IMPROVED HYPER-TEMPORAL FEATURE EXTRACTION METHODS FOR LAND COVER CHANGE DETECTION IN SATELLITE TIME SERIES

By

Brian Paxton Salmon

Submitted in partial fulfilment of the requirements for the degree

Philosophiae Doctor (Electronic)

in the

Faculty of Engineering, Built Environment and Information Technology

Department of Electrical, Electronic and Computer Engineering

UNIVERSITY OF PRETORIA

August 2012

SUMMARY

IMPROVED HYPER-TEMPORAL FEATURE EXTRACTION METHODS FOR LAND COVER CHANGE DETECTION IN SATELLITE TIME SERIES

by

Brian Paxton Salmon

Promoter:	Prof J.C. Olivier
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Philosophiae Doctor (Electronic)
Keywords:	classification, clustering, change detection, extended Kalman filter, Fourier transform, satellite, time series

The growth in global population inevitably increases the consumption of natural resources. The need to provide basic services to these growing communities leads to an increase in anthropogenic changes to the natural environment. The resulting transformation of vegetation cover (e.g. deforestation, agricultural expansion, urbanisation) has significant impacts on hydrology, biodiversity, ecosystems and climate. Human settlement expansion is the most common driver of land cover change in South Africa, and is currently mapped on an irregular, ad hoc basis using visual interpretation of aerial photographs or satellite images. This thesis proposes several methods of detecting newly formed human settlements using hyper-temporal, multi-spectral, medium spatial resolution MODIS land surface reflectance satellite imagery. The hyper-temporal images are used to extract time series, which are analysed in an automated fashion using machine learning methods. A post-classification change detection framework was developed to analyse the time series using several feature extraction methods and classifiers. Two novel hyper-temporal feature extraction methods are proposed to characterise the seasonal pattern in the time series. The first feature extraction method extracts Seasonal Fourier features that exploits the difference in temporal spectra inherent to land cover classes. The second feature extraction method extracts state-space vectors derived using an extended Kalman filter. The extended Kalman filter is optimised using a novel criterion which exploits the information inherent

in the spatio-temporal domain. The post-classification change detection framework was evaluated on different classifiers; both supervised and unsupervised methods were explored. A change detection accuracy of above 85% with false alarm rate below 10% was attained. The best performing methods were then applied at a provincial scale in the Gauteng and Limpopo provinces to produce regional change maps, indicating settlement expansion.

OPSOMMING

VERBETERDE HOË TYD-RESOLUSIE KENMERKONTTREKKINGSMETODES VIR DIE DETEKSIE VAN VERANDERING IN LANDBEDEKKING MET BEHULP VAN ’N SATELLIETTYDREEKS.

deur

Brian Paxton Salmon

Promotor:	Prof J.C. Olivier
Departement:	Elektriese, Elektroniese en Rekenaar Ingenieurswese
Universiteit:	Universiteit van Pretoria
Graad:	Philosophiae Doctor (Elektronies)
Sleutelwoorde:	klassifikasie, groepering, veranderingopsporing, uitgebreide Kalman-filter, Fourier-transform, satelliet, tydsreekse

Die groei in die globale bevolking veroorsaak verhoogde verbruik van natuurlike hulpbronne. Die behoefte om basiese dienste te lewer aan hierdie groeiende gemeenskappe lei tot ’n toename in antropogeniese veranderinge aan die natuurlike omgewing. Die gevolglike transformasie van plantbedekking (bv. ontbossing, landbou-uitbreiding, verstedeliking) het ’n beduidende impak op hidrologie, ekosisteme en die klimaat. Nedersettingsuitbreiding is die mees algemene oorsaak van landbedekkingsverandering in Suid-Afrika en informasie oor waar en wanneer nuwe nedersettings, voorkom word tans op ’n onreëlmatige basis bekom deur die visuele interpretasie van lugfotos of satellietbeelde. Hierdie tesis stel verskeie metodes voor vir die opsporing van nuutgestigte nedersettings met behulp van hiper-temporale, multi-spektrale, medium ruimtelike resolusie MODIS-grondoppervlakte reflektansie satellietbeelde. Die hiper-temporale beelde word gebruik om tydsreekse te onttrek, wat dan outomaties ontleed word met behulp van masjienleer metodes. ’n *Post*-klassifikasie veranderingopsporingsraamwerk is ontwikkel om tydsreekse te analiseer deur gebruik te maak van verskeie kenmerkonttrekkingsmetodes en klassifiseerders. Twee nuwe hiper-temporale kenmerkonttrekkingsmetodes word voorgestel om die seisoenale patroon in die reeks te karakteriseer. Die eerste kenmerkonttrekkingsmetode onttrek Seisoen Fourier-eienskappe

uit die tydsreeks, wat die temporale spektrum eienskappe van verskillende landbedekkingsklasse beklemtoon. Die tweede kenmerkonttrekkingsmetode onttrek toestand-ruimte vektore uit die tydsreeks, wat verkry word met behulp van ’n uitgebreide Kalman-filter. Die uitgebreide Kalman-filter is geoptimeer deur gebruik te maak van ’n nuwe maatstaf wat gebaseer is op die inligting in die ruimtelike-temporale domein. Die *post*-klassifikasie veranderingopsporingsraamwerk is geëvalueer met verskillende klassifiseerders; beide toesig en sonder-toesig metodes is ondersoek. ’n Veranderingopsporingsakkuraatheid bo 85% met ’n valsalarmkoers onder 10% is behaal. Die beste metodes is toegepas op ’n provinsiale skaal in die Gauteng- en Limpopo-provinsies om plaaslike veranderings kaarte te produseer.

This thesis is dedicated to:

God Almighty, for all the countless opportunities that He has given me;
My loving family and friends, thank you for all your love, support, and sacrifice throughout my life.

We all grow up with the weight of history on us. Our ancestors dwell in the attics of our brains as they do in the spiraling chains of knowledge hidden in every cell of our bodies. - Shirley Abbott

ACKNOWLEDGEMENT

The author would like to thank the following people and institutions, without whose help this thesis would not have been possible:

- The Council for Scientific and Industrial Research for supporting me on their PhD studentship programme.
- My study leader, Prof J.C. Olivier, for all the advice and guidance he has given me throughout the course of my studies.
- My co-promoters, Dr. Frans van den Bergh and Dr. Konrad Wessels, for all their insight, advice and help.
- My fellow student, Waldo Kleynhans, for all his useful suggestions and advice.
- The University of Pretoria's computer clusters maintained by Hans Grobler, which greatly aided in my simulations.
- Karen Steenkamp for providing me with the necessary data used for training and validation purposes.
- Willem Marais for providing me with custom developed image processing software.
- The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

LIST OF ABBREVIATIONS

Autocorrelation Function	ACF
Aikaike Information Criterion	AIC
Atmospheric Infrared Sounder	AIRS
Autocovariance Least Squares	ALS
Ante Meridiem	AM
Advanced Microwave Scanning Radiometer	AMSR
Advanced Microwave Sounding Unit	AMSU-A
Artificial Neural Network	ANN
Advanced Spaceborne Thermal Emission and Reflection radiometer	ASTER
Algorithm Theoretical Basis Document	ATBD
Advanced Very High Resolution Radiometer	AVHRR
Break For Additive Seasonal and Trend	BFAST
Broyden-Fletcher-Goldfarb-Shanno	BFGS
Best Matching Unit	BMU
Bidirectional Reflectance Distribution Function	BRDF
Bias-Variance Equilibrium Point	BVEP
Bias-Variance Score	BVS
Bias-Variance Search Algorithm	BVSA
Clouds and the Earth’s Radiant Energy System	CERES
Change Vector Analysis	CVA
Chandra X-ray Center	CXC
Coastal Zone Color Scanner	CZCS
Discrete Fourier Transform	DFT
Extended Kalman Filter	EKF

Expectation Maximization	EM
Earth Observation System	EOS
Earth Resource Technology Satellite	ERTS
Enhanced Thematic Mapper Plus	ETM+
Enhanced Vegetation Index	EVI
Foreign Argicultural Services	FAS
Fast Fourier Transform	FFT
Farm Service Agency	FSA
Gigabit	Gb
Gross Domestic Product	GDP
Group on Earth Observations	GEO
Global Earth Observation System of Systems	GEOSS
Geographical Information System	GIS
Global Positioning System	GPS
Hierarchical Data Format	HDF
High Resolution Infrared Spectrometer	HIRS
Humidity Sounder for Brazil	HSB
Inverse Discrete Fourier Transform	IDFT
Inverse Fast Fourier Transform	IFFT
Instantaneous Field of View	IFOV
Least Squares	LS
Line Spread Function	LSF
Linear Spectral Mixture Analysis	LSMA
Land Use Land Change	LULC
Multi-angle Imaging SpectroRadiometer	MISR
Multilayer Perceptron	MLP
MODerate-resolution Imaging Spectroradiometer	MODIS

Measurements of Pollution in the Troposphere	MOPITT
Multi-Spectral Scanner	MSS
National Aeronautics and Space Administration	NASA
National Argicultural Statistics Services	NASS
Normalized Difference Vegetation Index	NDVI
Near InfraRed	NIR
National Land Cover	NLC
Ordinary Least Squares	OLS
Principal Component Analysis	PCA
Post Meridiem	PM
Point Spread Function	PSF
Radial Basis Function	RBF
Red Green Blue	RGB
Resilient backpropagation	RPROP
Smithsonian Astrophysical Obervatory	SAO
Seasonal Fourier Features	SFF
Signal-to-Noise Ratio	SNR
Self Organizing Map	SOM
Satellite Pour l'Observation de la Terre	SPOT
Signal-to-Quantization Noise Ratio	SQNR
Sum of Squares Error	SSE
Support Vector Machine	SVM
Thematic Mapper	TM
United Nations	UN
United States Department of Argiculture	USDA
Vegetative Cover Conversion	VCC
Vegetation Index	VI

CONTENTS

Contents

CHAPTER ONE
INTRODUCTION

1.1 PROBLEM STATEMENT

Reliable monitoring of land cover and its transformation is an important component of environmental and natural resources management. Land cover is defined as the physical composition of material on the surface of the Earth, while land use is a description of how the land is used for socio-economic reasons [1]. Land cover is distinctly different from land use, but these two terms will be used interchangeably, as the focus of this thesis is the detection of land cover transformation of natural vegetation to newly formed human settlements. Several studies have investigated the global effects of anthropogenic activities on the planet, and it is estimated that more than a third of the Earth's land surface has been transformed by human activities [2]. The increase in human population is one of the major drivers of settlement expansion within geographical areas, which further increases the utilisation of the remaining natural resources [3]. Geographic information on land use and land cover change is highly sought after at local and global scales.

Land cover change often indicates land use change with major socio-economic impacts, while the transformation of vegetation cover (e.g. deforestation, agricultural expansion, urbanisation) has significant impacts on hydrology, ecosystems and climate [4, 5]. All these changes affect the environment and have a detrimental impact on the habitat of the human race. This raises the question whether the human's demand for natural resources is sustainable.

Sustainability is the long-term maintenance plan that will ensure the future of mankind's endeavours. The most widely quoted definition of sustainability and sustainable development was stated by the Brundtland Commission of the United Nations (UN) on March 20, 1987 as [6]:

> *Sustainable development is development that meets the needs of the present without compromising the ability of future generations to meet their own needs.*

The well-being of the environment is one of the major factors that contributes to sustainability. The UN General Assembly's discussion on sustainable human settlements concluded that countries' local governments need to plan, implement, develop, and manage human settlements [7]. It was further stated that the local government needs to manage existing settlements and prevent the establishment of any new unplanned settlements. The ability to determine where new settlements are formed, creates opportunities for the local government to provide running water supplies, sewage- and refuse removal services, which ties in directly with the UN's Millennium Development goals. The UN proposes a systematic development of sustainable cities for newly formed settlements. The South African government incorporated this vision into its local policies by focusing on service delivery to these newly formed settlements. Human settlement expansion is currently the most pervasive form of land cover change in South Africa [8]. Most of the new settlements are informal, unplanned and are usually built without the legal consent of the land owner [9, 10]. This thesis focuses on the detection of new human settlements formed in South Africa.

Satellite-based remote sensing is widely recognised by agencies, such as the United States Department of Agriculture (USDA)'s Farm Service Agency (FSA), the USDA's National Agricultural Statistics Services (NASS), and USDA's Foreign Agricultural Services (FAS), as a cost-effective method of acquiring information on the Earth's land surface [11]. Monitoring environmental dynamics, and classifying and detecting land cover change, require this type of cost-effective, systematic observations. The remote sensing science has thus progressed rapidly to meet the need to monitor global environmental change activities [12, 13]. Visually inspecting large volumes of high spatial resolution images for monitoring land cover is time-consuming and resource-intensive [14].

Earth observation satellites with wide swath widths provide the means of monitoring large areas on a frequent basis (high temporal resolution) [15]. These satellites are equipped with multiple coarse to medium spatial resolution sensors to record land surface information, in different spectral bands on a daily basis. Land cover surveillance of large geographical areas is augmented by the information inherent in the hyper-temporal satellite images, and therefore the analysis of these long-term data sets has attracted much attention [16, 17]. Owing to the complexity and non-parametric nature of land cover classification and change detection, machine learning methods are widely regarded as the most viable option for classification and change detection [14, 18]. The use of machine learning methods enables digital change detection, which encompasses the quantification of temporal phenomena from multi-date imagery that is most commonly acquired by satellite-based multi-spectral sensors [19].

Two types of land cover changes are usually investigated [20]: land cover modification and land cover transformation. Land cover modification is caused by internal changes within a particular land cover class. These changes affect the current state of the land cover class, but do not change the land

cover class, i.e. seasonal variation of natural vegetation. Land cover transformation of a particular geographical area involves change from one land cover class to another. This thesis focuses on land cover transformation of natural vegetation to newly formed human settlements, although the methods are applicable to other forms of land cover transformation. In the rest of this thesis the terms land cover transformation and land cover change are used interchangeably.

Change detection studies usually rely on image differencing, post-classification comparison methods, and change trajectory analysis [20–26], and the data are mostly treated as hyper-dimensional, but not necessarily as hyper-temporal. These methods therefore do not fully capitalise on the high temporal sampling rate which captures the dynamics of different land cover types. Satellites with high temporal acquisition rates provide information on the seasonal dynamics of a particular land cover type [15]. Incorporating the temporal information into a change detection algorithm allows the method to distinguish between land cover conversion and natural seasonal variations.

Main problem statement: *To detect land cover conversion of natural vegetation to newly formed human settlements reliably. The land cover change detection algorithm should incorporate temporal information to distinguish the change from seasonal variations. The land cover change detection algorithm should also be able to detect new human settlements that only span a small geographical area using coarse spatial resolution satellite imagery.*

1.2 OBJECTIVE OF THIS THESIS AND PROPOSED SOLUTION

Primary objective: *Develop a change detection algorithm that operates on multiple spectral bands, which exploits the richness of information inherent in hyper-temporal images*.

Secondary objective: *Develop a change detection algorithm that is sufficiently near automated, requiring minimal human interaction.*

As stated previously, machine learning methods are the more viable solution when analysing high dimensional data sets. A post-classification change detection approach detects change by classifying a geographical area into different classes over time. Land cover change is defined here as the transition in class label of a pixel's time series from one class to another class, after which it remains in the newly assigned class for the remainder of the time series [20]. A flow diagram for the proposed solution is shown in figure 1.1.

A set of images of a particular geographical area is obtained. The interval between two consecutive images must be short, which implies hyper-temporal acquisitions. The hyper-temporal images in this thesis were acquired by the MODerate resolution Imaging Spectroradiometer (MODIS) sensor

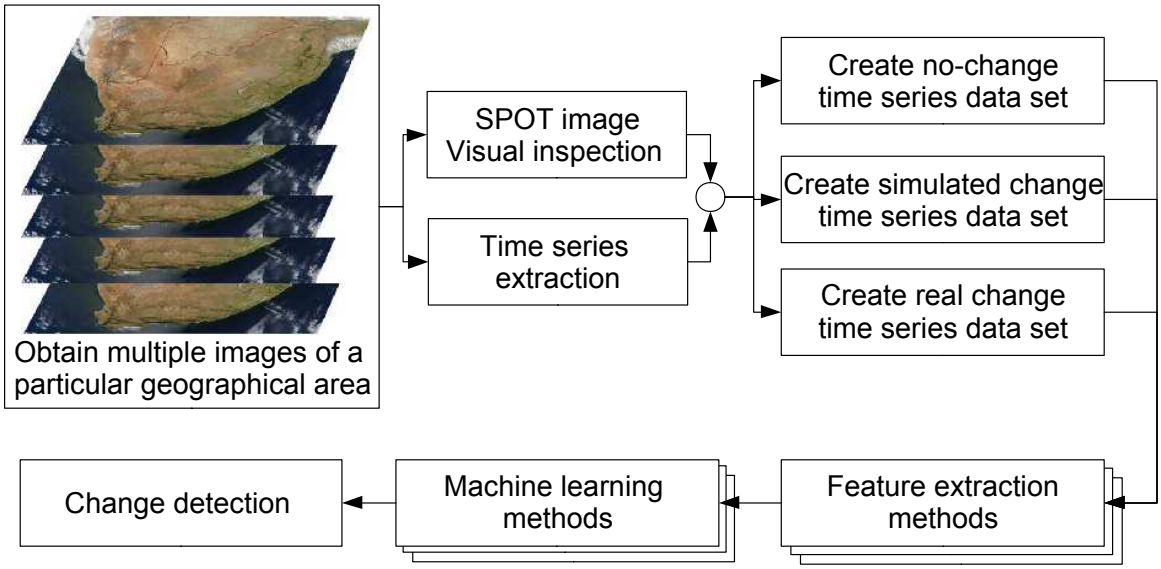


FIGURE 1.1: A flow diagram which depicts the steps followed to realise the proposed solution.

on board the Terra and Aqua satellites and are freely available. The MCD43A4 product provides hyper-temporal, multi-spectral (7 spectral bands) medium spatial resolution (500 metre) land surface reflectance data. The Bidirectional Reflectance Distribution Function (BRDF) correction models all the pixels in an image to a nadir view, which significantly reduces the anisotropic scattering effects of surfaces under different illumination and observation conditions [27, 28]. Time series of reflectance values were extracted for each spectral band over a particular geographical area (500 metre by 500 metre) from the multi-spectral hyper-temporal MODIS data set (February 2000 – January 2008).

Since the hyper-temporal images are coarse to medium spatial resolution, high spatial resolution satellite data are required for ground truth. Satellite Probatoire d'Obervation de la Terre (SPOT) images are high spatial resolution images, which are analysed by operators to identify areas that experienced land cover change or no land cover change.

Land cover change is a rare event on a regional scale and vital information, such as the date of change and rate of change, is usually not known. Therefore land cover change was simulated to enable a detailed assessment of change detection methods, which could not be performed on the real land cover change data set. A simulated land cover change time series data set is created by blending time series of two different land cover classes which did not change. The simulated land cover change data are used to test the functionality of the change detection methods, after which tests are performed on real examples of land cover change mapped using high spatial resolution images. Several contributions are made in this thesis that provide solutions to the primary and secondary objectives.

Contribution 1: *Develop of a novel land cover change detection method. The method is a post-classification approach and will operate on the Seasonal Fourier Features (SFF). SFF are*

hyper-temporal features extracted from time series.

The SFF are hyper-temporal features extracted without experiencing the usual pitfalls encountered with subsequence clustering [29]. The use of the SFF is then compared to another method proposed by Kleynhans *et al.* [30], referred to as the Extended Kalman Filter (EKF) feature extraction method. The drawback with this method is that it requires an offline optimisation phase, which must be performed by an operator. This does not satisfy the secondary objective (full automation) of this thesis, but has shown promising results.

Contribution 2: *Extend the EKF feature extraction method to a higher dimensions to improve change detection capabilities.*

The second objective concerned with full automation of the EKF extraction method is addressed in the following contribution.

Contribution 3: *Propose a novel criterion that is referred to as the Bias-Variance Equilibrium Point (BVEP). The BVEP is the point where the tracking of the reflectance values within time series are improved and the internal stability of the EKF is optimised. Define a Bias-Variance Score (BVS) that will measure the current system in relation to the BVEP.*

The BVEP criterion also provides statistical information on the seasonal vegetation activity cycle, which could provide vital insight into environmental dynamics [31, 32]. The optimisation of the BVS requires an unsupervised search method, which adjusts the variables to satisfy the BVEP criterion.

Contribution 4: *Design a new search algorithm, referred to as the Bias-Variance Search Algorithm (BVSA), that can effectively optimise the BVS to the BVEP criterion for optimal EKF performance.*

1.3 OUTLINE OF THESIS

The outline of the thesis is as follows:

- Chapter 2 gives a brief overview of the study area and an introduction to remote sensing principles. The chapter discusses several trade-offs that should be considered when selecting a sensor to solve the problem statement. The chapter concludes with an overview of some of the most common change detection methods found in the literature.

- Chapter 3 gives an introduction to supervised classification and in particular the Multilayer Perceptron (MLP). The chapter further discusses the pursuit of acceptable performance, and concludes with an overview on design considerations for a supervised classifier.
- Chapter 4 gives an introduction to unsupervised classification and provides several motivations for using an unsupervised classifier. The chapter also covers the disadvantages of unsupervised clustering and methods to mitigate them with proper cluster design.
- Chapter 5 defines four different feature extraction methods and their application to time series. These features are expected to provide good separation between natural vegetation and human settlement signals.
- Chapter 6 introduces the novel SFF and provides an in-depth investigation of the limitation of time series analysis mentioned by Keogh and Lin [29]. The chapter concludes with evidence of how the SFF provides a solution to this limitation.
- Chapter 7 introduces the BVEP, BVS, and Bias-Variance Search Algorithm (BVSA) used to optimise the EKF, in order to improve the quality of the extracted features.
- Chapter 8 presents the results of all experiments conducted in the thesis. These experiments report on classification accuracies, and change detection accuracies. These experiments are first conducted on a labelled data set within a particular province, and then expanded to run on a complete province, the Gauteng and Limpopo provinces of South Africa.
- Chapter 9 gives concluding remarks, as well as suggesting possible future research that could expand on the concepts introduced in this thesis.

CHAPTER TWO

REMOTE SENSING USED FOR LAND COVER CHANGE DETECTION

2.1 OVERVIEW

Remote sensing is the acquisition of information about an object without any direct contact with the object [33, Ch. 1]. Sensors are usually used to measure reflected wavelengths obtained from an object, which are then analysed for specific applications. A satellite-based sensor measures the reflected electromagnetic radiation of the Earth's surface and these measurements are then used to infer changes in surface reflectances caused by either environmental dynamics or anthropogenic activities.

Many international organisations and national governments have identified remote sensing as a beneficial field of study, and have made major joint investments in building better Earth observation systems. The objective of this chapter is to give the reader insight on how satellite-based sensors can be used to detect the formation of new human settlements on the Earth's surface.

2.2 SPONTANEOUS SETTLEMENTS

The standard of living in a country usually improves when sustainable economic growth is maintained. The government pursues a variety of projects to control the quality of economic growth [34]. Economical growth in developing countries is usually constrained by the lack of skilled labour, availability of resources, and necessary equipment. This lack of progress is aggravated by the pressure of a rapid growth in population and a backlog in housing development projects [9].

This backlog creates a shortage in the supply of affordable houses to the public, which results in the construction of temporary dwellings. These temporary dwellings are usually built without the legal consent of the land owner. The construction of temporary dwellings is not region-specific and has become a global phenomenon, although different characteristics are observed in the development of

these dwellings in each region [35]. A cluster of such temporary dwellings is formally known as a spontaneous settlement [9], which is a form of informal settlement [36, 37].

Social, economical, and political processes drive the migration of communities to certain regions, which often results in the development of informal settlement. This motivates the need for the local government to progressively track settlement expansion and migration [38, 39]. Settlement expansion is currently mapped on an irregular, ad hoc basis at great financial cost, using expensive visual interpretation of aerial photographs or satellite images. Regional information on settlement expansion gives the government the ability to plan the provision of services such as water, sanitation and electricity to these new or growing communities.

The behaviour of urban settlement migration and expansion has been empirically studied and predicted in various studies, but for several reasons cannot be applied to spontaneous settlements [9]. In this thesis no prior assumptions are made when attempting to find new or expanding settlements other than the decrease in seasonal behaviour associated with settlements.

Another motivation for tracking these spontaneous settlements is that their formation is currently one of the most pervasive forms of land cover change in South Africa [40]. The transformation of natural vegetation by practises such as deforestation, agricultural expansion and urbanisation has significant impacts on hydrology, ecosystems and the climate [4, 5, 41]. The area of interest in this thesis is the Limpopo province and Gauteng province located within South Africa.

2.2.1 Limpopo province

The Limpopo province is situated in the northern part of South Africa (Figure 2.1). The name of the province was derived from the river that separates South Africa from its neighbouring countries, Zimbabwe and Botswana. The province shares its southern borders with the Mpumalanga, Gauteng and North-West provinces.

The province is largely covered by natural vegetation, which is used for grazing by cattle and wildlife. It houses the largest hunting industry in South Africa. The province is also rich in numerous different tea and coffee plantations. The area is cultivated, with a range of agriculture focused on sunflowers, cotton, maize, peanuts, bananas, litchis, pineapples and mangoes.

The government departments within the province cannot currently capture and process all the necessary data on the different land cover types and anthropogenic activities throughout the province. This constraint is brought about by a limited budget, which motivates the pursuit of a less expensive alternative. Remote sensing (section 2.3) has been adopted by several governments as a less expensive option to augment the current processes of gathering information. If the government had access to more complete information, it could assist in the development of a management system to control and

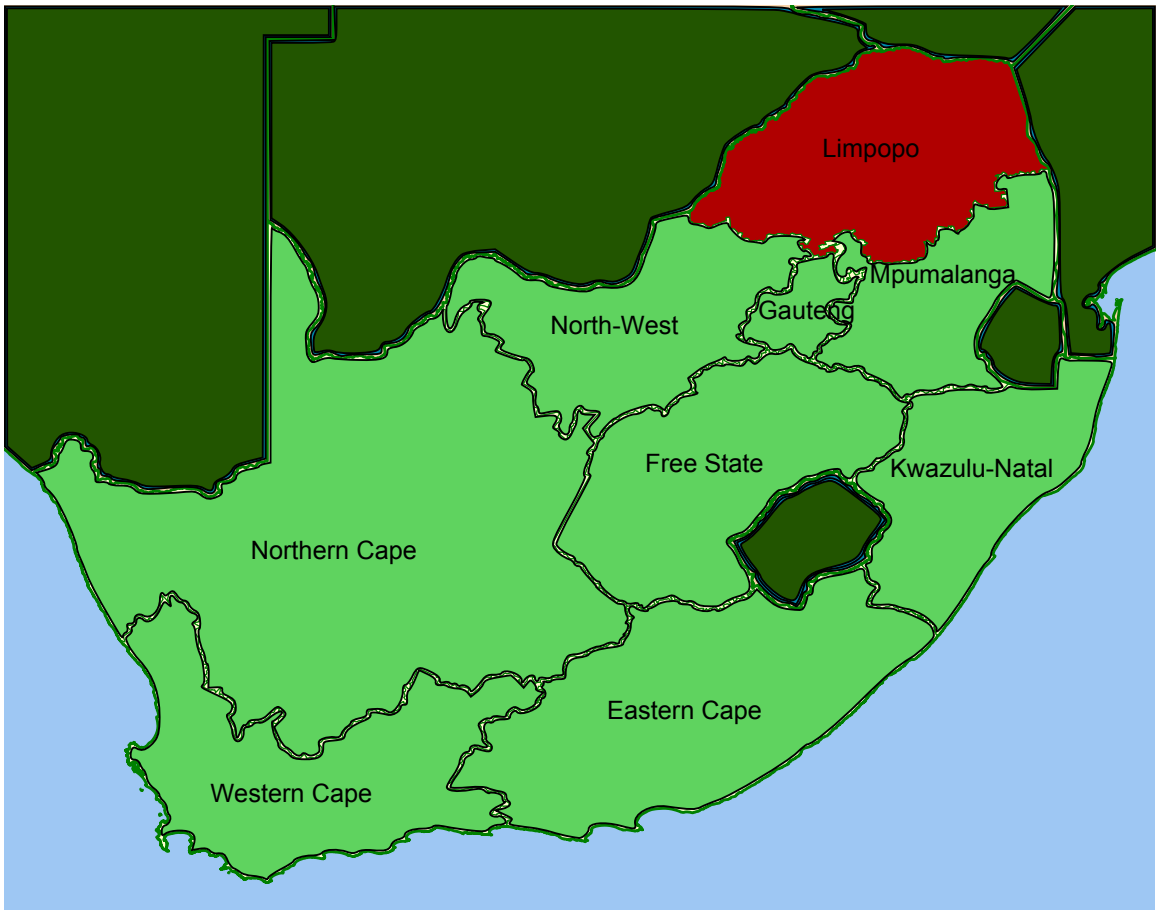


FIGURE 2.1: The Limpopo province is located in the northern part of South Africa.

monitor resources for the people throughout the province.

2.2.2 Gauteng province

The Gauteng province is situated in the highveld of South Africa (Figure 2.2). The name Gauteng comes from the Sesotho (indigenous language) word meaning *place of gold*. This is a common reference to the gold discovered in the city of Johannesburg in 1886. The province shares its borders with the Limpopo, Mpumalanga, North-West, and Free State provinces.

Gauteng is a landlocked province in the highveld, which is a high-altitude grassland. The province is the most urbanised one in the country. The province houses 20% of the country's population and only covers 1.4% of the country's total land area. A total population growth of over 30% was recorded between the years 2001 and 2010. Even though small in size, the province contributes 33.9% of South Africa's gross domestic product (GDP), which equates to 10% of the entire African continent.

In May 2008, the South African government identified problems caused by the massive influx of foreign nationals and provincial migration towards the Gauteng province. These problems range from

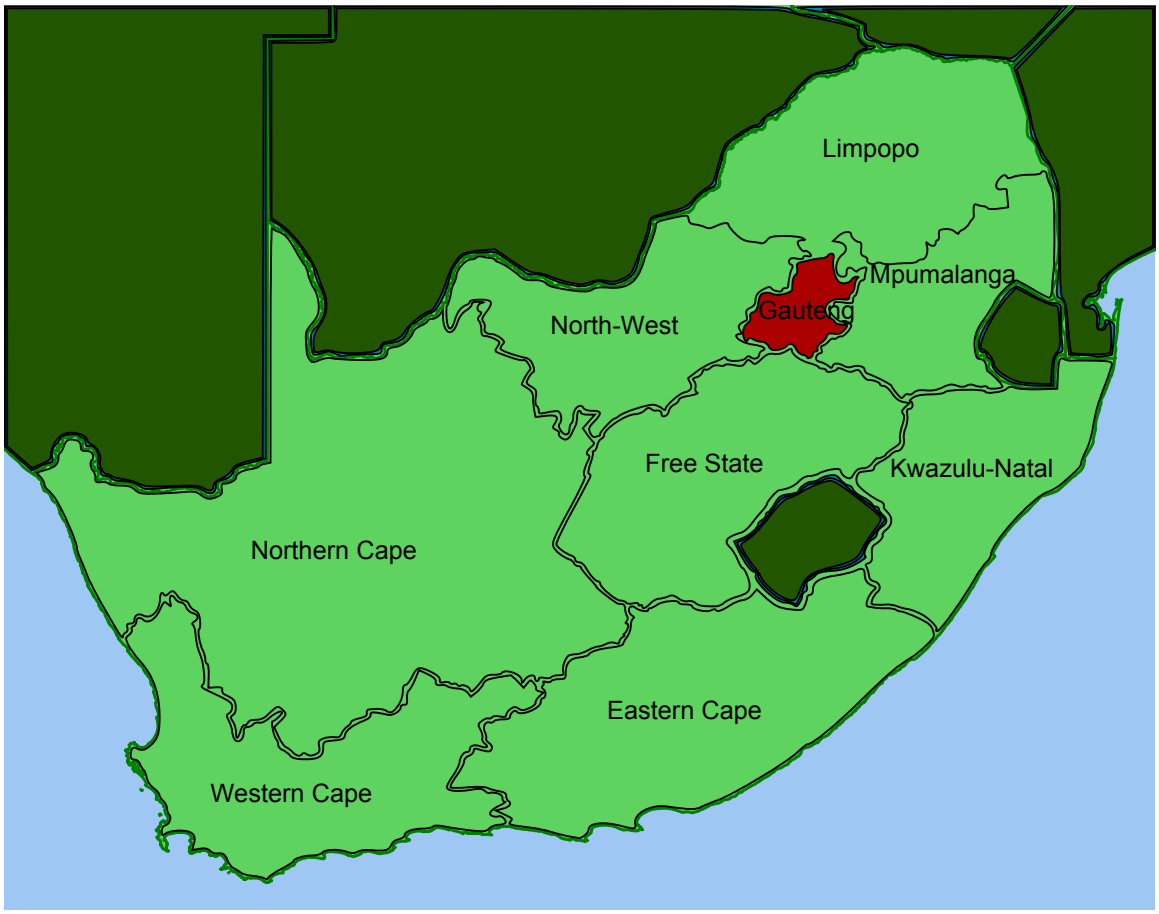


FIGURE 2.2: The Gauteng province is located in the highveld of South Africa.

social integration of multiple different cultures to proper service delivery. The active migration is motivated by a high median annual income for working adults and diverse employment opportunities. The province is rapidly growing to house cities that will be among the largest in the world. A projected population of 15 million people is expected by the year 2015.

2.3 OVERVIEW OF REMOTE SENSING

The Earth's surface is continually undergoing transformation caused by environmental change and anthropogenic activities. Many environmental problems stem from this continual transformation, of which some are; water shortage, soil degradation, greenhouse gas emissions, deforestation, biodiversity loss, etc. [33, Ch. 1].

The ability to evaluate the environmental dynamics will require periodic observation for analysis. Remote sensing is formally defined as the analysis of remotely acquired information on a particular object. This is usually accomplished using a sensor that is not in direct contact with the object [42, Ch. 1].

Earth observation satellites are non-military reconnaissance satellites that are used by the remote sensing community to acquire periodic observations of the Earth. These satellites use sensors to capture electromagnetic radiation which is reflected from or emitted by the Earth. The first Earth observation satellite that was developed was the Earth Resource Technology Satellite (ERTS-1), which was renamed to Landsat 1. It was designed to acquire multi-spectral medium resolution imagery on a systematic and recurring basis [43, Ch. 1].

Numerous additional remote sensing systems were commissioned and deployed through various agencies around the world after the success of the ERTS-1 mission. The Group on Earth Observations (GEO) was created in February 2005 to unite 60 national governments and 40 international organisations to implement the Global Earth Observation System of Systems (GEOSS). The main objective is to create high-quality, long-term, global observations in a timely fashion at minimal cost. The GEOSS system will ultimately monitor all aspects of the Earth's system to study global change.

A host of nations have launched hundreds of satellites into orbit since 1957, and this created a range of specifications that must be considered when choosing a sensor on a satellite for a specific application [43, Ch. 2]. The various permutations of the specifications are passive versus active sensors, the range of electromagnetic spectrum sensed, spectral bandwidth of each sensor, temporal acquisition rate, spatial resolution, radiometric resolution, etc. These specifications are discussed in successive sections along with the interaction of various components within a remote sensing system.

2.4 ELECTROMAGNETIC RADIATION

Electromagnetic radiation is a disturbance produced by an oscillation or acceleration of an electric charge. This disturbance consists of electromagnetic waves that comprise electric and magnetic fields which propagate perpendicular to one another with a set of time and spatial properties.

The electromagnetic wave oscillates through a medium with successive cycles and the distance between each completed cycle is called a wavelength. The energy density of the wave is defined by the amplitude. All electromagnetic waves radiate to the same wave theory and travel at the speed of light in a vacuum.

The electromagnetic wave acts according to its wavelength when it comes into contact with an object and can either reflect, refract, diffract or interfere. Electromagnetic radiation is classified into several categories according to wavelength: long waves, radio waves, microwaves, infrared, visible, ultraviolet, X-rays and Gamma rays. The categorised wavelengths are shown in figure 2.3.

One of the major sensor specifications on board a satellite is the deployment of either an active or passive sensor. An active sensor illuminates a scene with its own source of electromagnetic radiation.

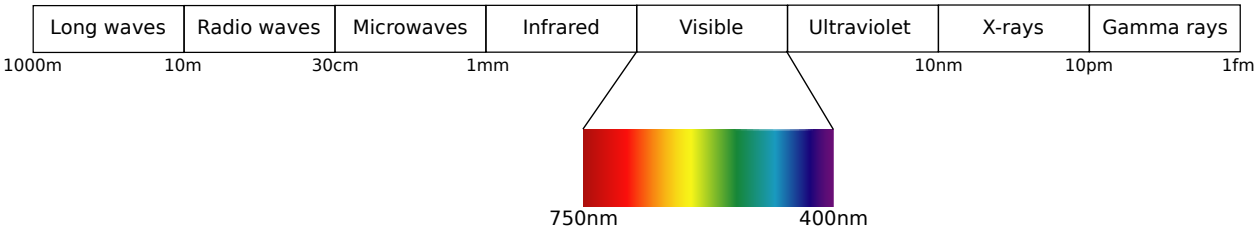


FIGURE 2.3: The electromagnetic spectrum [42, Ch. 1].

The source is set to a range of wavelengths of interest, which is typically in the 2.4 cm–107 cm range.

A passive sensor relies on the sun's radiation to illuminate a scene. A passive sensor is also called an optical sensor, as it operates in the visible and infrared spectrum. The visible spectrum is the most popular range in the electromagnetic spectrum, as it can be sensed by biological organisms.

The properties of the sun's radiance are of importance for a passive sensor, as it produces a wide range of wavelengths with a non-uniform energy distribution. Planck's law states that the spectral radiance is a function of the object's temperature and wavelength of the electromagnetic radiation [44]. The sun's peak emission is in the 400 nm–750 nm spectrum range, which is referred to as the visible spectrum. The spectral distribution across the spectrum remains relatively unchanged as it propagates through space [43, Ch. 2], but the reduction in intensity is subjected to the inverse-square law of the distance between the sun and the Earth [44].

2.5 EARTH'S ENERGY BUDGET

The Earth receives incoming energy from the sun and stars, while losing energy either through absorption, reflectance and transmittance [45, 46]. The conservation of energy states that an equilibrium between the incoming and outgoing energy must be preserved. This equilibrium is a function of the wavelength λ and is expressed as

$$\mathrm{E_I}(\lambda) = \mathrm{E_R}(\lambda) + \mathrm{E_A}(\lambda) + \mathrm{E_T}(\lambda), \tag{2.1}$$

where $\mathrm{E_I}(\lambda)$ denotes the incoming energy, $\mathrm{E_R}(\lambda)$ denotes the reflected energy, $\mathrm{E_A}(\lambda)$ denotes the absorbed energy and $\mathrm{E_T}(\lambda)$ denotes the transmitted energy. The total flux of the incoming energy $\mathrm{E_I}(\lambda)$ is a combination of solar radiation, geothermal energy, tidal energy (moon gravity) and heat energy (fossil fuel consumption). The outgoing energy is partitioned into either reflected, absorbed or transmitted radiation. The partitioning of the outgoing energy into either reflected, absorbed or transmitted radiation varies for different wavelengths, atmospheric conditions and geographical properties [42, Ch. 1].

A sensor on board a satellite measures only the reflected energy $\mathrm{E_R}$; to put the emphasis on the reflected energy, equation (2.1) is rewritten as

$$\mathrm{E_R}(\lambda) = \mathrm{E_I}(\lambda) - \mathrm{E_A}(\lambda) - \mathrm{E_T}(\lambda). \tag{2.2}$$

Approximately 30% of all incoming energy is reflected back into space. The contributions made to the reflected energy by geothermal energy, tidal energy and heat energy are negligibly small when compared to the reflected solar radiation [42, Ch. 1]. The average reflectance of 30% of the incoming energy $\mathrm{E_I}(\lambda)$ is further subdivided: atmospheric reflectance of 6%, cloud reflectance of 20% and the Earth's surface reflectance of 4% [47–49]. A brief overview is given of all the interacting media within the energy budget in the following sections.

2.5.1 Interaction with the atmosphere

Electromagnetic radiation penetrates the atmosphere, which consists of five layers of gases that are retained by the planet's gravitational field [50]. Power and spectral properties of electromagnetic radiation are altered as they propagate through the atmosphere. The atmosphere can either scatter or absorb electromagnetic radiation. The five layers of atmosphere are; the exosphere, thermosphere, mesosphere, stratosphere and troposphere.

The exosphere is the outer layer of the atmosphere. It is a very thin layer where the atoms and molecules leave the atmosphere and dissipate into outer space.

The thermosphere is the second layer that electromagnetic radiation penetrates and this is where most of the Earth Observation satellites orbit. The thermosphere extends between 90 km and 1000 km above sea level. The temperature in the layer is strongly affected by solar activities.

The mesosphere is the middle layer of the atmosphere and extends between 50 km to 90 km above sea level. The majority of the meteors originating from outer space burn up in this layer. It is difficult to measure the properties of the mesosphere, as only sounding rockets can be used at these altitudes.

The stratosphere is the second closest layer to the Earth's surface and is positioned at an altitude of between 8 km and 50 km. The ozone layer is situated within the stratosphere and absorbs most of the harmful solar radiation. An aircraft can fly through the stratosphere because of the temperature stratification within the layer.

The troposphere is the closest layer to the surface of the Earth and rises up to 20 km above sea level. Most weather activities occur within this layer, which holds nearly all water vapour and dust particles. Solar electromagnetic radiation heats up the surface of the Earth and in turn is transferred back to the troposphere.

The atmosphere alters the intensity and spectral composition of electromagnetic radiation before it

is sensed by a sensor on board a satellite. These effects are mainly categorised into either atmospheric scattering or absorption [42, 43].

2.5.1.1 Atmospheric scattering

The principal mechanisms affecting electromagnetic radiation as it propagates through the atmosphere are the scattering and absorption effects. Atmospheric scattering occurs when solar radiation is randomly diffused within the atmosphere. The behaviour of atmospheric scattering is determined by analysing the ratio of the particle's diameter to the wavelength of the electromagnetic wave. Atmospheric scattering is classified into three general categories [42, 43];

- *Rayleigh scattering* is the most common scattering effect in the atmosphere. This scattering occurs when a particle's diameter is much smaller than that of the interacting electromagnetic wave. Rayleigh scattering is inversely proportional to the fourth power of a radiating wavelength. This means that shorter wavelengths are more prone to scatter in the atmosphere than longer wavelengths.
- *Mie scattering* occurs when a particle's diameter is equal to an electromagnetic wave's wavelength. The major causes of Mie scattering are: pollen, dust, smoke, water vapour, and other particles situated in the lower portion of the atmosphere.
- *Non-selective scattering* occurs when an atmospheric particle's diameter is much larger than a radiating wavelength. Non-selective scattering mostly affects the visible, near infrared and mid-infrared spectrums. In this case, all the wavelengths are scattered equally regardless of their length. Non-selective scattering is found in water droplets, which give clouds and fog a white appearance.

2.5.1.2 Atmospheric absorption

Atmospheric absorption is caused by gaseous components that retain electromagnetic radiation within the atmosphere. Atmospheric absorption allows different wavelengths to be absorbed in different parts of the atmosphere. This absorption rate into different layers is illustrated in figure 2.4. The gases that absorb most solar radiation are: water vapour, carbon dioxide, and ozone [42, 43].

Earth observation satellites are limited, as they can only acquire images from wavelengths that are not absorbed into the atmosphere. The range of wavelengths that is not absorbed into the atmosphere is commonly referred to as the *atmospheric window* [42, Ch. 1]. A spectral sensor is usually set to measure a narrow band of spectrum within the atmospheric window.

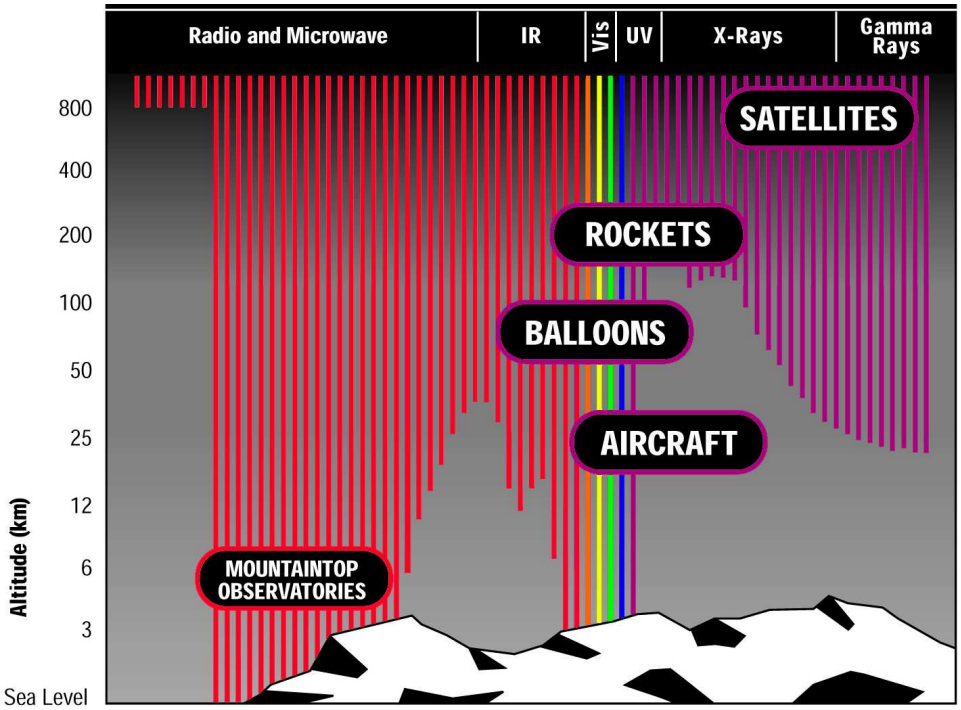


FIGURE 2.4: Atmospheric absorption allows different wavelengths to be absorbed in different parts of the atmosphere. This figure shows the different elevations at which electromagnetic radiation is absorbed into the atmosphere. Image supplied by NASA/CXC/SAO.

2.5.1.3 Atmospheric correction

The electromagnetic radiation recorded at a sensor is not a true reflection of the Earth's surface owing to the effects of atmospheric scattering and absorption. A critical preprocessing step for creating oceanic and land surface products is the correction of these atmospheric disturbances [51, 52].

Two general methods are used in correcting atmospheric disturbances: relative and absolute correction. Relative atmospheric correction is exactly as the term implies a relative histogram match of an image to a reference image. This method requires an accurate reference image for a specified geographical area and any adjoining areas.

Absolute atmospheric correction is further subdivided into empirical and physical methods. The absolute empirical method is not popular, as it has a tendency to over-simplify the corrections applied to an image.

The absolute physical method, on the other hand, uses a mathematical model to extract the effects of various gaseous components and then to compensate for these effects accordingly. A radiative transfer model is a form of the absolute physical method which extracts the gaseous concentrations directly from an image in order to estimate the corrected radiance for the image.

2.5.2 Interaction with the Earth's surface

The Earth's surface interacts with incoming electromagnetic radiation and can either absorb, reflect and/or transmit the radiation. The reflected electromagnetic radiation excites the components within the sensor. The amount of reflected electromagnetic radiation is a function of the wavelength and the properties of the surface. The surface has several properties that affect the amount of reflectance: mineral profile, surface contour, surface roughness, etc. Reflected electromagnetic waves are mostly affected by the surface's roughness and are divided into two general modes: specular (smooth) and diffuse (rough or Lambertian) [33, Ch. 4].

The Rayleigh criterion determines the level of roughness for a medium and is calculated as

$$h \leq \frac{\lambda}{8\cos(\theta)}. \tag{2.3}$$

The variable h denotes the surface irregularity height, λ denotes the wavelength and θ denotes the angle of incidence measured to the azimuth. If equation (2.3) is satisfied, then the surface is considered to be diffuse, otherwise it is specular [42, 43].

A specular surface reflects electromagnetic radiation according to Snell's law, which states that the outgoing energy is exactly reflected at a perpendicular angle to the azimuth of the incoming energy. A diffuse surface reflects the incoming electromagnetic radiation in all directions off the surface. A Lambertian (perfect diffuse) surface reflects the incoming energy uniformly in all directions off the surface.

Most natural surfaces are imperfect diffuse reflectors (specular component present) in the visible and near infrared spectrum. This makes remote sensing possible, as reflected electromagnetic radiation can be captured at most viewing angles. This would not be possible if the surface was completely specular, as it would have a high reflectance value at a single specific viewing angle and relatively low reflective values at all other viewing angles [53, 54].

2.5.3 Interaction with a satellite-based sensor

The principal concept of remote sensing is to observe an object remotely. In a satellite-based application it is the recording of electromagnetic radiation that has interacted with an object. A sensor, as defined in this thesis, is a device that measures a physical quantity and converts it into an electrical signal.

The advantage, when considering the interaction of radiation with the sensor, is that it can be designed to measure the environment optimally. A satellite sensor's specifications that will be discussed briefly are: the spatial, spectral, radiometric and temporal resolutions.

Spatial resolution is the geographical size that is recorded on a two-dimensional pixel in the image. The size of the area represented in a pixel is determined by the altitude, viewing angle and sensor characteristics. All these characteristics are influenced by the instantaneous field of view (IFOV) of the sensor [33, Ch. 4]. The IFOV of the sensor is time-dependent, as the satellite is not perfectly stable in its orbit. The distance between the satellite and the Earth varies continually, altering the physical size of the geographical area that is captured within a single pixel.

Another limiting factor is the point spread function (PSF) of the sensor. The PSF is the system impulse response between the geographical area and the sensor. This function describes the degree of illumination spreading from the adjacent area to the geographical area of interest. The PSF results in a blending or spreading effect on areas with relatively bright or dark objects within the IFOV of the sensor. This leads to high contrast features becoming indiscernible on satellite images even though their widths are less than the sensor's spatial resolution.

Spectral resolution is the bandwidth of the electromagnetic spectrum recorded by the sensor. A sensor that senses a shorter spectrum range of wavelengths (smaller bandwidth) has an improved ability to capture the spectral signature of an object within the spectral band when compared to a sensor that measures a larger spectrum range of wavelengths (larger bandwidth).

The disadvantage of increasing the spectral resolution is that the signal-to-noise ratio (SNR) decreases. Recorded radiance at the sensor is adversely affected by some form of noise. The physical propagation of electromagnetic radiation to the sensor can be seen as a time-variant multi-path propagation of the reflected electromagnetic wave of a geographical area with a certain level of additive noise. The additive noise in the sensor is made up mostly of thermal noise. The thermal noise does not decrease if a smaller bandwidth is sensed, although the instantaneous radiance in the sensor is reduced for a higher spectral resolution sensor as it is exposed to a shorter range of spectrum. The thermal noise remains the same regardless of the range of spectrum that is being sensed. To summarise: reducing the reflected power within the sensor (reducing the bandwidth) will inadvertently reduce the SNR.

Optimal spectral resolution is obtained when a sensor mitigates the effect of additive noise and has a spectral bandwidth that captures the best matched spectral signature for the intended remotely sensed object. Remote sensing systems usually use multi-spectral or hyper-spectral sensors. This is an array of sensors that capture different ranges of spectrum at the same time. A multi-spectral sensor has less than 100 unique spectral bands, while a hyper-spectral sensor has more than 100.

Radiometric resolution is the accuracy of converting electromagnetic radiation at the satellite sensor

to a digital binary format. A higher radiometric resolution enables the satellite sensor to distinguish between more levels of intensity.

It is possible to encode electromagnetic radiation as an information source at a rate that is close to its entropy [55, Ch. 6]. This is unfortunately limited by the storage space available on the satellite, which induces a certain level of distortion in the sampling of the electromagnetic radiation. The reason is that electromagnetic radiation is an analog source and requires an infinite number of binary bits to store.

A loss in precision is caused by the finite storage space, which induces a distortion that is directly related to the number of quantisation levels (number of binary bits per radiance sample). It should be noted that the number associated with each quantisation level is not a direct measure of the captured electromagnetic radiation, but rather the steps into which a range of physical values is divided.

In an effort to distribute the captured electromagnetic radiation more evenly over the range of quantisation levels, some sensors apply either non-linear quantisation mapping functions or an amplifier with an automatic gain control mechanism. This alters the intensity of the captured electromagnetic radiation and distributes it over a range of different quantisation levels without creating a saturated buffer in the remotely sensed image.

The total number of quantisation levels and the method of distributing radiation across the levels affect the level of distortion in the stored values. This rate of distortion is defined by the signal-to-quantisation-noise ratio (SQNR), which is expressed as

$$\mathrm{SQNR} = \frac{P_x}{P_{\tilde{x}}}. \tag{2.4}$$

The variable $P_{\tilde{x}}$ is the quantisation-noise power and P_x is the power of the radiation before quantisation.

Low-quality sensors have low SQNR, which equates to low radiometric resolution. The disadvantage in increasing the radiometric resolution is the costs and complexity of adding a higher resolution analogue-to-digital converter device and the increase in required storage space for storing the binary values of the digital image. For example, the Quickbird satellite owned by DigitalGlobe has a radiometric resolution of 11 bits. This enables the sensor to distinguish between 2048 (2^{11}) levels of radiance. The satellite has 128 Gb storage capacity, which equates to 57 images stored on board. The sensor can distinguish between 65536 (2^{16}) levels of radiance if the radiometric resolution is set to 16 bits. The problem is that only 39 images can be stored on board, which results in a 32% reduction in storage capacity.

Temporal resolution is the periodic rate of acquisition of a geographical area by the same satellite sensor. This is important for investigating any change in land surface and the monitoring of global environmental processes. The orbit, altitude, swath width, and priority tasking of the sensor on board

the satellite determines the temporal rate at which an area of interest can be imaged [42, Ch. 6]. Sensors are tasked from a mission control center to acquire images of geographical areas. Areas of interest are assigned a priority task, which improves the temporal acquisitions for this area. The temporal resolution varies from less than an hour to more than a few months [43, Ch. 2]. Fixed temporal resolution is a sensor that has a fixed viewing angle, repetitive orbital track and a fixed swath width.

The swath width is the trade-off between the temporal resolution and the spatial resolution. The wider the swath width, the shorter the revisit time period for a geographical area, while the narrower the swath width, the better the spatial resolution (for the same number of pixels).

TABLE 2.1: Specification of different remote sensing sensors.

Sensor	Temporal resolution (Revisit period)	Spatial resolution	Wavelength range	Number of spectral bands
Enhanced Thematic Mapper Plus (ETM+)	16 days	15 m – 60 m	0.45 μm–12.50 μm	8
MODerate-Resolution Imaging Spectroradiometer (MODIS)	1–2 days	250 m – 1000 m	0.405 μm–14.385 μm	36
Advanced Very High Resolution Radiometer (AVHRR)	Daily	1100 m – 4000 m	0.58 μm–12.50 μm	5

How to choose a sensor: This thesis focuses on expanding settlements. Finding newly developed housing requires several considerations when selecting the right remote sensing sensor.

High spatial resolution sensors have the ability to detect much smaller objects in an area. The drawback is that higher spatial resolution means lower temporal resolution. These images are thus not regularly acquired and are financially expensive.

Detecting new settlements is possible when comparing two high spatial resolution images taken at two different dates. The problem is that similar land cover types can appear significantly different at various times of the year. These seasonal changes in the land cover can be mitigated if the temporal resolution is high enough to capture these trends [15]. This makes the use of high temporal resolution sensors much more useful for change detection.

A list of specifications for three different satellites used to image the land surface is shown in table 2.1. The specifications for these three satellites are used to illustrate the range of trade-offs to consider when selecting a sensor.

The Enhanced Thematic Mapper Plus (ETM+) operates on a very high spatial resolution of 15 m – 60 m, with a low temporal revisit time of 16 days.

The Advanced Very High Resolution Radiometer (AVHRR) has a high temporal resolution of one day, but captures a geographical area at a spatial resolution of 1100 metres. The large swath width is necessary to obtain a high temporal resolution at the expense of the spatial resolution.

The MODerate-resolution Imaging Spectroradiometer (MODIS) is a newer instrument, which was

specifically designed for global land surface monitoring and is the chosen sensor for this study, as it has a high temporal resolution and medium spatial resolution capabilities [16]. MODIS has a temporal resolution of 1–2 days, which is close to the temporal resolution of the AVHRR sensor. MODIS also has a medium spatial resolution (250 m – 1000 m) and a wider variety of spectral bands.

2.6 MODERATE RESOLUTION IMAGING SPECTRORADIOMETER

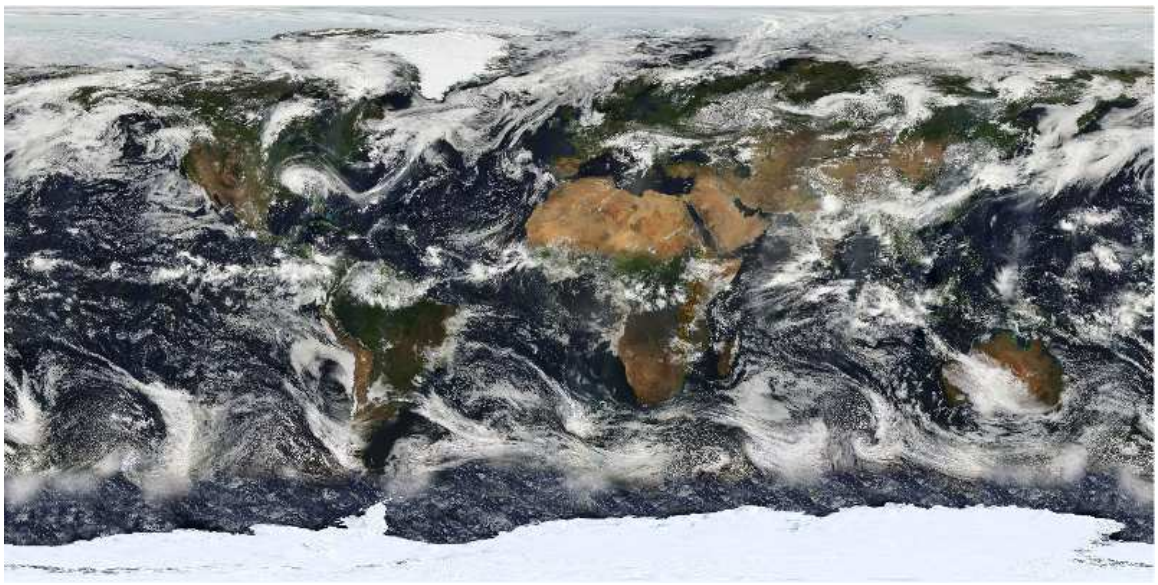


FIGURE 2.5: Multiple MODIS images concatenated to form a image of the Earth.

MODIS is an experimental scientific sensor launched into the Earth's thermosphere by NASA on board the Terra EOS-AM-1 satellite on December 18, 1999. A second MODIS sensor was launched on board the Aqua EOS-PM-1 satellite on May 4, 2002.

The Terra EOS satellite was the first NASA scientific research satellite to carry the MODIS instrument into orbit. The Terra satellite was launched from the Vandenberg Air Force base into a sun-synchronous orbit at an altitude of 705 km [56]. Terra is Latin for *Earth*. The Terra EOS satellite carries a total of five remote sensing sensors which record measurements of the Earth's climate system: Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER), Clouds and the Earth's Radiant Energy System (CERES), Multi-angle Imaging SpectroRadiometer (MISR), MODIS and Measurements of Pollution in the Troposphere (MOPITT).

The Aqua EOS satellite was the second NASA scientific research satellite to carry a MODIS instrument into orbit. The Aqua satellite was launched from the Vandenberg Air Force base into an afternoon equatorial crossing orbit at an altitude of 705 km [56]. Aqua is Latin for *water*. The Aqua EOS satellite carries a total of six remote sensing sensors that collects information about the Earth's

TABLE 2.2: MODIS spectral bands properties and characteristics.

Spectral bands	Wavelengths (nanometres)	Resolution (metres)	Property or characteristic	Spectral range
Band 1	620–670	250	Absolute Land Cover Transformation, Vegetation Chlorophyll	Visible (Red)
Band 2	841–876	250	Cloud Amount, Vegetation Land Cover Transformation	Near Infrared
Band 3	459–479	500	Soil/Vegetation Differences	Visible (Blue)
Band 4	545–565	500	Green Vegetation	Visible (Green)
Band 5	1230–1250	500	Leaf/Canopy Differences	Short Infrared
Band 6	1628–1652	500	Snow/Cloud Differences	Short Infrared
Band 7	2105–2155	500	Cloud Properties, Land Properties	Short Infrared
Band 8	405–420	1000	Chlorophyll	Visible (Blue)
Band 9	438–448	1000	Chlorophyll	Visible (Blue)
Band 10	483–493	1000	Chlorophyll	Visible (Blue)
Band 11	526–536	1000	Chlorophyll	Visible (Green)
Band 12	546–556	1000	Sediments	Visible (Green)
Band 13	662–672	1000	Atmosphere, Sediments	Visible (Red)
Band 14	673–683	1000	Chlorophyll Fluorescence	Visible (Red)
Band 15	743–753	1000	Aerosol Properties	Near Infrared
Band 16	862–877	1000	Aerosol Properties, Atmospheric Properties	Near Infrared
Band 17	890–920	1000	Atmospheric Properties, Cloud Properties	Near Infrared
Band 18	931–941	1000	Atmospheric Properties, Cloud Properties	Near Infrared
Band 19	915–965	1000	Atmospheric Properties, Cloud Properties	Near Infrared
Band 20	3660–3840	1000	Sea Surface Temperature	Mid wave Infrared
Band 21	3929–3989	1000	Forest Fires & Volcanoes	Mid wave Infrared
Band 22	3929–3989	1000	Surface/Cloud Temperature	Mid wave Infrared
Band 23	4020–4080	1000	Surface/Cloud Temperature	Mid wave Infrared
Band 24	4433–4498	1000	Cloud Fraction, Troposphere Temperature	Mid wave Infrared
Band 25	4482–4549	1000	Cloud Fraction, Troposphere Temperature	Mid wave Infrared
Band 26	1360–1390	1000	Cloud Fraction (Thin Cirrus), Troposphere Temperature	Mid wave Infrared
Band 27	6535–6895	1000	Mid Troposphere Humidity	Mid wave Infrared
Band 28	7175–7475	1000	Upper Troposphere Humidity	Long wave Infrared
Band 29	8400–8700	1000	Surface Temperature	Long wave Infrared
Band 30	9580–9880	1000	Total Ozone	Long wave Infrared
Band 31	10780–11280	1000	Cloud Temperature, Forest Fires & Volcanoes, Surface Temperature	Long wave Infrared
Band 32	11770–12270	1000	Cloud Height, Forest Fires & Volcanoes, Surface Temperature	Long wave Infrared
Band 33	13185–13485	1000	Cloud Fraction, Cloud Height	Long wave Infrared
Band 34	13485–13785	1000	Cloud Fraction, Cloud Height	Long wave Infrared
Band 35	13785–14085	1000	Cloud Fraction, Cloud Height	Long wave Infrared
Band 36	14085–14385	1000	Cloud Fraction, Cloud Height	Long wave Infrared

water cycle. The six sensors are: the Atmospheric Infrared Sounder (AIRS), Advanced Microwave Sounding Unit (AMSU-A), Humidity Sounder for Brazil (HSB), Advanced Microwave Scanning Radiometer for EOS (AMSR-E), MODIS, and CERES.

NASA's strategy is to use the MODIS sensors to investigate and acquire hyper-temporal, multi-spectral and multi-angular observations of the Earth on a daily basis. MODIS was launched to continue the monitoring of the Earth from older sensors such as: Coastal Zone Colour Scanner (CZCS), the Advanced Very High Resolution Radiometer (AVHRR), the High Resolution Infrared Spectrometer (HIRS), and the Thematic Mapper (TM). The MODIS sensors were built by the Santa Barbara Remote Sensing Institute according to the specifications provided by NASA. NASA has gone to great lengths to ensure proper sensor calibration to generate an accurate long-term data set for global studies [57].

MODIS is a passive remote sensing instrument with 490 detectors, which are arranged to form 36 spectral bands that measure the 405 nm–14385 nm spectrum. Each detector in the sensor has a 12-bit

TABLE 2.3: Table description of the available MODIS land cover products.

Product	Short Description	Composition time	Spatial Resolution	Satellites	Product Code
Snow product	Snow cover land and snow albedo	Daily/8-day	500m/1km	Terra or Aqua	MOD10/MYD10 MOD29/MYD29
Land surface temperature	Land surface temperature and emissivity daily levels	Daily/8-day/ Monthly	1km/6km	Terra or Aqua	MOD11/MYD11
Land cover dynamic product	Decision tree classify 34 classes of land cover	Yearly	500m/1km	Terra or Aqua	MOD12/MYD12
Thermal Anomalies/ Fire products	Fire detection	Daily/8-day	1km	Terra or Aqua	MOD14/MYD14
LAI/FPAR products	Measure surface photosynthesis, evapotranspiration, and net primary production	8-day	1km	Terra, Aqua or combined	MOD15/MYD15/ MCD15
Gross Primary Production product	Measures growth of terrestrial vegetation	8-day	1km	Terra or Aqua	MOD17/MYD17
Surface Reflectance	Spectral reflectance and atmospheric scattering	Daily/8-day	250m/500m/ 1km	Terra or Aqua	MOD09/MYD09
Global Vegetation Indices	Calculates the NDVI and EVI indices	16-day/Monthly	250m/500m/ 1km	Terra or Aqua	MOD13/MYD13
Vegetation Cover Conversion	Estimate proportions of life form, leaf type, and leaf longevity	Yearly	500m	Terra	MOD44
BRDF/Albedo products	Mathematical models to describe BRDF and derive Albedo measurements	8-day/16-day	500m/1km	Terra, Aqua or combined	MOD43/MYD43/ MCD43
Burned Area product	Burning and quality information and survey for rapid changes on surfaces	Monthly	500m	Combined	MCD45

radiometric resolution and can acquire a swath of 2330 km (cross track) by 10 km (nadir track). The wide swath width of MODIS enables it to record the entire Earth's surface every two days. MODIS spectral bands are recorded at a different spatial resolutions: spectral bands 1–2 are measured at 250 m spatial resolution, spectral bands 3–7 are measured at 500 m spatial resolution and spectral bands 8–36 are measured at 1 km spatial resolution. The spatial resolution is reported at a nadir viewing angle. It should be noted that an increase in spatial resolution is experienced in the scan direction, which causes pixels to be partially overlapping at off-nadir angles. This phenomenon is known as the bowtie effect and is a source of variability over the revisit cycle.

The spectral bands are designed to provide observations of global environmental processes occurring in the troposphere: cloud activity, radiation budget, oceanographic occurrences and land cover monitoring (Full listing in Table 2.2). The images acquired by MODIS are converted with a set of preprocessing steps on a daily basis into terrestrial, atmospheric and oceanic products (Full product listing in Table 2.3).

The prefix MOD and MYD in the product code (table 2.3) refers to the product derived from the data acquired from the Terra and Aqua satellites respectively. The prefix MCD in the product code refers to the product derived using data from both satellites [27, 28, 58–60]. The composition time

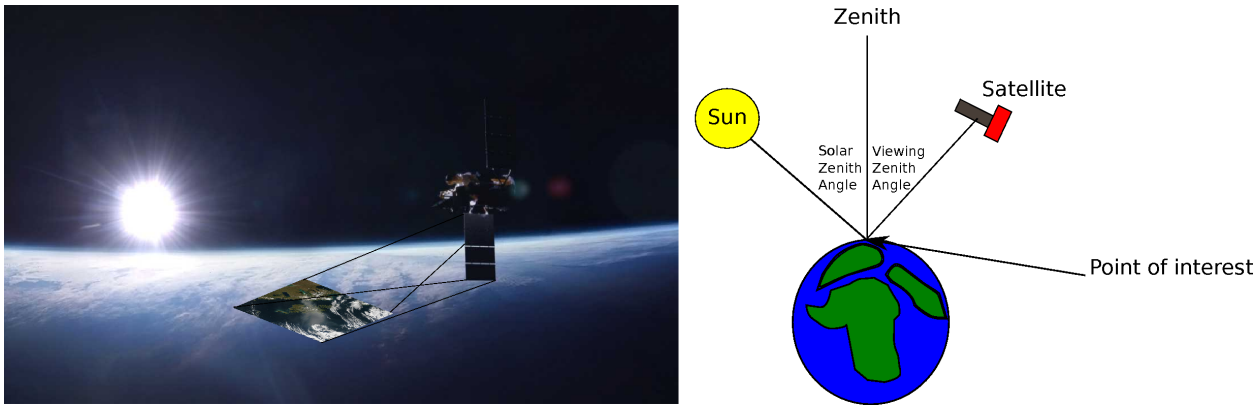


FIGURE 2.6: Example of a passive orbiting satellite acquiring an image from earth.

(table 2.3) reports the temporal resolution at which an acquisition for the product becomes available and the spatial resolution at which the products are produced.

The MODIS product chosen for this thesis is the MCD43A4 land surface reflectance product. The product is defined as a nadir viewed land surface reflectance, which is atmospherically corrected [61, 62]. The adjusted land spectral reflectance product significantly reduces the anisotropic scattering effects of surfaces under different illumination and observation conditions [27, 28]. This BRDF/Albedo product is also used as an input to derive land classifications for the *Land Cover Dynamic Product*. The MCD43A4 product uses the first 7 spectral bands, which are often referred to as the land surface bands. The 7 spectral bands are used because of the minimal atmospheric absorption of atmospheric gases.

The larger swath width on MODIS enables the surveying of every geographical area at least every two days. The MODIS instrument has an orbital repeat cycle of 16 days, which is a problem with the large swath width, as the viewing angles (at the same ground location) between successive observations might differ dramatically. This means that every 16 days an image is acquired of the same geographical area with similar viewing angles.

The disadvantage of acquiring images from a polar orbiting passive satellite is the variation in the reflected signal that is caused by the change in the surface reflectance during the composition period (Figure 2.6). This variation in signal is contributed by many different environmental and external sources such as: solar zenith angle, viewing zenith angle, seasonality, sensor angle, etc.

This disadvantage created the need to consider the distribution of the electromagnetic radiation as a function of the observation and illumination angles. The BRDF is a mathematical function which describes the variability in surface reflection based on the illumination and viewing angles [63].

Estimation of the BRDF enables the adjustment of the reflectance values as if they were taken from a nadir view. The MODIS MCD43A4 product uses a 16-day rolling window of acquisitions from both Terra and Aqua satellites, together with a semi-empirical kernel-driven bidirectional reflectance model

to determine the global set of parameters describing the BRDF. The hemispherical reflectance and the bi-hemispherical reflectance at the solar zenith angle are derived from the BRDF parameters to produce a coarse resolution composite image every 8 or 16 days [28].

A weighted linear sum of kernel functions is used for a BRDF model to correct for illumination and viewing angles. This BRDF model is a 4-variable function that sums together an isotropic parameter and two functions of viewing and illumination geometry to determine the reflectance [28]. The BRDF model is given by

$$\begin{aligned} R(\theta_{\text{sol}}, \theta_{\text{view}}, \theta_{\text{rel}}, \lambda) = f_{\text{iso}}(\lambda) + f_{\text{vol}}(\lambda) K_{\text{vol}}(\theta_{\text{sol}}, \theta_{\text{view}}, \theta_{\text{rel}}, \lambda) \\ + f_{\text{geo}}(\lambda) K_{\text{geo}}(\theta_{\text{sol}}, \theta_{\text{view}}, \theta_{\text{rel}}, \lambda), \end{aligned} \tag{2.5}$$

where θ_{sol} denotes the solar zenith angle and θ_{view} denotes the viewing angle. The variable θ_{rel} denotes the relative azimuth angle and λ denotes the wavelength.

The RossThick kernel function is currently best suited for the volume scattering radiative transfer model used in the kernel function $K_{\text{vol}}(\theta_{\text{sol}}, \theta_{\text{view}}, \theta_{\text{rel}}, \lambda)$ for the MODIS MCD43A4 product. The LiSparce kernel function is at present best suited for the geometric shadow casting theory used in the kernel function $K_{\text{geo}}(\theta_{\text{sol}}, \theta_{\text{view}}, \theta_{\text{rel}}, \lambda)$ [28].

The BRDF model's parameters are derived by the MODIS MOD43B1 product and are used to compute the albedos using the solar illumination geometry. The approximation of terrestrial albedo at a particular solar zenith angle, requires a weighted sum of the black-sky (directional-hemispherical) albedo and the white-sky (bi-hemispherical) albedo. The black-sky albedo is defined as albedo in the absence of a diffuse component and is a function of the solar zenith angle. The white-sky albedo is defined as albedo in the absence of a direct component when the diffuse component is isotropic [28]. The product uses the black-sky and white-sky model for albedo estimation.

