Multivariate Locally Stationary Wavelet Analysis with the mvLSW R Package

Simon A. C. Taylor
Lancaster University

Timothy Park
Lancaster University

Idris A. Eckley
Lancaster University

Abstract

This paper describes the R package **mvLSW**. The package contains a suite of tools for the analysis of multivariate locally stationary wavelet (LSW) time series. Key elements include: (i) the simulation of multivariate LSW time series for a given multivariate evolutionary wavelet spectrum (EWS); (ii) estimation of the time-dependent multivariate EWS for a given time series; (iii) estimation of the time-dependent coherence and partial coherence between time series channels; and, (iv) estimation of approximate confidence intervals for multivariate EWS estimates. A demonstration of the package is presented via both a simulated example and a case study with `EuStockMarkets` from the **datasets** package.

Keywords: mvLSW, multivariate evolutionary wavelet spectrum, coherence, R.

1. Introduction

Technological advances in sensors and other data recording mechanisms have led to an increased need to efficiently and accurately analyze multivariate time series. Areas of research where such methods are commonly required include economics (Rua and Nunes 2009), medicine (Cribben 2012), telecommunications (Bardwell, Eckley, Fearnhead, Smith, and Spott 2019) and environmental science (Shama 2007). Historically, one might approach such a multivariate time series challenge by making the assumption that the underlying process was second order stationary. However, there are an increasing number of cases where such global stationarity assumptions are not tenable due to the second order structure of the process changing over time. Such series are said to be non-stationary.

Over the years, a number of popular approaches for modeling non-stationary time series have been proposed, though activity has predominantly focused on the univariate time series setting. Notable contributions in the univariate setting include the seminal work of Priestley

on evolutionary processes (Priestley 1965), time varying moving average methods (Hallin 1986) and locally stationary processes (Dahlhaus 1997; Nason, von Sachs, and Kroisandt 2000). For a recent review of the literature in this area, we refer readers to the excellent review article by Dahlhaus (2012). Implementations of these models in R (R Core Team 2019) include the **LSTS** (Olea, Palma, and Rubio 2015) and **wavethresh** (Nason 2008, 2016). These respectively implement the locally stationary Fourier and wavelet approaches for univariate time series. A multidimensional implementation of a locally stationary framework is also available for regular lattice processes (see Eckley, Nason, and Treloar 2010; Eckley and Nason 2011, for details).

We are, of course, by no means the first to consider the challenge of modeling non-stationary multivariate time series. Early contributions include those by Ombao, von Sachs, and Guo (2005) who extend the smoothed localized exponential (SLEX) model to the multivariate setting and Sanderson, Fryzlewicz, and Jones (2010) who apply the wavelet methodology to bivariate and multivariate time series respectively. The SLEX framework, originally proposed by Ombao, Raz, von Sachs, and Guo (2002), seeks to segment a non-stationary time series into dyadic stationary blocks that then permit the employment of standard methods to form a time-dependent Fourier analysis. The multivariate extension of SLEX, proposed by Ombao *et al.* (2005), enables the estimation of the cross-spectrum and investigations into the coherence structure. Conversely, the work of Sanderson *et al.* (2010) and Cho and Fryzlewicz (2015) decompose the dependence within a locally stationary time series into two multiscale components; the within-channel spectral structure and the cross-spectrum (between-channel) structure for the bivariate and p-variate time series cases respectively. However neither of these recent wavelet-based contributions directly address a key modeling challenge for truly multivariate non-stationary signals, namely the identification of whether whether the connection between two channels is either direct or indirect, i.e., driven by other observed channel(s).

This article presents the R implementation of an alternative formulation for multivariate locally stationary wavelet (LSW) model, proposed by Park, Eckley, and Ombao (2014). The work extends the locally stationary wavelet framework of Nason *et al.* (2000) to a multivariate setting, also enabling the estimation of the within-channel spectral structure and cross-spectrum. Park *et al.* (2014) also introduce the concepts of local coherence and, crucially, local partial coherence within the multivariate locally stationary wavelet setting. The **mvLSW** package (Taylor, Park, Eckley, and Killick 2019) implements the work of Park *et al.* (2014), building upon the univariate locally stationary wavelet time series implementation in **wavethresh** (Nason 2008, 2016). Specifically, **mvLSW** provides functionality for: the simulation of multivariate LSW time series for a given evolutionary wavelet spectrum (EWS); estimation of the multivariate EWS for a given time series with point-wise confidence intervals; and, estimation of the local coherence and local partial coherence between time series channels. The **mvLSW** package is available for download from the comprehensive R archive network (CRAN). Note that throughout this article all **mvLSW** package references relate to version 1.2.3 of the package, as available on CRAN.

The paper is structured as follows. The framework of multivariate LSW modeling is briefly introduced in Section 2 together with the introduction of the **mvLSW** package for defining a multivariate EWS and simulating a multivariate LSW time series. Section 3 describes estimation of the multivariate EWS and its implementation in R. The concept and estimation of localized coherence and partial coherence is discussed in Section 4. We then introduce the functions developed to implement this coherence estimation framework, prior to summarizing

the mechanism by which approximate confidence intervals can be constructed in Section 5. A complete demonstration of the **mvLSW** package is presented in Section 6 including a case study which makes use of the European financial index time series `EuStockMarkets` that is accessible from the **datasets** package.

2. Framework

This section briefly introduces the multivariate locally stationary wavelet (LSW) time series and introduces its implementation within R in the **mvLSW** package. Section 2.1 introduces the multivariate LSW framework of Park *et al.* (2014) and a description of the time-frequency power decomposition by the multivariate evolutionary wavelet spectrum (EWS). Section 2.2 presents a worked example for a trivariate LSW process and demonstrates how to simulate a time series with the specified spectral form.

2.1. The multivariate LSW process

Adopting the notation of Park *et al.* (2014), the doubly indexed P-variate stochastic process $\{\mathbf{X}_{t;T}\}$ for $t = 1, \ldots, T$ with dyadic length, i.e., $T = 2^J$ for some $J \in \mathbb{N}$, is said to be a multivariate LSW process if it is represented by:

$$\mathbf{X}_{t;T} = \sum_{j=1}^{\infty} \sum_{k} \mathbf{V}_j(k/T)\psi_{j,k}(t)\mathbf{z}_{j,k}. \tag{1}$$

As in the earlier univariate LSW work of Nason *et al.* (2000), the $\{\psi_{j,k}\}$ denote the set of discrete non-decimated wavelets for each level j and location k pair such as the Daubechies compactly supported wavelet (Daubechies 1990). The random vectors $\{\mathbf{z}_{j,k} = (z_{j,k}^{(1)}, \ldots, z_{j,k}^{(P)})^\top\}$ are uncorrelated innovations with zero expectation and $P \times P$ identity variance-covariance matrix. Finally, $\mathbf{V}_j(u)$ denotes the lower-triangular $P \times P$ transfer function matrix for the given level index at rescaled time $u := t/T \in (0,1)$. Some smoothness conditions are assumed for the elements of the transfer function matrix, controlling its behavior (see Park *et al.* 2014, for details). For notational ease, the explicit dependence on T shall henceforth be suppressed but its dependence on the process and derived estimates shall naturally be assumed.

