^* - 1$ individual orthogonal contrasts, forming a $J^* - 1 \times J^*$ contrast matrix. For example, the matrix we use for $J^* = 3$ is

$$R = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & -2 \end{pmatrix}, \quad (121)$$

and for $J^* = 5$ we have a 4×5 matrix;

$$R = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 & -1 \\ 1 & 1 & -4 & 1 & 1 \end{pmatrix}. \quad (122)$$

The complete contrast should identify departures from local white noise that consist of different values at consecutive scales (rows 1 and 2 of R in Equation (122)), monotonically changing power over scales (row 3), or different power at the central scale (row 4). The complete contrast contains more comparisons than the single contrasts, so should identify a greater range of processes in the alternative hypothesis set.

Quadratic form test statistic

Rather than using contrasts, the mean of \mathbf{B} can be estimated and subtracted from the spectrum, giving centred spectrum values. This centring can be written as a matrix operation, by considering a $J^* \times J^*$ centring matrix, H which has values $1/J^*$ on the off-diagonal, and $(J^* - 1)/J^*$ on the diagonal. The centred values are calculated as $\tilde{\mathbf{B}} = H\mathbf{B}$, with null distribution $\tilde{\mathbf{B}} \sim^A N(\mathbf{0}, H\Sigma_B H')$, where $\mathbf{0}$ is a J^* -length vector of zeroes.

The test statistic at t_k is defined as the simple quadratic form of the centred values; $U_q = \tilde{\mathbf{B}}'\tilde{\mathbf{B}}$. Using Mathai and Provost (1992), Equation 4.1.2, we have the representation $U_q = \sum_{p=1}^r \lambda_p Z_p^2$, where $\{\lambda_p\}_{p=1, \dots, r}$ are the eigenvalues of $H\Sigma_B H'$, and $\{Z_p\}_{p=1, \dots, r}$ are independent standard normal variables. Hence the test statistic is a sum of scaled χ^2 variables.

As shown in Section 5.3, matrix Σ_B is known up to the process variance and excess kurtosis, so we estimate the covariance matrix using the estimates of these, as in Section 5.4.3. Duchesne and de Micheaux (2010) compared methods of calculating the distribution of a sum of independent χ^2 variables and recommended that exact methods such as the Imhof (1961) algorithm should be used as approximations tend to be poor. To implement this the `imhof` function of the `CompQuadForm` package of Duchesne and de Micheaux (2010) is used.

5.5 Simulations

In this section we demonstrate properties of the LWN test in a series of settings. To implement the test the LWN package is used, written predominantly by the author, with additions by Guy Nason. The manual for a working version of this package is included in Appendix C, and a fully tested version of this package will be released via CRAN in due course. Throughout these simulations, the rectangular kernel is used for smoothing.

5.5.1 Multiple test correction

When the LWN test is calculated on many segments of a series, multiple test correction is applied to the p -values to guard against spurious rejections of the null hypothesis caused by repeated testing. For this the Holm (1979) method is used, which follows the following algorithm. If a series is tested locally at K positions, this results in ordered p -values

$p_{(1)} < p_{(2)} < \dots < p_{(K)}$ associated with the times $t_{(1)}, t_{(2)}, \dots, t_{(K)}$. For level α , if $p_{(1)} < \alpha(K)^{-1}$, reject the null hypothesis at $t_{(1)}$, otherwise none of the local hypotheses can be rejected. If the test with lowest p -value rejects the null, the testing of p -values continues sequentially until $p_{(k)} > \alpha(M + 1 - k)^{-1}$ for some k . Then the null hypothesis at positions $t_{(1)}, \dots, t_{(k-1)}$ are rejected, and the remaining tests are not rejected at level α .

5.5.2 Variance and kurtosis estimation

Crucial to being able to test for non-Gaussian white noise is estimation of the variance and excess kurtosis of the original series. Here a wavelet method is used to estimate the variance at time position m , taking the mean of the estimated β -spectrum over J^* scales and $(2s + 1)$ time points; $\hat{\sigma}_m^2 = \{J^*(2s + 1)\}^{-1} \sum_{j=1}^{J^*} \sum_{k=m-s}^{m+s} I_{j,k}$.

The excess kurtosis is present in the covariance formula multiplied by σ^4 ; $\{\mathbb{E}(X_t^4) - 3\sigma^4\}$, and to avoid estimating the variance twice, we estimate the fourth moment in the excess kurtosis using the estimator $\hat{\mu}_{4,m} = (2s+1)^{-1} \sum_{k=m-s}^{m+s} X_k^4$. Then when estimating the covariance matrix at position m we replace $\{\mathbb{E}(X_t^4) - 3\sigma^4\}$ with the estimate $\{\hat{\mu}_{4,m} - 3(\hat{\sigma}_m^2)^2\}$.

In the following simulation, we test the performance of different estimators of σ^2 for a series of length $T = 64$. The series each have variance 1, the Gaussian distribution is standard $\mathcal{N}(0, 1)$, the Laplace distribution has location parameter $\mu = 0$ and scale parameter $b = 1/\sqrt{2}$, and the Uniform distribution is $U(-\sqrt{3}, \sqrt{3})$. Excess kurtosis for the Gaussian distribution is zero, for Laplace is 3, and for Uniform is -1.2 . Table 11 includes the standard time-domain estimator of variance, `var`, which includes all of the 64 observations. The LWN estimators of the variance, which calculate variance using the wavelet spectrum, are tested over different maximum scales, J^* , and smoothing intervals, s . For low J^* and s , the estimate is in effect being carried out using a smaller sample than the comparison estimator `var`. Whilst most of the estimators appear unbiased, using the Shannon wavelet spectrum with $J^* = 5$ is not an unbiased estimator. This is evident throughout the choices of s , and the different innovation distributions. The biased nature of this estimator may be due to the Shannon wavelet not decaying quickly enough within the interval.

Excluding the $J^* = 5$ case, the Haar and Shannon estimators have similar bias and MSE. The single value case, $s = 0$, where the estimate of variance is the mean of the spectrum at

Estimator of variance	Gaussian		Laplace		Uniform	
	Bias	MSE	Bias	MSE	Bias	MSE
var	0.59	3.27	-0.78	7.78	-0.10	1.30
Haar, $J = 3$ $ss = 0$	0.23	70.43	-6.44	107.43	0.25	43.86
Haar, $J = 4$ $ss = 0$	-0.76	55.57	-5.89	78.55	-1.10	33.41
Haar, $J = 5$ $ss = 0$	-0.52	41.22	-4.60	58.94	-1.59	26.94
Haar, $J = 3$ $ss = 15$	0.11	7.54	-1.49	16.44	-0.25	3.51
Haar, $J = 4$ $ss = 15$	0.10	8.50	-1.95	15.81	-1.01	4.07
Haar, $J = 5$ $ss = 15$	0.19	9.21	-0.92	17.03	-0.80	5.83
Haar, $J = 3$ $ss = 31$	0.24	4.01	-1.32	8.28	-0.58	1.96
Haar, $J = 4$ $ss = 31$	0.15	4.95	-1.56	9.16	-1.11	2.48
Haar, $J = 5$ $ss = 31$	-0.41	6.20	-1.44	11.30	-1.61	3.86
Shannon, $J = 3$ $ss = 0$	-1.64	67.77	-3.90	88.47	-1.66	53.13
Shannon, $J = 4$ $ss = 0$	-0.74	52.74	-2.65	67.99	-2.40	37.94
Shannon, $J = 5$ $ss = 0$	-10.85	36.30	-12.08	46.76	-11.98	25.95
Shannon, $J = 3$ $ss = 15$	0.07	9.20	-2.61	15.75	-0.69	5.38
Shannon, $J = 4$ $ss = 15$	0.91	10.76	-2.47	15.61	-1.03	6.15
Shannon, $J = 5$ $ss = 15$	-9.52	8.80	-11.93	13.29	-10.88	5.73
Shannon, $J = 3$ $ss = 31$	0.35	5.11	-1.53	8.73	0.01	3.08
Shannon, $J = 4$ $ss = 31$	0.50	6.52	-1.68	9.81	-0.80	3.67
Shannon, $J = 5$ $ss = 31$	-9.85	6.08	-11.30	9.31	-10.69	4.22

Table 11: Bias and MSE ($\times 100$) of different estimators of σ^2 , calculated using 1000 simulations of length $T = 64$ of each distribution.

the central time point, is unbiased but has high MSE. To stabilise the estimator smoothing over time is recommended. The MSE is lowest with highest s , where the average is taken over more time values. When $s \neq 0$, the choice of J^* makes little difference in most cases. The best LWN estimators of innovation variance that include the entire comparison interval perform similarly to the standard time domain estimator of variance.

To compare these estimators in more detail, four empirical densities are plotted in Figure 23. All are approximately centred on the true value 1, with the **var** estimator having smallest spread around the mean. This plot shows the Haar LWN estimator with $J^* = 3$ and $s = 31$ has smaller spread than the Shannon estimator with the same parameters, and both are better than the Haar estimator when dropping the smoothing interval to $s = 15$.

The kurtosis is estimated using the sample fourth moment, $\hat{\mu}_{4,m} = (2s + 1)^{-1} \sum_{k=m-s}^{m+s} X_k^4$. The bias and MSE of this estimator for the same samples as in Table 11 are found in Table 12. The true $\mathbb{E}(X_t^4)$ for the modes are 3 for Gaussian, 6 for Laplace, and 1.8 for Uniform. As Table 12 shows, the performance of the fourth moment estimator varies across distributions. For the Laplace distribution, the estimator is incredibly variable, which is of concern for the result of our test. The range of this estimator over 1000 simulations

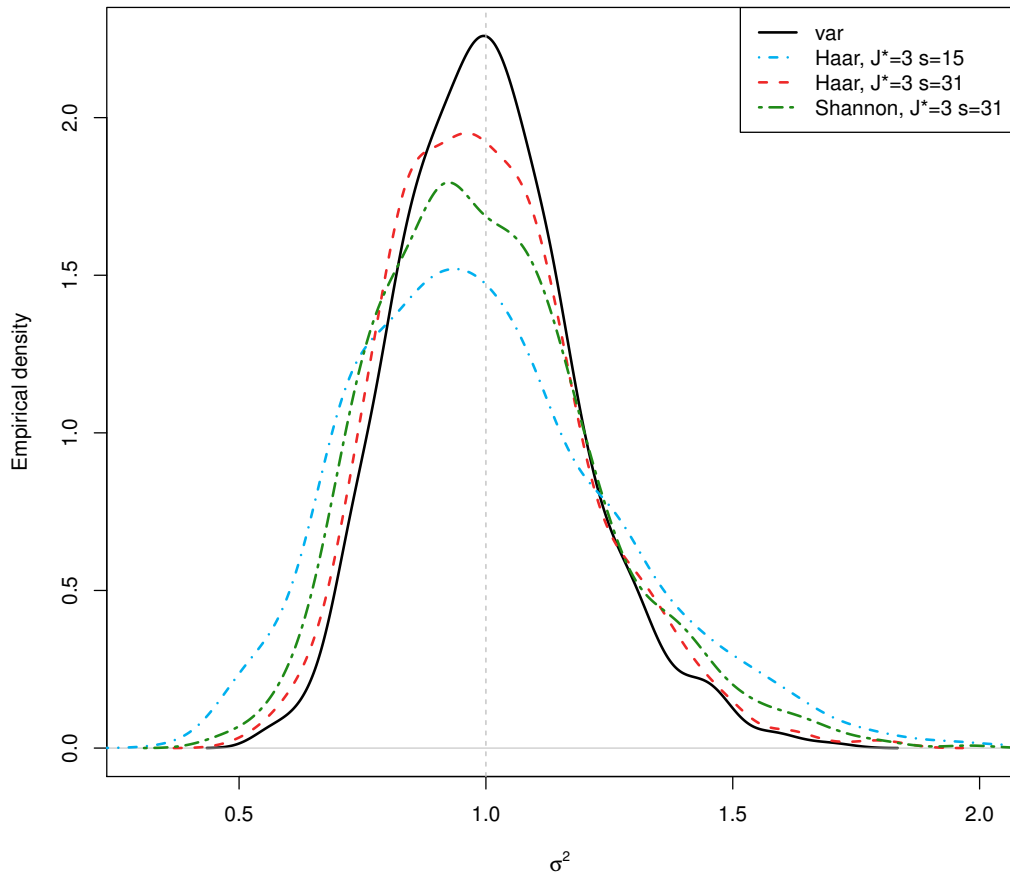


Figure 23: Empirical density of innovation variance estimates using the standard estimator, and three LWN estimators. This plot is produced using the function `density` with default smoothing arguments.

of Laplace white noise with $s = 31$ was 0.36-83.08. As expected, the estimator for each distribution performs better as s increases, and this leads to the recommendation to not use $s < 15$ in the local white noise tests as the kurtosis estimation has high MSE. Indeed, if the segment is stationary, s should be chosen so that smoothing occurs over the entire segment, $s = 2^{-1}(t_2 - t_1) - 1$.

5.5.3 Size and power properties as a global test

The LWN test is first tested for different observation lengths, using the test statistics described in Section 5.4.3. Throughout this section, the complete, alternate, and opposite contrasts will be denoted ' C_{com} ', ' C_{alt} ', ' C_{opp} ', respectively, and the quadratic form statistic will be labelled 'Q'. If the test run assumes Gaussianity of the white noise it does not include the excess kurtosis term, and this is denoted 'asG'.

s	Gaussian		Laplace		Uniform	
	Bias*	MSE	Bias*	MSE	Bias*	MSE
0	-20.85	62.02	-28.97	776.62	-1.84	5.64
15	9.40	3.41	-22.09	85.86	-1.00	0.18
31	5.00	1.67	-16.97	36.91	-0.54	0.09

Table 12: Performance of the fourth moment estimator $\hat{\mu}_{4,32}$ over 1000 simulations of white noise of length $T = 64$. All distributions have parameters set to achieve variance 1. * the bias values have been multiplied by 100 for readability.