The black-sky model is given as

$$\begin{aligned} \alpha_{\text{BS}} = f_{\text{iso}}(\lambda)(g_{0,\text{iso}} + g_{1,\text{iso}}\lambda^2 + g_{2,\text{iso}}\lambda^3) \\ + f_{\text{vol}}(\lambda)(g_{0,\text{vol}} + g_{1,\text{vol}}\lambda^2 + g_{2,\text{vol}}\lambda^3) \\ + f_{\text{geo}}(\lambda)(g_{0,\text{geo}} + g_{1,\text{geo}}\lambda^2 + g_{2,\text{geo}}\lambda^3). \end{aligned} \tag{2.6}$$

The coefficients for the black-sky model for the isotropic (iso), the RossThick (vol) and LiSparce (geo) can be substituted into equation (2.6) to simplify to

$$\alpha_{\mathrm{BS}} = f_{\mathrm{iso}}(\lambda) + f_{\mathrm{vol}}(\lambda)(-0.007574 - 0.070987\lambda^2 + 0.307588\lambda^3) + f_{\mathrm{geo}}(\lambda)(-1.284909 - 0.166314\lambda^2 + 0.04184\lambda^3). \quad (2.7)$$

The white-sky model is given as

$$\alpha_{\mathrm{WS}} = f_{\mathrm{iso}}(\lambda)g_{\mathrm{iso}} + f_{\mathrm{vol}}(\lambda)g_{\mathrm{vol}} + f_{\mathrm{geo}}(\lambda)g_{\mathrm{geo}}. \quad (2.8)$$

The coefficients for the white-sky model are also substituted into equation (2.8), which equates to

$$\alpha_{\mathrm{WS}} = f_{\mathrm{iso}}(\lambda) + 0.189184 f_{\mathrm{vol}}(\lambda) - 1.377622 f_{\mathrm{geo}}(\lambda). \quad (2.9)$$

The solar zenith angle is then transformed to a nadir angle at local sensor noon using the BRDF model.

Cloud obscuration reduces the number of observations that are available for processing even when both satellites are combined within a product. Fortunately, according to a global analysis conducted, South Africa has more than an 80% probability of acquiring enough non-cloudy images within 16 days to produce a reliable 8 day composite land reflectance MODIS product [64].

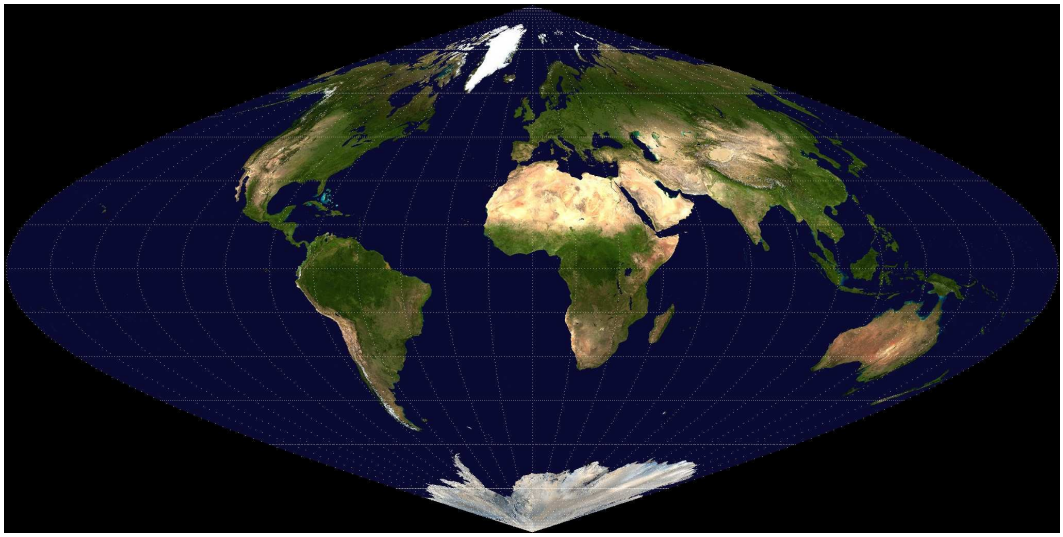


FIGURE 2.7: Sinusoidal projection of the the planet Earth.

The land surface reflectance products are sinusoidally projected and stored in a Hierarchical Data Format - Earth Observing System (HDF-EOS) format [65]. A sinusoidal projection of the planet Earth is shown in figure 2.7. The sinusoidal projection is a pseudocyclindrical projection, which translates images to retain relative geographical sizes between areas accurately. These images are then gridded to form an equal-sized gridded map. The disadvantage is that it distorts the shapes and orientation within the maps when viewing the images.

The PSF of the MODIS sensor was not measured prelaunch; instead a line spread function (LSF) was measured in the scan direction to derive the PSF [66]. The MODIS PSF induced radiation from adjacent areas which is mostly caused by clouds. A correction for this unwanted radiation entering the sensor is computed using both the PSF and the approximation of the radiance measured by the saturated spectral bands. This prior knowledge of the radiance received is usually discarded in most products, as it requires long computing times. The largest impact is the low radiance measured in MODIS oceanic products, which are in close proximity to highly reflective objects such as clouds, coastlines, or sun glint. The PSF introduces a small amount of straylight into the MODIS measurements, which does not have a large impact on land surface products.

2.7 VEGETATION INDICES

Vegetation indices were created to assist in the study of terrestrial vegetation in large-scale global environmental dynamics. Vegetation indices are spectral transformations of a set of spectral band combinations. The vegetation indices enhance the vegetation characteristics within an image, which facilitates the comparison of terrestrial photosynthetic activity variations [67].

2.7.1 Normalised Difference Vegetation Index

The Normalised Difference Vegetation Index (NDVI) is a scalar index that enhances vegetation characteristics in a multi-spectral image. The NDVI was inspired by phenology, which is the study of the periodical growth cycle of plants and how this cycle is influenced by seasonal and inter-annual variability in the ecosystem [68]. A global NDVI coverage map is shown in figure 2.8. NDVI is a normalised ratio that uses the λ_{RED} (Red spectrum band 0.63 μm – 0.69 μm) and λ_{NIR} (Near Infrared spectrum band 0.76 μm – 0.90 μm) spectral bands and is computed as

$$\mathrm{NDVI} = \frac{\lambda_{\mathrm{NIR}} - \lambda_{\mathrm{RED}}}{\lambda_{\mathrm{NIR}} + \lambda_{\mathrm{RED}}}. \tag{2.10}$$

The NDVI index capitalises on the differences in absorption rates between the two spectral bands when interacting with natural vegetation. The RED spectral band's electromagnetic radiation is absorbed by the natural vegetation for photosynthesis and the NIR spectral band's electromagnetic radiation is reflected by the natural vegetation because of the vegetation's cellular structure. The NDVI index exploits the low reflectance values in the RED spectral band and high reflectance values in the NIR spectral band for natural vegetation [69, 70]. The NDVI ratio shown in equation (2.10) produces positive values near 1 ($\mathrm{NDVI} \approx 1$) for areas containing a dense vegetation canopy and small positive values ($\mathrm{NDVI} \approx 0$) for bare soils.

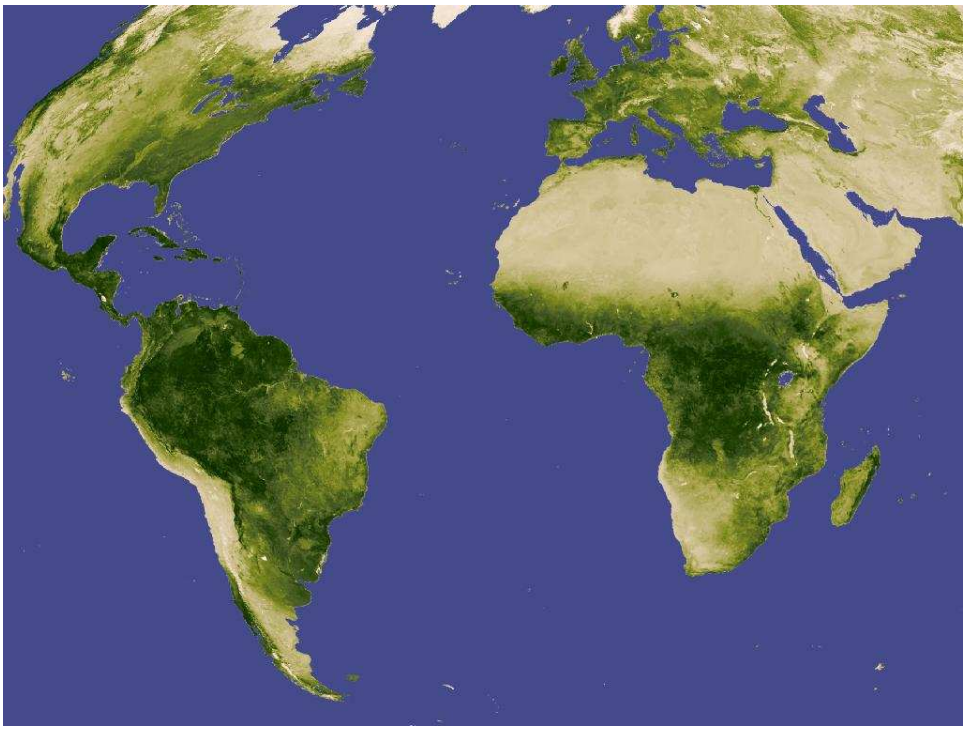


FIGURE 2.8: Global NDVI index coverage map created using MODIS. Image supplied by NASA.

The general use of the NDVI index is demonstrated in large regional environmental models, which include: leaf area index, biomass, chlorophyll, net plant productivity, fractional vegetation cover, accumulated rainfall, etc. Several studies tend to over-use the NDVI index in many applications for which it was not specifically designed [71]. The normalised difference between these two spectral bands only illustrates a relationship in the original information, while other important information is discarded. Whether the discarded information is relevant depends on the process of analysis and geographical area. The NDVI index is sensitive to numerous environmental factors, including atmospheric effects, thin cloud coverage (ubiquitous cirrus), moistness of the soil (precipitation or evaporation), difference in soil colour, anisotropic effects, and spectral effects (different sensors provide different NDVI).

Several alternatives to NDVI have been proposed to address a variety of limitations in analysing satellite acquired imagery. These include: the Perpendicular Vegetation Index [72], the Soil-adjusted Vegetation Index [73], the Atmospherically Resistant Vegetation Index [74], and the Global Environment Monitoring Index [71].

2.7.2 Enhanced Vegetation Index

The Enhanced Vegetation Index (EVI) is an improved version of the NDVI vegetation index. The EVI does not tend to saturate as quickly as the NDVI does in areas with high biomass. The EVI decouples

the canopy background reflectance, and is computed as

$$\mathrm{EVI} = G\,\frac{\lambda_{\mathrm{NIR}} - \lambda_{\mathrm{RED}}}{\lambda_{\mathrm{NIR}} + C_1\lambda_{\mathrm{RED}} - C_2\lambda_{\mathrm{BLUE}} + L}. \tag{2.11}$$

The variable λ_{NIR} denotes the surface reflectance of the near infrared band and λ_{RED} denotes the surface reflectance of the red spectral band. The variable λ_{BLUE} denotes the surface reflectance of the blue spectral band and L denotes the canopy background adjustment term. The coefficients C_1 and C_2 denote the aerosol resistance term and G is the gain coefficient.

The scaling coefficients are used to minimise the effects of aerosols. The blue spectral band is atmospherically sensitive and is used to adjust the red spectral band for aerosol influences. The coefficients used by MODIS to calculate EVI are substituted into equation (2.11) as

$$\mathrm{EVI} = 2.5\,\frac{\lambda_{\mathrm{NIR}} - \lambda_{\mathrm{RED}}}{\lambda_{\mathrm{NIR}} + 6\lambda_{\mathrm{RED}} - 7.5\lambda_{\mathrm{BLUE}} + 1}. \tag{2.12}$$

NDVI is the most widely used vegetation index, which could be attributed to its low computational costs. The use of EVI always raises two questions:

1. Does the sensor measure the blue spectral band independently?

2. Are the scaling coefficients used in computing EVI applicable to the current geographical area?

NDVI is a good vegetation index if properly used and was included in this thesis because of its popularity and to create a base performance for comparison [75, 76]. It should be noted that all methods proposed in this thesis could be adapted to operate with other sets of spectral bands and vegetation indices.

2.8 LAND COVER CHANGE DETECTION METHODS

Change detection can be viewed from a prototype theory mindset [77]. The prototype theory states that the performance of the results generated from a change detection method is based on the user's requirements. This creates a paradigm that there is no single solution for detecting change for all applications [18, 20]. Change detection methods are designed for a specific application and have their own merits and limitations.

An example to demonstrate the user's specific needs is shown in figure 2.9. A change in land cover type from natural vegetation to human settlement is experienced in the red polygon, while only seasonal change in the vegetation has occurred in the blue polygon. Applications and issues of change detection in the remote sensing community are summarised into several categories [24], namely:

1. land cover classification and change detection [78, 79],
2. forest monitoring [80, 81],
3. fire detection [82, 83],
4. urban expansion and change [84, 85],
5. natural environment change [86, 87], and
6. specialised applications [88, 89].

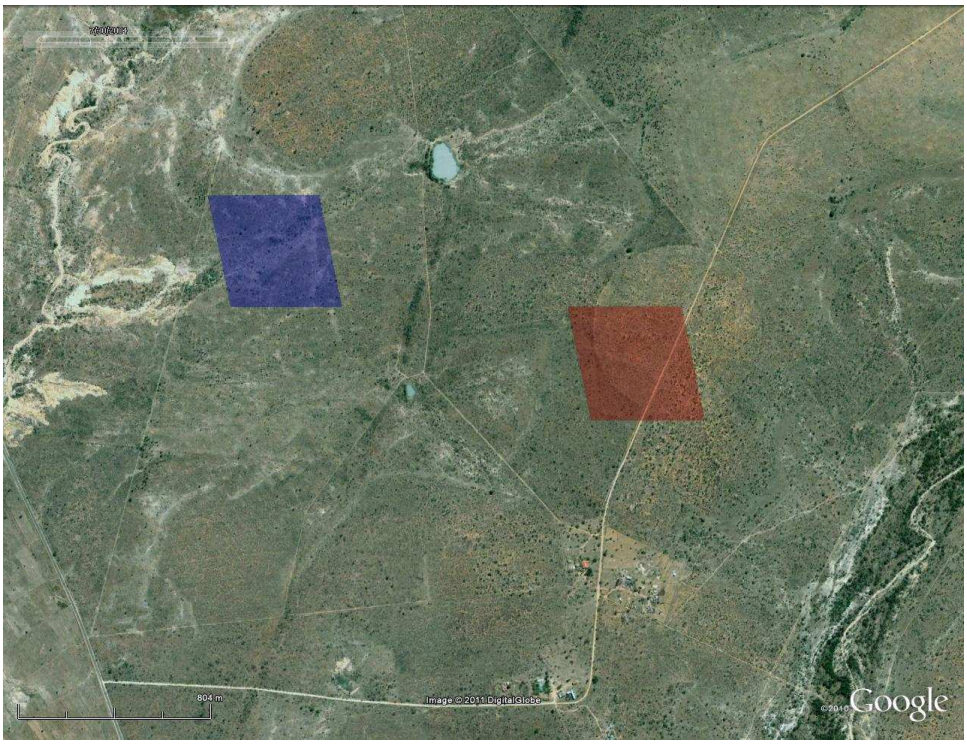
The remote sensing community's monitoring capabilities keep improving with the development and deployment of new technologies. Global data sets are becoming more accessible and computational resources are becoming more affordable [14]. These data sets come from several different sensors. The more popular are: Landsat Multi-Spectral Scanner (MSS), TM, MISR, SPOT, AVHRR and MODIS. The type of land cover change of interest also changes with technologies, which requires continuous pursuit of new change detection methods [18, 20].

There are four major steps involved when constructing a change detection framework [90]. The first step is image preprocessing to ensure the image is corrected by removing any unwanted artifacts [18, 20]. Preprocessing spatially registers and environmentally corrects each image to a minimum product's quality index. The product's quality index is reached by using topographical correction, spatial registration, radiometric calibration, atmospheric calibration and normalisation between multi-temporal imagery.

The purpose of the preprocessing is the assurance that the images acquired over a geographical area remain consistent through time and any changes in the reflectance values are not caused by processing artifacts. Incorrect preprocessing has adverse effects on the accuracy of the change detection methods [91, 92]. For example, if images are not correctly spatially registered, the geographical location of a pixel in one image will not correspond with the geographical location of the same pixel in another image.

The second step is proper feature extraction and selection. Suitable meaningful features must be obtained from the images to give the change detection method the ability to detect change. A renowned quotation is: *If you can measure it, you can improve it - William Thomson*. If no measurable feature exists to detect the change, then no change detection method will be able to detect it.

The third step is to develop a suitable change detection method that uses the features to detect changes according to the user's requirements. The method must be reliable and robust in most operating environments.



(a) Quickbird image taken on 30 July 2004 (courtesy of Google^{TM}Earth).

(b) Quickbird image taken on 31 December 2008 (courtesy of Google^{TM}Earth).

FIGURE 2.9: A change in land cover type is shown by the red polygon in (a) and (b), while only a seasonal change has occurred in the blue polygon.

The fourth step is the assessment of the previous three steps. How well did the change detection method satisfy the requirements set by the user? The overall accuracy assessed in the system is affected

by several factors, including [24]; (1) the quality of the preprocessing, (2) availability of reliable ground truth, (3) complexity of the environmental case study, (4) useful feature extraction, (5) feature analysis and processing, (6) change detection algorithms used, (7) the analyst's skills, (8) knowledge and information about the study area, (9) critical assessment of the system's outputs, and (10) time and cost constraints.

Standard statistical tests are used to measure the performance of the change detection algorithm quantitatively and are supported by visual assessment of the geographical areas. Change detection methods are divided into multi-temporal and hyper-temporal change detection methods. Change detection methods operating on multi-temporal images require only a few images; usually in the order of 2–5 images of the same geographical area. Change detection methods operating on hyper-temporal images usually requires hundreds of images taken at regular constant intervals; usually 8–30 days between acquisitions.

Most change detection methods found in the literature can either provide change information or a change alarm [93, 94]. A change alarm uses a threshold to provide binary *change*/*no change* information from the images. A change information algorithm uses post-classification to provide a *from-to* change.

Multi-temporal change detection methods evaluate local patterns in the reflectance values between images to detect change. The change detection method should compensate for the difference in environmental conditions, illumination conditions, and local trends in each of the images [95]. Multi-temporal change detection methods are grouped into several categories [24]: (1) algebra, (2) transformation, (3) classification, (4) advanced models, (5) Geographical Information System approach (GIS), (6) visual analysis, and (7) other methods.

The algebraic approach entails methods such as [24]: image differencing, image regression, image ratioing, index differencing, trajectory vector analysis, and background subtraction [93, 94]. These methods have low complexity and use manually adjusted thresholds to define change in the local vicinity.

The advantage of using an algebraic approach is the ease of interpreting the execution of the method. Another advantage is that it can operate on data sets which were captured in different environmental conditions. The disadvantage of these methods is that they have the potential to enhance the system noise, which effectively reduces the methods' performances. Another disadvantage is the setting of the threshold. The threshold has to be manually adjusted for each new data set. The methods are sensitive to features with little separability or features that are subjected to external events or time dependence.

The transformation approach uses methods to reduce the number of dimensions in the remote

sensing reflectance data set to create a new manifold [24]. The advantage of this approach is the removal of redundant dimensions and it puts emphasis on the information-carrying components [96, 97]. This approach includes transformation algorithms such as principal component analysis (PCA), Gramm-Schmidt, Chi-square, independent component analysis, etc. The disadvantage is the interpretation of the new manifold and the change trajectory of the geographical area.

The classification approach is characterised by classification methods such as: spectral combined analysis, expectation-maximisation (EM) algorithm, hybrid classification, hierarchical classification, and artificial neural networks (ANN). These methods require initial training on a set of labelled pixels. Afterwards the method is applied using the information gathered to classify a set of unknown labelled pixels. The advantage of using such a classification method is that it provides a change information matrix. These methods are robust to external environmental conditions [8, 98]. The disadvantage is the dependency on periodic updating of the training data sets.

The advanced model approach transforms the spectral reflectance values from multi-temporal spectral reflectance values to physical process parameters. The advantage is that the extracted process parameters are easier to interpret than the spectral reflectance values [99, 100]. Methods commonly used in this category are: Linear Spectral Mixture Analysis (LSMA), Li-Strahler reflectance model, spectral mixture models, and biophysical parameter estimation [24]. The disadvantage is finding a suitable model for the conversion and the intensive procedure of converting the reflectance values.

The GIS-based approach uses a GIS system to analyse satellite imagery. The advantage of a GIS system is the ability to incorporate several different layers of meta-data and satellite images for analysis [101]. The disadvantage is that different data sets have different product quality standards and when used together will degrade the results of the overall performance [24].

Visual interpretation of images can exploit the full capabilities of a remote sensing analyst's experience and knowledge. A skilled analyst can compensate for environmental conditions when looking for change [102]. The disadvantage of this approach is the processing time, and labour cost required for large geographical areas and the variability of skill level of the analyst.

There are many different change detection methods that cannot be grouped into the afore-mentioned categories. These methods produce new approaches to the field of change detection and have their associated advantages and disadvantages [103–105].

Land cover change is a function of time and can be abrupt or gradual. The ability to detect the difference between abrupt and gradual change is based on the temporal acquisition rate, the change detection method and the number of acquisitions.

Gradual change is defined as the slow change from one type of land cover to another. For example, settlement expansion is the process of clearing the indigenous vegetation and constructing a new human

settlement, which could take several months. Abrupt change is defined as a fast change in land cover type, for example, wild fire that can destroy all the natural vegetation in an area within a few hours [106].

Multi-temporal change detection methods flag all their land cover changes as abrupt. Previous studies have shown that multi-temporal change detection methods' performance is limited by the differences produced in the seasonal growth of vegetated areas [107]. Variations in surface reflectance values are observed in vegetated areas when the images are acquired at different times of the intra-annual growth cycle [19]. These phenological cycles induce variations that could raise the false change detection rate, as they are flagged as land cover change when it is only a natural seasonal variation. To overcome this limitation, a high temporal acquisition rate is required to capture the seasonal variations of a particular land cover [108]. This motivates the investigation into hyper-temporal change detection methods, as these methods can distinguish between phenological cycles, gradual and abrupt change [106].

Hyper-temporal change detection methods are used on multiple images acquired from a satellite with a short periodic revisit cycle and can be used to complement a multi-temporal change detection method [109]. The hyper-temporal acquisition rate provides continuous monitoring of the Earth, and is not limited by the availability of costly high-resolution images. This is used to augment information about which areas should rather be tasked for acquisition of high spatial resolution imagery. For example, a hyper-temporal change detection method maps the geographical areas with the highest probability of land cover change at low costs, after which a costly high-resolution image is acquired to confirm the change.

2.8.1 Hyper-temporal change detection methods

Majority of the change detection methods found in the literature are based on medium to high spatial resolution multi-temporal image analysis [18, 20]. Certain multi-temporal change detection methods can be extended to hyper-temporal images by applying the methods sequentially to subsets of multi-temporal images. The approaches that have been extended for the hyper-temporal case are: image differencing [110], image regression [111], image ratioing [112], index differencing [113], Principle Component Analysis (PCA) [75, 76], and Change Vector Analysis (CVA) [114].

These multi-temporal change detection studies rely on bi-temporal and trajectory analysis [20, 21, 24] and the data are mostly treated as hyper-dimensional, but not necessarily as hyper-temporal. These methods therefore do not fully capitalise on the temporal dimension, which captures the dynamics of different land cover types.

Hyper-temporal change detection methods attempt to understand the underlying force structuring

the data in the time dimension by identifying patterns and trends, detecting changes, clustering, modelling and forecasting [8, 40]. Hyper-temporal change detection methods are broadly divided into three categories: regression analysis, spectrum analysis, and temporal metrics.

2.8.1.1 Regression analysis

Regression analysis is a parametric method used to model the underlying structure of the data. The parameters of the model are estimated using the data set. For example, Kleynhans *et al.* assumed the MODIS NDVI time series could be modelled as a triply modulated cosine function [30]. The parameters for this model were estimated using an EKF. A labelled data set was used to estimate the models' covariance matrices manually to improve separability between different land cover classes. The estimated parameters were evaluated to detect changes in land cover.

Regression is also used to fit time series to a hypothetical temporal trajectory [109]. A temporal trajectory is a defined map of a finite sequence of points describing the expected observed values in a time series. The goodness of fit of a particular time series is computed for a set of hypothetical temporal trajectories and is measured using least squares. A set of hypothetical temporal trajectories is derived for forest disturbance dynamics in [109], which is used to describe the type of change.

The advantage of these methods is that there is no need to set a threshold. The disadvantage of both these methods is the assumption in the form of the model or temporal trajectories. Are all the changes that could realistically occur encapsulated in the model? Is the model able to adapt by inserting more parameters or creating a larger set of hypothetical temporal trajectories?

2.8.1.2 Spectrum analysis

Spectrum analysis is the analysis of harmonic frequencies within a time series. Fourier analysis is a type of spectral analysis which uses a Fourier transform to express a time series as a sum of a series of cosine and sine waves with varying frequencies, amplitudes and phases [115, Ch. 3]. The frequency of each wave component is related to the number of completed cycles defined in the time series. In many applications, the Fourier transform of time series is used for classification and segmentation [116]. Lhermitte *et al.* proposed a classification method that only evaluates the mean and seasonal Fourier transform components. The reason for this is due to the high sampling rate of a strong seasonal component in vegetation time series [116]. These components are then clustered using a post-classification change detection method [40].

Verbesselt *et al.* proposed the BFAST (Break For Additive Seasonal and Trend) approach, which uses trend, seasonal and remainder components to detect changes in the phenological cycles of plants [106]. The seasonal component is derived using the Fourier transform and has been shown to be more

stable than a piecewise linear seasonal model [117].

The advantage of these methods is that they are not dependent on a predefined model. They extract the harmonic frequencies from the time series, which means they allow the evaluation of all frequency components. The disadvantage of these methods is that the time series is assumed to be stationary and that enough harmonic frequencies are properly sampled within the time series.

2.8.1.3 Temporal metric

A temporal metric is derived from the time series by evaluating inter-annual differences in five temporal units: annual maximum, annual minimum, annual range, annual mean and temporal vector. Spatial information can also be included in some of these temporal metrics, such as: spatial mean and spatial standard deviation. The temporal metric is compared to a predefined threshold to determine whether change has occurred.

An example of a temporal metric is the evaluation of a moving average window's standard deviation on a time series. A time series is declared as a changed area when two different windows' standard deviation significantly differ from one another [118].

Another temporal metric is known as the disturbance index. The disturbance index is used to detect large-scale ecosystem distance [119]. The disturbance index measures the ratio between annual maximum land surface temperature and annual maximum EVI to the multiple year mean annual maximum land surface temperature and multiple year mean annual maximum EVI. If the current annual maximums are significantly higher than the long-term maximum, a disturbance is flagged. The difference between the two is evaluated with a predefined threshold to categorise the level of disturbance.

The annual NDVI differencing method is another temporal metric proposed by Lunetta *et al.* [19], which calculates the difference between consecutive summation of the annual NDVI time series. The pixel is flagged as change if a certain predefined threshold is exceeded in this difference. The threshold is usually determined using standard normal distribution statistics.

The EKF change detection method is a temporal metric proposed by Kleynhans *et al.* [120], which evaluates the Euclidean distance between parameters derived with an EKF within a spatio-temporal window. The EKF fits a triply modulated cosine function to a time series to model the seasonal variations. The pixel is flagged as change if the Euclidean distance exceeds a predefined threshold.

The autocorrelation function (ACF) change detection method is a temporal metric proposed by Kleynhans *et al.* [121], which evaluates the stationarity of a time series. The ACF of a time series in question is compared to the ACF of time series that did not change in the local geographical vicinity. The pixel is flagged as change if the deviation between the two ACFs exceeds a predefined threshold.

The advantage of using a temporal metric is that it operates on the raw time series data. This enables observation of abnormal behaviour that is usually filtered out by regression and spectrum analysis. The disadvantage of using a temporal metric is the selection of the threshold and the negative impact of the additive noise in the time series has on the performance.

The noise is reduced by creating methods that operate on annual statistics, which reduces the effective time series measurements significantly. For example, an original MODIS NDVI time series for 10 years (+450 time samples) can be reduced down to only 10 annual measurements represented by a temporal metric.

2.8.2 MODIS land cover change detection product

Since the launch of MODIS, several different products have been developed (see table 2.3 on page 22 for a listing). Only a few specific change detection products have been developed for a small range of applications. Thus there is currently no operational MODIS product to detect any changes in land cover. There have been two previous attempts to create an operational land cover change detection product [122–124].

The first attempt was the MODIS land use and land cover (LULC) algorithm, which detects land cover changes at a 1 km resolution using a CVA approach [114, 124]. The direction of the change vector is compared to a predefined threshold value and when exceeded, a change is flagged. It was suggested that neural network classifiers be used on a pixel-by-pixel basis to track the probability that a specific pixel changed over time [124]. The neural network is a supervised classifier and is used to derive a parameter for land cover classification. This parameter is used to determine if the new data of a geographical area are mapped to an existing category or to create a new category for the area. The monitoring of current and previous observations are used with the land cover parameter to declare change.

The second attempt at a MODIS LULC product was the MODIS Vegetative Cover Conversion (VCC) product. The VCC product uses the first two spectral bands of MODIS at a spatial resolution of 250 m to detect any changes caused by anthropogenic activities or extreme natural events [123]. Five different change detection methods were proposed in the VCC product:

1. RED-NIR space partitioning method: A two-dimensional map is created of the brightness and greenness at two separate time intervals and is used to detect change. The brightness is computed as the mean between the first two spectral bands. The greenness is computed as the difference between spectral bands 2 and 1.

2. RED-NIR space change vector: A change vector is mapped onto a spectral space (spectral band

1 and 2) between two different dates for the same pixel. The magnitude and trajectory of the change vector between the two dates are used to determine if changed occurred.

3. Modified Δ-space threshold: Uses a polar notation to define the differences in the RED and NIR values for a pixel at two different dates. The type of change is defined by the resulting vector in the polar plane.

4. Texture thresholding: Measures a coefficient of variation within a 3x3 spatial kernel at two different times. The coefficient of variation is calculated as the ratio between the standard deviation and mean within the kernel. Change is declared when the coefficient of variation exceeds a predefined threshold.

5. Linear feature thresholding: The method computes the mean and absolute difference of a pixel value for each neighbouring pixel in a 3x3 spatial kernel. A threshold determines whether a linear feature is present.

Neither the MODIS LULC [114] nor the MODIS VCC [123] product fully capitalises on the temporal dimension, as only two dates are compared. A multi-temporal change detection method was attempted, while disregarding the potential of a hyper-temporal change detection method, which has been used successfully in other fields [125, 126]: telecommunications, voice recognition, control systems, etc. Even though one of the primary objectives before the launch of the MODIS sensors was an operational land cover change detection product, to date no operational product has been developed.

2.9 SUMMARY

In this chapter, the use of remote sensing for monitoring geographical areas was discussed. The joint investment of many international organisations and national governments has led to the creation of numerous Earth observation satellites for various different applications. The chapter focused on the importance of using satellite remote sensing to detect new human settlement development in certain regions of South Africa.

The method of choosing a satellite-based sensor was discussed by considering the spatial, spectral, radiometric, and temporal resolutions. After considering multiple factors, the MODIS sensor was chosen, followed by a detailed description of its properties, with emphasis on the benefits of the BRDF corrected data products. The chapter concluded with a review of some of the popular multi-temporal change detection methods, and expanded to the use case of hyper-temporal change detection methods.

CHAPTER THREE

SUPERVISED CLASSIFICATION

3.1 OVERVIEW

Using machine learning methods to classify data sets is a recognised solution in many remote sensing applications. In this chapter several design considerations are introduced that should be heeded when implementing a supervised classifier. This is important, since less than 30% of new designs are correctly assessed [127]. In the previous chapter it was found that machine learning methods are more readily used in modern research because of the large volumes of data sets becoming readily available to the research community, and the great benefit of analysing these data sets in higher dimensional feature space. This chapter focuses on discussing strong, feasible approaches when a supervised classifier is used to solve real world problems.

3.2 CLASSIFICATION

Classification is the process of finding important similarities between objects and then grouping these objects into several subjective classes (categories).

Conceptual clustering is a modern process of classification by which conceptual descriptions are derived from objects, which is followed by the classification of the object according to these descriptions. Conceptual clustering was promoted from a machine learning background. There are two general methods of categorisation that apply to conceptual clustering, namely supervised and unsupervised learning [98, 128]. Supervised learning is the process of supplying category labels to objects in the machine learning algorithm, while an unsupervised learning algorithm attempts to extract the categories without any labels. The way in which the two learning methods operate are completely different. A supervised learning method uses the labels of multiple objects to extract the information from the descriptions that will accurately predict the correct category. An unsupervised

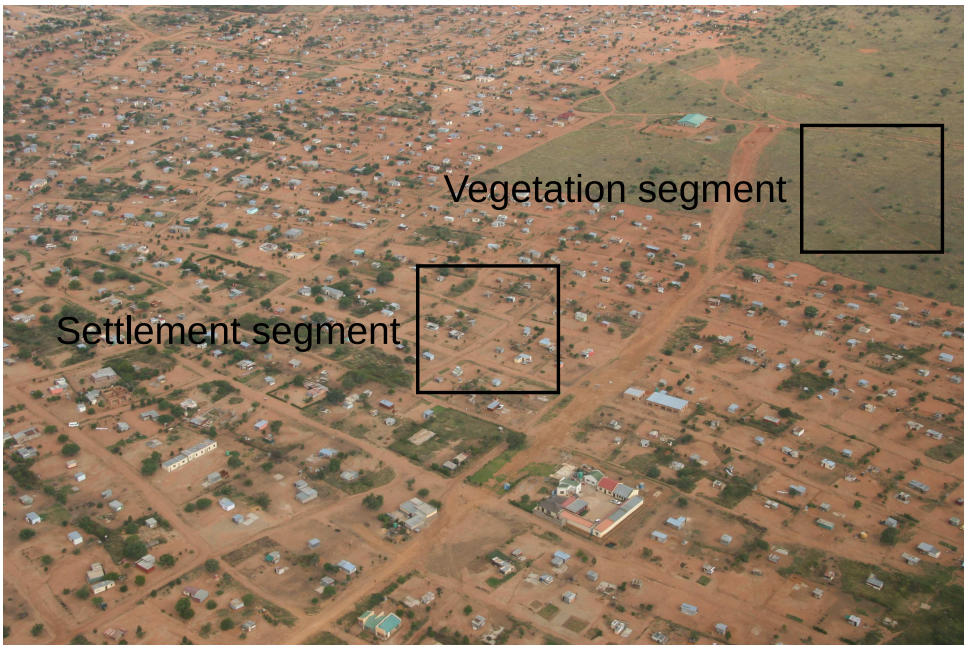


FIGURE 3.1: An aerial photo taken in the Limpopo province, South Africa of two different land cover which are indicated by a natural vegetation segment and settlement segment. A segment is defined as a collection of pixels within a predefined size bounding box.

learning method examines the inherent structure between all objects, to create categories using the most similar descriptions.

3.3 SUPERVISED CLASSIFICATION

Supervised classification is a form of conceptual clustering and is the process of allocating a predefined class label to a certain input pattern. Several concepts will be introduced throughout this thesis in considering a hypothetical problem of separating different land cover types in an image. In figure 3.1, an aerial photo is used to illustrate two different land cover types: natural vegetation and human settlement. Input patterns to the supervised classifier will be labelled as either natural vegetation or human settlement. The supervised classifier is given a set of descriptors to infer a function that assigns a predefined label to each segment of the image. This function produces output values, denoted by y, as either discrete, continuous or probabilistic in nature. The supervised classifier assigns a class label to the output value y that best matches the given input pattern and is denoted by $\mathcal{C}_k, k = 1, 2, \ldots K$, where K is equal to the number of output classes.

Land cover example: In the case of the land cover example shown in figure 3.1, K is equal to two and the output value that the supervised classifier produces will be assigned accordingly to either the natural vegetation class or the human settlement class. □

Observations from different data sources are often grouped together to form an input vector $\vec{x}$, also referred to as an input pattern. These input data sources are usually in descriptive forms that can be interpreted by humans.

Land cover example: In the case of the land cover example, the input data sources provide a colour metric that is either ordinal or real. The input data source in this instance is a set of real number values derived from the green, blue and red colours extracted from the RGB (Red Green Blue) colour buffer of all the pixels within a segment. This input data source is used to form a single input vector with three dimensions, which is defined as

$$\vec{x} = [(\text{Red value})\ (\text{Green value})\ (\text{Blue value})], \tag{3.1}$$

where $\vec{x}$ denotes the input vector. □

3.3.1 Mapping of input vectors

The ability of the supervised classifier to map the input vector $\vec{x}$ to the desired output value y is based on the performance of the learning algorithm. Given a set of input vectors $\{\vec{x}\}$ and the set of corresponding desired output values $\{y\}$, the learning algorithm seeks to infer a function that will satisfy

$$y \approx \mathcal{F}(\vec{x}). \tag{3.2}$$

This implies that the input space is approximately mapped to the desired output space by using a mapping function denoted by $\mathcal{F}$. The mapping function $\mathcal{F}$ is optimised by introducing a scoring function that evaluates the current mapping function's performance.

The learning algorithm tries to find a solution to the mapping function that will maximise the scoring function. There are two general approaches to solving equation (3.2) when a scoring function is used: empirical risk minimisation and structural risk minimisation. Empirical risk minimisation attempts to find the optimal inferred function that will minimise the error in the mapping of the input space to the output space. Structural risk minimisation includes a penalty term that provides control between the bias and variance trade-off within the learning algorithm [129]. Both approaches try to minimise the mapping error between the input and output space.

In regression analysis, the learning algorithm attempts to model the conditional distribution of the desired output values, given a set of input vectors. The desired output values will also be termed target values. Mapping typically uses an error function to determine the goodness of fit between the input and output space, and is based on the principle of maximum likelihood [130, Ch. 6 p. 195]. The likelihood $\mathcal{L}$ is computed as

$$\mathcal{L} = \prod_{p=1}^{P} p(T_{\mathcal{C}}^{p} \mid \vec{\tilde{x}}^{\,p}) P(\vec{\tilde{x}}^{\,p}), \tag{3.3}$$

where $P(\vec{\tilde{x}}^{\,p})$ denotes the probability of observing the p^{th} input vector and $p(T_{\mathcal{C}}^{p} \mid \vec{\tilde{x}}^{\,p})$ denotes the conditional probability density of observing the target value $T_{\mathcal{C}}^{p}$, given that the input vector $\vec{\tilde{x}}^{\,p}$ is present. The error function $\mathcal{E}$ is derived by converting equation (3.3) into the negative log-likelihood, which is defined as

$$\mathcal{E} = -\ln \mathcal{L} = -\sum_{p=1}^{P} p(T_{\mathcal{C}}^{p} \mid \vec{\tilde{x}}^{\,p}) - \sum_{p=1}^{P} P(\vec{\tilde{x}}^{\,p}). \tag{3.4}$$

The minimisation of the error in the mapping requires the minimisation of error function $\mathcal{E}$. The minimisation of the error function $\mathcal{E}$ in equation (3.4) will result in the maximisation of the likelihood in equation (3.3). A popular method of defining the error in mapping is the Sum of Squares Error (SSE). The minimisation of the SSE is equivalent to minimising the error function $\mathcal{E}$ in equation (3.4). The SSE equation over P patterns is given as

$$\mathcal{E} = 0.5 \sum_{p=1}^{P} \left\| \mathcal{F}(\vec{\tilde{x}}^{\,p}) - T_{\mathcal{C}}^{p} \right\|^{2}. \tag{3.5}$$

The vector $\vec{\tilde{x}}^{\,p}$ denotes the p^{th} input vector and $T_{\mathcal{C}}^{p}$ denotes the corresponding target value of the supervised classifier.

In regression analysis, the mapping derived by using equation (3.5) is regarded as optimal as long as the following three conditions are met [130, Ch. 6 p. 203]. These three conditions are:

1. The input vector set $\{\vec{\tilde{x}}\}$ is sufficiently large to capture the underlying data structure.
2. The mapping between the input space and the output space is flexible enough.
3. The optimisation of the mapping is done with a good learning algorithm to minimise equation (3.5) effectively.

In classification analysis, the learning algorithm tries to model the posterior probability of the class label. The SSE function was not specifically designed for classification problems, as it assumes that the target values are generated from a smooth deterministic function with additive zero-mean Gaussian distributed noise. The decision to use error functions within classification requires discrete class labels with optional corresponding class membership probabilities [130, Ch. 6 p. 222]. Many different approaches have been used to rescale the output values in regression problems to match the

class membership probabilities [130, Ch. 6 p. 223]. The error function shown in equation (3.4) is reformulated for a classification problem as

$$\mathcal{E} = -\sum_{p=1}^{P} p(T_{\mathcal{C}}^{p} \mid \vec{\tilde{x}}^{p}) - \sum_{p=1}^{P} P(\vec{\tilde{x}}^{p}) = -\sum_{p=1}^{P}\sum_{k=1}^{K} p(\mathcal{C}_k \mid \vec{\tilde{x}}^{p})\delta_{T_{\mathcal{C}}^{p}} - \sum_{p=1}^{P} P(\vec{\tilde{x}}^{p}). \tag{3.6}$$

If the p^{th} input vector $\vec{\tilde{x}}^{p}$ is from class $\mathcal{C}_k$ then $\delta_{T_{\mathcal{C}}^{p}} = 1$, where δ denotes the Kronecker delta symbol. The symbol k denotes the class label of interest and K denotes the number of output classes.

The output values of the supervised classifier correspond to the Bayesian posterior probabilities if the SSE function is minimised as shown in equation (3.6) [131, 132]. In a regression application it is acceptable to assume Gaussian residuals when using the SSE function, but for classification problems the target values are discrete and the additive zero-mean Gaussian distributed noise is not a good description. A more intuitive approach is to use a binomial distribution which leads to the definition of the cross-entropy error function [133].

Cross-entropy starts by observing the probability that the set of target values is $T_{\mathcal{C}_k}^{p} = \delta_{T_{\mathcal{C}}^{p}}$ when the p^{th} input pattern $\vec{\tilde{x}}^{p}$ is from class $\mathcal{C}_k$. This results in the output of a supervised classifier denoting a class membership probability $p(\mathcal{C}_k|\vec{\tilde{x}}^{p})$ [130, Ch. 6 p. 237]. The value of the conditional distribution is then expressed as

$$\mathcal{L} = \prod_{p=1}^{P} p(T_{\mathcal{C}}^{p} \mid \vec{\tilde{x}}^{p}) P(\vec{\tilde{x}}^{p}) = \prod_{p=1}^{P} \left(\prod_{k=1}^{K} (y^{p})^{T_{\mathcal{C}_k}^{p}} \right) P(\vec{\tilde{x}}^{p}), \tag{3.7}$$

which equates to the cross-entropy error function defined as

$$\mathcal{E} = -\sum_{p=1}^{P}\sum_{k=1}^{K} T_{\mathcal{C}_k}^{p} \ln\left(\frac{y^{p}}{T_{\mathcal{C}_k}^{p}}\right). \tag{3.8}$$

To ensure that the output values of the supervised classifier equates to the posterior probabilities, the following condition must hold, given as [130, 134]

$$\frac{l'(1-y)}{l'(y)} = \frac{1-y}{y}, \tag{3.9}$$

where a class of functions l which satisfies this condition is given by

$$l(y) = \int y^{r}(1-y)^{r-1}\mathrm{d}y. \tag{3.10}$$

Both the cross-entropy error function and SSE function comply with the condition set in equation (3.9). Either of these two error functions can be used in minimising the error in the mapping between the input space and output space for a given classification application. The SSE function is more

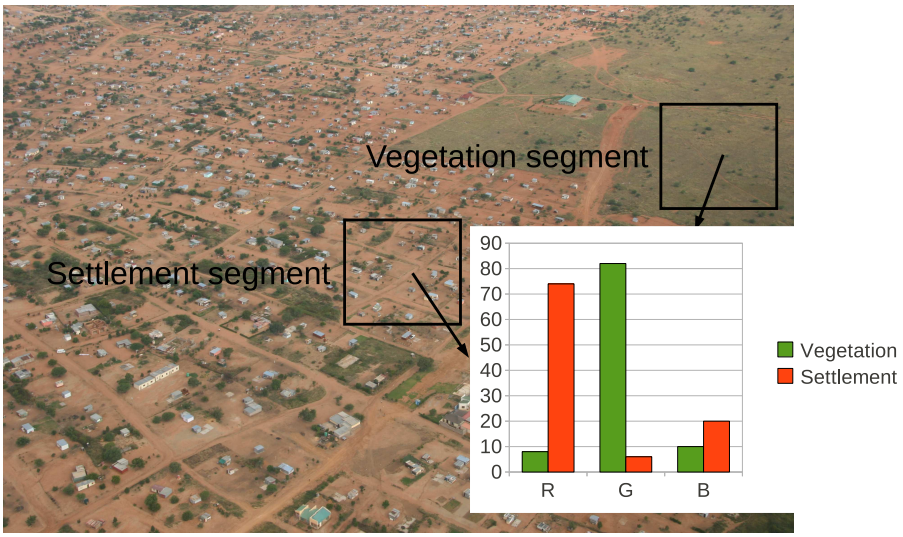


FIGURE 3.2: The same aerial photo over the Limpopo province as shown in figure 3.1, with an RGB histogram overlay showing the attributes of the two segments.

attractive owing to the ease of implementation.

Land cover example: In the case of the land cover example, a mapping of the input space to the output space is planned. The output space has two categories and the class labels are defined as; $\mathcal{C}_k \in \{\mathcal{C}_1, \mathcal{C}_2\} = \{\text{natural vegetation, human settlement}\}$. The input vectors are grouped as shown in equation (3.1). The learning algorithm infers a function that will map the input vector to the corresponding output value. These output values are grouped according to their respective class label for analysis of the supervised classifier. The learning algorithm will attempt to map the correct intensities of the RGB buffer values that will prove to be the most probable match between the input vector and the correct class membership. The learning algorithm uses a scoring system, like the SSE, to minimise the number of incorrect class memberships that are present in the current mapping. To demonstrate the results of the mapping, a histogram of each segment is shown in figure 3.2 with all participating pixels. The supervised classifier assigns segments with dominant red intensity to human settlement and segments with dominant green intensity to natural vegetation. □

The external evaluation of the mapping of the input space to the output space requires sound empirical validation. It was shown that less than 30% of new classifiers and learning algorithms are correctly assessed with proper empirical validation [127]. To ensure proper analysis, the results can be assessed by running the supervised classifier on actual (non-synthetic) data sets. This approach will ensure strong support in using the supervised classifier to solve real problems. A second approach to proper external evaluation is the subdivision of the data set into several partitions. These partitioned

data sets allow proper tuning of the supervised classifier and are used to perform cross-validation [127]. A good method of tuning a supervised classifier is to subdivide the labelled data set (input vectors with known class labels) into three different subsets:

1. A training data set, which is used to train the learning algorithm to derive a mapping function that will minimise the errors on the entire set of input vectors $\{\vec{\tilde{x}}\}$.

2. A validation data set, which is used to test the performance periodically and to mitigate any negative design effects of the supervised classifier [135]. The performance is bounded by the intrinsic noise within the training data [130, Ch. 9 p. 372].

3. A test data set, which is used to verify the performance of the supervised classifier on unseen data. The test data set is used to approximate the generalisation error; this data set is not included in the training phase or optimisation phase of the classifier.

3.3.2 Converting to feature vectors

Preprocessing of the input vector $\vec{\tilde{x}}$ before the learning algorithm and postprocessing of the output vector $\vec{y}$ after the learning algorithm is an optional procedure used to improve an algorithm's performance. The performance improves even when evaluating the outputs derived from the learning algorithm that is using a noisy and inconsistent data set [136]. Let $\vec{x}$ denote the preprocessed version of the input vector $\vec{\tilde{x}}$, and $\vec{\hat{y}}$ denote the postprocessed version of the output vector $\vec{y}$. This processing chain is illustrated in figure 3.3.

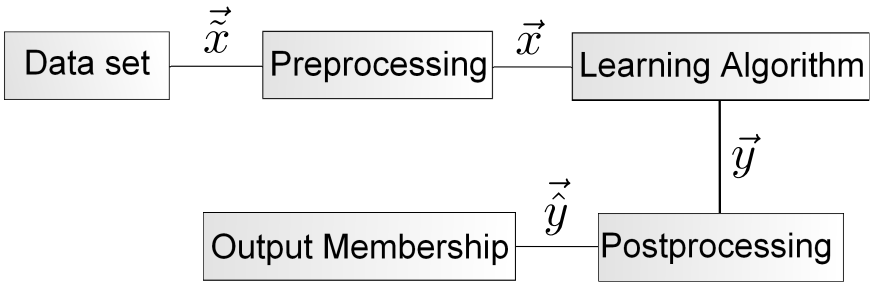


FIGURE 3.3: Flow diagram illustrating the processing steps that includes preprocessing and postprocessing.

The input data set $\{\vec{\tilde{x}}\}$ contains information from several input data sources and the information from each individual source can either be real numbers, ordinal numbers, nominal numbers or an 1-of-c coding. An adjective used to describe the numerical ranking of an object's position in a set is known as an ordinal number. A nominal number is a set of numbers used for labelling purposes alone and do not provide an indication of any other type of measurement. A 1-of-c coding is a vector representation

of the input which is an all-zero vector except in one location. The input data sets must have the same cardinality regardless of the form of the input source.

Preprocessing is the processing of raw data supplied from the input data set $\{\vec{x}\}$ to another space that can be more effectively analysed. Most machine learning algorithms learn faster and provide better performance if the input data set $\{\vec{x}\}$ is preprocessed. Numerous different methods are used for preprocessing, including: sampling, transformation, denoising, standardisation and feature extraction.

1. Sampling selects representative subsets from a large population of input patterns to perform a range of functions such as generalisation, cross-validation, etc.

2. Transformation translates the raw data set to another mathematical domain.

3. Denoising includes several techniques used to reduce the noise on samples in the input data set.

4. Standardisation refers to the scaling of the variables within the input pattern from multiple input data sources to a common scale. This common scale allows the underlying properties of the input data sources to be compared fairly within a machine learning algorithm.

5. Feature extraction extracts specific characteristics from the input patterns.

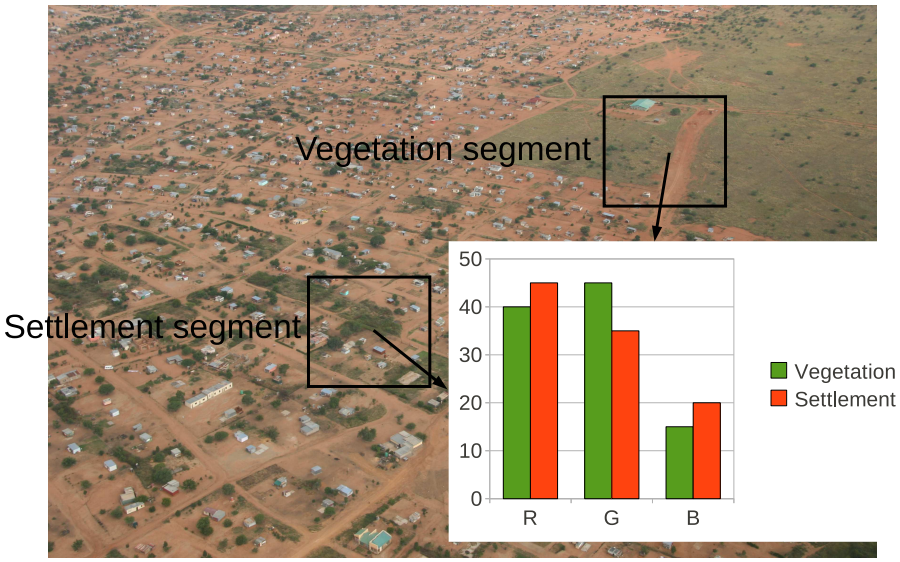


FIGURE 3.4: An alternative selection of natural vegetation and human settlement segments of the aerial photo taken in the Limpopo province using the same input vector.

Land cover example: Revisiting the aerial photo, the advantage of feature extraction as a preprocessing step can be shown when new segments are selected as shown in figure 3.4. High correlation is observed in the histogram of the three RGB buffer values when the new

segments are captured with the original input vector defined in equation (3.1). This results in poor separability within the input space and significant deterioration in the performance of the machine learning algorithm. Both segments appear highly similar in figure 3.4, and will require a complex classifier to separate the segment into the two predefined classes.

A feature extraction method is proposed in the example to extract both the moisture and reflectivity of each segment. Once extracted, these features can be placed into a feature vector $\vec{x}$ of two dimensions, which is defined as

$$\vec{x} = [(\text{Moisture})\ (\text{Reflectivity})]. \tag{3.11}$$

By using the feature vector, the human settlement segment in the example has high reflectivity and low moisture retention due to the bare soil. The natural vegetation segment has high moisture retention and low reflectivity, as shown in figure 3.5. This creates an improved feature space for the classifier to separate the two classes, regardless of the geographical positions of the segments. □

Postprocessing is an important component in the analysis phase of the design [137]. Postprocessing is the procedure of converting the output set $\{\vec{y}\}$, produced by the supervised classifier, back into either the space of the original data set or to a more user-friendly format. This extracts information from the results produced by the learning algorithm and is used to improve the overall system performance.

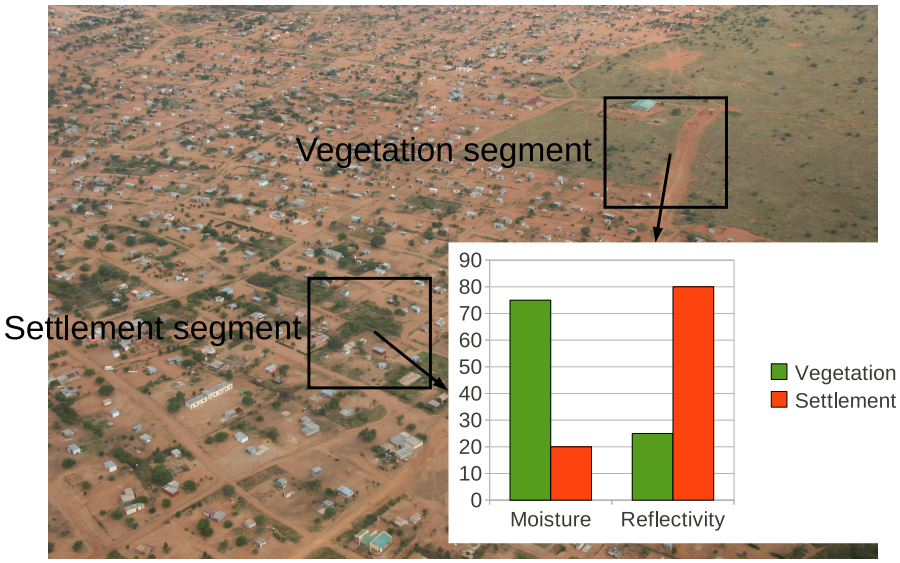


FIGURE 3.5: A new histogram created by extracting the feature vectors of the new segments selected in figure 3.4.

Numerous methods are used for postprocessing, which are categorised as: knowledge filtering, interpretation, evaluation and knowledge integration [137].

1. Knowledge filtering is the filtering of the outputs produced by the supervised classifier. This filtering improves the results when the mapping function in the supervised classifier is sensitive to the noise within the training data set.

2. Interpretation is a form of knowledge discovery where input vectors are processed by the supervised classifier and converted to an user-friendly format for human analysis. These postprocessed outputs are analysed to interpret the effect of the input vectors has on the supervised classifier. This creates a new knowledge base for further improving the results of the supervised classifier for the given application.

3. Evaluation is an approach that transforms the output values into a performance metric that is used to evaluate the performance of the current supervised classifier. Typical performance metrics include: classification accuracy, comprehensibility, computational complexity, visual interpretation, etc.

4. Knowledge integration is the process of including additional selected information sources to improve the performance of the supervised classifier.

Land cover example: In the case of the land cover example, the evaluation approach is used as a postprocessing step. The classification accuracy is used as the performance metric to evaluate the segment classification within the aerial photo. The supervised classifier produces an output vector $\vec{y}$ of either discrete, continuous or probabilistic in nature.

Let the output vector $\vec{y}$ in this example denote the vector containing all the posterior class probability values. The mapping of this vector to a class is expressed as

$$\mathcal{C}_k = \begin{cases} \mathcal{C}_1(\text{natural vegetation}) & \text{if } y_1 > y_2 \\ \mathcal{C}_2(\text{human settlement}) & \text{if } y_2 \geq y_1. \end{cases} \tag{3.12}$$

The output vector $\vec{y}$ is classed as natural vegetation when the largest value in the vector is in the first position and human settlement when in the second position. The classification accuracy is maximised by selection of the most appropriate supervised classifier and feature extraction method. □

The preprocessing of the input vector $\vec{\tilde{x}}$ will produce a new input vector $\vec{x}$ that is commonly referred to as the feature vector. Feature vectors will be used throughout the thesis as it is assumed that with proper feature extraction the overall system performance will improve.

3.4 ARTIFICIAL NEURAL NETWORKS

An artificial neural network (ANN) is a computational learning method that was inspired by the neural activities within the human brain [138]. ANNs have a range of capabilities to operate on non-linear and non-parametric data sets. The advantage of the ANN is that it can model a non-linear relationship between the input and output variables. The ANN is trained on a partial set of known data to perform either classification, estimation, simulation or prediction of underlying structures within the data.

3.4.1 Network architecture

3.4.1.1 Perceptron

The first design consideration that will be evaluated is the network architecture, as several different ANN architectures are proposed in the literature. The simplest architecture is the single-layer perceptron, which is a linear feedforward neural network that was first proposed by Frank Rosenblatt at the Cornell Aeronautical Laboratory in 1957 [139]. The perceptron is discussed, as several other concepts expand on it, as well as the important limitation the perceptron has in terms of the range of functions it can represent. The perceptron is classified as a feedforward network, as the activation of the neuron is propagated in one direction from the feature vector $\vec{x}$ to the output value y. The relationship between the feature vectors and the output is stored within the ANN's weight vector (also referred to as the synaptic strengths within the ANN), and is defined within the network as

$$y = \mathcal{F}(\vec{x}, \vec{\omega}). \tag{3.13}$$

The variable y denotes the corresponding ANN's output value and $\vec{\omega}$ denotes the weight vector. The feature vector presented to the network is denoted by $\vec{x}$ and $\mathcal{F}$ denotes the function inferred by the ANN. The weight vector $\vec{\omega}$ and the feature vector $\vec{x}$ are multiplied such that equation (3.13) expands in the case of the perceptron to

$$y = \mathcal{F}\left(\omega_0 + \sum_{i=1}^{N} x_i\omega_i\right) = \mathcal{F}\left(\omega_0 + \vec{x}\cdot\vec{\omega}\right). \tag{3.14}$$

The symbol $\mathcal{F}$ denotes the activation function and the network inputs are denoted by the feature vector $\vec{x} = \{x_1, x_2, \ldots, x_N\}$. The weight vector for the network is denoted by $\vec{\omega} = \{\omega_1, \omega_2 \ldots, \omega_N\}$ and the neuron bias by ω_0.

The perceptron is trained with the perceptron learning rule, which minimises the error function by evaluating the output value produced for a given feature vector. The perceptron learning rule processes individual feature vectors $\vec{x}$ by presenting them to the network and adjusting the weight

vector $\vec{\omega}$ iteratively to improve the classification accuracy. The perceptron learning rule attempts to fit a linear hyperplane through the feature space. The perceptron learning rule is limited by the network architecture and will only converge if the classes are linearly separable within the feature space [140, 141]. Other applications involving multiple separation regions are catered for by using multiple perceptrons in parallel, with each output value corresponding to a specific region.

3.4.1.2 Multilayer perceptron

A more popular network architecture is the multilayer perceptron (MLP). A MLP is a feedforward ANN model that contains multiple layers of neurons. The multilayer architecture allows the MLP to distinguish feature vectors within a feature space that are not linearly separable. A two-layer network architecture of a MLP, which has one hidden node layer, is illustrated in figure 3.6.

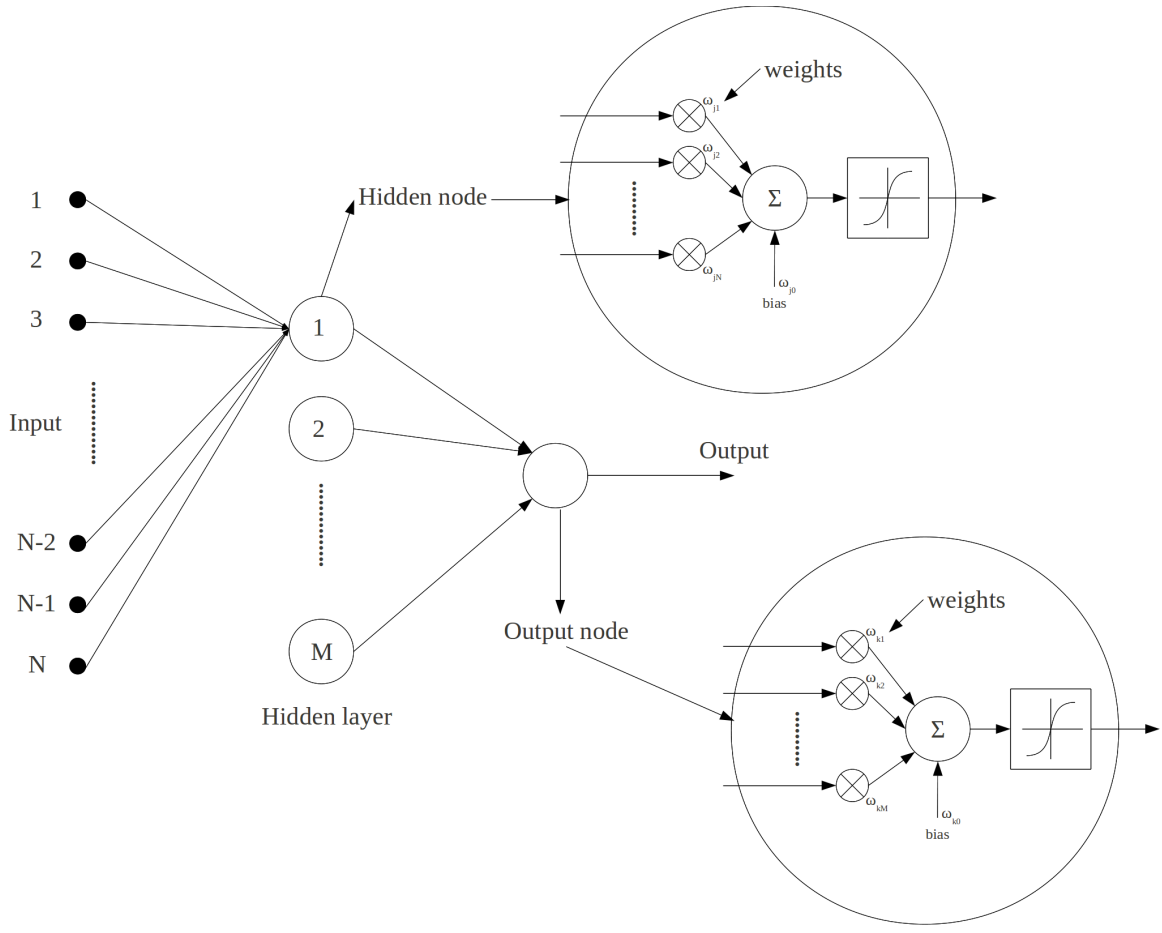


FIGURE 3.6: The topology of a feedforward multilayer perceptron with a single hidden layer.

This fully connected two-layer network's links are mathematically expressed as

$$y_k = \mathcal{F}_2\left(\omega_{k0} + \sum_{j=1}^{M} \omega_{kj}\, \mathcal{F}_1\left(\omega_{j0} + \sum_{i=1}^{N} x_i \omega_{ji}\right)\right), \tag{3.15}$$

which is more compactly expressed in vector notation as a linear multiplication between vectors as

$$y_k = \mathcal{F}_2\left(\omega_{k0} + \vec{\omega}_k \cdot \mathcal{F}_1\left(\omega_{j0} + \vec{x} \cdot \vec{\omega}_j\right)\right). \tag{3.16}$$

The network consists of N input nodes denoted by the vector $\vec{x} = \{x_1, x_2, \ldots, x_N\}$. The weight vector that connects the input nodes to the j^{th} hidden node is denoted by the vector $\vec{\omega}_j = \{\omega_{j1}, \omega_{j2} \ldots, \omega_{jN}\}$, with a corresponding neuron bias denoted by ω_{j0}. Similarly, the weight vector that connects the hidden nodes to the k^{th} output node is denoted by the vector $\vec{\omega}_k = \{\omega_{k1}, \omega_{k2} \ldots, \omega_{kM}\}$, with a corresponding neuron bias denoted by ω_{k0}. The MLP allows the use of multiple output nodes to produces an output vector that expands equation (3.16) to

$$\vec{y}_k = \mathcal{F}_2\left(\omega_{k0} + \vec{\omega}_k \cdot \mathcal{F}_1\left(\omega_{j0} + \vec{x} \cdot \vec{\omega}_j\right)\right), \tag{3.17}$$

with an output vector $\vec{y}_k$ that uses a *one-of-c* coding.

Introducing a unity input on each neuron, $x_0 = 1$, the weight vector is expanded to include the neuron bias as $\vec{\omega}_j = \{\omega_{j0}, \omega_{j1} \ldots, \omega_{jN}\}$ for the hidden nodes and $\vec{\omega}_k = \{\omega_{k0}, \omega_{k1} \ldots, \omega_{kM}\}$ for the weight vector for the output nodes. This simplifies equation (3.17) to

$$\vec{y}_k = \mathcal{F}_2\left(\vec{\omega}_k \cdot \mathcal{F}_1\left(\vec{x} \cdot \vec{\omega}_j\right)\right). \tag{3.18}$$

Monotonic functions are usually used as activation functions. Neural networks typically use a sigmoid activation transfer function in the hidden layers given in equation (3.18) as

$$\mathcal{F}(a) = \frac{1}{1 + e^{-a}}. \tag{3.19}$$

The sigmoid activation function is non-linear and allows the outputs of the neural network to be interpreted as a posterior class probability [130, Ch. 6 p. 234]. If all the activation functions within the network are converted to linear functions, then an equivalent single layer linear network without any hidden layers can be derived. This follows from the observation that the composition of successive linear transformations is itself a linear transformation [130, Ch. 4 p. 121].

By applying a linear transformation to equation (3.19), a tangent activation function is derived as

$$\mathcal{F}(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}}. \tag{3.20}$$

The tangent activation function is of interest as through empirical simulations it has been proven to provide faster training of the network (section 3.4.4) [130, Ch. 4 p. 127].

The number of layers and hidden nodes within each layer are flexible design parameters. The general rule is that the layers and nodes are chosen to best model the feature space. It is known from the Kolmogorov theorem that a two-layer network with finitely many discontinuities can closely approximate any decision boundary to arbitrary precision using a sufficient number of hidden nodes with sigmoidal activation functions [142].

Several different network architectures exist and are constructed on similar concepts. The focus of this chapter will be on the MLP, but different ANNs will be briefly discussed in this chapter.

3.4.2 Regression using a multilayer perceptron

Regression analysis is a method for modelling and analysing a set of variables that focuses on the mapping relationship between a dependent variable and multiple independent variables. This extends to the understanding of inherent changes in the dependent variable when any one of the independent variables is altered. An ANN is seen as a flexible non-linear regression method, which is readily deduced from equation (3.18), where the network uses a training algorithm to find a weight $\vec{\omega}$ to map a relationship between the feature vectors and the output vectors.

The training algorithm trains the network by presenting the patterns of the training set to the network, and adjusting the weights (synapse strengths) to minimise the error function. The training algorithm derives the optimal weight by using the error function given in equation (3.4) as

$$\vec{\omega}_{opt} = \underset{\vec{\omega}\in\Omega}{\operatorname{argmin}}\{\mathcal{E}\} = \underset{\vec{\omega}\in\Omega}{\operatorname{argmin}}\left\{ -\sum_{p=1}^{P} p(T_{\mathcal{C}}^{p} \mid \vec{x}^{p}) - \sum_{p=1}^{P} P(\vec{x}^{p}) \right\}. \tag{3.21}$$

The vector $\vec{\omega}_{opt}$ denotes the optimised weight that provides the optimal fit for the mapping that is found within the weight space Ω. $P(\vec{x}^{p})$ denotes the probability of observing the p^{th} feature vector and $p(T_{\mathcal{C}}^{p} \mid \vec{x}^{p})$ denotes the conditional probability density of the target value $T_{\mathcal{C}}^{p}$ given that the feature vector $\vec{x}^{p}$ is present. The probability of observing the p^{th} feature vector denoted by $P(\vec{x}^{p})$ is an additive constant in equation (3.21), and can not be improved through the network architecture or learning algorithm procedures [130, Ch. 6 p. 195]. This term is dropped to simplify equation (3.21) to

$$\vec{\omega}_{opt} = \underset{\vec{\omega}\in\Omega}{\operatorname{argmin}}\left\{ -\sum_{p=1}^{P} p(T_{\mathcal{C}}^{p} \mid \vec{x}^{p}) \right\}. \tag{3.22}$$

The SSE function given in equation (3.5) is usually used as the error function in the MLP and is substituted into equation (3.22) to compute the optimised weight as

$$\vec{\omega}_{opt} = \underset{\vec{\omega} \in \Omega}{\operatorname{argmin}} \left\{ 0.5 \sum_{p=1}^{P} \left\| \mathcal{F}(\vec{x}^p, \vec{\omega}) - T_{\mathcal{C}}^p \right\|^2 \right\}. \tag{3.23}$$

The symbol $\mathcal{F}$ denotes the MLP's inferred map and $\vec{x}^p$ denotes the p^{th} feature vector with the corresponding target value denoted by $T_{\mathcal{C}}^p$. The training algorithm attempts to find the optimal weight $\vec{\omega}_{opt}$ that provides the smallest error function value $\mathcal{E}$.

3.4.3 Classification using a multilayer perceptron

The case was made that an ANN can be interpreted as a non-linear regression model in section 3.4.2. A regression model is used to construct a classifier, which is used to interpret the dependent variable as a posterior class membership probability. These posterior probabilities yield the most likely class for each feature vector.

The reconstruction of the regression model to behave like a classifier starts by using a 1-of-c coding output vector as shown in equation (3.18). The output layer responds like a logistic regression model when sigmoid activation functions are used in each output node [130, Ch. 6 p. 232].

By setting the target value for each training pattern to the desired posterior class probability, with a 1-of-c coding , the MLP is trained in the same manner as a regression model to obtain the optimal weight $\vec{\omega}_{opt}$. Using the optimal weight $\vec{\omega}_{opt}$, the ANN maps the feature vectors to their corresponding desired posterior class probabilities.

Since each MLP output node represents the posterior class probability for each class, a mapping function is used to select the class that has the largest posterior probability. The mapping function $\mathcal{Z}$ is expressed as

$$\mathcal{C}_k = \mathcal{Z}(\vec{y}), \tag{3.24}$$

where $\mathcal{C}_k$ denotes the class membership and $\vec{y}$ denotes the MLP output vector.

Deriving the optimal weight $\vec{\omega}_{opt}$ will assign the highest posterior class probability to the correct class membership $\mathcal{C}_k$ for the corresponding feature vector $\vec{x}$ and is expressed as

$$P(\mathcal{C}_k = \mathcal{C}_f | \vec{x}) > P(\mathcal{C}_k = \mathcal{C}_g | \vec{x}) \qquad \forall (f \neq g), \tag{3.25}$$

where $P(\mathcal{C}_k = \mathcal{C}_f | \vec{x})$ denotes the probability of class membership of $\mathcal{C}_k$ being equal to $\mathcal{C}_f$, given the feature vector $\vec{x}$ was presented to the MLP.

The probability of error is equal to the probability of falling within the incorrect decision region [143]. The probability of error for the class membership $(\mathcal{C}_k = \mathcal{C}_c)$ of the MLP is computed as

$$P_e = 1 - \int_{\mathcal{R}_c} p(\vec{x}|\mathcal{C}_k = \mathcal{C}_c)P(\mathcal{C}_k = \mathcal{C}_c)\mathrm{d}\,\vec{x}. \tag{3.26}$$

The procedure of minimising the probability of error P_e on the global population group of feature patterns, requires that the complete population's class memberships be known. This is not possible for most actual data sets (non-synthetic), as acquiring the class membership on all feature vectors is infeasible. The objective of the training algorithm is to minimise the probability of error P_e on the global population by only using a subset of feature vectors with known class membership.

An external evaluation process is used for minimising the probability of error, as discussed in section 3.3.1, that is used to improve overall system performance. The subdivision of the labelled data set (feature vectors with known class memberships) for the MLP is briefly discussed:

1. A training data set is used to train the ANN to minimise the mapping errors on the data set by means of adaptation of the weights. A popular method of calculating the error in the mapping is the SSE shown in equation (3.5). The minimisation of the error is accomplished by initialisation the weights with random values, followed by presenting the training data set to the network to adjust the weights accordingly. Several different training algorithms exist in the literature that attempts to minimise the error on the training data set.

2. A validation data set is periodically used to test the network performance to mitigate the effects of overfitting [135]. A neural network with more hidden nodes has the ability to learn a more complex mapping [144]. A complex mapping in the feature space has the ability to isolate complex regions [145]. If proper design of the MLP is not adhered to, the network not only extracts the characteristics of the feature space, but also memorises the noise within the training data set.

3. A test data set is used to validate the performance of the MLP. The test data set is used to estimate the generalisation error, and this data set is not included in the training phase or optimisation phase.

3.4.4 Training of neural networks

As stated previously, the MLP network relies on the weights to assign the feature vector to the class membership that has the largest posterior probability. This is under the assumption that the optimal weight $\vec{\omega}_{opt}$ is used to provide the decision regions. The design of a proper MLP requires the estimation of a weight $\vec{\omega}$ that will minimise the error function and generalisation error for an application.