Within the above modeling framework, arguably the key element of interest is the transfer function matrix. Since the innovations are orthogonal with unit variance and the wavelets are not channel specific, then all forms of dependency between and within channels must be encapsulated by the transfer function matrix. For instance, the diagonal element $V_j(u)^{(p,p)}$ for channel p, for $p = 1, \ldots, P$, defines the time-varying amplitudes of the respective wavelet at level j akin to the univariate scenario of Definition 1 in Nason *et al.* (2000). Furthermore, if the off-diagonal element $V_j(u)^{(p,q)}$, for $p > q$, is non-zero then there is a time-varying dependence between channels p and q, but if this element is zero for all time and levels then the two channels are uncorrelated.

The transfer function matrix is also useful in forming the spectral representation of a multivariate LSW time series. Specifically, the power contained within a multivariate LSW process is described by the multivariate evolutionary wavelet spectrum (EWS) taking the form:

$$\mathbf{S}_j(u) = \mathbf{V}_j^\top(u)\mathbf{V}_j(u), \tag{2}$$

where $\mathbf{V}_j^\top(u)$ denotes the transpose of $\mathbf{V}_j(u)$. The diagonal element $S_j^{(p,p)}(u)$ denotes the auto-spectrum for channel p, whilst the off-diagonal element $S_j^{(p,q)}(u)$, for $p \neq q$, denotes the cross-spectrum between the channel pair p and q, and it describes how the power in the process is shared between channels.

2.2. Trivariate worked example

The multivariate EWS defined in Equation 2 is a concise description of the time-varying distribution of power for a multivariate LSW time series. To demonstrate this, we introduce a trivariate LSW process which will be used as a worked example throughout the paper. We assume that the process is built using discrete Haar wavelets, and that the EWS for this process has non-zero power at only the second finest level, $j = 2$, given by:

$$\mathbf{S}_2(u) \;\; = \;\; \begin{bmatrix} 4 + 16u & 2 + 8u & 2 + 8u \\ 2 + 8u & 6 & 1 + 4u \\ 2 + 8u & 1 + 4u & 20 - 14u \end{bmatrix}, \quad \text{for } u \in (0, 1). \tag{3}$$

Note that the auto-spectrum for the second channel is constant and so this channel is marginally stationary. The auto-spectrum for the first channel increases over the unit time interval and so, with the spectrum at all other levels equal to zero, the variability of this channel steadily increases. Conversely, the auto-spectrum for the third channel decreases over time. Dependence between the channels is also time-varying within this example.

The following extract defines the trivariate EWS in R using the **mvLSW** package for a time series of length $T = 1024$ (i.e., $J = 10$):

```
R> library("mvLSW")
R> P <- 3          ## Number of channels
R> T <- 1024       ## Time series length
R> J <- log2(T)    ## Number of levels
R> Spec <- array(0, dim = c(P, P, J, T))
R> Spec[1, 1, 2, ] <- seq(from = 4, to = 20, length = T)
R> Spec[2, 2, 2, ] <- 6
R> Spec[3, 3, 2, ] <- seq(from = 20, to = 6, length = T)
R> Spec[1, 2, 2, ] <- Spec[2, 1, 2, ] <- seq(from = 2, to = 10, len = T)
R> Spec[1, 3, 2, ] <- Spec[3, 1, 2, ] <- seq(from = 2, to = 10, len = T)
R> Spec[2, 3, 2, ] <- Spec[3, 2, 2, ] <- seq(from = 1, to = 5, len = T)
R> True_mvEWS <- as.mvLSW(x = Spec, filter.number = 1,
+    family = "DaubExPhase", names = c("X1", "X2", "X3"))
```

The command `as.mvLSW` converts a 4D array into an object having S3 class `mvLSW`. This class is central to the **mvLSW** package and consists of two components: the multivariate EWS as a 4D numerical array and a list of sundry information about the spectrum such as its dimensions, analyzing wavelet and channel names. The input arguments for `as.mvLSW` are:

- `x`: multivariate EWS as a 4D array ($P \times P \times J \times T$).

- `filter.number`, `family`: analyzing wavelet (see Nason 2016).
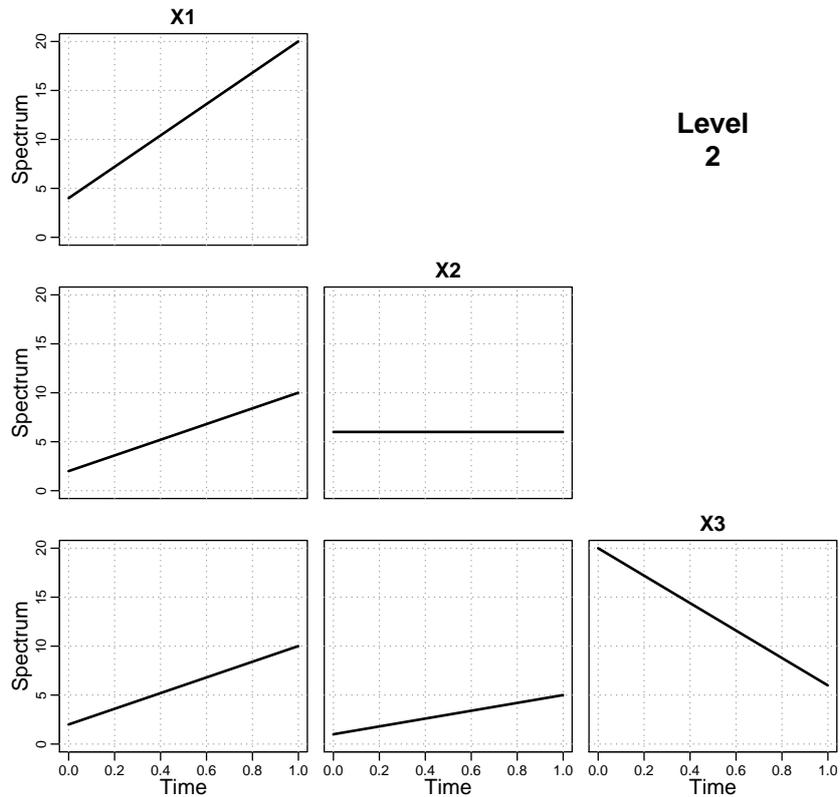
- `names`: channel names (optional).

Figure 1: True trivariate EWS at level 2 defined by Equation 3.

Due to the dimensions of a multivariate EWS, visualizations that illustrate the distribution of power over levels and locations can only be obtained via slices through this 4D object. For instance, the plot in Figure 1 displays the trivariate EWS at the second level defined in Equation 3 that is invoked by the following command:

```
R> plot(True_mvEWS, style = 2, info = 2, ylim = c(0, 20), lwd = 2)
```

The essential input arguments used here are:

- `x`: an object of `mvLSW` class.

- `style`: numerical index defining the plotting style:

  1. Single spectra plot between channel pair (p, q) and at level j is specified via `info = c(p, q, j)`.
  2. Panel plot across channels at level j is specified via `info = j`.
  3. Panel plot across levels for channel pair (p, q) is specified via `info = c(p, q)`,
  4. As `style = 3`, but presented as an image plot.

Additional arguments such as `ylim` and `lwd` can be supplied to customize the generated plot. Point-wise approximate confidence intervals introduced in Section 5 can be drawn by supplying a list containing the upper and lower bounds via the optional `Interval` argument.

Furthermore, the inclusion of the logical argument `diag`, when `style = 2`, suppresses the drawing of the diagonal plots on the panel. This is particularly useful for plotting the local coherence and partial coherence estimates discussed in Section 4.

