Bayesian hyperspectral image segmentation with discriminative class learning

Janete da Silva Borges

Thesis submitted to the Sciences Faculty of University of Porto, Portugal, for the partial fulfillment of the requirements for the degree of Doctor of Philosophy made under the supervision of Doctor André R.S. Marçal, Assistant Professor at the Faculty of Sciences of the University of Porto, Portugal and Doctor José Bioucas-Dias, Associate Professor at the Technical University of Lisbon, Portugal.

April 2008

To Óscar, my parents,
and my brother

Resumo

Reconhecimento de padrões e detecção remota são áreas de investigação que, nos últimos anos, têm sofrido grandes desenvolvimentos. O facto da detecção remota ser uma área propícia às aplicações dos algoritmos de reconhecimento de padrões, não é alheio a esses desenvolvimentos.

A segmentação de imagens tem sido um dos problemas mais estudados em reconhecimento de padrões. A sua aplicabilidade a um vasto número de domínios tem levado a diversas abordagens, formulações e ferramentas. A detecção remota é um dos domínios onde a segmentação de imagens tem um papel de extrema importância.

O processamento de imagens hiper-espectrais tem sido um dos grandes desafios para os algoritmos de reconhecimento de padrões dado a problemática relacionada com o fenómeno de Hughes. A informação detalhada acerca das assinaturas espectrais fornecida pelos sensores hiper-espectrais, levou ao desenvolvimento de novos algoritmos capazes de lidar com a elevada dimensionalidade dos dados. Contudo, esta é ainda uma área de investigação em desenvolvimento.

Este trabalho, foca-se no desenvolvimento de métodos de classificação e segmentação capazes de lidar com a elevada dimensionalidade dos dados, nomeadamente das imagens hiper-espectrais.

Esta tese apresenta um novo algoritmo de segmentação Bayesiano com aprendizagem discriminativa das classes. O método proposto é composto por duas

partes: a aprendizagem das densidades das classes e a inclusão de informação espacial. O facto de a aprendizagem dos parâmetros necessários em cada parte do processo ser feita em dois passos consecutivos e não simultneos conduz a procedimentos computacionalmente mais leves. As densidades das classes são determinadas utilizando o algoritmo discriminativo proposto nesta tese: o algoritmo rápido de regressão multinomial esparso (Fast Sparse Multinomial Regression (FSMLR)). O algoritmo FSMLR introduz uma modificação no procedimento iterativo usado no SMLR. De forma a impor a esparcidade, o SMLR utiliza um *prior* de Laplace. Em conjunto com a modificação iterativa, o uso de um *prior* alternativo (o *prior* de Jeffreys) é também proposto de forma a evitar a afinação o parâmetro de esparcidade.
A informação contextual é incluída na forma de dependências espaciais impostas por um *prior* de Markov-Gibbs multi-nível (MLL). A segmentação óptima é dada pela solução de um problema de optimização discreto, que é determinada pelo algoritmo α-Expansion.

A performance da abordagem proposta é ilustrada num conjunto de experiências executadas em diversas condições, tendo em conta a dimensão do conjunto de treino. Quer o passo de estimação das densidades das classes, quer o passo de segmentação são avaliados separadamente e os resultados comparados com resultados de métodos de classificação/segmentação recentes.

Abstract

Pattern recognition and remote sensing are two areas of research that have suffered great developments in recent years. The fact that remote sensing is one of the most suitable areas for the application of pattern recognition algorithms is not strange to those developments.

Image segmentation has been one of the most studied problems in pattern recognition. Its application to a wide range of domains has lead to many different approaches, formulations and tools. Remote sensing is one of the domains where image segmentation plays a role of great importance.

Hyperspectral imaging has been one of the major challenges to pattern recognition algorithms due the problematic related to the Hughes phenomenon. The detailed information about spectral signatures provided by hyperspectral sensors lead to the development of new algorithms capable of properly handling the high dimensionality of the data. Nevertheless, this is still an active area of research.

This work focus on the development of methods for classification and segmentation capable of dealing with high dimensional datasets, namely with hyperspectral images.

This thesis presents a new Bayesian segmentation algorithm with discriminative class learning. The proposed method comprises two parts: the learning of the class densities and the inclusion of spatial information. The fact that the parameters required for each part of the process are learnt in two consecutive,

but nonsimultaneous steps, conducts to lighter computational procedures. The class densities are determined using a discriminative algorithm proposed in this thesis: the Fast Sparse Multinomial Regression (FSMLR) algorithm. FSMLR algorithm introduces a modification to the iterative method used in SMLR. To enforce sparsity, the SMLR uses the Laplacian prior. In addition to the iterative modification, the use of an alternative prior (the Jeffreys prior) is also proposed to avoid the tuning of the sparsity parameter.
The contextual information is added in the form of spatial dependencies enforced by a Multi-Level (MLL) Markov-Gibbs prior. The optimal segmentation is given by the solution of a discrete optimization problem, which is efficiently solved through the α-Expansion graph cut based algorithm

The performance of the proposed approach is illustrated in a set of experiments carried out in different conditions, regarding the size of the training set. Both the class density step and segmentation step are evaluated separately and results are compared with recently introduced hyperspectral classification/segmentation methods.

Résumé

La reconnaissance de formes et la télédétection sont deux domaines de recherche qui ont, ces dernières années, souffert de grands développements. Le fait que la télédétection est l'un des secteurs les plus appropriés pour l'application des algorithmes de reconnaissance de formes n'est pas étrange à ces développements. La segmentation d'image a été l'un des problèmes les plus étudiés en la reconnaissance des formes. Son applicabilité à des nombreux domaines a conduit à plusieurs différents approches, formulations et outils. La télédétection est l'un des domaines où la segmentation d'image joue un rôle d'extrême importance.
Le traitement d'images hyperspectrales a été l'un des principaux défis qui s'ont présenté aux algorithmes de reconnaissance de formes, à cause de la problématique relative au phénomène de Hughes.
L'information détaillée sur les signatures spectrales fournie par les capteurs hyper-spectrales a mené au développement de nouveaux algorithmes capables de manipuler correctement la grande dimensionnalité des données. Néanmoins, celle-ci est un domaine de recherche encore actif.

Ce travail se focalise dans le développement de méthodes de classification et segmentation capables de traiter la grande dimensionnalité des données, notamment des images hyperspectrales.
Cette thèse présente un nouvel algorithme Bayésien de segmentation avec apprentissage discriminatif des classes. La méthode proposée comporte deux

parties : l'apprentissage des densités des classes et l'inclusion d'information spatiale. Le fait de l'apprentissage des paramètres nécessaires en chaque partie de la procédure être fait dans deux étapes consécutives et non simultanées conduit à des procédures informatiquement plus légères. Les densités des classes sont déterminées en utilisant l'algorithme discriminatif proposé dans cette thèse: l'algorithme de régression polynômiale clairsemé rapide (Fast Sparse Multinomial Regression (FSMLR)). L'algorithme FSMLR introduit une modification dans la procédure itérative utilisée dans SMLR. De manière à imposer l'éparsement, SMLR utilise un *prior* de Laplace. En plus de la modification itérative, on propose aussi l'utilisation d'un *prior* alternatif (le *prior* de Jeffreys) pour éviter le raffinage du paramètre d'éparsement.
L'information contextuelle est ajoutée sous forme de dépendances spatiales imposées par un *prior* multiniveaux (Multi-Level, MLL) de Markov-Gibbs. La segmentation optimale est donnée par la solution d'un problème d'optimisation discrète, qui est efficacement résolu par l'algorithme α-Expansion.

L'exécution de l'approche proposée est illustré dans un ensemble d'expŕiences effectués aux conditions diverses, concernant la dimension de l'ensemble d'entraînement. L'étape de densité de classe et l'étape de segmentation sont évaluées séparément et les résultats sont comparés à ceux de méthodes de classification/segmentation récemment présentées.

Acknowledgments

It is a pleasure to express my gratitude to all the persons that somehow contributed to this thesis. The collaboration and help demonstrated by everyone surrounding me in the last four years ought to be acknowledge.

Special words have to be directed to my supervisors, Professor André Marçal and Professor Bioucas-Dias. I thank Professor André Marçal for all the help, ideas and comments given during this work and for providing guidance and support since 2001. I also gratefully thank Professor Bioucas-Dias, for his important contributions, valuable opinions, discussions and ideas.

I would like to thank also to my colleagues at DMA and IST for the support provided whenever I asked. *Merci* Alexandra ;)

I would like to acknowledge David Landgrebe for providing the Indian Pines data, Paolo Gamba for providing the Pavia data, and Vladimir Kolmogorov for the max-flow/min-cut C++ code made available on the web (see [18] for more details). I also thank the financial support from FCT - Fundação para a Ciência e a Tecnologia, grant SFRH/BD/17191/2004.

Para o meu marido, meus pais e meu irmão, não há palavras que traduzam toda a minha gratidão para com eles. Devo-lhes tudo, e por tudo... muito obrigada! É um orgulho ser *vossa*. À minha família e amigos, obrigada por fazerem da minha vida num mar de sorrisos.

Contents

List of Tables

List of Figures

Symbols and Abbreviations

Symbols

$\mathbf{y}$	Image of labels
$\mathbf{x}$	Feature images
x	input vector
$\mathcal{S}$	Set of pixels that compose the image
c	Number of columns of a digital image
l	Number of lines of a digital image
d	Dimensional space size
N	Number of patterns of a data set
K	Number of classes
$\mathcal{N}_i$	Neighbourhood of site i
β_{MLL}	Parameter that controls spatial continuity
σ_N	Variance noise of simulated feature images.

Abbreviations

BSDCL	Bayesian Segmentation with Discriminative Class Learning
FSMLR	Fast Sparse Multinomial Logistic Regression
EM	Electromagnetic
UV	Ultraviolet
IR	Infrared
OA	Overall Accuracy
AVIRIS	Airborne Visible/Infrared Imaging Spectrometer
ROSIS	Reflective Optics System Imaging Spectrometer
MLL	Multi-Level Logistic
ML	Maximum Likelihood
IRLS	Iteratively Reweighted Least Squares
MAP	Maximum *a posteriori*
MSE	Mean Square Error
USGS	United States Geological Survey
LDA	Linear Discriminant Analysis
GS	Gauss-Seidel

Chapter 1

Introduction

Pattern recognition is a tool of vital interest in the remote sensing. The primary goal of pattern recognition, correctly classify a pattern into one of several available classes, is frequently the solution for several problems in remote sensing. Conversely, the intensive use of pattern recognition techniques in the remote sensing field brought new developments to pattern recognition. This thesis is an example of that.

Pattern recognition holds several techniques with direct applications in remote sensing. Image classification and segmentation, feature extraction and selection, matching, target detection and unmixing algorithms are among the most used techniques in remote sensing area.

Although remarkably advances in pattern recognition have been made, the advances in remote sensing still present new challenges to pattern recognition.

Hyperspectral imaging expands and improves capability of multispectral imaging taking advantage of hundreds of contiguous spectral bands to uncover materials that usually cannot be resolved by multispectral sensors. This area has been showing to be a fast growing one in remote sensing. However, the large amount of data made available by hyperspectral sensors also comes with a price

that is related to the Hughes phenomenon: the difficulty in learning in high dimensional densities from a limited number of training samples. This has been one of the major problems to pattern recognition algorithms deal with.

Image classification is one of the most used tasks in remote sensing information processing. The classification of remote sensing imagery for production of land cover maps is much needed in several areas like forestry, agriculture, geology, hydrology, cartography, economics, geography, etc. There is a wide range of classification algorithm suitable for remote sensing image classification. The major developments in statistical pattern recognition theory were during the 1960's and 1970's, with the formulation of pattern recognition as a Bayes decision theory problem, nearest neighbour decision rules and density estimation, Fisher linear discriminant, K-means algorithm, among others techniques. Since the latter part of 1980's new algorithms like neural networks and support vector machines have been intensively applied to the classification of remote sensing images. However, when applied to hyperspectral images, the majority of these algorithms reveal problems in dealing with such an amount of information. Support vector machines are more suitable for this type of data, having shown good performance in dealing with high dimensional datasets.

The afore mentioned classification algorithms work based on the information of each pixel considered as an individual. To improve the results of this type of classification, contextual information should be added to the spectral information available. Although the statistical dependence of neighbouring pixels was considered in [113, 48], only recently the inclusion of contextual information in remote sensing problems has becoming more frequent. The mathematical foundation proposed by Geman and Geman [50] allowed for many posterior work on Markov random field models. These models provide a rigorous mathematical characterization of contextual information of textures from neighbouring pixels on an image. Nevertheless, the application of powerful classification and segmentation algorithms to hyperspectral images is compromised due the high

dimensionality of this type of data. The classification as well as the segmentation of high-dimensional data are therefore active areas of research.

Both the problem of learning in high dimensional spaces, as well as the inclusion of spatial information in the classification process motivated the work developed in this thesis. The work here presented has as main goal the development of classification and segmentation algorithms capable of deal with high dimensional datasets, namely the hyperspectral images. Based on this, one may divide the methodology here developed in two major parts: (i) the learning of the spectral densities and (ii) the introduction of spatial information. With respect to the former, we based our work on discriminative class learning algorithms due to their capacity of learning directly the densities of the labels given the featues, and their ability to produce sparse solutions. The sparsity of a classifier is of major importance when dealing with high dimensional datasets.

Our main contibution consists in a modification to the Sparse Multinomial Regression (SMLR) algorithm [66] wich resulted in the Fast Sparse Multinomial Regression algorithm [14]. The algorithm here proposed is abble to learn the class densities at a much lower computational cost than the original one, allowing its application to high-dimensional datasets.

The SMLR originally proposed makes use of a Laplacian prior to enforce the sparsity of the classifier. However, this implies the tunning of a sparsity parameter which, when dealing with high dimensional datasets leads to computational problems. In this work we propose the application of an alternative prior - the Jeffreys prior - to enforce the sparseness of the FSMLR [17] avoiding the parameter tunning.

Regarding the second part of the methodology - the inclusion of spatial information - we worked based on the MRF theory. We present a new Bayesian approach to hyperspectral image segmentation that boosts the performance of the discriminative classifiers [15]. This is achieved by combining class densities

based on discriminative classifiers with a Multi-Level Logistic Markov-Gibbs prior. The discrete optimization problem one is led to is solved efficiently via graph cut tools.

A study of the proposed methods in their different stages is carried out trough their application to hyperspectral images. Both synthetic and benchmarked datasets are used to evaluate the performance of either the densities learning stage, as well as the segmentation process. This evaluation is made trough the analysis of the overall accuracies of final results, as well as trough the analysis of the degree of sparseness promoted in each case. The response of the methods to different sizes of training sets will also be considered.

The present thesis is organized as follows:
Chapter 2 introduces the basic concepts of remote sensing. We start by reviewing the physics of remote sensing by presenting the information about the electromagnetic spectrum followed by the interaction of the electromagnetic radiation with both the atmosphere and ground targets. The process of how multispectral and hyperspectral sensors acquire and register the energy radiation ends this chapter.
Chapter 3 introduces the data classification problem. Most of the topics, like the Bayes decision theory, are well known and established theories and may be skipped. Nevertheless, they are used to introduce the main notation used throughout the text. General concepts of a data classification problems are given in the first section. We then focus our atention to the classification of multispectral and hyperspectral images. The chapter is concluded with the problematic of the spatial context in image classification.
In Chapter 4, we present the developed methodology to classify hyperspectral images with a bayesian discriminative approach wich includes spatial information using a multi-level Markov Gibbs prior. The SMLR method is first reviewd both with a Laplacian and a Jeffreys prior, and then we propose the Fast-

SMLR. The chapter carries on with the inclusion of contextual information in the process trough the introduction of a multi-level Markov-Gibbs prior. We conclude this chapter with the MAP segementation description with the α-expansion algorithm.

The datasets used to test and evaluate the developed methods, as well as the conducted experimental procedures, are presented in Chapter 5.

Chapter 6, presents the results of the application of the proposed method to several datasets. The results are grouped by dataset. For each dataset, experiments with different conditions were made based on the type of prior, the type of input function and inclusion of contextual information.

The final discussions and conclusions are presented in Chapter 7, ending with the outline of future work.

Chapter 2

Remote Sensing Fundamentals

Remote sensing is the science (and to some extent, art) of acquiring information about the Earth's surface without actually being in contact with it. This is done by sensing and recording reflected or emitted energy and processing, analyzing, and applying that information.

This is definition of Remote Sensing is given by the Canada Centre for Remote Sensing (http://www.ccrs.nrcan.gc.ca). To easily understand the meaning of remote sensing, let us say that remote sensing is a rather simple, familiar activity that we all do as a matter of daily life: observe all that surround us. Our eyes are our *remote sensors* that capture the light, sending a signal to our brain (our *processor*) which records the data and interprets this into information. Human vision may be used as a parallel to better understand what remote sensing is, however this is not usually interpreted as remote sensing. In practice, remote sensing refers to instrument-based techniques rather then human visual capacities. Hence, remote sensing is the science of obtaining information about an object, area or phenomenon trough the analysis of data acquired by a recording device that is not in physical or intimate contact with the object, area or phenomenon under study. The devices that measure or collect the

information of some property of an object, area or phenomenon are the sensors (camera, lasers, radio frequency receivers, radar systems, sonar, seismographs, gravimeters, magnetometers, etc.). The information acquired, depending on the sensor used, may be measurements of force fields, electromagnetic radiation, or acoustic energy.

This chapter is devoted to introducing the basic concepts of Remote Sensing (RS), restricted to the electromagnetic radiation. We will first introduce the physics of RS and then we will proceed to a characterization of multispectral and hyperspectral sensors.

2.1 The Physics of Remote Sensing

2.1.1 The Electromagnetic Spectrum

Electromagnetic (EM) energy can best be described as waves of electric and magnetic energy moving together through space, this type of energy is emitted by natural sources like the sun, the earth and the ionosphere.

EM energy-based sensors collect information based on the way that a body emit and reflect the EM energy, which is in the form of EM radiation. All EM radiation has fundamental properties and behaves in predictable ways according to the basics of wave theory. Depending on the behaviour of the wavelength, the EM radiation is generally classified into radio, microwave, infrared, the visible region we perceive as light, ultraviolet, X-rays and gamma rays (from longer to shorter wavelengths). The behaviour of EM radiation depends on its wavelength. Also, the wavelength is inversely related to the frequency. Higher frequencies have shorter wavelengths, and lower frequencies have longer wavelengths. Understanding the characteristics of EM radiation in terms of their wavelength and frequency is crucial to understanding the information to

be extracted from remote sensing data. The EM spectrum (figure 2.1) is the

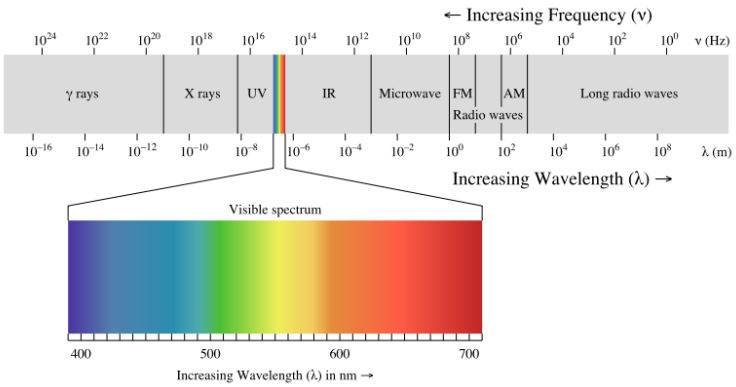


Figure 2.1: The electromagnetic spectrum.

range of all possible EM radiation wavelengths (or frequencies), from the shorter wavelengths (including gamma and x-rays) to the longer wavelengths (including microwaves and broadcast radio waves). The ultraviolet (UV), the visible, the infrared (IR) and the microwave portion of the EM spectrum are of particular interest in remote sensing.

Any body whose temperature is above absolute zero (0 K or -273degC) radiates EM energy. Therefore, terrestrial objects also radiate at several wavelengths. EM radiation with a wavelength between approximately 400nm and 700nm can be detected by the human eye and perceived as visible light. However, there is a lot of radiation around us which is *invisible* to our eyes, but can be detected by other remote sensing instruments and used to our advantage. Regarding the sections of the EM spectrum considered in remote sensing, there are certain objects that are easily detectable at specific wavelengths. For example, while rocks and minerals fluoresce or emit visible light when illuminated by UV radiation; the green vegetation has a higher response in the IR region. The possibility of gather EM information in a wider section of the EM spectrum than just the visible portion increases the potential of remote sensing, allowing

to better distinguish different targets.

There are two types of EM energy-based sensors: the passive and the active sensors. The passive sensors are characterised for capturing the energy that is naturally available. This means that it should exist an external energy source to illuminate the target (generally the sun), and for that reason, this type of sensors can only be used when that source is available. On the other hand, the active sensors are characterised for providing their own source of energy to illuminate the target of interest. This property allows the use of these sensors at any time of day or season. However, when compared with the passive sensors, the cost of using this sensors is very high. The type of sensors used in this work are passive sensors.

2.1.2 Atmosferic interactions

The primary source of energy that illuminates natural targets is the sun, but it is not the unique source. Independently of the source, once the EM radiation enters into and propagates through the earth′s atmosphere, the particles and gases in the atmosphere affect its properties including the speed and direction of propagation, the wavelength, the intensity, and the spectral distribution. The intensity and spectral composition of radiation available to any sensor are therefore affected. These effects are mainly caused by the mechanisms of *absorption* and *scattering* [74].

Absorption is the phenomena in which some gases that comprise our atmosphere absorb radiation in certain wavelengths. Ozone, carbon dioxide, and water vapour are the three main atmospheric constituents which absorb radiation. The EM spectrum areas where this phenomenon occurs are known as *absorption bands*. The wavelength ranges in which the atmosphere is particularly transmissive of energy are referred to as *atmospheric windows*. Figure 2.2 shows

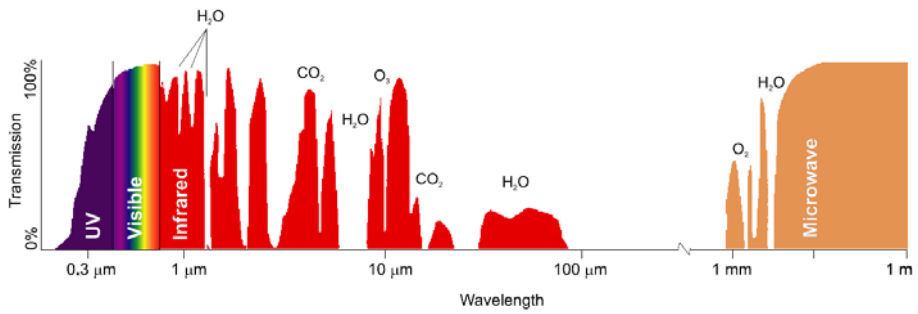


Figure 2.2: Atmospheric windows and absorption bands.

the percentage of light transmitted at various wavelengths, and the sources of atmospheric opacity are also given. Both passive and active remote sensing technologies do best are usually limited to operate within the atmospheric windows. Outside these windows, there is no radiation from the sun to detect because the atmosphere has blocked it.

Scattering is the other main mechanism at work when electromagnetic radiation interacts with the atmosphere. Scattering occurs when radiation is reflected or refracted by atmospheric particles. Examples of those particles are gas molecules, dust, smoke, pollen, cloud droplets, raindrops, etc. When propagating through the atmosphere, radiation may be unpredictably redirected (scattered) into various directions by these atmospheric particles. The size of atmospheric particles relative to wavelength of incident radiation affects the occurrence of different scattering types. The degree of scattering effect depends on several factors such as the geometric shape and abundance of the particles or gases, the wavelength of radiant energy, and the distance the radiant flux travels through the atmosphere. The scattering is usually divided into three categories:

- *Rayleigh scattering*: Rayleigh scattering occurs when the size of atmospheric particles are much smaller than the wavelength of the incident EM radiation. The amount of scattered energy by Rayleigh scattering

is inversely proportional to the fourth power of wavelength of radiation causing shorter wavelengths of radiation to be scattered much more than longer wavelengths. The blue sky and red sunset are typical examples of Rayleigh scattering. Rayleigh scattering is the dominant scattering mechanism in the upper atmosphere.

- *Mie scattering*: exists when the size of atmospheric particles such as smoke, haze, pollen and dust are comparable to the wavelength radiation. This scattering, compared to Rayleigh scattering, tends to affect longer wavelengths. Also, it mostly occurs in the lower portions of the atmosphere where larger particles are more abundant, and dominates when cloud conditions are overcast.

- *Non-selective scattering*: it occurs when the particles are much larger than the wavelength. The effect of nonselective scattering is approximately the same in all scattering directions and is almost independent of wavelength. This is why fog and clouds appear white.

Due to these atmospheric effects, the sensor's incoming information is largely contaminated and do not directly characterize the reflectance of surface objects. To use this information efficiently in remote sensing applications, these effects should be removed. The objective of *atmospheric correction* is to retrieve the surface reflectance (that characterizes the surface properties) from remotely sensed imagery by removing the atmospheric effects. Atmospheric correction algorithms basically consist of two major steps. First, the optical characteristics of the atmosphere are estimated either by using special features of the ground surface or by direct measurements of the atmospheric constituents or by using theoretical models. Various quantities related to the atmospheric correction can then be computed by the radiative transfer algorithms given the atmospheric optical properties. Second, the remotely sensed imagery can be corrected by

inversion procedures that derive the surface reflectance. Atmospheric correction has been shown to significantly improve not only the quality of the observed earth surface imaging but also the accuracy of classification of the ground objects.

2.1.3 Interaction with targets

The radiation that manages to pass through the atmosphere (is not absorbed or scattered) will reach and interact with objects/materials at the surface of Earth. Three fundamental interactions will when energy strikes, or is incident upon the surface: *absorption*, *transmission* and *reflection*. The total incident energy is the sum of these three interactions, the proportions of each will depend on the wavelength of the energy and the material and condition of the feature. Absorption occurs when, at a given wavelength, the EM energy incident on a given surface is absorbed and converted to other forms of energy. Transmission is the process by which the incident EM energy on a surface propagates trough that surface. Reflection occurs when EM energy is moves away from the target at specific angles and/or scatters away from the target at various angles, depending on the surface roughness and the angle of incidence of the rays.

Remote sensing instruments are mostly devoted to measuring and registering the radiance reflected by the targets or areas of interest. There are two extreme types of reflecting surfaces that interact with EM radiation: *specular* (smooth) and *diffuse* (rough). Reflection is said to be specular when radiation is reflected according to Snell's Law which states that the angle of incidence is equal to the angle of reflectance, i.e., show a mirror like behaviour. Reflection is said to be diffuse when the surface is rough and the energy is reflected almost uniformly in all directions. Figure 2.3 illustrates the geometric characterization of ideal and near-perfect specular and diffuse reflectors. In general, natural

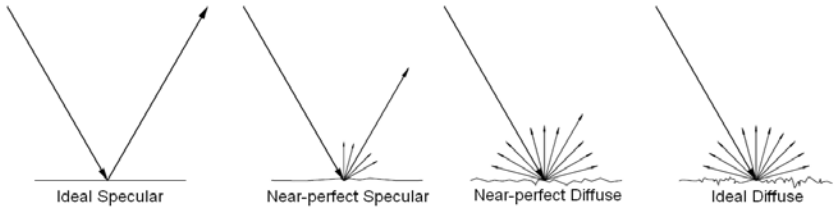


Figure 2.3: Specular vs. diffuse reflectors (adapted from [74]).

surfaces are almost always diffuse and depart significantly from specular at shorter wavelengths (into the infrared) and may still be somewhat diffuse in the microwave region.

As we saw, there are several aspects that affect the radiance of a target registered by a sensor. Depending on the chemical composition and physical characteristics of the target of interest, and the wavelengths of radiation involved, we can observe very different responses to the mechanisms of absorption, transmission, and reflection. For any given material, the amount of radiation that it reflects, absorbs, transmits, or emits varies with wavelength. Plotting the reflectance as a function of wavelength, results in a spectral reflectance curve. The spectral signature of a given material uniquely identifies that material, accordingly with the measured reflectance at varying wavelengths. Figure 2.4 shows an example of spectral signatures for three materials present in earth surface: healthy vegetation, dry bare soil and clear lake water. This important property of matter gives us a powerful tool to discriminate between different materials present in the earth surface. The variability of the spectral signatures between different targets allow us to identify the type and/or condition of those targets. Although a spectral response for the same type of target may be quite variable (depending for example on time and/or location recorded), spectral signatures are a suitable choice to identify the land cover type in remotely sensed imagery. The spectral signatures may be successfully used as input vectors to a pattern recognition system.

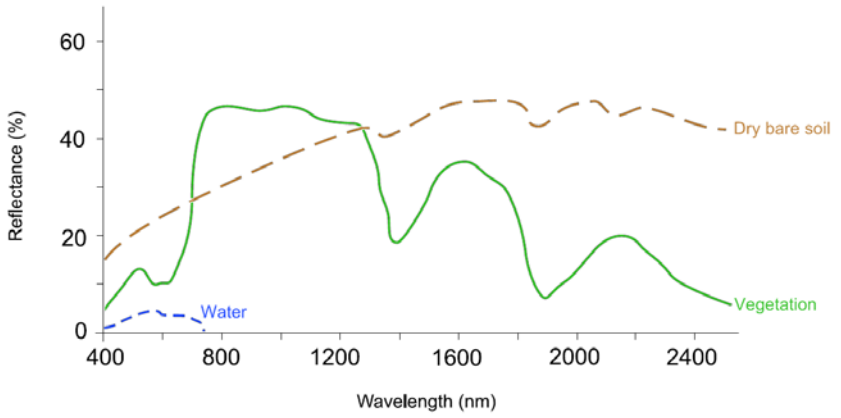


Figure 2.4: Typical reflectance curves for vegetation, soil, and water.

