Dynamic detection of anomalous regions within distributed acoustic sensing data streams using locally stationary wavelet time series

Rebecca E. Wilson¹ · Idris A. Eckley² · Matthew A. Nunes³ · Timothy Park⁴

Received: 15 November 2017 / Accepted: 27 September 2018 / Published online: 20 February 2019
© The Author(s) 2019

Abstract

Distributed acoustic sensing technology is increasingly being used to support production and well management within the oil and gas sector, for example to improve flow monitoring and production profiling. This sensing technology is capable of recording substantial data volumes at multiple depths within an oil well, giving unprecedented insights into production behaviour. However the technology is also prone to recording periods of anomalous behaviour, where the same physical features are concurrently observed at multiple depths. Such features are called ‘stripes’ and are undesirable, detrimentally affecting well performance modelling. This paper focuses on the important challenge of developing a principled approach to identifying such anomalous periods within distributed acoustic signals. We extend recent work on classifying locally stationary wavelet time series to an online setting and, in so doing, introduce a computationally-efficient online procedure capable of accurately identifying anomalous regions within multivariate time series.

Keywords Distributed acoustic sensing · Wavelets · Locally stationary time series · Coherence · Dynamic classification · Stripe detection

Responsible editor: Alípio Jorge, Rui L. Lopes, German Larrazabal.

This work has been partially supported by the Engineering and Physical Sciences Research Council under Grants: EP/L015692/1 and Shell Research Ltd.

✉ Idris A. Eckley
i.eckley@lancaster.ac.uk

¹ STOR-i Centre for Doctoral Training, Lancaster University, Lancaster LA1 4YF, UK

² Department of Mathematics and Statistics, Lancaster University, Lancaster LA1 4YF, UK

³ School of Mathematical Sciences, University of Bath, Bath BA2 7AY, UK

⁴ Statistics and Data Science, Shell Global Solutions International BV, Amsterdam, The Netherlands

1 Introduction

The ability to accurately analyse geoscience data at, or close to, real time is becoming increasingly important. For example, within the oil and gas sector this need can arise as a consequence of (1) the sheer volume of data now being collected and (2) operational considerations. It is this setting that we consider in this article, seeking to enable the rapid identification of certain anomalous features within Distributed Acoustic Sensing data obtained from an oil producing facility. Specifically, we seek to build on recent work within the non-stationary time series community to develop an approach that permits the online monitoring of these complex signals.

The technology used to generate the data considered in this article, Distributed Acoustic Sensing (DAS), involves the use of a fibre-optic cable as a sensor in which the entire length of the fibre is used to measure acoustic or thermal disturbances. DAS originates from the defence industry where it is commonly used in security and border monitoring (Owen et al. 2012). Recently, the technology has been applied within the oil and gas industry, for example in pipeline monitoring and management (Williams 2012; Mateeva et al. 2014a,b). The use of DAS to monitor production volumes and composition within a well requires the installation of a fibre-optic cable along the length of the well combined with an interrogator unit on the surface (Paleja et al. 2015). This unit sends light pulses down the cable and processes the back-scattered light. The installation of such technology has become popular as it can be a cost effective way to obtain continuous, real-time and high-resolution information.

When monitoring the behaviour of wells it is important to be able to detect unusual occurrences, including potential corruptions of the data. Striping is one particular form of corruption that can have a particularly deleterious effect, rendering data potentially unusable in a specific time region. Stripes are characterised by sudden, and distinctive changes in the structure of the signal over time, see Mateeva et al. (2014a,b) and Ellmauthaler et al. (2017) for examples. These features can be present simultaneously across all channels or only apparent across a subset of channels, for example from the surface to a set depth within the well. Crucially, the occurrence of stripes simultaneously at different locations indicates that these features are not physical. Instead stripes can occur for a number of reasons, including a disturbance of the fibre-optic cable near the unit, or problems with the electronics due to the high sampling rate.

Visually, stripes can manifest themselves in a variety of ways. Some are visually obvious within the DAS data, such as the stripe that occurs at around 4000 ms in Fig. 1a. Other occurrences can be more subtle, and therefore more challenging to detect. For example, the stripe could be a change in the second-order structure. Critically such features can make it difficult to carry out further analysis of the data, such as flow rate analysis. For this reason, there is significant interest in being able to detect regions of striping as soon as they occur, so that they can be removed whilst keeping as much of the original signal intact as possible. It is this challenge of dynamically detecting striping regions that motivates the work presented in this article.

There exist a variety of techniques for the classification of time series in the statistical and machine learning literature. An exhaustive review is beyond the scope of this article, but popular classification methods include hidden Markov models (HMM) (see e.g. Rabiner 1989; Ephraim and Merhav 2002; Cappé et al. 2009); support vec-

Fig. 1 Time series plots of DAS amplitude at four different well depths over the same time period: **a** original series; **b** detrended series. The highlighted regions in **a** indicate three examples of stripping

tor machines (Cortes and Vapnik 1995; Muller et al. 2001; Kampouraki et al. 2009); Gaussian mixture models (McLachlan and Peel 2004; Povinelli et al. 2004; Kersten 2014); nearest neighbour classifiers (Zhang et al. 2004; Wei and Keogh 2006) and multiscale methods (Chan and Fu 1999; Mörchén 2003; Aykroyd et al. 2016) to name but a few. More recent contributions for large-scale (online) classification include the MOA machine learning framework (Bifet et al. 2010; Read et al. 2012). For a recent overview of classification in the time series context, see for example Fu (2011). Dependent on the application being considered, one might adopt various modelling choices. For example, some classifiers have distinct advantages, such as simplicity of implementation, speed or suitability for massive online applications. However many, such as GMM or SVM-based approaches, do not explicitly allow temporal dependence or are limited to a narrow class of series structure (HMMs), which is seen as crucial to classification of time series in the majority of realistic settings (see e.g. Bifet et al. 2013). Complex hidden dependence structure is typical of the DAS data studied in this article (see Fig. 2).

Our approach to the dynamic stripe identification problem builds on recent work within the time series literature. Wavelet approaches to modelling time series have become very popular in recent years, principally because of their ability to provide time-localised measures of the spectral content inherent within many contemporary data (e.g. Killick et al. 2013; Nam et al. 2015; Chau and von Sachs 2016; Nason et al. 2017). This *locally stationary* modelling paradigm is flexible enough to represent a wide range of non-stationary behaviour and has also been extended to enable the modelling and estimation of multivariate non-stationary time series structures (e.g. Sanderson et al. 2010; Park et al. 2014). Typically these settings assume that the data have already been collected, and are available for offline analyses.

The novel contribution in this article is to employ the MvLSW modelling framework of Park et al. (2014) to represent the DAS data, using a moving window approach, thereby extending previous work to the online dynamic classification setting. This modelling framework allows us to classify multivariate time series with complex dependencies both *within* and *between* channels of the series, including those which exhibit visually subtle changes in behaviour over time. Reusing data calculations allows us to also produce a computationally efficient nondecimated wavelet transform in the online setting.

Our article is organised as follows. Section 2 contains an overview of the Multivariate Locally Stationary Wavelet (MvLSW) model and existing dynamic classification method. In Sect. 3, we describe the proposed online classification method. Section 4 contains a simulation study evaluating the performance of the proposed classifier using synthetic data, further justifying the use of time-varying coherence as a feature for classification. A case study using an acoustic sensing dataset is then described in Sect. 5, where we discuss the utility of the proposed classifier as a stripe detection method. Finally, Sect. 6 includes some concluding remarks.

