Vniversitat de València

Kernel Feature Extraction Methods for Remote Sensing Data Analysis

Author: Emma Izquierdo-Verdiguier

Advisors: Luis Gómez-Chova
Gustau Camps-Valls

VNIVERSITAT
DE VALÈNCIA

Facultat de Física

PhD Thesis:
Doctoral studies in
Remote Sensing

Kernel Feature Extraction Methods for Remote Sensing Data Analysis

Emma Izquierdo-Verdiguier

Thesis advisors

Luis Gómez-Chova
Gustau Camps-Valls

Kernel Feature Extraction Methods for Remote Sensing Data Analysis

Emma Izquierdo-Verdiguier, 2014.

Book cover art by Jorge Ruiz Domínguez.

Departamento de Física de la Tierra y Termodinámica

Facultat de Física

D. LUIS GOMEZ-CHOVA, Doctor en Ingeniería Electrónica, Profesor Titular de Universidad del Departamento de Ingeniería Electrónica de la Escuela Técnica Superior de Ingeniería de la Universitat de València y

D. GUSTAU CAMPS VALLS, Doctor en Físicas, Profesor Titular de Universidad del Departamento de Ingeniería Electrónica de la Escuela Técnica Superior de Ingeniería de la Universitat de València

HACEN CONSTAR QUE:

la Licenciada en Físicas Emma Izquierdo Verdiguier ha realizado bajo su dirección el trabajo titulado ''*Kernel Feature Extraction Methods for Remote Sensing Data Analysis*'', que se presenta en esta memoria para optar al grado de Doctor (con Mención Internacional) por la Universitat de València.

Y para que así conste a los efectos oportunos, firmamos el presente certificado, en Valencia a 30 de Mayo de 2014.

Luis Gómez-Chova

Gustau Camps-Valls

Tesis Doctoral: Kernel Feature Extraction Methods for Remote Sensing Data Analysis

Autor: D.ª EMMA IZQUIERDO VERDIGUIER

Directores: Dr. LUIS GOMEZ CHOVA
Dr. GUSTAU CAMPS I VALLS

El tribunal nombrado para juzgar la Tesis Doctoral arriba citada, compuesto por:

Presidente: ______________________________

Vocal: ______________________________

Secretario: ______________________________

Acuerda otorgarle la calificación de ____________________

Y para que así conste a los efectos oportunos, firmamos el presente certificado.

Valencia, a

Agradecimientos

Cuando un proyecto como la realización de una Tesis alcanza su final no siempre es debido al empeño, ganas y dedicación que el autor ponga en su realización. Detrás del trabajo cumplido hay mucha gente que apoya y cree en la persona que lo realiza. Son personas que animan a seguir adelante brindando, de diferentes maneras, su solidaridad.

En mi caso a lo largo de todos estos años he tenido la gran suerte de poder contar con esas personas cerca. Y gracias a ellas no sólo ha sido posible ponerle el punto final a esta Tesis sino que además he tenido la oportunidad de aprender y ''crecer'' a su lado. Echando la vista atrás, una de las primeras personas a las que quiero mostrar mi gratitud y a la que le tengo un gran cariño y aprecio es al *Dr. Calpe-Maravilla*. Depositó su confianza en mí hace ya unos cuantos años y me dio la oportunidad de poder entrar a trabajar dentro de un impresionante grupo, descubriéndome el mundo de la investigación.

A los *Drs. Camps-Valls* y *Gómez-Chova*, se me hace ''bola'' llamaros así con lo fácil que es Gus y Luis, mis directores. No solo por guiarme, orientarme, ayudarme y aconsejarme sino por hacer que disfrute con mi trabajo. Gracias a ellos he aprendido mucho a lo largo de estos años, y no solo de clasificadores y kernels. Entre otras muchas cosas me han dado la oportunidad de descubrir mundo. No sólo en los viajes a congresos a los que me llevaban o en las estancias a las que me mandaban, cada vez más lejos por si en alguna de aquellas me perdía por el camino. También he tenido la oportunidad de encontrar islas paradisiacas y repúblicas que ni conocía ya que, eso de etiquetar nubes por todo el planeta, lo requería (Luis, luego no querías que Gus y yo te puenteásemos). No me gustaría que estos agradecimientos, y por ser los directores, no reflejasen la gran calidad humana que tienen ambos y que transmiten a la gente que se forma con ellos. Gus y Luis, no tengo palabras para agradeceros todo el trabajo y tiempo que habéis dedicado en mi formación y en esta Tesis.

Por supuesto a todos los miembros que forman parte del grupo de *Image and Signal Processing* (ISP), por todas las veces que no han dudado en ayudarme, horas de trabajo que hemos pasado juntos, FFT, papers, congresos... pero sobretodo por el gran ambiente de trabajo que se respira a vuestro lado. En especial a *Julia*, tantos años compartiendo despacho y vivencias. Has hecho que tuviera grandes momentos y me has animado y apoyado en los malos pero sobre todo...haces que el trabajo día a día sea mucho más afable. Por otro lado, es un gusto poder ''frikear'' de linux, tablets y demás con *Jordi*. ¡Anda que no te has tirado tiempo echándome un cable, eliminando errores y warnings del LaTeX!. Al *Dr. Laparra* persona esencial en este trabajo (sin lugar a dudas eres un medio director de la presente), gracias por tu colaboración y ayuda. Esta Tesis no hubiera sido igual si no te hubiera contado el KECA a mi vuelta de Noruega. A Valero decirle que gracias por todo, sabes que los vídeos son lo de menos, una mera excusa para desconectar del trabajo. A *Jesús* y *Luis Alonso* por las grandes charlas en la hora de la comida que nos amenizan esos momentos de agobio y estrés. Y a *MariaPi* por las veraniegas conversaciones en el despacho. Por supuesto, no me olvido del resto de integrantes del ISP: *Sandra*, *Antonio* y *Raúl*, y otros miembros que han pasado por el grupo como *Vicent* y *Devis*. A *Neus*, *Jorge* y el resto de integrantes del grupo *LEO*.

La realización de una Tesis es un laborioso trabajo que conlleva años de trabajo. En este caso, no hubiera sido posible sin la financiación de la beca FPI (BES-2009-013518) del extinguido (*requiescat in pace*) Ministerio de Ciencia e Innovación (*#nonosvamosnosechan*) y el contrato $4000101773/10/I - LG$ Option ''Improvement and integration of cloud screening products, (CPI-13-271)'' de la ESA. Dentro de la beca del ministerio, se incluyeron las distintas estancias que me han dado la posibilidad de conocer y aprender otras formas de trabajar e investigar.

Aprovecho para agradecer al *Prof. Lorenzo Bruzzone* la posibilidad de dejarme conocer la *città di Trento*, así como el norte de Italia. A los integrantes del *Remote Sensing Laboratory*, en especial a las *Dras. Francesca Bovolo* y *Begüm Demir* por la acogida y cariño recibido durante mis meses de estancia. Fuera del ámbito universitario tuve la oportunidad de conocer a distintas personas que hicieron que mi estancia en Italia fuera muy agradable: *Anna*, *Jacopo*, *Manuel*, *Miren*, *Nicola* y *Ronny*.

Al *Prof. Jerónimo Arenas-García* por la posibilidad que me dio de volver a conocer la ciudad donde nací y pasé los primeros años de infancia. A los integrantes del departamento de *Teoría de la Señal y Comunicaciones* de la *Universidad Carlos III* de Madrid, por darme una visión completamente distinta de tratamiento y análisis de imágenes. Y a *Sergio* por la ayuda aportada para que el *SS-KOPLS* saliera adelante.

La última estancia realizada fue en el *Deparment of Physics and Technology* de la *Universidad de Tromsø*. Gracias al *Prof. Robert Jenssen* no solo aprendí de Teoría de Información, sino que descubrí unas remotas y frías tierras que, en verano, merece la pena explorar. Agradecer a *Gema*, *Ricardo*, *Jonas*, *Muhammad*, *Thomas*, *Marta* y *Matthieu* que mi estancia en la isla de Trom no fuese gélida, a pesar de poder hacer esquí de fondo en julio.

En los agradecimientos de esta tesis no pueden faltar las personas que indirectamente han hecho posible la finalización de este capítulo de mi vida. Mi madre y el resto mi familia, por aguantar mi estrés y estar siempre ahí, pero sobre todo a los que no han podido acompañarme durante estos años, porque por mucho tiempo que pase os seguiré recordando. Y como no, mi segunda familia porque siempre me brindáis vuestra ayuda y apoyo. Porque ya sea en una gran celebración o un mal momento, siempre estáis ahí. Por recargar mi energía y por todo el cariño que me dais: *Álvaro, Amparo, Amparito, Andrés, Carlos, Cris, Gabriela, Jordi, Jorge, Isa, Isaías, Jose, Juanmi, Juan, Lambies, Miguel Ángel, Migueler, Patri, Pitu, Rojillez, Virgil...*

Emma Izquierdo Verdiguier.

''El trabajo que nunca se empieza es el que tarda más en finalizarse.''
J. R. R. Tolkien

''It's the job that's never started as takes longest to finish.''
J. R. R. Tolkien

A mi padre

Contents

Abstract in Spanish

Esta Tesis aborda el análisis de datos de teledetección empleando métodos de aprendizaje máquina. En particular, en este trabajo se proponen diferentes métodos kernel para la extracción de características relevantes a partir de imágenes hiperspectrales adquiridas por satélites de observación de la Tierra. El gran volumen de datos adquiridos, debido a la cada vez mayor resolución tanto espacial y temporal como espectral de las imágenes, hace casi imprescindible el empleo de técnicas que permitan reducir la dimensionalidad de los datos manteniendo la información relevante. Por otro lado, la heterogeneidad y las relaciones no lineales presentes en este tipo de datos sugiere el uso de métodos avanzados no lineales que sean capaces de adaptarse a las particularidades y propiedades de estas imágenes. El fin último del análisis de las características extraídas es, en definitiva, el de interpretar la información contenida en los datos y extraer conocimiento. Si se consigue capturar la estructura subyacente en los datos correctamente, estas características pueden emplearse directamente en tareas generales como clasificación, regresión, segmentación, compresión, o visualización facilitando su uso y mejorando los resultados. En este contexto, esta Tesis presenta diferentes métodos núcleo (*kernel*) de extracción de características con dos objetivos principales: 1) incluir conocimiento a priori sobre el problema a resolver y 2) aprender la distribución de los datos disponibles.

Además, uno de los problemas en la mayoría de aplicaciones de teledetección es la dificultad para obtener muestras etiquetadas por lo que es pertinente el estudio de algoritmos supervisados, no supervisados y semisupervisados. Por tanto, las propuestas que se plantean en la presente Tesis son las siguientes. En los métodos supervisados, la calidad de los resultados depende de las muestras etiquetadas que en muchas ocasiones son escasas y no contemplan todos los escenarios posibles, lo cual puede afectar a la capacidad de generalización y robustez de los modelos. En estos casos, cuando se conoce el problema y frente a qué variables el modelo debería ser invariante, forzamos esa invarianza generando muestras de entrenamiento virtuales que incluyan esa información. En los métodos no supervisados, el número de muestras de las cuales se puede aprender no está limitado, pero al no disponer de muestras etiquetadas, la aproximación más usual para reducir la dimensionalidad de los datos es la de encontrar las proyecciones que preservan mejor la varianza de los datos. Sin embargo, cuando los datos no siguen una distribución Gaussiana, podemos usar otros criterios como la teoría de la información para encontrar las proyecciones que alternativamente maximicen la entropía.

Por otro lado, un problema común en los métodos no supervisados es que no se dispone de información para ajustar los parámetros libres del modelo. Para aliviar este problema se propone un kernel generativo capaz de medir similitudes locales y globales al considerar diferentes agrupamientos a diferentes escalas, lo cual hace el modelo prácticamente independiente de ningún parámetro.

Por último, cuando se dispone de un número limitado de muestras etiquetadas, las características extraídas pueden aprovechar la información contenida en las abundantes muestras sin etiquetar de la imagen analizada. Para ello, se propone un método semisupervisado que combina las similitudes del kernel supervisado y el kernel generativo aprendido a partir de todos los datos.

Los métodos de extracción de características propuestos se ilustran en bases de datos estándar e imágenes de teledetección de diferentes características. En general los resultados confirman, por un lado las hipótesis de partida mostrando una clara ventaja de los métodos kernel respecto de las versiones lineales. Y por otro lado, los métodos propuestos que para cada uno de los posibles escenarios de aprendizaje resuelven o mitigan los problemas más frecuentes impuestos por datos de teledetección.

Abstract

This Thesis faces the challenging problem of remote sensing data analysis from a machine learning perspective. In particular, different kernel feature extraction methods are proposed to discover the most relevant features form multi and hyperspectral images acquired by Earth observation satellites. The huge data volume acquired by these sensors –basically due to the increasing spatial, temporal and spectral resolution of the images– makes almost mandatory using techniques that allow us to reduce the data dimensionality while keeping the relevant information. On the other hand, the heterogeneity and nonlinear relations present in this type of data suggests the use of advanced nonlinear methods adaptable to the particularities and properties of the images. The analysis of the extracted features is ultimately aimed at interpreting the data information and knowledge discovery about the problem. If the underlying data structure is properly captured, these features can be directly used in general tasks such as classification, regression, clustering, compression or visualization, making easier the analysis and improving the results. In this context, this Thesis presents different kernel feature extraction methods with two main objectives: 1) to include a priori knowledge about the problem to be solved, and 2) to learn the data distribution from the available samples. In addition, one of the problems in most remote sensing applications is the difficulty to obtain labeled samples, which makes also pertinent to study supervised, unsupervised and semisupervised approaches. Therefore, proposals in the present Thesis can be summarized as follows. In supervised methods, the quality of the results depends on the available labeled samples, which are usually scarce and do not cover all possible scenarios. This, in fact, may affect the generalization capability and robustness of the models. Therefore, when one knows the problem and to which variables the model should be invariant, one can force this invariance by generating virtual training samples that encode this information. In the unsupervised methods, the number of unlabeled samples used for learning is almost unlimited and, in the unsupervised feature extraction context, the most common approximation to reduce the dimensionality is based on preserving the variance of the data. However, when the data do not follow a Gaussian distribution, one can resort to information theory concepts in order to find the projections maximizing entropy. On the other hand, a common problem of unsupervised methods is that there are no labeled samples to tune the free parameters of the models. In order to mitigate this problem, a generative kernel based on the cluster assumption is proposed. The kernel captures local and global similarities in the data manifold by clustering the data at different scales, which makes the model almost independent of any parameter. Finally, when a limited number of labeled samples is available, extracted features can also exploit the information contained in the wealth of unlabeled samples from the analyzed image. Therefore, a semisupervised method combining the similarities of the supervised kernel and the generative kernel learned from the whole dataset is proposed. The feature extraction methods proposed in this Thesis are illustrated in standard machine learning problems and in a wide range of remote sensing images with different characteristics. The results confirm the hypotheses showing a clear advantage of kernel methods with respect to the linear versions. Moreover, results show how the proposed methods solve or mitigate the problems present in remote sensing data.

Preface

Context and overview

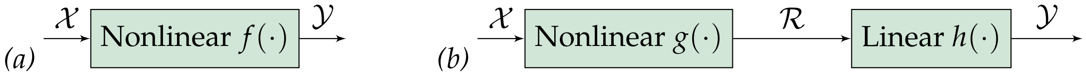
Earth observation by remote sensing is an interdisciplinary field of Science focused on monitoring our planet using a wide range of instruments that capture information of the observed scenes at different electromagnetic wavelengths. Materials can be identified using the different interactions with the electromagnetic radiation emitted, absorbed and reflected by objects depending on their composition. The spectral information is also very useful to estimate relevant biophysical parameters characterizing the processes on Earth. Examples of remote sensing products and applications include land cover thematic maps, land use inventories, as well as temperature and chlorophyll content maps, just to name a few. Remote sensing actually involves a plethora of broader applications with great economical and societal values, such as urban monitoring, fire detection or flood damage evaluation among others. In all of them, analyzing the acquired images by the sensors efficiently is of paramount importance. In this context, Earth observation by remote sensing implies nowadays many different fields of Science and Engineering, such as signal and image processing, statistics, computer vision, and physics. All of them follow different approaches to tackle the same fundamental scientific challenge: extracting useful information from the acquired images.

Before answering the question of how to extract the useful information from images, one should question what makes optical remote sensing images so distinctive. Statistically, multi and hyperspectral images are not very different from natural grayscale and colour photographic images. Grayscale images are spatially *smooth*: the joint probability density function (*pdf*) of the luminance samples is highly nonuniform, the covariance matrix is highly non-diagonal, the autocorrelation functions are broad and have generally a $1/f$ band-limited spectrum. In the case of color images, the correlation between the tristimulus values of the natural colours is typically high. Despite all these commonalities, the analysis of multi- and hyperspectral images turns out to be more difficult, especially because of the high dimensionality of the pixels and the spatial-spectral patches, the high spatial and spectral redundancy, the particular noise sources present in the acquired data, the typically non-Gaussian nature of the data, and perhaps more importantly their potential nonlinear nature. Such nonlinearities can be related to a plethora of factors, including the multi-scattering in the acquisition process, the heterogeneity at subpixel level, as well as the impact of atmospheric and geometric distortions. The imaging process may lead to non-Gaussian pixel distributions, as well as pixels typically embedded on distinct nonlinear manifolds within the higher-dimensional feature spaces. The high spectral sampling of hyperspectral images, for instance also leads to strong collinearity issues. Finally, the spatial variability of the spectral signature increases the internal class variability. All these factors, in conjunction to the few labeled examples typically available, make information extraction a very challenging problem. As a result, the accuracy obtained with standard parametric models, either for data classification, regression or *pdf* estimation, is very often compromised.

The problems raised by the high dimensionality of the data and the high spectral and spatial collinearity are commonly referred to as the *Hughes phenomenon* in the remote sensing community. This problem is known as the *curse of dimensionality* in the machine learning community. The problem is ubiquitous in remote sensing data processing. Inference in high-dimensional low-sized datasets turns to be very challenging because the lack of samples to properly cover the space volume increased by dimensionality. There are two obvious solutions to this situation: either increase the number of samples or to reduce the data dimensionality. On the one hand, increasing the number of labeled samples is aimed to obtain a statistically more robust and reliable representation. Nevertheless, to achieve this objective, the number of samples has to increase exponentially with the dimensionality, which is not easy in remote sensing given the high cost associated to data labelling. On the other hand, reducing the dimensionality of the data can be done either through feature selection or feature extraction approaches. Feature selection, or variable selection, tries to select a subset of relevant input features (or variables) that optimally summarize the data information. The main advantage is that the retained features keep their meaning and units, but these approaches may be hampered by strong nonlinear feature relations and scarcity of labeled samples to guide the optimization criterion. Alternatively, one can resort to *feature extraction* techniques, which aim to find a transformation of the data to a lower dimensional space while retaining most of the information content. The obtained features are combinations of *all* the original input features, so in principle they can be more appropriate to deal with complex feature relations.

Motivation and Objectives

This Thesis will investigate the recurrent problem in remote sensing data processing of dealing with high dimensional low-sized datasets. The previous alternatives to solve the problem will be investigated. On the one hand, we will focus on methods that can incorporate potentially informative samples without additional sampling: we will exploit *prior knowledge* about the problem and the expected feature relations to include synthetic examples, and we will exploit the information contained in the wealth of unlabeled examples present in the scene to better model the data distributions. On the other hand, we will concentrate on reducing data dimensionality via feature extraction in order to obtain potentially useful data representations. In both approaches we will have to deal with nonlinear feature relations and non-Gaussian scenarios. Therefore, we will consider nonlinear inference functions for the task at hand. The general aim is then learning a function $f(\cdot)$ that, departing from input sensory data $x \in \mathcal{X}$ can predict an output target variable $y \in \mathcal{Y}$. The problem can be approached directly with nonlinear models implementing $f(\cdot)$, e.g. with neural networks or kernel machines. Despite its efficiency, this approach leads to hidden representations that are hard to analyze and visualize. Alternatively, one can approach the problem by learning an intermediate transformation $g(\cdot)$ from the original, potentially high-dimensional feature space $\mathcal{X}$, to an accessible representation space of fewer dimensions, $\mathcal{R}$. Then one can use the data projected into $\mathcal{R}$ to perform a simple linear transform, $h(\cdot)$, to infer the output variable. This approach delivers two important advantages: 1) the first nonlinear step leads to an accessible feature space of lower dimensionality, and 2) the second linear step typically involve solving simpler, faster and convex optimization problems. The Thesis will focus on this second approximation, and in particular will design kernel methods to learn the nonlinear feature extraction transformation.