2.2 Multispectral and Hyperspectral sensors

In the beginning of this chapter we identified the goal of remote sensing as the acquisition of information about the earth surface. That information can be retrieved trough the EM response of the different materials present at the earth surface. In previous sections we have been discussing the principal sources of EM radiation, and the interaction of this radiation with both the atmosphere and ground targets. We will now focus on how this EM radiation is acquired and registered.

To the devices that capture the EM energy from the objects at the earth surface, we call sensors. The energy may be detected either photographlly or electronically. While the process of photography detects and records the energy variations using a light-sensitive film; electronic sensors generate an electrical signal that corresponds to the energy variations in the original scene [74]. We will consider the last type of sensors, electronic ones. The process of generating the electrical signal is initialized when the radiative flux of a given point over a target's surface strikes a photosensitive device in the sensor. Composed by a

number of lines or arrays of photodiodes[1], this device is sensitive to EM radiation within a discrete portion of the EM spectrum in each of the arrangements of photodiodes, each of them is termed *spectral channel* or *spectral band*. The amount of electric charge converted by each photiode is directly proportional to the incident radiative flux, and is then quantized to a discrete number, the *digital value*.

Typically, each digital value is stored in a two-dimensional array of discrete *picture elements*, or *pixels*, forming an *image channel*. To each of the bands of differing wavelengths, there is a corresponding image channel. The set of these image channels form a *multispectral image*. Figure 2.5 shows a collection of seven image channels.

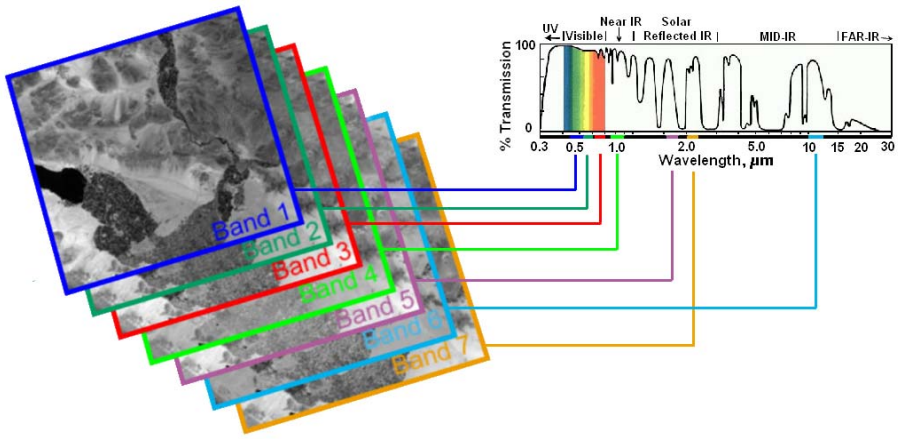


Figure 2.5: A multispectral image as a collection of image channels.

Resolution allows to measure of the ability of an optical system to distinguish between signals that are spatially near or spectrally similar [61]. There are four major resolutions associated with a remote sensing system: temporal, spatial, radiometric and spectral resolution. These resolutions are also used

[1]A photodiode is a type of photodetector capable of converting light into either current or voltage, depending upon the mode of operation.

to characterize a remote sensor and the correspondent imagery.

Temporal resolution is defined as the frequency at which images are registered in a particular area at the same viewing angle. The more frequently it is captured, the better or finer the temporal resolution is said to be. In the case of on board satellite sensors, the temporal resolution is related to the time it takes to the satellite to complete one entire orbit cycle.

The instantaneous field of view (IFOV) of a sensor primarily defines the spatial resolution of a sensor. The IFOV is the angular cone of visibility of the sensor and determines the area on the Earth's surface which is seen from a given altitude at one particular moment in time. For practical purposes, the spatial resolution gives the ground area that is represented by each pixel in a remotely sensed image. Therefore, a higher the spatial resolution gives more detailed information about the ground cover. In the case of fixed-distance platforms (the IFOV is fixed), such as EO satellites, the spatial resolution is simply the dimension of the ground-projected IFOV. When this is not the case, the size of the area viewed may be determined by multiplying the IFOV by the distance from the ground to the sensor.

Radiometric resolution is defined by the sensor sensitivity to the magnitude of the EM energy reflected or emitted by the target. Radiometric resolution is usually expressed as a number of levels or a number of bits, for example 8 bits or 256 levels. The finer the radiometric resolution of a sensor, the better subtle differences of intensity or reflectivity can be represented.

The spectral resolution of a sensor is defined by the number and dimension of specific wavelength intervals in the EM spectrum to which a remote sensing instrument is sensitive. There are certain regions of the EM that are optimum for obtaining information on biophysical parameters (see figure 2.2). Careful selection of the spectral bands may therefore improve the EM energy quality

received by the sensor.

Regarding the spectral resolution, there are two major types of sensor: the multispectral and the hyperspectral sensors. Multispectral sensors record radiant energy in few bands of the EM spectrum. These spectral bands are usually non-contiguous and broad, reflecting a lower spectral resolution. The hyperspectral sensors acquire images in many, very narrow, contiguous spectral bands. While the number of spectral bands of a multispectral sensor is on the order of tens, the hyperspectral sensors have hundreds of spectral bands.

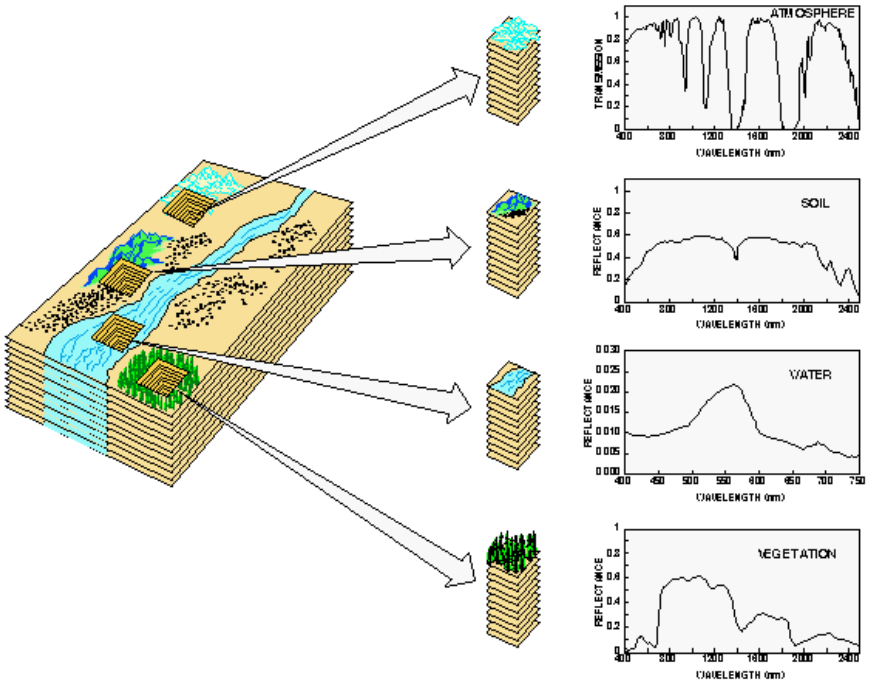


Figure 2.6: The imaging spectrometry concept.

Figure 2.6 shows the concept of hyperspectral imaging (or imaging spectrometry): the hundreds of adjacent narrow spectral bands collect within a given spectral range, enables the construction of an almost continuous reflectance spectrum for every pixel in the scene allowing the identification of the surface

materials. It is known however that in practice, a pixel's signature generally includes a mixture of more than one type of material. To deal with this problem, spectral unmixing algorithms are applied. This subject will be referred in more detail in section 3.4.

The stable platform in which the sensor resides to collect and record the reflected energy from a target surface is usually situated on an aircraft or on a spacecraft (or satellite). Depending on the location, within or outside the Earth's atmosphere, the sensors are known as airborne or spaceborne sensors, respectively. The main difference between these two type of sensors has to do with the altitude at which the data is collected and the field of view (FOV) of each sensor. These two aspects are intimately related and highly influence the EM energy detected by the sensors inducing both spectral and geometrical effects in the images.

Concerning the altitude of each sensor, the fact that airborne sensors are able to fly at lower altitudes than spaceborne sensors, allow them to provide higher spatial resolution images. Another significant difference between these two type of sensors has to do with the FOV[2]. Since the satellite altitude is higher, the FOV of spaceborne sensors is much less wider than the airborne sensors (around 45deg off nadir in this case, and around 5deg in the former one). This wide FOV of airborne sensors will conduct to a geometric distortion in pixel size from nadir to maximum viewing angle.

In the case of satellite imagery, the fact of imaging through the entire atmosphere leads to atmospheric effects not detected in the airborne sensors. On the other hand, airborne sensors with wide FOV optics have problems related to imaging trough widely varying slant paths through the atmosphere while narrow FOV satellite sensors have not. Also, significant changes in viewing geometry across

[2]Note that while IFOV has to do with a single photodiode, FOV is the angle through which EM radiation is detected by the sensor

the extent of an aircraft image may result in changes in spectral signatures due the variations in the BRDF of the terrain, while such effects are less significative in satellite imagery. In exchange, the true atmospheric characteristics generally do not vary across the limited extent of the aircraft imagery, while dramatic changes in haze and visibility often occur in satellite imagery.

Geometric distortions are also added to airborne imagery due to the platform motion. The variations in flight conditions, lead to changes in aircraft position (roll, pitch and yaw) that cause non-uniform ground sampling.

As seen, concerning to the *quality* of the images, both types of sensors have advantages and disadvantages. In either case the imagery should be geometrically and atmospherically corrected with adequate postprocessing methods.

With respect to the acquisition of the images of a specific area, with given temporal and seasonal characteristics, it is easier to achieve that with airborne sensors due the higher flexibility of scheduling a flight with those requirements. Another positive aspect of airborne sensors has to do with their general capacity of providing a larger amount of data.

In this thesis images from two airborne hyperspectral sensors were considered: from the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) and from the Reflective Optics System Imaging Spectrometer (ROSIS).

The AVIRIS, which was first flown in 1987, was one of the first airborne systems with imaging spectroscopy and was produced by NASA's Jet Propulsion Lab (JPL). This sensor was developed to increase the spectral and spatial coverage of the first airborne imaging spectrometer (AIS) created in 1982 by the same laboratory.

AVIRIS typically flown onboard the NASA/ARC aircraft, however, there are more three aircraft platforms also used (Twin Otter International's turboprop,

Scaled Composites' Proteus, and NASA's WB-57). The image used in this work was acquired with a ER-2 flight. The flight altitude defines the data pixel size and swath width. Considering the ER-2, it usually flies at approximately 20 km above sea level, has a 30deg FOV and a IFOV of 1.0mrad, producing images where each pixel cover an area of approximately 20m diameter on the ground. The AVIRIS instrument uses a *whiskbroom* scanning mirror (sweeps back and forth), producing 677 pixels in each scan thus yielding a ground swath about 11km wide. An AVIRIS *scene* is a set of 512 lines of data, and corresponds to an area about 10km long on the ground.

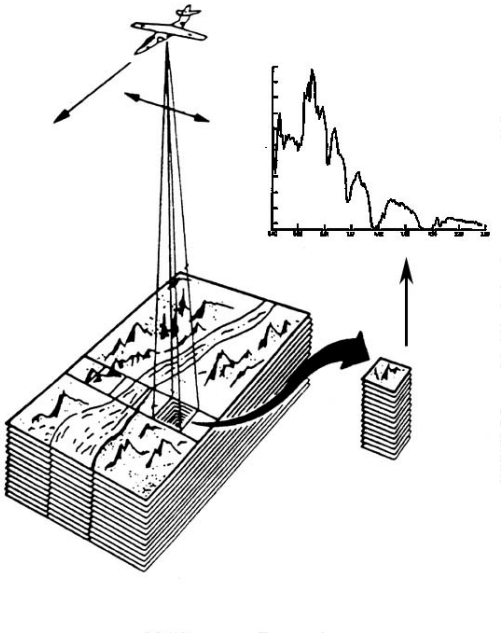


Figure 2.7: Conceptual representation of AVIRIS data acquisition.

This instrument manages to cover the entire spectrum range between 380nm and 2500nm, by using 224 different detectors with a spectral bandwidth (wavelength sensitive range) of approximately 10nm. Therefore, each pixel contains information of each of the 224 spectral bands. When the data from each detector is plotted on a graph, it yields a complete VIS-NIR-SWIR spectrum [51, 53].

The data are recorded in 12bits (values from 0 to 4095). Figure 2.7 presents a conceptual representation of AVIRIS data acquisition.

The ROSIS sensor was developed prior to 1992 by Dornier Satellite Systems (DSS formerly MBB) in cooperation with the GKSS Research Centre (Institute of Hydrophysics) and with the DLR [49]. The detection of fine spectral structures in coastal and inland waters was the original goal of this sensor. This determined the selection of the spectral range, the bandwidth, the number of channels, the radiometric resolution and the possibility of off-nadir pointing to avoid sun glint. Nevertheless, this sensor had been used for monitoring spectral features ashore or within the atmosphere. An example is it inclusion in the Hysens project.

The DLR Hysens project included two hyperspectral sensors (DAIS and ROSIS) and hyperspectral processing facilities with the aim of introduce airborne hyperspectral imaging techniques and optical calibration techniques to environmental scientists. This project was active between 2000 and 2002 [84].

ROSIS makes use of a two-dimensional CCD array. The first dimension is used to scan a narrow cross track line on the ground, the second one is for acquiring the spectral information of each scanned pixel. This gives the possibility for imaging simultaneously 115 spectral bands of 512 picture elements.

Some data characteristics depend on the flight altitude. Within the HySens project, a FOV of 16deg and a IFOV of 0.56mrad were defined with a pushbroom scan principle and 512 pixels per line. This resulted in a ground resolution from 1m to 6m. The data was recorded in 14bit. The spectral area covered by the ROSIS is between 430 and 860nm, with a spectral sampling of 4nm. The preprocessing of the ROSIS data, including system corrections and radiometric calibration, was carried out by the DLR in the scope of the HySens project.

The main characteristics of the airborne AVIRIS and ROSIS hyperspectral

remote sensing systems are summarized in table 2.1.

Table 2.1: Characteristics of the airborne AVIRIS and ROSIS hyperspectral remote sensing systems

	AVIRIS	ROSIS
Technology	Whiskbroom linear array	Pushbroom area array CCD
Spectral Resolution (nm)	400-2500	430-860
Spectral Interval (nm)	10	4
Data Collection Mode	224 bands	115 bands
Dynamic Range (bits)	12	14
IFOV (mrad)	1.0	0.56
FOV(deg)	30	16

Chapter 3

Data Classification

This chapter is devoted to introducing basic data classification concepts and notation used throughout the thesis. Initially the general concepts of classification are presented, both in supervised and unsupervised approaches and the validation techniques. Then the problem of image classification is addressed, and more specifically directioned to multi and hyper-spectral images. The chapter is finalized with the introduction of spatial context in the classification processes.

3.1 General Concepts

In every day task, we automatically identify objects, persons, sounds, smells, and every thing that surround us almost without thinking about it. In some situations we make a greater effort to do that identification by associating characteristics observed or felt to something that we had already seen or felt before, or to something that someone already told us about. This ability is taken for granted because we have it since we are born. Also, we are continuously learning and improving this capacity. This process can be called as classification or pattern recognition. This mechanism of classification is assumed to be easy

for us, until we face the task of teaching a machine to do the same.

Jain et. *al* [59] define pattern recognition as *the study of how machines can observe the environment, learn to distinguish patterns of interest from their background, and make sound and reasonable decisions about the categories of the patterns.*

Automatic recognition is widely used in a large range of applications like biology, psychology, medicine, marketing, computer vision, artificial intelligence and remote sensing. In all these areas there are constantly problems related to the identification of grouping patterns, classification of individuals/observations, and the automatic recognition is a powerful tool to help and make decisions related with these problems. Automatic recognition often produces faster decisions with no need for human intervention, for example in the identification of number plates in a parking lot, fingerprints, post distribution machines, etc.

Analogously to what happens with human decision process, the automatic recognition is based on the object/individual characteristics observed. For example, if we want to identifying our umbrella left among others, we start by identify the colour, size and shape. An appropriate functional dependency or relationship between those variables lead us to a decision (is it mine or not?). The evaluated characteristics are called features or variables, and in the statistical approach they can be represented in terms of a d-dimensional vector where the position j, $j \in \{1, ..., d\}$, measures the response of the observed object/individual in a given characteristic. Let us then represent real world objects as a feature vector $x_i = [x_{i1}, \ldots, x_{id}]$ where each component is associated with each one of the d features. This object representation is known as input vector, also called observation, individual, predictor or independent variables. The goal of automatic recognition is to predict the group to each individuals belong, given the observed features.

It is desirable that these groups form compact and disjoint regions of points in the d-dimensional feature space. The set of objects to be grouped will be represented as $\mathbf{x} = [x_1, \ldots, x_n]$, where each x_j, $i = 1, \ldots, n$ presents a single input vector.

3.1.1 Supervised and unsupervised approaches

Statistical classification consists in assigning a label to each object present in $\mathbf{x}$, given their observed features. Classification is said to be **supervised** if there is a set of objects which have known labels. These samples can be represented as the set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, were y_i is the label (or class) identified with object x_i. In this case, the classifier is generated based in the information given by these paired samples, called training samples.
Supervised learning may be seen as a learning system where there is a *teacher* that provides a category label for each pattern in a training set, and seeks to reduce the sum of the costs of these patterns [38]. Under these circumstances, the cost is nothing less than the *price to pay* for assigning a wrong label to an object. However, there are problems for which training data, of known class labels, are not available. When this happens (no knowledge about the object labels), we have **unsupervised** classification or clustering. In this type of problem, we are given a set of feature vectors x and the goal is to unravel the underlying similarities, and group similar vectors together. Using the *teacher* metaphor, in the case of unsupervised learning there is no explicit *teacher*, and the system forms clusters or *natural groupings* of the input patterns [38]. To measure the similarities between groups and/or observations several similarities measures have been developed. The most commonly used are simple distance measures such as Euclidean distance and L1 distance.

These two types of classification methods comprise a vast number of algorithms.

In fact, each of these two classification areas may be divided in other groups of classification algorithms, this subdivision may vary from author to author.
In the case of unsupervised learning, algorithms generally are based on the following two popular clustering techniques: iterative square-error partitional clustering and hierarchical clustering [59].
Hierarchical algorithms rely on ideas of matrix and graph theory to produce either increasing (divisive algorithms) or decreasing (agglomerative algorithms) number of clusters each step, thus producing a hierarchy of clusters. Each level of the hierarchy represents a particular grouping of the data into disjoint clusters of observations. This nested sequence of groups is represented by a dendogram or a tree. Square error partitional algorithms attempt to obtain that partition which minimizes the within-cluster scatter or maximizes the between-cluster scatter. The optimum partition is computed iteratively using differential calculus concept. Well-known algorithms of this type are the k-means [75] and Fuzzy K-means [40]. Both in hierarchical and partitional algorithms it is the user that should define the number of classes K, which sometimes may not be an easy task, mainly because every clustering algorithms will find clusters in a given dataset whether they exist or not. To better deal with this problem, there are some methods to automatically select the number of classes present in the image [77]. Another challenging problem in clustering algorithms is the selection of the appropriate measure of similarity to define clusters which, in general, is both data (cluster shape) and context dependent.

Supervised classification methods are extensively used in pattern recognition. This type of methods can be categorised according to several criteria. Some authors [103] suggest three main groups: Bayesian classifiers, linear classifiers and non-linear classifiers.

3.1.2 Bayesian classifiers

Bayesian decision theory is the basis of Bayesian classifiers. To briefly review this theory, let us assume that there are K possible classes, and consequently each pattern y_i should take one of these values $k \in \{1, \ldots, K\}$. The probability model for a classifier is given by a conditional model $P(y_i = k|x_i)$, $k \in \{1, \ldots, K\}$ and $i = 1, \ldots, N$. These conditional probabilities represent the likelihood of a given object x_i belong to each of the K classes involved. The Bayes classifier gives us a reasonable solution which says that we classify x_i to the most probable class, using the conditional distribution. The optimal Bayes decision rule can be stated as: assign the input pattern x_i to class k if

$$p(y_i = k|x_i) > p(y_i = l|x_i),\ \forall\ k \neq l,\ k,\ l \in \{1, \ldots, K\}\,. \tag{3.1}$$

The problem is then how to estimate these *a posteriori probabilities* $p(\mathbf{y}|\mathbf{x})$.
Bayes decision theory assumes that the *a priori probabilities* (or simply *prior*) $p(\mathbf{y})$ and the class-conditional probability density $p(\mathbf{x}|\mathbf{y})$ are known. The prior probabilities reflect our prior knowlodge of how likely we are to get a given class. The class-conditional probability density function, gives us the *likelihood* of $\mathbf{y}$ with respect to $\mathbf{x}$ for which $p(\mathbf{x}|\mathbf{y})$ is large is more "'likely"' to be the true class. When these terms are not known, what is done is estimate them from the training data.
Using Bayes formula it is possible to convert the prior probability to the *a posteriori* probability by observing the value of $\mathbf{x}$:

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}, \tag{3.2}$$

where $p(\mathbf{x})$ is the probability density function describing the training data, irrespective of its class. This term can be viewed as merely a scale factor that guarantees that the posterior probabilities sum to one. This formula can be expressed informally in English by saying that

$$posterior\ =\ \frac{likelihood\ \times\ prior}{evidence}. \tag{3.3}$$

Based in the rule exposed in 3.1, the labelling generated by this classifier is then given by:

$$\mathbf{y} = argmax_{y_i=1,\dots,K} p(\mathbf{y_i}|\mathbf{x}) = argmax_{y_i=1,\dots,K} \left\{ p(\mathbf{x}|\mathbf{y_i}) p(\mathbf{y_i}) \right\}. \tag{3.4}$$

This type of classifiers family is also known as the Maximum Likelihood (ML) classifier, which is one of the most widely used in pattern recognition, and that will be later used in this thesis.

3.1.3 Linear and non-linear classifiers

Linear classifiers and non-linear classifiers are designed irrespective of any assumptions on the distribution describing the training data [103]. Linear classifiers use linear discriminant functions, which can be interpreted as hyperplanes that divide the pattern space in two partitions: all points on one side of the hyperplane are classified as *yes*, while the others are classified as *no*. Due to this dichotomy, linear classifiers are often designed for a two-class problems. However, there are methods to extend this type of methods for a multi-class problem: *one-vs-all*, error-correcting code and single-machine approaches (see [94] for a review and discussion of multiclass classification using binary classifiers).

Examples of well-known linear algorithms include Least Square methods, the Percepton algorithm, Linear Discriminant Analysis (LDA) (or Fisher's linear discriminant), Logistic Regression (LR) and Support Vector Machines (SVM).

Non-linear classifiers emerge as a necessity when dealing with problems that are not linearly separable. One of the most popular non-linear algorithm is the K-Nearest Neighbour classifier (K-NN) [47]. In addition, the linear classifier algorithms listed above can be converted into non-linear algorithms operating on a different input space, using the *kernel trick*.

The *kernel trick* consists in mapping the data into a high dimensional feature

space, where each coordinate corresponds to one feature of the data items, transforming the data into a set of points in an Euclidean space. In that space, a variety of methods can be used to find relations in the data. These methods are known as kernel methods [32] and have received a great deal of attention in the past few years, mainly due to their capacity for solving problems involving the classification and analysis of high-dimensional or complex data.

3.1.4 Generative and discriminative classifiers

Besides the supervised/unsupervised dichotomy, in statistical pattern recognition another dichotomy may be considered depending on the kind of information available about the class-conditional densities. If the form of these densities is known (e.g, multivariate Gaussian), but some parameters of the densities are unknown, we have a parametric decision problem. On contrary, if the form of class-conditional densities is unknown, then we operate in a nonparametric mode [59].
In the case of supervised parametric learning of classifiers, the determination of parameters lead us to two broad classes of methods: the generative (informative) and the discriminative models [96].

Generative models learn the conditional density functions $p(\mathbf{x}|\mathbf{y})$ separately from the training data and make their predictions using the Bayes rules to calculate $p(\mathbf{y}|\mathbf{x})$. Examples of such algorithms include LDA and Naive Bayes classifier.

Discriminative models learn the posterior $p(\mathbf{y}|\mathbf{x})$ directly from the data, this means that the class-conditional densities are not explicit modelled. This property is one of the several compelling reasons for using discriminative rather than generative models, as succinctly articulated by Vapnik [110]: *one should solve the classification problem directly and never solve a more general problem as an intermediate step.* This class of models include linear and logistic discrimination,

K-NN, tree classifiers, feedforward neural networks, SVM and other kernel methods.

It should be mentioned that the work developed along this thesis uses the logistic regression model. As mentioned before, logistic regression learns $p(\mathbf{y}|\mathbf{x})$ directly. It starts by assuming a parametric form for the distribution $p(\mathbf{y}|\mathbf{x})$, then directly estimates its parameters from the training data. The form of the distribution is conveniently chosen to lead to simple linear expressions for classification. To classify any pattern, the rule 3.1 is applied. A common method to estimate the model parameters is by maximum likelihood estimation. The maximization of the log-likelihood function is generally achieved using Newton-Raphson algorithm [55]. More details about the multinomial logistic regression method for classification will be addressed in section 4.1.1.

3.1.5 The pattern recognition process

A pattern recognition system can be viewed as a cycle that includes several processing steps. A popular system scheme is the one presented in [38] which is composed of the following activities:

- **Data Collection**: it should be sufficiently enough to assure good performance in the fielded system. Note that independently of the classification or decision rule used, the performance of it depends on both the number of available samples as well as the values of the samples.
- **Feature Choice**: a critical step and depends on the characteristics of the problem domain. In this step, prior knowledge should be incorporated.
- **Model Choice**: there is a great number of learning algorithms that the user should choose in this step. These algorithms can be grouped in different class model as mentioned previously.

- **Training**: is the process of using available data to build the classifier, also referred to as learning the classifier. In this step, the algorithm chosen in the previous step should be trained using the available training samples.

- **Evaluation**: the performance of the system is measured and the need for improvements is analysed.

We have exposed the main class models to perform the learning part of a pattern recognition system. But it should be highlighted the extreme importance of the last step in the pattern recognition system. Recall that the goal of designing a recognition system is to classify future test samples which are likely to differ from the training samples used to build the classifier. Therefore, building an excessive complex system that perfectly predict the class of training samples is unlikely to perform well on new patterns. This situation is known as overfitting. Instead, the system should be capable of producing good predictions in test patterns which were not used during the training stage. This ability is referred as the generalization performance of a learning method. The assessment of this performance is of great importance in practice since it guides the choice of learning method or model, and gives us a measure of the quality of the ultimately chosen model [55].

Poor generalization ability may be caused for different reasons. Some of them may be: (i) the small number of samples relative to the number of features (known as *curse of dimensionality* to which we will refer later); (ii) the classifier is too intensively optimized on the training set (overfitting); (iii) the number of unknown parameters associated with the classifier is too large [59].

To assess the performance of a learning method one should have in mind that there are two separate goals: the model selection and the model assessment. The first goal is related with the estimation of the performance of different models in order to choose the (approximate) best one. The latter one is related to the

estimation of the prediction error (generalization error) on new data, when using the final model.

The accomplishment of these two goals should not be a problem if there is enough data to perform the train, the validation and the test tasks. In this case, the dataset should be divided in three parts: a training set, a validation set and a test set. The first one is used to fit the models; the second one to estimate the prediction error for model selection and the last one to assess the generalization error of the final chosen model. The problem is that in most practical problems obtaining enough data to execute these three steps is not always possible. In addition it is difficult to give a general rule on how much training data is enough and how to choose the number of patterns in each of the three parts.

To overcome the problem of insufficient data to split it into three parts, there are methods that approximate the validation step either (i) analytically or (ii) by efficient sample re-use. Examples of analytical methods are the Akaike Information Criterion (AIC) [1], the Bayesian Information Criterion (BIC) [98], Minimum Description Length (MDL) [95] and Vapnik's Structural Risk Minimization (SRM) [110]. These methods estimate the prediction error estimating the *optimism*[1] and add it to the to the training error rate. However, these methods only work for a special class of estimates that are linear in their parameters. The second type of methods, which include the well-known cross-validation [102] and bootstrap methods [41], are direct estimates of the extra sample error and can be used with any loss function, and with nonlinear, adaptative fitting techniques [55]. In this work, cross-validation techniques were used in the validation step.

[1] *Optimism* is defined as the expected difference between the *in-sample* error and the training error [55].

3.2 Image Classification

The previous section presented the basics of pattern recognition, in particular the classification of objects or patterns. As mentioned, in the statistical approach an object to be classified is represented as a d-dimensional vector, where each component represents a characteristic of that object. This object, individual, or observation may be an endless spectrum of things. As an individual to be classified we may think of a real object (a car, a chair, a clothing piece, etc), a sound, words in a text, a person, a face, an image, etc. In this section we will focus on the classification of specific objects, namely the pixels that compose an image.

A digital image is a representation of a two dimensional image using numerical values. A digital image is composed by a finite set of elements called pixels. A pixel is a contraction of the term PIcture ELement. Digital images are made up of small squares (the pixels), just like a tile mosaic. Pixels are very small and so when the image is displayed on a computer monitor we do not normally see the individual pixels. The digital image looks smooth and continuous just like a regular photograph but it is actually composed of thousands or millions of tiny squares as shown in figure 3.1.