The error function $\mathcal{E}(\vec{\omega})$ is improved with a training algorithm by searching through the weight space Ω, that uses the SSE metric given in equation (3.5), which is continuous and twice differentiable in $\mathbb{R}^{|\vec{\omega}|}$, where $|\vec{\omega}|$ denotes the total number of weights in the network.

A local minimum of $\mathcal{E}(\vec{\omega})$ is defined as a vector $\vec{\omega}_{\text{local}}$, such that $\mathcal{E}(\vec{\omega}_{\text{local}}) \leq \mathcal{E}(\vec{\omega})$ for all $|\vec{\omega}_{\text{local}} - \vec{\omega}| < D_{\vec{\omega}}$ in $\mathbb{R}^{|\vec{\omega}|}$, where $D_{\vec{\omega}}$ is a predefined constant.

It is possible that $\mathcal{E}(\vec{\omega})$ may contain multiple local minima. Let S_{local} denote the set of all such local minima of $\mathcal{E}(\vec{\omega})$ on $\mathbb{R}^{|\vec{\omega}|}$. The global minimiser of $\mathcal{E}(\vec{\omega})$ is then defined as

$$\vec{\omega}^* = \underset{\vec{\omega} \in S_{\text{local}}}{\operatorname{argmin}} \mathcal{E}(\vec{\omega}). \tag{3.27}$$

Note that $\mathcal{E}(\vec{\omega}^*) \leq \mathcal{E}(\vec{\omega}), \forall\, \vec{\omega} \in \mathbb{R}^{|\vec{\omega}|}$. In addition, the derivative of the error function, $\nabla\mathcal{E}(\vec{\omega})$, is zero for all $\vec{\omega} \in S_{\text{local}}$.

Owing to the non-linear nature of the error function $\mathcal{E}(\vec{\omega})$, no closed form solution can be obtained. Many iterative algorithms can be applied to minimise the error function $\mathcal{E}(\vec{\omega})$, most of which iteratively adjust the current weight $\vec{\omega}_i$ such that

$$\vec{\omega}_{(i+1)} = \vec{\omega}_i + \Delta\vec{\omega}_i, \tag{3.28}$$

where $\Delta\vec{\omega}_i$ is typically chosen such that $\mathcal{E}(\vec{\omega}_{i+1}) < \mathcal{E}(\vec{\omega}_i)$. The manner in which $\Delta\vec{\omega}_i$ is determined at each epoch i, will allow the algoirthm to converge to either a local minimum or a global minimum of the error function $\mathcal{E}(\vec{\omega})$.

Owing to the inherent difficulty of reliably locating the global minimum $\vec{\omega}^*$ of the error function $\mathcal{E}(\vec{\omega})$, most algorithms instead attempt to find the best local minimum, given a finite number of iterations, which may be called an *acceptable local minimum* for a given training data set.

Another important aspect that should be considered is that the global minimum of the error function $\mathcal{E}(\vec{\omega})$ on a given training data set may not necessarily result in the best generalisation performance for the application, hence it is typically sufficient to find an *acceptable local minimum* [130, Ch. 6 p. 194]. Several different approaches to calculating the weight update set $\vec{\omega}_i$ in equation (3.28) will now be discussed.

3.4.5 First order training algorithms

3.4.5.1 Gradient descent

The gradient of the error function $\mathcal{E}(\vec{\omega})$ always points in the direction in which $\mathcal{E}(\vec{\omega})$ will decrease most rapidly in its local vicinity. Algorithms that exploit the gradient information can typically locate a minimum in fewer iterations than algorithms that do not use gradients. The gradient descent algorithm

propagates along the negative slope of the error function [146]. The weight update $\Delta\vec{\omega}_i$ given in equation (3.28) is iteratively computed in the gradient descent approach at each epoch i as

$$\Delta\omega_i = -\mathfrak{L}_i \nabla\mathcal{E}|_{\vec{\omega}_i} + \mathcal{M}\Delta\omega_{(i-1)}. \tag{3.29}$$

The variable $\mathfrak{L}_i$ denotes the learning rate and $\mathcal{M}$ denotes the momentum parameter. The derivative of the error surface evaluated at weight $\vec{\omega}_i$ is denoted by $\nabla\mathcal{E}|_{\vec{\omega}_i}$. The algorithm incorporates a learning rate parameter $\mathfrak{L}_i$ that scales the rate of propagation of the weight down the negative slope. The correct adjustment of the learning rate improves the convergences onto a local minimum of $\mathcal{E}(\vec{\omega})$. If the learning rate is set too high, the algorithm has difficulty in stabilising the weight and might cause $\vec{\omega}_i$ to oscillate around the minimum, preventing convergence. When the learning rate is set too low, the algorithm takes a long time to converge. Common practice states a gradual decrease in the learning rate $\mathfrak{L}_i$ during training minimises the chance of oscillations within the training process.

Additional information for the training algorithm is acquired from the eigenvalues of the Hessian matrix of the error. The learning rate can be set to $\mathfrak{L}_i = (2/\lambda_{\max})$ to improve the performance further, where $\lambda_{\max}$ denotes the largest eigenvalue in the Hessian matrix [147]. The disadvantage is that the Hessian matrix varies as the weight is updated at each iteration with $\Delta\omega_i$ and calculating the Hessian matrix is computationally expensive.

If the Hessian matrix is calculated, a metric is defined for characterising the expected rate of convergence of steepest descent. This metric is the ratio of the smallest eigen value $\lambda_{\min}$ and the largest eigen value $\lambda_{\max}$ and is expressed as

$$R(\lambda) = \frac{\lambda_{\min}}{\lambda_{\max}}. \tag{3.30}$$

A very small value of $R(\lambda)$ usually means that the error surface contours are highly elongated elliptical in shape and the progress to the minimum will be extremely slow when using steepest gradient descent. The momentum parameter $\mathcal{M}$ is used for compensating when the ratio $R(\lambda)$ is small [148]. The momentum term leads to faster convergence towards the minimum without causing divergent oscillations, which may appear when the learning rate is too large. The momentum parameter acts as a lowpass filter to incorporate recent trends in movement along the error surface. Inclusion of momentum generally leads to a significant improvement in the performance of gradient descent.

3.4.5.2 Resilient backpropagation

Resilient backpropagation (RPROP) is a first-order heuristic algorithm that is used for training a feedforward neural network [149]. The RPROP algorithm is based on the notion that the optimal

step size, at a given iteration, will differ for each dimension of $\vec{\omega}_i$. RPROP thus maintains a separate weight update step $\Delta\vec{\omega}_{i,j}$ for each dimension j. A heuristic is employed to adjust each $\Delta\vec{\omega}_{i,j}$ at every epoch as follows; if the sign of the gradient dimension j has changed from that of the previous epoch, reduce the step size $\Delta\vec{\omega}_{i,j}$ and reverse its sign, otherwise increase the step size $\Delta\vec{\omega}_{i,j}$.

The reasoning is that the gradient sign in dimension j will change if the algorithm has moved over a local minimum, thus the algorithm must take smaller steps in the following iterations to approach the minimum. This is analogous to implementing standard steepest descent, but with a separate adaptive learning rate for each dimension.

3.4.5.3 Quickprop

The last heuristic first order training algorithm that will be discussed in the section is the Quickprop algorithm [150]. Quickprop treats each weight within the network as quasi-independent. The idea is to approximate the error surface with a quadratic polynomial function. The gradient information derived with backpropagation is used to determine the coefficients of the polynomial. The step sizes are fixed within the weight to ensure that the algorithm will converge to a minimum. The Quickprop algorithm uses a local quadratic surface and cannot distinguish between propagating upwards or downwards on the error surface. This drawback is easily overcome by determining the propagation direction by using an algorithm such as the gradient descent algorithm in the first epoch.

3.4.5.4 Line search

The line search is a one dimensional minimisation problem, which finds the minimum of the error function along a particular search direction [151]. It is used in several different algorithms to reduce computational complexity and will be discussed briefly. Suppose that a certain algorithm is considering a particular search direction $\vec{d}_i$ through the weight space for a potential future weight update (equation (3.28)), the minimum along that particular search direction is calculated as

$$\vec{\omega}_{(i+1)} = \vec{\omega}_i + \Delta_d \vec{d}_i, \tag{3.31}$$

where the step size parameter Δ_d is calculated as

$$\mathcal{E}(\Delta_d) = \underset{\Delta_d \in \mathbb{R}}{\operatorname{argmin}}\, \mathcal{E}(\vec{\omega}_i + \Delta_d \vec{d}_i). \tag{3.32}$$

In summary, the line search finds the optimal step size for a selected search direction. The line search algorithm itself has several constraints, as every line minimisation involves several internal error function evaluations, which could be computationally expensive. Line search introduces additional

parameters whose values will determine the termination criterion for each line search.

3.4.5.5 Conjugate gradient

The concept of choosing improved search directions is the main principle behind the conjugate gradient algorithm [130, 152]. The conjugate gradient algorithm evaluates the performance of conjugate directions with line search algorithms. The conjugate gradient algorithm is an iterative approach and is applied with ease to applications having feature vectors with several dimensions. The conjugate gradient algorithm operates under the assumption of a quadratic error function with a positive definite Hessian matrix [130, Ch. 7 p. 276].

Owing to the fact that most data sets have a non-quadratic error surface, there is a high probability that if the step size is small enough, the evaluation of $\mathcal{E}(\vec{\omega}_i + \Delta\vec{\omega}_i)$ will fall on an error surface that is approximately quadratic in its local vicinity. This may lead to fast convergence to a minimum. Under similar reasoning, if the local vicinity of the error surface is non-quadratic, the conjugate gradient algorithm will converge slowly to the minimum.

The performance of the conjugate gradient algorithm is dependent on the type of line search algorithm used. Line search allows the conjugate gradient algorithm to find the step size without evaluating the Hessian matrix.

3.4.6 Second order training algorithms

The successive use of the local gradient vector as the search direction does not always result in the most optimal search trajectory. The local gradient does not necessarily point directly at the minimum, which may cause oscillating behaviour in a steepest descent algorithm. This slow progression to the minimum can even be present with a quadratic error surface for poorly conditioned networks. The convergence speed can be improved by evaluating and choosing superior search directions while propagating down the error surface.

3.4.6.1 Newton method

The Newton method is an algorithm that calculates the Newton direction by assuming a positive definite Hessian matrix and a quadratic error surface. The trajectory from the current weight to a nearby minimum is known as the Newton direction. There are three obstacles when using the Newton method [130, Ch. 7 p. 286]:

1. The calculation of the Hessian matrix is computationally expensive for a non-linear MLP which requires $\mathcal{O}(P|\vec{\omega}|^2)$ operations to compute, where P is the number of feature vectors to evaluate

and $|\vec{\omega}|$ is the dimension of the weights.

2. The calculation of the inverted Hessian matrix is also computationally expensive, as it requires $\mathcal{O}(|\vec{\omega}|^3)$ iterations to compute.

3. Regardless of whether the Hessian matrix is positive definite, the Newton direction can point to either a maximum or a minimum.

The third obstacle can be resolved by using a model trust region approach that adds a positive definite symmetrical matrix to the Hessian matrix [130, Ch. 7 p. 287], which is expressed as

$$\mathbf{H}_{\text{new}} = \mathbf{H}_{\text{old}} + A\mathbf{I}. \tag{3.33}$$

The matrix $\mathbf{H}_{\text{old}}$ is the current Hessian matrix and $\mathbf{H}_{\text{new}}$ is the adjusted Hessian matrix. The identity matrix is denoted by $\mathbf{I}$ and A denotes a constant factor. Equation (3.33) provides the Newton direction if the constant factor A is set to a small value or it can provide the negative gradient descent direction if the constant factor A is set to a large value [130, Ch. 7 p. 287].

The last consideration is the step size along the Newton direction. The step size calculated within the Newton method is made under the assumption that the error surface is quadratic in shape. Most real data sets have non-quadratic error surfaces and when the step size is too large, the algorithm may fail to converge.

3.4.6.2 Quasi-Newton method

A more practical implementation of the Newton method is the Quasi-Newton method. The Quasi-Newton method is an approximation of the Newton method, as the Hessian matrix is computationally expensive for complex neural networks [153]. The Quasi-Newton method approximates the inverted Hessian matrix over several iterations, using only the first derivative of the error function. After each iteration the estimated inverse Hessian matrix approximates more closely the real inverse Hessian matrix for a given weight.

A popular quasi-Newton algorithms is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. The BFGS algorithm updates the estimated Hessian matrix in each epoch to converge to the actual Hessian matrix. The algorithm starts with the identity matrix to ensure that the minimum is tracked and not the maximum. The length of the Newton step is calculated using a proper line search to ensure stability. The accuracy of the line search is not as critical as it was with the conjugate gradient algorithm [154].

The disadvantages of the Newton and the Quasi-Newton methods are the storage requirements and the number of iterations to approximate the Hessian matrix [130, Ch. 7 p. 289]. Because of the

non-quadratic error surface of most data sets, the approximate Hessian matrix must be estimated after each weight update to ensure correct minimisation of the error function. The second disadvantage of these methods is the introduction of the model trust region constant factor A and the correct scaling of this constant.

3.4.6.3 Levenberg-Marquardt algorithm

The last second order training algorithm that will be discussed in the section is the Levenberg-Marquardt algorithm [155, 156]. The Levenberg-Marquardt algorithm is an approach to derive the second-order derivative without computing the Hessian matrix, as with the Quasi-Newton method. The Levenberg-Marquardt algorithm is specifically designed to minimise the SSE. This is accomplished by approximating the function in equation (3.5) with linearisation as

$$\mathcal{F}(\vec{x}_i, \vec{\omega}_i + \Delta\omega_i) \approx \mathcal{F}(\vec{x}_i, \vec{\omega}_i) + \vec{J}_i \Delta\omega_i. \tag{3.34}$$

The vector $\vec{J}_i$ is a gradient row vector of $\mathcal{F}$ with respects to $\vec{\omega}_i$ and is computed as

$$\vec{J}_i = \frac{\partial \mathcal{F}(\vec{x}_i, \vec{\omega}_i)}{\partial\, \vec{\omega}_i}. \tag{3.35}$$

Substituting the approximation of equation (3.34) into equation (3.5) is expressed as

$$\mathcal{E}(\vec{\omega} + \Delta\omega_i) = 0.5 \sum_{p=1}^{P} \left\| \mathcal{F}(\vec{x}^p, \vec{\omega}) + \vec{J}_i \Delta\omega_i - T_{\mathcal{C}}^p \right\|^2. \tag{3.36}$$

By setting the derivative as

$$\frac{\partial \mathcal{E}(\vec{\omega} + \Delta\omega_i)}{\partial\, \Delta\omega_i} = 0, \tag{3.37}$$

equation (3.36) can be expressed as

$$(\mathbf{J}^{\mathrm{T}}\mathbf{J})\Delta\omega_i = \mathbf{J}^{\mathrm{T}} \left(0.5 \sum_{p=1}^{P} \left\| \mathcal{F}(\vec{x}^p, \vec{\omega}) - T_{\mathcal{C}}^p \right\|^2 \right). \tag{3.38}$$

The Jacobian matrix is denoted by $\mathbf{J}$, with each row containing $\vec{J}_i$. This Jacobian matrix contains the first derivatives of the neural network's error. Levenberg added a non-negative damping factor λ_{damp}, which is adjusted at each epoch. This is expressed as

$$(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda_{\mathrm{damp}}\mathbf{I})\Delta\omega_i = \mathbf{J}^{\mathrm{T}} \left(0.5 \sum_{p=1}^{P} \left\| \mathcal{F}(\vec{x}^p, \vec{\omega}) - T_{\mathcal{C}}^p \right\|^2 \right). \tag{3.39}$$

A smaller damping factor λ_{damp} value allows the algorithm to behave more like the Newton method, while a larger damping factor λ_{damp} value allows the algorithm to behave like the gradient descent method.

If the damping factor λ_{damp} value is set too high, the inversion of $(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda_{\text{damp}}\mathbf{I})$ contributes nothing to the algorithm. Marquardt then contributes a variable that will scale each component of the gradient according to the curvature. This results in the Levenberg-Marquardt equation given as

$$(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda_{\text{damp}}\text{diag}(\mathbf{J}^{\mathrm{T}}\mathbf{J}))\Delta\omega_i = \mathbf{J}^{\mathrm{T}}\left(0.5\sum_{p=1}^{P}\left\|\mathcal{F}(\vec{x}^{\,p},\vec{\omega}) - T_{\mathcal{C}}^{p}\right\|^2\right), \tag{3.40}$$

where the identity matrix $\mathbf{I}$ in equation (3.39) is replaced to ensure larger propagation in the desired direction when the gradient becomes smaller.

3.5 OTHER VARIANTS OF ARTIFICIAL NEURAL NETWORKS USED FOR CLASSIFICATION

3.5.1 Radial basis function network

The radial basis function (RBF) network is another ANN that is discussed in this chapter [130, 157]. In the case of the MLP, the hidden neurons create multi-dimensional hyperplanes to separate different classes within the feature space. In the case of the RBF network, the network uses local kernel functions, which are represented by a prototype vector within each hidden neuron to model different classes. The activation of the hidden neurons is based on the distance from the prototype vector, which in effect creates a spherical multi-dimensional hypersphere. The RBF network can be used for classification; the posterior class probabilities of the network at the output is computed as

$$p(\mathcal{C}_k|\vec{x}) = \sum_{d=1}^{D}\vec{\omega}_{kd}\varphi_d(\vec{x}). \tag{3.41}$$

The RBF uses D basis functions that are denoted by φ_d. The φ_d basis function in the network's hidden neurons is expressed as a normalised basis function given by

$$\varphi_d(\vec{x}) = \frac{p(\vec{x}\,|\,d)P(d)}{\sum_{e=1}^{E}p(\vec{x}\,|\,e)P(e)} = p(d\,|\,\vec{x}). \tag{3.42}$$

The d^{th} basis function evaluating feature vector $\vec{x}$ is denoted by $\varphi_d(\vec{x})$ [130, Ch. 5 p. 181]. The denominator is used to normalised the basis function by iterating through all the basis functions within the network with variable e. The outputs of all the radial basis functions are linearly combined with a weight vector to form an output vector. The weight vector for each output node is given by

$$\vec{\omega}_{kd} = \frac{p(d \,|\, \mathcal{C}_k) P(\mathcal{C}_k)}{P(d)} = p(\mathcal{C}_k \,|\, d). \tag{3.43}$$

The radial basis function network can be designed in a fraction of the time required to train a MLP, but requires a large sample of input vectors to train reliably [158].

3.5.2 Self organising map

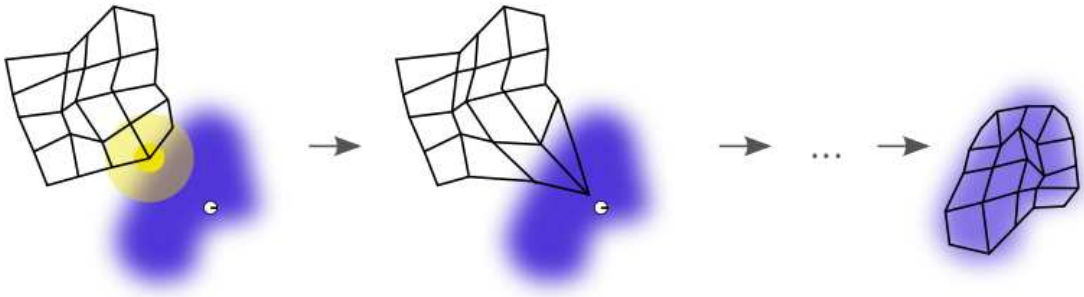


FIGURE 3.7: The training of the SOM will map the gridded topological map to the training data set.

Another popular ANN design is the Self Organising Map (SOM) [159, 160]. The SOM is trained with an unsupervised learning algorithm to convert a high dimensional data set to a lower dimensional representation of the data, typically two-dimensional. The SOM converts the higher dimensional data set to a lower dimension using a topological map that comprises prototype neurons. This topological map is used to illustrate the relationship between feature vectors by placing similar feature vectors in close vicinity to each other on the map and dissimilar feature vectors further apart. Each prototype neuron has a prototype vector; these are comparable to weights in other ANNs, and are initialised to either random samples or uniform subsampling of the feature vector set.

The training algorithm used on the SOM is a competitive learning algorithm which searches for the part of the network that strongly responds to the given feature vector. The response is evaluated by presenting a feature vector $\vec{x}$ to the SOM's prototype neurons to determine the Euclidean distances to all prototype vectors. The prototype neuron with the most similar prototype vector is termed as the best matching unit (BMU). The prototype vector within the BMU is adjusted towards the feature vector. The prototype neurons in close vicinity of the BMU in the topological map are known as the neighbouring neurons and are also updated to a certain degree towards the current feature vector. The magnitude of the adaptation of the neighbouring neurons decreases with epochs and distance from the BMU.

A SOM is trained in batch mode, where all the feature vectors are presented to the network and only the BMU is trained. A monotonically increasing penalty factor is added to that feature vector to

ensure that a particular feature vector does not dominate the training algorithm. In the beginning of the training phase, the neighbourhood relationship within the topological map is large, but with each epoch the mapping of neighbourhood size shrinks within the map and the network converges (Figure 3.7). The creation of a topological map, particularly if the data are not intrinsically two-dimensional, may lead to suboptimal placement of the feature vectors [130, Ch. 5 p. 188].

3.5.3 Hopfield networks

The third ANN briefly discussed is the Hopfield network. A Hopfield network is a recurrent network with feedback loops between the outputs and the inputs [161–163]. The neurons in the Hopfield network have binary threshold activation functions and the internal state of the network evolves to a stable state that is a local minimum of the Lyapunov function. The Lyapunov function is a monotonically decreasing energy function that puts less emphasis on the previous set of feature vectors than on the current set of feature vectors. A Hopfield network is an associative memory, which enables it to train on a set of target vectors, and when a new set of feature vectors are presented it will cause the network to settle into an activation pattern corresponding to the most closely resembling target vector presented in the training phase. The drawback of the Hopfield network is that it can only retrieve all the fundamental memorised target vectors [164].

3.5.4 Support vector machine

A Support Vector Machine (SVM) is a supervised learning algorithm that was developed in the AT&T Bell laboratories in 1995. SVM is based on the principle of structural risk minimisation, which involves constructing a non-linear hyperplane with kernel functions to separate the feature space into several output regions [129].

The SVM training algorithm attempts to fit a non-linear hyperplane through the feature space. It focuses on maximising the distance between the decision boundary and the sets of feature vectors. The SVM is a maximum margin classifier and does this by identifying the feature vectors within the feature space that prohibits the training algorithm from increasing the margin between the output regions. These feature vectors are called the support vectors within the feature space.

The method by which the SVM handles non-separable feature vectors is relaxing the constraints on the hyperplane that maximises the separability. This is accomplished by including a cost function into the separating marginal regions and penalises the feature vectors that severely hinders the SVM's performance.

The advantage of a SVM is that it uses a weighted sum of kernel functions to separate the feature vectors in the feature space. The kernel functions reduce the number of dimensions and decouples the

computational complexity of the SVM from the feature vector's dimensionality. Another advantage is that it is less prone to overfitting. If the hyperplanes are properly designed, the results of the SVM are similar to a properly designed MLP classifier [165].

A disadvantage in the SVM is that the choice of kernel used in the algorithm is very important. Several adjustable dimensions of the parameters are encapsulated within the kernel, which only leaves the penalty parameter available for adjustment. Proper choice of kernel is still an active research field; using prior knowledge during kernel selection usually improves performance. Further disadvantages are potentially slow training and substantial memory usage during training. It is observed that the speed is significantly reduced when training on larger data sets [129].

The last design consideration is the proper setting of the penalty term used to classify non-separable feature vectors. This penalty term must be optimised either through brute force searching or any other heuristic search methods.

3.6 DESIGN CONSIDERATION: SUPERVISED CLASSIFICATION

In this section a brief overview is given of some considerations when designing a supervised classifier. The first consideration is the investigation of the input vector set $\{\vec{x}\}$ and the desired output vector set $\{\vec{y}\}$. The first question is whether a plausible mapping function exists that can successfully map the input space to the output space with meaningful descriptors. Should the input vector set $\{\vec{x}\}$ be preprocessed into a feature vector set $\{\vec{x}\}$ and should the output vector set $\{\vec{y}\}$ be postprocessed to improve overall performance? This analysis provides insight into all further design decisions.

On completing the analysis, the next step is finding a suitable supervised classifier. The choice of ANN and the corresponding training algorithm is critical in finding acceptable performance in the mapping. The reason why only acceptable performance is pursued, rather than optimal, is that finding the best feature vector set and the optimal supervised classifier requires an exhaustive search, which is not feasible in terms of computational costs.

The adaptation for using a supervised classifier optimally entails the use of a proper training algorithm. Training algorithms typically focus on monotonically decreasing the value of the error function. Unfortunately, this type of training algorithm is more prone to becoming trapped in a local minimum when a small incremental steps are used. If the incremental step size is too large, the training algorithm will overshoot the minima. The convergence rate of the training algorithm is hindered even more when the direction of the propagation in the error surface does not point to the minimum. Several different training algorithms try to find the direction to the minimum since the local gradient does not always point straight at the minimum.

The training algorithm utilises training patterns in two general methods: iterative and batch

learning. Batch learning is an offline learning method that evaluates all the available training patterns before adapting the network. Iterative learning can either be online or offline, as it only evaluates sequentially a single training pattern before adapting the network [166]. An offline system stores all its patterns in a data set, while an online system processes and discards a pattern.

Another important consideration is that most ANNs are prone to overfit. This can be controlled by proper implementation of an early stopping criteria. The most common methods of stopping a training algorithm are:

1. The preset number of epochs is reached.
2. The predetermined computational time has expired in the execution of the training algorithm.
3. The training algorithm is stopped when a predefined lower threshold of the error function is reached.
4. The training algorithm is stopped when the first derivative of error function falls below a predefined lower threshold.
5. The error on the validation data set (section 3.4.3) is minimised.

It is commonly believed that a MLP with many hidden neurons has a high generalisation error, as the network is more prone to overfit [130, Ch. 1 p. 14]. This excess capacity (large number of hidden neurons) offers the MLP the ability to learn more complex models. If too much training is applied on a MLP, with excess capacity, it starts to learn the intrinsic noise within the data set. This is an undesirable property in most applications of a supervised classifier and much emphasis is placed on limiting the capacity of the network to prevent overfitting (Occam Razor's principle). It is also commonly believed that a MLP network with a large number of hidden neurons requires a large number of training vectors (section 3.4.3) to find a suitable mapping function between the feature and output space [167].

This common knowledge was questioned when a contradiction was shown by Caruana *et al.* [168]. They showed that a MLP with excess capacity has better generalisation error than a MLP with sufficient capacity. A MLP can be trained to map highly non-linear regions with a large number of hidden neurons, but still have the ability to retain a proper mapping of the linear regions [168] with a limited number of training patterns.

The concept is based on a slowly converging training algorithm that will first train the linear regions and then progress to the non-linear regions. If a good stopping criterion is adhered to, the training algorithm will terminate properly before it overfits. Some second-order methods, e.g. conjugate gradient descent algorithm, do not exhibit this property, as they have fast convergence, and will indeed overfit if the network has excess capacity.

This behaviour is intrinsically built into the slower training algorithms, as the set of weights $\{\vec{\omega}\}$ is usually initialised with small non-zero values and only after many epochs do certain values within the weights tend to large values. This implies that the MLP first considers simple mapping functions before exploring more complex functions [168, 169].

Small initial values are used within the weights to ensure that there is no saturation of the sigmoidal activation function. This initialisation ensures that contours are created on the error surface when backpropagation is applied in the training phase, otherwise the saturation of the sigmoidal activation functions will create a very flat error surface.

The last design consideration is the choice of initial weights, which is very important in achieving good results. A suitable initial choice has the potential of allowing the training algorithm to train fast and efficiently. Even stochastic algorithms, such as gradient descent, which have the possibility of escaping from local minima, can be sensitive to the initial weights used. This results in the rule of thumb to run several training phases with different initial weights in parallel to evaluate the performance of different minima [130, Ch. 7 p. 260].

The ANN used in this thesis is the MLP with a stochastic gradient descent as used by Caruana *et al.* [168]. The gradient descent uses a learning and momentum parameter in the training process to speed up convergences and a validation data set to apply proper early stopping.

3.7 SUMMARY

This chapter presented a methodology for designing a supervised classifier for real world applications. Emphasis was placed on the design of a proper mapping function between the input and output space. The mapping function's fit was then measured using a suitable error function. The performance of the classifier improves when a training method is used which adapts the network to minimise the error function.

This can be seen as a regression approach to determine the relationship between the dependent and independent variables within the network. The output values produced by the network can be interpreted as a set of posterior class probabilities under certain assumptions. The chapter concludes with a range of good practice notes on how to design and develop a good supervised classifier.

CHAPTER FOUR

UNSUPERVISED CLASSIFICATION

4.1 OVERVIEW

In this chapter a brief overview is given of the notion of grouping objects into different categories without any supervision. The previous chapter described a supervised approach to grouping objects and how the relationship between the desired class membership and input vectors was derived using labels. The possibility is now explored of grouping objects based on their perceived intrinsic similarities. A formal definition is provided on an unsupervised method known as clustering, followed by the advantages of exploring an unsupervised approach. The design considerations behind producing good clustering results are then explored, followed by the challenges inherent when using clustering methods to solve real world problems.

Clustering algorithms are broadly divided into hierarchical and partitional clustering approaches [40, 170]. Four popular hierarchical clustering methods and two partitional clustering methods are discussed with their corresponding properties. The chapter concludes with a discussion on how clusters can be converted to classes to obtain a supervised classifier.

4.2 CLUSTERING

Clustering is a form of conceptual clustering, which is an unsupervised method used for grouping unlabelled input vectors into a set of categories. Clustering groups a set of input vectors through perceived intrinsically similar or dissimilar characteristics.

Let $\{y^k\}$, $y^k \in \mathbb{N}$, $1 \leq y^k \leq K$, denotes the set of cluster labels. Let $\mathcal{F}_{\mathcal{C}} : \mathbb{R}^n \rightarrow \{y^k\}$ denote the function that maps the input vector $\vec{x}^p$, $\vec{x}^p \in \mathbb{R}^n$, to a cluster label. The variable p denotes the index of the vector within the input vector set. The function $\mathcal{F}_{\mathcal{C}}$ is said to cluster the input vector set $\{\vec{x}^p\}$ into K clusters.

Several motivations exist to justify the use of clustering algorithms for many non-synthetic data sets:

1. Significant costs are involved when gathering information about the data set to create reliable class labels for supervised classification.
2. The underlying data structure of a large unlabelled data set can be captured to provide reliable clustering on a smaller labelled data set.
3. Accommodate a dynamic input space. This is when the input space changes over time or in response to a triggered event.
4. Assisting in creating a well-conditioned input vector from the input space to gain insight into what improves the cluster label allocation.

4.2.1 Mapping of vectors to clusters

A cluster label is derived by evaluating several different input data sources from the input space. These data sources are grouped together to form an input vector $\vec{\vec{x}}$. These input vectors are the same as with the supervised classifier and have descriptive forms that can be interpreted. The preprocessing and postprocessing of the input and output vectors is an optional procedure used to improve the clustering algorithm's performance [136]. Using feature vectors $\vec{x}$ and postprocessed output value y is assumed to improve the performance significantly and is used throughout this chapter.

The clustering algorithm constructs a function $\mathcal{F}_{\mathcal{C}}$ to determine the cluster label and is based on the set of feature vectors $\{\vec{x}^p\}$. The mapping function is expressed as

$$y^k = \mathcal{F}_{\mathcal{C}}(\vec{x}^p). \tag{4.1}$$

The clusters typically encapsulate properties of the non-synthetic data set; each cluster should have a homogeneous set of feature vectors.

4.2.2 Creating meaningful clusters

No theoretical guideline exists on how to extract the optimal feature vector set from the input vector set for a specific clustering application. Owing to the limited intrinsic information within the feature vector set, it is difficult to design a clustering algorithm that will find clusters to match the desired cluster labels.

This constraint is created by a clustering algorithm, as it tends to find clusters in the feature space irrespective of whether any real clusters exist. This constraint motivates the notion that any two

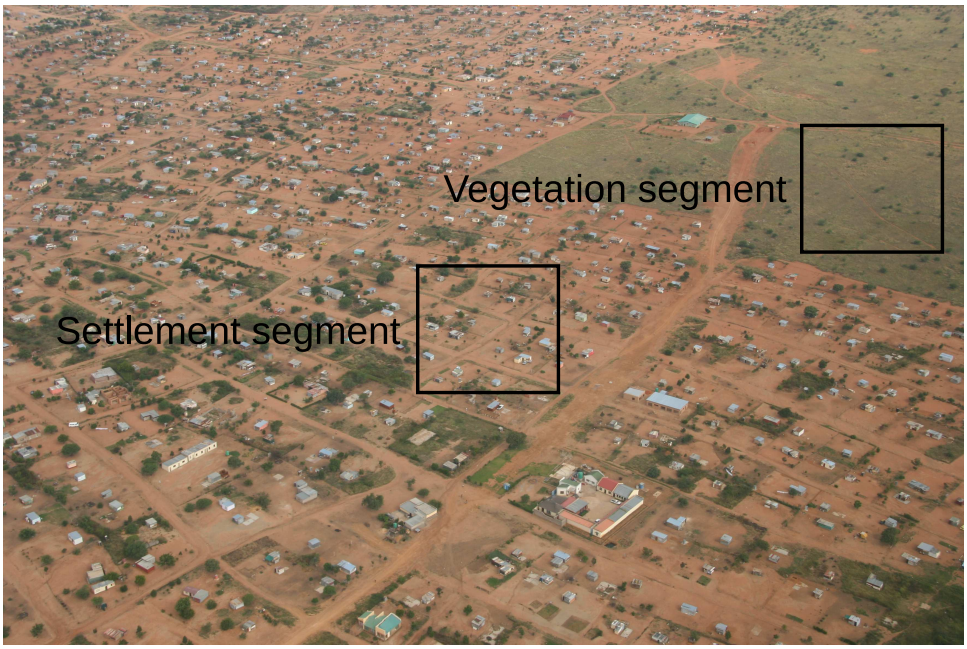


FIGURE 4.1: An aerial photo taken in the Limpopo province, South Africa of two different land cover which are indicated by a natural vegetation segment and settlement segment. A segment is defined as a collection of pixels within a predefined size bounding box.

arbitrary patterns can be made to appear equally similar when evaluating a large number of dimensions of information in the feature space. This will result in defining a meaningless clustering function $\mathcal{F}_C$. This makes clustering a subjective task in nature, which can be modified to fit any particular application.

The advantage in this versatility is that the clustering algorithm can be used as either an exploratory or a confirmatory analysis tool [170]. Clustering used as an exploratory analysis tool is there to explore the underlying structures of the data. No predefined models or hypotheses are needed when exploring the data set. Clustering used as a confirmatory analysis tool is to confirm any set of hypotheses or assumptions. In certain applications, clustering is used as both; first to explore the underlying structures to form new hypotheses. Second, to test these hypotheses by clustering the feature vector set. This makes clustering a data-driven learning algorithm and any domain knowledge that is available can improve the forming of clusters [170].

Domain knowledge is used to reduce complexity by aiding in processes such as feature selection and feature extraction. Proper domain knowledge leads to good feature vector representation that will yield exceptional performance with the most common clustering algorithms, while incomplete domain knowledge leads to poor feature vector representation that will only yield acceptable performance with a complex clustering algorithm.

An aerial photo is used to illustrate the clustering of different land cover types in figure 4.1. In this

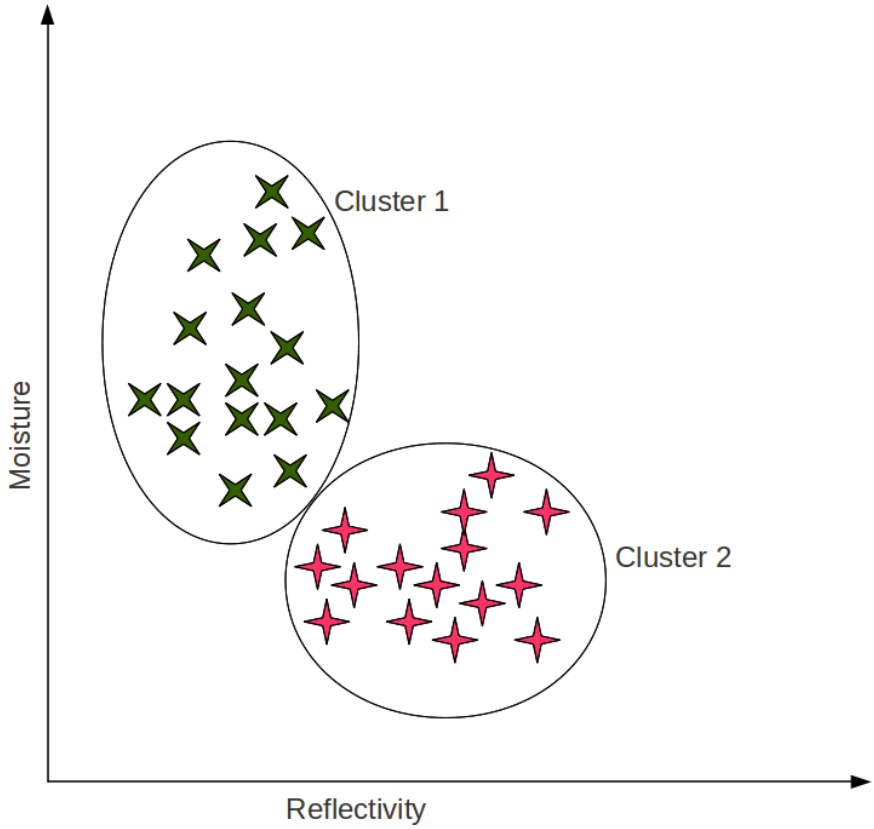


FIGURE 4.2: A two-dimensional illustration of feature vectors within the feature space. The green feature vectors represent the natural vegetation class and the red feature vectors represent the human settlement class.

image two land cover types are of interest: natural vegetation and human settlement.

Land cover example: In the case of the land cover example shown in figure 4.1, domain knowledge is used for feature extraction and selection. Let it be assumed that the domain knowledge provided information that the feature vector given in equation (4.2) will provide better separability between the two categories.

$$\vec{x} = [(\text{Moisture})\ (\text{Reflectivity})]. \tag{4.2}$$

The natural vegetation segments have feature vectors with low reflectivity and high moisture levels, while the human settlement segments have feature vectors with high reflectivity and low moisture levels. This is illustrated in a two-dimensional plot shown in figure 4.2. When natural clusters exist in the feature space and the number of clusters is set to K=2, a well-designed clustering algorithm will produce two perfect clusters, as shown in figure 4.2. □

Domain knowledge in many fields is incomplete or unavailable. Verifying the domain knowledge

from actual (non-synthetic) data sets is extremely resource-expensive and is difficult to relate to the feature space. The most practical approach for designing an unsupervised learning algorithm is to *learn from example* [171]. The *learning from example* approach requires that the clustering algorithm be subjected to an external evaluation process. The external evaluation is hampered by the fact that thousands of different clustering algorithms have been developed and evidence suggests that none of them is superior to any other [172]. This is addressed in the *impossibility theorem*, which states three criteria which no clustering algorithm can satisfy [172]. The three criteria to satisfy in the *impossibility theorem* are:

1. Scale invariance; the scaling of the feature vectors should not change the assigned cluster labels.
2. Richness; the clustering algorithm must be able to achieve all possible partitions in the feature space.
3. Consistency; the change in distance within all clusters will not change the assigned cluster labels.

Based on the *impossibility theorem*, each clustering application is different and requires an unique design to obtain good clustering results. This emphasises the importance of obtaining *acceptable performance* in the search for a clustering algorithm, as it is infeasible to search through all the permutations of clustering designs. The admissibility criterion is a more practical approach to consider when applying external evaluation to a clustering algorithm [170]. The admissibility criterion comprises three important design considerations:

1. The manner in which the clusters are formed.
2. The intrinsic structure of the feature vectors.
3. The sensitivity of the clusters created.

4.2.3 Challenges of clustering

Humans cluster with ease in two and three dimensions, while a machine learning method is required to cluster in higher dimensions. Several design implications arise when clustering in higher dimensions [171]:

- Determining the number of clusters K (section 4.6).
- Determining whether the feature vectors carry representative information to produce clusters that will hold a relation to the desired classes for the application (section 4.2.2).

- Deciding which pairwise similarity metric should be used to evaluate the feature space (section 4.3).

- Determining how the feature vectors should be evaluated to form clusters. Clustering algorithms are broadly divided into hierarchical and partitional clustering approaches [40, 170]. The first approach is hierarchical clustering, which produces a nested hierarchy of clusters of discrete groups (section 4.4). The second approach is partitional clustering, which creates an unnested partitioning of the data points with K clusters [173] (section 4.5).

4.3 SIMILARITY METRIC

A clustering algorithm defines clusters with feature vectors that are similar to one another, and separate them from feature vectors that are dissimilar. This similarity between feature vectors is usually measured using a distance function.

Let $\{\vec{x}\}$, $\vec{x} \in \mathbb{R}^N$ denote a set of N-dimensional feature vectors. Let $D : \mathbb{R}^N \rightarrow \mathbb{R}_+$ denote the distance function that calculates the distance between the vector $\vec{x}^p$ and $\vec{x}^q$. The function D is said to return the distance (similarity metric) between the two feature vectors.

The properties of the distance function D are:

- Non-negative, $D(\vec{x}^p, \vec{x}^q) \geq 0$.

- Identity axiom, $D(\vec{x}^p, \vec{x}^q) = 0$, iff $p = q$.

- Triangle inequality, $D(\vec{x}^o, \vec{x}^p) + D(\vec{x}^p, \vec{x}^q) \geq D(\vec{x}^o, \vec{x}^q)$.

- Symmetry axiom, $D(\vec{x}^p, \vec{x}^q) = D(\vec{x}^q, \vec{x}^p)$.

The non-negative and identity axioms produce a positive definite function. The distance metric is as important in the design as the clustering algorithm itself. Proper selection of a distance metric will result in the distance between feature vectors of the same cluster being smaller than the distance between the feature vectors of other clusters.

Choosing a distance function opens a broad class of distance metrics. The first to consider is the general Minkowski distance, which is used to derive some of the most common distance functions used in clustering applications. The Minkowski distance D_{mink} is expressed as

$$D_{\text{mink}}(\vec{x}^p, \vec{x}^q) = \left(\sum_{n=1}^{N} |x_n^p - x_n^q|^m \right)^{\frac{1}{m}}. \tag{4.3}$$

The variable $m, m \in \mathbb{N}$, is the Minkowski parameter that is used to adjust the nature of the distance metric. The Minkowski distance simplifies to the popular Euclidean distance D_{ed} if the Minkowski parameter m is set to 2 in equation (4.3). The Euclidean distance is computed as

$$D_{\text{ed}}(\vec{x}^p, \vec{x}^q) = \sqrt{\sum_{n=1}^{N} |x_n^p - x_n^q|^2}. \tag{4.4}$$

The advantage of the Euclidean distance is that it is invariant to translation or rotation of the feature vector $\vec{x}$. The Euclidean distance however does vary under an arbitrary linear transformation.

The squared Euclidean distance is an alteration to the Euclidean distance, as it places a greater weight on a set of vectors that are considered to be outliers in the vector space. The squared Euclidean distance is expressed as

$$D_{\text{sq}}(\vec{x}^p, \vec{x}^q) = \sum_{n=1}^{N} |x_n^p - x_n^q|^2. \tag{4.5}$$

If the Minkowski parameter is set to m=1, equation (4.3) simplifies to the Manhattan distance. The Manhattan distance is the sum of the absolute difference between vectors. The Manhattan distance is expressed as

$$D_{\text{man}}(\vec{x}^p, \vec{x}^q) = \sum_{n=1}^{N} |x_n^p - x_n^q|. \tag{4.6}$$

The Mahalanobis distance metric is used in statistics to measure the correlations between multivariante vectors. The Mahalanobis distance metric D_{mahal} is expressed as

$$D_{\text{mahal}}(\vec{x}^p, \vec{x}^q) = \sqrt{(\vec{x}^p - \vec{x}^q) G_{\text{mahal}}^{-1} (\vec{x}^p - \vec{x}^q)}, \tag{4.7}$$

where G_{mahal} denotes the covariance matrix.

4.4 HIERARCHICAL CLUSTERING ALGORITHMS

A clustering algorithm uses a set of feature vectors $\{\vec{x}^p\}$, cluster parameters and a similarity metric to construct a mapping function $\mathcal{F}_{\mathcal{C}}$. Let $\vartheta = (\cup_{q=1}^{Q} \vartheta^q)$ denote the set of cluster parameters that the clustering algorithm needs to determine when constructing $\mathcal{F}_{\mathcal{C}}$.

As stated previously, clustering algorithms are broadly divided into either a hierarchical or partitional clustering approach [40, 170]. The hierarchical clustering approach produces a nested hierarchy of clusters of discrete groups according to a certain linkage criterion. The nested clusters are

recursively linked in either an agglomerative mode or divisive mode. The second approach to clustering is partitional clustering, which creates an unnested partitioning of the vectors into K clusters [173]. In hierarchical clustering using an agglomerative mode, the clustering parameter set $\{\vartheta\}$ is determined iteratively in four steps:

Step 1: The clustering algorithm starts by allocating each feature vector to its own cluster. The initialisation phase is defined as

$$\vartheta_I^p = \vec{x}^p, \qquad \forall p \text{ and } I = 0. \tag{4.8}$$

The variable ϑ_I^p denotes the p^{th} set of cluster parameters at epoch I, with I set to zero for the initialisation phase. The vector $\vec{x}^p$ denotes the p^{th} feature vector.

Step 2: The similarity between two clusters is defined by a linkage criterion. The linkage criterion evaluates two clusters using a similarity metric (section 4.3) to compute the dendrogrammatic distance $T(\vartheta_I^l, \vartheta_I^k)$. The dendrogrammatic distance is computed as

$$T(\vartheta_I^l, \vartheta_I^k) = \beta(\vartheta_I^l, \vartheta_I^k), \tag{4.9}$$

where the linkage criterion is denoted by the function β, $\beta \in \{T_{\text{sing}}, T_{\text{com}}, T_{\text{ave}}, T_{\text{ward}}\}$.

This expression states that all the feature vectors in cluster y^l must be compared to all the feature vectors in cluster y^k using a predefined argument. The linkage criterion's function β returns a dendrogrammatic distance between the two clusters.

Step 3: Select the shortest dendrogrammatic distance $T(\vartheta_I^l, \vartheta_I^k)$ between all pairs of clusters. Let $\vartheta_I^{l^*}$ and $\vartheta_I^{k^*}$ be selected such that

$$[\vartheta_I^{l^*}, \vartheta_I^{k^*}] = \underset{l,k \in [1,K]; l \neq k}{\operatorname{argmin}} \; T(\vartheta_I^l, \vartheta_I^k). \tag{4.10}$$

Step 4: Merge the two clusters with index l^* and k^* as

$$\vartheta_{(I+1)}^{l^*} = \left(\vartheta_I^{l^*} \cup \vartheta_I^{k^*}\right), \tag{4.11}$$

$$\vartheta_{(I+1)}^{k^*} = \emptyset. \tag{4.12}$$

Steps 2–4 are repeated until all the clusters are merged into a single cluster. The sequence of merging clusters can be graphically presented by a tree diagram, called a dendrogram. The dendrogram is a multi-level hierarchy with two clusters merging at each level.

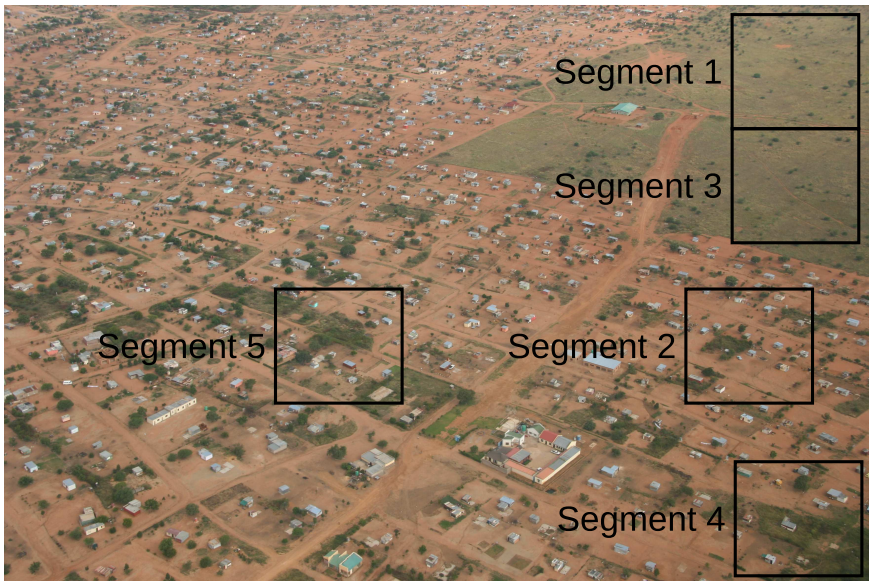


FIGURE 4.3: An alternative selection of five new segments of the aerial photo taken in the Limpopo province which indicates different types of land cover types.

Land cover example: Five new segments are defined in figure 4.3. A hierarchical clustering algorithm operating in agglomerative mode creates a dendrogram shown in figure 4.4 when applied to the five segments. In the first iteration the similarity between segment 4 and segment 5 is the highest (shortest dendrogrammatic distance). These segments are merged to form a new cluster. The dendrogrammatic distances between the merging clusters are indicated on the vertical axis. The shorter the distance on the vertical axis, the more similar the two joining clusters. In the second iteration, segment 1 and segment 3 are joined as being the next most similar clusters. These two newly formed clusters are joined together, as they are more similar to each other than to segment 2. Segment 2 is joined to form a single cluster containing all segments, which completes the dendrogram.

In the divisive mode, the clustering algorithm starts by placing the entire feature vector set in a single cluster. In this mode, a comparison is made between all the feature vectors within the cluster to determine which feature vectors are the most dissimilar and split the cluster into two separate clusters. This process is repeated until every single cluster retains a single feature vector. The sequence of separating the clusters is also represented on a dendrogram. Only the agglomerative mode was considered, as it is a bottom-up approach and the concept could easily be derived for a divisive mode with the same methodology in a top-down approach.

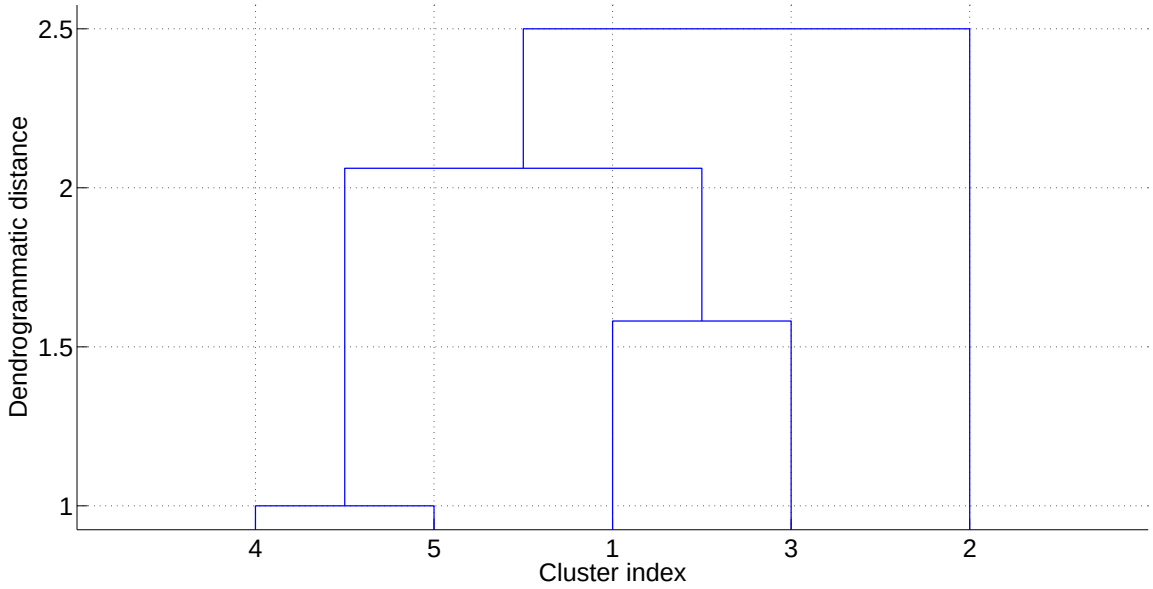


FIGURE 4.4: An illustration of an hierarchical clustering approach operating in agglomerative mode.

4.4.1 Linkage criteria

4.4.1.1 Single linkage criterion

The merging of clusters is based on the dendrogrammatic distance between clusters. The dendrogrammatic distance is computed using a linkage criterion. The single linkage criterion is the first linkage criterion that is considered, as it searches for the shortest distance between two feature vectors; each residing in two different clusters. The single linkage criterion $T_{\text{sing}}(\vartheta_I^l, \vartheta_I^k)$ is expressed as

$$T_{\text{sing}}(\vartheta_I^l, \vartheta_I^k) = \min\{D(\vec{x}^p, \vec{x}^q)\} \quad \forall \vec{x}^p \in \vartheta_I^l,\ \vec{x}^q \in \vartheta_I^k \text{ and } l \neq k. \tag{4.13}$$

The variable $\vec{x}^p$ denotes the p^{th} feature vector and $\vec{x}^q$ denotes the q^{th} feature vector. The similarity metrics shown in section 4.3 (equation (4.3)–(4.7)) or any other distance metric found in the literature can be used as the distance metric $D(\vec{x}^p, \vec{x}^q)$. The single linkage criterion has a chaining effect as a characteristic trait when forming clusters. This results in clusters that are straggly and elongated in shape [174]. The advantage of elongated clusters is that they can extract spherical clusters from the feature space.

4.4.1.2 Complete linkage criterion

The complete linkage criterion computes a dendrogrammatic distance by finding the maximum possible distance between two feature vectors that reside in different clusters. The complete linkage

criterion $T_{\text{com}}(\vartheta_I^l, \vartheta_I^k)$ is expressed as

$$T_{\text{com}}(\vartheta_I^l, \vartheta_I^k) = \max\{D(\vec{x}^p, \vec{x}^q)\} \quad \forall \vec{x}^p \in \vartheta_I^l,\ \vec{x}^q \in \vartheta_I^k \text{ and } l \neq k. \tag{4.14}$$

The variable $\vec{x}^p$ denotes the p^{th} feature vector and $\vec{x}^q$ denotes the q^{th} feature vector. The complete linkage criterion has the characteristic trait of forming tightly bounded compact clusters. The complete linkage criterion creates more useful clusters in many actual (non-synthetic) data sets than the single linkage criterion [170, 175].

4.4.1.3 Average linkage criterion

The average linkage criterion is the most intuitive linkage criterion, as it calculates a dendrogrammatic distance between two clusters by finding the average distance among all pairs of feature vectors residing in different clusters. The average linkage criterion $T_{\text{ave}}(\vartheta_I^l, \vartheta_I^k)$ is expressed as

$$T_{\text{ave}}(\vartheta_I^l, \vartheta_I^k) = \frac{1}{|\vartheta_I^l||\vartheta_I^k|} \sum_{\vec{x}^p \in \vartheta_I^l} \sum_{\vec{x}^q \in \vartheta_I^k} D(\vec{x}^p, \vec{x}^q), \quad l \neq k. \tag{4.15}$$

$|\vartheta_I^l|$ denotes the number of feature vectors in cluster ϑ_I^l and $|\vartheta_I^k|$ denotes the number of feature vectors in cluster ϑ_I^k. The average linkage criterion is a compromise between the complete linkage criterion's sensitivity to outliers and the chaining effect produced by the single linkage criterion.

4.4.1.4 Ward criterion

The Ward criterion computes a dendrogrammatic distance between clusters by finding the clusters that will maximise the coefficient of determination R^2 [176]. The Ward criterion $T_{\text{ward}}(\vartheta_I^l, \vartheta_I^k)$ is expressed as

$$\begin{aligned} T_{\text{ward}}(\vartheta_I^l, \vartheta_I^k) \quad = \quad & \sum_{p \in \left(\vartheta_I^l \cup \vartheta_I^k\right)} \left\| \vec{x}^p - E\left[\vartheta_I^l \cup \vartheta_I^k\right] \right\|^2 - \sum_{p \in \vartheta_I^l} \left\| \vec{x}^p - E\left[\vartheta_I^l\right] \right\|^2 - \\ & \sum_{p \in \vartheta_I^k} \left\| \vec{x}^p - E\left[\vartheta_I^k\right] \right\|^2. \end{aligned} \tag{4.16}$$

The expected value of the feature vectors in the cluster is denoted by $E[\vec{x}^p]$. The Ward criterion attempts to minimise the variance between the K clusters and only uses the Euclidean distance. Most linkage criteria in the literature are variants of the single linkage, complete linkage, average linkage or Ward criterion.

4.4.2 Cophenetic correlation coefficient

A dendrogram is created iteratively as the function $\mathcal{F}_{\mathcal{C}}$ is derived with a hierarchical clustering algorithm. The dendrogram illustrates the dendrogrammatic distances obtained with the linkage criterion (section 4.4.1). The cophenetic correlation coefficient is a statistical measure of correlation between the dendrogrammatic distances and the similarity distances for all pairs of feature vectors [177]. The cophenetic correlation coefficient is computed as

$$D_{\text{cc}} = \frac{\sum_{q=2}^{P}\sum_{p=1}^{q}(D(\vec{x}^{\,p},\vec{x}^{\,q}) - E[D(\vec{x}^{\,p},\vec{x}^{\,q})])(T(\vartheta_0^l,\vartheta_0^k) - E[T(\vartheta_0^l,\vartheta_0^k)])}{\sqrt{\sum_{q=2}^{P}\sum_{p=1}^{q}(D(\vec{x}^{\,p},\vec{x}^{\,q}) - E[D(\vec{x}^{\,p},\vec{x}^{\,q})])^2(T(\vartheta_0^l,\vartheta_0^k) - E[T(\vartheta_0^l,\vartheta_0^k)])^2}}, \tag{4.17}$$

with $\vec{x}^{\,p} \in \vartheta_0^l$ and $\vec{x}^{\,q} \in \vartheta_0^k$. The function $D(\vec{x}^{\,p},\vec{x}^{\,q})$ denotes the distance between the feature vector $\vec{x}^{\,p}$ and $\vec{x}^{\,q}$ as shown in section 4.3. The $T(\vartheta_0^l,\vartheta_0^k)$, $\vec{x}^{\,p} \in \vartheta_0^l$, $\vec{x}^{\,q} \in \vartheta_0^k$, denotes the dendrogrammatic distance between the feature vector $\vec{x}^{\,p}$ and $\vec{x}^{\,q}$ as shown in equation (4.9). The higher the correlation, the better the dendrogram preserves the information of the feature space when using a particular linkage criterion. The cophenetic correlation coefficient is used to evaluate several different distance metrics and linkage criteria that will best retain the original distances of the feature space in the dendrogram [177].

4.5 PARTITIONAL CLUSTERING ALGORITHMS

A partitional clustering algorithm operates on the actual feature vectors, which significantly reduces the required space and computations to operate, which makes it more suitable for larger data sets when compared to hierarchical clustering [173].

Let $\{y^k\}$, $k \in \mathbb{N}$, $1 \leq k \leq K$ denote the set of cluster labels. Let $\mathcal{F}_{\mathcal{C}} : \mathbb{R}^N \rightarrow \{y^k\}$ denote the function that maps feature vectors $\{\vec{x}\}$, $\{\vec{x}\} \in \mathbb{R}^N$, onto the clusters. Then $\mathcal{F}_{\mathcal{C}}$ is said to cluster $\vec{x}$ into K clusters.

In a general case of partitional clustering, a set of clustering parameters is determined when constructing the mapping function $\mathcal{F}_{\mathcal{C}}$. Let $\{\vartheta_I^k\}$, $\{\vartheta_I^k\} \in \Omega_\vartheta$, denote the set of clustering parameters. The variable k, $1 \leq k \leq K$, denotes the index in the set $\{\vartheta_I^k\}$ which refers to the cluster label y^k. The variable I denotes the current epoch. The partitional clustering algorithm uses a distance metric $D(\vec{x}^{\,\text{p}},\vartheta_I^k)$ to measure the distance between the p^{th} feature vector $\vec{x}^{\,\text{p}}$ and cluster y^k. The feature vector $\vec{x}^{\,\text{p}}$ is then mapped onto $\{y^k\}$ using the function $\mathcal{F}_{\mathcal{C}}$, such that

$$\mathcal{F}_{\mathcal{C}}(\vec{x}^{\,\mathrm{p}}) = \underset{y^k \in \{y^k\}}{\operatorname{argmin}} \left\{ D(\vec{x}^{\,\mathrm{p}}, \vartheta_I^k) \right\}. \tag{4.18}$$

Intuitively, the function $\mathcal{F}_{\mathcal{C}}$ maps a vector $\vec{x}^{\,\mathrm{p}}$ to the nearest cluster.

The function $\mathcal{F}_{\mathcal{C}}$ is constructed by determining the set of cluster parameters $\{\vartheta_I^k\}$ to minimise the overall distance between a given set of feature vectors $\{\vec{x}\}$ and the K corresponding clusters. One possible definition of this process is

$$\left\{\vartheta_I^{k*}\right\} = \underset{\left\{\vartheta_I^k\right\} \in \Omega_\vartheta}{\operatorname{argmin}} \left\{ \sum_{p=1}^{P} D\big(\vec{x}^{\,\mathrm{p}}, \vartheta_I^{\mathcal{F}_{\mathcal{C}}(\vec{x}^{\,\mathrm{p}})}\big) \right\}. \tag{4.19}$$

The clustering algorithm simultaneously determines the parameters ϑ_I^k of each cluster, as well as the cluster assignment of each feature vector $\vec{x}^{\,\mathrm{p}}$.

4.5.1 K-means algorithm

The first partitional clustering algorithm explored is the popular K-means algorithm [178]. The K-means algorithm attempts to find the center points of the natural clusters. The K-means clustering algorithm accomplishes this by partitioning the feature vectors into K mutually exclusive clusters.

K-means is a heuristic, hill-climbing algorithm that attempts to converge to the center mass point of the natural clusters. It can be viewed as a gradient descent approach which attempts to minimise the sum of squared error of each feature vector to the nearest cluster centroid [179]. The clusters created with the K-means algorithm are compact and isolated in nature.

Minimising the SSE has been shown to be a NP-hard problem, even for a two-cluster problem [180]. This gives rise to a variety of heuristic approaches to solving the problem for practical applications. The most common method of implementing the K-means algorithm is the Lloyd's approach. The Lloyd's approach is an iterative method which comprises three steps:

Step 1: Initialise a set of K centroids $\{\vartheta_I^k\}$.

Step 2: Assign each feature vector to its closest centroid. This is accomplished by creating K empty sets $\vec{s}^{\,k} = \emptyset, k = 1, 2, \ldots, K$, for each of the corresponding centroids $\{\vartheta_I^k\}$. The assignment step is expressed as

$$\vec{s}^{\,k} = \left\{ \{\vec{x}^{\,p}\} : D(\vec{x}^{\,p}, \vartheta_I^k) < D(\vec{x}^{\,p}, \vartheta_I^l), \forall l \neq k \right\}. \tag{4.20}$$

The vector $\vec{x}^{\,p}$ denotes the p^{th} feature vector and D denotes the distance function.

Step 3: The update step adjusts the centroids' position to minimise the sum of distance given in

equation (4.19). The adjustment is made for each centroid as

$$\vartheta^k_{(I+1)} = \frac{1}{|\vec{s}^k|} \sum_{\vec{x}^p \in \vec{s}^k} \vec{x}^p, \qquad \forall\, k. \tag{4.21}$$

$|\vec{s}^k|$ denotes the number of elements in the set.

Steps 2–3 are repeated until all the feature vectors within each cluster remain unchanged or a predefined stopping criterion is reached.

The performance of the K-means algorithm is dependent on the density distribution of the feature vectors in the feature space. K-means will minimise the SSE with high probability to the global minimum if the feature vectors are well separated [181]. The ability of the K-means algorithm to handle a large number of feature vectors enables the parallel execution of multiple replications with different initial seeds to avoid local minima. The K-means clustering algorithm is usually used as a benchmark against other algorithms, and has been used successfully in many other fields [171].

4.5.2 Expectation-maximisation algorithm

The Expectation-Maximisation (EM) algorithm is another partitional clustering algorithm, which attempts to fit a mixture of probability distributions on the set of feature vectors [182]. The EM algorithm was designed on the assumption that the feature vectors are extracted from a feature space with a multi-modal distribution.

Given a set of observable vectors $\{\vec{x}\}$ and unknown variables $\{y^k\}$, the EM algorithm finds the maximum likelihood or maximum *aposterior* estimates for the parameters $\vec{\omega}$, $\vec{\omega} \in \Omega$. The maximum likelihood estimation of the parameters $\vec{\omega}_{\mathrm{ML}}$ is expressed as

$$\vec{\omega}_{\mathrm{ML}} = \underset{\vec{\omega}\in\Omega}{\operatorname{argmax}} \left\{ \log p(\vec{x}|\vec{\omega}) \right\} = \underset{\vec{\omega}\in\Omega}{\operatorname{argmax}} \left\{ \mathcal{J}(\vec{\omega}) \right\}. \tag{4.22}$$

The log-likelihood of the conditional probability in equation (4.22) is expanded to incorporate the unknown variables y^k as

$$\mathcal{J}(\vec{\omega}) = \log p(\vec{x}|\vec{\omega}) = \log \sum_k p(\vec{x}, y^k|\vec{\omega}) = \log \sum_k q(y^k|\vec{x},\vec{\omega}) \frac{p(\vec{x}, y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})}. \tag{4.23}$$

The function $q(y^k|\vec{x},\vec{\omega})$ is an arbitrary density over y^k. Considering the following lower bound inequality to equation (4.23) as

$$\log \sum_k q(y^k|\vec{x},\vec{\omega}) \frac{p(\vec{x}, y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})} \geq \sum_k q(y^k|\vec{x},\vec{\omega}) \log \frac{p(\vec{x}, y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})}, \tag{4.24}$$

which for convenience is rewritten as

$$\mathcal{J}(\vec{\omega}) \geq \sum_{k} q(y^k|\vec{x},\vec{\omega}) \log \frac{p(\vec{x},y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})}. \tag{4.25}$$

It is easier if the EM algorithm instead attempts to maximise the lower bound shown in equation (4.25). The EM algorithm iteratively adjusts the parameters of the distributions in two steps. The first step is the expectation step (E-step) which calculates the log likelihood function, with respect to the conditional distribution of y^k given $\vec{x}$ with the current estimate of the parameter $\vec{\omega}$ as

$$q(y^k|\vec{x},\vec{\omega})^{\text{new}} = \underset{q(y^k|\vec{x},\vec{\omega})}{\text{argmax}} \left\{ \sum_{k} q(y^k|\vec{x},\vec{\omega}) \log \frac{p(\vec{x},y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})} \right\}. \tag{4.26}$$

Calculating the E-step requires the vector $\vec{\omega}$ to be fixed while attempting to optimise over the space of distributions. The second step is the maximisation step (M-step), which tries to maximise the vector $\vec{\omega}$ using the result from equation (4.26). The M-step is computed as

$$\vec{\omega}^{\text{new}} = \underset{\vec{\omega}}{\text{argmax}} \left\{ \sum_{k} q(y^k|\vec{x},\vec{\omega})^{\text{new}} \log \frac{p(\vec{x},y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})^{\text{new}}} \right\}. \tag{4.27}$$

The EM algorithm iterates through both steps until it converges to a local maximum. The feature vector is assigned to a cluster that maximises the *aposterior* probabilities of a given distribution.

The disadvantage of the EM algorithm is that even though the probability of the feature vectors does not decrease, it does not guarantee that the algorithm will converge to the global maximum for a multi-modal distribution. This implies that the EM algorithm can converge to a local maximum. This can be avoided with multiple replications of the algorithm executed with different initial seeds. The EM algorithm is well suited to operate on data sets that contain missing vectors and data sets with low feature space dimensionality.

4.6 DETERMINING THE NUMBER OF CLUSTERS

The most difficult design consideration is to determine the correct number of clusters that should be extracted from the data set. Hundreds of methods have been developed to determine the number of clusters within a data set. The choice in determining the number of clusters K is always ambiguous and is a distinct issue from the process of actually solving the unsupervised clustering problem.

The problem if the number of clusters K is increased without penalty in the design phase (which defeats the purpose of clustering), is that the number of incorrect cluster assignments will steadily decrease to zero. In the extreme case; each feature vector is assigned to its own cluster, which results in zero incorrect clustering allocations. Intuitively this makes the choice in the number of clusters a

balance between the maximum compression of the feature vectors into a single cluster and complete accuracy by assigning each feature vector to it own cluster.

The silhouette value is used as a measure of how close each feature vector is to its own cluster when compared to feature vectors in neighbouring clusters [183]. The silhouette value $\mathcal{S}(\vec{x}^p, K)$ for the feature vector $\vec{x}^p$ is computed as

$$\mathcal{S}(\vec{x}^p, K) = \frac{\min\{\mathcal{S}_{\text{BD}}(\vec{x}^p, l) - \mathcal{S}_{\text{WD}}(\vec{x}^p)\}}{\max\{\mathcal{S}_{\text{WD}}(\vec{x}^p), \min\{\mathcal{S}_{\text{BD}}(\vec{x}^p, k)\}\}}, \quad \forall k, l. \tag{4.28}$$

The function $\mathcal{S}_{\text{WD}}(\vec{x}^p)$ denotes the average distance for the feature vector $\vec{x}^p$ to the other feature vectors in the same cluster. The cluster index is denoted by $k, k \in \mathbb{N}, 1 \leq k \leq K$, and $\mathcal{S}_{\text{BD}}(\vec{x}^p, k)$ denotes the average distance for the feature vector $\vec{x}^p$ to the feature vectors in the k^{th} cluster. The average distance within the same cluster $\mathcal{S}_{\text{WD}}(\vec{x}^p)$ for the feature vector $\vec{x}^p$ is computed as

$$\mathcal{S}_{\text{WD}}(\vec{x}^p) = \left\{ \sum_{q=1}^{|\vartheta^{\mathcal{F}_\mathcal{C}(\vec{x}^p)}|} \frac{D(\vec{x}^p, \vec{x}^q)}{|\vartheta^{\mathcal{F}_\mathcal{C}(\vec{x}^p)}| - 1} : \forall \vec{x}^q \in \vartheta^{\mathcal{F}_\mathcal{C}(\vec{x}^p) \setminus \vec{x}^p} \right\}. \tag{4.29}$$

The variable $|\vartheta^{\mathcal{F}_\mathcal{C}(\vec{x}^p)}|$ denotes the number of feature vectors in the cluster where $\vec{x}^p$ reside. The average distance between the feature vector $\vec{x}^p$ and the k^{th} cluster is computed as

$$\mathcal{S}_{\text{BD}}(\vec{x}^p, k) = \left\{ \sum_{q=1}^{|\vartheta^{\mathcal{F}_\mathcal{C}(\vec{x}^q)}|} \frac{D(\vec{x}^p, \vec{x}^q)}{|\vartheta^{\mathcal{F}_\mathcal{C}(\vec{x}^q)}|} : \forall \vec{x}^q \in \vartheta^{\mathcal{F}_\mathcal{C}(\vec{x}^q)}, \vec{x}^q \notin \vartheta^{\mathcal{F}_\mathcal{C}(\vec{x}^p)}, \mathcal{F}_\mathcal{C}(\vec{x}^q) = y^k \right\}. \tag{4.30}$$

The variable $|\vartheta^{\mathcal{F}_\mathcal{C}(\vec{x}^q)}|$ denotes the number of feature vectors within the k^{th} cluster.

The silhouette value $\mathcal{S}(\vec{x}^p, K)$ ranges from -1 to 1. A silhouette value $\mathcal{S}(\vec{x}^p, K) \to 1$ indicates that the feature vector $\vec{x}^p$ is very distant from the neighbouring K clusters. A silhouette value $\mathcal{S}(\vec{x}^p, K) \to 0$ indicates the feature vector $\vec{x}^p$ is close to the decision boundary between two clusters. A silhouette value $\mathcal{S}(\vec{x}^p, K) \to -1$ indicates that the feature vector $\vec{x}^p$ is probably in the wrong cluster.