Tables 13-17 contain the results of simulations of MA(1) processes with standard normal innovations. For the first three tables, J^* is chosen to be $\log_2(t_2 - t_1) - 2$, so the coarsest two scales are not included in the test. As the wavelet spectrum of MA(1) processes has monotonically changing values of the spectrum over scales it is expected that the tests involving opposite rather than alternate contrasts will perform better. Whilst there are differences at alternate scales, the differences best manifest when comparing the coarse scales to the fine scales.

In Table 13, where $T = 32$, the test using Shannon wavelets and alternate contrasts performs worst for both positive and negative MA parameters, as expected. In general, the other LWN tests perform well in comparison to the Bartlett and Ljung-Box tests for negative MA parameters, but have lower power for MA parameters 0.7 and 0.9. The size of all of the LWN test variants is in the range 0.03-0.05, showing better calibration than the Bartlett test and Ljung-Box test with 20 lags.

As the observation length increases, the power improves for all of the tests. For $T = 64$ in Table 14 all but the C_{alt} contrast LWN tests have power 1 for MA parameter -0.9, higher than the comparison tests. However the best performing LWN test for MA parameter 0.9 achieves 0.80 power, whereas the the Bartlett test has power of 0.99. The Haar quadratic form test has much higher power than the Shannon version for positive MA parameters. The LWN tests with highest overall power are Shannon C_{com} and Haar Q.

Even for observation length $T = 128$ in Table 15, the Shannon C_{alt} test performs poorly when considering MA processes as the alternative hypothesis. The test that does not include the excess kurtosis estimation, C_{com} asG, has higher power for MA parameter -0.3 than C_{com} , but lower power for positive parameters. For this observation length, the C_{com} test performs best in terms of power out of the LWN tests.

To investigate the role of J^* in the LWN test properties, Tables 16 and 17 reduce the

Wavelet	Test	MA parameter								
		-0.9	-0.7	-0.5	-0.3	0	0.3	0.5	0.7	0.9
Shannon	C_{com} asG	0.74	0.65	0.46	0.26	0.04	0.10	0.19	0.24	0.29
Shannon	C_{com}	0.64	0.56	0.36	0.18	0.04	0.16	0.28	0.36	0.43
Shannon	C_{alt}	0.02	0.01	0.01	0.02	0.04	0.10	0.13	0.14	0.15
Shannon	C_{opp}	0.76	0.66	0.44	0.20	0.04	0.14	0.24	0.32	0.34
Shannon	Q	0.56	0.46	0.25	0.10	0.03	0.13	0.21	0.27	0.28
Haar	Q	0.77	0.70	0.50	0.28	0.05	0.16	0.29	0.44	0.51
	Bartlett	0.70	0.64	0.44	0.23	0.02	0.10	0.27	0.50	0.59
	L-B 5	0.60	0.53	0.38	0.22	0.07	0.15	0.31	0.46	0.57
	L-B 20	0.52	0.47	0.35	0.20	0.10	0.17	0.32	0.46	0.51

Table 13: Simulated power and size using 1000 simulations of series with $T = 32$, $s = 15$, $J^* = 3$.

Wavelet	Test	MA parameter								
		-0.9	-0.7	-0.5	-0.3	0	0.3	0.5	0.7	0.9
Shannon	C_{com} asG	1	0.99	0.91	0.51	0.07	0.20	0.43	0.58	0.66
Shannon	C_{com}	1	0.97	0.83	0.38	0.06	0.28	0.57	0.74	0.80
Shannon	C_{alt}	0.44	0.34	0.16	0.08	0.04	0.12	0.17	0.21	0.23
Shannon	C_{opp}	1	0.98	0.79	0.32	0.05	0.18	0.28	0.34	0.41
Shannon	Q	0.99	0.92	0.61	0.16	0.04	0.16	0.26	0.35	0.36
Haar	Q	0.99	0.98	0.90	0.46	0.06	0.25	0.44	0.59	0.66
	Bartlett	0.99	0.98	0.87	0.49	0.04	0.34	0.8	0.94	0.99
	L-B 5	0.97	0.94	0.75	0.36	0.05	0.31	0.69	0.91	0.97
	L-B 20	0.80	0.74	0.52	0.29	0.08	0.24	0.51	0.70	0.76

Table 14: Simulated power and size using 1000 simulations of series with $T = 64$, $s = 31$, $J^* = 4$.

number of scales used to 4 and 3, respectively, while keeping $T = 128$. Using fewer scales in general gives similar, or greater powers across the tests. The Haar Q test features improved power for positive parameters when using fewer scales, with Haar Q, $J^* = 3$ having highest power out of all the tests at $T = 128$ in Tables 15-17. At $T = 128$ the C_{com} test dominates the C_{alt} and C_{opp} tests for all J^* . For the quadratic form test statistics, the Shannon test is dominated by the test using Haar wavelets.

As the Shannon C_{com} and Haar Q tests have performed best in the MA simulations, these tests are used for further comparison, however it is noted that for other alternative models different tests may perform best. The results of Tables 15-17 lead us to suggest using a value of $J^* = \text{floor}\{\log_2(t_2 - t_1)\} - 2$ for the Shannon C_{com} version of the test, and $J^* = \text{floor}\{\log_2(t_2 - t_1)\} - 4$ for the Haar Q version, noting that for particularly short local segments it may not be practical to use such small values of J^* .

Table 18 compares the two LWN tests to results in Nason and Savchev (2014), for their

Wavelet	Test	MA parameter								
		-0.9	-0.7	-0.5	-0.3	0	0.3	0.5	0.7	0.9
Shannon	C_{com} asG	1	1	1	0.87	0.08	0.42	0.81	0.95	0.96
Shannon	C_{com}	1	1	1	0.79	0.06	0.53	0.88	0.98	0.99
Shannon	C_{alt}	0.56	0.39	0.17	0.03	0.05	0.17	0.29	0.33	0.33
Shannon	C_{opp}	1	1	0.98	0.62	0.07	0.11	0.18	0.24	0.27
Shannon	Q	1	1	0.94	0.37	0.04	0.15	0.29	0.34	0.38
Haar	Q	1	1	0.99	0.67	0.06	0.29	0.50	0.67	0.75
	Bartlett	1	1	1	0.83	0.04	0.77	0.99	1	1
	L-B 5	1	1	0.99	0.71	0.05	0.67	0.98	1	1
	L-B 20	0.99	0.99	0.87	0.46	0.07	0.44	0.85	0.98	1

Table 15: Simulated power and size using 1000 simulations of series with $T = 128$, $s = 63$, $J^* = 5$.

Wavelet	Test	MA parameter								
		-0.9	-0.7	-0.5	-0.3	0	0.3	0.5	0.7	0.9
Shannon	C_{com} asG	1	1	0.99	0.80	0.06	0.48	0.86	0.97	0.98
Shannon	C_{com}	1	1	0.99	0.72	0.04	0.57	0.90	0.98	0.99
Shannon	C_{alt}	0.81	0.69	0.44	0.14	0.05	0.17	0.25	0.26	0.28
Shannon	C_{opp}	1	1	0.98	0.55	0.05	0.29	0.49	0.57	0.62
Shannon	Q	1	1	0.96	0.42	0.05	0.26	0.5	0.64	0.71
Haar	Q	1	1	1	0.80	0.06	0.48	0.84	0.97	0.99

Table 16: Simulated power and size using 1000 simulations of series with $T = 128$, $s = 63$, $J^* = 4$.

Wavelet	Test	MA parameter								
		-0.9	-0.7	-0.5	-0.3	0	0.3	0.5	0.7	0.9
Shannon	C_{com} asG	1	1	0.97	0.67	0.04	0.52	0.88	0.98	0.99
Shannon	C_{com}	1	1	0.97	0.63	0.05	0.57	0.90	0.98	0.99
Shannon	C_{alt}	0.1	0.08	0.08	0.05	0.04	0.11	0.19	0.21	0.22
Shannon	C_{opp}	1	1	0.98	0.66	0.05	0.46	0.74	0.86	0.90
Shannon	Q	1	1	0.95	0.54	0.04	0.40	0.79	0.93	0.97
Haar	Q	1	1	1	0.81	0.06	0.71	0.97	1	1

Table 17: Simulated power and size using 1000 simulations of series with $T = 128$, $s = 63$, $J^* = 3$.

Wavelet	Test	Model			
		MA(2) $T = 64$	ARMA(1,2)		
			$T = 16$	$T = 32$	$T = 64$
Shannon	$C_{com}, J^* = 4$	0.26	-	-	1
Shannon	$C_{com}, J^* = 3$	0.22	0.60	0.97	1
Shannon	$C_{com}, J^* = 2$	-	0.55	0.96	-
Haar	$Q, J^* = 4$	0.40	-	-	1
Haar	$Q, J^* = 3$	0.58	0.85	0.99	1
Haar	$Q, J^* = 2$	-	0.85	0.99	-
	HWWN	0.27	0.43	0.93	1
	L-B 1	0.17	0.93	1	1
	L-B 20	0.35	-	0.79	1
	Bartlett	0.38	0.81	1	1

Table 18: Simulated power of local white noise tests against HWWN, Ljung-Box and Bartlett results taken from Table III in Nason and Savchev (2014).

Haar wavelet white noise test (HWWN), as well as the Ljung-Box and Bartlett tests. The MA(2) model has parameters $\beta_1 = 0$ and $\beta_2 = 0.5$, and the ARMA(1,2) model has parameters $\alpha_1 = -0.4$, $\beta_1 = -0.8$ and $\beta_2 = 0.4$. As shown in previous results, the number of scales used in our LWN test for these models is important, with the Shannon C_{com} test having lower power with smaller J^* in Table 18, and the opposite for the Haar Q test. The LWN models perform well compared to the other methods, with a particular improvement when using Haar Q with $J^* = 3$ for the MA(2) model.

Table 19 contains equivalent results to Table 13, but with Uniform innovations with variance one rather than standard normal innovations. Both of the LWN tests are well calibrated for this type of white noise, and the resulting power values are similar to the standard normal results for all of the tests except the Ljung-Box 20 test, which has lower power for negative MA parameters than for normal innovations. Table 20 contains results using Laplace innovations with variance one. The power drops a little for the LWN tests for this distribution, with the Haar Q test having higher power than the Shannon C_{com} test. In this case, the Haar Q test also outperforms both of the Ljung-Box tests. For negative MA parameters, the Haar Q test is most powerful, and for positive parameters the Bartlett test has greatest power, so without further knowledge about the alternative hypothesis it would be difficult to choose between these tests.

Wavelet	Test	MA parameter								
		-0.9	-0.7	-0.5	-0.3	0	0.3	0.5	0.7	0.9
Shannon	C_{com}	0.64	0.61	0.43	0.24	0.07	0.19	0.31	0.42	0.47
Haar	Q	0.75	0.69	0.51	0.28	0.05	0.16	0.28	0.43	0.47
	Bartlett	0.70	0.63	0.42	0.23	0.03	0.12	0.27	0.46	0.55
	L-B 5	0.47	0.43	0.29	0.14	0.04	0.12	0.22	0.37	0.43
	L-B 20	0.18	0.16	0.09	0.04	0.01	0.12	0.27	0.46	0.55

Table 19: Simulated power and size using 1000 simulations of series with $T = 32$, $s = 15$, $J^* = 3$ and Uniform innovations.

Wavelet	Test	MA parameter								
		-0.9	-0.7	-0.5	-0.3	0	0.3	0.5	0.7	0.9
Shannon	C_{com}	0.51	0.46	0.25	0.08	0.01	0.10	0.21	0.32	0.34
Haar	Q	0.74	0.69	0.52	0.24	0.04	0.14	0.27	0.42	0.46
	Bartlett	0.70	0.61	0.44	0.19	0.02	0.10	0.25	0.44	0.57
	L-B 5	0.43	0.41	0.28	0.12	0.02	0.08	0.19	0.33	0.40
	L-B 20	0.14	0.13	0.06	0.02	0.00	0.03	0.06	0.12	0.12

Table 20: Simulated power and size using 1000 simulations of series with $T = 32$, $s = 15$, $J^* = 3$ and Laplace innovations.

5.5.4 Size and power properties as a local test

The previous section shows that the local white noise test works well compared to frequently used tests when used as a global test on small sample sizes. In this section, the advantage of using a local test is shown, on examples that would be described as spurious white noise using the terminology of Goerg (2012). These examples are time-varying MA(1) processes, with parameter that averages to zero over the time interval. The innovations are standard normal random variables.

The MA parameters of the three tested processes are plotted in Figure 24, along with the test sections which are centred at values $t = 32, 128, 224$. The smoothing parameter is $s = 31$ and $J^* = 4$ scales are used throughout the following simulations. As shown in the previous simulations, the LWN test has better power for negative parameters, so it is expected that the sections with negative TVMA parameter will be rejected more often than those with positive parameters. The local white noise tests are compared to using the currently available tests, Bartlett and Ljung-Box applied globally. The LWN test is described as rejecting the hypothesis of global white noise if any of the section tests are inconsistent with the local white noise hypothesis. Process A has local white noise in section 2, and has accordingly low power (size) here. As expected, the power is higher for section 3, where the MA parameter is negative, compared to section 1. The LWN tests

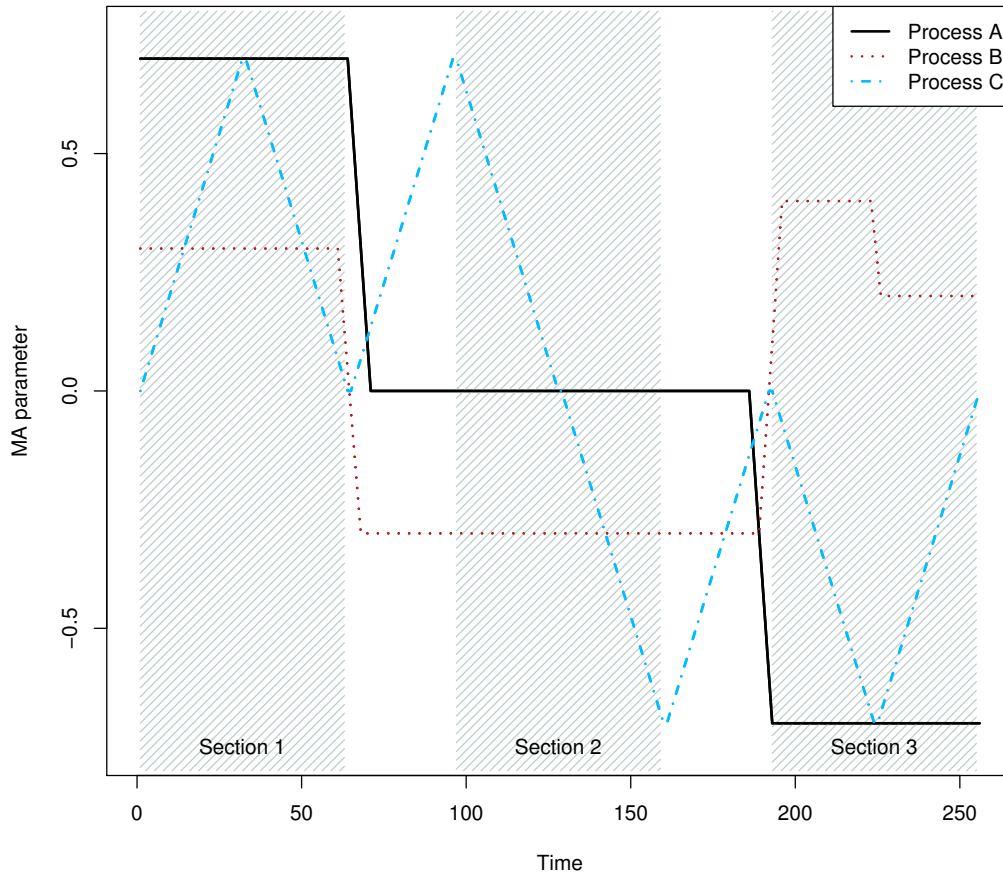


Figure 24: MA parameters of the TVMA(1) processes used in simulations.