To form an image, the pixels are disposed in a regular grid of a fixed number of rows and columns. Each pixel holds a value that represents the brightness of a given colour at any specific point of the image. Those points are spatially identified by a pair of spatial plane coordinates. Each pixel can be referenced via its x and y coordinates. In digital images the point of reference is usually taken as the upper left corner, so that the x coordinate increases top-to-bottom and the y coordinate increases left-to-right. Using this idea, the digital images

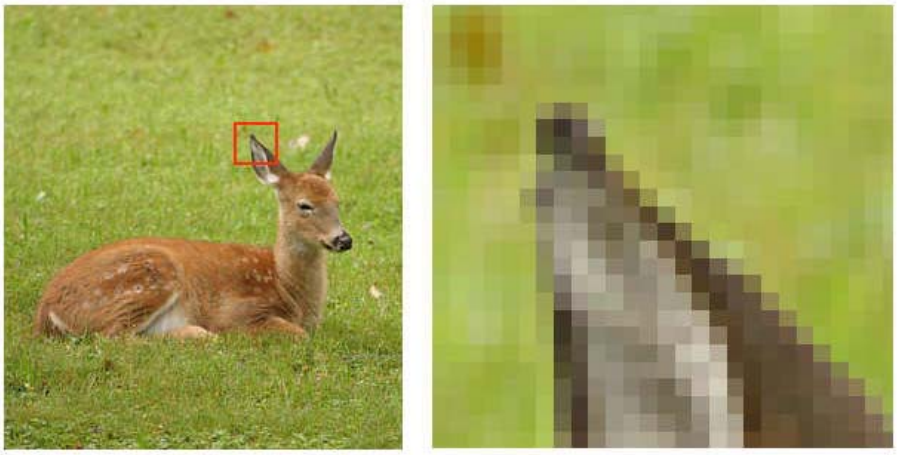


Figure 3.1: On the left the full image, on the right the area in the red square magnified to show individual pixels.

can be handled as a matrix:

$$\begin{bmatrix} (0,0) & (0,1) & \dots & (0,c-1) \\ (1,0) & (1,1) & \dots & (1,c-1) \\ \vdots & \vdots & & \vdots \\ (l-1,0) & (l-1,1) & \dots & (l-1,c-1) \end{bmatrix},$$

where c is the number of columns, l the number of lines of the digital image.

The pixels values may not be a single value, depending on the number and nature of those values, digital images may be classified as:
(i) binary images: each pixel has only two possible values and is stored as a single bit (0 or 1);
(ii) greyscale images: each pixel carries a single value that expresses the intensity information. This value ranges between a minimum and a maximum, where the former is total absence (black) and the maximum is total presence (white), with any fractional values in between;
(iii) colour images: these images are characterized for the fact that each pixel has more than one value. Usually the pixels that compose this type of image

are represented by a vector. A RGB (Red Green Blue) image is an example with three dimensions, while multi-spectral and hyper-spectral images are other examples that will be detailed forward.

Figure 3.2 shows a scheme of how a multi-band image is formed. In this example, each image pixel is represented by a four dimension vector.

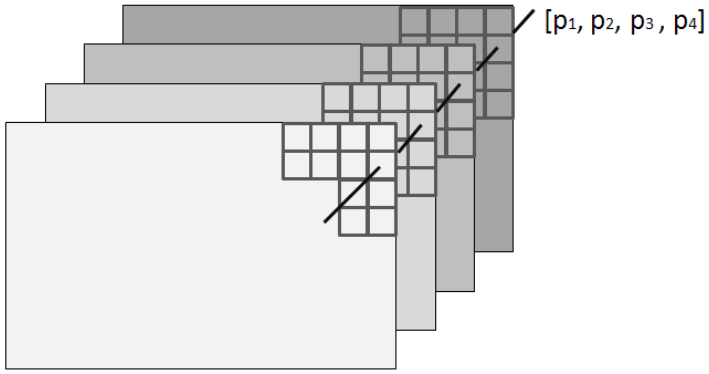


Figure 3.2: Schematic showing how a pixel of multi-band image is formed from the corresponding pixel values of its four components.

Image processing techniques have been developed to give solution to a wide range of practical applications, such as medical imaging, face recognition, fingerprint recognition, machine vision and remote sensing.

Image classification is one of many digital image processing techniques. The intent of the classification process is to categorize all pixels in a digital image into one of several classes (in the case of supervised learning), or create groups of pixels based on natural groupings present in the image values (in the case of unsupervised classification). This categorized data will result in a labelled image which may then be used to identify image sections of interest and, depending on the application, may be used to diagnosis, measure volumes or areas, locate objects, etc.

Recalling the concepts and methods presented in section 3.1, each image pixel

will be consider as the input vector x_i to the classification system, and the response y_i will be the class attributed to pixel i. The set of all pixels that compose the digital image will be denoted as $\mathbf{x}$, and $\mathbf{y}$ will represent the product of the classification, the labelled image. Note that $\mathbf{x}$ is an array of dimension $l \times c \times d$ where l the number of lines, c is the number of columns and d the components of the image, and $\mathbf{y}$ is an array of $l \times c$.

This thesis is dedicated to image classification methods for remote sensing digital images, namely, to images produced by multi and hyperspectral sensors. From now on, the thesis will focus in this type of images.

Remotely sensed image pixels contain the spectral information of different material present in the image. The dimension of the input vector x_i to the classification algorithm will depend on the type of sensor that collected the image. Hyperspectral sensors produce input vectors with higher dimension than multispectral ones (a pixel from a multispectral image has dimension of the order of tens while the hyperspectral image pixels have dimension in the order of hundreds). This characteristic of hyperspectral images has the advantage that more detailed information about the material present in a pixel is available, but on the other side, this will produce high dimensional datasets which will difficult the classification process. This subject will be analysed in detail in section 3.4.

The final remotely sensed image classification result is a labelled image where each pixel represents a land cover class or type. This product is frequently used to produce thematic maps or land cover maps.
In principle, any classification algorithm should be capable of producing such a result, this may not be possible if for example the classification algorithms chosen is not capable of dealing with large datasets. Nevertheless, if traditional classification algorithms are applied to the image pixels, what frequently happens is that the final labelled image do not present an homogeneous aspect of

the land cover distribution: the image classified has a *pixelised* aspect. This happens because the algorithm only has into account the spectral information of each individual pixel. When we look to an image and try to classify some objects there, we subconsciously know that adjacent pixels are more likely to have the same class. What we are doing is to include spatial information in our own classification process. This is not considered in the traditional pixel-based classification. In order to introduce the information of neighbouring pixel in the classification process, some methods have been developed recently. This information is very important but until a few years ago it was rarely considered in the remote sensing field. This matter will be addressed latter in this chapter.

3.3 Multispectral Remote Sensing Image Classification

In a multispectral image, a pixel represents the spectral signature (measured in a limited set of bands of the electromagnetic spectrum) of the materials present in the area captured by that pixel. Figure 3.3 shows the response of a pixel in a given spectral band. The dimension of the input vector will be then equal to the number of selected spectral bands of the sensor that captured the image. Some bands may be discarded by the user if for instance they are noisy bands, or have redundant information. These will be the input features used in the classification algorithm. The final product will be a labelled image where each pixel represents a single land cover class type. Figure 3.4 presents an example of a thematic map produced by the classification of a multispectral image of an agricultural area. The production of thematic-maps has a wide range of applications such as geological, agricultural, forestry, ecology, urban and land use applications.

Throughout the years, different types of classification algorithms have been

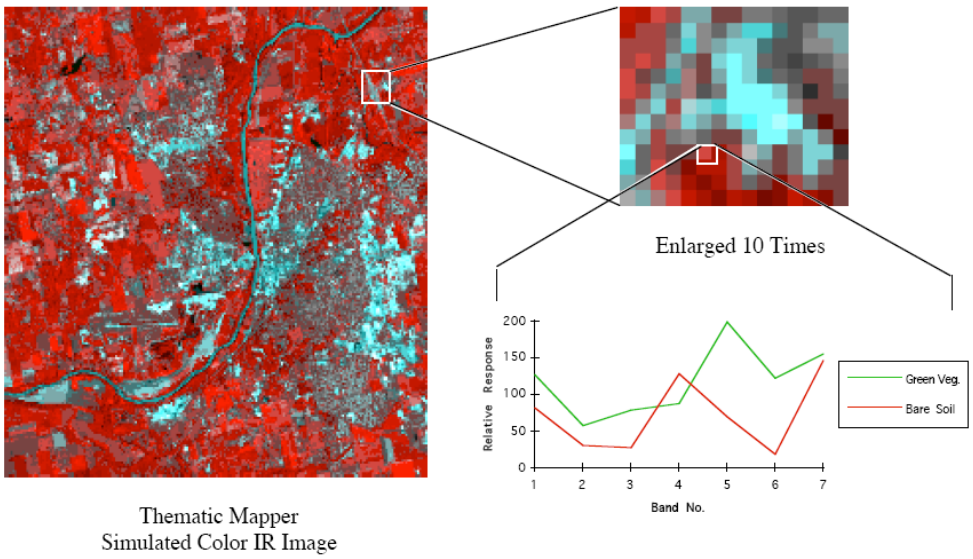


Figure 3.3: The multispectral concept: the signal from a pixel expressed as a graph of response vs. spectral band (figure from [69]).

applied to multispectral images, both with supervised and unsupervised approaches. However, the most commonly used is the supervised approach. Recall that in supervised algorithms the classes are defined *a priori* by the user, and then, each image pixel is associated with one of those classes. In unsupervised classification there is no prior information about the land cover classes searched for, and the image pixels are grouped accordingly with their spectral similarity. This will produce clusters that only roughly match some of the actual land cover classes.

Classification algorithms like the Maximum Likelihood, Minimum Distance, Parallelepiped, K-NN and LDA are among the most widely used algorithms in supervised classification. Their popularity is mainly due their simplicity and because almost every image processing software has them available. These algorithms are, being used as a reference to evaluate other classification methods, as reported for example in [11, 2, 97, 23, 118, 12]. Unsupervised algorithms

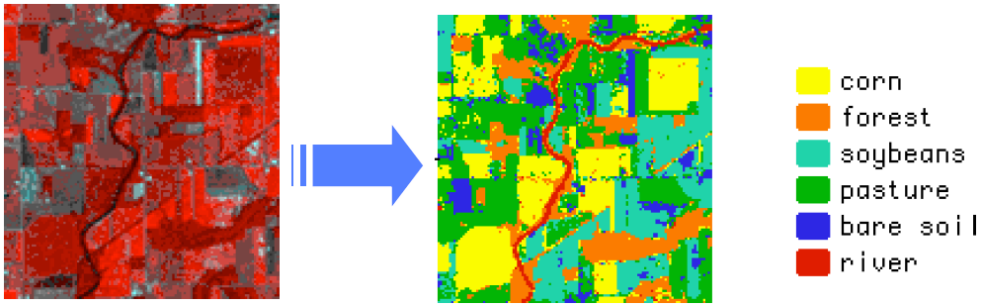


Figure 3.4: A thematic map of an agricultural area created from Thematic Mapper multispectral data (figure from [69]).

frequently used are the K-Means, ISODATA, and hierarchical clustering. In [39] some examples of unsupervised classification methods applied to multi-spectral images are presented. Neural Networks methods are also widely used among the remote sensing community [11, 2, 99] and new developments based on the traditional NN and directioned to the classification of remotely sensed images have been presented [97].

All these algorithms work on a pixel-based classification and for that reason reveal the problem identified previously: they do not include spatial neighbourhood information. This lack of information will result on thematic maps with 'spots', i.e., isolated pixels of a given class will appear in a zone of another class. Observe for example the yellow square in the top right of the thematic map in figure 3.4. That area corresponds to a corn area, however, there are some isolated pixels in the middle of the square identified as forest. If this thematic map were made by a human, all the square would be classified as corn and the limits of each different region of land cover type would be smoother because of our real world perception. To avoid this type of problems, new algorithms were developed to include this information. These algorithms include spatial information in several ways: texture, shapes and neighbouring pixels spectral information [35]. This subject will be addressed later, in section 3.5. Comparisons between

pixel-based algorithms and object-based algorithms are available in recent works [23, 118, 11]. Other examples of object oriented classification of multispectral images may be found in [78, 106, 79, 82, 4, 88].

Recently developments in classification algorithms have been made related to methods that mixture the ideas of supervised and unsupervised classification algorithms, and sparse algorithms. Both developments intend to improve the classifiers response to the problem of the curse of dimensionality: learn a classifier in a high-dimensional feature space with a small number of data samples (we will refer to this problem next section).

The first subject reflects the idea of making use of both labelled and unlabeled data for training - typically a small amount of labelled data with a large amount of unlabeled data. There are two approaches to this technique: the semi-supervised learning [121], and transductive learning (see [110] pp. 339-371). The importance of this technique is related to the cost of having enough labelled training samples to perform the classification task in a remotely sensed image. Several works used this techniques with good results [85, 76, 21, 30].

Support Vector Machines (SVM) are probably the most popular sparse classifier in remote sensing. Sparse classifiers are charactherized for setting automatically to zero irrelevant/redundant parameters. The advantage of sparse classifiers is therefore obvious: it will lead to a structural simplification of the estimated function. Moreover, it will improve the generalization performance of the classifier. In remote sensing classification, where problems related with curse of dimensionality are common, this type of method proved to work well [78, 76, 21, 12]. Sparse classification algorithms include not only SVMs, this family of algorithms include also the Relevance Vector Machine [105], the Sparse Probit Regression [45, 43], sparse online Gaussian processes [34], the Informative Vector Machine [71] and the Joint Classifier and Feature Optimization algorithm [67, 68]. These

algorithms are considered to be among the current state-of-the art in supervised learning [45, 43, 66].

3.4 Hyperspectral Remote Sensing Image Classification

Hyperspectral data provide the capability to discriminate among nearly any set of classes, expanding and improving the capability of multispectral image analysis. Hyperspectral imaging take advantage of hundreds of contiguous spectral channels to capture more subtle details of spectral response of objects on the ground that usually cannot be resolved by multispectral sensors. Hyperspectral imaging has therefore expanded the capability of multispectral imaging in numerous applications in agriculture, ecology, geology, environmental monitoring, military intelligence, law enforcement, and chemical and biological defence. However, this advantage also comes with a price, namely, the knowledge needed to effectively use the spectral information resulting from these hundreds of bands to perform various tasks in data exploitation.

In section 2.2 the main difference between hyperspectral and multispectral sensors where shown to rely on the number of bands captured for each sensor. Figure 3.5 illustrates the type of image produced and the information made available for each type of sensor. As one can observe, it is obvious the increment of information introduced by an hyperspectral sensor. As mentioned before, there is an increase in the spectral information available for each pixel, but this also increases the amount of data to process, resulting in high dimensional datasets.

It is not straightforward which type of image will produce the best final result in a classification task. This happens mainly because the lack of hyperspectral and

multispectral datasets collected simultaneously over the same area. Moreover, the type of classification algorithm applied to hyperspectral datasets usually differ from those applied to multispectral datasets which turns the comparison between these two types of datasets not feasible. Even so, there are some works where hyperspectral and multispectral images are used with the same goal and are then compared [116, 52, 6, 63]. It may be said however that it is consensual that, as a relatively new analytical technique, the full potential of hyperspectral imaging has not yet been discovered, and therefore the comparison with traditional multispectral imaging is not complete.

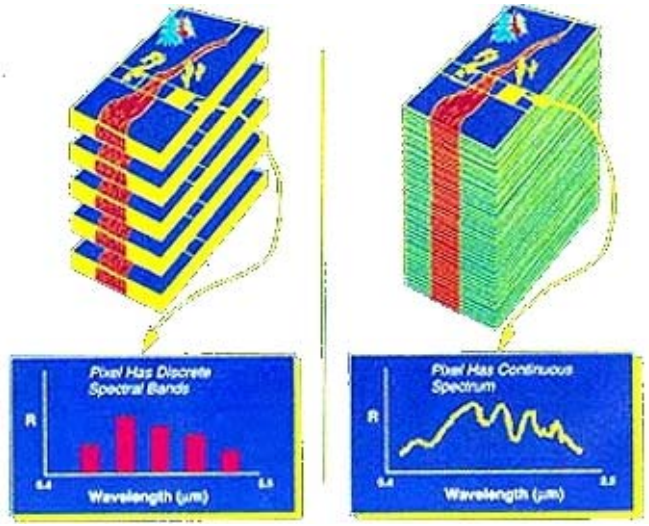


Figure 3.5: An illustration of the structure of multispectral images (right), and hyperspectral images (left).

In the past few years, due to the increase of spectral resolution brought by hyperspectral images, two image processing areas have been suffering great developments: the subpixel detection and mixed pixel classification algorithms. These new developments are due to the challenges brought by three factors:
(i) the presence of pure pixels is very improbable in an image scene (a pixel usually contain two or more material substances) which leaded to new to unmixing

algorithms [86, 87, 24];
(ii) that the signal sources provided by these images may include targets with size smaller than the ground truth sampling distance (they are generally embebed in a single pixel and cannot be visualized by visual inspection), leading to improvements on sub-pixel classification algorithms;
(iii) the sensor characteristics are not ideal which means that the signal recorded for a pixel is in fact obtained from a point spread function, resulting in a mixture of signal from the pixel itself and its neighbours.
More detailed information on these matters may be found in [28].

One of the major problems in statistical learning algorithms is related to the well-known Hughes phenomena [58]. The Hughes phenomena, or curse of dimensionality [5], refers to the exponential growth of hypervolume as a function of dimensionality. In statistics it relates to the fact that the convergence of any estimator to the true value of a smooth function defined on a space of high dimension is very slow. In terms of supervised classification algorithms, this means that, a priori, we need an enormous amount of training samples to obtain a good estimate of classifier. This of course is a major problem in hyperspectral remote sensing applications since the ratio between the number of training samples available and the number of features is frequently small.

To deal with the Hughes phenomena, one can identify three approaches: (i) reduce the data dimensionality, (ii) choose simpler models, or (iii) use unlabelled data to learn the classifier (semi-supervised learning). Initially the most common choice reported in the literature is probably the first one, which indirectly reduces the number of parameters to be estimated. Several feature reduction algorithms have been applied to the supervised classification of hyperspectral images. Some examples may be found in [16, 88]. However, as can be seen from a result in [31] reducing data dimensionality may lead to a more complex classification problem. More recently new methods have been developed

to produce simpler models, namely the discriminative approaches which are able to produce sparse solutions, like we previously saw in section 3.1. These approaches hold the state-of-the art in hyperspectral supervised learning due to their capacity of dealing with small class distances, high dimensionality, and limited training samples. The SVMs are one of the most consolidated discriminative supervised classification tools and have been successfully used for hyperspectral data classification. Examples of the application of sparse classifiers to hyperspectral datasets may be found in [3, 26, 42, 119]. From recent results reported in the scientific literature, one may say that, generally, the second approach is gaining adepts since it produces better resuls in terms of generalization, accuracy and computational economy when dealing with high dimensional datasets. A comparison between these two approaches may be found in [8].

The method developed in this thesis relies on the approach of reducing the classifier complexity trough the inclusion of conditions (priors) that produce sparse classifiers. This will be addressed in detail in section 4.1. Examples of semi-supervised and transductive learning applied to hyperspectral datasets may be found in works recently published [25, 22].
Although many progresses have been made to successfully deal with the Hughes phenomena in hyperspectral imaging, it is still an active area of research, which should see many developments in the coming years.

3.5 The introduction of spatial context

Common and traditional classification algorithms treat each pixel in an image as spatially independent. In remote sensing applications this usually produces thematic maps unlikely to form a patch-like and easily interpretable pattern. This problem has already been introduced in section 3.3, where the figure 3.4

exemplifies the problems related to the result of a pixel-based classification. Better classification outcomes can be achieved if the pixel's spatial context is introduced in the process.

In remotely sensed imagery, there are factors that cause neighbouring pixels to exhibit some level of mutual characteristics. Examples of such factors can be atmospheric interaction, the spatial and spectral resolution of a sensor, and the mechanism of the pixel being generated. Also, when mapping the pixels to landscape patterns, if a pixel is identified as *water*, it will be most likely that the surrounding pixels have the same class. The classification accuracy can be improved if such spatial interaction is well modelled [107].

Image segmentation is an important problem in image analysis, appearing in many applications including pattern recognition, object detection, and medical imaging. This subject has been one of the most studied problems in computer vision. Image segmentation may be shortly defined has a process in which image elements representing the same tissue class are grouped together and labelled [20]. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. Each of the pixels in a region are similar with respect to some characteristic or computed property, such as colour, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic.

In remote sensing, the terms *classification* and *segmentation* are often used to refer to the same process. However, it is important to distinguish these two methods. In this work, classification refers to the process of assigning a class to a pixel based only on its spectral signature. Segmentation is the process where the spatial context of each pixel (it dependency to neighbouring pixels) is added

to the spectral information, improving the results of classification.

In the past couple of decades, many different approaches, formulations and tools have been proposed in computer vision. Two research paths can be used to classify image segmentation works [44]: (i) development of image features, and feature models, as relevant and informative as possible for segmentation goal; and (ii) development of methods that enforce some form of spatial regularity to the segmentation, i.e., that integrate local cues (from features) into a globally coherent segmentation.

There are several examples of developments in both research paths. First some concepts with respect to the first path are introduced.
With the intention of simulate the human image segmentation, some recent proposals combine intensity, texture, and contour-based features [80]. Methods based on pixels intensity have been developed using morphological profiles [7, 91, 42, 82]. Opening and closing morphological transforms are used in order to isolate bright (opening) and dark (closing) structures in images, where bright/dark means brighter/darker than the surrounding features in the images. A different approach uses the multiscale structure built by a specific algorithm to measure and incorporate various properties such as intensity contrast, isotropic texture, and boundary integrity [100].
In remote sensing, a popular type of segmentation is the object-based segmentation. This type of segmentation is easily available in some commercial software (e.g. Definiens, ENVI, ERDAS). It uses a hierarquical scheme starting with one-pixel objects and in numerous subsequent steps, smaller image objects are merged into bigger ones. Throughout the segmentation procedure, the whole image is segmented and image objects are generated based upon several adjustable criteria of homogeneity or heterogeneity in colour and shape. Some examples of hierarquical segmentation may be found in [23, 118, 78, 106, 79].
Segmentation algorithms based on texture are also frequently used. Literature

in texture features and models is quite vast. A recent survey over this subject is presented in [93]. Nevertheless, some of the classical examples of texture-based algorithms can be listed here: Garbor features [60]; wavelet based features [108]; co-occurence matrix [54] and features derived from MRF local texture model [33, 37]. Some examples of the application of these algorithms may be found in [11, 106]. Nonparametric statistical measures of texture similarity may be also used to perform segmentation by resorting to pairwise clustering techniques [57].

In respect to the second research path, there are also various approaches that enforce some form of spatial coherence. Graph-based segmentation methods treat image segmentation as a graph partitioning problem and use a given criterion for segmenting the graph [101, 112, 115].
A segmentation algorithm considers a given class of image partitions. Spatial regularity may also be achieved by constraining these classes. For example, while [56] presents an image segmentation algorithm which represents images by polygonal segments, [89] considers quad-tree-like partitions. Another form of achieving spatial coherence is by imposing some prior on the length or the smoothness of the region boundaries [120]. Some recent works may be found in [62] and references therein. In a probabilistic Bayesian approach, Markov Random Field (MRF) theory allows some form of spatial coherence by incorporating a MRF prior [73].

The work developed in this thesis uses the MRF theory to enforce spatial dependencies, more specifically, it uses a Multi-Level (MLL) Markov-Gibbs prior which will be detailed in section 4.2. To give an idea of MRF theory, one may say that MRF theory is a branch of probability theory used to analyse the spatial or contextual dependencies of physical phenomena. It is frequently used in visual labelling to establish probabilistic distributions of interacting labels since it provides a convenient and consistent way to model context-dependent entities such as image pixels and correlated features. This is accomplished by

defining interactions between neighbouring pixels and building structures using such local interactions. From a computational viewpoint, these local structures allow analysis that are limited to pixels involving sites and their neighbours, and can be performed in parallel.
The use of MRF in hyperspectral image segmentation has been increasingly used in recent years. Some examples of such application can be found in [119, 88].

The basic concepts of pattern recognition and remote sensing were introduced in these first chapters. The following chapters present the original work developed.

Chapter 4

Methodology Developed

This chapter presents the theoretical basis and the methods developed in this work. An introduction to the Sparse Multinomial Logistic Regression (SMLR) method is given, a sparse method to classify high dimensional datasets. Nevertheless it was observed that the direct application to hyperspectral images conducted to computational problems due the very high dimensionality of this type of classification problem. This application difficulty lead to the development of a new approach to the SMLR, the Fast-SMLR. This method is explained in section 4.1.2. Two priors are considered in this problem: the Laplacian and the Jeffreys prior.

The addition of spatial information results in a new bayesian segmentation method. This information is added by means of a MLL Markov Gibd prior which is described in detail in section 4.2. Section 4.3 presents the method used to estimate the optimal segmentation.

4.1 The Fast Sparse Multinomial Logistic Regression method

This section introduces the FSMLR classification method adopting two different priors: Laplacian and Jeffreys. The FSMLR is based in the Multinomial Logistic Regression models, and promotes the sparseness of the classifier using a sparsity promoting prior. First the SMLR proposed by Krishnapuram et. al [66] is reviewed, then two priors are introduced. Finally the *Fast* implementation is presented.

4.1.1 The Sparse Multinomial Logistic Regression method

The SMLR algorithm learns a multi-class classifier based on the multinomial logistic regression. By incorporating a prior, this method performs simultaneously feature selection, to identify a small subset of the most relevant features, and learns the classifier itself.

Let us start first with a review on the multinomial logistic regression theory, and then the priors theory will be introduced.

The logistic regression model arises from the desire to model the posterior probabilities of the K classes via linear functions in $\boldsymbol{x}$, while at the same time ensuring that they sum to one and remain in the range $[0, 1]$ [55].

Let $\boldsymbol{x} = [x_1, \ldots, x_d]^T \in R^d$ be the d observed features. The goal is to assign to each $\boldsymbol{x}$ the probability of belonging to each of the K classes, given K sets of feature weights, one for each class. In particular, if $\boldsymbol{y} = [y^{(1)}, \ldots, y^{(K)}]^T$ is a 1-of-K encoding vector of the K classes, such that $y^{(k)} = 1$ if x corresponds to an example belonging to class k and $y^{(k)} = 0$ otherwise; and if $\boldsymbol{w}^{(k)}$ is the feature weight vector associated with class k, then the probability that a given

sample $\boldsymbol{x}_i$ belongs to class k is given by

$$P\left(y^{(k)}=1|\boldsymbol{x}_i,\boldsymbol{w}\right)=\frac{\exp\left(\boldsymbol{w}^{(k)^T}\boldsymbol{x}_i\right)}{\sum_{k=1}^{K}\exp\left(\boldsymbol{w}^{(k)^T}\boldsymbol{x}_i\right)} \tag{4.1}$$

for $k \in \{1,\ldots,K\}$, where $\boldsymbol{w}=[w^{(1)^T},\ldots,w^{(K)^T}]^T$ and the superscript ^{T} denotes the vector transpose. For binary problems ($K=2$) this is known as a logistic (linear) regression model; for $K>2$ it is known as multinomial logistic (linear) regression.

The model is specified in terms of $K-1$ log-odds or logit transformations (reflecting the constraint that the probabilities sum to one). For this reason, the weight vector for one of the classes need not to be estimated because we have that:

$$P\left(y^{(k)}=1|\boldsymbol{x}_i,\boldsymbol{w}\right)=\frac{\exp\left(\boldsymbol{w}^{(k)^T}\boldsymbol{x}_i\right)}{\sum_{k=1}^{K-1}\exp\left(\boldsymbol{w}^{(k)^T}\boldsymbol{x}_i\right)},\ k\in\{1,\ldots,K-1\}$$

and consequently:

$$P\left(y^{(K)}=1|\boldsymbol{x}_i,\boldsymbol{w}\right)=\frac{1}{\sum_{k=1}^{K-1}\exp\left(\boldsymbol{w}^{(k)^T}\boldsymbol{x}_i\right)}.$$

Without loss of generality, $w^{(K)}$ is set to zero, and the only parameters to be learned are the weight vectors $w^{(k)}$ for $k \in \{1,\ldots,K-1\}$. From now on, $\boldsymbol{w}$ will denote the $(d(K-1))$-dimensional vector of parameters to be learned.

To extend the linear logistic model to include non-linear transformations of the input features, a function $h(x)$ is introduced in equation 4.1:

$$P\left(y^{(k)}=1|\boldsymbol{x}_i,w\right)=\frac{\exp\left(\boldsymbol{w}^{(k)^T}h(\boldsymbol{x}_i)\right)}{\sum_{k=1}^{K}\exp\left(w^{(k)^T}h(x_i)\right)} \tag{4.2}$$

where $h(x)=[h_1(x),\ldots,h_l(x)]^T$ is a vector of l fixed functions of the input, often termed features.

Possible choices for $h(x)$ function are:

- linear: $h(\boldsymbol{x}_i) = [1, x_{i,1}, \ldots, x_{i,d}]^T$, where $x_{i,j}$ is the j^{th} component of x_i, in which case $\boldsymbol{w}$ is a $(d+1)$ dimensional vector;
- non-linear: $h(\boldsymbol{x}) = [1, \phi_1(x), \ldots, \phi_K(x)]^T$, where $\phi_i(\cdot)$ are nonlinear functions. In this case, the dimensionality of $\boldsymbol{w}$ is $K+1$;
- kernel: $h(\boldsymbol{x}) = [1, \mathtt{K}(x, x_1), \ldots, \mathtt{K}(x, x_n)]^T$, where $\mathtt{K}(\cdot,\cdot)$ is some symmetric kernel function [32]. Here, the dimensionaliy of $\boldsymbol{w}$ is $n+1$.

Kernels are nonlinear mappings, thus ensuring that the transformed samples are more likely to be linearly separable. A popular kernel used in image classification is the Gaussian Radial Basis Function (RBF): $\mathtt{K}(\mathbf{x}, \mathbf{z}) = -\exp(\|\mathbf{x} - \mathbf{z}\|^2 / 2\sigma^2)$.