The transfer function matrix is obtained by factorizing the multivariate EWS matrix using Cholesky decomposition. A realization of a multivariate LSW process with a pre-specified EWS is therefore simulated using Equation 1 with innovations sampled independently from some defined distribution that has zero mean and unit variance. This procedure is implemented by the `rmvLSW` command. For example, the following extract generates a realization of a trivariate LSW time series, with EWS structure as defined by Equation 3 with Gaussian innovations:

```
R> set.seed(1)
R> X <- rmvLSW(Spectrum = True_mvEWS, noiseFN = rnorm)
R> plot(x = X, main = "Gaussian mvLSW time series")
```

The input arguments for `rmvLSW` are:

- `Spectrum`: multivariate EWS as an `mvLSW` object.

- `noiseFN`: function for generating the orthogonal innovation process with zero expectation and unit variance. Additional arguments provided to `rmvLSW` are passes to the supplied function.

The `S3` `simulate` method can alternatively be called to simulate a realization of a multivariate LSW time series for a provided multivariate EWS. An example of simulated trivariate LSW time series is presented in Figure 2.

3. Multivariate EWS estimation

The multivariate LSW framework presented in the previous section enables the generation of a time series that possess spectral properties defined by a specified multivariate EWS. Conversely, interest often lies in understanding the spectral structure for a given multivariate LSW time series. Such a scheme was proposed by Park *et al.* (2014), estimating the multivariate EWS, $\{\hat{\mathbf{S}}_{j,k}\}$ for levels $j = 1, \ldots, J$ and locations $k = 0, \ldots, T - 1$. To begin, each channel of the time series is independently transformed based on a given wavelet to determine the empirical wavelet coefficient vector, $\mathbf{d}_{j,k} = [d_{j,k}^{(1)}, \ldots, d_{j,k}^{(P)}]^\top$, with elements:

$$d_{j,k}^{(p)} = \sum_{t=1}^{T} X_t^{(p)} \psi_{j,k}(t). \tag{4}$$

The wavelet periodogram is then defined by the matrix set for each level and location pair as:

$$\mathbf{I}_{j,k} = \mathbf{d}_{j,k}\mathbf{d}_{j,k}^\top. \tag{5}$$

Mirroring a result of Nason *et al.* (2000) for univariate time series, Park *et al.* (2014) established that Equation 5 is both biased and has non-vanishing estimator variance. These
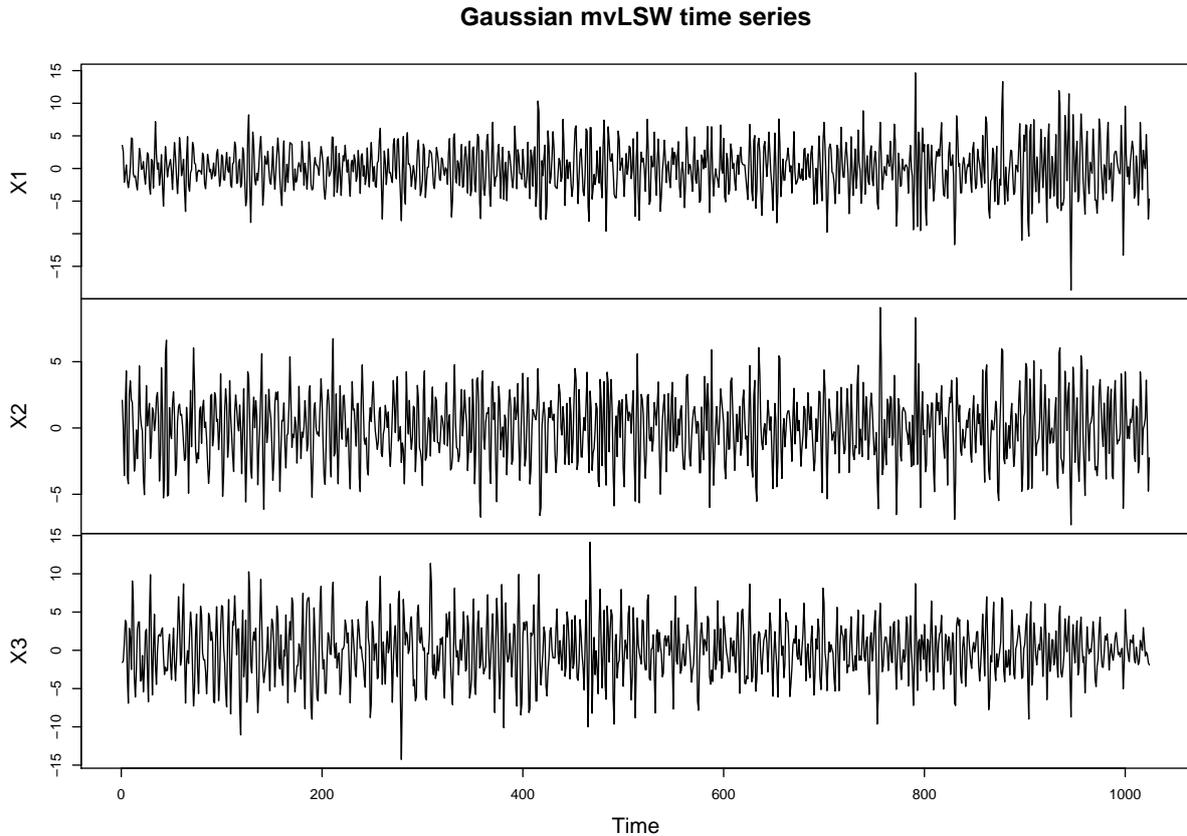
Gaussian mvLSW time series

Figure 2: A realization of a (Gaussian) trivariate locally stationary wavelet process having EWS structure as defined in Equation 3.

issues are overcome by smoothing and correcting the raw periodogram estimate in defining the multivariate EWS estimator:

$$\hat{\mathbf{S}}_{j,k} = \sum_{l=1}^{J} \sum_{m=-M}^{M} (\mathbf{A}^{-1})_{j,l} \, w_M(m) \mathbf{I}_{l,k+m}. \tag{6}$$

Here, $w_M(m)$ denotes the symmetric kernel function on the compact support $m \in [-M, M]$ (Ombao, Raz, Strawderman, and von Sachs 2001) and $(\mathbf{A}^{-1})_{j,l}$ is the (j, l)-element from the inverted autocorrelation wavelet inner product matrix (Eckley and Nason 2005).

Perhaps unsurprisingly, the bias correction in Equation 6 means that the spectral matrix estimates $\{\hat{\mathbf{S}}_{j,k}\}$ are no longer guaranteed to be positive definite. This leads to difficulties in evaluating quantities that are derived from the multivariate EWS, such as the localized coherence and partial coherence which we introduce in Section 4. The matrices of the multivariate EWS estimate must therefore be regularized using, for example, the approach of Schnabel and Eskow (1999) to ensure that subsequently derived estimates are themselves valid.

3.1. Trivariate worked example continued

Estimation of the multivariate EWS is implemented using the `mvEWS` function. The following code extract estimates the trivariate EWS for the time series simulated in Section 2.2. The

wavelet transform is performed using the Haar wavelet and smoothing of the periodogram is performed using the rectangular kernel $w_M(m) = (2M + 1)^{-1}$ for $m \in [-M, M]$ with parameter $M = \sqrt{T} = 32$.

```
R> EWS <- mvEWS(X = X, filter.number = 1, family = "DaubExPhase",
+     kernel.name = "daniell", kernel.param = sqrt(T), bias.correct = TRUE,
+     tol = 1e-10)
```

The arguments of `mvEWS` are:

- `X`: regular *P*-variate time series with dyadic length of class `matrix` or `ts`, `xts` (Ryan and Ulrich 2018) or `zoo` (Zeileis and Grothendieck 2005).