Process	Wavelet	Test	Power			
			Global	Section 1	Section 2	Section 3
A	Shannon	C_{com}	0.86	0.48	0.02	0.78
	Haar	Q	0.97	0.37	0.02	0.95
		Bartlett	0.11	-	-	-
		L-B 5	0.13	-	-	-
		L-B 20	0.19	-	-	-
B	Shannon	C_{com}	0.34	0.10	0.20	0.11
	Haar	Q	0.41	0.07	0.30	0.11
		Bartlett	0.06	-	-	-
		L-B 5	0.08	-	-	-
		L-B 20	0.10	-	-	-
C	Shannon	C_{com}	0.36	0.15	0.02	0.26
	Haar	Q	0.49	0.15	0.03	0.42
		Bartlett	0.08	-	-	-
		L-B 5	0.12	-	-	-
		L-B 20	0.16	-	-	-

Table 21: Power of tests applied to a TVMA(1) process over 1000 simulations using standard normal innovations. The LWN global test refers to a rejection at any of the three sections tested. LWN tests have $s = 31$ and $J^* = 4$.

have much higher power (0.86-0.97) than the global white noise tests which have power 0.11-0.19 for Process A.

The power of the global white noise tests is also low for Process B. Here there is no local white noise, but as the MA parameters are small the departures from white noise are difficult to detect. Again, the section with negative MA parameter is detected most by the LWN tests, and section 3 with changing positive parameter is rejected slightly more than the stationary section 1. Although the LWN tests do not achieve high power, they are a marked improvement on the global tests.

For Process C there is no local white noise. Section 1 has far lower power than the equivalent section for Process A as for Process C the average of the MA parameter over this section is much lower. Similarly for section 3. The central section is spurious local white noise as the parameter averages to zero, and the local white noise test has very low power here.

These simulations show the benefit of testing locally for white noise, as many more of the realisations reject the null hypothesis of white noise when using our local test.

5.6 Examples

In this section, the local white noise test is used on the residuals from the GNAR models fitted in Section 4.6. In both examples, the rectangular kernel is used to smooth the β -spectrum estimate.

5.6.1 GDP example

In Section 4.6.1, a GNAR(2,[2,0]) model was fitted to the GDP data up to $T = 51$, and shown to outperform AR and VAR models for prediction accuracy. In this section, the LWN test is used to check the residuals of the GNAR model.

The residuals corresponding to times 4-35 are selected at each node, and nodes with missing values are not tested. The Haar Q LWN test with $J^* = 3$ and $s = 15$ is applied to the residuals at the 24 nodes where these values are present. The results are shown in Table 22. Of all of these tests, only the test of the residuals at the Belgium node rejects the null hypothesis at the nominal level 5%. The tests were conducted separately, with no multiple error correction, so one rejection would be expected when testing 24 times with

Node	p -value	Test Statistic	$\hat{\mu}_4 - 3(\hat{\sigma}^2)^2$	$\hat{\sigma}^2$
AUS	0.698	0.051	-0.856	0.973
AUT	0.471	0.044	-0.110	0.614
BEL	0.020	0.383	0.647	0.741
CAN	0.796	0.035	-0.411	1.011
DNK	0.731	0.016	0.088	0.581
FIN	0.200	0.295	-0.847	1.070
FRA	0.915	0.010	-0.318	0.878
DEU	0.501	0.046	-0.327	0.657
GRC	0.835	0.018	3.268	0.786
ISL	0.241	0.314	-0.920	1.179
IRL	0.790	0.022	-0.678	0.798
ITA	0.234	0.094	0.571	0.631
JPN	0.465	0.080	1.063	0.814
KOR	0.735	0.021	0.825	0.663
LUX	0.235	0.102	0.723	0.656
MEX	0.984	0.003	0.290	1.090
NLD	0.954	0.002	-0.241	0.478
NOR	0.954	0.009	-1.682	1.167
PRT	0.660	0.103	-1.085	1.279
ESP	0.382	0.160	-0.526	1.033
SWE	0.231	0.099	-0.107	0.650
CHE	0.212	0.327	1.471	1.138
GBR	0.325	0.241	0.009	1.167
USA	0.479	0.145	-1.157	1.132

Table 22: Results of Haar Q LWN test with $J^* = 3$ and $s = 15$ applied to residuals of the GNAR model in Section 4.6.1 at $t = 4 - 35$.

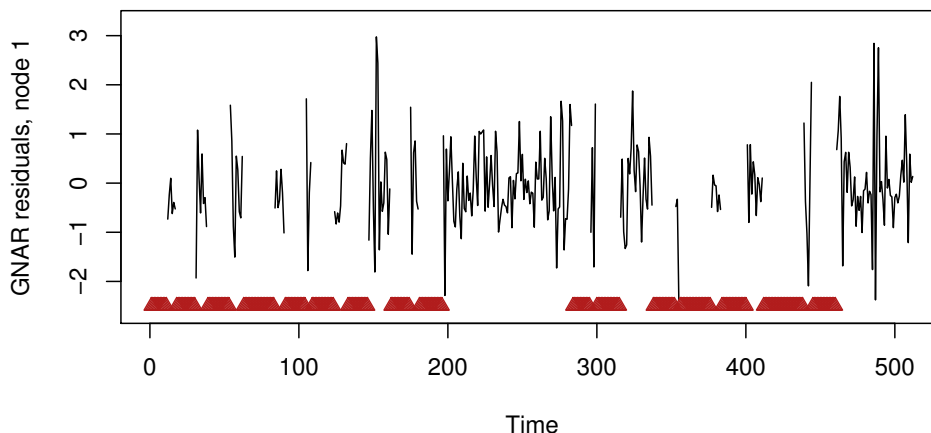


Figure 25: Residuals after GNAR model fit. Red triangles indicate the location of missing observations.

nominal value 5%, therefore this rejection is not investigated further. Indeed, if Holm's multiple test correction is used, none of the residual series are rejected as white noise. Although the GNAR model was chosen to minimise the prediction error (rather than checking residuals), these tests show that the residuals of this model are also consistent with white noise. The estimated excess kurtosis values are not equal across the nodes, and the variance estimates also show some variability. This emphasises the need for the GNAR model of Chapter 4 to allow for residuals that are not identically distributed across nodes.

5.6.2 Bristol Traffic

In this section, the residuals from the GNAR model fit of the Bristol Traffic data from Section 4.6.2 are tested for local white noise. As in Section 3.4.2, the residuals at the first node are used here.

Figure 25 shows the GNAR residuals, with many gaps in the data. As the LWN test requires complete observations of power of two length, the residuals at $t = 197 - 260$ are selected for testing. By testing this set of residuals, the GNAR fit in this section can be evaluated. Both the Barlett test, and the Ljung-Box test with maximum lag 5 performed on the interval of length 64 return p -values over 0.9, so indicate residuals that are consistent with white noise. As the behaviour of the residuals on the segment appears

Test Position	p -value (adjusted)	p -value	Test Statistic	$\hat{\mu}_4 - 3(\hat{\sigma}^2)^2$	$\hat{\sigma}^2$
212	1	0.515	1.329	0.566	0.473
244	1	0.681	0.768	0.008	0.272

Table 23: Local white noise test results applied to GNAR residuals at times 197-260. The LWN test used was Shannon C_{com} , with running mean bandwidth $s = 15$ and $J^* = 3$.

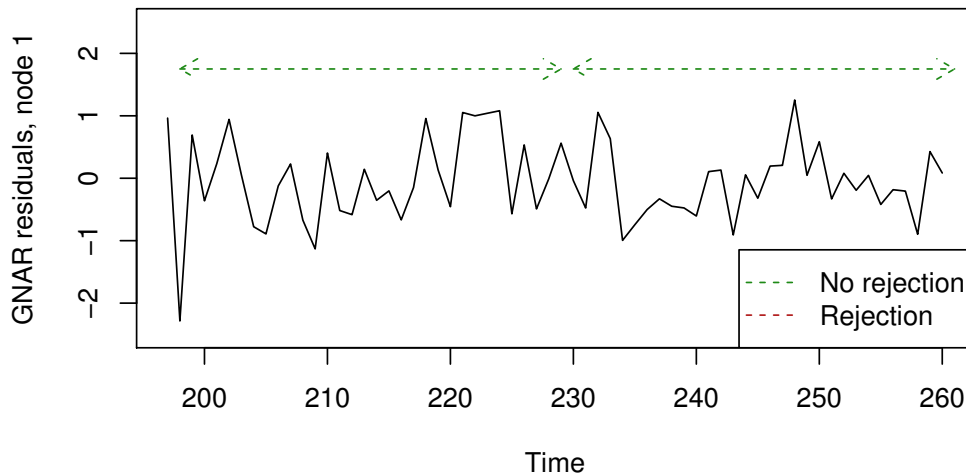


Figure 26: Residuals after GNAR fit with results of two local white noise tests. Arrows indicate the tested segments using Shannon C_{com} , with running mean bandwidth $s = 15$ and $J^* = 3$.

to change at the later values, local white noise tests are performed at two time positions. For these tests, Shannon wavelets with the complete contrast test statistic were used on $J^* = 3$ scales, and running mean smoothing with $s = 15$ was used on the β -spectrum estimate. Table 23 shows that both segments are not rejected by the local white noise tests. This is shown graphically in Figure 26. This indicates that the GNAR model fitted in Section 4.6.2 fits the tested segment, but the resulting noise may not be identically distributed as there are differences in the estimated excess kurtosis of the two segments.

The simulation results of Section 5.5.3 compared the local white noise test to global white noise tests, by effectively treating the local test as a global test. In this example, the local white noise test was applied to data that contains local white noise, rather than global white noise, as it is not identically distributed over the entire interval.

5.7 Conclusion

In this chapter a test for local white noise was presented. The test is wavelet-based and new results about wavelet quantities were developed in Section 5.3, including derivation of the Shannon autocorrelation wavelet. This test has been shown to be effective and robust to use of different distributions of noise through simulations.

In Section 5.4.3 different test statistics were presented that could be used in our local white noise test. Of these, the Shannon wavelet with complete contrast statistic, and the Haar wavelet with quadratic form statistics were the best performing tests in the simulations of Section 5.5. If additional information about the model of a local series segment under the alternative hypothesis were known, then a particular test statistic for the situation could be chosen. The test statistics of Section 5.4.3 are by no means exhaustive, and other adaptations to our local white noise test could be made. One potential adaptation would be to use wavelet packets (Definition 2.27), which may have additional power compared to the method described in this chapter.

Our test also requires the selection of parameters J^* and s , the number of scales to use in the spectrum, and smoothing bandwidth, respectively. Again, these parameters can be better chosen with additional information about the data, with s recommended to be as great as possible while only smoothing over a stationary segment of the series. As a rule of thumb, from the simulations in this chapter and other results not shown, when using the contrast test statistic $J^* = J - 2$ works well, where $J = \text{floor} \{ \log_2(t_2 - t_1) \}$, but for the quadratic form use of fewer scales is preferable, such as $J^* = J - 4$.

Additional theoretical work using the estimated β -spectrum would be beneficial, to show normality of the general kernel smoothed spectrum with an analogous result to Proposition 5.1. In addition, the sum of squared Shannon wavelets in Equation (118) may be available as a closed form expression.

Although many white noise tests are currently available, few, if any, are designed to test series locally. Applying this test following the test of aliasing and local white noise by Eckley and Nason (2018) would allow for the two alternative hypotheses to be separated. Our test is designed for local application, but our current software implementation has limitations for practical use, such as needing power of two inputs and requiring complete data. Inputs of any length could be tested using the developed local white noise framework

with adaptations made to the current software, such as padding inputs to achieve a power of two length. As in Chapter 3, a second-generation wavelet method could be used to produce a local white noise test robust to missing observations, which may be suitable for use on the wider class of irregularly-sampled time series.

6 Discussion

This section summarises the main ideas and achievements of the new methodology described within this thesis, and discusses areas of further development of the work.

6.1 Stationarity testing with missing observations

Chapter 3 presented a new test for stationarity, designed for use on data with missing observations. Our test uses the second-generation wavelet transform to deal with the data gaps, and is an application of the spectrum estimation technique for irregular data of Knight et al. (2012).

Key contributions of Chapter 3

- Novel test of stationarity designed for use when data is incomplete.
- Performs well in terms of power and size against leading tests on complete data.
- Performs well against most of the simulated stationary and non-stationary series in Section 3.3 with up to 20% data loss.