A silhouette graph is a visual representation of the silhouette values and is a visual aid used to determine the number of clusters. The x-axis denotes the silhouette values and the y-axis denotes the cluster labels. The silhouette graph shown in figure 4.5 was created from a larger set of segments defined in the example of land cover classification (figure 4.3). In this silhouette graph; cluster 3 has high silhouette values present, which implies that the current feature vectors within cluster 3 are well separated from the other two clusters. Cluster 1 also has high silhouette values, but with a few feature vectors considered to be ill-positioned. Cluster 2 has significantly lower silhouette values and most of its feature vectors are closely positioned at the boundary between clusters. This might suggest that

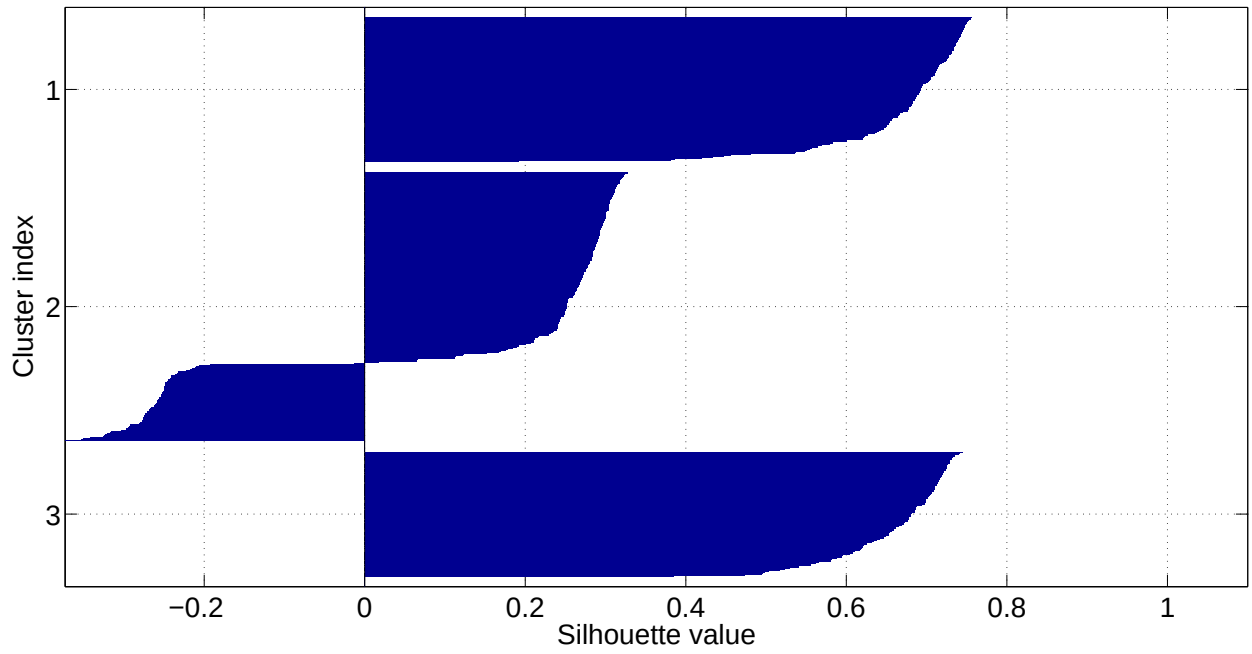


FIGURE 4.5: A silhouette plot of 3 clusters formed of example given in figure 4.3.

cluster 2 can be subdivided into two separate clusters.

An analytical method of deciding on the correct number of clusters K, is the computation of the average of the silhouette value. The average silhouette value is calculated as

$$\mathcal{S}_{\text{ave}}(\{\vec{x}\}, K) = \sum_{p=1}^{P_{\max}} \mathcal{S}(\vec{x}^{\,p}, K), \tag{4.31}$$

where $P_{\max}$ denotes the total number of feature vectors in set $\{\vec{x}\}$. A range of K can be evaluated without any prior knowledge to determine the performance of the clustering algorithm. The number of clusters K that produces the highest average silhouette value is then selected.

4.7 CLASSIFICATION OF CLUSTER LABELS

Clusters typically encapsulate properties of the feature vector set and this homogeneous property motivates the assignment of class labels to the clusters. The class labels are assigned using a supervised classifier, which assigns a set of class labels $\{\mathcal{C}_k\}$ to the K cluster labels [171].

The supervised classifier assigns a class label to a cluster with the most frequently occurring class label from the labelled training data set. Assigning the class labels to the cluster labels with a supervised classifier is expressed as

$$\mathcal{C}_k = \mathcal{Z}(y^{\,k}). \tag{4.32}$$

Owing to the fact that there is no *one cluster represents one class* property, feature vectors of a certain class might end up in the incorrect cluster and therefore be assigned the wrong class label.

Land cover example: The clustering algorithm uses a function $\mathcal{F}_{\mathcal{C}}$ to assign a cluster label to each of the two segments in figure 4.1. The supervised classifier is then used to assign a class label to each of the clusters. In this example the number of clusters K is set to two and the supervised classifier will assign either the natural vegetation class or the human settlement class to the cluster label. This is accomplished by mapping the cluster label y^k, as

$$\mathcal{C}_k = \begin{cases} \mathcal{C}_1(\text{natural vegetation}) & \text{if } y^k = 1 \\ \mathcal{C}_2(\text{human settlement}) & \text{if } y^k = 2. \end{cases} \tag{4.33}$$

The cluster label y^k is classified as natural vegetation when the label is in the first cluster and human settlement when the label is in the second cluster. □

4.8 SUMMARY

In this chapter a methodology was presented to aid in the design process of an unsupervised classifier. The way in which a clustering method tends to find clusters in the feature space irrespective of whether any real clusters exist was discussed. This shows that proper design criteria must be adhered to and the most practical approach to designing a clustering method is to *learn from example* [171].

The design of the clustering method requires the simultaneous optimisation of the:

- feature extraction and feature selection,
- clustering algorithm, and
- similarity metric.

Six popular clustering algorithms were explored. These algorithms are based on basic concepts, which explore the properties of the feature vectors. Thousands of clustering algorithms have been developed in the last couple of decades and most of them only use different permutations and combinations of the concepts defined in these six clustering algorithms. These basic concepts will provide insight into the intrinsic properties of the feature vectors that populate a high-dimensional feature space.

CHAPTER FIVE
FEATURE EXTRACTION

5.1 OVERVIEW

In this chapter, four different feature extraction methods that could be used on time series are investigated. The chapter starts with a discussion on how a series of images are used to create a time series of reflectance values for a particular geographical area. From there the feature extraction methods are discussed, which are:

- EKF,
- least squares model fitting,
- M-estimator model fitting, and
- Fourier transform.

The EKF is a regression approach which uses a process model and an internal state space. The least squares and M-estimator methods are regression approaches that aim to minimise the fitting error (residuals) of a predefined model on a time series. The Fourier transform is a frequency analysis approach, which decomposes time series into several harmonic frequencies.

5.2 TIME SERIES REPRESENTATION

A time series is a sequence of data points measured at successive (often uniformly spaced) time intervals. A time series $\mathbf{x}$ of length $\mathcal{I}$, is defined as

$$\mathbf{x} = [\vec{x}_1\ \vec{x}_2\ \ldots\ \vec{x}_{\mathcal{I}}], \tag{5.1}$$

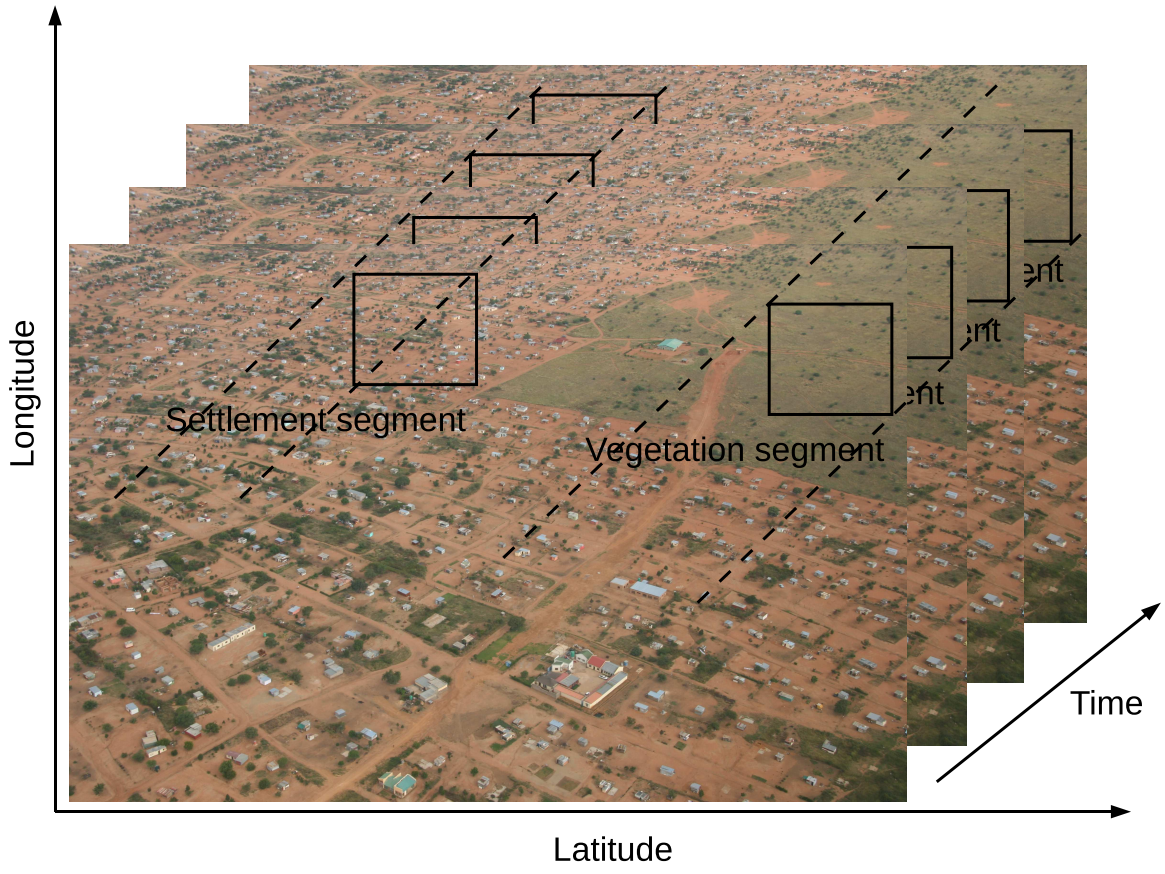


FIGURE 5.1: Multiple aerial photos are acquired in the Limpopo province at different time intervals of the same geographical area. Natural vegetation and human settlement segments are mapped out to form a set of time series.

with

$$\vec{x}_i = [x_{i,1}\ x_{i,2}\ \ldots\ x_{i,T}]. \tag{5.2}$$

The variable T denotes the number of elements in vector $\vec{x}_i$.

The analysis of time series comprises methods that attempt to understand the underlying structure of the data gathered. Analysing the structure allows the identification of patterns and trends, detection of change, clustering, modelling and forecasting [40]. A time series which is extracted from multiple images is used in this chapter to illustrate various concepts.

Land cover example: In figure 5.1, multiple aerial photos are acquired of the same geographical area with segments mapped out over a duration of time. These segments illustrate an example of two different land cover types which do not change over time. The two land cover types are: natural vegetation and human settlement. These hyper-temporal segments are processed to provide a single reflectance value for a given geographical segment at each time interval. A

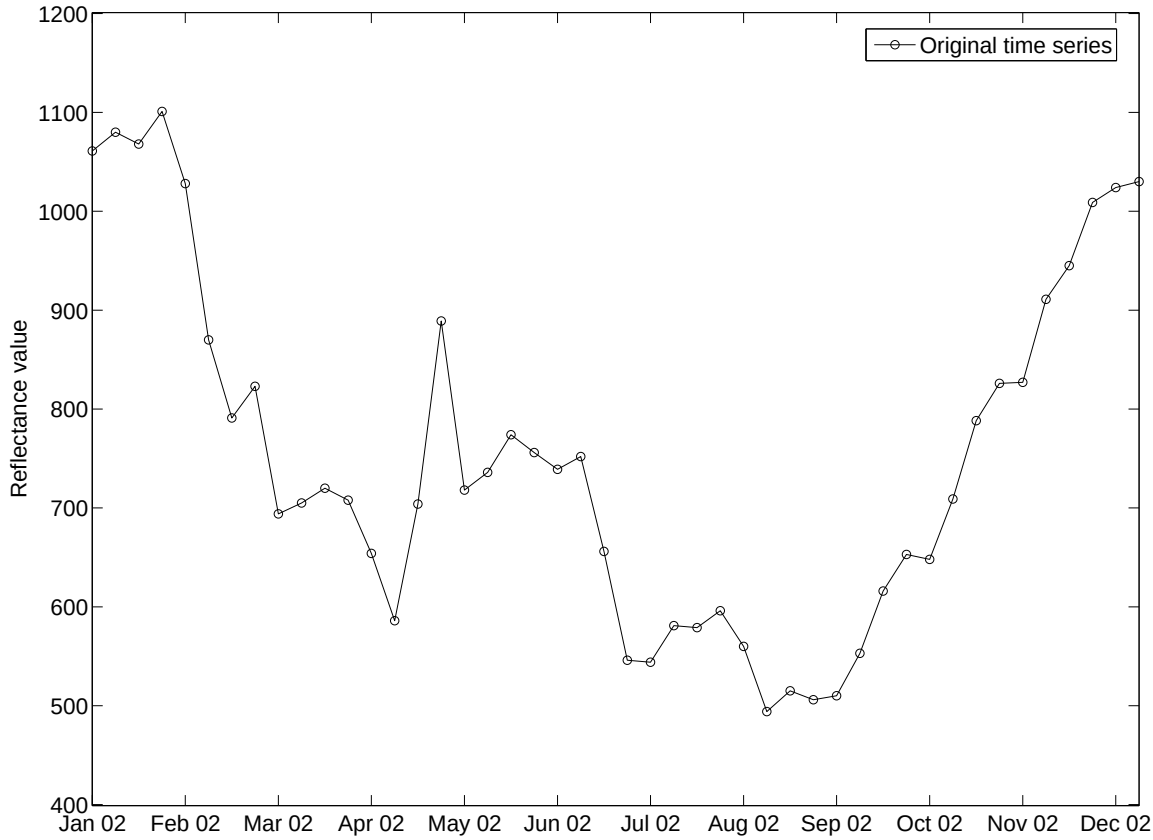


FIGURE 5.2: Time series consisting of reflectance values reported through time for a single image segment shown in figure 5.1.

single reflectance value is obtained from a linear mixture of all the intensities within a segment. The reflectance values for a segment creates a time series shown in figure 5.2. It is observed that the reflectance values in the time series undergo seasonal changes through the course of the year. □

5.3 STATE-SPACE REPRESENTATION

Numerous real world systems are approximated with an underlying process description. This process determines the output of a system which is driven by an internal state. The behaviour at time i of such a system can be predicted based on the information observed from the system at time $(i-1)$. This description of a system's internal operation is known as a state-space model. It was originally developed by control engineers [184, Ch. 3 p. 41]. A state-space model is a mathematical representation frequently used to model a system with a set of state-space variables. The state-space model uses a set of state-space variables to predict the next output of the system.

The state-space variables in most applications are a function of time; as such the use of a time

domain representation is a convenient method for analysing the state-space model of a system [184, Ch. 3 p. 41]. The current state is thus represented by a first order differential function in the time domain. The assumption thus far has been that the process function used within the state-space model and the set of state-space variables are known and that all the system's internal operations have been incorporated. This is usually not the case, as both should be estimated. This results in an erroneous prediction of the output, which leads to assessing the accuracy of the system.

Assessing the accuracy of the state-space model requires the comparison of the actual system's output to the predicted output. The output is usually observed with the addition of noise [185, Ch. 1]. The noise is contributed by several factors, which include:

1. the limited description of the process function,
2. the state-space variables that are not estimated perfectly, and
3. any unknown internal or external source of noise.

This leads to two models required to express the dynamic model: the process model and observation model. The process model is used to describe the adaptation of the state-space variables from time $(i-1)$ to time i. The state-space variables are encapsulated at time i in a state-space vector $\vec{W}_i$ as

$$\vec{W}_i = [W_{i,1} \;\; W_{i,2} \;\; \dots \;\; W_{i,S}], \tag{5.3}$$

where S denotes the number of elements in the state-space vector. The adaptation of the state-space vector is known as the prediction step. The state-space vector $\vec{W}_i$ for time i is predicted using the transition equation, which is given as

$$\vec{W}_i = \mathbf{f}(\vec{W}_{i-1}) + \vec{z}_{i-1}. \tag{5.4}$$

The relation between $\vec{W}_i$ and $\vec{W}_{i-1}$ is described by a known transition function $\mathbf{f}$. A process noise vector $\vec{z}_{i-1}$ is added owing to the incomplete description ability inherent in function $\mathbf{f}$ and/or any previous incorrect estimates of the state-space vector $\vec{W}_{i-1}$. The noise vector $\vec{z}_{i-1}$ is assumed to be a stochastic vector with a zero-mean and covariance matrix $\mathcal{Q}_{i-1}$.

The observation model is used to describe the relation between the state-space vector $\vec{W}_i$ and the actual output of the system at time i. The actual output at time i is termed the observation vector $\vec{x}_i$ and is used in the updating step. The updating step uses a measurement equation which is given as

$$\vec{x}_i = \mathbf{h}(\vec{W}_i) + \vec{v}_i. \tag{5.5}$$

The state-space vector $\vec{W}_i$ is related to the observation vector $\vec{x}_i$ by means of the known measurement function $\mathbf{h}$. The measurement function $\mathbf{h}$ and state-space vector $\vec{W}_i$ might not be perfectly estimated. This is compensated for by including an observation noise vector $\vec{v}_i$, where the noise vector $\vec{v}_i$ is a stochastic vector with zero mean and covariance matrix $\mathcal{R}_i$. Equations (5.4) and (5.5) are known as the state-space form of a linear dynamic model. The time domain approach to state-space model representation provides an iterative model that recursively processes each observation vector sequentially.

It is assumed that both the noise vectors $\vec{z}_{i-1}, \vec{z}_{i-1} \sim \mathcal{N}_u(0, \mathcal{Q}_{i-1})$, and $\vec{v}_i, \vec{v}_i \sim \mathcal{N}_u(0, \mathcal{R}_i)$, are uncorrelated and distributed by a known distribution $\mathcal{N}_u$ for all time increments. This property is expressed as

$$\begin{pmatrix} \vec{z}_{i-1} \\ \vec{v}_i \end{pmatrix} = \mathcal{N}_u \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \mathcal{Q}_{i-1} & 0 \\ 0 & \mathcal{R}_i \end{pmatrix} \right), \quad \forall i. \tag{5.6}$$

It is also assumed that the noise vectors are uncorrelated with the initial state-space vector $\vec{W}_0$, which is expressed as

$$E[\vec{W}_0 \vec{z}_{i-1}] = E[\vec{W}_0 \vec{v}_i] = 0, \quad \forall i. \tag{5.7}$$

The recursive nature of a linear dynamic model requires that a state-space vector must be adapted at each time increment i using the newest observation vector $\vec{x}_i$. This requires the derivation of a posterior probability density function of the state-space vector, given that all previous observation vectors are available [185, Ch. 1]. This is accomplished by obtaining the initial state-space vector $P(\vec{W}_i)$, after which the posterior probability density function $p(\vec{W}_i|\vec{x}_i, \vec{x}_{i-1}, \dots \vec{x}_0)$ is recursively estimated using the predict (equation (5.4)) and update (equation (5.5)) steps. The posterior probability $p(\vec{W}_i|\vec{x}_{i-1}, \vec{x}_{i-2}, \dots \vec{x}_0)$ is obtained using the Chapman-Kolmogoroff equation given as

$$p(\vec{W}_i|\vec{x}_{i-1}, \vec{x}_{i-2}, \dots \vec{x}_0) = \int p(\vec{W}_i|\vec{W}_{i-1}) p(\vec{W}_{i-1}|\vec{x}_{i-1}, \vec{x}_{i-2}, \dots \vec{x}_0) d\vec{W}_{i-1}. \tag{5.8}$$

The conditional probability density function $p(\vec{W}_i|\vec{W}_{i-1})$ is estimated using the transition equation shown in equation (5.4) and known covariance matrix $\mathcal{Q}_{i-1}$. In this prediction step the transition equation expands the current state-space probability density function. The measurement equation then uses the newest observation vector $\vec{x}_i$ to tighten the state-space probability density function [185, Ch. 1]. The state-space probability density function is updated using the observation vector $\vec{x}_i$ via Bayes' rule as

$$p(\vec{W}_i|\vec{x}_i, \vec{x}_{i-1}, \dots \vec{x}_0) = \frac{p(\vec{x}_i|\vec{W}_i)p(\vec{W}_i|\vec{x}_{i-1}, \vec{x}_{i-2}, \dots \vec{x}_0)}{p(\vec{x}_i|\vec{x}_{i-1}, \vec{x}_{i-2}, \dots \vec{x}_0)}, \tag{5.9}$$

which is expanded to

$$p(\vec{W}_i|\vec{x}_i, \vec{x}_{i-1}, \dots \vec{x}_0) = \frac{p(\vec{x}_i|\vec{W}_i)p(\vec{W}_i|\vec{x}_{i-1}, \vec{x}_{i-2}, \dots \vec{x}_0)}{\int p(\vec{x}_i|\vec{W}_i)p(\vec{W}_i|\vec{x}_{i-1}, \vec{x}_{i-2}, \dots \vec{x}_0)d\vec{W}_i}. \tag{5.10}$$

The conditional probability density function $p(\vec{x}_i|\vec{W}_i)$ is calculated using equation (5.5) and known covariance matrix $\mathcal{R}_i$. The accuracy of the state-space vector can be measured if knowledge of the posterior probability density function $p(\vec{W}_i|\vec{x}_i, \vec{x}_{i-1}, \dots \vec{x}_0)$ is available [185, Ch. 1].

5.4 KALMAN FILTER

The Kalman filter was originally developed by Rudolf Kalman in 1960 and was published in two journals [186, 187]. The Kalman filter was designed to recursively solve the state-space form of the linear dynamic model given in equations (5.4) and (5.5). The Kalman filter assumes that the transition function $\mathbf{f}$ is a known linear matrix $\mathbf{F}$ and the process noise vector $\vec{z}_{i-1}, \vec{z}_{i-1} \sim \mathcal{N}(0, \mathcal{Q}_{i-1})$, is normally distributed. This simplifies the transition equation given in equation (5.4) to

$$\vec{W}_i = \mathbf{F}\vec{W}_{i-1} + \vec{z}_{i-1}. \tag{5.11}$$

The Kalman filter also assumes that the measurement function $\mathbf{h}$ is a known linear matrix $\mathbf{H}$ and the observation noise vector $\vec{v}_i, \vec{v}_i \sim \mathcal{N}(0, \mathcal{R}_i)$, is normally distributed. This simplifies the measurement equation given in equation (5.5) to

$$\vec{x}_i = \mathbf{H}\vec{W}_i + \vec{v}_i. \tag{5.12}$$

The distributions $p(\vec{W}_i|\vec{x}_{i-1}, \dots, \vec{x}_0)$, $p(\vec{W}_{i-1}|\vec{x}_{i-1}, \dots, \vec{x}_0)$ and $p(\vec{W}_i|\vec{x}_i, \dots, \vec{x}_0)$ in equation (5.8) and equation (5.10) are assumed to be normally distributed. The posterior probability $p(\vec{W}_i|\vec{x}_{i-1}, \dots \vec{x}_0)$ is thus expressed as

$$p(\vec{W}_i|\vec{x}_{i-1}, \dots \vec{x}_0) = \sqrt{|2\pi\mathfrak{P}_{(i|i-1)}|}\exp(A_1), \tag{5.13}$$

with

$$A_1 = -\frac{1}{2}(\vec{W}_i - \hat{\vec{W}}_{(i|i-1)})^{\mathrm{T}}\mathfrak{P}^{-1}_{(i|i-1)}(\vec{W}_i - \hat{\vec{W}}_{(i|i-1)}). \tag{5.14}$$

The matrix $\mathfrak{P}_{(i|i-1)}$ denotes the covariance matrix at time i, given all the previous covariance matrices

up to and including time $(i-1)$. The vector $\vec{\hat{W}}_{(i|i-1)}$ denotes the estimate of the state-space vector $\vec{W}$ at time i, given all estimates of state-space vectors up to and including time $(i-1)$. The other posterior probability given in equation (5.8) is expressed as

$$p(\vec{W}_{i-1}|\vec{x}_{i-1},\dots\vec{x}_0) = \sqrt{|2\pi\mathfrak{P}_{(i-1|i-1)}|}\exp(A_2), \tag{5.15}$$

with

$$A_2 = -\frac{1}{2}(\vec{W}_{i-1} - \vec{\hat{W}}_{(i-1|i-1)})^{\mathrm{T}}\mathfrak{P}^{-1}_{(i-1|i-1)}(\vec{W}_{i-1} - \vec{\hat{W}}_{(i-1|i-1)}). \tag{5.16}$$

The matrix $\mathfrak{P}_{(i-1|i-1)}$ denotes the covariance matrix at time $(i-1)$, given all the previous covariance matrices up to and including time $(i-1)$. The vector $\vec{\hat{W}}_{(i-1|i-1)}$ denotes the estimate of the state-space vector $\vec{W}$ time $(i-1)$, given all the previous estimates of state-space vectors up to and including time $(i-1)$. The posterior probability given in equation (5.10) is expressed as

$$p(\vec{W}_i|\vec{x}_i,\dots\vec{x}_0) = \sqrt{|2\pi\mathfrak{P}_{(i|i)}|}\exp\left(-\frac{1}{2}(\vec{W}_i - \vec{\hat{W}}_{(i|i)})^{\mathrm{T}}\mathfrak{P}^{-1}_{(i|i)}(\vec{W}_i - \vec{\hat{W}}_{(i|i)})\right), \tag{5.17}$$

where $\mathfrak{P}_{(i|i)}$ denotes the covariance matrix at time i, given all the previous covariance matrices up to and including time i. The vector $\vec{\hat{W}}_{(i|i)}$ denotes the estimate of the state-space vector $\vec{W}$ at time i, given all estimates of state-space vectors up to and including time i.

The Kalman filter recursively estimates the probability density functions given in equations (5.13)–(5.17). The prediction parameters used in the prediction step (equation (5.4)) include the predicted state-space vector $\vec{\hat{W}}_{(i|i-1)}$ and predicted covariance matrix $\mathfrak{P}_{(i|i-1)}$. The predicted state-space vector's estimate $\vec{\hat{W}}_{(i|i-1)}$ is computed as

$$\vec{\hat{W}}_{(i|i-1)} = \mathbf{F}\vec{\hat{W}}_{(i-1|i-1)}, \tag{5.18}$$

and the predicted estimate of the covariance matrix is computed with

$$\mathfrak{P}_{(i|i-1)} = \mathcal{Q}_{i-1} + \mathbf{F}\mathfrak{P}_{(i-1|i-1)}\mathbf{F}^{\mathrm{T}}. \tag{5.19}$$

The parameters used in the updating step (equation (5.5)) include the posterior estimate of the state-space vector $\vec{\hat{W}}_{(i|i)}$ and posterior estimate of the covariance matrix $\mathfrak{P}_{(i|i)}$. These parameters require the computation of the innovation term and optimal Kalman gain. The innovation term $\mathcal{S}_i$ is computed as

$$\mathcal{S}_i = \mathbf{H}\mathfrak{P}_{(i|i-1)}\mathbf{H}^{\mathrm{T}} + \mathcal{R}_i. \tag{5.20}$$

The optimal Kalman gain $\mathfrak{K}_i$ is computed as

$$\mathfrak{K}_i = \mathfrak{P}_{(i|i-1)}\mathbf{H}^{\mathrm{T}}\mathcal{S}_i^{-1}. \tag{5.21}$$

The posterior estimate of the state-space vector $\hat{\vec{W}}_{(i|i)}$ is computed as

$$\hat{\vec{W}}_{(i|i)} = \hat{\vec{W}}_{(i|i-1)} + \mathfrak{K}_i(\vec{x}_i - \mathbf{H}\hat{\vec{W}}_{(i|i-1)}), \tag{5.22}$$

and the posterior estimate of the covariance matrix $\mathfrak{P}_{(i|i)}$ is computed as

$$\mathfrak{P}_{(i|i)} = \mathfrak{P}_{(i|i-1)} - \mathfrak{K}_i\mathcal{S}_i\mathfrak{K}_i^{\mathrm{T}}. \tag{5.23}$$

If the process function is precise and the initial estimates of $\hat{\vec{W}}_{(0|0)}$ and $\mathfrak{P}_{(0|0)}$ are accurate, then the following five properties will hold. The first two properties, which are relevant to the state-space vector's estimate, are

$$E[\vec{W}_i - \hat{\vec{W}}_{(i|i)}] = E[\vec{W}_i - \hat{\vec{W}}_{(i|i-1)}] = 0, \tag{5.24}$$

$$E[\vec{x}_i - \mathbf{H}\hat{\vec{W}}_{(i|i-1)}] = 0. \tag{5.25}$$

The last three properties hold a relation to the covariance matrices, which accurately reflect the estimated covariance as

$$\mathfrak{P}_{(i|i)} = \mathrm{cov}(\vec{W}_i - \hat{\vec{W}}_{(i|i)}), \tag{5.26}$$

$$\mathfrak{P}_{(i|i-1)} = \mathrm{cov}(\vec{W}_i - \hat{\vec{W}}_{(i|i-1)}), \tag{5.27}$$

$$\mathcal{S}_i = \mathrm{cov}(\vec{x}_i - \mathbf{H}\hat{\vec{W}}_{(i|i-1)}). \tag{5.28}$$

The performance of the Kalman filter is usually inhibited by the poor estimation of the observation noise's covariance matrix $\mathcal{R}_i$ and the process noise's covariance matrix $\mathcal{Q}_{i-1}$. The Kalman filter is unable to compute the mean and covariance of the Gaussian posterior probability $p(\vec{W}_i|\vec{x}_i, \vec{x}_{i-1}, \ldots \vec{x}_0)$ accurately if poor initial estimates are made of the observation and process noise's covariance matrices.

5.5 EXTENDED KALMAN FILTER

The EKF is the non-linear extension of the standard Kalman filter in estimation theory. The EKF has been considered to be the de facto standard in the theory of non-linear state estimate, navigation systems and global positioning system (GPS) [188].

The EKF is similar to the standard Kalman filter as a state-space vector $\vec{W}_i$ is estimated at each time increment i. The state-space vector $\vec{W}_i$ is estimated at time i recursively by using the set of observation vectors $\{\vec{x}_i, \vec{x}_{i-1}, \ldots, \vec{x}_0\}$. The state-space model's equations are reformulated for the EKF in this section. The transition equation in equation (5.11) is rewritten as

$$\vec{W}_i = \mathbf{f}(\vec{W}_{i-1}) + \vec{z}_{i-1}. \tag{5.29}$$

The transition function $\mathbf{f}$ is a non-linear function, and the process noise vector $\vec{z}_{i-1}, \vec{z}_{i-1} \sim \mathcal{N}(0, \mathcal{Q}_{i-1})$, is assumed to be normally distributed. The measurement equation in equation (5.12) is rewritten as

$$\vec{x}_i = \mathbf{h}(\vec{W}_i) + \vec{v}_i. \tag{5.30}$$

The measurement function $\mathbf{h}$ is a non-linear function and the observation noise vector $\vec{v}_i, \vec{v}_i \sim \mathcal{N}(0, \mathcal{R}_i)$ is assumed to be normally distributed. The idea behind the EKF is that the non-linear transition function $\mathbf{f}$ and non-linear measurement function $\mathbf{h}$ can be sufficiently described using local linearisation of the two functions.

The posterior probability density function $p(\vec{W}_i|\vec{x}_i, \ldots, \vec{x}_0)$ is approximated by means of a Gaussian distribution, which implies that equations (5.13)–(5.17) described in the Kalman filter section (section 5.4) still hold. Prediction parameters and updating parameters are reformulated to take into account the non-linear transition and measurement functions. The predicted state-space vector's estimate $\hat{\vec{W}}_{(i|i-1)}$ is expressed as

$$\hat{\vec{W}}_{(i|i-1)} = \mathbf{f}(\hat{\vec{W}}_{(i-1|i-1)}), \tag{5.31}$$

where $\mathbf{f}$ denotes the non-linear transition function. The predicted estimate of the covariance matrix $\mathfrak{P}_{(i|i-1)}$ is expressed as

$$\mathfrak{P}_{(i|i-1)} = \mathcal{Q}_{i-1} + \mathbf{F}_{\text{est}} \mathfrak{P}_{(i-1|i-1)} \mathbf{F}_{\text{est}}^{\text{T}}. \tag{5.32}$$

The matrix $\mathbf{F}_{\text{est}}$ is the local linearisation of the non-linear transition function $\mathbf{f}$. The matrix $\mathbf{F}_{\text{est}}$ is defined as the Jacobian evaluated at $\hat{\vec{W}}_{(i-1|i-1)}$ as [185, Ch. 2]

$$\mathbf{F}_{\text{est}} = \left\| \left[\frac{\partial}{\partial W_{i,1}} \; \cdots \; \frac{\partial}{\partial W_{i,S}} \right] \mathbf{f}^{\text{T}}(\vec{W}_i) \right\|_{\vec{W}_i = \vec{\hat{W}}_{(i-1|i-1)}}. \tag{5.33}$$

In the case of the updating parameters, the posterior estimate of the state-space vector $\vec{\hat{W}}_{(i|i)}$ is expressed as

$$\vec{\hat{W}}_{(i|i)} = \vec{\hat{W}}_{(i|i-1)} + \mathfrak{K}_i(\vec{x}_i - \mathbf{h}(\vec{\hat{W}}_{(i|i-1)})). \tag{5.34}$$

The function $\mathbf{h}$ denotes the non-linear measurement function and $\mathfrak{K}_i$ denotes the EKF's optimal Kalman gain given as

$$\mathfrak{K}_i = \mathfrak{P}_{(i|i-1)}\mathbf{H}_{\text{est}}^{\text{T}}\mathcal{S}_i^{-1}. \tag{5.35}$$

The matrix $\mathbf{H}_{\text{est}}$ is the local linearisation of the non-linear measurement function $\mathbf{h}$. The matrix $\mathbf{H}_{\text{est}}$ is defined as the Jacobian evaluated at $\vec{\hat{W}}_{(i|i-1)}$ as [185, Ch. 2]

$$\mathbf{H}_{\text{est}} = \left\| \left[\frac{\partial}{\partial W_{i,1}} \; \cdots \; \frac{\partial}{\partial W_{i,S}} \right] \mathbf{h}^{\text{T}}(\vec{W}_i) \right\|_{\vec{W}_i = \vec{\hat{W}}_{(i|i-1)}}. \tag{5.36}$$

The innovation term for the EKF is defined as

$$\mathcal{S}_i = \mathbf{H}_{\text{est}}\mathfrak{P}_{(i|i-1)}\mathbf{H}_{\text{est}}^{\text{T}} + \mathcal{R}_i. \tag{5.37}$$

The posterior estimate of the covariance matrix $\mathfrak{P}_{(i|i)}$ is expressed as

$$\mathfrak{P}_{(i|i)} = \mathfrak{P}_{(i|i-1)} - \mathfrak{K}_i\mathcal{S}_i\mathfrak{K}_i^{\text{T}}. \tag{5.38}$$

Land cover example: The time series example given in figure 5.1 produces a time series which is shown in figure 5.2. Kleynhans *et al.* proposed a triply modulated cosine function for the process function [30]. The triply modulated cosine function is expressed as

$$\vec{x}_i = \mu_i + \alpha_i \cos(2\pi f_{\text{samp}} i + \theta_i). \tag{5.39}$$

The variable i denotes the time index and f_{samp} denotes the temporal sampling rate of the image acquisitions. The cosine function is characterised by three variables: μ_i, α_i and θ_i. These three variables form the state-space vector, which is defined as

$$\vec{W}_i = [W_{i,1}\; W_{i,2}\; W_{i,3}] = [W_{i,\mu}\; W_{i,\alpha}\; W_{i,\theta}]. \tag{5.40}$$

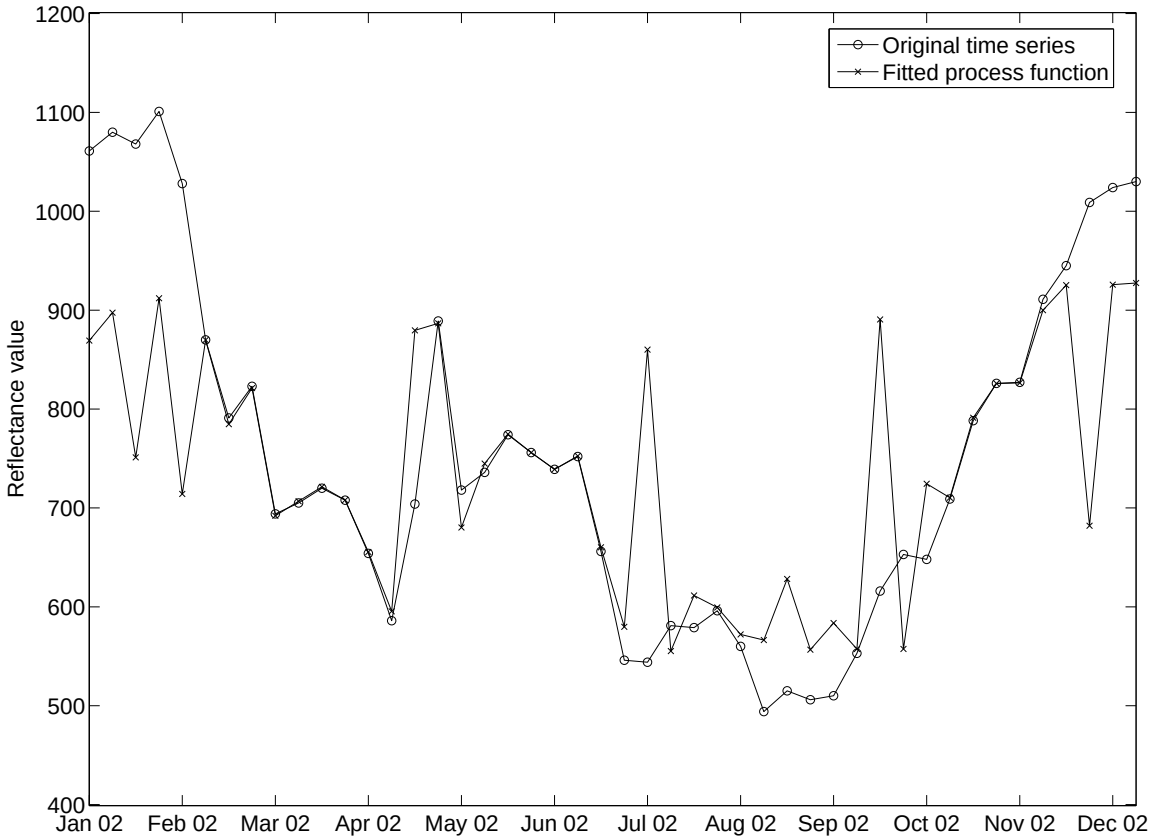


FIGURE 5.3: The Extended Kalman filter estimates the parameters of the state-space vector $\vec{W}_i$ to fit the triply modulated cosine function onto the time series shown in figure 5.2. The estimated state-space vector is used to create a fitted process function to measure the accuracy of the fit.

The triply modulated cosine function is a non-linear function and the EKF was proposed to solve the state-space model. It is assumed that the state-space vector remains constant from one time increment to the next. This reduces the transition equation given in equation (5.29) to

$$\vec{W}_i = \vec{W}_{i-1} + \vec{z}_{i-1}. \tag{5.41}$$

The measurement equation shown in equation (5.30) is defined for this example as

$$\vec{x}_i = \mathbf{h}(\vec{W}_i) + \vec{v}_i, \tag{5.42}$$

where the measurement function $\mathbf{h}$ is the triply modulated cosine function given in equation (5.39) as

$$\mathbf{h}(\vec{W}_i) = W_{i,\mu} + W_{i,\alpha}\cos(2\pi f_{\text{samp}} i + W_{i,\theta}). \tag{5.43}$$

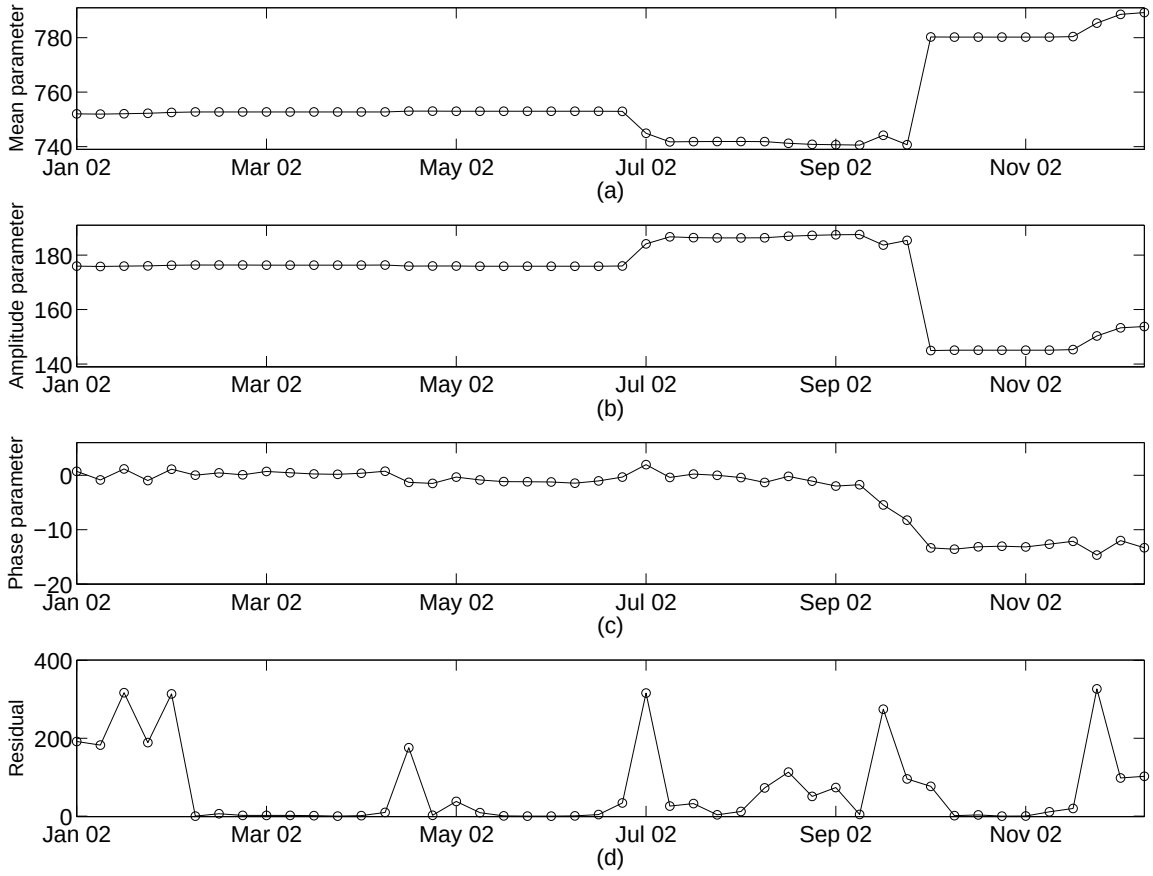


FIGURE 5.4: The Extended Kalman filter estimates the parameters in the state-space vector $\vec{W}_i$. Figure (a) shows the mean parameter μ_i estimates. Figure (b) shows the amplitude parameters α_i estimates. Figure (c) shows the phase parameter θ_i estimates. Figure (d) shows the absolute error in tracking the output of the system.

It should be noted that the measurement function produces a vector with a single dimension. Thus for this example, equation (5.42) is further reduced to a single output as

$$x_i = \mathbf{h}(\vec{W}_i) + v_i. \tag{5.44}$$

The predicted state-space vector's estimate $\hat{\vec{W}}_{(i|i-1)}$ shown in equation (5.31) is rewritten by substituting the transition function with the identity matrix for the example as

$$\hat{\vec{W}}_{(i|i-1)} = \mathbf{f}(\hat{\vec{W}}_{(i-1|i-1)}) = \hat{\vec{W}}_{(i-1|i-1)}. \tag{5.45}$$

The matrix $\mathbf{F}_{\text{est}}$ is an identity matrix, which simplifies the predicted estimate for the covariance matrix $\mathfrak{P}_{(i|i-1)}$ shown in equation (5.32) to

$$\mathfrak{P}_{(i|i-1)} = \mathcal{Q}_{i-1} + \mathbf{F}_{\text{est}}\mathfrak{P}_{(i-1|i-1)}\mathbf{F}_{\text{est}}^{\text{T}} = \mathcal{Q}_{i-1} + \mathfrak{P}_{(i-1|i-1)}. \tag{5.46}$$

The posterior estimate of the state-space vector $\hat{\vec{W}}_{(i|i)}$ shown in equation (5.34) is expressed for this example as

$$\begin{aligned}
\hat{\vec{W}}_{(i|i)} &= \hat{\vec{W}}_{(i|i-1)} + \mathfrak{K}_i(\vec{x}_i - \mathbf{h}(\hat{\vec{W}}_{(i|i-1)})) \\
&= \hat{\vec{W}}_{(i|i-1)} + \mathfrak{K}_i(\vec{x}_i - \mathbf{H}_{\text{est}}(\hat{\vec{W}}_{(i|i-1)})) \\
&= \hat{\vec{W}}_{(i|i-1)} + \mathfrak{K}_i\left(\vec{x}_i - \left\|\left[\frac{\partial \mathbf{h}^{\text{T}}(\vec{W}_i)}{\partial W_{i,\mu}} \; \frac{\partial \mathbf{h}^{\text{T}}(\vec{W}_i)}{\partial W_{i,\alpha}} \; \frac{\partial \mathbf{h}^{\text{T}}(\vec{W}_i)}{\partial W_{i,\theta}}\right]\right\|_{\vec{W}_i = \hat{\vec{W}}_{(i|i-1)}}\right),
\end{aligned} \tag{5.47}$$

with

$$\frac{\partial \mathbf{h}(\vec{W}_i)}{\partial W_{i,\mu}} = 1 \tag{5.48}$$

$$\frac{\partial \mathbf{h}(\vec{W}_i)}{\partial W_{i,\alpha}} = \cos(2\pi f_{\text{samp}} i + \hat{\vec{W}}_{(i|i-1),\theta}) \tag{5.49}$$

$$\begin{aligned}
\frac{\partial \mathbf{h}(\vec{W}_i)}{\partial W_{i,\theta}} &= -\hat{\vec{W}}_{(i|i-1),\alpha}\Big[\sin(2\pi f_{\text{samp}} i)\cos(\hat{\vec{W}}_{(i|i-1),\theta}) + \\
&\qquad \cos(2\pi f_{\text{samp}} i)\sin(\hat{\vec{W}}_{(i|i-1),\theta})\Big].
\end{aligned} \tag{5.50}$$

The time series shown in figure 5.2 is fitted with the triply modulated cosine function by estimating a state-space vector $\vec{W}_i$ for each time increment. The estimated output of the EKF, using the newest available observation vector at time i, is plotted with the actual observation vector $\vec{x}_i$ in figure 5.3. It is observed that the EKF requires an initial number of observations before the state-space vector starts to stabilise. The stabilised state-space vector corresponds to a more accurate tracking of the actual observations.

The progressive estimation of the state-space vectors is shown in figure 5.4. Figure 5.4(a) illustrates the estimation of the mean parameter μ_i (the first element in the state-space vector denoted by $W_{i,\mu}$). Figure 5.4(b) illustrates the estimation of the amplitude parameter α_i (the second element in the state-space vector denoted by $W_{i,\alpha}$). Figure 5.4(c) illustrates the estimation of the phase parameter θ_i (the third element in the state-space vector denoted by $W_{i,\theta}$). The absolute error in the tracking of the output is illustrated in figure 5.4(d).

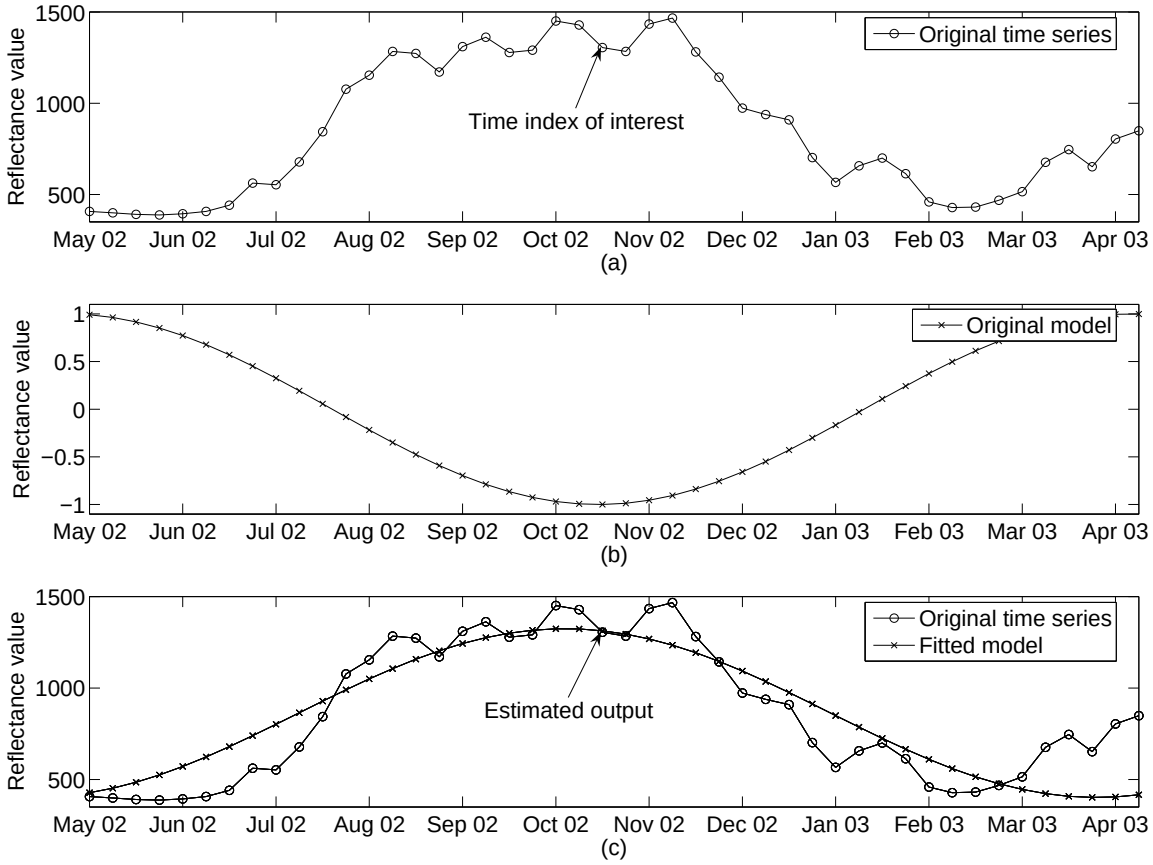


FIGURE 5.5: Least squares estimates the parameter vector $\vec{W}_i$ to fit the model onto the time series.

5.6 LEAST SQUARES MODEL FITTING

The least squares method was first discovered by Carl Friedrich Gauss in 1795 and was later published by the French mathematician Legendre in 1805. The least squares is a method used to fit the triply modulated cosine model with a parameter vector $\vec{W}_i$. It estimates the parameter vector by evaluating the fit of the model to the actual observation vector. The parameter vector in this context can be viewed as the state-space vector defined in the state-space model and the model can be viewed as the process function (section 5.3).

The least squares is a linear regression method, which uses a model $\mathbf{h}$ to predict a set of dependent parameter vectors $\{\vec{W}_i\}$ from a set of independent observation vectors $\{\vec{x}_i\}$. The least squares' goal is to find a parameter vector $\vec{W}_i$ that will minimise the sum of squares between the observation vectors $\vec{x}_i$ and the model's estimated output vector $\hat{\vec{x}}_i$. The sum of squares is computed as a summation of the error residuals to measure the performance and is expressed as

$$\mathcal{E}_{\text{LS}} = \sum_{i=1}^{\mathcal{I}} (\vec{x}_i - \hat{\vec{x}}_i)^2 = \sum_{i=1}^{\mathcal{I}} (\vec{x}_i - \mathbf{h}(\vec{x}_i, \vec{W}_i))^2. \tag{5.51}$$

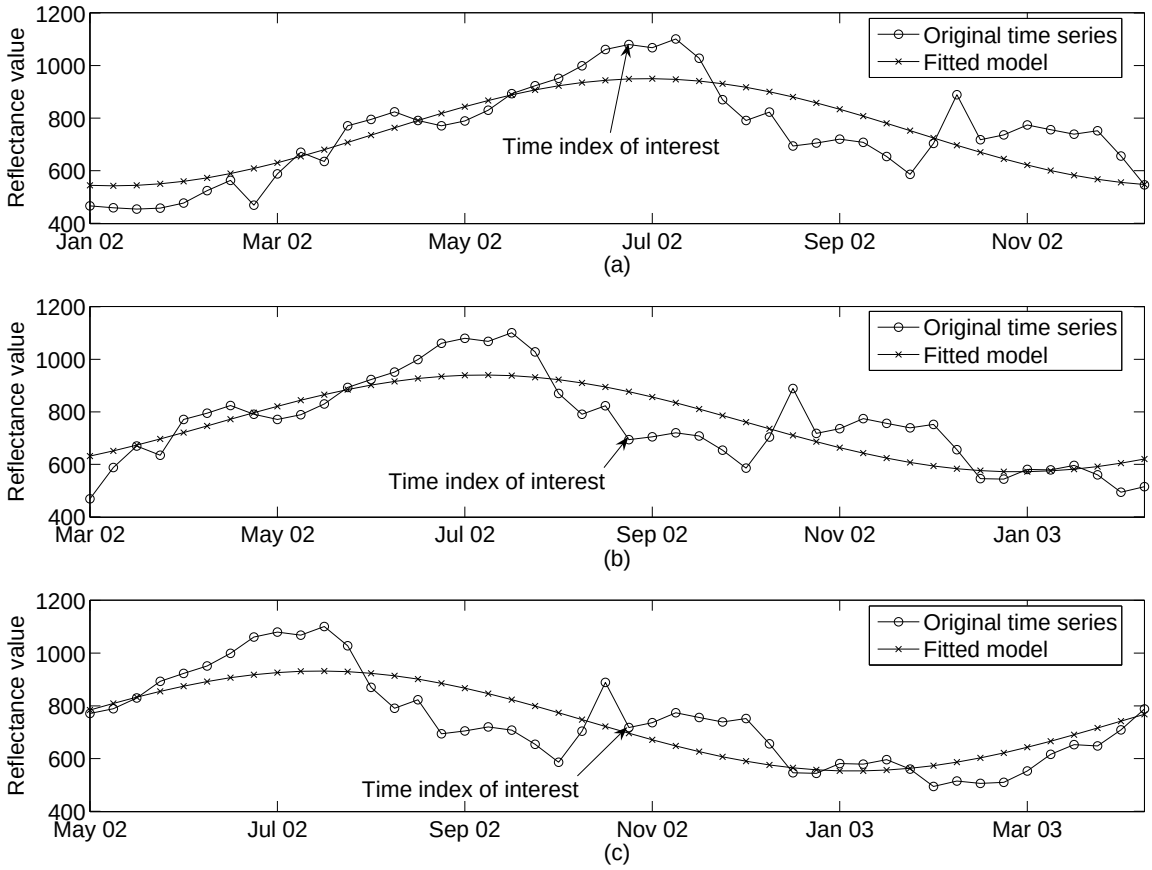


FIGURE 5.6: Least squares estimates the parameter vector $\vec{W}_i$ by shifting the model over the time series.

The variable $\mathcal{E}_{\mathrm{LS}}$ denotes the sum of squares and $\mathbf{h}$ denotes the model. The sum of squares can be minimised using standard approaches, which evaluate the partial derivatives. The partial derivative of the sum of squares is solved as

$$\frac{d\mathcal{E}_{\mathrm{LS}}}{d\vec{W}_i} = 2\sum_{j=1}^{\mathcal{I}}(\vec{x}_j - \hat{\vec{x}}_j)\frac{d(\vec{x}_j - \hat{\vec{x}}_j)}{d\vec{W}_i} = 0, \qquad \forall i. \tag{5.52}$$

Several variations of the least squares exist; the most popular method is the ordinary least squares (OLS) algorithm. The OLS assumes the observation noise vector $\vec{v}_i$ is normally distributed and the model $\mathbf{h}$ is linear.

The least squares is considered optimal when a set of criteria is met in the estimates of the parameter vector. These criteria are:

1. The observation vectors are randomly sampled from a well defined data set.

2. The underlying structure within the data set is linear.

3. The difference between the observation vector $\vec{x}_i$ and the fitted model has an expected zero mean.

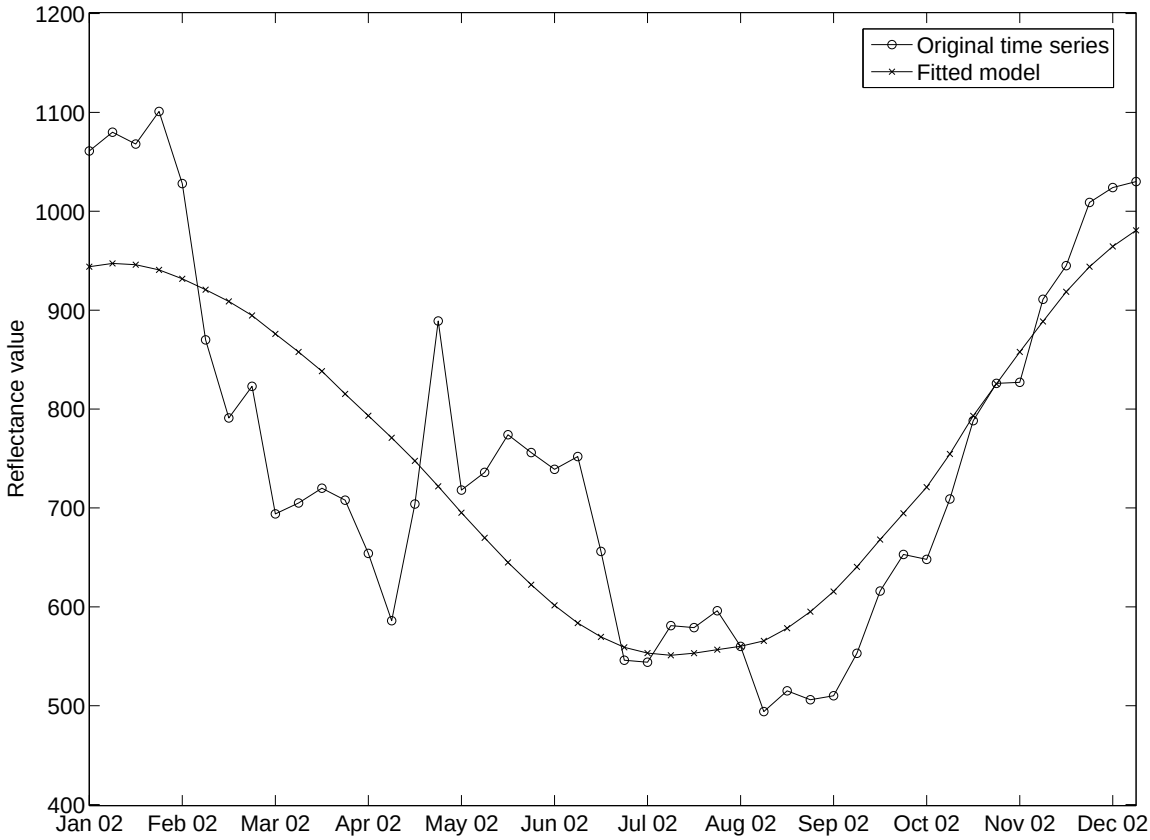


FIGURE 5.7: Least squares estimates the parameter vector $\vec{W}_i$ to fit triply modulated cosine model onto a time series.

4. The parameter vector's variables are linearly independent from each other.

5. The difference between the observation vector $\vec{x}_i$ and the fitted model is normally distributed and uncorrelated to the parameter vector.

In addition to the five criteria stated, if the Gauss-Markov condition also holds; then the OLS estimates are considered to be equivalent to the maximum likelihood estimates of the parameter vectors. More sophisticated adaptations have been made to the OLS and the most frequently used of these are: the weighted least squares, alternating least squares and partial least squares.

The OLS can be extended to include the field of non-linear models. The drawback is that the standard approach of evaluating the derivative of a non-linear model in equation (5.52) is not always possible. This is because the derivatives of $d(\vec{x}_j - \hat{\vec{x}}_j)/d\vec{W}_i$ are functions which are dependent on both the observation vectors $\{\vec{x}_i\}$ and the parameter vectors $\{\vec{W}_i\}$.

This changes the least squares from a closed-form solution for the linear case to a non closed-form solution for the non-linear case. This requires that the estimation of the set of parameter vectors $\{\vec{W}_i\}$ is derived using an analytical iterative algorithm. The algorithm iterates through the parameter vector's

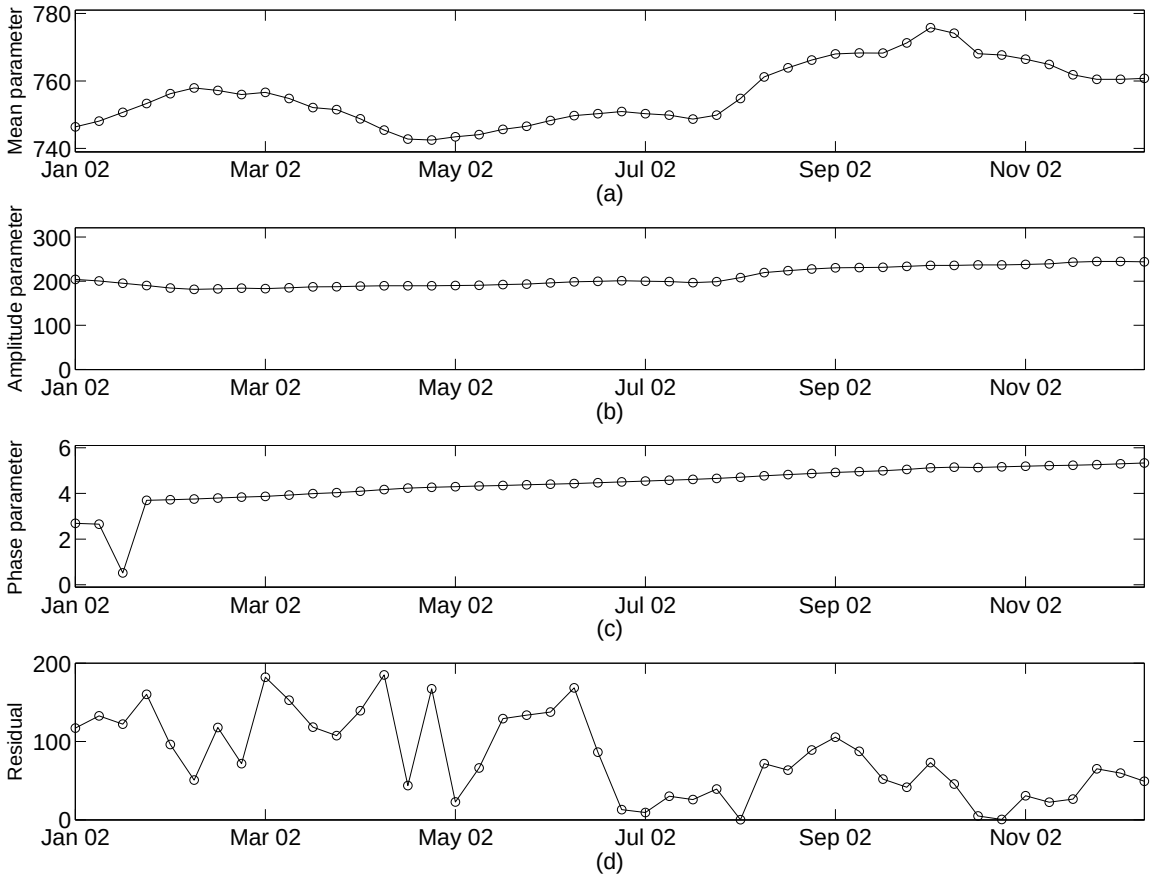


FIGURE 5.8: Least squares estimates the parameter vector $\vec{W}_i$ to fit triply modulated cosine model onto a time series.

space using the derivative of the sum of squares $\mathcal{E}_{\text{LS}}$ at each epoch. The gradient descent algorithm is a popular iterative method used in this case.

Land cover example: In this example the least squares predicts the set of parameter vectors for the time series shown in figure 5.2. The problem lies in the fact that the least squares requires a set of observation vectors $\{\vec{x}_i\}$ to estimate a single parameter vector $\vec{W}_i$. The lowest number of observation vectors required to estimate the parameter vector is $(|\vec{W}_i| + 1)$.

This concept is illustrated in figure 5.5 by using a set of observation vectors the length of a single year. In figure 5.5(a) the time series in figure 5.2 is shown with a time index of interest. The parameter vector $\vec{W}_i$ for observation vector $\vec{x}_i$ is estimated using the set $\{\vec{x}_{i-N}, \vec{x}_{i-N+1}, \ldots, \vec{x}_{i+N-1}, \vec{x}_{i+N}\}$ of observation vectors. The variable N is chosen to encapsulate the entire period of the model shown in figure 5.5(b). The parameter vector $\vec{W}_i$ is then determined using the least squares to minimise the sum of squares to produce the fitted model shown in figure 5.5(c).

The next step is to estimate a parameter vector $\vec{W}_i, \forall i$. This is accomplished by moving the

model across the time index. The parameter vector $\vec{W}_{i+c}$ for observation vector $\vec{x}_{i+c}$ is estimated using the set $\{\vec{x}_{i-N+c}, \vec{x}_{i-N+c+1}, \ldots, \vec{x}_{i+N+c-1}, \vec{x}_{i+N+c}\}$. This iterative approach to moving the model is shown in three different figures in figure 5.6.

After shifting through the entire time series, the predicted output of the least squares is plotted, along with the actual observation vectors in figure 5.7.

The progressive estimation of the parameter vectors is shown in figure 5.8. Figure 5.8(a) illustrates the estimation of the model's mean parameter μ_i. Figure 5.8(b) illustrates the estimation of the model's amplitude parameter α_i. Figure 5.8(c) illustrates the estimation of the model's phase parameter θ_i. The absolute error in tracking of the output is illustrated in figure 5.8(d). □

5.7 M-ESTIMATE MODEL FITTING

Various attempts have been made to create robust statistical estimators, which are used to fit models. M-estimates rely on the maximum likelihood approach to estimate the parameters of a particular statistical model. An M-estimator is generally defined as a zero of the estimating function, while the estimating function is usually the derivative of a statistical function of interest. The advantage of a M-estimator is that it does not assume that the residuals are normally distributed. M-estimators attempt to minimise the mean absolute deviation in the residuals for a given distribution using a maximum likelihood approach.

The assessment of different distributions in the M-estimator allow for different weighting functions to be associated with outliers. Normally distributed residuals usually associate greater weights to outliers when compared to a Lorentzian distribution of residuals [189, Ch. 15]. This deviant behaviour in relative weighting points in a model makes it difficult to apply standard gradient descent. The Nelder-Mead method is thus the chosen optimisation method, as it only requires function evaluations and not the derivatives [189, Ch. 15].

The Nelder-Mead algorithm was first proposed by John Nelder and Roger Mead in 1965 [190]. The Nelder-Mead algorithm is a non-linear method which estimates the parameter vector $\vec{W}_i$ for a particular model. The Nelder-Mead algorithm is a well-defined numerical method that operates on a twice differentiable, unimodal, multi-dimensional function. The method makes use of a direct search by evaluating a function at the vertices of a simplex. A N-simplex is a N-dimensional polytope which is the convex hull of (N+1) vertices. The algorithm then iteratively moves and scales the simplex's vertices through the set of dimensions in search of the minimum. It continually attempts to improve the evaluated function until a predefined bound is reached.

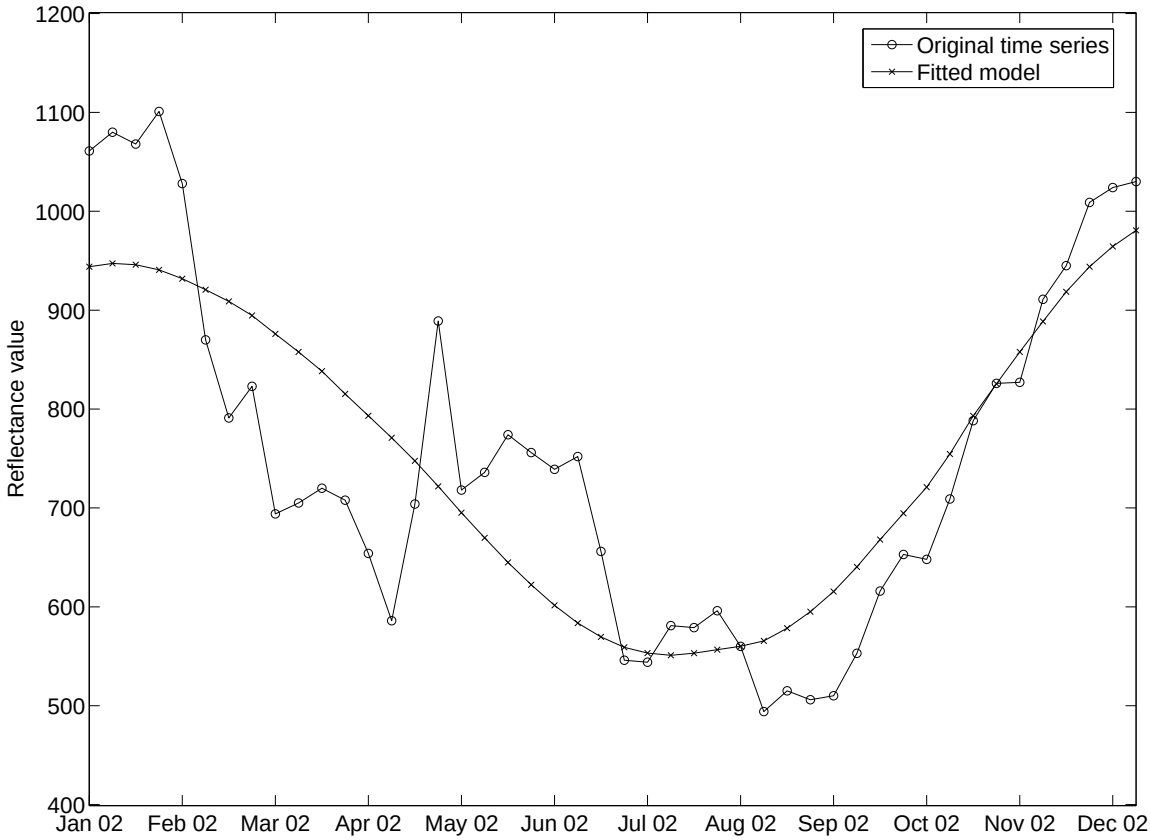


FIGURE 5.9: M-estimator estimates the parameter vector $\vec{W}_i$ to fit the triply modulated cosine model onto a time series.

Each epoch requires the execution of six steps to compute the new position of the simplex. The algorithm in summary starts with initialising the vertices of the simplex. It then iteratively rejects and replaces the worst performing vertex point with a new vertex point. This process of setting new vertex points creates a sequence of new N-simplexes. The initialisation with a small initial N-simplex converges rapidly to a local minimum, while a large N-simplex becomes trapped in non-stationary points in the vector space.

Land cover example: In this example the M-estimator predicts a set of parameter vectors for the time series shown in figure 5.2. The same problem exists for the M-estimator, as for the least squares, when estimating the sequence of parameter vectors. The parameter vector $\vec{W}_i$ for observation vector $\vec{x}_i$ is estimated using the set $\{\vec{x}_{i-N}, \vec{x}_{i-N+1}, \ldots, \vec{x}_{i+N-1}, \vec{x}_{i+N}\}$ of observation vectors. This is rectified by shifting the model through all the time indices. The initial estimate of the M-estimator is contained in a certain parameter space by using the mean and standard deviation of the time series as the initial parameter vector for the model. The previous parameter vector $\vec{W}_{i-1}$ is then used to initialise the M-estimator when determining the current parameter vector $\vec{W}_i$.

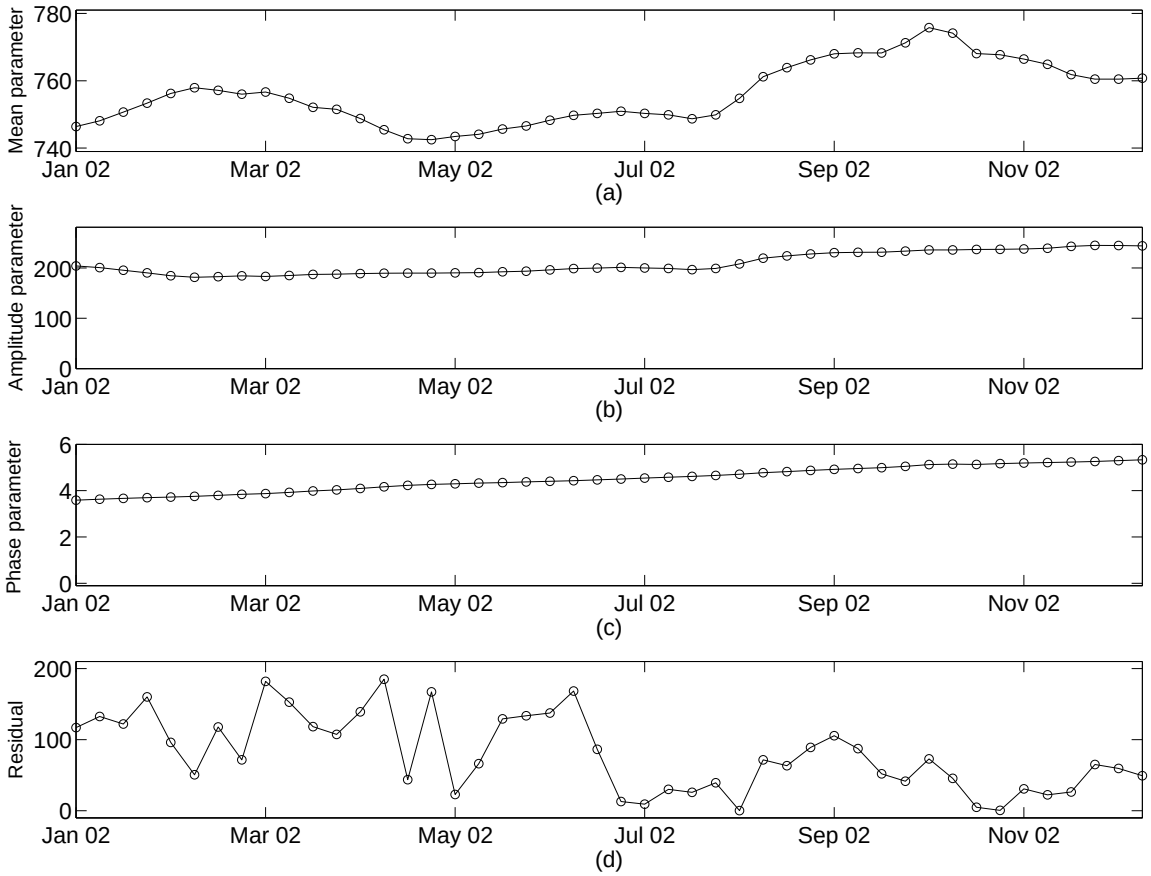


FIGURE 5.10: M-estimator estimates the parameter vector $\vec{W}_i$ to fit the triply modulated cosine model onto a time series.