- `filter.number`, `family`: wavelet, passed to `wd` from **wavethresh**.

- `kernel.name`, `kernel.param`: defines the smoothing kernel that is evaluated by the base command `kernel`. The argument `kernel.name` is a character string that names any kernel type evaluated by `kernel`.

- `bias.correct`: logical variable, indicating whether the biased or corrected estimator should be estimated.

- `tol`: threshold applied in matrix regularization.

Estimation of the trivariate EWS involves a number of steps including wavelet transformation, smoothing, bias correction and regularization. The function returns an object of class `mvLSW`. Details about each step are stored within this object, under the information list. A summary of these details can be examined by invoking the `print` or `summary` command.

```
R> summary(EWS)

== Dimensions ==
P : 3
J : 10
T : 1024

== Wavelet Transform ==
Family        : DaubExPhase
Filter Number : 1

== Smoothing ==
Type   : all
Method : Daniell(32) - GCV criterion = 38.68027

== Sundries ==
Applied Bias Correction : TRUE
Minimum Eigenvalue      : 1e-10
```

The printed summary states the dimension of the object, the wavelet and smoothing method used in deriving the estimate. The item `Type` under the smoothing heading identifies that the same kernel function is applied to all levels. Smoothing can be performed on a by-level basis. For further details, the reader is referred to the package's manual. A measure of the smoothing performance is calculated using a generalized cross validation (GCV) criterion (Ombao *et al.* 2001). The sundries information states whether or not the bias correction has been applied and the minimum eigenvalue across all time and level pairs to confirm whether all estimated spectral matrices are positive definite.

We should note that the package requires a time series of dyadic length. In practice, not all series will be of length 2^J for some $J \in \mathbb{N}$. As Nason and Silverman (1994) describe, various techniques can be used to overcome this challenge. These include the use of reflective or periodic boundary conditions; zero padding and truncating either the head or tail of the data. Naturally, there can be various advantages and disadvantages associated with each of these approaches, depending on the underlying data generation process (Percival and Walden 2000, Chapter 4.11).

4. Localized coherence and partial coherence

Dependence within a multivariate time series can occur both within and between channels. Coherence provides a measure of the linear dependence between any channel pair. However, if the multivariate time series is non-stationary then the coherence measure may vary over time. Park *et al.* (2014) introduced the concept of local coherence based on the locally stationary wavelet framework defined in Section 2.1. At a particular level, the local coherence matrix function is defined by:

$$\boldsymbol{\rho}_j(u) \quad = \quad \mathbf{D}_j(u)\mathbf{S}_j(u)\mathbf{D}_j(u), \tag{7}$$

where $\mathbf{D}_j(u) = \text{diag}\{[S_j^{(p,p)}(u)]^{-1/2} : p = 1, \ldots, P\}$. The off-diagonal elements of the localized coherence matrices may take any value within $[-1, 1]$ where strong linear dependence is identified by values near ± 1. Whilst this coherence measure provides an indication of strong dependence between a channel pair, it cannot distinguish whether this relationship is direct or indirect. In other words, it cannot identify whether the relationships occur because of direct dependencies with other (observed) channels. To quantify direct linear dependence, Park *et al.* (2014) introduced the local partial coherence matrix function. For any given level, this is defined as follows:

$$\Gamma_j(u) \quad = \quad -\mathbf{H}_j(u)\mathbf{G}_j(u)\mathbf{H}_j(u), \tag{8}$$

where $\mathbf{G}_j = \mathbf{S}_j(u)^{-1}$ and $\mathbf{H}_j(u) = \text{diag}\{[G_j^{(p,p)}(u)]^{-1/2} : p = 1, \ldots, P\}$.

4.1. Trivariate worked example continued

Within the **mvLSW** package both forms of coherence can be estimated using the `coherence` command. This implements a "plug-in" estimator proposed by Park *et al.* (2014). The input arguments for `coherence` are:
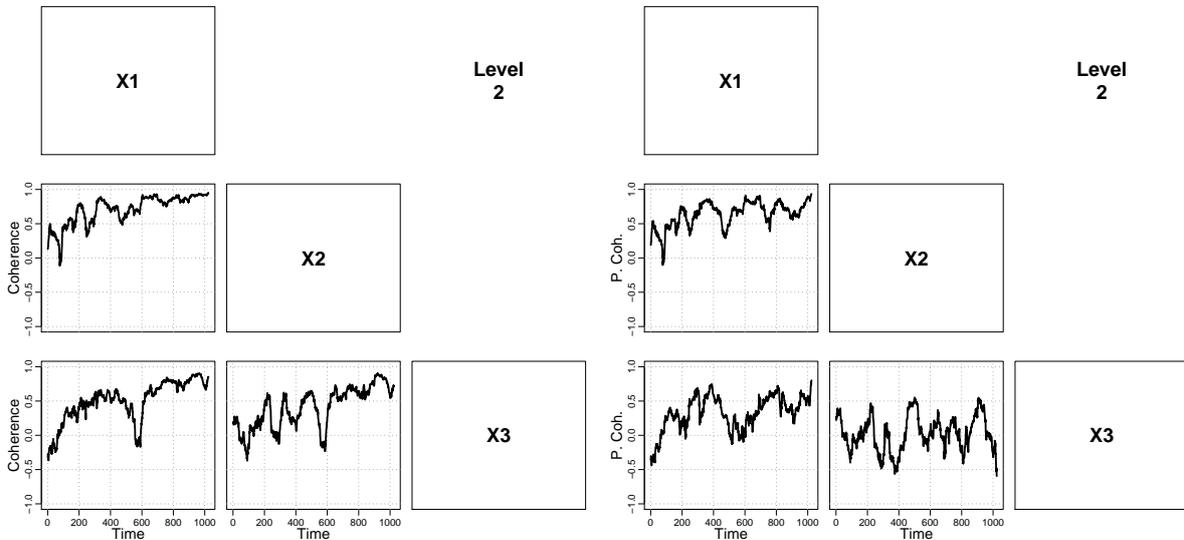
Figure 3: Localized coherence (left) and partial coherence (right) estimates at level 2 for the simulated trivariate time series.

- `object`: multivariate EWS as an `mvLSW` object.

- `partial`: logical variable, indicating whether to evaluate the partial coherence.

Here, the distinction between evaluating the localized coherence and localized partial coherence is simply achieved by toggling the logical argument as demonstrated below with the trivariate EWS estimate derived in Section 3.1:

```
R> RHO <- coherence(object = EWS, partial = FALSE)
R> GAMMA <- coherence(object = EWS, partial = TRUE)
```

As with the multivariate EWS, the returned coherence measure is stored within a 4D array of class `mvLSW`. A visualization of a slice through this object can be generated by invoking the `plot` command as described in Section 2.2 for appropriate `style` and `info` arguments. Note that the diagonal elements of both coherence matrices do not contain any useful information and so can be suppressed when generating the image by toggling the `diag` logical argument as demonstrated in the following extract for the trivariate example:

```
R> plot(x = RHO, style = 2, info = 2, diag = FALSE, ylim = c(-1, 1),
+    lwd = 2, ylab = "Coherence")
R> plot(x = GAMMA, style = 2, info = 2, diag = FALSE, ylim = c(-1, 1),
+    lwd = 2, ylab = "P. Coh.")
```

The localized coherence and partial coherence estimates at the second level are presented in Figure 3. Recalling the form of the EWS in Equation 3, it is perhaps unsurprising that the estimates of the localized coherence at the second level are mostly positive and gradually increase over time. However, the localized partial coherence estimate between the second and third channels is closer to zero compared to its localized coherence estimate, suggesting that

the measured linear dependence can mostly be explained by their dependence with the first channel.

