

CHAPTER 4

HYPERTEMPORAL TECHNIQUES

The chapter provides the technical details of all the sequential and non-sequential hypertemporal classification and change detection algorithms that were investigated in this thesis. The chapter is divided into three main sections, namely *simulation* (Section 4.1), *classification* (Section 4.2) and *change detection* (Section 4.3).

Simulation is the creation of synthetic data in such a way that the synthetic data accurately represent real world data or phenomena. Simulated datasets are used for algorithm development, testing and validation, as well as for optimising instrument specifications. Simulated data are a valuable tool and is often used by the remote sensing community [26, 68]. Most remote sensing simulators are constructed by using a deductive approach, which means that they rely on the biophysical laws that govern the reflection of light [26, 27]. In Section 4.1.2, an inductive multispectral hypertemporal reflectance simulator is proposed. In contrast to deductive simulators, an inductive simulator uses a mathematical model that is built from the statistical properties of an existing dataset. The fact that an inductive model is built up from the statistical properties of an existing dataset enables an inductive model to augment datasets. The inductive simulator from Section 4.1.2 will be used to generate data for the data-intensive CUSUM algorithm presented in Section 4.3.3. The inductive model that will be used consists of two components, namely an SHO [3] (Section 4.1.1) to model the deterministic underlying noise-free signal and the Ornstein-Uhlenbeck process [4] (Section 4.1.2.1) to model the residual after the SHO has been subtracted. The two-component model will be referred to, in this chapter, as the CSHO [2], which is discussed in detail in Section 4.1.2.2. The possibility of using the parameters of the CSHO model as features for classification is discussed in Section 4.2.4.2.

Classification is the act of arranging or organising according to class or category. Land cover classi-

fication using remotely sensed data is a critical first step in large-scale environmental monitoring, resource management and regional planning [14]. A good review of different classification approaches is given in [14]. As mentioned in Chapter 1 the thesis focuses on hypertemporal classifiers. The minimum distance classifier [16], the time-varying maximum likelihood classifier [23], and the three feature groups \mathbf{l} , θ and ζ are discussed in Section 4.2.2, Section 4.2.3 and Section 4.2.4.2 respectively. The classification results obtained after applying these approaches to the datasets in Section 2.8 can be found in Chapter 5.

Change detection is the process of identifying differences in the state of an object or phenomenon by observing it at different times. Essentially, it involves the ability to quantify temporal effects using multitemporal data [69]. There have been quite a number of reviews on change detection in the remote sensing field, namely [13, 69–73]. As mentioned in Chapter 1 the thesis also focuses on hypertemporal change detection techniques. The band differencing algorithm [7] and the windowless CUSUM algorithm [6] are discussed in Section 4.3.2 and Section 4.3.3 respectively. The change detection results obtained after applying the aforementioned techniques to the datasets in Section 2.8 are presented in Chapter 5.

4.1 SIMULATION

As stated in the previous section, most remote sensing simulators are constructed by using a deductive approach, which means that they employ the biophysical laws that govern the reflection of light [26, 27]. The simulator proposed in this chapter, however uses an inductive approach. An inductive approach tries to fit a mathematical model on the observed time-series directly, which is then used to simulate realistic reflectance values. The CSHO simulator proposed in this chapter is based on a stochastic inductive model. A stochastic inductive model tries to model the observed stochastic process, not just the noise-free underlying signal. The proposed simulator is not the first such approach used in the remote sensing literature [31]. Usually it is applied to a single time-series to enable forecasting, as is the case in [31]. The proposed simulator supplements [31], by making the concurrent simulation of multiple dependent time-series (multispectral) possible. On the other side of the inductive spectrum lies the complementary noise-free inductive models [74], which are used for noise reduction. The noise-free signals are then used to extract phenological markers. These two inductive approaches do not compete against each other, since they have different aims, noise reduction versus time-series generation.

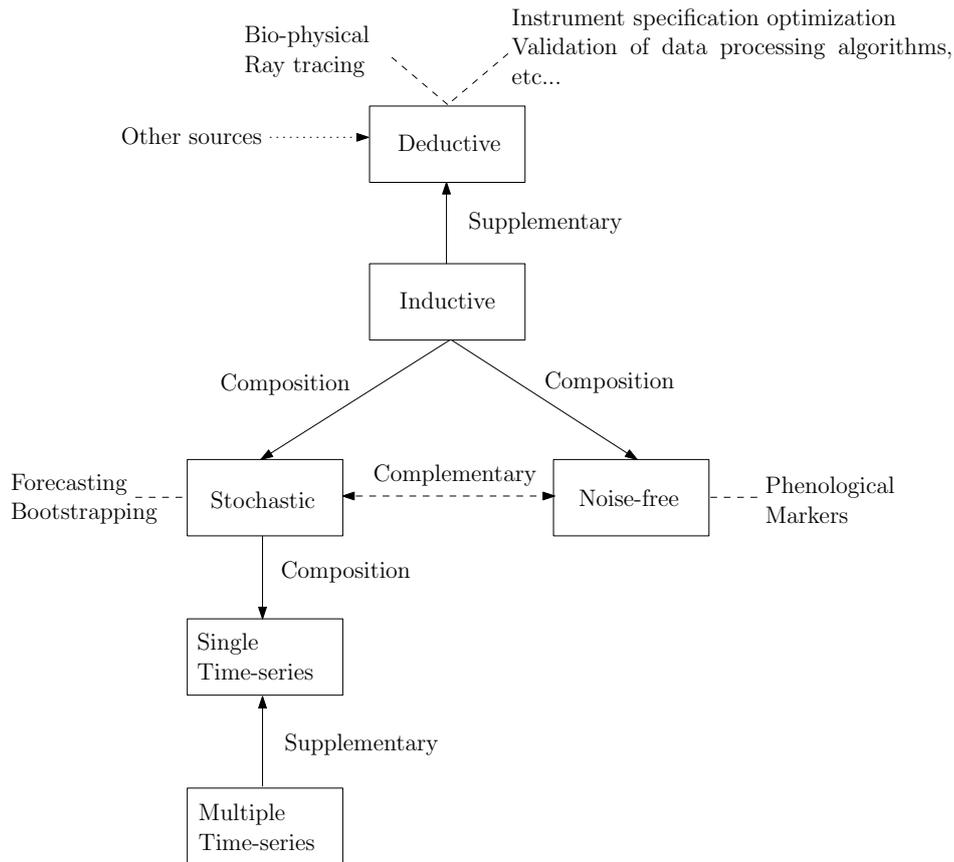


Figure 4.1: Functional (scientific) positioning of the proposed simulator.

The deductive simulators that were mentioned earlier use bio-physical parameters such as chlorophyll content, which can be derived from ancillary sources (for example direct measurement). An inductive simulator is an example of one such an ancillary source (in specific applications). For example, an inductive simulator could be used to forecast Leaf Area Index (LAI), which is a parameter required by the PROSPECT + Scattering by Arbitrary Inclined Leaves (PROSAIL) deductive simulator [27, 75]. Since an inductive simulator is actually a possible deductive simulator input source, they are not directly comparable. It is important to point out that inductive simulators are supplementary tools when used with deductive simulators, as they are not required by deductive simulators, which can function independently from inductive simulators. Figure 4.1 illustrates the scientific positioning of the simulator proposed in this chapter relative to existing simulators and models.

4.1.1 Noise-free inductive models

In this section, a short overview will be given of some of the different noise-free inductive models that are currently in use. The proposed stochastic inductive simulator uses an underlying inductive noise-free model (deterministic part) as its base. To make the simulator stochastic, a stochastic model is added to the deterministic base, which is the primary differentiating factor between noise-free modelling and stochastic modelling. In stochastic modelling the statistical properties of an observed class are replicated, while deterministic modelling wants to determine the shape of the underlying noise-free signal, and as such provides complementary functionality. The stochastic model does not necessarily require the best possible underlying noise-free model, as long as the model used for the residual preserves the statistical properties of the original signal.

The SHO model is an example of a noise-free inductive model [3] and is given by

$$A \sin(2\pi f_s t + \phi) + C, \quad (4.1)$$

where

$$\{A, C\}$$

are the harmonic features proposed by [5, 10] and $T_s = \frac{1}{f_s}$ is the period of the model. Many other models have been proposed as an improvement on the SHO model [31, 74, 76–78].

In particular, Carrão *et al.* [74] modelled MODIS time-series with a harmonic non-linear solution of a chaotic attractor

$$C + A \cos(2\pi f_s t + \phi + \alpha \cos(2\pi f_s t + \zeta)).$$

The function of each parameter used by Carrão's model is discussed below:

- C is a linear parameter that represents the annual mean of the model.
- A is the amplitude for the sine wave that fixes the peak deviation from the annual mean of the model.
- ϕ is the annual phase (produces a specific season of a given land cover class).
- α controls the non-linear strength of the model. When $\alpha = 0$, the model reduces to a simple harmonic oscillator, whereas $\alpha > 0$ introduces non-symmetry (bi-annual behaviour) in the model.

- ζ is the annual nonlinear phase. This phase allows time to “slow down” and to “accelerate” in order to reproduce asymmetries in variations (increases versus decreases).

Figure 4.2 illustrates the effect of the model parameters, as well as some of the different wave shapes that Carrão’s model can represent.

