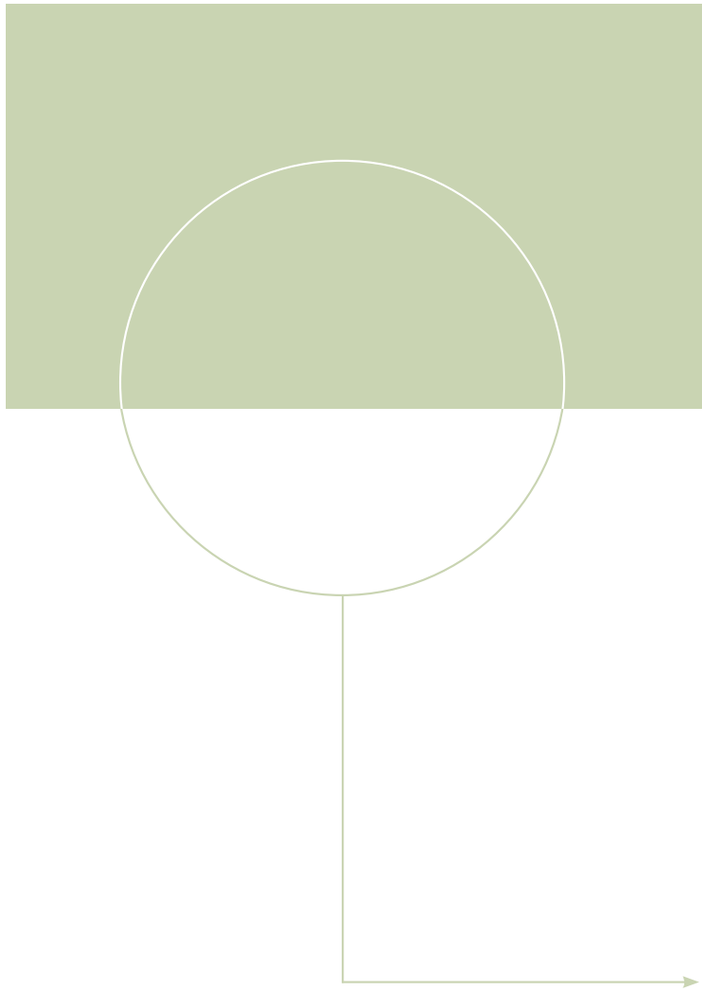

Master's thesis

2019

Eivind Grimstad

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Computer Science

Eivind Grimstad

Evolutionary multi-objective optimization for band selection of hyperspectral imagery using a cluster-based representation

July 2019

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Evolutionary multi-objective optimization for band selection of hyperspectral imagery using a cluster-based representation

Eivind Grimstad

Computer Science
Submission date: July 2019
Supervisor: Pauline Catriona Haddow

Norwegian University of Science and Technology
Department of Computer Science

Abstract

Our planet is changing, now more than ever before. Understanding these changes and how they impact the environment is crucial for preserving the Earth for the coming generations. Improvements in remote sensing technology and data collection allows us to harvest more data than ever. Hyperspectral remote sensors gather data about electromagnetic radiation reflected off the Earth in nearly the entire spectrum of light emitted from our sun. This data can enable the classification and monitoring of changes in vegetation, agricultural areas, water contents, human habitation, natural disasters and so much more. However, the high dimensionality and redundancy of this data provides a unique challenge for machine recognition. This work presents a new technique for doing unsupervised spectral band selection of hyperspectral data, based on clustering bands in highly correlated subspaces and multi-objective evolutionary search using NSGA-II. Experiments show promising results on several popular hyperspectral datasets compared to other similar recent methods, indicating that this is an interesting avenue for further research.

Preface

My journey toward this thesis started when I in the summer of 2018, somewhat on a whim, asked Professor Pauline Haddow in an email about opportunities for a specialization project connecting bio-inspired computation and space technology. She immediately responded and got in touch with Dr. Didier Keymeulen at JPL, NASA. He offered to set up a joint project between NTNU and JPL to investigate the use of evolutionary algorithms to optimize hyperspectral image sensors. Although much has happened and changed since then and the planned trip to work at JPL in Los Angeles had to be cancelled, I am indebted to Dr. Keymeulen for introducing me to this field of research and for his supervision and hard work in the early stages of the project.

I would like to deeply thank my main supervisor, Professor Haddow, for her guidance throughout the entire process. Without her feedback, encouragement and insight, the thesis would not be what it is now. She has always pushed me and shown I could do better, while still trusting me in my decisions. Every meeting and discussion renewed my faith in the work and provided me with new and useful angles of approach. Thank you.

Deserving of many thanks is also my wonderful girlfriend Frida, who has been at my side with unyielding faith. She has been patient and encouraging, always there to listen to my problems and give useful comments from outside the master bubble. I could not have done this without you.

Eivind Grimstad Trondheim, July 9, 2019

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Hyperspectral images provide high dimensional data able to detect and classify surface feature to a high degree of certainty. Instead of relying primarily on the spatial shape of materials, which would require a high spatial resolution, it records images in a large number of spectral bands, so that different land covers can be recognized according to their shape in the spectral dimension. This technology allows aircraft and satellites to cover large areas that can be used for many applications like agriculture, environmental change observation, urban evolution, mineral mapping and more. The high dimensionality of these images in the spectral dimension provide not only an unprecedented recognition ability but large challenges to data storage, transmission and processing. Since spectral bands close to each other in the electromagnetic spectrum share properties, much of data volume collected is redundant and leads to difficulties for classification algorithms such as the curse of dimensionality. For this reason, much research has been done in the field of dimensionality reduction of hyperspectral images. Specifically band selection, which aims to pick out some selected bands from the available has been a popular technique. Band selection techniques need a measure of the information content and redundancy between bands to select the best ones accordingly, which can be either supervised or unsupervised. While supervised techniques are great for optimizing the bands selected for a specific application, it needs already labelled data samples to function, which require expert knowledge to acquire. Unsupervised techniques require only the information present in the image itself to select the best bands and is therefore good both since it requires no expert knowledge and because it is not tailored to a specific application and can thus be more generalizable.

This thesis explores the use of evolutionary algorithms for doing unsupervised band selection of hyperspectral images. It establishes a state-of-the art within the field and proposes a new algorithm that makes use of the redundancy between bands in order to

create clusters of similar bands where a single band can be chosen as the representative. It is done through a multi-objective genetic algorithm that provides a set of optimal solutions with a different number of selected bands that can be chosen between in order to support different needs. The proposed algorithm is experimentally evaluated and compared with other recent evolutionary band selection algorithms, and shows competitive, stable performance on several different hyperspectral datasets.

1.1 Goals

The goal of the thesis is to provide insight into the following research question:

> How can low redundancy band clustering be used together with multi-objective evolutionary search in order to select representative bands from a hyperspectral image in an unsupervised manner?

The goal is decomposed into the following research questions will be employed to guide the experiments conducted:

Research questions

RQ1 How good is NSGA-II at selecting band subsets of different lengths that provide stable and predictable results?

RQ2 What divergence measure best captures the different information captured in adjacent spectral bands?

RQ3 How do the number of subspaces affect classification accuracy?

RQ4 What is the relative importance of individual band informativeness compared to the redundancy between selected bands?

RQ5 How does the presence of noisy bands affect classification accuracy?

1.2 Methods

There are two primary methods used in this thesis, a literature review establishing state of the art, and experimental studies. The purpose of the literature review is to provide an insight into the motivation of choosing the research goal and to present relevant recent work that is used as a comparison to the proposed model. The experimental studies

are conducted in order to evaluate how the proposed model can answer the research questions and ultimately the research goal.

1.3 Pre-project

In the fall of 2018 a pre-masters project was undertaken for TDT4501 entitled *Hardware optimization of remote sensing imaging spectrometers: A review*. It was supervised by Professor Haddow at NTNU and Dr Keymeulen at JPL, NASA. The goal of the project was to establish the state of the art for the online configuration optimization of push-broom hyperspectral sensor hardware. The goal was to identify how evolutionary techniques could be applied to the optimisation of the focal plane array with the intention to continue the project as a masters thesis at JPL in Los Angeles. Unfortunately, restrictions imposed by NASA regarding both access to documentation and hardware involved prevented continuation of the project to the experimental phase, despite much effort by Dr Keymeulen. As such, the project was stopped and a new masters topic was identified.

While this thesis shares some of the theoretical background, particularly the basic theory of remote sensing, the project has moved to hyperspectral data processing rather than hyperspectral sensor hardware.

1.4 Outline

The thesis is organized as follows. Chapter 2 gives an introduction to the theoretical background used in the work. It is separated into four sections: Remote sensing, evolutionary algorithms, information theory and classification. Then, chapter 3 describes the literature review work done in establishing the state of the art in band selection of hyperspectral imaging. It gives an overall discussion of what techniques have been applied and their strengths and weaknesses, as well as an in-depth discussion of some selected related work. Chapter 4 presents the proposed model used for band selection. It also introduces the simulator implemented in order to test and verify the model. Chapter 5 presents the plan and results of the experiments done to explore and verify the properties and quality of the model and how it compares to the other related work. Lastly, 6 will be a discussion of the results, the main contributions and recommendations for future work.

Chapter 2

Background and Theory

This chapter gives an insight into the theoretical background of the work done in the thesis. The background can be separated into four different fields, in the intersection of which this thesis lies. The source of the data used comes from the field of remote sensing, described in section 2.1. This section describes the basics of electromagnetic radiation and how it is used to record information and gain knowledge about the Earth's surface. It also discusses the novel challenges with processing hyperspectral data, to lay the basis for the methods used in the rest of the thesis.

Then, section 2.2 provides an introduction to evolutionary optimization and search, both for single-objective and multi-objective problems, and describes some of the algorithms used. In order to facilitate the discussion of how evolutionary computation is fit to solve some of the challenges in the processing of hyperspectral data, the section includes a discussion of the advantages and disadvantages of using evolutionary optimization.

Since the task at hand is unsupervised processing of hyperspectral data, there is a need for establishing inherent quality metrics for recorded hyperspectral data. This is covered in section 2.3, where information theory is used to create measures of information content in an image and measures of the difference between images.

Finally, one of the basic and useful applications of hyperspectral data, is land cover classification. Section 2.4 describes classification as a machine learning task and what challenges hyperspectral data provide for this task.

2.1 Remote Sensing

Remote sensing can be defined as the act of gathering information about a target without direct contact. In the broadest form the information to be gathered can be anything, and the target can be anything, but usually when discussing remote sensing, the information comes from electromagnetic radiation and the target is the Earth, observed from either airborne or on-orbit platforms. Therefore, remote sensing is also often referred to as Earth observation. As such, remote sensing is a remarkably powerful and interesting tool for gathering information about how the surface of our planet is changing and how human civilization is affecting it. The application areas are widespread, from observing changes in rain forest density, agriculture, disaster relief, mineral mapping and certainly to surveillance applications. This theory section will describe the basics of remote sensing, starting with a description of the information source, electromagnetic radiation, and a summary of different types of sensors used in remote sensing and their advantages and disadvantages.

This section will describe the basics of hyperspectral imaging. First it the basic principles of spectroscopy, and then give an overview of how it is being used within remote sensing.

2.1.1 Electromagnetism

Electromagnetism is one of the most basic and omnipresent phenomena in physics. It is used to study both vast cosmological events and the tiniest atomic ones. Most objects that have mass interact with electromagnetism in one way or another, either through transmission, emission or reflection. This interaction is a useful tool for understanding the composition of the object. It is established that electromagnetic radiation can equivalently be thought of both as a wave and as a stream of particles called photons. The wave representation is useful since it allows the radiation to be classified according to wavelength or frequency. Since the speed of light is a constant c, the relationship between the wavelength λ and the frequency ν can be expressed as

$$\nu = \frac{c}{\lambda} \tag{2.1}$$

Since the wavelength and frequency are inversely proportional, the two properties can be used interchangeably to describe the radiation. The wavelength places it somewhere in the electromagnetic spectrum, as shown in Figure 2.1.

Most electromagnetic signals observed are not composed of a single wavelength, but rather a mixture of many component wavelengths with different intensity. The view of electromagnetic radiation as a stream of photons is useful to describe this intensity. Since

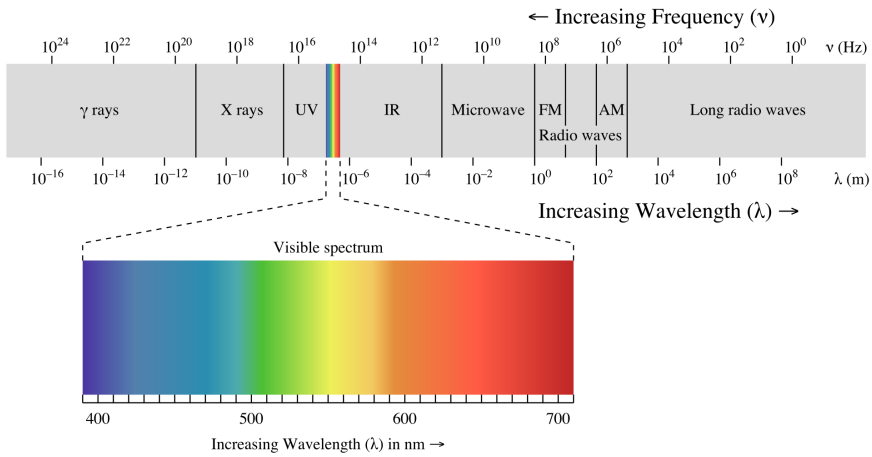

Figure 2.1: The electromagnetic spectrum

Source: EM spectrum, used under CC BY 3.0

the photon energy, contained within a single photon, is proportional to the wavelength of the radiation (h being the Planck constant), the energy of the radiation is, in a sense, the number of photons, same as the number of electrons defines the current in a circuit. The energy of an electromagnetic signal is measured in joules (J), and by differentiating with regards to time, the resulting quantity is called the **radiant flux** of the system, denoted Φ and having the unit of watts (W).

Electromagnetic radiation has a source, being an object that either transmits the radiation from another source, emits it through some process, or reflects it. Based on the properties of the object, the radiant flux of the signal coming from the object may vary based on the direction it is viewed from. The directional quantity that describes the radiant flux differentiated by the viewing direction is called the **radiance** of the object. Conversely, when an object receives electromagnetic radiation the quantity used to denote the energy present is called **irradiance**. This is the spectral flux received differentiated by the area of the surface.

Considering that all light consists of several wavelengths, the quantities of radiant flux, radiance and irradiance can also be differentiated with regards to the wavelength of the light, in which case the quantities are called spectral flux, spectral radiance and spectral irradiance.

2.1.2 Sources of electromagnetic radiation

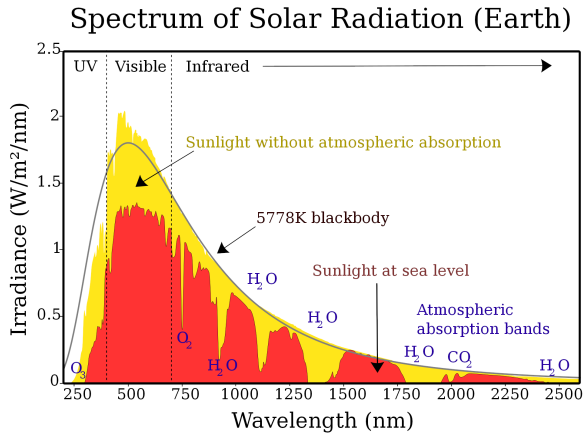

Figure 2.2: **Electromagnetic radiation from the sun.** The black line shows the spectral signature of the reference blackbody with the same temperature as the sun. The yellow area shows the received sunlight above the atmosphere, while the red area is received sunlight at sea level.

Source: Solar spectrum by Wikipedia user Nick84, used under CC BY-SA 3.0

The unarguably most important source of emitted electromagnetic radiation in the universe are stars. Stars are dense balls of hot plasma that emit electromagnetic radiation due to fusion of hydrogen and helium at its core. The frequency of the initially produced light is in the gamma ray part of the spectrum, but as the light works its way towards the surface it interacts with the material in the sun and is converted into lower frequency light. A useful model for describing solar radiation is the black body model. The black body model assumes that all the radiation coming from an object is from emitted energy. It is black in the sense that it absorbs all light it receives. In the early 20th century, physicist Max Planck discovered that the spectral radiance emitted by a blackbody is dependent on its absolute temperature T. This discovery is formalized in Planck's law, which is be used to determine the radiance of emitted solar radiation, by modelling the Sun as a black body with a temperature of 5778 K. Using the distance between the Sun and the Earth, this enables determining the irradiance of solar radiation at the Earth. The black curve in Figure 2.2 shows the predicted solar irradiance using the black body model. The yellow area shows observed spectral irradiance of sunlight at the Earth above the atmosphere. The black body curve pretty closely determines sunlight without factoring in the atmosphere. It shows that sunlight consists of wavelengths from around

200 nm in the ultra violet (UV) part of the spectrum, through visible (VIS, 380 nm-740 nm), near-infrared (NIR, 740 nm-1400 nm), to some parts of the short wavelength infrared spectrum (SWIR, 1400 nm-3000 nm).

The Earth itself also emits electromagnetic radiation according to its black body equivalent, but at wavelengths a lot higher than the Sun, due to the Earth having a much lower temperature than the Sun.

2.1.3 Electromagnetic interactions in the atmosphere

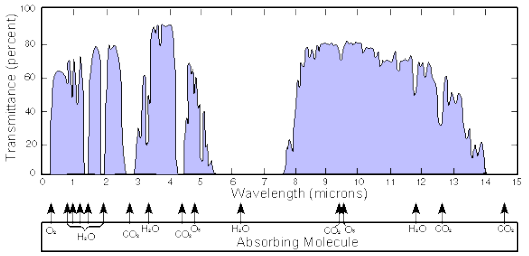

Figure 2.3: **Electromagnetic transmittance through the atmosphere of the Earth.**

Before the sunlight reaches the Earth's surface, it has to be transmitted through the atmosphere. The atmosphere of the Earth is composed of many different gases, like nitrogen (N), oxygen (O_2), carbon dioxide (CO_2), ozone (O_3) and water (H_2O). The transmittance of sunlight through the atmosphere is thus dependent on its interaction with these gases and their absorption spectra. The transmitted radiant flux can be denoted Φ_t. The transmittance of the medium is then

$$T = \frac{\Phi_t}{\Phi}$$

where Φ is the incident radiant flux. An approximate model of the transmittance of sunlight through the atmosphere is shown in Figure 2.3. The figure shows transmittance in microns (micrometers). It is the part of the figure that is between 0.2 and 2.4 microns, the solar emitted wavelengths that are interesting for this study. Ultra violet (UV) rays of wavelengths most dangerous for life on Earth, 200 nm to 300 nm, are absorbed by the layer of ozone in the stratosphere. The other atmospheric gases, most notably oxygen, carbon dioxide and and water vapour also close off the atmosphere for some wavelengths. This closing off is shown by the low transmittance values at some parts in the spectrum. In the solar emitted wavelengths, it is especially water vapour that is the most troublesome, blocking out large parts of the NIR and SWIR spectra. This

phenomenon can also be seen in Figure 2.2 where the red area shows the sunlight that reaches the surface. Water vapour content is additionally notoriously difficult to model, since it can have large local variations, both in time and space.

The sunlight that has been so lucky to get through the atmospheric windows are ready to be reflected off the surface of the Earth. Although different kinds of material reflect the light in different ways, its reflection can usual be modelled as Lambertian (Figure 2.4). A Lambertian reflection off a surface means that the radiance is constant in any direction.

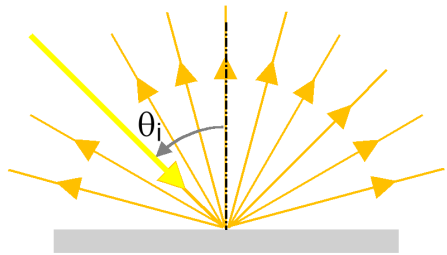

Figure 2.4: **A Lambertian surface** has a constant radiance across any viewing angle θ

Source: Cropped from Lambertian reflection by Wikipedia user Panjasan, used under CC BY 3.0

This kind of reflection is also called diffuse reflection, and is a good approximation since most surfaces are rough and will therefore reflect light in many directions. Non-diffuse surfaces are for example mirrors, which are specular reflectors.

However, depending on the material the incident radiant flux Φ is distributed between reflection Φ_r, Φ_a absorption and transmission Φ_t. Thus,

$$\Phi = \Phi_r + \Phi_a + \Phi_t$$

It is also useful to define reflectance as

$$R = \frac{\Phi_r}{\Phi}$$

As with Φ_t, Φ_r is also a function of wavelength. This function indicates the ratio of the incident light to the reflected light of each wavelength, and is essentially what the human eye interprets as the color of the material in the visible spectrum. Figure 2.5 shows the reflectances of some materials.

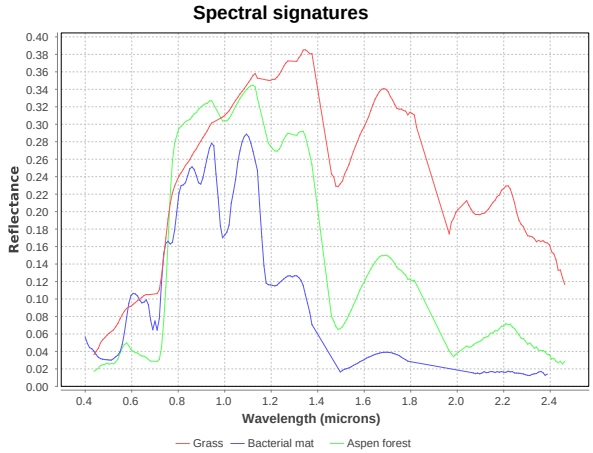

Figure 2.5: Reflectance signatures across solar emitted wavelengths of some different materials. Data fetched from Kokaly et al., 2017

Since different types of material have significantly different spectral reflectances, they can be used as a "signature" to identify the material. The figure shows that the three different materials have significant differences in reflectance values across the entire range of solar emitted wavelengths. So it is not only what is visible light to humans (0.4 - 0.7 microns) that is useful for recognizing a material, but infrared light also. Visible light has no physical significance apart from the fact that it is in the wavelengths the sun emits the most energy of, so eyes sensitive to those wavelengths will see better.