In a supervised learning context, the components of $\boldsymbol{w}$ are estimated from the training data $\mathcal{D} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$. Usually, this estimation is done using a maximum likelihood (ML) procedure to obtain the components of $\boldsymbol{w}$ from the training data, simply by maximizing the log-likelihood function [55]:

$$\begin{aligned} l(\boldsymbol{w}) &= \sum_{i=1}^{n} log P\left(\boldsymbol{y}_i | \boldsymbol{x}_i, \boldsymbol{w}\right) \\ &= \sum_{i=1}^{n} \left[\sum_{k=1}^{K} y_i^{(k)} \boldsymbol{w}^{(k)^T} \boldsymbol{x}_i - \log \sum_{k=1}^{K} \exp\left(\boldsymbol{w}^{(k)^T} \boldsymbol{x}_i\right) \right] \end{aligned} \tag{4.3}$$

The ML estimate for $\boldsymbol{w}$ can be determined using various algorithms like Newton's method, coordinate ascent, conjugate gradient ascent, fixed-Hessian Newton method, quasi-Newton, dual optimization or iterative scaling (see [83] and references therein). Nevertheless, the most widely used method is the Newton's method also known as Iteratively Reweighted Least Squares (IRLS) method since there is no other that clearly outperforms IRLS [83]. However, the log-likelihood function (eq.4.3) can be made arbitrarily large when the training data is separable. For this reason it is crucial the introduction of a prior on $\boldsymbol{w}$.

A sparsity promoting prior on the entries of $\boldsymbol{w}$ is then incorporated, in order to achieve sparsity in the estimate of $\boldsymbol{w}$.

A sparse estimate for $\boldsymbol{w}$ corresponds to an estimate in which irrelevant or redundant components are exactly zero. The sparseness property of a classifier is desirable for several reasons, namely because (i) it leads to a structural simplification of the estimated function and (ii) it often increases the generalization performance, namely when kernel classifiers are used [32, 110]. Moreover, in a sparse classifier, only a subset of the training data has to be kept. This characteristic is therefore of extreme importance when large datasets are considered, as is the case of hyperspectral images.

With the inclusion of a prior over $\boldsymbol{w}$, a maximum *a posteriori* (MAP) is used instead of typical ML criterion for multinomial logistic regression. The estimates of $\boldsymbol{w}$ are then given by:

$$\hat{\boldsymbol{w}}_{MAP} = \arg\max_{\boldsymbol{w}} L(\boldsymbol{w}) = \arg\max_{\boldsymbol{w}} \left[l(\boldsymbol{w}) + \log p(\boldsymbol{w}) \right] \tag{4.4}$$

where $p(\boldsymbol{w})$ is a prior on $\boldsymbol{w}$.

A common way to achieve sparse estimates is by adopting a zero-mean Laplacian (rather than Gaussian) prior on $\boldsymbol{w}$. The Laplacian prior has been widely used as it's capacity to produce sparseness solutions are well-known, and thus has been exploited in several research areas [27, 29, 72, 104, 114]. The sparsity of this prior is achieved due the presence of the l_1 penalty, which sets some components of $\hat{\boldsymbol{w}}$ to zero [104].

The Laplacian prior on $\boldsymbol{w}$ has the form:

$$p(\boldsymbol{w}) \propto \exp(-\lambda \|\boldsymbol{w}\|_1) \tag{4.5}$$

where $\|\boldsymbol{w}\|_1 = \sum_i w_i$ denotes the l_1 norm and λ acts as a tunable regularization parameter.

However, the use of this prior enforces the setting of the λ parameter that controls the degree of sparseness of the obtained estimates. The process of selecting/estimating the *optimum* λ is usually done by cross-validation through

the training process. In the case of high dimensional datasets, such as hyperspectral images, this search often becomes a time consuming task and do not optimally utilize the available data.

In order to overcame this *disadvantage* of the Laplacian prior, a parameter-free prior is introduced in the estimation of class densities: the Jeffreys prior [9]. The Jeffreys prior is a non-informative prior that expresses the notion of ignorance/invariance with respect to changes in measurement scale [9, 46].

Using the Jeffreys prior, $p(\boldsymbol{w})$ is given by

$$p(\boldsymbol{w}) \propto 1/\|\boldsymbol{w}\|_1 \tag{4.6}$$

which is a parameter-free prior, having no longer a sparsity parameter to tune. As will be seen in the experimental tests, this prior will also produce sparse solutions

Considering the adoption of these two priors, the following subsections describe how the estimation of the weights w is done in each case.

4.1.1.1 SMLR with Laplacian prior

The inclusion of a Laplacian prior does not allow for the use of the classical IRLS method to solve the maximization problem in equation 4.4. The bound optimization framework supplies a tool to tackle this optimization problem. The central concept in bound optimization is the iterative replacement of the function to be optimized, in this case $L(\boldsymbol{w}) = l(\boldsymbol{w}) + \log p(\boldsymbol{w})$, with a surrogate function Q [70], such that,

$$L(\boldsymbol{w}^{(t+1)}) \geq L(\boldsymbol{w}^{(t)}) \tag{4.7}$$

and

$$\boldsymbol{w}^{(t+1)} = \arg\max_{\boldsymbol{w}} Q(\boldsymbol{w}|\widehat{\boldsymbol{w}}^{(t)}). \tag{4.8}$$

Conditon 4.7 is acomplished if the surrogate function satifies the key condition that $L(\boldsymbol{w}) - Q(\boldsymbol{w}|\widehat{\boldsymbol{w}}^{(t)})$ attains its minimum at $\boldsymbol{w} = \widehat{\boldsymbol{w}}^{(t)}$. With this condition, it is shown that $L(\boldsymbol{w})$ is increased during the process:

$$\begin{aligned} & L(\widehat{\boldsymbol{w}}^{(t+1)}) \\ & \quad = L(\widehat{\boldsymbol{w}}^{(t+1)}) - Q(\widehat{\boldsymbol{w}}^{(t+1)}|\widehat{\boldsymbol{w}}^{(t)}) + Q(\widehat{\boldsymbol{w}}^{(t+1)}|\widehat{\boldsymbol{w}}^{(t)}) \\ & \quad \geq L(\widehat{\boldsymbol{w}}^{(t)}) - Q(\widehat{\boldsymbol{w}}^{(t)}|\widehat{\boldsymbol{w}}^{(t)}) + Q(\widehat{\boldsymbol{w}}^{(t+1)}|\widehat{\boldsymbol{w}}^{(t)}) \\ & \quad \geq L(\widehat{\boldsymbol{w}}^{(t)}) - Q(\widehat{\boldsymbol{w}}^{(t)}|\widehat{\boldsymbol{w}}^{(t)}) + Q(\widehat{\boldsymbol{w}}^{(t)}|\widehat{\boldsymbol{w}}^{(t)}) \\ & \quad = L(\widehat{\boldsymbol{w}}^{(t)}) \end{aligned} \tag{4.9}$$

A surrogate function that satisfies the key condition can often be achieved by purely analytical methods. Since $l(\boldsymbol{w})$ is concave and C^2, its surrogate function, $Q_l(\boldsymbol{w}|\widehat{\boldsymbol{w}}')$ can be determined using a bound on its Hessian H [13]. Let B be a negative definite matrix such that $H(\boldsymbol{w}) - B$ is positive semi-definite, i.e., $H(\boldsymbol{w}) \succ B$ for any $\boldsymbol{w}$. A valid surrogate function is

$$Q(\boldsymbol{w}|\hat{\boldsymbol{w}}^{(t)}) = \boldsymbol{w}^T \left(g(\hat{\boldsymbol{w}}^{(t)}) - B\hat{\boldsymbol{w}}^{(t)} \right) + \frac{1}{2}\boldsymbol{w}^T B \boldsymbol{w}, \tag{4.10}$$

where

$$B \equiv -\frac{1}{2}\left[I - 11^T/K \right] \otimes \sum_{i=1}^{\mathcal{S}} \boldsymbol{x}_i \boldsymbol{x}_i^T \tag{4.11}$$

where $\otimes$ is the Kronecker matriz product and $1 = [1, 1, \ldots, 1]^T$, $g(\boldsymbol{w})$ is the gradient of $l(\boldsymbol{w})$ given by

$$g(\boldsymbol{w}) = \sum_{i=1}^{\mathcal{S}} \left(\boldsymbol{y}_i' - p_i(\boldsymbol{w}) \right) \otimes \boldsymbol{x}_i, \tag{4.12}$$

with $\boldsymbol{y}_i' = \left[y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(K-1)} \right]^T$ and $p_i(\boldsymbol{w}) = \left[p_i^{(1)}(\boldsymbol{w}), \ldots, p_i^{(K-1)}(\boldsymbol{w}) \right]^T$.

Concerning the l_1 norm $\|\boldsymbol{w}\|_1 = \sum_k |w_k|$, it is worth noting that, for a given $w_k^{(t)}$,

$$-|w_k| \geq -1/2 w_k^2 / |w_k^{(t)}| + c^{te}, \tag{4.13}$$

where c^{te} is a constant. Thus, both terms of $L(\boldsymbol{w})$ have a quadratic bound. Since the sum of functions is lower bounded by the sum of the correspondent lower bounds, there is a quadratic bound for $L(\boldsymbol{w})$, given by

$$Q(\boldsymbol{w}|\widehat{\boldsymbol{w}}^{(t)}) = \boldsymbol{w}^T \left(g(\hat{\boldsymbol{w}}^{(t)}) - B\hat{\boldsymbol{w}}^{(t)} \right) + \frac{1}{2}\boldsymbol{w}^T B \boldsymbol{w} + \frac{1}{2}\boldsymbol{w}^T \Lambda^{(t)} w \boldsymbol{w} \tag{4.14}$$

where

$$\Lambda^{(t)} = \text{diag}\left\{ \left| \hat{\boldsymbol{w}}_1^{(t)} \right|^{-1}, \ldots, \left| \hat{w}_{d(K-1)}^{(t)} \right|^{-1} \right\}. \tag{4.15}$$

The maximization of (4.14) leads to

$$\hat{w}^{(t+1)} = (B - \lambda\Lambda^{(t)})^{-1} \left(B\hat{w}^{(t)} - g(\hat{w}^{(t)}) \right). \tag{4.16}$$

Numerically, Eq.4.16 is equivalent to solve [66]:

$$\hat{w}^{(t+1)} = \Gamma^{(t)} \left(\Gamma^{(t)} B \Gamma^{(t)} - \lambda I \right)^{-1} \Gamma^{(t)} \left(B\hat{w}^{(t)} - g(\hat{w}^{(t)}) \right), \tag{4.17}$$

where

$$\Gamma^{(t)} = \text{diag}\left\{ \left| \hat{w}_1^{(t)} \right|^{1/2}, \ldots, \left| \hat{w}_{d(K-1)}^{(t)} \right|^{1/2} \right\}. \tag{4.18}$$

Equation 4.17 is set to this form to avoid inverse weight estimates, because some of them are expected to be zero.

It is now possible to perform exact MAP multinomial logistic regression under a Laplacian prior, with the same cost as the original IRLS algorithm for ML estimation (see [92]). However, an important issue remains - the adjustment of the sparseness parameter λ in Eq.4.17. This should be done by cross-validation, which may result in a time consuming task. To avoid this, we adopt a Jeffreys prior on the weights. The next section describes how the ML multinomial logistic regression is performed with this prior.

4.1.1.2 SMLR with Jeffreys prior

The use of a Jeffreys prior removes the sparseness parameter λ from the model, since it is a parameter-free prior. As will be shown experimentally, this prior

strongly induces sparseness and yields state-of-the-art performance in image segmentation.

The Jeffreys prior is a heavy-tailed non-informative prior which has been used in several processing applications (like classification, regression [43] and image deconvolution [10]) to produce sparse solutions and avoid the search for the parameter that controls sparsity of the algorithms.

Here it will be shown how the bound optimization algorithm presented in previous section is applied when the Jeffreys prior is introduced. When using the Jeffreys prior, minor changes are required in the estimation of the weights $\boldsymbol{w}$ described previously for the Laplacian prior.

In this case, the function to be optimized is

$$L(\boldsymbol{w}) = l(\boldsymbol{w}) - log(\|\boldsymbol{w}\|_1),$$

instead of

$$L(\boldsymbol{w}) = l(\boldsymbol{w}) - \lambda \|\boldsymbol{w}\|_1 .$$

In this way, the surrogate function for $l(\boldsymbol{w})$, $Q(\boldsymbol{w}|\widehat{\boldsymbol{w}}^{(t)})$ (eq.4.10) is kept, and also the inequality 4.13 remains valid. Consequently both terms of $L(\boldsymbol{w})$ continue to have a quadratic bound given by equation 4.14. However, the introduction of the Jeffreys prior conducts to a new matrix $\Lambda^{(t)}$. This matrix is now given by

$$\Lambda^{(t)} = \text{diag}\left\{ \left| \hat{\boldsymbol{w}}_1^{(t)} \right|^{-2}, \ldots, \left| \hat{\boldsymbol{w}}_{d(K-1)}^{(t)} \right|^{-2} \right\}. \tag{4.19}$$

The removal of the sparseness parameter from the iterative equation 4.17 is other modification.

The maximization of (4.19) leads now to the iteration equation:

$$\hat{w}^{(t+1)} = (B - \Lambda^{(t)})^{-1} \left(B\hat{w}^{(t)} - g(\hat{w}^{(t)}) \right), \tag{4.20}$$

which numerically is equivalent to solving:

$$\hat{w}^{(t+1)} = \Gamma^{(t)} \left(\Gamma^{(t)} B \Gamma^{(t)} - I \right)^{-1} \Gamma^{(t)} \left(B\hat{w}^{(t)} - g(\hat{w}^{(t)}) \right), \tag{4.21}$$

similarly to what was done in the previous section. Here $\Gamma^{(t)}$ is given by

$$\Gamma^{(t)} = \operatorname{diag} \left\{ \left| \hat{w}_1^{(t)} \right|, \ldots, \left| \hat{w}_{d(K-1)}^{(t)} \right| \right\} \tag{4.22}$$

We now have two linear systems to solve, one for Laplacian prior (eq. 4.17) and another one for Jeffreys prior (eq. 4.21), to determine the $\boldsymbol{w}$ estimates.

4.1.2 The iterative modification to SMLR

Independently of the prior used, the update equations include a non-constant matrix $\left(\Gamma^{(t)} B \Gamma^{(t)} - I\right)$, that need to be inverted at each iteration. This process leads to a high computational cost when solving the linear system in (4.17) and (4.21) at each iteration, which becomes often prohibitive. The cost at each iteration is of the order of $((dK)^3)$, turning the application of SMLR to large datasets very difficult, either because the original number of features is very large, or because a very large training dataset is used. In the case of hyper-spectral image segmentation, the problem is the number of bands (d), which is usually very large.

In order to circumvent this problem, a modification to the iterative method used in SMLR is introduced. This modification results in a faster and more efficient algorithm: the Fast-SMLR (FSMLR) [14]. FSMLR uses the Block Gauss-Seidel (GS) method [92] to solve the system used in the IRLS method. The modification consists, at each iteration, in solving blocks corresponding to the weights belonging to the same class, instead of computing the complete set of weights.

The system to be solved in the case of Laplacian prior is presented in equation 4.16. When the Jeffreys prior is considered, the rationale is equivalent but

without the parameter λ and the matrix Λ, and is given by expression 4.19 instead of expression 4.15. So only the Laplacian prior will be explained here.

If one considers the update equation given in Eq.4.16, and proceed to some algebric modifications, one has:

$$\begin{aligned}
&\hat{w}^{(t+1)} = \left(B - \lambda\Lambda^{(t)}\right)^{-1}\left(B\hat{w}^{(t)} - g(\hat{w}^{(t)})\right) \\
&\Leftrightarrow \left(B - \lambda\Lambda^{(t)}\right)\hat{w}^{(t+1)} = \left(B\hat{w}^{(t)} - g(\hat{w}^{(t)})\right) \\
&\Leftrightarrow \Lambda^{(t)}\left(\Lambda^{(t)^{-1}}B - \lambda I\right)\hat{w}^{(t+1)} = \left(B\hat{w}^{(t)} - g(\hat{w}^{(t)})\right) \\
&\Leftrightarrow \left(\Lambda^{(t)^{-1}}B - \lambda I\right)\hat{w}^{(t+1)} = \Lambda^{(t)^{-1}}\left(B\hat{w}^{(t)} - g(\hat{w}^{(t)})\right) \qquad (4.23)
\end{aligned}$$

Recall that $\Lambda^{(t)^{-1}}$ is a diagonal matrix, where each block k corresponds to the weights of that class and has size $d \times d$, so, $\Lambda^{(t)^{-1}}$ has dimension $(d*(K-1)) \times (d*(K-1))$. Matrix B (eq. 4.11) has dimension $(d*(K-1)) \times (d*(K-1))$ and it can be decomposed as a block matrix, were each block corresponds to a class:

$$B = \begin{bmatrix} B_{1,1} & \dots & B_{1,K-1} \\ \vdots & & \vdots \\ B_{K-1,1} & \dots & B_{K-1,K-1} \end{bmatrix}$$

where $B_{k,k}$ corresponds to the block correspondent to class k and is of dimension $d \times d$. In addition, if we set

$$Y = \Lambda^{(t)^{-1}}\left(B\hat{w}^{(t)} - g(\hat{w}^{(t)})\right)$$

and W the matrix of class weights to be updated at each iteration such as

$$W = \begin{bmatrix} w_1 \\ \vdots \\ w_{K-1} \end{bmatrix},$$

where w_k is the block corresponding to the weights of class k, we have that solving the linear systems in (4.17) with the block GS iterative procedure, with

the blocks coinciding with the class weights is equivalent to solve:

$$\left(\begin{bmatrix} \left|\hat{w}_1^{(t)}\right| & & \\ & \ddots & \\ & & \left|\hat{w}_{K-1}^{(t)}\right| \end{bmatrix} \begin{bmatrix} B_{1,1} & \dots & B_{1,K-1} \\ \vdots & & \vdots \\ B_{K-1,1} & \dots & B_{K-1,K-1} \end{bmatrix} - \lambda I \right) \begin{bmatrix} w_1 \\ \vdots \\ w_{K-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_{K-1} \end{bmatrix}.$$

Writing this set of linear systems in a simpler way, we have

$$\begin{bmatrix} A_{1,1} & \dots & A_{1,K-1} \\ \vdots & & \vdots \\ A_{K-1,1} & \dots & A_{K-1,K-1} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_{K-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_{K-1} \end{bmatrix}.$$

This means that in each iteration we solve the following system for each block k

$$A_{k,k} w_k = y_k - \sum_{i \neq k} A_{i,k} w_i, \tag{4.24}$$

where matrix A is a block diagonal matrix resulting from $\Lambda^{(t)^{-1}} B - \lambda I$.

Using this technique, what happens is that, at each iteration, K systems of equal dimension to the number of samples are solved. This results in an improvement in terms of computational effort of the order of (K^2).

4.2 The inclusion of contextual information

The application of FSMLR with a Laplacian or a Jeffreys prior enforces sparsity in the estimation of class densities, providing a competitive method for the classification of hyperspectral images. However, this can be improved by adding information about the neighbourhood of each pixel. The inclusion of contextual

information together with the spectral information will model the piecewise smoothness of real world images. The addition of the contextual information will result in a label image with a finite set of nonoverlapping homogeneous regions. This process is known as segmentation.

The goal of segmentation is to estimate the label image having observed the feature images. Let us represent the observed hyper-dimensional images (the feature images) as $\mathbf{x} = \left\{\boldsymbol{x}_i \in R^d, i \in \mathcal{S}\right\}$, where d is the number of spectral bands and $\mathcal{S}$ the total number of pixels of the scene considered. The goal is to assign to each $\boldsymbol{x}_i, i \in \mathcal{S}$, a label $\boldsymbol{y}_i \in \mathcal{L} = \{1, 2, \ldots, K\}$, resulting in an image of labels $\mathbf{y} = \{\boldsymbol{y}_i\}_{i \in \mathcal{S}}$.

In a Bayesian framework, the estimation of $\mathbf{y}$ having observed $\mathbf{x}$ is done by maximizing the posterior distribution

$$p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{y})p(\mathbf{y}) \tag{4.25}$$

where $p(\mathbf{x}|\mathbf{y})$ is the likelihood function (or the probability of feature image given the labels) and $p(\mathbf{y})$ is the prior over the classes.

To determinate the likelihood function, the FSMLR is adopted to learn the densities of labels. The likelihood is given by $p(\boldsymbol{x}_i|\boldsymbol{y}_i) = p(\boldsymbol{y}_i|\boldsymbol{x}_i)p(\boldsymbol{x}_i)/p(\boldsymbol{y}_i)$. The class densities $p(\boldsymbol{y}_i|\boldsymbol{x}_i)$ are learned by the discriminative classifier presented in the previous section, the FSMLR. In addition, since $p(x_i)$ does not depend on the labeling $\mathbf{y}$, we have

$$p(\mathbf{x}|\mathbf{y}) \propto \prod_{i \in \mathcal{S}} p(y_i|x_i)/p(y_i), \tag{4.26}$$

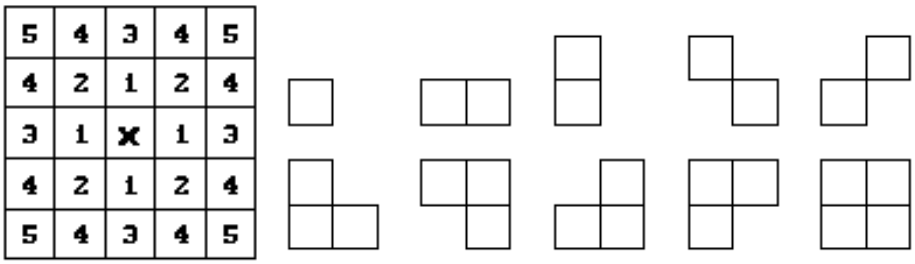
where conditional independence is understood. Also, in this approach, the classes are assumed as likely probable: $p(y_i) = 1/K$. Although this assumption may not be the ideal, it still leads to very good results. The class densities can be tilted, if required, towards other distribution by using the method described in [81].

Assuming the Bayesian model given in expression 4.25, it is possible to introduce the contextual information by modelling the prior over classes $p(\mathbf{y})$.

MRF models allow one to incorporate contextual constraints in a principled manner. The MLL prior is a MRF that models the piecewise continuous nature of the image elements, considering that adjacent pixels are likely to belong to the same class. This prior is a generalization of the Ising model [50] and has been widely used in image segmentation problems [73].

The Hammersly-Clifford theorem is a key theoretical result that provides a method to write the density of a MRF. To define that density, some definitions are required first.

Let us define ∂s as the neighbourhood of pixel s. This neighbourhood system should be symmetric $r \in \partial s \Rightarrow s \in \partial r$ also $s \notin \partial s$. A clique is a set of pixels, c, that are neighbours of one another: $\forall s, r \in c, r \in \partial s$. The cliques can have several forms and number of pixels, depending on the neighbourhood order considered (Figure 4.1). Figure 4.1a presents different orders of neighbourhood (the numbers indicate the outermost neighbouring sites in the n^{th} order neighbourhood system). In this work, 2nd order neighbourhoods will be considered.



(a) Neighbourhood systems

(b) Clique types for the 2^{nd} order neighbourhood

Figure 4.1: Neighbourhood systems and 2^{nd} order neighbourhood cliques (figure adapted from [73])

According to the Hammersly-Clifford theorem, the density associated with a MRF is a Gibb's distribution [50]. Therefore, the prior model for segmentation has the structure

$$p(\mathbf{y}) = \frac{1}{Z} \exp\left(- \sum_{c \in C} V_c(\mathbf{y}) \right), \tag{4.27}$$

where Z is the normalizing constant for the density and the sum in the exponent is over the prior potentials $V_c(\mathbf{y})$ for the set of cliques C over the image. The $\sum_{c \in C} V_c(\mathbf{y})$ is known as energy function, and the potential is of the form:

$$-V_c(\mathbf{y}) = \begin{cases} \alpha_{y_i} & if \quad |c| = 1 \quad \text{(single clique)} \\ \beta_c & if \quad |c| > 1 \quad \text{and } \forall_{i,j \in c}\, y_i = y_j \\ -\beta_c & if \quad |c| > 1 \quad \text{and } \exists_{i,j \in c}\, y_i \neq y_j \end{cases} \tag{4.28}$$

where β_c is a non-negative constant.

The definition of the potencial function in 4.28 encourages neighbours to have the same label. By varying the set of cliques and the parameters α_{y_i} and β_c, MLL offers a great deal of flexibility. For example, the model generates texture-like regions if β_c depends on c and blob-like regions otherwise.

Equation (4.27) can be written as

$$p(\mathbf{y}) = \frac{1}{Z} e^{\beta n(\mathbf{y})} \tag{4.29}$$

where $n(\mathbf{y})$ denotes the number of cliques having the same label, if we let $\alpha_k = \alpha$ and $\beta_c = \frac{1}{2}\beta > 0$. This choice gives no preference to any label nor to any direction.

The conditional probability $p(y_i = k | y_j, j \in \mathcal{S} - i)$ is then given by

$$p(y_i = k | y_{\mathcal{N}_i}) = \frac{e^{\beta n_i(k)}}{\sum_{k=1}^{K} e^{\beta n_i(k)}}, \tag{4.30}$$

where $n_i(k)$ is the number of sites in the neighbourhood of site i, $\mathcal{N}_i$, having the label k.

We have now a powerful mechanism for modelling spatial continuity, by using the neighbourhood information in the MLL prior. The next step is the parameter estimation problem. The answer to this is given by a MAP optimization problem which will be addressed in the next section.

4.3 MAP segmentation

After learning the class densities $p(\mathbf{x}|\mathbf{y}) \propto \prod_i p(y_i|x_i)$ with FSMLR and modelling the prior over classes $p(\mathbf{y})$ by a MLL prior, one has a MAP segmentation problem, given by

$$\begin{aligned} \hat{\mathbf{y}} &= \arg\max_{\mathbf{y}} p(\mathbf{x}|\mathbf{y})p(\mathbf{y}) \\ &= \arg\max_{\mathbf{y}} \sum_{i\in\mathcal{S}} \log p(x_i|y_i) + \beta n(\mathbf{y}) \\ &= \arg\min_{\mathbf{y}} \sum_{i\in\mathcal{S}} -\log p(x_i|y_i) - \beta \sum_{i,j\in c} \delta(y_i - y_j), \end{aligned} \tag{4.31}$$

where δ is the unit impulse function. The minimization of (4.31) is a hard combinatorial optimization problem. The combinatorial nature of this optimization problem limits the number of algorithms available to achieve the optimum solution.

The function to be minimized can be viewed as an energy function where the first term penalizes the solutions that are inconsistent with the observed data and the second term enforces some kind of spatial coherence. Efficient energy minimization algorithms have been recently developed to tackle this kind of problem. Examples of those developments are based in optimization methods like Graph Cuts [19], Loopy Belief Propagation [117, 111] and tree-reweighted message passing [65].

Graph cut techniques from combinatorial optimization can be used to find the global minimum for some multi-dimensional energy functions. Two algorithms

that use the graph cuts technique to compute local minimum are proposed in [19]. The swap move algorithm and the expansion move algorithm are the most popular graph cuts algorithms. These algorithms rapidly compute a local minimum, in the sense that no 'permitted move' will produce a labelling with lower energy.

The swap move takes some subset of the pixels currently given the label α and assigns them the label β, and vice-versa, given a pair of labels α and β. The swap move algorithm finds a local minimum such that there is no swap move, for any pair of labels α, β that will produce a lower energy labelling. The expansion move for a label α increases the set of pixels that are given this label. The expansion move algorithm finds a local minimum such that no expansion move, for any label α, yields a label with lower energy.

These algorithms guarantee the identification of the optimal labelling if the energy function is equivalent to a semi-metric in the swap, and metric in the expansion algorithm [64].

It can be shown that the pairwise interaction term on the right hand side of (4.31) is equivalent to a metric. The metric is obtained by simply adding β to terms $-\beta\delta(y_i - y_j)$. This equivalence lead us to apply the α-Expansion algorithm since it guarantees very good approximations [19].

The combinatorial optimization literature provides several min-cut/max-flow algorithms on graphs as a useful tool for exact or approximate energy minimization. The α-Expansion algorithm makes use of a min-cut/max-flow algorithm presented by [18], by iteratively running this algorithm in appropriate graphs. Boykov's min-cut/max-flow algorithm consistently proved to be faster than several standard algorithms, and in some applications made near real-time performance possible [18]. The implementation of this algorithm was made available by the authors upon request for research purposes.

Chapter 5

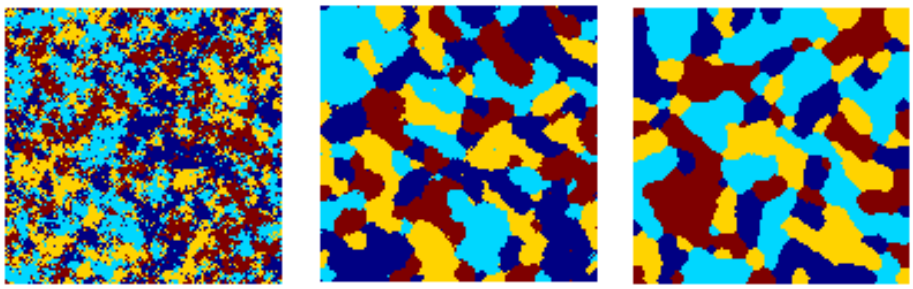
Experimental Setup

The final goal of the development of theorical methods is their application to real problems to give the solutions needed. An essential previous step is the study of the behaviour of the Bayesian image Segmentation method with Discriminative Class Learning (BSDCL) in controled enviroments. In this particular case, that step is the application of the proposed algorithms to simulated hyperspectral datasets where all the data characteristics are controlled and known by the user. In this chapter, the datasets used to test the methods developed in this work are presented, as well as the experimental procedures applied to infer the quality of the BSDCL method. The datasets include synthetic hyperspectral images as well as benchmarked hyperspectral datasets and are presented in sections 5.1, 5.2 and 5.3. In section 5.4 the experimental procedures are presented.