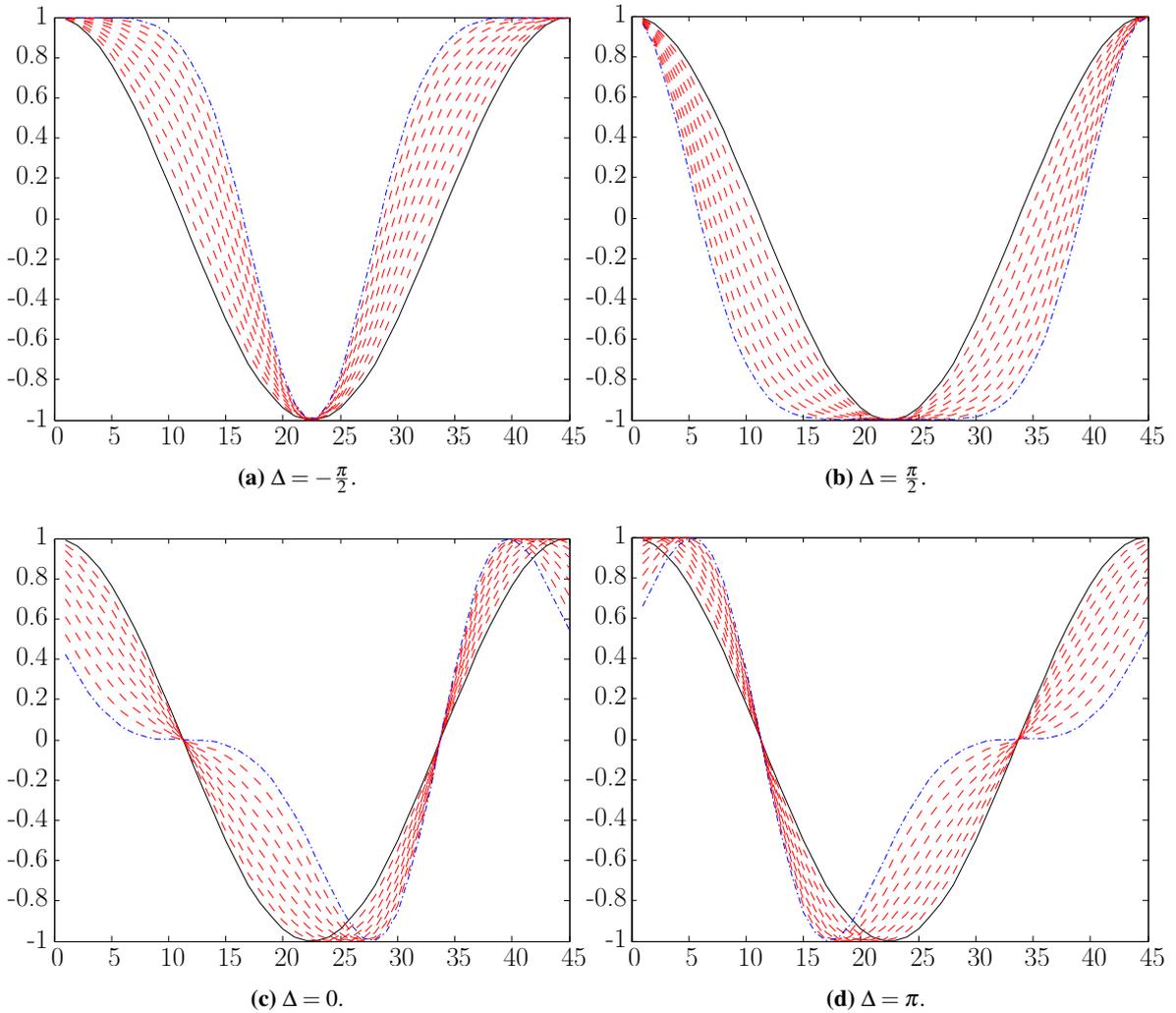


Figure 4.2: The equation for Carrão’s model is, $C + A \cos(2\pi f_s t + \phi + \alpha \cos(2\pi f_s t + \zeta))$ and $\Delta = \phi - \zeta$. The blue line is when $\alpha = 1$, while the black line is when $\alpha = 0$ and the red lines are for $0 < \alpha < 1$. When the parameter $\alpha > 1$ Carrão’s model will exhibit bi-annual variation. To get the specific graphs $\phi = 0$, which only functions as a translation parameter.

Kleynhans *et al.* [78] modelled NDVI time-series with a triply modulated cosine function

$$A(t) \sin(2\pi f_s t + \phi(t)) + C(t).$$

Jönsson *et al.* [76] modelled vegetation index time-series using Asymmetric Gaussian (AG) functions, while Zhang *et al.* [77] used piecewise-defined local Double Logistic (DL) functions. The AG and DL global model functions both have the following form

$$y(t) = \sum_{h=1}^H c_{1,h} + c_{2,h}g(t;A_h),$$

where H denotes the number of local model functions to use. The linear parameters $c_{1,h}$ and $c_{2,h}$ respectively govern the mean and the amplitude of the local function $g(t,A_h)$, while the meta-parameter $A_h = \{a_1, \dots, a_r\}$ determines the shape of the local function $g(t,A_h)$. The AG local function is equal to

$$g(t, a_1, \dots, a_5) = \begin{cases} \exp \left[-1 \left(\frac{t-a_1}{a_2} \right)^{a_3} \right], & \text{if } t \geq a_1 \\ \exp \left[-1 \left(\frac{a_1-t}{a_4} \right)^{a_5} \right], & \text{if } t < a_1. \end{cases} \quad (4.2)$$

In Equation 4.2, a_1 determines the position of the maxima (or minima) of g , while a_2 and a_3 determine the width and flatness of the right half of the function g . Similarly a_4 and a_5 determine the width and flatness of the left half of the function g . The DL local function is represented by

$$g(t; a_1, \dots, a_4) = \frac{1}{1 + \exp \left(\frac{a_1-t}{a_2} \right)} - \frac{1}{1 + \exp \left(\frac{a_3-t}{a_4} \right)},$$

where a_1 and a_3 determine, respectively, the position of the left and right inflection points and a_2 and a_4 fix the rates of change at those points. The global AG and DL functions therefore respectively require $H \times 7$ and $H \times 6$ parameters.

Verbesselt *et al.* [19,20] proposed a seasonal-trend model

$$C_1 + C_2t + \sum_{j=1}^k A_j \sin(2\pi j f_s t + \phi_j),$$

where C_1 is the mean of the model, C_2 is the slope of the linear trend and A_j and ϕ_j are responsible for reproducing the seasonal behaviour.

From all the inductive models discussed up to this point the SHO was selected as the deterministic base of the proposed CSHO simulator. It is a well-known fact that the SHO model is a good first order noise-free model of a remotely sensed time-series [3].

A short discussion follows below to justify the SHO as the underlying noise-free model for the proposed simulator. In the discussion, reasons are provided for not selecting the other noise-free inductive models (in this section). Kleynhans's model is not a possible candidate for the deterministic base of the proposed simulator, since it is not a parsimonious model. The stochastic model that will be used

in the end should be parsimonious so that it can also be used to extract classification features. The seasonal-trend model is also not suitable, as the trend term of the model implies that the model should be used with a window. The model used in the end should be parsimonious and be able to simulate a multi-year time-series. It has been shown that Carrão's model is better than the remaining models, i.e. better than AG and DL, as using Carrão's model leads to lower fitting errors [74]. Carrão's model can be used as a simulator by adding white noise to it. All of the models mentioned (in particular Carrão's model) are definitely more accurate than the SHO model over a one-year window, but are also computationally more intensive than the Fourier transform used by the SHO.

In particular Carrão's method uses phase unwrapping, Levenberg-Marquardt $\times 2$ and Ordinary Least Squares (OLS) as functional blocks for estimating the parameters of the model [79, 80]. When the time-series becomes multi-year and there is inter-annual variation in the data, the long-term fitting-error made by the SHO is on average far less when compared to most of the other shapes that can be produced by Carrão's model (the SHO is one of the shapes Carrão's model can produce and is obtained when $\alpha = 0$). A summary of some of the shapes Carrão's model can generate can be found in Figure 4.2.

In other words, when the time-series becomes multi-year the SHO is actually a very good model candidate, while the extra versatility offered by Carrão's model becomes redundant (especially if the first harmonic component dominates the remaining harmonic components). In the case of multi-year time-series the increased accuracy (if any when compared to an SHO owing to the possibility of local minima, which is relevant for Levenberg-Marquardt), benefit obtained by using Carrão's model no longer outweighs the computational cost of the parameter estimation technique used by Carrão's method (compared to the SHO). It is also important to realize that when Carrão's model is used on each year individually, its parsimoniousness is compromised.

4.1.2 Proposed simulator

The proposed simulator uses the CSHO model. The CSHO consists of two components, a deterministic component and a stochastic component. The SHO is used for the deterministic component, while the Ornstein-Uhlenbeck process is used for the stochastic component. The Ornstein-Uhlenbeck process is used to model the remaining residual after the SHO is subtracted from the observed time-series. As the SHO is very general, there will be a high degree of dependence between the observations of the residual. The Ornstein-Uhlenbeck process can model a time-series with dependent observations

(first order), since this process is the continuous-time analogue of the discrete-time AR(1) process. The dependence implies colouredness, which is where the name of the simulator comes from. The Ornstein-Uhlenbeck process can be used to generate coloured noise as well as white noise [81]. The harmonic parameters of the SHO are estimated with the Fourier transform, while the parameters of the Ornstein-Uhlenbeck process are estimated with maximum likelihood parameter estimation. The objective of the CSHO simulator is to simulate multispectral time-series with an inherent correlation structure. In this thesis the simulator is used to augment datasets for data-intensive classification and change detection algorithms (Section 4.3.3). In selective cases, statistical inductive models similar to the CSHO have been used to forecast a single time-series [31]. The complex issue of incorporating multispectral correlation into a simulator was however not addressed in [31]. The CSHO simulator incorporates the average class noise correlation between the different spectral bands and reproduces class-specific spectral behaviour (spectral dependence) by enforcing the statistical restrictions imposed by different parameters (like mean and seasonal amplitude) of each spectral band in a class on one another.

In Section 4.1.2.1 the Ornstein-Uhlenbeck process is discussed, which is followed by Section 4.1.2.2 that discusses the CSHO in detail. Section 4.1.2.3 describes the algorithm used to estimate the parameters of the CSHO. The algorithm for simulating a MODIS pixel with the CSHO is presented in Section 4.1.2.7. The details of how the CSHO simulator enforces spectral dependence and correlation are presented in Section 4.1.2.4, Section 4.1.2.5 and Section 4.1.2.6.

4.1.2.1 Ornstein-Uhlenbeck

The Ornstein-Uhlenbeck process is widely used in mathematical finance for the modelling of the dynamics of interest rates and volatilities of asset prices. The Ornstein-Uhlenbeck process is the continuous-time analogue of the discrete time AR(1) process and, when initialised with the equilibrium distribution, is also stationary, Gaussian, Markovian and mean reverting. A stochastic process $\eta(t)$ is

- stationary if, for all $t_1 < t_2 < \dots < t_n$ and $h > 0$, the random n -vectors $(\eta(t_1), \eta(t_2), \dots, \eta(t_n))$ and $(\eta(t_1 + h), \eta(t_2 + h), \dots, \eta(t_n + h))$ are identically distributed;
- Gaussian if, for all $t_1 < t_2 < \dots < t_n$, the n -vector $(\eta(t_1), \eta(t_2), \dots, \eta(t_n))$ is multi-variate normally distributed;

- Markovian if, $\forall B \in \mathbb{R}$ and for all $t_1 < t_2 < \dots < t_n$, $P(\eta(t_n) \leq B | \eta(t_1), \eta(t_2), \dots, \eta(t_{n-1})) = P(\eta(t_n) \leq B | \eta(t_{n-1}))$ (in lay man terms it means that the future is determined only by the present and not the past).

Moreover, the Ornstein-Uhlenbeck stochastic process satisfies the following stochastic differential equation:

$$d\eta(t) = \lambda(\mu - \eta(t))dt + \sigma dW(t), \quad (4.3)$$

where $\lambda > 0$ is the rate of mean reversion, μ is the long-term mean of the stochastic process, $\sigma > 0$ is the volatility or average magnitude, per square-root time, of the random fluctuations and $W(t)$ is a standard Brownian motion on $t \in [0, \infty]$, implying that $dW(t) \sim \mathcal{N}(0, \sqrt{dt})$. The solution to Equation 4.3 is given by

$$\eta(t) = \eta(0)e^{-\lambda t} + \mu(1 - e^{-\lambda t}) + \int_0^t \sigma e^{\lambda(s-t)} dW(s),$$

where the integral on the right-hand side is an Itô integral. The equilibrium density of the Ornstein-Uhlenbeck process is equal to $\mathcal{N}(\mu, \frac{\sigma^2}{2\lambda})$. If the random fluctuations in the process are ignored, it becomes clear that $\eta(t)$ has an overall drift towards the process mean μ . The process $\eta(t)$ reverts to the mean exponentially, at a rate λ , with a magnitude in direct proportion to the distance between the current value of $\eta(t)$ and μ [82].