2 Wavelets and time series

The problem of modelling and analysing non-stationary time series can be approached in a number of ways that often involve assuming the changing second-order structure adopts a time varying spectrum or autocovariance. Examples within the existing literature include the oscillatory processes (Priestley 1981), Locally Stationary Fourier model (Dahlhaus 1997) and time-varying autoregressive processes (Dahlhaus et al. 1999). Due to the high frequency nature of acoustic sensing data, we focus our attention on wavelet-based methods such as the Locally Stationary Wavelet (LSW) processes, introduced by Nason et al. (2000). The use of wavelets allows for the time-scale decomposition of a signal using computationally efficient transform algorithms, whilst also allowing the structure to change over time.

Methods for the classification of non-stationary time series can broadly be divided into two categories; static or dynamic. Static classification approaches attempt to assign an entire test signal to a particular class. They differ in the way in which they

Fig. 2 Hidden time-varying coherence structure of the DAS series in Fig. 1 at selected wavelet scales (resolutions): **a** coherence between series 1 and 2; **b** coherence between series 3 and 4

choose to model the nonstationarity, including through Locally Stationary Fourier processes (Sakiyama and Taniguchi 2004), the smooth localised exponentials (SLEX) framework (Huang et al. 2004) and wavelets (Fryzlewicz and Ombao 2009; Krzemieniewska et al. 2014). In contrast, dynamic classification approaches allow for the class assignment of the test signal to vary over time which allows for more flexibility in the classification and covers problems where the underlying nonstationarity is due to class switching. The method that we introduce is an online analogue of the dynamic classification approach of Park et al. (2018) which looks to detect subtle changes in the dependence structure of a multivariate signal in a fast and efficient manner.

Before introducing our proposed method in Sect. 3, we first outline some details of the locally stationary wavelet framework of Park et al. (2014) together with the (offline) dynamic classification method introduced in Park et al. (2018) that forms the basis of our online approach. We begin with some introductory concepts of wavelets. For a more comprehensive introduction to the area, see Nason (2008) or Vidakovic (1999).

2.1 Discrete wavelet transforms

Succinctly, wavelets can be seen as oscillatory basis functions with which one can represent signals in a multiscale manner. More specifically, for a function (signal) $f \in L^2(\mathbb{R})$, we can write

$$f(x) = \sum_{k \in \mathbb{Z}} c_{0,k} \phi_{0,k}(x) + \sum_{j \leq J} \sum_{k \in \mathbb{Z}} d_{j,k} \psi_{j,k}(x),$$

for scales j and locations k , and where the wavelet $\psi_{j,k}(x) = 2^{-j/2} \psi(2^{-j}x - k)$ is a basis function formed as a dilation and translation of a “mother” wavelet ψ ; scaling functions $\phi_{j,k}$ are similarly formed as dilated and translated versions of a father wavelet ϕ . The wavelet coefficients $d_{j,k}$ capture local oscillatory behaviour of the signal at a scale (frequency) j , whereas the scaling coefficients $c_{j,k}$ represent the signal’s smooth (mean) behaviour at different scales. More specifically, fine scale coefficients capture the local characteristics of a signal; coarse scale coefficients describe the overall behaviour of the signal.

The discrete wavelet transform (DWT). Computation of wavelet coefficients resulting from traditional wavelet transforms is performed using the so-called *Discrete Wavelet Transform* (DWT), first introduced by Mallat (1989). The algorithm proceeds by alternately applying high- and low-pass filtering and decimation (subsampling) operations to the observed data.

Let $\mathcal{H} := \{h_k\}$ and $\mathcal{G} := \{g_k\}$ be a low- and high-pass filter pair associated with a given wavelet, such as the quadrature mirror filters used in the construction of compactly supported wavelets introduced by Daubechies (1992). Following Nason and Silverman (1995), let $\mathcal{D}_0$ denote the even decimation operator that selects every even-indexed element in a sequence, in other words $(\mathcal{D}_0 x)_l = x_{2l}$. The detail coefficients

of the DWT of a time series $X = \{x_t\}_{t=1}^T$ (where $T = 2^J$ for some positive integer J) can be found using

$$\mathbf{d}_j = \mathcal{D}_0 \mathcal{G}(\mathcal{D}_0 \mathcal{H})^{J-j-1} X,$$

for $j = 0, \dots, J-1$. Similarly, the scaling or smooth coefficients of the DWT are given by

$$\mathbf{c}_j = (\mathcal{D}_0 \mathcal{H})^{J-j} X,$$

for $j = 0, \dots, J$.

The information contained in the original time series X can thus be fully described by the set of coefficients $\{\mathbf{d}_{J-1}, \mathbf{d}_{J-2}, \dots, \mathbf{d}_0, \mathbf{c}_0\}$.

The nondecimated discrete wavelet transform (NDWT). The *nondecimated* wavelet transform (NDWT) is a modification of the DWT outlined above in which the decimation step is not carried out, resulting in 2^J smooth and detail coefficients at each level of the transform. This allows for a fuller description of the local characteristics of the data in the decomposition, a feature that turns out to be particularly helpful for describing time series. A more detailed treatment of the NDWT can be found in Nason and Silverman (1995), see also Coifman and Donoho (1995) and Percival (1995). In the context of streaming data, the transform is such that only a small number of coefficients need to be recomputed at each time step, recycling previously evaluated coefficients. This computationally efficient algorithm will be used within our online dynamic classification technique described in Sect. 3.

2.2 Multivariate locally stationary wavelet (MvLSW) processes

We now turn to consider the application of wavelets within non-stationary time series models. Specifically we focus on the recently proposed multivariate locally stationary wavelet (MvLSW) framework introduced by Park et al. (2014), which we later use to model the DAS data described in Sect. 1. This approach provides a flexible model for multivariate time series that is able to capture (second order) nonstationarity, as well as temporally inhomogeneous dependence structure between channels of a multivariate series.

Following Park et al. (2014), a P -variate locally stationary wavelet process $\{\mathbf{X}_{t,T}\}_{t=1}^T$ can be represented as

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

where $T = 2^J \geq 1$ and $\mathbf{V}_j(k/T)$ is the lower-triangular transfer function matrix. Each element of the transfer function matrix is assumed to be a Lipschitz continuous function with Lipschitz constants, L_j , that satisfy $\sum_{j=1}^{\infty} 2^j L_j^{(p,q)} < \infty$ for each pair of channels (p, q) . The vectors $\psi_j = (\psi_{j,0}, \dots, \psi_{j,(N_j-1)})$ are discrete non-decimated

wavelets associated to a low- / high-pass filter pair, $\{\mathcal{H}, \mathcal{G}\}$, constructed according to Nason et al. (2000) as

$$\begin{aligned}\psi_{1,n} &= \sum_k g_{n-2k} \delta_{0,k} = g_n \quad \text{for } n = 0, 1, \dots, N_1 - 1, \\ \psi_{j+1,n} &= \sum_k h_{n-2k} \psi_{j,k} \quad \text{for } n = 0, 1, \dots, N_{j+1} - 1.\end{aligned}$$

In the equations above, $\delta_{0,k}$ is the Kronecker delta function and $N_j = (2^j - 1)(N_h - 1) + 1$ where N_h is the number of non-zero elements of the filter $\mathcal{H} = \{h_k\}_{k \in \mathbb{Z}}$. The random vectors $\mathbf{z}_{j,k}$ in (1) are defined such that $E(\mathbf{z}_{j,k}) = \mathbf{0}$ and $\text{cov}(\mathbf{z}_{j,k}^{(i)}, \mathbf{z}_{j',k'}^{(i')}) = \delta_{i,i'} \delta_{j,j'} \delta_{k,k'}$.