5.1 Synthetic test data

The generation of the simulated hyperspectral datasets can be viewed as a two step procedure: (i) generation of label images, $\mathbf{y}$; (ii) generation of features images, $\mathbf{x}$.

The label images were generated using a MLL distribution (see Section 4.2) using the Gibbs sampler [50] with a 2^{nd} order neighbourhood, where the 2^{nd} order neighbourhood of a site (i, j) is considered as the set of sites $\mathcal{N}_{i,j} = \{(i, j+1), (i-1, j+1), (i-1, j), (i-1, j-1), (i, j-1), (i+1, j-1), (i+1, j), (i+1, j+1)\}$. The shape of these label images depends on a parameter (β_{MLL}) that controls the spatial continuity. A higher value for β produces a more homogeneous spatial continuity image. Figure 5.1 shows three examples of these label images with 4 classes, for $\beta_{MLL} = 0.5$, $\beta_{MLL} = 1$ and $\beta_{MLL} = 2$, all of 120×120 pixel size.



(a) $\beta_{MLL} = 0.5$ (b) $\beta_{MLL} = 1$ (c) $\beta_{MLL} = 2$

Figure 5.1: Image labels with four classes generated by a MLL distribution for different values of β.

To apply the methods developed, different types of image labels were generated. Images with 4 and 10 classes and with different degrees of spatial continuity (for example, with $\beta_{MLL} = 1; 1.2; 1.4; 1.6; 1.8$ and 2) were generated, all with 120×120 pixel size.

The simulated feature images, $\mathbf{x}$, were generated according to a Gaussian density $p(\mathbf{x}|\mathbf{y})$, were the prior $p(\mathbf{y})$ follows a MLL density. The feature images were obtained by adding zero-mean Gaussian independent noise with standard deviation σ_N to a source matrix of mineral signatures. In this way, the simulated spectral vector x_i for $i \in \mathcal{S}$, given the label y_i, is Gaussian distributed with mean $\mu(y_i)$

and covariance matrix $\sigma_N^2\mathbf{I}$, i.e, $x_i \sim \mathcal{N}\left(\mu(y_i), \sigma_N^2\mathbf{I}\right)$. The means $\mu(y_i)$, playing the role of spectral signatures, were extracted from a source matrix provided by a Matlab data file, and were extracted from the USGS spectral library [109]. Each mineral signature is evaluated in 221 spectral bands, resulting in a dataset of dimension 120×120×221.

The noise variance can be tunned to test the behaviour of the algorithm. Values of $\sigma_N^2 = 0.01, 0.1$ and 1 were tested in FSMLR algorithm. However, the majority of tests were done by setting $\sigma_N^2 = 1$, corresponding to a signal-to-noise ratio below one, resulting in a hard classification problem.

5.2 Indian Pines Dataset

One of the most popular hyperspectral image data sets used to test image processing techniques is the well-known hyperspectral AVIRIS spectrometer Indian Pines 92 from Northern Indiana [69]. This benchmarked dataset has been frequently used to test several techniques in the processing of hyperspectral images, providing a good evaluation exercise. The ground truth data image

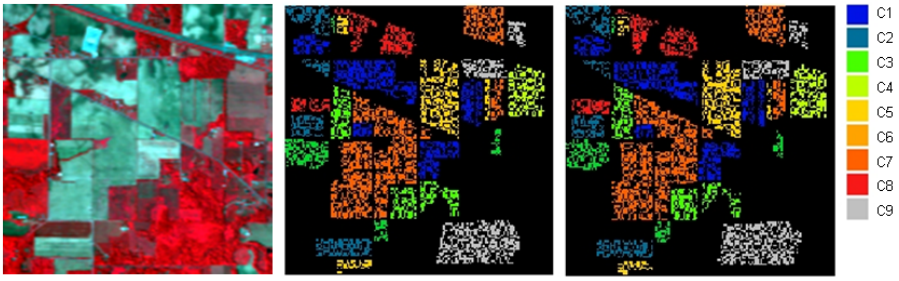


Figure 5.2: AVIRIS image used for testing. Left: original image band 50 (near infrared); Centre: training areas; Right: validation areas.

consists of 145×145 pixels of the AVIRIS image in 220 contiguous spectral bands, at 10 nm intervals in the spectral region from 0.40 to 2.45 μm. The

spatial resolution of these images is 20m.

Four of the 224 original AVIRIS bands contained no data or zero values and were thus removed. In the remaining 220 bands, there are 20 spectral bands that correspond to spectral regions where there is a significant absorption of radiation by the atmosphere due to water vapour. Bands 104-108, 150-163 and 220 are for that reason considered noisy bands. The image covers an agricultural

Table 5.1: Number of training and validation samplesin the AVIRIS Indian Pines hyperspectral image.

Class	Training	Validation
C1 - Corn-no till	742	692
C2 - Corn-min till	442	392
C3 - Grass/Pasture	260	237
C4 - Grass/Trees	389	358
C5 - Hay-windrowed	236	253
C6 - Soybean-no till	487	481
C7 - Soybean-min till	1245	1223
C8 - Soybean-clean till	305	309
C9 - Woods	651	643
	4757	4588

portion of North-West Indiana with 16 identified classes. The data set represents a very challenging land-cover classification scenario, in which the primary crops of the area (mainly corn and soy-beans) were very early in their growth cycle, with only about 5% canopy cover. Discriminating among the major crops under this circumstances can be very dificult (in particular, given the moderate spatial resolution of 20m).

Due to the insufficient number of training samples, seven classes were discarded,

leaving a dataset with 9 classes distributed by 9345 elements. This dataset was randomly divided into a set of 4757 training samples and 4588 validation samples. The number of samples per class and the class labels are presented in table 5.1 and their spatial distribution within the image ca be seen in figure 5.2.

5.3 Pavia Datasets

The Pavia hyperspectral dataset was also used to test the BSDCL method. The Pavia datasets were collected by the optical sensor ROSIS 03 (Reflective Optics System Imaging Spectrometer) in the framework of HySens project managed by DLR (German Aerospace Agency) [36]. The images from the ROSIS spectrometer have 115 spectral bands with a spectral coverage from 0.43 to 0.86 μm. In the particular case of the images over Pavia, the flight altitude was chosen as the lowest available for the airplane, which resulted in a spatial resolution of 1.3m per pixel. Two scenes over Pavia were made availabe, a scene over the city centre and another over Pavia University. Three different subsets of the full data were then considered:

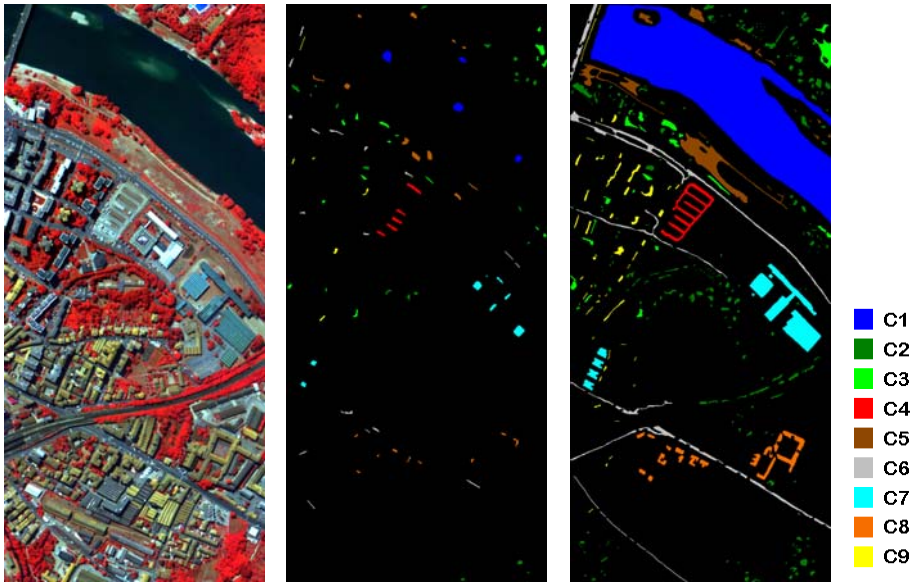
- *Dataset 1* - Image over Pavia city centre with 492 by 1096 pixel in size, 102 spectral bands (without the noisy bands) and nine ground-truth classes distributed by 5536 training samples and 103539 validation samples (Fig.5.3).
- *Dataset 2* - Image over Pavia University with 310 by 340 pixel in size, 103 spectral bands (without the noisy bands) and nine ground-truth classes distributed by 3921 training samples and 42776 validation samples (Fig.5.5).
- *Dataset 3* - Superset of the scene over Pavia city centre, including a dense residential area, with 715 by 1096 pixel in size and nine ground-truth classes distributed by 7456 training samples and 148152 validation samples (Fig.5.4).

In the following subsections the images are shown as well as the spatial distribution of the training and test samples, and the distributions of pixels per class.

5.3.1 Pavia Centre

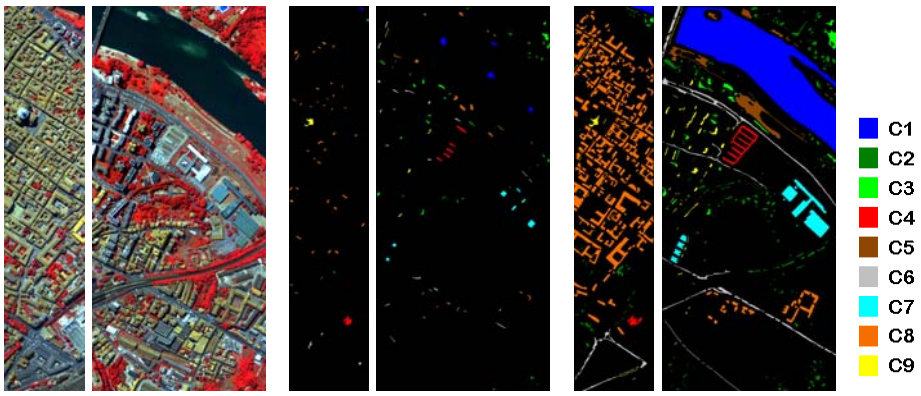
The image over Pavia city center presents a dense residential area on one side of the river Ticino and open areas on the other side. Originally, this image was of size 1096 by 1096 pixels, however this image included a 381 pixel wide black strip in the middle of the image. For correct processing, the stripe with no information was removed, resulting in a *two part* image with 715 by 1096 pixels – *Dataset 3* (Fig. 5.4a).

The right part of the original image was considered individually. It resulted in a 492 by 1096 pixels size image – *Dataset 1* (Fig.5.3a).



(a) RGB Color Composite (80,45,10) (b) Training Set (c) Ground Truth

Figure 5.3: Pavia Dataset 1



(a) RGB Color Composite (80,45,10) (b) Training Set (c) Ground Truth

Figure 5.4: Pavia Dataset 3

Some spectral bands were removed due to noise thus resulting in datasets of dimension 102. Nine land cover classes were identified in the city centre area: water, trees, meadow, bricks, bare soil, asphalt, bitumen, tiles and shadow. The number of samples per class and the class labels are presented in table 5.2, while the spatial distribution of training sites of datasets 1 and 3 are presented in figures 5.3b and 5.4b, respectively. The spatial distribution of ground truth pixels used to test the performance of the segmentation procedure are presented in figures 5.3c and 5.4c, respectively.

5.3.2 Pavia University

The second test site is around the Engineering School at the University of Pavia (*Dataset 2*). The University area image is 610 by 340 pixels (Fig.5.5a). Nine classes were also identified, however they are not all the same as in Pavia centre. Information classes for *Dataset 2* are: asphalt, meadow, gravel, trees, metal, bare soil, bitumen, brick, and shadow.

The number of samples per class and the class labels are presented in table 5.3.

Table 5.2: Number of training and validation samples of dataset 1 and 3

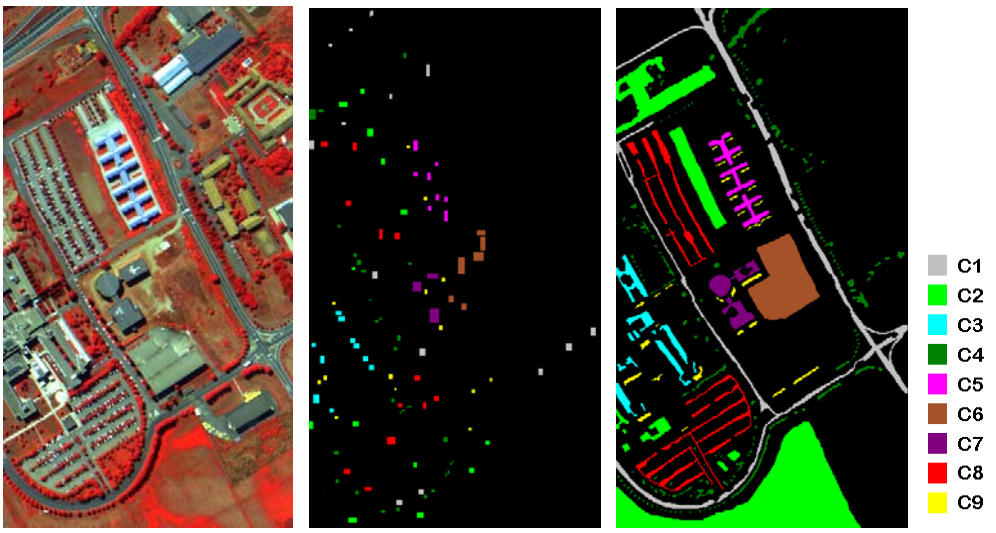
	Dataset 1		Dataset 3	
Class	Training	Validation	Training	Validation
C1 - Water	745	65278	824	65971
C2 - Trees	785	6508	820	7598
C3 - Meadow	797	2905	824	3090
C4 - Bricks	485	2140	808	2685
C5 - Bare Soil	820	6549	820	6584
C6 - Asphalt	678	7585	816	9248
C7 - Bitumen	808	7287	808	7287
C8 - Tiles	223	3122	1260	42826
C9 - Shadow	195	2165	476	2863
	5536	103539	7456	148152

The spatial distribution of training sites and ground truth pixels are presented in figures 5.5b and 5.5c, respectively.

5.4 Experimental Procedures

This section outlines the experimental procedures performed to test the proposed classification and segmentation methods (FSMLR and BSDCL, respectively).

Both in FSMLR as in BSDCL methods, there are parameters to be tuned by the user. A simple and common way to estimate the optimal set of parameters is the cross-validation method [38]. In k-fold cross validation method, the set of labelled training samples is randomly divided in k disjoint parts of equal size.



(a) Color Composite (b) Training Set (c) Ground Truth

Figure 5.5: Pavia Dataset 2

The classifier, or any other algorithm, is then trained k times, each time with a different set, held out as a validation set. The estimated performance is the mean of those k errors and the set of optimal parameters is selected as the one that gives the lower validation error. It is essential that the validation set does not include points used for training the parameters.

In each scenario analysed, a k-fold cross validation was applied to the training set, to access the optimal set of parameters. A measure to evaluate each method is given by the Overall Accuracy (OA), which is calculated over an independent test set. The OA are simply the ratio of the number of correct classified pixels over the total number of pixels in the independent test set.

The simulated datasets presented in section (Sec. 5.1) allows for an extensive evaluation of the performance of each method since the labels of all image pixels are known. The tests with simulated images are of great interest since it is relatively easy to generate different scenarios regarding several parameters, such as size of training sets, number of classes, spatial dispersion of classes and degree of noise presented in the feature images.

Table 5.3: Number of training and validation samples of Dataset 2

Class	Training	Ground Truth
C1 - Asphalt	548	6631
C2 - Meadow	540	18649
C3 - Gravel	392	2099
C4 - Trees	524	3064
C5 - Metal	265	1345
C6 - Bare Soil	532	5029
C7 - Bitumen	375	1330
C8 - Brick	514	3682
C9 - Shadow	231	947
	3921	42776

In the simulated hyperspectral images, all the pixels present in the image were used to study the methods. Training and independent test samples were randomly generated with various dimensions, and the k-fold cross validation method applied: the training sets were used to access the set of optimal parameters for each method and the independent validation set used to determine the OA, as well as other measures that allow a more complete evaluation of the performance of the methods, such as the degree of sparseness, for example.

Both Indian Pines and Pavia datasets present distinct sets for training and testing (Sec. 5.2 and Sec. 5.3). The Indian Pines training and test sets were created according with the work by Camps-Valls [26] to make possible a correct comparison of different classification methods for hyperspectral images. The Pavia dataset was already provided with the training and test datasets defined. In an analogous manner to what was done with the simulated images, also with the benchmarked datasets the k-fold cross validation method was applied to the

training sets to estimate the optimal set of parameters. The test set was used to analyse the OA and sparseness degree of the classification methods.
The results of these experiments in several scenarios are presented in the next chapter.

Chapter 6

Results

In this chapter the results from the experimental tests done over datasets introduced in chapter 5 are presented and discussed. Three main sections compose this chapter, as a consequence of the number of test datasets used. The results are discussed for each kind of dataset to facilitate the exploitation of the characteristics, advantages and performance of each method presented in chapter 4. Each section starts by analizing the FSMLR method, both with $h(x)$ linear and RBF. Studies over the two types of prior (Laplacian and Jeffreys) are also explored. Finally results from the MRF segmentation procedure are presented.

Figure 6.1 presents a scheme with the different types of algorithms tested for

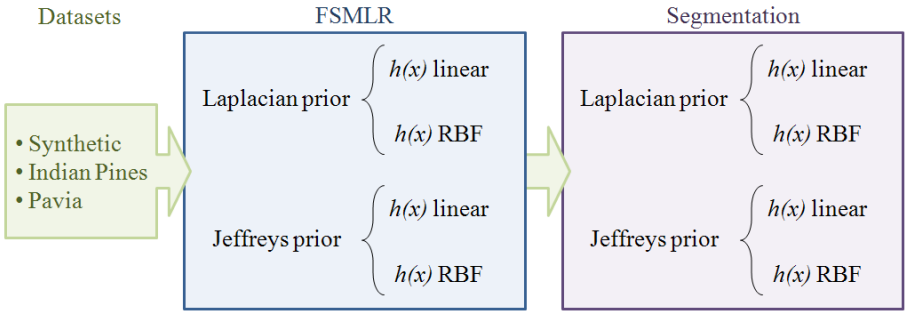


Figure 6.1: The different types of algorithms tested for each dataset.

each dataset. Both for FSMLR classification as for the segmentation, results for different priors and input functions are presented.

6.1 Synthetic test data

This section is concerned with the study of the performance of the methods proposed in chapter 4 when applied to different synthetic images. Synthetic hyperspectral images were generated to evaluate the performance of the methods in controlled conditions, having in mind that those performances will vary accordingly with the characteristics of each image. Tests were done with different spatial continuity image labels, different number of classes, different degrees of noise present in feature images and different sizes of training samples. In addition, the definition of some parameters in each method is also analysed in this section.

6.1.1 FSMLR with Laplacian Prior

The Laplacian prior enforces the sparsity of the classification methods trough a regularization parameter λ. This parameter has to be defined by the user. The choice of the best λ can be done by evaluating the OA over the test sets in the cross validation procedures. This can result in an intensive search that should be performed each time a classification task is needed. Depending on the goal of the analysis, some experiments were done using a fixed value for λ, like for example when the goal is to evaluate the performance of the method to different training set sizes.

6.1.1.1 $h(x)$ Linear

To analyse the performance of the FSMLR with a Laplacian prior and when using a linear input function ($h(x)$) over the features, synthetic hyperspectral datasets were generated according with the methodology described in section 5.1. The characteristics of the images tested in this subsection are described in table 6.1. A total of 36 images were created (2×6×3).

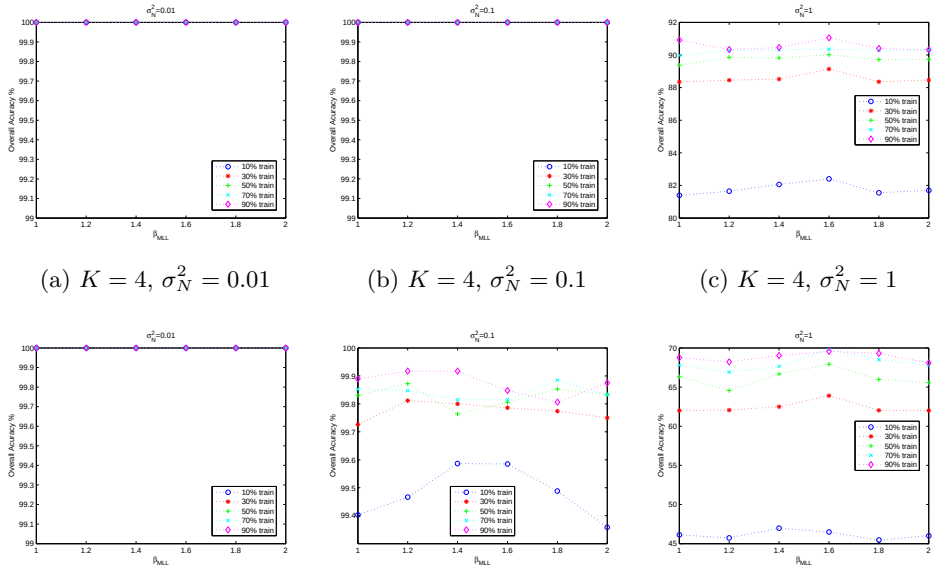
Table 6.1: Characteristics of simulated images to test FSMLR with $h(x)$ linear and Laplacian Prior

Number of Classes (K)	4 and 10
Spatial dispersion of label images (β_{MLL})	1, 1.2, 1.4, 1.6, 1.8 and 2
Noise variance of feature images (σ_N^2)	0.01, 0.1, 1

The variation of the parameters β_{MLL} and σ_N was made to evaluate the response of the FSMLR to the spatial continuity of label images and to the amount of noise present in the feature data.

The size of training samples is of great importance and usually it greatly affects the classification results. In this evaluation, tests were made using 10%, 30%, 50%, 70% and 90% of the image pixels as training set. Datasets with the characteristics described were simulated ten times for each set of parameters, in order to better evaluate the FSMLR classification results. A total of 1800 cases were evaluated (36 images × 5 training set sizes × 10 repetitions).

In this experiment, the sparseness parameter of FSMLR was set to $\lambda = 0.0005$. For each dataset, the OA was calculated. For each set of ten datasets with the same characteristics, the mean of all OA was taken. The graphic representation of the mean OA is presented in figure 6.2 as a function of the spatial continuity (β_{MLL}).



(a) $K = 4$, $\sigma_N^2 = 0.01$ (b) $K = 4$, $\sigma_N^2 = 0.1$ (c) $K = 4$, $\sigma_N^2 = 1$

(d) $K = 10$, $\sigma_N^2 = 0.01$ (e) $K = 10$, $\sigma_N^2 = 0.1$ (f) $K = 10$, $\sigma_N^2 = 1$

Figure 6.2: Overall accuracies as a function of the spatial continuity (β_{MLL}) of the label images for different training set sizes and different noise variance σ_N.

Comparing the results for different number of classes, it is possible to observe that the FSMLR performs better for smaller number o classes, namely when dealing with a hard classification problem ($\sigma_N^2 = 1$). When the σ_N^2 values are lower, there is no significant difference between the results for $K = 4$ and $K = 10$.

Globally, as expected, the higher the values of noise variance in the simulated hyperspectral feature images, the lower the value of OA. The increase of noise variance highly affects the performance of FSMLR method. It also should be noted that when the noise variance values are low (σ_N^2=0.01 and 0.1), the OA are always 100% or very close.

With respect to the variation of spatial continuity of image labels (β_{MLL}), there is no significant change in OA. The spatial heterogeneity of label images does

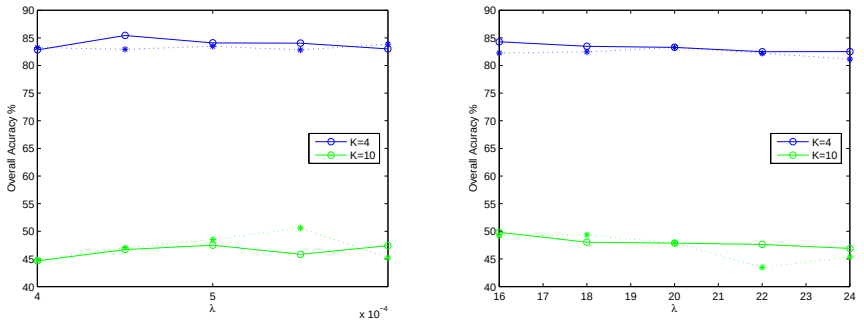
not seem to affect the performance of the FSMLR.

Regarding the size of training sets used and when lower noise variance is present there is also no significant difference in OA. However, when a higher noise variance is considered, the use of smaller training sets degrades the performance of FSMLR. This behaviour is frequent in real problems where normally there are high levels of noise in images. In those situations, a higher training set size helps achieving better performance for all classification methods.

Setting $\sigma_N^2 = 1$, corresponding to a signal-to-noise ratio below one, results in a hard classification problem. And as been observed, lower values for σ_N^2 do not affect significantly the results of FSMLR. For that reason, in the following experiments only images with this level of noise were tested.

In order to evaluate the influence of the sparseness parameter λ, experiments were done using 10% of image pixels as training set of images with 4 and 10 classes, and $\beta = 1$ and 2. The choice of the sparseness parameters was made in a way to test very small values and higher values. In this way, λ took values between 0.0004 and 0.0006 by steps of 0.00005; and between 16 and 24, by steps of 2.

Figure 6.3: Overall accuracies as function of sparseness parameter (λ), for 4 and 10 classes and $\beta_{MLL} = 1$ and 2 (lines and dotted lines, respectively).



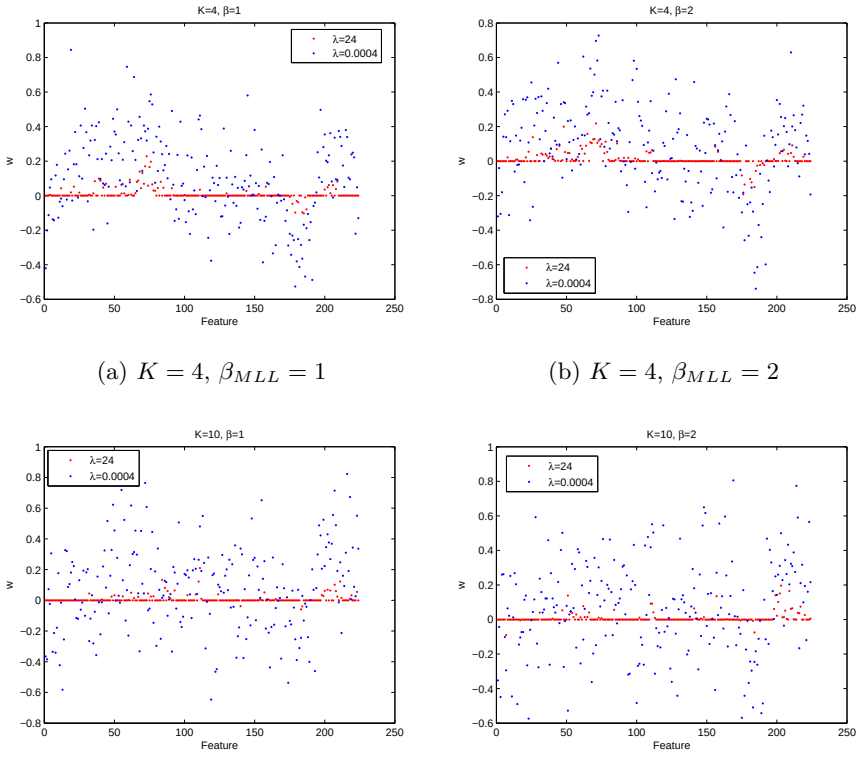
Analysing the OA presented in figure 6.3 it is possible to observe that the

variation of λ does not greatly affects the results in terms of OA. However, a lower value for λ produces results slightly better (about 1% or 2%). Regarding the case for 4 classes, the variation of the OA are between approximately 82% and 85%. When the number of classes is higher, and similarly to what was observed in figure 6.2, the OA are very low. This has to do with the small size of the training set used in this experiment (10%). Nevertheless, the behaviour of the classifier as function of the variation of λ is similar to what happens with 4 classes: there is no evidence of great variations in OA related to the choice of a high or low value for λ.

The OA is not the only quality measurement a classifier. When dealing with large dimension datasets, such as hyperspectral images, it is also important to evaluate the generalization capacity of the classifier and this can be done by analysing the sparseness of the classifier. As mentioned before, the λ parameter is a parameter that controls the sparsity of the FSMLR classification. A feature is selected whenever the correspondent weight is non zero. The number of significant features selected by each classifier is then the number of non-zero feature weights.

Figure 6.4 shows the features weights vector estimated by the FSMLR for $\lambda = 0.0004$ and 24. From this figure it is easily observed that a higher value for λ largely decreases the number of features selected by the FSMLR method, improving in this way the sparsity of the classifier.

Recall that the difference between the OA for $\lambda = 0.0004$ and $\lambda = 24$ is low (around 1% or 2% depending on the label image characteristics: number of classes and spatial continuity), while the number of features selected from each λ value is very different. Selecting a lower number of features results in classification algorithm with higher generalization capacity and less expensive computationaly.



(a) $K = 4$, $\beta_{MLL} = 1$

(b) $K = 4$, $\beta_{MLL} = 2$

(c) $K = 10$, $\beta_{MLL} = 1$

(d) $K = 10$, $\beta_{MLL} = 2$

Figure 6.4: Feature weights of different sparseness parameters (λ), for 4 and 10 classes and $\beta_{MLL} = 1$ and 2.

6.1.1.2 $h(x)$ RBF

To test the performance of FSMLR with a RBF function, datasets with the characteristics described in table 6.2 were generated.