4.1.2.2 Coloured Simple Harmonic Oscillator

Let $\mathbf{x}_c(t) = \{x_c^b(t)\}_{b \in \{1, \dots, 7\}}$ denote a MODIS pixel at time t with assigned class label $c \in \mathcal{C}$, where $x_c^b(t)$ denotes the b^{th} spectral band's reflectance at time t . The c is omitted if the class of the MODIS pixel is unknown.

Each observed signal belonging to the same class is a sample path of a stochastic process $X_c^b(t)$. Each MODIS class c is therefore modelled as a set of correlated (spectrally) stochastic processes $\mathbf{X}_c(t) = \{X_c^b(t)\}_{b \in \{1, \dots, 7\}}$. Since $X_c^b(t)$ is a stochastic process, an analytic expression can be assigned (if such an expression exists) to each sample path (MODIS pixel) $x_c^b(t; \boldsymbol{\theta}_c^b)$ of $X_c^b(t)$, where $\boldsymbol{\theta}_c^b$ is a set of random values with a joint probability density function. It is important to realise that real world MODIS pixels are also spatially correlated, while the proposed model assumes spatial independence.

The proposed analytic expression for each MODIS pixel in each band (sample path) is given by

$$x_c^b(t; \theta_c^b) = s_c^b(t; \{A_c^b, \phi_c^b, C_c^b\}) + \eta_c^b(t; \{\mu_c^b, \lambda_c^b, \sigma_c^b\}), \quad (4.4)$$

where $s_c^b(t; \{A_c^b, \phi_c^b, C_c^b\})$ is the SHO model given in Equation 4.1 with period $T_s = \frac{1}{f_s} = 45$. The noise process $\eta_c^b(t; \{\mu_c^b, \lambda_c^b, \sigma_c^b\})$ is an Ornstein-Uhlenbeck process that satisfies the stochastic differential equation given in Equation 4.3.

For each class and band, it is expected that μ_c^b will be insignificant relative to C_c^b , as $\mu_c^b = 0$ if the parameter C_c^b is estimated without error. For convenience θ_c^b will sometimes be omitted from $x_c^b(t; \theta_c^b)$.

The distribution of θ_c^b is determined by the parameter set $\{A_c^b, \phi_c^b, C_c^b, \lambda_c^b, \sigma_c^b\}$ and it follows that $\theta_c = \{\theta_c^b\}_{b \in \{1, \dots, 7\}} = \{A_c^b, \phi_c^b, C_c^b, \lambda_c^b, \sigma_c^b\}_{b \in \{1, \dots, 7\}} = \{\theta_1, \dots, \theta_{35}\}$. The probability density function associated with θ_c is denoted with $f_c(\theta_c)$. When NDVI is included in the parameter set the notation $\tilde{\theta}_c$ will be used. The same convention applies for $\tilde{\mathbf{X}}_c(t)$ and $\tilde{\mathbf{x}}_c(t)$. NDVI is excluded when constructing the probability density function $f_c(\theta_c)$, since NDVI must be constructed from bands 1 and 2. The notation for a MODIS pixel (plus NDVI) is represented graphically in Figure 4.3 (where $\tilde{\mathbf{x}}[i]$ is the discrete analogue of $\tilde{\mathbf{x}}(t)$).

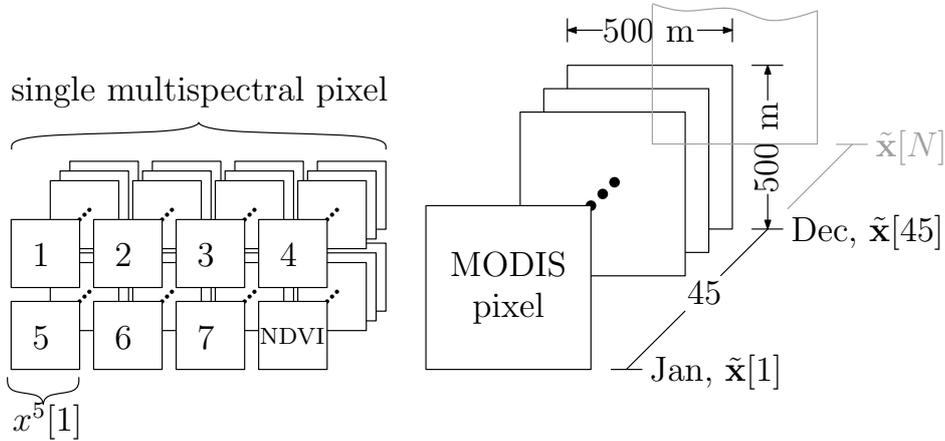


Figure 4.3: Time-series data representation for a single pixel, plus NDVI [2] © IEEE 2012.

The ensemble mean for $\tilde{\mathbf{X}}_c(t)$ is defined as

$$\tilde{\mathbf{y}}_c(t) = \{\mathbb{E}[X_c^b(t)]\}_{b \in \{1, \dots, 7, \text{NDVI}\}}, \quad (4.5)$$

and is assumed to be periodic, i.e. $\tilde{\mathbf{y}}_c(t) = \tilde{\mathbf{y}}_c(t + 45j)$, $\forall j \in \mathbb{N}$.

The autocorrelation of $\tilde{\mathbf{x}}_c(t)$ is defined as $\tilde{\mathbf{R}}_c(\tau) = \{\mathcal{R}_c^b(\tau)\}_{b \in \{1, \dots, 7, \text{NDVI}\}}$, where

$$\mathcal{R}_c^b(\tau) = \frac{(x_c^b(t) - \mathbb{E}[x_c^b(t)])(x_c^b(t + \tau) - \mathbb{E}[x_c^b(t)])}{\text{var}(x_c^b(t))}. \quad (4.6)$$

Although the spatial correlation is not fully incorporated into the CSHO model it can still be quantified with the same notation. The spatial correlation of class c in band $b \in \{1, \dots, 7, \text{NDVI}\}$ can be represented with a correlation matrix $\boldsymbol{\rho}_b^c$, with elements

$$\rho_{b_{m,n}}^c = \frac{\mathbb{E}[(x_{m,c}^b(t) - \mathbb{E}[x_{m,c}^b(t)])(x_{n,c}^b(t) - \mathbb{E}[x_{n,c}^b(t)])]}{\text{std}(x_{m,c}^b(t))\text{std}(x_{n,c}^b(t))},$$

where $x_{m,c}^b(t)$ is the m -th pixel in a set of P MODIS pixels belonging to class c . The average spatial correlation is then equal to

$$\tilde{\boldsymbol{\rho}}^c = \mathbb{E}\{\{\boldsymbol{\rho}_b^c\}_{b \in \{1, \dots, 7, \text{NDVI}\}}\}. \quad (4.7)$$

The CSHO does enforce a limited amount of spatial correlation through $f_c(\boldsymbol{\theta}_c)$ (for instance the sample paths of the CSHO pixels are reasonably in phase, have slight differences in long-term mean and seasonal amplitude). As such CSHO pixels are less correlated (spatially) than the actual MODIS pixels.

4.1.2.3 Parameter estimation

To estimate the harmonic parameters of Equation 4.4 the Fourier transform is used, while the noise parameters will be estimated via maximum-likelihood parameter estimation. The Fourier transform \mathcal{F} of an observed MODIS pixel $\tilde{\mathbf{x}}(t)$ is defined as

$$\tilde{\mathbf{X}}(f) = \{\mathcal{F}[x^b(t)]\}_{b \in \{1, \dots, 7, \text{NDVI}\}}.$$

The subscript c is omitted here, since the class to which the MODIS pixel belongs is unknown.

For each band b the harmonic parameters $\{\hat{A}^b, \hat{\phi}^b, \hat{C}^b\}$ are estimated as follows:

$$\begin{aligned} \hat{A}^b &= 2|\mathcal{F}[x^b(t)](f_s)| \\ \hat{\phi}^b &= \arg(\mathcal{F}[x^b(t)](f_s)) \\ \hat{C}^b &= |\mathcal{F}[x^b(t)](0)|. \end{aligned}$$

In practice $\hat{\phi}^b$ is calculated by using a minimum squared error sinusoidal fit. The mean and amplitude of the sinusoid used to calculate $\hat{\phi}^b$ are set to \hat{A}^b and \hat{C}^b respectively.

The parameters $\hat{\mu}^b, \hat{\lambda}^b$ and $\hat{\sigma}^b$ for $x^b(t)$ are estimated by using maximum likelihood parameter estimation. The first step is to calculate the residual by using

$$\hat{\eta}^b(t) = x^b(t) - \hat{A}^b \sin(2\pi ft + \hat{\phi}^b) + \hat{C}^b.$$

Now let $\eta^b[i]$ be the discrete time analogue of $\eta^b(t)$, with Δt being the time step of $\eta^b[i]$, i.e. $t = i\Delta t$, and I the total number of discrete time samples that are available of $\eta^b(t)$. The log-likelihood function of $\eta^b[i]$ is given by [83]

$$\begin{aligned} L(\mu^b, \lambda^b, \bar{\sigma}^b) &= -\frac{I}{2} \ln(2\pi) - I \ln(\bar{\sigma}^b) - \dots \\ &\dots - \frac{1}{2(\bar{\sigma}^b)^2} \sum_{i=1}^I [\eta^b[i] - \eta^b[i-1]\alpha^b - \mu^b(1 - \alpha^b)]^2, \end{aligned} \quad (4.8)$$

where

$$(\bar{\sigma}^b)^2 = (\sigma^b)^2 \frac{1 - e^{2\alpha^b}}{2\lambda^b} \quad (4.9)$$

and

$$\alpha^b = e^{-\lambda^b \Delta t}. \quad (4.10)$$

By respectively setting the partial derivative of Equation 4.8 with respect to $\mu^b, \lambda^b, \bar{\sigma}^b$ equal to 0 and respectively solving for $\mu^b, \lambda^b, \bar{\sigma}^b$, such that μ^b is independent of λ^b and $\bar{\sigma}^b$, the following maximum likelihood estimators are obtained

$$\begin{aligned} \hat{\mu}^b &= \frac{\eta_l \eta_{kk} - \eta_k \eta_{kl}}{I(\eta_{kk} - \eta_{kl}) - (\eta_k^2 - \eta_k \eta_l)}, \\ \hat{\lambda}^b &= -\frac{1}{\Delta t} \ln \frac{\eta_{kl} - \hat{\mu}^b \eta_k - \hat{\mu}^b \eta_l + I(\hat{\mu}^b)^2}{\eta_{kk} - 2\hat{\mu}^b \eta_k + I(\hat{\mu}^b)^2}, \\ \hat{\sigma}^b &= \frac{1}{I} [\eta_{ll} - 2\hat{\alpha}^b \eta_{kl} + (\hat{\alpha}^b)^2 \eta_{kk} \dots \\ &\quad - 2\hat{\mu}^b (1 - \hat{\alpha}^b)(\eta_l - \hat{\alpha}^b \eta_k) + I(\hat{\mu}^b)^2 (1 - \hat{\alpha}^b)^2], \end{aligned}$$

with

$$\begin{aligned} \eta_k &= \sum_{i=1}^I \hat{\eta}^b[i-1], \quad \eta_l = \sum_{i=1}^I \hat{\eta}^b[i], \\ \eta_{kk} &= \sum_{i=1}^I \hat{\eta}^b[i-1]^2, \quad \eta_{kl} = \sum_{i=1}^I \hat{\eta}^b[i-1] \hat{\eta}^b[i], \quad \eta_{ll} = \sum_{i=1}^I \hat{\eta}^b[i]^2, \end{aligned}$$

where the relation between $\hat{\sigma}^b$ and $\bar{\sigma}^b$ is defined in the same way as in Equation 4.9 and $\hat{\alpha}^b$ is defined in the same manner as in Equation 4.10. The estimated parameters can now be used as classification features.