The predicted output of the M-estimator is plotted with the actual observation vectors $\vec{x}_i$ in figure 5.9.

The progressive estimation of the parameter vectors are shown in figure 5.10. Figure 5.10(a) illustrates the estimation of the model's mean parameter μ_i. Figure 5.10(b) illustrates the estimation of the model's amplitude parameter α_i. Figure 5.10(c) illustrates the estimation of the model's phase parameter θ_i. The absolute error in the tracking of the output is illustrated in figure 5.10(d). □

5.8 FOURIER TRANSFORM

The Fourier transform of a discrete time series is a representation of the sequence in terms of the complex exponential sequence $\{e^{j2\pi fi}\}$, where f is the frequency variable. The Fourier transform representation of a time series, if it exists, is unique and the original time series can be recovered by applying an inverse Fourier transform [115, Ch. 3].

Let $\mathbf{x}$, $\mathbf{x} = [x_1\ x_2\ \ldots\ x_{\mathcal{I}}]$, denote the time series and let $\mathcal{I} \to \infty$, then the Fourier transform $\mathcal{X}(e^{j2\pi f})$ is defined as

$$\mathcal{X}(e^{j2\pi f}) = \sum_{i=-\infty}^{\infty} x_{(\mathcal{I}/2)} e^{j2\pi f i}. \tag{5.53}$$

The Fourier transform $\mathcal{X}(e^{j2\pi f})$ is a complex function and is written in rectangular form as

$$\mathcal{X}(e^{j2\pi f}) = \mathcal{X}_{\text{real}}(e^{j2\pi f}) + j\mathcal{X}_{\text{imag}}(e^{j2\pi f}), \tag{5.54}$$

where $\mathcal{X}_{\text{real}}(e^{j2\pi f})$ denotes the real part and $\mathcal{X}_{\text{imag}}(e^{j2\pi f})$ denotes the imaginary part of $\mathcal{X}(e^{j2\pi f})$. The components of the rectangular form are expressed as

$$\mathcal{X}_{\text{real}}(e^{j2\pi f}) = |\mathcal{X}(e^{j2\pi f})| \cos\theta_{\mathcal{X}}, \tag{5.55}$$

$$\mathcal{X}_{\text{imag}}(e^{j2\pi f}) = |\mathcal{X}(e^{j2\pi f})| \sin\theta_{\mathcal{X}}. \tag{5.56}$$

The quantity $|\mathcal{X}(e^{j2\pi f})|$ denotes the magnitude function of the Fourier transform. The quantity $\theta_{\mathcal{X}}$ denotes the phase function, which is given as

$$\theta_{\mathcal{X}} = \arctan\left(\frac{\mathcal{X}_{\text{imag}}(e^{j2\pi f})}{\mathcal{X}_{\text{real}}(e^{j2\pi f})}\right). \tag{5.57}$$

In the case of a finite length time series $\mathbf{x}$, $\mathbf{x} = [x_1\ x_2\ \ldots\ x_{\mathcal{I}}]$, $\mathcal{I} \in \mathbb{N}$, $\mathcal{I} < \infty$, there is a simpler relation between the time series and its corresponding Fourier transform $\mathcal{X}(e^{j2\pi f})$ [115, Ch. 3]. For a time series $\mathbf{x}$ of length $\mathcal{I}$, only $\mathcal{I}$ values of $\mathcal{X}(e^{j2\pi f})$ at $\mathcal{I}$ distinct harmonic functions at frequency points, $0 \leq f \leq \mathcal{I}$, are sufficient to construct the unique time series $\mathbf{x}$. This leads to the concept of a second transform domain representation that operates on a finite length time series [115, Ch. 3].

This second transform is known as the discrete Fourier transform (DFT). The relation between a finite length time series $\mathbf{x}$, $\mathbf{x} = [x_1\ x_2\ \ldots\ x_{\mathcal{I}}]$, and its corresponding Fourier transform $\mathcal{X}(e^{j2\pi f})$ is obtained by uniformly sampling $\mathcal{X}(e^{j2\pi f})$ on the frequency domain between $0 \leq f \leq 1$ at increments of $f = i/\mathcal{I}$, $0 \leq i \leq (\mathcal{I}-1)$. The DFT is computed by sampling equation (5.53) uniformly as

$$\mathcal{X}_i = \mathcal{X}(e^{j2\pi f})\Big|_{f=i/\mathcal{I}} = \sum_{n=0}^{\mathcal{I}-1} x_n e^{j2\pi i n/\mathcal{I}}, \quad 0 \leq i \leq (\mathcal{I}-1). \tag{5.58}$$

The inverse discrete Fourier transform (IDFT) is given by

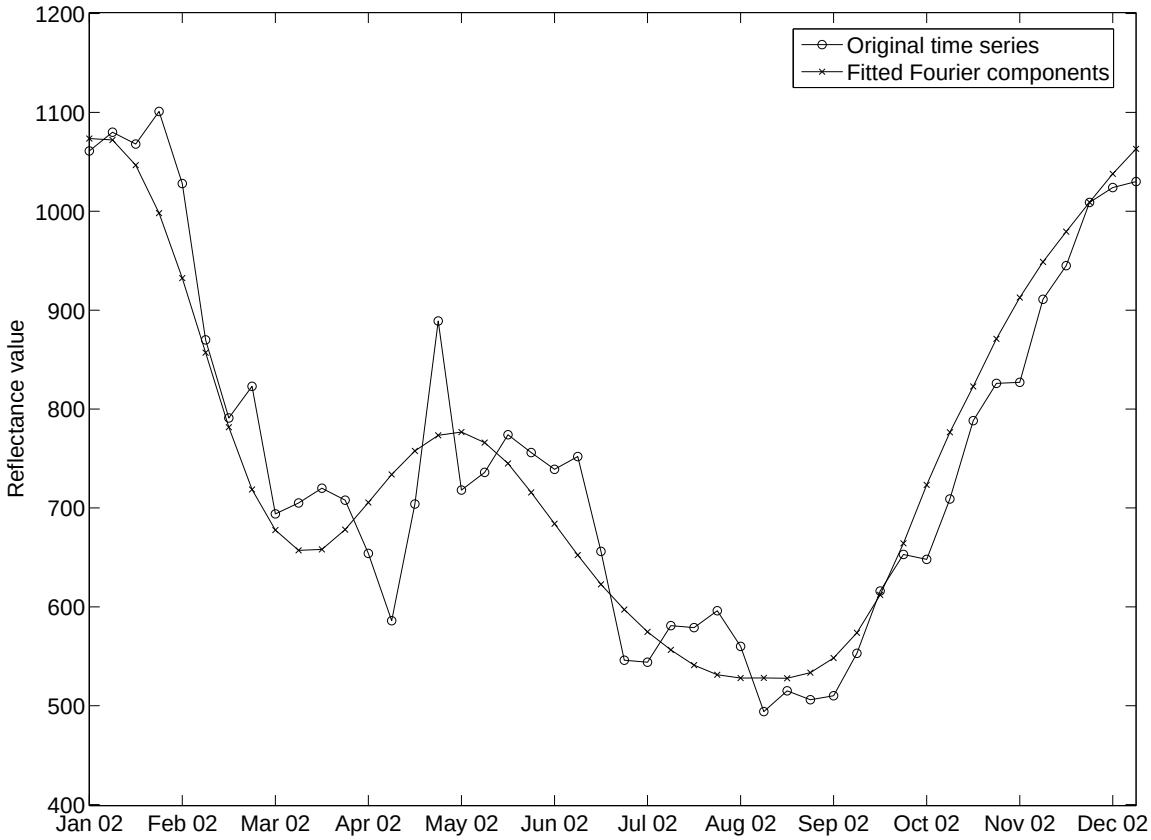


FIGURE 5.11: Fast Fourier transform (FFT) estimates the parameters of the vector $\vec{W}_i$ to fit multiple harmonics onto time series $\mathbf{x}$.

$$x_n = \sum_{i=0}^{\mathcal{I}-1} \mathcal{X}_i e^{-j2\pi in/\mathcal{I}}, \quad 0 \leq n \leq (\mathcal{I}-1). \tag{5.59}$$

The computation of the DFT and IDFT requires $\mathcal{O}(\mathcal{I}^2)$ complex multiplications and $\mathcal{O}(\mathcal{I}^2 - \mathcal{I})$ complex additions. A fast Fourier transform (FFT) refers to an algorithm that has been developed to reduce the computational complexity of computing the DFT to about $\mathcal{O}(\mathcal{I}(\log_2 \mathcal{I}))$ operations. As there is no loss in precision in using these fast computing algorithms, they will be used throughout this thesis when referring to the DFT of a time series. Similarly, an inverse fast Fourier transform (IFFT) algorithm has been developed to compute the IDFT efficiently.

The FFT function is denoted by $\mathfrak{F}$ and is mathematically computed as

$$\mathcal{X} = \mathfrak{F}(\mathbf{x}). \tag{5.60}$$

The sequence $\mathcal{X}$ is the DFT of the time series $\mathbf{x}$. The time series $\mathbf{x}$ is a process in the time domain and the value of $\mathbf{x}$ is dependent on the corresponding time index i. The DFT $\mathcal{X}$, on the other hand, is a process in the frequency domain by which the process is defined by the amplitude $|\mathbf{x}_f|$ and phase $\angle \mathbf{x}_f$

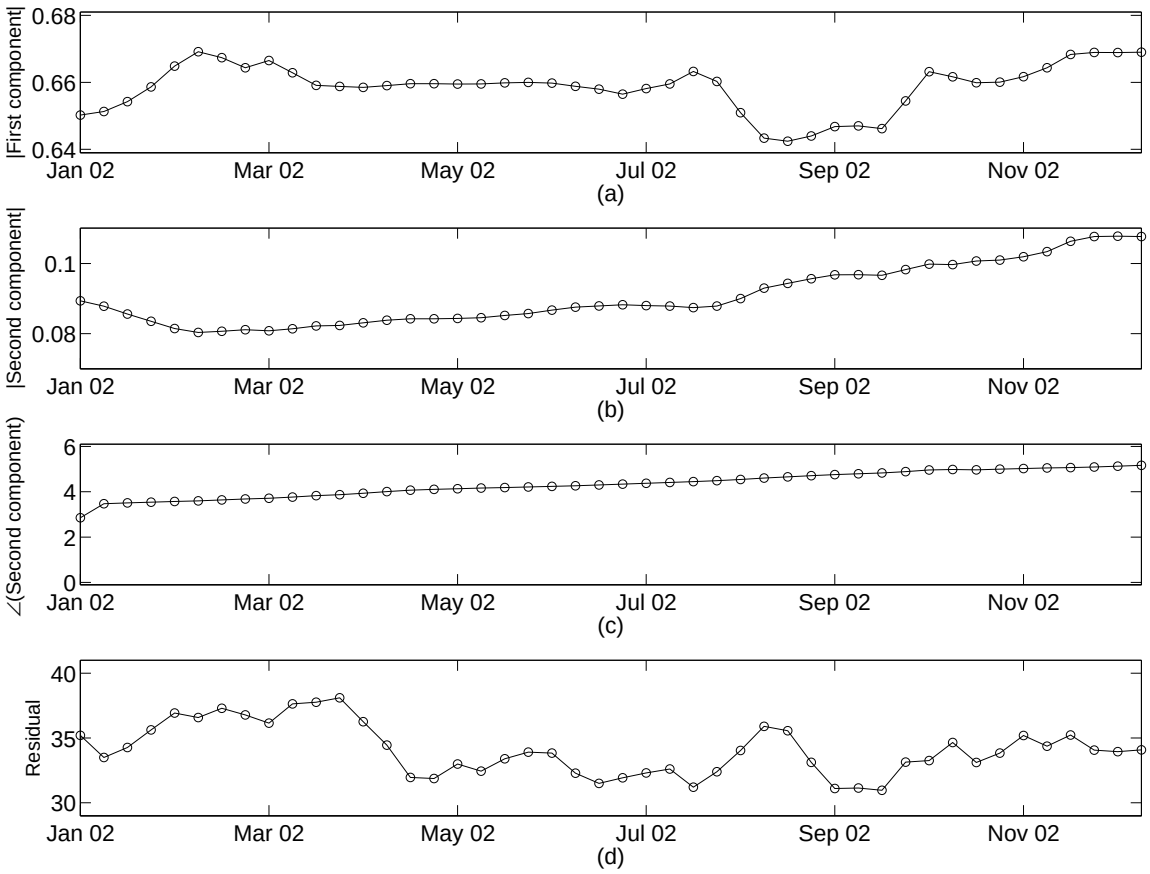


FIGURE 5.12: Fast Fourier transform (FFT) estimates the parameters of the vector $\vec{W}_i$ to fit multiple harmonics onto time series $\mathbf{x}$.

of harmonic frequency samples f, $f \in \{-\infty, \infty\}$.

The inverse Fourier transform is denoted by $\mathfrak{F}^{-1}$ and is mathematically computed as

$$\mathbf{x} = \mathfrak{F}^{-1}(\mathcal{X}). \tag{5.61}$$

The conversion to the frequency domain allows the analysis of periodic (such as seasonal) effects and trends within the time series $\mathbf{x}$.

Land cover example: In this example the fast Fourier transform is used to predict a set of Fourier components for the time series shown in figure 5.2.

The Fourier components are stored in a vector $\vec{W}_i$ for observation vector $\vec{x}_i$ and are estimated using the set $\{\vec{x}_{i-N}, \vec{x}_{i-N+1}, \ldots, \vec{x}_{i+N-1}, \vec{x}_{i+N}\}$ of observation vectors. The variable N is chosen to capture enough energy in each harmonic function of interest. This happens to be the entire process function of a complete phenological cycle of one year.

A set of harmonic functions is stored in the state-space model as

$$\vec{W}_i = [W_{i,1}\ W_{i,2}\ W_{i,3}] = [W_{i,\mu}\ W_{i,\alpha}\ W_{i,\theta}] = [|\mathcal{X}_1|\ \ 2|\mathcal{X}_2|\ \ \angle(\mathcal{X}_2)]. \tag{5.62}$$

The next step is to estimate a vector $\vec{W}_i, \forall i$. This is accomplished by moving a window across the time index. The vector $\vec{W}_{i+c}$ for observation vector $\vec{x}_{i+c}$ is estimated using the set $\{\vec{x}_{i-N+c}, \vec{x}_{i-N+c+1}, \ldots, \vec{x}_{i+N+c-1}, \vec{x}_{i+N+c}\}$. This iterative approach moves the window of the DFT similar to the least squares and M-estimator. The predicted output of the Fourier components is plotted along with the actual observation vectors in figure 5.11.

The progressive estimation of the vectors is shown in figure 5.12. Figure 5.12(a) illustrates the estimation of the magnitude of the first frequency component in $\mathcal{X}$. Figure 5.12(b) illustrates the estimation of the magnitude of the second frequency component in $\mathcal{X}$. Figure 5.12(c) illustrates the phase of the second frequency component $\mathcal{X}$. The absolute error in tracking of the output is illustrated in figure 5.12(d). □

5.9 SUMMARY

In this chapter, four different feature extraction methods were investigated. The feature extraction methods are all based on the same principle of fitting a cosine model to the time series. The first three methods; EKF, least squares model fitting and M-estimator model fitting, are regression approaches, which attempt to estimate the mean, amplitude, and phase component of the cosine function. All three features are comparable among the three regression methods. The Fourier transform method is similar to the other three methods, except for the fact that a complex vector is estimated, which contains the combined power of both a cosine and sine function. The feature vectors extracted using these methods will be used by machine learning methods to determine the corresponding class labels.

CHAPTER SIX

SEASONAL FOURIER FEATURES

6.1 OVERVIEW

In this chapter, the concept of extracting meaningful features from a time series is investigated. The chapter starts by defining the difference between the concept of whole clustering and subsequence clustering. It continues by exploring a fundamental pitfall inherent when using subsequence clustering to analyse time series. This is motivated at the hand of an experiment presented by Keogh [29] and a worked-out visual example. A key feature extraction method, that will extract the Seasonal Fourier Features (SFF) is presented in section 6.4, which will overcome the disadvantage of using subsequence clustering. The chapter concludes by defining how this SFF is used in a post-classification change detection algorithm to detect change in time series.

6.2 TIME SERIES ANALYSIS

A time series is a sequence of measurements, typically recorded at successive time intervals [191]. Time series have a distinct natural temporal ordering. This induces a high correlation between measurements taken at a shorter interval from a system, when compared to measurements taken at a longer interval from the same system. Time series analysis comprises methods for analysing time series to extract statistics and underlying characteristics. Several different types of analysis can be applied to time series and are categorised as: exploration, description, prediction and forecasting.

1. Exploration provides in-depth information on serial dependence and any cyclic behaviour patterns within time series. The time series can also be graphically examined to observe any salient characteristics.

2. Description provides information of underlying structures hidden within the time series. Algorithms were developed to decompose time series into several components to examine any hidden trends, seasonality, slow and fast variations, cyclic irregularities and anomalies.

3. Prediction provides information on any near future event in the time series and can be used as feedback to control a system's behaviour that is providing the data points of the time series.

4. Forecasting uses statistical models to generate variations of the time series to observe alternative possible events that might occur in the future.

Clustering is the most frequently used exploration tool in data mining algorithms. The vast quantities of important information typically hidden in time series have attracted substantial attention [29]. Clustering is used in many algorithms as either: rule discovery [192], indexing [193], classification [194], prediction [195], or anomaly detection [196]. Clustering of time series is broadly divided into two categories: *whole clustering* and *subsequence clustering* [29].

Whole clustering: Whole clustering is similar to the conventional clustering of discrete objects. Each time series is viewed as an individual discrete object and is thus clustered into groups with other time series. □

Subsequence clustering: Subsequence clustering is when multiple individual time series (subsequences) are extracted with a sliding window from a single time series. Let $\mathbf{x}$, $\mathbf{x} = [\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_{\mathcal{I}}]$, denote a time series of length $\mathcal{I}$. A subsequence extracted from time series $\mathbf{x}$ is given as

$$\mathbf{x}_p = \left(\vec{x}_p,\ \vec{x}_{p+1},\ \ldots,\ \vec{x}_{p+Q-1} \right), \tag{6.1}$$

for $1 \leq p \leq \mathcal{I}$-$Q$+1, where Q is the length of the subsequence. The sequential extraction of subsequences in equation (6.1) is achieved by using a temporal sliding window that has a length of Q and position p, $p \in \mathbb{N}_0$, that is incremented with a natural number $\mathbb{N}$ to extract sequential subsequences $\mathbf{x}_p$ from $\mathbf{x}$. This set of subsequences are clustered into groups, similar to how *whole clustering* clusters an entire time series. □

6.3 MEANINGLESS ANALYSIS

Recently the data mining community's attention was drawn to a fundamental limitation in the clustering of subsequences that are extracted with a sliding window from a time series [29]; the sliding window

causes the clustering algorithms to create meaningless results. This is due to the fact that clusters extracted from the subsequences are forced to obey a certain constraint that is pathologically unlikely to be satisfied by any data set. The term meaningless originates from the effect of creating random clusters when applying a clustering algorithm to such subsequences [29].

It should be noted that it is well understood that clustering in a high-dimensional feature space usually produces useless results if proper design considerations are not followed [197, 198]. For example, the K-nearest neighbour algorithm produces fewer useful clusters in higher dimensions. This is because the ratio between the nearest neighbour and the average neighbour distance rapidly converges to one in higher dimensions. However, the analysis on time series usually results in high dimensionality, which typically has a low intrinsic dimensionality [199]. This is not the limitation that will be discussed in this chapter.

Keogh and Lin [29] made a surprising claim, which called into question dozens of published results. The problem identified lies in the way the features are extracted from the sliding window when presented to the clustering algorithm. This claim is supported by the following experiment.

Experiment presented in [29]: The variability in the clusters formed will be tested using the same clustering design considerations and methodology on different data sets containing time series. It is shown that any partitional or hierarchical clustering algorithm would suffice in this experiment, and under this assumption the K-means was used for its robustness in forming reliable clusters. The K-means clustering algorithm forms clusters, which are used to define a set of functions.

Let $\vartheta(a) = \{\vartheta^1(a), \vartheta^2(a), \ldots, \vartheta^K(a)\}$ denote the cluster centroids derived with the K-means algorithm from the first data set.

Let $\vartheta(b) = \{\vartheta^1(b), \vartheta^2(b), \ldots, \vartheta^K(b)\}$ denote the cluster centroids derived with the K-means algorithm from the second data set.

Let $D_{\text{ed}}(\vartheta^i, \vartheta^j)$ denote the Euclidean distance between two cluster centroids. The distance metric $D_{\text{ed}}(\vartheta^i, \vartheta^j)$ determines the shortest possible distance for an one-to-one mapping of two sets of centroids $\vartheta(a)$ and $\vartheta(b)$.

The difference between the two sets of cluster centroids is defined as

$$D_{\mathcal{M}}(\vartheta(a), \vartheta(b)) = \sum_{i=1}^{K} \min_{j} [D_{\text{ed}}(\vartheta^i(a), \vartheta^j(b))]. \tag{6.2}$$

The consistency of a clustering algorithm to form similar sets of clusters is measured if the first data set used to find cluster centroids $\vartheta(a)$ and the second data set used to find cluster centroids

$\vartheta(b)$ is the same data set. A more important measurement is to determine the similarity between the centroids when they are not the same data set.

Keogh and Lin [29] proposed a clustering meaningfulness index as

$$C_{\mathcal{M}}(\vartheta(a), \vartheta(b)) = \frac{D_{\mathcal{M}}(\vartheta(a), \vartheta(a))}{D_{\mathcal{M}}(\vartheta(a), \vartheta(b))}. \tag{6.3}$$

The clustering meaningfulness index measures the similarity between two data sets' clusters despite the fact that two different data sets are used.

Intuitively, if proper clustering design considerations were applied the numerator in equation (6.3) should converge to zero. In contrast to this statement, if the data sets are unrelated, then the denominator should tend to a large number. This in effect naturally makes the clustering meaningfulness index $C_{\mathcal{M}}(\vartheta(a), \vartheta(b)) \to 0$.

The results produced in this experiment were unexpected. When a random walk data set was compared to a stock market data set, the clustering meaningfulness index averaged between 0.5 and 1 when *subsequence clustering* was applied to the time series. This means that if clustering was performed on the stock market data set, the centroids derived could be re-used for the random walk data set and the difference in clustering results could not be observed.

The same was not true when *whole clustering* was used on these two data sets. The clustering meaningfulness index converged to zero when the stock market data set and random walk data set were clustered using a *whole clustering* approach. Several additional experiments were conducted in [29] to motivate this behaviour as a property of the sliding window. □

The sliding window causes the clustering algorithm to create meaningless results, as it forms sine wave cluster centroids regardless of the data set, which clearly makes it impossible to distinguish one data set's clusters from another. Furthermore, the sine waves within the cluster centroids are always out of phase with each other by exactly $1/K$ period [29]. The inability to produce meaningful cluster centroids revealed a new question: how do the cluster centroids obtain this special structure [29]? In this section a visual example is shown to illustrate why the clustering algorithm produces meaningless results.

Visual example: Assume a triply modulated cosine function, which is given as

$$x_i = \mu_i + \alpha_i \cos(2\pi f i + \theta_i), \tag{6.4}$$

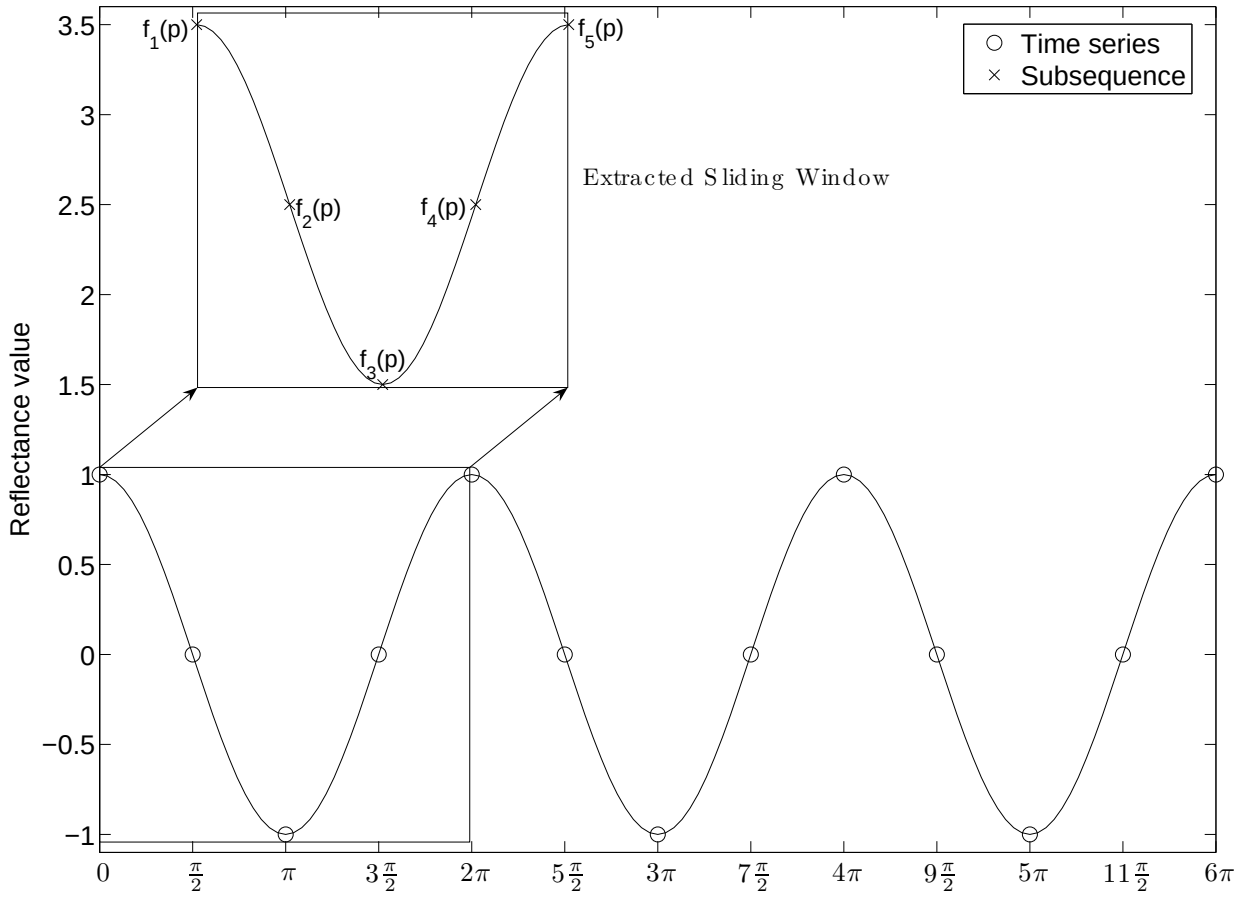


FIGURE 6.1: The five feature points, separated by a period of $\frac{\pi}{2}$, are extracted from the sliding window, and is denoted by the set $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$.

where the mean μ_i, amplitude α_i, frequency f, and phase θ_i are fixed for all time increments in this example. A visual plot of this triply modulated cosine function is shown in figure 6.1. A sliding window is placed on the time series with features extracted from the window at multiples of $\frac{\pi}{2}$ of the period.

The five features are extracted at interval $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi\}$ from the sliding window and are denoted by $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$. The position of the sliding window is denoted by the variable p, $p \in \mathbb{N}_0$. This is mathematically expressed as

$$\begin{aligned} \mathbf{x}_p &= \Big(f_1(p),\ f_2(p),\ f_3(p),\ f_4(p),\ f_5(p)\Big) \\ &= \Big(x_{p\pi/2},\ x_{(p+1)\pi/2},\ x_{(p+2)\pi/2},\ x_{(p+3)\pi/2},\ x_{(p+4)\pi/2}\Big). \end{aligned} \tag{6.5}$$

The initial extracted features, $p = 0$, are extracted from the sliding window and are expressed as

$$\begin{aligned}\mathbf{x}_0 &= \Big(f_1(0),\ f_2(0),\ f_3(0),\ f_4(0),\ f_5(0)\Big) \\ &= \Big(x_0,\ x_{\pi/2},\ x_{\pi},\ x_{3\pi/2},\ x_{2\pi}\Big).\end{aligned} \tag{6.6}$$

It should be noted that the length of the sliding window in this example is set at Q=5. The position of the sliding window is incremented by 1 (equivalent shift of $\frac{\pi}{2}$) to evaluate a new range of observations in the time series (figure 6.2), which is expressed as

$$\begin{aligned}\mathbf{x}_1 &= \Big(f_1(1),\ f_2(1),\ f_3(1),\ f_4(1),\ f_5(1)\Big) \\ &= \Big(x_{\pi/2},\ x_{\pi},\ x_{3\pi/2},\ x_{2\pi},\ x_{5\pi/2}\Big).\end{aligned} \tag{6.7}$$

As the position is incremented, the five features extracted from the time series in set $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$ are presented to a clustering method. To understand the claim of Keogh [29], focus will only be placed on the first feature $f_1(p)$ without loss of generality. The feature extracted at point $f_1(p)$ for the sliding window at position p is expressed as

$$f_1(p) = x_{p\pi/2}. \tag{6.8}$$

Equation (6.8) is used to create a time series $\mathbf{f}_1$ for all the values of $f_1(p)$ for all positions p of the sliding window and is expressed as

$$\mathbf{f}_1 = \Big(x_0,\ x_{\pi/2},\ x_{\pi},\ \ldots x_{(\mathcal{I}-Q)\pi/2}\Big). \tag{6.9}$$

The values of the triply modulated cosine function is substituted into $\mathbf{f}_1$ as

$$\mathbf{f}_1 = \Big(\alpha_i,\ \mu_i,\ -\alpha_i,\ \mu_i, \alpha_i\ \ldots \alpha_i\Big). \tag{6.10}$$

This shows that inadvertly all the features are sequentially presented to every dimension of the feature vector. The fundamental problem becomes intuitive, as every feature dimension is sequentially attempting to learn the same thing. This is better illustrated by tabulating the set of features $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$. Table 6.1 shows what each feature point measures as a function of the sliding window increments. □

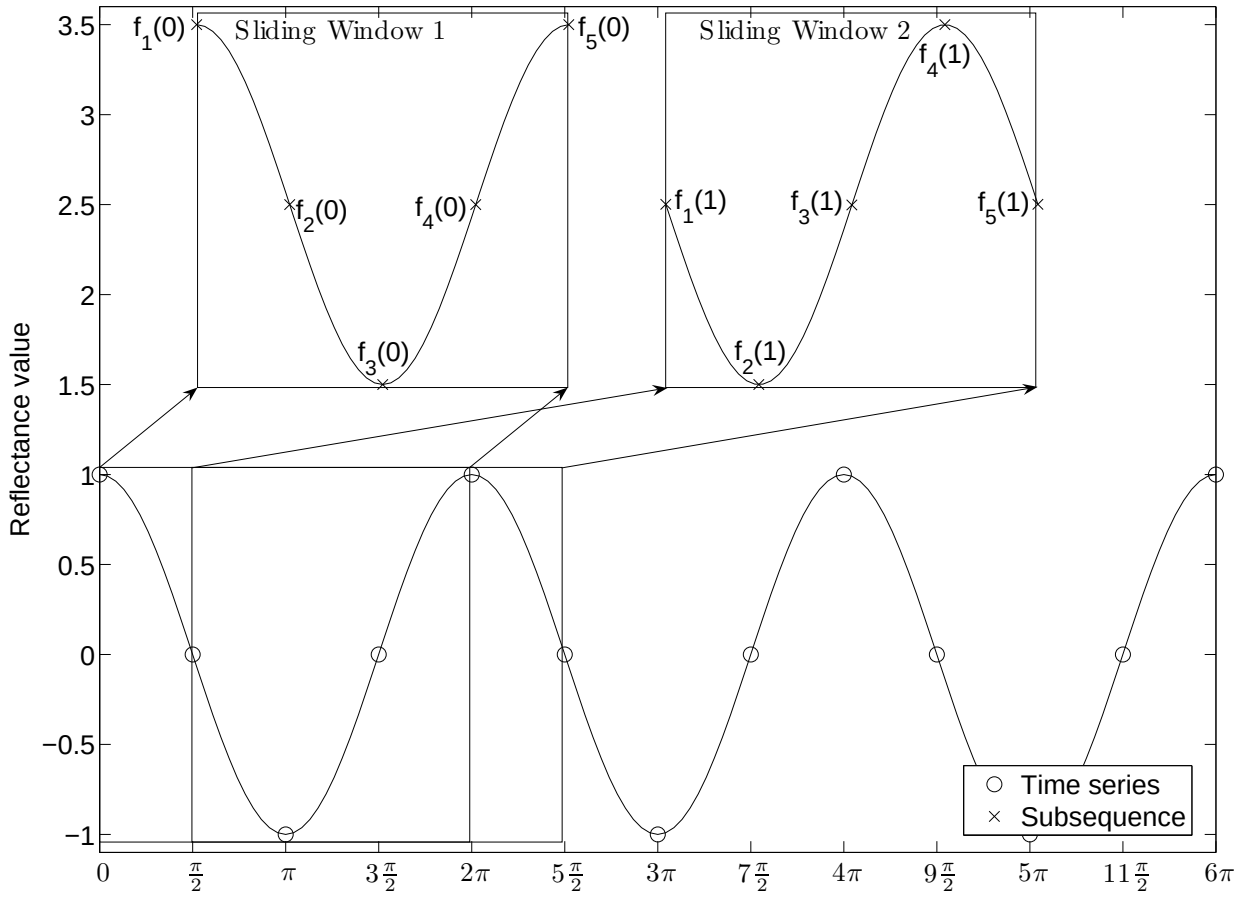


FIGURE 6.2: Two sets of five feature points $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$, are separated by a period of $\frac{\pi}{2}$, are shown to be extracted by two sliding windows.

Table 6.1: The sequence of features extracted as a function of the sliding window's position from figure 6.2.

Sliding window position	**Time increment**	**Feature points**				
		f_1	f_2	f_3	f_4	f_5
0	0	α_i	μ_i	-α_i	μ_i	α_i
1	$\frac{\pi}{2}$	μ_i	-α_i	μ_i	α_i	μ_i
2	π	-α_i	μ_i	α_i	μ_i	-α_i
3	$\frac{3\pi}{2}$	μ_i	α_i	μ_i	-α_i	μ_i
4	2π	α_i	μ_i	-α_i	μ_i	α_i

The intuition behind understanding this problem is to imagine an arbitrary data point somewhere in the time series which enters the sliding window and the contribution this data point makes to the overall mean of the sliding window. As the sliding window passes by, the data point first appears as the rightmost value in the window and then sequentially appears exactly once in every possible location within the sliding window. Thus all feature points will present the same information at different times and different dimensions to the clustering algorithm. This is equivalent to only presenting one data

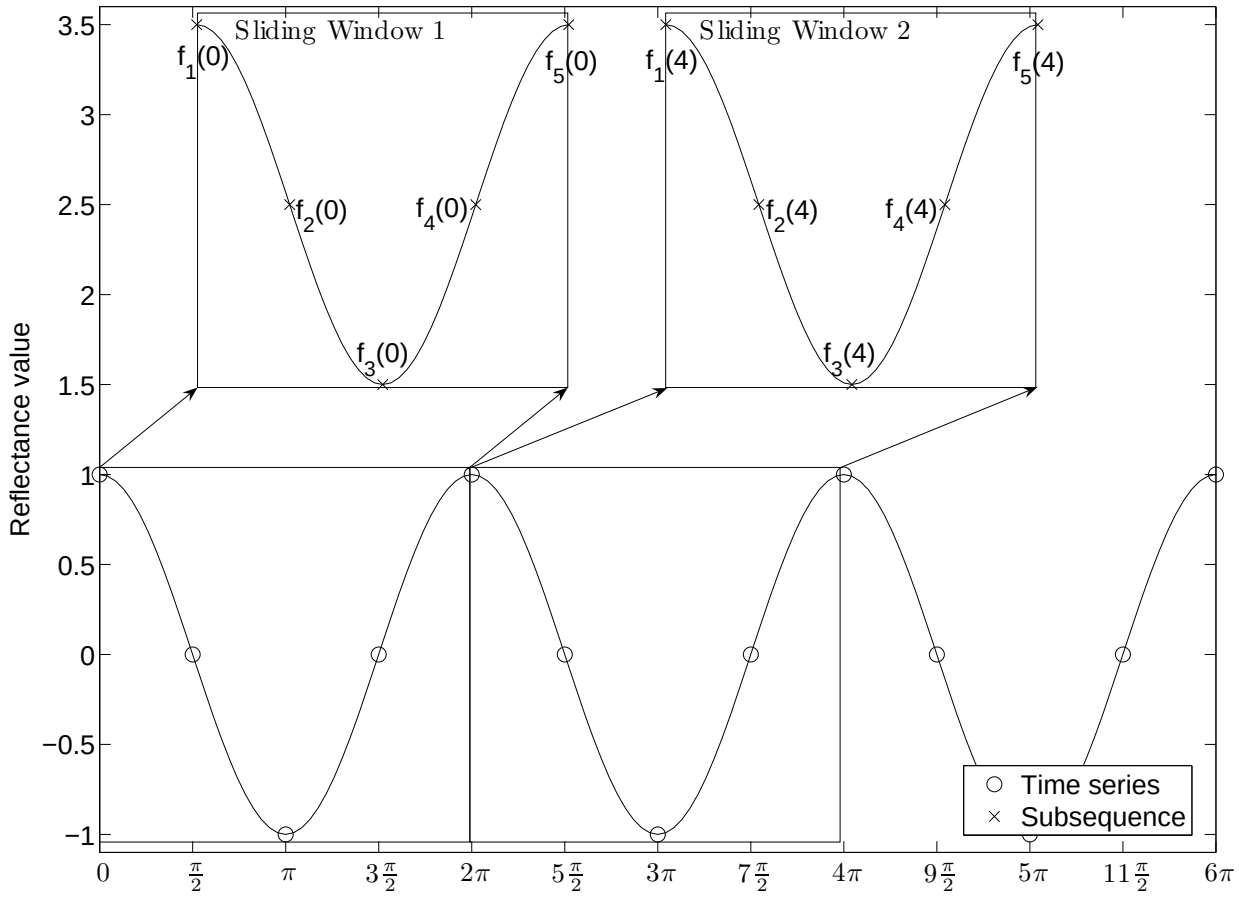


FIGURE 6.3: Two sets of five feature points $\{f_1(p), f_2(p), f_3(p), f_4(p), f_5(p)\}$, are separated by a period of 2π, are shown to be extracted by two sliding windows.

point to a clustering algorithm and sequentially shifting through the time series.

Several ideas were formulated on how to create meaningful clusters [29]. The first idea was to increment the position of the sliding window by more than the length of the sliding window. This does not solve the problem, as the *subsequence clustering* becomes a *whole clustering* application. The second idea considered by Keogh and Lin [29] was to set the number of clusters much higher than the true number of clusters within the data set. Empirically this only worked if the number of clusters was set impractically high. The authors concluded that there is no simple solution to the problem of *subsequence clustering*.

Proposition 6.3.1 *A tentative solution was presented by Keogh and Lin [29] to find meaningful clusters using subsequence clustering. The example is in essence whole clustering, but it does emphasise an interesting property. The tentative solution proposes a single time series with a repetitive pattern, as shown in figure 6.3. The sliding window is shifted by exactly one period of the repetitive pattern within the time series. The new features are extracted and presented to the clustering algorithm. The solution becomes more intuitive if the features are tabulated in sequence of extraction.*

Table 6.2: The sequence of features extracted as a function of the sliding window's position from figure 6.3.

Sliding window position	**Time increment**	**Feature points**				
		f_1	f_2	f_3	f_4	f_5
0	0	α_i	μ_i	-α_i	μ_i	α_i
1	2π	α_i	μ_i	-α_i	μ_i	α_i
2	4π	α_i	μ_i	-α_i	μ_i	α_i
3	6π	α_i	μ_i	-α_i	μ_i	α_i
4	8π	α_i	μ_i	-α_i	μ_i	α_i

Table 6.2 now shows that each feature point is acquiring a single property of the time series. Through feature selection it becomes apparent that features f_3–f_5 can be discarded. This tentative solution provides meaningful clusters when the sliding window position p is incremented by the period of the repetitive pattern.

This however becomes a whole clustering *solution if the sliding window's position is incremented by more than its length. This results in analysing non-overlapping sliding windows.* □

Since remote sensing time series data have a strong periodic component due to the seasonal vegetation dynamics, the extracted sequential time series could potentially be processed to yield usable features. A feature extraction method is proposed in the next section that will reduce the feature space's dimensionality and removes the restriction of the tentative solution proposed in [29]. The removal of the restriction on the sliding window's position p will enable effective subsequence clustering that does not suffer from the afore-mentioned limitations.

6.4 MEANINGFUL CLUSTERING

In this section a method is shown that will create usable features from a subsequence $\mathbf{x}_p$ extracted from a MODIS MCD43A4 time series data set. The fixed acquisition rate of the MODIS product and the seasonality of the vegetation in the study area make for an annual periodic signal $\mathbf{x}$ that has a phase offset that is correlated with rainfall seasonality and vegetation phenology. The FFT [200] of $\mathbf{x}_p$ is computed, which decomposes the time sequence's values into components of different frequencies with phase offsets. This is often referred to as the frequency (Fourier) spectrum of the time series. Because the time series $\mathbf{x}_p$ is annually periodic, this would translate into frequency components in the frequency spectrum that have fixed positions with varying phase offsets. The varying phases limits the shifting of the sliding window's position p to exactly a periodic cycle [29], except if the clustering algorithm can cater for the varying phases.

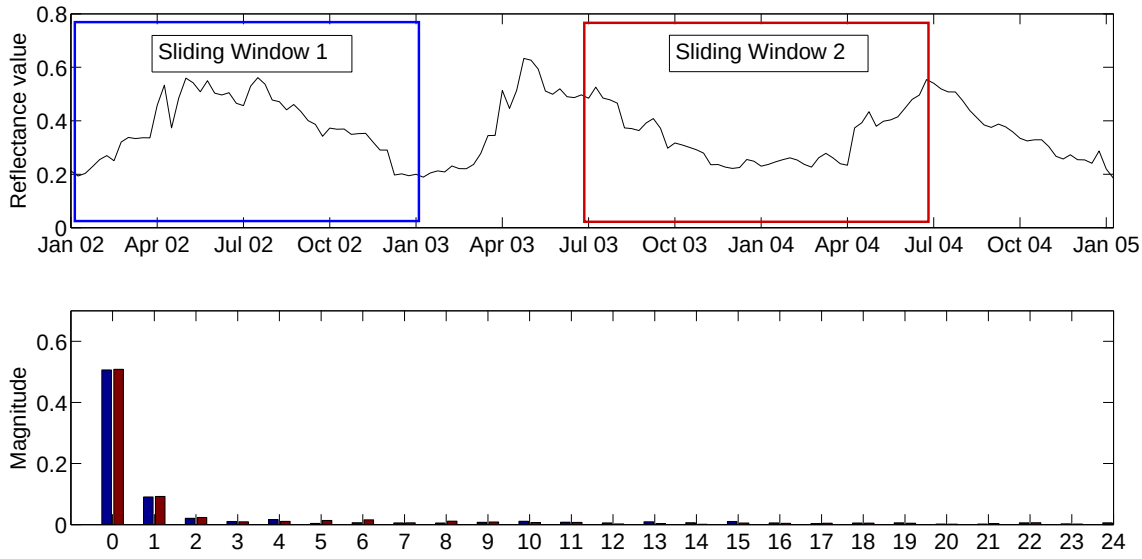


FIGURE 6.4: The feature components $X_p(f)$ extracted from two sliding windows at random positions using equation (6.11) yields similar features.

This limitation is addressed by computing the magnitude of all the FFT components, which removes all the phase offsets. This makes it possible to compensate for both the restrictive position p of the sliding window and the seasonality. This means that p, which is the position of the sliding window, does not have to be incremented by only a fixed annual period, but can be incremented by any natural number. The features for the clustering method are extracted from the sliding window $\mathbf{x}_p$ by the methodology discussed above, and are termed as the SFF $\mathcal{X}_p$. The SFF is computed as

$$\mathcal{X}_p = |\,\mathfrak{F}(\,\mathbf{x}_p\,)\,|, \tag{6.11}$$

where $\mathfrak{F}(\cdot)$ represents the Fourier transform. From the discussion above, a sliding window of any length can be applied to the MODIS time series and moved along the time axis at any rate as long as the feature extraction rule in equation (6.11) is applied. Figure 6.4 illustrates how the SFFs that are extracted using two different sliding window positions in time maintain their position in the feature space, even though the two sliding windows are arbitrarily positioned in time.

The seasonal attribute typically associated with MODIS time series and the slow temporal variation relative to the acquisition interval [15], makes the first few FFT components dominate the frequency spectrum. This reduces the number of features needed to represent the feature space and thus reduces the dimensionality, making clustering an even more feasible option [201].

The mean and annual FFT components from equation (6.11) were considered, as it was shown by Lhermitte [116] that considerable class separation can be achieved from these components. Many

FFT-based classification and segmentation methods consequently only consider a few FFT components [116, 202, 203].

6.5 CHANGE DETECTION METHOD USING THE SEASONAL FOURIER FEATURES

In this section the meaningful clustering approach discussed in section 6.4 is incorporated into a land cover change detection method. The change detection method operates on multiple spectral bands, as shown in figure 6.5.

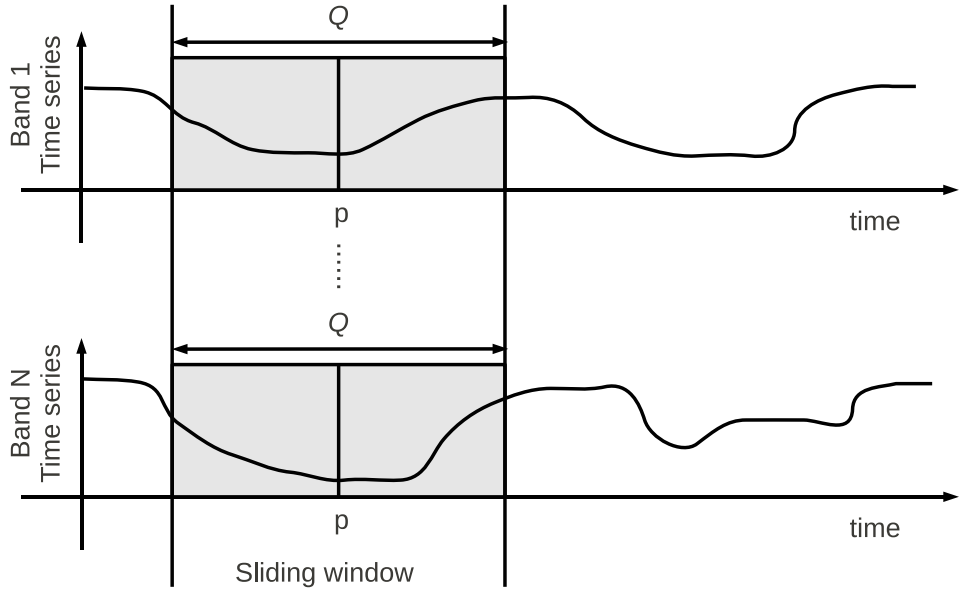


FIGURE 6.5: Temporal sliding window used to define a subsequence of the time series for classification and change detection.

The mean μ and annual α component of the SFF were considered from each of the MODIS spectral bands. These features are expressed using the same methodology discussed above as

$$\mathcal{X}_{bp} = |\, \mathfrak{F}_{b\mu}(\, \mathbf{x}_{bp}) \;\; \mathfrak{F}_{b\alpha}(\, \mathbf{x}_{bp}\,)\,|, \tag{6.12}$$

where $\mathfrak{F}_{b\mu}$ denotes the mean component extracted from the b^{th} spectral band's Fourier transform. The function $\mathfrak{F}_{b\alpha}$ denotes the annual component extracted from the b^{th} spectral band's Fourier transform. The subsequence $\mathbf{x}_{bp}$ is extracted from the b^{th} spectral band at position p.

This selection of frequency components reduces the number of features to represent the feature space and thus reduces the dimensionality. A feature vector is defined to encapsulate multiple spectral bands' SFF. The feature vector is defined as

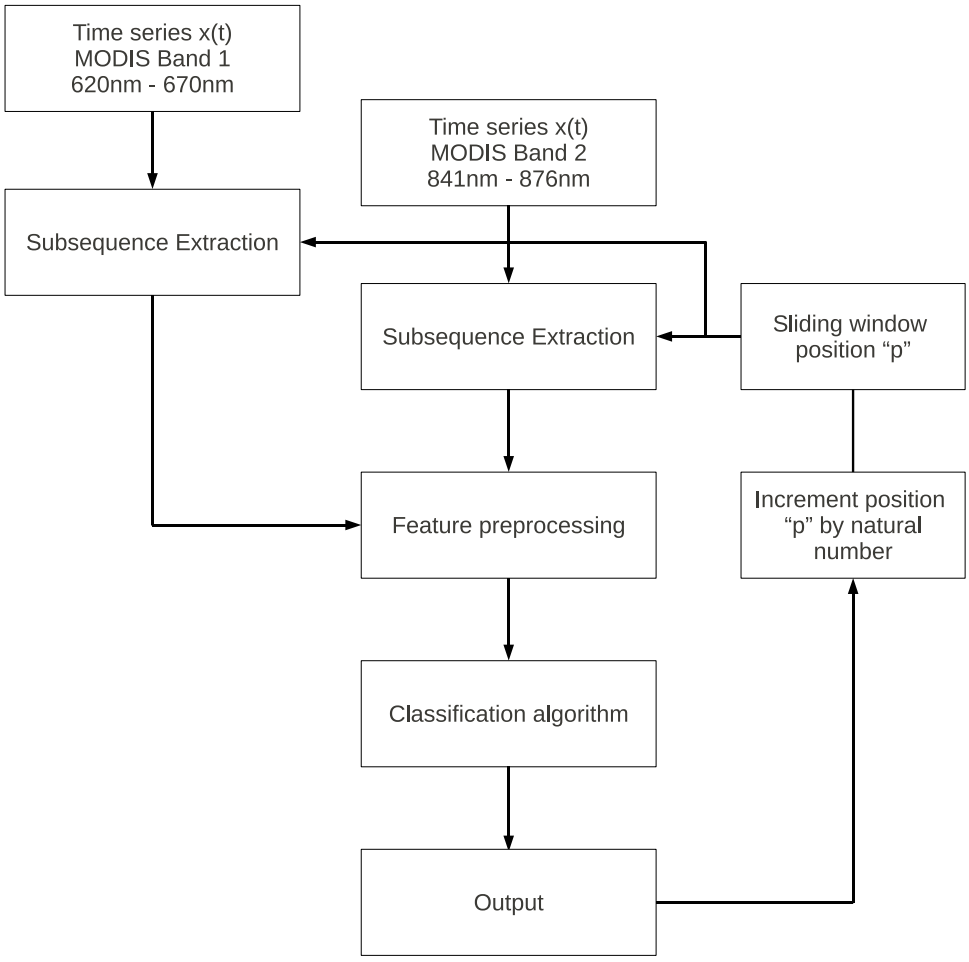


FIGURE 6.6: Subsequences of the time series extracted from the two spectral MODIS bands are processed for clustering and change detection.

$$\mathcal{X}_p^N = [\,\mathcal{X}_{1p}\;\; \mathcal{X}_{2p}\;\; \ldots \mathcal{X}_{Np}\,]. \tag{6.13}$$

Here N denotes the number of spectral bands, and p, $p \in [1, (\mathcal{I} - Q)]$, the position of the sliding window. The first feature vector is the NDVI time series (N=1), which is denoted by $\mathcal{X}_p^1$. This is where the NDVI is computed for $\mathcal{X}_{bp}$ in equation (6.1), which uses a combination of the first two spectral bands (RED and NIR spectral bands) of the MODIS instrument. The second feature vector is to use the first two spectral bands separately (N=2), which is denoted by $\mathcal{X}_p^2$. The last feature vector uses all seven spectral bands separately (N=7), which is denoted by $\mathcal{X}_p^7$.

These SFFs are processed by a machine learning algorithm to detect change. The processing chain for the two spectral bands feature vector $\mathcal{X}_p^2$ is shown as an illustration in figure 6.6. The outputs produced a time series of classifications for a given pixel as a function of the sliding window position p. Land cover change is defined then as the transition in class label of a pixel's time series from one class to another class, after which it remains in the newly assigned class for the remainder of the time

series.

6.6 SUMMARY

In this chapter a detailed overview was given of the pitfall of creating meaningless clusters. An example was presented to illustrate the real limitation of subsequence clustering, followed by a few tentative solutions proposed by Keogh and Lin [29] to solve this problem. Keogh and Lin admit that these solutions are not a fully worked out solution to the problem, but with further investigation a possible solution could be identified. In section 6.5, the SFF was proposed as a solution for a particular data set, which in this case was a time series that had inherent seasonal variations. The SFF will be one of the extracted features used in chapter 8 to detect land cover change.

CHAPTER SEVEN

EXTENDED KALMAN FILTER FEATURES

7.1 OVERVIEW

In this chapter, the Extended Kalman filter (EKF) is used as a feature extraction method, and is studied in-depth. The chapter discusses how the state space variables are used within the EKF, followed by how these are used to separate a set of time series into several classes. The importance of the initial parameters used to set the EKF is discussed in section 7.2.3, illustrating how the behaviour is dependent on these initial parameters.

A novel criterion called the Bias-Variance Equilibrium Point (BVEP), is proposed in section 7.2.4, which defines a desired set of initial parameters that will provide optimal performance. The BVEP criterion is derived using both the temporal and spatial information to design a system with desirable behaviour. A specifically designed search algorithm called the Bias-Variance Search Algorithm (BVSA) is proposed that will adjust the Bias-Variance Score (BVS) to best satisfy the BVEP criterion that will provide good initial parameters for the EKF. The chapter concludes by briefly overviewing the Autocovariance Least Squares (ALS) method, which will be used as benchmark when evaluating the method proposed in section 7.2.4.

7.2 CHANGE DETECTION METHOD: EXTENDED KALMAN FILTER

7.2.1 Introduction

An EKF is discussed as a feature extraction method in this section, which is based on the assumption that the parameters of the underlying model can be used to separate a set of time series into different classes. The model is based on the seasonal behaviour of a specific land cover class. It should be noted that a certain model would better describe a particular land cover class than another and that proper model selection must be done for each different land cover class. It follows that more separable

parameters derived by the EKF make it easier to detect changes in the assigned classes.

Lhermitte *et al.* proposed a method that separates different land cover classes using a Fourier analysis of NDVI time series [116]. It was concluded that good separation is achievable when evaluating the magnitude of the coefficients of the Fourier transform associated with the NDVI signal's mean and amplitude components. Kleynhans *et al.* proposed a method which jointly estimates the mean and seasonal component of the Fourier transform using a triply modulated cosine function [30]. The EKF uses the triply modulated cosine function to model NDVI time series by updating the mean (μ), amplitude (α), and phase (θ) parameters for each time increment.

The method proposed in this section expands on the method of Kleynhans [30] *et al.* by modelling the spectral bands separately and addresses the second constraint of the manual estimation of the initial parameters for the EKF to ensure proper tracking of the observation vectors. The initial parameters include the initial state-space vector, process noise covariance matrix and observation noise covariance matrix. An operator typically uses a training set to supervise the adjustment of the initial parameters until acceptable performance is obtained for a set of time series.

7.2.2 The method

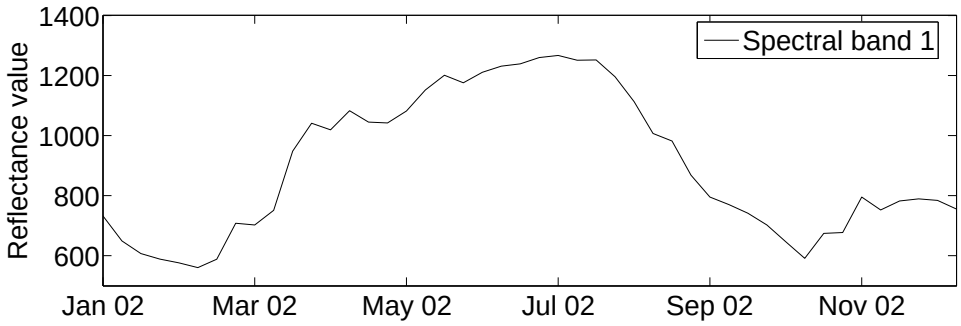
The EKF is a non-linear estimation method, which estimates the unobserved parameters using noisy observation vectors of a related observation model. The EKF has been used in the remote sensing community for parameter estimation of values related to physical, biogeochemical processes or vegetation dynamics models [204, 205].

In figure 7.1, a Fourier transform is used to observe that the majority of the signal energy is contained in the mean and seasonal component of the first spectral band. This implies that the time series in spectral band 1 are well represented in the time domain as a single cosine function with a mean offset, amplitude and phase, as shown in figure 7.2.

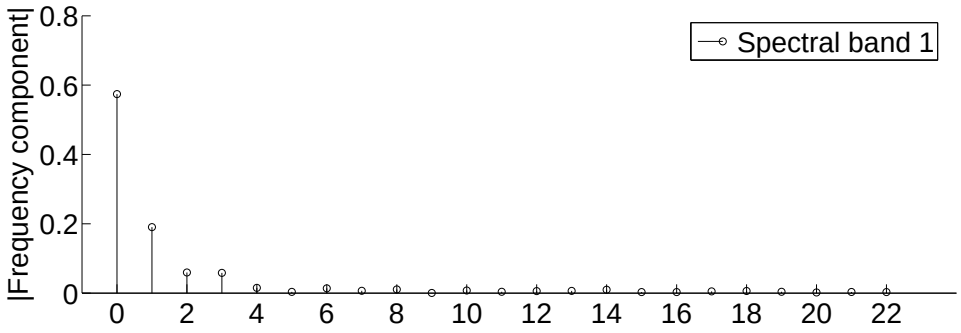
This single cosine model is, however, not a good representation if the time series is non-stationary, which is often the case; for example, inter-annual variability or land cover change. The triply modulated cosine function proposed in [30] is extended here to model a spectral band as

$$x_{i,k,b} = \mu_{i,k,b} + \alpha_{i,k,b} \cos(2\pi f_{\text{samp}} i + \theta_{i,k,b}) + v_{i,k,b}. \tag{7.1}$$

The variable $x_{i,k,b}$ denotes the observed value of the b^{th} spectral band's time series, $b \in \{1, 7\}$, of the k^{th} pixel, $k \in [1, N]$, at time index i, $i \in [1, \mathcal{I}]$. The noise sample of the k^{th} pixel at time i for each spectral band is denoted by $v_{i,k,b}$. The noise is additive with an unknown distribution on all the spectral bands. The cosine function model is separately fitted to each of the spectral bands and is based on several different parameters; the frequency f_{samp} can be explicitly calculated based on the annual



(a) Time series of reflectance values recorded by the MODIS spectral band 1.



(b) Discrete Fourier transform of the time series shown in (a).

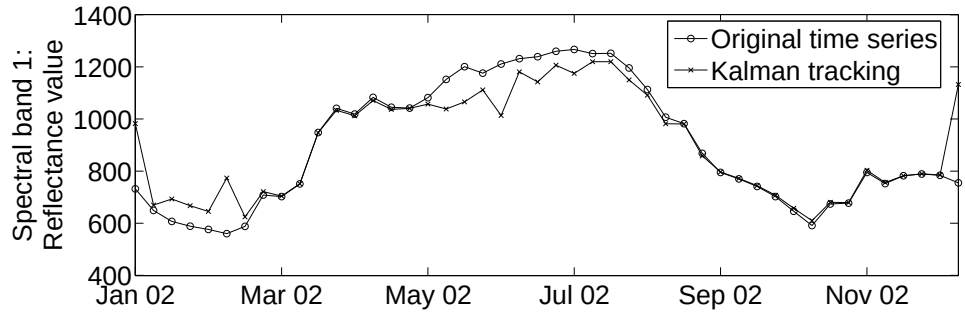
FIGURE 7.1: The time series recorded by the first spectral band for a geographical area is shown in (a) with the corresponding discrete Fourier transform shown in (b).

vegetation growth cycle, and the sampling rate of the MODIS sensor. Given the 8 daily composite MCD43A4 MODIS data set, f_{samp} is set to $\frac{8}{365}$. The non-zero mean of the b^{th} spectral band of the k^{th} pixel at time index i is denoted by $\mu_{i,k,b}$, the amplitude by $\alpha_{i,k,b}$ and the phase by $\theta_{i,k,b}$. The values of $\mu_{i,k,b}$, $\alpha_{i,k,b}$ and $\theta_{i,k,b}$ are dependent on time and must be estimated for each pixel k, $\forall k, k \in [1, N]$, given the spectral band observation vectors $x_{i,k,b}$ for i, $\forall i, i \in [1, \mathcal{I}]$, and b, $b \in \{1, 7\}$.

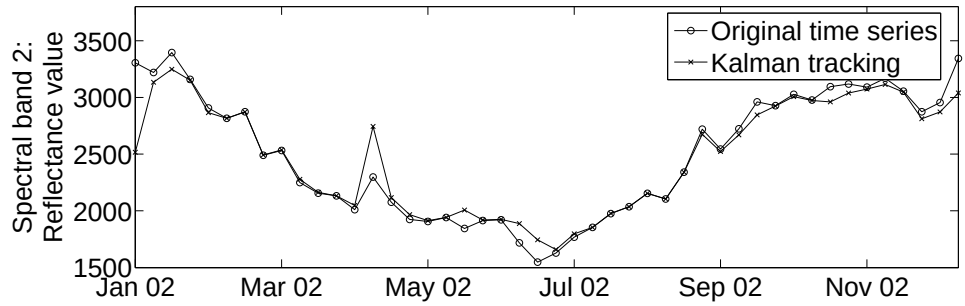
The MODIS spectral bands however are assumed to be uncorrelated and are treated independently in this method. The index b is omitted for convenience, with no loss in generality in the description of this method. A state-space vector is estimated by the EKF at each time increment i for each spectral band and contains all the parameters. This is expressed as

$$\vec{W}_{i,k} = [W_{i,k,1} \;\; W_{i,k,2} \;\; W_{i,k,3}] = [W_{i,k,\mu} \;\; W_{i,k,\alpha} \;\; W_{i,k,\theta}]. \tag{7.2}$$

For the present example of land cover classification, it is assumed that the state-space vector $\vec{W}_{i,k}$ does not change significantly through time; hence, the process model is linear. The measurement model, however, contains the cosine function and, as such, is evaluated via the standard Jacobian formulation, through linear approximation of the non-linear measurement function around the current state-space vector. The state-space vector $\vec{W}_{i,k}$ is related to the observation vector $x_{i,k}$ via a non-linear measurement function. Both the transition function and measurement function are assumed to be



(a) Extended Kalman filter tracking the observation vectors extracted from spectral band 1.



(b) Extended Kalman filter tracking the observation vectors extracted from spectral band 2.

FIGURE 7.2: The tracking of the first two spectral bands using the triply modulated cosine function.

non-perfect, so the addition of process and observation noise is required.

Converting state-space vectors to land cover classes

A machine learning algorithm is used to process the estimated state-space vectors to assign class labels. A class label is assigned to each state-space vector for each pixel at each time increment. This is expressed as

$$\mathcal{C}_{i,k} = \mathcal{F}_{\mathcal{C}}(W_{i,k,1}, \ldots, W_{i,k,S}) = \mathcal{F}_{\mathcal{C}}(\vec{W}_{i,k}), \tag{7.3}$$

where the function $\mathcal{F}_{\mathcal{C}}$ denotes either a supervised or unsupervised classifier. The class label for the k^{th} pixel at time i is denoted by $\mathcal{C}_{i,k}$. Change is declared when a pixel k changes in class label as a function of time i. This is expressed as

$$\mathcal{C}_{i,k} \neq \mathcal{C}_{j,k}, \quad 0 \leq i \leq j, \forall i, j. \tag{7.4}$$

The importance of the initial parameters will be discussed in the next section.

7.2.3 Importance of the initial parameters

The EKF recursively solves the state-space form of a linear dynamic model [185, Ch. 1]. In this section the importance of the initial estimates of the system's variables is shown.

Let $\mathbf{x}_k = \{\vec{x}_{i,k}\}_{i=1}^{i=\mathcal{I}}$, $k \in [1, N]$, denote the k^{th} time series in the set of time series consisting of observation vectors, with each observation vector denoted by $\vec{x}_{i,k} = x_{i,k}$ as the spectral bands are treated independently. Let $\vec{W}_{i,k} = \{W_{i,k,s}\}_{s=1}^{s=S}$ denote the corresponding state-space vector for $x_{i,k}$. Then it is said that the EKF solves the state-space form recursively using the transition equation given as

$$\vec{W}_{i,k} = \mathbf{f}(\vec{W}_{(i-1),k}) + z_{(i-1),k}, \tag{7.5}$$

and the measurement equation given as

$$\vec{x}_{i,k} = \mathbf{h}\left(\vec{W}_{i,k}\right) + v_{i,k}. \tag{7.6}$$

The transition function is denoted by $\mathbf{f}$ and the measurement function is denoted by $\mathbf{h}$. A brief overview of the operations of the EKF which is shown in section 5.5 is revisited for convenience. It is well known from estimation theory that many prediction results simplify when Gaussian distributions are assumed. The process noise vector and observation noise vector are thus assumed to be Gaussian distributed. The process noise vector is thus denoted by $z_{(i-1),k}, z_{(i-1),k} \sim \mathcal{N}(0, \mathcal{Q}_{(i-1),k})$, and the observation noise vector is denoted by $v_{i,k}, v_{i,k} \sim \mathcal{N}(0, \mathcal{R}_{i,k})$.

The EKF recursively adapts the state-space vector for each incoming observation vector by predicting and updating the vector. In the prediction step the state-space vector $\vec{\hat{W}}_{(i|i-1),k}$ and covariance matrix $\mathfrak{B}_{(i|i-1),k}$ are predicted. The predicted state-space vector's estimate $\vec{\hat{W}}_{(i|i-1),k}$ is computed as

$$\vec{\hat{W}}_{(i|i-1),k} = \mathbf{f}\left(\vec{\hat{W}}_{(i-1|i-1),k}\right), \tag{7.7}$$

and the predicted covariance matrix $\mathfrak{B}_{(i|i-1),k}$ is computed as

$$\mathfrak{B}_{(i|i-1),k} = \mathcal{Q}_{(i-1),k} + \mathbf{F}_{\text{est}}\mathfrak{B}_{(i-1|i-1),k}\mathbf{F}_{\text{est}}^{\text{T}}. \tag{7.8}$$

The matrix $\mathbf{F}_{\text{est}}$ is the local linearisation of the non-linear transition function $\mathbf{f}$. In the updating step, the posterior estimate of the state-space vector $\vec{\hat{W}}_{(i|i),k}$ is computed as

$$\vec{\hat{W}}_{(i|i),k} = \vec{\hat{W}}_{(i|i-1),k} + \mathfrak{K}_{i,k}\left(\vec{x}_{i,k} - \mathbf{h}\left(\vec{W}_{i,k}\right)\right), \tag{7.9}$$

using the optimal Kalman gain denoted by $\mathfrak{K}_{i,k}$ which is computed as

$$\mathfrak{K}_{i,k} = \mathfrak{B}_{(i|i-1),k}\mathbf{H}_{\text{est}}^{\text{T}}\mathcal{S}_{i,k}^{-1}. \tag{7.10}$$

The matrix $\mathbf{H}_{\text{est}}$ is the local linearisation of the non-linear measurement function $\mathbf{h}$. The matrix $\mathcal{S}_{i,k}$ denotes the innovation term, which is computed as

$$\mathcal{S}_{i,k} = \mathbf{H}_{\text{est}}\mathfrak{B}_{(i|i-1),k}\mathbf{H}_{\text{est}}^{\text{T}} + \mathcal{R}_{i,k}. \tag{7.11}$$

The posterior estimate of the covariance matrix $\mathfrak{B}_{(i|i),k}$ is computed as

$$\mathfrak{B}_{(i|i),k} = \mathfrak{B}_{(i|i-1),k} - \mathfrak{K}_{i,k}\mathcal{S}_{i,k}\mathfrak{K}_{i,k}^{\text{T}}. \tag{7.12}$$

The tracking performance of the EKF is assessed by evaluating the stability of the state-space vector and error in estimating the observation vector. The error in estimating the observation vector is computed as the absolute error between the estimated observation vector $\vec{\hat{x}}_{i,k}$ and the actual observation vector $\vec{x}_{i,k}$. This is expressed as

$$\mathcal{E}_{\vec{x},i,k} = |\vec{x}_{i,k} - \vec{\hat{x}}_{i,k}| = \left|\vec{x}_{i,k} - \mathbf{h}\left(\vec{W}_{(i|i),k}\right)\right|. \tag{7.13}$$

In equation (7.13), it is observed that the state-space vector $\vec{\hat{W}}_{(i|i),k}$ determines the observation error $\mathcal{E}_{\vec{x},i,k}$. Thus the state-space vector $\vec{\hat{W}}_{(i|i),k}$ can be selected to minimise the observation error. The MODIS spectral bands are assumed to be uncorrelated and only produce a single reflectance value for each pixel. This simplifies equation (7.13) to

$$\mathcal{E}_{\vec{x},i,k} = |x_{i,k} - \hat{x}_{i,k}| = \left|x_{i,k} - \mathbf{h}\left(\vec{W}_{(i|i),k}\right)\right|. \tag{7.14}$$

The observation error is easily minimised by significantly varying $\vec{\hat{W}}_{(i|i),k}$ to accommodate the fluctuation in observation vectors. This does not bode well if the underlying structure of the system is also being analysed. A significantly varying state-space vector $\vec{\hat{W}}_{(i|i),k}$ is indicative of an unstable model. The conclusion is that the state-space model must be kept stable, while also attempting to minimise equation (7.14).