This Thesis is aimed to support the scientific and technological interest in kernel feature extraction for remote sensing data analysis. Two major points motivate the combination of feature extraction methods with kernel methods. On the one hand, feature extraction methods reduce the dimensionality of data looking for data projections that better describe the data. On the other hand, kernel methods implicitly transform the original data to a feature space in which the nonlinear relations are likely to be reduced. The data projected into the most relevant features therein may be more useful than in the original input space because they account for nonlinearities and non-Gaussian distributions efficiently.

The Thesis in a nutshell

- **What is the main goal?** To develop algorithms for dimensionality reduction better adapted to the statistical characteristics of remote sensing data: high dimensionality, low number of labeled samples, nonlinear feature relations and non-Gaussian distributions.
- **Why is the topic important?** the goal is important and timely given the increasing number of heterogeneous satellite images acquired by current and upcoming satellite constellations. Reducing the data dimensionality while keeping the information content is a relevant research and a technological opportunity. The goal is also challenging methodologically as it implies developing new machine learning methods adapted to specific data characteristics.

- **How do we plan to address it?** Kernel methods provide a solid mathematical framework to tackle nonlinear dimensionality reduction. Kernel machines also allow us to design nonlinear algorithms easily, and are able to incorporate *prior* knowledge, invariances and regularization terms. All these ingredients are aimed to generate data representations with high expressive power, i.e. compact and informative.

Research objectives

The present Thesis contributes with kernel feature extraction developments to deal with remote sensing data. We will develop methods that can cope with nonlinearities and non-Gaussian data distributions, as well as with methods that deal with few labeled samples, incorporate prior knowledge and invariances through regularization. These objectives will be guided by the following research questions:

- **On the remote sensing data manifold characteristics.**

  1. *Can kernel methods cope with global and local manifold structure?* Kernel methods can capture nonlinear feature relations efficiently. Typically they rely on a kernel function that reflects *global* data similarities. However, *local* relations can play a fundamental role in the feature extraction, especially since uneven sampling is a common place in remote sensing datasets. By designing kernel functions that account for *local* relations in the manifold we aim to answer about how much local relations can be learned, what are the extent of such relations, and in what cases local information is more useful than global information. The problem is even more challenging in unsupervised scenarios because kernel parameters must be inferred without supervision. We will investigate these issues under the learning paradigms of *semisupervised learning* in general and *generative kernels* in particular.
  2. *Should kernel feature extraction be guided by maximizing the variance or entropy components?* As we will see, all kernel feature extraction methods derive projections that optimize a particular criterion in feature spaces, either the variance of the projections, the correlation or alignment with the labels. The latter methods cannot be obviously applied in unsupervised settings. Therefore, we will investigate alternative measures to variance compaction in feature spaces. In particular we will focus on *entropy* as a measure of information content of the data. We aim to answer when and in what situations seeking for maximum entropy kernel components is more appropriate than seeking for maximum kernel variance components. These fundamental question is aimed to study the Gaussian nature of the remote sensing data, as for Gaussian distributions variance and entropy are equivalent.

- **On the inclusion of prior knowledge.**

    1. *Can virtual data help feature extraction in supervised settings?* Supervised feature extraction methods applied to image analysis may need a large amount of labeled samples to yield efficient and effective representation spaces due to the curse of dimensionality. This problem can be solved by increasing the number of labeled samples per dimension but this is very costly and time-demanding (human and computationally). An alternative could be to generate virtual samples from the original training data to fill in the space. Including informative samples is related to *encode invariances* in the extraction, as one can generate synthetic samples that reinforce the prior belief of feature relations based on solid physical knowledge about the problem. In turn, encoding invariances in a learning machine is a clear form of *regularization* since, roughly speaking, one tries to impose smoothness on the most plausible class of functions.
    2. *Can unlabeled data help feature extraction in semisupervised settings?* Semisupervised learning has the advantage of accounting for both labeled samples and the information in the unsupervised data. In remote sensing data problems, e.g. image segmentation or biophysical parameter retrieval, this is a common situation. Departing from the previous generative kernels, we aim to investigate whether inclusion of both labeled and unlabeled samples offer a more accurate description of the remote sensing data manifolds, what is the information content of each counterpart, and in what situations the combination is beneficial.

These research questions and objectives have led us to develop a set of new kernel methods in supervised, unsupervised and semisupervised settings. Methods performance will be examined in illustrative bidimensional toy examples, on classical machine learning databases, and on remote sensing problems (image segmentation and biophysical parameter estimation) with images of different spatial and spectral resolutions.

Organization

This Thesis is organized in seven chapters covering an introduction to remote sensing, a review of the background of kernel methods and multivariate analysis, the proposed supervised, unsupervised and semisupervised methods, and a discussion and conclusions obtained from the work. The outline is summarized as follows:

> **Chapter 1** reviews the fundamental basis of optical remote sensing data processing, summarizes the main concepts involved in Earth observation and remote sensing imaging, and details the main problems in a classical processing chain.

Chapter 2 reviews the fundamentals of *kernel methods*, which is the used statistical learning framework in the Thesis to classify images, estimate biophysical parameters, cluster data and to validate all the proposed algorithms.

Chapter 3 reviews the framework of multivariate analysis to extract linear and nonlinear features, along with a compilation of state-of-art feature extraction methods applied to remote sensing data.

Chapter 4 addresses the study of encoding *invariances* via data synthesis, as well as it analyzes the impact of these samples on supervised nonlinear feature extraction methods. Furthermore, the chapter introduces the virtual support vector machine (VSM) in remote sensing field.

Chapter 5 addresses two unsupervised nonlinear feature extraction methods based on kernels. The first one proposes an optimization of the kernel decomposition based on Kernel Entropy Component Analysis (KECA). The second method is the Probabilistic Cluster Kernel (PCK), a free-parameter generative kernel learned from data that captures the local and global data manifold structure.

Chapter 6 addresses the semisupervised kernel feature extraction problem. The chapter reviews the state-of-art in semisupervised learning and explains the proposed method: SemiSupervised Kernel Partial Least Square (SS-KPLS) and its orthonormalized extension, the SemiSupervised Orthonormalized Kernel Partial Least Square (SS-KOPLS). Both proposals exploit the unlabeled information of the data along with the supervised information.

Chapter 7 summarizes the accomplished objectives and discusses the main conclusions obtained throughout this work.

Appendix A presents the linear algebra tools and the main matrix factorization fundamentals.

Chapter 1

Introduction to remote sensing data processing

Contents

Earth observation through the analysis of remote sensing data has contributed with real-life applications with great societal benefits. For instance urban monitoring, fire detection or flood prediction from remotely sensed multispectral or radar images have a great impact on economical and environmental issues. To treat efficiently the acquired data and provide accurate products, remote sensing has evolved into a multidisciplinary field, where machine learning and signal processing algorithms play an important role nowadays (Tuia and Camps-Valls, 2009a; Camps-Valls, 2009). All the applications, from a machine learning and signal/image processing perspective, are tackled under specific formalisms, such as classification and clustering, regression and model inversion, image coding, restoration and enhancement, source unmixing, data fusion or feature selection and extraction. In general, statistical machine learning has proven successful in many disciplines of Science and Engineering (Hastie et al., 2009). Machine learning is a multidisciplinary field used in several domains such as computer science, signal and image processing, computer vision, etc. In the last decade, machine learning has found widespread adoption in remote sensing and geosciences as well. Nowadays, for instance, statistical inference algorithms are regularly used for image classification, target detection and biophysical parameter retrieval applications, just to name a few (Camps-Valls et al., 2011).

In this chapter, a brief introduction to remote sensing is first given. The electromagnetic radiance, the kind of sensors and their different resolutions are described to frame the Thesis. The standard remote sensing data processing chain that goes from the signal acquisition to the final product is then detailed. Finally, new trends in machine learning for remote sensing data processing are commented.

1.1 Introduction to remote sensing

Earth observation is a multi-disciplinary field embracing physics, chemistry, electronic, cartography, geology, forestry and computer science, among others. Earth observation via remote sensing data analysis aims to study the Earth's surface (and interactions with the atmosphere) as a complex, evolving system. The information captured using different remote sensing imaging sensors is relevant: they essentially measure the electromagnetic radiation reflected, absorbed, and emitted in a different way by materials in a scene depending of their molecular composition and shape. Such acquired signals help monitoring the processes occurring on Earth, and lately thanks to new satellite sensors this can be done with unprecedented accuracy.

The Electromagnetic (EM) Radiation changes according to the wavelength. The energy distribution of all electromagnetic waves is known as *electromagnetic spectrum*. The electromagnetic spectrum covers regions from short to long wavelengths: Gamma-Rays, X-Rays, Ultra-Violet (UV), Visible, Infrared, Microwaves and Radio waves:

- Gamma-Rays and X-Rays are highly energetic waves that are absorbed by the higher layers of the atmosphere, and thus are not favorable for Earth observation.

- Ultra-Violet are also used in studies about surface of planets with and without atmosphere.

- Visible is the most widely used range in Remote Sensing together with the Infrared region. The Visible region covers a narrow interval (0.4 to $0.7\mu m$) of the EM spectrum where we can find Red ($0.6 - 0.7\mu m$), Green ($0.5 - 0.6\mu m$) and Blue ($0.4 - 0.5\mu m$) bands (RGB) that human visual system is adapted to.

- Infrared (IR) can be divided in three ranges: Near, Medium and Far Infrared. The two first intervals are close to the visible region, and along with it, are the most used by spectrometers, radiometers, polarimeters and lasers. The far IR is referred to as the emitted infrared or "thermal energy" considering that the radiation on this interval is directly related to the body's heat.

- The Microwave range is used in microwave radiometers and radar systems, such as the Synthetic Aperture Radar (SAR) imaging systems.
- The last region of the *electromagnetic spectrum* corresponds to the Radio waves, which is used by active sensors like radio altimeters.

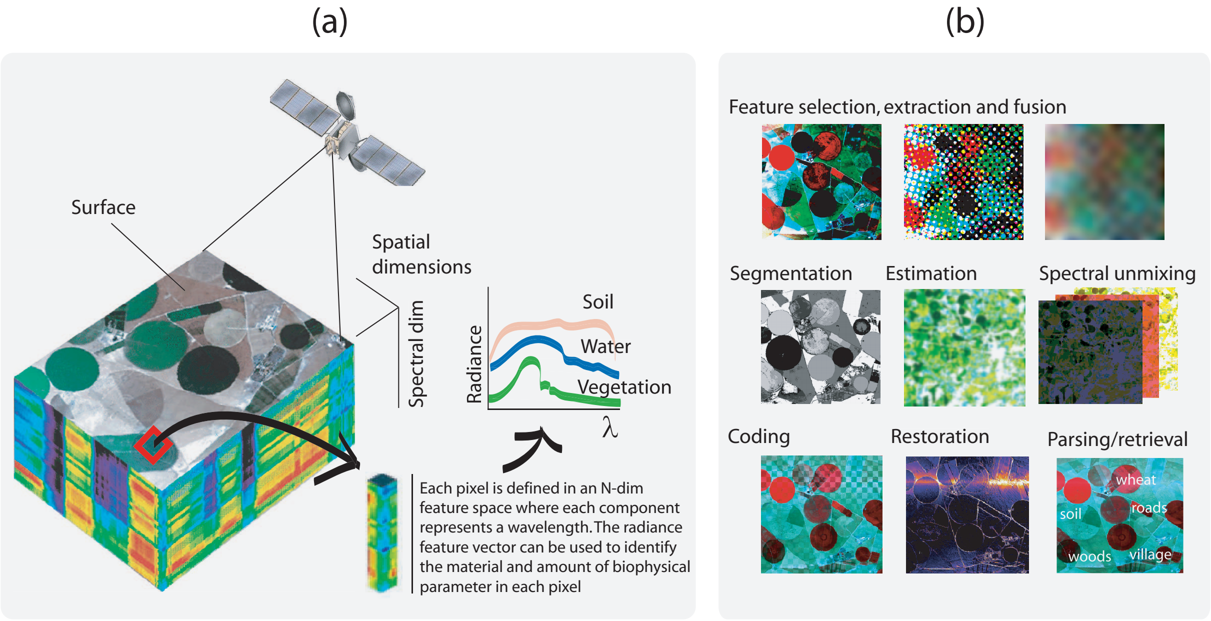
Remote sensing exploits the *radiation-matter iteration* and deals with the acquisition of information about a scene (or specific object) at a short, medium or long distance. According to the type of energy sources involved in the data acquisition, remote sensing imaging instruments can be *passive* or *active*:

- Passive systems: These are sensors that rely on solar radiation as the source of illumination. Some examples of passive sensors are multi- and hyperspectral imaging sensors.
- Active systems: These are sensors that accomplish a dual function, that is, to produce a signal and to register it after interacting with the observed system. Radar systems or Laser Imaging Detection and Ranging (LIDAR) are examples of systems for active remote sensing.

This thesis concentrates on passive systems for Earth monitoring. Passive systems usually exploit solar radiation to capture the reflected radiation, which is acquired by airborne or satellite spectrometers at different wavelengths. The quality of the information collected is based on the *resolution* of the sensors. Higher resolution means more information taken from the scene, but it involves greater information storage. There exist different types of resolution in a remote sensing image:

1. Radiometric resolution is the number of gray levels in which the radiation is divided for storage. The gray level is a numeric value that represents the radiance captured by the sensor.
2. Temporal resolution is related to the revisit time of the remote sensing platform.
3. Spatial resolution is the surface's distance between *image elements* or *pixels*. The spatial resolution not only depends on the type of sensor, but also on the altitude and angle from which the scene is captured.
4. Spectral resolution is related to the number and separation of spectral channels (or bands) recorded by the sensor.

In this Thesis, we will focus on optical images only. Figure 1.1(a) shows the basic principle of imaging spectroscopy to perform satellite remote sensing. Image pixels are linked to two spatial coordinates that, together with the spectral (wavelength) dimension, form a "hypercube" or data cube (Shaw and Manolakis, 2002; Lillesand et al., 2008). Hence, a pixel is a vector where each dimension informs about the radiance captured by the sensor at the different wavelengths. Each pixel in the scene has its own spectrum or *spectral signature*, which can be used to identify and classify the object into different thematic classes automatically.



Each pixel is defined in an N-dim feature space where each component represents a wavelength. The radiance feature vector can be used to identify the material and amount of biophysical parameter in each pixel

Figure 1.1: The concept of imaging spectroscopy. Airborne/Spaceborne imagery is acquired, compressed, preprocessed and analyzed. The latter step includes classification for mapping applications, regression for biophysical parameter estimation, anomaly detection and target recognition. Source: Camps-Valls (2009).

1.2 Standard techniques for remote sensing image processing

Remote sensing image processing departs from the acquired hypercube and applies a set of techniques and methods that allow capturing information from the observed scene. The main objective is to provide a ready-to-use product to the users that is obtained from the acquired image. Figure 1.1(b) shows some of the main problems involved in remote sensing image processing, including the data storage/coding and transmission, pre-processing (e.g. feature selection and extraction), and processing (e.g. segmentation, unmixing). The final aim of this Thesis is to develop *feature extraction methods* for land cover classification and bio-geo-physical parameter retrieval, which is a critical step before the classification or regression steps, respectively. The main stages in the remote sensing image processing chain are detailed in what follows.

1.2.1 Remote sensing image transmission and coding

Along with the increasing demand of hyperspectral data, the sensor technology used to capture remote sensing images has been significantly developed in the last decade, improving, among others, the spatial and spectral resolutions. Such improvements on quality leads to an increasing demand on storage and bandwidth transmission capabilities. Both lossy and lossless image coding have been investigated extensively for multispectral imagery, but more important for hyperspectral images (García-Vílchez et al., 2011). The current recommendation are based on a transform stage, where data is decorrelated in the spatial domain using a wavelet transform (plus a bit plane encoder stage), thus following the latest standard JPEG2000 for grayscale images. Other well-known wavelet-based coding systems are SPIHT-3D and SPECK-3D (Karami et al., 2012). In order to improve the coding performance, a common strategy is to decorrelate first the image in the spectral domain (Keerthana and Sivasankar, 2013).

1.2.2 Remote sensing image preprocessing tasks

From all the possible preprocessing steps aimed at improving the image and product quality, three steps are intimately related to the objectives of this Thesis: *image restoration*, *feature extraction/selection* and *image fusion*.

Image restoration

Image restoration is an important step in the remote sensing image processing chain since it is intended to correct the distortions affecting the image formation. Several corrections are typically applied:

- *Radiometric correction* deals with the corrections of the observed values that are usually related to the imaging sensor and the data acquisition and transmission, such as missing pixels or lines of an image (Yan and Shaker, 2014).
- *Atmospheric correction* is the process of correcting the distortions in the radiance values observed at the sensor level caused by the atmosphere (Fuyi et al., 2013).
- *Topographic correction* is the process of correcting terrain effects, mainly due to the surface elevation and observation/illumination geometries (Schlapfer et al., 2012).
- *Geometric correction* is the process where the geometry of the image is corrected to provide an exact location to the image pixels in a specific surface projection (Hu and Tang, 2011).

In addition to the previous distortions, different noise amounts and sources are present in remote sensing data. To mitigate the noise problem, the most common way in hyperspectral images is by means of PCA (Chen and Qian, 2009). Nevertheless, an alternative is the widely used minimum noise fraction (MNF) algorithm (Green et al., 1988) and its recent nonlinear version (Gómez-Chova et al., 2011). Furthermore, the noise covariance estimation is a more challenging problem and other techniques have been recently proposed, such as anisotropic diffusion (Mendez-Rial and Martín-Herrero, 2012), wavelet shrinkage (Chen and Qian, 2011), or kernel multivariate methods (Camps-Valls and Bruzzone, 2009).

Feature selection and extraction

The high collinearity between spectral bands pose a challenging problem when working with hyperspectral images. Essentially, machine learning algorithms suffer from a risk of overfitting thus giving rise to poor generalization capabilities. In these situations, feature selection and extraction are central tasks because of the *curse of dimensionality* (Guyon et al., 2006). Standard feature selection methods select features that minimize a given criterion, e.g. the classification error, by means of *filter methods*. This approach typically rely on correlation or the mutual information between the bands and the class labels to discard irrelevant or redundant channels. The approach has been extensively studied in remote sensing data processing (Pal and Foody, 2010). Recent advances focus on *wrapper methods*, which select features that minimize the classification error directly (Bolón-Canedo et al., 2013). This approaches are typically greedy and thus computationally expensive.

Feature extraction pursue a different direction: roughly speaking, all input variables are used and combined to derive a reduced set of new features that maximally preserve the information content of the original data. The most common method is linear PCA but recent advances have been proposed using nonlinear methods such as locally linear embedding or isometric mapping (Jia et al., 2013). In the last years, multivariate kernel-based feature extraction methods have been proposed to address nonlinearities in the data (Arenas-García et al., 2013). In all these methods, one can also use *spectral and spatial filters* to extract edges or geometrical features (Richards and Jia, 1999) although one typically exploit *morphological operations* to further improve object detection (Izquierdo-Verdiguier et al., 2011). This thesis will concentrate on multivariate kernel-based algorithms for feature extraction.

Remote sensing image fusion

As we have seen in section 1.1, there are different sensor resolutions: spectral, spatial and temporal. Spatial resolution of sensors is often limited with respect to their spatial resolution. Panchromatic sensors provide high spatial resolution whereas multispectral or hyperspectral sensors give a high spectral resolution with low spatial detail. Sensors with both high spectral and spatial resolutions would be technology challenging and extraordinarily expensive, if at all physically realizable. An alternative comes from the field of imaging processing: image fusion methods are used to generate an image with both (good spatial, good spectral) characteristics. There are specific approaches developed for remote sensing image processing based on wavelets (N. Indhumadhi, 2011) or for multisource fusion such as Ehlers et al. (2010) that combines SAR with optical panchromatic images.

1.2.3 Remote sensing image processing tasks

The last step of the remote sensing image processing chain is intended to provide products obtained from the remote sensing images that can be directly used or interpreted by the final users. There are many products worth mentioning: abundance maps of materials of interest, land cover and land use maps that allow monitoring, estimated biophysical parameter maps that permit phenology studies, saliency maps that highlight interesting portions of the scene, etc. The most relevant remote sensing problems analyzed in this Thesis are land cover classification and biophysical parameter retrieval only. They are typically tackled by particular *classification* and *regression* algorithms, respectively. We will instead focus on learning nonlinear feature transforms that lead to rich data representations where soimple (ideally linear) classification or regression should suffice. In the following sections, we review applications in these two fields.