Training sets with 10% of image pixels were used to evaluate the performance of FSMLR when using a RBF for $h(x)$. The use of this type of function enforces the user to define a parameter σ_h. In these experiments, several values for λ were also tested. Recall that λ controls the sparsity of FSMLR method. The parameter λ took values between 0.0004 and 0.0006, with increments of 0.00005

Table 6.2: Characteristics of simulated images to test FSMLR with $h(x)$ RBF and Laplacian Prior

Number of Classes (K)	4 and 10
Spatial dispersion of label images (β_{MLL})	1 and 2
Noise variance of feature images (σ_N^2)	1

and the σ_h took values between 0.48 and 0.72, with increments of 0.06. This resulted in 25 possible solutions for the classification of each image generated with the characteristics described in table 6.2. A total of 1000 cases were tested (4 image types $\times$ 25 classifiers $\times$ 10 repetitions).

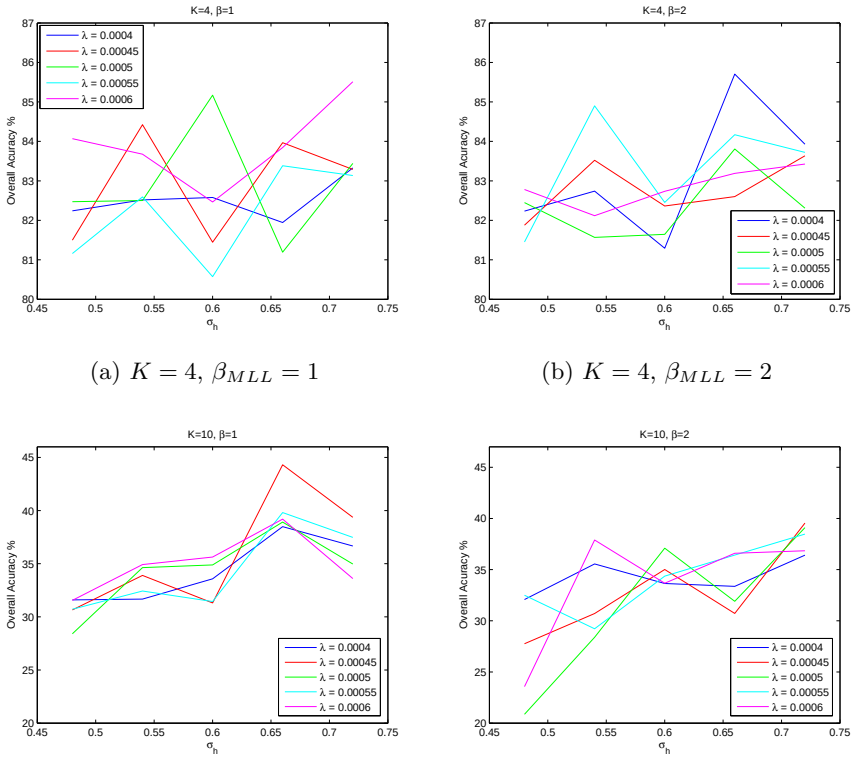
Figure 6.5 presents the OA produced by the FSMLR classification method under these conditions. The OA are presented as function of σ_h to search for a relation between the change of the parameter σ_h and the final OA. The five lines presented in each figure, correspond to the five values of λ tested.

When 4 classes are considered ($K = 4$), the OA results vary around 5% (from 80% to 85% for $\beta_{MLL} = 1$, and 81% to 86% for $\beta_{MLL} = 2$). However, these variations do not seem to be directly related with the variation of σ_h.

When 10 classes are considered ($K = 10$), it is possible to observe that, the increase of σ_h results in higher OA. But similarly to what happened in the $h(x)$ linear case, when only 10% of pixels are considered to train the classifier, the results of the FSMLR classifier are very poor (below 50%).

6.1.2 FSMLR with Jeffreys Prior

The Jeffreys prior is a parameter-free prior. In this way, it removes the sparseness parameter λ from the FSMLR while at the same time it is capable of



(a) $K = 4$, $\beta_{MLL} = 1$

(b) $K = 4$, $\beta_{MLL} = 2$

(c) $K = 10$, $\beta_{MLL} = 1$

(d) $K = 10$, $\beta_{MLL} = 2$

Figure 6.5: FSMLR classification with $h(x)$ RBF OA, as function of the variation of σ_h for 4 and 10 classes and $\beta_{MLL} = 1$ and 2.

keeping the method sparsity. This property of the Jeffreys prior facilitates the search for the best model to classify an image since there is no need to search fo the best λ, as was the case with the Laplacian prior. In this subsection, results of the use of the Jeffreys prior are shown in order to evaluate the sparseness capacity of this prior as well as the capacity to correctly classify a hyperspectral image. The results using the Jeffreys prior are also compared with those from the Laplacian prior.

Table 6.3: Overall accuracy of FSMLR using different training sets, with $h(x_i)$ Linear and $K = 4$, using a Laplacian and a Jeffreys prior.

		Size of training set				
		10%	20%	30%	40%	50%
$\beta = 1$	Laplacian Prior	83.74%	87.55%	89.27%	90.02%	90.34%
	Jeffreys Prior	78.75%	85.00%	86.95%	88.26%	89.05%
$\beta = 2$	Laplacian Prior	83.36%	87.62%	89.54%	89.70%	90.45%
	Jeffreys Prior	79.15%	84.75%	87.29%	87.94%	88.98%

6.1.2.1 $h(x)$ Linear

To evaluate the method performance depending on the size of the training sets, tests were made using 10%, 20%, 30%, 40% and 50% of image pixels as training set, when a linear $h(x_i)$ was considered. The image labels analysed were generated using $\beta_{MLL} = 1$ and $\beta_{MLL} = 2$.

Tables 6.3 and 6.4 present the OA obtained, considering four and ten classes, respectively. For each case, experiments were carried out using a Laplacian prior (setting $\lambda = 0.0005$) and the Jeffreys prior. Tables 6.3 and 6.4 present both OA results.

In both cases, four and ten classes, the OA from the Jeffreys prior are lower than with the Laplacian prior.

Considering the case with $K = 4$, these differences go from around 5% to 1%, depending on the size of training set used. The higher the training set size is, the smaller is the difference in the OA values between the Laplacian and Jeffreys prior. Likewise to what was observed earlier, the use of a larger training set highly improves the OA of the classification. The differences in OA when using

Table 6.4: Overall accuracy of FSMLR using different training sets, with $h(x_i)$ Linear and $K = 10$, using a Laplacian and a Jeffreys prior.

		Size of training set				
		10%	20%	30%	40%	50%
$\beta = 1$	Laplacian Prior	46.80%	57.11%	62.68%	65.08%	67.37%
	Jeffreys Prior	45.29%	53.53%	58.82%	61.53%	63.42%
$\beta = 2$	Laplacian Prior	48.27%	57.49%	62.42%	65.10%	67.58%
	Jeffreys Prior	45.04%	54.48%	58.50%	61.19%	62.79%

10% or 50% of pixels to train go from 7% to 10%.

When ten classes are considered, the OA obained with the Laplacian prior are also higher than with the Jeffreys prior, from around 5% to 2%, depending on the training set size use, identically to what was observed for four classes. However in this case, these OA differences are smaller when smaller training set sizes are considered, unlike to what happened with four classes. Globally, the OA in this case are much smaller than when four classes were used. Similarly to what was observed in previous sections, when only 10% of the image pixels were taken as training set, the OA are very small, below 50%. When the training set size increases, the OA also increases (near 20%) but never reaches the levels achieved in the problem with four classes.

Concerning the sparseness promoted by each prior, table 6.5 presents the number of selected features by each prior, in the case where four classes were considered.

The OA from Jeffreys prior revealed to be lower than with the Laplacian prior. However, analysing the sparsity of both priors, it is possible to observe that the Jeffreys prior produces a sparseness solution highly reducing the number of features selected to perform the classification. This improves the generalization

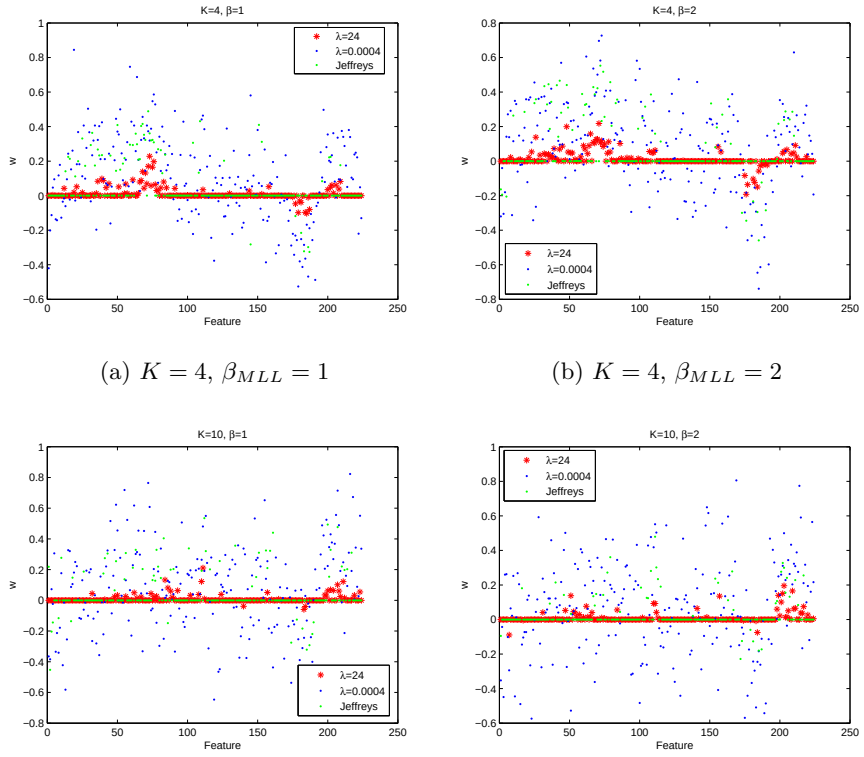
Table 6.5: Number of significant features selected (from 224) by each prior, whith $h(x_i)$ linear and $K = 4$.

	Size of training set				
	10%	20%	30%	40%	50%
Laplacian Prior	224	223	223	223	223
Jeffreys Prior	98	27	157	146	127

capacity of the classification and reduces the computational effort.

Like observed in section 6.1.1.1, a higher value for λ conducts a smaller number of selected features. To better evaluate the sparsity of Jeffreys prior, results from section 6.1.1.1 (where several values for λ were tested to evaluate the sparsity of the Laplacian prior) are now compared to the results from Jeffreys prior in the same conditions (with 10% of image pixels as training set).

The feature weight vector estimated by the FSMLR when using a Laplacian prior with $\lambda = 0.0004$ and 24, and with the Jeffreys prior are presented in figure 6.6. In this figure it is evident that the Jeffreys prior produces more sparse solutions than when the Laplacian prior is used with a small value for λ. However, when higher values for λ are considered, the Laplacian prior improves the sparseness achieved with the Jeffreys prior. Nevertheless, it is important to note that the use of the Laplacian prior enforces the user to define the λ value. This task may not be easy for the user if he does not have sensibility for this problem. The Jeffreys prior do not need any parameter to be defined by the user and even in this context can achieve sparse solutions competitive with the Laplacian prior.



(a) $K = 4$, $\beta_{MLL} = 1$

(b) $K = 4$, $\beta_{MLL} = 2$

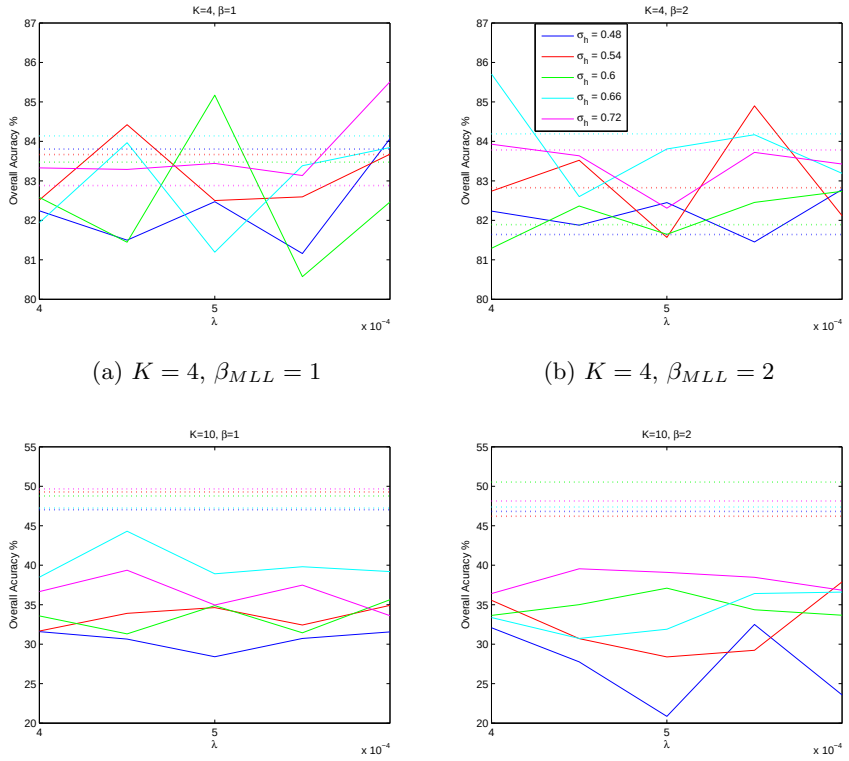
(c) $K = 10$, $\beta_{MLL} = 1$

(d) $K = 10$, $\beta_{MLL} = 2$

Figure 6.6: Feature weights generated by Laplacian prior (for different sparseness parameters, λ) and Jeffreys prior, for 4 and 10 classes and $\beta_{MLL} = 1$ and 2.

6.1.2.2 $h(x)$ RBF

To analyse the performance of the Jeffreys prior, when a RBF is used in $h(x)$, tests were carried out in the same conditions as the ones in section 6.1.1.2: training sets with 10% of image pixels, image labels with four and ten classes, $\beta_{MLL} = 1$ and 2. Since a RBF has to be defined in $h(x)$, there is the σ_h parameter that has to be defined by the user. In a similar way to what was done in section 6.1.1.2, σ_h was made to vary from 0.48 to 0.72, by steps of 0.06.



(a) $K = 4$, $\beta_{MLL} = 1$

(b) $K = 4$, $\beta_{MLL} = 2$

(c) $K = 10$, $\beta_{MLL} = 1$

(d) $K = 10$, $\beta_{MLL} = 2$

Figure 6.7: FSMLR classification with $h(x)$ RBF OA, as function of the variation of λ and with reference lines for Jeffreys prior (dotted lines), for 4 and 10 classes and $\beta_{MLL} = 1$ and 2.

Figure 6.5 showed no direct variation of OA with σ_h values. For this reason, the results of OA presented in figure 6.7 are shown as function of λ, and include a reference line of the OA produced by the Jeffreys prior. In this way it is easier to compare the OA of both priors, for different values of σ_h and λ.

In the case of four classes, the OA produced by the Jeffreys prior competes with the OA from the Laplacian priors for different values of λ, having indeed higher values of OA in the majority of cases for various values of σ_h.

When ten classes are considered, the use of the Jeffreys prior improved the OA

achieved with the Laplacian prior. Recall that the OA of FSMLR with Laplacian priors when using a small training set were very poor, below 40%. The Jeffreys prior improved these results in 10% to 20%, depending on the σ_h considered.

Note that the values used for λ in these experiments are very low, producing solutions not very sparse, but with higher OA than if a higher value for λ was used. This is a very good indicator of the good performance of the Jeffreys prior.

6.1.3 Segmentation with MRF

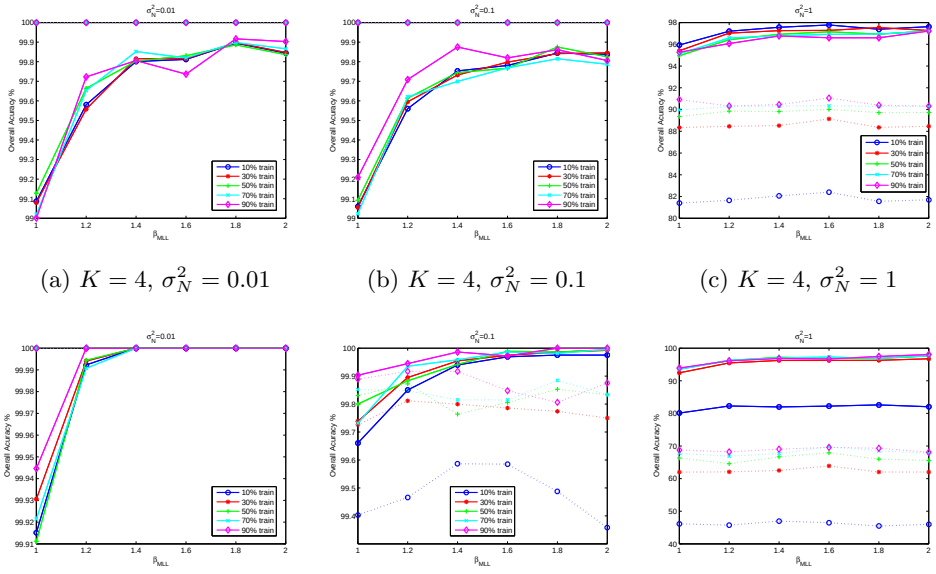
The application of the FSMLR to classify an hyperspectral image can be interpreted as step in the segmentation procedure presented in this thesis. In this section results of segmentation process are presented. The results are presented in two parts depending on the type of $h(x)$ used – linear or RBF. The results of the two priors, Laplace and Jeffreys, are presented together to a better comparison of results. The majority of results presented here are based in the experiments performed in sections 6.1.1 and 6.1.2 and can thus be interpreted as the conclusion of the process.

In the segmentation process there is a parameter that should be defined by the user in the α-Expansion algorithm. This smoothness parameter, β, has to do with the spatial heterogeneity of label images. In this set of experiments over synthetic images, β was set to be equal to β_{MLL} and variations of ± 0.1 over β_{MLL} were also tested.

6.1.3.1 Laplacian and Jeffreys Prior with $h(x)$ Linear

Based on the experiments performed in section 6.1.1.1, figure 6.8 presents the comparison of OA of FSMLR classification with a Laplacian prior with $\lambda =$

0.0005 and of MRF segmentation process, as function of β_{MLL} for 4 and 10 classes. Dotted lines corresponds to the OA for the FSMLR classification and solid lines for the MRF segmentation, with $\beta = \beta_{MLL}$. The graphics are displayed for different values of feature noise: $\sigma = 0.01, 0.1$ and 1. Different training set sizes were also considered. In some plots, the OA of the FSMLR classifier (dotted lines) are nearly 100%, and the lines are thus not visible in those plots.



(a) $K = 4$, $\sigma_N^2 = 0.01$ (b) $K = 4$, $\sigma_N^2 = 0.1$ (c) $K = 4$, $\sigma_N^2 = 1$

(d) $K = 10$, $\sigma_N^2 = 0.01$ (e) $K = 10$, $\sigma_N^2 = 0.1$ (f) $K = 10$, $\sigma_N^2 = 1$

Figure 6.8: Segmentation overall accuracies as function of spatial continuity (β_{MLL}) of the label images for different training set sizes and different noise variance σ_N.

In the case of $K = 4$ (Figures 6.8a, 6.8b and 6.8c) and for larger values of β_{MLL} and $\sigma_N^2 = 0.01$ or 0.1, the results from FSMLR classifier and MRF segmentation are similar. When the noise increases, the MRF outperforms the FSMLR by over 5%. As expected for low values of β_{MLL}, the performance of the MRF segmentation is slightly worse. However, the segmentation method

clearly outperforms the FSMLR classification when the noise is high ($\sigma_N^2 = 1$). The different sizes used for the training set do not seems to affect the results, except for the case of $\sigma_N^2 = 1$, where the use of a smaller training set degrades the performance of the FSMLR.

In the case of $K = 10$ (Figures 6.8d, 6.8e and 6.8f), the results for $\sigma_N^2 = 0.01$ and $\sigma_N^2 = 0.1$ are very similar for both methods. For $\sigma = 0.1$, it is nevertheless possible to see a small improvement of the OA achieved with MRF segmentation. Once again, when higher noise in the feature image is considered ($\sigma_N^2 = 1$), MRF segmentation clearly outperforms FSMLR classifier, by over 30%.

Likewise to the analysis performed in section 6.1.1.1, also with the MRF segmentation no significative differences were observed when low values for σ_N^2 are considered. For that reason, the following experiments were taken using a fixed value for $\sigma_N^2 = 1$.

The MRF segmentation performance was also evaluated when different priors (Laplacian and Jeffreys) in the feature density estimation procedure are considered. In these experiments, the influence of different training set sizes is also analysed using training set sizes of 10%, 20%, 30%, 40% and 50% of image pixels. The smoothness parameter of α-Expansion algorithm was set to be equal to β_{MLL}. Tables 6.6 and 6.7 present the OA of the MRF segmentation for $\beta_{MLL} = 1$ and 2, for 4 and 10 classes, respectively.

Considering the case of four classes, and independently of the β_{MLL} used, the performances in terms of OA is very similar for both priors used in the densities estimation step. The differences in the OA for both prior are minimum. Regarding the size of training sets used, it is observed that the increase of the training size once again improves the OA. It is also interesting to note that the difference between using 20% or 50% sizes of training sets produces very similar OA. The main difference happens when the training size is changed from 10%

Table 6.6: Overall accuracy of MRF segmentation using different training sets, with $h(x_i)$ Linear and $K = 4$, using a Laplacian and a Jeffreys prior.

		Size of training set				
		10%	20%	30%	40%	50%
$\beta_{MLL} = 1$	Laplacian Prior	96.85%	98.38%	98.79%	98.88%	98.94%
	Jeffreys Prior	96.65%	98.15%	98.44%	98.63%	98.81%
$\beta_{MLL} = 2$	Laplacian Prior	98.51%	99.13%	99.32%	99.32%	99.33%
	Jeffreys Prior	97.62%	98.61%	99.05%	99.10%	99.07%

to 20%.

Tests with ten classes revealed lower OA than with four classes, similarly to what was observed in the FSMLR classification problems. However, the improvement achieved by the segmentation procedure was very good (around 30%). In the case of ten classes and $\beta_{MLL} = 1$ the use of Jeffreys prior produced lower OA than the Laplacian one for all training set sizes, except when 10% of pixels were considered. The differences in the OA between both priors are around 2%. When $\beta_{MLL} = 2$, these differences are higher, varying from 16% to 4%, depending on the size of training set. When lower training sets are used, the differences between the OA of both priors become higher. Increasing the size of the training sets, smoothes the differences between the OA for both priors. In either ways, the OA achieved with 50% of pixels as training samples produced very good results for segmentation OA, comparing with the OA achieved by the FSMLR classification (see table 6.4).

Regarding the choice of β parameter from the α-Expansion algorithm, results with $\beta = \{\beta_{MLL} - 0.1; \beta_{MLL}; \beta_{MLL} + 0.1\}$ are presented in table 6.8. The sizes of training sets considered were 10%, 50% and 90% of image samples. Image

Table 6.7: Overall accuracy of MRF segmentation using different training sets, with $h(x_i)$ Linear and $K = 10$, using a Laplacian and a Jeffreys prior.

		Size of training set				
		10%	20%	30%	40%	50%
$\beta = 1$	Laplacian Prior	70.36%	88.36%	92.36%	93.67%	93.97%
	Jeffreys Prior	72.67%	84.76%	89.79%	91.89%	91.62%
$\beta = 2$	Laplacian Prior	83.45%	93.74%	95.63%	95.14%	96.42%
	Jeffreys Prior	66.87%	84.02%	89.16%	90.58%	92.26%

labels having 4 and 10 classes were tested.

The case of $K = 4$ do not show any significative variation in the OA promoted by the change in β values. When ten classes are considered and the size of training set is small (10%), the variation in the OA reaches 5%. When the size of training set increases, the difference between OA decrease.

6.1.3.2 Laplacian and Jeffreys Prior with $h(x)$ RBF

This final section of tests with synthetic images analyses the performance in terms of the segmentation algorithm OA when a RBF function over the features is considered in the densities estimation step with the FSMLR algorithm.

As seen previously, in sections 6.1.1.2 and 6.1.2.2, once a RBF function is considered there is an extra parameter to be defined by the user – σ_h. The definition of this parameter could influence the segmentation results, so this parameter should be considered in the analysis of segmentation OA. The type of prior used in the FSMLR method for estimation the features densities should alse be taken in account, as well as the smoothness parameter from the α-

Table 6.8: Overall accuracies for the proposed segmentation method for different values of β.

		$k=4$			$k=10$		
Training set size:		10%	50%	90%	10%	50%	90%
	$\beta = 0.9$	96.26%	98.91%	98.96%	66.18%	95.27%	95.85%
$\beta_{MLL} = 1$	$\beta = 1$	96.33%	98.92%	99.48%	71.76%	94.99%	94.91%
	$\beta = 1.1$	96.55%	98.93%	98.82%	68.54%	94.49%	94.77%
	$\beta = 1.9$	98.63%	99.36%	99.38%	89.59%	96.61%	97.40%
$\beta_{MLL} = 2$	$\beta = 2$	98.49%	99.27%	98.96%	89.38%	97.56%	97.54%
	$\beta = 2.1$	98.75%	99.27%	99.38%	87.68%	97.51%	94.88%

Expansion algorithm.

Regarding these three variables in the whole segmentation procedure, first the influence of changing the β parameter from the α-Expansion algorithm is considered. Then, the analysis will focus on the influence of the type of prior used and the variation in σ_h values.

Similarly to sections 6.1.1.2 and 6.1.2.2, all experiments were done considering 10% of image pixels as training samples, and the label images were generated using $\beta_{MLL} = 1$ and $\beta_{MLL} = 2$.

Table 6.9 presents the OA of segmentation algorithm setting $\beta = \beta_{MLL} - 0.1$; β_{MLL}; $\beta_{MLL} + 0.1$ for images with 4 classes. The OA from the FSMLR classification method used to estimate the features densities is also presented in order to facilitate the analysis of the improvement achieved with the segmentation procedure.

As can be observed, the variation of β results in small variations in OA, around

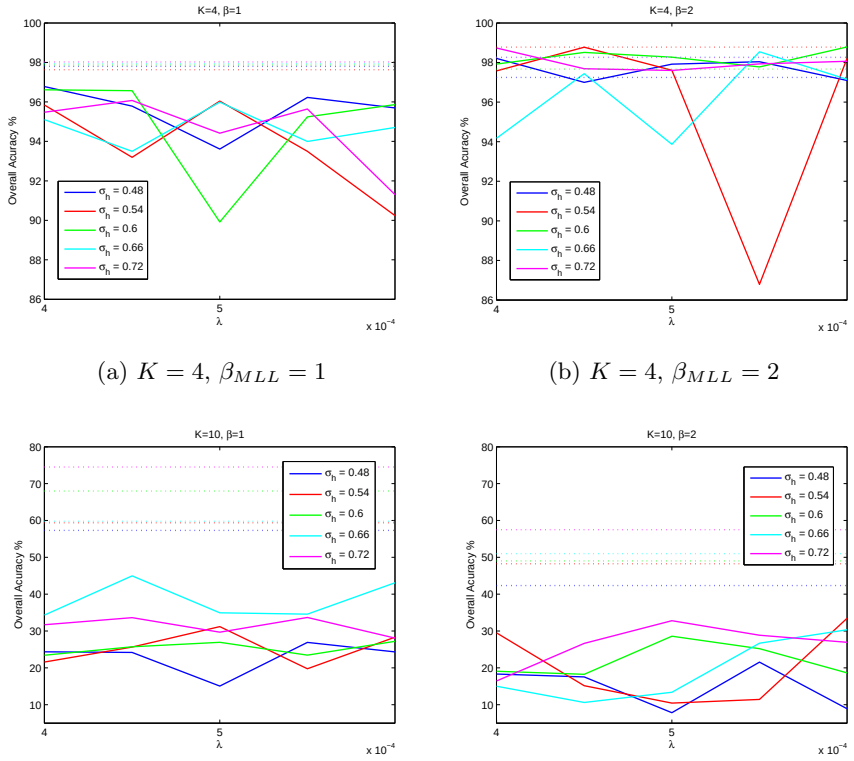
Table 6.9: Overall accuracies using a RBF kernel in the estimation of class densities, for the proposed segmentation method and FSMLR classification, using 10% of pixels as training data.

	$\beta = 0.9$	$\beta = 1$	$\beta = 1.1$	FSMLR
$\beta_{MLL} = 1$	96.27%	96.91%	97.23%	83.53%
	$\beta = 1.9$	$\beta = 2$	$\beta = 2.1$	FSMLR
$\beta_{MLL} = 2$	97.88%	97.82%	97.77%	83.77%

2% for $\beta_{MLL} = 1$ and no significant variation for $\beta_{MLL} = 2$. Like what happened in the Linear case, also here, the improvement achieved by the segmentation procedure was around 14%.

Figure 6.9 presents the OA for the segmentation of images with 4 and 10 classes, and $\beta_{MLL} = 1$ and $\beta_{MLL} = 2$, considering different values for σ_h and different priors. Dotted lines correspond to the OA achieved with the Jeffreys prior, solid lines correspond to the Laplacian prior, with λ varying from 0.0004 to 0.0006 by steps of 0.00005.

Analysing the case where $K = 4$ and when $\beta_{MLL} = 1$, the OA of the segmented images using the Jeffreys prior outperform the ones from the Laplacian prior independently of the σ_h or λ values considered. The major differences between both priors happens for $\sigma_h = 0.6$ and $\lambda = 0.0005$; $\lambda = 0.0005$ and $\sigma_h = 0.54$ and 0.72. In these cases, the differences are around 8%. In the remaining cases, the differences are between 4% and 1%.
When $\beta_{MLL} = 2$, the differences between OA promoted by each prior are very small except in the case where $\sigma_h = 0.66$ is considered and the case where $\lambda = 0.00055$ and $\sigma_h = 0.54$, where the OA for the Laplacian prior is around 10% below the OA from the Jeffreys prior.