The local wavelet spectral (LWS) matrix and the wavelet coherence of a multivariate signal are key quantities of interest in the dynamic classification problem. Given a MvLSW signal $\mathbf{X}_t$ with associated transfer function matrix, $\mathbf{V}_j(z)$, the local wavelet spectral matrix $\mathbf{S}_j(z)$ is defined as

$$\mathbf{S}_j(z) = \mathbf{V}_j(z) \mathbf{V}_j(z)^\top. \quad (2)$$

This quantity describes the cross-covariance between channels at each scale and (rescaled) location z . The coherence is a measure of the dependence between the channels of a multivariate signal at a particular time and scale. Following Park et al. (2014), the wavelet coherence matrix $\boldsymbol{\rho}_j(z)$ is given by

$$\boldsymbol{\rho}_j(z) = \mathbf{D}_j(z) \mathbf{S}_j(z) \mathbf{D}_j(z), \quad (3)$$

where $\mathbf{S}_j(z)$ is the LWS matrix from (2) and $\mathbf{D}_j(z)$ is a diagonal matrix with entries given by $S_j^{(p,p)}(z)^{(-1/2)}$. It is these spectral and coherence quantities which we use to enable us to accurately classify multichannel signals with time-varying dependence and second order structure.

Estimation of MvLSW spectral and coherence components. In practice, the coherence and LWS matrix are unknown for an observed multivariate series and need to be estimated. The LWS matrix of a multivariate signal can be estimated by first calculating the empirical wavelet coefficient vector $\mathbf{d}_{j,k} = \sum_t \mathbf{X}_t \psi_{j,k-t}$ at locations k and scales j . The raw wavelet periodogram matrix is then defined as $\mathbf{I}_{j,k} = \mathbf{d}_{j,k} \mathbf{d}_{j,k}^\top$.

Park et al. (2014) establish that the raw wavelet periodogram is a biased and inconsistent estimator of the true LWS matrix, $\mathbf{S}_j(z)$. However, they show that (asymptotically) this bias is described by the well known inner product matrix of discrete autocorrelation wavelets, $\mathbf{A}$. The elements of $\mathbf{A}$ are given by $A_{j\ell} = \sum_\tau \Psi_j(\tau) \Psi_\ell(\tau)$ where $\Psi_j(\tau) = \sum_k \psi_{jk}(0) \psi_{jk}(\tau)$ (see Eckley and Nason 2005 or Nason et al. 2000 for further information). The bias inherent within the raw wavelet periodogram can therefore be removed using the inverse of this inner product matrix. To obtain consistency, the resulting estimate must be smoothed in some way, for example using a rectangular kernel smoother (Park et al. 2014). This results in an (asymptotically) unbiased, and consistent, estimator of the LWS matrix, $\mathbf{S}_{j,k}$, given by

$\hat{\mathbf{S}}_{j,k} = (2M + 1)^{-1} \sum_{m=k-M}^{k+M} \sum_l A_{jl}^{-1} \mathbf{I}_{lm}$, where M denotes the kernel bandwidth corresponding to a smoothing window of length $2M + 1$. Estimation of the wavelet coherence matrix is then straightforward, simply using a plug-in estimator, substituting $\hat{\mathbf{S}}$ into Eq. (3).

With the key modelling notation established, we now briefly summarise an approach to dynamic classification based upon the MvLSW framework. This will be the cornerstone of the approach that we propose in Sect. 3.

2.3 Dynamic classification

Following their work on MvLSW processes, Park et al. (2018) introduced an approach to dynamically classify a Multivariate Locally Stationary Wavelet signal $\mathbf{X}_t$ whose class membership may change over time. The approach assumes that at any time t , the signal $\mathbf{X}_t$ can belong to one of $N_c \geq 2$ different classes, where N_c is known. Let $C_X(t)$ denote the class membership of X_t at time t where $C_X(t) \in \{1, 2, \dots, N_c\}$. Following Park et al. (2018), the signal $\mathbf{X}_t$ in (1) can be then represented as

$$\mathbf{X}_t = \sum_j \sum_k \sum_{c=1}^{N_c} \mathbb{I}_c[C_X(k)] \mathbf{V}_j^c \psi_{j,k}(t) \mathbf{z}_{j,k},$$

where $\mathbf{V}_j^c$ is the class specific transfer function which is constrained to be constant over time and $\mathbb{I}_c[C_X(k)]$ represents an indicator function which equals 1 if $C_X(k) = c$ and 0 otherwise.

To classify the multivariate signal $\mathbf{X}_t$, the approach makes use of a set of N_i training signals, denoted $\{Y_t^{(i)}\}$ for $i \in \{1, 2, \dots, N_i\}$. It is assumed that the class membership of these training signals over time, $C_{Y^{(i)}}(t)$, is known. For each training signal, the LWS matrix $\hat{\mathbf{S}}_{jk;Y^{(i)}}$ and coherence matrix $\rho_{jk;Y^{(i)}}$ can be estimated, as discussed in Sect. 2.2.

The aim of this classification method is to calculate the probability of the signal belonging to a particular class at a given time point. To do this, the likelihood of the signal belonging to each class given the information contained in the training signals is calculated. It is necessary to apply a Fisher-z transform to the coherence estimates to ensure that the estimates can be approximated by a Gaussian distribution. For a class c , the transformed coherence $\zeta_j^{(c)}$ is given by

$$\zeta_j^{(c)} = \tanh^{-1} \rho_j^{(c)}. \quad (4)$$

The mean and variance of the transformed coherence for class c can be estimated using the transformed coherence for the training signals that are known to belong to that particular class. Note that in practice, the Gaussian distribution will be an approximation to the true distribution of the (finite sample) Fisher z-transformed coherence estimates. We recommend that, as for any such analysis, this assumption is validated for any data set analysed.

As in Krzemieniewska et al. (2014), classification is performed using a subset of wavelet coefficients that show the largest difference between the classes in terms of the transformed coherence. The subset, denoted by $\mathcal{M}$, consists of the scale and channel indices (j, p, q) for $p < q$. $\mathcal{M}$ is made up of the coefficients that have the largest values of the discrepancy measure $\Delta_j^{(p,q)}$ given by

$$\Delta_j^{(p,q)} = \sum_{c=1}^{N_c} \sum_{g=c+1}^{N_c} \left| \frac{\widehat{\xi}_j^{(p,q)(c)} - \widehat{\xi}_j^{(p,q)(g)}}{\sqrt{\text{var}(\widehat{\xi}_j^{(p,q)(c)}) + \text{var}(\widehat{\xi}_j^{(p,q)(g)})}} \right|. \quad (5)$$

In practice, a proportion $\wp$ is typically chosen and the subset $\mathcal{M}$ are those $\wp\%$ of time-scale indices with the largest discrepancies (Krzemieniewska et al. 2014). In order to estimate the probability that the signal $\mathbf{X}_t$ belongs to a particular class at a given time, the transformed coherence $\widehat{\xi}_{jk;X}$ is first estimated, before using Bayes' theorem to obtain

$$\Pr[C(k) = c | \widehat{\xi}_{jk;X}] \propto \Pr[C(k) = c] \mathcal{L}(\widehat{\xi}_{jk;X} | \xi_j(k/T) = \xi_j^{(c)} \quad \forall j), \quad (6)$$

where $\mathcal{L}(\theta|x)$ is the likelihood and $\Pr[C(k) = c]$ is a prior probability (Park et al. 2018). Due to the use of the Fisher-z transform in (4), the likelihood $\mathcal{L}(\theta|x)$ takes the form of a Gaussian likelihood with mean vector $\mu^{(c)}$ and covariance matrix $\Sigma^{(c)}$. The vector $\mu^{(c)}$ consists of the elements of $\widehat{\xi}_j^{(p,q)(c)} \quad \forall (j, p, q) \in \mathcal{M}$, whilst similarly $\widehat{\mu}_k$ contains the elements $\widehat{\xi}_{jk;X}^{(p,q)} \quad \forall j, p, q \in \mathcal{M}$. The Gaussian likelihood hence takes the form

$$\mathcal{L}(\widehat{\xi}_{jk;X} | \xi_j(k/T) = \xi_j^{(c)} \quad \forall j) \propto |\Sigma^{(c)}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \{(\widehat{\mu}_k - \mu^{(c)})^\top (\Sigma^{(c)})^{-1} (\widehat{\mu}_k - \mu^{(c)})\}\right). \quad (7)$$

In practice, the true mean vectors and covariance matrices of $\widehat{\xi}_{jk;X}$ are unknown, and they are estimated using the training data. In the examples provided in Sect. 4, we use a flat (uninformative) prior. However, of course, many other prior specifications could be used in the formulation above to reflect beliefs from application-specific expert knowledge.