4.1.2.4 Parameter probability density function

All the estimated parameters (of all pixels in a specific class) are represented by the vector $\Theta_c = \{\Theta_1, \Theta_2, \dots, \Theta_{35}\}$, where Θ_i is a random variable and θ_i (or rather $\hat{\theta}_i$) is a realisation of it. Note that NDVI is excluded from the parameter probability density function, as it is created from MODIS land bands 1 and 2. The joint density of Θ_c is assumed to be Gaussian distributed and expressed with

$$f_c(\theta_c) = \frac{1}{\sqrt{(2\pi)^{|\theta_c|} |\Sigma|}} \exp \left[-\frac{1}{2} (\theta_c - \mu) \Sigma^{-1} (\theta_c - \mu) \right]. \quad (4.11)$$

In Equation 4.11, $\mu = \mathbb{E}[\Theta_c]$ and Σ is the covariance matrix with elements $\Sigma_{n,m} = \mathbb{E}[(\Theta_n - \mu_{\Theta_n})(\Theta_m - \mu_{\Theta_m})]$, $\forall m, n \in \{1, \dots, |\theta_c|\}$.

4.1.2.5 Parameter and noise correlation

The parameter correlation matrix P_p^c has elements $P_{n,m} = \frac{\mathbb{E}[(\Theta_n - \mu_{\Theta_n})(\Theta_m - \mu_{\Theta_m})]}{\sigma_{\Theta_n} \sigma_{\Theta_m}}$, $\forall m, n \in \{1, \dots, |\theta_c|\}$. The parameter correlation matrix P_p is used to get an indication of the dependence between the model parameters of each class and is used to model class-specific spectral behaviour.

In addition to P_p^c , the noise correlation P_η^c is measured between the different MODIS bands. To determine the noise correlation, $dW^b(t)$ from Equation 4.3 needs to be estimated, since $dW^b(t)$ induces the random behaviour in the noise. To estimate $dW^b(t)$, $\eta^b(t)$ is discretised with timesteps of length Δt . An exact formula that holds for $\Delta t = 1$ is [83]

$$\eta^b[i] = e^{-\lambda^b} \eta^b[i-1] + (1 - e^{-\lambda^b}) \mu^b + \sigma^b \sqrt{\frac{(1 - e^{-2\lambda^b})}{2\lambda^b}} \Delta W^b[i], \quad (4.12)$$

where $\Delta W^b[i] \sim \mathcal{N}(0, 1)$ and is equal to $\Delta W^b[i] = W^b[i] - W^b[i-1]$.

By making $\Delta W^b[i]$ the subject of Equation 4.12, it can be used to estimate (or approximate) the *independent*, normally distributed innovation terms for each timestep of each MODIS band. This, in turn, allows the computation of the correlation matrix P_η^c of the innovation terms across the spectral bands with $P_{n,m} = \frac{\mathbb{E}[(\Omega_n - \mu_{\Omega_n})(\Omega_m - \mu_{\Omega_m})]}{\sigma_{\Omega_n} \sigma_{\Omega_m}}$, $\forall m, n \in \{1, \dots, 7\}$, where Ω_n is the random variable with realisations ΔW^n and n refers to the MODIS band.

4.1.2.6 Generating correlated innovations

Independent, correlated innovations are generated by following the approach presented in [84]. Consider d independent standard (i.e. unit variance) white noise processes $\overline{\Delta W}^1, \dots, \overline{\Delta W}^d$ each of length I , where I is the number of observations that needs to be simulated. Let furthermore a (deterministic and constant) matrix

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_{11} & \delta_{12} & \cdots & \delta_{1d} \\ \delta_{21} & \delta_{22} & \cdots & \delta_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{71} & \delta_{72} & \cdots & \delta_{7d} \end{bmatrix}$$

be given, and consider the seven-dimensional process $\Delta \mathbf{W}_c$, defined by

$$\Delta \mathbf{W}_c = \boldsymbol{\delta} \overline{\Delta \mathbf{W}}, \quad (4.13)$$

where

$$\Delta \mathbf{W}_c = [\Delta W_c^1, \dots, \Delta W_c^7]^T.$$

Moreover, assume that the rows of $\boldsymbol{\delta}$ have unit length, i.e.

$$\|\boldsymbol{\delta}_{i\#}\|_2 = 1, \quad i = 1, \dots, 7. \quad (4.14)$$

Then each of the components $\Delta W_c^1, \dots, \Delta W_c^7$ separately is also a standard (i.e. unit variance) white noise process, with instantaneous correlation given by

$$\mathbf{P}_\eta^c = \boldsymbol{\delta} \boldsymbol{\delta}^*. \quad (4.15)$$

Given a positive definite correlation matrix \mathbf{P}_η^c , $\boldsymbol{\delta}$ can be obtained by using Cholesky factorisation (Section A.3) [40], such that Equation 4.14 is automatically satisfied.

4.1.2.7 Simulating a MODIS pixel

Let $\boldsymbol{\sigma}_c = \{\sigma_c^b\}_{b \in \{1, \dots, 7\}}$, $\boldsymbol{\lambda}_c = \{\lambda_c^b\}_{b \in \{1, \dots, 7\}}$, $\boldsymbol{\mu}_c = \{\mu_c^b\}_{b \in \{1, \dots, 7\}}$, $\mathbf{C}_c = \{\mathbf{C}_c^b\}_{b \in \{1, \dots, 7\}}$, $\mathbf{A}_c = \{\mathbf{A}_c^b\}_{b \in \{1, \dots, 7\}}$, $\boldsymbol{\phi}_c = \{\phi_c^b\}_{b \in \{1, \dots, 7\}}$, $\mathbf{s}_c(t) = \{s_c^b(t)\}_{b \in \{1, \dots, 7\}}$ and $\boldsymbol{\eta}_c(t) = \{\eta_c^b(t)\}_{b \in \{1, \dots, 7\}}$. If the CSHO model is used to simulate a MODIS pixel which belongs to class c the following steps are required:

1. Draw $\boldsymbol{\theta}_c$ randomly from $f_c(\boldsymbol{\theta}_c)$ (assuming that $f_c(\boldsymbol{\theta}_c)$ has already been constructed by using the procedure discussed in Section 4.1.2.4).
2. Generate correlated seven-dimensional innovations $\Delta\mathbf{W}_c$ that are characterised by the correlation matrix \mathbf{P}_η^c (assuming that \mathbf{P}_η^c has already been estimated via the procedure discussed in Section 4.1.2.5) by using the procedure discussed in Section 4.1.2.6.
3. Use $\boldsymbol{\sigma}_c$ and $\boldsymbol{\lambda}_c$ (from $\boldsymbol{\theta}_c$) together with $\Delta\mathbf{W}_c$ and Equation 4.12 to generate $\boldsymbol{\eta}_c(t)$ under the assumption that $\boldsymbol{\mu}_c = \mathbf{0}$ and $\boldsymbol{\eta}_c(0) = \mathbf{0}$. The first 45 observations must be ignored, to allow $\boldsymbol{\eta}_c(t)$ to reach a state of equilibrium.
4. Use \mathbf{C}_c , \mathbf{A}_c and $\boldsymbol{\phi}_c$ (from $\boldsymbol{\theta}_c$) and Equation 4.1 to generate $\mathbf{s}_c(t)$.
5. Generate $\mathbf{x}_c(t)$ using $\mathbf{s}_c(t)$, $\boldsymbol{\eta}_c(t)$ and Equation 4.4.
6. Generate NDVI from $x_c^1(t)$ and $x_c^2(t)$.

4.2 CLASSIFICATION

As mentioned in the chapter introduction, *classification* is the act of arranging or organising according to class or category. Land cover classification using remotely sensed data is a critical first step in large-scale environmental monitoring, resource management and regional planning [14]. At this point it is prudent to point out the subtle difference between *land cover* and *land use*. Land cover refers to the (physical) surface cover, such as vegetation, urban infrastructure, water, bare soil etc., whereas land use refers to the (functional) purpose that the land serves, such as agriculture, recreation, or wildlife habitat [23].

The main focus of this section will be on land cover classification. In Section 4.2.1 a short literature review is given of land cover classification techniques, followed by the presentation of three hypertemporal classifiers in Section 4.2.2, Section 4.2.3 and Section 4.2.4. The CSFO feature set is discussed in Section 4.2.4.2.

4.2.1 Literature review

According to [85] there are two types of analytic approaches for creating land cover maps, namely *photointerpretation* and *machine analysis*. Photointerpretation relies on a human analyst to interpret an enhanced image. Machine analysis on the other hand, uses statistical or numerical algorithms to perform the labelling of multispectral datasets. A good review of different machine analysis techniques (henceforth described only as classification techniques) is available in [14]. According to [14], remote sensing classification approaches can be grouped using a taxonomy. The proposed taxonomy in [14] can be found in Figure 4.4. A short description of each category found in Figure 4.4 is given in [14, 23]. In [14], *the classification elements* are used as the primary attribute for grouping classification techniques together. In this section the classification of elements will also be used as the primary attribute for grouping classification techniques together. The different classification elements

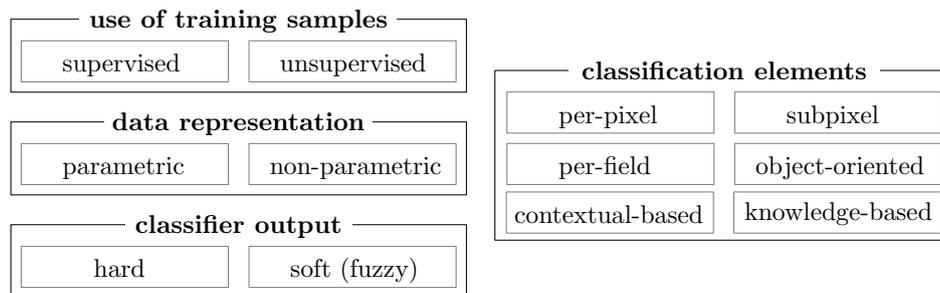


Figure 4.4: A taxonomy of fixed sample size land cover classification techniques (from [23]).

and some of the most popular algorithms used by each type of element are discussed in the following list:

1. In *per-pixel* classification, each pixel is classified as belonging to a specific land cover class. Per-pixel classification clearly assumes homogeneous pixels, which becomes unrealistic when the spatial resolution is decreased [23]. Per-pixel classifiers can be parametric or non-parametric. *Maximum likelihood classification* is probably the most commonly used *per-pixel* parametric classification approach and is presented clearly in [85]. The most frequently used non-parametric per-pixel classifiers are *Artificial Neural Networks (ANNs)* [86], *decision trees* [87, 88] and *SVMs* [89–91]. There are some remaining advanced per-pixel-based classification algorithms worth mentioning, which include: the *spectral angle classifier* [92], *Independent Component Analysis (ICA)* [93, 94], a *model-based approach* [95, 96] and several *nearest neighbour approaches* [97–99].