(a) $K = 4$, $\beta_{MLL} = 1$

(b) $K = 4$, $\beta_{MLL} = 2$

(c) $K = 10$, $\beta_{MLL} = 1$

(d) $K = 10$, $\beta_{MLL} = 2$

Figure 6.9: Segmentation OA, with $h(x)$ RBF as function of the variation of λ and with reference lines for Jeffreys prior (dotted lines), for 4 and classes and $\beta_{MLL} = 1$ and 2.

Considering the 10 classes case ($K = 10$), the results from the segmentation that included the Laplacian prior in the estimation of features densities step, were very low for either cases ($\beta_{MLL} = 1$ and $\beta_{MLL} = 2$). The Jeffreys prior produced better results specially when $\beta_{MLL} = 1$. Nevertheless, the results of OA segmentation for 10 classes are again (likewise results presented in other sections) lower than the OA for images with lower number of classes.

6.2 Indian Pines dataset

This section presents the results of the application of the FSMLR classification algorithm and the segmentation with MRF over the benchmarked dataset Indian Pines. The characteristics of this dataset are described in section 5.2. The learning process of each method was performed over subsets of the training set and the OA presented here were measured in the independent test set described in 5.2.

The Indian Pines dataset has been widely used to test and evaluate several hyperspectral image processing algorithms, including classification and segmentation ones. The analysis of the results in this section is for that reason done together with the comparison of results from methods presented in other works.

6.2.1 FSMLR with Laplacian Prior

In this subsection, the results are presented using the complete set of bands and discarding the 20 noisy bands. The objective was to observe the influence of a coarse feature selection on the classifiers performance.

The tuning process for each classifier was done by first dividing the training set considered into a subset with approximately 10% of training samples, which was used to learn the classifier, and the remaining 90% used to compute an estimate of the OA. This process was repeated 20 times in order to obtain the parameter that maximizes the OA in the remaining training sets.

The OA presented are a result of the application of the optimal set of parameter to the independent test set. These results are compared with the ones achieved by Camps-Valls et. al [26].

Table 6.10: Best λ and number of support vectors (SV) used with $h(x)$ linear.

	220 Bands			200 Bands		
	10%	20%	100%	10%	20%	100%
λ	16	22	18	16	18	18
SV	28	22	85	24	39	37

6.2.1.1 $h(x)$ Linear

The learning process of the FSMLR classifier when $h(x)$ is linear was performed using different training set sizes to evaluate the response of the classifier to this variable. Training sets with 10%, 20% and 100% of the original training set were considered.

Since one is dealing with a Laplacian prior, the definition of the λ parameter was also an object of analysis. Tests were done using $\lambda = 16$, 18, 20, 22 and 24.

Table 6.10 presents the parameters that provided the highest OA for each experimental scenario and respective number selected features (number of support vectors). As one can see, there is in fact a large reduction on the number of features needed to built the classifier.

The OA in table 6.11 are presented to compare results between the use of 220 spectral bands and without the noisy bands. It can be observed that the improvement in OA due to the coarse selection is not significant. In some cases, the use of all 220 bands gives better results than with 200 bands (without the noisy bands). However, it is worth noting that the differences are not significant in both cases.

In order to better evaluate these results, a comparison was made with the results obtained using other kernel-based methods in the same dataset [26]. Although

Table 6.11: Results with $h(x)$ linear using 10%, 20% and the complete training set.

	10%	20%	100%
220 bands	76.55%	79.69%	85.77%
200 bands	75.57%	81.60%	85.24%

Table 6.12: Comparison of the FSMLR classification with the results from [26].

	SMLR Linear	LDA [26]
220 bands	85.77%	82.32%
200 bands	85.24%	82.08%

there were some limiting factors in the practical application of the proposed method, due to limitations in Matlab processing capacity, the results obtained are very encouraging. The performance of FSMLR linear proved to be superior to Linear Discriminant Analysis (LDA) [26] as it is summarised in table 6.12.

6.2.1.2 $h(x)$ RBF

Adopting a RBF for $h(x)$ lead, the user to define an extra parameter, the σ_h value. Since the parameters are defined using the cross validation method, the addition of one more parameter turns the computational process heavier. For that reason, experiments were carried out using subsets with 10%, 20% and 50% of the training samples to learn FSMLR the classifier with a RBF function and a Laplacian prior. The values tested for these parameters were $\lambda = 0.0004, 0.00045, 0.0005, 0.00055, 0.0006$ and $\sigma = 0.48, 0.54, 0.6, 0.66, 0.72$.

In table 6.13 an example of the tuning process over one subset of 20% of the

Table 6.13: OA of a FSMLR classification with $h(x)$ RBF and a Laplacian prior, using 20% of the training samples.

$\lambda \backslash \sigma_h$	0.48	0.54	0.6	0.66	0.72
0.0004	85.06%	85.53%	85.37%	84.93%	84.40%
0.00045	85.14%	85.56%	85.32%	84.98%	84.38%
0.0005	85.24%	85.50%	85.32%	84.82%	84.27%
0.00055	85.45%	85.43%	85.22%	84.66%	84.43%
0.0006	85.48%	85.40%	84.95%	84.56%	84.38%

training samples and 220 spectral bands is showed. In this example we take $\sigma = 0.54$ as the best σ. Then we fixed this value and looked for the best λ running 20 times the same procedure for different subsets of the same size. The same process was carried out to achieve the best λ and σ_h using 10% and 50% of the training set.

Table 6.14 presents the OA results considering different sizes of training sets and different number of spectral bands considered. Once again, it is patent the

Table 6.14: OAs of FSMLR with $h(x)$ RBF and Laplacian prior, using 10%, 20% and 50% of training samples.

	10%	20%	50%
220 bands	82.93%	87.12%	90.12%
200 bands	84.98%	86.73%	90.52%

influence of the training set size used to train the classifier in the final results. The OAs measured in the test set increases with the size of the training set used.

Regarding the influence of the number of spectral bands used, it is not detectable a significant influence in using or not the 20 noisy bands, likewise the linear case. However, it is worth pointing out that the use of a higher number of spectral bands increases the computational complexity of the FSMLR classification method. Also, considering that those bands are noisy bands the use of them will not add important information to the classification process. For these reasons, in the experiments reported in the next sections, the 20 noisy bands will be excluded from the experiments and only 200 spectral bands will be considered.

Comparing the results of FSMLR classification with the ones from a SVM-RBF classification (from [26]), they are very similar. The values presented in table 6.15 for SVM-RBF are approximate values extracted from graphical data presented in figure 6 of [26]. Although for RBF kernels our method did not outperform the method used in [26], the sparsity of FSMLR can be an advantage for large datasets.

Table 6.15: Comparison of the FSMLR classification with the results from [26].

	FSMLR RBF(50%)	SVM-RBF (50%) [26]
220 bands	90.12%	$\simeq 91\%$
200 bands	90.52%	$\simeq 91\%$

6.2.2 FSMLR with Jeffreys Prior

This subsection presents the results of using a Jeffreys prior in the FSMLR classification method. The OA achieved with this prior is compared with the OA obtained using the Laplacian prior. Throughout this section only 200 spectral bands were used.

6.2.2.1 $h(x)$ Linear

Considering a linear function, experiments were carried out with subsets of training set with 10%, 20%, 50% and 100% of training samples.

Table 6.16 presents the OA validated in the independent test set, for each training set size used to learn the classifier. Once again, the OA results increase with the size of training set used.

The OA produced by the Jeffreys prior are lower than the ones from FSMLR classification with a Laplacian prior. The differences vary from 1% to 3%, depending on the size of the training set used, being smaller when a bigger training set is used.

Table 6.16: OA of FSMLR classification using 10%, 20%, 50% and the complete training set, with $h(x_i)$ Linear, using a Laplacian and a Jeffreys prior.

	Size of training set			
	10%	20%	50%	100%
Laplacian Prior	75.78%	78.92%	85.00%	86.44%
Jeffreys Prior	72.99%	76.33%	83.26%	85.24%

However, looking to the level of sparsity of each prior, the Jeffreys prior uses less number of features, producing in this way more sparse solutions than the Laplacian prior.

Table 6.17 shows the number of selected features by each prior. It can be observed that for all sizes of the training set, the Jeffreys prior selects around half the number of features that the Laplacian prior.

Recall that we are dealing with hyperspectral images where the dimension is very

Table 6.17: Number of significant features selected by each prior, whith $h(x_i)$ linear.

	Size of training set			
	10%	20%	50%	100%
Laplacian Prior	34	49	71	105
Jeffreys Prior	18	27	39	51

high and for that reason a method that produces good OA with less number of features selected is preferable since it will have a significant impact in the computational burden. Also, the Jeffreys prior avoids the search for the best sparsity parameter (λ) needed in the Laplacian prior. For these two reasons, and although the OA produced by the Jeffreys are slightly lower, the Jeffreys prior may be a better option.

6.2.2.2 $h(x)$ RBF

To test the use of a Laplacian prior when a RBF is considered to estimate the features densities, training sets with 10%, 20% and 50% of training samples were used.

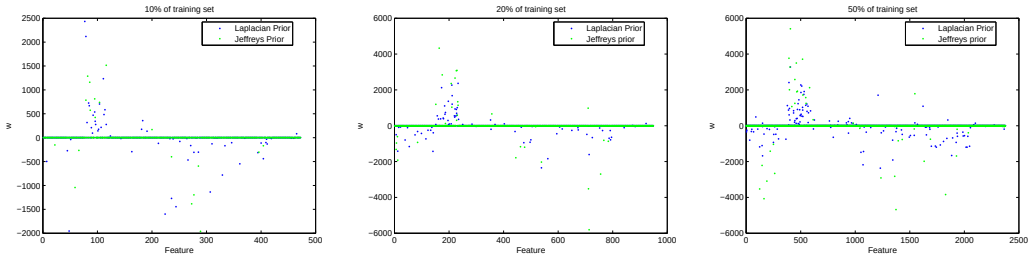
The OA for each training set size, and both priors are presented in table 6.18. Once more, the OA increases with the size of training sets used. The use of the RBF function in the FSMLR improved the OA when compared to the linear case (see 6.16), achieving with 50% of training samples better results than the linear case with the complete test set.

Comparing the performance of Jeffreys and Laplacian priors, also with a RBF function the Laplacian prior produces better OA. Here, the differences between the OA of each prior vary between 5% and 2%, being the highest difference for

Table 6.18: OA of FSMLR classification using 10%, 20% and 50% of training set, with $h(x_i)$ RBF, using a Laplacian and a Jeffreys prior.

	Size of training set		
	10%	20%	50%
Laplacian Prior	83.70%	86.44%	90.61%
Jeffreys Prior	78.77%	84.72%	88.64%

the smaller training set size used.



(a) 10% (b) 20% (c) 50%

Figure 6.10: Feature weigths estimated from Laplacian and Jeffreys priors, for different sizes of training set.

In terms of sparseness, once again the Jeffreys prior retrieves solutions with a lower number of significative features, as can be seen in figure 6.10.

6.2.3 Segmentation with MRF

This section presents the OA obtained by the MRF based segmentation algorithm proposed in chapter 4. Experiments were carried out in the same conditions used in previous sections (6.2.1 and 6.2.2). Results are analysed together for Jeffreys and Laplacian prior for each type of function defined in

$h(x)$ (linear and RBF).

6.2.3.1 Laplacian and Jeffreys Prior with $h(x)$ Linear

Experiments with $h(x)$ linear were carried out with 10%, 20%, 50% and the complete training set. In the GCαExpansion method a $\beta = 1.5$ was defined when the complete training set is considered, and $\beta = 4$ for subsets of training data. Table 6.19 shows the OA for the MRF segmentation method adopting a linear function in the feature estimation procedure.

Table 6.19: OA of MRF segmentation using 10%, 20%, 50% and the complete training set, with $h(x_i)$ Linear, using a Laplacian and a Jeffreys prior.

	Size of training set			
	10%	20%	50%	100%
Laplacian Prior	86.05%	89.45%	89.69%	95.60%
Jeffreys Prior	86.18%	88.58%	90.43%	95.66%

By the analysis of table 6.19, it is possible to observe that the Jeffreys prior achieves competitive results with Laplacian prior. The performance of the classifier was found to be nearly independent of the prior used for all training sets tested, the variations on the OA are minimal. The increase in size of the training set results in better OA for all methods, as seen before.

Comparing the results from the segmentation method with the ones presented in sections 6.2.2.1 and 6.2.1.1 it is evident the importance of the addition of spatial information by the segmentation process. The improvement in OA promoted by the segmentation process goes from 5% to 13%, being the majority of these improvements around 10%, independently of the size of training set. It is also interesting to note that segmentation algorithm using only 10% of

training samples in the learning process, achieves the same OA that the FSMLR classification method with the complete training set.

6.2.3.2 Laplacian and Jeffreys Prior with $h(x)$ RBF

In these experiments, training sets with 10%, 20% and 50% of the original training set were considered. In the GCαExpansion method a $\beta = 4$ was considered to model to spatial distribution of pixels in the segmentation procedure. The OA for the MRF segmentation method adopting a RBF function in the feature estimation procedure are shown in Table 6.20.

Table 6.20: OA of MRF Segmentation using 10%, 20% and 50% of training set, with $h(x_i)$ RBF, using a Laplacian and a Jeffreys prior.

	Size of training set		
	10%	20%	50%
Laplacian Prior	92.11%	94.62%	97.86%
Jeffreys Prior	89.84%	95.07%	96.71%

In comparison with the linear case, the use of an RBF function improved the OA, likewise observed in the FSMLR classification analysis.

Regarding the type of prior used, the results from the Jeffreys prior are competitive with the Laplacian revealing once again that the OA of the segmentation process seems to be independent from the prior used. The differences observed between OA in table 6.18 are now even smaller.

The improvement promoted by the use of spatial information is also significant, varying between 7% and 11%. Using only 10% of training samples in the segmentation process gives us higher accuracies than using 50% of training

samples in the FSMLR classification.

The introduction of Jeffreys prior was able to keep the good performance of the Bayesian segmentation method proposed. It should be noted that with this prior there is no need for searching the parameter that best controls sparsity, something that has to be done with the Laplacian prior. This reduces significantly the time needed to classify the image. The reduction is of the order of the number of sparsity parameters to be tested. Moreover, the sparsity achieved by the FSMLR when using a Jeffreys prior is higher than with the Laplacian prior.

6.3 Pavia datasets

This section presents the application of both the FSMLR classification and the MRF based segmentation methods presented in chapter 4, to three urban hyperspectral images over the town of Pavia, Italy. The details of these datasets were described in section 5.3.

Experiments were carried out to access the efficiency of the presented classification and segmentation procedures when compared to recent algorithms developed for processing hyperspectral imagery, presented in [90] like SVM, MRF based characterization with Discriminant Analysis Feature Extraction and Extended Morphological Profiles.

Since the goal was to compare the results from the FSMLR classification and MRF based segmentation algorithms here presented with the results from [90], experiments were carried out in the same conditions of that work. For that reason some methods were not applied to all images. In the following sections this aspect will be detailed.

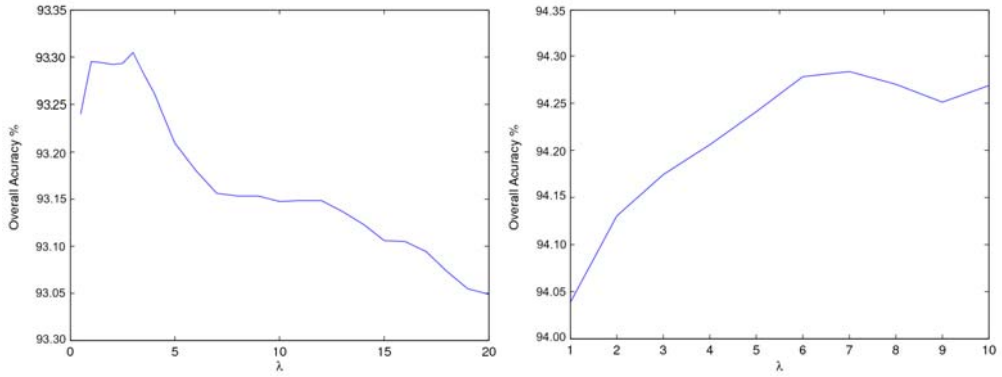
6.3.1 FSMLR with Laplacian Prior

To analyse the performance of the Laplacian prior in the Pavia datasets, several values for λ were considered, depending on the type of function defined for $h(x)$. Experiments were carried out using the complete training sets defined in section 5.3 and with subsets of this set. All the OA presented were measured in the independent test set.

6.3.1.1 $h(x)$ Linear

Dataset 1 and *Dataset 3* were used to test the performance of FSMLR with linear function. Experiments were carried out with the complete training set and OA measured in the independent test set.

The λ values tested for *Dataset 1* were $\lambda = \{0.5 : 0.5 : 4; 5 : 20\}$. For *Dataset 3*, λ values from 1 to 10 were tested. Figure 6.11 shows the OA as function of λ. As it can be observed, the variation in OA promoted by the sparsity parameter is not significant in both datasets. The variations in OA are less than 0.5% in both cases.



(a) *Dataset1* (b) *Dataset3*

Figure 6.11: FSMLR classification OAs, as function of λ, with $h(x)$ linear.

Comparing the results of *Dataset 1* with the SVM with poly kernel (96.03%, from [90]), it is observed that the FSMLR does not outperform the SVM. However, it is important to recall that a linear function is being used, while results from [90] were obtained with a poly function which can improve the results since it has higher flexibility to adapt to data.

Regarding the results achieved with *Dataset 3*, one can say that theseare good results, considering that the results presented for this dataset in [90] (OA in the order of 97%) were achieved using methods that provide integration of spatial details into the classification based on comparison of the spectra.

In order to evaluate the performance of FSMLR classification method when smaller training samples are considered, 5 subsets from the whole training set of *Dataset 1*, with 10%, 20%, 40%, 60% and 80% of each class were randomly selected to learn the classification algorithm. A five-fold cross-validation method was used to access the parameters of the FSMLR algorithm. The OA were evaluated on the complete validation set.

Training set size	10%	20%	40%	60%	80%
Overall Accuracy	85.21%	87.01%	88.25%	88.70%	89.38%

Table 6.21: OA of the FSMLR classification with linear mapping, using different subsets of the training set.

Table 6.21 summarizes the results obtained. Although the OA increase with the size of training sets used, it is possible to conclude that the FSMLR classification method is not drastically affected by the high dimensionality of training samples, and good generalization performance is obtained – with only 10% of training samples, 85% of accuracy is reached.

6.3.1.2 $h(x)$ RBF

When a RBF kernel is considered, the computational complexity increases and the process of finding the $h(x)$ parameters that gives the highest OA becomes a very slow task when a large training set is used.

Considering the *Dataset 2*, a subset with 10% of each class present in the training samples was randomly selected to learn the classification algorithm, and the OA was measured over the complete validation set. With 10% of the training set, an OA of 84.88% was achieved. Results over the same dataset in [90], but using the complete training set were of 80.99% produced by a SVM with RBF kernel and 85.22% with Extended Morphological Profiles. Comparing the results of classifiers that use only spectral information, the FSMLR with RBF function outperformed the SVM with RBF kernel in 4%, using only 10% of training samples. It is also of value to note that the accuracy achieved by the FSMLR with RBF function is competitive with the results from Extended Morphological Profiles where spatial information is added to the process.

Training set size	10	20	40	60	80	100
FSMLR-RBF	86.08	89.82	92.23	93.29	94.27	94.84
SVM-RBF [90]	93.85	94.51	94.51	94.71	95.36	95.29

Table 6.22: OAs of the FSMLR classification with RBF function, using different subsets of the training set, and results from [90].

The FSMLR classification method using RBF kernels, was also evaluated using *Dataset 1*. Subsets with 10, 20, 40, 60, 80 and 100 samples of each class were randomly selected from the training set, and the OA were calculated over the complete test set. The results are presented in table 6.22. From that table it is possible to observe that in this dataset, and regardless of the size of the training set, the FSMLR classification does not outperform the SVM-RBF algorithm

used in [90]. However, with the increase of training set, the differences between both methods tends to decrease. It should be pointed out the high generalization capacity of FSMLR: with only 40 training pixels per class, more than 90% OA is reached.

6.3.2 FSMLR with Jeffreys Prior

As seen in previous experiments, the Jeffreys prior showed to be competitive with Laplacian prior both in terms of OA and sparsity. Also, the Jeffreys prior has the advantage of being a parameter free prior.

This section present results of OA over the same datasets tested in section 6.3.1. The OA results are compared for both priors and also the analysis of the sparsity achived by each prior is done both for $h(x)$ linear and RBF.

6.3.2.1 $h(x)$ Linear

When a linear function is adopted to $h(x)$, tests with Jeffreys prior were carried out with *Dataset 1* and *Dataset 3* considering the complete training set to learn the classifier, as it was done in section 6.3.1. OA were measured in the independent test set.

The FSMLR classification of *Dataset 1* with the Jeffreys prior resulted in a OA of 95.15%. Recall that the best OA achieved by the Laplacian prior in the same conditions was 93.30%, for a $\lambda = 3$ (see figure 6.11). So, in terms of OA, the Jeffreys prior outperformed the FSMLR linear classification with Laplacian prior and approximated the results presented in [90] with a SVM Poly kernel (OA of 96.03%).

In respect to the classification methods sparsity, it is possible to observe in figure 6.12 the number of significant features selected by each prior. Note that λ was

set to 3 in the Laplacian prior. From this figure it is possible to observe the

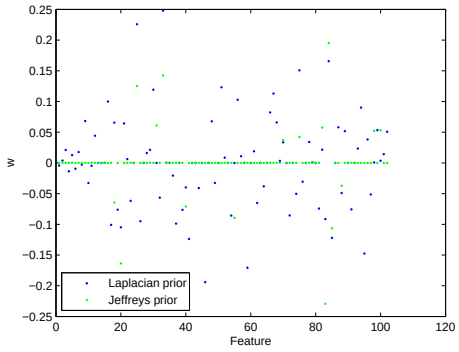


Figure 6.12: Feature weights for Jefreys and Laplacian ($\lambda = 3$) priors, with $h(x)$ linear.

higher sparsity achieved by the Jeffreys prior. The number of weights estimated by the this prior with non-zero value is much less than by the Laplacian prior. This capacity together with the fact that it does not require a search the best λ, and the good performance in terms of OA, make the FSMLR Linear classification with Jeffreys prior an excellent option.

Considering the classification problem with FSMLR Linear and with Jeffreys prior, *Dataset 3* experiments were carried with the complete training set to learn the classifier and the independent test set to evaluate the OA. The result of this classification resulted in an OA of 96.95%. This result outperformed the result achieved with the Laplacian prior in 2% and showed to be competitive with the results from [90] (97%) achieved with methods that integrate spatial information.

In terms of sparsity, comparing with the Laplacian prior with $\lambda = 7$ (the parameter that returned the best OA), the Jeffreys prior once again gave solutions with higher level of sparsity. Figure 6.12 shows the weights estimate for both priors.

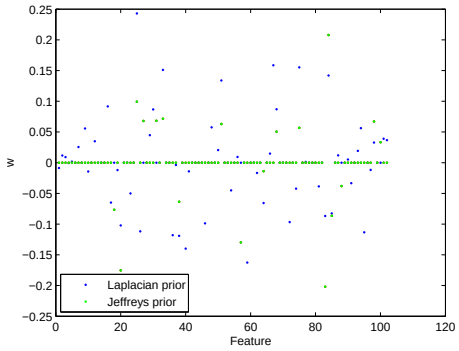


Figure 6.13: Feature weights for Jefreys and Laplacian ($\lambda = 3$) priors, with $h(x)$ linear.

Observing 6.12, it is easily noticeable the higher number of weights with non-zero values for the Laplacian prior, revealing that this prior uses a higher number of features to execute the classification process. This will evidently increase the computational cost of this task. Moreover, this type of prior enforces the user to define the sparseness parameter, which can become a computational demanding task, specially when high dimension datasets are considered.

6.3.2.2 $h(x)$ RBF

The Jeffreys prior was also used to classify when a RBF function is considered for $h(x)$. When a RBF function is used, it is necessary to define the σ parameter. This was done empirically through a 5-fold cross-validation procedure in the training set.

Considering *Dataset2*, and similarly to what was done in the Laplacian prior case, only 10% of training set samples were used to learn the classifier. The OA were measured in the test set. In this case, an OA of 83.78% was achieved. Recall that in the Laplacian prior, the OA achieved was of 84.43%, using a $\lambda = 0.001$. Once more the Jeffreys prior proves to be competitive with the Laplacian one,

without the need to define any parameter, and also outperformed the results presented in [90] over the same dataset with the complete training set to learn the classifier with a SVM with RBF kernel.

The analysis of sparsity can be done analyzing the number of features set to zero by each prior. Figure 6.14 shows the weight values estimated by each prior. Once more it is visible the higher degree of sparsity promoted by the Jeffreys

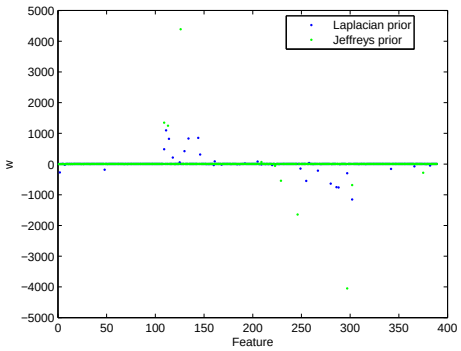


Figure 6.14: Feature weights for Jefreys and Laplacian ($\lambda = 0.001$) priors, with $h(x)$ RBF.

prior improving the generalization capacity. This property of the Jeffreys prior, together with the high OA achieved and the fact that there is no need to seek the best λ, makes the Jeffreys prior a good choice.

6.3.3 Segmentation with MRF

This section presents the results of the segmentation procedure using a MRF. The addition of spatial information is expected to improve the FSMLR classification results. However, it will also increase the complexity of the process, and consequently the processing time. The segmentation process enforces the definition of the parameter β, that controls spatial homogeneity of the label images. This parameter is adjusted empirically to maximize the OA measured

in the test set. Segmentation and classification results will be compared as well as a comparison with results from [90] will be addressed.

6.3.3.1 Laplacian and Jeffreys Prior with $h(x)$ Linear

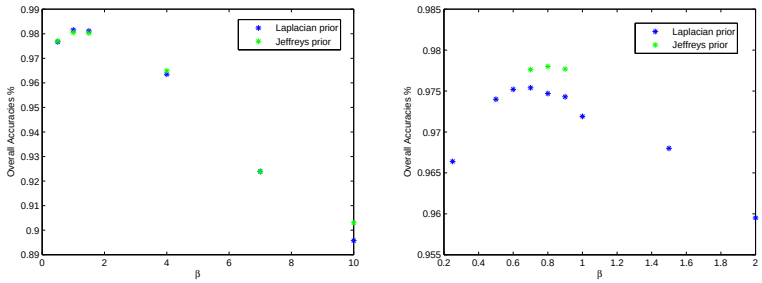
The performance of the segmentation method with linear function was analyzed with *Dataset 1* and *Dataset 3* using the complete training set to learn the segmentation algorithm, and the complete set of validation samples was used to access the OA (Table 6.23).

	Dataset 1	*Dataset 3*
MRF*Seg* Lap	98.18%	98.46%
MRF*Seg* Jef	98.05%	97.78%
Results from [90]	96.03%	97.27%

Table 6.23: OA of the MRF segmentation with linear mapping both for Laplacian and Jeffreys prior, and the results from [90], using the complete training set.

The results from [90] presented in table 6.23 for *Dataset 1* were achieved with a SVM with a Poly kernel. The results for *Dataset 3* are a product of a MRF-based spatial characterization where a discriminant analysis feature extraction was applied before in order to increase spectral separability. The application of the proposed segmentation method with a linear mapping managed to improve the results under the same conditions, without any pre-processing to increase the spectral separability, independently of the prior used.

The segmentation process implemented is dependent of the spatial parameter β. Several values for this parameter were tested to access the one that retrieved the best OA. Figure 6.15 shows the OA variation as function of the β parameter for both datasets. As can be observed in both figures, the behavior of the



(a) *Dataset 1* (b) *Dataset 3*

Figure 6.15: Segmentation OA values as function of spatial continuity parameter (β) for Jefreys and Laplacian priors, with $h(x)$ linear.

segmentation algorithm depends on the value of β choosen, but the type of prior used does not seem to highly influence the OA values.

To access the improvement promoted by the inclusion of spatial information when small samples are considered, experiments were carried out for *Dataset 1*, with 10%, 20%, 40%, 60% and 80% of each class from the training set. The OA values were evaluated on the complete validation set and are presented in table 6.24.