Biophysical parameter retrieval and model inversion

The estimation of biophysical parameters represents a paramount scientific challenge in remote sensing in order to better understand the environment dynamics at local and global scales (Lillesand et al., 2008). The inversion of analytical models introduces a high level of complexity and computational burden, and sensitivity to noise becomes an important issue. As a direct consequence, the use of *empirical models* adjusted to learn the relationship between the acquired spectra and the actual ground measurements has become very attractive in recent years. *Parametric* models have some important drawbacks, which typically lead to poor prediction results on unseen (test) data. As a consequence, *non-parametric* and potentially *nonlinear* regression techniques have been effectively introduced for the estimation of biophysical parameters from remotely sensed images. Different models and architectures of neural networks have been considered for the estimation of biophysical parameters (Vilas et al., 2011). Recently the use of support vector regression (SVR) and other Bayesian nonparametric methods have been presented as efficient alternatives to neural networks for modeling some biophysical parameters (Verrelst et al., 2012b). We will further review the emerging field of kernel machines for biophysical parameter retrieval in the next chapter.

Image classification

A relevant application in remote sensing is to create classification maps for urban monitoring, catastrophe assessment, change or target detection. Depending on the available data, it is possible to divide classification methods in three main families: i) *supervised methods*, ii) *unsupervised methods* and iii) *semisupervised methods*. Supervised methods are probably the most common in remote sensing, and neural networks (Pacifici and Del Frate, 2010) and support vector machines (Mountrakis et al., 2011) are popular algorithms. The latter method has been applied in several problems for urban monitoring (Schwert et al., 2013) or multi-temporal classification (Niu and Ban, 2013), among others. Unsupervised methods rely on clustering image pixels depending on their similarity. Finally, semisupervised methods involve combining both labeled and unlabeled data in the same model. These methods exploit the information conveyed by abundant unlabeled data to generate, for example, improved land cover maps (Muñoz Marí et al., 2012; Kiyasu et al., 2011).

Table 1.1 summarizes each step of the processing chain in remote sensing. We include the main steps, the fields of engineering and sciences involved, the learning paradigms applicable, a brief summary of the objectives, examples of applications and the main current methods and techniques used in each of them. The summary does not pretend to be exhaustive but to serve

Table 1.1: A taxonomy for remote sensing methods and applications, based on (Tuia and Camps-Valls, 2009a).

Topic	Fields & Tools	Objectives & Problems	Examples	Methods & Techniques
Coding	Transform coding and vision computing	Compress the huge amount of acquired data	Transmission of data to Earth station, avoid redundancy and errors, realistic quick-looks	PCA, DCT, Wavelets, SPIHT, kernel feature extraction
Feature Selection	Filters/Wrappers	Ranking and channel selection	Efficient transmission, model development, compression.	SFFS, RFE, Network pruning, GA, kernel dependence estimation
Feature Extraction	Statistical, denoising, machine learning	Seek the best data direction according to measures of relations among data	data description, model development, multitemporal	Multivariate analysis (PCA), neural networks, kernel methods
Restoration	Denoising, deblurring	interpretation, feat. extract.	Acquisition noise, transmission	Wiener, wavelets, advanced denoising, kernel methods
Data fusion	Image/Signal Proc.	Different sensors, temporal acquisitions, resolutions	Multi-temporal analysis, change detection	Multi-resolution, fusion
Signal Unmixing	Signal Processing and machine learning	Independizing the mixture of spectra, restoration, classification with pure pixels	Unmixing and subpixel techniques	ICA, linear/non-linear unmixing, kernels and pre-images.
Model inversion	Regression	Monitoring Earth's Cover at a local/global scale	Water quality, desertification, vegetation indexes, temperature concentration , biomass, ozone, ...	Linear regression, neural networks, kernel methods.
Classification	Pattern recognition	Monitoring evolution and changes of Earth's cover	Urban monitoring, mineral detection, change detection, ...	k-NN, LDA, neural networks, kernel methods

as a comprehensive summarizing view of the field.

1.3 Advanced machine learning for remote sensing data processing

As we have already motivated before, the high dimensionality of the data, the high spectral and spatial collinearity, or the few labeled examples motivates the use of machine learning techniques in remote sensing data processing. Regularization of the models or semisupervised learning are techniques that allow introducing additional information to prevent overfitting or to solve an ill-posed problem. In this section, we summarize some recent learning paradigms and the main techniques applied in remote sensing that are related to the core of the Thesis.

1.3.1 Manifold Learning

Manifold learning considers that high dimensional data can be mapped to a lower dimensional space without information loss of the original data. This recent field is closely related to dimensionality reduction and nonlinear feature extraction. The manifold learning framework embraces a large set of algorithms. In order to circumvent the problems associated to traditional linear dimensionality reduction methods, many algorithms have been proposed in the machine learning community, such as spectral, graph-based, neural networks, principal curves and projection pursuit methods. They are however seldom used in remote sensing problems. Among them, the most common algorithms applied to remote sensing are Local Linear Embedding (Crawford et al., 2011), Isomap (Feilhauer et al., 2011) and Lapacian eigenmaps methods (Shi et al., 2013). Lately, as will be discussed in upcoming chapters, the family of kernel methods have been exploited to tackle manifold learning problems (Arenas-García et al., 2013).

1.3.2 Semisupervised Learning

Regularization is a way to introduce more information than the available in the data to reduce the complexity of the models. A specific way of regularization is by means of semisupervised learning. The main idea is to develop the model not only with the supervised information but also adding the information of unlabeled data in order to improve the supervised model. These methods can be divided in *generative*, which are based on probabilistic models, or *discriminative*, which directly learn class boundaries. In remote sensing, generative models have been applied in segmentation tasks (Li et al., 2010), urban monitoring (Tuia and Camps-Valls, 2011), or pixel based classification (Maulik and Chakraborty, 2012). Furthermore, it is possible to model the data representation by cluster kernels (Gómez-Chova et al., 2010) or graphs applied in target detection, regression, and classification (Camps-Valls et al., 2014). Other semisupervised approaches are the transductive SVM (Bruzzone and Demir, 2014) and transductive Multiple-Kernel Learning (Sun et al., 2014), which have been applied in image classification. The field of semisupervised learning is tightly related to the previous manifold learning paradigm, since the aim is to learn or encode the information of the manifold structure by exploiting all available data.

1.3.3 Transfer Learning

Finally, it is worth mentioning the field of transfer learning, that aims at transferring knowledge (or a model) trained or adjusted in one (source) domain to another (target) domain. This is an ubiquitous problem in remote sensing data processing. For example, land-cover maps are only updated by classifying image time series when training samples collected at a particular time are available, which commonly requires re-training classifiers. Transfer learning would avoid that by updating the classifier or the data representation. The field of transfer learning is also known as *domain adaptation*. The problem was initially tackled with partially unsupervised classifiers, under parametric formalisms and neural networks. The approach was then successfully extended to the use of SVM (Bruzzone and Marconcini, 2009). A related problem is also that of classifying an image with samples from different images, which induces the sample selection bias or covariate shift problems. These problems have been recently presented by defining proper kernel machines (Gómez Chova et al., 2008). The field is also related to the general problem of learning and inference in manifolds.

1.4 Summary

This Chapter has briefly introduced remote sensing data processing to the reader. The electromagnetic spectrum, the active and passive sensors, different image resolutions and the resulting data have been described. In addition, from the data acquisition to the product generation, we have seen the most common processing problems. After imaging and acquisition, remote sensing needs to be stored and transmitted, and hence advances in image coding come to place. We have also introduced the three main pre-processing steps from an image processing perspective: image denoising, fusion and selection and feature extraction. The latter field of feature extraction will be analyzed in depth in the next chapters since it is the core of this Thesis. The last image processing step has briefly reviewed the main algorithms used to generate land-cover maps and biophysical parameter maps. The last part of the chapter has summarized some machine learning paradigms that provided the basis to develop the tools and algorithms proposed in this Thesis.

Chapter 2

Introduction to Kernel Methods

Contents

This chapter provides a summary of applications and recent theoretical developments of kernel methods in remote sensing data analysis. Section 2.1 summarizes the use of kernel methods in remote sensing and reviews the main applications. Section 2.2 presents a brief introduction to kernel methods, fixes notation, and reviews their basic properties. Section 2.3 reviews the classification setting, under paradigms of supervised, and unsupervised classification. Furthermore, the section presents kernel methods for density estimation and regression settings. We intentionally leave the treatment of kernel feature extraction for the next chapter, as it is the core of the Thesis. For more information about kernel algorithms see (Shawe-Taylor and Cristianini, 2004), (Schölkopf and Smola, 2002) and (Camps-Valls and Bruzzone, 2009).

The chapter is partly based on the paper:

□ E. Izquierdo-Verdiguier, L. Gómez-Chova and G. Camps-Valls, *"Kernels for remote sensing image classification,"* *Wiley Encyclopedia of EEE (Submitted by invitation, expected publication 2015).*

2.1 Kernel methods in remote sensing

Kernel methods are a standard tool in machine learning and pattern recognition, and are very suitable for remote sensing data processing (Camps-Valls and Bruzzone, 2009). Actually, in the last decade, kernel methods have been widely used in remote sensing data processing. This is mainly due to the fact that they fit the needs of the field, and can efficiently cope with the problems posed in many remote sensing data analysis. In particular, kernel methods are a solid mathematical framework to develop nonlinear algorithms that are easy to parametrize, fast and intuitive, and are typically robust to high dimensional data and problems with low number of labeled samples, which is often the case in Earth observation applications. Furthermore, it is possible to combine multi-source heterogeneous data (i.e. images) by means of the use of particular compositions of kernels dedicated to process the spatial, spectral or temporal information (Camps-Valls et al., 2006b).

The information of the images acquired by imaging systems allow the characterization, identification, and classification of the land covers present in the scenes (Richards and Jia, 1999). However, the high input data dimensionality in, for example, hyperspectral images degrade the performance of traditional classifiers such as artificial neural networks or Gaussian maximum likelihood. These methods are highly impacted by datasets with low ratios of samples per number of dimensions, essentially because either they estimate poorly the parameters in the presence of noise or data sampling hampers the challenging problem of density estimation they aim to solve. The issue of low number of samples per dimension is known in the literature as the *curse of dimensionality* and has been largely reported in remote sensing image classification problems (Hughes, 1968; Fukunaga and Hayes, 1989). In such situations, one is forced to impose some constraints on the solution, by mainly restricting the *capacity* (i.e. flexibility, number of parameters) of the model to enforce simpler decision functions. This has been approached by pruning weights in neural nets, enforcing sparsity in regression models, including informative features, and encoding invariances in the model. All of these approaches can be casted as different forms of *regularization*.

In the last decades, the use of support vector machines (SVM) widespread in many fields of engineering and science (Shawe-Taylor and Cristianini, 2004; Schölkopf and Smola, 2002). The field of remote sensing has witnessed a similar wide adoption of SVMs, especially to tackle image classification problems (Camps-Valls and Bruzzone, 2009). The SVM has integrated three main processes in the same single algorithm. First of all, SVM makes a *feature extraction* step since data are mapped to a higher dimensional space in which they are classified with a simple (linear) algorithm. Second, in SVM, it is possible to control the complexity of the model by means of an efficient *regularization procedure*. And finally, an upper bound of the generalization error is minimized, thus the SVM follows the Structural Risk Minimization (SRM) principle (Vapnik, 1998), which not only focuses on minimizing the training error but also limiting the capacity of the classifier. The application of SVMs has demonstrated very good performance in multispectral, hyperspectral, and multi-source image classification, see e.g. (Camps-Valls and Bruzzone, 2005; Camps-Valls et al., 2006b; Tarabalka et al., 2010; Tuia et al., 2011b,a; García et al., 2011). In this chapter, we will review the SVM formulation since this is the most widely used kernel method nowadays.

Another different concern in remote sensing problems is related to situations where the training set is incomplete and/or not representative. We should stress that actually few attention has been paid to the case of having an incomplete knowledge of the classes present in the investigated scene. Very often we find problems in which there is only one class of interest to be detected, e.g. in anomaly and change detection problems, urban monitoring, and cloud detection, just to name a few scenarios. In order to solve these types of problems, SVM has played an important role with extensions such as the *one-class SVM*, which only considers samples belonging to the class of interest in order to learn the underlying data class support. The method was originally introduced for anomaly detection (Mercier and Girard-Ardhuin, 2006), then analyzed for dealing with incomplete and unreliable training data (Muñoz Marí et al., 2007), it has also been recently reformulated for change detection (Muñoz Marí et al., 2010), and recently was reformulated to introduce the user's

Remote sensing image classification is restricted by both the quality and the number of labeled samples. This is due to the fact that acquiring ground truth information is very challenging and costly, mainly in complex and heterogeneous geographical areas. In order to alleviate this problem, the semisupervised learning (SSL) paradigm (Chapelle et al., 2006) introduced in machine learning has been also exploited in remote sensing image classification. SSL employs the information contained in the abundant unlabeled samples along with the low number of labeled samples. In several image classification applications, semisupervised learning has been applied ranging from: image segmentation (Mitra et al., 2004), image classification by means of semisupervised one-class SVM (Muñoz Marí et al., 2010), deformation kernels with the Laplacian SVM (Gómez-Chova et al., 2008) for cloud detection, transductive SVMs in remote sensing image classification (Chi and Bruzzone, 2005; Bruzzone et al., 2005, 2006), and semisupervised data mining applications (Vatsavai et al., 2005).

Kernel methods have also found application in regression and function approximation (Smola and Schölkopf, 2004), and also found application in some particular Earth observation applications. Remote sensing very often deals with inverting a forward model or to approximate bio-geo-physical parameters from acquired imaging spectra. The goal is thus to obtain a robust model able to predict/estimate different parameters, such as temperature, vegetation variables like the fraction of vegetation cover (FVC) or leaf area index (LAI), water vapor, etc. While support vector machines (SVM) were introduced in the mid-90s for classification and regression, only recently the regression variation (*support vector regression*, SVR) gained popularity for continuous biophysical parameter retrieval. For instance Karimi et al. (2008) used the SVR model for estimating various crop physiological parameters (plant height, leaf nitrogen content, and leaf chlorophyll content) from hyperspectral data. The same approach was applied by Yang et al. (2011), who found that SVR performed superior compared to linear nonparametric methods. One of the emerging powerful kernel-based regression methods involves *kernel ridge regression* (KRR), also known as least squares support vector machines (LS-SVM). KRR proved to be very promising because of its excellent performance. Wang et al. (2011) compared KRR against linear nonparametric methods (multiple linear regression and PLSR) for LAI estimation and concluded that KRR yielded the most accurate estimates. In the recent years, kernel-based model inversion has dominated the field of non-linear and non-parametric biophysical parameter estimation (Camps-Valls et al., 2011; Verrelst et al., 2012a; Bioucas-Dias et al., 2013; Okujeni et al., 2013). In this Thesis, we will intentionally obviate all these powerful nonlinear regression approaches, as our main goal is to analyze (regression) problems from the perspective of *nonlinear* feature extraction plus *linear* regression.

2.2 Introduction to kernel methods

This section includes a brief introduction to kernel methods (Camps-Valls and Bruzzone, 2009; Gómez-Chova et al., 2011; Izquierdo-Verdiguier et al., 2015). After setting the scenario and fixing the used notation, we give the main properties of kernel methods, namely the notion of kernel function, the property of positive definiteness, and the reproducing property. We also pay attention to kernel methods development by means of particular properties drawn from linear algebra and functional analysis (Golub and Van Loan, 1996; Reed and Simon, 1980). The field of kernel machines is very vast. We however omit many interesting aspects in this review, such as bounds of performance, capacity control and convergence and stability issues. This is done intentionally for the sake of simplicity and to treat only the issues relevant to the topic of this Thesis.

2.2.1 Measuring similarity with kernels

Machine learning deals with the very ambitious goal of learning and recognizing patterns from data automatically. These patterns may come in different forms depending on the inference problem: in classification problems one is interested in detecting relevant features that separate samples belonging to different classes maximally, while in regression problems the interest is in learning an approximating a function of the underlying system that generated the data. Many methods exist to tackle the different problems: classification and regression trees, neural networks, boosting algorithms, projection pursuit and random projections, etc. Despite the different principles guiding each particular algorithm, there is a common thing in all of them:

machine learning algorithms try to learn feature representations that highlight similarities (or dissimilarities) among points. In this context, the field of kernel methods can be seen as an appropriate framework to define statistical inference problems. Kernel methods essentially rely on the notion of similarity between examples through the concept of kernel function. As we will see, kernel methods can formalize many algorithms through this *kernel function*.

All machine learning problems start with the collection of a representative dataset describing the problem at hand. In remote sensing data processing this may imply for example a terrestrial campaign that measures a biophysical parameter/variable on the ground at the same time an airborne/satellite sensor overpasses the area of study thus yielding spectra. The dataset is thus formed by pairs of spectra and corresponding parameter, and the problem can then be formalized as a prediction or regression problem. Formally, let us define a set of empirical data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, where $\mathbf{x}_i$ are the *inputs* (often called the independent variables) taken from $\mathcal{X}$ and $y_i \in \mathcal{Y}$ are called the *outputs* (or dependent variable). The problem of *learning from examples* implies to use these sample pairs to *predict well on test (unseen) examples*. To develop machines or models that generalize well, kernel methods try to exploit the structure of the data and thus define a similarity between all pairs of samples available in the training set. Then, with such learned (or inferred) similarity measure fixed, a new incoming test example is, roughly speaking, simply assigned a prediction corresponding to the most similar example in the training set. Therefore one can (simplistically) see that kernel methods are a kind of memory-based algorithms.

The main problem encountered in machine learning is that very often $\mathcal{X}$ is not appropriate to work, that is the input space has not a proper notion of similarity. In such case, the input features are not discriminative of the different classes or do not carry enough information for prediction. Machine learning methods typically solve the problem by introducing nonlinearities that transform the data with the aim of separating or aligning examples in a richer representation space. Kernel methods do the same: examples are mapped to a (dot product) space $\mathcal{H}$, using a *feature mapping* $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{H}, \mathbf{x} \mapsto \boldsymbol{\phi}(\mathbf{x})$. The similarity between the vectors (points, or elements) in $\mathcal{H}$ can now be estimated using its associated dot product in that feature space, i.e $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Here, we define a function that computes that similarity, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that $(\mathbf{x}, \mathbf{x}') \mapsto K(\mathbf{x}, \mathbf{x}')$. This *kernel function* needs to satisfy:

$$K(\mathbf{x}, \mathbf{x}') = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') \rangle_{\mathcal{H}}, \tag{2.1}$$

where the mapping $\boldsymbol{\phi}$ is the *feature map*, the space $\mathcal{H}$ is called the *feature space*, and K is a reproducing kernel in Hilbert space (RKHS). This equality is the core of all kernel methods: intuitively the expression tells us that we may find a (similarity) kernel function that works with data in $\mathcal{X}$ that is *implicitly* reproducing the similarity between these data mapped to a higher dimensional feature space. The important thing here is that we do not need to map explicitly the data points, nor to have access to their coordinates: we will just simply work with kernel functions and original space data points. The equality has of course to fulfill the properties of existence and uniqueness, whose formal demonstration can be found, e.g. in (Camps-Valls and

Bruzzone, 2009)[chapter 2].

2.2.2 Positive definite kernels

An important property of kernel methods is that of *positive definiteness*. This is because the class of kernels that can be written in the form of the previous equality (2.1) coincides with the class of positive definite kernels.

Definition 1. *A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel* if and only if *there exists a Hilbert space $\mathcal{H}$ and a feature map $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{H}$ such that for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ we have $K(\mathbf{x}, \mathbf{x}') = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') \rangle_{\mathcal{H}}$.*

In practice, a real (symmetric) matrix $\mathbf{K}$ of size $n \times n$, whose entries are $K(\mathbf{x}_i, \mathbf{x}_j)$ or simply K_{ij}, is named *positive definite* if for all $c_1, \ldots, c_n \in \mathbb{R}$, $\sum_{i,j=1}^{n} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0$. Note that a positive definite kernel is equivalent to a positive definite Gram matrix in the *feature space*. Throughout this Thesis we will work with positive definite kernels, and when proposing new kernel functions we will demonstrate that the property is fulfilled. Being a positive kernel (Gram) matrix implies that all the eigenvalues must be positive and has some implications on the spectral analysis of the proposed kernels in this Thesis.

Therefore, algorithms operating on the data only in terms of dot products can be used with any positive definite kernel by simply replacing $\langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') \rangle_{\mathcal{H}}$ with kernel evaluations $K(\mathbf{x}, \mathbf{x}')$. This technique is also known in the statistical inference community as the *kernel trick* (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004). Another direct consequence is that, for a positive definite kernel, one does not need to know the explicit form of the feature map since it is implicitly defined through the definition of the kernel.

2.2.3 Basic operations with kernels

For the interest of the developments in this Thesis, we now review some basic properties of kernels functions. We want to stress that, although the space $\mathcal{H}$ can be very high-dimensional, some basic operations can still be performed therein implicitly:

Translation. A translation in feature space can be written as the modified feature map $\tilde{\boldsymbol{\phi}}(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x}) + \boldsymbol{\Gamma}$ with $\boldsymbol{\Gamma} \in \mathcal{H}$. Then, the translated dot product for $\langle \tilde{\boldsymbol{\phi}}(\mathbf{x}), \tilde{\boldsymbol{\phi}}(\mathbf{x}') \rangle_{\mathcal{H}}$ can be computed if we restrict $\boldsymbol{\Gamma}$ to lie in the span of the functions $\{\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)\} \in \mathcal{H}$. The property is widely used to define particular kernel functions that are robust (or invariant) to undesired data translations. We will however use this operation just to center data in feature spaces.