5. Confidence intervals for the multivariate EWS estimate

Under reasonably mild assumptions, Park (2014) established that the approximate, point-wise $100(1 - \alpha)\%$ confidence interval for a given element of the multivariate EWS, $S_{j,k}^{(p,q)}$, is given by:

$$
\left\{ \hat{S}_{j,k}^{(p,q)} - \Phi\left(1 - \frac{\alpha}{2}\right) \sqrt{\text{Var}\left[\hat{S}_{j,k}^{(p,q)}\right]}, \quad \hat{S}_{j,k}^{(p,q)} + \Phi\left(1 - \frac{\alpha}{2}\right) \sqrt{\text{Var}\left[\hat{S}_{j,k}^{(p,q)}\right]} \right\}, \tag{9}
$$

where $\Phi(x)$ denotes the standard Gaussian cumulative distribution function. The asymptotic variance of an element of the multivariate EWS estimator is easily shown to be:

$$
\text{Var}\left[\hat{S}_{j,k}^{(p,q)}\right] = \sum_{l_1,l_2=1}^{J} \sum_{m_1,m_2=k-M}^{k+M} (A^{-1})_{j,l_1} (A^{-1})_{j,l_2} \, w_M(m_1) w_M(m_2) \, \text{Cov}\left[I_{l_1,m_1}^{(p,q)}, I_{l_2,m_2}^{(p,q)}\right], \tag{10}
$$

where the covariance between any pair of wavelet periodogram elements is:

$$
\text{Cov}\left[I_{j,k}^{(p,q)}, I_{l,m}^{(p,q)}\right] = \left[\sum_{h=1}^{J} B_{j,l,h}(m-k) S_h^{(p,q)}\left(\frac{k+m}{2T}\right)\right]^2
$$

$$
+ \prod_{r=p,q} \sum_{h=1}^{J} B_{j,l,h}(m-k) S_h^{(r,r)}\left(\frac{k+m}{2T}\right) + \mathcal{O}\left(T^{-1}\right). \tag{11}
$$

Here, $B_{j,l,h}(\lambda) = \sum_{\tau} \Psi_{j,h}(\tau) \Psi_{l,h}(\tau - \lambda)$ is the inner product of cross-level autocorrelation wavelet inner product function where $\Psi_{j,l}(\tau) = \sum_k \psi_{j,k}(0) \psi_{l,k+\tau}(0)$ (Fryzlewicz and Nason 2006). Note that the special case $\{A_{j,l} = B_{j,j,l}(0)\}$ for levels $j, l = 1, \ldots, J$ denotes the inner product matrix that quantifies the leakage bias of the raw periodogram, see Section 3. The variance is dependent on knowing the true EWS but this is typically unavailable. Consequently, the variance is estimated by substituting the multivariate EWS estimate $\hat{S}_{j,k}$.

5.1. Trivariate worked example continued

Before calculating the asymptotic variance, the autocorrelation wavelet inner product function $B_{j,l,h}(\lambda)$ must first be calculated. The command `AutoCorrIP` evaluates the inner product functions for a given wavelet such as the Haar wavelet demonstrated below:

```
R> HaarACWIP <- AutoCorrIP(J = J, filter.number = 1, family = "DaubExPhase")
```

The `AutoCorrIP` command returns a $(2T + 1) \times J \times J \times J$ array that corresponds to the lag λ and level indices j, l and h respectively.

As with the localized coherence and partial coherence estimates, the variance is evaluated as a "plug-in" estimator based on a given multivariate EWS estimate. This is implemented by `varEWS` as demonstrated below for the trivariate EWS estimate evaluated in Section 3.1:

```
R> VAR <- varEWS(object = EWS, ACWIP = HaarACWIP)
```
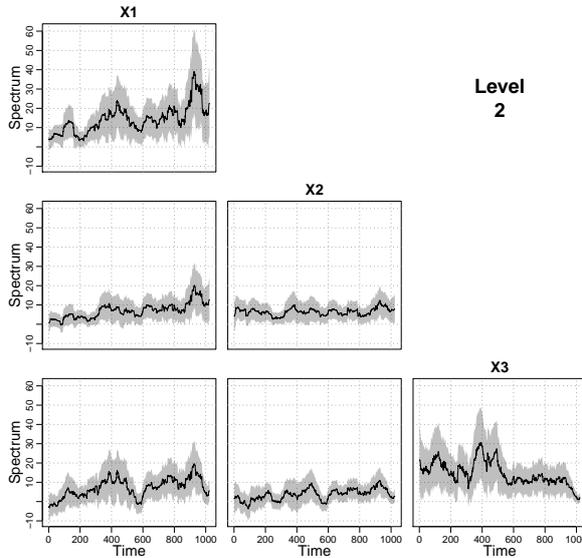
Figure 4: Panel plot of the trivariate EWS estimate with approximate, point-wise 95% confidence intervals at the second level for the simulated time series with Gaussian innovations.
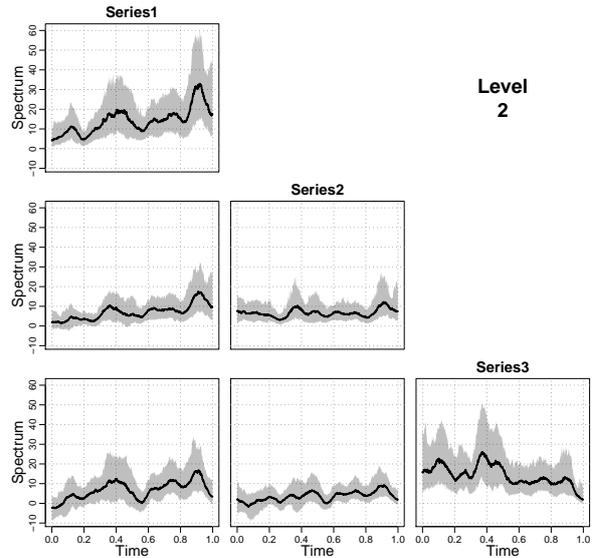
Figure 5: Panel plot of the bootstrapped median and central, point-wise 95% intervals for the trivariate EWS estimate at the second level for the simulated time series with Gaussian innovations.

The arguments of `varEWS` are:

- `object`: multivariate EWS as an `mvLSW` object.

- `ACWIP`: 4D array containing the autocorrelation wavelet inner product function.

Given the variance estimate, the approximate, point-wise 95% confidence interval for the trivariate EWS elements is derived using `ApxCI`.

```
R> CI <- ApxCI(object = EWS, var = VAR, alpha = 0.05)
```

The arguments required for the `ApxCI` function are:

- `object`: multivariate EWS as an `mvLSW` object.

- `var`: variance estimate of the multivariate EWS as an `mvLSW` object.