The dynamic classification method described here is an offline approach that calculates the probability of belonging to a particular class at each time point. Since we are interested in detecting stripes in DAS data in an online setting, we adapt the existing method to allow for classification of data streams. We describe our approach below.

3 Online dynamic classification of multivariate series

In order to adapt the existing dynamic classification method outlined in Sect. 2.3 to an online setting, we make use of a moving window approach. The use of such a window encapsulates the constraint in many data streaming applications that there is only a limited data storage and memory with which to perform analysis.

Our online dynamic classification technique proceeds as follows. For a window of length $w = 2^J$ the first step of our algorithm is to calculate the set of discriminative indices as defined in Eq. (5) using a set of training signals of length w . For reasons of efficiency, the discriminative indices are used in the classification step for each window of the data. Although window-specific indices could be used, in our experience, updating the set of discriminative indices for each window increases computational complexity without providing significant accuracy improvement. The dynamic classification method described in Sect. 2.3 is applied to the first window of data to obtain the probability that the signal belongs to a particular class for the time points in the window.

Upon arrival of a new data point, the window then shifts by one, and the data under analysis consists of the old data together with the new data point, but we also lose the first data point contained in the previous window. The online wavelet transform is then used to efficiently update the wavelet coefficients and the transformed coherence estimate for the new window. Using the information previously calculated from the training signals, we can then obtain the probability that the signal belongs to a particular class for the time points contained in the new window. The algorithm continues by repeatedly moving the window for each new data point and estimating the probability of each data point belonging to a class until we reach the end of the data stream.

During our classification algorithm, we obtain multiple estimates for the probability that a signal belongs to a particular class (at each time point) from the different windows into which a data point falls. For example, for a time series of length T analysed with a moving window of length $w < T$, we obtain w estimates for the probabilities of an individual time point t belonging to a given class c , which we denote $p_{t,i}^{(c)}$ for window i .

A question that arises as a result of the iterative approach is how to combine the estimates from different windows to obtain an overall probability that the time point belongs to a particular class, and hence classify the signal. In what follows, for computational simplicity we use a simple average, but other more sophisticated combination methods could be used. In other words, our final probability estimates are given by

$$p_t^{(c)} = \frac{1}{w} \sum_{i=1}^w p_{t,i}^{(c)} \quad \text{for } t = 1, 2, \dots, T. \quad (8)$$

In some applications, an overall classification of the signal is required rather than probability estimates. In this case, the class c that has the largest probability $p_t^{(c)}$ is assigned to the time point t for all $t \in \{1, 2, \dots, T\}$.

A summary of our method for estimating the probability that a given multivariate signal belongs to a particular class c at a particular time is given in Algorithm 1.

Online dynamic classification:

1. Let X be a P -variate signal of length T that we wish to classify using a moving window of length w .
 2. Calculate the set of discriminative indices using a set of P -variate training signals of length w , whose class assignments are known.
 3. Apply dynamic classification method to the first window of data $X[1 : w]$ to obtain the probability that the signal belongs to a particular class c for the time points in the window, denoted $p_{t,1}^{(c)}$ for $t = 1, 2, \dots, w$.
 4. Iterate for i in 2 to $T - w + 1$
 - (a) Apply the online wavelet transform to the new window of data $X[i : i + w - 1]$ to update the wavelet coefficients.
 - (b) Update the transformed coherence at the set of discriminative indices using the wavelet coefficients calculated in the previous step.
 - (c) Apply dynamic classification method to obtain the probability that the signal belongs to a particular class for window i , denoted $p_{t,i}^{(c)}$ for $t = i, i + 1, \dots, i + w - 1$.
 5. Average probability estimates for each window using (8) to obtain the final probability that the signal X belongs to a particular class c over time, $p_t^{(c)}$ for $t = 1, 2, \dots, T$.
-

Alg. 1: Finding the average probability that a multivariate signal belongs to a particular class over time.

4 Synthetic data examples

We now turn to assess the performance of our proposed online dynamic classification approach. To this end, a simulation study is designed to test the ability of this wavelet-based approach to classify data streams exhibiting various characteristics. More specifically, the study consists of three different scenarios. These scenarios are chosen to mimic signals arising in practice:

Scenario 1: Signal of length 1024, short time segments of length 100 between changes in class, nine class changes in total.

Scenario 2: Signal of length 1024, alternating long/short segments of length 300 and 100 between changes, five class changes in total.

Scenario 3: Signal of length 2048, long segments of length 300 between changes, six class changes in total.

For all scenarios, the generated series randomly switch classes between time segments. A window length of 256 is used when implementing the online dynamic classification method and the training data consists of 10 signals, some of which contain changes in class. The R packages **wavethresh** (Nason 2016) and **mvLSW** (Taylor et al. 2017) are used to calculate the wavelet coefficients and transformed coherence that are used in the online dynamic classification.

Long segments of length 300 between class changes are chosen to ensure that there is a maximum of one class change in each dynamic classification window. In the situation where the class changes are reasonably far apart, we expect the online dynamic

classification algorithm to classify the signal well. As a contrast, short segments of length 100 are also chosen to demonstrate some potential limitations of the method. In particular, when the signal contains multiple class changes that are close together, there is a possibility that our approach will misclassify the signals.

For each scenario, we consider a number of examples of generating processes for the classes in the multivariate series. The first example we examine consists of three classes where each class is defined by a trivariate normal signal with mean $\mu = (0, 0, 0)$ and differing cross-channel dependence structure. More specifically, the classes are defined by the three covariance matrices

$$\Sigma^{(1)} = \begin{pmatrix} 1 & 0 & 0.3 \\ 0 & 1 & 0.7 \\ 0.3 & 0.7 & 1 \end{pmatrix}, \Sigma^{(2)} = \begin{pmatrix} 1 & 0.6 & 0.1 \\ 0.6 & 1 & -0.4 \\ 0.1 & -0.4 & 1 \end{pmatrix} \text{ and } \Sigma^{(3)} = \begin{pmatrix} 1 & -0.5 & -0.2 \\ -0.5 & 1 & 0.1 \\ -0.2 & 0.1 & 1 \end{pmatrix}.$$

Example simulated data for this process using the different class switching scenarios above are shown in Fig. 3a.