2.1.4 Remote Sensor Technology

Figure 2.6: **Sentinel-1B and Sentinel 2.** Two ESA satellites used for remote sensing. Sentinel-1B uses active radar technology to observe the Earth, while Sentinel-2 uses a passive multispectral sensor.

Source: ESA

A remote sensor system must minimally consist of the sensor itself, capable of recording incoming light, a platform the sensor is mounted upon, a storage unit that can store the data, and usually some communication device that can transmit the data to the ground. The following will briefly describe some these components, focusing on the deployment platforms, the sources of radiation that can be used, and sensor technologies. It will not be a thorough study but will provide enough information needed for the thesis work.

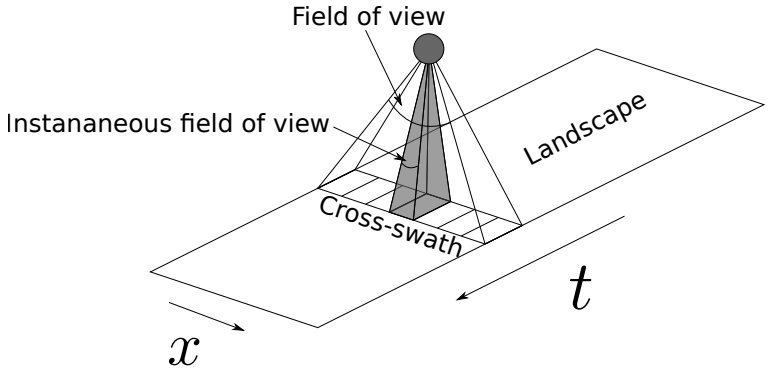

Figure 2.7: Simple model of a remote sensor

Platform deployment In its simplest form a remote sensor can be modelled as in Figure 2.7. It shows a remote sensing platform, the grey circle, moving in the t direction at a certain height across a landscape.

A big difference between remote sensing and conventional photography is that remote sensing platforms are "scanning" the surface as it moves, gathering row by row of information instead of forming an entire image with the camera standing still. The remote sensor has has a field of view (FOV), which is the angle that determines the area that is visible for the sensor at a particular point in time. On the surface, this angle forms the *swath width* (also called cross-swath in the figure). The instantaneous field of view (IFOV) is the angle available to one sensor element in the remote sensor, which on the surface becomes the spatial resolution of the sensor image. The concept of sensor elements will be explained further.

Remote sensing platforms can be deployed either on aircraft or in orbit mounted on a satellite. Aircraft mounted remote sensors have the advantages that they are easy to deploy to a specific location, and since they are closer to the surface they are sensing, a sensor with the same IFOV will have a higher spatial resolution. On the other hand, a satellite mounted sensor both moves a lot quicker in the t direction, and will have a much wider swath width. The orbit chosen for the satellite determines to a large degree how effective it can be at recording data. Remote sensors usually want to cover the entire surface of the Earth in a shortest possible amount of time. To do this, they are normally deployed in a polar orbit. This is an orbit that is rotated 90 deg from the equator of the Earth, and means that for each revolution of the orbit, the Earth will have rotated underneath the sensor so it covers new ground every orbit.

Radiation source Remote sensing platforms can be either active or passive. Active remote sensors, like the Sentinel-1B in Figure 2.6 have light emitting devices attached. Examples of active sensor technologies are RADAR which emits radio waves, and LIDAR which emits visible or infrared light. Active remote sensing has the advantage that it can observe a target at any time of day, without requiring direct sunlight on the target. Additionally, active sensors that use high wavelength radiation such as radio waves for RADAR, are uninterrupted by atmospheric conditions. However, active sensors require a significant amount of energy to operate.

Passive remote sensing, on the other hand, usually gets the radiation it measures from the sun. For observing surface features on the Earth, using solar emitted radiation is the most useful. Being dependent on the sun may reduce the availability of data from the sensor, but there are also solutions to this problem. For example, the ESA satellite Sentinel 2, shown in Figure 2.6 is placed in a sun-synchronous orbit, meaning the ground below it will have equal daylight at all times.

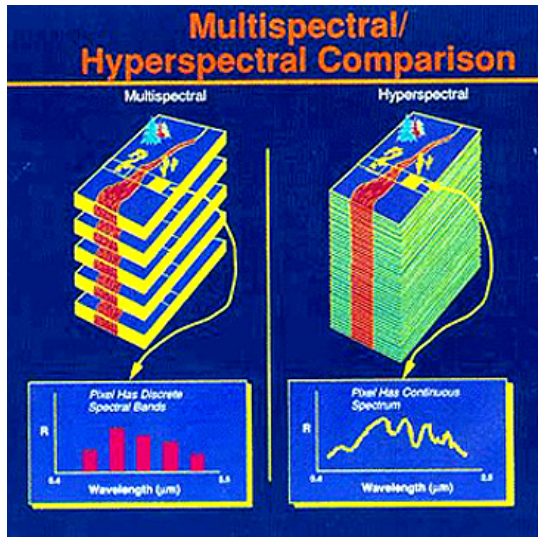

Figure 2.8: **Hyperspectral imaging compared to multispectral.** Hyperspectral images consist of narrow, contiguous bands, with each pixel a continuous function of wavelength, while multispectral images are discrete.

Source: NASA

Sensor array Although all remote sensors have some sensor array to capture the incoming light, the nature and configuration of this sensor array will vary greatly, both in terms of what light it gathers and how it functions. Generally though, it exists of a set of mirrors and lenses that disperse the light into wavelength bands and focuses the light on light sensitive devices such as a diodes. The diodes create electric currents which can be integrated with a capacitor or other processes. At equal intervals, called the exposure time, the charge in the capacitors are read and they are discharged. The charge recorded is then proportional to the radiance of the light received.

The lenses in the sensor disperse the light into bands, consisting of light in a small part of the spectrum, but how wide these bands or how many there are will vary greatly from sensor to sensor. If the light is dispersed into only a few bands which are not adjacent, the sensor is called *Multispectral*. One such sensor was the Landsat-1, one of the first remote sensing platforms deployed for Earth observation. The attached MSS (NASA, 2019a) sensor gathered radiation in four different discrete wavelength bands, making it a multispectral sensor.

While multispectral sensors are useful, they only gather information about a few

bands in the spectrum, as shown in the left side of Figure 2.8. This means that its capability to recognize spectral signatures such as the ones shown in 2.5 is limited. Often the bands in a multi-spectral system are chosen to recognize specific materials.

For improving the capabilities of the system to recognize a wide range of spectral signatures, *Hyperspectral* sensors were developed. These sensor disperse the light into much smaller bands which are adjacent to each other. This means that the data recorded from the sensor is able to approximate the real spectral signatures of a wide range of materials and land covers to a much larger degree. A model of a hyperspectral sensor is shown in Figure 2.9.

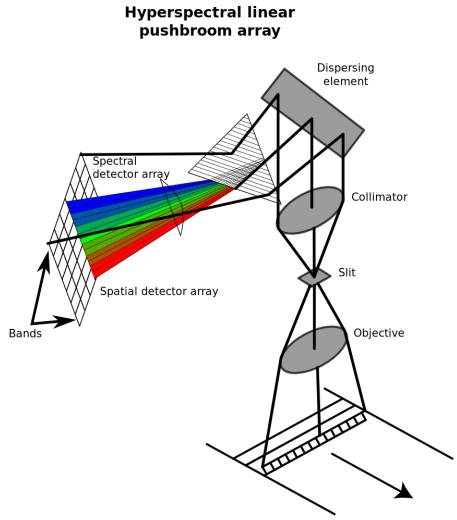

Figure 2.9: A hyperspectral sensor with a focal plane array (FPA)

Source: NASA

Light is dispersed and focused upon a two-dimensional array called the focal plane array, in which one dimension is the spectral and the other is one of the spatial dimension. As the platform scans the surface it produces a two-dimensional array of radiance values at each time interval, which taken together forms a *hyerspectral image cube* like the one shown on the right side of Figure 2.8.

One of the first and definitely most important hyperspectral sensors is the AVIRIS, the Airborne Visible-Infrared Imaging Spectrometer, developed by NASA in the 1980's and still in use today (NASA, 2019b). The sensor has a swath width of 11 km when deployed to an airplane a height of 20 km. The AVIRIS gathers data in 220 adjacent

wavelengths, from $400 - 2400\,\text{nm}$, which is across the entire range of solar emitted wavelengths, as seen in Figure 2.2.

2.1.5 Data processing

Recognizing land covers and surface features on the Earth is useful for many applications. However, recognizing detailed features only based on their shape in the spatial dimension might in some cases require an unfeasibly high resolution. This is one of the main motivations for using hyperspectral sensors in data analysis, as described by D. Landgrebe (2002), as it allows the recognition of land covers based on their spectral "shape" rather than their spatial. Still, hyperspectral sensors also require large amounts of data due to its generality, and for a specific application much of this data would be redundant and even counterproductive. Therefore, much research in hyperspectral imaging have focused on techniques to select the necessary and desired features for a specific task, as a form of dimensionality reduction. See Chapter 3 for a discussion of the state of the art.

2.2 Evolutionary Optimization

Evolutionary optimization is a sub-field of *evolutionary computation*, whose purpose and goal is to design optimization algorithms inspired by natural, biological processes such as evolution, genetics or animal behavior. Since the prevailing theory of how humans and the other life forms have been shaped throughout the ages can be interpreted as a continuous and heuristic optimization of our ability to survive in the environment, it is both tempting and almost obvious to try to harness this power on our own optimization tasks. Due to this sub-symbolic and model-free nature, evolutionary algorithms may be readily applied to a wide range of problems.

Evolutionary computation has a long history of research, stretching almost as far back as information theory and modern computing itself. One of the first developed and most well-known algorithms in evolutionary computation is the *Genetic algorithm* (GA). The genetic algorithm will be the main search framework implemented for the work in this thesis, and so a significant portion of this chapter is devoted to explaining how it works, as well as advantages and limitations. Another evolutionary algorithm that will be briefly described is the Artificial Bee Colony (ABC), implemented in this thesis so as to compare the proposed method against another work that uses ABC.

2.2.1 Optimization problems

An *optimization problem* is a task where the purpose is to find an x that maximizes (or minimizes), a function $f(x)$. The function f is often called an objective function. x can be anything, such as a number, a vector of numbers or even a string. The set of allowed values for x is usually called the set of *feasible solutions* to the problem, while the domain of x on f is called the *search space*. The maximum value of f across the entire domain is called the global maximum. f might also have several local maxima, defined as a point where all neighboring points have a lower value. Conversely, f will have local and global minima, which would be the targets for optimization if minimization is the goal.

An optimization algorithm has to search through the search space of x in order to find the value that maximizes f. The size and shape of the search space has a large effect on what optimization algorithm should be used for the problem. If f is a simple continuous mathematical function defined over all real numbers, calculus can provide a provably global maximum. However, most real world problems can not be easily modelled as such a continuous, differentiable function. In these cases it might be difficult to find provably optimal solutions in reasonable (polynomial) time, and the only feasible way of solving the problem is to find a solution that is "good enough", given some constraints. A well-known class of such problems are the NP-hard problems. In simplified terms,

NP-hard are problems where finding an optimal value of f is difficult, but verifying the value of f for a given x is easy. The definition of "easy" in computational complexity theory is that the number of computations required to solve the problem is bounded by a polynomial expression with regards to the size of the input problem.

In order to illustrate and explain the optimization algorithms described in this chapter, the famous NP-hard travelling salesman problem (TSP) will be used. The goal of the TSP problem is to find the shortest route to visit all cities, once and only once, and then return to where one started. It can be represented as a graph, where the nodes represent cities and the edges are paths between them. The edge weights are the distance between the cities. A feasible solution as described above is then called a Hamiltonian cycle. Figure 2.10 shows the problem instance that will be used.

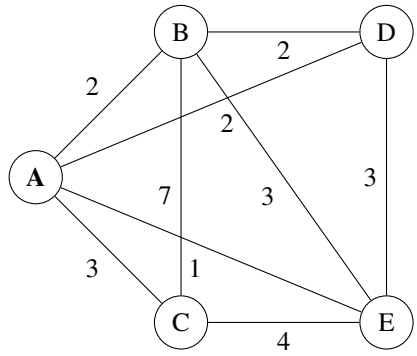

Figure 2.10: An instance of TSP. The starting and ending city is **A**

In TSP, x is a route between the cities, and its feasible values are all Hamiltonian cycles in the graph, and f is the total weight of all the edges visited. The search space of the TSP problem will grow exponentially as the number of cities and paths between them increases, and therefore finding an optimal solution is generally infeasible. However, finding the length of a given route, calculating f, is an easy procedure.

2.2.2 Genetic algorithm

The genetic algorithm is one of the most well-known kinds of evolutionary algorithms, and has been studied extensively, in many variations, since it was first proposed in the 1950's. A simple genetic algorithm is illustrated in Figure 2.11. The basic premise is to investigate the search space guided by a meta-heuristic inspired by evolution and natural selection. The algorithm operates on a set of candidate solutions, called the "population". At every iteration of the search, also called "generation", the algorithm

combines and mutates solutions in the population to form new ones, and then selects the next generation based on the fitness of the individuals. The selection procedure can be implemented in many different ways. This process is done until some termination condition is met, which can be either a fixed number of generations, or that a good enough solution has been found.

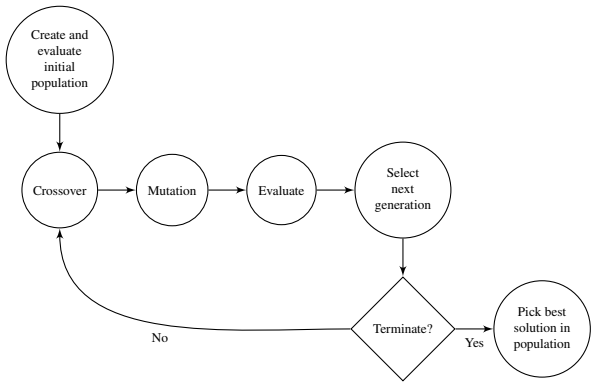

Figure 2.11: Genetic algorithm

The next paragraphs will introduce the various steps in a genetic algorithm.

Representation

When designing a genetic algorithm, the programmer has to decide on how to represent a solution to the problem at hand. How the solution is represented has a great effect on both the performance and efficiency of the algorithm. From terms used in genetics, the representation of the problem used for performing genetic operators is called the "genotype". It is often a fixed-length string of numbers, to make it easy to apply general genetic operators. However, depending on the problem to be solved, it might be anything. For TSP, a naive representation for a solution would a string of all the cities to be visited. An example is:

$$S = ACEDB$$

$$S = 13542$$

Since the algorithm should to be able to combine and mutate solutions in order to create new and better solutions, it is desirable that sub-strings of the genotype in some sense represents a solution to a sub-problem. If this is the case, recombination should be

able to take good parts from different solutions to create an even better one. This can be easily seen in the TSP example above, as it may be that visiting city *C* after city *A* is a good solution, but not the rest.

Initialization

The first thing that needs to be done when starting a run of a genetic algorithm is to initialize the population. The simplest way to do this is to initialize them all at random. Since the algorithm does not yet have any data on the viability of each solution or the topology of the problem space, random initialization should make sure the initial population is well spread out in the space. However, using heuristics based on the problem domain to initialize the population might lead to better initial conditions for the search and therefore also a quicker convergence to the optimum. For the TSP a heuristic for initialization could be a nearest neighbor approach, where cities are iteratively added based on the one nearest to the previously added. Heuristic initialization can also hamper the search, though, by limiting the exploration of the search space and converging prematurely on suboptimal solutions.

Genetic operators

The technique genetic algorithms use to discover new solutions in the search space is to apply genetic operators. There are two primary kinds of genetic operators, crossover and mutation. Crossover operators have two solutions as its input, while mutation operators only have one. The genetic parallel to crossover is reproduction, where the two inputs are the parents and the output is the child, while mutation is supposed to represent random genetic switches. It is widely believed that evolution could not happen without mutation, since without it there would be no way of generating entirely new combinations. In this way, it can be said that mutation operators are needed in order to **exploit** already existing good solutions, while crossover is needed to **explore** the search space for novel solutions.

In order to select which individuals should be chosen for the genetic operators, a standard procedure is called tournament selection. In tournament selection, *k* individuals are chosen randomly from the population, and the one with the best fitness value is chosen for the genetic operator. This allows for a good trade-off between exploiting good solutions by further improving them, and allowing exploration.

Crossover operators The n-point crossover is a standard crossover operator. It chooses n random points in the genetic string and then alternately selects subsets from each parent from each substring divided by the random points. For many applications though, this crossover will generate infeasible solutions, and new operators have to be designed.

For the TSP problem, a crossover operator could be the following. Select a random chunk from one of the parents, and place it in the same position in a new solution, the child. Then, sequentially add the cities not already added from the other parent. Two generate two new solutions, the same procedure could be done selecting a chunk from the second parent.

Mutation operators Mutation operators generally make a single simple change to a solution. If the genetic string is a bit string, it could involve just flipping a random bit. In TSP, being a combinatorial problem, a mutation operator could involve choosing two random cities and swapping their positions.

Selection

After the algorithm has applied operators in order to create new solutions, as well as evaluated their fitness, it needs to select which solutions that should be used as the population in the next iteration. This is the equivalent of the famous term "survival of the fittest". Selection operators can differ greatly in different genetic algorithms, but generally, only choosing the solutions with the highest values of the objective functions is a recipe for premature convergence into local optima. There usually needs to be a mechanism for letting some randomness into the process, so that the search space can be explored more widely.

A method for selecting the next generation is called roulette-wheel selection, in which the probability of each individual to be chosen is proportional to its fitness. This means that good solutions are more likely to be chosen, but suboptimal ones also has a chance. A variation is the rank selection, in which the probability is based on an individuals rank within the population, instead of the absolute fitness value. This ensures that large differences in fitness values does not mean that some solutions will have extremely low/high probabilities of being chosen.

There might however, be an advantage to retain some of the best solutions in each iteration. This is called *elitism* and is often used to ensure that good solutions are not lost as the algorithm progresses.

Termination

The termination of a genetic algorithm is usually done either after a fixed number of iteration, or if no improved solutions are created, indicating that the algorithm has converged.

Hyperparameters

The genetic algorithm has several parameters, called hyperparameters, that can be tuned to optimize its performance. The optimal values of these parameters are generally determined by the dataset although some sensible defaults are available.

The population size of genetic algorithms can influence its performance greatly. Generally the population size should be at least big enough that a diverse population can be kept. When it is large enough can depend on the dimensionality of the problem and the topology of the search space, and must often be established through trial-and-error. Large population sizes might often lead to better results, but also require more computational power.

For applying genetic operators, there needs to be established the parameters crossover rate and mutation rate. These rates determine how many of the individuals are selected for the application of crossover and mutation operators. It is understood through experience that the crossover rate should be significantly higher than the mutation rate, so a value of 0.9 to crossover and 0.1 for mutation is usually a good starting point. Like the population size, the values should be explored in trial-and-error.

2.2.3 Artificial Bee Colony

Artificial Bee Colony (ABC) is another metaphor based heuristic optimization algorithm, that takes its inspiration from the behavior of foraging honey bees. The candidate set of solutions ABC works on, is a population of honey sources that the bees are attracted to. The bees are divided into three types, the employed bees, the onlooking bees and the scout bees. The employed bees attach themselves to the best food sources in the population and try to improve them, by exploring neighboring food sources. The onlooking bees look at the food sources attached to the employed bees and randomly chooses one to improve based on its fitness, while the scout bees replace stale food sources with entirely new sources. These three steps, as well as a recording of the currently best solution, happen at each iteration of the algorithm.

2.2.4 Multi-objective optimization

While many problems can be modelled using one objective function, there are also problems in which there are several, competing objectives that are not easily weighed against each other or chosen between. These problems are called multi-objective optimization problems (MOP).

The goal of multi-objective optimization is to extend optimization algorithms to handle the optimization of more than one objective function simultaneously. It means that

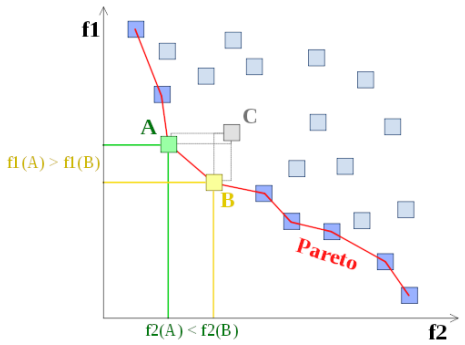

Figure 2.12: A Pareto front between objectives f_1 and f_2. A and B are in the Pareto front since no other solution is better in both objectives, while C is not, since both A and B have better values in both objectives.

Source: Front pareto, used under CC BY 3.0

the objective $\mathbf{F}(\mathbf{x}) = \{f_1, ..., f_n\}$ if n is the number of objective functions. This is useful particularly when these objective function represent competing axes of optimization and a suitable trade-off between them is difficult to establish. In this type of problem, an optimal solution is not represented by a single parameter, but rather by a set of parameters that provide an optimal value for different trade-offs between the objective functions. This set of solutions is known as the Pareto front of the given problem. Any single solution along the Pareto front has the property that it is impossible to improve the solution by putting more weight on one of the objective-functions, since it will lead to a decrease in optimization for one of the other objective-functions. From this, it can be said the multi-objective optimization attempts to approximate the Pareto front of the problem as close as possible. An example Pareto front is shown in Figure 2.12

While multi-objective optimization can be applied in many contexts, the focus here will mainly be how it can be used in order to extend genetic algorithms to handle multi-objective problems. Since the calculation of fitness is essential for a genetic algorithm, there needs to be a way to express the "Pareto fitness" of a given solution, and to compare solutions in a way that makes selection and evolution approach the Pareto front. It is also desirable that the solutions are as evenly spread out across the Pareto front as possible to give access to as many different unique trade-offs between the objective functions as possible.