Examples of air quality (PM10) observations and Bristol Traffic car speed data in Section 3.4 demonstrated the use of our test on real-world datasets with missingness. Due to the method of testing, further information was available about the non-stationary behaviour of the air quality dataset, such as changes over time featuring in all of the artificial scales.

Our stationarity test deals with a particular type of irregularly-spaced data, where observations are missing at random from an evenly spaced series. Extending this test for use with irregularly-spaced time series is a future step for this work, including the development of a method for generating bootstrap samples of irregularly-spaced processes. In addition, further simulations and testing while changing the number of trajectories and bootstrap samples would enable a reasoned choice of these test parameters.

6.2 Generalised Network Autoregressive Processes

Chapter 4 describes a generalised model for network time series, GNAR, which allows for network change over time.

Key contributions of Chapter 4

- Unique model for network time series, allowing for covariate and neighbour set inclusion.
- GNAR package provides a suite of tools for fitting, predicting, and simulating GNAR series.
- GNAR model can be used in settings where VAR fails, such as missing data, or data with a small number of time observations compared to the number of nodes. Can also be used for changing networks over time.
- Improved prediction performance compared to AR and VAR models in simulation and GDP example, with fewer parameters needed for prediction.

Simulations of GNAR models in Section 4.5 demonstrated the theoretical consistency results of the parameters and BIC criterion. In the GDP example of Section 4.6.1 our model is used to construct a network that works well for multivariate time series prediction. Compared to the standard VAR and AR models, the GNAR model prediction using the constructed network is much closer to the true observations of GDP.

Further work in the area of generalised network autoregressive processes includes development of the GNAR package to enable fitting and simulating of the full range of GNAR processes, including those with non-identically distributed innovations at each node.

6.3 Local White Noise Testing

The definition of local white noise and a corresponding local white noise test were developed in Chapter 5. This test utilised the wavelet transform and compared values of the smoothed raw wavelet periodogram at different scales, as these values should all be the same under the null hypothesis of local white noise. Different tests were presented to compare these values, including contrasts and quadratic form statistics, and these test statistics were compared to their asymptotic distribution under the null hypothesis.

Key contributions of Chapter 5

- New derivation of Shannon autocorrelation wavelet, and sum of squared Haar wavelets.
- Novel test for local white noise.
- LWN package provides a suite of tools to perform the local white noise test.
- Test performs comparatively with global white noise tests.
- Using the definition of local white noise, poor model fit can be detected in sections of time series residuals where global tests fail.

This test was used in Section 5.6 to demonstrate the lack of structure in the residuals of the GNAR models fitted in Chapter 4.

Further work in this area includes testing a wider range of test statistics, including wavelet packet quantities which may increase the power of the test against some alternatives. Further theoretical developments of the quantities used in the autocovariance equations would also be beneficial, particularly for Shannon wavelets. In addition, the normality of the kernel smoothed wavelet periodogram is conjectured but not proven. Whilst a preliminary software package for using our test is available, the functionality could be expanded to allow for data of any length (not just a power of two). Another expansion of this method would include providing a test for use with missing observations, which could use second-generation wavelets in a similar way to Chapter 3.

A LiftToS example documentation

liftToS *Function to return the test statistic of the lifting test of stationarity.*

Description

This function calculates the test statistic of the lifting test of stationarity. This function performs the NLT algorithm on the data using `fwtpperm` from the `nlt` package, generates an estimate of the evolutionary wavelet spectrum from the resulting details, and returns the average variance of this spectrum after scaling.

Usage

```
liftToS(x, grid=1:length(x), nsims=100, nscales=9, M=0, ...)
```

Arguments

<code>x</code>	a time series of length <code>n</code> , where <code>n</code> is the number of observations recorded
<code>grid</code>	time positions of the recorded <code>n</code> observations, if the time series has no missing data the grid is <code>1, ..., n</code>
<code>nsims</code>	number of trajectories to use in the NLT algorithm
<code>nscales</code>	number of artificial scales to use in the evolutionary wavelet spectrum estimate
<code>M</code>	smoothing parameter for the spectrum

Value

A list containing

<code>stat</code>	the test statistic
<code>ews</code>	the evolutionary wavelet spectrum estimate, a matrix of dimension <code>nscales x n</code>

Author(s)

Matt Nunes, Marina Knight, Kathryn Leeming and Guy Nason.

References

Knight, M. I. and Nason, G. P.(2009) 'A 'nondecimated' lifting transform', *Statistics and Computing*, 19:1-16.

Knight, M. I., Nunes, M. A., and Nason, G. P.(2012) 'Spectral estimation for locally stationary time series with missing observations', *Statistics and Computing*, 22:877-895.

Examples

```
##perform test on MA(1) process with randomly missing observations
x <- arima.sim(model=list(ma=0.7), n=100)
obs <- sort(sample(1:100, 80))
liftToS(x[obs], grid=obs, nsims=100, nscales=5, M=5)
```

B GNAR documentation

Package ‘GNAR’

November 2, 2018

Type Package**Title** Methods for Fitting Network Time Series Models**Version** 0.2.9**Date** 2018-11-01**Author** Kathryn Leeming [aut],
Guy Nason [aut],
Matt Nunes [aut, cre],
Marina Knight [ctb]**Maintainer** Matt Nunes <nunesrpackages@gmail.com>**Description** Simulation of, and fitting models for, Generalised Network Autoregressive (GNAR) time series models which take account of network structure. Such models are described in Knight et al. (2016), see <arXiv:1603.03221>.**Depends** igraph, wordcloud**License** GPL-2**Encoding** UTF-8**LazyData** true**NeedsCompilation** no**Repository** CRAN**Date/Publication** 2018-11-02 05:20:12 UTC**R topics documented:**

coef.GNARfit	2
fitted.GNARfit	3
fiveNet	4
fiveNode	4
fiveVTS	5
gdp	5
gdpVTS	6
GNAR	6
GNARdesign	7

2		<i>coef.GNARfit</i>
	GNARfit	8
	GNARpredict	9
	GNARsim	11
	GNARtoigraph	12
	GNARtoWMat	12
	igraphtoGNAR	13
	is.GNARfit	14
	is.GNARnet	15
	lmToBIC	16
	matrixtoGNAR	17
	na.row	17
	NofNeighbours	18
	plot.GNARnet	19
	print.GNARfit	19
	print.GNARnet	20
	residToMat	21
	residuals.GNARfit	21
	seed.nos	22
	seedToNet	23
	summary.GNARfit	23
	summary.GNARnet	24
	vswind	25
	windnetplot	26
	Index	27

<code>coef.GNARfit</code>	<i>Function to return coefficients of GNARfit objects</i>
---------------------------	---

Description

`coef.GNARfit` returns the vector of coefficients from a GNARfit object.

Usage

```
## S3 method for class 'GNARfit'
coef(object,...)
```

Arguments

<code>object</code>	the output of a GNARfit or GNARpredict call
<code>...</code>	additional arguments, unused here.

Details

S3 method for class "GNARfit".

fitted.GNARfit

3

Value

`coef.GNARfit` returns a vector of coefficient values.

Examples

```
#get the coefficients of the fiveNode data GNAR fit
data(fiveNode)
coef(GNARfit())
```

`fitted.GNARfit` *Function to return fitted values of GNARfit objects*

Description

`fitted.GNARfit` returns the fitted values of a GNARfit object as a matrix.

Usage

```
## S3 method for class 'GNARfit'
fitted(object,...)
```

Arguments

`object` the output of a GNARfit or GNARpredict call
`...` additional arguments, unused here.

Details

S3 method for class "GNARfit".

Value

`fitted.GNARfit` returns a ts object of fitted values, with `t-alphaOrder` rows and `nnodes` columns.

Examples

```
#get the fitted values of the fiveNode GNAR fit
data(fiveNode)
fitted(GNARfit())
```

4

fiveNode

<code>fiveNet</code>	<i>Example network</i>
----------------------	------------------------

Description

This example GNARnet object is a network containing five vertices connected by five undirected edges, each of length 1.

Usage

```
fiveNet
```

Format

`fiveNet` is a GNARnet object containing `$edges` and `$dist`.
`edges` is a list of length five, with `edges[[i]]` containing the vertices that node `i` is connected to.
`dist` is a list of length five, with `dist[[i]]` containing the length of the vertices that node `i` is connected to.

Examples

```
#Plot this network
data(fiveNode)
plot(fiveNet)
```

<code>fiveNode</code>	<i>Example Network Time Series</i>
-----------------------	------------------------------------

Description

A multivariate time series `fiveVTS` and corresponding network `fiveNet`.

Usage

```
data("fiveNode")
```

Format

This dataset contains two R objects:
`fiveVTS` is a `ts` object with a matrix of 200 rows (`t=200`) and 5 columns (`n=5`)
`fiveNet` is a GNARnet object containing `$edges` and `$dist`.
`edges` is a list of length five, with `edges[[i]]` containing the vertices that node `i` is connected to.
`dist` is a list of length five, with `dist[[i]]` containing the length of the vertices that node `i` is connected to.

fiveVTS

5

Examples

```
data(fiveNode)
plot(fiveNet)
image(fiveVTS)
```

<i>fiveVTS</i>	<i>Simulated GNAR data using fiveNet</i>
----------------	--

Description

This dataset is simulated from the GNAR(1,[1]) model, using *fiveNet* as the accompanying network. The parameters used were $\alpha_1 = 0.2$ for each node, and $\beta_1 = 0.5$. See *GNARfit* for the model description.

Usage

```
fiveVTS
```

Format

fiveVTS is a ts object with a matrix of 200 rows ($t=200$) and 5 columns ($n=5$)

Examples

```
data(fiveNode)
#Plot using ts S3 function
plot(fiveVTS)

#Plot as heatmap
image(fiveVTS)
```

<i>gdp</i>	<i>GDP example data.</i>
------------	--------------------------

Description

Data and seed values for GDP example.

Usage

```
data("gdp")
```

Format

This dataset contains two R objects:
gdpVTS is a ts object with a matrix of 52 rows ($t=52$) and 35 columns ($n=35$) *seed.nos* is a vector containing integer values.

6

GNAR

Examples

```
data(gdp)
image(gdpVTS)
plot(seedToNet(seed.nos[1], nnodes=35, graph.prob=0.15))
```

gdpVTS

*Differenced GDP values for 35 countries***Description**

This dataset is from the OECD (OECD (2018), Quarterly GDP (indicator). <doi:10.1787/b86d1fc8-en> (Accessed on 29 January 2018)) and is differenced annual growth rate for 35 countries.

Usage

```
gdpVTS
```

Format

gdpVTS is a ts object with a matrix of 52 rows (t=52) and 35 columns (n=35)

Examples

```
data(gdp)
#Plot using 'ts' S3 function, can only plot up to 10 columns at once
plot(gdpVTS[,1:5])

#Plot as heatmap
image(gdpVTS)
```

GNAR

*GNAR package***Description**

A package to fit, predict, and simulate time series using the Generalised Network AutoRegressive (GNAR) model. The main functions are GNARfit, which fits the model to a time series and network(s), GNARpredict which fits the GNAR model and produces a within-sample prediction of the last time point, and GNARsim which simulates from a GNAR model with specified parameters. For details of the model, see GNARfit. The package also contains an example network time series in data file fiveNode, with network fiveNet, and simulated time series fiveVTS.

References

Knight, M.I., Nunes, M.A. and Nason, G.P. Modelling, detrending and decorrelation of network time series. arXiv preprint.

GNARdesign

7

GNARdesign *Function to create the GNAR design matrix*

Description

Creates the design matrix necessary for fitting the GNAR model.

Usage

```
GNARdesign(vts = GNAR::fiveVTS, net = GNAR::fiveNet, alphaOrder = 2, betaOrder = c(1,1),
fact.var = NULL, globalalpha=TRUE, tvnets=NULL, netsstart=NULL)
```

Arguments

vts	a matrix or ts object containing the multivariate time series to be modelled. The i, j entry of this matrix should be for time i and vertex/node j .
net	the (first) network associated with the time series, containing a list with entries \$edges and \$dist. This network should have the same number of nodes as the number of columns of the vts matrix.
alphaOrder	a non-negative integer specifying the maximum time-lag to model.
betaOrder	a vector of length alphaOrder specifying the maximum neighbour set to model at each of the time-lags.
fact.var	a vector of factors indicating which nodes belong to each set with different parameters to be fitted.
globalalpha	a TRUE/FALSE value indicating whether to use global alpha parameters.
tvnets	a list of additional networks. Currently only NULL (the static network case) is supported.
netsstart	a vector of times corresponding to the first time points for each network of tvnets. Currently only NULL (the static network case) is supported.

Value

GNARdesign	returns a matrix containing $(t - \alphaOrder)n_{nodes}$ rows and a column for each parameter to be fitted. The columns are in time-lag order, eg for GNAR(2,[1,0]) the columns are alpha1, beta1.1, alpha2. When a factor variable is specified the columns are labelled with the factor.
------------	--

Examples

```
#Design matrix to fit GNAR(2,[1,1]) to the fiveVTS data
data(fiveNode)
GNARdesign()
```

8

GNARfit

GNARfit *Fitting function for GNAR models*

Description

Fits the GNAR model to the given inputs using GNARdesign and lm.

Usage

```
GNARfit(vts=GNAR::fiveVTS, net=GNAR::fiveNet, alphaOrder=2, betaOrder=c(1,1),
fact.var=NULL, globalalpha=TRUE, tvnets=NULL, netsstart=NULL, ErrorIfNoNei=TRUE)
```

Arguments

- vts a matrix containing the multivariate time series to be modelled. The i, j entry of this matrix should be for time i and vertex/node j .
- net the (first) network associated with the time series, containing a list with entries \$edges and \$dist. This network should have the same number of nodes as the number of columns of the vts matrix.
- alphaOrder a non-negative integer specifying the maximum time-lag to model.
- betaOrder a vector of length alphaOrder specifying the maximum neighbour set to model at each of the time-lags.
- fact.var a vector of factors indicating which nodes belong to different sets with different parameters to be fitted.
- globalalpha a TRUE/FALSE value indicating whether to use global alpha parameters.
- tvnets a list of additional networks. Currently only NULL (the static network case) is supported.
- netsstart a vector of times corresponding to the first time points for each network of tvnets. Currently only NULL (the static network case) is supported.
- ErrorIfNoNei a TRUE/FALSE value indicating whether to stop the function call with an error if betaOrder specifies more neighbour sets than exist in the network. If FALSE the function will continue and some parameters will be NA.