2. *Subpixel* techniques are especially used with medium or coarse resolution remote sensing data, since heterogeneous pixels are quite common at those spatial resolutions. Typically subpixel classification is done with either *fuzzy sets* [100, 101] or *Spectral Mixture Analysis (SMA)* [102–104]. Other prominent approaches to subpixel classification include ANNs [105], *Dempster-Shafer theory*, *certainty factors* [106] and a maximum likelihood approach [107].
3. One way to handle pixel heterogeneity is to employ *per-field* classification. In per-field classification pixels are no longer evaluated individually, but in “fields” consisting of the same land cover type, such that the noise can be averaged out over larger areas, implying that the fields are more homogeneous than the pixels that make up the fields (see for example [108, 109]). *Object-oriented* classification is similar to per-field-based classification. The main difference is that object-oriented methods use only raster data, whereas per-field approaches use vector and raster data. The reference list [110–112], provides additional information on object-oriented classification. A frequently used object-oriented approach is eCognition, which is described in (among others) [113].
4. *Contextual-based* approaches to land cover classification take the spatial distribution of pixels into account in an attempt to minimise the effects of intra-class variations [114]. In [115] a selection of early ad hoc contextually based classifiers are compared. More recently it has been shown that the Markov and Gibbs random fields are effective approaches that can use spatial information [116, 117]. Markov and Gibbs random fields were introduced to image processing by the seminal paper [118]. There are also spectral-contextual classifiers of which [119] is a good example.
5. *Knowledge-based* methods use ancillary data sources (such as a digital elevation map, a soil map, housing, etc.) on top of the contextual information that is available for a region to perform classification (see [120] for an example).

4.2.1.1 Dimensionality reduction

The large amount of training data that hyperspectral data provide needs to be reduced, as classifiers that use large training datasets become impractical very quickly [23]. An effective way of reducing training datasets is to use *dimensionality reduction*, which is closely related to *feature extraction*. Dimensionality reduction algorithms have to be able to select the most prevailing elements from a

dataset, while skipping the unimportant elements. Several approaches to dimensionality reduction exist, including *Principal Component Analysis (PCA)*, *minimum noise fraction transform*, *discriminant analysis* [121–123], *decision boundary feature extraction* [124], *Gaussian mixture model feature extraction* [95], *wavelet transform* [125] and *SMA* [126].

4.2.1.2 Hypertemporal classification

Most of the classification techniques discussed in the literature review up to now have been single-date classifiers. It has been shown that multitemporal and hypertemporal classification is more reliable than single-date classification [15, 127, 128], since single-date reflectance values between different classes may be unseparable due to the fact that land-cover classes could have similar spectral characteristics during certain times of the year [15]. A second reason that motivates hypertemporal classification is that most of the earth (landmass) is covered by vegetation. Vegetation species have unique phenologies, which make remote classification possible [39]. The most prominent hypertemporal classification techniques in literature are PCA [17, 129, 130], phenological metrics [18, 131, 132], Fourier analysis [5, 133–136], wavelet analysis [137], minimum distance classification [16, 23] and time-varying maximum likelihood classification [23].

The chapter focuses on hypertemporal classification techniques. In particular it revolves around the parameters of the CSHO. The parameters of the CSHO will be used as features that will be fed into an SVM classifier. The proposed technique extends the approach in [5], which is based on Fourier features. In [5], it is shown that efficient separability can in fact be achieved when using only the mean and seasonal harmonic components. The Ornstein-Uhlenbeck process, which is a component of the CSHO, summarises the less important Fourier features that by themselves do not contribute significantly to classification up with two average model parameters that could possibly contribute significantly to classification accuracy. The SVM classifier with CSHO features is compared to the minimum distance classifier [16], the time-varying model classifier [23] and SVMs fed with temporal and harmonic features in Section 5.3.

4.2.2 Minimum distance classifier

The minimum distance classifier classifies the observed signal $\tilde{\mathbf{x}}(t)$ as class c by choosing the class with the lowest model error [16, 23]. Where the model error for each class c is defined as the accumulated euclidean distance between the observed signal $\tilde{\mathbf{x}}(t)$ and the signal model (yearly ensemble

mean) $\tilde{\mathbf{y}}_c(t)$, mathematically it can be written as, find a c such that the following optimisation problem is minimised:

$$\inf_{c \in \mathcal{C}} \int_0^I \|\tilde{\mathbf{x}}(t) - \tilde{\mathbf{y}}_c(t)\|_2 dt.$$

Any subset of $\tilde{\mathbf{x}}(t)$ and $\tilde{\mathbf{y}}_c(t)$ can be used for classification, as long as both subsets are constructed from the same spectral bands. The euclidean differences are normalised with the difference between the maximum and minimum observed value in each band.

4.2.3 Time-varying maximum likelihood classifier

The time-varying maximum likelihood classifier uses the time-varying model [23]. The background theory used in this section was discussed in detail in Section 3.4. The time-varying model is a discrete model and $\tilde{\mathbf{X}}_c(t)$ therefore needs to be discretised. Let the discretised form of $\tilde{\mathbf{X}}_c(t)$ be denoted by $\{\tilde{\mathbf{X}}_c[k]\}_{k=\{1,2,\dots\}}$. The time-varying model is equivalent to the first order statistical description of $\{\tilde{\mathbf{X}}_c[k]\}_{k=\{1,2,\dots\}}$. Usually the phrase “first order statistical description” is only associated with a single stochastic process, but here the first order statistical description is connected with a set of stochastic processes. The first order statistical description of $\{\tilde{\mathbf{X}}_c[k]\}_{k=\{1,2,\dots\}}$ is equal to the set of probability density functions at each time step k , $\{q_k^c\}_{k=\{1,2,\dots\}}$. If it is assumed that the MODIS data contain no inter-annual variation, in other words it is assumed that the MODIS time-series is periodic (45 observations in a year), then it is true that $q_k^c = q_{k+45n}^c$, $n = \{1,2,\dots\}$. Note that q_k^c is an eighth-dimensional density and that the density at k can also be constructed for a smaller number of bands. When only a subset of the bands is used the notation $q_k^{c,\mathbf{b}}$ is used, where \mathbf{b} can be any subset of $\{1, \dots, 7, \text{NDVI}\}$. The same rule applies for $\tilde{\mathbf{X}}_c^{\mathbf{b}}(t)$ and $\tilde{\mathbf{x}}_c^{\mathbf{b}}(t)$. Assume now that the class label c is equal to either a $v \equiv 0$ or a $s \equiv 1$ if the observed MODIS pixel belongs to either the vegetation or settlement class. Any unlabelled MODIS pixel $\{\tilde{\mathbf{x}}^{\mathbf{b}}[k]\}_{k \in \mathbb{N}}$ obeys one of two statistical hypotheses:

$$\mathcal{H}_0 : \tilde{\mathbf{x}}^{\mathbf{b}}[k] \sim Q_k^{0,\mathbf{b}}, k = 1, 2, \dots$$

versus

$$\mathcal{H}_1 : \tilde{\mathbf{x}}^{\mathbf{b}}[k] \sim Q_k^{1,\mathbf{b}}, k = 1, 2, \dots;$$

where for each time step k , $Q_k^{0,\mathbf{b}}$ and $Q_k^{1,\mathbf{b}}$ are two $|\mathbf{b}|$ -dimensional probability distributions with associated densities $q_k^{0,\mathbf{b}}$ and $q_k^{1,\mathbf{b}}$, respectively. Further assume that hypothesis \mathcal{H}_1 occurs with prior probability π and \mathcal{H}_0 with prior probability $1 - \pi$.

Now define the posterior sequence to be

$$\pi_k^\pi = \frac{\pi_{k-1}^\pi q_k^{1,\mathbf{b}}(\bar{\mathbf{x}}^{\mathbf{b}}[k])}{\pi_{k-1}^\pi q_k^{1,\mathbf{b}}(\bar{\mathbf{x}}^{\mathbf{b}}[k]) + (1 - \pi_{k-1}^\pi) q_k^{0,\mathbf{b}}(\bar{\mathbf{x}}^{\mathbf{b}}[k])}, \quad k = 1, 2, \dots,$$

where $\pi_0^\pi = \pi$. The maximum likelihood classification of the time-varying classification task is then given by

$$\delta_k = \begin{cases} 0, & \text{if } \pi_k^\pi \leq 0.5 \\ 1, & \text{if } \pi_k^\pi > 0.5. \end{cases}$$

If thresholds are introduced to the time-varying maximum likelihood classifier then the time-varying maximum likelihood classifier becomes sequential in nature. Let $\{\pi_U, \pi_L\}$ be those thresholds. If π_k^π crosses $\{\pi_U, \pi_L\}$ a decision can be made. The decision rule now becomes

$$\delta_k = \begin{cases} 0, & \text{if } \pi_k^\pi \leq \pi_L \\ 1, & \text{if } \pi_k^\pi > \pi_U. \end{cases}$$

It can easily be shown (see Section 3.4.2 for more details) that the sequential time-varying maximum likelihood classifier is equivalent to the time-varying SPRT (in terms of classification accuracy and delay), where the time-varying SPRT is obtained by casting the classification problem presented in this section into the likelihood domain. Currently the classification problem is solved using a posterior sequence.