Non-domination and NSGA-II

Non-domination is a technique for solving MOO problems with evolutionary optimization. It is based on building a set of non-dominated solutions to constitute the Pareto front. A solution is dominated by another if it has a better value in every objective function. Using this notion of domination, the EA can sort the individuals in the population by the number of solutions it is dominated by.

The most well-known and used non-domination based evolutionary algorithm is NSGA-II, proposed by Deb et al. in 2002. NSGA-II is based on the genetic algorithm, and uses non-domination sorting to determine the most fit solutions in the population. The main improvement of NSGA-II over earlier non-domination based genetic algorithm is that it requires fewer hyperparameters in the algorithm. Earlier algorithm for example necessitated determining a sharing parameter to ensure the even spread of solutions in the Pareto front. Having fewer parameters means the algorithm is easier to generalize.

NSGA-II differs from the general GA process shown in Figure 2.11 only in the "Select next generation" step. The selection is based on a non-dominated sort process, shown in Figure 2.13. In iteration i with the population P_i of length N, the genetic operators create new solutions Q_i based on P_i. NSGA-II combines these two into $R_i = P_i \cup Q_i$. Then R_i is sorted using the non-dominated sort and crowding distance sort algorithms, and the N best individuals are chosen as the next generation.

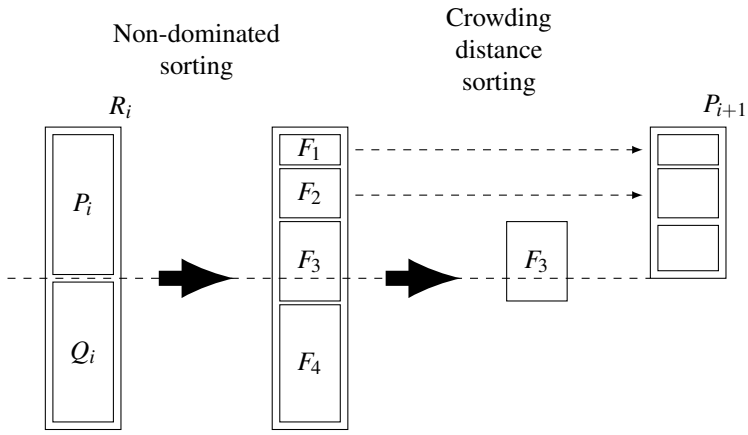

Figure 2.13: Selecting the next generation in NSGA-II. Adapted from Deb et al., 2002

The non-dominated fronts F_i in Figure 2.13 are created by sorting the individuals according to how many other individuals they are dominated by. The individuals in F_1

are not dominated by any solutions, while F_2 are dominated by 1 other solution and so on. The non-dominated front that lies in the cutoff line where solutions should be included in the next generation, F_3 in the example, is internally sorted using the crowding distance sorting algorithm. This algorithm sorts the individuals based on their distance in all of the objectives to the other individuals, so the most diverse set of individuals is sorted first and thus selected for the next generation.

When the termination condition has been met the Pareto front is the first non-dominated front.

Decomposition

Decomposition-based MO is based on decomposing the problem into a set of single objective optimization problems. This decomposition is done by creating a set of trade-off weights between the objectives and assigning a weight to each individual in the population. Then, the fitness of the individual is determined based on the weighed sum between the objectives. The weights ensure that each individual exists in an even distribution across the Pareto front.

A popular multi objective decomposition technique, that is used in the literature, and implemented as a comparative method in this thesis, is the Tchebycheff decomposition method. In the Tchebycheff optimization method, the fitness value of each subproblem in the population is defined as

$$g^{tch}(\mathbf{F}(\mathbf{x}), \mathbf{w}, \mathbf{z}^*) = \max_{i=1}^{n} w_i(f_i(\mathbf{x}) - z_i^*)$$

where $\mathbf{w}$ is the weights between the objective functions and $\mathbf{z}^*$ are the ideal points of the objectives. The ideal point of each objective is the minimal value found so far.

The Tchebycheff decomposition applies genetic operators only through a crossover operation where a subproblem is crossed with one of its neighboring subproblems. The neighborhood of a subproblem is defined as the k solutions with the closest weight values in Euclidian distance. k is a tunable hyperparameter. When the crossover operator is applied, the solution is replaced by the new one, if it has a better fitness value.

2.3 Information Theory

Quantity	Formula
Entropy	$H(X) = -\sum_{x \in X} P(x) log_2 p(x)$
Kullback-Leibler divergence	$D_{KL}(P,Q) = \sum_{x \in X} P(x) log_2 \frac{P(x)}{Q(x)}$
Joint entropy	$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} P(x,y) log_2 Q(x,y)$
Mutual information	$MI(X,Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) log_2 \frac{P(x,y)}{P(x)Q(y)}$
Disjoint information	$DI(X,Y) = H(X,Y) - MI(X,Y)$

Table 2.1: **Information theoretic quantities**. X and Y are discrete random variables taking values $\{x_1,...,x_n\}$ and $\{y_1,...,y_n\}$. $P(X)$ is the probability mass function of X, and $Q(Y)$ is the probability mass function of Y. $P(X,Y)$ is the joint probability mass function of X and Y.

Information theory is a field that concerns the basic aspects of how information is stored and transmitted. It revolves around a very general term called information entropy, which describes the expected number of bits required to store or transmit a message, based on the probability distribution of message values. Information theory was introduced by Claude Shannon (Shannon, 1948), and can be applied to a wide range of applications. For this project, information theory will be used to quantify the amount of information present in hyperspectral images, both in absolute term and the relative overlap of information that exists between bands. The relevant quantities are listed in Table 2.1 and will be discussed in the following sections.

2.3.1 Information entropy

Information entropy measures how much information is expected to be gained by observing a value of a random variable. When using 2 as base for the logarithm, the unit for the entropy H is in bits, which is obviously useful from a computer science perspective.

To consider some simple examples, one can look at the flipping of a coin and the

rolling of a dice. For flipping a coin, X can take the values $\{Heads, Tails\}$, and $P(X) = 0.5$. That means that

$$H(X) = -(0.5 \times log_2(0.5) + 0.5 \times log_2(0.5)) = 1$$

So the act of flipping a coin has an information entropy of one bit. The interpretation of this can be that if the person flipping the coin were to communicate the result of the flip to another person, he would have communicate the answer to one binary (Yes/No) question. A related example is rolling a dice. The entropy of that action is $H(X) \approx 2.584$. In that case the person rolling the dice would expect to need, in average, to answer 2.584 questions. For example, if the result of a roll is 4:

1. Is it larger than 3? YES
2. Is it larger than 5? NO
3. Is it 5? NO

In the example, the number of required questions was three, but if the answer had been 6, there would only be two required questions. This is why Shannon entropy is considered to be the expected, or average, information content in a message.

Another interesting and relevant interpretation of information entropy, is as an upper bound for the lossless compression of data. If a receiver of a file knows and expects a file to be a long string of one of two characters (with equal probability), a sender only has to transfer one bit of data. On the other hand, if each character in the message can be on of many, with a given probability function, the entropy would be much higher.

2.3.2 Kullback-Leibler divergence

Kullback-Leibler divergence (KL divergence) is another measure in information theory, introduced by Solomon Kullback and Richard Leibler (Kullback and Leibler, 1951). It also sometimes called relative entropy, and is used to measure the difference between two probability distributions defined over the same value space. A popular interpretation of the Kullback-Liebler divergence is that it represents the number of additional bits one would need if the person receiving the message thinks it is comes from the distribution Q, while it actually comes from P.

KL divergence is not symmetric, so $D(P,Q) \neq D(Q,P)$. This makes the KL divergence unfavorable for use as a distance metric, and for this reason it is usually symmetrised. Symmetrised KL divergence is simply $D_{sym}(P,Q) = D(P,Q) + D(Q,P)$. The symmetrised KL divergence will be zero if P and Q is the same distribution, and non-negative.

2.3.3 Mutual information

Mutual information (MI) is quantity that akin to the KL divergence, describes a relationship between two distributions. While KL divergence is defined over one variable with different distributions, mutual information is defined over two different variables. It uses the joint probability distribution between them. Mutual information is a measure of the similarity between the two variables, and can be interpreted as how much information is gained about Y by observing X.

2.3.4 Disjoint information

Disjoint information (DI) is a quantity defined from the joint entropy and mutual information, that describes the information in Y and X that are not mutual between them. As such it is related to the KL divergence since it describes a divergence rather than a similarity.

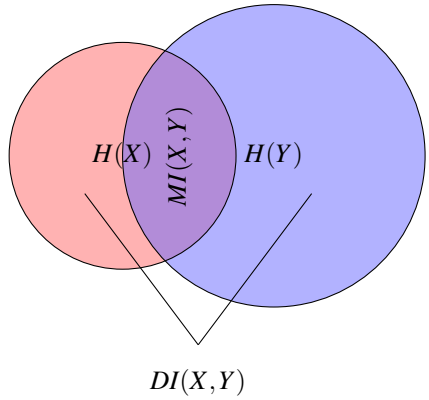

Figure 2.14: **Venn diagram showing how different quantities relate.** The red circle shows the entropy of variable X, $H(X)$, while the blue circle shows $H(Y)$. The entire colored area is the joint entropy, $H(X,Y)$ The overlapping area, in purple, is the mutual information $MI(X,Y)$, which means the divergent information $DI(X,Y)$ is the colored area excluding the purple area.

2.3.5 Correlation coefficient

The correlation coefficient is not strictly a information theory measure, as it does not use probability measures. Instead it draws on samples of any two variables. In any case it

is a useful measure that similar to KL divergence, MI and DI can describe the similarity between two variables. The correlation coefficient between to variables b_i and b_j each taking n different values, can be defined as

$$R(i,j) = \frac{\sum_{k=1}^{n}(b_{i,k} - \bar{b_i})(b_{j,k} - \bar{b_j})}{\sqrt{\sum_{k=1}^{n}(b_{i,k} - \bar{b_i})^2}\sqrt{\sum_{k=1}^{n}(b_{j,k} - \bar{b_j})^2}} \tag{2.2}$$

where $\bar{b_i}$ and $\bar{b_j}$ are the average values of the two variables.

2.3.6 In image processing

Information theoretic quantities can be applied to any message as long as its probability mass function (pmf) P can be established. In order to apply it to hyperspectral images, there must be established a way to find the pmf of an image. Since a hyperspectral image can be considered to be an image with N channels, where N is the number of spectral bands, each band can be considered a one channel grey scale image. The pmf of each channel can then be found using the normalized grey level histogram. The grey level histogram is found by putting all the pixels of the image in one of N bins. For an exact histogram, N should be the number of grey levels in the image, so 2^d where d is the bit depth of the image. If the bit depth is one byte there should be $2^8 = 256$ bins.

To illustrate, this chapter will use the Indian Pines dataset recorded by the AVIRIS sensor.

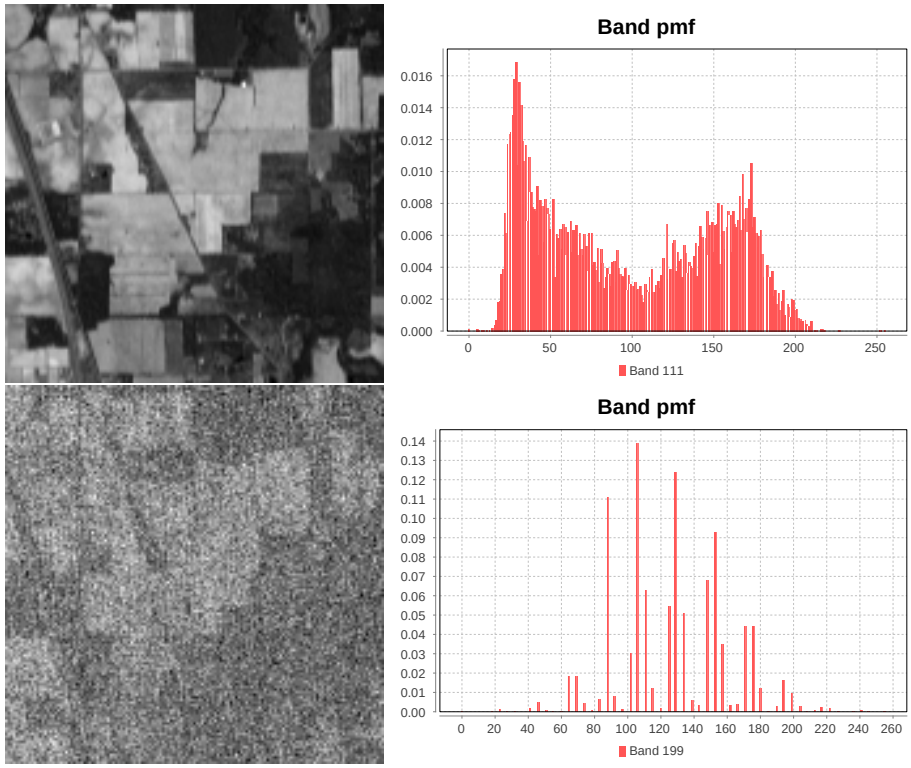

Figure 2.15: Two bands from the Indian Pines dataset together with their probability mass functions.

Entropy Figure 2.15 shows two bands from the Indian Pines dataset. The first image shows the 111th band, which has high entropy. This can be visually confirmed by seeing that the images has little noise and high contrast, and that the features can be easily distinguished. The pmf is well distributed across all of the bands, which in turn explains the high entropy. The second image shows band number 199, which has low entropy. As the image shows, it has high noise and it is difficult to distinguish the same features as in the other image. The pmf shows that the intensity levels are concentrated in a few bins with high probability, which is why the entropy is low. These two images confirm that entropy is a good measure for describing the quality of a hyperspectral image. Figure 2.16 shows the entropy of all the bands in the Indian Pines dataset. The two example bands used in Figure 2.15 shows that band 111 and 199 are respectively one of the highest and lowest entropy bands in the image.

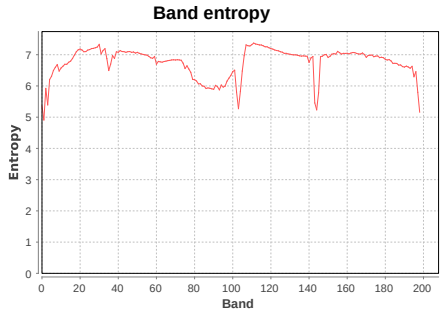

Figure 2.16: Entropies of all bands in the Indian Pines dataset.

Divergence measures This chapter has introduced several quantities that can be used to compare information content of different random variables and pmfs. Most of them measure divergence, except for mutual information which measures the similarity. To visualize these measures on hyperspectral images, the adjacent bands in the image will be compared as shown in figure 2.17.

These tables show the divergence measures applied to adjacent pair of bands, so between bands (0,1), (1,2), (2,3) and so on. The KL-divergence measure shows a topology with a minimum value of zero where there are quite a few "spikes" in the spectrum where there is divergence between the bands. The maximum value is between bands 0 and 1 and the reason for this is that these are both low entropy bands (see Figure 2.16). Since low entropy bands usually means they are noisy, they difference between them will naturally be higher. The MI graph is "upside-down" compared to the others since it measures similarity rather than divergence. It shows a similar topology to the KL divergence in the way that bands near the edge shows low mutual information (so high divergence). Generally, the local extremes are at the same points in the spectrum, but the relative importance in between them is different. DI is closely related to MI. The main difference that can be seen between them is that DI has higher values in extremes that are at bands with high entropy. This is the main purpose of introducing the DI. For example, the edges of the spectrum does not have nearly as high relative values as in the other measures. Lastly is the correlation coefficients. This measure is unrelated to the information theory measures and only uses statistical measures on the band samples themselves. It is therefore natural that this measure is significantly different from the others. Most of the local minima in the correlation coefficient graph lies between band 70 and 110. It is not obvious why this is the case by looking at the available visualizations.

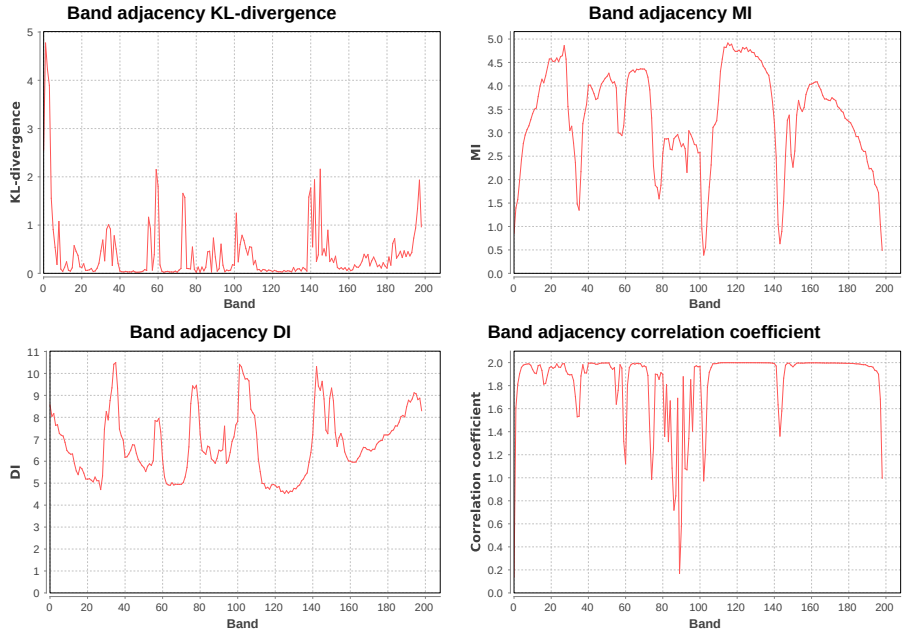

Figure 2.17: Four comparison measures visualized on adjacent bands of the Indian Pines dataset.

2.3.7 Summary

This section has introduced the measures that will be used to identify the information content in hyperspectral images. Several competing measures have been introduced to describe the relative information content between bands. See Chapter 3 for how these have been applied in the literature, and Chapter 5 for experiments on the comparison of using the measures as objective functions for the proposed algorithm.

2.4 Classification

Classification is an important problem in statistics and data analysis that seeks to place data points into discrete sets, called classes (y). A data point is simply a vector of scalars $\mathbf{x}$, or any other property that can be meaningfully represented as such. The basic premise of classification is that there exists some f, so that

$$f(\mathbf{x}) = y$$

for every $\mathbf{x}$ in the domain, where y is one of the classes. This f might be arbitrarily complex, but a mapping must exist for predictive classification to be meaningful. By using existing, labelled examples taken from f, a classification algorithm, or classifier, is supposed to "learn" an approximation of f, which can be called $\hat{f}$, given the data it has available. Then, previously unseen data can be fed into $\hat{f}$. Algorithms that use prelabelled data points are called **supervised**, while techniques that do not require this are called **unsupervised**.

For the work in this thesis, classification will be used as an example application for hyperspectral data, used for performance verification in experiments. This section will give some insight into the unique challenges with classification in such high-dimensional data as hyperspectral images are. It will also give a brief introduction to the classifier that will be used in the experiments, the support vector machine (SVM), as implemented in the eminent LIBSVM (C.-C. Chang and Lin, 2011).

2.4.1 High-dimensional data

When the feature vector $\mathbf{x}$ has a high dimensionality, many classifiers suffer from a problem known as the "curse of dimensionality" or alternatively called Hughes phenomenon. This problem involves that the achieved classification accuracy (number of correctly labelled classes) will start to decrease as the number of features reaches a certain value. The reason why this problem occurs is that the space of feature values increases dramatically for each new dimension is introduced, so the amount of labelled data needed to properly estimate f also increases.

In order to alleviate this problem, the dimensionality of the feature vector should be lowered somehow. There are two main techniques to do this, feature selection and feature extraction. Feature selection tries to find a subset of the features in x that ensures that as much as possible of the information of the original features are retained. Feature extraction, on the other hand, seeks to map the feature space to a new, lower dimensional space.

2.4.2 Support vector machines

Support vector machines (SVM) is a classification algorithm that is based on finding the hyperplane in the feature space that can split the data into two classes with the widest possible margin. A hyperplane is any vector

$$\mathbf{w} * \mathbf{x} - b = 0$$

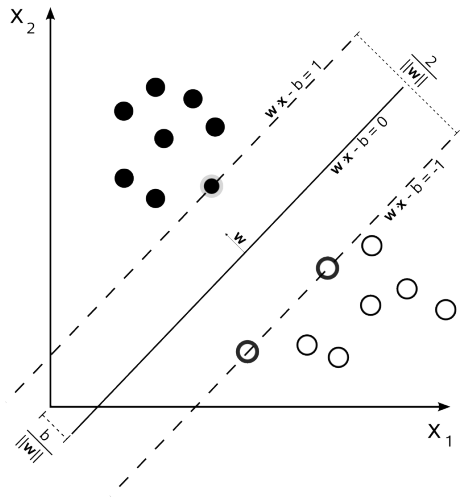

Figure 2.18: A maximum margin hyperplane separating data points belonging to two different classes. This hyperplane is the output model of an SVM classifier.

SVM finds the maximum margin hyperplane by using two additional hyperplanes

$$\mathbf{w} * \mathbf{x} - b = 1$$

and

$$\mathbf{w} * \mathbf{x} - b = -1$$

The goal of SVM is then to find $\mathbf{w}$ and b to maximize the distance between the two margin hyperplanes, without any of the data points falling in between the two planes. Since not all data is linearly separable, the algorithm has a "soft margin" version which allows data to fall inside the margins at the penalty of some loss function. Then the soft-margin SVM should optimize based on a trade-off between maximizing the margin and

ensuring that as few as possible of the data points are inside the margin. This trade-off is often encoded as a parameter C in the algorithm.

In this most basic form it is a binary linear classifier, meaning it can classify linear data into one of two classes. However, many extensions of the algorithm exists to allow it to handle both multi-class problems and non-linear data.