The initial estimates provided to the EKF will now be discussed to illustrate their importance. A stable state-space vector requires a small adaptation from $\vec{\hat{W}}_{(i-1|i-1),k}$ to $\vec{\hat{W}}_{(i|i),k}$. The initial estimated state-space vector $\vec{\hat{W}}_{(0|0),k}$, $\vec{\hat{W}}_{(0|0),k} \in \mathcal{W}$, at the first observation vector $\vec{x}_{0,k}$ is optimised using a local search method or domain knowledge which satisfies

$$\hat{\vec{W}}_{(0|0),k} = \underset{\vec{W}\in\mathcal{W}}{\operatorname{argmin}} \left\{ \left| \vec{x}_{0,k} - \mathbf{h}\left(\vec{W}\right) \right| \right\}, \tag{7.15}$$

then

$$\mathcal{E}_{\vec{x},0,k} = \left| \vec{x}_{0,k} - \mathbf{h}\left(\hat{\vec{W}}_{(0|0),k,b}\right) \right|. \tag{7.16}$$

The recursive adaptation of the state-space vector's estimate $\hat{\vec{W}}_{(i|i),k}$ is then calculated using the predicted step given in equation (7.7) and the updating step in equation (7.9). Equation (7.7) is substituted into equation (7.9) to yield

$$\hat{\vec{W}}_{(i|i),k} = \mathbf{f}\left(\hat{\vec{W}}_{(i-1|i-1),k}\right) + \mathfrak{K}_{i,k}\left(\vec{x}_{i,k} - \mathbf{h}\left(\mathbf{f}\left(\hat{\vec{W}}_{(i-1|i-1),k}\right)\right)\right). \tag{7.17}$$

The Kalman gain $\mathfrak{K}_{i,k}$ determines the rate of change in the error between the predicted and estimated state-space vector. If the observation error is large and the Kalman gain is large, then large changes will be made to the current state-space vector. If the observation error is large and the Kalman gain is small, then the state-space's estimate $\hat{\vec{W}}_{(i|i),k}$ will adapt slowly, which typically leads to a large observation error $\mathcal{E}_{\vec{x},i,k}$ (equation (7.13)) until it eventually converges. If the observation error is small and the Kalman gain is large, then the state-space vector will struggle to converge, as it will continually overshoot the desired state-space vector that will minimise equation (7.13). Substituting the optimal Kalman gain given in equation (7.10) into equation (7.17) expands it to

$$\hat{\vec{W}}_{(i|i),k} = \mathbf{f}\left(\hat{\vec{W}}_{(i-1|i-1),k}\right) + \mathfrak{B}_{(i|i-1),k}\mathbf{H}_{\text{est}}^{\text{T}}\mathcal{S}_{i,k}^{-1}\left(\vec{x}_{i,k} - \mathbf{h}\left(\mathbf{f}\left(\hat{\vec{W}}_{(i-1|i-1),k}\right)\right)\right). \tag{7.18}$$

The Kalman gain is dependent on the predicted covariance matrix $\mathfrak{B}_{(i|i-1),k}$ and innovation term $\mathcal{S}_{i,k}$. The innovation term controls the trust region within the state-space vector's space. This is dependent on the predicted covariance matrix $\mathfrak{B}_{(i|i-1),k}$ and observation covariance noise $\mathcal{R}_{i,k}$. Substituting the innovation term given in equation (7.11) into equation (7.18) results in

$$\begin{aligned} \hat{\vec{W}}_{(i|i),k} \quad = \quad & \mathbf{f}\left(\hat{\vec{W}}_{(i-1|i-1),k}\right) + \mathfrak{B}_{(i|i-1),k}\mathbf{H}_{\text{est}}^{\text{T}}(\mathbf{H}_{\text{est}}\mathfrak{B}_{(i|i-1),k}\mathbf{H}_{\text{est}}^{\text{T}} + \\ & \mathcal{R}_{i,k})^{-1}\left(\vec{x}_{i,k} - \mathbf{h}\left(\mathbf{f}\left(\hat{\vec{W}}_{(i-1|i-1),k}\right)\right)\right). \end{aligned} \tag{7.19}$$

The last term to evaluate is the predicted covariance matrix $\mathfrak{B}_{(i|i-1),k}$. The predicted covariance matrix $\mathfrak{B}_{(i|i-1),k}$ is substituted to yield an updated state-space vector as

$$\begin{aligned}\hat{\vec{W}}_{(i|i),k} &= \mathbf{f}\left(\hat{\vec{W}}_{(i-1|i-1),k}\right) + (\mathcal{Q}_{(i-1),k} + \mathbf{F}_{\text{est}}\mathfrak{B}_{(i-1|i-1),k}\mathbf{F}_{\text{est}}^{\text{T}})\mathbf{H}_{\text{est}}^{\text{T}} \\ &\quad (\mathbf{H}_{\text{est}}(\mathcal{Q}_{(i-1),k} + \mathbf{F}_{\text{est}}\mathfrak{B}_{(i-1|i-1),k}\mathbf{F}_{\text{est}}^{\text{T}})\mathbf{H}_{\text{est}}^{\text{T}} + \mathcal{R}_{i,k})^{-1} \\ &\quad \left(\vec{x}_{i,k} - \mathbf{h}\left(\mathbf{f}\left(\hat{\vec{W}}_{(i-1|i-1),k}\right)\right)\right). \end{aligned} \tag{7.20}$$

The transition function $\mathbf{f}$ and measurement function $\mathbf{h}$ are assumed to be known. The observation vector $\vec{x}_{i,k}$ is supplied by the real system. The only variables left within equation (7.20) are: (1) previous state-space vector's estimate $\hat{\vec{W}}_{(i-1|i-1),k}$, (2) process noise's covariance matrix $\mathcal{Q}_{(i-1),k}$, (3) previous estimate of covariance matrix $\mathfrak{B}_{(i-1|i-1),k}$, and (4) observation noise's covariance matrix $\mathcal{R}_{i,k}$.

The previous estimation of the covariance matrix $\mathfrak{B}_{(i-1|i-1),k}$ will be briefly explored, as it is part of equation (7.20). The covariance matrix $\mathfrak{B}_{(i-1|i-1),k}$ is updated with

$$\mathfrak{B}_{(i-1|i-1),k} = \mathfrak{B}_{(i-1|i-2),k} - \mathfrak{K}_{(i-1),k}\mathcal{S}_{(i-1),k}\mathfrak{K}_{(i-1),k}^{\text{T}}. \tag{7.21}$$

Substituting the Kalman gain of equation (7.10) into equation (7.21) yields

$$\mathfrak{B}_{(i-1|i-1),k} = \mathfrak{B}_{(i-1|i-2),k} - (\mathfrak{B}_{(i-1|i-2),k}\mathbf{H}_{\text{est}}^{\text{T}}\mathcal{S}_{(i-1),k}^{-1})\mathcal{S}_{(i-1),k}(\mathfrak{B}_{(i-1|i-2),k}\mathbf{H}_{\text{est}}^{\text{T}}\mathcal{S}_{(i-1),k,b}^{-1})^{\text{T}}. \tag{7.22}$$

Substituting the innovation term of equation (7.11) into equation (7.22) yields

$$\begin{aligned}\mathfrak{B}_{(i-1|i-1),k} &= \mathfrak{B}_{(i-1|i-2),k} - (\mathfrak{B}_{(i-1|i-2),k}\mathbf{H}_{\text{est}}^{\text{T}}(\mathbf{H}_{\text{est}}\mathfrak{B}_{(i-1|i-2),k}\mathbf{H}_{\text{est}}^{\text{T}} + \mathcal{R}_{(i-1),k})^{-1}) \\ &\quad (\mathbf{H}_{\text{est}}\mathfrak{B}_{(i-1|i-2),k}\mathbf{H}_{\text{est}}^{\text{T}} + \mathcal{R}_{(i-1),k})(\mathfrak{B}_{(i-1|i-2),k}\mathbf{H}_{\text{est}}^{\text{T}}(\mathbf{H}_{\text{est}}\mathfrak{B}_{(i-1|i-2),k} \\ &\quad \mathbf{H}_{\text{est}}^{\text{T}} + \mathcal{R}_{(i-1),k})^{-1})^{\text{T}}. \end{aligned} \tag{7.23}$$

The predicted covariance matrix $\mathfrak{B}_{(i-1|i-2),k}$ given in equation (7.8) is substituted into equation (7.23), which yields

$$\begin{aligned}\mathfrak{B}_{(i-1|i-1),k} &= (\mathcal{Q}_{(i-2),k} + \mathbf{F}_{\text{est}}\mathfrak{B}_{(i-2|i-2),k}\mathbf{F}_{\text{est}}^{\text{T}}) - ((\mathcal{Q}_{(i-2),k} + \mathbf{F}_{\text{est}}\mathfrak{B}_{(i-2|i-2),k}\mathbf{F}_{\text{est}}^{\text{T}})\mathbf{H}_{\text{est}}^{\text{T}} \\ &\quad (\mathbf{H}_{\text{est}}(\mathcal{Q}_{(i-2),k} + \mathbf{F}_{\text{est}}\mathfrak{B}_{(i-2|i-2),k}\mathbf{F}_{\text{est}}^{\text{T}})\mathbf{H}_{\text{est}}^{\text{T}} + \mathcal{R}_{(i-1),k})^{-1})(\mathbf{H}_{\text{est}}(\mathcal{Q}_{(i-2),k} + \\ &\quad \mathbf{F}_{\text{est}}\mathfrak{B}_{(i-2|i-2),k}\mathbf{F}_{\text{est}}^{\text{T}})\mathbf{H}_{\text{est}}^{\text{T}} + \mathcal{R}_{(i-1),k})((\mathcal{Q}_{(i-2),k} + \mathbf{F}_{\text{est}}\mathfrak{B}_{(i-2|i-2),k}\mathbf{F}_{\text{est}}^{\text{T}})\mathbf{H}_{\text{est}}^{\text{T}} \\ &\quad (\mathbf{H}_{\text{est}}(\mathcal{Q}_{(i-2),k} + \mathbf{F}_{\text{est}}\mathfrak{B}_{(i-2|i-2),k}\mathbf{F}_{\text{est}}^{\text{T}})\mathbf{H}_{\text{est}}^{\text{T}} + \mathcal{R}_{(i-1),k})^{-1})^{\text{T}}. \end{aligned} \tag{7.24}$$

Equation (7.20) is computed for every newly obtained observation vector. The state-space vector's estimate $\hat{\vec{W}}_{(i|i),k}$ requires the results from equation (7.24) to compute the current estimates. The transition function $\mathbf{F}_{\text{est}}$ and measurement function $\mathbf{H}_{\text{est}}$ are known, then the only variables left to compute in equation (7.24) are: (1) initial covariance matrix $\mathfrak{B}_{(0|0),k}$, (2) process covariance matrix $\mathcal{Q}_{(i-1),k}$, and (3) observation noise's covariance matrix $\mathcal{R}_{i,k}$. The conclusion from equation (7.20) and equation (7.24) is that the initial parameters of importance are:

1. the initial state-space vector's estimate $\hat{\vec{W}}_{(0|0),k}$,
2. the initial covariance matrix estimate $\mathfrak{B}_{(0|0),k}$,
3. the process covariance matrix $\mathcal{Q}_{(i-1),k}$, and
4. the observation covariance matrix $\mathcal{R}_{i,k}$.

The initial state-space vector's estimate $\hat{\vec{W}}_{(0|0),k}$ is initialised using equation (7.15). Even if an incorrect estimate is used, the state-space vector $\hat{\vec{W}}_{(i|i),k}$ should converge to the correct vector as $i \to \infty$. The same is true about the initial covariance matrix $\mathfrak{B}_{(0|0),k}$. As $i \to \infty$, the covariance matrix $\mathfrak{B}_{(i|i),k}$ should tend to converge to the correct matrix. The usual operation of the EKF sets the initial covariance matrix equal to an identity matrix.

The initial covariance matrix $\mathfrak{B}_{(0|0),k}$ will stabilise, as equation (7.8) is known as a discrete Riccati equation, and under certain circumstances will converge, which results in equation (7.24) converging to a stable state [206]. The conditions for convergences of the discrete Riccati equation are:

1. the process covariance matrix $\mathcal{Q}_{(i-1),k}$ is a positive definite matrix,
2. the observation covariance matrix $\mathcal{R}_i, k$ is a positive definite matrix,
3. the pair $(\mathbf{F}_{\text{est}}, z_{(i-1),k})$ is controllable, *i.e.*,

$$\operatorname{rank}\left[z_{(i-1),k} | \mathbf{F}_{\text{est}} z_{(i-1),k} | \mathbf{F}_{\text{est}}^2 z_{(i-1),k} | \dots | \mathbf{F}_{\text{est}}^{N-1} z_{(i-1),k}\right] = N, \tag{7.25}$$

4. and the pair $(\mathbf{F}_{\text{est}}, \mathbf{H}_{\text{est}})$ is observable, *i.e.*,

$$\operatorname{rank}\left[\mathbf{H}_{\text{est}}^{\text{T}} | \mathbf{F}_{\text{est}}^{\text{T}} \mathbf{H}_{\text{est}}^{\text{T}} | (\mathbf{F}_{\text{est}}^{\text{T}})^2 \mathbf{H}_{\text{est}}^{\text{T}} | \dots | (\mathbf{F}_{\text{est}}^{\text{T}})^{N-1} \mathbf{H}_{\text{est}}^{\text{T}}\right] = N, \tag{7.26}$$

with $N \in \mathbb{N}$. Under the above conditions, the predicted covariance matrix $\mathfrak{B}_{(i|i-1),k}$ converges to a constant matrix

$$\lim_{i\to\infty} \mathfrak{B}_{(i|i-1),k} = \mathfrak{B}_{\text{const}}, \tag{7.27}$$

where $\mathfrak{B}_{\text{const}}$ is a symmetric positive definite matrix. $\mathfrak{B}_{\text{const}}$ is a unique positive definite solution of the discrete Riccati equation and $\mathfrak{B}_{\text{const}}$ is independent of the initial distribution of the initial state-space vector's estimate $\hat{\vec{W}}_{(0|0),k}$.

The system can also estimate $\hat{\vec{W}}_{(0|0),k}$ and $\mathfrak{B}_{(0|0),k}$ using an offline training phase. Offline refers to observation vectors that are stored and are used recursively for estimation. The process covariance matrix $\mathcal{Q}_{(i-1),k}$ and observation covariance matrix $\mathcal{R}_{i,k}$ are assumed to be constant throughout the recursive estimation of the observation vector. This is usually manually set by a system analyst in an offline training phase through successive adjustments. In this thesis the initial EKF is defined as:

1. The initial state-space vector $\hat{\vec{W}}_{(0|0),k}$ is estimated offline.
2. The initial covariance matrix $\mathfrak{B}_{(0|0),k}$ is estimated offline.
3. The process covariance matrix $\mathcal{Q}_{(i-1),k}$ is set to a fixed matrix.
4. The observation covariance matrix $\mathcal{R}_{i,k}$ is set to a fixed matrix.

The EKF will track the observation vectors with minimum residual and have a stable internal state-space vector if all initial parameters are properly estimated.

7.2.4 Bias-Variance Equilibrium Point

The general approach to estimating and initialising the state-space vectors, as well as the observation and process noise's covariance matrices for the EKF, is usually for an analyst to determine these offline using a training data set. Proper estimation of the initial parameters through various methods leads to good feature vectors from the EKF, while improper estimation could cause system instability, which leads to delayed tracking or abnormal system behaviour.

A novel BVEP criterion is proposed in this section that will use temporal and spatial information to design a parameter space where desirable system behaviour is expected. This is accomplished by first observing the dependencies between the initial parameters. The proposed criterion uses an unsupervised BVSA to adjust the BVS iteratively to determine proper initial parameters for the EKF. The characteristics of the initial parameters are first explored before describing the criterion. The first parameter is the observation covariance matrix $\mathcal{R}_{i,k}$. The observation covariance matrix $\mathcal{R}_{i,k}$ is defined as

$$\mathcal{R}_{i,k} = E[(x_{i,k} - E[x_{i,k}])^2]. \tag{7.28}$$

This is due to the fact that the spectral bands are assumed to be uncorrelated and that the MODIS sensor only produces a single reflectance value per pixel per spectral band. The second parameter is the process covariance matrix $\mathcal{Q}_{i,k}$. The process covariance matrix $\mathcal{Q}_{i,k}$ is defined as

$$\mathcal{Q}_{i,k} = \begin{pmatrix} E[(W_{i,k,1}-E[W_{i,k,1}])(W_{i,k,1}-E[W_{i,k,1}])] & \dots & E[(W_{i,k,1}-E[W_{i,k,1}])(W_{i,k,S}-E[W_{i,k,S}])] \\ \vdots & \ddots & \vdots \\ E[(W_{i,k,S}-E[W_{i,k,S}])(W_{i,k,1}-E[W_{i,k,1}])] & \dots & E[(W_{i,k,S}-E[W_{i,k,S}])(W_{i,k,S}-E[W_{i,k,S}])] \end{pmatrix}. \quad (7.29)$$

The state-space variables within the state-space vector are assumed to be uncorrelated; the process covariance matrix simplifies to

$$\mathcal{Q}_{i,k} = \text{diag}\{E[(W_{i,k,s}-E[W_{i,k,s}])^2]\}, \ \ \forall s. \quad (7.30)$$

The setting of the initial parameters has a major effect on the overall system performance. The initial state-space vector $\vec{W}_{(0|0),k}$ for the first observation vector $\vec{x}_{0,k}$ is optimised using equation (7.15). The initial estimated covariance matrix $\mathfrak{B}_{(0|0),k}$ is usually set to the identity matrix. This only leaves the estimation of the observation covariance matrix $\mathcal{R}_{i,k}$ and process covariance matrix $\mathcal{Q}_{i,k}$. Let the uncorrelated observation covariance matrix's diagonals be placed into a vector called the observation candidate vector $\Upsilon_{\mathcal{R},i,k}$, were $\Upsilon_{\mathcal{R},k}$ is selected from the space $\upsilon_{\mathcal{R}}$, and it is expressed as

$$\Upsilon_{\mathcal{R},i,k} = 10^{\zeta_{i,k}/10}, \quad (7.31)$$

with

$$\zeta_{i,k} = 10\log_{10}\left(E[(\vec{x}_{i,k}-E[\vec{x}_{i,k}])^2]\right). \quad (7.32)$$

Let the uncorrelated process covariance matrix's diagonals be placed into a vector called the process candidate vector $\Upsilon_{\mathcal{Q},i,k}$, were $\Upsilon_{\mathcal{Q},k}$ is selected from space $\upsilon_{\mathcal{Q}}$, which is expressed as

$$\Upsilon_{\mathcal{Q},i,k} = 10^{[\varsigma_{i,k,1} \ \dots \ \varsigma_{i,k,S}]/10} = 10^{\vec{\varsigma}_{i,k}/10}, \quad (7.33)$$

with

$$\varsigma_{i,k,s} = 10\log_{10}\left(E[(W_{i,k,s}-E[W_{i,k,s}])^2]\right). \quad (7.34)$$

It should be noted that the EKF only updates recursively the state-space vector $\vec{W}_{(i|i),k}$, and covariance matrix $\mathfrak{B}_{(i|i),k}$. The time index of the observation covariance matrix $\mathcal{Q}_{i,k}$ has been left

inserted to emphasise the time effect in a dynamic linear system. The EKF, however, does not alter the observation covariance matrix at each time increment and is thus constant for all time indices. This is formally stated as $\mathcal{Q}$=$\mathcal{Q}_i$, $\forall i$. The process covariance matrix is also retained as a constant for all time indices and this is stated as $\mathcal{R}$=$\mathcal{R}_i$, $\forall i$. This concludes that the observation covariance matrix and process covariance matrix are independent of time. This property allows the observation candidate vector to be rewritten as

$$\Upsilon_{\mathcal{R},k} = 10^{\zeta_k/10} \quad \forall k, \tag{7.35}$$

and the process candidate vector rewritten as

$$\Upsilon_{\mathcal{Q},k} = 10^{[\varsigma_{k,1} \ \cdots \ \varsigma_{k,S}]/10} = 10^{\vec{\varsigma}_k/10} \quad \forall k. \tag{7.36}$$

It was stated earlier that the performance of the Kalman filter is measured by the residual error in tracking the observation vectors and the internal stability of the state-space vector. A parameter space is thus defined to describe the system behaviour.

The first desired behaviour is the tracking of the observation vector with minimal residual. This desired behaviour is expressed as the minimal achievable sum of absolute residuals $\sigma_{\mathcal{E}}$, which is computed as

$$\sigma_{\mathcal{E}} = \min_{\Upsilon_{\mathcal{R},k}\in\upsilon_{\mathcal{R}},\Upsilon_{\mathcal{Q},k}\in\upsilon_{\mathcal{Q}}} \left\{ \sum_{k=1}^{N}\sum_{i=1}^{\mathcal{I}} \left\| \hat{x}_{i,k} - x_{i,k} \right\| \right\}, \tag{7.37}$$

then

$$[\mathcal{R}_{\sigma_{\mathcal{E}}}, \mathcal{Q}_{\sigma_{\mathcal{E}}}] = \underset{\Upsilon_{\mathcal{R},k}\in\upsilon_{\mathcal{R}},\Upsilon_{\mathcal{Q},k}\in\upsilon_{\mathcal{Q}}}{\operatorname{argmin}} \left\{ \sum_{k=1}^{N}\sum_{i=1}^{\mathcal{I}} \left\| \hat{x}_{i,k} - x_{i,k} \right\| \right\}. \tag{7.38}$$

Thus $\sigma_{\mathcal{E}}$ is the minimal residual, and $[\mathcal{R}_{\sigma_{\mathcal{E}}}, \mathcal{Q}_{\sigma_{\mathcal{E}}}]$ represents the parameters required to achieve this value. The minimal residual is computed as

$$\sigma_{\mathcal{E}} = \sum_{k=1}^{N}\sum_{i=1}^{\mathcal{I}} \left\| \hat{x}_{i,k} - x_{i,k} \right\| \Bigg|_{\mathcal{R}=\mathcal{R}_{\sigma_{\mathcal{E}}},\mathcal{Q}=\mathcal{Q}_{\sigma_{\mathcal{E}}}}. \tag{7.39}$$

The second criterion is to have internal stability of the state-space vector. This can be measured as the variations in each of the state-space variables. The second desired behaviour is expressed as the minimal achievable absolute deviation in state-space variables, which is computed as

$$\sigma_s = \min_{\Upsilon_{\mathcal{R},k}\in\upsilon_{\mathcal{R}},\Upsilon_{\mathcal{Q},k}\in\upsilon_{\mathcal{Q}}} \left\{ \sum_{k=1}^{N}\sum_{i=1}^{\mathcal{I}} \left\| W_{i,k,s} - E[W_{i,k,s}] \right\| \right\}, \quad \forall s, \tag{7.40}$$

then

$$[\mathcal{R}_{\sigma_s}, \mathcal{Q}_{\sigma_s}] = \underset{\Upsilon_{\mathcal{R},k}\in\upsilon_{\mathcal{R}},\Upsilon_{\mathcal{Q},k}\in\upsilon_{\mathcal{Q}}}{\operatorname{argmin}} \left\{ \sum_{k=1}^{N}\sum_{i=1}^{\mathcal{I}} \left\| W_{i,k,s} - E[W_{i,k,s}] \right\| \right\}, \quad \forall s. \tag{7.41}$$

Thus σ_s is the minimal absolute deviation in the state-space variable s. The set $[\mathcal{R}_{\sigma_s}, \mathcal{Q}_{\sigma_s}]$ represents the parameters required to achieve this value. The minimal absolute deviation is computed as

$$\sigma_s = \sum_{k=1}^{N}\sum_{i=1}^{\mathcal{I}} \left\| W_{i,k,s} - E[W_{i,k,s}] \right\| \Bigg|_{\mathcal{R}=\mathcal{R}_{\sigma_s},\mathcal{Q}=\mathcal{Q}_{\sigma_s}}. \tag{7.42}$$

The spatial information is included through the use of a set of time series all located in a specific geographical area. The set of N time series for a geographical area is denoted by $\{\vec{x}_{i,k}\}$. Let $q_{i,\mathcal{E}}$ denote the probability density function derived at time index i from the residuals given over the set of observations $\{x_{i,k}\}_{k=1}^{k=N}$ such that $P[a \leq \mathcal{E} \leq b] = \int_a^b f(e)\mathrm{d}e = \int_a^b f(e,\mathcal{R},\mathcal{Q})\mathrm{d}e$ i.e.,

$$P[a \leq \mathcal{E} \leq b] = \int_a^b q(e,\mathcal{R},\mathcal{Q})\mathrm{d}e = \int_a^b q_{i,\mathcal{E}}\mathrm{d}e. \tag{7.43}$$

Let $q_{i,s}$ denote the probability density function for the state-space variable s derived at time index i from the deviations given over the set of state-space vectors $\{W_{i,k,s}\}_{k=1}^{k=N}$ such that $P[a \leq s \leq b] = \int_a^b f(s')\mathrm{d}s' = \int_a^b f(s',\mathcal{R},\mathcal{Q})\mathrm{d}s'$ i.e.,

$$P[a \leq s \leq b] = \int_a^b q(s',\mathcal{R},\mathcal{Q})\mathrm{d}s' = \int_a^b q_{i,s}\mathrm{d}s'. \tag{7.44}$$

A conditioned observation probability density function $q^*_{i,\mathcal{E}}$ is defined as the probability density function $q_{i,\mathcal{E}}$ in equation (7.43), which uses the set $[\mathcal{R}_{\sigma_\mathcal{E}}, \mathcal{Q}_{\sigma_\mathcal{E}}]$ to satisfy the condition given in equation (7.39) as

$$P[a \leq \mathcal{E} \leq b] = \int_a^b q(e,\mathcal{R}_{\sigma_\mathcal{E}},\mathcal{Q}_{\sigma_\mathcal{E}})\mathrm{d}e = \int_a^b q^*_{i,\mathcal{E}}\mathrm{d}e. \tag{7.45}$$

A conditioned process probability density function $q^*_{i,s}$ is defined as the probability density function $q_{i,s}$ in equation (7.44), which uses the set $[\mathcal{R}_{\sigma_s}, \mathcal{Q}_{\sigma_s}]$ to satisfy the condition given in equation (7.42) as

$$P[a \leq\ s \leq b] = \int_a^b q(s',\mathcal{R}_{\sigma_s},\mathcal{Q}_{\sigma_s})\mathrm{d}s' = \int_a^b q^*_{i,s}\mathrm{d}s'. \tag{7.46}$$

The performance of the current estimate $\Upsilon_{\mathcal{R},k}$ and $\Upsilon_{\mathcal{Q},k}$ is defined by a criterion that evaluates how

well the conditions stated in equation (7.37) and equation (7.40) are satisfied. The current estimates are recursively updated and are denoted by $\hat{\Upsilon}^{\iota}_{\mathcal{R},k}$ and $\hat{\Upsilon}^{\iota}_{\mathcal{Q},k}$, where ι denotes the iteration number. The current estimates $\hat{\Upsilon}^{\iota}_{\mathcal{R},k}$ and $\hat{\Upsilon}^{\iota}_{\mathcal{Q},k}$ are used to derive the set of probability density functions $\{\hat{q}^{\iota}_{i,\mathcal{E}}\}$, $\forall i$, and $\{\hat{q}^{\iota}_{i,s}\}$, $\forall i$.

A f-divergent distance known as the Hellinger distance [207, 208] is used to measure the similarity between the probability density functions $\hat{q}^{\iota}_{i,\mathcal{E}}$ and $q^{*}_{i,\mathcal{E}}$. The modified Hellinger distance $\mathcal{H}_{i,\mathcal{E}}$, $\mathcal{H}_{i,\mathcal{E}} \in [0, 1]$, is computed as

$$\mathcal{H}_{i,\mathcal{E}} = 1 - \sqrt{1 - \sqrt{\int_{-\infty}^{\infty} \hat{q}^{\iota}_{i,\mathcal{E}}\, q^{*}_{i,\mathcal{E}} \mathrm{d}e}}, \tag{7.47}$$

where a value of $\mathcal{H}_{i,\mathcal{E}} \rightarrow 1$ means high similarity between $\hat{q}^{\iota}_{i,\mathcal{E}}$ and $q^{*}_{i,\mathcal{E}}$, while $\mathcal{H}_{i,\mathcal{E}} \rightarrow 0$ means low similarity. The modified Hellinger distance is also used to measure the similarity for the state-space variables. The modified Hellinger distance $\mathcal{H}_{i,s}$, $\mathcal{H}_{i,s} \in [0, 1]$, is computed as

$$\mathcal{H}_{i,s} = 1 - \sqrt{1 - \sqrt{\int_{-\infty}^{\infty} \hat{q}^{\iota}_{i,s}\, q^{*}_{i,s} \mathrm{d}s'}}, \tag{7.48}$$

where a value of $\mathcal{H}_{i,s} \rightarrow 1$ means high similarity between $\hat{q}^{\iota}_{i,b,s}$ and $q^{*}_{i,b,s}$, while $\mathcal{H}_{i,s} \rightarrow 0$ means low similarity.

The BVS is defined to encapsulates all similarity metrics as

$$\Gamma_i = \min\Big(\{\mathcal{H}_{i,s}\}_{s=1}^{s=S} \cup \{\mathcal{H}_{i,\mathcal{E}}\}\Big). \tag{7.49}$$

Finding optimal estimates for $\hat{\Upsilon}^{\iota}_{\mathcal{R},k}$ and $\hat{\Upsilon}^{\iota}_{\mathcal{Q},k}$ requires a stable covariance matrix $\mathfrak{B}_{(i|i),k}$. Equation (7.27) states that the predicted covariance matrix $\mathfrak{B}_{(i|i),k}$ should converge to a constant matrix under certain prerequisite conditions. Let $\mathcal{I}_T$, $\mathcal{I}_T \ll \mathcal{I}$, denote the number of time steps required to ensure that the predicted covariance matrix $\mathfrak{B}_{(\mathcal{I}_T|\mathcal{I}_T-1),k}$ converges to ensure a stable covariance matrix $\mathfrak{B}_{(\mathcal{I}_T|\mathcal{I}_T),k}$. The BVS is deemed accurate at $\mathcal{I}_T$, which is defined as

$$\Gamma_{\mathcal{I}_T} = \min\Big(\{\mathcal{H}_{\mathcal{I}_T,s}\}_{s=1}^{s=S} \cup \{\mathcal{H}_{\mathcal{I}_T,\mathcal{E}}\}\Big). \tag{7.50}$$

The BVEP criterion is defined as the BVS, which optimally maximises the conditions. The BVEP criterion is defined as

$$\Gamma^{*}_{\mathcal{I}_T} = \max_{\Upsilon^{\iota}_{\mathcal{R},k} \in \upsilon_{\mathcal{R}}, \Upsilon^{\iota}_{\mathcal{Q},k} \in \upsilon_{\mathcal{Q}}} \{\Gamma_{\mathcal{I}_T}\}. \tag{7.51}$$

If the reflectance values of the spectral bands are correlated, then the BVS is expanded to compensate

for this as

$$\Gamma_{\mathcal{I}_T} = \min\left\{\left\{\{\mathcal{H}_{\mathcal{I}_T,b,s}\}_{s=1}^{s=S}\right\}_{b=1}^{b=B}\{\mathcal{H}_{\mathcal{I}_T,b,\mathcal{E}}\}_{b=1}^{b=B}\right\}. \tag{7.52}$$

In this thesis however it was assumed that the spectral bands were uncorrelated.

7.2.5 Bias-Variance Search algorithm

The BVSA is proposed in this section, which will attempt to estimate $\hat{\Upsilon}^{\iota}_{\mathcal{R},k}$ and $\hat{\Upsilon}^{\iota}_{\mathcal{Q},k}$ to satisfy the BVEP criterion using the BVS given in equation (7.50). The BVSA starts by creating ideal operating conditions for each parameter in the EKF, followed by using a hill-climbing algorithm to search for a set of $\hat{\Upsilon}^{\iota}_{\mathcal{R},k}$ and $\hat{\Upsilon}^{\iota}_{\mathcal{Q},k}$ that will satisfy at best the ideal operating conditions for all the parameters within the EKF.

The first ideal condition is a system that employs perfect tracking of the observation vectors. This ideal condition is used to create the probability density function $q^*_{i,\mathcal{E}}$. This is obtained by

$$q^*_{i,\mathcal{E}} = \left\{q_{i,\mathcal{E}} : \{\zeta_k\} \to -\infty; \{\varsigma_{k,s}\} \to \infty, \forall\, s\right\}. \tag{7.53}$$

Under perfect conditions the probability density function $q^*_{i,\mathcal{E}}$ should tend to be an impulse of unity power situated around the zero position, meaning zero error residual is measured.

The second ideal condition is a system that employs a stable state-space variable. This ideal condition is used to create the probability density function $q^*_{i,s}$. This is obtained by

$$q^*_{i,s} = \left\{q_{i,s} : \{\zeta_k\} \to \infty; \{\varsigma_{k,\{s\}\setminus s}\} \to \infty; \{\varsigma_{k,s}\} \to -\infty\right\}. \tag{7.54}$$

This condition creates an environment which attempts to track the state-space variable s with the smallest variation.

After the ideal observation conditions' probability density functions $q^*_{i,\mathcal{E}}$ and $q^*_{i,s}$ have been computed, a hill-climbing search algorithm is applied to find a set of initial parameters that will best satisfy all these ideal conditions. The BVSA iteratively searches the parameter space and is described briefly below.

Step 1: The BVSA starts with the initial parameters set as $\zeta^0_k = 0$dB, $\forall\, k$, and $\varsigma^0_{k,s} = 0$dB, $\forall\, k, s$.

Step 2: Compute the state-space vector $\vec{W}_{(\mathcal{I}_T|\mathcal{I}_T),k}$ at time $\mathcal{I}_T$ using the same $\hat{\Upsilon}^{\iota}_{\mathcal{R},k} = \zeta^{\iota}_k$ and $\hat{\Upsilon}^{\iota}_{\mathcal{Q},k} = \{\zeta^{\iota}_k\}_{s=1}^{s=S}$ for every time series in the set $\{\mathbf{x}_k\}_{k=1}^{k=N}$.

Step 3: Obtain the probability density function of the residual errors $q^{\iota}_{i,\mathcal{E}}$ over the N time series at time index $\mathcal{I}_T$.

Step 4: Obtain the probability density function of the residual errors $q^{\iota}_{i,s}$ of the state-space variable s over the N time series at time index $\mathcal{I}_T$.

Step 5: Compute the modified Hellinger distance $\mathcal{H}_{\mathcal{I}_T,\mathcal{E}}$ as shown in equation (7.47).

Step 6: Compute the modified Hellinger distance $\mathcal{H}_{\mathcal{I}_T,s}$ as shown in equation (7.48).

Step 7: Determine the best performing condition $\mathcal{H}_{\text{best}}$ as

$$\mathcal{H}_{\text{best}} = \max\left\{\{\mathcal{H}_{\mathcal{I}_T,\mathcal{E}}\}\ \{\mathcal{H}_{\mathcal{I}_T,s}\}\right\}. \tag{7.55}$$

Step 8: Determine the worst performing condition $\mathcal{H}_{\text{worst}}$ as

$$\mathcal{H}_{\text{worst}} = \min\left\{\{\mathcal{H}_{\mathcal{I}_T,\mathcal{E}}\}\ \{\mathcal{H}_{\mathcal{I}_T,s}\}\right\}. \tag{7.56}$$

Step 9: Adjust the new ζ^{ι}_k according to its relative position to the best and worst performing parameters using a threshold $\rho_{\mathcal{H}}$, $\rho_{\mathcal{H}} \in [0,1]$, $\rho_{\mathcal{H}} \in \mathbb{R}$. The adjustment is made as

$$\zeta^{\iota+1}_k = \begin{cases} \zeta^{\iota}_k + \gamma^{\iota} & \text{if } \left(\frac{\mathcal{H}_{\mathcal{I}_T,\mathcal{E}} - \mathcal{H}_{\text{worst}}}{\mathcal{H}_{\text{best}} - \mathcal{H}_{\text{worst}}} > \rho_{\mathcal{H}}\right) \\ \zeta^{\iota}_k - \gamma^{\iota} & \text{if } \left(\frac{\mathcal{H}_{\mathcal{I}_T,\mathcal{E}} - \mathcal{H}_{\text{worst}}}{\mathcal{H}_{\text{best}} - \mathcal{H}_{\text{worst}}} \leq \rho_{\mathcal{H}}\right) \end{cases}. \tag{7.57}$$

The variable γ^{ι} is a decreasing scalar measured in decibels and is a non-negative real number.

Step 10: Adjust the new ς^{ι}_k according to its relative position to the best and worst performing parameters using a threshold $\rho_{\mathcal{H}}$, $\rho_{\mathcal{H}} \in [0,1]$, $\rho_{\mathcal{H}} \in \mathbb{R}$. The adjustment is made as

$$\varsigma^{\iota+1}_{k,s} = \begin{cases} \varsigma^{\iota}_{k,s} + \gamma^{\iota} & \text{if } \left(\frac{\mathcal{H}_{\mathcal{I}_T,s} - \mathcal{H}_{\text{worst}}}{\mathcal{H}_{\text{best}} - \mathcal{H}_{\text{worst}}} > \rho_{\mathcal{H}}\right) \\ \varsigma^{\iota}_{k,s} - \gamma^{\iota} & \text{if } \left(\frac{\mathcal{H}_{\mathcal{I}_T,s} - \mathcal{H}_{\text{worst}}}{\mathcal{H}_{\text{best}} - \mathcal{H}_{\text{worst}}} \leq \rho_{\mathcal{H}}\right) \end{cases}. \tag{7.58}$$

The variable γ^{ι} is a decreasing scalar measured in decibels and is a non-negative real number.

Repeat steps 2–10 until one of the parameters ζ_k or $\varsigma_{k,s}$ stabilises. After the search algorithm converges, the estimates $\hat{\Upsilon}^{\iota}_{\mathcal{R},k}$ and $\hat{\Upsilon}^{\iota}_{\mathcal{Q},k}$ are used to initialise the EKF.

7.3 AUTOCOVARIANCE LEAST SQUARES METHOD

In this section a method known as the ALS is investigated as an alternative for setting the initial parameters of the EKF. If complete system knowledge about the measurement function $\mathbf{h}$ and transition

function $\mathbf{f}$ were known, then the EKF only requires knowledge of the observation covariance matrix $\mathcal{R}$ and process covariance matrix $\mathcal{Q}$. Several different approaches have been formulated to solve the estimation of these covariance matrices [209–211]. All these methods assumed that the noise-shaping matrix in the transition equation is known. In the absence of information on the noise-shaping matrix the linear dynamic model is modelled as a Gaussian noise vector. The method that is investigated is the ALS method, which operates in the absence of information on the noise shaping matrix [212]. The ALS method assumes that:

1. both the measurement function $\mathbf{h}$ and transition function $\mathbf{f}$ are known,
2. enough observation vectors are available to ensure internal covariance matrix $\mathfrak{B}_{(i|i)}$ becomes stable, and
3. the residuals at different time increments are uncorrelated.

The method estimates the observation covariance matrix $\mathcal{R}$ and process covariance matrix $\mathcal{Q}$ by minimising an objective function [212]. The objective function is a function of the measurement function $\mathbf{h}$, transition function $\mathbf{f}$ and the noise-shaping matrix (if present). The motivation for using this method is that it avoids a complicated non-linear estimation approach used by methods that employ a maximum likelihood estimation approach [213].

7.4 SUMMARY

In this chapter a novel BVEP criterion was proposed, which computes the process covariance matrix and observation covariance matrix using spatial and temporal information. This criterion could easily be extended, as shown in equation (7.52), to include spectral information if the spectral bands are correlated.

The derived matrices in the BVS were then used to initialise the EKF, which is used as a feature extraction method. The BVSA provides covariance matrices that could be used for a variety of different applications. A variety of different search algorithms can be used with the BVEP criterion, such as interior point, active set, simulated annealing, etc. These methods will be explored in chapter 8.

CHAPTER EIGHT

RESULTS

8.1 OVERVIEW

The first part of the chapter studies the effects of different parameter settings to determine their influence on the quality of the solutions. The second part of the chapter explores the classification accuracies of several different methods, while the last part investigates the change detection accuracies of the best performing methods. The chapter concludes with the processing of these methods on large regional scale areas and assessing the outcome.

8.2 GROUND TRUTH DATA SET

A labelled data set, offering ground truth, is required to evaluate the performance of different land cover change detection algorithms. The performance of the methods is measured with a variety of tests to assess accuracy and robustness. Two study areas were investigated in this chapter, namely the Limpopo and Gauteng provinces.

Limpopo province: The Limpopo province is located in the northern parts of South Africa and is largely covered by natural vegetation. The expansion of human settlements, often informal and unplanned, is the most pervasive form of land cover change in the province. Areas were identified where new settlements were known to have been built over the last decade.

Gauteng province: The Gauteng province is located in the highveld of South Africa and is the most urbanised province in the country. The province contributes 33% of the country's national economy. Active migration to the province from other provinces is motivated by the prospect of higher incomes and more diverse employment opportunities. An average growth of 249 310

(a) Quickbird image taken on 1 March 2004 (courtesy of Google^{TM}Earth).

(b) Quickbird image taken on 9 July 2008 (courtesy of Google^{TM}Earth).

(c) Quickbird image taken on 11 December 2009 (courtesy of Google^{TM}Earth).

FIGURE 8.1: Three high resolution images acquired over a residential area called Midstream estates located in Midrand, Gauteng, South Africa. The area was zoned for residential use in 2003 and new settlements were erected only after 9 July 2008.

persons per year within the province has been estimated over the past decade [214, 215]. It should be noted that the Gauteng province only covers 1.4% of the country's total land area, while housing over 20% of the population.

8.2.1 MODIS time series data set

The performance of different land cover change detection methods will be evaluated on a per pixel basis using a set of different spectral bands' time series, which are extracted from the MODIS land surface reflectance product. The MODIS (MCD43A4, Collection V005) 500 metre, Nadir and BRDF adjusted spectral reflectance bands were used, as these significantly reduce the anisotropic scattering effects of surfaces under different illumination and observation conditions [27, 28]. The first two spectral bands (RED and NIR spectral bands) are the only spectral bands available at a spatial resolution of 250 metre, and are not BRDF adjusted. The 500 metre resolution spectral bands were considered to illustrate the

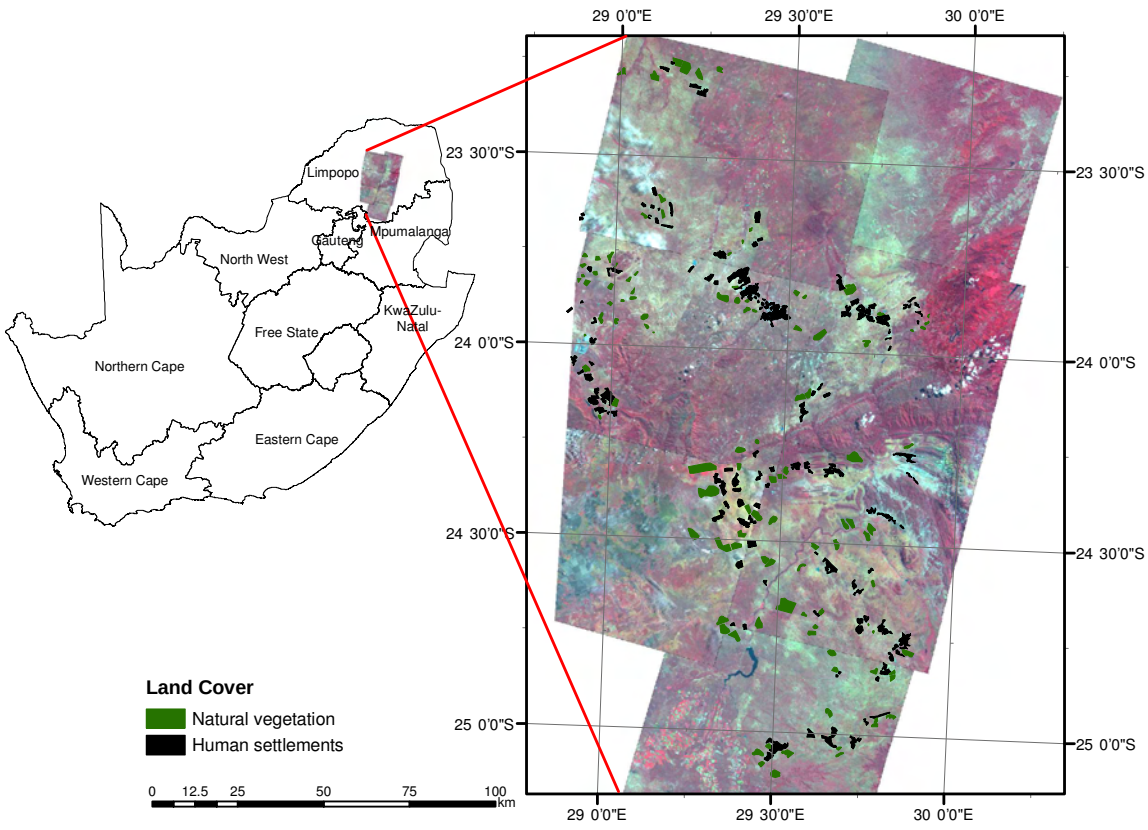


FIGURE 8.2: The Limpopo province study area has land cover types polygons overlayed using Albers projection on SPOT5 RGB 321 imagery that was acquired between March 2006 and May 2006. The SPOT2 images were acquired of the same area in May 2000 [8].

advantages of using additional spectral bands in the analysis. A time series is extracted for all 7 spectral bands from the data set (MODIS tile H20V11) for each pixel in each study area (year 2000–2008).

8.2.2 Manual inspection of study areas

Identification of no change areas: Visual interpretation of SPOT2 (year 2000) and SPOT5 (year 2006 / 2008) high spatial resolution images was used to verify that none of the areas classified as no change, experienced any form of land cover change during the study period.

Identification of change areas: This data set was captured using the same procedure explained for the no change areas, except that areas where new human settlements had formed during the study period were captured.

Even though human settlement expansion is one of the most pervasive forms of land cover change in South Africa, information on this form of land cover change is poorly documented, and vital details

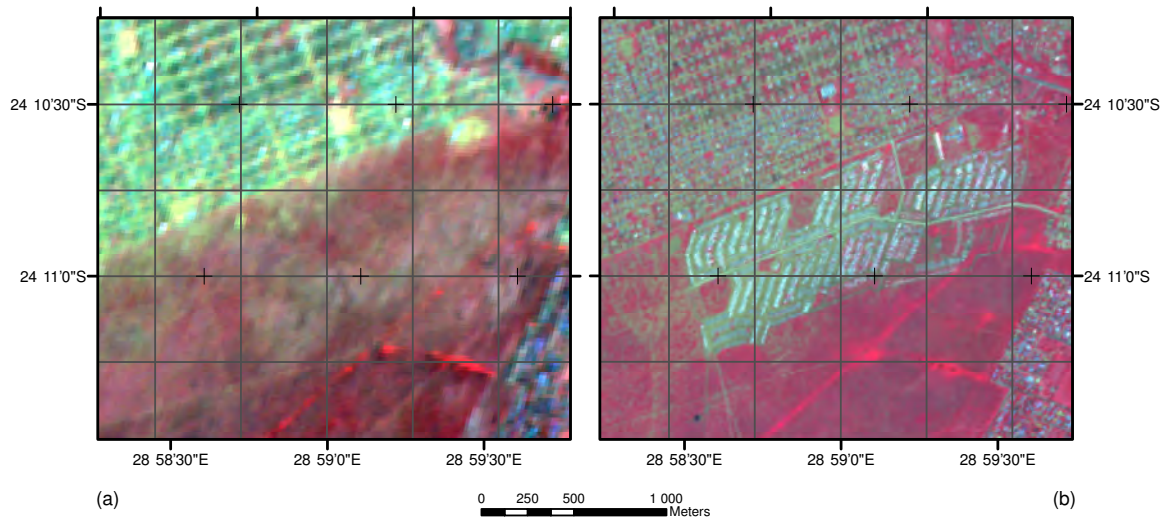


FIGURE 8.3: A land cover change of natural vegetation to human settlement in Sekuruwe. Sekuruwe is a human settlement that is located in the Limpopo province, South Africa. The SPOT2 image (RGB 321) was acquired on 2 May 2000 of the natural vegetation area (a) and a SPOT5 (RGB 321) image was acquired on 1 May 2007 of a newly developed human settlement (b). The SPOT2 and SPOT5 image is projected to a MODIS sinusoidal WGS84 projection and is overlaid with a MODIS 500 metre coordinate grid [8].

such as the date of land cover conversion cannot be determined reliably. An example of inaccurate information is shown in figure 8.1. The local municipality demarcated new roads in a suburban area for future expansion. Unfortunately, no newly developed settlements had been built until quite recently. A good estimate on the date of land cover conversion can be made if regular acquisitions are obtained for a particular area. In this example, if only the images in figure 8.1(a) and figure 8.1(c) were available, then the date of change could be somewhere between March 2004 and December 2009. The real land cover change only occurred after July 2008, which illustrates the importance of the vital statistic of knowing when change occurred.

Once the areas have been identified as change or no change, they are mapped with polygons on the geocoded SPOT imagery, as shown in figure 8.2. The SPOT images are then projected to a MODIS sinusoidal WGS84 projection and is overlaid with a MODIS 500 metre coordinate grid (Figure 8.3). The MODIS grid blocks, which contain the mapped polygons, are thus marked for extraction from the MODIS MCD43A4 product.

8.2.3 Google^{TM}Earth used for visual inspection

Google^{TM}Earth is being used more routinely in visually displaying and validating of geographical areas [216, 217]. As an additional validation step, the MODIS pixel coordinates of interest were transformed into a KML (Keyhole Markup Language) file and visually inspected in Google^{TM}Earth. The true colour of the high resolution Quickbird images available in Google^{TM}Earth made a good platform to illustrate some of the findings presented in this chapter.

Google^{TM}Earth operates on a free sharing policy of images and does not have a mandate to buy regular imagery of certain geographical areas. This means that only areas in which suitable images were acquired before and after the settlement formation could be validated using Google^{TM}Earth.

8.2.4 Simulated land cover data set

Accurate date-of-change information was not available for the ground truth data set, preventing the measurement of the delay in detecting change of the proposed methods. Land cover change events were simulated by combining data from natural vegetation and human settlement time series, with the advantage of a known date of change and transition duration [8].

Four testing data subsets were created, based on concatenating time series of different combinations of classes:

- Subset 1: natural vegetation time series (class 1) concatenated to settlement time series (class 2).
- Subset 2: settlement time series (class 2) concatenated to natural vegetation time series (class 1).
- Subset 3: settlement time series (class 2) concatenated to another settlement time series (class 2).
- Subset 4: natural vegetation time series (class 1) concatenated to another natural vegetation time series (class 1).

These four subsets were used to test if the change detection algorithm can detect change reliably on subsets 1 and 2, while not falsely detecting change for subsets 3 and 4.

8.3 SYSTEM OUTLINE

In this section an overall system outline is provided to explain how all the different methods interconnect with one another (figure 8.4) to create a change detection framework. The system starts with the input of time series extracted from the MODIS MCD43A4 land surface reflectance

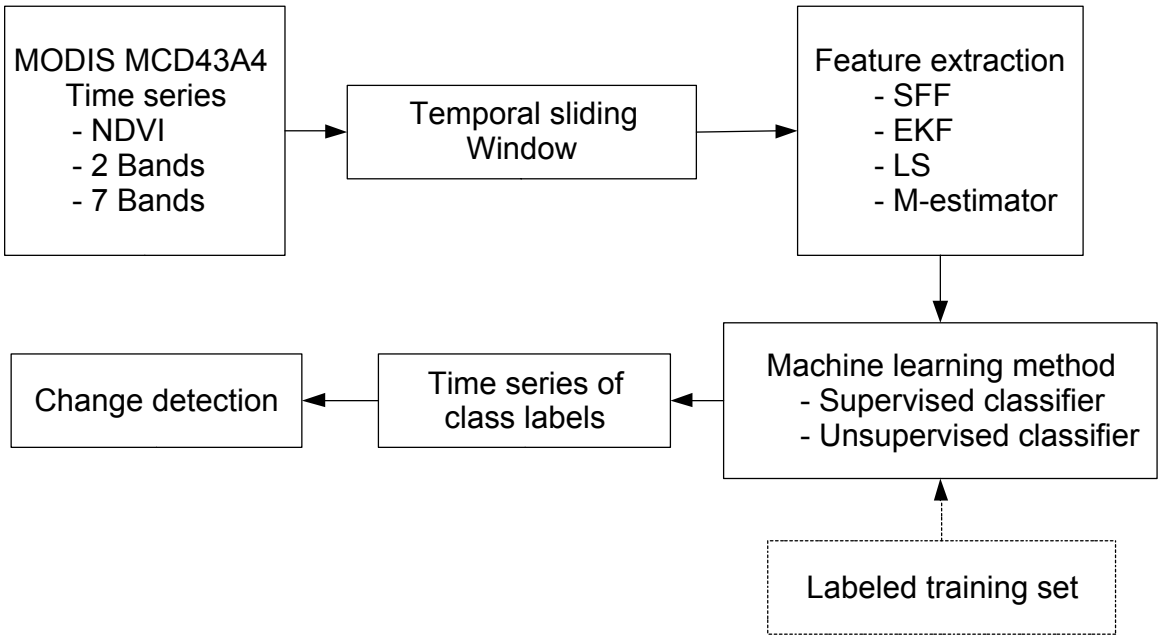


FIGURE 8.4: A flow diagram which provide a complete system outline used in this chapter in all the experiments.

product (section 2.6). The time series used as input can either be one of the following spectral band combinations as listed with the number of dimensions in the feature space as:

- NDVI (2-dimensions),
- first two spectral bands (RED and NIR spectral bands, 4-dimensions), and
- all seven spectral bands (land bands, 14-dimensions).

A temporal sliding window is used to extract sequential subsequences from the time series for analysis. The length of the temporal sliding window is varied, depending on the feature extraction method used. The feature extraction methods applied to these subsequences are listed with their corresponding temporal sliding window length as:

- SFF (6, 12, and 18 months),
- least squares (12 months, see section 8.5.3),
- M-estimator (12 months, see section 8.5.3), and
- EKF (8 days).

The extracted feature vectors are then processed by a machine learning method, which assigns a class label to each feature vector. The machine learning method can be either a supervised classifier, or

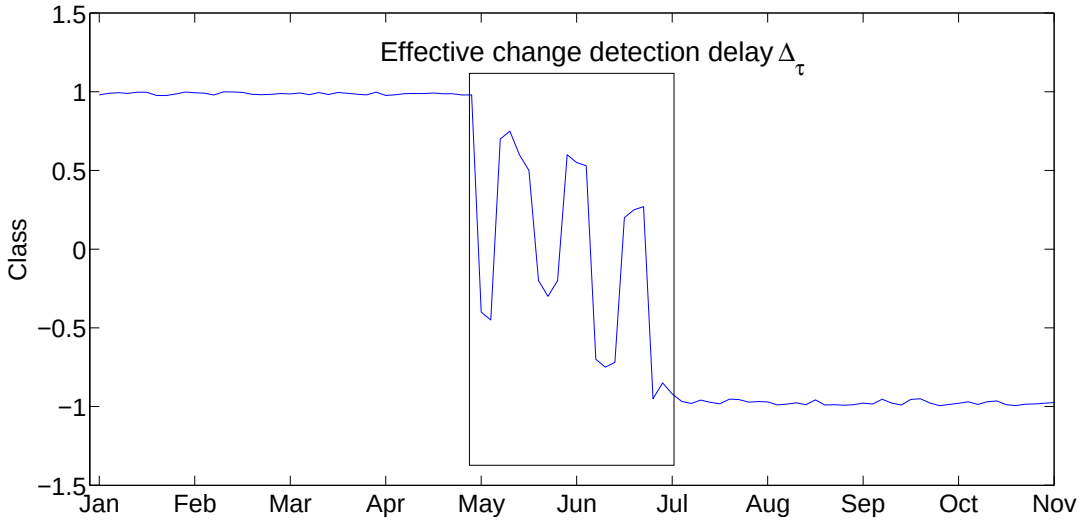


FIGURE 8.5: An illustrative example of the effective change detection delay Δ_τ, which is defined as the time duration it takes after the first acquisition of change in the MODIS time series for the land cover change detection algorithm to detect it.

an unsupervised classifier. The class labels produced by the machine learning method form a new time series, where each time index corresponds to a classification of an extracted temporal subsequence. An example of such a time series consisting of class labels is given in figure 8.5. The class labels in the time series start in the class label *1* (natural vegetation class), and transitions to the class label *-1* (human settlement class), as the position of the temporal sliding window is incremented. It is clear from the illustration that a change in the land cover has occurred in the time series.

A simulated land cover change data set was created in response to the lack of information about when the actual land cover changed (section 8.2.4). In the simulated land cover change data set, the exact position (date) of land cover change in the time series is known. This creates another dimension of evaluation, which enables the quantification of how quickly the land cover change can be detected by the land cover change detection algorithm.

This delay in detecting a change in land cover is termed the effective change detection delay Δ_τ, and is defined as the time duration in which the change detection algorithm is unable to detect the simulated land cover change in subset 1, and subset 2 after the date of change. The concatenation process (section 8.2.4) in the simulated land cover change data set produces an abrupt change in the time series, which does not necessarily represent the reality of human-induced change such as settlement expansion, which could take several months to develop. A blending period (linear blend over 12 and 24 months) from one land cover time series to another was initially considered, but it turned out that it did not affect the ability to detect the land cover change correctly, as this is a property that is exploited in the post-classification change detection approach. The blending model does not

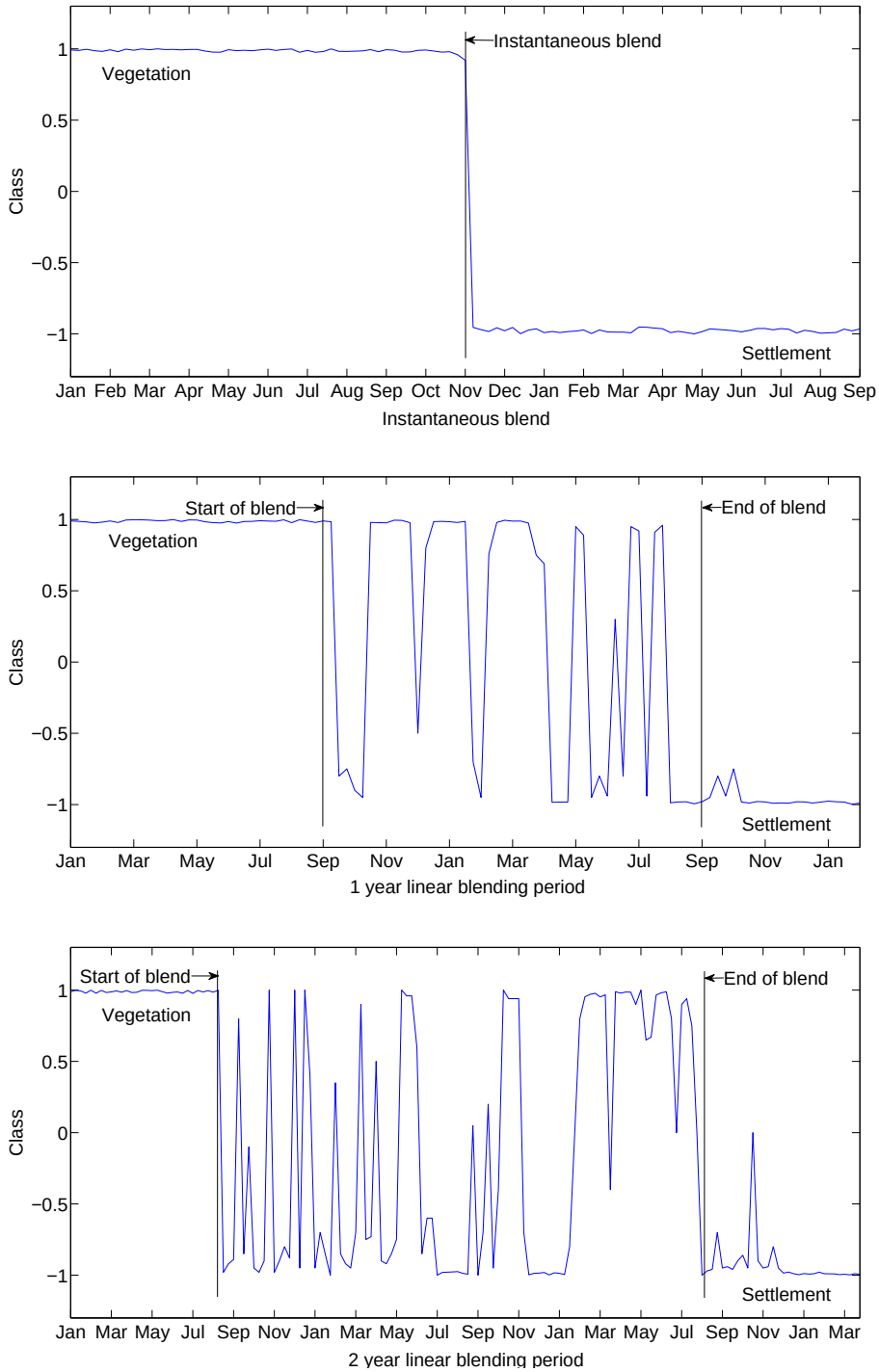


FIGURE 8.6: Class label time series for simulated land cover change from natural vegetation to human settlement. The top panel is for instantaneous simulated land cover change, the middle panel is for a land cover change over a 12 month blending period and the bottom panel is for a land cover change over a 24 month blending period.

faithfully simulate all forms of actual land cover change, but it does delay the date on which the change is declared (figure 8.6). It was concluded that only abrupt concatenation should be used when

measuring the lower limit of effective change detection Δ_τ time.

8.4 EXPERIMENTAL PLAN

In this section an overview is given of the experiments conducted in this chapter. The experiments were conducted in the Limpopo and Gauteng provinces. The number of pixels per data set in each province is given in table 8.1.

Table 8.1: Number of pixels per land cover type, per study area used for training, validation and testing data sets.

Province	Class	Number of time series
Limpopo	Vegetation - No change	1497
	Settlement - No change	1735
	Simulated land cover change	500
	Real land cover change	118
	Complete Province	590212
Gauteng	Vegetation - No change	591
	Settlement - No change	371
	Simulated land cover change	124
	Real land cover change	180
	Complete Province	78702

The experiments conducted in this chapter are grouped into four categories:

1. Parameter exploration (section 8.5),
2. Classification (section 8.6),
3. Change detection (section 8.7),
4. Provincial experiments (section 8.9).

A set of general experiments were conducted in section 8.5 to optimise the parameters which are used in the remaining sections (section 8.6 – section 8.9). The first set of experiments is used to determine the optimal network architecture for the MLP (section 8.5.1) that will minimise the generalisation error. The second set of experiments is used to explore two different training methods for the MLP (section 8.5.2): batch mode and iteratively retrained mode. The third set of experiments is used to optimise the length of the sliding window for the least squares method (section 8.5.3). The fourth set of experiments is used to compare the performance of the EKF when using the BVEP criterion (denoted by EKF_{BVEP}) and ALS methods (denoted by EKF_{ALS}, section 8.5.4). The fifth set

of experiments is used to investigate the setting of the BVEP criterion using the BVSA (section 8.5.5). The sixth set of experiments is used to investigate the performance of each of the regression methods (section 8.5.6). The seventh set of experiments is used to determine the number of clusters to use in the unsupervised classifier (section 8.5.7). The last set of experiments is used to determine the average silhouette value for different clustering algorithms (section 8.5.8).

In section 8.6, the classification accuracy is computed for each of the two classes in a range of experiments on the no change data set. In each section the average classification accuracy is reported, along with the corresponding standard deviation. Different combinations of feature extraction methods and machine learning methods are investigated in these experiments. The feature extraction methods that were explored are:

- least squares model fitting,
- M-estimator model fitting,
- SFF, and
- EKF_{BVEP}.

The classification experiments are divided into supervised classification experiments and unsupervised classification experiments. The machine learning method determines the category of the classifier. The machine learning methods that were explored are:

1. Supervised classifier:
    - Multilayer Perceptron (section 8.6.1).
2. Unsupervised classifier:
    - Hierarchical clustering, single linkage criterion (section 8.6.3),
    - Hierarchical clustering, average linkage criterion (section 8.6.3),
    - Hierarchical clustering, complete linkage criterion (section 8.6.3),
    - Hierarchical clustering, Ward clustering method (section 8.6.4),
    - Partitional clustering, K-means algorithm (section 8.6.5),
    - Partitional clustering, EM algorithm (section 8.6.6).

The objective of the classification experiments is to identify combinations of methods which have high classification accuracies and minimal corresponding standard deviations.

The change detection algorithms in this thesis are based on a post-classification approach, and are thus dependent on the classification accuracies reported in section 8.6. The classification accuracies are used to identify a set of methods that will provide acceptable change detection accuracies (section 8.7).

The first set of experiments is used to determine the change detection accuracies on the simulated land cover change data set. The number of time series blended to simulate the land cover change in each province is given in table 8.1. The true positives and false positives are reported on the simulated land cover data set in section 8.7.1.

The second set of experiments is used to determine the change detection accuracies on the real land cover change data set. The number of time series that experienced actual land cover change in the labelled data set of each province is given in table 8.1. In these experiments only the true positives are reported on the real land cover data set in section 8.7.2.

The third set of experiments is used to determine the effective change detection delay Δ_τ on the simulated land cover change data set. The number of time series blended to simulate land cover change with the exact time index known of change in each province is given in table 8.1. The effective change detection delay is reported in days in section 8.7.3.

The change detection algorithms are then applied to the complete province in section 8.9. The total number of time series in each province is given in table 8.1. The entire province is classified and areas which experienced land cover change are mapped, followed by the calculation of summary statistics.

8.5 PARAMETER EXPLORATION

8.5.1 Optimising the multilayer perceptron

The MLP comprises an input layer, one hidden layer and an output layer. All hidden and output nodes used a tangent sigmoid activation function. The input layer accepts feature vectors for classification, while the output layer represents the likelihood that an input belongs to a specific class. The MLP output was in the range [-1;1], where 1 represents a 100% certainty of class membership to class 1 (natural vegetation) given the feature vector, while -1 represents a 100% certainty of class 2 (settlement).

The weights of the MLP were determined using a steepest descent gradient optimisation method in the training phase, with gradients estimated using backpropagation [130, Ch. 4 p. 140]. A validation set was used for initial MLP architecture optimisation by evaluating the generalisation error to identify overfitting of the network for each study area. The MLP architecture was optimised for different lengths of sliding window Q, number of spectral bands and training mode. In table 8.2 the number

TABLE 8.2: The number of hidden nodes used within the MLP for each experiment.

Province	Algorithm	Window length	Spectral Band		
			NDVI	2 Bands	7 Bands
Limpopo	SFF, Iteratively retrained	6 months	7	6	6
		12 months	8	10	9
		18 months	8	9	7
	SFF, Batch mode	12 months	8	10	9
	Least squares	12 months	9	8	11
	M-estimator	12 months	9	10	7
	EKF_{BVEP}	n/a	7	5	5
	EKF_{ALS}	n/a	15	13	11
Gauteng	SFF, Iteratively retrained	6 months	8	8	7
		12 months	7	7	8
		18 months	7	6	5
	SFF, Batch mode	12 months	7	7	8
	Least squares	12 months	8	10	5
	M-estimator	12 months	11	10	9
	EKF_{BVEP}	n/a	9	4	2
	EKF_{ALS}	n/a	14	6	5

of hidden nodes used in each experiment are reported. The learning rate was set to 0.01 and the momentum parameter was set to 0.9. The maximum number of epochs in each training phase was set to 10000, and used the generalisation error on the validation set as an early stopping criterion.

8.5.2 Batch mode versus iterative retrained mode

In this section the notion of an iterative retrained training mode is explored and is compared to a classical batch training mode. The change detection method extracts feature vectors sequentially from a time series using a temporal sliding window. These feature vectors must be processed to yield a class label for each feature vector.

A MLP operating on the SFFs extracted from the temporal sliding window was used to explore the difference in classification accuracies between the batch mode and iteratively retrained mode. In the batch mode [130, Ch. 7 p. 263] all the incremental sliding windows between the year 2000 and the year 2008 were used as initial training inputs to the MLP. The experiments were conducted for the 8 years without any retraining.

The iteratively retrained MLP is proposed to compensate for the inter-annual variability between years due to the rainfall variability. The iteratively retrained MLP is trained to recognise data from

Table 8.3: Classification accuracy of the batch mode and iteratively retrained MLP on the validation set. Each entry gives the average classification accuracy for each mode, calculated over 10 repeated independent experiments along with the corresponding standard deviation. The average classification accuracy is given in percentage for each of the classes over a temporal sliding window length of 12 months and different sets of spectral band combinations (NDVI, 2 spectral bands and all 7 spectral bands).

Province	**Spectral Band**	**Class**	**Mode**	
			Batch mode	Iteratively retrained
Limpopo	NDVI	Vegetation	67.7 ± 9.5	72.8 ± 5.3
		Settlement	83.0 ± 4.9	83.2 ± 3.7
	2 Bands	Vegetation	80.5 ± 5.6	83.1 ± 4.1
		Settlement	87.2 ± 2.0	86.8 ± 2.7
	7 Bands	Vegetation	94.5 ± 2.1	94.4 ± 1.6
		Settlement	94.8 ± 1.2	95.2 ± 1.1
Gauteng	NDVI	Vegetation	94.6 ± 4.1	96.2 ± 2.0
		Settlement	82.3 ± 8.9	88.0 ± 6.3
	2 Bands	Vegetation	96.6 ± 1.4	96.7 ± 1.6
		Settlement	92.2 ± 3.2	95.6 ± 2.3
	7 Bands	Vegetation	97.2 ± 0.4	99.8 ± 0.3
		Settlement	95.7 ± 0.4	99.3 ± 0.7

the training set within the sliding window at position p in the time series, and is then used to classify the data from the testing set within the sliding window at position p. This retraining at each time increment caused a small adaptation of the weights, and has low complexity because of the small incremental MLP weight changes over each 8 day increment of MODIS. These small MLP weight changes only required 300 epochs at each time increment for network adaptation.

The iteratively retrained mode provided slightly higher mean classification accuracies when compared to the classical batch training mode. The reason why the iteratively retrained mode performed better than the batch mode (table 8.3) is that the iteratively retrained mode had the advantage of learning the most recent spectral properties of the land cover types, as time progressed. The iteratively retrained mode takes cognisance of what is within the temporal sliding window to compensate for short-term inter-annual climate variability and adapts to longer term trends in climate without confusing any of these with a particular land cover type, which has often been a problem with other regional land cover studies [218, 219]. It should be noted that these benefits of using the iteratively retrained mode comes at the cost of having shorter predictive spans, as predicting future events will require retraining with an training data set that is unavailable. The benefits of using iteratively retrained mode resulted in it being used in the remainder of this chapter.

8.5.3 Optimising least squares

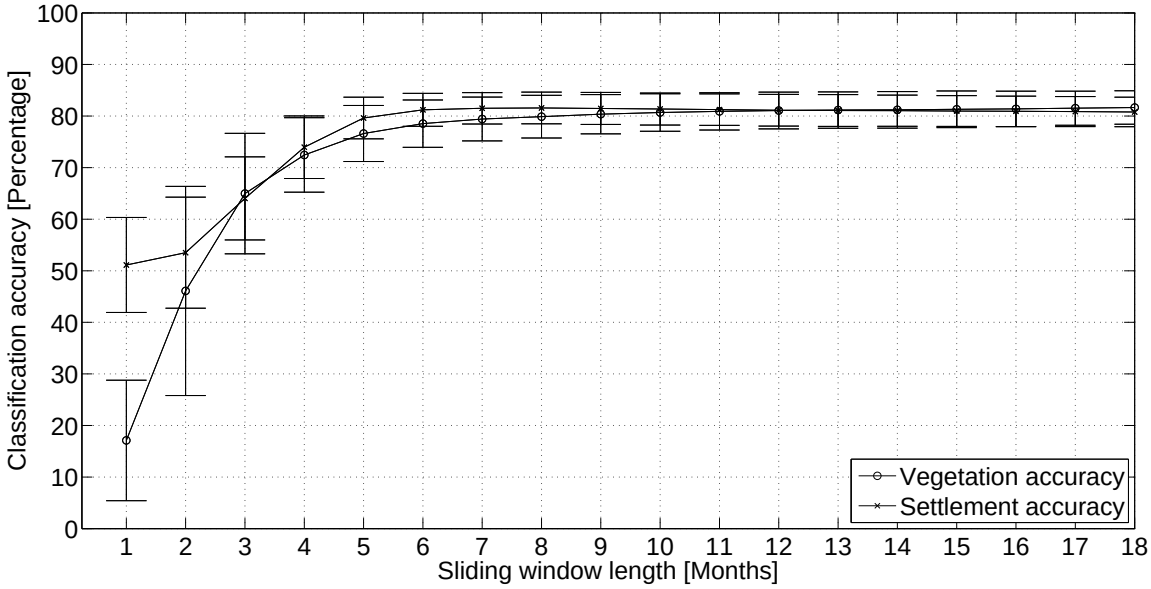
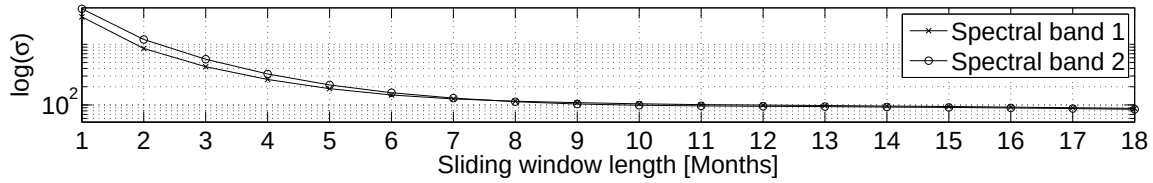


FIGURE 8.7: Classification accuracy reported by the K-means algorithm using the model fitted with a least squares model approach. The average classification accuracy is measured in percentage for each of the classes over a range of temporal sliding window length.

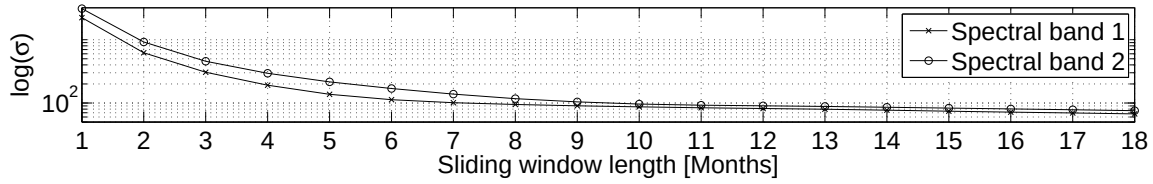
In this section an experiment was conducted to determine the optimal length of the sliding window when using the least squares approach to fit a model. The model is a triply modulated cosine model and the estimated parameters are used by a machine learning method for classification and change detection. The sliding window length was evaluated against classification accuracy, the model parameters' standard deviation and residuals of the fitted model. The classification accuracies were computed using the K-means algorithm operating on the first two spectral bands that were extracted from the Limpopo province study area. In figure 8.7, the classification accuracies are plotted as a function of the sliding window length, which is reported in the number of months.

It was observed that the settlement classification accuracy stabilised above 80% when the sliding window length surpassed the 5 month mark. The vegetation classification accuracy only stabilised above 80% after the sliding window had a length longer than 9 months. Similar classification accuracies and corresponding standard deviations were observed for both classes when the sliding window length increased beyond 11 months.