Details

The GNAR model of order (p, S) is defined as

$$X_{i,t} = \sum_{j=1}^p \left(\alpha_{i,j} X_{i,t-j} + \sum_{c=1}^C \sum_{r=1}^{S_j} \beta_{j,r,c} \sum_{q \in N_t^{(r)}(i)} \omega_{i,q,c}^{(t)} X_{q,t-j} \right) + u_{i,t}$$

where p is the maximum time lag, $S = (S_1, \dots, S_p)$ and S_j is the maximum stage of neighbour dependence for time lag j , $N_t^{(r)}(i)$ is the r th stage neighbour set of node i at time t , $\omega_{i,q,c}^{(t)}$ is the connection weight between node i and node q at time t if the path corresponds to covariate c . Here,

GNARpredict

9

we consider a sum from one to zero to be zero and $\{u_{i,t}\}$, are assumed to be independent and identically distributed at each node i , with mean zero and variance σ_i^2 . Currently, only a single network GNAR model can be fitted. The connection weight, $\omega_{i,q,c}^{(t)}$, is the inverse of the distance between nodes i and q , normalised so that they sum to 1 for each i , t . See *is.GNARnet* for GNARnet object information and example construction.

Value

`mod` the lm output from fitting the GNAR model.
`y` the original response values, with NAs left in.
`dd` the output of *GNARdesign* containing the design matrix, with NAs left in.
`frbic` inputs to the *GNARfit* function.

References

Knight, M.I., Nunes, M.A. and Nason, G.P. Modelling, detrending and decorrelation of network time series. arXiv preprint.

Examples

```
#Fit the GNAR(2,[1,1]) model to the fiveVTS data
data(fiveNode)
GNARfit()

#Convert the residuals to matrix form
residToMat(GNARfit())$resid
```

`GNARpredict`*Fits and predicts using the GNAR model*

Description

Fits the GNAR model to up to observation $t-1$ and produces a within-sample prediction for time t .

Usage

```
GNARpredict(vts=GNAR::fiveVTS, net=GNAR::fiveNet, alphaOrder=2, betaOrder=c(1,1),
  fact.var=NULL, globalalpha=TRUE, tvnets=NULL, netsstart=NULL, ErrorIfNoNet=TRUE)
```

Arguments

`vts` a matrix containing the multivariate time series to be modelled. The i, j entry of this matrix should be for time i and vertex/node j .
`net` the (first) network associated with the time series, containing a list with entries `$edges` and `$dist`. This network should have the same number of nodes as the number of columns of the `vts` matrix.
`alphaOrder` a non-negative integer specifying the maximum time-lag to model.

10	<i>GNARpredict</i>
<code>betaOrder</code>	a vector of length <code>alphaOrder</code> specifying the maximum neighbour set to model at each of the time-lags.
<code>fact.var</code>	a vector of factors indicating which nodes belong to different set with different parameters to be fitted.
<code>globalalpha</code>	a TRUE/FALSE value indicating whether to use global alpha parameters.
<code>tnets</code>	a list of additional networks. Currently only NULL (the static network case) is supported.
<code>netsstart</code>	a vector of times corresponding to the first time points for each network of <code>tnets</code> . Currently only NULL (the static network case) is supported.
<code>ErrorIfNoNei</code>	a TRUE/FALSE value indicating whether to stop the function call with an error if <code>betaOrder</code> specifies more neighbour sets than exist in the network. If FALSE the function will continue and some parameters will be NA.

Details

See `GNARfit` for GNAR model information. Note that the prediction is for the final time observation of the input data, to predict out-of-sample add a row of zeros to the input data matrix. Only coefficients with p-value smaller than 0.05 are used to calculate prediction. See `is.GNARnet` for GNARnet object information and example construction.

Value

<code>pred</code>	the prediction for time <code>t</code> .
<code>mod</code>	the <code>lm</code> output from fitting the GNAR model up to <code>t-1</code> .
<code>ys</code>	the original response values up to <code>t-1</code> , with NAs left in.
<code>ds</code>	the output of <code>GNARdesign</code> containing the design matrix up to <code>t-1</code> , with NAs left in.
<code>ypred</code>	the original response values at <code>t</code> .
<code>dpred</code>	the time <code>t</code> entries of the design matrix.
<code>frbic</code>	inputs to the <code>GNARpredict</code> function.

Examples

```
#Fit and predict the fiveVTS data with the GNAR(2,[1,1]) model
data(fiveNode)
GNARpredict()
```

GNARsim

11

GNARsim *Simulates a GNAR process*

Description

Simulates a GNAR process with Normally distributed innovations.

Usage

```
GNARsim(n=200, net=GNAR::fiveNet, alphaParams=list(c(rep(0.2,5))),
betaParams=list(c(0.5)), sigma=1, tvnets=NULL, netsstart=NULL)
```

Arguments

n	time length of simulation.
net	network used for the GNAR simulation.
alphaParams	a list containing vectors of auto-regression parameters for each time-lag.
betaParams	a list of equal length as alphaParams containing the network-regression parameters for each time-lag.
sigma	the standard deviation for the innovations.
tvnets	Only NULL is currently supported.
netsstart	Only NULL is currently supported.

Details

Parameter lists should not be NULL, set unused parameters to be zero. See GNARfit for model description.

Value

GNARsim returns the multivariate time series as a ts object, with n rows and a column for each of the nodes in the network.

References

Knight, M.I., Nunes, M.A. and Nason, G.P. Modelling, detrending and decorrelation of network time series. arXiv preprint.

Examples

```
#Simulate a GNAR(1,[1]) process with the fiveNet network
data(fiveNode)
GNARsim()
```

12

GNARtoWMat

GNARtoigraph *Converts a GNAR network to a weighted igraph object*

Description

Takes an input network and neighbour stage and returns it in igraph form.

Usage

```
GNARtoigraph(net=GNAR::fiveNet, stage=1, normalise=FALSE)
```

Arguments

```
net          a GNARnet object containing $edges and dist.
stage       the neighbour set that the adjacency matrix is created for.
normalise   whether to normalise each to non-zero row to have sum one.
```

Details

With normalisation this is a non-invertible transform. See NofNeighbours for neighbour set definition. See is.GNARnet for GNARnet object information and example construction.

Value

GNARtoigraph returns an 'igraph' object with weights as the inverse distances of the input network.

Examples

```
#fiveNet as an igraph object
data(fiveNode)
GNARtoigraph()
```

GNARtoWMat *Converts a GNAR network to a weighted adjacency matrix*

Description

Takes an input GNARnet and neighbour stage and outputs the corresponding adjacency matrix.

Usage

```
GNARtoWMat(net=GNAR::fiveNet, stage=1, normalise=FALSE)
```

igraphtoGNAR

13

Arguments

net a GNARnet object containing \$edges and \$dist.
stage the neighbour set that the adjacency matrix is created for.
normalise whether to normalise each to non-zero row to have sum one.

Details

With normalisation this is a non-invertible transform. See NofNeighbours for neighbour set definition.

Value

GNARtoWMat returns a square matrix with the number of rows and columns equal to the length of the \$edges list. Entry i, j of the matrix will be non-zero if node j is in the stage neighbour set of i .

Examples

```
#fiveNet as an adjacency matrix
data(fiveNode)
GNARtoWMat()
```

<i>igraphtoGNAR</i>	<i>Converts an igraph to GNAR network</i>
---------------------	---

Description

Converts an 'igraph' to the GNARnet form for use as an input to GNAR functions.

Usage

```
igraphtoGNAR(ig)
```

Arguments

ig an 'igraph' object.

Details

The values in the \$dist list are the reciprocal of the values from the weighted adjacency matrix.

Value

igraphtoGNAR returns a GNARnet: a list with elements \$edges and \$dist.

14

*is.GNARfit***Examples**

```
#Convert fiveNet to igraph and back again
data(fiveNode)
igraphtoGNAR(GNARtoigraph(fiveNet))
```

is.GNARfit *Function to check GNARfit objects*

Description

is.GNARfit returns either TRUE or FALSE according to a series of GNARfit checks.

Usage

```
is.GNARfit(x)
```

Arguments

x the object to be tested

Details

The *is.GNARfit* function checks whether the object passes a series of tests that correspond to it being the output of GNARfit or GNARpredict:

- Is it a list containing \$mod and \$frbic
- Does it contain either \$y and \$dd or \$ys and \$ds
- Is \$mod a lm object
- Does \$frbic have the components to calculate the BIC with lmToBIC

Value

is.GNARfit returns TRUE or FALSE corresponding to passing the above tests.

Examples

```
#check that the example fit meets the criteria above
data(fiveNode)
is.GNARfit(GNARfit())
#also check the predict function
is.GNARfit(GNARpredict())
```

is.GNARnet

15

*is.GNARnet**Functions to check and create GNARnet objects***Description**

is.GNARnet returns either TRUE or FALSE according to a series of GNARnet checks. *as.GNARnet* returns a GNARnet object from an input weights matrix, 'igraph' object, or a GNARnet without assigned class.

Usage

```
is.GNARnet(x)
as.GNARnet(x)
```

Arguments

x the network to be tested or object to be converted

Details

The *is.GNARnet* function checks whether the network passes a series of tests:

- Is it a list containing *\$edges* and *\$dist*
- Are the *\$edges* and *\$dist* lists of the same length
- Are each of the elements of *\$edges* the same length as the corresponding *\$dist* element
- Do the edges only contain valid entries, 1,...,nnodes
- Is it labelled as GNARnet class
- Are no duplicate edges present

The *as.GNARnet* function converts igraph objects to GNARnet form, other possible inputs are adjacency matrices, and lists with *\$edges* and *\$dist* entries of the correct form.

Value

is.GNARnet returns TRUE or FALSE corresponding to passing the above tests. *as.GNARnet* returns a GNARnet object.

Examples

```
#check that the example network meets the criteria above
data(fiveNode)
is.GNARnet(fiveNet)

#convert to igraph and back again
as.GNARnet(GNARtoigraph(fiveNet))

#generate a new network with three nodes
```

16

lmToBIC

```

#edges 1->2, 2->1, 2->3
#dist 1, 2, 1
#note 1->2 and 2->1 are of different lengths
threeEdge <- list(c(2), c(1,3), NULL)
threeDist <- list(c(1), c(2,1), NULL)
threeNet <- list(edges=threeEdge, dist=threeDist)
#check if this is a GNARnet
is.GNARnet(threeNet)
#use as.GNARnet to change the class
threeNet <- as.GNARnet(threeNet)
#check if this is a GNARnet now
is.GNARnet(threeNet)

```

lmToBIC

Calculates the BIC for a GNAR model fit

Description

Returns the value of the BIC for a GNARfit or GNARpredict object.

Usage

```
lmToBIC(GNARobj.in=GNARpredict())
```

Arguments

GNARobj.in output of GNARfit or GNARpredict.

Details

For a GNAR($p, [S]$) model, the BIC is calculated as

$$BIC(p, S) = \ln |\Sigma_{p,S}| + T^{-1} M \ln(T)$$

where $\Sigma_{p,S} = T^{-1} U' U$, U is the matrix of residuals, and $M = np + C \sum_{j=1}^p S_j$ when `globalalpha=FALSE`, $M = p + C \sum_{j=1}^p S_j$ when `globalalpha=TRUE`, and C is the number of factors.

For definiteness, we seek to minimize the BIC. Often, better models have a small BIC.

Value

lmToBIC returns the value of the BIC for the input model fit.

Examples

```

#BIC for the GNAR(2,[1,1]) model using GNARpredict on fiveVTS
data(fiveNode)
lmToBIC()

```

matrixtoGNAR

17

<i>matrixtoGNAR</i>	<i>Converts an adjacency matrix to GNAR network</i>
---------------------	---

Description

Converts an adjacency matrix to the GNARnet form for use as an input to GNAR functions.

Usage

```
matrixtoGNAR(input.mat)
```

Arguments

<code>input.mat</code>	an adjacency matrix whose dimension equals the number of nodes in the resulting network.
------------------------	--

Details

The values in the `$dist` list are the reciprocal of the values from the weighted adjacency matrix.

Value

`matrixtoGNAR` returns a GNARnet list with elements `$edges` and `$dist`.

Examples

```
#Convert fiveNet to an adjacency matrix and back again
data(fiveNode)
matrixtoGNAR(GNARtoWMat())
```

<i>na.row</i>	<i>Identifies which rows of a matrix have NAs</i>
---------------	---

Description

Returns a vector with elements TRUE/FALSE identifying which rows contain NA elements.

Usage

```
na.row(mat)
```

Arguments

<code>mat</code>	a matrix object.
------------------	------------------

18

*NofNeighbours***Details**

This function is used in the unstacking of residuals into a residual matrix and replacing NAs where they were previously present.

Value

`na.row` returns a vector of length equal to the number of rows in `mat`. Each element is either TRUE or FALSE.

Examples

```
#Check if there are and NAs in fiveVTS
data(fiveNode)
na.row(fiveVTS)
```

`NofNeighbours`*Calculates stage-neighbours of a network*

Description

Calculates neighbour sets of a particular node in the network and their distances.

Usage

```
NofNeighbours(node=1, stage=2, net=GNAR::fiveNet)
```

Arguments

`node` is an integer specifying which node to calculate the neighbours of.
`stage` is an integer specifying the maximum neighbour-stage to calculate to.
`net` a GNARnet object with edge list and distance list.

Details

Note that the distances are calculated as the sum along the shortest path; do not use this with a weights (rather than distance) list. Stage- r neighbours of node i are denoted $N^{(r)}(i)$, and are nodes that are r edges (but no fewer) away from i . Hence stage-1 neighbours are the immediate neighbours, stage-2 neighbours are the neighbours of neighbours and so on.