Non-linear data is handled through the use of feature space mappings called kernels. The kernels transform the original feature space into a higher dimensional space in which the data is linear. Then the maximum margin hyperplane can be found for the transformed space, even if none can be found in the original space. One such popular kernel is the radial basis function, the RBF kernel. This kernel transforms the space based on the euclidean distance between the data points, and requires the specification of another parameter, often called γ.

For multi-class classification, SVM can be applied by training a set of classifiers, each trained on two classes. LIBSVM implements the "one-against-one" one method. It involves training one classifier for each pair of classes. When the classifier is do determine the class of a new, unseen point, all of the classifiers vote on which class they think it belongs to, and the class with the highest vote count wins.

Chapter 3

Motivation and State of the Art

This chapter will establish the state of the art within the field of band selection of hyperspectral images. First it will go through the literature review protocol to describe which publications are important to the field and how papers were reviewed and selected. Then there will be a general overview of the various techniques that have been applied for band selection to get an idea about the current state of research. Then there will be a more thorough discussion of two applied techniques that are used as the basis for the work conducted in this thesis. Finally there will be a summary of the findings.

3.1 Literature review protocol

Important publications

- IEEE International Geoscience and Remote Sensing Symposium
- IEEE Transactions on Geoscience and Remote Sensing
- IEEE Geoscience and Remote Sensing Letters
- Applied Soft Computing

Keywords

- Unsupervised
- Hyperspectral

- Band selection
- Multi-objective
- Evolutionary optimization
- Subspace decomposition
- Shannon entropy
- Mutual information
- Kullback-Leibler divergence

Search engines

- **IEEE Xplore** is the publisher of most of the important publications identified and has a powerful search engine for search entire database or within a given publication
- **Web of Science** is used for exploring relationships between papers, finding either parent or child references

Research paper criteria Papers found using the keywords, important publications on the search engines are chosen based on whether they meet some criteria

- They should either
  1. Describe a relevant unsupervised objective function, or
  2. Describe an evolutionary algorithm, preferably multi-objective, for solving band selection
- They should describe a new technique that is not been applied to hyperspectral band selection before
- They should report competitive results compared with other similar band selection algorithms

3.2 State of the art

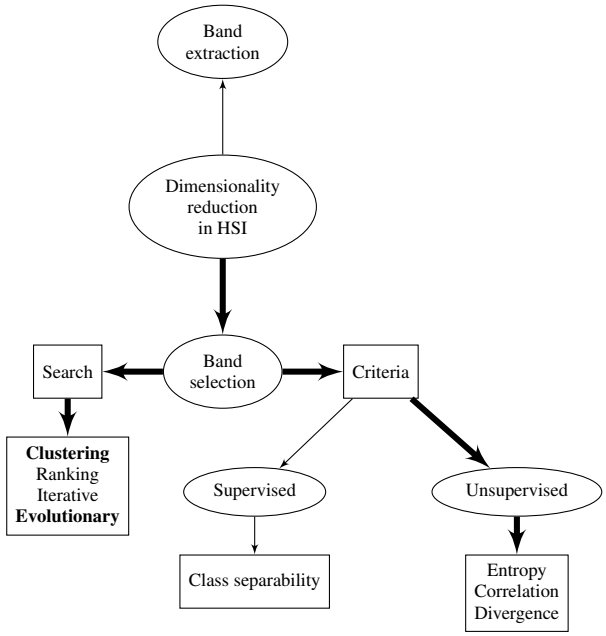

Figure 3.1: **Overview of techniques.** The thick lines follow the main line of focus.

With the introduction of multispectral and hyperspectral sensors for remote sensing in the 1970's and 1980's, for example the AIS developed by NASA (Vane, Goetz, and Wellman, 1984), it was established that there were several new challenges with this type of sensor. Wiersma and D. A. Landgrebe, 1980 identifies that dimensionality reduction is necessary in order to decrease computational load, lower the amount of training data needed for classification, and to optimize general purpose sensor hardware for specific problem domains. Since multi-and hyperspectral sensors are complex and expensive to build and launch, it would be desirable that sensors could be built for general purpose, with many spectral bands available, and that the useful bands for a specific application can be chosen thereafter. And although this optimization can be done by sensor configurability, Wiersma and D. A. Landgrebe, 1980 also concludes it is useful in the post-processing and classification of the data.

In hyperspectral imaging especially, in which the spectral bands collected are adjacent to each other in the spectrum, dimensionality reduction is furthermore important due to the large redundancy between adjacent bands.

A large and varied set of techniques for selecting representative bands from hyperspectral images have emerged over the years. Many early efforts were focused on component analysis for feature extraction. Since bands are highly correlated, Principal component analysis (PCA) (Harsanyi and C.-I. Chang, 1994) and related methods such as Independent component analysis (Jing Wang and Chein-I Chang, 2006) have been applied in order to extract a small set of uncorrelated features.

Several authors, such as Ye Zhang et al., 1999, have criticized feature extraction methods for not retaining the original radiance values of the bands, and thus losing possibly crucial information about the observed scene. An important contribution in the effort to mitigate this was Ye Zhang et al., 1999, who proposed a method to adaptively separate the hyperspectral bands into similar subspaces and run PCA on each of the subspaces. They used the correlation coefficient metric to describe the similarity of bands. Although this helps in retaining local structures in the data, feature extraction methods still makes the data less useful for further processing and physical modelling (Petrie, Heasler, and Warner, 1998). Therefore, band selection methods are useful.

Band selection methods attempt to pick out the bands in the image cube that contain the most representative information about the observed scene. Several challenges appear in band selection, and the literature describes a varied set of ways to solve them.

Band selection is a special instance of the general feature selection problem, which seeks to select the best features in any given feature vector. Feature selection algorithms usually consist of two components. The first component is how the method judges the amount of information present in a feature or a set of features. This is a measure of the quality of a feature, and determines whether or not it should be present in the selected feature subset. The second aspect of feature selection algorithms is the search procedure. The search procedure determines how the algorithm navigates the space of possible feature combination in order to find the one that maximizes the information content.

3.2.1 Band quality measures

For measuring the quality of a band or set of bands, methods can primarily be either supervised or unsupervised. Supervised quality measure take into account already labelled training data, and select bands that maximally discriminate between the target classes, for example using Bhattacharyya distance (Serpico and Bruzzone, 2001), a measure often used in feature selection for determining the distance between variables belonging to different classes. These methods are great at targeting a specific problem area, since the bands can be optimized for the given labels exactly, but consequently are also not very generalizable and require manual labelling of data.

On the other hand, unsupervised quality measures are based on the inherent infor-

mation present in the hyperspectral data, and are usually based on information theoretic measures such as Shannon entropy, mutual information and Kullback-Leibler divergence, or statistical measures such as correlation coefficients. Since these methods do not require any manual labelling of data, they are much less dependent on expert knowledge.

The use of information theoretic measures for band selection of hyperspectral images has been considered since the early 1990's (Conese and Maselli, 1993; Sotoca, Pla, and Klaren, 2004) and used together with many different search algorithms. There are two main approaches to using information theory. Some authors choose to approach it via the entropy of selected bands (Gong, M. Zhang, and Yuan, 2016), while some approach it from redundancy and mutual information between the bands (Guo et al., 2006). Approaching from the entropy side of things will make sure that only the most representative bands are selected, but will not guarantee that the selected bands have different information contained in them. The mutual information approach, however, does not guarantee that the bands selected have a high entropy.

This is especially apparent when using the standard mutual information measure. Several authors (Hossain, Jia, and Pickering, 2012; R. Yang et al., 2017) point out that it does not work as a distance metric properly. Its inadequacy as a measure can furthermore be shown by considering the Venn diagram of information measures, shown in Figure 2.14. Since it only measures the overlap between the two variables, a high mutual information can either happen for very similar low entropy variables, or less similar high entropy variables. In the context of HSI, this means that bands with high noise and therefore low entropy will score high on the mutual information scale and encourage the selection of these bands. Hossain, Jia, and Pickering, 2012 proposed to use a measure called the normalized mutual information (NMI) to handle this. NMI divides the mutual information by the product of the marginal entropies of the two variables, to remove the dependency on entropies of the individual variables. Another measure was proposed in R. Yang et al., 2017 called the disjoint information (DI), which instead of dividing by the marginal entropies, takes the joint entropy of the two variables and subtract the mutual information. This way, the measure will always favour variables with large individual entropies, a favorable property for band selection. Figure 2.17 shows visualizations of the various measures on hyperspectral data.

Many authors choose to consider the problem as a combination between the two properties of redundancy and information content (Feng et al., 2016; Lorencs, Mednieks, and Sinica-Sinavskis, 2018; M. Zhang, Gong, and Chan, 2018), as a way to ensure that high entropy bands with high mutual information are not used together. Lorencs, Mednieks, and Sinica-Sinavskis, 2018 solved this in a greedy band selection algorithm by normalizing the entropy on their correlation with already selected bands, while M. Zhang, Gong, and Chan, 2018 proposed a criterion called maximum information, minimum redundancy (MIMR). Using a combination of the two measures helps to make sure

that the selected bands have low redundancy, plus ensuring that no noisy, low-entropy bands are chosen.

3.2.2 Search techniques

Since band selection is established to be a computationally difficult problem to solve optimally, most methods use a heuristic search technique based on its quality measures. Some techniques are based on greedy selection of bands (Lorencs, Mednieks, and Sinica-Sinavskis, 2018; Du and H. Yang, 2008). These techniques sequentially choose the best bands (Lorencs, Mednieks, and Sinica-Sinavskis, 2018; Du and H. Yang, 2008) based on their relation with previously selected bands. Another class of techniques is clustering (Martinez-Uso et al., 2007; Wang, F. Zhang, and Li, 2018; Xie et al., 2019), in which adjacent bands are clustered together in highly correlated subsets. These methods take advantage of the redundancy of adjacent spectral bands by selecting representative bands from different subspaces in the spectrum.

The algorithm proposed in Martinez-Uso et al., 2007 is based on a hierarchical clustering method, where clusters are iteratively merged based on their similarity. They start the algorithm with all bands in their own cluster and then iteratively merges clusters one by one. They propose variations using several different measures of similarity between bands, including the KL-divergence and mutual information. For selecting the representative band for each cluster, they select the band with the highest average similarity to the other bands in the cluster. R. Yang et al., 2017 uses a k-means clustering based approach based on the disjoint information measure. Selection of the representative band for each cluster is based on an iterative procedure that maximizes similarity with bands in the cluster and minimizes similarity with selected bands from other clusters. Both these methods suffer in the presence of noisy bands, since the selection of representative bands are only based on similarity and not information content, and can therefore risk the selection of noisy bands to represent clusters. Datta, S. Ghosh, and A. Ghosh, 2015 proposed an algorithm that combines the features of clustering and ranking algorithms. They first do clustering as a way to remove redundant bands, and then rank the bands according to their informativeness to select the best ones. They also introduce the possibility of having bands not included in any cluster at all, and rather defined as noisy bands not belonging in any cluster. This helps the algorithm perform better than the other methods.

Another class of algorithms use evolutionary techniques that optimize some objective function of the selected bands through heuristic search procedures. Evolutionary techniques are of interest in this domain due to their search capabilities and ability to find solutions in complex spaces. For high-dimensional problems which hyperspectral images are, evolutionary search have favourable properties since it is able to find good solutions without an exhaustive search of the search space. L. Zhang et al., 2007 used a

artificial immune system (AIS) and clonal selection based algorithm for supervised band selection. A solution was represented as a binary string where value i indicated whether or not the ith band should be selected, and search was based on antibody cloning and mutation. Particle Swarm Optimization (PSO) has also been applied to the supervised band selection problem in Su et al., 2014. They designed a two-level PSO algorithm that was capable of both selecting an optimized band combination and the number of selected bands. Since it is supervised however, it either needs labelled examples or reference spectral signatures. The algorithm will then optimize a solution for the given labels or reference spectra, which is not the same goal as for unsupervised band selection.

Paul et al., 2015 use a genetic algorithm combined with spatial clustering for unsupervised band selection. They first use a clustering algorithm in order to cluster pixels in the spatial dimension and find the mean of each cluster for each band, resulting in a matrix. They then apply a genetic algorithm where the representation is the numbers of the bands to be selected, with a fixed length N. As an objective function, they use the sum of the Kullback-Leibler divergences between adjacent bands in the individual. Since the Kullback-Leibler divergence can have high values for noisy bands, this might be a downside. The genetic algorithm is based around a crossover operator that merges two random parents and selects N random bands from either. It has no mutation operators, which means that no new bands can ever be added. This means that the genetic algorithm might be sensitive to the random initialization, as that decides what bands "can" be chosen.

In 2016 it was proposed for the first time to use multi-objective optimization for band selection, in Gong, M. Zhang, and Yuan, 2016. It is difficult to determine the number of bands that is needed in the subset selected, and for an unsupervised algorithm, it might not even make sense, since it should be deferred to the actual use of the data. For one dataset there might be several different applications, according to how many classes it should differentiate between, or how similar these classes are. Therefore, Gong, M. Zhang, and Yuan, 2016 proposed a multi-objective algorithm they called MOBS that resulted in a set of optimal solutions, each with a different number of selected bands. They used a Tchebycheff decomposition based approach, so each individual in the population is evaluated based on a given weighed sum between the objectives. For objective functions, they used the number of bands in the solution and the sum of the entropies in the selected bands. The achieved trade-off is thus between preserving the information among the bands and reducing the number of bands selected. They do not include an objective function for ensuring the redundancy between bands, but do include it when doing crossover. Their crossover operator works in such a way that two children are created, and the solution with the highest Kullback-Leibler divergence between them is selected. This will only guide the search to a small degree, as solutions are still selected and discarded based on their objective functions. The algorithm also only includes a crossover operator, making it potentially sensitive to initialization conditions.

Another multi-objective band selection algorithm was reported in Xu, Shi, and Pan, 2017, termed IRMoBS. This algorithm is also based on Tchebycheff decomposition, but still differs somewhat from the goal of MOBS. IRMoBS does not focus on outputting a set of solutions with different number of bands for decision makers, but constructs a variation of the decomposition approach that aims at converging to one specific solution. Therefore, they also require an user input on how many bands that should be selected. Therefore, in IRMoBS, multi-objective optimization is only a tool to reach one specific solution, rather than a technique to generate many.

The authors of Gong, M. Zhang, and Yuan, 2016 proposed a new multi-objective algorithm in M. Zhang, Gong, and Chan, 2018. Instead of trading off between number of bands and information content, this algorithm, termed BOMBS, trades off between redundancy and information content. They argue that different datasets and different applications have different requirements between the two objectives. So BOMBS again, similar to IRMoBS, leaves the concept of selecting band subsets with different lengths simultaneously, and encodes the subset length as an input parameter instead. BOMBS uses a non-domination based AIS algorithm for optimization. In order to select a solution from the Pareto front, they use a trial-and-error-approach.

Lastly, an evolutionary algorithm was proposed in Xie et al., 2019. Their algorithm, ISD-ABC, is based on a two step-process, where the spectrum is first decomposed into a number of subspaces using a modified version of the ASD algorithm (Ye Zhang et al., 1999). Then, equal number of bands are selected from each subspace using the Artificial Bee Colony (ABC) algorithm, with the entropy sum of the selected bands as the objective function. Therefore they achieve a use of both the redundancy reduction, through the subspace decomposition, and maximizing information content through the ABC algorithm. Neither the subspace decomposition algorithm nor the ABC is based on multi-objective optimization however, which might lead to less optimal trade-offs between the two.

3.3 Selected related work

For this thesis, two of the algorithms from in the literature will be studied more in-depth and used as a comparison for the herein proposed algorithm. Since the proposed algorithm is based on a multi-objective algorithm capable of selecting band subsets of different sizes simultaneously, the MOBS algorithm will be used as the first comparison. The second algorithm that will be used is the ISD-ABC algorithm, since it is also based on evolutionary optimization, and makes use of subspace decomposition. The next two sections will describe these algorithms.

3.3.1 MOBS

The MOBS algorithm (Gong, M. Zhang, and Yuan, 2016) is built around a Tchebycheff decomposition multi-objective genetic algorithm and two competing objectives in order to solve the band selection problem. The algorithm is capable of finding a set of optimal bands subsets with different lengths in a single run.

The follow sections will describe how the MOBS algorithm works.

Representation

The MOBS algorithm uses a direct representation where a chromosome is the set S of bands that should be selected. Each value x in S is then an integer value ranging from 0 to $N-1$ where N is the number of bands in the hyperspectral image. If S includes a value 42 that means that the 42nd band of the image is selected.

Objective functions

Since MOBS is a multi-objective algorithm, it uses two competing objective functions to guide its search. They are the total entropy of the selected bands

$$f_1 = \sum_{x \in S} \frac{1}{H(x)}$$

and the number of selected bands

$$f_2 = ||S||$$

Subject to minimization, f_1 seeks to ensure that information rich bands are retained, while f_2 limits the number of bands. Since H is a non-negative function, it can be seen that using only f_1 would select all bands, while only using f_2 would select no bands at all. This means that the Pareto front between them should spread out to include many different number of bands between 0 and N.

Search algorithm

The multi-objective algorithm used in MOBS is based on the genetic algorithm using Tchebycheff decomposition to decompose the multi-objective problem into a set of single-objective problems. See chapter 2.2 for a description of the Tchebycheff decomposition method for solving MOO.

Crossover MOBS generates new solutions by the application of a crossover algorithm between two solutions. The crossover is based around creating difference sets between the bands selected by the two parents, and taking bands from each difference sets to generate two new solutions. Then, in order to choose which of the two solutions should replace the old in the population, the KL divergence between every pair of bands in the solution is calculated, and the solution with the maximum value is chosen. This is done to preserve the non-redundancy of information between the bands.

Discussion

The MOBS algorithm introduces a novel technique for selecting band subsets of several different lengths in one run of the algorithm. It is based on a trade-off between the number of selected bands and the entropy sum, and uses a decomposition approach to ensure that solutions are evenly distributed on the Pareto front. It uses a divergence measure in its crossover selection where two children are created and the solution with the largest divergence between every pair of selected bands is chosen. Still, the central objective functions are only centered around choosing the maximal informative bands for a certain subset length, and gives little guarantee that the information contained within the selected bands is non-redundant. Additionally, since the genetic algorithm is only based on a crossover operation selecting bands from the two selected parents, there is no random mutation able to add bands not present in the initial population. This may potentially make the algorithm sensitive to the initialization conditions, which in MOBS is done randomly.

3.3.2 ISD-ABC

ISD-ABC is an unsupervised band selection algorithm proposed in 2019 by Xie et al. It is based on the combination of two techniques. First, the spectral bands of the hyperspectral image are decomposed into a number of subspaces, and then the Artificial Bee Colony algorithm is used to select a number of bands from each subspace.

ISD

The first phase of the algorithm, Improved subspace decomposition (ISD) is based on the Adaptive subspace decomposition (ASD) method, proposed in Ye Zhang et al., 1999. The ASD method was used as a decomposition step in order to improve the performance of PCA for band extraction, by making sure spectrally local features were retained. To do this, it splits the spectrum into subspaces by using local minima in the correlation coefficient matrix of the bands. Two bands are defined to be in the same subspace if

the correlation coefficient between them is more than a defined threshold value. The ASD method was criticized by Xie et al. for using only the correlation coefficient. They improved upon the ASD method by using the shape of the reflectance spectrum. To illustrate this improvement they use the Indian Pines dataset which was introduced in chapter 2.3. The correlation coefficients between adjacent bands and irradiance spectrum of this dataset is shown in Figure 3.2. They argue that even though the correlation coefficient technique would split up the bands between 75 and 103 into many small subspaces, it should actually be only one subspace, considering the shape of the irradiance spectrum. As far as can be understood from the paper, the irradiance spectrum was used to manually merge the subspaces. This means that the ISD method requires a manual processing step for every dataset the algorithm should be applied to.

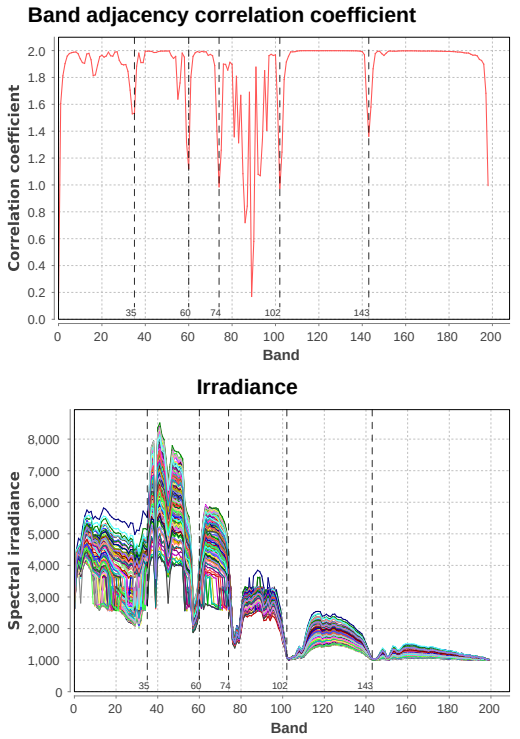

Figure 3.2: Correlation coefficients and irradiance spectrum of the Indian Pines dataset, with ISD-calculated subspace boundaries.

ABC

The second phase of the algorithm involves running the ABC algorithm to select k bands to be used as representatives for each subspace. For the ABC algorithm, the authors use a representation where each honey source is k selected bands from each subset. The maximum entropy

$$f = \sum_{x \in S} H(x)$$

is used as the objective function f for the selected bands.

Discussion

Since this is only single-objective optimization, the algorithm needs to run several times with different values of k in order to produce different numbers of selected bands. Since k is also the same for each subspace, the number of bands selected can only be multiples of the number of subspaces produced by the ISD method. The results show that the number of subspaces can vary greatly from dataset to dataset. On the Pavia University dataset, the number of subspaces is two, in comparison to Indian Pines where there are six subspaces.

When the ISD method has found the subspaces, the ABC method disregards any notion of redundancy or correlation between the bands selected. The only measure is the total entropy of the selected bands. This effectively means that the algorithm establishes that there are no more subspace structures in the dataset than those found by the ISD method, and the best bands within each subspace is simply the ones with the highest information content. Using the sum of the entropies is similar to the MOBS algorithm.