To investigate the potential of our proposed approach further, we studied an example with a time-varying moving average (VMA) process, with three classes defined by the following coefficient matrices:

$$\begin{aligned} \text{Class 1: } \mathbf{X}_t &= \mathbf{Z}_t + \begin{pmatrix} 1 & 0 & 0.6 \\ 0 & 1 & 0.3 \\ 0.6 & 0.3 & 1 \end{pmatrix} \mathbf{Z}_{t-1} + \begin{pmatrix} 1 & 0.2 & 0.9 \\ 0.2 & 1 & 0.5 \\ 0.9 & 0.5 & 1 \end{pmatrix} \mathbf{Z}_{t-2} \\ \text{Class 2: } \mathbf{X}_t &= \mathbf{Z}_t + \begin{pmatrix} 1 & -0.7 & -0.3 \\ -0.7 & 1 & 0.4 \\ -0.3 & 0.4 & 1 \end{pmatrix} \mathbf{Z}_{t-1} + \begin{pmatrix} 1 & 0.9 & -0.3 \\ 0.9 & 1 & 0 \\ -0.3 & 0 & 1 \end{pmatrix} \mathbf{Z}_{t-2}, \\ \text{Class 3: } \mathbf{X}_t &= \mathbf{Z}_t + \begin{pmatrix} 1 & -0.4 & 0.2 \\ -0.4 & 1 & -0.6 \\ 0.2 & -0.6 & 1 \end{pmatrix} \mathbf{Z}_{t-1} + \begin{pmatrix} 1 & 0.1 & -0.5 \\ 0.1 & 1 & -0.3 \\ -0.5 & -0.3 & 1 \end{pmatrix} \mathbf{Z}_{t-2}, \end{aligned}$$

where $\mathbf{Z}_t, \mathbf{Z}_{t-1}$ and $\mathbf{Z}_{t-2}$ are IID multivariate Gaussian white noise (see Fig. 3b).

The third example we consider is a vector autoregressive process with intra- and cross-channel changes in dependence between each class (see Fig. 3c). The three classes in the example are defined by

$$\begin{aligned} \text{Class 1: } \mathbf{X}_t &= \begin{pmatrix} 0.2 & 0.3 & 0 \\ 0.3 & 0.5 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{X}_{t-1} + \begin{pmatrix} 0.6 & -0.1 & 0 \\ -0.1 & -0.3 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{X}_{t-2} + \epsilon_1, \\ \text{Class 2: } \mathbf{X}_t &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0.4 & -0.4 \\ 0 & -0.4 & 0.4 \end{pmatrix} \mathbf{X}_{t-1} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & -0.6 & 0.2 \\ 0 & 0.2 & 0.3 \end{pmatrix} \mathbf{X}_{t-2} + \epsilon_2, \\ \text{Class 3: } \mathbf{X}_t &= \begin{pmatrix} -0.1 & 0 & 0.4 \\ 0 & 0 & 0 \\ 0.4 & 0 & -0.5 \end{pmatrix} \mathbf{X}_{t-1} + \begin{pmatrix} 0.2 & 0 & -0.2 \\ 0 & 0 & 0 \\ -0.2 & 0 & -0.3 \end{pmatrix} \mathbf{X}_{t-2} + \epsilon_3, \end{aligned}$$

(a) Scenario 1, multivariate Gaussian series

(b) Scenario 2, vector moving average series

(c) Scenario 3, vector autoregressive series

Fig. 3 Example realisations of generating processes for the different scenarios used in the simulation study. **a** Short segments of length 100 between class changes; **b** alternating long/short segments of length 300 and 100 between changes; **c** long segments of length 300 between changes

where the noise vectors ϵ_i are zero-mean multivariate normal realisations, distributed with covariances

$$\Sigma_{\epsilon_1} = \begin{pmatrix} 3 & 0.3 & 0.9 \\ 0.3 & 3 & 1.4 \\ 0.9 & 1.4 & 3 \end{pmatrix}, \quad \Sigma_{\epsilon_2} = \begin{pmatrix} 2 & 1.3 & 0.4 \\ 1.3 & 1.8 & 0.3 \\ 0.4 & 0.3 & 2 \end{pmatrix}, \quad \Sigma_{\epsilon_3} = \begin{pmatrix} 5 & 3.3 & 2.5 \\ 3.3 & 4.5 & 2.8 \\ 2.5 & 2.8 & 3.5 \end{pmatrix}.$$

Competitor methods. In the simulation study, we compare our proposed method with a number of alternative classification techniques. Firstly we consider a Hidden Markov Model (HMM) approach—a probabilistic model of the joint distribution of observed variables, together with their “hidden” states (in this setting, classes). Such methods have previously been used for classification in the literature, see for example Ainsleigh et al. (2002). In this model, it is assumed that (1) the observed data at a particular time is independent of all other variables, given its class and (2) given the previous class, the class at a time is independent of all other variables (i.e. the changes in class are Markovian). This means that we assume that the probability of changing class does not depend on time or previous class membership, which can be an unrealistic assumption to make in practice. Furthermore, HMMs can be computationally intensive to implement especially in multiclass settings, requiring procedures such as the EM algorithm for tractable model fitting, see e.g. Cappé et al. (2009). An introduction to HMMs and their applications can be found in Zucchini and MacDonald (2009).

A sequential HMM approach is applied to both the full test signal and its transformed coherence at the set of discriminative indices, using the *R* package **HMM** (Himmelman 2010). In both cases, the model is initialized to have equal state probabilities, and then trained using the initial data. When a new data point arrives, the probabilities of belonging to each state are computed. This process of increasing the number of data points and computing the probabilities is repeated until we reach the end of the signal. As with the online dynamic classification approach, multiple estimates for the probability of belonging to a state at a particular time point are obtained. This is because each time a data point falls within a window, probabilities associated to the time point belonging to a particular class are calculated. For each time point, the estimates are averaged and the overall classification of the signal is then defined to be the most likely state at each point. We also considered a third variant of the sequential HMM approach that was applied to each window of the data used in the online dynamic classification and the corresponding transformed coherence. However this produced poor results so we omit them from the comparisons below.

To demonstrate the importance of accounting for the dependence structure within the test series, we also apply a support vector machine (SVM) classifier to the series, available in the *R* package **e1071** (Meyer et al. 2015), as well as the mixture modelling approach from the **mclust** *R* package (Fraley et al. 2017) (denoted *GMM*). These methods do not explicitly allow temporal dependence in the classification rules, and so we would expect them to perform poorly in cases where this dependence features in the test series. Specifically, we used a radial basis kernel for the SVM classifier. The GMM approach implemented allows for potentially different numbers of mixture components and covariance structures for each class, with the number of components chosen with the Bayesian information criterion (BIC). Similar to the HMM method

described above, we show results on the SVM and GMM methods applied to the transformed coherence measure—the results for the techniques on the raw series performed poorly and so they aren't reported in the tables. In addition, we compare our method to the Naïve Bayes (NB) classifier in the **RMOA** (Wijffels 2014) suite of online methods (again using the transformed coherence). This latter technique uses a Bayesian classification rule similar to that in (6), and hence provides a useful comparison to our proposed use of time-varying wavelet coherence in a Bayesian rule. We also investigated the performance of several of the ensemble classification techniques implemented in the **RMOA** package, however their performance was similar to the NB classifier so we omit these results for brevity.

Training procedure details. The training data for both the online dynamic classification and the sequential HMM approaches consists of ten signals of length 256. Of the ten signals, we simulate two each from Class 1, 2 and 3 and the remaining four signals contain a mixture of all three. For the competitor methods that are applied to the transformed coherence measure, the training data has a slightly different form. In this case, the training signals are simulated with class memberships as defined above but the approaches are trained on the transformed coherences of these signals at the set of discriminative indices rather than the raw data. For the different scenarios and generating processes considered, in practice we find that the subset of most discriminative indices tends to consist of the finest scales, i.e. scales 1–3, but that all channel indices appear to be important.

For each of the scenarios, 100 replications of the test signals are simulated and three different classification evaluation measures are considered. In particular, the number of class changes detected is recorded along with the V-measure (Rosenberg and Hirschberg 2007) and the true positive detection rate, defined to be the proportion of each signal that is correctly classified. A change is detected if the signal switches class and this change lasts for longer than four time points. The V-measure assesses the quality of a certain segmentation (given the truth) and is measured on the $[0, 1]$ scale where a value of 1 represents perfect segmentation.