Centering. Note that the previous translation allows us to center data $\{\mathbf{x}_i\}_{i=1}^{n} \in \mathcal{X}$ in the *feature space*, i.e. $\{\boldsymbol{\phi}(\mathbf{x}_i)\}_{i=1}^{n}$. The mean of the data in $\mathcal{H}$ is $\boldsymbol{\phi}_{\mu} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\phi}(\mathbf{x}_i)$, which is a linear combination of the span of functions, and thus fulfills the requirement for $\boldsymbol{\Gamma}$. One can center data in $\mathcal{H}$ by computing $\tilde{\mathbf{K}} = \mathbf{HKH}$ where $\mathbf{H}$ is called the "centering matrix" whose entries are $H_{ij} = \delta_{ij} - \frac{1}{n}$ with the Kronecker symbol $\delta_{i,j} = 1$ if $i = j$ and zero otherwise.

Computing distances. We have seen that the kernel function corresponds to a dot product in a Hilbert space $\mathcal{H}$, and thus one can compute distances between mapped samples entirely in terms of kernel evaluations:

$$d(\mathbf{x},\mathbf{x}') = \|\boldsymbol{\phi}(\mathbf{x}) - \boldsymbol{\phi}(\mathbf{x}')\|_{\mathcal{H}} = \sqrt{K(\mathbf{x},\mathbf{x}) + K(\mathbf{x}',\mathbf{x}') - 2K(\mathbf{x},\mathbf{x}')}.$$

This property can be useful to estimate the degree of nonlinear distortion introduced by a particular kernel, or a particular choice of its parameters.

Normalization. Exploiting the previous property, one can also normalize data in feature space:

$$K(\mathbf{x},\mathbf{x}') \leftarrow \left\langle \frac{\boldsymbol{\phi}(\mathbf{x})}{\|\boldsymbol{\phi}(\mathbf{x})\|}, \frac{\boldsymbol{\phi}(\mathbf{x}')}{\|\boldsymbol{\phi}(\mathbf{x}')\|} \right\rangle = \frac{K(\mathbf{x},\mathbf{x}')}{\sqrt{K(\mathbf{x},\mathbf{x})K(\mathbf{x}',\mathbf{x}')}}$$

Note that for some particular kernel functions, e.g. those with sample self-similarities $K(\mathbf{x},\mathbf{x}) = 1$ as in radial basis function kernels, normalization does not impact the resulting kernel.

Representer's theorem (Kimeldorf and Wahba, 1971)

A very relevant property in the context of kernel methods is the Representer's theorem. The statement of the theorem presented here is a particular example of (Kimeldorf and Wahba, 1971; Schölkopf et al., 2001):

Theorem 1. *Let $\mathcal{X}$ be a non-empty set and K a positive-definite real-valued kernel on $\mathcal{X} \times \mathcal{X}$ with corresponding reproducing kernel Hilbert space $\mathcal{H}$. Given a training sample $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n) \in \mathcal{X} \times \mathbb{R}$, a strictly monotonically increasing real-valued function $\Omega\colon [0,\infty) \to \mathbb{R}$, and an arbitrary empirical risk function $V\colon (\mathcal{X} \times \mathbb{R}^2)^m \to \mathbb{R} \cup \{\infty\}$, then for any $f^* \in \mathcal{H}$ satisfying*

$$f^* = \text{argmin}_{f\in\mathcal{H}} \left\{ V\left((\mathbf{x}_1, y_1, f(\mathbf{x}_1)), ..., (\mathbf{x}_n, y_n, f(\mathbf{x}_n))\right) + \Omega\left(\|f\|^2_{\mathcal{H}}\right)\right\},$$

f^* admits a representation of the form:

$$f^*(\cdot) = \sum_{i=1}^{n} \alpha_i K(\cdot, \mathbf{x}_i),$$

where $\alpha_i \in \mathbb{R}$ for all $1 \leq i \leq n$.

This theorem is of special relevance to develop and analyze kernel machines, as many machine learning problems can be defined in such general form, and hence we are given a common representation of the solution. We should stress the generality of both the loss function, V, and the regularization term, Ω. Note that the cost function may actually adopt complex penalization terms depending not only on the labels and the predictions but also on the samples, hence opening the field to cost-sensitive learning. On the other hand, the regularization term admits flexible definitions, thus allowing to enforce smooth solutions of f, and opening the field to design efficient priors and encoding invariances. Also note that, when plugging an admissible representation of f into $\|f\|_{\mathcal{H}}$, the regularizer reduces to an energy constraint depending on the solution and the kernel matrix, $\boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathbb{R}^{n\times 1}$. This observation returns the discussion back to the centrality and relevance of the kernel definition not only to allow flexible signal approximations but also as powerful regularizer.

2.2.4 Standard kernels

As we have seen before, the bottleneck for any kernel method is the definition of a feature mapping $\boldsymbol{\phi}$ that maps data to a feature space endorsed with a dot product. The naive idea would be to design such mappings. However, we have already observed that this is actually not necessary, provided that we can construct reproducing kernel functions that may accurately reflect the similarity among mapped samples. However, a new problem arises: not all kernel similarity functions are permitted. In fact, valid kernels are only those fulfilling Mercer's Theorem (roughly speaking, being positive definite similarity matrices). The most common ones are: the linear $K(\mathbf{x},\mathbf{z}) = \mathbf{x}^\top\mathbf{z}$, the polynomial $K(\mathbf{x},\mathbf{z}) = (\mathbf{x}^\top\mathbf{z}+1)^d$, $d \in \mathbb{R}^+$, and the Radial Basis Function (RBF), $K(\mathbf{x},\mathbf{z}) = \exp\left(-\|\mathbf{x}-\mathbf{z}\|^2/(2\sigma^2)\right)$, $\sigma \in \mathbb{R}^+$. Note that, by Taylor series expansion, the RBF kernel can be casted as a polynomial kernel with infinite degree. Thus the corresponding Hilbert space is infinite dimensional, which corresponds to a mapping into the space of smooth functions $\mathbb{C}^\infty$. The RBF kernel is also of practical convenience –stability and only one parameter to be tuned–, and it is the preferred kernel function in standard applications. The use of RBF kernels also allows some connections to the theory of density estimation, Parzen's windows and theoretic-information learning, as we will see later in this chapter, and extensively in a kernel method proposed in the Thesis.

2.2.5 Kernel development

Taking advantage of some algebra and functional analysis properties (Golub and Van Loan, 1996; Reed and Simon, 1980), one can derive very useful properties of kernels. Be K_1 and K_2 two positive definite kernel functions on $\mathcal{X} \times \mathcal{X}$, $\mathbf{A}$ a symmetric positive (semi)definite matrix, f any increasing, differentiable, monotone and enclosed function, and $\mu > 0$. Then, the following kernels are valid (Schölkopf and Smola, 2002):

$$
\begin{aligned}
K(\mathbf{x},\mathbf{x}') &= K_1(\mathbf{x},\mathbf{x}') + K_2(\mathbf{x},\mathbf{x}') && (2.2)\\
K(\mathbf{x},\mathbf{x}') &= \mu K_1(\mathbf{x},\mathbf{x}') && (2.3)\\
K(\mathbf{x},\mathbf{x}') &= K_1(\mathbf{x},\mathbf{x}') \cdot K_2(\mathbf{x},\mathbf{x}') && (2.4)\\
K(\mathbf{x},\mathbf{x}') &= K(f(\mathbf{x}), f(\mathbf{x}')) && (2.5)
\end{aligned}
$$

These basic properties give rise to the construction of refined similarity measures that could be better fitted to the data characteristics. In remote sensing data processing, one can sum dedicated kernels to spectral, contextual or even temporal information of pixels through (2.2). A scaling factor to each kernel can also be added (Eq. 2.3). Also, we want to stress property (2.5): if a particular preprocessing point-wise function f is useful one can either apply it before hand or alternatively find a kernel function that implicitly encodes it. This latter property will be used indirectly when proposing probabilistic kernels, being such f the maximum a posteriori probability induced by a clustering algorithm.

There are many additional tricks and tips to construct kernel functions from previous ones. A complete review can be found in (Shawe-Taylor and Cristianini, 2004). Nowadays the field of kernel construction is very active, and one may find kernels to deal with heterogeneous data sources (e.g. strings, images, language, etc.), kernels that combine multiple modalities, and kernels designed to deal with complex data representations (graphs, tensors, fractals), just to name a few. In what follows, we simply review some recent advances for kernel development that are important to this Thesis:

Convex combinations. By exploiting (2.2) and (2.3), one can build kernels by linear combinations of kernels working on feature subsets:

$$K(\mathbf{x},\mathbf{x}') = \sum_{m=1}^{M} d_m K_m(\mathbf{x},\mathbf{x}').$$

This field of research is known as multiple kernel learning (MKL), and different algorithms exist to optimize the weights and kernel parameters jointly (Rakotomamonjy et al., 2008). Note that this kernel offers some insight in the problem, since relevant features receive higher values of d_m, and the corresponding kernel parameters yield information about pairwise similarity scales. In the context of this thesis we will focus merely on simple composite functions to derive semisupervised feature extraction algorithms. Nevertheless, it does not escape to our knowledge that MKL extensions could be explored as well.

Generative kernels. Exploiting Eq. (2.5), one can construct kernels from probability distributions by defining $K(\mathbf{x},\mathbf{x}') = K(\mathbf{p},\mathbf{p}')$, where $\mathbf{p}$, $\mathbf{p}'$ are defined on the space $\mathcal{X}$. This kind of kernels is known as *probability product kernels between distributions* and is defined as:

$$K(\mathbf{p},\mathbf{p}') = \langle \mathbf{p},\mathbf{p}' \rangle = \int_{\mathcal{X}} \mathbf{p}(\mathbf{x})\mathbf{p}'(\mathbf{x})\mathrm{d}\mathbf{x}.$$

This kernel construction is implicit behind our proposals for unsupervised learning in chapter 5. On the one hand, generative kernels can be seen as particular constructions to estimate densities, hence somewhat related to our optimized kernel entropy component analysis. On the other hand, and with a more explicit relation, we will exploit posterior probabilities to construct multiscale generative kernels in our probabilistic cluster kernel.

2.3 Examples of kernel methods

In this section, we summarize the most important instantiations of kernel methods for classification, clustering, density estimation and dependence estimation. These are the central problems in this Thesis. Consequently, we intentionally do not treat the vast literature of kernel methods for regression, function approximation, time series analysis, signal and target detection, visualization or sorting. It is also worth noting that all the reviewed methods have provided excellent performance in remote sensing data analysis.

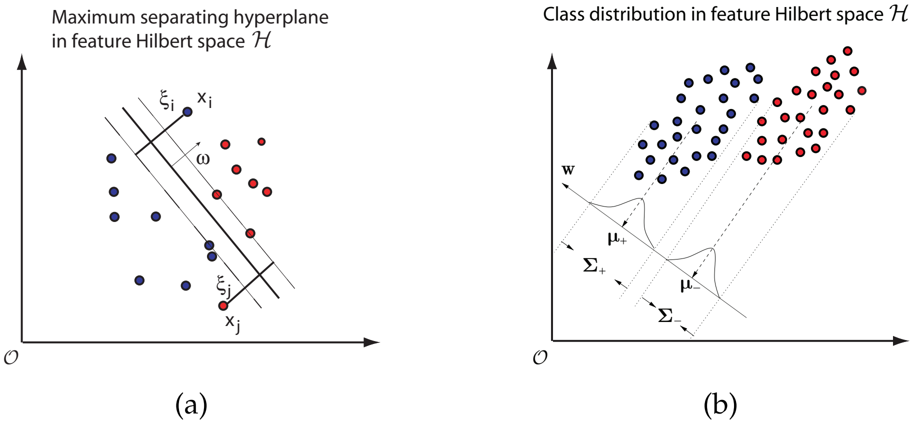

Maximum separating hyperplane in feature Hilbert space $\mathcal{H}$
ξ_i
x_i
ω
ξ_j
x_j
$\mathcal{O}$
(a)
Class distribution in feature Hilbert space $\mathcal{H}$
$\mathbf{w}$
μ_+
Σ_+
μ_-
Σ_-
$\mathcal{O}$
(b)

Figure 2.1: Illustration of kernel classifiers. (a) SVM: Linear decision hyperplanes in a nonlinearly transformed, feature space, where *slack* variables ξ_i are included to deal with errors. (b) KFD: Kernel Fisher's Discriminant separates the classes by projecting them onto a hyperplane where the difference of the projected means (μ_1, μ_2) is large, and the variance around means σ_1 and σ_2 is small. Source: (Gómez-Chova et al., 2011).

2.3.1 Support Vector Machine (SVM)

The most important development of a kernel method is the Support Vector Machine (SVM) for classification (Vapnik, 1998). The method has found wide application in many subfields of Engineering and Science, and nowadays it constitutes the main classification algorithm in remote sensing data processing as well. The reasons for this widespread adoption by our community is the excellent performance in very high dimensional feature spaces and low number of examples, robustness to different noise levels and sources, and the capability to combine heterogeneous information sources via kernel construction (Camps-Valls and Bruzzone, 2005; Camps-Valls et al., 2006b; Camps-Valls and Bruzzone, 2009). The SVM is a supervised linear classifier that tries to separate classes maximally. This is why it is also known as a *maximum margin classifier*. In order to classify data, the SVM finds the hyperplane with largest margin between the two classes (Fig. 2.1). Such operation can be done in the original input space, hence yielding a linear classifier, or implicitly in feature space, thus leading to a nonlinear classifier. Either way, the hyperplane is defined by the points called support vectors, which lie on the margin.

In the typically more powerful case of the nonlinear SVM, first the pixels are mapped to a higher dimensional space, $\Phi : \mathbb{R}^d \to \mathcal{H}$. Then, a linear classifier is used therein, and hence the mapped pixels are classified linearly with maximum margin in $\mathcal{H}$. Notationally, we are given a labeled training data set $\{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^N$ and $y_i \in \{-1, +1\}$, and given a nonlinear mapping $\boldsymbol{\phi}(\cdot)$, the SVM method solves:

$$\min_{\mathbf{w},\xi_i,b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \right\} \tag{2.6}$$

constrained to:

$$y_i(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b) \geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \tag{2.7}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, n \tag{2.8}$$

where $\mathbf{w}$ and b define a linear classifier in the feature space, and ξ_i are positive slack variables enabling to deal with permitted errors (Fig. 2.1a). Appropriate choice of the nonlinear mapping $\boldsymbol{\phi}$ guarantees that the transformed samples are *more likely* to be linearly separable in the (higher dimension) feature space. The regularization parameter C controls the generalization capability of the classifier, and it must be selected by the user. Primal problem (2.6) is solved using its dual problem counterpart (Schölkopf and Smola, 2002), and the decision function for any test vector $\mathbf{x}_*$ is finally given by

$$f(\mathbf{x}_i) = \text{sing}(\sum_{i=1}^{n} y_i \alpha_i \underbrace{\boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_j)}_{\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)} + b)$$

where α_i are Lagrange multipliers corresponding to constraints in (2.7), being the support vectors (SVs) those training samples $\mathbf{x}_i$ with non-zero Lagrange multipliers $\alpha_i \neq 0$; $K(\mathbf{x}_i, \mathbf{x}_*)$ is an element of a kernel matrix $\mathbf{K}$ defined as in equation (2.1); and the bias term b is calculated by using the *unbounded* Lagrange multipliers as $b = 1/k \sum_{i=1}^{k}(y_i - \langle \boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w} \rangle)$, where k is the number of *unbounded* Lagrange multipliers ($0 \leqslant \alpha_i < C$) and $\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \boldsymbol{\phi}(\mathbf{x}_i)$ (Schölkopf and Smola, 2002). It is worth noting that, as in any other kernel method, the classification only depends on the dot products between the transformed samples which is implicitly estimated through the kernel function. Therefore, even in the nonlinear case, we do not need to know the mapping Φ explicitly, only the dot products among mapped samples, that is, the kernel function among the samples.

The SVM has extensively reported good accuracy in remote sensing applications, as reported in Camps-Valls and Bruzzone (2009). SVMs have been applied to both multispectral (Chen et al., 2012; Dalponte et al., 2012) and hyperspectral (Tarabalka et al., 2010; Li et al., 2011a; Chen et al., 2013) data in a wide range of domains, including object recognition (Duro et al., 2012), land cover and multi-temporal classification (Silva et al., 2011; Petropoulos et al., 2012; Leiva-Murillo et al., 2013), urban monitoring (Kamusoko et al., 2013) and agriculture land mapping (Amorós-López et al., 2011; Zolfaghari et al., 2013).

2.3.2 Kernel Fisher's Discriminant (KFD)

The kernel Fisher's discriminant (KFD) analysis algorithm is a common method for supervised data classification. The algorithm extends the linear Fisher's discriminant to the nonlinear case with kernels. The use of this method in remote sensing data processing has found many applications (Kuo et al., 2009; Jia et al., 2013). Interestingly, we should mention here that close connections between Fisher's discriminant analysis and many multivariate data analysis techniques have been established in the literature (Arenas-García et al., 2013). These links extend to the kernel counterparts as well, and this is the main motivation of reviewing KFD in this chapter.

Notationally, let us assume that, n_1 out of n training samples belong to class -1 and n_2 to class $+1$, so $n = n_1 + n_2$. Let $\boldsymbol{\mu}$ be the mean of the whole set, and $\boldsymbol{\mu}_-$ and $\boldsymbol{\mu}_+$ the means for classes -1 and $+1$, respectively. Analogously, let $\boldsymbol{\Sigma}$ be the covariance matrix of the whole set, and $\boldsymbol{\Sigma}_-$ and $\boldsymbol{\Sigma}_+$ the covariance matrices for the two classes.

The Linear Fisher's Discriminant (LFD) seeks for projections that maximize the interclass variance and minimize the intraclass variance (Fisher, 1936; Hastie et al., 2009). By defining the *between class scatter matrix* $\mathbf{S}_B = (\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)(\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)^\top$ and the *within class scatter matrix* $\mathbf{S}_W = \boldsymbol{\Sigma}_- + \boldsymbol{\Sigma}_+$, the problem reduces to maximize

$$J(\mathbf{w}) = \frac{\mathbf{w}^\top \mathbf{S}_B \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_W \mathbf{w}} \tag{2.9}$$

The Kernel Fisher's Discriminant (KFD) is obtained by defining the LFD in a high dimensional *feature* space $\mathcal{H}$. Now, the problem reduces to maximize:

$$J(\mathbf{w}) = \frac{\mathbf{w}^\top \mathbf{S}_B^{\boldsymbol{\phi}} \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_W^{\boldsymbol{\phi}} \mathbf{w}} \tag{2.10}$$

where now $\mathbf{w}$, $\mathbf{S}_B^{\boldsymbol{\phi}}$ and $\mathbf{S}_W^{\boldsymbol{\phi}}$ are defined in $\mathcal{H}$, $\mathbf{S}_B^{\boldsymbol{\phi}} = (\boldsymbol{\mu}_-^{\boldsymbol{\phi}} - \boldsymbol{\mu}_+^{\boldsymbol{\phi}})(\boldsymbol{\mu}_-^{\boldsymbol{\phi}} - \boldsymbol{\mu}_+^{\boldsymbol{\phi}})^\top$, and $\mathbf{S}_W^{\boldsymbol{\phi}} = \boldsymbol{\Sigma}_-^{\boldsymbol{\phi}} + \boldsymbol{\Sigma}_+^{\boldsymbol{\phi}}$. We need to express (2.10) in terms of dot-products only. According to the reproducing kernel theorem (Schölkopf and Smola, 2002), any solution $\mathbf{w} \in \mathcal{H}$ can be represented as a linear combination of training samples in $\mathcal{H}$. Therefore $\mathbf{w} = \sum_{i=1}^n \alpha_i \boldsymbol{\phi}(\mathbf{x}_i)$ and then

$$\mathbf{w}^\top \boldsymbol{\mu}_i^{\boldsymbol{\phi}} = \frac{1}{n_i} \sum_{j=1}^{n} \sum_{k=1}^{n_i} \alpha_j K(\mathbf{x}_j, \mathbf{x}_k^i) = \boldsymbol{\alpha}^\top \mathbf{M}_i \tag{2.11}$$

where $\mathbf{x}_k^i$ represents samples of class i, and $(\mathbf{M}_i)_j = \frac{1}{n_i} \sum_{k=1}^{n_i} K(\mathbf{x}_j, \mathbf{x}_k^i)$. Taking the definition of $\mathbf{S}_B^{\boldsymbol{\phi}}$ and (2.11), the numerator of (2.10) can be rewritten as $\mathbf{w}^\top \mathbf{S}_B^{\boldsymbol{\phi}} \mathbf{w} = \boldsymbol{\alpha}^\top \mathbf{M} \boldsymbol{\alpha}$, and the denominator as $\mathbf{w}^\top \mathbf{S}_W^{\boldsymbol{\phi}} \mathbf{w} = \boldsymbol{\alpha}^\top \mathbf{N} \boldsymbol{\alpha}$, where

$$\mathbf{M} = (\mathbf{M}_- - \mathbf{M}_+)(\mathbf{M}_- - \mathbf{M}_+)^\top \tag{2.12}$$

$$\mathbf{N} = \textstyle\sum_{j=\{-1,+1\}} \mathbf{K}_j (\mathbf{I} - \mathbf{1}_{n_j}) \mathbf{K}_j^\top \tag{2.13}$$

$\mathbf{K}_j$ is a $n \times n_j$ matrix with $(\mathbf{K}_j)_{nm} = K(\mathbf{x}_n, \mathbf{x}_m^j)$ (the kernel matrix for class j), $\mathbf{I}$ is the identity matrix and $\mathbf{1}_{n_j}$ a matrix with all entries set to $1/n_j$. Finally, Fisher's linear discriminant in $\mathcal{H}$ is solved by maximizing

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^\top \mathbf{M} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top \mathbf{N} \boldsymbol{\alpha}}, \tag{2.14}$$

which is solved as in the linear case. The projection of a new sample $\mathbf{x}$ onto $\mathbf{w}$ can be computed through the kernel function:

$$\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) \tag{2.15}$$

2.3.3 Kernel k-means

Kernel methods have been also used to cluster data in general and remote sensing data in particular. Note that, when clusters are compact and well separated, linear algorithms may work well. Nevertheless, in other cases data contain arbitrarily shaped clusters of different densities and hence clusters are not linearly separable. In such cases, separation may be easier in a high dimensional feature space. The kernel k-means algorithm extends the linear k-means to RKHS by means of mapping functions $\boldsymbol{\phi}(\cdot)$. Since the k-means formulation can be expressed solely in terms of dot products, kernel functions can replace these expressions returning the value of the dot product in the RKHS directly.