3.4 Summary

There have been developed a wide range of algo The use of multi-objective evolutionary algorithms for unsupervised band selection is a relatively new idea. The MOBS algorithm was the first proposed such algorithm, and had as a goal to be able to co-evolve solution with different band subset sizes in one run. Later multi-objective algorithms have not continued on this track, rather focusing on multi-objective optimization combined with a decision making strategy in order to only select one solution, for a specific number of bands. The main drawback of MOBS is that it does not focus very much on the redundancy between the selected bands, and so risks selecting highly informative

but redundant bands that do not provide any additionally useful information for classification or other tasks. Therefore it would be of interest to look into how one can use the ideas from band clustering and subspace decomposition algorithms in order to take advantage of the redundancy between adjacent bands and select a better set of bands for each subset size. Additionally, since non-dominated evolutionary algorithms are also capable of creating diverse Pareto fronts, it is interesting to look into applying the NSGA-II algorithm to the problem, to alleviate some of the problems of the decomposition algorithm in MOBS like the sensitivity to the initialization. An algorithm proposed in this thesis to look into this research avenue, called DUMB, is introduced in the next chapter.

Chapter 4

Model and Implementation

This chapter will give an overview of the architecture of the proposed algorithm. It will describe the properties of the genetic algorithm, including the novel representation, objective functions used, crossover and mutation operators, and several other considerations.

4.1 Model: DUMB

The algorithm proposed for doing unsupervised band selection of hyperspectral data, DUMB (Decomposition-based Unsupervised Multi-objective Band selection), is primarily based around its novel indirect representation used for search. Its other components are the multi-objective search framework based on NSGA-II and the objective functions used to determine fitness. This section will describe each of these components and how they interact.

4.1.1 Representation

One of the novel properties of the DUMB algorithm is the way the problem is represented. Contrary to the representation in MOBS and ISD-ABC (See Sections 3.3.1 3.3.2) DUMB uses a unique cluster based representation.

The representation is based on the subspace decomposition technique, where the idea is that the spectrum consists of adjacent clusters of bands that contain similar information. This is similar to the ISD approach. However, while ISD determines exactly how many subspaces the spectrum of a specific dataset contains and sets up to select-

ing several bands from each subspace, DUMB will only select one representative band from each subspace, and rather adjust the number of subspaces to choose the number of bands. More concretely, a candidate solution for a band selection problem with N bands in the DUMB representation can be expressed as

$$S = \{b_0, ...b_M\}$$

where:

- $x_i \in [1..N-2]$
- $M \in [0..N-2]$

This means that each b_i represents a boundary between subspaces. For example, having a band selection problem with $N = 200$, Figure 4.1 illustrates the solution $S = \{21, 79, 130, 180\}$. This solution produces five subspaces. The first and the last subspaces implicitly start and end at the first and last band, so there is no need to include them in the representation.

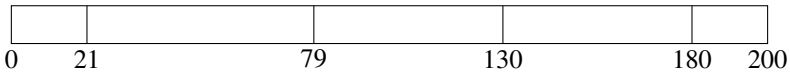

Figure 4.1: Example solution using the DUMB representation

This way, it can be seen that S can also be expressed as a set set of subspaces

$$S_{sub} = \{T_0, ...T_{N-1}\}$$

where T_i is a set of adjacent bands, disjoint from every other T_j.

In order to convert this genotypic representation of the problem into the selected bands, there should be picked one representative band from each subspace. So,

$$B = \{\underset{x \in T}{\operatorname{argmax}} f(x) : T \in S_{sub}\}$$

where $f(x)$ is a function describing the "representativeness" of band x. DUMB will use the information entropy $H(x)$, as described in chapter 2.3.

This representation has a variable length and the same properties as a set, since it has no concept of ordering and can only include unique elements. The lowest cardinality possible is the empty set in which case all bands are in the same subspace, while the highest is equal to the number of bands in the image minus two, when all bands have their own subspace.

4.1.2 Objective functions

Along with this novel, indirect representation, there must also be devised new objective functions to capture the fitness of a given solution. An important consideration when choosing objective functions is having them trade off in the length of the solution. This is, after all, the purpose of viewing the problem as a multi-objective one. Decision makers are presented with a set of optimal solutions of different lengths to choose from. And while earlier algorithms like MOBS explicitly encoded this trade-off, this algorithm will do it indirectly through its objectives.

The problem is defined for simplicity as a minimization problem, which means that lower values of a given objective represents a more optimal solution. Since the algorithm should be unsupervised, the objective functions should only be based on inherent information present in the images. See chapter 2.3.

Objective function 1: Internal divergence (f_{INT})

$$f_{INT}(S) = \sum_{T \in S_{sub}} \sum_{i=0}^{|T|-1} divergence(x_i, x_{i+1})|T| \tag{4.1}$$

The purpose of this objective function is to minimize the internal divergence within each subspace of the solution. This will make sure that the bands in each subspace contains the same information, so that one band can be chosen to represent the information within the entire subspace with minimal loss. Studying this objective function it can be seen that alone it would tend towards putting every band into its own subspace, in which case the value would be zero. Conversely, the maximum value is reached when all the bands are in one subspace.

The divergence value of each subspace is multiplied by its length. This is done to penalize really large subspaces, and guide the search towards creating subspaces that are more evenly spread out across the spectrum.

Objective function 2: Boundary divergence (f_B)

$$f_B(S) = \sum_{i=0}^{|S|} \frac{1}{divergence(x_i, x_i + 1)} \tag{4.2}$$

The purpose of this objective function is to maximize the divergence between the subsets, so they all contain different kinds of information. It does this by summing up the divergence between all the subspace borders. Looking closer into the properties of this objective function, it can be seen that it tends toward the opposite as f_{INT}. Here,

when there is only one subspace, there would be no borders, so the value would be zero, while the maximum value is reached when every band has its own subspace.

Divergence measure Both the objective function use a *divergence* function that takes two bands. There are several possible functions that can be used here, based on information theoretic measures:

- Kullback-Leibler divergence $D_{KL}(X,Y)$
- Mutual information $\frac{1}{MI(X,Y)}$
- Correlation coefficient $\frac{1}{Corr(X,Y)}$
- Disjoint information $DI(X,Y)$

In the experiments section there will be a discussion of these measures of divergence and how they affect the subspace search.

4.1.3 Search algorithm

The search algorithm in DUMB is based on NSGA-II, a non-domination sorting based multi-objective genetic algorithm. See Chapter 2.2. NSGA-II is chosen as the search algorithm since it is a well established algorithm in the literature showing good performance and generalizability. No big alterations have been made to NSGA-II as a framework, so the main components of the search algorithm that is unique for the application is the genetic operators designed to handle the subspace representation. The values for the parameters of the genetic algorithm such as the crossover rate, mutation rate and population size have been established through trial-and-error using sensible default values.

Genetic operators

Given the properties of the representation there need to be designed genetic operators for crossing and mutating solutions. The operators designed are relatively simple, based on one-point crossover and random mutations.

Crossover The crossover operator used in the proposed algorithm is a adapted version of the one-point crossover. Given two parents S_1 and S_2, a random boundary in S_1 is selected. The new solution C is constructed by taking the left side of the boundary from

S_1 and the right side from S_2. Using this crossover aims to retain good subspace decompositions from parts of the spectrum, and combining them with others. An example is illustrated below.

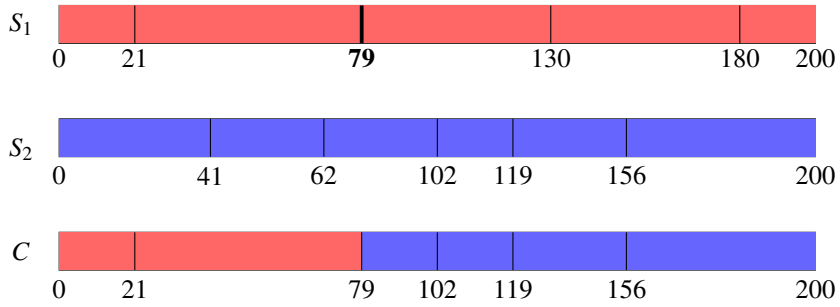

Figure 4.2: **Example crossover operation**. The crossover point chosen from the first parent is band 79.

The example crossover operation in Figure 4.2 will in addition to combining the subspaces of the parent solution, create a new subspace between bands 79 and 102 resulting in a new selected band.

Mutation While crossover can combine parts of different solutions into new and better ones, the algorithm also needs a way to exploit good solutions with local search. This is done through mutation. The mutation operator chosen for the algorithm will either add a boundary at a random band in the solution, splitting a subspace in two, or remove an existing boundary to merge two subspaces. It chooses between these two operations with equal probability.

Selection Individuals are selected for mutation and crossover by using tournament selection, as described in 2.2. The winner of the tournament is chosen by sorting the individuals by the non-dominated sort algorithm.

Termination

The search is terminated either if a predefined max number of iterations have been reached, or if no new solutions have been introduced into the non dominated front after 100 iterations.

4.2 Implementation

The proposed model is implemented in a simulator using the Java programming language. The implementation is architected in such a way that it is simple to swap out objective functions, representations, multi-objective frameworks and datasets. This way the program lays the groundwork for doing thorough comparisons of how different components and aspects of the band selection algorithm work together to produce results, and enables the comparison to previously proposed band selection methods. Visualizations are produced for both information theoretic quantities in the dataset, Pareto fronts and classification results, using the JFreeChart library (cite?).

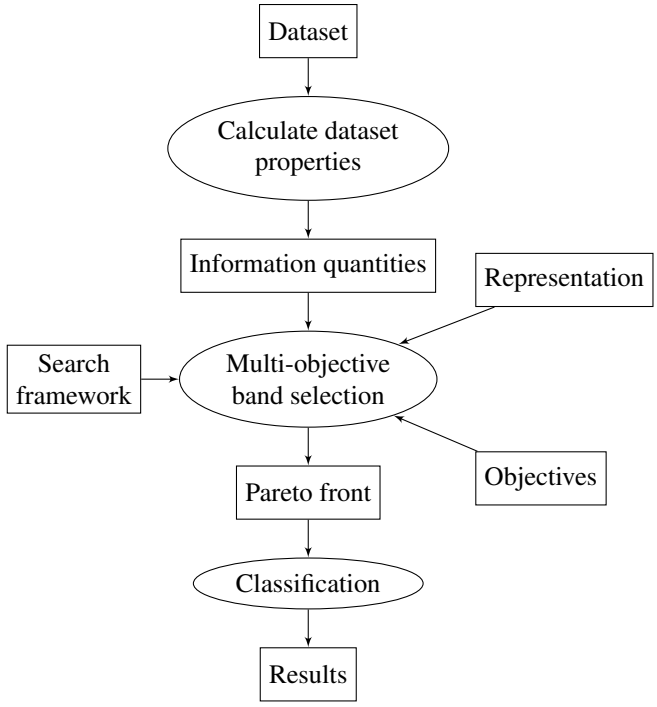

Figure 4.3: Overview of implementation architecture. Elliptical nodes are calculation steps, while the rectangular nodes are input/output data.

4.2.1 Representations

There are two implemented problem representations. The first is the direct representation, as used in the MOBS model (Gong, M. Zhang, and Yuan, 2016), called band representation. The second one is the newly proposed indirect representation based on subspaces, called the subspace representation, introduced in the previous section.

4.2.2 Objective functions

The objective functions that are implemented are the ones used in the algorithms that are compared between. They are listed below, together with the algorithms that employ them.

- Boundary divergence (DUMB)
- Internal divergence (DUMB)
- Number of bands (MOBS)
- Total entropy (MOBS, ISD-ABC)

4.2.3 Search frameworks

Several search frameworks are implemented in order for a comparison to be made between them.

NSGA-II The NSGA-II algorithm is implemented based on the original paper (Deb et al., 2002), and has several configurable parameters, which are

- Maximum number of iterations
- Population size
- Tournament size
- Crossover rate
- Mutation rate

Tchebycheff decomposition The decomposition based multi-objective framework is implemented as described in the MOBS algorithm (Gong, M. Zhang, and Yuan, 2016), explained in section 3.3.1. The configurable parameters are

- Maximum number of iterations
- Population size (Number of subproblems)
- Neighborhood size

Artificial Bee Colony Since ABC is a single objective search algorithm, it requires the specification of a k, how many bands to select. ABC is used as a part of the ISD-ABC algorithm, together with ISD. The configurable parameters are

- Maximum number of iterations
- Population size
- Number of bands to select
- Stale food source limit

ISD based search The ISD approach proposed by (Xie et al., 2019) decomposes the spectrum into a number of subspaces. The ISD based search method can then apply the other search frameworks on each individual subspace and then combine them to achieve one solution. When merging the solutions from the subspaces, the algorithm will combine them so that there are the same number of bands selected in each subspace.

4.2.4 Pruning of selected solutions

Due to the indirect nature of the representation, there is no guarantee that the algorithm will produce exactly one solution per number of bands. Considering there is no meaningful way for a decision maker to choose between two solutions with the same number of bands, the algorithm should prune the solutions so there is only one per number of selected bands. Since the algorithm should be unsupervised, the achieved classification accuracy can not be used. Since all of the solutions with a given number of bands are found using the same algorithm, they should both include a diverse set of bands. Given this assumption, a good way to choose between the solutions is to pick the one with the maximum total entropy between all the selected bands, maximizing the information content.

Chapter 5

Experiments and Results

This chapter contains the experimental work conducted to evaluate the performance of the DUMB algorithm. Firstly, it reviews the research goal introduced in the introduction of the thesis and constructs a series of research questions that is used to guide what experiments should be conducted. Secondly, it introduces the experimental plan. It includes a description of the datasets that will be used, the classification algorithm and the metrics used for evaluating classification performance. Lastly, the results are presented and briefly discussed. A more through discussion of the experimental results and their relation to the research questions can be found in 6.

5.1 Research goal

The research goal introduced in Chapter 1 is as follows:

> How can low redundancy band clustering be used together with multi-objective evolutionary search in order to select representative bands from a hyperspectral image in an unsupervised manner?

For this research goal, an algorithm called DUMB has been proposed as described in Chapter 4. In order to evaluate how the algorithm can answer the research goal, several research question have been formulated. These questions are also listed in Section 1.1. The research goal states that the goal is to select "representative" bands. The experiments will judge the representativeness on how well it performs on a supervised classification task. This is not mentioned in the research goal since the algorithm is unsupervised and the representativeness is unrelated to the task the selected bands should be applied to.

For running experiments however, classification is used as the task.

5.2 Experiment Setup

5.2.1 Datasets

Name	Sensor	Bands	Samples	Labelled samples	# classes
Indian Pines	AVIRIS	220 (0.4 μm - 2.4 μm)	$145 \cdot 145$	10249 (49 %)	16
Botswana	HYPERION	145 (0.4 μm - 2.5 μm)	$256 \cdot 1476$	3248 (0.9 %)	14
Pavia University	ROSIS	103 (0.43 μm - 0.96 μm)	$340 \cdot 610$	42776 (20 %)	9

Table 5.1: Three hyperspectral datasets used for experiments

Experiments will be conducted on three popular hyperspectral datasets, all of them available online at `http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes`. They are recorded by three different sensors at three different locations, to explore the generalizability of the algorithm for various spatial, spectral and geographical conditions. The three datasets include two artifacts; the hyperspectral image itself, and the ground truth. The hyperspectral image is a three dimensional matrix consisting of the irradiance values received at the sensor focus point. Each pixel of the image is considered to be a vector of the irradiance values at a specific spatial coordinate across the available band. This vector does *not* form a reflectance signature like the ones shown in Figure 2.5. The irradiance signature is affected by the solar intensity at each band (Figure 2.2) and the atmospheric interaction (Figure 2.3), in addition to the reflectance of the surface. However, since the datasets are collected in local time and space, it is reasonable to assume that both the atmospheric interaction and solar intensity is constant across all pixels, so the only meaningful difference between the pixels is due to the surface reflectance. Notice also that the numeric values of the spectral irradiance have no meaningful unit.

The ground truth of the image is expert labelled by mapping the pixels to discrete labels. The ground truth includes class labels on a subset of the pixels in the image. Following is a description of each of the three datasets and their unique features.

In addition to the hyperspectral data itself, a set of artifacts is produced that is used by the algorithms. These are

- **Entropy** The entropy value of each band
- **Correlation coefficients** The correlation coefficient between adjacent bands

- **KL-divergence** The Kullback-Leibler divergence between adjacent bands
- **MI** The mutual information between adjacent bands
- **DI** The disjoint information between adjacent bands

and are graphically displayed in Appendix A.

Indian Pines

The Indian Pines dataset was captured by the AVIRIS NASA, 2019b sensor in an area in Indiana, USA. The image contains different agricultural crops, forests, and some human infrastructure. The image itself consists of 145 by 145 pixels, but the available ground truth mapping only labels 10249 (49 %) of the pixels. Ground truth and class sample numbers are shown in Figure 5.1, and spectral irradiance in Figure 5.2.

The Indian Pines dataset is likely the most widely used test dataset for classification of hyperspectral data, used all over the literature. It has some interesting challenges for classification. The number of labelled examples are relatively low, and the number of samples vary greatly from class to class. The class "Oats" for example, only has 20 samples, which means that with a training set size of 10 % there are only 2 instances available for training. Since the different classes have different spectral signatures, the number of bands needed in order to classify them all correctly could be high.

The dataset is also interesting because it is available in two variations. One variation is with all the 220 recorded bands from the sensor. However, some of the bands in the spectral range that AVIRIS records in are known water absorption bands (See figure 2.3). These bands are at indexes [103-107], [149-162] and 200. This enables experimenting on how well the band selection algorithms can handle the presence of noisy bands, by comparing the achieved results with and without the noisy bands removed. Experiments done on the Indian Pines dataset will be marked "Indian Pines (All)" if it is done on the dataset without manual band removal.

Botswana

The Botswana dataset was captured by the Hyperion sensor (Pearlman et al., 2003) aboard the NASA EO-1 satellite. The data depicts a location on the Okavango Delta in Botswana. It is a much larger image than Indian Pines, with a size of 1476 by 256 pixels. The Hyperion sensor records data in 220 spectral bands, but the dataset only has 145 of the bands, which is without noisy and uncalibrated bands. The dataset with all bands is not available, as it is with Indian Pines. Class sample numbers and ground truth is shown in Figure 5.3, and irradiance values in Figure 5.4.

Class	Samples
Alfalfa	46
Corn-notill	1428
Corn-mintill	830
Corn	237
Grass-pasture	483
Grass-trees	730
Grass-pasture-mowed	28
Hay-windrowed	478
Oats	20
Soybean-notill	972
Soybean-mintill	2455
Soybean-clean	593
Wheat	205
Woods	1265
Building-Grass-Trees-Drives	386
Stone-Steel-Towers	93

Figure 5.1: Indian Pines dataset

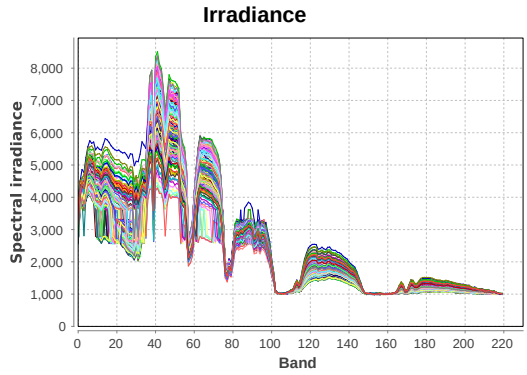

Figure 5.2: Irradiance values of 200 random pixels in the Indian Pines (All) dataset

Pavia University

The Pavia University image was captured by the ROSIS sensor (Kunkel et al., 1988) and depicts part of the Pavia University campus in Northern Italy. Classes and their

Class	Samples
Water	270
Hippo grass	101
Floodplain grasses1	251
Floodplain grasses2	215
Reeds1	269
Riparian	269
Firescar2	259
Island interior	203
Acacia woodlands	314
Acacia shrublands	248
Acacia grasslands	305
Short mopane	181
Mixed mopane	268
Exposed soils	95

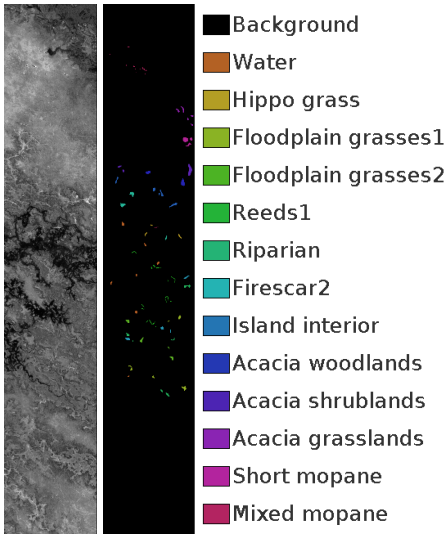

Figure 5.3: Botswana dataset

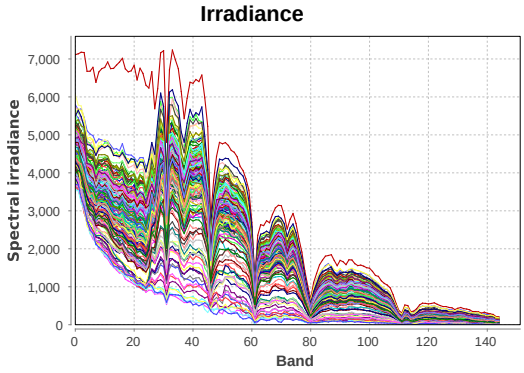

Figure 5.4: Spectral irradiance values of 200 random pixels from the Botswana dataset

quantities are shown in Figure 5.5, and irradiance values in 5.6.