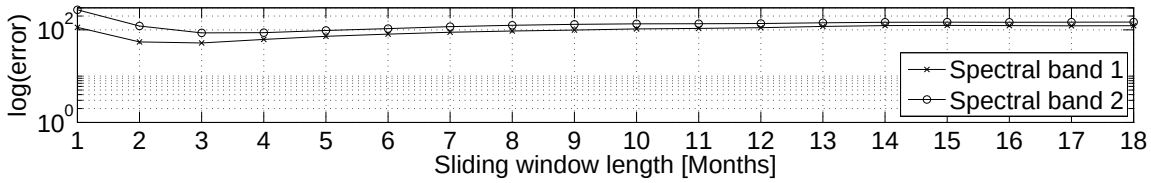
The model parameters' standard deviation for both the mean and amplitude parameters are shown in figure 8.8(a) and figure 8.8(b) respectively. It was observed that the model parameters' standard deviation for both the mean and amplitude parameters reduced as the length of the sliding window was increased. The mean parameter's standard deviation for both spectral bands started to decrease more



(a) Mean parameter



(b) Amplitude parameter



(c) Absolute error

FIGURE 8.8: The standard deviation for the mean and amplitude parameter are illustrated in (a) and (b) when using a least squares approach to fit a triply modulated cosine model to the first two spectral bands of MODIS. The absolute error between the fitted model and the actual MODIS time series is shown in (c).

slowly when the sliding window length was longer than 9 months. The amplitude parameter's standard deviation for both spectral bands started to decrease more slowly when the sliding window length was longer than 10 months.

The opposite was observed with the absolute error, which measures the difference between the fitted model and the actual MODIS time series. A shorter sliding window length had a smaller measured residuals, except if the window was too short and was severely affected by the additive noise in the MODIS time series. A sliding window of 2–3 months had the smallest measured residuals (figure 8.8(c)).

The length of the sliding window was determined based on the classification accuracies, owing to the inverse relationship between the standard deviations of the model's parameters and the absolute error. On the basis of this experiment it was decided to set the sliding window length to 12 months for all experiments using least squares to fit a model. The similarity between the results produced by the least squares and M-estimator supports the choice of a 12 month window for the M-estimator too. No significant variations in the parameter vector were found when sliding the window through the time series and using the least squares or the M-estimator.

8.5.4 BVEP versus autocovariance least squares

Table 8.4: Classification accuracy of the MLP using either the BVEP criterion or the ALS approach to fine tune the parameters of the Extended Kalman filter. Each entry gives the average classification accuracy for each mode, calculated over 10 repeated independent experiments along with the corresponding standard deviation. The average classification accuracy is given as a percentage for each of the classes over a number of spectral band combinations (NDVI, 2 spectral bands and all 7 spectral bands).

Province	**Spectral Band**	**Class**	**Mode**	
			EKF_{ALS}	EKF_{BVEP}
Limpopo	NDVI	Vegetation	66.6 ± 9.1	80.2 ± 4.4
		Settlement	79.2 ± 6.2	82.7 ± 3.7
	2 Bands	Vegetation	79.3 ± 2.7	87.2 ± 1.6
		Settlement	85.9 ± 2.1	89.7 ± 1.3
	7 Bands	Vegetation	86.6 ± 3.7	95.3 ± 0.7
		Settlement	90.6 ± 1.9	96.1 ± 0.6
Gauteng	NDVI	Vegetation	89.3 ± 4.8	91.4 ± 5.7
		Settlement	72.1 ± 16.9	86.9 ± 9.1
	2 Bands	Vegetation	90.6 ± 2.9	98.6 ± 1.0
		Settlement	87.6 ± 3.2	96.2 ± 1.5
	7 Bands	Vegetation	95.3 ± 1.8	99.9 ± 0.1
		Settlement	94.8 ± 2.4	99.9 ± 0.1

In this section two different methods used for setting the parameters of the EKF are investigated. The first method that is investigated is the ALS method discussed in section 7.3. The second method investigated is the BVEP criterion approach discussed in section 7.2.4.

In table 8.4, the classification accuracies for both provinces are reported when the EKF is used to extract the features. The average classification accuracy is calculated with cross-validation using 10 repeated independent experiments [127]. From these results it was concluded that the EKF_{BVEP} performed better than any experiment conducted using the EKF_{ALS}. This could be owing to the fact that the BVEP criterion utilises spatial information that is inherent in the set of time series.

8.5.5 Optimisation of Kalman filter parameters

In this section the results obtained by using the BVSA are discussed. The BVSA is an iterative algorithm that moves the BVS through a defined space. In each epoch the algorithm attempts to minimise the standard deviation of all the state space variables while simultaneously minimising the residual between the triple modulated cosine function's output and the actual observations.

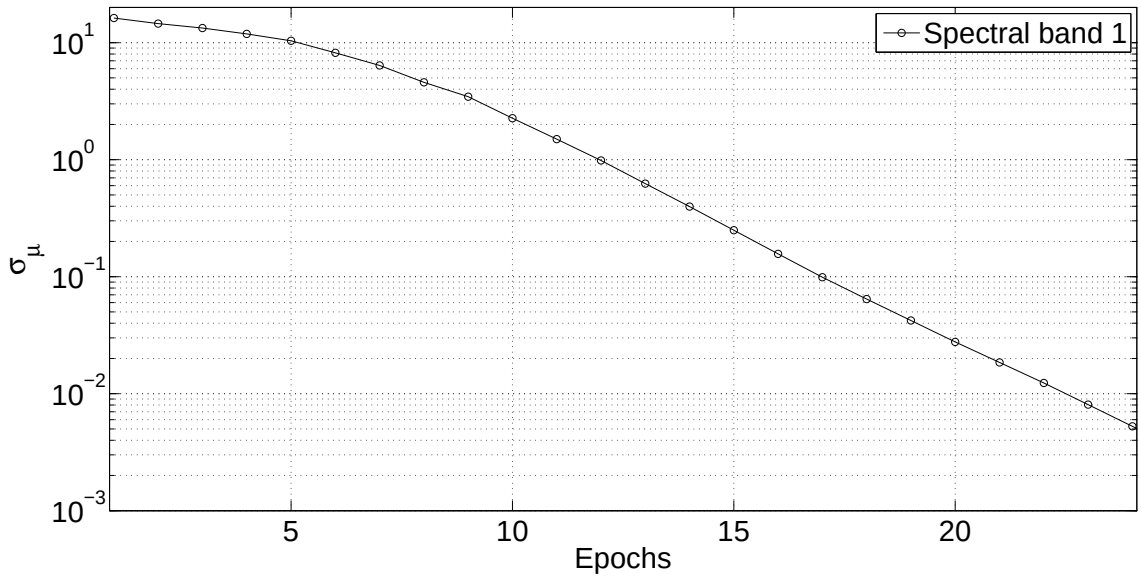


FIGURE 8.9: The expected standard deviation of the mean parameter computed for the first MODIS spectral band on the Limpopo province study area as a function of epoch.

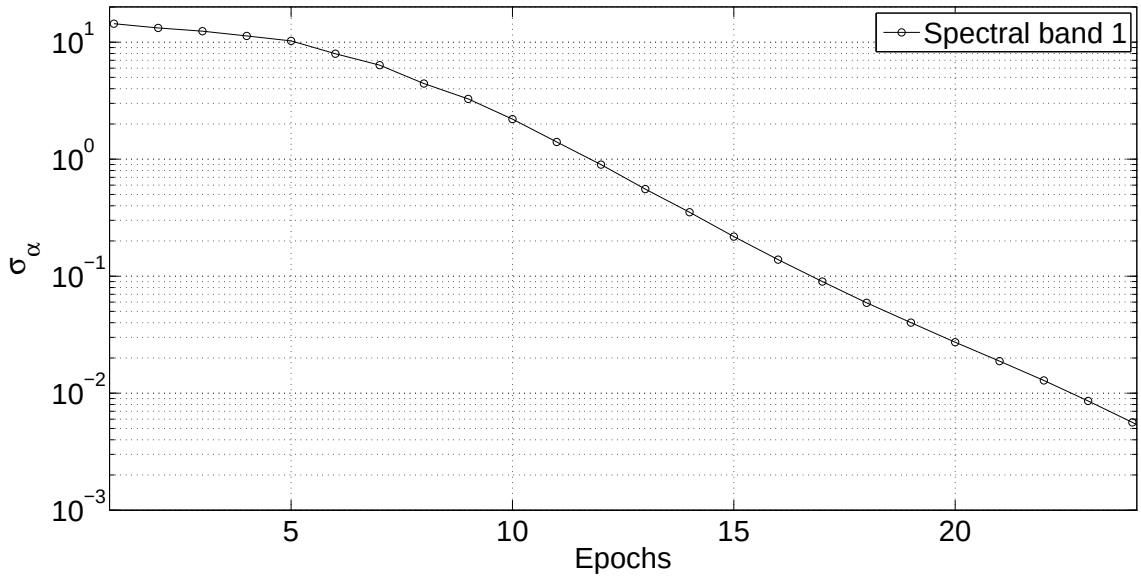


FIGURE 8.10: The expected standard deviation of the amplitude parameter computed for the first MODIS spectral band on the Limpopo province study area as a function of epoch.

In figure 8.9, the standard deviation σ_μ of the mean parameter obtained by fitting the cosine model to the first MODIS spectral band is illustrated as a function of epoch in the BVSA. The standard deviation reported here is the average standard deviation found over all the time series extracted from the Limpopo province study area. It is clear from the graph that the standard deviation decreases as more epochs are processed, which implies that the mean parameter appears to become more stable with each iteration.

The standard deviation σ_α of the amplitude parameter that is used to fit the first MODIS spectral band is illustrated as a function of epoch of the BVSA in figure 8.10. The standard deviation reported here is the average standard deviation found over all the time series extracted from the Limpopo province study area. It is clear from the graph that the standard deviation decreases as more epochs are processed, implying increasing stability with further iterations.

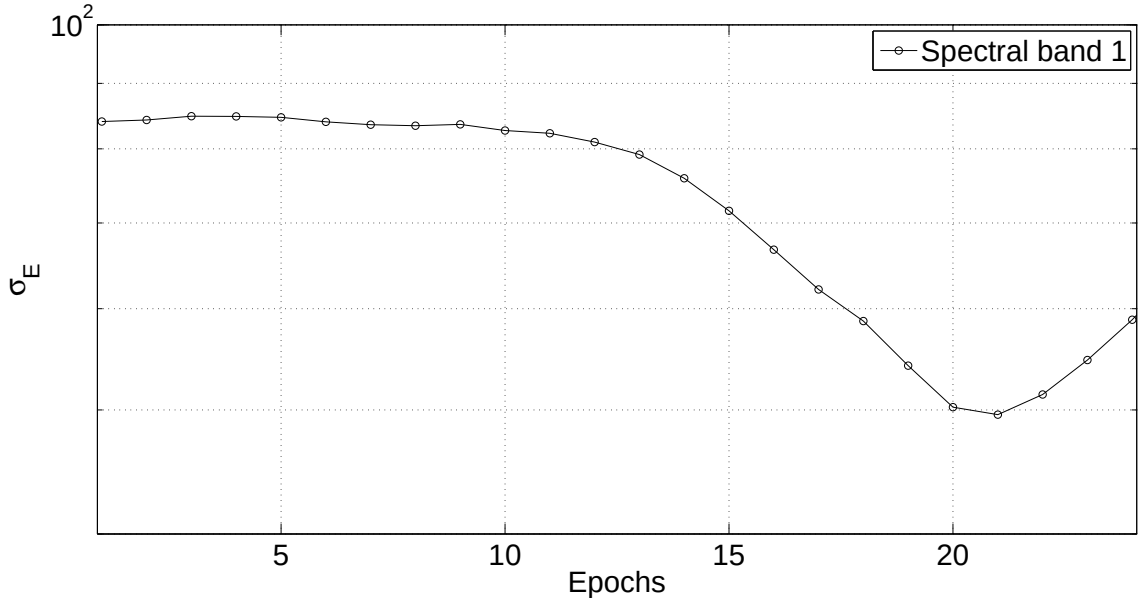


FIGURE 8.11: The expected residuals computed for the first MODIS spectral band on the Limpopo province study area as a function epoch.

In figure 8.11, the mean residual $\sigma_\mathcal{E}$ over all the time series' difference between the actual observations and EKF output is illustrated as a function of epoch in the BVSA. It is observed that the residual decreases significantly after the 10^{th} epoch. Overfitting appears towards the end of the optimisation process. This overfit can occur on any metric and in this experiment the overfit is observed on the $\sigma_\mathcal{E}$ metric after the 21^{st} epoch. This overfit defines the end of the search and is used as an early stopping criterion.

Table 8.5: Parameter evaluation of two different search methods that were compared in the Limpopo province study area.

Algorithm	**Parameter evaluation**		
	σ_μ	σ_α	$\sigma_\mathcal{E}$
Simulated Annealing	14.5	12.6	94.6
BVSA	0.04	0.02	87.1

The process covariance matrix $\mathcal{Q}$ and observation covariance matrix $\mathcal{R}$ used in the 21^{st} epoch are then used to initialise the EKF for the experiments. The BVSA is applied independently to each of the

seven spectral bands and NDVI time series to obtain a process covariance matrix $\mathcal{Q}$ and observation covariance matrix $\mathcal{R}$ for each spectral band.

Table 8.6: Parameters evaluation of all four methods for the Limpopo province study area. The measurements are made on all seven MODIS spectral bands and NDVI.

Province	**Spectral Band**		**Mode**			
			Least squares	M-estimator	EKF_{ALS}	EKF_{BVEP}
Limpopo	NDVI	$\sigma_{\mathcal{E}}$	0.04	0.04	0.001	0.03
		σ_{μ}	0.02	0.01	0.04	0.02
		σ_{α}	0.02	0.02	0.05	0.001
	Band 1	$\sigma_{\mathcal{E}}$	118.6	118.7	144.0	87.1
		σ_{μ}	28.8	28.1	29.8	0.04
		σ_{α}	36.4	36.1	21.8	0.02
	Band 2	$\sigma_{\mathcal{E}}$	145.2	144.7	179.9	95.7
		σ_{μ}	38.5	37.4	29.6	0.01
		σ_{α}	56.4	57.6	25.2	0.36
	Band 3	$\sigma_{\mathcal{E}}$	58.1	58.0	62.3	47.9
		σ_{μ}	13.6	13.1	20.9	0.06
		σ_{α}	18.9	18.3	14.7	0.05
	Band 4	$\sigma_{\mathcal{E}}$	65.6	65.6	81.0	58.3
		σ_{μ}	14.2	14.1	25.5	0.05
		σ_{α}	19.7	20.8	18.0	0.04
	Band 5	$\sigma_{\mathcal{E}}$	154.6	154.3	171.1	97.3
		σ_{μ}	36.7	36.2	29.6	0.01
		σ_{α}	48.6	49.1	24.9	0.01
	Band 6	$\sigma_{\mathcal{E}}$	198.5	198.4	242.4	166.9
		σ_{μ}	46.6	45.8	33.8	0.01
		σ_{α}	67.8	68.1	27.3	0.01
	Band 7	$\sigma_{\mathcal{E}}$	232.1	232.0	302.0	201.1
		σ_{μ}	79.3	76.5	31.3	0.02
		σ_{α}	77.9	76.4	26.1	0.03

It should be noted that other optimisation algorithms were also explored, based on the objective function defined in the BVEP criterion (equation (7.50)) to evaluate the performance of the BVSA. The algorithms used to set the BVS are: (1) the interior point method [220], (2) active set method [221], and (3) simulating annealing [222]. It is observed from the active set method that larger and more aggressive step sizes are required, which supports the BVSA described on page 135. Simulated annealing (500 epochs, 5 function evaluations per epoch) produced better results than either the active set method or the interior point method. Table 8.5 compares simulated annealing to BVSA.

By evaluating the propagation direction of the simulating annealing method, it was concluded that

the method would eventually find the same solution identified by the BVSA, and yield the exact same performance. The advantage of the BVSA was the speed of convergence, which is attributed to the fact that it only requires a single function evaluation per epoch and converged in 21 epochs in this experiment.

8.5.6 BVSA parameter evaluation

Table 8.7: Parameters evaluation of all four methods for the Gauteng province study area. The measurements are made on all seven MODIS spectral bands and NDVI.

Province	**Spectral Band**		**Mode**			
			Least squares	M-estimator	EKF_{ALS}	EKF_{BVEP}
Gauteng	NDVI	$\sigma_{\mathcal{E}}$	0.04	0.04	0.001	0.003
		σ_{μ}	0.01	0.01	0.07	0.05
		σ_{α}	0.009	0.01	0.06	0.01
	Band 1	$\sigma_{\mathcal{E}}$	96.6	96.6	90.8	44.8
		σ_{μ}	17.7	17.4	21.3	0.01
		σ_{α}	22.5	22.2	17.3	15.3
	Band 2	$\sigma_{\mathcal{E}}$	156.4	155.9	204.2	123.4
		σ_{μ}	49.1	47.2	29.8	0.01
		σ_{α}	54.9	55.3	25.5	0.5
	Band 3	$\sigma_{\mathcal{E}}$	55.1	55.1	46.7	38.5
		σ_{μ}	10.2	9.8	14.9	0.03
		σ_{α}	14.0	13.5	12.2	0.02
	Band 4	$\sigma_{\mathcal{E}}$	63.3	63.3	57.0	42.7
		σ_{μ}	12.6	12.6	19.2	0.04
		σ_{α}	14.7	15.4	14.5	0.03
	Band 5	$\sigma_{\mathcal{E}}$	153.2	153.0	162.9	105.3
		σ_{μ}	47.4	46.2	26.6	0.01
		σ_{α}	54.2	53.8	22.6	0.01
	Band 6	$\sigma_{\mathcal{E}}$	157.3	157.4	130.5	87.3
		σ_{μ}	29.8	30.0	24.9	0.01
		σ_{α}	34.8	36.6	22.2	0.01
	Band 7	$\sigma_{\mathcal{E}}$	158.0	157.8	151.9	71.9
		σ_{μ}	27.8	27.0	23.0	0.02
		σ_{α}	35.0	34.3	21.7	20.5

In this section the derived parameters for each regression method are compared along with the residuals. The comparison is based on the standard deviation σ_{μ} of the mean parameter, the standard deviation σ_{α} of the amplitude parameter, and the residuals $\sigma_{\mathcal{E}}$. A mean (amplitude) parameter with a small standard deviation indicates a stable variable. A small $\sigma_{\mathcal{E}}$ indicates a well-estimated output when

compared to the actual observations.

An analysis of the standard deviation of the parameters extracted from the Limpopo province data is presented in table 8.6. It was observed that the M-estimator generally performs similarly to least squares, and in some cases performed slightly better. The EKF_{ALS} method generally increased the residuals to improve the parameter stability when compared to the M-estimator. The EKF_{BVEP} outperformed all the methods in all the experiments, except for the NDVI experiments. The EKF_{BVEP} however did yield comparable results to the other methods in the NDVI experiments.

In table 8.7, the same comparison was made as in table 8.6 for the Gauteng province study area. The M-estimator again performed similar to the least squares and in a few experiments performed slightly better. The relation between the EKF_{ALS} method and M-estimator did not hold in the Gauteng province study area. The EKF_{ALS} method increased its residuals in spectral bands 2 and 5 to improve the parameters' stability when compared to the M-estimator. In spectral bands 1, 3 and 4 the mean parameter's standard deviation σ_μ was increased to improve the other two metrics. In spectral bands 6 and 7, EKF_{ALS} outperformed the M-estimator in all the metrics. In the NDVI case the EKF_{ALS} decreased its residuals at the cost of parameter stability when compared to the M-estimator.

The EKF_{BVEP} outperformed all methods in all the experiments, except for the NDVI experiments. A peculiar observation was made for the EKF_{BVEP} in spectral bands 1 and 7. For the first spectral band case overfitting was observed in the amplitude parameter early in the BVSA, which is used as an early stopping criterion. For the seventh spectral band case the standard deviation σ_α of the amplitude parameter slowly monotonically decreased for each epoch of the BVSA until an overfit was reported on the residuals $\sigma_\mathcal{E}$ at the 22^{nd} epoch. If the overfit did not occur, the standard deviation σ_α of the amplitude parameter would still steadily decrease. In the remainder of the chapter only the optimised EKF using the BVEP criterion (EKF_{BVEP}) will be considered and will be referred to as the EKF method.

8.5.7 Determining the number of clusters

Determining the number of clusters is one of the most difficult design considerations. The number of clusters K must be determined that provides maximum compression of information in the feature vectors with minimal error in classification on the data set.

The average silhouette value $\mathcal{S}_{\text{ave}}$ (equation (4.31) on page 82) is the metric used to determine the number of clusters. The nature of selecting only natural vegetation and human settlement areas in the labelled time series data set, and the resolution of the MODIS sensor, suggested a strong tendency of $\mathcal{S}_{\text{ave}}$ to have a high value at lower values of K. This is due to the fact that the labelled data set contains two distinct classes. At 500 metre resolution, the MODIS pixels are quite large, and are therefore

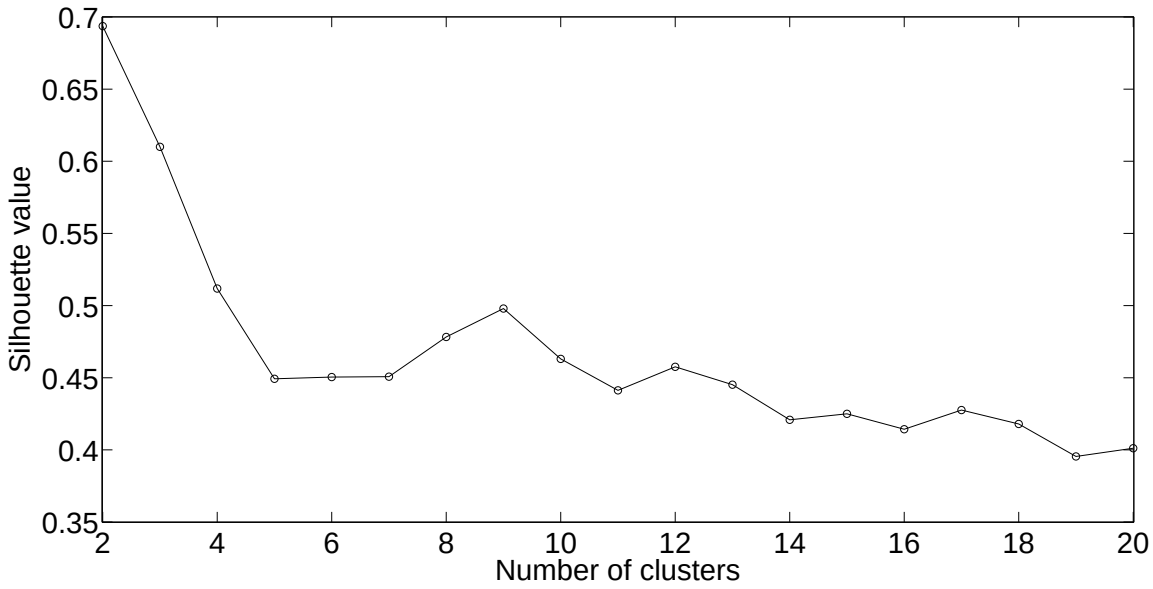


FIGURE 8.12: The average silhouette value $\mathcal{S}_{\text{ave}}$ computed over a range of different number of clusters in the Gauteng province.

likely to contain a mixture of different vegetation types. Nevertheless, it is reasonable to assume that the variability within the broader vegetation class will be large enough to justify splitting the vegetation class into subclasses. This however was not the case in the labelled data sets in this study.

In figure 8.12, an experiment was performed to compute the average silhouette value $\mathcal{S}_{\text{ave}}$ for a range of K. The experiment was conducted in Gauteng province using the EKF on the first two spectral bands. The feature vectors were then clustered using the K-means algorithm, followed by the computing of the silhouette values. The highest average silhouette value of 0.69 was recorded at two classes and steadily decreased as K increased. The experiment was repeated for all the other clustering methods, with K=2 producing the highest silhouette value in all the cases. The same experiments were conducted in the Limpopo province study area and yielded similar results.

8.5.8 Results: Cophenetic correlation coefficient

In this section the cophenetic correlation coefficient D_{cc} was computed for a range of hierarchical clustering methods: single linkage criterion (section 8.6.3), average linkage criterion (section 8.6.3), complete linkage criterion (section 8.6.3) and Ward clustering (section 8.6.4).

The cophenetic correlation coefficient evaluates how the created dendrogram retains the original placement of the feature vectors within the feature space. A high cophenetic correlation coefficient, $D_{\text{cc}} \rightarrow 1$, denotes that the distance representation is well preserved in the dendrogram. The cophenetic correlation coefficient was computed in the Limpopo province for a range of experimental

Table 8.8: The Cophenetic correlation coefficient computed for a range of hierarchical clustering methods on the Limpopo province's no change data set.

Algorithm	**Feature extraction**	**Window length**	**Spectral Band**		
			NDVI	2 Bands	7 Bands
Single linkage criterion	SFF	6 months	0.50	0.31	0.33
		12 months	0.51	0.32	0.33
		18 months	0.52	0.32	0.33
	Least squares	12 months	0.49	0.32	0.38
	M-estimator	12 months	0.49	0.32	0.39
	EKF	n/a	0.46	0.28	0.29
Average linkage criterion	SFF	6 months	0.59	0.64	0.61
		12 months	0.59	0.65	0.61
		18 months	0.59	0.65	0.62
	Least squares	12 months	0.60	0.62	0.61
	M-estimator	12 months	0.60	0.62	0.60
	EKF	n/a	0.59	0.62	0.59
Complete linkage criterion	SFF	6 months	0.64	0.64	0.62
		12 months	0.64	0.65	0.63
		18 months	0.64	0.66	0.63
	Least squares	12 months	0.60	0.61	0.62
	M-estimator	12 months	0.60	0.62	0.62
	EKF	n/a	0.62	0.63	0.64
Ward clustering	SFF	6 months	0.69	0.71	0.68
		12 months	0.69	0.72	0.68
		18 months	0.70	0.72	0.69
	Least squares	12 months	0.67	0.73	0.69
	M-estimator	12 months	0.67	0.73	0.69
	EKF	n/a	0.68	0.74	0.69

parameters (table 8.8): hierarchical clustering methods, feature extraction methods, and spectral band combinations.

A small improvement in the cophenetic correlation coefficient is observed when the sliding window length is increased. It is concluded that the cophenetic correlation coefficient is highly dependent on the clustering method used, as all feature extraction methods performed similarly when using a particular clustering method.

The single linkage criterion provided the lowest cophenetic correlation coefficients among the clustering methods. The average linkage criterion provided much better cophenetic correlation coefficients than the experiments using the single linkage criterion. A small improvement is observed

in the NDVI experiments when the complete linkage criterion is compared to the average linkage criterion. Similar results were observed for the average and complete linkage criteria in the two and seven spectral band experiments. A small improvement was observed in all the experiments when Ward clustering was used instead of the complete linkage criterion.

The same trend in cophenetic correlation coefficients was observed in the Gauteng province when all the experiments were compared to the results produced in the Limpopo province. The cophenetic correlation coefficient confirms the trend, which is observed in classification accuracies through sections 8.6.3–8.6.4. This is an important experiment, as this result was derived in an unsupervised manner, meaning the class labels for each time series were not used in the cluster process. It was concluded from the experiments conducted in this section that creating spherical clusters with minimum internal variance preserves the inherent distance between feature vectors within the feature space, which results in a higher cophenetic correlation coefficient.

8.6 CLASSIFICATION

8.6.1 Classification accuracy: Multilayer perceptron

Table 8.9: Classification accuracy of the MLP using SSFs on the no change data set. Each entry gives the average classification accuracy in percentage along with the corresponding standard deviation.

Province	Spectral Band	Class	Sliding window length		
			6 months	12 months	18 months
Limpopo	NDVI	Vegetation	69.7 ± 7.8	72.8 ± 5.3	73.9 ± 4.8
		Settlement	81.5 ± 5.0	83.2 ± 3.7	84.8 ± 3.1
	2 Bands	Vegetation	81.4 ± 4.3	83.1 ± 4.1	85.2 ± 3.7
		Settlement	86.3 ± 3.4	86.8 ± 2.7	88.1 ± 2.2
	7 Bands	Vegetation	93.1 ± 2.1	94.4 ± 1.6	94.7 ± 1.4
		Settlement	93.8 ± 1.6	95.2 ± 1.1	96.3 ± 0.9
Gauteng	NDVI	Vegetation	94.4 ± 3.7	96.2 ± 2.0	95.8 ± 2.2
		Settlement	79.5 ± 11.5	88.0 ± 6.3	88.5 ± 7.2
	2 Bands	Vegetation	95.1 ± 2.8	96.7 ± 1.6	97.2 ± 1.9
		Settlement	90.7 ± 6.7	95.6 ± 2.3	95.8 ± 2.5
	7 Bands	Vegetation	99.3 ± 0.7	99.8 ± 0.3	99.8 ± 0.3
		Settlement	98.1 ± 1.4	99.3 ± 0.7	99.6 ± 0.6

In this section the classification accuracies are evaluated for a MLP using a range of feature extraction methods. In table 8.9, the classification accuracies for both provinces are reported using SFFs. The average classification accuracy and corresponding standard deviation were calculated with

Table 8.10: Classification accuracy of the MLP using regression methods to extract features on the no change data set. Each entry gives the average classification accuracy in percentage along with the corresponding standard deviation.

Province	**Spectral Band**	**Class**	**Method**		
			Least squares	M-estimator	EKF
Limpopo	NDVI	Vegetation	72.5 ± 5.3	72.8 ± 5.4	80.2 ± 4.4
		Settlement	83.3 ± 3.4	84.6 ± 3.4	82.7 ± 3.7
	2 Bands	Vegetation	82.2 ± 4.3	83.1 ± 4.3	87.2 ± 1.6
		Settlement	86.4 ± 2.8	87.7 ± 2.5	89.7 ± 1.3
	7 Bands	Vegetation	92.5 ± 2.3	92.5 ± 1.9	95.3 ± 0.7
		Settlement	92.6 ± 1.2	92.4 ± 1.4	96.1 ± 0.6
Gauteng	NDVI	Vegetation	92.5 ± 4.9	93.1 ± 4.4	91.4 ± 5.7
		Settlement	88.6 ± 6.4	88.8 ± 6.0	86.9 ± 9.1
	2 Bands	Vegetation	97.5 ± 1.8	97.3 ± 1.9	98.6 ± 1.0
		Settlement	95.1 ± 2.6	94.9 ± 2.9	96.2 ± 1.5
	7 Bands	Vegetation	99.8 ± 0.4	99.9 ± 0.4	99.9 ± 0.1
		Settlement	99.2 ± 0.5	99.3 ± 0.9	99.9 ± 0.1

cross-validation using 10 repeated independent experiments. The accuracy is reported for each class over a range of temporal sliding window lengths (6, 12 and 18 months) and different spectral band combinations (NDVI, 2 spectral bands and all 7 spectral bands).

It is observed that a longer sliding window has a higher classification accuracy in all the experiments, as well as a reduction in standard deviations. Overall, the trend was that the classification performance improved for a longer sliding window. Another trend that was observed was an increase in overall performance when more spectral bands were used as input to a MLP classifier. This is supported by a higher classification accuracy for the first two spectral bands when compared to the NDVI, and the highest classification accuracy was reported for all seven spectral bands.

In table 8.10, the classification accuracies for both provinces are reported using regression methods to extract the features. The regression methods attempted to fit a triply modulated cosine function to the MODIS time series. The sliding window length was set to 12 months for both the least squares and M-estimator approaches. A similar improvement is observed as in table 8.9 when more spectral bands are used in the experiments.

From all the experiments it was concluded that a significant improvement is obtained when using the first two spectral bands rather than the NDVI. A further improvement was observed when the MLP operated on all seven spectral bands. The experiments conducted in the section are repeated in the following sections using different clustering algorithms.

8.6.2 Clustering experimental setup

In the following sections (section 8.6.3–8.6.4), different clustering approaches are analysed in a range of experiments. The first set of experiments conducted in each section is the measurement of the classification accuracy of the labelled time series using SFFs. The experiments were conducted for three different lengths of sliding window: 6 months (23 MODIS samples), 12 months (46 MODIS samples), and 18 months (69 MODIS samples). The experiments also explore the use of different spectral bands: NDVI, the first two spectral bands, and all seven spectral bands. In each experiment the classification accuracy along with the standard deviation is reported for the two classes: natural vegetation and human settlement.

The class labels in the experiments are assigned to minimise the overall error. This is accomplished in the Limpopo province by assigning the cluster containing majority of the feature vectors to the settlement class, as there are more settlement class time series than vegetation class time series (table 8.1). In the experiments conducted in the Gauteng province, the cluster containing majority of the feature vectors is assigned to the vegetation class, as there are more vegetation class time series than settlement class time series (table 8.1).

The second set of experiments conducted in each section is the measurement of classification accuracies of the labelled time series using different regression methods to extract features. The experiment is conducted on three different regression methods: least squares model fitting, M-estimator model fitting, and EKF. The experiments were also conducted to explore the use of different spectral bands in the similar method as in the first set of experiments. In each experiment the classification accuracy along with the standard deviation is reported for the two classes. The class labels are again assigned to minimised the overall error.

8.6.3 Clustering accuracy: Single, Average and Complete linkage criterion

In this section the viability of using hierarchical clustering based on the single, average and complete linkage criteria are investigated. Table 8.11 shows the classification accuracy on the experiments conducted using the SFFs, which were clustered based on the single, average and complete linkage criteria.

It is clear from the experiments that the first two spectral band outperforms NDVI.The first two spectral band also offered a slight improvement over the all seven spectral band. It is important to note that the all seven spectral band feature vector already encapsulate the first two spectral band. The reason for the decrease in classification accuracy is attributed to the fact that the seven spectral band feature vector requires more clusters (number of clusters K must increase) to cater for the increase in feature dimensionality. It was observed in an independent experiment that the classification accuracy

Table 8.11: Classification accuracy of a hierarchical clustering algorithm using the single, average and complete linkage criteria with the SFFs on the no change data set. Each entry gives the average classification accuracy in percentage along with the corresponding standard deviation for a sliding window length of 12 months.

Province	**Spectral Band**	**Class**	**Sliding window length**		
			Single linkage	Average linkage	Complete linkage
Limpopo	NDVI	Vegetation	45.8 ± 26.7	46.2 ± 25.7	52.1 ± 28.8
		Settlement	70.3 ± 21.1	71.0 ± 18.9	67.1 ± 21.9
	2 Bands	Vegetation	72.1 ± 16.7	76.4 ± 17.6	78.8 ± 15.9
		Settlement	80.0 ± 10.1	83.5 ± 9.5	85.7 ± 11.3
	7 Bands	Vegetation	71.4 ± 17.0	76.5 ± 25.2	75.5 ± 19.1
		Settlement	77.5 ± 9.9	83.0 ± 12.8	80.6 ± 24.0
Gauteng	NDVI	Vegetation	60.9 ± 18.2	65.3 ± 11.2	64.8 ± 9.9
		Settlement	36.9 ± 25.4	40.8 ± 21.8	42.1 ± 20.0
	2 Bands	Vegetation	80.1 ± 16.1	82.8 ± 14.8	81.6 ± 11.7
		Settlement	66.4 ± 35.1	67.0 ± 33.8	69.2 ± 29.4
	7 Bands	Vegetation	79.2 ± 16.3	80.2 ± 15.1	80.5 ± 12.2
		Settlement	64.4 ± 34.2	64.8 ± 34.1	65.9 ± 30.1

rapidly improves for the seven spectral band case if K is larger than 10. The number of clusters was not increased as the objective of the use of the unsupervised classifier is to evaluate a completely unsupervised change detection method. A supervised algorithm must then be applied onto the clusters if more clusters are included.

The first two spectral band experiments offered acceptable performance in both provinces. It should be noted that these classification accuracies could only be obtained with these three hierarchical clustering methods when performing proper outlier removal. The outliers were identified by applying principle component analysis to the feature vectors and calculating the Hotellier T^2 distance between the principal components and each of the transformed feature vectors. The outliers were then selected with distances exceeding a predefined threshold. The other clustering methods did not require the removal of outliers and for this reason the single linkage, average linkage and complete linkage criteria will not be further evaluated in this chapter.

8.6.4 Clustering accuracy: Ward clustering method

In this section the viability of using the Ward clustering method is investigated. Table 8.12 and table 8.13 show the results for the experiments that were produced using the Ward clustering method.

The Ward clustering method provided no acceptable classification accuracies when clustering on the NDVI time series. The Ward clustering method did however provide reasonable classification

Table 8.12: Classification accuracy of the Ward clustering method using the SFFs on the no change data set. Each entry gives the average classification accuracy in percentage along with the corresponding standard deviation.

Province	**Spectral Band**	**Class**	**Sliding window length**		
			6 months	12 months	18 months
Limpopo	NDVI	Vegetation	45.3 ± 19.4	45.4 ± 17.5	46.3 ± 17.2
		Settlement	64.6 ± 12.8	66.3 ± 11.9	66.6 ± 11.7
	2 Bands	Vegetation	79.0 ± 14.2	80.9 ± 13.8	81.7 ± 13.4
		Settlement	78.2 ± 11.1	77.5 ± 10.2	77.3 ± 10.3
	7 Bands	Vegetation	72.4 ± 16.5	73.8 ± 15.6	73.8 ± 15.8
		Settlement	73.6 ± 11.9	74.5 ± 11.5	74.7 ± 11.1
Gauteng	NDVI	Vegetation	66.4 ± 10.8	67.4 ± 8.8	67.5 ± 8.7
		Settlement	35.2 ± 28.9	38.7 ± 28.6	38.9 ± 29.0
	2 Bands	Vegetation	81.3 ± 14.5	86.8 ± 13.1	86.8 ± 12.7
		Settlement	68.0 ± 31.9	69.8 ± 31.8	69.9 ± 32.0
	7 Bands	Vegetation	77.4 ± 15.6	78.2 ± 17.8	76.3 ± 18.3
		Settlement	24.5 ± 19.0	26.2 ± 18.7	27.9 ± 23.1

Table 8.13: Classification accuracy of Ward clustering with the regression methods to extract features on the no change data set. Each entry gives the average classification accuracy in percentage along with the corresponding standard deviation.

Province	**Spectral Band**	**Class**	**Method**		
			Least squares	M-estimator	EKF
Limpopo	NDVI	Vegetation	68.0 ± 16.4	68.8 ± 15.7	66.3 ± 16.5
		Settlement	78.8 ± 13.4	78.5 ± 13.4	77.5 ± 13.4
	2 Bands	Vegetation	79.9 ± 15.1	80.0 ± 15.0	85.7 ± 12.3
		Settlement	76.9 ± 11.1	76.9 ± 11.1	77.7 ± 10.9
	7 Bands	Vegetation	72.8 ± 17.5	72.8 ± 17.6	74.1 ± 14.9
		Settlement	72.8 ± 14.3	72.8 ± 14.2	75.4 ± 9.3
Gauteng	NDVI	Vegetation	94.6 ± 10.8	94.7 ± 10.9	85.1 ± 12.1
		Settlement	27.9 ± 12.5	28.1 ± 12.9	36.9 ± 23.3
	2 Bands	Vegetation	84.5 ± 14.5	84.5 ± 14.5	88.7 ± 10.2
		Settlement	68.6 ± 32.1	68.8 ± 32.0	87.9 ± 14.3
	7 Bands	Vegetation	79.6 ± 17.3	79.6 ± 17.4	78.8 ± 18.0
		Settlement	27.5 ± 22.7	27.4 ± 22.6	44.0 ± 25.2

accuracies when the first two spectral bands and the all seven spectral bands were used in the Limpopo province. Classification accuracies of above 75% were reported for the first two spectral band experiments. The EKF features using the first two spectral bands yielded classification accuracies higher than 87.9% in the Gauteng province when compared to all the other regression methods, which

yielded classification accuracies below 70%.

In the seven spectral bands experiments an interesting trend was observed in all the hierarchical clustering experiments. The classification accuracies were lower in higher dimensions (7 spectral bands) than in lower dimensions (2 spectral bands). The question that was raised was whether the feature vectors became more separable in higher dimensions. The answer was confirmed with the MLP in section 8.6.1, where the MLP reported higher classification accuracies in the seven spectral band experiments when compared to the two spectral band experiments.

This reverts back to the statement made in section 4.2.2 on page 70 that clustering in a high-dimensional feature space usually provides meaningless results if proper design considerations are not followed [197, 198]. This is usually attributed to the notion that the ratio between the nearest neighbour and average neighbourhood distance rapidly converges to one in higher dimensions.

The remedy for this reduction in classification accuracy in the seven spectral band experiments is the implementation of a more complex clustering algorithm or a more in-depth feature selection criterion. The complex clustering algorithm will create non-linear mappings as with the MLP to obtain the desired classification accuracies. The shortcoming is the need to over design the clustering algorithm for a particular data set. Feature selection is the other approach that can be used to improve clustering performance, as it is used as a dimensionality reduction procedure, which uses fewer spectral bands to improve the performance. The problem is that different combinations of spectral bands will perform better on different data sets.

Based on the impossibility theorem, the emphasis is placed on obtaining acceptable performance in the clustering algorithm. As stated previously, the Ward clustering method does provide acceptable classification accuracies when using the first two spectral bands.

8.6.5 Clustering accuracy: K-means clustering

In this section the viability of using the K-means partitional clustering method is investigated. Table 8.14 and table 8.15 illustrate the classification accuracies for the experiments conducted with the K-means clustering algorithm.

The clustering of the NDVI time series using K-means provided acceptable classification accuracies when the regression method was used in the Limpopo province (table 8.15). This however was not the case in the Gauteng province, from which it can be concluded that the performance of clustering NDVI time series with K-means was unacceptable as it is only usable in the Limpopo province.

The first two spectral band experiments provided better classification accuracy performance when compared to any similar hierarchical clustering method. The EKF approach was deemed the best

Table 8.14: Classification accuracy of K-means with the SFFs on the no change data set. Each entry gives the average classification accuracy in percentage along with the corresponding standard deviation.

Province	**Spectral Band**	**Class**	**Sliding window length**		
			6 months	12 months	18 months
Limpopo	NDVI	Vegetation	53.2 ± 12.8	54.4 ± 8.3	54.8 ± 9.2
		Settlement	58.7 ± 7.1	59.9 ± 5.3	59.7 ± 7.3
	2 Bands	Vegetation	81.7 ± 4.7	82.9 ± 3.7	83.4 ± 3.5
		Settlement	81.4 ± 2.2	82.0 ± 2.4	81.8 ± 2.2
	7 Bands	Vegetation	75.8 ± 5.0	76.2 ± 4.6	76.3 ± 4.3
		Settlement	74.9 ± 2.8	75.2 ± 2.3	75.2 ± 2.1
Gauteng	NDVI	Vegetation	61.3 ± 8.0	63.1 ± 5.3	65.5 ± 6.7
		Settlement	42.3 ± 28.3	39.8 ± 30.2	38.9 ± 29.9
	2 Bands	Vegetation	85.1 ± 9.1	90.0 ± 7.3	90.4 ± 7.2
		Settlement	72.6 ± 19.4	70.9 ± 21.3	71.2 ± 21.7
	7 Bands	Vegetation	76.5 ± 13.2	77.3 ± 13.1	77.3 ± 13.4
		Settlement	38.7 ± 7.6	41.2 ± 6.8	41.6 ± 6.3

Table 8.15: Classification accuracy of K-means with the regression methods to extract features on the no change data set. Each entry gives the average classification accuracy in percentage along with the corresponding standard deviation.

Province	**Spectral Band**	**Class**	**Method**		
			Least squares	M-estimator	EKF
Limpopo	NDVI	Vegetation	69.9 ± 5.7	71.4 ± 5.7	70.5 ± 6.8
		Settlement	79.3 ± 3.5	81.2 ± 3.4	79.1 ± 4.7
	2 Bands	Vegetation	81.5 ± 3.5	81.5 ± 3.6	84.4 ± 0.2
		Settlement	80.7 ± 3.1	80.6 ± 3.0	82.3 ± 0.2
	7 Bands	Vegetation	76.7 ± 3.8	76.7 ± 3.7	76.3 ± 0.2
		Settlement	74.3 ± 2.8	74.5 ± 2.7	75.1 ± 0.1
Gauteng	NDVI	Vegetation	94.4 ± 5.2	94.4 ± 5.2	68.3 ± 14.2
		Settlement	29.2 ± 2.7	29.3 ± 2.6	39.9 ± 32.2
	2 Bands	Vegetation	87.2 ± 7.6	87.2 ± 7.6	92.3 ± 0.4
		Settlement	73.9 ± 20.1	73.9 ± 20.2	84.7 ± 2.2
	7 Bands	Vegetation	75.9 ± 12.5	76.0 ± 12.4	75.9 ± 1.9
		Settlement	24.5 ± 6.6	24.5 ± 6.6	33.2 ± 0.7

performing feature extraction method in view of the small standard deviation in classification accuracy.

A similar observation was made for the partitional clustering as for the hierarchical clustering when clustering in higher dimensions. A small decrease of 6% was measured in classification accuracy when the first two spectral band experiments were compared to the all seven spectral band experiments in

the Limpopo province. A large decrease of over 30% was measured in classification accuracy when comparing the same experiments in the Gauteng province. This suggested that the same approach as described in section 8.6.4 must be followed.

8.6.6 Clustering accuracy: Expectation-Maximisation

In this section the viability of using the EM clustering algorithm is investigated. Table 8.16 and table 8.17 illustrate the results for the experiments conducted with the EM clustering algorithm. It was concluded from the experiments that the K-means clustering algorithm and EM clustering algorithm perform similarly, as the experimental results were almost exactly the same.

Table 8.16: Classification accuracy of EM algorithm with the SFFs on the no change data set. Each entry gives the average classification accuracy in percentage along with the corresponding standard deviation.

Province	**Spectral Band**	**Class**	**Sliding window length**		
			6 months	12 months	18 months
Limpopo	NDVI	Vegetation	51.3 ± 12.8	52.4 ± 8.5	52.9 ± 11.7
		Settlement	58.7 ± 7.1	58.8 ± 6.5	57.7 ± 7.3
	2 Bands	Vegetation	80.7 ± 4.6	81.9 ± 3.7	81.4 ± 3.6
		Settlement	81.4 ± 2.2	81.1 ± 2.2	80.6 ± 2.1
	7 Bands	Vegetation	75.8 ± 5.0	76.3 ± 4.5	76.3 ± 4.3
		Settlement	75.0 ± 2.9	75.2 ± 2.3	75.2 ± 2.1
Gauteng	NDVI	Vegetation	61.3 ± 8.0	63.1 ± 5.3	65.5 ± 6.7
		Settlement	42.3 ± 28.3	39.8 ± 30.2	39.0 ± 29.9
	2 Bands	Vegetation	85.1 ± 9.1	90.0 ± 7.4	90.4 ± 7.2
		Settlement	72.6 ± 19.4	70.9 ± 21.1	71.2 ± 21.7
	7 Bands	Vegetation	76.5 ± 13.2	77.3 ± 13.2	77.3 ± 13.4
		Settlement	38.7 ± 7.6	41.2 ± 6.8	41.6 ± 6.3

The EM clustering algorithm did however have a slightly lower classification accuracy at a negligible increase in standard deviation in a few of the experiments. For this reason the K-means clustering algorithm was chosen for its lower computational complexity.

8.6.7 Summary of classification results

In this section the results of the classification accuracies for section 8.6 are summarised. The first classifier that was considered in this section was the supervised MLP, which had the advantage of modelling a non-linear relationship between the input and output vectors.

The prospect of detecting land cover change was confirmed as possible by either using the NDVI time series or the first two spectral bands time series of the MODIS data, as this was supported by

Table 8.17: Classification accuracy of EM algorithm with the regression methods to extract features on the no change data set. Each entry gives the average classification accuracy in percentage along with the corresponding standard deviation.

Province	**Spectral Band**	**Class**	**Method**		
			Least squares	M-estimator	EKF
Limpopo	NDVI	Vegetation	69.9 ± 5.9	71.3 ± 5.7	69.5 ± 6.9
		Settlement	79.3 ± 3.5	81.3 ± 3.4	79.0 ± 4.7
	2 Bands	Vegetation	81.5 ± 3.5	81.5 ± 3.5	84.3 ± 0.2
		Settlement	80.7 ± 3.1	80.6 ± 3.1	81.3 ± 0.2
	7 Bands	Vegetation	76.7 ± 3.8	76.8 ± 3.8	76.3 ± 0.2
		Settlement	74.5 ± 2.4	74.4 ± 2.5	75.0 ± 0.1
Gauteng	NDVI	Vegetation	94.4 ± 5.2	94.4 ± 5.2	68.3 ± 14.2
		Settlement	29.2 ± 2.6	29.3 ± 2.9	40.1 ± 31.2
	2 Bands	Vegetation	87.2 ± 8.4	87.2 ± 8.3	92.2 ± 0.4
		Settlement	73.1 ± 22.0	73.1 ± 22.0	83.9 ± 2.1
	7 Bands	Vegetation	75.8 ± 12.3	75.9 ± 12.5	75.8 ± 1.9
		Settlement	24.5 ± 6.8	24.4 ± 6.6	33.2 ± 0.7

the results in [223]. The classification accuracies produced by the MLP were however found to be the highest when using all seven spectral bands.

The MLP was deemed to be the best classifier in this chapter when the feature vectors were extracted with the EKF. Classification accuracies of 95.3% with a standard deviation of 0.7% for the vegetation class, and 96.1% with a standard deviation of 0.6% for the settlement class were reported in the Limpopo province. In the Gauteng province classification accuracies of 99.9% with a standard deviation of 0.1% for the vegetation class and 99.9% with a standard deviation of 0.1% for the settlement class were reported.

It should be noted that the MLP classifier can be replaced with a variety of other classifiers. The MLP performed the best of all the classifiers in this thesis, but like most other supervised machine learning methods, the MLP is dependent on a training set and is required to be robust to any errors occurring within the training set [14]. The drawback in the remote sensing field is that the training data set has to be created with the aid of high spatial resolution imagery, and because of the temporal component must be updated periodically. These periodic updates are a costly endeavour, which justifies the consideration of unsupervised classification methods.

An unsupervised classifier is usually designed by *learning from example*. Thus several clustering methods were evaluated to make deductions about the nature of the feature vectors in the feature space.

Acceptable performance was only obtained with the single, average and complete linkage criteria with proper outlier removal. The other clustering methods did not require the removal of outliers and

for this reason was not explored further.

Ward's clustering method produced the best results of all the hierarchical clustering methods. It was concluded from the experiments conducted that creating spherical clusters with minimum internal variance preserves the inherent distance between feature vectors in the feature space. The algorithm provided acceptable performance for all experiments conducted in the Limpopo province, with the exception that acceptable performance was only observed for the first two spectral band experiments in the Gauteng province.

K-means and EM clustering algorithms were investigated as representative partitional clustering methods, with both methods performing very similarly. The experiments showed empirically that the partitional clustering methods outperformed all the hierarchical clustering methods in the Limpopo province. The partitional clustering methods had the same outcome as the Ward clustering method in the Gauteng province, with similar poor performances in the NDVI- and seven spectral band experiments. The partitional clustering methods were deemed to be better than the Ward clustering method, as they presented classification accuracies with lower standard deviations. The K-means algorithm was the preferred partitional clustering method for its reduced computational complexity.

In the next section the change detection capabilities of the algorithms are explored. Only a few methods were explored, since the change detection in this chapter is based on a post-classification approach. The algorithms that provided acceptable classification performance, which will be explored in the next section, are:

1. the Multilayer perceptron,
2. the Ward clustering method, and
3. the K-means algorithm.

8.7 CHANGE DETECTION

8.7.1 Simulated land cover change detection

A simulated land cover change data set was created to assess the land cover change detection algorithm objectively. The time series data set is used to ensure that the change detection algorithm is able to detect a transition between classes, while analysing the transition.

In table 8.18, the first set of change detection experiments are shown that were conducted in the Limpopo province. All the viable classification approaches that yielded acceptable performance in section 8.6 are shown in these experiments. Each entry in table 8.18 gives the average change detection

Table 8.18: The land cover change detection accuracies are given on the simulated land cover change data set in the Limpopo province. Each entry gives the true positives in percentage (false positives in parentheses).

Algorithm	**Feature extraction**	**Window length**	**Spectral Band**		
			NDVI	2 Bands	7 Bands
MLP	SFF	6 months	69.2 (30.0)	77.6 (22.4)	90.5 (9.6)
		12 months	70.2 (29.5)	78.2 (21.3)	90.8 (9.4)
		18 months	71.9 (29.2)	78.7 (20.7)	91.0 (8.9)
	Least squares	12 months	68.4 (31.8)	77.5 (22.3)	90.0 (10.1)
	M-estimator	12 months	69.0 (31.1)	77.2 (23.4)	90.2 (10.0)
	EKF	n/a	70.0 (30.3)	79.8 (20.2)	91.7 (8.7)
Ward clustering	SFF	6 months	51.2 (50.5)	71.1 (25.7)	68.3 (30.5)
		12 months	52.4 (48.5)	71.6 (25.5)	68.7 (30.3)
		18 months	52.6 (42.8)	72.2 (24.5)	69.2 (30.1)
	Least squares	12 months	65.4 (33.7)	69.8 (27.9)	67.6 (32.1)
	M-estimator	12 months	65.8 (33.7)	70.1 (28.0)	67.7 (32.3)
	EKF	n/a	59.8 (38.1)	73.0 (22.2)	66.6 (30.8)
K**-means**	SFF	6 months	50.0 (46.8)	71.3 (26.8)	64.3 (33.7)
		12 months	52.7 (46.1)	72.6 (26.5)	65.0 (33.0)
		18 months	53.5 (40.4)	72.9 (24.5)	65.7 (33.7)
	Least squares	12 months	63.4 (36.1)	70.4 (29.8)	65.4 (35.8)
	M-estimator	12 months	63.5 (36.3)	70.6 (29.5)	65.4 (35.8)
	EKF	n/a	57.9 (42.0)	72.8 (22.7)	64.8 (33.8)

accuracies, with the corresponding false alarm rate in parentheses. The change detection accuracies (true positives) are measured on subset 1 and subset 2, which were discussed in section 8.2.4, and the false alarm rates (false positives) are measured on subset 3 and subset 4.

The worst performing experiment was the method that employs the NDVI time series. The overall change detection accuracies were well below 70%, with a reported false alarm rate higher than 30%. In the first two spectral band experiments, acceptable performance was measured across all the methods, with overall change detection accuracies of above 70%, and a reported false alarm rate usually below 26%.

The seven spectral band experiment yielded similar behaviour when compared to the results observed in the classification accuracies. The MLP (supervised classifier) performed exceptionally by reporting overall change detection accuracies above 90% and a false alarm rate below 10%. The unsupervised classifiers, Ward clustering and K-means, reported change detection accuracies which are lower in the higher dimensions (7 spectral bands) than in the lower dimensions (2 spectral bands).

Table 8.19: The land cover change detection accuracies are given on the simulated land cover change data set in the Gauteng province. Each entry gives the true positives in percentage (false positives in parentheses).

Algorithm	**Feature extraction**	**Window length**	**Spectral Band** NDVI	2 Bands	7 Bands
MLP	SFF	6 months	81.2 (16.3)	89.7 (11.1)	97.3 (2.7)
		12 months	83.8 (16.3)	91.8 (10.5)	98.5 (1.5)
		18 months	83.9 (16.4)	92.0 (8.9)	98.5 (1.4)
	Least squares	12 months	78.1 (20.2)	90.0 (13.4)	97.5 (3.4)
	M-estimator	12 months	80.1 (18.9)	90.2 (13.0)	97.6 (3.2)
	EKF	n/a	82.5 (14.0)	93.2 (8.4)	98.4 (1.3)
Ward clustering	SFF	6 months	27.7 (28.8)	77.6 (25.4)	32.6 (31.6)
		12 months	33.2 (31.5)	80.0 (21.6)	36.9 (35.1)
		18 months	35.6 (34.6)	81.1 (19.8)	39.3 (35.4)
	Least squares	12 months	24.5 (17.4)	78.9 (19.7)	33.5 (28.6)
	M-estimator	12 months	24.5 (17.0)	79.2 (19.4)	33.4 (28.7)
	EKF	n/a	25.1 (17.2)	86.1 (7.2)	42.7 (26.0)
K**-means**	SFF	6 months	37.2 (42.9)	77.2 (26.6)	50.4 (41.3)
		12 months	43.8 (41.6)	80.3 (23.4)	51.2 (46.9)
		18 months	45.9 (46.7)	80.4 (24.6)	55.8 (38.7)
	Least squares	12 months	28.6 (21.3)	74.6 (28.5)	50.6 (45.7)
	M-estimator	12 months	28.6 (21.3)	75.0 (28.3)	51.3 (45.4)
	EKF	n/a	36.1 (37.8)	83.8 (5.9)	50.7 (40.8)

The reduction in change detection accuracies can be attributed to the reduction in classification accuracies shown in section 8.6.4 and section 8.6.5. The remedy for this reduction in change detection accuracy in the seven spectral band experiment is again either a more complex clustering algorithm or a more detailed selection of features. The more complex clustering algorithm typically requires a non-linear clustering region to obtain higher change detection accuracies. It is reported in the literature that this shortcoming can typically be solved by over designing the clustering algorithm for a particular data set. The second approach to remedy this reduction is to apply dimensionality reduction, which implies selecting different combinations of spectral bands. The potential risk is that different combinations of spectral bands will perform better on different data sets.

The emphasis in this thesis is placed on obtaining acceptable performance with the clustering algorithm based on the impossibility theorem. Acceptable performance is reported for all methods employing the first two spectral bands, and exceptional performance is reported for the MLP employing all seven spectral bands.

In table 8.19, the second set of change detection experiments are shown that were conducted in the Gauteng province. The same setup is used in these experiments as in the experiments conducted in the Limpopo province. The best performing algorithms were the methods that employ the MLP. The overall change detection accuracies were above 80% with a false alarm rate below 17%. A significant increase in change detection accuracy is observed when the two spectral bands are evaluated when compared to the NDVI. Both the NDVI and two spectral bands' experiments uses the same spectral bands, which implies that using the two spectral bands separately is better.

The worst performing experiments were the methods that employed either the NDVI or all seven spectral bands with an unsupervised classifier. It was observed that experiments conducted with the first two spectral bands along with an unsupervised classifier yielded acceptable performance. The reported overall change detection accuracies were above 75% with a false alarm rate below 30%.

8.7.2 Real land cover change detection

Table 8.20: The land cover change detection accuracy on the real land cover change data set in the Limpopo province. Each entry gives the true positives in percentage (false positives in parentheses).

Algorithm	**Feature extraction**	**Window length**	**Spectral Band**		
			NDVI	2 Bands	7 Bands
MLP	SFF	6 months	65.4 (32.5)	75.1 (19.5)	84.8 (9.3)
		12 months	66.1 (28.2)	75.3 (18.9)	85.3 (7.9)
		18 months	68.0 (28.7)	76.0 (18.8)	85.3 (8.2)
	Least squares	12 months	64.8 (28.6)	73.8 (23.1)	84.3 (10.1)
	M-estimator	12 months	64.7 (29.9)	73.4 (22.8)	84.3 (9.9)
	EKF	n/a	64.2 (24.6)	78.6 (16.7)	86.8 (8.7)
Ward clustering	SFF	6 months	38.8 (44.7)	67.3 (26.7)	58.7 (35.5)
		12 months	40.3 (52.1)	70.7 (25.9)	63.0 (32.9)
		18 months	40.5 (50.3)	70.0 (25.2)	63.3 (32.6)
	Least squares	12 months	57.6 (36.8)	65.4 (29.0)	62.8 (32.8)
	M-estimator	12 months	57.0 (36.3)	65.4 (28.5)	62.2 (32.8)
	EKF	n/a	52.8 (41.7)	71.8 (26.4)	63.5 (31.1)
K**-means**	SFF	6 months	44.8 (41.1)	70.2 (25.8)	59.8 (29.8)
		12 months	46.0 (42.0)	70.5 (25.4)	60.6 (31.1)
		18 months	46.9 (42.3)	70.5 (25.4)	61.0 (31.4)
	Least squares	12 months	59.8 (37.3)	68.4 (31.1)	61.0 (32.0)
	M-estimator	12 months	59.0 (36.5)	69.0 (30.3)	61.5 (33.4)
	EKF	n/a	51.7 (40.1)	72.0 (24.4)	63.0 (29.9)

In this section, the real land cover change data set (section 8.2.2) is used to measure the performance of the land cover change detection algorithms. This data set is used to test the validity of the algorithms for real world applications [127].

In table 8.20, the first set of change detection experiments are reported that were conducted in the Limpopo province. In these experiments all the viable classifiers identified in section 8.6.7 are explored. Each entry in table 8.20 gives the change detection accuracies (true positives), with corresponding false alarm rates (false positives) in parentheses.

The worst performing methods were those that employed the NDVI spectral band. Overall change detection accuracies in these experiments were observed to be well below 70%. On the other hand, acceptable performance was reported across all the methods using the first two spectral bands, except for the unsupervised classifiers operating on the features extracted with the least squares, and M-estimator.

Table 8.21: The land cover change detection accuracy on the real land cover change data set in the Gauteng province. Each entry gives the true positives in percentage (false positives in parentheses).

Algorithm	**Feature extraction**	**Window length**	**Spectral Band**		
			NDVI	2 Bands	7 Bands
MLP	SFF	6 months	82.3 (20.5)	86.5 (9.8)	94.3 (2.2)
		12 months	82.3 (16.8)	90.0 (8.8)	95.1 (1.1)
		18 months	83.7 (15.3)	90.4 (8.9)	95.1 (1.0)
	Least squares	12 months	80.0 (16.7)	87.7 (11.8)	94.3 (2.5)
	M-estimator	12 months	80.0 (17.5)	87.7 (10.9)	92.9 (2.8)
	EKF	n/a	83.4 (17.0)	92.1 (9.9)	95.5 (1.6)
Ward clustering	SFF	6 months	15.8 (24.2)	80.1 (21.2)	28.7 (29.8)
		12 months	20.7 (27.0)	80.3 (21.5)	31.3 (30.1)
		18 months	21.2 (28.8)	80.3 (21.4)	31.3 (30.3)
	Least squares	12 months	18.8 (18.0)	78.0 (23.1)	29.7 (29.4)
	M-estimator	12 months	18.1 (17.7)	75.5 (22.2)	30.5 (29.6)
	EKF	n/a	17.8 (17.5)	82.3 (11.3)	38.8 (24.8)
K**-means**	SFF	6 months	32.9 (34.4)	79.2 (24.2)	40.9 (38.9)
		12 months	38.3 (35.1)	79.2 (24.1)	44.7 (42.0)
		18 months	36.0 (34.7)	80.8 (22.7)	46.2 (40.4)
	Least squares	12 months	24.3 (23.9)	75.1 (26.6)	42.3 (40.1)
	M-estimator	12 months	22.8 (23.1)	75.1 (26.2)	44.7 (42.0)
	EKF	n/a	33.3 (29.8)	80.6 (9.8)	43.5 (43.2)

The MLP performed better, by reporting overall change detection accuracies above 84% when using all seven spectral bands. The unsupervised classifiers performed better on the first two spectral

bands than on all seven spectral bands. This was expected, as a similar trend was observed in the classification accuracies.

In table 8.21, the same set of experiments for the real land cover change data set were conducted in Gauteng results are reported. The best performing set of experiments is again the methods that employ the MLP. The overall change detection accuracies are above 80% with false alarm rates below 20%. A significant increase in change detection accuracy is observed when the two spectral spectral bands are evaluated when compared to the NDVI. Because both the NDVI and two spectral bands' experiments uses the same spectral bands, it can be concluded that using the two spectral bands separately is better. This claim is supported by all the previous experiments in this chapter.

The worst performing methods are those that employ either the NDVI or all seven spectral bands with an unsupervised classifier. Meanwhile, similar experiments conducted with the first two spectral bands with an unsupervised classifier yielded acceptable performance. The reported overall change detection accuracies were above 75%, with a false alarm rate below 25%.

The conclusion from both sets of experiments is that using the first two spectral bands with any change detection methods yields acceptable performance. At the same time, experiments using all seven spectral bands with a supervised classifier offered the best reported performance.

8.7.3 Effective change detection delay

In this section, the effective change detection delay Δ_τ is reported. The results of the experiments are presented in table 8.22 for the Limpopo province, and table 8.23 for the Gauteng province. The experiments' results are reported in the average number of days (1 MODIS sample = 8 days) for the ensemble of time series in the simulated land cover change data set.

The MLP was deemed the best performing classifier, as it achieved the shortest effective change detection delay. The MLP's effective change detection delay improved as more spectral bands were included. The best performing feature extraction method was the SFF with a temporal sliding window length of 6 months. The overall trend was that a shorter temporal sliding window length had a shorter effective change detection delay. This is intuitive as fewer data points contribute to the current state of the output class membership. The SFFs outperform the least squares and M-estimator using a similar temporal sliding window length of 12 months.

The unsupervised classifiers (Ward clustering method and K-means) reported an overall increase in effective change detection delay when compared to the MLP classifier. A similar observation is made here as in the discussion of classification accuracy in section 8.6.7. The first two spectral bands outperformed the NDVI and all seven spectral band combinations. This is due to the improved

Table 8.22: Effective change detection delay for simulated land cover change conducted in the Limpopo province. Each entry gives the average number of days for each study area, calculated over 10 repeated independent experiments.

Algorithm	**Feature extraction**	**Window length**	**Spectral Band**		
			NDVI	2 Bands	7 Bands
MLP	SFF	6 months	88	76	73
		12 months	117	101	92
		18 months	178	120	106
	Least squares	12 months	130	109	102
	M-estimator	12 months	146	118	109
	EKF	n/a	110	96	91
Ward clustering	SFF	6 months	132	92	116
		12 months	177	113	160
		18 months	253	176	218
	Least squares	12 months	185	130	166
	M-estimator	12 months	189	125	186
	EKF	n/a	163	104	151
K**-means**	SFF	6 months	127	94	119
		12 months	169	107	154
		18 months	233	164	216
	Least squares	12 months	186	127	165
	M-estimator	12 months	186	123	179
	EKF	n/a	166	105	151

classification accuracies reported in section 8.6.3–8.6.6 for the first two spectral bands.

Most experiments conducted in the Limpopo province had the K-means algorithm producing shorter effective change detection delays than the Ward clustering method, while no distinguishing difference was observed in the Gauteng province. In these experiments a clear improvement in the effective change detection delay is observed when the SFF is compared to the least squares and M-estimator with a similar sliding window length.

8.7.4 Summary of change detection results

In this section the results of the change detection experiments are summarised. In section 8.7.1, true positives and false positives were reported for the experiments conducted on the simulated land cover change data set. In section 8.7.2, the true positives were reported for the experiments conducted on the real land cover change data set. In section 8.7.3, the average effective change detection delays were reported for the experiments conducted on the simulated land cover change data set.

Table 8.23: Effective change detection delay for simulated land cover change conducted in the Gauteng province. Each entry gives the average number of days for each study area, calculated over 10 repeated independent experiments.

Algorithm	**Feature extraction**	**Window length**	**Spectral Band**		
			NDVI	2 Bands	7 Bands
MLP	SFF	6 months	84	69	65
		12 months	111	87	81
		18 months	153	114	109
	Least squares	12 months	122	98	94
	M-estimator	12 months	127	99	97
	EKF	n/a	108	89	81
Ward clustering	SFF	6 months	117	84	102
		12 months	146	103	139
		18 months	168	140	168
	Least squares	12 months	155	120	146
	M-estimator	12 months	164	123	154
	EKF	n/a	151	97	138
K**-means**	SFF	6 months	118	88	110
		12 months	139	112	143
		18 months	172	157	189
	Least squares	12 months	153	126	149
	M-estimator	12 months	157	128	153
	EKF	n/a	137	106	134

The MLP was considered the best classifier used for change detection. The MLP had better change detection accuracies and effective change detection delays when using more spectral bands. It was also found that a trade-off existed in the length of the temporal sliding window when comparing the difference between change detection accuracy and effective change detection delay. A longer temporal sliding window length improves the classification accuracy at the cost of a longer effective change detection delay. A shorter temporal sliding window length reacts faster to change in the time series at the loss in change detection accuracy.

Poor performance with the unsupervised methods used for clustering on the NDVI time series and all seven spectral bands' time series indicated that classes could not be well encapsulated in the clusters. The first two spectral bands, on the other hand, resulted in acceptable performance across all the change detection experiments and effective change detection delay's experiments.

The K-means algorithm and Ward clustering method performed similarly in all the experiments, except that the Ward clustering method had slightly higher change detection accuracies while the

Table 8.24: A list of different combinations of change detection algorithms that will be tested at a regional scale.

Feature extraction	**Sliding window length**	**Spectral band**	**Machine learning method**
SFF	12 months	2 Bands, 7 Bands	MLP
	12 months	2 Bands	Ward clustering method
	12 months	2 Bands	K-means algorithm
EKF		2 Bands, 7 Bands	MLP
		2 Bands	Ward clustering method
		2 Bands	K-means algorithm

K-means algorithm had a shorter effective change detection delay. This observation could be attributed to the K-means classification experiments, which yielded a very small standard deviation when compared to the Ward clustering method. In all the experiments conducted in this section (section 8.7), it was observed that the SFFs and EKF features outperformed the least squares and M-estimator features in the performance metrics. It is concluded from these experiments that the combinations given in table 8.24 yielded the best performance and will be evaluated on a regional scale.

8.8 CHANGE DETECTION ALGORITHM COMPARISON

In this section the change detection accuracies measured in section 8.7 are compared to other change detection algorithms found in the literature. The change detection methods used for comparison are:

- the annual NDVI differencing method (denoted by NDVI_{CDM}) [19],
- the EKF change detection method (denoted by EKF_{CDM}) [120], and
- the ACF change detection method (denoted by ACF_{CDM}) [121].

All three these methods listed above are supervised in nature, as a training data set is required to set a threshold, which is used to declare change. These three methods are compared in table 8.25 to a few methods listed in table 8.24.

The worst performing method was the NDVI_{CDM} method, having a change detection accuracy of 69% with a false alarm rate of 13% in the Limpopo province, and a change detection accuracy of 57% with a false alarm rate of 14% in the Gauteng province. A possible explanation for this poor performance is given in [224], which is that the method assumes that the annual NDVI difference between years is normally distributed, which could imply that it has difficulty in detecting land cover

Table 8.25: Comparison of the change detection accuracies in percentage (false alarm rate in parentheses) of the proposed change detection algorithms to other change detection algorithms found in the literature.

Algorithm	**Province**	
	Limpopo province	Gauteng province
EKF_{CDM} [19]	89% (13%)	75% (13%)
ACF_{CDM} [120]	81% (12%)	92% (15%)
NDVI_{CDM} [121]	69% (13%)	57% (14%)
EKF_{BVEP}, MLP, 7 spectral bands	87% (9%)	96% (2%)
EKF_{BVEP}, MLP, 2 spectral bands	79% (23%)	92% (10%)
EKF_{BVEP}, K-means, 2 spectral bands	72% (24%)	81% (10%)
EKF_{BVEP}, Ward clustering, 2 spectral bands	72% (26%)	82% (11%)

change in heterogeneous areas. The method performed the poorest in the Gauteng province owing to the land cover diversity [224].

The EKF_{CDM} had the highest change detection accuracy of 89% in the Limpopo province, with a false alarm rate of 13%. This was attributed to the fact that most of the province is covered by natural vegetation, which is the result of the high correlation between the parameter sequences of the neighbouring pixels in the spatio-temporal window [224]. The relative difference between the change and no change parameter streams was high enough to detect change. The EKF_{CDM} method's performance was lower in the Gauteng province, which was attributed in [224] to the land cover diversity.

The ACF_{CDM} exploits the non-stationary property of the change time series when compared to the no change time series. The method was applied to the 4^{th} spectral band of MODIS, as it offered the best performance [224]. The method reported a higher change detection accuracy in the Gauteng province when compared to the Limpopo province.

The performance of the two unsupervised classifiers (K-means and Ward clustering) operating on the first two spectral bands was similar. Both methods had better change detection accuracies and false alarm rates when compared to the NDVI_{CDM} method. The methods had a 6% higher change detection accuracy when compared to the EKF_{CDM} in the Gauteng province, but a 17% decrease in the Limpopo province.

The MLP operating on the EKF_{BVEP} features computed from the first two spectral bands had the same change detection accuracy as the ACF_{CDM} in the Gauteng province, but had the advantage of having a 5% lower false alarm rate. The reverse was observed in the Limpopo province, as the MLP operating on the first two spectral bands had a 2% lower change detection accuracy and 11% higher false alarm rate when compared to the ACF_{CDM} method.

The MLP operating on the EKF_{BVEP} features computed on all seven spectral bands was deemed the best change detection method in this section. The method had the highest change detection accuracy and lowest false alarm rate in the Gauteng province. It had the second highest change detection accuracy (2% lower than the highest) and the lowest false alarm rate in the Limpopo province.

8.9 PROVINCIAL EXPERIMENTS

A list of the best performing change detection algorithms is given in table 8.24, which is to be evaluated on a regional scale. The areas that will be evaluated are the entire Limpopo and Gauteng provinces.