4.2.4 Support Vector Machine

An SVM works by creating a hyperplane or set of hyperplanes in a high or infinite dimensional space, and as such can be used for classification, regression, or to perform other similar functions [1, 138, 139]. An SVM works on the principle of finding a hyperplane, such that the hyperplane has the furthest distance from the training data of any class (which is known as the functional margin). The training data for the classifier are a set of n points of the form

$$\mathcal{D} = \{(\mathbf{r}^{(i)}, \psi^{(i)} | \mathbf{r}^{(i)} \in \mathbb{R}^p, \psi^{(i)} \in \{-1, 1\}\} \quad (4.16)$$

where $\psi^{(i)}$ is a label denoting class membership, and $\mathbf{r}^{(i)}$ is a p -dimensional real feature vector. If the data are linearly separable, a maximum-margin hyperplane is calculated to divide the data into points belonging to the class with label -1 or 1 perfectly. The maximum-margin hyperplane is represented by the following

$$\mathbf{w}^T \cdot \mathbf{r} + b = 0, \quad (4.17)$$

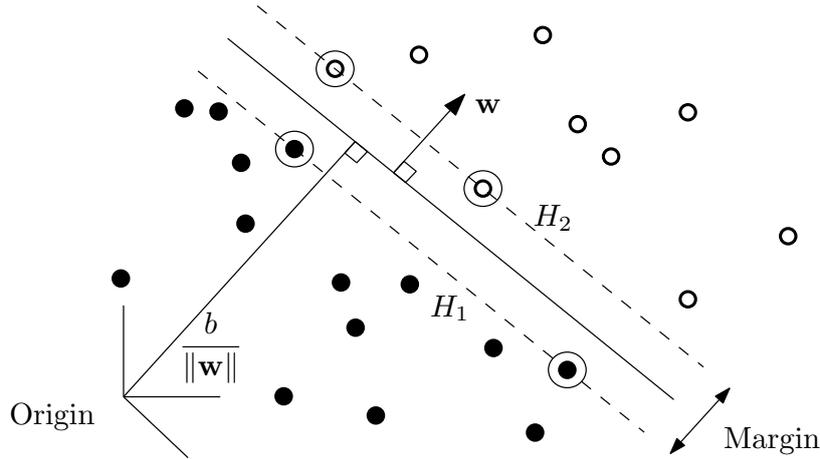


Figure 4.5: Example of a maximum-margin hyperplane of a linear SVM (from [23]).

where vector \mathbf{w} is perpendicular to hyperplane, Equation 4.17, and $\frac{b}{\|\mathbf{w}\|}$ is the offset of hyperplane, Equation 4.17, from the origin in the direction of \mathbf{w} . The maximum-margin hyperplane is calculated by choosing \mathbf{w} and b to maximise the distance between the hyperplanes $\mathbf{w}^T \cdot \mathbf{x} + b = -1$ (which corresponds to hyperplane H_1 in Figure 4.5) and $\mathbf{w}^T \cdot \mathbf{x} + b = 1$ (which corresponds to hyperplane H_2 in Figure 4.5). These two hyperplanes are as far a part as possible although they still correctly classify each training data point. The problem of maximising the distance between the hyperplanes $\mathbf{w}^T \cdot \mathbf{x} + b = -1$ and $\mathbf{w}^T \cdot \mathbf{x} + b = 1$ reduces to the following optimisation problem:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & \psi_i(\mathbf{w}^T \cdot \mathbf{r} + b) \geq 1, \quad i = 1, \dots, n. \end{aligned} \quad (4.18)$$

Equation 4.18 is known as the *primal problem*. The optimisation problem is actually solved by using the so-called *dual problem*, which is the Lagrangian reformulation of the primal problem. There are mainly two reasons for rather solving the dual problem, namely the constraints of Equation 4.18 are supplanted by constraints of the Lagrange multipliers themselves, which are much easier to deal with, and in the dual problem inner products are used in both the training and testing algorithms, which makes it possible to effortlessly generalise to non-linear SVMs [23]. Readers interested in the dual problem are referred to [139], which is a comprehensive tutorial on SVMs. To extend the

approach to non-separable datasets the optimisation problem is reformulated to obtain:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & \psi_i(\mathbf{w}^T \cdot \mathbf{r} + b) \geq 1 - \xi_i, \quad i = 1, \dots, n \\ & \xi_i \geq 0, \quad i = 1, \dots, n. \end{aligned}$$

The method entails, introducing slack variables, ξ_i , which measure the degree of misclassification of \mathbf{r}_i . The parameter C controls the relative weighting between the slack variables and the goal $\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$.

An SVM was chosen as classification technique since SVMs, unlike neural networks, are robust to the over-fitting problem (increased spectral view increases feature set sizes). The first documented use of SVMs in remote sensing was in [140]. A good review of the application of SVMs in the remote sensing field can be found in [91], of which [89, 141–143] are worth singling out. It is also worth mentioning [15, 144–148], as these works applied SVMs to MODIS data.

4.2.4.1 Example

Consider the following linearly separable binary classification problem:

$$\begin{aligned} \boldsymbol{\rho} &= [w_1, w_2, b]^T, \quad \mathbf{w} = [w_1, w_2]^T. \\ \mathbf{r} &= [r_1, r_2]^T. \end{aligned}$$

The aim is to find a hyperplane $\mathbf{w}^T \mathbf{r} + b = 0$ that separates the binary classes perfectly, while the margin between $\mathbf{w}^T \mathbf{r} + b = -1$ and $\mathbf{w}^T \mathbf{r} + b = 1$ is also maximised. The following six training examples are given,

$$\begin{aligned} \mathbf{r}^{(1)} &= [1, 1]^T = (1, 1), \\ \mathbf{r}^{(2)} &= [1, 2]^T = (1, 2), \\ \mathbf{r}^{(3)} &= [2, 1]^T = (2, 1), \\ \mathbf{r}^{(4)} &= [3, 3]^T = (3, 3), \\ \mathbf{r}^{(5)} &= [2, 4]^T = (2, 4), \\ \mathbf{r}^{(6)} &= [4, 5]^T = (4, 5). \end{aligned}$$

With classification given by the label $\psi^{(i)} \in \{-1, +1\}$ for each training example,

$$\boldsymbol{\psi} = [-1, -1, -1, 1, 1, 1].$$

The given binary classification problem is represented graphically in Figure 4.6. The hyperplane

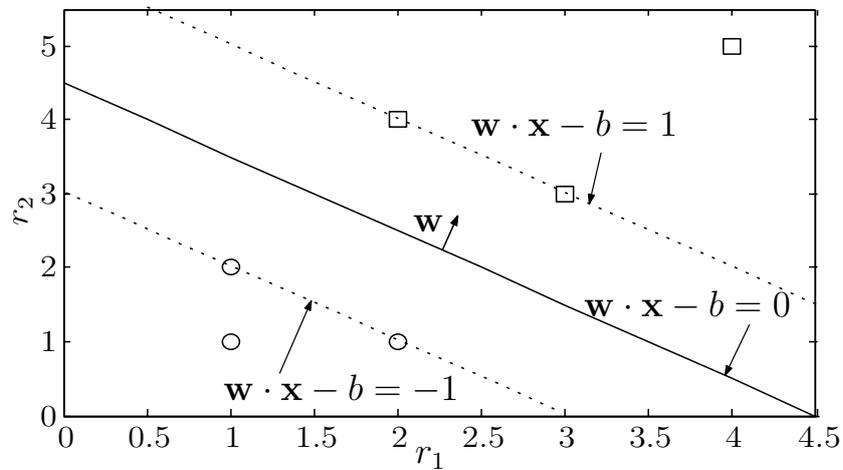


Figure 4.6: An example of an SVM classification problem.

$\mathbf{w}^T \mathbf{r} + b = 0$ for this example can be found by solving the following primal minimisation problem.

$$\begin{aligned}
 \text{Minimize} \quad & f(\boldsymbol{\rho}) = \frac{1}{2} \|\mathbf{w}\|^2 = \frac{1}{2} \mathbf{w}^T \mathbf{w}, & (4.19) \\
 \text{such that} \quad & g_1(\boldsymbol{\rho}) = w_1 + w_2 + b + 1 \leq 0, \\
 & g_2(\boldsymbol{\rho}) = w_1 + 2w_2 + b + 1 \leq 0, \\
 & g_3(\boldsymbol{\rho}) = 2w_1 + w_2 + b + 1 \leq 0, \\
 & g_4(\boldsymbol{\rho}) = -2w_1 - 4w_2 - b + 1 \leq 0, \\
 & g_5(\boldsymbol{\rho}) = -3w_1 - 3w_2 - b + 1 \leq 0, \\
 & g_6(\boldsymbol{\rho}) = -4w_1 - 5w_2 - b + 1 \leq 0,
 \end{aligned}$$

which can be transformed to have only *equality constraints* by introducing the auxiliary variables, κ_j . After introducing the auxiliary variables Equation 4.19 reduces to:

$$\begin{aligned}
 &\text{minimise} && f(\boldsymbol{\rho}) = \frac{1}{2} \|\mathbf{w}\|^2 && (4.20) \\
 &\text{such that} && h_1(\boldsymbol{\rho}) = w_1 + w_2 + b + 1 + \kappa_1^2 = 0, \\
 &&& h_2(\boldsymbol{\rho}) = w_1 + 2w_2 + b + 1 + \kappa_2^2 = 0, \\
 &&& h_3(\boldsymbol{\rho}) = 2w_1 + w_2 + b + 1 + \kappa_3^2 = 0, \\
 &&& h_4(\boldsymbol{\rho}) = -2w_1 - 4w_2 - b + 1 + \kappa_4^2 = 0, \\
 &&& h_5(\boldsymbol{\rho}) = -3w_1 - 3w_2 - b + 1 + \kappa_5^2 = 0, \\
 &&& h_6(\boldsymbol{\rho}) = -4w_1 - 5w_2 - b + 1 + \kappa_6^2 = 0.
 \end{aligned}$$

The constraints in Equation 4.19 follow from the requirement that

$$\psi^{(i)} \left(\mathbf{w}^T \mathbf{r}^{(i)} + b \right) \geq 1, \quad \forall i = 1, \dots, 6. \quad (4.21)$$

Now the Lagrange function (Section A.4) [40] of Equation 4.20 can easily be constructed, which is equal to

$$\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\kappa}, \boldsymbol{\lambda}) = f(\boldsymbol{\rho}) + \sum_{j=1}^6 \lambda_j (g_j(\boldsymbol{\rho}) + \kappa_j^2)$$

with $\boldsymbol{\lambda}$ being the Lagrange multiplier. The Lagrange function can be used for solving Equation 4.19 by setting each of its partial derivatives to zero and solving the set of equations formed. Taking the partial derivative of the Lagrange function with respect to each of its variables yields the following

set of equations:

$$\begin{aligned}
 \frac{\partial \mathcal{L}}{\partial w_1} &= w_1 + \lambda_1 + \lambda_2 + \lambda_3 - 2\lambda_4 - 3\lambda_5 - 4\lambda_6, \\
 \frac{\partial \mathcal{L}}{\partial w_2} &= w_2 + \lambda_1 + 2\lambda_2 + \lambda_3 - 4\lambda_4 - 3\lambda_5 - 5\lambda_6, \\
 \frac{\partial \mathcal{L}}{\partial b} &= \lambda_1 + \lambda_2 + \lambda_3 - \lambda_4 - \lambda_5 - \lambda_6, \\
 \frac{\partial \mathcal{L}}{\partial \kappa_j} &= 2\lambda_j \kappa_j \quad \forall i \ 1 \dots 6, \\
 \frac{\partial \mathcal{L}}{\partial \lambda_1} &= w_1 + w_2 + b + 1 + \kappa_1^2, \\
 \frac{\partial \mathcal{L}}{\partial \lambda_2} &= w_1 + 2w_2 + b + 1 + \kappa_2^2, \\
 \frac{\partial \mathcal{L}}{\partial \lambda_3} &= 2w_1 + w_2 + b + 1 + \kappa_3^2, \\
 \frac{\partial \mathcal{L}}{\partial \lambda_4} &= -2w_1 - 4w_2 - b + 1 + \kappa_4^2, \\
 \frac{\partial \mathcal{L}}{\partial \lambda_5} &= -3w_1 - 3w_2 - b + 1 + \kappa_5^2, \\
 \frac{\partial \mathcal{L}}{\partial \lambda_6} &= -4w_1 - 5w_2 - b + 1 + \kappa_6^2.
 \end{aligned}$$