Value

`edges` is a list of length `stage`, where `edges[[i]]` is a vector containing the nodes that are stage- i neighbours of node.
`dist` is a list of length `stage`, where `dist[[i]]` is a vector containing the distances from node to its stage- i neighbours, with ordering as in `edges[[i]]`.

plot.GNARnet

19

Examples

```
#First and second stage neighbours of node 1 in fiveNet
data(fiveNode)
NofNeighbours()
```

<i>plot.GNARnet</i>	<i>Plot function for GNAR networks</i>
---------------------	--

Description

Plots a GNAR network using the 'igraph' package.

Usage

```
## S3 method for class 'GNARnet'
plot(x, ...)
```

Arguments

x the networkGNARnet object associated with the time series, containing a list with entries \$edges and \$dist.

... additional arguments for the igraph plotting function, see plot.igraph.

Details

S3 method for class "GNARnet".

Examples

```
#Plot fiveNet
data(fiveNode)
plot.GNARnet(fiveNet)
```

<i>print.GNARfit</i>	<i>Function to print the model and coefficients of GNARfit objects</i>
----------------------	--

Description

print.GNARfit prints model, call, and coefficients of a GNARfit object.

Usage

```
## S3 method for class 'GNARfit'
print(x,...)
```

20

*print.GNARnet***Arguments**

x the output of a GNARfit or GNARpredict call
 ... additional arguments, unused here.

Details

S3 method for class "GNARfit".

Examples

```
#print the information of the fiveNode GNAR fit
data(fiveNode)
print(GNARfit())
```

```
print.GNARnet            Print function for GNAR networks
```

Description

Prints information about a GNAR network.

Usage

```
## S3 method for class 'GNARnet'
print(x, ...)
```

Arguments

x the networkGNARnet object associated with the time series, containing a list with entries \$edges and \$dist.
 ... additional arguments, unused here.

Details

S3 method for class "GNARnet".

Examples

```
#print fiveNet information
data(fiveNode)
print.GNARnet(fiveNet)
```

residToMat

21

<code>residToMat</code>	<i>Converts the output of a GNARfit or GNARpredict to fitted and residual value matrices</i>
-------------------------	--

Description

Unstacks the entries of the GNARfit or GNARpredict fitted and residual values to return matrices of a similar form to the multivariate time series input.

Usage

```
residToMat(GNARobj=GNARpredict(), nnodes=5)
```

Arguments

<code>GNARobj</code>	the output from the GNARfit or GNARpredict function
<code>nnodes</code>	the number of nodes in the original network time series

Details

This function also replaces the NAs that were removed in fitting.

Value

<code>resid</code>	is the matrix of residual values, with <code>t-alphaOrder</code> rows and <code>nnodes</code> columns.
<code>fit</code>	is the matrix of fitted values, with <code>t-alphaOrder</code> rows and <code>nnodes</code> columns.

Examples

```
#Get residual and fitted matrices from GNARpredict fit of fiveVTS
data(fiveNode)
residToMat()
```

<code>residuals.GNARfit</code>	<i>Function to return residuals of GNARfit objects</i>
--------------------------------	--

Description

`residuals.GNARfit` returns the residuals of a GNARfit object as a matrix.

Usage

```
## S3 method for class 'GNARfit'
residuals(object,...)
```


22

*seed.nos***Arguments**

`object` the output of a GNARfit or GNARpredict call
`...` additional arguments, unused here.

Details

The function first checks if the object is of GNARfit class, then uses `residToMat` to return the residuals.

Value

`residuals.GNARfit` returns a 'ts' object of residuals, with `t-alphaOrder` rows and `nnodes` columns.

Examples

```
#get the residuals of the fiveNode GNAR fit
data(fiveNode)
residuals(GNARfit())
```

<code>seed.nos</code>	<i>Vector of seed numbers</i>
-----------------------	-------------------------------

Description

Seed numbers for reproducible random graphs.

Usage

```
seed.nos
```

Format

`seed.nos` is a vector of length 10,000 containing integers.

Examples

```
data(gdp)
g <- seedToNet(seed.nos[1], nnodes=35, graph.prob=0.15)
plot(g, vertex.label=colnames(gdpVTS), vertex.size=0)
```

`seedToNet`

23

<code>seedToNet</code>	<i>Produces a random network from a seed value</i>
------------------------	--

Description

Produces a reproducible undirected Erdos-Reyni random network using a particular seed value.

Usage

```
seedToNet(seed.no, nnodes=34, graph.prob=0.5)
```

Arguments

<code>seed.no</code>	a valid number to set the seed to.
<code>nnodes</code>	the number of nodes in the produced network.
<code>graph.prob</code>	the probability that each pair of nodes is connected.

Details

`graph.prob` effectively controls the sparsity of the network. All distances are set to 1.

Value

A GNARnet object.

Examples

```
#Generate the random graph from seed 10, with 5 nodes and connection prob 0.5
seedToNet(10,nnodes=5,graph.prob=0.5)
```

<code>summary.GNARfit</code>	<i>Returns model summary for a GNAR model fit</i>
------------------------------	---

Description

Returns the summary of a GNARfit or GNARpredict object, including BIC.

Usage

```
## S3 method for class 'GNARfit'
summary(object, ...)
```

Arguments

<code>object</code>	output of GNARfit or GNARpredict.
<code>...</code>	additional arguments affecting the summary produced.

24

*summary.GNARnet***Details**

The output is the summary of the fit using `summary.lm`, and BIC calculated using `lmToBIC`.

Value

`summary.GNARfit` prints the model summary and the value of the BIC.

Examples

```
#summary for the GNAR(2,[1,1]) model using GNARpredict on fiveVTS
data(fiveNode)
summary(GNARpredict())
```

<code>summary.GNARnet</code>	<i>Summary function for GNAR networks</i>
------------------------------	---

Description

Prints brief information about a GNAR network.

Usage

```
## S3 method for class 'GNARnet'
summary(object, ...)
```

Arguments

<code>object</code>	the networkGNARnet object associated with the time series, containing a list with entries <code>\$edges</code> and <code>\$dist</code> .
<code>...</code>	additional arguments, unused here.

Details

S3 method for class "GNARnet".

Examples

```
#print fiveNet summary information
data(fiveNode)
summary.GNARnet(fiveNet)
```

vswind

25

vswind

*Wind Speed example network time series***Description**

A suite of data objects concerning wind speed analysis. The dataset contains a multivariate time series of wind speeds, two network descriptions, a vector of names for weather stations, and the coordinates of the weather stations.

Usage

```
data("vswind")
```

Format

This dataset contains six R objects:

vswindts is a ts object with a matrix of 721 rows (t=721) and 102 columns (n=102). This corresponds to 721 observations made through time at 102 weather stations. vswindnetD is a GNARnet object containing \$edges and \$dist.

edges is a list of length 102, with edges[[i]] containing the vertices that node i is connected to.

dist is a list of length 102, with dist[[i]] containing the length of the vertices that node i is connected to.

vswindnet is the same as vswindnetD except that all the distances are replaced by 1.

vswindnames is a character vector of length 102 containing the wind speed site names and

vswindcoords is a matrix with 102 rows (one for each wind station) and two columns providing

the x and y coordinates of the weather stations.

Source

The base data were obtained from the <http://www.metoffice.gov.uk> UK Met Office Weather-ObservationsWebsite distributed under the UK Open Government License <http://www.nationalarchives.gov.uk/doc/open-government-licence/version/1/> Contains public sector information licensed under the Open Government Licence v1.0.

See Also

```
windnetplot
```

Examples

```
data(vswind)
#
# The name entry for Bristol
#
vswindnames[77]
#[1] "BRIST"
#
# plot the distance network
#
## Not run: windnetplot()
```

26

windnetplot

windnetplot	<i>Produce bespoke plot of the wind data network</i>
-------------	--

Description

Plots the wind speed data network with distance information.

Usage

```
windnetplot()
```

Arguments

None.

Details

The wind speed data is to be found in the `vswind` data set. This function contains commands, using functionality from the `wordcloud` package, to plot the network, with node names and edges. Distances between nodes are plotted next to the edges.

See Also

`vswind`

Examples

```
data(vswind)
## Not run: windplotnet()
```

Index

*Topic **datasets**
 fiveNode, 4
 gdp, 5
 vswind, 25

as.GNARnet (is.GNARnet), 15

coef.GNARfit, 2

fitted.GNARfit, 3
 fiveNet, 4, 4, 6
 fiveNode, 4, 6
 fiveVTS, 4, 5, 6

gdp, 5
 gdpVTS, 6
 GNAR, 6
 GNARdesign, 7
 GNARfit, 5, 6, 8, 10, 11, 14, 21
 GNARpredict, 6, 9, 14, 21
 GNARsim, 6, 11
 GNARtoigraph, 12
 GNARtoWMat, 12

igraph, 12, 15
 igraphtoGNAR, 13
 is.GNARfit, 14
 is.GNARnet, 9, 10, 12, 15

lm, 14
 lmToBIC, 14, 16, 24

matrixtoGNAR, 17

na.row, 17
 NofNeighbours, 12, 13, 18

plot.GNARnet, 19
 plot.igraph, 19
 print.GNARfit, 19
 print.GNARnet, 20

residToMat, 21, 22
 residuals.GNARfit, 21

seed.nos, 22
 seedToNet, 23
 summary.GNARfit, 23
 summary.GNARnet, 24
 summary.lm, 24

ts, 3–7, 11, 25

vswind, 25, 26
 vswindcoords (vswind), 25
 vswindnames (vswind), 25
 vswindnet (vswind), 25
 vswindnetD (vswind), 25
 vswindts (vswind), 25

windnetplot, 25, 26

C LWN documentation

Package ‘LWN’

March 9, 2019

Type Package**Title** Local White Noise Tests**Version** 0.1.6**Author** Kathryn Leeming and Guy Nason**Maintainer** Kathryn Leeming <kathryn.leeming@bristol.ac.uk>**Description** Performs local white noise tests using the wavelet spectrum, and contains associated wavelet functions.**Depends** AutoSpec, locits, CompQuadForm, wavethresh, graphics, fractal**License** GPL-2**Encoding** UTF-8**LazyData** true**R topics documented:**

covKRM	2
covKRMmat	3
getBetaMat	3
getBetaMatS	4
haarsqsum2	5
JNC	5
lwntest	6
makeRJ	7
psi.jl	8
psi.k	9
PsiShannon	9
shannonmo	10
shannonpsi2	10
shannonsqsum2	11
sigToP	11
waldByHand	12
Index	13

2

covKRM

covKRM	<i>Covariance of the running mean smoothed wavelet spectrum under white noise</i>
--------	---

Description

Returns the value of the expected covariance between scales j, l of the running mean smoothed wavelet spectrum under the null hypothesis of white noise. This covariance is between scales at the same time point.

Usage

```
covKRM(j, l, s, sigma2=1, wavelet, kurts4, smoothing="RM", kerin=NULL)
```

Arguments

j	Scale of the spectrum.
l	Scale of the spectrum.
s	Kernel bandwidth of the running mean smoother, s points either side of the central observation are included in smoothing.
sigma2	Variance of the white noise process.
wavelet	Either "Haar" or "Shannon", wavelet used in construction of the spectrum.
kurts4	Excess kurtosis of the white noise process, for Gaussian white noise this is 0.
smoothing	Type of kernel smoothing used on the spectrum.
kerin	The kernel used for smoothing

Details

This function may be slow for large s as it calculates and sums $2s+1$ values. Called in covKRMmat to create the covariance matrix.

Value

The value of the theoretical covariance using estimated excess kurtosis.

Examples

```
#covariance between different scales with Shannon wavelet should be zero
covKRM(j=2, l=3, s=10, wavelet="Shannon", kurts4=0)
#non-zero at different scales
covKRM(j=2, l=2, s=10, wavelet="Shannon", kurts4=0)
```


covKRMmat

3

covKRMmat	<i>Covariance matrix of the running mean smoothed wavelet spectrum under white noise</i>
-----------	--

Description

Constructs a matrix giving the expected covariance of the wavelet spectrum under the null hypothesis of white noise.

Usage

```
covKRMmat(nscases, ss, ssigma, wavelet, krt, smoothtype="RM")
```

Arguments

nscases	Number of scales to calculate the covariance for.
ss	Kernel bandwidth for smoothing, set to 0 for no smoothing. Running mean smooth is carried out with ss observations either side of the central observation.
ssigma	Variance of the white noise process.
wavelet	Either "Haar" or "Shannon".
krt	Excess kurtosis of the white noise process, for Gaussian noise this is 0.
smoothtype	Type of kernel smoothing used for the spectrum.

Details

Calls covKRM for each entry of the matrix. May be slow for large smoothing parameter.

Value

A nscases x nscases matrix of covariances.

Examples

```
## Generate the covariance matrix for the first three scales of an
## unsmoothed Shannon spectrum of a standard Gaussian noise process
covKRMmat(nscases=3, ss=0, ssigma=1, wavelet="Shannon", krt=0)
```

getBetaMat	<i>Returns Haar beta spectrum as a matrix</i>
------------	---

Description

Performs the Haar wavelet transform and returns the details as a matrix.

Usage

```
getBetaMat(x, xjnc)
```

4 *getBetaMatS*

Arguments

`x` Input time series.
`xjnc` Maximum scale of the beta spectrum to return.

Value

A matrix of the Haar wavelet beta spectrum with `xjnc` rows and `length(x)` columns.

Examples

```
## The function is currently defined as
getBetaMat(rnorm(32), xjnc=3)
```

`getBetaMatS` *Returns Shannon beta spectrum as a matrix*

Description

Performs the Shannon wavelet transform and returns the details as a matrix.

Usage

```
getBetaMatS(x, xjnc)
```

Arguments

`x` Input time series.
`xjnc` Maximum scale to return of the Shannon beta spectrum.

Value

A matrix of dimension `xjnc` x `T`, where `T` is the length of `x`.