- `alpha`: type I error.

The returned object is a list with two items named `"L"` and `"U"`. These are objects having class `mvLSW`, that contain the point-wise lower and upper interval bounds respectively. The point-wise confidence intervals can be included on a plot of the multivariate EWS estimate using the `plot` command demonstrated in Section 2.2 with the additional `Interval` argument. The 95% confidence intervals for the trivariate EWS estimate in Figure 4 is generated by the following command:

```
R> plot(x = EWS, style = 2, info = 2, Interval = CI)
```

To demonstrate the accuracy of the 95% confidence intervals, the estimates are compared against the 95% bootstrap intervals. The following extract evaluates the bootstrap intervals from 100 simulated time series with power spectrum defined by the estimate trivariate EWS in Section 3.1. The interval estimates for the second level are presented in Figure 5. It is clear from comparing these estimates that the analytically derived intervals are close to those obtained via bootstrapping.

```
R> Bsamp <- array(NA, dim = c(P, P, J, T, 100))
R> for (b in seq_len(100)) {
R>   X_boot <- rmvLSW(Spectrum = EWS, noiseFN = rnorm)
R>   EWS_boot <- mvEWS(X = X_boot, filter.number = 1, family = "DaubExPhase",
+     kernel.name = "daniell", kernel.param = sqrt(T), bias.correct = TRUE)
R>   Bsamp[, , , , b] <- EWS_boot$spectrum
R> }
R> Bl <- as.mvLSW(x = apply(Bsamp, 1:4, quantile, prob = 0.025))
R> Bm <- as.mvLSW(x = apply(Bsamp, 1:4, quantile, prob = 0.5))
R> Bu <- as.mvLSW(x = apply(Bsamp, 1:4, quantile, prob = 0.975))
R> BInt <- list(L = Bl, U = Bu)
R> plot(Bm, style = 2, info = 2, Interval = BInt, lwd = 2)
```

6. European financial indices

We now illustrate the application of the package using `EuStockMarkets`, the European financial indices from the **datasets** package. The data set contains the daily index of the German (DAX), Swiss (SMI), French (CAC) and British (FTSE) markets. The log-returns are evaluated to remove any trend features and the series is truncated to produce a time series of length $T = 1024$. Figure 6 presents a trace plot of the time series. Each channel of the
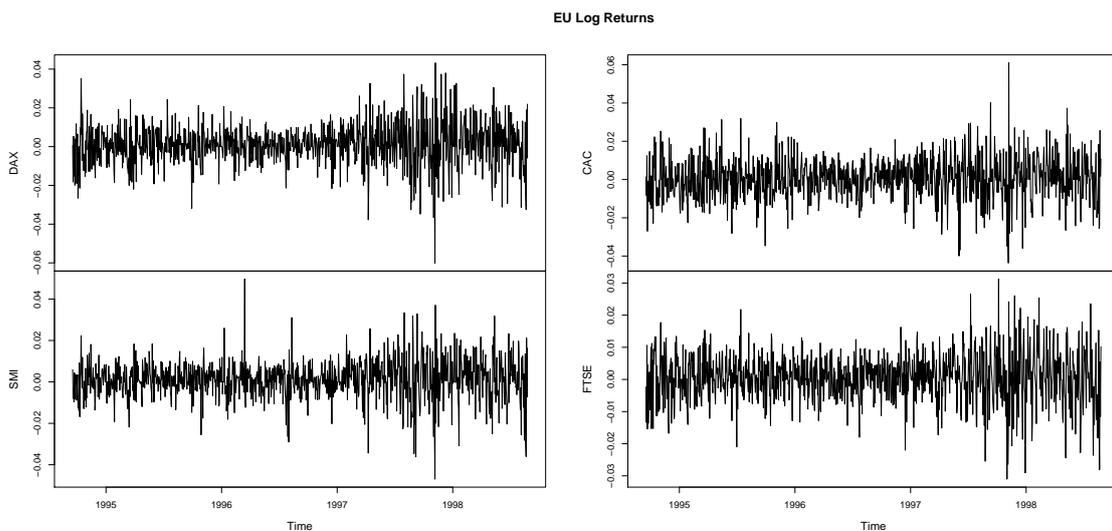

Figure 6: Log-returns time series of four European financial markets. Top-left: German DAX. Top-right: French CAC. Bottom-left: Swiss SMI. Bottom-right: British FTSE.
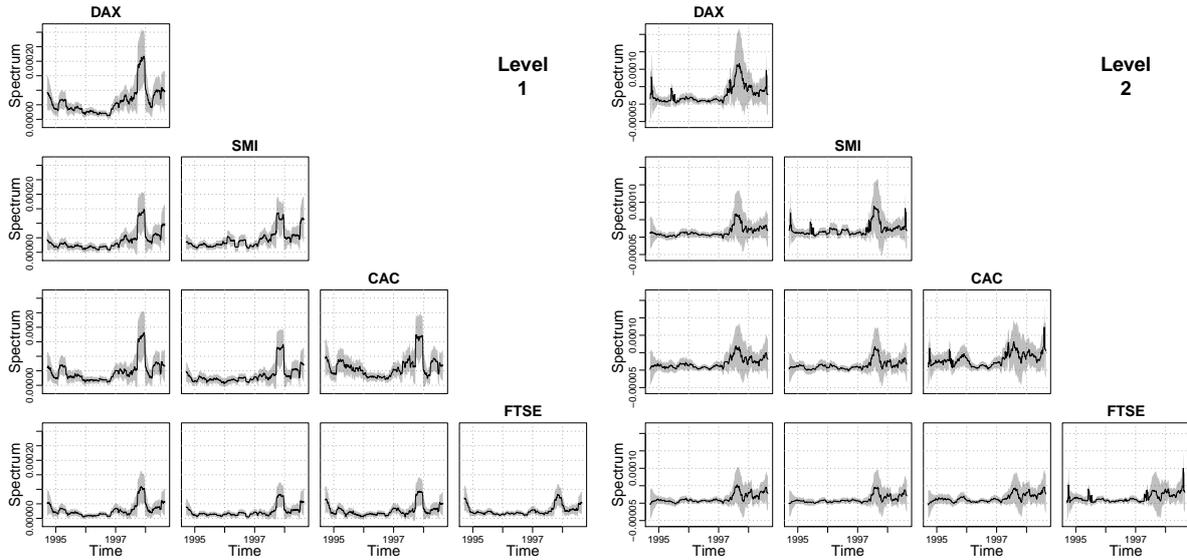
Figure 7: Panel plots at the finest (left) and second (right) levels of the multivariate EWS estimate and approximate, point-wise 95% confidence intervals for the European market log-returns.

time series is independently deemed to be non-stationary according to the test implements by `BootTOS` from the **costat** package (Cardinali and Nason 2013).

```
R> data("EuStockMarkets", package = "datasets")
R> T <- 1024
R> J <- log2(T)
R> N <- nrow(EuStockMarkets)
R> EU.lret <- diff(log(EuStockMarkets))
R> EU.lret <- window(EU.lret, start = c(1994, 186))
R> plot(x = EU.lret, main = "EU Log Returns", nc = 2)
```