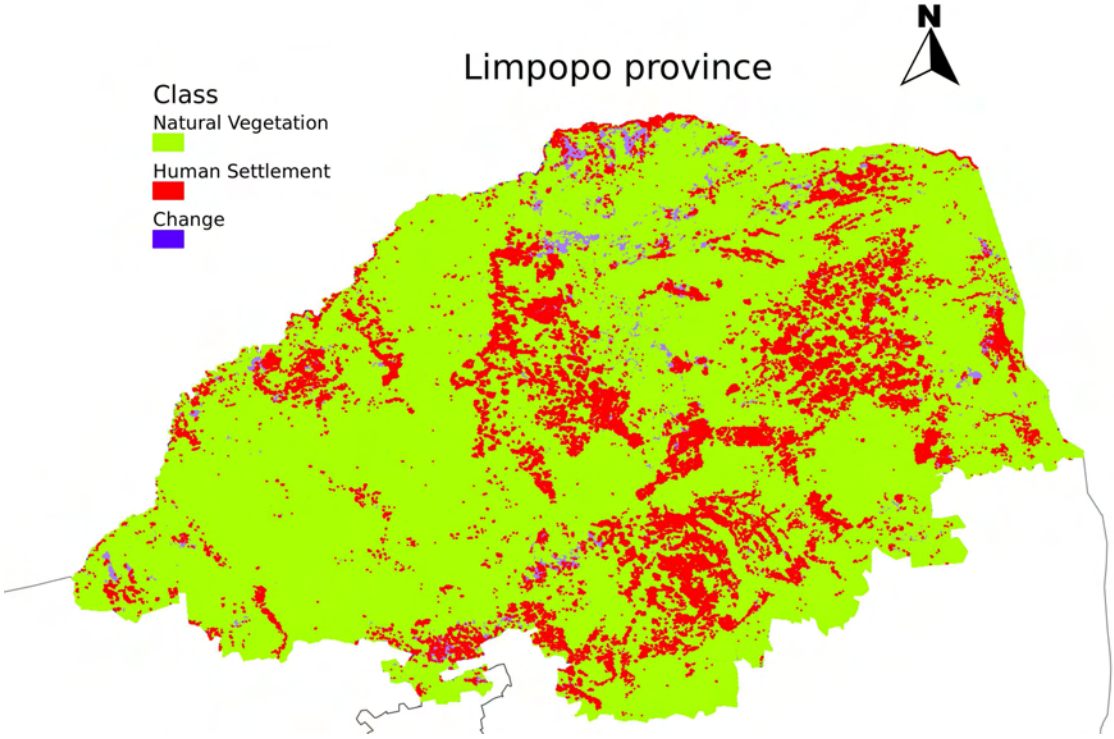


FIGURE 8.13: A classification/ change detection map of the entire Limpopo province.

The results obtained from processing the entire Limpopo province are presented in table 8.26. The table divides the results into three categories: natural vegetation, human settlements, and change. An illustration of one of these experiments is shown in figure 8.13, which represents the Limpopo province. The overall trend throughout all the methods was that natural vegetation covered 85%–88% of the province, and that human settlement covered 9%–12% of the province. This signifies that majority of the province is still largely covered by natural vegetation. The land cover change that is reported here is the transformation of natural vegetation to human settlement. The land cover change that was reported ranged from 1%–4% of the total area in the province. This is a significant area that has changed in

the province over the past decade, since the total human settlement class has expanded by 12%–40% in the study period. This suggests that some of the algorithms might be too sensitive towards change events or that the labelled data set should be expanded to incorporate a larger variety of classes. On the other hand, it should be noted that the controlled experiments that were conducted on the labelled data set involved land cover that transformed from natural vegetation to human settlement. This did not include any examples of other land cover transformations, which could exist in the province. This could be rectified, as the algorithms are versatile enough to include other classes to improve the classification, and in turn change detection accuracies. Future expansion of the work could entail collecting agricultural land cover information in each of the provinces.

Table 8.26: The classification and change detection results produced for the entire Limpopo province. The results are presented in percentage cover of total area in the province.

Feature extraction	**Algorithm**	**Spectral Band**	**Class allocation [%]**		
			Natural vegetation	Human settlement	Land cover change
SFF	MLP	2 Bands	86.94	10.31	2.75
		7 Bands	87.69	10.61	1.70
	Ward clustering	2 Bands	86.33	9.64	4.03
	K-means	2 Bands	86.05	10.02	3.93
EKF	MLP	2 Bands	85.74	11.57	2.69
		7 Bands	86.33	12.11	1.56
	Ward clustering	2 Bands	86.20	10.32	3.48
	K-means	2 Bands	85.81	10.90	3.29

Closer inspection of table 8.26 allows the deduction of some interesting trends. These trends cannot be confirmed, as no ground truth exists for the current results, which are only based on observations. The MLP consistently detected more human settlement than the unsupervised classifiers, while indicating a reduced number of detected land cover changes. This puts emphasis on the classification at the beginning of the time series, as both the detected land cover change class and the human settlement class agree that the time series ends in the human settlement class. This could be attributed to the fact that the province experienced a rainfall shortage in 2001/2002 (beginning of the study period).

The unsupervised classifiers detected more land cover change when compared to the MLP. In some experiments the size of changed areas that were reported almost doubled. Another observation among the unsupervised classifiers is that the Ward clustering method flagged more land cover changes than the K-means algorithm. This trend was also observed in the controlled experiments and was deduced

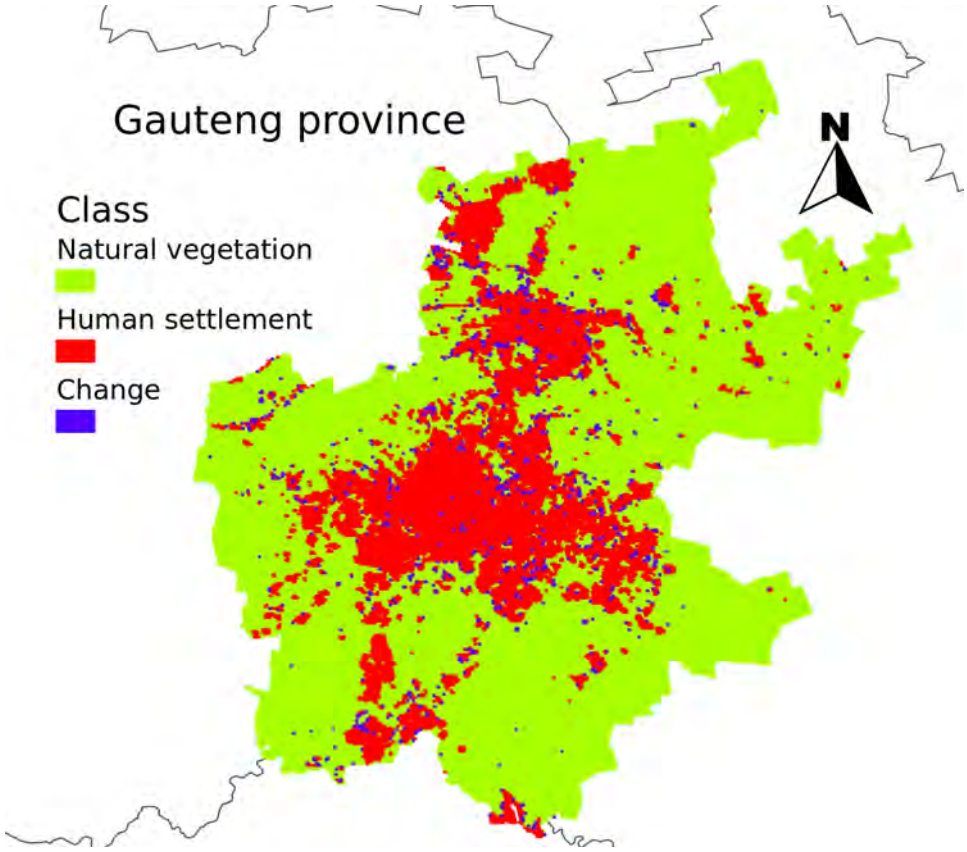


FIGURE 8.14: A classification/ change detection map of the entire Gauteng province.

from the observation that the Ward clustering method had a wider standard deviation in its classification accuracies than the K-means.

Table 8.27: The classification and change detection results produced for the entire Gauteng province. The results are presented in percentage cover of total area in the province.

Feature extraction	**Algorithm**	**Spectral Band**	**Class allocation [%]**		
			Natural vegetation	Human settlement	Land cover change
SFF	MLP	2 Bands	76.65	20.12	3.23
		7 Bands	77.33	21.39	1.28
	Ward clustering	2 Bands	75.53	19.90	4.57
	K-means	2 Bands	75.43	20.46	4.11
EKF	MLP	2 Bands	76.01	20.92	3.07
		7 Bands	76.89	21.46	1.17
	Ward clustering	2 Bands	76.22	19.56	4.22
	K-means	2 Bands	76.08	19.96	3.96

The same experiment was conducted in the Gauteng province and its results are presented in

table 8.27. The results were produced by processing the entire Gauteng province into the three defined categories. An illustration of one of these experiments is shown in figure 8.14, which represents the Gauteng province. The overall trend in this province was significantly different from the results produced in the Limpopo province, as this province is mostly urbanised. The natural vegetation class covered 75%–78% of the province, while human settlements covered 19%–22%. This result supports the concept that Gauteng is a heavily urbanised province.

The land cover change which was flagged ranged from 1%–5% of the total area in the province. This is a significant large area that has changed in the study period, as the total human settlement class has expanded by 5%–23% in the province. The same trends that were observed in the results produced for the Limpopo province with regard to the nature of the change detection algorithm were observed in the Gauteng province.

8.10 COMPUTATIONAL COMPLEXITY

In this section a comparison is made of the complexity of extracting the EKF features and the SFFs. A time series $\mathbf{x}$ of length $\mathcal{I}$, is defined as

$$\mathbf{x} = [\vec{x}_1\ \vec{x}_2\ \ldots\ \vec{x}_{\mathcal{I}}], \tag{8.1}$$

with

$$\vec{x}_i = [x_{i,1}\ x_{i,2}\ \ldots\ x_{i,T}]. \tag{8.2}$$

The variable T denotes the number of elements in vector $\vec{x}_i$. If the state-space vector $\vec{W}_i$ used in the EKF has S elements, then the complexity of filtering a single time series is at least $\mathcal{O}(\mathcal{I}S^2)+\mathcal{O}(\mathcal{I}T^{2.4})$. In the case of the EKF features extracted from a triply modulated cosine function on uncorrelated spectral bands, S=3 and T=1.

The complexity of extracting the SFF is based on the complexity of the FFT algorithm and the length of the temporal sliding window. If the time series is length $\mathcal{I}$ and the length of the temporal sliding window is Q, then the processing of a single time series is equal to $\mathcal{O}((\mathcal{I}-Q)Q\log_2 Q)$, with $Q \ll \mathcal{I}$.

A timing experiment was conducted on a cluster node to calculate the computational time of both feature extraction methods and the results are reported in table 8.28. The computer's specifications used for this experiment are:

- Dell PowerEdge 1955 blade, Intel Xeon 5355 (Quad-Core) 2.66 GHz, 8 GB RAM, 1333 MHz

Table 8.28: The computational time required to extract features from 25000 time series using either the EKF feature extraction method or SFF extraction method. The results is reported in milliseconds per time series.

Feature	Millisecond per time series
SFF	0.47
EKF	22.81

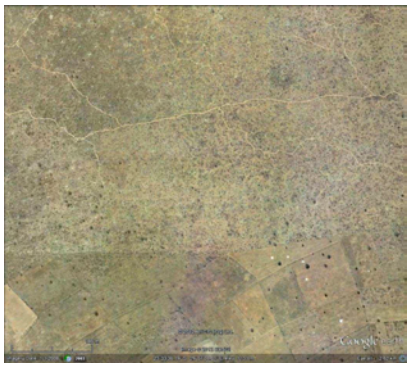
FSB, Gigabit Ethernet, 4x 2.1 kW redundant power supplies (3+1), 2x Gigabit Switch Modules, 1x Avocent Digital Access KVM switch, Software Debian Testing AMD64 with MATLAB R2012a.

The experiment was conducted over 25000 time series and it was concluded that the SFF could be extracted from the time series 48.5 times faster than the EKF features. The next requirement addressed is the time required to optimise the EKF features using the BVEP criterion. The BVSA is an iterative search algorithm that sets the EKF parameters within the BVS in an attempt to best satisfy the BVEP criterion. If the BVSA requires E_{BVSA} iterations to set the EKF parameters, the the extraction of $\mathrm{EKF}_{\mathrm{BVEP}}$ features takes at least $48.5E_{\mathrm{BVSA}}$ times longer than the SFF. The typical range of iterations used for E_{BVSA} in these experiments were between 20 and 30.

8.11 SUMMARY

In this section a summary is provided of the observarions made in this chapter. It was found that the supervised classifier outperformed the unsupervised methods. The downside was the costs involved in producing a labelled training data set. The best performance was obtained when the MLP was optimally set to operate on all seven spectral bands of MODIS. The training method adopted was the iteratively retrained mode, which compensates for the inter-annual variability. A temporal sliding window length of 12 months used on either the SFF, least squares, or M-estimator offered the best trade-off between parameter variability, effective change detection delay and change detection accuracy. Similar gains were obtained in the trade-off with the EKF features if the parameters were optimised with the BVEP criterion.

The change detection algorithms yielded better performance in the Gauteng province than the Limpopo province. This could be attributed to the more dense natural vegetation found in the Gauteng province. Figure 8.15 illustrates a difference between the informal settlements and natural vegetation found in both provinces. The Gauteng province houses more compact informal settlements and more dense natural vegetation when compared to the Limpopo province.



(a) Natural vegetation located in the Limpopo province.

(b) Informal settlements located in the Limpopo province.

(c) Natural vegetation located in the Gauteng province.

(d) Informal settlements located in the Gauteng province.

FIGURE 8.15: Four high resolution images acquired in the two provinces; Limpopo and Gauteng. (a) A natural vegetation area located in the Limpopo province. (b) An informal settlement located in the Limpopo province. (c) A natural vegetation area located in the Gauteng province. (d) An informal settlement located in the Gauteng province. (courtesy of Google^{TM}Earth).

A general trend of performance improvement was observed when the first two spectral bands (Red and NIR spectral bands) were used instead of the NDVI. The use of the first two spectral bands as input was deemed superior, as the same spectral bands are used to compute the NDVI. Further improvement was observed when using all seven spectral bands with a supervised classifier.

The SFFs and EKF features yield better performance in detecting land cover change when compared to the features extracted using least squares and M-estimator methods. The EKF features only provided better separation between classes than the SFFs when the BVEP criterion was used to set the EKF parameters. The consequence of this is that the SFF was deemed the better approach when compared to the EKF features, as the EKF-extracted features required the computation of the covariance matrices using the BVEP criterion. This improvement into separation in classes was not significant, and the SFF was deemed better owing to its lower computational time (section 8.10).

CHAPTER NINE
CONCLUSION

9.1 CONCLUDING REMARKS

The importance of reliable land cover monitoring and detection of land cover change was discussed in chapter 1, and has been shown to be of great benefit to the global community [11]. Each country or region faces its own challenges in monitoring the land; in South Africa the transformation of natural vegetation to new human settlements is the most pervasive form of land cover change [7].

South Africa's National Land Cover (NLC) was mapped in 1995–1997 using manual photo interpretation [225] of Landsat imagery, while the NLC of 2000 was based on digital classification of Landsat images by regional experts [226]. Both of these took a number of years to complete. Subsequently land cover has been mapped by provincial governments on an ad hoc basis through private companies using a variety of methods. Since the methods have not been standardised through time and space, reliable land cover change data cannot be generated from successive national land cover data sets. The Landsat-based land cover mapping efforts furthermore relied on single date imagery, which resulted in neighbouring images being acquired on widely varying dates containing seasonal effects that hampered multi-spectral land cover classification. The hyper-temporal, time-series analysis approach described here capitalises on seasonal dynamics to characterise land cover and land cover change in a repeatable, standardised method that can be applied over large areas.

The satellite images used in this thesis were acquired by the MODIS sensor. The MODIS sensor is used to produce a hyper-temporal, multi-spectral medium spatial resolution land surface reflectance data product. This sequence of images is used to construct a time series, which can be analysed with a change detection algorithm to detect the formation of newly developed human settlements. A post-classification change detection framework was developed to detect land cover change occurring in time series. The framework classifies the geographical area for each time index and declares change if a permanent transition in class label is observed. Two novel hyper-temporal feature extraction methods

were proposed in this thesis, which are used in the post-classification change detection framework. The two types of features extracted with these novel feature extraction methods are:

- the Seasonal Fourier Features (SFF), and
- the Extended Kalman Filter (EKF) features optimised using the Bias-Variance Equilibrium Point (BVEP) criterion.

The SFF is a hyper-temporal feature vector that extracts information from multiple spectral bands, which exploits the seasonal spectral signature in the temporal dimension of a geographical area. SFF is the first type of novel hyper-temporal feature in this thesis that incorporates temporal information, allowing the analysis of seasonal surface reflectance variations of different land cover classes. SFF (extracted from the MODIS time series) allows the post-classification change detection framework to be sensitive enough to detect new human settlements as small as 0.25 km^2.

The second novel hyper-temporal feature extraction method is an improvement on the method proposed by Kleynhans *et al.* [30]. The first contribution made to this method is the extension to higher dimensions, which improves the land cover change detection accuracies. This contribution is supported by all the experiments conducted in chapter 8. The second contribution made to the method proposed by Kleynhans *et al.* [30] is the definition of the novel BVEP criterion, which defines the condition that improves the tracking of time series, while simultaneously improving the internal stability of the EKF.

This criterion allows the evaluation of the EKF performance in an unsupervised fashion. The drawback with the method proposed by Kleynhans *et al.* is that it requires an offline optimisation phase, which must be performed by an operator with a training set. This drawback is overcome by defining a scoring function such as the Bias-Variance Score (BVS) to evaluate how well a particular set of parameters satisfy the BVEP criterion. The EKF parameters are adjusted using a search algorithm such as the Bias-Variance Search Algorithm (BVSA) in an attempt to best satisfy the BVEP criterion. This led to another contribution, namely the development of the BVSA; the BVSA is an unsupervised search algorithm that can effectively optimise the BVS using the BVEP criterion for optimal EKF performance. It was found in chapter 8 that the BVSA performed similarly to other popular search algorithms, but had the advantage of having a faster convergence time. All these contributions led to the full automation of the method proposed by Kleynhans *et al.* [30]. The BVS optimised using the BVEP criterion provides statistical information on the phenological growth cycle, which could also be used to provide vital insight to environmental dynamics [31, 32].

The post-classification change detection framework uses a machine learning method to classify a geographical area at each time index and can be either a supervised or an unsupervised classifier. In chapter 8 the ability of the hyper-temporal features to separate different land cover classes was

investigated. A classification experiment was used to evaluate class separation; a Multilayer Perceptron (MLP) was used to represent supervised classifiers. Unsupervised methods were represented by a selection of clustering methods. The supervised classifier performed significantly better than the unsupervised methods, but it requires labelled examples derived from commercial high resolution satellite imagery, making the unsuperivsed methods more attractive for operational implementation.

A range of experiments were conducted for different combinations of spectral bands: NDVI, first two MODIS spectral bands, and all seven MODIS spectral bands. It was observed that the experiments using the first two spectral bands yielded better results than the experiments using NDVI. This is a well-known property in the machine learning community, that better separation is usually obtained in higher dimensions [130, Ch. 1 p. 4]. This was supported by classification experiments in chapter 8, where the MLP reported general improvements with an increase in the number of spectral bands. The performance of the unsupervised methods improved when going from two-dimensional features (NDVI) to four-dimensional features (first two spectral bands), but the performance deteriorated when going to 14-dimensional features (all seven spectral bands), suggesting that complex decision boundaries are required to maximise performance in 14-dimensions.

The goal for this thesis was the development of a novel land cover change detection method. The method had to be sufficiently near automated with minimal human interaction. A post-classification change detection framework was used to evaluate two features extraction methods to improve land cover separability, which in turn improved the land cover change detection. The SFF is a novel introduced feature and was compared to the EKF feature presented by Kleynhans *et al.* [30]. The EKF features were improved using the novel BVEP criterion, which resulted in an optimised EKF that gave the best performance. The downside was that the EKF features could only provide better results if the BVEP criterion was used in the optimisation phase. These improvements over the SFF features were small when compared to the computational requirement of the optimisation phase. Therefore, it was concluded that the SFF is more practical for operational applications.

9.2 FUTURE RECOMMENDATIONS

In this section a brief overview is given of potential future research that could stem from the work presented in this thesis.

- **Spatial information analysis:** In chapter 2 it was discussed that algorithms are usually designed to provide acceptable performance for an application in a particular geographical area. This is caused by the inherent differences between geographical areas. The BVEP criterion can be used to analyse a particular geographical area by studying the statistical parameters derived, such as

the standard deviation of model parameters. This information can be used in a statistical test to determine whether a region of the study area can be expanded to cover a larger area. An example of such a test is the use of the Aikaike Information criterion (AIC) to determine if the size of the current study area is acceptable. The AIC is given as

$$\mathrm{AIC} = \ln(K) - 2\ln(L), \tag{9.1}$$

where K is the number of model parameters and L is the likelihood of the model which incorporates the standard deviation. The criterion is used to balance the cost of increased complexity (more small regions) against the loss of performance when using fewer, larger regions.

- **Spectral band selection criterion:** In chapter 4 it was discussed that proper domain knowledge leads to proper definition of feature vectors. Feature selection is always a relevant topic in remote sensing, as new sensors are continually being developed with more sophisticated capabilities. In chapter 3, an approach to training a neural network was presented which was proposed by Caruana *et al.* [168]. The training algorithm starts by mapping all the linear regions in the feature space and then progresses to map more complex non-linear regions. In a neural network architecture context, input nodes that contribute to the output nodes are assigned larger synaptic weights, while input nodes that contribute little information to the output nodes are assigned smaller synaptic weights. The distribution of the synaptic weights can be used to infer a spectral band selection criterion.

- **Internal covariance matrix analysis:** In the computation of the BVS, it is assumed that the internal covariance matrix $\mathfrak{P}_{(i|i)}$ (equation 5.38) is set to the identity matrix. The matrix will then converge to a stable internal covariance matrix $\mathfrak{P}_{(\mathcal{I}_T|\mathcal{I}_T)}$ at time $\mathcal{I}_T$ if the Riccati condition holds and enough observation vectors are supplied. This convergence should be almost constant and can be expressed as

$$\left\| \frac{\mathrm{d}^2\mathfrak{P}_{(i|i)}}{\mathrm{d}i^2} \right\| \leq \varepsilon, \tag{9.2}$$

where $\|\cdot\|$ is a suitable matrix norm, e.g. induced norm or Frobenius norm. An in-depth study is proposed on the behaviour of the EKF's internal covariance matrix $\mathfrak{P}_{(i|i)}$ with regards to land cover change. The internal covariance matrix $\mathfrak{P}_{(i|i)}$ should fluctuate when experiencing a non-stationary process such as land cover change. These fluctuations can be used to define a change threshold $T_{\mathfrak{P}}$ that flags a change when

$$\left\| \frac{\mathrm{d}^2 \mathfrak{P}_{(i|i)}}{\mathrm{d}i^2} \right\| > T_{\mathfrak{P}}. \tag{9.3}$$

- **Complex model design:** In chapter 5 the emphasis was placed on using a triply modulated cosine model to describe the MODIS time series. The next phase is to explore more complex models, which could be used to model the time series. For example, the triply modulated cosine model given in equation (5.44) can be expanded to incorporate multiple models as

$$x_i = \sum_{m}^{M} \mathbf{h}_m(\vec{W}_i) + v_i, \tag{9.4}$$

with measurement function defined as

$$\mathbf{h}_m(\vec{W}_i) = W_{i,\mu,m} + W_{i,\alpha,m} \cos(2\pi f_{\text{samp}} i + W_{i,\theta,m}). \tag{9.5}$$

Another proposed expansion to the SFF feature is to consider more Fourier components for analysis. The sinusoidal behaviour is not a true representation of all different land cover classes, which motivates a further exploration of new models.

REFERENCES

[1] A. Comber, P. Fisher, and R. Wadsworth, "What is land cover?" *Environment and planning B: Planning and design*, vol. 32, no. 2, pp. 199–209, 2005.

[2] P. Vitousek, H. Mooney, J. Lubchenco, and J. Melillo, "Human domination of Earth's ecosystems," *Science*, vol. 277, pp. 494–499, July 1997.

[3] G. Daily and P. Ehrlich, "Population, sustainability, and Earth's carrying capacity," *Bioscience*, vol. 42, no. 10, pp. 761–771, November 1992.

[4] R. DeFries, L. Bounoua, and G. Collatz, "Human modification of the landscape and surface climate in the next fifty years," *Global Change Biology*, vol. 8, no. 5, pp. 438–458, May 2002.

[5] J. Foley, R. DeFries, G. Asner, C. Barford, G. Bonan, S. Carpenter, F. Chapin, M. Coe, G. Daily, H. Gibbs, J. Helkowski, T. Holloway, E. Howard, C. Kucharik, C. Monfreda, J. Patz, I. Prentice, N. Ramankutty, and P. Snyder, "Global consequences of land use," *Science*, vol. 309, no. 5734, pp. 570–574, July 2005.

[6] G. Brundtland, "Report of the World Commission on environment and development: Our common future," Brundtland Commission, United Nations General Assembly, Tech. Rep. A/42/427, 1987.

[7] C. Olver, "South Africa's review report for the sixteenth session of the United Nations commission on sustainable development," Department of Environmental Affairs and Tourism Pretoria, Tech. Rep. CSD-16, March 2008.

[8] B. Salmon, J. Olivier, W. Kleynhans, K. Wessels, F. van den Bergh, and K. Steenkamp, "The use of a Multilayer Perceptron for detecting new human settlements from a time series of MODIS images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 13, no. 6, pp. 873–883, December 2011.

[9] P. van den Berg, "Transformasie van winterveld: Veranderde grondbenutting en nedersettingsverdigting," Master's thesis, Department of Geography, University of Pretoria, Pretoria, South Africa, October 1994.

[10] H. Eva, A. Brink, and D. Simonetti, "Monitoring land cover dynamics in sub-Saharan Africa," Institute for Environmental and Sustainability, Tech. Rep. EUR 22498 EN, 2006.

[11] C. Johannsen, P. Carter, D. Morris, B. Erickson, and K. Ross, "Potential applications of remote sensing," Site-Specific Management Guidelines SSMG-22, Potash and Phosphate Institute, Tech. Rep., 1999.

[12] R. Myneni and J. Ross, *Photon-vegetation Interactions: Applications in Optical Remote Sensing and Plant Physiology*, 1st ed. New York, USA: Springer, 1991.

[13] S. Liang, *Quantitative Remote Sensing of land surfaces*, 1st ed. New York, USA: Wiley Interscience, 2004.

[14] R. DeFries and J. Chan, "Multiple criteria for evaluating machine learning algorithms for land cover classification from satellite data," *Remote Sensing of Environment*, vol. 74, no. 3, pp. 503–515, December 2000.

[15] R. S. Lunetta, D. Johnson, J. Lyon, and J. Crotwell, "Impacts of imagery temporal frequency on land-cover change detection monitoring," *Remote Sensing of Environment*, vol. 89, no. 4, pp. 444–454, February 2004.

[16] J. Townshend and C. Justice, "Selecting the spatial resolution of satellite sensors required for global monitoring of land transformations," *International Journal of Remote Sensing*, vol. 9, no. 2, pp. 187–236, February 1988.

[17] M. Hansen and R. DeFries, "Detecting long-term global forest change using continuous fields of tree-cover maps from 8-km Advanced Very High Resolution Radiometer (AVHRR) data for the years 1982-99," *Ecosystems*, vol. 7, no. 7, pp. 695–716, November 2004.

[18] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote Sensing*, vol. 28, no. 5, pp. 823–870, January 2007.

[19] R. Lunetta, J. Knight, J. Ediriwickrema, J. Lyon, and L. Worthy, "Land-cover change detection using multi-temporal MODIS NDVI data," *Remote Sensing of Environment*, vol. 105, no. 2, pp. 142–154, November 2006.

[20] P. Coppin, I. Jonckheere, K. Nackaerts, B. Muys, and E. Lambin, "Digital change detection methods in ecosystem monitoring: a review," *International Journal of Remote Sensing*, vol. 25, no. 9, pp. 1565–1596, May 2004.

[21] S. Gopal, C. Woodcock, and A. Strahler, "Fuzzy neural network classification of global land cover from a 1 degree AVHRR data set," *Remote Sensing of Environment*, vol. 67, no. 2, pp. 230–243, February 1999.

[22] G. Carpenter, S. Gopal, S. Macomber, S. Martens, C. Woodcock, and J. Franklin, "A neural network method for efficient vegetation mapping," *Remote Sensing of Environment*, vol. 70, no. 3, pp. 326–338, December 1999.

[23] B. Braswell, S. Hagen, S. Frolking, and W. Salas, "A multivariable approach for mapping sub-pixel land cover distributions using MISR and MODIS: application in the Brazilian Amazon region," *Remote Sensing of Environment*, vol. 87, no. 2-3, pp. 243–256, October 2003.

[24] D. Lu, P. Mausel, E. Brondizio, and E. Moran, "Change detection techniques," *International Journal of Remote Sensing*, vol. 25, no. 12, pp. 2365–2407, June 2004.

[25] H. Nemmour and Y. Chibani, "Neural network combination by fuzzy integral for robust change detection in remotely sensed imagery," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 14, pp. 2187–2195, January 2005.

[26] T. Westra and R. de Wulf, "Monitoring Sahelian floodplains using Fourier analysis of MODIS time-series data and artificial neural networks," *International Journal of Remote Sensing*, vol. 28, no. 7, pp. 1595–1610, January 2007.

[27] W. Wanner, A. H. Strahler, B. Hu, P. Lewis, J. Muller, X. Li, C. Schaaf, and M. Barnsley, "Global retrieval of bidirectional reflectance and albedo over land from EOS MODIS and MISR data: Theory and algorithm," *Journal of Geophysical Research*, vol. 102, no. D14, pp. 17 143–17 161, 1997.

[28] C. Schaaf, F. Gao, A. Strahler, W. Lucht, X. Li, T. Tsang, N. Strugnell, X. Zhang, Y. Jin, J. Muller, P. Lewis, M. Barnsley, P. Hobson, M. Disney, G. Roberts, M. Dunderdale, C. Doll, R. d'Entremont, B. Hu, S. Liang, J. Privette, and D. Roy, "First Operational BRDF, Albedo and Nadir Reflectance Products from MODIS," *Remote Sensing of Environment*, vol. 83, no. 1, pp. 135–148, November 2002.

[29] E. Keogh and J. Lin, "Clustering of time-series subsequences is meaningless: implications for previous and future research," *Knowledge and Information systems*, vol. 8, no. 2, pp. 154–177, August 2005.

[30] W. Kleynhans, J. Olivier, K. Wessels, F. van den Bergh, B. Salmon, and K. Steenkamp, "Improving land-cover class separation using an extended Kalman filter on MODIS NDVI time-series data," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 2, pp. 381–385, April 2010.

[31] M. Jakubauskas, D. Legates, and J. Kastens, "Harmonic analysis of time-series AVHRR NDVI data," *Photogrammetric Engineering of Remote Sensing*, vol. 67, no. 4, pp. 461–470, April 2001.

[32] S. Lhermitte, J. Verbesselt, K. Nackaerts, and P. Coppin, "A segmentation of vegetation-soil-climate complexes for South Africa based on SPOT vegetation time series," in *2nd International Vegetation User Conference*, vol. 1, Antwerp, Belgium, March 24–26, 2004, pp. 1–7.

[33] S. Liang, *Advances in land remote sensing: System, modeling, inversion and application*, 1st ed. New York, USA: Springer, 2008.

[34] W. Derman and S. Whiteford, *Social impact analysis and development planning in the third world*, 1st ed. Colorado, USA: Westview Press, 1985.

[35] F. Hudson, *A Geography of settlements*, 2nd ed. London, UK: Macdonald and Evans Ltd, 1976.

[36] P. Harrison, "The policies and politics of informal settlements in South Africa: A historical perspective," *Journal of Africa Insights*, vol. 22, no. 1, pp. 14–22, 1992.

[37] A. Gilbert and J. Gugler, *Cities, poverty and development: Urbanization in the third world*, 1st ed. London, UK: Oxford University Press, 1982.

[38] A. Christopher, "Apartheid and urban segregation levels in South Africa," *Journal of Urban Studies*, vol. 27, no. 3, pp. 421–440, June 1990.

[39] C. de Wet, *Moving together drifting apart: Betterment planning and villagisation in a South African homeland*, 1st ed. Johannesburg, South Africa: Witwatersrand University Press, 1995.

[40] B. Salmon, J. Olivier, K. Wessels, W. Kleynhans, F. van den Bergh, and K. Steenkamp, "Unsupervised land cover change detection: Meaningless sequential time series analysis," *IEEE Transactions Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 2, pp. 327–335, June 2011.

[41] G. Gutman, A. Janetos, C. Justice, E. Moran, J. Mustard, R. Rindfuss, D. Skole, B. Turner, and M. Cochrane, *Land Change Science: Observing, Monitoring, and Understanding Trajectories of Change on the Earths Surface*, 1st ed. New York, USA: Springer, 2004.

[42] T. Lillesand and R. Kiefer, *Remote Sensing and Image Interpretation*, 4th ed. New York, NY: John Wiley and Sons, 2000.

[43] P. Gibson, *Introductory Remote Sensing: Principles and Concepts*, 1st ed. New York, NY: Routledge, 2000.

[44] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of Physics*, 1st ed. John Wiley and Sons, 1997.

[45] H. Pollack, S. Hurter, and J. Johnson, "Heat flow from the Earth's interior: Analysis of the global data set," *Reviews of Geophysics*, vol. 31, no. 3, pp. 267–280, 1993.

[46] B. Nordell and B. Gervet, "Global energy accumulation and net heat emission," *International Journal of Global Warming*, vol. 1, no. 1–3, pp. 378–391, 2009.

[47] R. Dickinson, "Land surface processes and climate-surface albedos and energy balance," *Advance Geophysics*, vol. 25, pp. 305–353, 1983.

[48] J. Foley, I. Prentice, N. Ramankutty, S. Levis, D. Pollard, S. Sitch, and A. Haxeltine, "An integrated biosphere model of land surface processes, terrestrial carbon balance, and vegetation dynamics," *Global Biogeochemical Cycles*, vol. 10, no. 4, pp. 603–628, 1996.

[49] R. Dickinson, "Land processes in climate models," *Remote Sensing of Environment*, vol. 51, no. 1, pp. 27–38, January 1995.

[50] P. Tyson and R. Preston-Whyte, *The weather and climate of southern Africa*, 2nd ed. Oxford University Press, 2002.

[51] J. Nagol, E. Vermote, and S. Prince, "Effects of atmospheric variation on AVHRR NDVI data," *Remote Sensing of Environment*, vol. 113, no. 2, pp. 392–397, February 2009.

[52] H. Ouaidrari and E. Vermote, "Operational atmospheric correction of Landsat TM data," *Remote Sensing of Environment*, vol. 70, no. 1, pp. 4–15, October 1999.

[53] R. Avissar and R. Pielke, "A parameterization of heterogeneous land-surface for atmospheric numerical models and its impact on regional meteorology," *Monthly Weather Review*, vol. 117, no. 10, pp. 2113–2136, October 1989.

[54] P. R.A., G. Dalu, J. Snook, T. Lee, and T. Kittel, "Nonlinear influence of mesoscale land use on weather and climate," *Journal of Climate*, vol. 4, no. 11, pp. 1053–1069, November 1991.

[55] J. Proakis and M. Salehi, *Communication systems engineering*, 2nd ed. Upper Saddle River, New Jersey, USA: Prentice Hall, 2002.

[56] "Draft of the MODIS level 1B Algorithm Theoretical Basis Document Version 2.0," SAIC/GSC MODIS Characterization Support Team (MCST), Tech. Rep., February 1997.

[57] C. Justice, E. Vermote, J. Townshend, R. Defries, D. Roy, D. Hall, V. Salomonson, J. Privette, G. Riggs, A. Strahler, W. Lucht, R. Myneni, Y. Knyazikhin, S. Running, R. Nemani, Z. Wan, A. Huete, W. van Leeuwen, R. Wolfe, L. Giglio, J. Muller, P. Lewis, and M. Barnsley, "The Moderate resolution imaging spectroradiometer (MODIS): Land remote sensing for global change research," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 4, pp. 1228–1249, July 1998.

[58] W. Lucht, C. Schaaf, and A. Strahler, "An Algorithm for the retrieval of albedo from space using semiempirical BRDF models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 2, pp. 977–998, March 2000.

[59] W. Lucht and J. Roujean, "Considerations in the Parametric Modeling of BRDF and Albedo from Multiangular Satellite Satellite Sensor Observations," *Remote Sensing Reviews*, vol. 18, no. 2-4, pp. 343–379, September 2000.

[60] W. Lucht and P. Lewis, "Theoretical noise sensitivity of BRDF and albedo retrieval from the EOS-MODIS and MISR sensors with respect to angular sampling," *International Journal of Remote Sensing*, vol. 21, no. 1, pp. 81–98, January 2000.

[61] E. Vermote and A. Vermeulen, "Atmospheric correction algorithm: Spectral reflectance (MOD09) algorithm theoretical basis document (ATBD)," Department of Geography, University of Maryland, Tech. Rep., 1999.

[62] E. Vermote, N. Saleous, and C. Justice, "Atmospheric correction of MODIS data in the visible to middle infrared: First results," *Remote Sensing of Environment*, vol. 83, no. 1–2, pp. 97–111, November 2002.

[63] F. Nicodemus, "Directional reflectance and emissivity of an opaque surface," *Journal of Applied Optics*, vol. 4, no. 7, pp. 767–773, May 1965.

[64] D. Roy, Y. Jin, P. Lewis, and C. Justice, "Prototyping a global algorithm for systematic fire-affected area mapping using MODIS time series data," *Remote Sensing of Environment*, vol. 97, no. 2, pp. 137–162, July 2005.

[65] R. Wolfe, D. Roy, and E. Vermote, "MODIS Land data storage, gridding, and compositing methodology: Level 2 grid," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 4, pp. 1324–1338, July 1998.

[66] W. Barnes, T. Pagano, and V. Salomonson, "Prelaunch characteristics of the Moderate Resolution Imaging Spectroradiometer (MODIS) on EOS-AM1," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 4, pp. 1088–1100, July 1998.

[67] A. Huete, K. Huemmrich, T. Miura, X. Xiao, K. Didan, W. van Leeuwen, F. Hall, and C. Tucker, "Vegetation Index greenness global data set," NASA ESDR/CDR, Tech. Rep. 1, April 2006.

[68] J. Rouse, R. Haas, D. Deering, and J. Schell, "Monitoring the vernal advancement and retrogradation (Green wave effect) of natural vegetation," Goddard Space Flight Center, Greenbelt, Maryland 20771, Tech. Rep., October 1973.

[69] P. Sellers, "Canopy reflectance, photosynthesis, and transpiration," *International Journal of Remote Sensing*, vol. 6, no. 8, pp. 1335–1372, August 1985.

[70] R. Myneni, F. Hall, P. Sellers, and A. Marshak, "The interpretation of spectral vegetation indexes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 2, pp. 481–486, March 1995.

[71] B. Pinty and M. Verstraete, "A non-linear index to monitor global vegetation from satellites," *Plant Ecology*, vol. 101, no. 1, pp. 15–20, July 1992.

[72] A. Richardson and C. Wiegand, "Distinguishing vegetation from soil background information," *Photogrammetric Engineering and Remote Sensing*, vol. 43, no. 2, pp. 1541–1552, December 1977.

[73] A. Huete, "A soil-adjusted vegetation index (SAVI)," *Remote Sensing of Environment*, vol. 25, no. 3, pp. 53–70, August 1988.

[74] Y. Kaufman and D. Tanre, "Atmospherically resistant vegetation index (ARVI) for EOS-MODIS," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 2, pp. 261–270, March 1992.

[75] F. Garcia-Haro, M. Gilabert, and J. Melia, "Monitoring fire-affected areas using Thematic Mapper data," *International Journal of Remote Sensing*, vol. 22, no. 4, pp. 533–549, March 2001.

[76] T. Fung and W. Siu, "Environmental quality and its changes, an analysis using NDVI," *International Journal of Remote Sensing*, vol. 21, no. 5, pp. 1011–1024, July 2000.

[77] E. Rosch, "Natural categories," *Cognitive Psychology*, vol. 4, no. 3, pp. 328–350, May 1973.

[78] T. Fung, "Land use and land cover change detection with Landsat MSS and SPOT HRV data in Hong Kong," *Geocarto International*, vol. 7, no. 3, pp. 33–40, September 1992.

[79] N. Gautam and G. Chennaiah, "Land-use and land-cover mapping and change detection in tripura using satellite Landsat data," *International Journal of Remote Sensing*, vol. 6, no. 3–4, pp. 517–528, March 1985.

[80] K. Price, D. Pyke, and L. Mendes, "Shrub dieback in a semiarid ecosystem: the integration of remote sensing and GIS for detecting vegetation change," *Photogrammetric Engineering and Remote Sensing*, vol. 58, no. 4, pp. 455–463, April 1992.

[81] D. Alves, J. Pereira, C. De Sousa, J. Soares, and F. Yamaguchi, "Characterizing landscape changes in central Rondonia using Landsat TM imagery," *International Journal of Remote Sensing*, vol. 20, no. 14, pp. 2877–2882, September 1999.

[82] D. Fuller, "Satellite remote sensing of biomass burning with optical and thermal sensors," *Progress in Physical Geography*, vol. 24, no. 4, pp. 543–561, December 2000.

[83] V. Cuomo, R. Lasaponara, and V. Tramutoli, "Evaluation of a new satellite-based method for forest fire detection," *International Journal of Remote Sensing*, vol. 22, no. 9, pp. 1799–1826, June 2001.

[84] J. Chan, K. Chan, and A. Yeh, "Detecting the nature of change in an urban environment: a comparison of machine learning algorithms," *Photogrammetric Engineering and Remote Sensing*, vol. 67, no. 2, pp. 213–225, February 2001.

[85] X. Li and A. Yeh, "Principal component analysis of stacked multitemporal images for the monitoring of rapid urban expansion in the Pearl River Delta," *International Journal of Remote Sensing*, vol. 19, no. 8, pp. 1501–1518, May 1998.

[86] J. Michalek, T. Wager, J. Luczkovich, and R. Stoffle, "Multispectral change vector analysis for monitoring coastal marine environments," *Photogrammetric Engineering and Remote Sensing*, vol. 59, no. 3, pp. 381–384, March 1993.

[87] G. Zhou, J. Luo, C. Yang, B. Li, and S. Wang, "Flood monitoring using multitemporal AVHRR and RADARSAT imagery," *Photogrammetric Engineering and Remote Sensing*, vol. 66, no. 5, pp. 633–638, May 2000.

[88] P. Agouris, A. Stefanidis, and S. Gyftakis, "Differential snakes for change detection in road segments," *Photogrammetric Engineering and Remote Sensing*, vol. 67, no. 12, pp. 1391–1399, December 2001.

[89] R. Dwivedi and T. Sankar, "Monitoring shifting cultivation using space-borne multispectral and multitemporal data," *International Journal of Remote Sensing*, vol. 12, no. 3, pp. 427–433, March 1991.

[90] W. Kleynhans, B. Salmon, J. Olivier, K. Wessels, and F. van den Bergh, "A comparison of feature extraction methods within a spatio-temporal land cover change detection framework," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 1, Vancouver, Canada, July 24–29, 2011, pp. 688–691.

[91] J. Townshend, C. Justice, C. Gurney, and J. McManus, "The impact of misregistration on change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 5, pp. 1054–1060, September 1992.

[92] X. Dai and S. Khorram, "The effects of image misregistration on the accuracy of remotely sensed change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 5, pp. 1566–1577, September 1998.

[93] R. Nelson, "Detecting forest canopy change due to insect activity using Landsat MSS," *Photogrammetric Engineering and Remote Sensing*, vol. 49, no. 9, pp. 1303–1314, September 1983.

[94] J. Lyon, D. Yuan, R. Lunetta, and C. Elvidge, "A change detection experiment using vegetation indices," *Photogrammetric Engineering and Remote Sensing*, vol. 64, no. 2, pp. 143–150, 1998.

[95] K. Green, D. Kempka, and L. Lackley, "Using remote sensing to detect and monitor land-cover and land-use change," *Photogrammetric Engineering and Remote Sensing*, vol. 60, no. 3, pp. 331–337, 1994.

[96] J. Jensen and D. Toll, "Detecting residential land use development at the urban fringe," *Photogrammetric Engineering and Remote Sensing*, vol. 48, no. 4, pp. 629–643, April 1982.

[97] P. Chavez and D. MacKinnon, "Automatic detection of vegetation changes in the southwestern United States using remotely sensed images," *Photogrammetric Engineering and Remote Sensing*, vol. 60, no. 5, pp. 571–583, May 1994.

[98] A. Singh, "Digital change detection techniques using remotely sensed data." *International Journal of Remote Sensing*, vol. 10, no. 6, pp. 989–1003, June 1989.

[99] J. Adams, D. Sabol, V. Kapos, R. Filho, D. Roberts, M. Smith, and A. Gillespie, "Classification of multispectral images based on fractions of endmembers: application to land-cover change in the Brazillian Amazon," *Remote Sensing of Environment*, vol. 52, no. 2, pp. 137–154, May 1995.

[100] S. Macomber and C. Woodcock, "Mapping and monitoring conifer mortality using remote sensing in the Lake Tahoe Basin," *Remote Sensing of Environment*, vol. 50, no. 3, pp. 255–266, December 1994.

[101] C. Lo and R. Shipman, "A GIS approach to land-use change dynamics detection," *Photogrammetric Engineering and Remote Sensing*, vol. 56, no. 11, pp. 1483–1491, November 1990.

[102] T. Stone and P. Lefebvre, "Using multitemporal satellite data to evaluate selective logging in Para, Brazil," *International Journal of Remote Sensing*, vol. 19, no. 13, pp. 2517–2526, January 1998.

[103] R. Lawrence and W. Ripple, "Calculating change curves for multitemporal satellite imagery: Mount St. Helens 1980–1995," *Remote Sensing of Environment*, vol. 67, no. 3, pp. 309–319, March 1999.

[104] T. Yue, S. Chen, B. Xu, Q. Liu, H. Li, G. Liu, and Q. Ye, "A curve-theorem based approach for change detection and its application to Yellow River Delta," *International Journal of Remote Sensing*, vol. 23, no. 11, pp. 2283–2292, June 2002.

[105] G. Henebry, "Detecting change in grasslands using measures of spatial dependence with Landsat TM data." *Remote Sensing of Environment*, vol. 46, no. 2, pp. 223–234, November 1993.

[106] J. Verbesselt, R. Hyndman, G. Newnham, and D. Culvenor, "Detecting trend and seasonal changes in satellite image time series," *Remote Sensing of Environment*, vol. 114, no. 1, pp. 106–115, January 2010.

[107] R. Lunetta, J. Ediriwickrema, D. Johnson, J. Lyon, and A. McKerrow, "Impact of vegetation dynamics on the identification of land-cover change in a biologically complex community in North Carolina, USA," *Remote Sensing of Environment*, vol. 82, no. 2–3, pp. 258–270, October 2002.

[108] T. Loveland, J. Merchant, J. Brown, D. Ohlen, B. Reed, P. Olson, and J. Hutchinson, "Seasonal land-cover regions of the United States," *Annals of the Association of American Geographers*, vol. 85, no. 2, pp. 339–355, June 1995.

[109] R. Kennedy, W. Cohen, and T. Schroeder, "Trajectory-based change detection for automated characterization of forest disturbance dynamics," *Remote Sensing of Environment*, vol. 110, no. 3, pp. 370–386, October 2007.

[110] F. Bovolo and L. Bruzzone, "A Split-based approach to unsupervised change detection in large-size multitemporal images: Application to Tsunami-damage assessment," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 6, pp. 1658–1670, June 2007.

[111] C. Jha and N. Unnia, "Digital change detection of forest conversion of a dry tropical Indian forest region," *International Journal of Remote Sensing*, vol. 15, no. 13, pp. 2543–2552, September 1994.

[112] P. Howarth and G. Wickware, "Procedures for change detection using Landsat digital data," *International Journal of Remote Sensing*, vol. 2, no. 3, pp. 277–291, August 1981.

[113] R. Townshend and C. Justice, "Spatial variability of images and the monitoring of changes in the normalized difference vegetation index," *International Journal of Remote Sensing*, vol. 16, no. 12, pp. 2187–2195, August 1995.

[114] E. Lambin and A. Strahler, "Indicators of land-cover change for change-vector analysis in multitemporal space at coarse spatial scales," *International Journal of Remote Sensing*, vol. 15, no. 10, pp. 2099–2119, July 1994.

[115] S. Mitra, *Digital signal processing: A computer-based approach*, 2nd ed. New York, USA: McGraw-Hill, 2002.

[116] S. Lhermitte, J. Verbesselt, I. Jonckheere, K. Nackaerts, J. van Aardt, W. Verstraeten, and P. Coppin, "Hierarchical image segmentation based on similarity of NDVI time series," *Remote Sensing of Environment*, vol. 112, no. 2, pp. 506–521, February 2008.

[117] J. Verbesselt, R. Hyndman, A. Zeileis, and D. Culvenor, "Phenological change detection while accounting for abrupt and gradual trends in satellite image time series," *Remote Sensing of Environment*, vol. 114, no. 12, pp. 2970–2980, December 2010.

[118] C. Potter, P. Tan, M. Steinbach, S. Klooster, V. Kumar, R. Myneni, and V. Genovese, "Major disturbance events in terrestrial ecosystems detected using global satellite data sets," *Global Change Biology*, vol. 9, no. 7, pp. 1005–1021, July 2003.

[119] D. Mildrexler, M. Zhao, and S. Running, "Testing a MODIS Global Disturbance Index across North America," *Remote Sensing of Environment*, vol. 113, no. 10, pp. 2103–2117, October 2009.

[120] W. Kleynhans, J. Olivier, K. Wessels, B. Salmon, F. van den Bergh, and K. Steenkamp, "Detecting land cover change using an Extended Kalman Filter on MODIS NDVI time series data," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 3, pp. 507–511, May 2011.

[121] W. Kleynhans, B. Salmon, J. Olivier, F. van den Bergh, K. Wessels, T. Grobler, and K. Steenkamp, "Land cover change detection using autocorrelation analysis on MODIS time-series data: Detection of new human settlements in the Gauteng province of South Africa," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, In press 2011.

[122] M. Hansen, R. DeFries, J. Townshend, M. Carroll, C. Dimiceli, and R. Sohlberg, "Global Percent Tree Cover at a spatial resolution of 500 meters: First results of the MODIS vegetation continuous fields algorithm," *Earth Interactions*, vol. 7, no. 10, pp. 1–15, October 2003.

[123] X. Zhan, R. Sohlberg, J. Townshend, C. DiMiceli, M. Carroll, J. Eastman, M. Hansen, and R. DeFries, "Detection of land cover changes using MODIS 250m data," *Remote Sensing of Environment*, vol. 83, no. 1-2, pp. 336–350, November 2002.

[124] A. Strahler, D. Muchoney, J. Borak, M. Friedl, S. Gopal, E. Lambin, and A. Moody, "MODIS Land Cover Product Algorithm Theoretical Basis Document (ATBD): MODIS Land Cover and Land-Cover Change," Boston: Boston University, Tech. Rep., May 1999.

[125] J. Vermaak and E. Botha, "Recurrent neural networks for short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 13, no. 1, pp. 126–132, February 1998.

[126] X. Wang, L. Xiu-Xia, and J. Sun, "A new approach of neural networks to time-varying database classification," in *IEEE Proceedings Machine Learning and Cybernetics*, vol. 4, Guangzhou, China, August 18–21, 2005, pp. 2050–2054.

[127] S. Salzberg, "On comparing classifiers: Pitfalls to avoid and a recommended approach," *Data mining and knowledge discovery*, vol. 1, no. 3, pp. 317–328, September 1997.

[128] L. Bruzzone and S. Serpico, "An iterative technique for the detection of land-cover transitions in multitemporal remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 4, pp. 858–867, July 1997.

[129] C. Burges, "A Tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, June 1998.

[130] C. Bishop, *Neural Networks for Pattern Recognition*, 2nd ed. New York, USA: Oxford University Press, 1995.

[131] M. Richard and R. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Computation*, vol. 3, no. 4, pp. 461–483, 1991.

[132] H. White, "Connectionist nonparametric regression: multilayer feedforward networks can learn arbitrary mappings," *Journal of Neural Networks*, vol. 3, no. 5, pp. 535–549, 1990.

[133] J. Hopfield, "Learning algorithms and probability distributions in feed-forward and feed-back networks," *Proceedings of the National Academy of Sciences*, vol. 84, no. 23, pp. 8429–8433, December 1987.

[134] J. Hampshire and B. Pearlmutter, "Equivalence proofs for multilayer perceptron classifiers and the Bayesian discriminant function," in *Proceedings of the 1990 Connectionist Models Summer School*, vol. 1, San Mateo, CA, USA, 1990, pp. 159–172.

[135] C. Bishop, "Novelty detection and neural network validation," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 141, no. 4, pp. 217–222, August 1994.

[136] P. Hartono and H. Shuji, "Learning from imperfect data," *Journal of Applied Soft Computing*, vol. 7, no. 1, pp. 353–363, January 2007.

[137] I. Bruha and A. Famili, "Postprocessing in machine learning and data mining," *ACM SIGKDD Explorations Newsletter - Special issue on Scalable data mining algorithms*, vol. 2, no. 2, pp. 110–114, December 2000.

[138] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. New Jersey, USA: Prentice Hall, 2002.

[139] F. Rosenblatt, "The perceptron – a perceiving and recognizing automaton," Cornell Aeronautical Laboratory, Tech. Rep. 85-460-1, 1957.

[140] M. Minsky and S. Papert, *Perceptron*, 1st ed. Cambridge, Massachusetts, USA: MIT Press, 1969.

[141] D. MacKay, *Information Theory, Inference, and Learning Algorithms*, 1st ed. Cambridge, United Kingdom: Cambridge University Press, 2003.

[142] A. Kolmogorov, "On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition," *Doklady Akademii. Nauk USSR*, vol. 114, pp. 679–681, 1957.

[143] R. Duda, P. Hart, and D. Stork, *Pattern classification*, 2nd ed. New York: Wiley-Interscience, 2000.

[144] A. Barron, "Universal approximation bounds for superposition of a sigmoidal function," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 930–945, May 1993.

[145] R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4–22, April 1987.

[146] D. Rumelhart and J. McClelland, *Parallel Distributed Processing*, 1st ed. Cambridge: MIT Press, 1987.

[147] Y. Le Cun, P. Simard, and B. Pearlmutter, "Automatic learning rate maximization by on-line estimation of the Hessian eigenvectors," *Advances in Neural Information Processing Systems*, vol. 5, pp. 156–163, 1993.

[148] D. Plaut, S. Nowlan, and G. Hinton, "Experiments on learning by back propagation," Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-86-126, 1986.

[149] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 1, San Francisco, CA, USA, 28 March – 1 April, 1993, pp. 586–591.

[150] S. Fahlman, "Faster-learning variation back-propagation: an empirical study," in *Proceedings of the 1988 Connectionist Models Summer School*, vol. 1, San Mateo, CA, USA, 1988, pp. 38–51.

[151] R. Brent, *Algorithms for minimization without derivatives*, 1st ed. Englewood Cliffs, NJ, USA: Prentice Hall, 1973.

[152] M. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 46, no. 6, pp. 409–436, 1952.

[153] J. Dennis and R. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, 1st ed. New Jersey, US: Society for Industrial Mathematics, 1987.

[154] D. Shanno, "Conjugate gradient methods with inexact searches," *Mathematics of Operations Research*, vol. 3, no. 3, pp. 244–256, 1978.

[155] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quaterly Journal of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[156] D. Marquardt, "An algorithm for least-squares estimation of non-linear parameters," *Journal of the Society of Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[157] J. Moody and C. Darken, "Fast learning in networks of locally tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.

[158] S. Chen, C. Cowan, and P. Grant, "Orthogonal least squares learning algorithm for Radial Basis Function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, March 1991.

[159] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, January 1982.

[160] ——, *Self-organization and associative memory*, 2nd ed. Berlin: Springer-Verlag, 1987.

[161] J. Hopfield and D. Tank, "Neural computations of decisions in optimization problems," *Biology and Cybernetics*, vol. 52, no. 3, pp. 1–25, July 1985.

[162] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of USA*, vol. 79, no. 8, pp. 2554–2558, April 1982.

[163] J. Li, A. Michel, and W. Porod, "Analysis and synthesis of a class of neural networks: linear systems operating on a closed hypercube," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 11, pp. 1405–1422, November 1989.

[164] M. Negnevitsky, *Artifical Intelligence: A guide to intelligent systems*, 1st ed. Essex, England, UK: Addison Wesley, 2002.

[165] V. Kecman, *Learning and soft computing; Support Vector Machines, Neural Networks and Fuzzy Logic Models*, 1st ed. Cambridge, Massachusetts: MIT Press, 2001.

[166] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*, 1st ed. Belmont, MA, USA: Athena Scientific, 1996.

[167] E. Baum and D. Haussler, "What size net gives valid generalization," *Neural Computation*, vol. 1, no. 1, pp. 151–160, 1989.

[168] R. Caruana, S. Lawrence, and C. Giles, "Overfitting and neural networks: conjugate gradient and backpropagation," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, vol. 1, Como, Italy, July 24–27, 2000, pp. 114–119.

[169] A. Weigend, "On overfitting and the effective number of hidden units," in *Proceedings of the 1993 Connectionist Models Summer School*, vol. 1, San Mateo, CA, USA, 1993, pp. 335–342.

[170] A. Jain, "Data clustering: 50 years beyond K-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, June 2010.

[171] A. Jain, M. Murty, and P. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, September 1999.

[172] J. Kleinberg, "An impossibility theorem for clustering," in *Advances in Neural Information Processing Systems 15*. Cambridge, MA: MIT Press, 2003, pp. 446–453.

[173] A. Jain and R. Dubes, *Algorithms for clustering data*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 1988.

[174] G. Nagy, "State of the art in pattern recognition," *Proceedings of the IEEE*, vol. 56, no. 5, pp. 836–863, May 1968.

[175] F. Backer and L. Hubert, "A graph-theoretic approach to goodness-of-fit in complete-link hierarchical clustering," *Journal American Statistical Association*, vol. 71, no. 356, pp. 870–878, December 1976.

[176] J. Ward, "Hierarchical grouping to optimize an objective function," *Journal of American Statistical Association*, vol. 58, no. 301, pp. 236–244, March 1963.

[177] R. Sokal and F. Rohlf, "The comparison of dendrograms by objective methods," *Taxon*, vol. 6, no. 2, pp. 33–40, February 1962.

[178] H. Steinhaus, "Sur la division des corp materiels en parties," *Bulletin of the Polish Academy of Science*, vol. 4, no. 1, pp. 801–804, 1956.

[179] M. Anderberg, *Cluster Analysis for Applications: Monographs and Textbooks on Probability and Mathematical Statistics*, 1st ed. New York, USA: Academic Press, Inc., 1973.

[180] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Machine learning*, vol. 56, no. 1–3, pp. 9–33, July 2004.

[181] M. Meila, "The uniqueness of a good optimum for k-means," in *Proceedings of the 23rd International Conference on Machine Learning*, vol. 1, Pennsylvania, USA, June 25–29, 2006, pp. 625–632.

[182] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[183] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, 9th ed. New Jersey: Wiley-Interscience, 1990.

[184] G. Goodwin, S. Graebe, and M. Salgado, *Control system design*, 1st ed. Upper Saddle River, New Jersey, USA: Prentice-Hall, 2001.

[185] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle Filters for Tracking Applications*, 1st ed. London, UK: Artech House, 2004.

[186] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions ASME Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[187] R. Kalman and R. Bucy, "New results in linear filtering and prediction theory," *Transactions ASME Journal of Basic Engineering*, vol. 83, no. Series D, pp. 95–107, 1961.

[188] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," in *Proceedings of the IEEE*, vol. 92, no. 3, March 2004, pp. 401–422.

[189] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C++: The art of scientific computing*, 2nd ed. Cambridge, UK: Cambridge Press, 2002.

[190] J. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[191] G. Carlson, *Signal and Linear system analysis*, 2nd ed. New York, USA: John Wiley and Sons Inc., 1998.

[192] G. Das, K. Lin, H. Mannila, G. Renganathan, and P. Smyth, "Rule Discovery from time series," in *Proceedings of the 4th International Conference on Knowledge Discovery and Data mining*, vol. 1, New York, USA, August 27–31, 1998, pp. 16–22.

[193] N. Radhakrishnan, J. Wilson, and P. Loizou, "An alternate partitioning technique tp quantify the regularity of complex time series," *International Journal of Bifurcation and Chaos*, vol. 10, no. 7, pp. 1773–1779, July 2000.

[194] P. Cotofrei, "Statistical temporal rules," in *Proceedings of the 15th Conference on Computational Statistical*, vol. 1, Berlin, Germany, August 24–28, 2002, pp. 24–28.

[195] C. Schittenkopf, P. Tino, and G. Dorffner, "The benefits of information reduction for trading strategies," Report series for adaptive information systems and management in economics and management science, Tech. Rep. 45, 2000.

[196] T. Yairi, Y. Kato, and K. Hori, "Fault detection by mining association rules in house-keeping data," in *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in space*, vol. 1, Montreal, Canada, June 18–22, 2001, pp. 18–21.

[197] C. Aggarwal, A. Hinneburg, and D. Keim, "On the surprising behaviour of distance metrics in high dimensional space," in *Proceedings of the 8th International Conference on Database Theory*, vol. 1, London, UK, January 4–6, 2001, pp. 420–434.

[198] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is nearest neighbour meaningful?" in *Proceedings of the 7th International Conference on Database Theory*, vol. 1, Jerusalem, Israel, January 10–12, 1999, pp. 217–235.

[199] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Journal of Knowledge and Information systems*, vol. 3, no. 3, pp. 263–286, August 2001.

[200] A. Oppenheim, R. Schafer, and J. Buck, *Discrete-Time Signal Processing*, 2nd ed. New Jersey, USA: Prentice-Hall Signal Processing series, 1999.

[201] R. Bellman, *Adaptive control processes: A guided tour*. Princeton, New Jersey: Princeton University Press, 1961.

[202] M. Jakubauskas, D. Legates, and J. Kastens, "Crop identication using harmonic analysis of the time-series AVHRR NDVI data," *Computers and Electronics in Agriculture*, vol. 37, no. 1-3, pp. 127–139, November 2002.

[203] R. Juarez and W. Liu, "FFT analysis on NDVI annual cycle and climatic regionality in northeast Brazil," *International Journal of Climatology*, vol. 21, no. 14, pp. 1803–1820, December 2001.

[204] M. Chen, S. Liu, L. Tieszen, and D. Hollinger, "An improved state-parameter analysis of ecosystem models using data assimilation," *Ecological Modelling*, vol. 219, no. 3–4, pp. 317–326, December 2008.

[205] O. Samain, J. Roujean, and B. Geiger, "Use of a Kalman filter for the retrieval of surface BRDF coefficients with a time-evolving model based on the ECOCLIMAP land cover classification," *Remote Sensing of Environment*, vol. 112, no. 4, pp. 1337–1346, April 2008.

[206] J. Mendel, *Lessons in digital estimation theory*, 1st ed. The University of Michigan: Prentice-Hall, 1987.

[207] M. Nikulin, D. Commenges, and C. Huber, *Probability, Statistics and Modeling in public health*, 1st ed. 233 Spring street, New York, USA: Springer, 2005.

[208] M. Nikulin, N. Limnois, N. Balakrishnan, W. Kahle, and C. Huber-Carol, *Advances in degradation modeling: Applications to reliability, survival analysis, and finance*, 1st ed. 233 Spring street, New York, USA: Springer, 2010.

[209] R. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 15, no. 12, pp. 175–184, April 1970.

[210] B. Carew and P. Belanger, "Identification of optimum filter steady-state gain for systems with unknown noise covariances," *IEEE Transactions on Automatic Control*, vol. 18, no. 6, pp. 582–587, December 1973.

[211] G. Noriega and S. Pasupathy, "Adaptive estimation of noise covariance matrices in real-time preprocessing of geophysical data," *IEEE Transactions on Geoscience Remote Sensing*, vol. 35, no. 5, pp. 1146–1159, September 1997.

[212] M. Rajamani and J. Rawlings, "Estimation of the disturbance structure from data using semidefinite programming and optimal weighting," *Automatica*, vol. 45, no. 1, pp. 142–148, January 2009.

[213] R. Shumway and D. Stoffer, "An approach to time series smoothing and forecasting using the em algorithm," *Journal of Time Series Analysis*, vol. 3, no. 4, pp. 253–264, July 1982.

[214] R. Hirschowitz, "Mid-year estimates Statistical release," Statistics South Africa, Tech. Rep. P0302, 2000.

[215] P. Lehohla, "Mid-year population estimates," Statistics South Africa, Tech. Rep. P0302, 2010.

[216] A. Beaudette, D.E. nad OGeen, "Soil-Web: An online soil survey for California, Arizona, and Nevada," *Computers and Geosciences*, vol. 35, no. 10, pp. 2119–2128, October 2009.

[217] M. Clark and T. Aide, "Virtual interpretation of Earth Web-interface tool (VIEW-IT) for collecting land-use/land-cover reference data," *Remote Sensing*, vol. 3, no. 3, pp. 601–620, March 2011.

[218] L. Olsson, L. Eklundhb, and J. Ardo, "A recent greening of the Sahel-trends, patterns and potential causes," *Journal of Arid Environments*, vol. 63, no. 3, pp. 556–566, November 2005.

[219] V. Vanacker, M. Linderman, F. Lupo, S. Flasse, and E. Lambin, "Impact of short-term rainfall fluctuation on inter-annual land cover change in sub-Saharan Africa," *Global Ecology and Biogeography*, vol. 14, no. 2, pp. 123–135, January 2005.

[220] S. Mehrotra, "On the implementation of a Primal Dual Interior Point method," *SIAM Journal on Optimization*, vol. 2, no. 4, pp. 575–601, 1992.

[221] P. Gill, W. Murray, M. Saunders, and M. Wright, "Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints," *ACM Transactions on Mathematical Software*, vol. 10, no. 3, pp. 282–298, September 1984.

[222] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[223] M. Friedl, D. Sulla-Menashe, B. Tan, A. Schneider, N. Ramankutty, A. Sibley, and X. Huang, "MODIS collection 5 global land cover: algorithm refinement and characterization of new datasets," *Remote Sensing of Environment*, vol. 114, no. 1, pp. 168–182, January 2010.

[224] W. Kleynhans, "Detecting land-cover change using MODIS time-series data," Ph.D. dissertation, Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa, September 2011.

[225] M. Thompson, "A standard land-cover classification scheme for remote sensing applications in South Africa," *South African Journal of Science*, vol. 92, no. 1, pp. 34–42, January 1996.

[226] M. Thompson, H. van den Berg, T. Newby, and D. Hoare, "Guideline procedures for the National Land-Cover mapping and change monitoring," Council for Scientific and Industrial Research and Agricultural Research Council, Tech. Rep., March 2001.

APPENDIX A

PUBLICATIONS EMANATING FROM THIS THESIS AND RELATED WORK

A.1 PAPERS THAT APPEARED IN THOMSON INSTITUTE FOR SCIENTIFIC INFORMATION JOURNALS

- Salmon B.P., Olivier J.C., Wessels K.J., Kleynhans W., van den Bergh F., Steenkamp K.C.*"The use of a Multilayer Perceptron for detecting new human settlements from a time series of MODIS images"*, International Journal of Applied Earth Observations and Geoinformation, vol. 13, no. 6, December 2011, pp 873–883

- Salmon B.P., Olivier J.C., Wessels K.J., Kleynhans W., van den Bergh F., Steenkamp K.C.*"Unsupervised land cover change detection: Meaningful Sequential Time Series Analysis"*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 4, no. 2, June 2011, pp 327–335

- Kleynhans W., Olivier J.C., Wessels K.J., Salmon B.P., van den Bergh F., Steenkamp K.C.*"Improving land cover class separation using an extended Kalman filter on MODIS NDVI time-series data"*, IEEE Geoscience and Remote Sensing Letters, vol. 7, no. 2, April 2010, pp 381–385

- Kleynhans W., Olivier J.C., Wessels K.J., Salmon B.P., van den Bergh F., Steenkamp K.C.*"Detecting Land Cover Change Using an Extended Kalman Filter on MODIS NDVI Time Series Data"*, IEEE Geoscience and Remote Sensing Letters, vol. 8, no. 3, 2011, pp 507–511

- Kleynhans W., Salmon B.P., Olivier J.C., van den Bergh F., Wessels K.J., T.L. Grobler and Steenkamp K.C.*"Land Cover Change Detection Using Autocorrelation Analysis on MODIS*

Time-Series Data: Detection of new human settlements in the Gauteng province of South Africa", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, In press

- Ackermann E.R., Grobler T.L., Kleynhans W., Olivier J.C., Salmon B.P., and van Zyl A.J. *"Cavalieri Integration: a Novel Integration Technique"*, Quaestiones Mathematicae, In press

- Grobler T.L., Ackermann E.R., van Zyl A.J., Olivier J.C., Kleynhans W., and Salmon B.P. *"Synthesizing Multispectral MODIS Surface Spectral Reflectance Time Series Data"*, IEEE Geoscience and Remote Sensing Letters, In Press

- Grobler T.L., Ackermann E.R., van Zyl A.J., Olivier J.C., Kleynhans W., and Salmon B.P. *"Using Pages Cumulative Sum Test on MODIS time series to detect land cover changes"*, IEEE Geoscience and Remote Sensing Letters, In Press

A.2 PAPERS PUBLISHED IN REFEREED ACCREDITED CONFERENCE PROCEEDINGS

- Salmon B.P., Kleynhans W., van den Bergh F., Olivier J.C., Marais, W.J., Grobler T.L., Wessels K.J.,*"A search algorithm to meta-optimize the parameters for an extended Kalman filter to improve classification on hyper-temporal images"*, Accepted for publication, IEEE Geoscience and Remote Sensing Symposium 2012, Munich, Germany, 22 July - 27 July 2012

- Salmon B.P., Kleynhans W., van den Bergh F., Olivier J.C., Wessels K.J.,*"Detecting land cover change by evaluating the internal covariance matrix of the extended Kalman filter"*, Accepted for publication, IEEE Geoscience and Remote Sensing Symposium 2012, Munich, Germany, 22 July - 27 July 2012

- Grobler T.L., Ackermann E.R., van Zyl A.J., Kleynhans W., Salmon B.P., Olivier J.C. *"Sequential classification of MODIS time series"*, Accepted for publication, IEEE Geoscience and Remote Sensing Symposium 2012, Munich, Germany, 22 July - 27 July 2012

- Kleynhans W., Salmon B.P., Olivier J.C., van den Bergh F., Wessels K.J., Grobler T.L. *"Detecting land cover change using a sliding window temporal autocorrelation approach"*, Accepted for publication, IEEE Geoscience and Remote Sensing Symposium 2012, Munich, Germany, 22 July - 27 July 2012

- Kleynhans W., Salmon B.P., Olivier J.C.,, Wessels K.J., van den Bergh F.,*"A comparison of feature extraction methods within a spatio-temporal land cover change detection framework"*,

IEEE Geoscience and Remote Sensing Symposium 2011, Vancouver, Canada, 25 July - 29 July 2011

- Salmon B.P., Olivier J.C., Kleynhans W., Wessels K.J., van den Bergh F.,*"Automated land cover change detection: The quest for meaningful high temporal time series extraction"*, IEEE Geoscience and Remote Sensing Symposium 2010, Honolulu, Hawaii, United States, 25 July - 30 July 2010

- Kleynhans W., Olivier J.C., Salmon B.P., Wessels K.J., van den Bergh F.,*"A spatio-temporal approach to detecting land cover change using an extended Kalman filter on MODIS time series data"*, IEEE Geoscience and Remote Sensing Symposium 2010, Honolulu, Hawaii, United States, 25 July - 30 July 2010

A.3 INVITED CONFERENCE PAPERS IN REFEREED ACCREDITED CONFERENCE PROCEEDINGS

- Salmon B.P., Olivier J.C., Kleynhans W., Wessels K.J., van den Bergh F.,*"The quest for automated land cover change detection using satellite time series data meaningful high temporal time series extraction"*, IEEE Geoscience and Remote Sensing Symposium 2009, Cape Town, South Africa, 12 July - 17 July 2009

- Kleynhans W., Olivier J.C., Salmon B.P., Wessels K.J., van den Bergh F.,*"Improving NDVI time series class separation using an extended Kalman filter temporal time series extraction"*, IEEE Geoscience and Remote Sensing Symposium 2009, Cape Town, South Africa, 12 July - 17 July 2009

- Kleynhans W., Salmon B.P., Olivier J.C.,, Wessels K.J., van den Bergh F.,*"An autocorrelation analysis approach to detecting land cover change using hyper-temporal time-series data"*, Joint invite for publication, IEEE Geoscience and Remote Sensing Symposium 2011, Vancouver, Canada, 25 July - 29 July 2011

A.4 PAPERS SUBMITTED TO REFEREED ACCREDITED CONFERENCE PROCEEDINGS

- Kleynhans W., Salmon B.P. *"Monitoring informal settlements using SAR polarimetry"*, Submitted for review, African Association of Remote Sensing of the Environement (AARSE) 2012

A.5 BEST PAPER AWARD

- Salmon B.P., Kleynhans W., van den Bergh F., Olivier J.C., Marais, W.J., Wessels K.J.,*"Meta-optimization of the extended Kalman filter's parameters for improved feature extraction on hyper-temporal images"*, IEEE Geoscience and Remote Sensing Symposium 2011, Vancouver, Canada, 25 July - 29 July 2011

LIST OF TABLES

LIST OF FIGURES