The classification results for the different examples described above can be seen in Tables 1, 2 and 3. Sequential HMM denotes the results for the full test signal and Embedded HMM denotes the results for the transformed coherence; similar descriptors are used for the SVM, GMM and NB classifiers applied to the transformed coherence of the raw data. We remind the reader that these classification methods performed very poorly on the original series, and so are not reported in the tables. In each case, we have recorded the average number of changes detected, V-measure and true positive rate (described above) over the 100 replications; the numbers within the brackets represent the standard deviation of the corresponding quantities. Recall that the number of true class changes for Scenarios 1, 2 and 3 are nine, five and six respectively.

For the three class multivariate normal example (Table 1), it can be seen that all three methods overestimate the number of changes detected. The online dynamic classification performs the best in terms of the average number of changes detected, only marginally overestimating the number of changes, and is competitive with other methods in terms of V-measure and average true positive rate. Both the sequential HMM approach and the Naïve Bayes classification rule perform well in this setting

Table 1 Performance of classification procedures over 100 replications of multivariate Gaussian series for different scenarios of class changes, using the evaluation measures described in the text

Method	Scenario 1 (nine changes)	Scenario 2 (five changes)	Scenario 3 (six changes)
<i>Average number of changes detected</i>			
Online dynamic classification ($w = 256$)	9.38 (0.65)	5.58 (0.88)	6.19 (0.44)
Sequential HMM	10.44 (3.80)	6.71 (5.65)	10.96 (13.78)
Embedded HMM	11.90 (3.29)	8.49 (3.53)	15.29 (3.74)
Embedded SVM	13.55 (2.34)	8.50 (2.02)	8.02 (1.56)
Embedded GMM	17.88 (3.58)	15.45 (3.10)	29.16 (5.34)
Embedded NB	12.71 (2.27)	7.43 (1.50)	6.16 (0.42)
<i>Average V-measure</i>			
Online dynamic classification ($w = 256$)	0.89 (0.02)	0.89 (0.03)	0.94 (0.01)
Sequential HMM	0.94 (0.05)	0.93 (0.09)	0.94 (0.09)
Embedded HMM	0.78 (0.05)	0.74 (0.10)	0.80 (0.04)
Embedded SVM	0.80 (0.03)	0.82 (0.04)	0.91 (0.02)
Embedded GMM	0.81 (0.02)	0.73 (0.04)	0.75 (0.03)
Embedded NB	0.82 (0.06)	0.86 (0.03)	0.95 (0.01)
<i>Average true positive rate</i>			
Online dynamic classification ($w = 256$)	0.91 (0.02)	0.94 (0.02)	0.97 (0.01)
Sequential HMM	0.93 (0.11)	0.95 (0.11)	0.94 (0.13)
Embedded HMM	0.59 (0.09)	0.64 (0.13)	0.75 (0.10)
Embedded SVM	0.70 (0.05)	0.85 (0.05)	0.95 (0.02)
Embedded GMM	0.57 (0.06)	0.64 (0.05)	0.79 (0.05)
Embedded NB	0.75 (0.06)	0.88 (0.04)	0.97 (0.01)

Numbers in brackets represent the standard deviation of estimation errors. Bold numbers indicate best result

according to the V-measure and the average true positive rate. However, we note here that the improvement over our proposed method is minimal considering the variability in the estimates.

As we introduce dependence into the series, the distinction between our proposed method and its competitors becomes more marked. For the moving average process (Table 2), the performance of the online dynamic classification method improves as we increase the length of the segments between class changes; on the other hand, the sequential HMM procedure (as with the other competitors applied to the original series) cannot cope with the dependence in the data, drastically overestimating the number of changes in the data.

The online dynamic classification algorithm outperforms the competitors consistently for the autoregressive series, as shown in Table 3. More specifically, it classifies the changes well in terms of the V-measure and true positive rate, i.e. a low misclassification rate. Provided that the set of training data accurately represents the range of classes present, we would expect the dynamic classification approach to be able to correctly detect both the location of the changes and the classes involved. In contrast,

Table 2 Performance of classification procedures over 100 replications of vector moving average series for different scenarios of class changes, using the evaluation measures described in the text

Method	Scenario 1 (nine changes)	Scenario 2 (five changes)	Scenario 3 (six changes)
<i>Average number of changes detected</i>			
Online dynamic classification ($w = 256$)	9.82 (1.10)	5.78 (0.93)	6.59 (0.87)
Sequential HMM	35.75 (7.38)	33.75 (8.82)	77.24 (16.54)
Embedded HMM	10.28 (3.71)	8.69 (2.74)	14.57 (4.89)
Embedded SVM	11.90 (1.90)	5.96 (1.10)	9.18 (2.28)
Embedded GMM	13.31 (2.89)	14.21 (3.32)	18.09 (4.56)
Embedded NB	12.07 (1.63)	5.87 (0.68)	11.38 (2.36)
<i>Average V-measure</i>			
Online dynamic classification ($w = 256$)	0.87 (0.02)	0.89 (0.03)	0.94 (0.01)
Sequential HMM	0.76 (0.04)	0.66 (0.05)	0.64 (0.06)
Embedded HMM	0.75 (0.07)	0.73 (0.08)	0.79 (0.05)
Embedded SVM	0.85 (0.02)	0.88 (0.04)	0.87 (0.03)
Embedded GMM	0.81 (0.02)	0.71 (0.03)	0.81 (0.03)
Embedded NB	0.84 (0.02)	0.86 (0.03)	0.87 (0.03)
<i>Average true positive rate</i>			
Online dynamic classification ($w = 256$)	0.89 (0.03)	0.93 (0.02)	0.97 (0.01)
Sequential HMM	0.54 (0.15)	0.57 (0.14)	0.50 (0.15)
Embedded HMM	0.62 (0.08)	0.67 (0.11)	0.68 (0.05)
Embedded SVM	0.83 (0.03)	0.91 (0.03)	0.92 (0.03)
Embedded GMM	0.70 (0.05)	0.62 (0.05)	0.72 (0.03)
Embedded NB	0.81 (0.03)	0.92 (0.02)	0.90 (0.03)

Numbers in brackets represent the standard deviation of estimation errors. Bold numbers indicate best result

whilst the comparative methods detect the location of class changes well resulting in high V-measure, they can struggle to identify which class the signal belongs to after the class change has occurred, resulting in a lower true positive rate (a higher overall rate of misclassification). This can potentially be a challenge if accurate detection of anomalous areas is important.

In addition, note that in nearly all cases across the examples and scenarios, there is less variability in the evaluation measures using our proposed online dynamic classification (indicated by lower standard deviations). We also note here that the use of the coherence measure improves the performance of all competitor methods, justifying its efficacy as a classification feature in many settings. Crucially, we also found that the online dynamic classification approach was faster than HMM-based methods for longer time series.