Once more it is evident the high improvement promoted by the usage of spatial

Training set size	10%	20%	40%	60%	80%
FSMLR Lap	85.21%	87.01%	88.25%	88.70%	89.38%
MRF Seg. Lap	94.03%	96.14%	95.85%	96.16%	96.75%

Table 6.24: OA of the FSMLR classification and MRF segmentation with linear mapping, using different subsets of the training set.

information together with spectral information. The OA increase vary from 9% to 7%. It is also important to note the high value of OA achieved with only 10% of the training samples, which demonstrates the high generalization capacity of

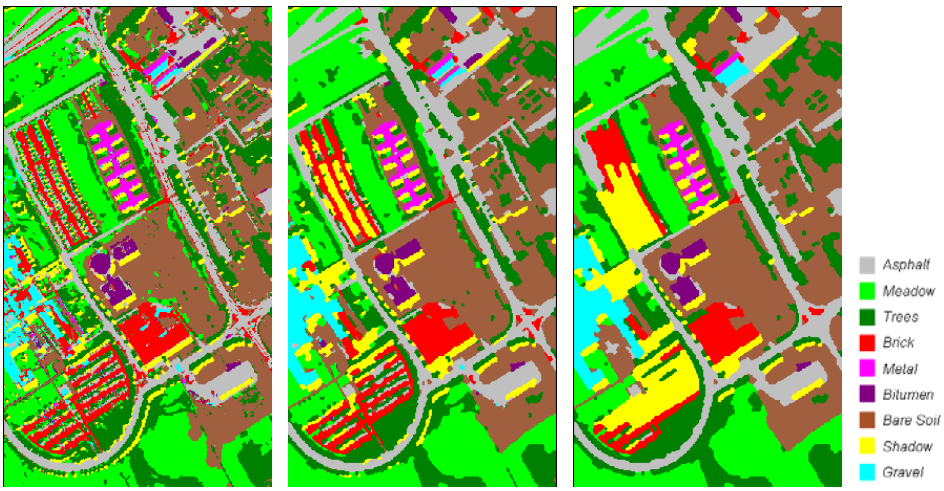
the method.

6.3.3.2 Laplacian and Jeffreys Prior with $h(x)$ RBF

The segmentation problem when a RBF function is used in the estimation of densities step is addressed with small subsets of training set from *Dataset 1* and with 10% of the training set from *Dataset 2*.

For both study cases several values of β were tested. The OA were evaluated in the independent test set. Recall that the β parameter controls spatial heterogeneity of label images. This parameter can be adjusted empirically using for example a cross validation procedure, or it can be defined in order to model the spatial dispersion of classes by the user, accordingly with the goal of segmentation.

Figure 6.16 shows the segmentation of *Dataset 2*, for different values of β ($\beta = 1$, 3.4 an 5), when a FSMLR with Laplacian prior was considered. As can be seen



(a) $\beta = 1$ (b) $\beta = 3.4$ (c) $\beta = 5$

Figure 6.16: Segmentation maps of Pavia Dataset 2, with RBF function.

in the three images of this figure, higher values of β produce maps with a higher

degree of homogeneity. This aspect can be of interest to the user, depending on the requirements of the image segmentation task.

Regarding the OA achieved with the segmentation process for *Dataset 2*, observing figure 6.17 it is visible the higher performance of the segmentation procedure for the Laplacian prior. Segmentation results in other datasets did not show such high differences on the OA resulting from the use of different priors. This can be due the low sparsity level considered in the Laplacian prior ($\lambda = 0.001$). Even so, results from the segmentation with Jeffreys prior are competitive with the results from [90] with algorithms that include spatial information. From this figure it is also visible the influence of the β parameter in the segmentation process.

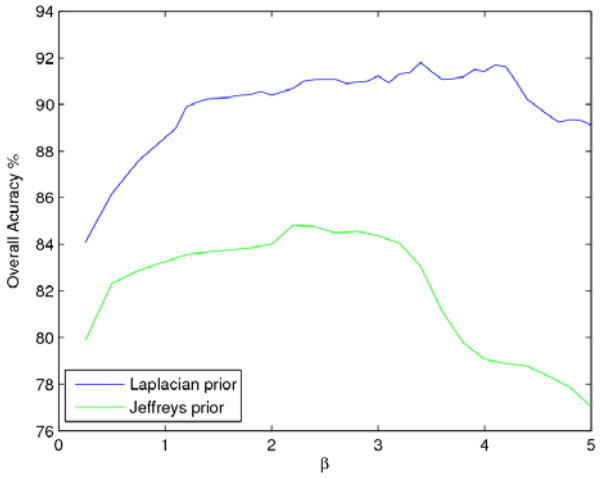


Figure 6.17: Segmentation OA for Laplacian and Jeffreys prior, with RBF function.

The MRF segmentation method proposed using RBF kernels in the class density estimation was also evaluated using the *Dataset 1*. Subsets with 10, 20, 40, 60, 80 and 100 samples of each class were randomly selected from the training set, and the OA were calculated over the complete test set. The results are presented in table 6.25, where it is possible to observe that, regardless of the size of the

training set, the MRF-Segmentation outperforms the SVM-RBF algorithm used in [90]. Also, the high improvement in OA promoted by the segmentation process should be noted analyzing the differences between the OA from the FSMLR classification and the MRF based segmentation. They vary between 3% and 11%, being higher when small training samples are considered.

Training set size	10	20	40	60	80	100
FSMLR-RBF	86.08	89.82	92.23	93.29	94.27	94.84
MRF*Seg* RBF	97.04	96.33	96.54	97.37	97.97	97.90
SVM-RBF [90]	93.85	94.51	94.51	94.71	95.36	95.29

Table 6.25: OA (%) of the MRF segmentation, using different subsets of the training set size, and results from [90].

The advantage of using a method that includes spatial information is well shown by the comparison of the OA achieved by both methods: with only 90 samples, the MRF based segmentation yielded an OA of 97.04%, while the SVM-RBF with the complete training set (5536 samples) achieved an OA of 96.45%.

In the experiments presented it is evident the high performance of the MRF segmentation process proposed. In all experiments the proposed segmentation algorithm outperformed results form other methods. The FSMLR method proposed for classification also performed well when compared to other methods. The choice of the input function $h(x)$ can have a significant influence in the results of classification. Good results were achieved both with Linear and RBF functions for $h(x)$. The usage of a RBF has the disadvantage of having to tune the σ parameter, but it can improve the results.

Both Laplacian and Jeffreys priors produce similar results in the FSMLR classification, and good results in terms of sparsity. However, since the Laplacian prior depends on the sparsity parameter λ, both values of spasrity and OA may

suffer big changes depending on the value of λ defined by the user. The search of the best λ can become a computational expensive task. The Jeffreys prior returned competitive results with the Laplacian prior, and does not need any parameter to be defined.

When the FSMLR method is incorporated in the segmentation process to estimate the features densities, the final result of the segmentation does not seem to depend on the type of prior, neither the type of function used in $h(x)$. This suggests the use of a linear function in $h(x)$ together with the Jeffreys prior, since it is a combination that does not use any parameter, resulting in a less expensive computational process.

Chapter 7

Conclusions

Hyperspectral images provide detailed information about spectral signatures, which would improve the discrimination between different land cover classes for the production of land cover maps. However, the high dimensionality of this type of data is a problem that pattern recognition algorithms have to deal with. Learning high dimensional densities from a limited number of training samples is a well known difficulty - the Hughes phenomenon.

In recent years, the wide availability of hyperspectral imagery, as well as other types of high dimensional data, has leaded the development of classification algorithms able to deal with high dimensional datasets. Discriminative algorithms are among the state-of-art in supervised image classification. Their ability to deal with small class distances, high dimensionality, and limited training samples has proved to be successful. SVM are probably the most well known example of a discriminative approach that retrieves very good results when applied to high dimensional data. SVM have shown excellent results in terms of computational cost, accuracy, robustness to noise and sparsity. When high dimensional datasets are considered, one of the most important properties of a classification algorithm is the capacity to produce sparse solutions. This will improve the generalization

capacity of the classifier, as well as the computational cost. Despite all these recent developments, classification algorithm for high dimensional datasets is still an area of intensive research.

A part of this thesis focuses in this research area. It presents a classification algorithm designed for efficiently deal with high dimensional datasets: the Fast Sparse Multinomial Logistic Regression (FSMLR) method [14]. FSMLR is based on SMLR [66], so it produces sparse solutions, but includes an iterative way for estimating the weights which turns the SMLR a faster and more efficient algorithm for the classification of hyperspectral images.
In addition to this improvement, this thesis also propose the use of an alternative prior to the one used in the original SMLR method. SMLR uses the Laplacian prior to improve the sparsity of the classifier. However, this prior needs the tuning of the sparsity parameter λ. When dealing with high dimensional datasets this process leads to high computational costs because it is done trough cross-validation methods. To avoid this, the use of the Jeffreys prior was proposed [17]. It is a parameter free prior which also gives good levels of sparsity, with no need for search for the best parameter.

The FSMLR method was applied to several hyperspectral images to access its efficiency when dealing with this type of data. Synthetic images and real hyperspectral images collected from AVIRIS and ROSIS sensors were used. When applying FSMLR classification algorithm one should choose the type of input function used. In this work linear and RBF functions were considered.
As expected, results proved that although the complexity of a RBF function may produce better OA than a linear input function, this complexity has the disadvantage of increasing the computational cost. In addition, an RBF function requires the tuning of a parameter which aggravates the algorithm time processing. Despite this, experiments carried out in synthetic images, linear functions achieved better results than with RBF functions, showing that simpler options

may achieve good results.

The sparsity of the classifier is a characteristic of extreme importance when dealing with high dimensional datasets. When using a Laplacian prior, the sparsity is controlled by a parameter which should be tuned by the user. In the experiments carried out, Jeffreys prior was able to produce competitive results with the use of a Laplacian prior in terms of OA with two advantages: no need for an intensive search for the sparsity parameter and retrieved higher levels of sparsity for similar OA accuracies. Higher sparsity results in a lower number of selected features necessary to the image classification which obviously reduces the computational effort.

When different sizes of training sets were considered, it was observed that, as expected, a higher number of training samples retrieved higher OA. However, in some experiments the good generalization capacity of FSMLR was patent when reduced size training sets achieved very good OA.

The use of benchmarked datasets allowed the comparison of the results achieved by FSMLR classification with results from state-of-art classification algorithms. Experiments with Indian Pines and Pavia datasets were performed in similar conditions to the ones published in [26] and [90], respectively.

Experiments with Indian Pines dataset, showed the quite satisfactory performance of FSMLR when compared to those from [26], where LDA and RBF SVM were used. With a linear input function, our method outperformed LDA; with a RBF function, FSMLR achieved approximately the same results that a RBF SVM.

Experiments over the Pavia datasets with a linear input function and Laplacian prior did not outperformed the results with a poly kernel presented in [90]. However, the results when using the Jeffreys prior revealed to be competitive with the ones presented in [90]. When a RBF input function was considered, the results from FSMLR outperformed the results with a RBF SVM from [90].

We may therefore state that FSMLR performance competes with state-of-art classification algorithms. Also, the use of the Jeffreys prior is desirable to the use of Laplacian because it is able to produce sparse solutions with no need for searching the optimum parameter and achieves similar OA to those from a Laplacian prior.

A way of improving the performance of discriminative classifiers (and others) consists in adding contextual information in the form of spatial dependencies, resulting in a segmented image. Image segmentation has been a widely studied problem in computer vision in several domains. Here again however, the application of segmentation algorithms to hyperspectral data is often difficult by the high dimensionality of the data. In this thesis we introduced a new Bayesian segmentation approach for hyperspectral images [15]. The Bayesian Hyperspectral Image Segmentation with Discriminative Class Learning methodology here presented enforces spatial dependencies by a Multi-Level Logistic (MLL) Markov-Gibbs prior. This density favours labelling in which neighbouring sites belong to the same class. The class densities were build on the FSMLR. Due its computationally efficiency and the property of yielding nearly optimum solutions, the α-Expansion graph cut based algorithm was adopted to compute the MAP segmentation.

The Bayesian segmentation method was applied to the hyperspectral datasets used to evaluate the FSMLR algorithm. In all cases, the use of contextual information lead to a substantial improvement of the OA results achieved with the discriminative classification. Very good values of OA were obtained (achieving 99% in some cases), which demonstrate the high potential of the use of both spatial and spectral information.

Since the segmentation algorithm is based on the class densities learned by the FSMLR, it was analysed the influence of the choice of the input function, as well as the type of prior used. Generally it was observed that the type of prior

considered (Laplacian or Jeffreys) does not greatly affects the OA segmentation results. This is an important fact to have in consideration because, as seen in the classification experiments, the Jeffreys prior conducts to good sparse solutions with no need for searching the best sparsity parameter. Regarding the type of input function used, RBF functions lead to higher values of OA.

The generalization capacity of the segmentation method should also be noticed: even when small training sets were considered, the proposed segmentation algorithm managed to achieve very good OA results.

When compared to recent techniques that also include spatial information along with the spectral information, our segmentation algorithm showed to be very competitive with them all. In fact the Bayesian segmentation method here proposed outperformed the methods presented in [90].

Trough the segmentation process, a parameter that controls the spatial dependency must be defined by the user. It was observed that this parameter highly influences the final result of segmentation. Although this can lead to extensive search for the parameter that retrieves higher OA, this flexibility of the segmentation algorithm can be of interest to the user, depending on the type of image to segment and the goal of segmentation.

As final remark, one should point out the good sparsity performance achieved with the use of a Jeffreys prior. The fact that this prior do not need the tuning for the parameter reduces the time processing of the classification, and therefore the segmentation process, while at the same time is able to achieve very good OA. Although the use of RBF function in the classification process could improve the results, the difference between the use of a linear and a RBF function in the segmentation process is not so significant. The final result of segmentation does not seem to depend either from the type of prior used. Therefore, this suggest the use of a linear input function together with a Jeffreys prior to achieve a segmentation process with good results of OA with no need for selection of

additional parameters besides the segmentation parameter.

This work presented contributions at several levels, including the proposal of new Bayesian hyperspectral segmentation algorithm. However, there is still room for improvement, namely by implementing accurate supervised learning of the model parameters and the development of semi-supervised techniques based on the FSMLR method proposed. Future work can be also developed to include different neighbourhood systems and implement an automatic routine to segment hyperspectral images.

Bibliography

[1] Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In Petrov B. N. and Csaki F., editors, *Proc. of the 2nd Int. Symp. on Information Theory*, pages 267–281, 1973.

[2] H. Bagan, Q.X. Wang, M. Watanabe, S. Kameyama, and Y.H. Bao. Land-cover classification using aster multi-band combinations based on wavelet fusion and som neural network. *Photogrammetric Engineering and Remote Sensing*, 74(3), 2008.

[3] T.V. Bandos, L. Bruzzone, and G. Camps Valls. Classification of hyperspectral images with regularized linear discriminant analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 47(3), 2009.

[4] R. Bellens, S. Gautama, L. Martinez Fonte, W. Philips, J.C.W. Chan, and F. Canters. Improved classification of vhr images of urban areas using directional morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 46(10), 2008.

[5] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[6] E. Bellucoa, M. Camuffob, S. Ferraria, L. Modeneseb, S. Silvestric, A. Maranib, and M. Marani. Mapping salt-marsh vegetation by multispec-

tral and hyperspectral remote sensing. *Remote Sensing of Environment*, 105(1), 2006.

[7] J.A. Benediktsson, J.A. Palmason, and J.R. Sveinsson. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions in Geoscience and Remote Sensing*, 43(3), 2005.

[8] A. Berge and A.S. Solberg. Improving hyperspectral classifiers: The difference between reducing data dimensionality and reducing classifier parameter complexity. pages 293–302, 2007.

[9] J. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 1980.

[10] J.M. Bioucas-Dias. Bayesian wavelet-based image deconvolution: a gem algorithm exploiting a class of heavy-tailed priors. *IEEE Transactions on Image Processing*, 15(4):937–951, 2006.

[11] H. Bischof, W. Schneider, and A.J. Pinz. Multispectral classification of landsat-images using neuralnetworks. *IEEE Transactions on Geoscience and Remote Sensing*, 30(3):482–490, 1992.

[12] E. Blanzieri and F. Melgani. Nearest neighbor classification of remote sensing images with the maximal margin principle. *IEEE Transactions on Geoscience and Remote Sensing*, 46(6), 2008.

[13] D. Böhning and B. Lindsay. Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, 40(4):642–663, 1988.

[14] J.S. Borges, J.M. Bioucas-Dias, and A.R.S. Marçal. Fast sparse multinomial regression applied to hyperspectral data. In *Image Analysis and*

Recognition, number 4142 in Lecture Notes in Computer Science, pages 700–709. Springer Berlin / Heidelberg, September, 2006 2006.

[15] J.S. Borges, J.M. Bioucas-Dias, and A.R.S. Marçal. Bayesian hyperspectral image segmentation with discriminative class learning. In *Pattern Recognition and Image Analysis*, number 4477 in Lecture Notes in Computer Science, pages 22–29. Springer Berlin / Heidelberg, June, 2007 2007.

[16] J.S. Borges, A.R.S. Marçal, and J.M. Bioucas-Dias. Evaluation of feature extraction and reduction methods for hyperspectral images. In Zbigniew Bochenek, editor, *Proceedings of the 26th EARSeL Symposium, New Developments and Challenges in Remote Sensing*, pages 266–274. Millpress, 2007.

[17] J.S. Borges, A.R.S. Marçal, and J.M. Bioucas-Dias. Hyperspectral image segmentation using fsmlr with jeffreys prior. In Derya Maktav, editor, *Proceedings of the 28th EARSeL Symposium, Remote Sensing for a Changing Europe*, pages 378–385. IOS Press, 2009.

[18] Y. Boykov and V. Kolmogorov. An experimental comparison of mincut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1124–1137, 2004.

[19] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[20] Yuri Boykov, Vivian S. Lee, Henry Rusinek, and Ravi Bansal. Segmentation of dynamic n-d data sets via graph cuts using markov models. In *In Proc. Medical Image Computing and ComputerAssisted Intervention*, pages 1058–1066, 2001.

[21] L. Bruzzone, M. Chi, and M. Marconcini. A novel transductive svm for semisupervised classification of remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(11), 2006.

[22] L. Bruzzone, Mingmin Chi, and M. Marconcini. Transductive svms for semisupervised classification of hyperspectral data. *IEEE IGARSS Proceedings*, 1.

[23] K.A. Budreski, R.H. Wynne, J.O. Browder, and J.B. Campbell. Comparison of segment and pixel-based non-parametric land cover classification in the brazilian amazon using multi-temporal landsat tm/etm+ imagery. *Photogrammetric Engineering and Remote Sensing*, 73(7), 2007.

[24] C. F. Caiafa, E. Salerno, A. N. Proto, and L. Fiumi. Blind spectral unmixing by local maximization of non-gaussianity. *Signal Process.*, 88(1), 2008.

[25] G. Camps Valls, T.V. Bandos Marsheva, and D. Zhou. Semi-supervised graph-based hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10), 2007.

[26] G. Camps-Valls and L. Bruzzone. Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 43(6):1351–1362, 2005.

[27] G.C. Cawley and N. L. C. Talbot. Gene selection in cancer classification using sparse logistic regression with bayesian regularisation. *Bioinformatics*, 22(19):2348–2355, 2006.

[28] "Chein-I Chang. *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*. Plenum Publishing Co., 2003.

[29] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[30] M.M. Chi and L. Bruzzone. An ensemble-driven k-nn approach to ill-posed classification problems. *Pattern Recognition Letters*, 27(4), 2006.

[31] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *Electronic Computers, IEEE Transactions on*, (3):326–334.

[32] N. Cristianini and J. Shawe-Taylor. *Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, 2000.

[33] G. Cross and A. Jain. *PAMI*, 5.

[34] Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

[35] O. Debeir, I. Van den Steen, P. Latinne, E. Wolff, and Ph. Van Ham. Spectral, spatial and contextual land cover classification using single and multiple classifiers. *Photogrammetric Engineering and Remote Sensing*, 68(6):597–605, 2002.

[36] F. Dell'Acqua, P. Gamba, A. Ferrari, J.A. Palmason, J.A. Benediktsson, and K. Arnason. Exploiting spectral and spatial information in hyperspectral urban data with high resolution. *IEEE Geoscience and Remote Sensing Letters*, 1(4):322–326, 2004.

[37] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using gibbs random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(1), 1987.

[38] R. Duda, Peter Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

[39] T. Duda and M. Canty. Unsupervised classification of satellite imagery: Choosing a good algorithm. *International Journal of Remote Sensing*, 23(11), 2002.

[40] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1974.

[41] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.

[42] M. Fauvel, J.A. Benediktsson, J. Chanussot, and J.R. Sveinsson. Spectral and spatial classification of hyperspectral data using svms and morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 46(11), 2008.

[43] M.A.T. Figueiredo. Adaptative sparseness using jeffreys prior. In *Advances in Neural Network Information Processing Systems 14*, pages 697–704. MIT Press, 2001.

[44] M.A.T. Figueiredo. Bayesian image segmentation using wavelet-based priors. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition - CVPR2005*, 2005.

[45] M.A.T. Figueiredo and A.K. Jain. Bayesian learning of sparse classifiers. In *In 2001 Conference on Computer Vision and Pattern Recognition (CVPR 2001*, pages 35–41. IEEE Press, 2001.

[46] M.A.T. Figueiredo and R.D. Nowak. Wavelet-based image estimation: an empirical bayes approach using jeffreys's noninformative prior. *IEEE Transactions on Image Processing*, 10(9):1322–1331, 2001.

[47] E. Fix and J. L. Hodges. Discriminatory analysis: Nonparametric discrimination:consistency properties. Technical Report Project 21-49-004, Report Number 4, USAF School of Aviation Medicine, 1951.

[48] K.S. Fu and T.S. Yu. *Statistical Pattern Classification Using Contextual Information*. Wiley, 1980.

[49] P. Gege, D. Beran, W. Mooshuber, J. Schulz, and H. van der Piepen. System analysis and performance of the new version of the imaging spectrometer rosis. In *First EARSeL Workshop on Imaging Spectroscopy*, 1998.

[50] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[51] A.F.H. Goetz, G. Vane, J.E. Solomon, and B.N. Rock. Imaging spectrometry for earth remote sensing. *Science*, 228:1147–1153, 1985.

[52] M. Govender, K. Chetty, V. Naiken, and H. Bulcock. Comparison of satellite hyperspectral and multispectral remote sensing imagery for improved classification and mapping of vegetation. *WaterSA*, 34(2):147–154, 2008.

[53] R.O. Green. Aviris operational characteristics. Technical report, 1994.

[54] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621.

[55] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning - Data Mining, Inference and Prediction*. Springer Series in Statistics. Springer, New York, 2001.

[56] Lothar Hermes, Student Member, and Joachim M. Buhmann. A minimum entropy approach to adaptive image polygonization. *IEEE Transactions on Image Processing*, 12:1243–1258, 2003.

[57] Thomas Hofmann, Jan Puzicha, and Joachim M. Buchmann. Unsupervised texture segmentation in a deterministic annealing framework. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8), 1998.

[58] G. Hughes. On the mean accuracy of statistical pattern recognizers. *Information Theory, IEEE Transactions on*, (1):55–63.

[59] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[60] Anil K. Jain and Farshid Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, (12):1167–1186.

[61] John R. Jensen. *Remote Sensing of the Environment: an Earth Resources Perspective*. Prentice Hall, Inc, 2000.

[62] Junmo Kim, John W. Fisher, Anthony Yezzi, Mjdat etin, and Alan S. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Trans. Image Processing*, 14:1486–1502, 2005.

[63] M. Koch, J. Inzana, and F. El-Baz. Applications of hyperion hyperspectral and aster multispectral data in characterizing vegetation for water resources studies in arid lands. In *The Geological Society of America Annual Meeting*, volume 37, 2005.

[64] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004.

[65] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

[66] B. Krishnapuram, L. Carin, M.A.T. Figueiredo, and A.J. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):957–968, 2005.

[67] Balaji Krishnapuram, Lawrence Carin, and Alexander J. Hartemink. Joint classifier and feature optimization for comprehensive cancer diagnosis using gene expression data. *J. Comput. Biol*, 11:227–242, 2004.

[68] Balaji Krishnapuram, Er J. Hartemink, Lawrence Carin, Mrio A. T. Figueiredo, and Senior Member. A bayesian approach to joint feature selection and classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1105–1111, 2004.

[69] D.A. Landgrebe. *Signal Theory Methods in Multispectral Remote Sensing*. John Wiley and Sons, Inc., New Jersey, 2003.

[70] K. Lange. *Optimization*. Springer Texts in Statistics. Springer-Verlag, New York, 2004.

[71] Neil Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems 15*, pages 609–616. MIT Press, 2003.

[72] M. Lewicki and T. Sejnowski. Learning overcomplete representations. *Neural Computaton*, 12:337–365, 2000.

[73] S. Z. Li. *Markov random field modeling in computer vision*. Springer-Verlag, London, UK, 1995.

[74] Thomas M. Lillesand, Ralph W. Kiefer, and Jonathan W. Chipman. *Remote Sensing and Image Interpretation*. Wiley.

[75] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[76] P. Mantero, G. Moser, and S.B. Serpico. Partially supervised classification of remote sensing images through svm-based probability density estimation. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3), 2005.

[77] A. R. S. Marçal and J. S. Borges. Estimating the natural number of classes on hierarchically clustered multi-spectral images. *Lecture Notes in Computer Science*, (3656):447–455, 2005.

[78] A. R. S. Marçal, J. S. Borges, J. A. Gomes, and J. P. Costa. Land cover update by supervised classification of segmented aster images. *International Journal of Remote Sensing*, 26(7):1347–1362, 2005.

[79] A. R. S. Marçal and Luisa Castro. Hierarchical clustering of multi-spectral images using combined spectral and spatial criteria. *IEEE Geoscience and Remote Sensing Letters*, 2(1):59–63, 2005.

[80] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, pages 530–549.

[81] G.J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Probability and Mathematical Statistics . Applied Probability and Statistics. John Wiley & Sons, New York, 1992.

[82] S.K. Meher, B. Uma Shankar, and A. Ghosh. Wavelet-feature-based classifiers for multispectral remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 45(6), 2007.

[83] T.P. Minka. A comparison of numerical optimizers for logistic regression. Technical report, Carnegie Mellon University, 2003.

[84] A. Muller, A. Hausold, and P. Strobl. Hysens - dais / rosis imaging spectrometers at dlr. In *SPIE, Proc. 8th Int. Symp. on Remote Sensing*, 2001.

[85] J. Munoz Mari, L. Bruzzone, and G. Camps Valls. A support vector domain description approach to supervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 45(8), 2007.

[86] J.M.P. Nascimento and J.M. Bioucas Dias. Does independent component analysis play a role in unmixing hyperspectral data? *IEEE Transactions on Geoscience and Remote Sensing*, 43(1), 2005.

[87] J.M.P. Nascimento and J.M.B. Bioucas Dias. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4), 2005.

[88] R. Neher and A. Srivastava. A bayesian mrf framework for labeling terrain using hyperspectral imaging. *IEEE Transactions on Geoscience and Remote Sensing*, 43(6), 2005.

[89] Robert D. Nowak and Mario A. T. Figueiredo. Unsupervised progressive parsing of poisson fields using minimum description length criteria. In *in IEEE Int. Conf. on Image Proc. — ICIP '99*, pages 26–29, 1999.

[90] A. Plaza, J. Benediktsson, J. Boardman, L. Brazile, J.and Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, J. Tilton, and G. Trianni. Advanced processing of hyperspectral images. *IEEE IGARSS Proceedings*, IV.

[91] A. Plaza, P. Martinez, J. Plaza, and R. Perez. Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3), 2005.

[92] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Number 37 in TAM. Springer-Verlag, New York, 2000.

[93] T. Randen and J.H Husoy. Filtering for texture classification: A comparative study. *IEEE Trans. Pattern Anal. Mach. Intell.*, (4):291–310.

[94] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, January 2004.

[95] J. Rissanen. A universal prior for integers and estimation by the minimum description length. *The Annals of Statistics*, 11(2):416–431, 1983.

[96] Y. Dan Rubinstein and Trevor Hastie. Discriminative vs informative learning. In *In Proc. Third Int. Conf. on Knowledge Discovery and Data Mining*, pages 49–53. AAAI Press, 1997.

[97] L. Samaniego, A. Bardossy, and K. Schulz. Supervised classification of remotely sensed imagery using a modified k-nn technique. *IEEE Transactions on Geoscience and Remote Sensing*, 46(7), 2008.

[98] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[99] B.U. Shankar, S.K. Meher, A. Ghosh, and L. Bruzzone. Remote sensing image classification: A neuro-fuzzy mcs approach. pages 128–139, 2006.

[100] E. Sharon, A. Brandt, and R. Basri. Segmentation and boundary detection using multiscale intensity measurements. pages I–469–I–476 vol.1.

[101] Jianbo Shi and Jitendra Malik. Normalized cut and image segmentation. Technical report, Berkeley, CA, USA, 1997.

[102] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B*, 36(1):111–147, 1974.

[103] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 3rd edition, 2006.

[104] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[105] Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.

[106] R. Trias Sanz, G. Stamon, and J. Louchet. Using colour, texture, and hierarchial segmentation for high-resolution remote sensing. *Journal of Photogrammetry and Remote Sensing*, 63(2), 2008.

[107] B.C.K. Tso and P.M. Mather. Classification of multisource remote sensing imagery using agenetic algorithm and markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 37:1255–1260, 1999.

[108] M. Unser. Texture classification and segmentation using wavelet frames. *Image Processing, IEEE Transactions on*, (11):1549–1560.

[109] USGS.

[110] V. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.

[111] Y. Weiss and W.T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13:2173–2200, 2001.

[112] Yair Weiss. Segmentation using eigenvectors: a unifying view. In *In International Conference on Computer Vision*, pages 975–982, 1999.

[113] J.R. Welch and K.G. Salter. A context algorithm for pattern recognition and image interpretation. *IEEE Trans. Syst. Man Cybernet*, 1:24–30, 1971.

[114] P. Williams. Bayesian regularization and pruning using a laplace prior. *Neural Computation*, 7:117–143, 1995.

[115] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11), 1993.

[116] B. Xu and P. Gong. Land-use/land-cover classification with multispectral and hyperspectral eo-1 data. *Photogrammetric Engineering and Remote Sensing*, 73(8), 2007.

[117] J.S. Yedidia, W.T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, pages 689–695, 2000.

[118] Q. Yu, P. Gong, N. Clinton, G. Biging, M. Kelly, and D. Schirokauer. Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery. *Photogrammetric Engineering and Remote Sensing*, 72(7), 2006.

[119] P. Zhong and R. Wang. Learning sparse crfs for feature selection and classification of hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 46(12), 2008.

[120] Song Chun Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 18:884–900, 1996.

[121] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.