The ROSIS sensor is different from AVIRIS and HYPERION in the spectral dimension. While the two former records data in the entire visible spectrum and some of the infrared, ROSIS only uses 430 nm to 960, which is the visible spectrum and slightly into IR, as shown in Figure 2.1. Since it uses a different spectral range, the shapes of the information theoretic quantities (Figure A.3 in) are also significantly different. The

entropy diverges little from band to band, and the irradiance spectrum does not show the distinct shape of water absorption bands like present in both Indian Pines and the Botswana datasets. Therefore, also the band similarity quantities like KL-divergence and mutual information show such different properties from the two other datasets to warrant the investigation of this dataset.

In the spatial dimension, the Pavia University differs from the others in that it has a large number of labelled samples, at least in absolute terms, and that there are fewer classes.

Class	Samples
Asphalt	6631
Meadows	18649
Gravel	2099
Trees	3064
Painted metal sheets	1345
Bare soil	5029
Bitumen	1330
Self-blocking bricks	3682
Shadows	947

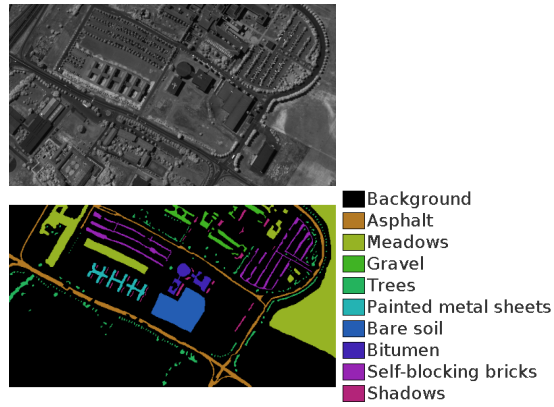

Figure 5.5: Pavia University dataset

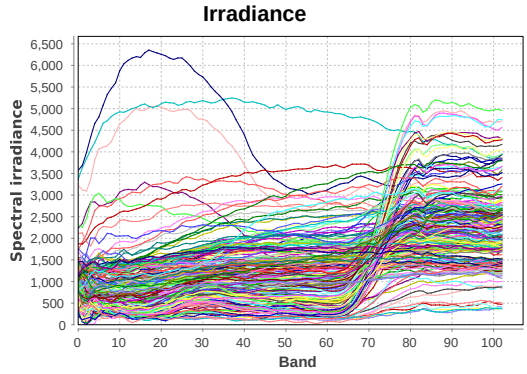

Figure 5.6: Spectral irradiance values of 200 random pixels from the Pavia dataset

5.2.2 Classification

Classifier	Soft-margin SVM (one-against-one)
Kernel	RBF
Hyperparameters	C and γ established through five-fold cross validation on training data
Training set size	10 % of each class
# runs per experiment	10
Max band subset length	50
Evaluation	Overall accuracy (OA), Average accuracy per class (AA)

Table 5.2: Properties of the classifier used for experiments

Classification of the dataset using the selected bands is done in order to evaluate the selected subset. For classification, each pixel in the dataset is considered to be a sample, and the given ground truth is the supervised label of that sample. This means that classification is only done using structures in the spectral dimension. In the spatial dimension, each pixel is considered an independent sample with no relation to each other.

Since the focus is on the band selection algorithm and not on the classifier itself, optimizing the classifier for performance will not be the main focus. SVM has the favorable properties that it works well with small training sets, has an existing high quality implementation (C.-C. Chang and Lin, 2011) and that it is not extremely computationally heavy. Therefore it is chosen as the classification algorithm. The SVM algorithm will be using the radial basis function as the kernel, and the kernel parameters C and γ will be established using grid search and five-fold cross validation of the training set.

For each band set received from the band selection algorithm, the classifier will do 10 evaluation runs, where a random 10 % of each class in the dataset will be picked out as the training set each run. Selecting the same percentage from each class ensures that the training set is weighed equally to the test set with regards to the amount of data in each class. The data will also be normalized per feature to a value between 0 and 1, a recommended preprocessing step for the SVM algorithm that ensures that each of the features have equal weight in the classification. Preliminary tests indicate that when a large number of bands are added, there is little improvement to the classification accuracy. This is likely due to how the SVM algorithm works, since it does not suffer the curse of dimensionality by losing classification accuracy, but it does hit a ceiling where adding more bands does little to increase the accuracy. For this reason, and to preserve computation time, the max number of bands that is classified for each run is set to 50.

The performance of the classification algorithm will be evaluated based on two metrics. First is the overall accuracy (OA). This is simply the number of correctly labelled test samples divided by the number of test samples. Both the mean and standard devi-

ation of this value will be recorded. Second is the average accuracy (AA). This is the average number of correctly labelled test samples of each class. This can reveal information about how the selected bands might be good at classifying some classes but bad at others. In some experiments, the accuracy for each class may also be listed.

5.2.3 Hyperparameters

Several hyperparameters are used in the algorithms implemented for these experiments. For the recent state of the art methods, the hyperparameters were based on recommendations from the original authors, while the parameters to DUMB were established through trial and error.

MOBS

- Number of iterations: 100
- Population size: 200
- Neighborhood size: 30

ISD-ABC

- Number of iterations: 150
- Population size: 30
- Stale honey source limit: 5

DUMB

- Max number of iterations: 1000
- Population size: Same as number of bands in the dataset
- Crossover rate: 0.75
- Mutation rate: 0.05
- Tournament size: 5

5.3 Results

5.3.1 Classification accuracy with all bands

Since the purpose of the experiments is to determine the performance of the band selection algorithm, rather than the classification algorithm, an initial experiment was run that uses all the bands in the hyperspectral image in order to determine a baseline classification accuracy. The results of this experiment is shown in Table 5.3.

Dataset	Overall accuracy	Average accuracy
Indian Pines (All)	0.76 ±0.003	0.63 ±0.02
Indian Pines	0.79 ±0.006	0.70 ±0.01
Botswana	0.90 ±0.01	0.91 ±0.01
Pavia university	0.94 ±0.002	0.91 ±0.005

Table 5.3: Classification accuracy using all bands

Class	Classification accuracy
Alfalfa	0.21 ±0.13
Corn-notill	0.72 ±0.03
Corn-mintill	0.61 ±0.06
Corn	0.53 ±0.07
Grass-pasture	0.85 ±0.04
Grass-trees	0.97 ±0.01
Grass-pasture-mowed	0.74 ±0.15
Hay-windrowed	0.99 ±0.01
Oats	0.16 ±0.12
Soybean-notill	0.70 ±0.03
Soybean-mintill	0.86 ±0.03
Soybean-clean	0.63 ±0.05
Wheat	0.97 ±0.02
Woods	0.96 ±0.02
Building-Grass-Trees-Drives	0.43 ±0.05
Stone-Steel-Towers	0.85 ±0.07

Table 5.4: Classification accuracy for each class using all bands on the Indian Pines dataset.

This results clearly shows that the Indian Pines and Indian Pines (All) datasets are the hardest to classify. It is obvious that Indian Pines (All) is worse, since it includes the water absorption bands, which are just noise. To explore why these datasets are harder to classify, there will follow a more thorough examination of the classification

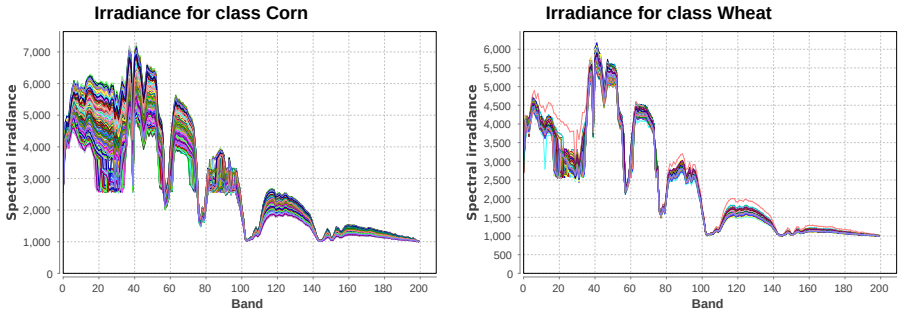

Figure 5.7: Irradiance values for the two classes "Corn" and "Wheat" in the Indian Pines dataset

performances of each individual class, shown in Table 5.4. Firstly, the "Oats" class has a very low accuracy. This is easily explained by the fact that there are only 20 samples of the class, so only two samples available for training. However, the two classes "Corn" and "Wheat" have almost the same amount of samples, but still have a large difference in accuracy. This discrepancy can be explained by looking at the irradiance signatures of all samples of the two classes (Figure 5.7). This shows that the irradiance values in the "Corn" class vary much more than in the "Wheat" class. Naturally the classifier struggles more with it. This result shows that the "Corn" class might actually be several distinct classes. This is a weakness of hand labelling data like done here, and cannot be avoided.

5.3.2 Multi-objective Search Algorithm (RQ1)

The search procedure in DUMB is guided by NSGA-II producing a set of solutions along the Pareto front.

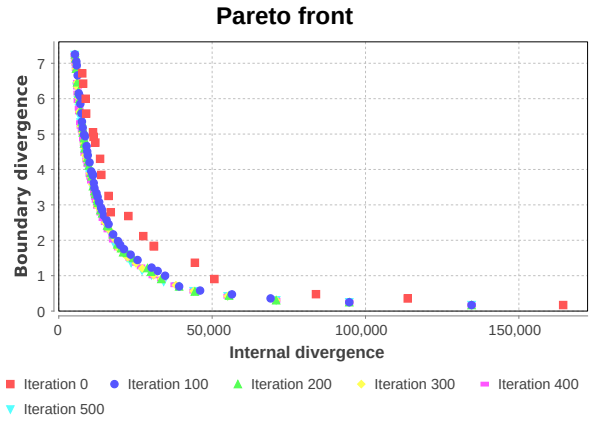

Figure 5.8: Convergence of Pareto front

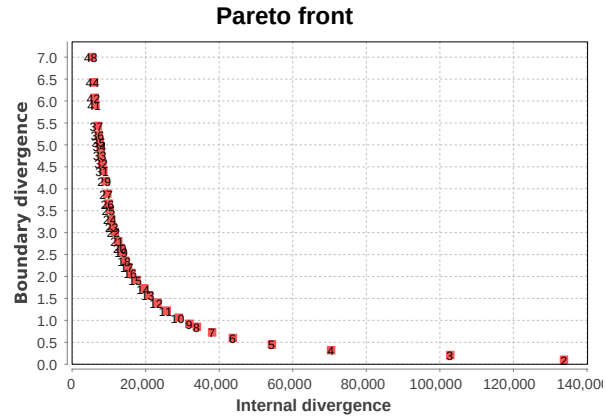

Figure 5.9: Achieved Pareto front of solutions

The results shown in Figures 5.9 and 5.8 show that the non-domination model is capable of creating a diverse set of solutions along the Pareto front. At the beginning of the run, there are fewer solutions with a unique number of bands, since there are more "holes" in the front. As discussed in Chapter 4, DUMB only chooses one solution per number of selected bands. After only some hundred iterations of the algorithm the solutions spread out better, as well as having more optimized values.

Figure 5.9 also show that the trade-off between the two objective functions creates

band subsets with different sizes. The numbers indicated on each point in the front indicates the length of the subset. High values of internal divergence create small subsets, while the other end of the front has large subsets.

Although all the figures only show the subsets selected with fewer bands than 50, interesting results can also be gleamed from looking at the entire Pareto front. Particularly the fact that the front consists of 100 solutions, on a dataset with 200 bands. Below 50 there are 41 unique solutions which means that the density is higher on subsets with fewer bands. This indicates that there is more to gain in terms of the objective functions when there are fewer bands selected.

Tables B.1 and B.2 show that to a large degree, having one more band in the subset equals adding a band to the subset with one less band. This matches intuition since the objective functions are designed around finding local minima/maxima in the spectrum. However, it is not always the case, which might both be a result of premature convergence of the search, or that the objective functions favor towards spreading the subsets evenly across the spectrum has an effect.

5.3.3 Divergence Measure for DUMB (RQ2)

The decomposition into subspaces by DUMB is driven by the measure used to quantify the divergence between a pair of adjacent bands. There are several options for divergence measures from the information theoretic background, as described in section 4.1.2, and these will be compared experimentally to select the most fit one.

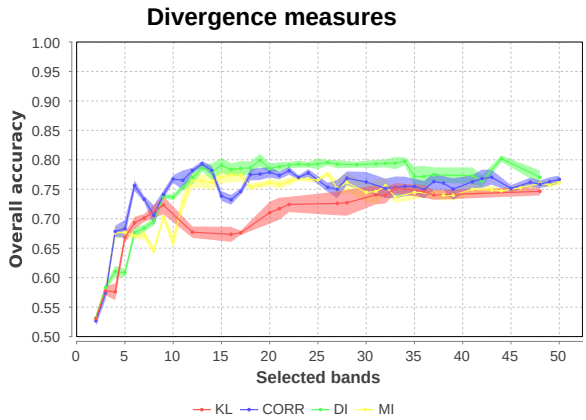

Figure 5.10: A comparison between the difference divergence measures considered for DUMB on the Indian Pines dataset.

Divergence measure	Subset boundaries	Selected bands	Overall accuracy
Correlation coefficient	12, 35, 59, 74, 90, 102, 118, 143, 159, 175, 186	8, 30, 41, 59, 74, 101, 111, 118, 155, 163, 176, 186	0.78±0.005
KL divergence	1, 2, 3, 4, 35, 59, 60, 73, 104, 142, 184	0, 1, 2, 3, 30, 41, 59, 71, 73, 111, 155, 185	0.68±0.01
Mutual information	15, 36, 57, 77, 93, 103, 119, 142, 150, 171, 195	14, 30, 41, 59, 77, 101, 111, 119, 149, 155, 173, 196	0.77±0.01
Disjoint information	12, 30, 36, 57, 78, 101, 111, 143, 156, 172, 188	8, 29, 30, 41, 59, 100, 110, 111, 155, 163, 173, 188	0.77±0.01

Table 5.5: The solutions for the four different divergence measures for 12 selected bands, on Indian Pines dataset.

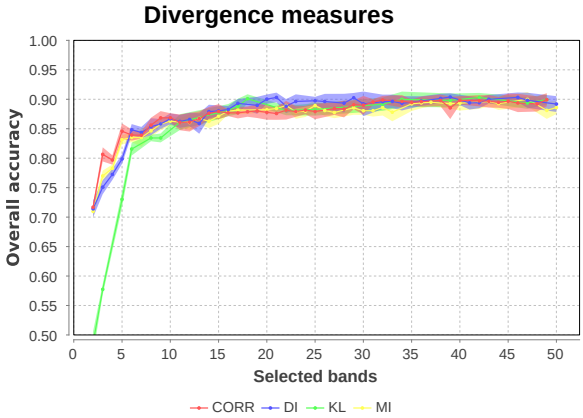

Figure 5.11: A comparison between the difference divergence measures considered for DUMB on the Botswana dataset.

The results of using different divergence measures as objective functions for DUMB are shown in Figure 5.10 for the Indian Pines dataset and in Figure 5.11 for the Botswana dataset. Indian Pines show larger differences and is therefore more interesting to analyse. The figure shows that there is quite large differences in the achieved classification accuracies using the different objective functions. The values of the different divergence measures on adjacent bands for Indian Pines can be viewed in Figure A.1.

The objective function that achieved the worst result was the Kullback-Leibler divergence. It starts promising with only a few bands, but then starts to dip quickly. Looking at the selected bands in detail, shown for a subset size of 12 in Table 5.5, the KL divergence quickly wants to add many of the low-wavelength bands. These are low entropy,

noisy bands, and since the KL-divergence between them is significant, they are heavily favored by the search algorithm. Not only are they favored early, but they form several, one-band subspaces. Since so many of these noisy bands are added early the classification accuracy falters. It rises steadily after these bands have been added, but never quite reaches the best functions.

The other objective functions show more similar performance, but the disjoint information (DI) seems to perform a slight bit better. This is due to its preference for high entropy bands. While both the correlation coefficient and mutual information start to consider the low entropy bands at the ends of the spectrum quite early, disjoint information stays away from it until around 35 bands, where the graph in Figure 5.10 has a small dip in accuracy.

On the Botswana dataset, the differences are much smaller. This might be due to the Botswana dataset being easier to classify and that the differences between the divergence measure values are smaller. See Figure A.2 for the information theoretic properties of the Botswana dataset. Since the dataset has been cleaned up and a good subset of bands that does not include noisy and bad bands has already been done, there is less to gain in selecting the exact best bands. The algorithm fares well with either measure since it is more important that the bands selected are well spread out across the spectrum, which DUMB favors in any case. The removal of noisy bands from the full set of 220 bands down to 140 also means that the bands that are adjacent in the dataset are not actually adjacent wavelengths. This means that the adjacent wavelength redundancy has less meaning.

5.3.4 Using manual band removal (RQ5)

In order to evaluate how DUMB handles noisy image bands, experiments have been run on both the Indian Pines and the Indian Pines (All) datasets. Results of this test is shown in Figure 5.12. This figure shows that while the two datasets rise quite similarly in classification accuracy in the first few subset sizes, the dataset with noisy images starts to falter after 15 bands. After this it lies steadily a few percent below the accuracy of the dataset with the noisy bands removed. This indicates that DUMB creates subspaces in which it is forced to select a noisy band. The main method it has to avoid selecting bands with noise is that it selects representative band through the max entropy criterion, which does not properly work if all the bands in a subspace have low entropy.

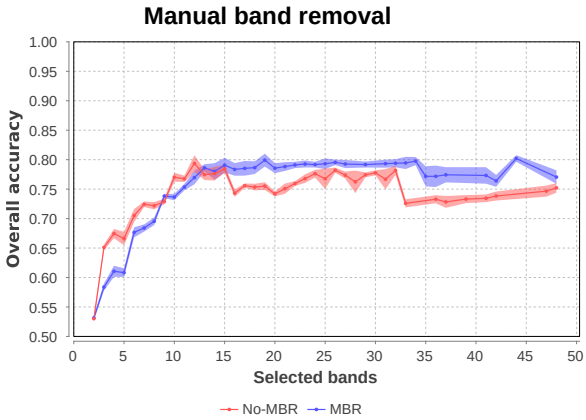

Figure 5.12: DUMB on the Indian Pines and Indian Pines (All) dataset. "No-MBR" indicates that no bands are manually removed

5.3.5 Comparison with other recent methods (RQ3, RQ4, RQ5)

In order to evaluate the performance of DUMB, it was compared with other recent band selection methods, namely MOBS and ISD-ABC. The comparison was run on all three datasets.

Figures 5.13, 5.14, 5.15 show that DUMB improves much faster than ISD-ABC. The difference between the methods is primarily that DUMB will partition the band into more subspaces, while ISD-ABC will select several high entropy bands from the fixed subspaces determined by the ISD method. So the subspace model seems to work well for low number of branches, and DUMB reaches a classification accuracy that compares with using all bands early. However, for all three datasets the improvement more or less stops there. Since SVM does not suffer much from the curse of dimensionality it might be difficult to reach a much higher classification accuracy than for all bands with any subset. This is particularly true for the Botswana and Pavia University datasets, which do not contain many low-entropy, noisy bands. For Indian Pines, on the other hand, there might be something more to gain, since quite a few of the bands are noisy. The DUMB algorithm misses out on these improvements however, as when the number of bands increases past 30, it starts to decompose some of the noisy parts of the spectrum and is forced to choose some of the noisy bands. The classification accuracy falters accordingly.

The algorithms can also be compared on the average entropies of the selected bands, shown in Figures 5.13, 5.14 and 5.15. This should give an indication of how important it

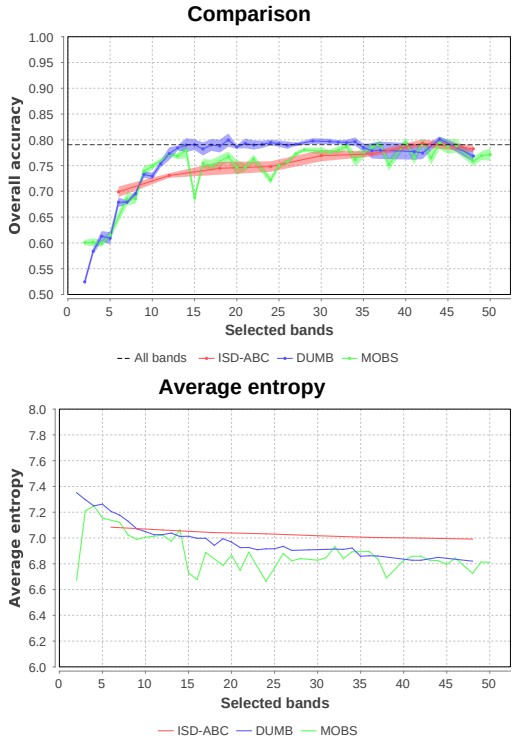

Figure 5.13: A comparison between DUMB and two other recent methods on the Indian Pines dataset.

is that the bands selected has a high entropy. The graphs show some different properties for the different algorithms and the different datasets, and provide a good framework for discussing the different approaches of the three algorithms. MOBS has a main focus on selecting high entropy bands, slightly guided to ensure divergence is retained. The results show that the classification accuracy can vary greatly from one number of bands to the next, unlike DUMB which is more stable as more and more bands are added. The entropy values have the same tendency and surprisingly maybe, the entropy is not much higher than DUMB, particularly on Indian Pines. The fluctuations in accuracy may be attributed to the sensitivity to initial conditions, discussed in 3.3.1. On Botswana, MOBS selects bands with quite a bit higher entropy than DUMB without it leading to any higher classification accuracy, which may indicate that non-redundancy is more important on the Botswana dataset. That may be explained by the fact that Indian Pines contains more noisy bands, and thus avoiding them is as important as ensuring the non-redundancy

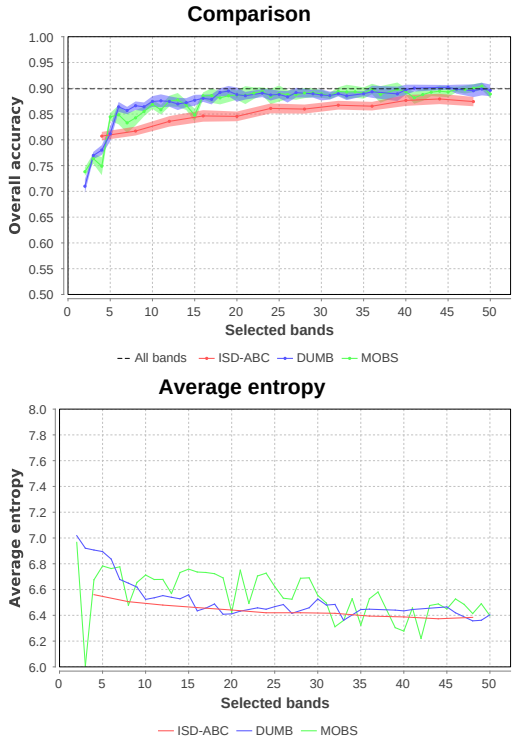

Figure 5.14: A comparison between DUMB and two other recent methods on the Botswana dataset.

between the bands.