Examples

```
getBetaMatS(rnorm(32), xjnc=3)
```

`haarsqsum2`

5

<code>haarsqsum2</code>	<i>Calculates sum of squared Haar wavelets</i>
-------------------------	--

Description

Returns the sum over squared Haar wavelets

Usage`haarsqsum2(1, j, m, n)`**Arguments**

<code>1</code>	Haar scale
<code>j</code>	Haar scale
<code>m</code>	Time location corresponding to scale 1
<code>n</code>	Time location corresponding to scale j

Details

$$\sum_t \psi_{1,m-t}^2 \psi_{j,n-t}^2$$

Value

A single value

Examples

```
##compare a non-overlapping segment
haarsqsum2(1,2,0,10)
##and an overlapping segment
haarsqsum2(1,2,0,1)
```

<code>JNC</code>	<i>Returns the number of non-cone scales for the Haar wavelet.</i>
------------------	--

Description

Integer of non-cone scales reflecting the length of series.

Usage`JNC(TT)`**Arguments**

<code>TT</code>	Length of series.
-----------------	-------------------

6

lwntest

Value

An integer.

Examples

```
##non-cone scales for T=32
JNC(32)
##non-cone scales increases with T
JNC(256)
```

lwntest	<i>Function to carry out local white noise test(s)</i>
---------	--

Description

Runs a specified number of local white noise tests on an input set of observations

Usage

```
lwntest(x, wavelet="Shannon", ss=0, nsc=NULL, asGaussian=FALSE,
        testtype="contrast", contrasttype="com",
        ntests=1, testsat=NULL, padj="holm",
        smoothtype="RM")
```

Arguments

x	Series of observations (needs to be power of two in length).
wavelet	Choice of wavelet for analysis, either "Haar" or "Shannon".
ss	Kernel bandwidth for running mean smooth, ss points either side of the test position are used for smoothing.
nsc	Number of scales to use in each test, or NULL if number of scales is the number of Haar non-cone scales.
asGaussian	TRUE/FALSE whether to treat input observations as Gaussian data. If TRUE the excess kurtosis is calculated but not used in covariance, if the default, FALSE, the excess kurtosis is used in calculations of the covariance.
testtype	Either "quad" or "contrast", choice of estimating and subtracting the mean from the spectrum ("quad"), or testing whether the values are the same using contrasts ("contrast").
contrasttype	The type of contrast to calculate, either "alt", "opp", or "com" for alternate, opposite, or complete contrasts. See makeRJ for details.
ntests	How many tests to perform on the observations, these will be evenly spaced throughout. If too many tests are specified for the length of observations and smoothing parameter this will result in an error.
testsat	A vector of positions specifying where tests should be performed, generally use ntests instead.
padj	Type of p-value adjustment to perform for multiple hypotheses, input to p.adjust.

makeRJ

7

smoothtype Type of smoothing to be performed on the spectrum, either "RM", "Epan", or "Tri", corresponding to running mean, Epanechnikov, and triangular kernel smoothing, respectively.

Value

An object of class "lwn" with the containing the following:

test A ntests row matrix with the following columns: p-value(adj)adjusted p-values of tests (only included if ntests is greater than 1) p-valuep-values of tests test statTest statistics of tests xkurtosisEstimated excess kurtosis sigmahatEstimated variance

times a list containing the intervals (starts, fins) and midpoints (mid) of the tested sections

xx the input time series

s the bandwidth used in smoothing

css NULL - added for future features of the test.

Note

S3 methods plot and print are available for "lwn" objects.

Examples

```
##run one local white noise test on Gaussian noise with defaults
x <- rnorm(128)
lwnetest(x)
##run two tests
lwnetest(x, ntests=2)
##add running mean smoothing and plot test results
plot(lwnetest(x, ntests=2, ss=30))
```

*makeRJ**Constructs contrast vector or matrix*

Description

Returns a matrix or vector containing contrasts

Usage

```
makeRJ(J, contype="alt")
```

Arguments

J Number of scales, or length of observations to test for equal mean.

contype "alt", "opp", or "com"; "alt" is alternate 1s and -1s, "opp" is 1s for the first half of the vector and -1s in the second half, and "com" is a complete set of contrasts.

Details

For contype="com", J can take a maximum value of 7.

8

*psi.jl***Value**

A matrix of dimension $1 \times J$ for "alt" or "com", and a matrix of dimension $J-1 \times J$ for "com".

Examples

```
##compare the three contrast matrices for J=5
makeRJ(5, contype="alt")
makeRJ(5, contype="opp")
makeRJ(5, contype="com")
```

psi.jl

Cross-correlation Haar wavelet

Description

Cross-correlation Haar wavelet

Usage

```
psi.jl(j,1,tau)
```

Arguments

j	Haar wavelet scale.
1	Haar wavelet scale.
tau	Value to evaluate the cross-correlation at.

Value

A single number

Examples

```
##Haar cross-correlation at the same scale has value 1 at 0
psi.jl(1,1,0)
psi.jl(2,2,0)
psi.jl(2,2,2)
```

`psi.k`

9

`psi.k` *Function to return the value of the Haar Autocorrelation wavelet.*

Description

Haar autocorrelation wavelet

Usage`psi.k(k, tau)`**Arguments**

`k` Haar wavelet scale.
`tau` Value to evaluate the autocorrelation wavelet at.

Value

A single number

Examples

```
##Haar autocorrelation wavelet is 1 at 0 for all scales
psi.k(1,0)
psi.k(10,0)
```

`PsiShannon` *Function to return the value of the Shannon Autocorrelation wavelet*

Description

Shannon Autocorrelation wavelet

Usage`PsiShannon(j,k)`**Arguments**

`j` Shannon wavelet scale.
`k` Value to evaluate the autocorrelation wavelet at.

Value

A single number.

Examples

```
##
PsiShannon(1,0)
PsiShannon(1,1)
PsiShannon(1,10)
```

10

shannonpsi2

shannonmo *Function to return the value of the Shannon mother wavelet*

Description

Shannon mother wavelet

Usage

shannonmo(x)

Arguments

x Input value

Value

A single number.

Examples

shannonmo(0.5)

shannonpsi2 *Returns the value of the Shannon wavelet*

Description

Shannon wavelet

Usage

shannonpsi2(x,j,k)

Arguments
x Time location
j Scale
k Position
Value

A single number.

Examples

shannonpsi2(0,1,0)

shannonsqsum2

11

<i>shannonsqsum2</i>	<i>Calculates the value of the sum of squared Shannon wavelets</i>
----------------------	--

Description

A numerical computation to find the value of the sum of squared Shannon wavelets

Usage

```
shannonsqsum2(j, l, m, n, maxt)
```

Arguments

j	Shannon wavelet scale.
l	Shannon wavelet scale.
m	Time position.
n	Time position.
maxt	Sum calculated numerically using values from -maxt to maxt.

Details

$$\sum_t \psi_{l,m-t}^2 \psi_{j,n-t}^2$$

This is a numerical calculation and may be slow for large maxt.

Value

A single number.

Examples

```
shannonsqsum2(1, 1, 0, 0)
##zero at different scales
shannonsqsum2(1, 2, 0, 0)
```

<i>sigToP</i>	<i>Calculates p-value of the quadratic form test statistic using Imhof's method</i>
---------------	---

Description

Converts the test statistic and covariance matrix into a p-value

Usage

```
sigToP(HSigma, stat)
```

12

*waldByHand***Arguments**

HSigma The covaraince matrix of the quadratic form.
 stat The value of the quadratic form.

Details

This function uses imhof from the CompQuadForm package.

Value

A single number in [0,1].

Note

If the p-value returned from the imhof function is negative, a warning is shown and the p-value is set to 0.

 waldByHand

Performs a contrast Wald test and returns the test statistic and p-value

Description

Contrast test using matrix multiplication

Usage

```
waldByHand(theta, Rmat, Sigmat, sigmahat)
```

Arguments

theta Estimated coefficients (spectrum)
 Rmat Contrast matrix or vector.
 Sigmat Covariance matrix of theta for a process with variance 1.
 sigmahat Estimated variance.

Value

tstat Value of the contrast test statistic.
 pval p-value of the Wald contrast test.

Index

covKRM, 2, 3
covKRMat, 2, 3
getBetaMat, 3
getBetaMatS, 4
haarsqsum2, 5
imhof, 12
JNC, 5
lwnetest, 6
makeRJ, 6, 7
p.adjust, 6
psi.jl, 8
psi.k, 9
PsiShannon, 9
shannonmo, 10
shannonpsi2, 10
shannonsqsum2, 11
sigToP, 11
waldByHand, 12

References

- ADAK, S. (1998) ‘Time-dependent spectral analysis of nonstationary time series.’ *Journal of the American Statistical Association*, **93**:1488–1501.
- AKAY, M. (1996) ‘Diagnosis of coronary artery disease using wavelet-based neural networks.’ In *Wavelets in Medicine and Biology*, ALDROUBI, A. and UNSER, M., eds. Boca Raton: CRC Press.
- ALPERT, B. K. (2002) ‘Construction of simple multiscale bases for fast matrix operations.’ In *Wavelets and Their Applications*, RUSKAI, M. B., BEYLKIN, G., COIFMAN, R., DAUBECHIES, I., MALLAT, S., MEYER, Y., and RAPHAEL, L., eds. Boston: Jones and Bartlett.
- ANACLETO, O. and QUEEN, C. (2017) ‘Dynamic chain graph models for time series network data.’ *Bayesian Analysis*, **12**:491–509.
- BARAGONA, R., BATTAGLIA, F., and CUCINA, D. (2018) ‘Portmanteau tests based on quadratic forms in the autocorrelations.’ *Communications in Statistics - Theory and Methods*, **47**:4355–4374.
- BARTLETT, M. S. (1955) *An Introduction to Stochastic Processes with Special Reference to Methods and Applications*. Cambridge: Cambridge University Press.
- BOSE, N. K., LERTRATTANAPANICH, S., and CHAPPALLI, M. B. (2004) ‘Superresolution with second generation wavelets.’ *Signal Processing: Image Communication*, **19**:387–391.
- BOX, G. E. P. and PIERCE, D. A. (1970) ‘Distribution of residual autocorrelations in autoregressive integrated moving average time series models.’ *Journal of the American Statistical Association*, **65**:1509–1526.
- BROCKWELL, P. J. and DAVIS, R. A. (2006) *Time series: theory and methods*. New York: Springer, 2nd edition.
- BUSETTI, F. and TAYLOR, A. M. R. (2005) ‘Stationarity tests for irregularly spaced observations and the effects of sampling frequency on power.’ *Econometric Theory*, **21**:757–794.

- CARDINALI, A. and NASON, G. P. (2010) ‘Costationarity of locally stationary time series.’ *Journal of Time Series Econometrics*, **2**:1941–1928.
- CARDINALI, A. and NASON, G. P. (2017) ‘Locally stationary wavelet packet processes: basis selection and model fitting.’ *Journal of Time Series Analysis*, **38**:151–174.
- CARDINALI, A. and NASON, G. P. (2018) ‘Practical powerful wavelet packet tests for second-order stationarity.’ *Applied and Computational Harmonic Analysis*, **44**:558–583.
- CHATFIELD, C. (2004) *The Analysis of Time Series: An Introduction*. Boca Raton: CRC Press, 6th edition.
- CHO, H. (2016a) ‘A test for second-order stationarity of time series based on unsystematic subsamples.’ *Stat*, **5**:262–277.
- CHO, H. (2016b) *unsystation: Stationarity Test Based on Unsystematic Sub-Sampling*. URL <https://people.maths.bris.ac.uk/~mahrc/research.html>, R package version 0.1.0.
- CHO, H. and FRYZLEWICZ, P. (2012) ‘Multiscale and multilevel technique for consistent segmentation of nonstationary time series.’ *Statistica Sinica*, **22**:207–229.
- CHUI, C. K. (1997) *Wavelets: a Mathematical Tool for Signal Analysis*. Philadelphia: Society for Industrial and Applied Mathematics.
- CONSTANTINE, W. and PERCIVAL, D. (2014) *fractal: Fractal Time Series Modeling and Analysis*. R package version 2.0-1.
- DAHLHAUS, R. (1996) ‘On the Kullback-Leibler information divergence of locally stationary processes.’ *Stochastic Processes and their Applications*, **62**:139–168.
- DAHLHAUS, R. (1997) ‘Fitting time series models to nonstationary processes.’ *The Annals of Statistics*, **25**:1–37.
- DAUBECHIES, I. (1992) *Ten Lectures on Wavelets*. Philadelphia: SIAM.
- DAVIS, R. A., ZANG, P., and ZHENG, T. (2016) ‘Sparse vector autoregressive modeling.’ *Journal of Computational and Graphical Statistics*, **25**:1077–1096.

- DETTE, H., PREUSS, P., and VETTER, M. (2011) ‘A measure of stationarity in locally stationary processes with applications to testing.’ *Journal of the American Statistical Association*, **106**:1113–1124.
- DOREIAN, P. (1981) ‘Estimating linear models with spatially distributed data.’ *Sociological Methodology*, **12**:359–388.
- DUCHESNE, P. and DE MICHEAUX, P. L. (2010) ‘Computing the distribution of quadratic forms: Further comparisons between the Liu-Tang-Zhang approximation and exact methods.’ *Computational Statistics and Data Analysis*, **54**:858–862.
- DWIVEDI, Y. and SUBBA RAO, S. (2011) ‘A test for second-order stationarity of a time series based on the discrete Fourier transform.’ *Journal of Time Series Analysis*, **32**:68–91.
- ECKLEY, I. A. and NASON, G. P. (2018) ‘A test for the absence of aliasing or local white noise in locally stationary wavelet time series.’ *Biometrika*, **105**:833–848.
- ECKLEY, I. A., NASON, G. P., and TRELOAR, R. L. (2010) ‘Locally stationary wavelet fields with application to the modelling and analysis of image texture.’ *Journal of the Royal Statistical Society: Series C*, **59**:595–616.
- FAN, J., LIAO, Y., and MINCHEVA, M. (2013) ‘Large covariance estimation by thresholding principal orthogonal complements.’ *Journal of the Royal Statistical Society: Series B*, **75**:603–680.
- FERREIRO, O. (1987) ‘Methodologies for the estimation of missing observations in time series.’ *Statistics & Probability Letters*, **5**:65–69.
- FIELD, D. J. (2000) ‘Wavelets, vision and the statistics of natural scenes.’ In *Wavelets: The Key to Intermittent Information?*, SILVERMAN, B. W. and VASSILICOS, J. C., eds. Oxford: Oxford University Press.
- FRYZLEWICZ, P. and NASON, G. P. (2006) ‘Haar-Fisz estimation of evolutionary wavelet spectra.’ *Journal of the Royal Statistical Society: Series B*, **68**:611–634.
- FRYZLEWICZ, P. and OMBAO, H. (2009) ‘Consistent classification of nonstationary time series using stochastic wavelet representations.’ *Journal of the American Statistical Association*, **104**:299–312.