Let $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ be a set of samples, mapped into a Hilbert space $\mathcal{H}$, thus giving $\{\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \dots, \boldsymbol{\phi}(\mathbf{x}_n)\}$. Then, the distances of each data to every centroid $j = 1, \dots, q$ is (Girolami, 2002a):

$$\sum_{i=1}^{n} \sum_{j=1}^{q} \left\| \boldsymbol{\phi}(\mathbf{x}_i) - \tilde{\mathbf{m}}_j \right\|^2, \tag{2.16}$$

where $\tilde{\mathbf{m}}_j = \frac{1}{n_j} \sum_{j=1}^{n} p_{i,j} \boldsymbol{\phi}(\mathbf{x}_i)$ are the centroids of the clusters into the feature space. Substituting $\tilde{\mathbf{m}}_j$ in equation (2.16) and applying the kernel trick (Schölkopf and Smola, 2002), we obtain the optimization problem of kernel k-means:

$$\mathbf{P}^* = \arg\max_{\mathbf{P}} \ \text{Tr}\{\mathbf{P}\mathbf{K}\mathbf{P}^\top\}, \tag{2.17}$$

where $\mathbf{K}$ is the kernel matrix and $\mathbf{P}^*$ is the optimal normalized cluster membership matrix.

2.3.4 Hilbert-Schmidt Independence Criterion (HSIC)

A recurrent problem in machine learning and signal processing is that of estimating dependencies between random variables. The most widely used method is mutual information (MI), which extends correlation accounting for higher-order dependencies (Cover and Thomas, 2005). The mutual information between two discrete unidimensional variables $\mathbf{x}, \mathbf{y} \in \mathbb{R}$ can be defined as:

$$I(\mathbf{x}, \mathbf{y}) = \sum_{y} \sum_{x} p(\mathbf{x}, \mathbf{y}) log \left(\frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x}) p(\mathbf{y})} \right),$$

where $p(\mathbf{x}, \mathbf{y})$ is the joint probability and $p(\mathbf{x})$ and $p(\mathbf{y})$ represent the marginal probability distributions. When the input and output data are independent, MI tends to zero.

The Hilbert-Schmidt Independence Criterion (HSIC) is a simple and effective method to estimate statistical dependence between possibly multidimensional random variables, extends linear correlation and can be computed very efficiently. Its basic idea is to evaluate all possible correlations in a reproducing kernel Hilbert space, which can be performed efficiently via the kernel trick (Gretton et al., 2005). For random variables $\mathbf{x} \in \mathbb{R}^{d_x}, \mathbf{y} \in \mathbb{R}^{d_y}$, HSIC corresponds to estimating the norm of the cross-covariance in feature space, whose empirical (biased) estimator is:

$$\text{HSIC} = \frac{1}{(n-1)^2} \text{Tr}(\mathbf{K}_x \mathbf{K}_y) = \frac{1}{(n-1)^2} \text{Tr}(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top \boldsymbol{\Psi}\boldsymbol{\Psi}^\top),$$

where $\mathbf{K}_x$ and $\mathbf{K}_y$ are kernels working on $\mathbf{x}$ and $\mathbf{y}$ respectively, and $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$ are the mapped data $\mathbf{x}$ and $\mathbf{y}$, respectively. When the input and output data are independent, HSIC tends to zero. Due to the kernelization, the empirical HSIC only depends on computable matrices of size $n \times n$. Both measures (HSIC and MI) will be used in this Thesis to measure differences between features obtained by several methods and to compare dissimilarity between kernels.

2.3.5 Kernel Density Estimation (KDE)

Kernel algorithms have been proposed to estimate the probability density function (pdf) of a random variable (Hwang et al., 1994), and lately in remote sensing (Mantero et al., 2005). An easy way of density estimation is by means of histograms (Pearson, 1895) that divides the samples space in bins and estimate the density by the fraction of the number of training data falling into the corresponding bin:

$$p_H(\mathbf{x}) = \frac{1}{n}\frac{\# \, of \; \mathbf{x}_{train} \; in \; same \; bin \; that \; \mathbf{x}}{width \; of \; bin}.$$

Note that the density estimation using histograms has several drawbacks. The histogram needs two parameters to be defined (number of bins and bin width), the results of the probability density estimation depends on the starting position of the bins, the discontinuities are due to the choice of bin locations and, some precaution should be taken with the dimensionality of the samples because of the exponential growth of the number of bins with the number of dimensions.

Another way to estimate the pdf is by means of *Parzen Windows* (Parzen, 1962). Supposing a region $\mathcal{R}$ defined by a hypercube. We can find an expression for the number of samples falling in this region defining a kernel function $K(\mathbf{x})$ that is also known as *Parzen window* which is given by

$$K(\mathbf{x}) = \begin{cases} 1 & \text{if } |\mathbf{x}^j| < \frac{1}{2} \;\; \forall j = 1, \ldots, d \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{x}^j$ is the value of the sample $\mathbf{x}$ in the j dimension and d is the number of dimensions. Thus, $K(\mathbf{x})$ is a unity cube centered at the origin that is equal to one if sample $\mathbf{x}$ falls into the hypercube, and zero otherwise. The samples that fall into a hypercube with side σ and center at $\mathbf{x}$ is

$$n_{bin} = \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{\sigma}\right)$$

where n is the total number of samples and σ is the *smoothing parameter* or *kernel width*. Therefore the density estimation at $\mathbf{x}$ by means of Parzen windows is

$$\hat{p}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sigma}K\left(\frac{\mathbf{x} - \mathbf{x}_i}{\sigma}\right).$$

This estimation can be seen as a superposition of n cubes of side σ, with one hypercube centered at each sample $\mathbf{x}_i$. The Parzen window estimation is similar to density estimation based on histograms. The difference is the use of hypercubes in Parzen estimation instead of bins intervals. However, the discontinuity problem is still present. To solve this problem, one common solution is to use a smooth kernel function $\int_{\S} K(\mathbf{x})d\mathbf{x} = 1$. If the kernel is a positive semi-definite function, it will fulfill the Mercer′ conditions and will link the Information Theory (IT) with kernel methods (Jenssen, 2009). The most commonly used kernel function is the Gaussian kernel

$$G(\mathbf{x}, \mathbf{x}_i|\sigma) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{1}{2}} e^{-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2^2\sigma}},$$

thus the pdf estimation is

$$\hat{p}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sigma}G(\mathbf{x}, \mathbf{x}_i|\sigma).$$

2.3.6 Kernel bandwidth selection

A crucial problem in kernel methods is to correctly select the kernel parameters. In the Gaussian RBF kernel, high values of kernel bandwidth, σ, yield an over-smoothed sample similarity (or density estimation for KDE) and mask the structure of the data while a small σ yields a peaky kernel matrix (or density estimation) very hard to interpret. Then, we would like to find a value of σ that maximizes the accuracy in classification or regression problems or that minimizes the error between the estimated density and the true density in density estimation problems. There are several ways to select the bandwidth. Here we summarize the most common ones used in this Thesis:

- Classification accuracy score by means of cross-validation is a supervised method that estimates the optimal bandwidth calculating the overall accuracy of the probabilistic classifier, that is, we calculate the probabilistic distribution *per* class for each sample and assign the class with higher probability to each sample. We obtain the overall accuracy for each K-fold and the K results are averaged to produce a single estimation. The maximum overall accuracy obtained gives us the optimum bandwidth (σ^*).

- Maximum likelihood (ML) *leave-one-out* is an algorithm for variable bandwidth estimation particularly in spaces of high dimensionality (Barnard, 2010). The Maximum Likelihood finds the parameters that maximize the probability distribution. Here, we use the leave-on-out method to fix the bandwidth. The leave-one-out consists on randomly partitioning the data set in K subsets being K the number of samples into the data set. Out of the K subsets, $K-1$ subsets are used as a training and the sample that we leave out is used to validate testing model. Then, the process is repeated K times using each of the K subsets exactly once as the validation data. The K results are averaged (or otherwise combined) to produce a single estimation.

- Silverman's rule (Silverman, 1986) is the classical rule of thumb in KDE and estimates the optimal bandwidth σ by means of the next formula:

  $$\sigma^* = \frac{1}{\sigma_i}\left(\frac{4}{n(d+2)}\right)^{1/(d+4)}, \tag{2.18}$$

  where n is the number of observations, d is the number of dimensions, and σ_i is the standard deviation *per* dimension, $i = 1, \ldots, d$.

2.4 Summary

This Chapter reviewed the main characteristics and properties of the framework of kernel methods. We have first summarized some key developments of kernel methods in remote sensing data analysis, with special focus on classification and regression developments, as they will be the main illustrative applications throughout this Thesis. We have payed special attention to the main properties of kernel functions, as we will rely on some properties of functional analysis to build new kernel functions by combination of elementary ones. Furthermore, the representer's theorem has been defined and different definitions of kernel mapping functions have been summarized. They will play a role in the definition of out-of-sample projections of the proposed kernel feature extraction methods. After this, we reviewed several state-of-the-art kernel methods for classification, clustering, dependence, and density estimation. They were included here as a mere illustration of the powerfulness of the framework, and for the sake of completeness. Kernel feature extraction will be summarized extensively in the following chapter, as it constitutes the core of the Thesis. Summarizing, we have seen that kernel methods allow to easily develop nonlinear algorithms through the use of reproducing kernel functions. Kernel methods can cope with problems involving high dimensionality and low number of examples, as well as permit the combination of heterogeneous sources of information. All these advantages make them a solid and convenient framework for pattern analysis in general and remote sensing data processing in particular.

Chapter 3

Kernel Multivariate Analysis in Remote Sensing Data Processing

Contents

In this Chapter, we will summarize different feature extraction methods and their use. Very often the terms of feature extraction, dimensionality reduction and manifold learning methods are used indistinctly given the tight relations between them. Feature extraction is exploited as a preprocessing step before classification and regression but classification or regression methods lead to black-box models which must be designed specifically and attached to the following application, the underlying idea in feature extraction is to find an appropriate data representation (typically via projection operators) which can then be used in *any* arbitrary application. This different perspective of addressing a problem leads to some interesting properties.

Essentially, both feature extraction and manifold learning seek for transformations (linear and nonlinear, parametric or not) of the data to a representation space that allows to capture most of the information of the data in fewer components. Feature extraction is typically concerned with matrix transformations such that the new data is embedded in a lower dimensional subspace. Manifold learning pursues a more challenging goal, and looks for transformations that describe the (typically nonlinear) characteristics of the data (Lee and Verleysen, 2007).

In recent years, a plethora of nonlinear dimensionality reduction methods has been presented trying to deal with manifolds that cannot be described with linear methods, such as the classical principal component analysis (PCA) (Jolliffe, 2010). See for example (Lee and Verleysen, 2007) for a comprehensive review on manifold learning and nonlinear feature extraction. Approaches to the problem range from local methods (Tenenbaum et al., 2000; Roweis et al., 2002; Verbeek et al., 2002; Teh and Roweis, 2003; Brand, 2003), kernel-based and spectral decompositions (Roweis and Saul, 2000; Schölkopf et al., 1998; Weinberger and Saul, 2004), neural networks (Kramer, 1991; Hinton and Salakhutdinov, 2006; Scholz et al., 2007), and projection pursuit approaches (Huber, 1985; Laparra et al., 2011).

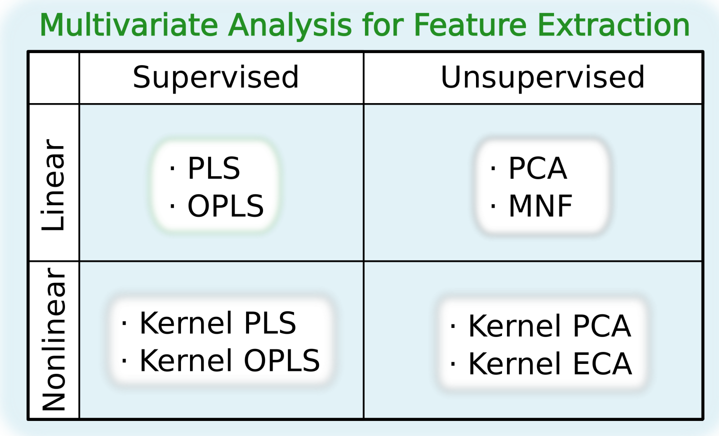
In this context, Multivariate Analysis (MVA) techniques constitute a family of methods to extract features, which has been used in several scientific areas (Arenas-García et al., 2013). In the following sections, we will review the main MVA methods used in the Thesis. We will start with the methods that take into account linear input-output relations, i.e. linear methods. Afterwards, we will continue with the nonlinear kernel MVA methods. We will conclude the chapter with a summary of the main properties of feature extraction methods.

3.1 Feature extraction applications in remote sensing data processing

Feature extraction has become an important topic in remote sensing data processing mainly due to the high dimensionality of data, as well as the high redundancy both between spectral bands and between neighboring pixels. This can cause the curse of dimensionality (Bellman, 1961), or the Hughes' phenomenon (Hughes, 1968). The problem is ubiquitous in remote sensing image analysis. Moreover, the high-dimensionality of remote sensing data is often increased by stacking spatial, spectral, temporal and multiangular features to the spectral channels for modelling additional information sources. In order to reduce the problems of dimensionality, feature extraction methods have been used in several remote sensing problems. Despite the fact that the most common method in remote sensing is PCA, there are also other useful methods depending on the application. The PLS method is applied in regression cases such as in (Hansen and Schjoerring, 2003). In this study, the authors did a regression analysis using hyperspectral images and they concluded that PLS regression analysis may provide a useful exploratory and predictive tool when applied to hyperspectral reflectance data. But in the case of remote sensing data processing, PLS has also been used in particular applications, such as mapping canopy nitrogen (Coops et al., 2003; Townsend et al., 2003), classifying salt marsh plants (Wilson et al., 2004), analyzing biophysical properties of forests (Naesset et al., 2005), and retrieving leaf fuel moisture (Li et al., 2007). In order to mitigate problems due to radiometric differences and noise, canonical correlation analysis (CCA/MAD) plus minimum noise fraction (MNF/MAF) transformations are becoming popular in unsupervised change detection (Nielsen, 2007). These methods, however, can only deal with affine transforms and additive uncorrelated Gaussian noise, and cannot be easily adapted to find a particular type of feature.

Remote sensing data often show nonlinearities: Backscattering, illumination changes, twisted distribution data or the relation between the reflectance with their parameters (Friedl and Brodley, 1997; Paola and Schowengerdt, 1995; Camps-Valls et al., 2008). This characteristic is very common in remote sensing data specially in supervised or semisupervised problems. The use of kernels is suited for remote sensing data processing (Camps-Valls and Bruzzone, 2009) due to the robustness to high dimensional images, to noise and to low number of labeled samples. In recent years, kernel methods have emerged as an excellent tool to develop nonlinear feature extraction methods (Camps-Valls and Bruzzone, 2009). Actually, most common linear feature extraction algorithms have been kernelized and applied to target detection (Kwon and Nasrabadi, 2005), channel selection (Serpico and Moser, 2007), classification of hyperspectral images (Li et al., 2011b), noise reduction (Gómez-Chova et al., 2011), and also to unsupervised change detection (Volpi et al., 2012). In addition, there are some works based on information theory such as (Gómez-Chova et al., 2012) that used Kernel Entropy Component Analysis (KECA) feature extraction method for clustering remote sensing data. Sparse kernel methods have also emerged in remote sensing due to computational burden. Arenas-García and Camps-Valls (2008) presented a reduced rank complexity KOPLS method (rKOPLS) for feature extraction. In (Arenas-García et al., 2013), the authors compared linear and kernel feature extraction methods in temperature estimation from infrared sounding data, and in hyperspectral image classification.

We can find other nonlinear feature extraction methods based on kernelization applied to multitemporal analysis such (Muñoz Marí et al., 2013), (Gómez-Chova et al., 2013). In the first work, the authors proposed the generalization of Kernel Canonical Correlation Analysis (KCCA) for several datasets, and applied it to multitemporal image classification. In the second work, the authors proposed a kernel change detection analysis (KCDA) that can be easily adapted to find a specific change of interest in satellite image time series while discarding undesired changes.



Multivariate Analysis for Feature Extraction

	Supervised	Unsupervised
Linear	· PLS · OPLS	· PCA · MNF
Nonlinear	· Kernel PLS · Kernel OPLS	· Kernel PCA · Kernel ECA

Figure 3.1: Different types of multivariate analysis for feature extraction used in this Thesis.

3.2 Introduction to multivariate analysis

Statistical multivariate analysis for feature extraction is commonly used to determine a system with less variables. This analysis synthesizes the information of the original system reducing its dimensionality. We use MVA in order to obtain a subset of independent variables (or features) from the original set by projecting the samples onto the most relevant directions of the data manifold. The projections are obtained by means of mathematical transformations: $f : \mathbb{R}^d \rightarrow \mathbb{R}^{d_f}$ where d is the dimension of samples in original space and d_f is the dimension in the transformed space, typically $d_f \leq d$. Depending on the available labeled data (supervised and unsupervised) and transformations (linear, nonlinear) that we will apply, the feature extraction methods can be divided in different groups. Figure 3.1 summarizes the different MVA methods used throughout the Thesis as starting points for our developments. Among all feature extraction methods, the two most common methods are principal component analysis (PCA) (Jolliffe, 2010) and partial least squares (PLS) (Rosipal and Krämer, 2006). Other methods focus on including information about the noise, such as the minimum noise fraction (MNF) transform (Green et al., 1988) or the related noise-adjusted principal components (NAPC) (Blackwell, 2005).

All previous methods assume that there exists a *linear* relation between the original features. In many situations, this linearity assumption does not hold, and a nonlinear feature extraction is needed to obtain acceptable performance. Different nonlinear versions of PCA and PLS have been developed, which can address nonlinear problems either by local approaches (Roweis and Saul, 2000), neural networks (Kramer, 1991), or kernel-based algorithms (Shawe-Taylor and Cristianini, 2004).

In the following sections, we first fix the notation used throughout the Thesis, and then summarize the formulation, properties and relations of both linear and kernel multivariate analysis techniques.