On all three datasets, DUMB starts with selecting high entropy bands, but the average entropy drops fast, together with the increase in classification performance. This shows that it is more important to select bands that are evenly spread out in the spectrum early, even though they might have a lower entropy. The ISD-ABC algorithm has an average entropy that decreases steadily, and the classification accuracy shows the same tendency in increasing.

ISD-ABC shows very stable performance on all three datasets. The tendency, however, is that the accuracy grows much slower than both MOBS and DUMB. On all three datasets it starts off equal, when it only selects one band from each subspace. However, while DUMB creates more subspaces and MOBS has mechanisms to ensure more even spread, the ABC algorithm optimizes very well towards its entropy sum objective function, and thus often selects adjacent, high entropy bands within each subspace. This is

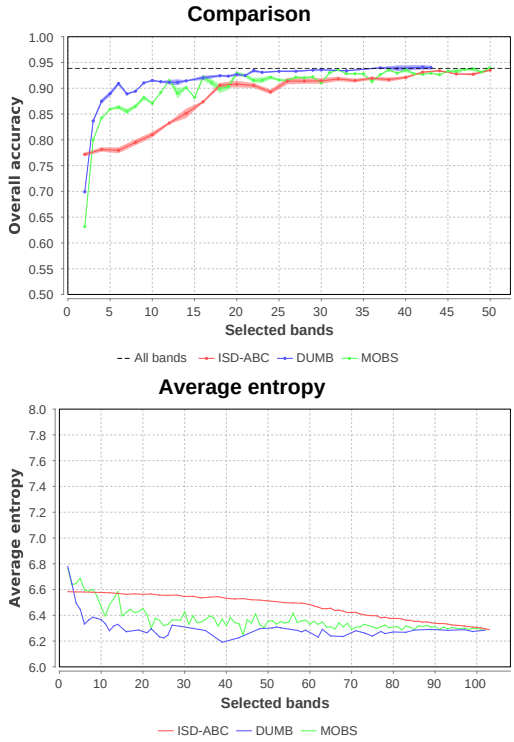

Figure 5.15: A comparison between DUMB and two other recent methods on the Pavia University dataset.

shown in the entropy graph for the Indian Pines dataset, and provides very little additional classification power. It should be noted that the authors of the ISD-ABC algorithm (Xie et al., 2019) did not report the same behavior and so did not discuss the issue. In any case, it does show some of the optimizing ability of ABC.

5.3.6 Achieved subspaces compared to ISD (RQ1, RQ3)

Since ISD separates the spectrum into a specific number of features using the correlation coefficients and the irradiance spectrum, it is interesting to compare the achieved subspaces with DUMB, which uses disjoint information. For the Indian Pines subset, the ISD method produces six subspaces. Figure 5.16 shows the comparison between these six subspaces and the six subspace solution produced by DUMB. The figure shows that four of the boundaries are almost at the same points in the spectrum, which corresponds

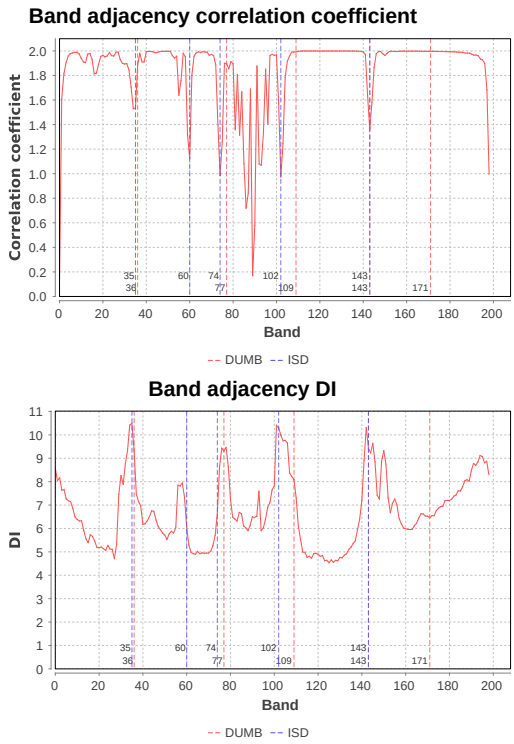

Figure 5.16: Six subspaces generated by ISD and DUMB on the Indian Pines dataset.

to the points where the correlation coefficient and disjoint information share local extremes. It is also worth noting that DUMB does not require the use of the irradiance spectrum in order to determine that there is only one subspace between band 77 and 109, which seems to indicate that the disjoint information is a more useful measure than the correlation coefficients. ISD and DUMB do not agree on where to put the fifth subspace, however. ISD places it at band 60, which is indeed also a local maximum in the disjoint information, while DUMB chooses to put one at 171. This does not make sense for the correlation coefficient, which is very flat in that area. Its not entirely obvious why its placed there based on the DI either, but it can explained by the fact that the f_{INT} objective function leans towards favouring solutions which splits the spectrum into more equally sized subspaces. However, one does not need to go further than the solution with one more subspace for DUMB to place a boundary at band 58.

Chapter 6

Discussion and Conclusion

This chapter will provide a discussion of the results yielded by the work in this thesis. It will start with a discussion of the experimental results in relation to the research questions, and how the proposed model stands up to other recent similar models. Then there will be a discussion on the limitations of the model, in terms of its applicability and generalizability. Then there will be a wrap-up of the most important contributions of the work, as well as a discussion of how it can be continued and improved in future projects.

6.1 Discussion of Results

The research goal and research questions that were introduced in Chapter 1 have served as a guide for the experiments conducted and described in Chapter 5. The following section discusses how the experiments provide insight into each of the research questions and lastly the overall research goal.

RQ1 - How good is NSGA-II at selecting band subsets of different lengths that provide stable and predictable results? NSGA-II was chosen as the multi-objective framework of choice due to it being a popular and widely implemented algorithm. The results show that it is capable of generating a diverse set of solutions across the trade-off between the two objective function, and that this trade-off corresponds to different subset lengths. The results are predictable in the sense that the bands included in a subset of length k one extra band compared to the subset of length $k-1$.

RQ2 - What divergence measure best captures the different information captured in adjacent spectral bands? The experiments conducted to answer this question show that it has a dependence on the dataset, particularly in relation to the noise conditions. It highlights a problem with the model and relying so heavily on measures of divergence between adjacent bands. Since noisy bands will obviously show some differences between them no matter the measure used, the algorithm will at some point be forced to select the bands. Using disjoint information does alleviate the problem somewhat, since it will favour high entropy bands first, but this only means that the low entropy bands will be chosen as boundaries at a higher number of subspaces.

RQ3 - How do the number of subspaces affect classification accuracy? The main finding the experiments show for this question is that the subspace model of DUMB is good at quickly reaching a high classification accuracy but that the model starts to break down or lose its meaning at some point. This may be because there exists no more meaningful local features in the adjacency divergence graph, or because it chooses subsets including only noisy bands. The exact position it starts to lose its usefulness depends greatly on the properties of the dataset.

RQ4 - What is the relative importance of individual band informativeness compared to the redundancy between selected bands? Experimental results show that this question also has different answers depending on the dataset. This question is also the most interesting to discuss when there are only a few bands selected since that is where there is a significant difference in the methods considered, both in terms of classification performance and average entropy. Generally, the results show that with few bands, higher average entropy does not necessarily equal higher classification accuracy.

RQ5 - How does the presence of noisy bands affect classification accuracy? The presence of noisy bands may affect the DUMB method more than the other models, according to the results, which can be explained by the fact that the other methods puts a higher emphasis on entropy. DUMB will, at some point, start so select noisy bands since the measures it uses to select non-redundant bands also indicate noise.

Research goal The goal of the research undertaken in this thesis was to investigate if clustering of bands in hyperspectral images could be used together with evolutionary techniques in order to select representative bands. The outset of the research was always to employ multi-objective evolutionary algorithms, both since the author finds it an interesting topic, and since early literature review indicated promising results in the field. The results of the experiments show that using a representation and objective functions that are based around the divergence between adjacent bands in the image is capable of

achieving similar and in some cases better results than other evolutionary techniques that has been applied. It is of particular importance that it is able to select good and stable bands that maximizes the potential of the data at a low number of bands, where each individual band is significant. The downside of the clustering technique is that it relies most heavily on the divergence measure, a quantity that will inevitably be high for noisy bands, so that noisy bands present in the image are often selected, degrading the performance. Although several efforts were made in order to limit this effect, such as using the disjoint information quantity and a subspace length coefficient in the objective functions, these only mitigate the problem rather than eliminating it. The algorithm showed good results, apart from the noise issues, on several datasets with different spectral and spatial properties showing robustness and generalizability.

6.2 Contributions

The main contribution of this thesis is the algorithm proposed for band selection in hyperspectral images. It continues a research avenue of the application of evolutionary algorithms for band selection, by showing that it can be effectively combined with the clustering idea, another part of the band selection field.

There has also been implemented a configurable and flexible band selection framework that can be easily extended to explore other objective functions, search frameworks and representations. The code is currently not open, but could be open sourced in the future.

This thesis has also been the first project at NTNU in the application of evolutionary techniques for hyperspectral imaging, and has as such laid the groundwork in the form of establishing theoretical knowledge, state-of-the-art, and provided a starting point for further research.

6.3 Further Work

Multi-objective optimization within the field of band selection of hyperspectral images is a young research path. This thesis explored using a non-domination sort genetic algorithm, in favor of one using decomposition, but there are many other search algorithms available. The Artificial Bee Colony showed good convergence rates for a single-objective version of the problem, so research into a multi-objective adaptation would be interesting.

One of the pain points of the clustering representation chosen for the algorithm was that it inevitably chose noisy or otherwise undesireable bands. Further work could in-

vestigate ways to extend the representation in order to handle gaps between clusters so that noisy bands can be handled better. It would also be interesting to employ the use of not only divergence measures between adjacent bands in the image, but between bands across the spectrum from each other, so clustering could be improved. Furthermore, the technique for choosing cluster representatives in this work was based on selecting the band with the maximum entropy in the cluster. This was done to ensure high entropy bands were chosen, but also led to bands that were outliers in a cluster being selected as representatives. Other clustering methods make use of other techniques such as mutual information in order to select representatives, and this could be further investigated.

Lastly is the topic of using these techniques for the real-time optimization of hyperspectral sensors. In the specialization project that was done in the Fall of 2018 is was established that one of the promising way of doing optimization of the sensors during flight was to do band merging. This would combine the incoming light of several adjacent wavelength to increase the value of the signal so the ratio to the noise is smaller. Since an intermediate output of the band selection algorithm proposed in this thesis is a clustering of adjacent bands, these clusters could be used for band merging. The algorithm would need a few adaptations to for it to work in this setting. Firstly, there needs to be done more research into the performance of the algorithm if it is to be executed in an online, continuous fashion. Secondly, the basis of the objective functions are measures based around a probability distribution that requires an entire image to function. Research must be done into how this probability function can be measured and estimated when there is a continuous stream of data flowing into the system. Lastly, for such an application it is not enough to output a set of Pareto optimal solutions, since a decision about how many clusters to use must be made. Therefore it would be useful to look into systems to support this decision making process, either by using the raw data acquired or some preferences on the desired separability between materials or what target is to be looked for.

This project only lasted for six months and has only begun to uncover the possibilities of evolutionary optimization in this field. Hopefully, more research will be made into the subject, at NTNU or elsewhere, so that the capabilities of hyperspectral images to give detailed data about how humanity best can cultivate, take care of, and make use of this Earth can be utilized to its fullest extent.

Bibliography

Kokaly, Raymond F et al. (2017). *USGS spectral library version 7*. Tech. rep. US Geological Survey, p. 61. DOI: https://doi.org/10.3133/ds1035.

NASA (2019a). *The Multispectral Scanner System*. URL: https://landsat.gsfc.nasa.gov/the-multispectral-scanner-system/.

— (2019b). *AVIRIS Concept*. URL: https://aviris.jpl.nasa.gov/aviris/concept.html.

Landgrebe, David (2002). "Hyperspectral image data analysis". In: *IEEE Signal Processing Magazine* 19.1, pp. 17–28. ISSN: 10535888. DOI: 10.1109/79.974718. URL: http://ieeexplore.ieee.org/document/974718/.

Deb, Kalyanmoy et al. (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2, pp. 182–197. ISSN: 1089778X. DOI: 10.1109/4235.996017. URL: http://ieeexplore.ieee.org/document/996017/.

Shannon, C E (1948). "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3, pp. 379–423. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

Kullback, S and R A Leibler (1951). "On Information and Sufficiency". In: *Ann. Math. Statist.* 22.1, pp. 79–86. DOI: 10.1214/aoms/1177729694. URL: https://doi.org/10.1214/aoms/1177729694.

Chang, Chih-Chung and Chih-Jen Lin (2011). "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2.3, pp. 1–27. ISSN: 21576904. DOI: 10.1145/1961189.1961199. URL: http://dl.acm.org/citation.cfm?doid=1961189.1961199.

Vane, Gregg, A. F. H. Goetz, and J. B. Wellman (1984). "Airborne imaging spectrometer: A new tool for remote sensing". In: *IEEE Transactions on Geoscience and Remote Sensing* GE-22.6, pp. 546–549. ISSN: 0196-2892. DOI: 10.1109/TGRS.1984.6499168. URL: http://ieeexplore.ieee.org/document/6499168/.

Wiersma, Daniel J. and David A. Landgrebe (1980). "Analytical design of multispectral sensors". In: *IEEE Transactions on Geoscience and Remote Sensing* GE-18.2, pp. 180–189. ISSN: 15580644. DOI: 10.1109/TGRS.1980.350271.

Harsanyi, J.C. and C.-I. Chang (1994). "Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach". In: *IEEE Transactions on Geoscience and Remote Sensing* 32.4, pp. 779–785. ISSN: 01962892. DOI: 10.1109/36.298007. URL: http://ieeexplore.ieee.org/document/298007/.

Jing Wang and Chein-I Chang (2006). "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis". In: *IEEE Transactions on Geoscience and Remote Sensing* 44.6, pp. 1586–1600. ISSN: 0196-2892. DOI: 10.1109/tgrs.2005.863297.

Ye Zhang et al. (1999). "Adaptive subspace decomposition for hyperspectral data dimensionality reduction". In: *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*. 3. IEEE, pp. 326–329. ISBN: 0-7803-5467-2. DOI: 10.1109/ICIP.1999.822910. URL: http://ieeexplore.ieee.org/document/822910/.

Petrie, G.M., P.G. Heasler, and T. Warner (1998). "Optimal band selection strategies for hyperspectral data sets". In: *IGARSS '98. Sensing and Managing the Environment. 1998 IEEE International Geoscience and Remote Sensing. Symposium Proceedings. (Cat. No.98CH36174)*. Vol. 99352. 509. IEEE, pp. 1582–1584. ISBN: 0-7803-4403-0. DOI: 10.1109/IGARSS.1998.691626. URL: http://ieeexplore.ieee.org/document/691626/.

Serpico, S.B. and Lorenzo Bruzzone (2001). "A new search algorithm for feature selection in hyperspectral remote sensing images". In: *IEEE Transactions on Geoscience and Remote Sensing* 39.7, pp. 1360–1367. ISSN: 01962892. DOI: 10.1109/36.934069. URL: http://ieeexplore.ieee.org/document/934069/.

Conese, Claudio and Fabio Maselli (1993). "Selection of optimum bands from TM scenes through mutual information analysis". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 48.3, pp. 2–11. ISSN: 09242716. DOI: 10.1016/0924-2716(93)90059-V. URL: https://linkinghub.elsevier.com/retrieve/pii/092427169390059V.

Sotoca, J.M., F. Pla, and A.C. Klaren (2004). "Unsupervised band selection for multispectral images using information theory". In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 3. 2. IEEE, pp. 510–513. ISBN: 0-7695-2128-2. DOI: 10.1109/ICPR.2004.1334578. URL: http://ieeexplore.ieee.org/document/1334578/.

Gong, Maoguo, Mingyang Zhang, and Yuan Yuan (2016). "Unsupervised Band Selection Based on Evolutionary Multiobjective Optimization for Hyperspectral Images". In: *IEEE Transactions on Geoscience and Remote Sensing* 54.1, pp. 544–557. ISSN:

0196-2892. DOI: 10.1109/TGRS.2015.2461653. URL: http://ieeexplore.ieee.org/document/7214263/.

Guo, Baofeng et al. (2006). "Band Selection for Hyperspectral Image Classification Using Mutual Information". In: *IEEE Geoscience and Remote Sensing Letters* 3.4, pp. 522–526. ISSN: 1545-598X. DOI: 10.1109/LGRS.2006.878240. URL: http://ieeexplore.ieee.org/document/1715309/.

Hossain, Md Ali, Xiuping Jia, and Mark Pickering (2012). "Improved feature selection based on a mutual information measure for hyperspectral image classification". In: *2012 IEEE International Geoscience and Remote Sensing Symposium*. Mi. IEEE, pp. 3058–3061. ISBN: 978-1-4673-1159-5. DOI: 10.1109/IGARSS.2012.6350780. URL: http://ieeexplore.ieee.org/document/6350780/.

Yang, Ronglu et al. (2017). "Representative band selection for hyperspectral image classification". In: *Journal of Visual Communication and Image Representation* 48, pp. 396–403. ISSN: 10959076. DOI: 10.1016/j.jvcir.2017.02.002. URL: http://dx.doi.org/10.1016/j.jvcir.2017.02.002.

Feng, Jie et al. (2016). "Unsupervised feature selection based on maximum information and minimum redundancy for hyperspectral images". In: *Pattern Recognition* 51, pp. 295–309. ISSN: 00313203. DOI: 10.1016/j.patcog.2015.08.018. URL: http://dx.doi.org/10.1016/j.patcog.2015.08.018.

Lorencs, A., I. Mednieks, and J. Sinica-Sinavskis (2018). "Selection of informative hyperspectral band subsets based on entropy and correlation". In: *International Journal of Remote Sensing* 39.20, pp. 6931–6948. ISSN: 0143-1161. DOI: 10.1080/01431161.2018.1468107. URL: https://doi.org/10.1080/01431161.2018.1468107%20https://www.tandfonline.com/doi/full/10.1080/01431161.2018.1468107.

Zhang, Mingyang, Maoguo Gong, and Yongqiang Chan (2018). "Hyperspectral band selection based on multi-objective optimization with high information and low redundancy". In: *Applied Soft Computing* 70, pp. 604–621. ISSN: 15684946. DOI: 10.1016/j.asoc.2018.06.009. URL: https://doi.org/10.1016/j.asoc.2018.06.009%20https://linkinghub.elsevier.com/retrieve/pii/S1568494618303326.

Du, Qian and He Yang (2008). "Similarity-Based Unsupervised Band Selection for Hyperspectral Image Analysis". In: *IEEE Geoscience and Remote Sensing Letters* 5.4, pp. 564–568. ISSN: 1545-598X. DOI: 10.1109/LGRS.2008.2000619. URL: http://ieeexplore.ieee.org/document/4656481/.

Martinez-Uso, Adolfo et al. (2007). "Clustering-Based Hyperspectral Band Selection Using Information Measures". In: *IEEE Transactions on Geoscience and Remote Sensing* 45.12, pp. 4158–4171. ISSN: 0196-2892. DOI: 10.1109/TGRS.2007.904951. URL: http://www.springerlink.com/index/10.1631/jzus.C1000304%20http://ieeexplore.ieee.org/document/4378560/.

Wang, Qi, Fahong Zhang, and Xuelong Li (2018). "Optimal Clustering Framework for Hyperspectral Band Selection". In: *IEEE Transactions on Geoscience and Remote Sensing* 56.10, pp. 1–13. ISSN: 0196-2892. DOI: 10.1109/TGRS.2018.2828161. URL: https://ieeexplore.ieee.org/document/8356741/.

Xie, Fuding et al. (2019). "Unsupervised band selection based on artificial bee colony algorithm for hyperspectral image classification". In: *Applied Soft Computing* 75, pp. 428–440. ISSN: 15684946. DOI: 10.1016/j.asoc.2018.11.014. URL: https://doi.org/10.1016/j.asoc.2018.11.014%20https://linkinghub.elsevier.com/retrieve/pii/S156849461830646X.

Datta, Aloke, Susmita Ghosh, and Ashish Ghosh (2015). "Combination of Clustering and Ranking Techniques for Unsupervised Band Selection of Hyperspectral Images". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8.6, pp. 2814–2823. ISSN: 1939-1404. DOI: 10.1109/JSTARS.2015.2428276. URL: http://ieeexplore.ieee.org/document/7112088/.

Zhang, Liangpei et al. (2007). "Dimensionality reduction based on clonal selection for hyperspectral imagery". In: *IEEE Transactions on Geoscience and Remote Sensing* 45.12, pp. 4172–4186. ISSN: 01962892. DOI: 10.1109/TGRS.2007.905311.

Su, Hongjun et al. (2014). "Optimized Hyperspectral Band Selection Using Particle Swarm Optimization". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7.6, pp. 2659–2670. ISSN: 1939-1404. DOI: 10.1109/JSTARS.2014.2312539. URL: http://ieeexplore.ieee.org/document/6783712/.