Table 3 Performance of classification procedures over 100 replications of vector autoregressive series for different scenarios of class changes, using the evaluation measures described in the text

Method	Scenario 1 (nine changes)	Scenario 2 (five changes)	Scenario 3 (six changes)
<i>Average number of changes detected</i>			
Online dynamic classification ($w = 256$)	9.79 (0.96)	5.36 (0.66)	6.64 (0.87)
Sequential HMM	12.03 (5.49)	9.30 (4.46)	17.27 (6.44)
Embedded HMM	13.11 (2.70)	10.44 (3.16)	17.83 (6.60)
Embedded SVM	12.80 (3.14)	7.59 (2.03)	12.37 (2.78)
Embedded GMM	16.82 (3.04)	6.32 (2.75)	19.64 (4.40)
Embedded NB	14.64 (3.13)	6.87 (1.40)	11.91 (2.62)
<i>Average V-measure</i>			
Online dynamic classification ($w = 256$)	0.87 (0.02)	0.89 (0.03)	0.92 (0.02)
Sequential HMM	0.81 (0.08)	0.73 (0.09)	0.75 (0.07)
Embedded HMM	0.79 (0.03)	0.71 (0.08)	0.74 (0.06)
Embedded SVM	0.78 (0.03)	0.83 (0.05)	0.84 (0.03)
Embedded GMM	0.75 (0.02)	0.61 (0.07)	0.73 (0.04)
Embedded NB	0.78 (0.06)	0.84 (0.04)	0.84 (0.03)
<i>Average true positive rate</i>			
Online dynamic classification ($w = 256$)	0.89 (0.02)	0.95 (0.02)	0.96 (0.01)
Sequential HMM	0.70 (0.11)	0.69 (0.09)	0.65 (0.11)
Embedded HMM	0.59 (0.09)	0.60 (0.10)	0.59 (0.10)
Embedded SVM	0.65 (0.05)	0.89 (0.04)	0.89 (0.04)
Embedded GMM	0.51 (0.04)	0.45 (0.04)	0.51 (0.05)
Embedded NB	0.64 (0.06)	0.89 (0.03)	0.89 (0.03)

Numbers in brackets represent the standard deviation of estimation errors. Bold numbers indicate best result

5 Case study

In the previous section we considered the efficacy of our approach against tried and tested examples. We now turn to consider an application arising from our collaboration with researchers working in the oil and gas industry.

The general philosophy is to apply our online dynamic classification method to acoustic sensing data provided by an industrial collaborator, with the aim of detecting stripping within these signals. The training data consists of ten quadvariate signals of length 4096 obtained from a subsampled version of an acoustic sensing dataset. The class assignments for each of the training signals have been decided by an industrial expert. The test signal is obtained from the same dataset and is a quadvariate signal of length 8192, unseen in the training signals. The test series exhibits autocorrelation as well as dependence between series (see Fig. 2). Due to the zero-mean assumption of the mvLSW model, in practice we detrend the series before analysis by taking first order differences of each component series (see Fig. 1b).

We apply the online dynamic classification approach with a moving window of length 4096 to the test signal. Based on the results from Sect. 4, for comparison the sequential HMM method is applied to the full test signal with the first 400 data points used to train a two-state model. We also apply the sequential HMM approach to the transformed coherence of the test signal, again training a two state model using the first 400 data points. Two-state models have been applied to demonstrate our belief that the acoustic sensing data contains areas of stable behaviour and striping.

In this case the true class membership of the test signal is unknown, therefore we compare the results visually. The classification results for each of the methods are found in Fig. 4; areas of the test signal for which a change in class is detected are shown with lighter shading. It can be seen that the online dynamic classification method performs best in that it detects the stripes in the test signal with only minimal areas of misclassification when compared to the expert's judgement. In contrast, applying the sequential HMM approach to the transformed coherence of the signal also results in a change of class being detected at the stripes but the class changes take place over a longer period than we would expect. Finally, applying the sequential HMM method to the full signal results in the stripes being detected but the end of the test signal being misclassified.

Recalling that the overall aim of this analysis has been to detect sudden regions of interest within (multivariate) acoustic sensing signals, as accurately as possible whilst minimising the number of falsely detected points—then the results look very positive. Specifically the classification results obtained by the online dynamic classification method compare very favourably. It is interesting to note that in these examples, coarser scales (i.e. scales 6–11) appear to play a key role in the classification.

When compared with a subjective analysis of the data as displayed in Fig. 1 we see that, each method correctly assigns 'non-stripe' regions with the following (correct) classification proportions; 0.944 for online dynamic classification, 0.792 for embedded HMM and 0.477 for sequential HMM.

6 Concluding remarks

In this article, we introduced an online dynamic classification method that can be used to detect changes in class within a data stream. We demonstrated the efficacy of the method using simulated data examples and an acoustic sensing dataset from an oil producing facility. The case study shows that our approach can be successfully used to detect anomalous periods in acoustic data, resulting in fewer areas of misclassification compared to more traditional classification methods, such as Markov Model approaches. Moreover, we have found that the use of a coherence measure in classification improves the performance of these methods.

In practice, we have found that a parsimonious choice of window is required: as with other moving window approaches, too short a window, and the results are not satisfactory; too long a window increases the computational time and potentially produces edge effects. We leave the challenge of automatically choosing window length as an avenue for future research. In addition, we have observed that, as with other competitor methods, our approach classifies well when the distance between

Fig. 4 Classification results obtained from applying **a** online dynamic classification, **b** sequential HMM and **c** embedded HMM approaches to acoustic sensing data, areas of the signal for which a change in class is detected are shown with lighter shading

Fig. 5 Comparison of computational cost of the classification methods described in Sect. 4 in terms of their scaling behaviour with length of test series

class changes is comparable to the window length but can struggle when we have shorter segments between changes.

Future work may consider the problem of detecting stripes that are characterised by more gradual changes in their properties. In practice, these features may be less obvious and we might wish to not just detect but also classify the type of stripe present in the acoustic sensing data. Our method could potentially be used to do this, provided that our training data represents the range of stripes that we wish to classify.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

A Comparison of computational cost of online classification methods

In this section, we provide an analysis of the computational cost of the various competitor classification methods outlined in Sect. 4. To this end, we run each online classification method on a set of test signals of increasing length, namely $T = 1024, 2048, 4096, 8192$. In particular, for each method and series length, we record the runtime of each method, averaged over $K = 25$ replications of series from the first example in Sect. 4. This allows us to compare how the runtime of each method scales with series length, T , removing external factors such as efficiency of coding and implementation programming language.

The results of the runtime analysis are shown in Fig. 5. As seen from the plot, as expected, each method increases in runtime with the length of the series. However, after an initial increase, our dynamic online classification method has a desirable near constant scaling with the length of the series. Its scaling profile is the best after the

NB classifier. Given the improvement in classification over the competitor methods across the range of examples studied in Sect. 4, we feel that this profile justifies the use of the proposed method.