Table 3.1: Specific notation for feature extraction methods.

n	Total number of samples
n_{train}	Total number of training samples
n_{test}	Total number of test samples
d	Dimension of input space
n_c	Dimension of output space
$\mathbf{X}$	Input data matrix(size $n \times d$)
$\mathbf{Y}$	Output data matrix(size $n \times n_c$)
$\tilde{\mathbf{X}}$	Centered input data matrix
$\tilde{\mathbf{Y}}$	Centered output data matrix
$\mathbf{C}_x, \mathbf{C}_y$	Input, output centered sample covariance matrices
$\mathbf{C}_{xy}$	Input-output centered sample cross-covariance matrix
d_f	Number of extracted features
$\mathbf{u}_i, \mathbf{v}_i$	ith projection vector for the input, output data
$\mathbf{U}, \mathbf{V}$	$[\mathbf{u}_1, \ldots, \mathbf{u}_{d_f}]$, $[\mathbf{v}_1, \ldots, \mathbf{v}_{d_f}]$. Projection matrices
$\mathbf{X}', \mathbf{Y}'$	Extracted features for the input, output data
$\mathcal{H}$	Reproducing kernel Hilbert Space
$\boldsymbol{\phi}(\mathbf{x})$	Mapping of $\mathbf{x}$ in feature space
$K(\mathbf{x}_i, \mathbf{x}_j)$	$\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle_{\mathcal{H}}$. Kernel function
$\boldsymbol{\Phi}$	$[\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)]^\top$. Input data in feature space
$\mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top$	Gram or Kernel Matrix
$\tilde{\mathbf{K}}$	Centered Gram or Kernel Matrix
$\mathbf{A}$	$[\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_{d_f}]$. Coefficients for $\mathbf{U} = \boldsymbol{\Phi}^\top \mathbf{A}$

3.3 Notation

Before reviewing the framework of MVA both in the linear and nonlinear cases, let us first fix the basic notation. Let $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ be a set of n data pairs, with $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{y}_i \in \mathbb{R}^{n_c}$ where d and n_c are the number of dimensions of the input and output data, respectively. By using matrix notation we can write, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^\top$, where superscript $^\top$ denotes matrix or vector transposition. We denote by $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ the centered versions of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Note that, the operation of centering removes the mean of every variable in the corresponding matrix. $\mathbf{C}_{xx} = \tilde{\mathbf{X}}^\top\tilde{\mathbf{X}}$ and $\mathbf{C}_{yy} = \tilde{\mathbf{Y}}^\top\tilde{\mathbf{Y}}$ are the sample covariance matrices of input and output data whereas $\mathbf{C}_{xy} = \tilde{\mathbf{X}}^\top\tilde{\mathbf{Y}}$ is the sample cross-covariance matrix.

In order to solve the MVA problem, we can obtain the projections via a linear transformation, $\tilde{\mathbf{X}}' = \tilde{\mathbf{X}}\mathbf{U}$, which for new test data $\mathbf{X}_*$ reduces to:

$$\tilde{\mathbf{X}}'_* \equiv \mathcal{P}(\tilde{\mathbf{X}}_*) = \tilde{\mathbf{X}}_*\mathbf{U}, \tag{3.1}$$

where $\mathbf{U}$ is the projection matrix to be estimated whose size is $d \times d_f$ and the projected test data $(\mathbf{X}_*)$ is a matrix of size $n_{test} \times d_f$.

Obtaining projections in kernel feature space for new test data $\mathbf{X}_*$ involves two operations. First, we have to map the data into the feature space, $\tilde{\boldsymbol{\Phi}}_*$. Second, we have to project these mapped data with $\mathbf{U}$, $\boldsymbol{\Phi}' = \boldsymbol{\Phi}\mathbf{U}$, which are expressed as a linear combination of the mapped samples, $\mathbf{U} = \tilde{\mathbf{\Phi}}^\top \mathbf{A}$ (Representer's Theorem, Section 2.2.3). Therefore the projected test data reduce to:

$$\mathcal{P}(\tilde{\mathbf{\Phi}}_*) = \tilde{\mathbf{\Phi}}_* \mathbf{U} = \tilde{\mathbf{\Phi}}_* \tilde{\mathbf{\Phi}}^\top \mathbf{A} = \tilde{\mathbf{K}}(\mathbf{X}_*, \mathbf{X})\mathbf{A}, \tag{3.2}$$

where $\mathbf{X}$ is the training data matrix, $\mathbf{A}$ columns contain the d_f extracted feature vectors produced by a specific kernel method, and $\tilde{\mathbf{K}}$ is the (centered) kernel containing as entries the similarities between $\mathbf{X}_*$ and $\mathbf{X}$, which are defined by the dot product between mapped samples $\tilde{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \tilde{\phi}(\mathbf{x}_i), \tilde{\phi}(\mathbf{x}_j) \rangle$ (more information about kernels and how to center data within Hilbert spaces in Section 2.2). The projected test data $\mathcal{P}(\tilde{\mathbf{\Phi}}_*)$ is a finite dimensional matrix of size $n_{test} \times d_f$. Table 3.1 summarizes the specific notation used in this chapter.

3.4 Linear multivariate analysis

As we have seen in the previous section, MVA statistical methods present linear and nonlinear versions. A common advantage of the linear methods is that they can be formulated using standard linear algebra, the solution reduces to data rotations so they are easily interpretable, and they can be implemented as standard or generalized eigenvalue problems that lead to convex optimization problems. There are several ways to implement these methods but we have focused on two approaches (more information in Appendix A): 1) in an iterative manner by calculating the top eigenvalue and deflating the matrix data, and 2) in a clockwise manner by solving the standard or generalized eigenvalue problems directly.

There are different approaches to linear MVA methods. Some disregard the target data (unsupervised) and some do not (supervised). There is another type of approaches regarding the kind of correlation criterion between variables which exploit the method to find a reduced set of relevant features. PCA seeks the directions of maximum variance of the data and it ignores the output data, i.e. class labels. On the contrary, PLS and orthonormalized PLS (OPLS), are supervised methods: while PLS looks for the directions that maximally align input and output data, OPLS reduces the Root Mean Square Error (RMSE) of the predictions using projected data. In this section, we will review the principles of PCA, PLS and OPLS.

3.4.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a widespread method for dimensionality reduction in real applications (Jolliffe, 2010). It consists in projecting the input data set onto the directions of largest input variance. Thus, PCA only considers the input data and does not take into account any target data set, i.e. it is an *unsupervised feature extraction* method. The criterion is expressed compactly as:

$$\begin{aligned} \text{PCA:} \quad & \mathbf{U} = \arg\max_{\mathbf{U}} \ \operatorname{Tr}\{\mathbf{U}^\top \mathbf{C}_{xx} \mathbf{U}\} \\ & \text{subject to: } \mathbf{U}^\top \mathbf{U} = \mathbf{I}, \end{aligned} \tag{3.3}$$

where $\mathbf{I}$ is the identity matrix of size d_f (number of extracted features), $\mathbf{C}_{xx} = \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$ is the centered sample covariance matrix of input data, and $\mathbf{U}$ is the projection matrix to be estimated.

Solving the problem in Equation (3.3) reduces to solve an eigenvalue problem of the input covariance matrix (more information in Appendix A):

$$\mathbf{C}_{xx}\mathbf{U} = \mathbf{\Lambda}\mathbf{U}, \tag{3.4}$$

where $\mathbf{\Lambda}$ is the square matrix whose main diagonal contains the eigenvalues, i.e. $\mathbf{\Lambda} = diag([\lambda_1, \ldots, \lambda_d])$ and $\mathbf{U}$ is the projection matrix, $\mathbf{U} \in \mathbb{R}^{d \times d_f}$.

There are different ways to solve the eigenvalue problem. One of them is by means of *Lagrange multipliers*, where the columns of the $\mathbf{U}$ matrix are the eigenvectors of the $\mathbf{C}_{xx}$ matrix associated with the d_f largest eigenvectors. The maximum number of features d_f is limited by $rank(\mathbf{C}_{xx})$ when $\mathbf{C}_{xx}$ has maximum rank, $d_f = d$. Another way to implement PCA is to extract the projection vectors one by one by means of sequential methods such as the deflation of the covariance matrix (Appendix A). We use the first way in the experiments that have been carried out in this Thesis.

PCA has multitude of applications such as a mere data exploration and visualization (Jolliffe, 2010). The main limitation of PCA apart of being a linear method, is that it does not consider the target variables $\mathbf{Y}$ for the input vectors but simply performs a coordinate rotation that aligns the transformed axes with the directions of maximum variance of the original data distribution. Thus, there is no guarantee that the directions of maximum variance will contain good features for discrimination or regression problems (see Figure 3.2).

3.4.2 Partial Least Squares (PLS)

The PLS algorithm, developed by Herman Wold (Wold, 1966), is probably one of the simplest methods for *supervised feature extraction*, since it considers the input data and the target data sets to define extracted features. The central idea of PLS is to find the projection vectors that maximize the cross-covariance between the projected input and output data, whose problem is expressed as:

$$\begin{aligned} \text{PLS:} \quad & \mathbf{U}, \mathbf{V} = \underset{\mathbf{U},\mathbf{V}}{\arg\max} \;\; \mathrm{Tr}\{\mathbf{U}^\top \mathbf{C}_{xy} \mathbf{V}\} \\ & \text{subject to: } \mathbf{U}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{V} = \mathbf{I}, \end{aligned} \tag{3.5}$$

where $\mathbf{C}_{xy} = \tilde{\mathbf{X}}^\top \tilde{\mathbf{Y}}$ is the covariance matrix of input and output centered data, and $\mathbf{V}$ is the projection matrix to be estimated for the output data set.

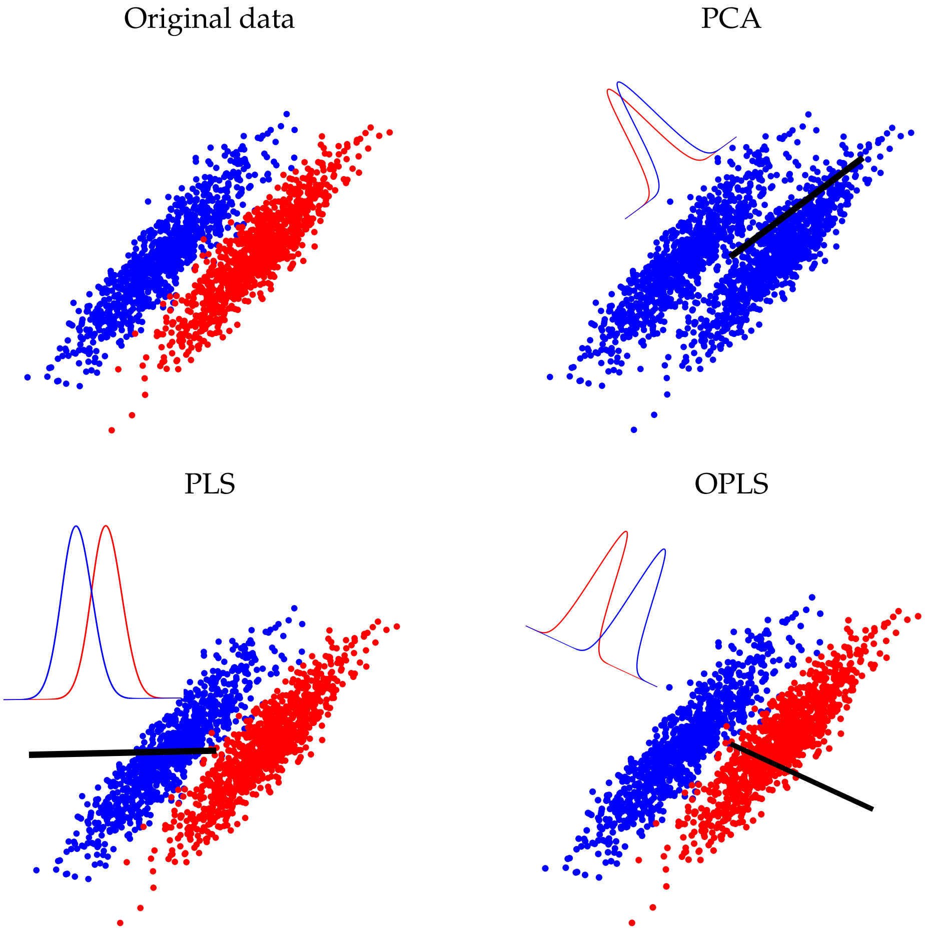


Figure 3.2: Toy example consisting of two overlapping Gaussian distributions, and the solution obtaining by PCA, PLS and OPLS first principal component and marginal pdfs of the two classes.

The equation (3.5) reduces to solve a Singular Value Decomposition (SVD) of the input-output covariance matrix (see more information in Appendix A):

$$\mathbf{C}_{xy}\mathbf{V} = \mathbf{\Lambda}\mathbf{U} \quad \text{or} \quad \mathbf{C}_{xy}\mathbf{U} = \boldsymbol{\beta}\mathbf{V}. \tag{3.6}$$

Some MVA methods consider also a feature extraction in the output space as is the PLS case, $\tilde{\mathbf{Y}}' = \tilde{\mathbf{Y}}_*\mathbf{V}$ where $\mathbf{V}$ is the projection matrix for the output data whose size is $n_c \times d_f$, where n_c is the number of data classes.

In this work, the SVD of $\mathbf{C}_{xy}$ has been used in order to solve the problem (Sampson et al., 1989) but it can be solved using different variants available in the literature. In our case, the maximum number of features d_f that PLS can extract is limited by the output dataset dimensionality (n_c) or the number of dataset samples (n), since d_f depends on $rank(\mathbf{C}_{xy})$.

Figure 3.2 shows the first component obtained by PLS in a 2D toy example. PLS method is capable to distinguish both classes while, PCA does not distinguish then because, being an unsupervised method, it does not take into account the labels.

3.4.3 Orthonormalized Partial Least Squares (OPLS)

In this subsection, we review a variation of the PLS method, known as Orthonormalized PLS (OPLS). The OPLS method consists of finding the projections vectors that maximize the cross-covariance between the input data with the output data (**Y**). Then, the two main differences between PLS and OPLS are: 1) In OPLS, the original data are *decorrelated*, i.e. their covariance is the identity matrix, and 2) output data are not projected, which means that OPLS only performs dimensionality reduction in the input space, not in output space. That, OPLS is defined by the following maximization problem:

$$\begin{aligned} \text{OPLS:} \quad & \mathbf{U} = \arg\max_{\mathbf{U}} \ \text{Tr}\{\mathbf{U}^\top \mathbf{C}_{xy}\mathbf{C}_{xy}^\top \mathbf{U}\} \\ & \text{subject to: } \mathbf{U}^\top \mathbf{C}_{xx}\mathbf{U} = \mathbf{I}, \end{aligned} \tag{3.7}$$

where $\mathbf{I}$ is the identity matrix of size d_f (number of extracted features), $\mathbf{C}_{xx} = \tilde{\mathbf{X}}^\top\tilde{\mathbf{X}}$ is the centered sample covariance matrix of input data, $\mathbf{C}_{xy} = \tilde{\mathbf{X}}^\top\tilde{\mathbf{Y}}$ is the centered sample cross-covariance matrix of input-output data and $\mathbf{U}$ is the projection matrix to be estimated. Figure 3.2 shows the first feature selected by OPLS method, which is alike that the PLS first feature, in the toy example.

Equation (3.7) reduces to solving a generalized eigenvalue problem:

$$\mathbf{C}_{xy}\mathbf{C}_{xy}^\top\mathbf{U} = \mathbf{\Lambda}\mathbf{C}_{xx}\mathbf{U}. \tag{3.8}$$

There are different possibilities in order to solve the OPLS generalized eigenvalue problem. One of them is to transform it into a standard eigenvalue problem, either premultiplying both sides of the equation by $\mathbf{C}_{xx}^{-1}$ or, more conveniently way, defining $\mathbf{W} = \mathbf{C}_{xx}^{-1/2}$ and after, premultiplying right side of the equation by $\mathbf{C}_{xx}^{-1/2}$. Then, we get a standard eigenvalue problem:

$$\mathbf{C}_{xx}^{-1/2}\mathbf{C}_{xy}\mathbf{C}_{xy}^\top\mathbf{C}_{xx}^{-1/2}\mathbf{W} = \mathbf{\Lambda}\mathbf{W}.$$

OPLS projections can be recovered from $\mathbf{W}$ as $\mathbf{U} = \mathbf{C}_{xx}^{-1/2}\mathbf{W}$. It is important to remark that the transform from the generalized eigenvalue problem to standard eigenvalue problem is possible only when $\mathbf{C}_{xx}$ is a full rank matrix. Even if the matrix $\mathbf{C}_{xx}$ is not a full rank matrix, it is still possible to solve the problem (Golub and Van Loan, 1996).

The maximum number of features obtained by OPLS method is limited by the rank of $\mathbf{C}_{xy}$. This means that the OPLS maximum number of features depends on the dimensions of the output data or the samples of input data since $d_f < min(n_c, n)$, as in the PLS method.

3.5 Kernel-based multivariate analysis

The previous subsection has reviewed some linear feature extraction methods. As we have seen, these methods are easily interpretable but the projections are limited to the assumption of linear feature relations. The extracted features have high quality when the data relations are linear but the performance may be seriously degraded otherwise. Linear feature extraction also suffers in case of strong collinearity of the inputs features, and also in the case of higher dimensionality than number of examples. Both situations are ubiquitous in many scientific problems, and are mainly related to poor estimation of covariance matrix in ill-posed situations. Furthermore, linear models can be wrong dealing with dataset that consists of more dimensions than samples.

Several authors have proposed nonlinear extensions of MVA methods in order to solve the problem of nonlinear data relations. The most widespread methods of transforming linear to nonlinear MVA methods is through the kernel framework. To obtain the kernel version, we must only replace the original centered data matrix $\tilde{\mathbf{X}}$ by the centered data in feature space $\tilde{\mathbf{\Phi}}$. Thus, the training dataset is $\{\tilde{\boldsymbol{\phi}}(\mathbf{x}_i), \mathbf{y}_i\}_{i=1}^{n}$, where $\tilde{\boldsymbol{\phi}}(\mathbf{x}_i) \in \mathcal{H}$ and $\mathbf{y}_i \in \mathbb{R}^{n_c}$. In the following subsections, we will show how through the Representer's Theorem (Section 2.2.3) the linear MVA algorithms are rewritten in terms of inner products reaching their nonlinear extensions.

3.5.1 Kernel PCA (KPCA)

The goal of KPCA is to find the projections that maximize the variance of the input data in the feature space. By simply replacing $\tilde{\mathbf{X}}$ by $\tilde{\mathbf{\Phi}}$ in (3.3), KPCA can be formulated in the following way:

$$\begin{aligned} \text{KPCA:} \quad & \mathbf{U} = \arg\max_{\mathbf{U}} \ \text{Tr}\{\mathbf{U}^\top \tilde{\mathbf{\Phi}}^\top \tilde{\mathbf{\Phi}} \mathbf{U}\} \\ & \text{subject to: } \mathbf{U}^\top \mathbf{U} = \mathbf{I}, \end{aligned} \tag{3.9}$$

where matrix $\tilde{\mathbf{\Phi}}$ contains the mapped data centered in the Hilbert space. Making use of the representer's theorem one can introduce $\mathbf{U} = \tilde{\mathbf{\Phi}}^\top \mathbf{A}$ into the previous formulation, and the maximization problem can be reformulated as follows:

$$\begin{aligned} \text{KPCA:} \quad & \mathbf{A} = \arg\max_{\mathbf{A}} \ \text{Tr}\{\mathbf{A}^\top \tilde{\mathbf{K}}_x \tilde{\mathbf{K}}_x \mathbf{A}\} \\ & \text{subject to: } \mathbf{A}^\top \tilde{\mathbf{K}}_x \mathbf{A} = \mathbf{I} \end{aligned} \tag{3.10}$$

The solution to the above problem can be obtained from the eigendecomposition of $\tilde{\mathbf{K}}_x \tilde{\mathbf{K}}_x$ represented by $\tilde{\mathbf{K}}_x \tilde{\mathbf{K}}_x \mathbf{A} = \mathbf{\Lambda} \tilde{\mathbf{K}}_x \mathbf{A}$, which has the same solution as

$$\tilde{\mathbf{K}}_x \mathbf{A} = \mathbf{\Lambda} \mathbf{A}. \tag{3.11}$$

The maximum number of features d_f is limited by the rank of the kernel matrix, thus $d_f \leq rank(\tilde{\mathbf{K}}_x)$.

Note that centering in feature space can be done implicitly via the simple kernel matrix operation $\mathbf{K} \leftarrow \mathbf{HKH}$, where $H_{ij} = \delta_{ij} - \frac{1}{n}$, δ represents the Kronecker delta $\delta_{i,j} = 1$ if $i = j$ and zero otherwise (Section 2.2.3). It is worth mentioning that throughout the present Thesis we considered to center data but recently, the centering operation has been questioned (Cadima and Jolliffe, 2009; Shawe-Taylor and Cristianini, 2004), and it is possible to find PCA and KPCA versions with uncentered data in original space (Jenssen, 2013b).