Estimation of the multivariate EWS is determined using Daubechies extremal phase wavelet with seven vanishing moments, and smoothing is performed with the rectangular kernel function with span parameter $M = \sqrt{T} = 32$. The approximate (point-wise) 95% confidence intervals for the multivariate EWS estimate are evaluated along with the localized coherence and partial coherence estimates. These estimates are obtained using the following R extract which produces plots of the multivariate EWS, Figure 7, localized coherence and localized partial coherence, Figure 8; all for the finest and second levels.

```
R> EU.EWS <- mvEWS(X = EU.lret, family = "DaubExPhase", filter.number = 7,
+    kernel.name = "daniell", kernel.param = sqrt(T), bias.correct = TRUE)
R> ACWIP_FN7 <- AutoCorrIP(J = J, family = "DaubExPhase", filter.number = 7)
R> EU.VAR <- varEWS(object = EU.EWS, ACWIP = ACWIP_FN7)
R> EU.CI <- ApxCI(object = EU.EWS, var = EU.VAR, alpha = 0.05)
R> EU.R <- coherence(object = EU.EWS)
R> EU.G <- coherence(object = EU.EWS, partial = TRUE)
R> plot(x = EU.EWS, style = 2, info = 1, Interval = EU.CI)
```
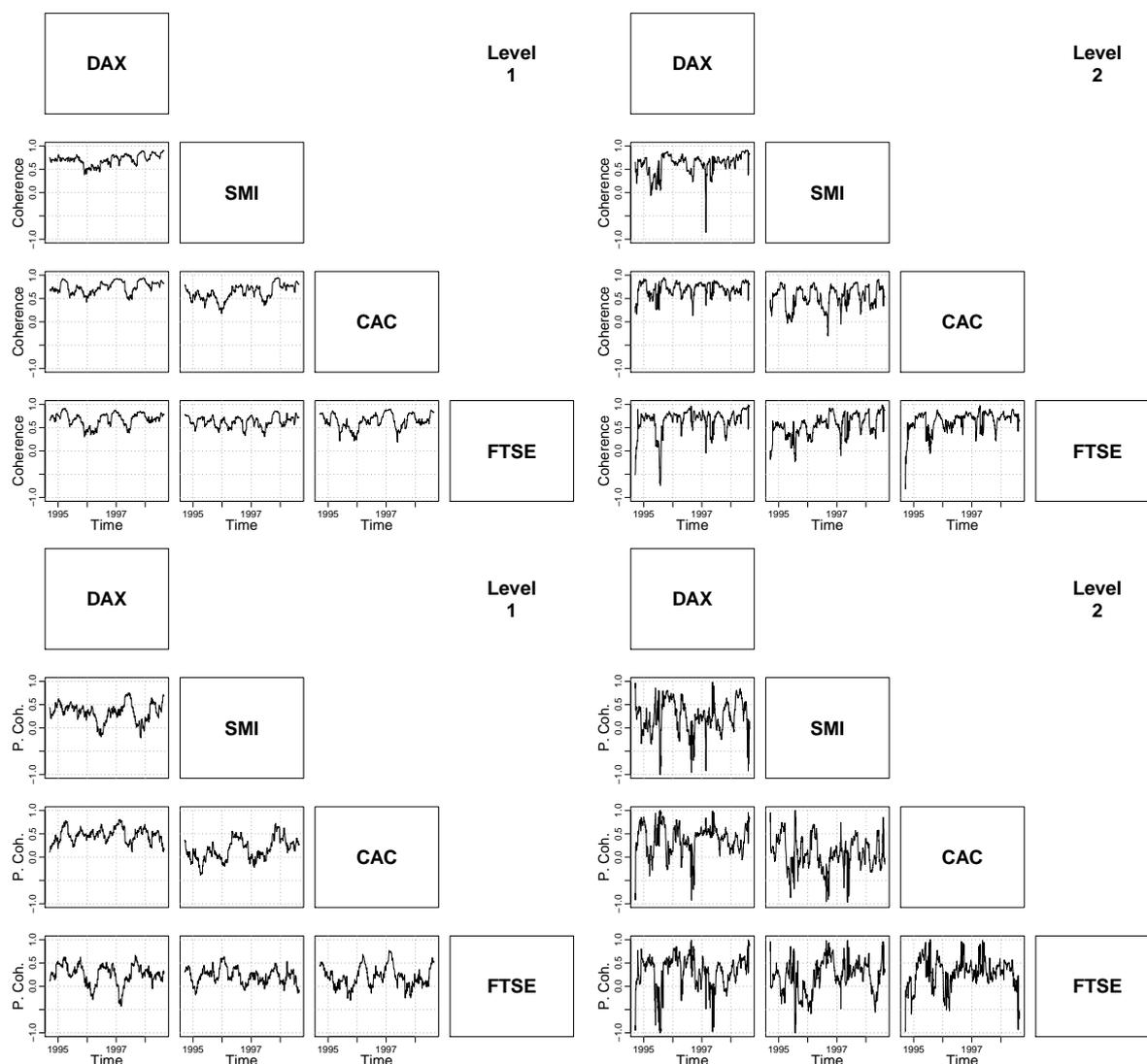
Figure 8: Panel plots at the finest (left) and second (right) levels of the localized coherence (top) and localized partial coherence (bottom) estimates for the European market log-returns.

```
R> plot(x = EU.EWS, style = 2, info = 2, Interval = EU.CI)
R> lim <- c(-1, 1)
R> lab1 <- "Coherence"
R> lab2 <- "P. Coh."
R> plot(x = EU.R, style = 2, info = 1, diag = F, ylim = lim, ylab = lab1)
R> plot(x = EU.R, style = 2, info = 2, diag = F, ylim = lim, ylab = lab1)
R> plot(x = EU.G, style = 2, info = 1, diag = F, ylim = lim, ylab = lab2)
R> plot(x = EU.G, style = 2, info = 2, diag = F, ylim = lim, ylab = lab2)
```

The non-stationary aspect of the time series is evident in Figure 7 as the spectral power peaks around mid-1997, at both the finest and second levels. The local coherence estimates, Figure 8 (top panels), identify that the four channels have strong linear dependence. However, the local partial coherence estimates in Figure 8 (bottom panels) are lower. This indicates that

some of the linear dependence between a given market pair occurs indirectly because of their relationship with the other (observed) markets. For example, at the finest level, the average coherence between SMI and CAC is 0.63 but its average partial coherence drops to 0.16. The partial coherence estimate at the second and coarser levels becomes more challenging to interpret as some matrices of the multivariate EWS are near singular.

7. Discussion

The **mvLSW** package contains a number of tools to assess a multivariate non-stationary time series under the locally stationary wavelet framework defined by Park *et al.* (2014). The main command `mvEWS` estimates the multivariate EWS for a LSW time series, Section 3.1, and is flexible for the user to specify the best wavelet transform and smoothing kernel that is most appropriate to explore the spectral structure of the time series. However, perhaps more usefully, the package also includes tools for analyzing the dependence structure that occurs between pairs of channels, as described by the localized coherence and partial coherence measures presented in Section 4.1. Uncertainty in the multivariate EWS estimate can be quantified analytically by the routines demonstrated in Section 5.1 in evaluating approximate, point-wise, 95% confidence intervals for the spectral elements.

A complete case study for examining the spectral structure of the European financial indices `EuStockMarkets` was demonstrated in Section 6. This showed that a major non-stationary feature occurs due to a peak in power at the finest level in 1997. Furthermore, analysis of the coherence structure identifies that the indices are strongly and positively linearly dependent with a notable fraction arising from the indirect relationship between all four markets.