References

- Ainsleigh PL, Kehtarnavaz N, Streit RL (2002) Hidden Gauss–Markov models for signal classification. *IEEE Trans Signal Process* 50(6):1355–1367
- Aykroyd RG, Barber S, Miller LR (2016) Classification of multiple time signals using localized frequency characteristics applied to industrial process monitoring. *Comput Stat Data Anal* 94:351–362
- Bifet A, Holmes G, Kirkby R, Pfahringer B (2010) MOA: massive online analysis. *J Mach Learn Res* 11:1601–1604
- Bifet A, Read J, Žliobaitė I, Pfahringer B, Holmes G (2013) Pitfalls in benchmarking data stream classification and how to avoid them. In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer, Berlin, pp 465–479
- Cappé O, Moulines E, Rydén T (2009) Inference in hidden markov models. In: *Proceedings of EUSFLAT conference*, pp 14–16
- Chan KP, Fu AWC (1999) Efficient time series matching by wavelets. In: *Proceedings of the 15th international conference on data engineering*. IEEE, pp 126–133
- Chau J, von Sachs R (2016) Functional mixed effects wavelet estimation for spectra of replicated time series. *Electron J Stat* 10(2):2461–2510
- Coifman RR, Donoho DL (1995) Translation-invariant de-noising. In: Antoniadis A, Oppenheim G (eds) *Wavelets and statistics*. Springer, Berlin, pp 125–150
- Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
- Dahlhaus R (1997) Fitting time series models to non-stationary processes. *Ann Stat* 25(1):1–37
- Dahlhaus R, Neumann MH, von Sachs R (1999) Nonlinear wavelet estimation of time-varying autoregressive processes. *Bernoulli* 5(5):873–906
- Daubechies I (1992) *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, Philadelphia
- Eckley IA, Nason GP (2005) Efficient computation of the discrete autocorrelation wavelet inner product matrix. *Stat Comput* 15(2):83–92
- Ellmauthaler A, Willis ME, Wu X, LeBlanc M (2017) Noise sources in fiber-optic distributed acoustic sensing vsp data. In: *79th eage conference and exhibition 2017*
- Ephraim Y, Merhav N (2002) Hidden Markov processes. *IEEE Trans Inf Theory* 48(6):1518–1569
- Fraley C, Raftery AE, Scrucca L, Murphy TB, Fop M (2017) mclust: Gaussian mixture modelling for model-based clustering, classification, and density estimation. R package version 5.4. <http://cran.r-project.org/package=mclust>. Accessed 3 Oct 2017
- Fryzlewicz P, Ombao HC (2009) Consistent classification of non-stationary time series using stochastic wavelet representations. *J Am Stat Assoc* 104(485):299–312
- Fu T-C (2011) A review on time series data mining. *Eng Appl Artif Intell* 24(1):164–181
- Himmelman L (2010) HMM: HMM-Hidden Markov Models. R package version 1.0. <https://cran.r-project.org/package=HMM>. Accessed 3 Oct 2017
- Huang HY, Ombao HC, Stoffer DS (2004) Discrimination and classification of nonstationary time series using the SLEX model. *J Am Stat Assoc* 99(467):763–774
- Kampouraki A, Manis G, Nikou C (2009) Heartbeat time series classification with support vector machines. *IEEE Trans Inf Technol Biomed* 13(4):512–518
- Kersten J (2014) Simultaneous feature selection and Gaussian mixture model estimation for supervised classification problems. *Pattern Recognit* 47(8):2582–2595
- Killick R, Eckley IA, Jonathan P (2013) A wavelet-based approach for detecting changes in second order structure within nonstationary time series. *Electron J Stat* 7:1167–1183
- Krzemieniewska K, Eckley IA, Fearnhead P (2014) Classification of non-stationary time series. *Stat* 3(1):144–157
- Mallat SG (1989) A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans Pattern Anal Mach Intell* 11(7):674–693

- Mateeva A, Lopez J, Potters H, Mestayer J, Cox B, Kiyashchenko D, Wills P, Grandi S, Hornman K, Kuvshinov B, Berlang W (2014a) Distributed acoustic sensing for reservoir monitoring with vertical seismic profiling. *Geophys Prospect* 62(4):679–692
- Mateeva A, Lopez J, Potters H, Mestayer J, Cox B, Kiyashchenko D, Wills P, Grandi S, Hornman K, Kuvshinov B, Berlang W, Zhaohui Y, Detomo R (2014b) Distributed acoustic sensing for reservoir monitoring with vertical seismic profiling. *Geophys Prospect* 62(4):679–692
- McLachlan G, Peel D (2004) *Finite mixture models*. Wiley, New York
- Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F, Chang CC, Lin CC (2015). e1071: misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien. R package version 1.6-7. <http://cran.r-project.org/package=e1071>. Accessed 3 Oct 2017
- Mörchen F (2003) Time series feature extraction for data mining using DWT and DFT, technical report 33. University of Marburg
- Muller KR, Mika S, Ratsch G, Tsuda K, Scholkopf B (2001) An introduction to kernel-based learning algorithms. *IEEE Trans Neural Netw* 12(2):181–201
- Nam CFH, Aston JAD, Eckley IA, Killick R (2015) The uncertainty of storm season changes: quantifying the uncertainty of autocovariance changepoints. *Technometrics* 57(2):194–206
- Nason GP (2008) *Wavelet methods in statistics with R*. Springer, New York
- Nason GP (2016) *Wavethresh: wavelets statistics and transforms*. R package version 4.6.8. <https://cran.r-project.org/package=wavethresh>. Accessed 3 Oct 2017
- Nason GP, Silverman BW (1995) The stationary wavelet transform and some statistical applications. *Lect Notes Stat* 103:281–300
- Nason GP, Von Sachs R, Kroisandt G (2000) Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum. *J R Stat Soc Ser B* 62(2):271–292
- Nason GP, Powell B, Elliott D, Smith PA (2017) Should we sample a time series more frequently? Decision support via multirate spectrum estimation. *J R Stat Soc Ser A (Stat Soc)* 180(2):353–407
- Owen A, Duckworth G, Worsley J (2012) OptaSense: fibre optic distributed acoustic sensing for border monitoring. In: *Intelligence and security informatics conference (EISIC), 2012 European*. IEEE, pp 362–364
- Paleja R, Mustafina D, Park T, Randell D, van der Horst J, Crickmore R, et al (2015) Velocity tracking for flow monitoring and production profiling using distributed acoustic sensing. In: *SPE annual technical conference and exhibition*. Society of Petroleum Engineers
- Park T, Eckley IA, Ombao HC (2014) Estimating time-evolving partial coherence between signals via multivariate locally stationary wavelet processes. *IEEE Trans Signal Process* 62(20):5240–5250
- Park T, Eckley IA, Ombao HC (2018) Dynamic classification using multivariate locally stationary wavelets. *Signal Process* (accepted for publication). <https://doi.org/10.1016/j.sigpro.2018.01.005>
- Percival DP (1995) On estimation of the wavelet variance. *Biometrika* 82(3):619–631
- Povinelli RJ, Johnson MT, Lindgren AC, Ye J (2004) Time series classification using Gaussian mixture models of reconstructed phase spaces. *IEEE Trans Knowl Data Eng* 16(6):779–783
- Priestley MB (1981) *Spectral analysis and time series*. Academic Press, London
- Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE* 77(2):257–286
- Read J, Bifet A, Holmes G, Pfahringer B (2012) Scalable and efficient multi-label classification for evolving data streams. *Mach Learn* 88(1–2):243–272
- Rosenberg A, Hirschberg J (2007) V-measure: a conditional entropy-based external cluster evaluation measure. In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (emnlp-conll)*, vol. 7. pp 410–420
- Sakiyama K, Taniguchi M (2004) Discriminant analysis for locally stationary processes. *J Multivar Anal* 90(2):282–300
- Sanderson J, Fryzlewicz P, Jones MW (2010) Estimating linear dependence between nonstationary time series using the locally stationary wavelet model. *Biometrika* 97(2):435–446
- Taylor S, Park T, Eckley IA, Killick R (2017) *mvLSW: Multivariate locally stationary wavelet process estimation*. R package version 1.2.1. <https://cran.r-project.org/package=mvLSW>. Accessed 3 Oct 2017
- Vidakovic B (1999) *Statistical modeling by wavelets*. Wiley, New York
- Wei L, Keogh E (2006) Semi-supervised time series classification. In: *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, pp 748–753
- Wijffels J (2014) *RMOA: connect R with MOA to perform streaming classifications*. R package version 1.0. <https://github.com/jwijffels/RMOA>. Accessed 3 Oct 2017

- Williams J (2012) Distributed acoustic sensing for pipeline monitoring. *Pipeline Gas J* 239(7)
- Zhang H, Ho TB, Lin MS (2004) A non-parametric wavelet feature extractor for time series classification. In: Dai H, Srikant R, Zhang C (eds) *Advances in knowledge discovery and data mining*. Springer, Berlin, pp 595–603
- Zucchini W, MacDonald IL (2009) *Hidden Markov models for time series: an introduction using R*. Chapman and Hall-CRC, New York

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.