3.5.2 Kernel PLS (KPLS)

KPLS is the nonlinear kernel-based extension of PLS (Arenas-García and Camps-Valls, 2008). The main difference between KPCA and KPLS is that while KPCA finds the projections containing the maximum variance of the input data in the feature space, KPLS extracts projections that account for both the projected input and target data. It is based on maximizing the variance between the projected data into a proper Hilbert space $\mathcal{H}$ and the target data matrix $\tilde{\mathbf{Y}}$ (i.e. the labels):

$$\begin{aligned} \text{KPLS:} \quad & \mathbf{U}, \mathbf{V} = \underset{\mathbf{U},\mathbf{V}}{\arg\max} \;\; \text{Tr}\{(\tilde{\mathbf{\Phi}}\mathbf{U})^\top \tilde{\mathbf{Y}}\mathbf{V}\} \\ & \text{subject to: } \mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}, \end{aligned} \tag{3.12}$$

By using again the representer's theorem, the maximization problem becomes:

$$\begin{aligned} \text{KPLS:} \quad & \mathbf{A}, \mathbf{V} = \underset{\mathbf{A},\mathbf{V}}{\arg\max} \;\; \text{Tr}\{\mathbf{A}^\top \tilde{\mathbf{K}}_x \tilde{\mathbf{Y}}\mathbf{V}\} \\ & \text{subject to: } \mathbf{A}^\top\tilde{\mathbf{K}}_x\mathbf{A} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}, \end{aligned} \tag{3.13}$$

As in the previous methods that we have reviewed, there are many ways of solving the problem. This is done either by solving an eigenvalue problem or as an iterative procedure. We can obtain the solution to this problem from the SVD of $\tilde{\mathbf{K}}_x\tilde{\mathbf{Y}}$. Alternatively, the problem can be efficiently solved using a deflation process. The many available variants of KPLS are defined by different forms of deflation. Rosipal and Krämer (2006) present an overview of PLS methods which can be adapted to nonlinear algorithms. Among all KPLS methods, we would emphasize the *KPLS Mode A* and the *dualPLS* (Shawe-Taylor and Cristianini, 2004). The first one is a deflation scheme that consists of the following two-steps iterative procedure:

1. Find the largest singular value of $\tilde{\mathbf{K}}_x\tilde{\mathbf{Y}}$, and the associated vector directions: $\{\boldsymbol{\alpha}_i, \mathbf{v}_i\}$, where $\boldsymbol{\alpha}_i$ and $\mathbf{v}_i$ are columns vectors of the $\mathbf{A}$ and $\mathbf{V}$ matrices.

2. Deflate the kernel matrix and labeled vector using:

$$\tilde{\mathbf{K}}_x \leftarrow \left[\mathbf{I} - \frac{\tilde{\mathbf{K}}_x\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^\top\tilde{\mathbf{K}}_x}{\boldsymbol{\alpha}_i^\top\tilde{\mathbf{K}}_x\tilde{\mathbf{K}}_x\boldsymbol{\alpha}_i}\right] \tilde{\mathbf{K}}_x \left[\mathbf{I} - \frac{\tilde{\mathbf{K}}_x\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^\top\tilde{\mathbf{K}}_x}{\boldsymbol{\alpha}_i^\top\tilde{\mathbf{K}}_x\tilde{\mathbf{K}}_x\boldsymbol{\alpha}_i}\right] \tag{3.14}$$

$$\mathbf{Y} = \mathbf{Y} - \tilde{\mathbf{K}}_x\boldsymbol{\alpha}_i\mathbf{Y}\frac{\tilde{\mathbf{K}}_x\boldsymbol{\alpha}_i}{||\tilde{\mathbf{K}}_x\boldsymbol{\alpha}||_2^2} \tag{3.15}$$

The *dualPLS* (Shawe-Taylor and Cristianini, 2004) uses the same deflation method above inside an iterative method known as the iterative power method (see Appendix A). This deflation procedure allows us to extract more features than output dimensions. This deflation method has been selected in this Thesis. For a more detailed description, as well as implementation details, the reader is referred to (Shawe-Taylor and Cristianini, 2004).

3.5.3 Kernel Orthonormalized PLS (KOPLS)

Kernel OPLS (KOPLS) presents the advantage of extracting the directions in feature space $\mathcal{H}$ that minimize the residuals of a multiregression that approximates the label matrix, i.e. $\arg\min_{\mathbf{U}} \; \|\tilde{\mathbf{Y}} - \tilde{\mathbf{\Phi}}\mathbf{U}\mathbf{W}\|_F^2$. It can be shown that this problem is equivalent to (Roweis and Brody, 1999):

$$\begin{aligned} \text{KOPLS:} \quad & \mathbf{U} = \arg\max_{\mathbf{U}} \; \text{Tr}\{\mathbf{U}^\top \tilde{\mathbf{\Phi}}^\top \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\top \tilde{\mathbf{\Phi}}\mathbf{U}\} \\ & \text{subject to: } \mathbf{U}^\top \tilde{\mathbf{\Phi}}^\top \tilde{\mathbf{\Phi}}\mathbf{U} = \mathbf{I}, \end{aligned} \tag{3.16}$$

whose dual form becomes:

$$\begin{aligned} \text{KOPLS:} \quad & \mathbf{A} = \arg\max_{\mathbf{A}} \; \text{Tr}\{\mathbf{A}^\top \tilde{\mathbf{K}}_x \tilde{\mathbf{K}}_y \tilde{\mathbf{K}}_x \mathbf{A}\} \\ & \text{subject to: } \mathbf{A}^\top \tilde{\mathbf{K}}_x \tilde{\mathbf{K}}_x \mathbf{A} = \mathbf{I}, \end{aligned} \tag{3.17}$$

where $\tilde{\mathbf{K}}_y = \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\top$. It can be shown that the columns of $\mathbf{A}$ are given by the solutions to the following *generalized* eigenvalue problem:

$$\tilde{\mathbf{K}}_x \tilde{\mathbf{K}}_y \tilde{\mathbf{K}}_x \mathbf{A} = \mathbf{\Lambda} \tilde{\mathbf{K}}_x \tilde{\mathbf{K}}_x \mathbf{A}. \tag{3.18}$$

We solved this problem with an iterative procedure that first computes the leading pair eigenvalue and eigenvector $\lambda_i, \boldsymbol{\alpha}_i$ (column vectors of $\mathbf{\Lambda}$ and $\mathbf{A}$, respectively) and then deflates the matrices. KOPLS method can only extract a maximum number of features given by the rank of $\tilde{\mathbf{K}}_x\mathbf{Y}$, while as we have already seen the KPCA and KPLS are limited by the rank of $\tilde{\mathbf{K}}_x$.

3.5.4 Kernel Entropy Component Analysis (KECA)

KECA was recently proposed as a general method for feature extraction and dimensionality reduction in pattern analysis and machine learning, (Jenssen, 2010, 2013a). The KECA algorithm is different from, but still intimately related to, the successful kernel multivariate signal processing methods such as kernel principal components analysis (KPCA), kernel canonical correlation analysis (KCCA) and kernel partial least squares (KPLS), (Arenas-García et al., 2013). On the one hand, KECA maintains a probabilistic input space interpretation, seeks to capture the entropy of the data in a reduced number of components, and constitutes a convergence point between kernel methods and information theoretic learning (Jenssen, 2009). On the other hand, KPCA, KPLS and KOPLS are also based on *kernel feature space* (reproducing kernel Hilbert space - RKHS).

KECA relies on the eigendecomposition of the (uncentered) kernel matrix, and sorts the eigenvectors according to the so-called entropy values of the projections. This is tightly related to information-theoretic concepts and the field of density estimation. The entropy-relevant dimensionality reduction transforms the dataset in a way that reveals cluster structure and hence information about the underlying class or cluster structures in the data (Jenssen, 2009, 2013a).

To be more precise, the measure of information used in Jenssen (2010) is the Renyi's second order entropy, given by

$$H = -\log \int p^2(\mathbf{x}) d\mathbf{x}, \tag{3.19}$$

where $p(\mathbf{x})$ is the pdf generating the data. Given a dataset $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of dimensionality d, the entropy may be estimated through kernel density estimation, KDE (Silverman, 1986) (as we will see Sec.2.3.5) as $-\log v$, where v is the so-called *information potential* (Principe, 2010):

$$v = \frac{1}{n^2} \mathbf{1}_n^\top \mathbf{K} \mathbf{1}_n \tag{3.20}$$

where $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ is any valid KDE kernel comprising the $(n \times n)$ kernel matrix and $\mathbf{1}_n$ is a n-dimensional vector of ones. Using the kernel decomposition introduced in Jenssen (2010):

$$\mathbf{K} = \mathbf{B}\mathbf{B}^\top = (\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}})(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^\top), \tag{3.21}$$

we may write

$$v = \sum_{j=1}^{d_f} \left(\sum_{i=1}^{n} \mathbf{B}_{ij} \right)^2 = \sum_{j=1}^{d_f} \left(\lambda_j^{\frac{1}{2}} \mathbf{1}_n^\top \mathbf{u}_j \right)^2. \tag{3.22}$$

In this expression, $\mathbf{U}$ contains the eigenvectors in columns, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]$, and $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{K}$, i.e. $\mathbf{\Lambda}_{ii} = \lambda_i$, and $d_f \leq n$ is the number of retained components. The terms $(\lambda_j^{\frac{1}{2}} \mathbf{1}_n^\top \mathbf{u}_j)^2$ denote the entropy values. Note that KECA and KPCA are somewhat related since the eigenvectors are the same in both cases. The difference resides in the criterion for selecting the most relevant directions: retained variance in KPCA or entropy in KECA.

3.6 Summary

Generally speaking, MVA methods look for projections of the input data that are "maximally aligned" with the targets, and the different methods are characterized by the particular objectives they maximize. Table 3.2 compares some of the most important properties of the methods described in this chapter. An interesting property of linear methods is that they are based on first and second order moments, and that some of their solutions can be formulated in terms of (generalized) eigenvalue problems. Thus, standard linear algebra methods can be readily applied. This property is shared by kernel methods as well. Table 3.2 shows the problem to be solved, the constraints involved, and the maximum number of features that each method can extract.

Table 3.2: Summary of linear and kernel MVA methods. For each method it is stated the objective to maximize (1st row), constraints for the optimization (2nd row), and maximum number of features (last row).Vectors $\mathbf{u}$ and α are column vectors in matrices $\mathbf{U}$ and $\mathbf{A}$, respectively. $r(\cdot)$ denotes the rank of a matrix. Based on Arenas-García et al. (2013).

PCA	PLS	OPLS	KPCA	KPLS	KOPLS	KECA
$\mathbf{u}^\top\mathbf{C}_x\mathbf{u}$	$\mathbf{u}^\top\mathbf{C}_{xy}\mathbf{v}$	$\mathbf{u}^\top\mathbf{C}_{xy}\mathbf{C}_{xy}^\top\mathbf{u}$	$\boldsymbol{\alpha}^\top\tilde{\mathbf{K}}_x^2\boldsymbol{\alpha}$	$\boldsymbol{\alpha}^\top\tilde{\mathbf{K}}_x\tilde{\mathbf{Y}}\mathbf{v}$	$\boldsymbol{\alpha}^\top\tilde{\mathbf{K}}_x\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\top\tilde{\mathbf{K}}_x\boldsymbol{\alpha}$	$\boldsymbol{\alpha}^\top\tilde{\mathbf{K}}_x^2\boldsymbol{\alpha}$
$\mathbf{U}^\top\mathbf{U}=\mathbf{I}$	$\mathbf{U}^\top\mathbf{U}=\mathbf{I}$ $\mathbf{V}^\top\mathbf{V}=\mathbf{I}$	$\mathbf{U}^\top\mathbf{C}_x\mathbf{U}=\mathbf{I}$	$\mathbf{A}^\top\tilde{\mathbf{K}}_x\mathbf{A}=\mathbf{I}$	$\mathbf{A}^\top\tilde{\mathbf{K}}_x\mathbf{A}=\mathbf{I}$ $\mathbf{V}^\top\mathbf{V}=\mathbf{I}$	$\mathbf{A}^\top\tilde{\mathbf{K}}_x^2\mathbf{A}=\mathbf{I}$	$\mathbf{A}^\top\mathbf{K}_x\mathbf{A}=\mathbf{I}$ sorted by max.H
$r(\tilde{\mathbf{X}})$	$r(\mathbf{C}_{xy})$	$r(\mathbf{C}_{xy})$	$r(\tilde{\mathbf{K}}_x)$	$r(\tilde{\mathbf{K}}_x)$	$r(\tilde{\mathbf{K}}_x\tilde{\mathbf{Y}})$	$r(\mathbf{K}_x)$

Once the feature extraction with different methods is done, we project data onto the main directions. We project test data using equation (3.1) in original space and (3.2) in feature space. Figure 3.3 shows the obtained projections in a 2D example. The toy example is composed by three sinusoidal snippets; each of them belonging to a different class (different colors). Linear methods fail in finding good projections since they reduce to rotations and thus they cannot cope with the nonlinear nature of the data distribution. Kernel methods find nonlinear projections that separate the classes better. The solution of KPCA does not allow to linearly separate the data. This is due to the fact that it is very difficult to tune the kernel parameter without labeled data, as previously studied in (Braun et al., 2008).

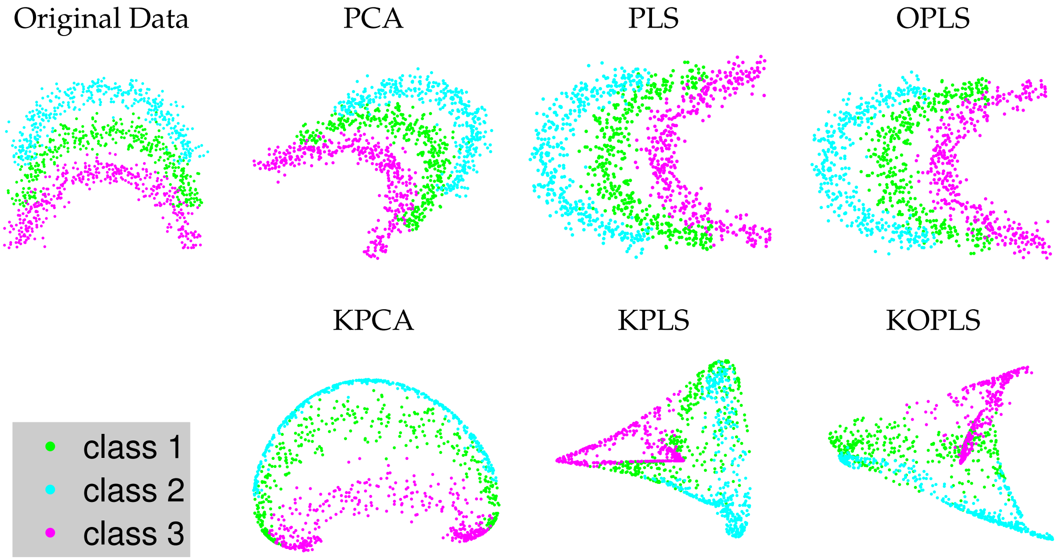


Figure 3.3: Projections extracted by different linear and nonlinear feature extraction methods in a 2D Toy example with three classes.

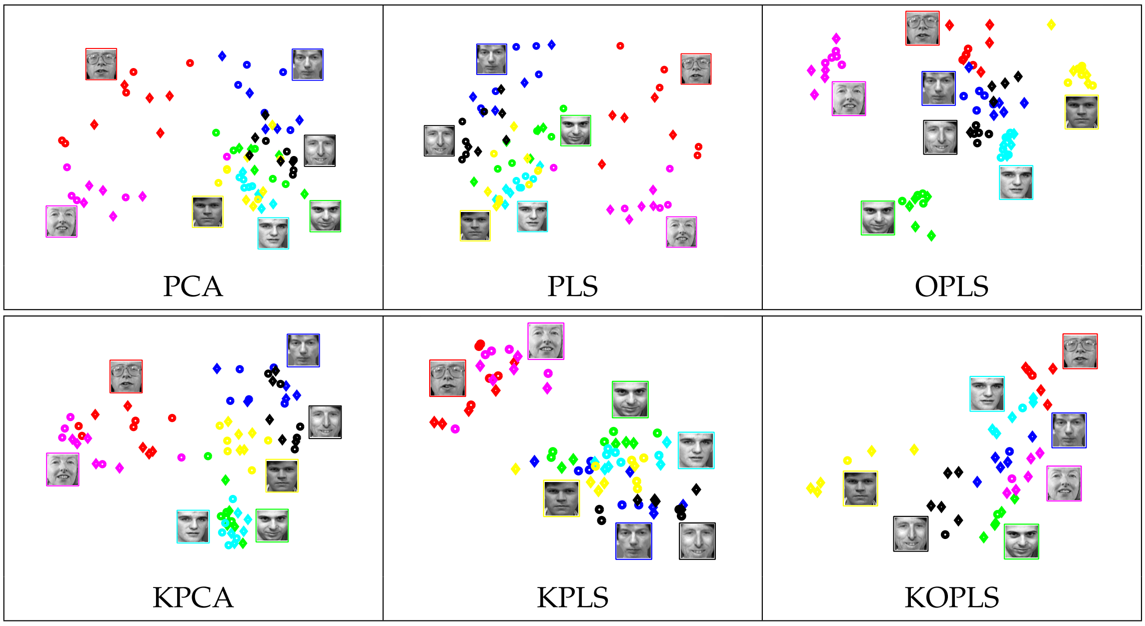


Figure 3.4: Faces projected onto the first two extracted features by different feature extraction methods.

Figure 3.4 shows another example to analyze the projections obtained by feature extraction methods. We used the Olivetti faces dataset [1]. The dataset is composed by 10 images of size 64 × 64 pixels *per person* taken with different poses. A total of 40 persons with different illumination conditions were acquired. We selected 7 different persons to evaluate the performance of the methods. Figure 3.4 illustrates the projected faces in a 2D space by the different feature extraction methods (each color represents a face). As in the previous example, linear methods have serious problems to group faces of the same class, except the red and pink classes. Also, the KPCA and KPLS methods have troubles to project data to discriminative representation, whereas the KOPLS method distinguishes the classes better that the rest of the methods, yielding clearer clusters with higher separability.

Several feature extraction methods have been presented in this chapter. As we have seen, kernel feature extraction methods help in the grouping of the data projection into the same class better than the linear methods. Nevertheless, this occurs in nonlinear data distributions. Therefore, before selecting a feature extraction method is recommendable to have prior knowledge of the original data. A Matlab implementation of the algorithms is available at `http://isp.uv.es/simfeat.html` for the interested reader.

[1] `http://www.cs.nyu.edu/~roweis/data.html`

Chapter 4

Supervised Kernel Feature Extraction including Invariances

Contents

This chapter introduces a simple method for including *invariances* in support vector machine (SVM) remote sensing image classification. We design explicit invariant SVMs to deal with the particular characteristics of remote sensing images. The problem of including data *invariances* can be viewed as a problem of encoding prior knowledge, which translates into incorporating informative support vectors that better describe the classification problem. The proposed method essentially generates new (synthetic) support vectors from the obtained by training a standard SVM with the available labeled samples. Then, original and transformed support vectors are used for training the Virtual SVM (VSVM). We first incorporate invariances to rotations and reflections of image patches for improving contextual classification. Then, we include invariance to object scale in patch-based classification. Finally, we focus on the challenging problem of including illumination invariances to deal with shadows in the images. Interestingly, the methodology can be applied to any kernel method, thus constituting a new research opportunity. Posteriorly, we will relate this new approach to the central core of this Thesis: feature extraction with kernel methods.

The chapter is partly based on the published papers:

- □ E. Izquierdo-Verdiguier, V. Laparra, L. Gómez-Chova, and G. Camps-Valls, *"Including invariances in SVM remote sensing image classification," 2012 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 7353–7356, 2012.
- □ E. Izquierdo-Verdiguier, V. Laparra, L. Gómez-Chova, and G. Camps-Valls, *"Encoding Invariances in Remote Sensing Image Classification With SVM," IEEE Geoscience and Remote Sensing Letters*, vol. 10 (5), pp. 981–985, 2013.

4.1 Introduction

As previously discussed, dimensionality of the remote sensing images is one of the main problems in the analysis of data (see Chapter 1). The large amount of acquired data in remote sensing favors the use of machine learning techniques. In Chapter 2 we presented different techniques to analyze and turn into information input data. To reduce the model complexity, we have applied a dimensionality reduction approach to the input data. There are plenty of features extraction methods, but MVA stands out (summarized in Chapter 3). Of all MVA methods previously explained, we will focus on supervised feature extraction methods in this chapter.