Acknowledgments

References

Bardwell L, Eckley IA, Fearnhead P, Smith S, Spott M (2019). "Most Recent Changepoint Detection in Panel Data." *Technometrics*, **61**(1), 88–98. `doi:10.1080/00401706.2018.1438926`.

Cardinali A, Nason GP (2013). "Costationarity of Locally Stationary Time Series Using **costat**." *Journal of Statistical Software*, **55**(1), 1–22. `doi:10.18637/jss.v055.i01`.

Cho H, Fryzlewicz P (2015). "Multiple-Change-Point Detection for High Dimensional Time Series via Sparsified Binary Segmentation." *Journal of the Royal Statistical Society B*, **77**(2), 475–507. `doi:10.1111/rssb.12079`.

Cribben I (2012). *Detecting Dependence Change Points in Multivariate Time Series with Applications in Neuroscience and Finance.* Ph.D. thesis, Columbia University. `doi:10.7916/D8JQ1CF0`.

Dahlhaus R (1997). "Fitting Time Series Models to Nonstationary Processes." *The Annals of Statistics*, **25**(1), 1–37. `doi:10.1214/aos/1034276620`.

Dahlhaus R (2012). "Locally Stationary Processes." In TS Rao, SS Rao, CR Rao (eds.), *Time Series Analysis: Methods and Applications*, volume 30 of *Handbook of Statistics*, chapter 13, pp. 351–413. Elsevier. `doi:10.1016/b978-0-444-53858-1.00013-2`.

Daubechies I (1990). "The Wavelet Transform, Time-Frequency Localization and Signal Analysis." *IEEE Transactions on Information Theory*, **36**(5), 961–1005. `doi:10.1109/18.57199`.

Eckley IA, Nason GP (2005). "Efficient Computation of the Discrete Autocorrelation Wavelet Inner Product Matrix." *Statistics and Computing*, **15**(2), 83–92. `doi:10.1007/s11222-005-6200-y`.

Eckley IA, Nason GP (2011). "**LS2W**: Implementing the Locally Stationary 2D Wavelet Process Approach in R." *Journal of Statistical Software*, **43**(3), 1–23. `doi:10.18637/jss.v043.i03`.

Eckley IA, Nason GP, Treloar RL (2010). "Locally Stationary Wavelet Fields with Application to the Modelling and Analysis of Image Texture." *Journal of the Royal Statistical Society C*, **59**(4), 595–616. `doi:10.1111/j.1467-9876.2009.00721.x`.

Fryzlewicz P, Nason GP (2006). "Haar-Fisz Estimation of Evolution Wavelet Spectra." *Journal of the Royal Statistical Society B*, **68**(4), 611–634. `doi:10.1111/j.1467-9868.2006.00558.x`.

Hallin M (1986). "Non-Stationary q-Dependent Processes and Time-Varying Moving Average Models." *Advances in Applied Probability*, **18**(1), 170–210. `doi:10.2307/1427242`.

Nason GP (2008). *Wavelet Methods in Statistics with R*. Springer-Verlag, New York. `doi:10.1007/978-0-387-75961-6`.

Nason GP (2016). **wavethresh**: *Wavelets Statistics and Transforms*. R package version 4.6.8, URL `https://CRAN.R-project.org/package=wavethresh`.

Nason GP, Silverman BW (1994). "The Discrete Wavelet Transform in S." *Journal of Computational and Graphical Statistics*, **3**(2), 163–191. `doi:10.1080/10618600.1994.10474637`.

Nason GP, von Sachs R, Kroisandt G (2000). "Wavelet Processes and Adaptive Estimation of the Evolutionary Wavelet Spectrum." *Journal of the Royal Statistical Society B*, **62**(2), 271–292. `doi:10.1111/1467-9868.00231`.

Olea R, Palma W, Rubio P (2015). **LSTS**: *Locally Stationary Time Series*. R package version 1.0, URL `https://CRAN.R-project.org/package=LSTS`.

Ombao H, Raz JA, von Sachs R, Guo W (2002). "The SLEX Model of a Non-Stationary Random Process." *Annals of the Institute of Statistical Mathematics*, **54**(1), 171–200. `doi:10.1023/a:1016130108440`.

Ombao H, von Sachs R, Guo W (2005). "SLEX Analysis of Multivariate Nonstationary Time Series." *Journal of the American Statistical Association*, **100**(470), 519–531. `doi:10.1198/016214504000001448`.

Ombao HC, Raz JA, Strawderman RL, von Sachs R (2001). "A Simple Generalised Cross-validation Method of Span Selection for Periodogram Smoothing." *Biometrika*, **88**(4), 1186–1192. `doi:10.1093/biomet/88.4.1186`.

Park T (2014). *Wavelet Methods for Multivariate Nonstationary Time Series*. Ph.D. thesis, STOR-i Centre for Doctoral Training, Lancaster University, Lancaster.

Park T, Eckley IA, Ombao HC (2014). "Estimating Time-Evolving Partial Coherence Between Signals via Multivariate Locally Stationary Wavelet Processes." *IEEE Transactions on Signal Processing*, **62**(20), 5240–5250. `doi:10.1109/tsp.2014.2343937`.

Percival DB, Walden AT (2000). *Wavelet Methods for Time Series Analysis*. Cambridge University Press, Cambridge. `doi:10.1017/CBO9780511841040`.

Priestley MB (1965). "Evolutionary Spectra and Non-Stationary Processes." *Journal of the Royal Statistical Society B*, **27**(2), 204–237. `doi:10.1111/j.2517-6161.1965.tb01488.x`.

R Core Team (2019). R*: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Rua A, Nunes LC (2009). "International Comovement of Stock Market Returns: A Wavelet Analysis." *Journal of Empirical Finance*, **16**(4), 632–639. `doi:10.1016/j.jempfin.2009.02.002`.

Ryan JA, Ulrich JM (2018). **xts***: eXtensible Time Series*. R package version 0.11-2, URL `https://CRAN.R-project.org/package=xts`.

Sanderson J, Fryzlewicz P, Jones M (2010). "Estimating Linear Dependence between Non-stationary Time Series Using the Locally Stationary Wavelet Model." *Biometrika*, **97**(2), 435–446. `doi:10.1093/biomet/asq007`.

Schnabel RB, Eskow E (1999). "A Revised Modified Cholesky Factorization Algorithm." *SIAM Journal on Optimization*, **9**(4), 1135–1148. `doi:10.1137/s105262349833266x`.

Shama AA (2007). "Simplified Procedure for Simulating Spatially Correlated Earthquake Ground Motions." *Engineering Structures*, **29**(2), 248–258. `doi:10.1016/j.engstruct.2006.04.018`.

Taylor S, Park T, Eckley IA, Killick R (2019). **mvLSW***: Multivariate, Locally Stationary Wavelet Process Estimation*. R package version 1.2.3, URL `https://CRAN.R-project.org/package=mvLSW`.

Zeileis A, Grothendieck G (2005). "**zoo**: S3 Infrastructure for Regular and Irregular Time Series." *Journal of Statistical Software*, **14**(6), 1–27. `doi:10.18637/jss.v014.i06`.

Affiliation:

Simon A. C. Taylor, Timothy Park, Idris A. Eckley
Department of Mathematics and Statistics
Lancaster University
Lancaster, United Kingdom
E-mail: s.taylor2@lancaster.ac.uk,
       parkt@lancaster.ac.uk,
       i.eckley@lancaster.ac.uk