Setting the above equations to zero and solving produces the following result:

$$\begin{bmatrix} w_1^* \\ w_2^* \\ b^* \\ \kappa_1^* \\ \kappa_2^* \\ \kappa_3^* \\ \kappa_4^* \\ \kappa_5^* \\ \kappa_6^* \\ \lambda_1^* \\ \lambda_2^* \\ \lambda_3^* \\ \lambda_4^* \\ \lambda_5^* \\ \lambda_6^* \end{bmatrix} = \begin{bmatrix} 2/3 \\ 2/3 \\ -3 \\ 0.8165 \approx 49/60 \\ 0 \\ 0 \\ 0 \\ 0 \\ \sqrt{2} \\ 0 \\ 0 \\ 4/9 \\ -2/9 \\ 2/3 \\ 0 \end{bmatrix} \quad (4.22)$$

The first three solutions of Equation 4.22 are also the solution of Equation 4.19, $\boldsymbol{\rho}^* = [2/3, 2/3, -3]$, which completely describes the hyperplane $\mathbf{w}^T \mathbf{r} + b = 0$. The hyperplane can be rewritten in terms of r_1 and r_2 .

$$\begin{aligned} \mathbf{w}^T \mathbf{r} + b &= w_1 r_1 + w_2 r_2 + b \\ &= 2/3 r_1 + 2/3 r_2 - 3 \\ &= 0 \end{aligned}$$

leading to the normal straight line $r_2 = -r_1 + 4.5$.

4.2.4.2 Proposed features

Three main sets of SVM features will be presented. The first feature set consists of the harmonic components of Equation 4.4 and is denoted by

$$\tilde{\mathbf{i}} = \{C^b, A^b\}_{b \in \{1, \dots, \text{NDVI}\}}. \quad (4.23)$$

Any spectral subset of $\tilde{\mathbf{i}}$ can also be selected, and is denoted by $\mathbf{i}^{\mathbf{b}}$, where \mathbf{b} is any subset of $\{1, \dots, \text{NDVI}\}$. Fourier (or spectral) analysis, on NDVI time-series in particular, has been used extensively for land cover classification (see for example [5, 78, 134, 149]), and it has been shown that reliable class separation can be achieved even when considering only the mean and seasonal spectral components [5, 78], i.e. Equation 4.23.

The second feature set consists of noise-harmonic features, i.e. consists of all the parameters in Equation 4.4 and is represented mathematically with

$$\tilde{\boldsymbol{\theta}} = \{C^b, A^b, \phi^b, \lambda^b, \sigma^b\}_{b \in \{1, \dots, \text{NDVI}\}}.$$

As in the case of $\tilde{\mathbf{i}}$, a spectral subset of $\tilde{\boldsymbol{\theta}}$ is denoted by $\boldsymbol{\theta}^{\mathbf{b}}$. The benefit of $\tilde{\boldsymbol{\theta}}$ when compared to $\tilde{\mathbf{i}}$ is that $\tilde{\boldsymbol{\theta}}$ also includes the parameters of the Ornstein-Uhlenbeck process, which are $\tilde{\boldsymbol{\lambda}}$ and $\tilde{\boldsymbol{\sigma}}$. The Ornstein-Uhlenbeck process summarises the less important Fourier features that by themselves do not contribute significantly to classification up with two model parameters that could contribute significantly to classification accuracy.

The third feature set is composed of temporal features. Selecting temporal features for classification purposes is a well-known approach [15]. If the most relevant reflectance values of a MODIS pixel $\tilde{\mathbf{x}}(t)$ are to be chosen, then those reflectance values from $\tilde{\mathbf{x}}(t)$ where the annual ensemble mean of two

different classes are at a maximum distance from each other need to be selected. Mathematically it can be expressed as follows: a τ should be selected such that the following optimisation problem is maximised.

$$\sup_{\tau \in \{1, \dots, 45\}} \|\tilde{\mathbf{y}}_{c_1}(\tau) - \tilde{\mathbf{y}}_{c_2}(\tau)\|_2,$$

where $\tilde{\mathbf{y}}$ represents the annual ensemble mean. The solution τ can be extended to a sequence $\boldsymbol{\tau}$; since the annual ensemble mean is periodic, a maximum can be attained more than once during the observation period I . Now the actual reflectance values from the observed MODIS pixel are selected,

$$\tilde{\boldsymbol{\zeta}} = \tilde{\mathbf{x}}(\boldsymbol{\tau}) = \{x_b(\boldsymbol{\tau})\}_{b \in \{1, \dots, 7, \text{NDVI}\}}.$$

A smaller $\boldsymbol{\zeta}^b$ can be constructed by using subsets of $\tilde{\mathbf{y}}_{c_1}(t)$, $\tilde{\mathbf{y}}_{c_2}(t)$ and $\tilde{\mathbf{x}}(t)$, as long as the subsets are constructed by using the same spectral bands. Now, $\tilde{\mathbf{l}}$, $\tilde{\boldsymbol{\theta}}$ or $\tilde{\boldsymbol{\zeta}}$ can be substituted into $\mathbf{r}^{(t)}$ (see Equation 4.16). As mentioned in Chapter 1, the shorthand notation \mathbf{l} , $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$ will be used to respectively refer to each feature set group, namely harmonic features, noise-harmonic features and temporal features.

4.3 CHANGE DETECTION

As stated in the chapter introduction, *change detection* is the process of identifying differences in the state of an object or phenomenon by observing it at different times. Essentially, it involves the ability to quantify temporal effects using multitemporal data [69]. Usually land cover changes are categorised into *land cover conversion* and *land cover modification* [71]. Land cover conversion is described in [71] as “complete replacement of one cover type by another”, whereas land cover modification is described as “more subtle changes that affect the character of land cover without changing its overall classification”.

In Section 4.3.1 a short literature review is given of land cover change detection techniques, followed by the presentation of two hypertemporal change detection techniques in Sections 4.3.2 and 4.3.3.

4.3.1 Literature review

There have been quite a number of reviews on change detection in the remote sensing field, namely [13, 69–73, 150]. The review written by Singh [69] in 1989 laid much of the foundation needed

to categorize remote sensing change detection approaches effectively [151]. The following main categories were proposed by Singh, namely *univariate image differencing*, *image regression*, *image rationing*, *vegetation index differencing*, PCA, *post-classification comparison*, *direct multi-date comparison* and *Change Vector Analysis (CVA)*. Later Lu *et al.* [13] improved on the categories proposed by Singh (more organised) [151]. The categories proposed by Lu *et al.* are briefly discussed below:

- *Algebra* – this category includes the methods that depend on a change metric, which is subsequently compared to a threshold value in order to declare a change or not. The change metric can be computed in a variety of ways, namely image differencing [152–155], image regression [156], image rationing [157], vegetation index differencing [158] and CVA [159].
- *Transformation* – this category comprises methods that reduce data dimensionality. Some of the possible approaches to reducing data redundancy are PCA [160], *Kauth-Thomas* [161], *Gramm-Schmidt* [161] and the *Chi-square transformation* [162].
- *Classification – post-classification comparison* [163], *Expectation Maximization* [164, 165] and ANN [166] are some of the constituent techniques that make up the classification category. The methods in this category use classified images and require a large amount of training data.
- *Advanced models* – this category includes, among others, the *Li-Strahler reflectance model* [167], SMA [102], and the *biophysical parameter estimation model* [168, 169]. The fundamental idea behind the methods in this category is that the reflectance values are converted to biophysical parameters, which are more interpretable than the original raw reflectance values.
- *Geographic Information System (GIS)* – the integrated GIS and remote sensing method [170] and the standard GIS approach [171] are some of the algorithms that fall into this category.
- *Visual interpretation* – visual interpretation requires manual interpretation of remote sensing images at different times followed by on-screen digitation of change polygons [172].
- *Other methods* – many categories have now been suggested to group the different change detection techniques together. There are however some techniques that do not fall into any of the above categories, namely *measures of spatial dependence* [173], *knowledge-based vision systems* [174], *change curves* [175], *generalised linear models* [176], the *curve theorem-based*

approach [177], *structure based approach* [178] and *spatial statistics-based approach* [179].

Most of the categories proposed by Lu *et al.* consist of methods that are multitemporal, which normally require only two images as input. There are however some reviews that explicitly discuss an additional category called *hypertemporal change detection techniques* or *temporal trajectory analysis* [71, 73]. The other categories proposed by [71, 73], will not be adopted, as the multitemporal techniques are grouped sufficiently using the categories proposed by [13]. All the methods that are applied to hypertemporal time-series fall into this additional category.

4.3.1.1 Hypertemporal techniques

When considering multi-date change detection, a serious consideration is the selection of optimal image dates. This problem can be circumvented by considering a hypertemporal time-series [10]. The last decade has seen a dramatic increase in the number of papers published in the field of hypertemporal change detection (remote sensing) [7, 12, 19–21, 28, 151, 180–193], some of which are discussed briefly below.

Temporal change metrics are used in [180] to detect land cover changes. The temporal change metrics are computed by computing the annual difference (year2-year1) of the annual maximum, annual minimum and annual range. In addition to the above metrics, the magnitude of the multitemporal change vector is also calculated. These metrics are then compared with a threshold to determine whether a change has occurred or not. In [7], a change is detected by identifying abnormal pixel behaviour. These pixels are identified by selecting pixels that show a significant deviation in the annual difference of the yearly total NDVI relative to other pixels from the same class and study area. In [184, 185] a disturbance index is computed to detect large-scale ecosystem disturbances. The disturbance index is calculated on an annual basis by dividing two ratios. The top ratio is calculated by dividing the annual maximum land surface temperature LST_{max} with the annual maximum Enhanced Vegetation Index (EVI) EVI_{max} , while the bottom ratio is calculated by dividing the multi-year mean of LST_{max} with the multi-year mean of EVI_{max} . The disturbance index is then compared with a predetermined threshold to determine whether a change flag is required. The departure from a model algorithm, the recursive merging algorithm and the yearly delta algorithm are some of the multitemporal techniques proposed in [28, 187]. A generic change detection approach is proposed in [19, 20] for NDVI time-series by detecting and characterising Breaks for Additive Seasonal and Trend (BFAST). Lastly, it is worth mentioning the sliding window approach documented in [21], the autocorrelation approach

proposed in [189, 190] and the Kalman filter approach presented in [191].