Paul, Arati et al. (2015). "Band selection in hyperspectral imagery using spatial cluster mean and genetic algorithms". In: *GIScience and Remote Sensing* 52.6, pp. 643–659. ISSN: 15481603. DOI: 10.1080/15481603.2015.1075180. URL: http://dx.doi.org/10.1080/15481603.2015.1075180.

Xu, Xia, Zhenwei Shi, and Bin Pan (2017). "A New Unsupervised Hyperspectral Band Selection Method Based on Multiobjective Optimization". In: *IEEE Geoscience and Remote Sensing Letters* 14.11, pp. 2112–2116. ISSN: 1545-598X. DOI: 10.1109/LGRS.2017.2753237. URL: http://ieeexplore.ieee.org/document/8057978/.

Hyperspectral Remote Sensing Scenes (2014). URL: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.

Pearlman, Jay S. et al. (2003). "Hyperion, a space-based imaging spectrometer". In: *IEEE Transactions on Geoscience and Remote Sensing* 41.6 PART I, pp. 1160–1173. ISSN: 01962892. DOI: 10.1109/TGRS.2003.815018.

Kunkel, B. et al. (1988). "ROSIS (Reflective Optics System Imaging Spectrometer) - A Candidate Instrument For Polar Platform Missions". In: *Optoelectronic Technologies for Remote Sensing from Space*. Ed. by C. Stuart Bowyer and John S. Seeley. Vol. 0868. April 1988, p. 134. DOI: 10.1117/12.943611. URL: http://

proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.943611.

Appendix A

Dataset properties

A.1 Indian Pines

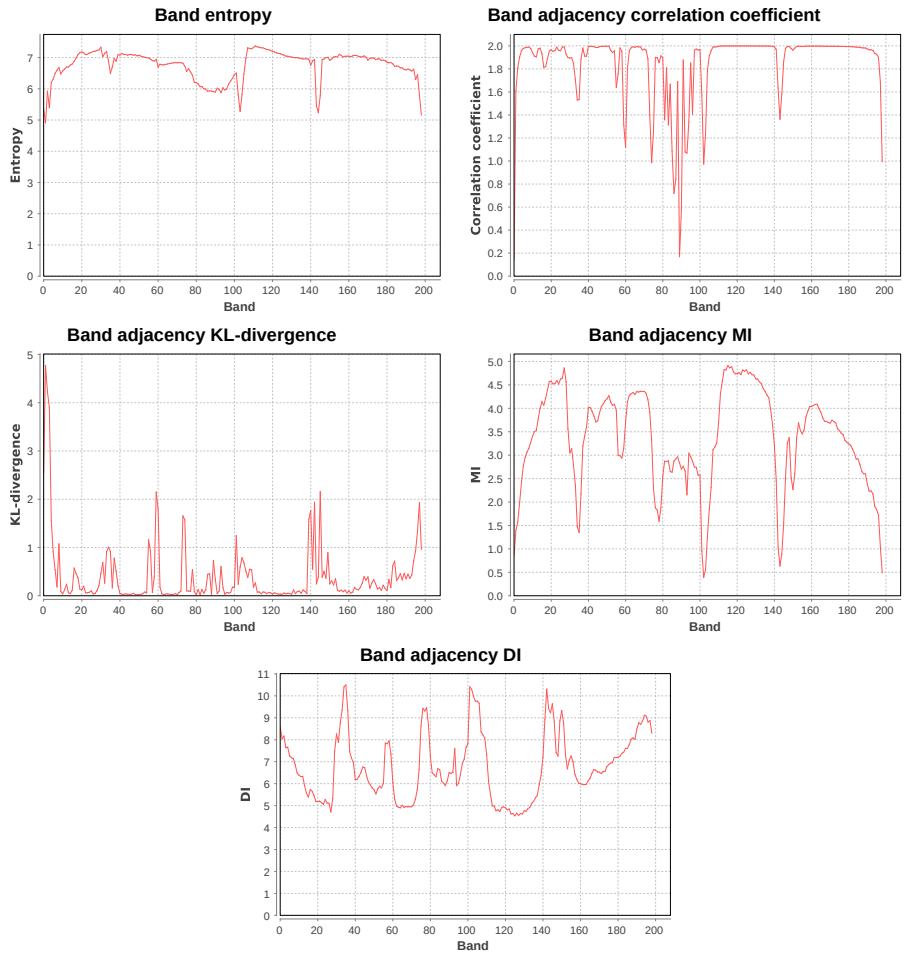

Figure A.1: Information theoretic properties of the Indian Pines dataset

A.2 Botswana

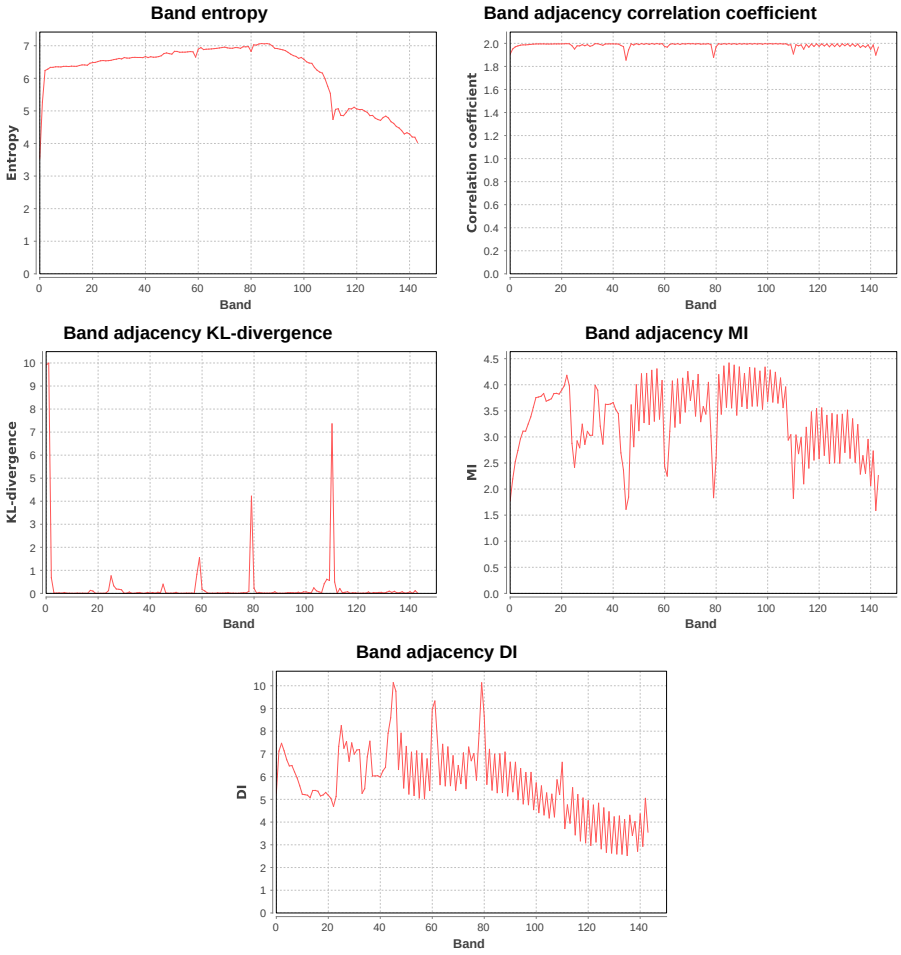

Figure A.2: Information theoretic properties of the Botswana dataset

A.3 Pavia University

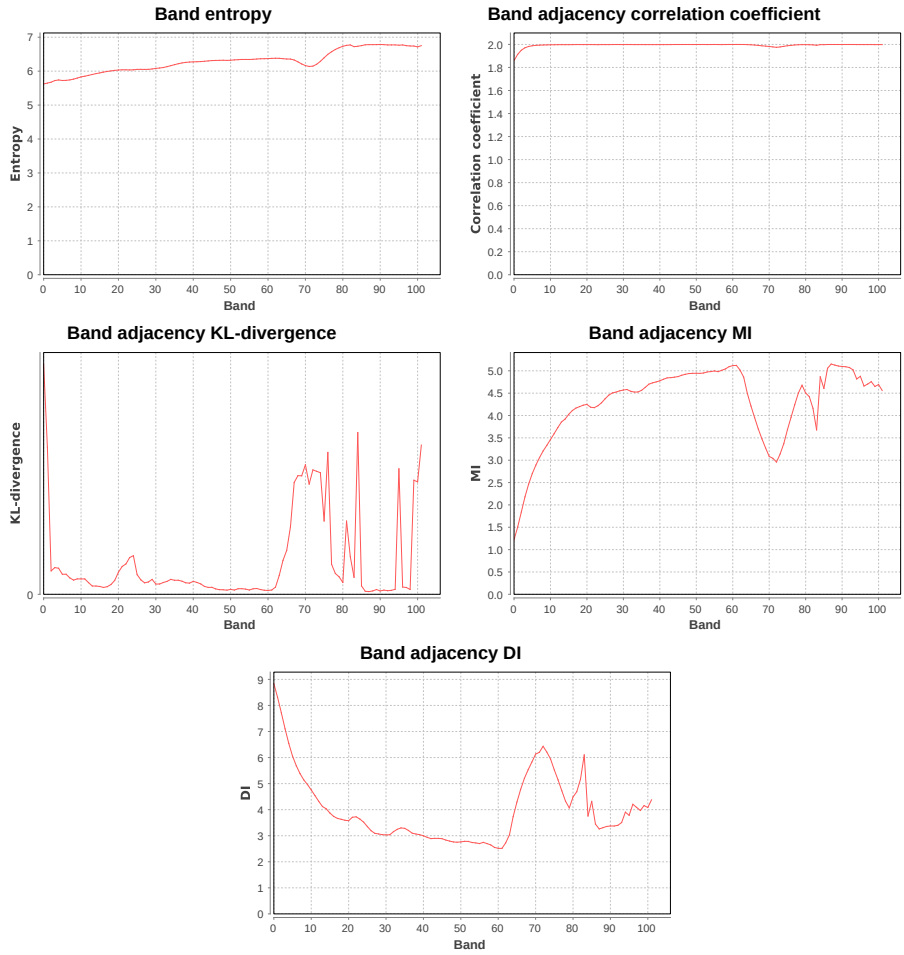

Figure A.3: Information theoretic properties of the Pavia University dataset

Appendix B

Results

B.1 Bands selected by DUMB

#	Bands selected	Boundaries
2	30, 111	104
3	30, 33, 111	33, 104
4	30, 33, 111, 155	33, 77, 143
5	30, 33, 107, 111, 155	33, 77, 109, 149
6	30, 41, 107, 111, 155, 173	36, 77, 109, 143, 171
7	30, 33, 59, 107, 111, 155, 173	33, 58, 77, 109, 143, 171
8	30, 33, 59, 107, 111, 155, 168, 185	33, 58, 78, 109, 143, 166, 184
9	30, 33, 59, 100, 107, 111, 149, 155, 178	33, 58, 78, 101, 109, 143, 151, 177
10	30, 33, 59, 100, 107, 111, 149, 155, 168, 185	33, 58, 78, 101, 109, 143, 151, 167, 184
11	8, 30, 41, 59, 77, 107, 111, 149, 155, 168, 183	12, 36, 58, 77, 101, 109, 143, 151, 167, 183
12	8, 29, 30, 41, 59, 100, 110, 111, 155, 163, 173, 188	12, 30, 36, 57, 78, 101, 111, 143, 156, 172, 188
13	8, 29, 30, 41, 59, 100, 107, 111, 121, 155, 163, 173, 188	12, 30, 36, 57, 78, 101, 109, 121, 143, 156, 172, 188
14	8, 29, 30, 41, 59, 100, 107, 111, 128, 155, 163, 173, 180, 191	12, 30, 36, 57, 78, 101, 109, 127, 143, 156, 171, 180, 189
15	8, 29, 30, 41, 46, 59, 76, 101, 110, 111, 128, 149, 155, 168, 185	12, 30, 36, 45, 57, 75, 90, 102, 111, 127, 143, 151, 166, 184
16	8, 29, 30, 41, 46, 59, 76, 101, 110, 111, 128, 155, 163, 173, 180, 191	12, 30, 36, 45, 57, 75, 90, 102, 111, 127, 143, 156, 171, 180, 189
17	8, 29, 30, 41, 46, 59, 76, 101, 110, 111, 128, 149, 155, 163, 173, 180, 194	12, 30, 36, 45, 57, 75, 90, 102, 111, 127, 143, 151, 161, 172, 180, 192
18	8, 29, 30, 41, 46, 59, 76, 84, 101, 110, 111, 128, 149, 155, 163, 173, 180, 194	12, 30, 36, 45, 57, 75, 84, 90, 102, 111, 127, 143, 151, 161, 172, 180, 192
19	6, 18, 30, 33, 41, 46, 59, 76, 101, 110, 111, 122, 135, 149, 155, 163, 173, 180, 194	7, 19, 32, 36, 44, 57, 75, 90, 102, 111, 122, 135, 143, 151, 161, 172, 180, 192
20	6, 18, 30, 33, 41, 46, 59, 76, 101, 105, 110, 111, 122, 135, 149, 155, 163, 173, 180, 194	7, 19, 32, 36, 44, 57, 76, 90, 102, 106, 111, 122, 135, 143, 151, 161, 172, 180, 192
21	6, 18, 30, 33, 41, 46, 59, 76, 83, 101, 105, 110, 111, 122, 135, 149, 155, 163, 173, 180, 194	7, 19, 32, 36, 44, 57, 76, 82, 90, 102, 106, 111, 122, 135, 143, 151, 161, 172, 180, 192
22	7, 18, 30, 33, 41, 46, 59, 71, 76, 83, 101, 105, 110, 111, 122, 135, 149, 155, 163, 173, 180, 194	8, 19, 32, 36, 44, 57, 63, 76, 82, 90, 102, 106, 111, 122, 135, 143, 151, 161, 172, 180, 192
23	6, 18, 30, 33, 41, 46, 59, 71, 76, 83, 101, 105, 110, 111, 122, 135, 149, 155, 163, 173, 180, 191, 194	7, 19, 32, 36, 44, 57, 63, 76, 82, 90, 102, 106, 111, 122, 135, 143, 151, 161, 172, 180, 189, 194
24	6, 18, 30, 33, 41, 46, 50, 59, 71, 76, 83, 101, 105, 110, 111, 122, 135, 149, 155, 163, 173, 180, 191, 194	7, 19, 32, 36, 44, 49, 57, 63, 76, 82, 90, 102, 106, 111, 122, 135, 143, 151, 161, 172, 180, 189, 194
25	6, 18, 27, 30, 37, 41, 46, 52, 59, 71, 76, 83, 101, 105, 110, 111, 122, 135, 149, 155, 163, 173, 180, 191, 194	7, 19, 28, 36, 39, 44, 52, 57, 63, 76, 82, 90, 102, 106, 111, 122, 135, 143, 151, 161, 172, 180, 189, 194
26	7, 18, 20, 30, 33, 39, 41, 46, 52, 59, 71, 76, 83, 101, 105, 110, 111, 122, 135, 149, 155, 163, 173, 180, 191, 194	8, 19, 26, 32, 36, 41, 44, 52, 57, 63, 76, 82, 90, 102, 106, 111, 122, 135, 143, 151, 161, 172, 180, 189, 194
27	6, 18, 27, 30, 37, 41, 46, 52, 59, 71, 76, 83, 91, 101, 106, 110, 111, 122, 135, 142, 153, 155, 163, 173, 180, 191, 194	7, 19, 28, 36, 39, 44, 52, 57, 63, 76, 82, 88, 94, 104, 107, 111, 122, 135, 141, 147, 155, 161, 172, 180, 189, 194
29	6, 18, 27, 30, 37, 41, 46, 52, 59, 71, 76, 83, 91, 101, 106, 110, 111, 122, 135, 142, 153, 155, 163, 168, 176, 178, 180, 191, 194	7, 19, 28, 36, 39, 44, 52, 57, 63, 76, 82, 88, 94, 104, 107, 111, 122, 135, 141, 147, 155, 159, 168, 174, 177, 180, 189, 194
31	7, 18, 20, 30, 33, 39, 41, 46, 48, 59, 71, 76, 83, 91, 100, 101, 106, 110, 111, 122, 135, 142, 153, 155, 163, 168, 176, 178, 180, 191, 194	8, 19, 26, 32, 36, 41, 43, 48, 58, 63, 76, 82, 88, 94, 101, 104, 107, 111, 122, 135, 141, 147, 155, 159, 168, 174, 177, 180, 189, 194

Table B.1: Bands selected by DUMB on the Indian Pines dataset (length 2-31)

#	Bands selected	Boundaries
32	7, 18, 20, 30, 33, 39, 41, 46, 48, 54, 59, 71, 76, 83, 91, 100, 101, 106, 110, 111, 122, 135, 142, 153, 155, 163, 168, 176, 178, 180, 191, 194	8, 19, 26, 32, 36, 41, 43, 48, 54, 58, 63, 76, 82, 88, 94, 101, 104, 107, 111, 122, 135, 141, 147, 155, 159, 168, 174, 177, 180, 189, 194
33	7, 18, 20, 30, 33, 39, 41, 46, 48, 54, 59, 71, 72, 76, 83, 91, 100, 101, 106, 110, 111, 122, 135, 142, 153, 155, 163, 168, 176, 178, 180, 191, 194	8, 19, 26, 32, 36, 41, 43, 48, 54, 58, 63, 72, 76, 82, 88, 94, 101, 104, 107, 111, 122, 135, 141, 147, 155, 159, 168, 174, 177, 180, 189, 194
34	8, 18, 20, 30, 33, 39, 41, 46, 48, 54, 59, 71, 76, 83, 91, 100, 101, 106, 110, 111, 122, 128, 135, 142, 153, 155, 163, 168, 173, 176, 178, 180, 191, 194	10, 19, 26, 32, 36, 41, 43, 48, 54, 58, 63, 76, 82, 88, 94, 101, 104, 107, 111, 122, 128, 135, 141, 147, 155, 159, 168, 170, 174, 177, 180, 189, 194
35	0, 5, 8, 18, 20, 30, 33, 39, 41, 46, 48, 59, 71, 76, 83, 91, 100, 101, 106, 110, 111, 122, 128, 135, 142, 153, 155, 163, 168, 173, 176, 178, 180, 191, 194	2, 6, 10, 19, 26, 32, 36, 41, 43, 48, 58, 63, 76, 82, 88, 94, 101, 104, 107, 111, 122, 128, 135, 141, 147, 155, 159, 168, 170, 174, 177, 180, 189, 194
36	0, 5, 8, 18, 20, 30, 33, 39, 41, 46, 48, 54, 59, 71, 76, 83, 91, 100, 101, 106, 110, 111, 122, 128, 135, 142, 153, 155, 163, 168, 173, 176, 178, 180, 191, 194	2, 6, 10, 19, 26, 32, 36, 41, 43, 48, 54, 58, 63, 76, 82, 88, 94, 101, 104, 107, 111, 122, 128, 135, 141, 147, 155, 159, 168, 170, 174, 177, 180, 189, 194
37	0, 5, 8, 18, 20, 30, 33, 39, 41, 46, 48, 54, 59, 65, 71, 76, 83, 91, 100, 101, 106, 110, 111, 122, 128, 135, 142, 153, 155, 163, 168, 173, 176, 178, 180, 191, 194	2, 6, 10, 19, 26, 32, 36, 41, 43, 48, 54, 58, 63, 66, 76, 82, 88, 94, 101, 104, 107, 111, 122, 128, 135, 141, 147, 155, 159, 168, 170, 174, 177, 180, 189, 194
41	0, 5, 8, 18, 20, 30, 33, 39, 41, 46, 48, 54, 59, 65, 71, 76, 83, 91, 100, 101, 106, 110, 111, 114, 122, 135, 137, 146, 149, 155, 163, 168, 173, 176, 180, 182, 185, 189, 191, 196, 197	2, 6, 10, 19, 26, 32, 36, 41, 43, 48, 54, 58, 63, 66, 76, 82, 88, 94, 101, 104, 107, 111, 114, 122, 135, 137, 143, 147, 151, 158, 167, 171, 176, 180, 182, 184, 189, 191, 195, 197
42	0, 5, 8, 18, 20, 30, 33, 39, 41, 46, 48, 54, 59, 61, 68, 71, 76, 83, 91, 100, 101, 106, 110, 111, 114, 122, 135, 137, 146, 149, 155, 163, 168, 173, 176, 180, 182, 185, 189, 191, 196, 197	2, 6, 10, 19, 26, 32, 36, 41, 43, 48, 54, 58, 60, 65, 69, 76, 82, 88, 94, 101, 104, 107, 111, 114, 122, 135, 137, 143, 147, 151, 158, 167, 171, 176, 180, 182, 184, 189, 191, 195, 197
44	2, 6, 14, 15, 20, 27, 30, 32, 33, 39, 41, 46, 48, 54, 59, 61, 68, 71, 76, 83, 91, 98, 101, 106, 110, 111, 114, 122, 135, 137, 146, 149, 155, 163, 168, 173, 176, 180, 182, 185, 189, 191, 196, 197	3, 7, 15, 16, 22, 28, 32, 33, 36, 41, 43, 48, 54, 58, 60, 65, 69, 76, 82, 88, 94, 99, 104, 107, 111, 114, 122, 135, 137, 143, 147, 151, 158, 167, 171, 176, 180, 182, 184, 189, 191, 195, 197
48	0, 5, 8, 11, 13, 17, 20, 27, 28, 30, 33, 39, 41, 46, 48, 54, 59, 61, 68, 71, 74, 76, 77, 84, 91, 98, 101, 106, 110, 111, 114, 122, 135, 137, 146, 149, 155, 163, 168, 173, 176, 180, 182, 185, 189, 191, 196, 197	2, 6, 9, 12, 14, 18, 23, 28, 29, 33, 36, 41, 43, 48, 54, 58, 60, 65, 69, 74, 76, 77, 84, 88, 94, 99, 104, 107, 111, 114, 122, 135, 137, 143, 147, 151, 158, 167, 171, 176, 180, 182, 184, 189, 191, 195, 197

Table B.2: Bands selected by DUMB on the Indian Pines dataset (length 32-48)