- FRYZLEWICZ, P., VAN BELLEGEM, S., and VON SACHS, R. (2003) ‘Forecasting non-stationary time series by wavelet process modelling.’ *Annals of the Institute of Statistical Mathematics*, **55**:737–764.
- GOERG, G. M. (2012) ‘Testing for white noise against locally stationary alternatives.’ *Statistical Analysis and Data Mining*, **5**:478–492.
- GRIMMETT, G. (2010) *Probability on Graphs: Random Processes on Graphs and Lattices*. Cambridge: Cambridge University Press.
- HAYS, J. C., KACHI, A., and FRANZESE JR., R. J. (2010) ‘A spatial model incorporating dynamic, endogenous network interdependence: A political science application.’ *Statistical Methodology*, **7**:406–428.
- HOLM, S. (1979) ‘A simple sequentially rejective multiple test procedure.’ *Scandinavian Journal of Statistics*, **6**:65–70.
- HONG, B. and CHEN, C. H. (2003) ‘Radial basis function neural network-based nonparametric estimation approach for missing data reconstruction of non-stationary series.’ In *Proceedings of the 2003 International Conference on Neural Networks and Signal Processing, 2003*, volume 1, 75–78.
- HUANG, K. and AVIYENTE, S. (2008) ‘Wavelet feature selection for image classification.’ *IEEE Transactions on Image Processing*, **17**:1709–1720.
- HURD, H. L. and GERR, N. L. (1991) ‘Graphical methods for determining the presence of periodic correlation.’ *Journal of Time Series Analysis*, **12**:337–350.
- HYNDMAN, R. J. and KHANDAKAR, Y. (2008) ‘Automatic time series forecasting: the forecast package for R.’ *Journal of Statistical Software*, **27**:1–22.
- IMHOF, J. (1961) ‘Computing the distribution of quadratic forms in normal variables.’ *Biometrika*, **48**:419–426.
- ISSERLIS, L. (1918) ‘On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables.’ *Biometrika*, **12**:134–139.
- JAFFARD, S., MEYER, Y., and RYAN, R. D. (2001) *Wavelets: Tools for Science and Technology*. Philadelphia: SIAM.

- JANSEN, M., NASON, G. P., and SILVERMAN, B. W. (2006) ‘Multivariate nonparametric regression using lifting.’ *Technical Report 08:07*, Statistics Group, Department of Mathematics, University of Bristol.
- JANSEN, M., NASON, G. P., and SILVERMAN, B. W. (2009) ‘Multiscale methods for data on graphs and irregular multidimensional situations.’ *Journal of the Royal Statistical Society: Series B*, **71**:97–125.
- JAWERTH, B. and SWELDENS, W. (1994) ‘An overview of wavelet based multiresolution analyses.’ *SIAM Review*, **36**:377–412.
- JIN, L., WANG, S., and WANG, H. (2015) ‘A new non-parametric stationarity test of time series in the time domain.’ *Journal of the Royal Statistical Society: Series B*, **77**:893–922.
- JOHNSTONE, I. M. (2000) ‘Wavelets and the theory of non-parametric function estimation.’ In *Wavelets: The Key to Intermittent Information?*, SILVERMAN, B. W. and VASSILICOS, J. C., eds. Oxford: Oxford University Press.
- KILLICK, R., KNIGHT, M. I., NASON, G. P., and ECKLEY, I. A. (2019) ‘The local partial autocorrelation function and some applications.’ (*Submitted for publication*).
- KNIGHT, M. and NUNES, M. (2012) *nlt: A nondecimated lifting transform for signal denoising*. R package version 2.1-3.
- KNIGHT, M. I. and NASON, G. P. (2009) ‘A ‘nondecimated’ lifting transform.’ *Statistics and Computing*, **19**:1–16.
- KNIGHT, M. I., NUNES, M. A., and NASON, G. P. (2012) ‘Spectral estimation for locally stationary time series with missing observations.’ *Statistics and Computing*, **22**:877–895.
- KNIGHT, M. I., NUNES, M. A., and NASON, G. P. (2016) ‘Modelling, Detrending and Decorrelation of Network Time Series.’ *ArXiv e-prints*.
- KOLACYZK, E. (2017) *Topics at the Frontier of Statistics and Network Analysis*. Cambridge: Cambridge University Press.
- KOVAC, A. and SILVERMAN, B. W. (2000) ‘Extending the scope of wavelet regression methods by coefficient-dependent thresholding.’ *Journal of the American Statistical Association*, **95**:172–183.

- KREBS, J. T. N. (2018) ‘Non-parametric regression for spatially dependent data with wavelets.’ *Statistics*, **52**:1270–1308.
- LEEMING, K., NASON, G. P., NUNES, M., and KNIGHT, M. (2018) *GNAR: Methods for Fitting Network Time Series Models*. R package version 0.3.4.
- LEENDERS, R. (2002) ‘Modeling social influence through network autocorrelation: constructing the weight matrix.’ *Social Networks*, **24**:21–47.
- LJUNG, G. M. and BOX, G. E. P. (1978) ‘On a measure of lack of fit in time series models.’ *Biometrika*, **65**:297–303.
- LÜTKEPOHL, H. (2005) *New Introduction to Multiple Time Series Analysis*. Berlin: Springer.
- MALLAT, S. (1989a) ‘Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$.’ *Transactions of the American Mathematical Society*, **315**:69–87.
- MALLAT, S. (1989b) ‘A theory for multiresolution signal decomposition: the wavelet representation.’ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**:674–693.
- MATHAI, A. M. and PROVOST, S. B. (1992) *Quadratic forms in random variables: theory and applications*. New York: M. Dekker.
- MEYER, Y. (1993) *Wavelets: algorithms and applications*. Philadelphia: SIAM.
- MUR, J. (1999) ‘Testing for spatial autocorrelation: moving average versus autoregressive processes.’ *Environment and Planning A*, **31**:1371–1382.
- NAM, C. F. H., ASTON, J. A. D., ECKLEY, I. A., and KILLICK, R. (2015) ‘The uncertainty of storm season changes: quantifying the uncertainty of autocovariance change-points.’ *Technometrics*, **57**:194–206.
- NASON, G. P. (2008) *Wavelet Methods in Statistics with R*. New York: Springer.
- NASON, G. P. (2013) ‘A test for second-order stationarity and approximate confidence intervals for localized autocovariances for locally stationary time series.’ *Journal of the Royal Statistical Society: Series B*, **75**:271–292.

- NASON, G. P. (2016a) *locits: Tests of stationarity and localized autocovariance*. R package version 1.7.1.
- NASON, G. P. (2016b) *wavethresh: Wavelets Statistics and Transforms*. R package version 4.6.8.
- NASON, G. P., NUNES, M., FRYZLEWICZ, P., and ECKLEY, I. (2012) *AutoSpec: Automatic stationary and locally stationary spectral estimation*. R package version 1.2.
- NASON, G. P. and SAVCHEV, D. (2014) ‘White noise testing using wavelets.’ *Stat*, **3**:351–362.
- NASON, G. P. and SILVERMAN, B. W. (1995) ‘The stationary wavelet transform and some applications.’ In *Wavelets in Statistics*, ANTONIADIS, A. and OPPENHEIM, G., eds., volume 103 of *Lecture Notes in Statistics*. Berlin: Springer.
- NASON, G. P., VON SACHS, R., and KROISANDT, G. (2000) ‘Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum.’ *Journal of the Royal Statistical Society: Series B*, **62**:271–292.
- NEWLAND, D. E. (2000) ‘Harmonic wavelets in vibration and acoustics.’ In *Wavelets: The Key to Intermittent Information?*, SILVERMAN, B. W. and VASSILICOS, J. C., eds. Oxford: Oxford University Press.
- NUNES, M. and NASON, G. P. (2005) ‘Stopping time in adaptive lifting.’ *Technical Report 05:15*, Statistics Group, Department of Mathematics, University of Bristol.
- OMBAO, H., RAZ, J., VON SACHS, R., and GUO, W. (2002) ‘The SLEX model of a non-stationary random process.’ *Annals of the Institute of Statistical Mathematics*, **54**:171–200.
- ORD, K. (1975) ‘Estimation methods for models of spatial interaction.’ *Journal of the American Statistical Association*, **70**:120–126.
- PAPARODITIS, E. (2009) ‘Testing temporal constancy of the spectral structure of a time series.’ *Bernoulli*, **15**:1190–1221.
- PAQUET, A. H., WARD, R. K., and PITAS, I. (2003) ‘Wavelet packets-based digital watermarking for image verification and authentication.’ *Signal Processing*, **83**:2117–2132.

- PARK, T., ECKLEY, I. A., and OMBAO, H. C. (2014) ‘Estimating time-evolving partial coherence between signals via multivariate locally stationary wavelet processes.’ *IEEE Transactions on Signal Processing*, **62**:5240–5250.
- PARK, T., ECKLEY, I. A., and OMBAO, H. C. (2018) ‘Dynamic classification using multivariate locally stationary wavelet processes.’ *Signal Processing*, **152**:118–129.
- PERCIVAL, D. and WALDEN, A. (2000) *Wavelet Methods for Time Series Analysis*. Cambridge: Cambridge University Press.
- PERCIVAL, D. B. and CONSTANTINE, W. L. B. (2006) ‘Exact simulation of Gaussian time series from nonparametric spectral estimates with application to bootstrapping.’ *Statistics and Computing*, **16**:25–35.
- PFAFF, B. (2008) ‘VAR, SVAR and SVEC models: Implementation within R package vars.’ *Journal of Statistical Software*, **27**:1–32.
- PRADHAN, B., SANDEEP, K., MANSOR, S., RAMLI, A. R., and SHARIF, A. R. B. M. (2007) ‘Second generation wavelets based GIS terrain data compression using Delaunay triangulation.’ *Engineering Computations*, **24**:200–213.
- PRIESTLEY, M. and SUBBA RAO, T. (1969) ‘A test for non-stationarity of time-series.’ *Journal of the Royal Statistical Society: Series B*, **31**:140–149.
- PRIESTLEY, M. B. (1981) *Spectral analysis and time series*. London: Elsevier Academic Press.
- R CORE TEAM (2018) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- RAMSEY, J. B. (2000) ‘The contribution of wavelets to the analysis of economic and financial data.’ In *Wavelets: The Key to Intermittent Information?*, SILVERMAN, B. W. and VASSILICOS, J. C., eds. Oxford: Oxford University Press.
- SALTER-TOWNSHEND, M., WHITE, A., GOLLINI, I., and MURPHY, T. (2012) ‘Review of statistical network analysis: Models, algorithms, and software.’ *Statistical Analysis and Data Mining*, **5**:243–264.

- SCHWARZ, G. (1978) ‘Estimating the dimension of a model.’ *The Annals of Statistics*, **6**:461–464.
- SHAO, X. (2011) ‘A bootstrap-assisted spectral test of white noise under unknown dependence.’ *Journal of Econometrics*, **162**:213–224.
- SHUMWAY, R. and STOFFER, D. (2000) *Time Series Analysis and Its Applications*. New York: Springer.
- SPENCER, S. E. F., HILL, S. M., and MUKHERJEE, S. (2015) ‘Inferring network structure from interventional time-course experiments.’ *The Annals of Statistics*, **9**:507–524.
- SWELDENS, W. (1998) ‘The lifting scheme: A construction of second generation wavelets.’ *SIAM Journal of Mathematical Analysis*, **29**:511–546.
- TINATI, M. A. and MOZAFFARY, B. (2006) ‘A wavelet packets approach to electrocardiograph baseline drift cancellation.’ *International Journal of Biomedical Imaging*, **2006**:1–9.
- TODA, H. Y. and MCKENZIE, C. R. (1999) ‘LM tests for unit roots in the presence of missing observations.’ *Mathematics and Computers in Simulation*, **48**:457–468.
- TSAY, R. S. (2014) *Multivariate Time Series Analysis*. Hoboken: Wiley.
- VAN BELLEGEM, S. and VON SACHS, R. (2008) ‘Locally adaptive estimation of evolutionary wavelet spectra.’ *The Annals of Statistics*, **36**:1879–1924.
- VARGA, R. S. (1962) *Matrix Iterative Analysis*. New Jersey: Prentice-Hall.
- VASILYEV, O. V. and BOWMAN, C. (2000) ‘Second-generation wavelet collocation method for the solution of partial differential equations.’ *Journal of Computational Physics*, **165**:660–693.
- VIDAKOVIC, B. (2009) *Statistical Modeling by Wavelets*. New York: Wiley.
- VON SACHS, R. and NEUMANN, M. H. (2000) ‘A wavelet-based test for stationarity.’ *Journal of Time Series Analysis*, **21**:597–613.
- WICKERHAUSER, M. V. (1994) *Adapted Wavelet Analysis from Theory to Software*. Massachusetts: A. K. Peters.

- XIE, Y., YU, J., and RANNEY, B. (2009) ‘Forecasting using locally stationary wavelet processes.’ *Journal of Statistical Computation and Simulation*, **79**:1067–1082.
- ZGHEIB, R., FLEURY, G., and LAHALLE, E. (2006) ‘New fast algorithm for simultaneous identification and optimal reconstruction of non stationary AR processes with missing observations.’ In *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*, 371–376.
- ZHU, X., PAN, R., LI, G., LIU, Y., and WANG, H. (2017) ‘Network vector autoregression.’ *The Annals of Statistics*, **45**:1096–1123.
- ŽIVANOVIC, G. D. and GARDNER, W. A. (1991) ‘Degrees of cyclostationarity and their application to signal detection and estimation.’ *Signal Processing*, **22**:287–297.