Most of the remote sensing hypertemporal change detection algorithms in the literature use some form of windowing, in other words only recent data are used to detect change. In contrast the hypertemporal change detection algorithm proposed in Section 4.3.3 (Page's original CUSUM algorithm [6]) is windowless, consequently no step is required to determine the window length.

There are several metrics by which a change detection algorithm should be evaluated. An obvious one is *detection delay*, which is the time taken for the change detection algorithm to declare that a change has occurred, given that a change in the data actually occurred. Then there is the question of how likely it is for the algorithm to declare that a change has occurred, given that the change in the data did in fact occur, a metric that is referred to as either *probability of detection* or the True Positive Rate (TP). There are more metrics that need to be considered. For example, there is the possibility that the algorithm will declare change, even though no change has occurred in the data, which can be referred to as either the *probability of false detection (alarm)* or the False Positive Rate (FP). As this chapter is presented in a statistical framework, the detection theory terms TP and FP will not be adopted. Then there is the question of how to eliminate the need for a windowing mechanism, in the sense that the proposed algorithm is *on-line* or sequential, i.e. it uses all the past data. This is possible when the algorithm has the property that it only starts behaving differently when an actual change has occurred. However, it is not common for the above four change detection criteria to be considered simultaneously in a remote sensing change detection context, and in that respect the proposed algorithm is novel, since it can sequentially detect change (vegetation pixels that are changed into settlement pixels) as accurately and quickly as possible, while staying below a certain probability of false alarm.

The proposed change detection algorithm (Section 4.3.3) uses Page's original CUSUM algorithm in order to process samples sequentially [6]. Windowed versions of the CUSUM algorithm have been used with MODIS in the past, typically in a bootstrapping [194] or in an in-control process mean context [28, 29]. The problem with using only recent data, which was extracted using a window, is that the average pixel behaviour might not be captured if the window is not long enough. The next example highlights the main drawback of using a window.

In Figure 4.7, the time-series of a vegetation pixel in Gauteng (that did not change from land cover type) over 8 years and its filtered output (which could be considered as the in-control process mean)

is presented in Figure 4.7. Clearly to select a proper history period is quite difficult in the case of

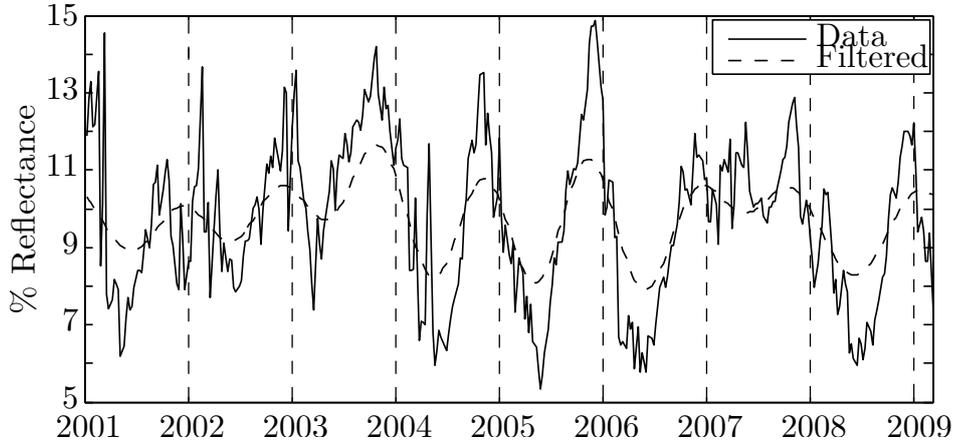


Figure 4.7: Temporal behaviour of a vegetation pixel (Gauteng) in MODIS band 1, and its filtered output (first 10 FFT components).

Figure 4.7. An improper estimated in-control process mean (due to a bad history period) will lead to wrongly estimated residuals (larger than they should be). Larger residuals cause an unnecessary amount of false alarms. Which is why CUSUM performs so badly for the approach presented in [28, 187].

CUSUM can be implemented without using a window, because in Page's original form the CUSUM statistic is derived from log-likelihood ratios, which can be obtained from densities estimated at every time-step of the year. The densities at each time-step thus circumvents intra-annual variation. The densities are constructed by using the CSHO, which can replicate average pixel behaviour which implies that the effect of inter-annual variation is also minimised (see Section 5.2.4.4) [2, 30, 32]. The CUSUM change detection algorithm is compared with the popular band differencing approach (Section 4.3.2) in Section 5.4 [7, 10, 28].

4.3.2 Lunetta et al.'s scheme

Let $\tilde{\mathbf{x}}_p[k] = \{x_p^b[k]\}_{b \in \{1, \dots, 7, \text{NDVI}\}}$ be the p -th discrete MODIS pixel in a set of P unlabelled pixels. The c subscript is omitted as the class of the pixel is unknown. The change detection scheme proposed by Lunetta *et al.* can be implemented with the following steps [28]:

1. The signal $\tilde{\mathbf{x}}_p[k]$ is first filtered, by keeping only the first v components of an I point Fast Fourier Transform (FFT), where I is equal to the temporal dimension of $\tilde{\mathbf{x}}_p[k]$.

2. For each pixel p compute the annual sum for each year of data (of which there are Y years). Let $\{\mathbf{a}_{p1}, \dots, \mathbf{a}_{pY}\}$ correspond to this list of annual sums, where $\mathbf{a}_{p1} = \{\sum_{k=1}^{45} x_p^b[k]\}_{b \in \{1, \dots, 7, \text{NDVI}\}}$, etc.
3. For each pixel p compute the difference between consecutive annual sums, i.e., $\{\mathbf{a}_{p2} - \mathbf{a}_{p1}, \mathbf{a}_{p3} - \mathbf{a}_{p2}, \dots, \mathbf{a}_{pY} - \mathbf{a}_{pY-1}\}$, where $\mathbf{a}_{p2} - \mathbf{a}_{p1} = \{a_{p2}^b - a_{p1}^b\}_{b \in \{1, \dots, 7, \text{NDVI}\}}$, etc. Let $\mathbf{d}_{pj} = \mathbf{a}_{pj+1} - \mathbf{a}_{pj}$.
4. For each pixel p compute the z -score $\left\{ \mathbf{z}_{pj} = \frac{\mathbf{d}_{pj} - \boldsymbol{\mu}_j}{\boldsymbol{\sigma}_j} \right\}$ for each of the $Y - 1$ values in $\{\mathbf{d}_{p1}, \mathbf{d}_{p2}, \dots, \mathbf{d}_{pY-1}\}$. This is done for each \mathbf{d}_{pj} by subtracting the mean ($\boldsymbol{\mu}_j = \mathbb{E}\{\mathbf{d}_{1j}, \mathbf{d}_{2j}, \dots, \mathbf{d}_{pj}\} = \{\mathbb{E}\{d_{1j}^b, d_{2j}^b, \dots, d_{pj}^b\}\}_{b \in \{1, \dots, 7, \text{NDVI}\}}$) and dividing by the standard deviation ($\boldsymbol{\sigma}_j = \sqrt{\mathbb{E}\{(\mathbf{d}_{1j})^2, (\mathbf{d}_{2j})^2, \dots, (\mathbf{d}_{pj})^2\} - (\boldsymbol{\mu}_j)^2}$, where $(\mathbf{d}_{1j})^2 = \{(d_{1j}^b)^2\}_{b \in \{1, \dots, 7, \text{NDVI}\}}$, etc.). Note that the mean and the standard deviation are computed across space. Let $\{\mathbf{z}_{p1}, \mathbf{z}_{p2}, \dots, \mathbf{z}_{pY-1}\}$ correspond to this list of z -scores.
5. For each pixel p compute the change score $\mathbf{c}_p = \{\max\{|z_{p1}^b|, |z_{p2}^b|, \dots, |z_{pY-1}^b|\}\}_{b \in \{1, \dots, 7, \text{NDVI}\}}$. A change or no change decision can now be reached in every band by comparing \mathbf{c}_p with the eight-dimensional threshold \mathbf{h}_l . If $\mathbf{c}_p > \mathbf{h}_l$ a change is declared.

4.3.3 Cumulative Sum

The CUSUM algorithm is discussed in detail in Section 3.6. To apply the CUSUM algorithm to MODIS time-series it needs to be modified slightly. The modified CUSUM algorithm that is presented in this section is also applied to the first order statistical description of $\{\tilde{\mathbf{X}}_c[k]\}_{k=\{1,2,\dots\}}$, which was introduced in Section 4.2.3. There is however a slight difference; the CUSUM algorithm will only be applied to individual bands, so that a fair comparison with Lunetta et al.'s scheme is possible (no multispectral densities are considered). The modified CUSUM stopping time is given by

$$\mathbf{T}_h^{\text{CUSUM}} = \inf\{\mathbf{k} \geq \mathbf{0} | \mathbf{g}_k \geq \mathbf{h}_c\},$$

where

$$\mathbf{g}_k = \begin{cases} \{(g_{k-1}^b + s_k^b)^+\}_{b \in \{1, \dots, 7, \text{NDVI}\}} & k \neq 0 \\ \mathbf{y} \in \mathbb{R}^{+8} & k = 0 \end{cases}$$

and

$$s_k^b = \ln \frac{q_k^{1,b}(x^b(k))}{q_k^{0,b}(x^b(k))}. \quad (4.24)$$

Under normal CUSUM operating conditions $\mathbf{y} = \mathbf{0} = \{0, 0, 0, 0, 0, 0, 0, 0\}$. As soon as $g_k^b \geq h_c^b$ a change can be declared in band b . It is important to realise that the optimality of CUSUM (and the optimality of the sequential time-varying classifier) can no longer be guaranteed because of the following list of shortcomings:

1. The identically distributed assumption is violated by Equation 4.24.
2. The MODIS time-series does not consist of independent observations.
3. The densities $q_k^{0,b}$ and $q_k^{1,b}$ are estimated and not known beforehand.
4. In reality the MODIS pixels are spatially correlated.

The densities at each time-step can be estimated from ground truth data or via a trained CSHO simulator. Furthermore, it should also be obvious to the reader that the CUSUM algorithm presented here is nothing more than a repeated time-varying SPRT (see Section 4.3.3) [59].

4.4 CONCLUSION

The chapter presented the details of all the sequential and non-sequential hypertemporal classification and change detection algorithms that were investigated in this thesis. The chapter was divided into three main sections, namely simulation (Section 4.1), classification (Section 4.2) and change detection (Section 4.3). The chapter primarily dealt with a new stochastic inductive model, the CSHO (Section 4.1.2.2) and the model's possible application in simulation (Section 4.1.2.2), classification (Section 4.2.4.2) and change detection (Section 4.3.3). The experimental results of all the algorithms presented in this chapter will be given in Chapter 5. Note that the time-varying maximum likelihood classifier (with thresholds) in Section 4.2.3 and the CUSUM algorithm in Section 4.3.3 are sequential approaches.