The introduction of supervised feature extraction methods in remote sensing has increased in the last decades. They are mostly used in classification or regression problems. Although Principal Component Analysis (PCA) has been a widely used method in remote sensing, there are other feature extraction methods that have been proposed to remote sensing problems such as partial least square (PLS) or Orthonormized PLS (OPLS) methods. PLS applied to regression is one of the most common techniques for spectral calibration and prediction to measure the reflectance of canopy biomass (Hansen and Schjoerring, 2003; Cho et al., 2007) or soil organic carbon (Cho et al., 2007), among others. Nevertheless, PLS and OPLS are possible to use them for image classification. In (Arenas-García et al., 2013), not only a comparison among different supervised feature extraction methods (linear and their kernel based methods) is presented, but they also are applied to pixel-based hyperspectral image classification and regression problems.

In the following sections, we will discuss three important issues about supervised methods. First, we will contrast supervised methods with the classical PCA and its nonlinear approach (KPCA). Second, we will discuss about the inclusion of *invariance* (Izquierdo-Verdiguier et al., 2012c, 2013b) not only into the classifiers, but also in feature extraction approaches. Finally, we will discuss the main conclusions about the studied supervised feature extraction methods with and without invariance encoding.

Table 4.1: UCI database description (n: number of samples, d: number of dimensions, n_c: number of classes, n_{train}: number of training samples, and n_{test}: number of test samples).

Database	n	d	n_c	n_{train}	n_{test}
Ionosphere	351	33	2	80	172
Letter	20000	16	26	260	1040
Pendigits	10992	16	9	180	900
Pima-Indians	768	8	2	200	330
Vowel	990	12	10	150	330
wdbc	569	30	2	80	344

4.2 Comparison of supervised kernel feature extraction methods

Within the MVA family, there are several kinds of supervised feature extraction methods. We will focus now on PLS and OPLS methods and their nonlinear kernel counterparts. We will evaluate the performance of different supervised linear and nonlinear feature extraction methods and will compare its results to classical unsupervised methods, namely PCA and KPCA. We will illustrate the results obtained in two cases evaluating the accuracy of methods for classification and for regression experiments in different databases and multispectral images.

4.2.1 Feature extraction for classification

In this section, we present the results obtained by applying different feature extraction methods to six real datasets from the University of California Irvine (UCI) Machine Learning Repository[1] and the Pavia remote sensing multispectral image classification problem. First, we will start explaining the data used in the experiments. And then, we will focus on the analysis and comparison of the accuracy and the robustness between the supervised feature extraction methods used prior the linear classifier.

Data collection

We selected some databases of the UCI repository: The Ionosphere dataset is a binary classification problem about the radar signal quality returned from the ionosphere; the goal for the Letter dataset is to detect each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet; the Pendigits problem deals with the pen-based recognition of handwritten digits; the Pima-Indians dataset constitutes a classical problem of diabetes diagnosis in patients from clinical variables; the Vowel dataset deals with the vowels detection problem in Japanese and contains data from a large number of time series of cepstrum coefficients taken from speakers; and finally wdbc is another clinical problem

[1] `http://archive.ics.uci.edu/ml/datasets.html`

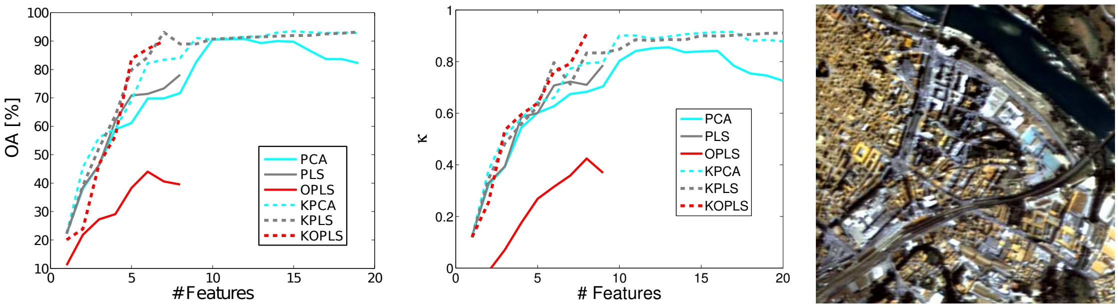


Figure 4.1: Overall accuracy (Left) and κ (Center) in an independent test set as function of different numbers of extracted features. Right: RGB Pavia image.

for diagnosis of breast cancer in malignant/benign classes. The datasets were intentionally selected either because of the observed high collinearity between input features or the diversity in number of classes. Table 4.1 gives details on the dimensionality, number of classes and training and test stes used in the experiments which are explained below.

The last experiment deals with the problem of pixel classification of a hyperspectral image. We applied different feature extraction methods plus a linear classifier to an hyperspectral remote sensing image in order to analyze the methods in a real image (see the RGB image in Fig. 4.1[right]). We used in this case an image acquired by the DAIS7915 sensor over the city of Pavia (Italy)[2], as it constitutes a challenging 9-class urban classification problem dominated by structural features and relatively high spatial resolution (5-meter pixels). Following previous works on the classification of this image, we took into account only 40 spectral bands in the range [0.5, 1.76] μm, and thus skipped thermal and middle infrared bands above 1958 nm. Training and test sets using this image are 36 and 1710 samples, respectively.
We considered the Gaussian RBF kernel since it is the most common in kernel methods. This kernel only introduces a scalar free parameter, σ. We obtained the RBF kernel by fixing the width parameter (σ) to the average Euclidean distance between all samples.

Experimental results

We extracted feature projections and projected train and test data using Eqs. (3.1) and (3.2), respectively. Then, a Least Squares (LS) linear regressor has been used because it is a simple and fast model. The basic idea is to find the best predictions that minimize the prediction error. In matrix notation, let $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{Y} \in \mathbb{R}^{n\times n_c}$ be a set of observations:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|^2,$$

whose solution reduces to the normal equations $\mathbf{W}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} = \mathbf{X}^\dagger\mathbf{Y}$, where † is the Moore-Penrose pseudoinverse, and predictions for test data $\mathbf{X}_*$ are obtained by $\hat{\mathbf{Y}}_* = \mathbf{X}_*\mathbf{W} = \mathbf{X}_*\mathbf{X}^\dagger\mathbf{Y}$.

[2] I would like to acknowledge Prof. Paolo Gamba for kindly providing the image.

Table 4.2: Kappa statistic for different UCI databases.

Database	d_f	PCA	PLS	OPLS	KPCA	KPLS	KOPLS
Ionosphere	1	0.51	0.38	-0.27	0.54	0.51	**0.55**
	2	0.11	0.37	0.52	0.39	**0.55**	–
Letter	1	0.005	0.04	0.04	-0.001	0.03	0.02
	10	0.34	0.33	0.35	0.30	0.36	0.32
	20	–	–	–	0.35	0.44	**0.47**
Pendigits	1	0.07	0.11	0.12	0.12	0.11	0.12
	8	0.70	0.73	0.72	0.72	0.78	**0.87**
	20	–	–	–	0.85	0.85	–
Pima-Indians	1	0.37	0.39	0.16	**0.43**	-0.09	0.24
	2	0.32	0.37	0.16	**0.43**	0.10	–
	3	0.14	–	–	0.13	0.16	–
Vowel	1	-0.002	0.08	0.10	-0.005	0.10	0.11
	9	0.37	0.35	0.38	0.39	0.51	**0.58**
	20	–	–	–	0.45	0.58	–
wdbc	1	0.80	0.77	0.53	0.81	**0.84**	0.75
	2	0.83	0.84	0.60	0.81	0.75	–
	3	0.83	–	–	0.81	0.75	–

For the particular case of classification, the linear model is followed by a "winner-takes-all" activation function. We used the overall accuracy OA[%] and the estimated Cohen's kappa statistic κ. The former is the mean value of the correct prediction obtained by the classifier, and the latter score measures the statistical agreement between observers. Both measures are obtained from the confusion matrix, see (Congalton and Green, 1999).

Results for the UCI databases are shown in Table 4.2. The number of d_f is variable according to the number of classes in the database. It is also variable according to the rank of the matrix applied to obtain the eigendecomposition which depends on the feature extraction method. This happens with all methods but the KPLS, which depends on the decomposition algorithm used (see Sec. 3.5.2). For this reason, Table 4.2 presents different values of d_f depending on the databases. The results were obtained for a different number of training and testing samples for each database (see Table 4.1). In general, nonlinear methods (KPCA, KPLS and KOPLS) outperform the linear approaches (PCA, PLS, OPLS). The supervised methods provide the best results in all databases except for Pima-Indians. Remarkably, the KOPLS outperforms KPLS in four out of five databases with less number of extracted features. Excluding Pendigits and Letter databases which all feature extraction methods show low κ results, note that, the lower κ values have obtained by OPLS method. This may suggest an OPLS overfitting problem.

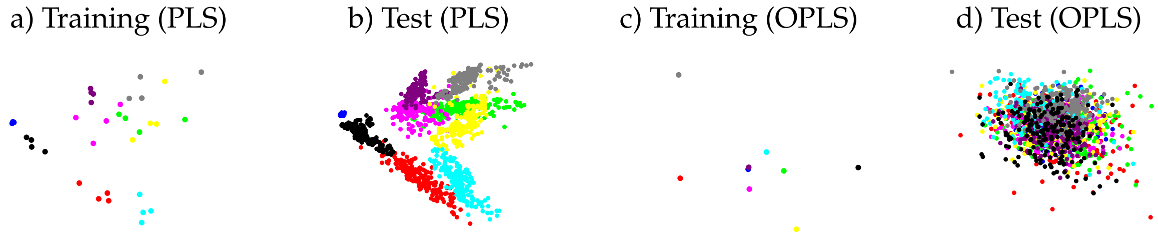


Figure 4.2: Training and test data projections on two dimensions obtained by PLS and OPLS.

Figure 4.1 shows the results for the hyperspectral Pavia image for a different number of extracted dimensions d_f. Nonlinear methods (dotted lines) generally lead to higher accuracy than the linear methods (solid line). The KOPLS slightly outperforms PLS with lower number of features. It is worth noting that the test accuracy obtained with the OPLS method is lower than 50%, even when all possible extracted features are used. This is because the OPLS method overfits the training data when the ratio n_{train}/d is very low since very low variance directions of the input space are used (Arenas-García et al., 2013). This issue of potential overfitting is further analyzed in Fig. 4.2, which shows the data projections obtained by PLS and OPLS in training and test datasets. The test data projected by PLS (Fig. 4.2(b)) allows to separate the different classes, while the OPLS (Fig. 4.2(d)) is not able to do it. Figure 4.2(c) shows the overfitting for the projections of the data using OPLS while this does not happen with PLS. We will come back to this issue in the next section, and will propose the inclusion of virtual examples to remedy the problem.

4.2.2 Feature extraction for biophysical parameter estimation

In this section, we tackle the estimation of oceanic chlorophyll concentration from multispectral MERIS measurements. The dataset simulates data acquired by the Medium Resolution Imaging Spectrometer (MERIS) (Bezy et al., 1997; Rast and Agency, 1999) onboard the Envisat satellite (MERIS dataset) and, in particular, the spectral behaviour of chlorophyll concentration in the subsurface waters. We selected the eight channels in the visible range (412-681 nm) to be used for retrieval. The range of variation of chlorophyll concentration in this dataset is $0.02 - 25\ mg/m^3$ (Camps-Valls et al., 2006a).

In this experiment, the predictions are obtained using a linear regression model on top of the projected data. We evaluate different quantitative measures of accuracy (with RMSE and MAE), bias (with ME) and goodness-of-fit (with the Pearson's correlation coefficient) for a varying number of extracted features. We compare the results obtained by 1) unsupervised linear PCA and its nonlinear kernel version, (KPCA), and 2) supervised feature extraction algorithms (PLS and OPLS their nonlinear version KPLS and KOPLS). Table 4.3 shows the obtained results with $n_{train} = 135$ and $n_{test} = 865$ samples to obtain the measures of accuracy in the test dataset. As in the classification section, the Gaussian RBF kernels have been obtained fixing the kernel

Table 4.3: Estimated results for the oceanic chlorophyll concentration retrieval problem versus the number of extracted features.

Model	**RMSE**	**MAE**	**|ME|**	**R**
PCA ($d_f = 1$)	0.484	0.385	**0.005**	0.221
PCA ($d_f = 2$)	0.352	0.279	**0.007**	0.704
PCA ($d_f = 3$)	**0.294**	**0.228**	**0.006**	**0.809**
PCA ($d_f = 4$)	**0.235**	**0.170**	0.006	**0.882**
PLS ($d_f = 1$)	0.429	0.339	0.008	0.502
OPLS ($d_f = 1$)	0.153	0.109	**0.005**	0.951
KPCA ($d_f = 1$)	0.486	0.390	0.008	0.194
KPCA ($d_f = 2$)	0.480	0.383	0.015	0.250
KPCA ($d_f = 3$)	0.368	0.292	0.014	0.673
KPCA ($d_f = 4$)	0.363	0.280	0.015	0.682
KPLS ($d_f = 1$)	0.401	0.317	0.022	0.589
KPLS ($d_f = 2$)	**0.350**	**0.278**	0.022	**0.709**
KPLS ($d_f = 3$)	0.339	0.269	0.008	0.730
KPLS ($d_f = 4$)	0.312	0.238	**0.005**	0.785
KOPLS ($d_f = 1$)	**0.143**	**0.066**	0.037	**0.961**

width parameter to the average Euclidean distance between all samples.

On average, nonlinear methods obtained better results than the linear approaches with few features (e.g. $d_f = 1$). Specifically, KOPLS method obtained the best results reducing the prediction error around 25% with respect to the KPCA and the linear PCA and PLS methods. Increasing d_f, the best predictions are obtained by PCA but in order to improve a result such as the KOPLS with $d_f = 1$, PCA requires at least four extracted features.

4.3 Invariant kernel feature extraction

Up to now, we have checked the robustness of the different methods using measures of classification (OA, κ) or regression (RMSE) accuracy in several toy examples and remote sensing data processing problems. All the previous feature extractors used the data directly, and did not consider the inclusion of prior knowledge about the problem. This limitation has actually led to eventual problems of overfitting and lack of expressive power. To circumvent these problems many forms of prior knowledge have been considered in remote sensing data processing. For example, in active learning, interaction with a user is the most naive form of incorporating knowledge as manual labeling corrects the posterior probability provided by a limited classifier (Tuia et al., 2011b). In the case of relying on the classical smoothness assumption of natural images, it is reasonable to regularize the solution by including contextual, multisource, multiangular or multitemporal information. Another example is to develop statistical models that invert radiative transfer models that encode plausible physical relations between features and biophysical parameters (Darvishzadeh et al., 2011).

Mathematical models, such as classifiers or regressors, should be robust to uninformative changes in the data representation. The property of such mathematical functions is called 'invariance', and the algorithm is referred to as being 'invariant', i.e. its decision function should be unaltered under transformations of data objects. The problem of encoding invariances in remote sensing image processing applications is ubiquitous. An algorithm for biophysical retrieval estimation should be resistant (invariant) to illumination changes and to canopy spectral invariants. Similarly, a classifier should be invariant to rotations of patches, to changes in illumination and shadows, or to the spatial scale of the objects to be detected. The question raised here is how to include *any* kind of prior invariant behavior into a large margin classifier.

One way to perform supervised classification of multispectral and hyperspectral remote sensing data is to use the Support Vector Machine (SVM, see Section 2.3.1) kernel algorithm which provides robustness and accuracy to the classification. Different ways of incorporating invariances in SVM were originally presented in (Schölkopf et al., 1996; DeCoste and Schölkopf, 2002; Chapelle and Schölkopf, 2002). Recently, other methods have been presented: Walder and Lovell (2005) proposed a penalization of the variance of the decision function across similar class memberships; while in Shivaswamy and Jebara (2006) the classifier is forced to be invariant to permutations of sub-elements within each sample. The work of DeCoste and Schölkopf (2002) considers two main solutions to the invariance problem: designing particular kernel functions that encode local invariance under transformations, or to generate artificial examples from the selected support vectors and train a SVM with them. The latter method is informally named Virtual SVM (VSVM) and, because of its simplicity and effectiveness, it is the one studied in this Thesis in the context of remote sensing image classification.

We will explain the proposed Virtual Support Vector Machine (VSVM) as a way to deal with the invariance problem. Subsequently, we show experimental results on three problems: encoding invariances to rotations and reflections of image patches for contextual classification; encoding invariances to the different scales of objects in the land cover classification, and encoding invariances to illumination changes to deal with shadows in the images.

4.3.1 Virtual SVM

The Virtual SVM (VSVM) implements invariances in a very simple and intuitive way. The method consists of three steps:

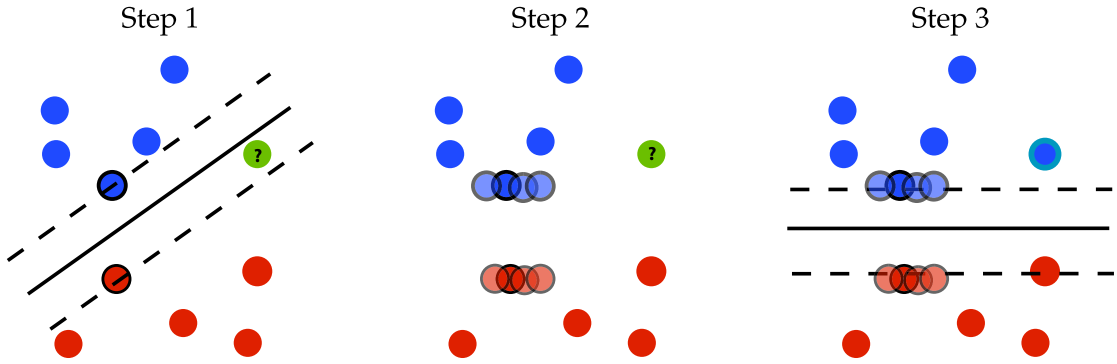


Figure 4.3: Illustration of the Virtual SVM in a binary toy example. The prior knowledge that we want to encode here is that the classification function should be invariant to transformations of the horizontal feature. The first step of the algorithm gives a wrong prediction to the green test sample because it does not fulfill the invariance assumption since the sample should belong to the blue class. The second step generates a set of *virtual* support vectors from the two found before: intuitively, the idea is to construct a more meaningful hyperplane by forcing the presence of SVs in those regions to which the classifier should be invariant. By training a SVM again with both SVs and VSVs, a correct hyperplane is obtained in the third step.

1. Train the standard SVM (see Section 2.3.1) with the available training data, and find the corresponding support vectors (SVs).
2. Perturb the features of the found SVs to which the solution should be invariant. This procedure generates a set of virtual SVs (VSVs).
3. Train a new SVM with both SVs and VSVs.

The method is intuitively illustrated in Fig. 4.3. The VSVM is general enough to encode any prior knowledge about the invariance of the classifier to specific features. The method was originally applied for handwritten digit recognition applications, in which the classifier should be invariant to rotation of the digits (Schölkopf et al., 1996; DeCoste and Schölkopf, 2002). Nevertheless, as we will see in the next section, encoding more challenging invariances may be harder in remote sensing image analysis.

4.3.2 Experimental results

In the experiments, we compare the standard SVM and the VSVM, both using the standard radial basis function (RBF) kernel with length-scale σ. A 10-fold cross-validation procedure was used to find the optimal SVM parameters, $\sigma \in [10^{-2}, ..., 10^{2}]$, $C \in [1, ..., 10^{3}]$. In all cases, the one-versus-one multiclass scheme implemented in LibSVM (Chang and Lin, 2011) was used. We report different figures of merit: overall accuracy (OA[%]), the estimated Cohen's kappa statistic (κ), and the rate of support vectors used after training both SVM and VSVM. Note that comparing the total number of SVs would constitute an unfair measure because the VSVM will always use by definition more training samples: SVs plus VSVs. We train the models with

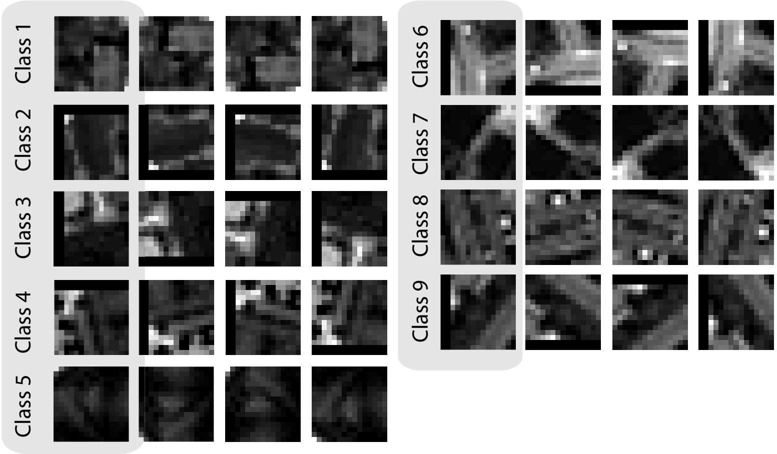


Figure 4.4: Example for encoding invariance to rotation. The leftmost patches (shaded) are illustrative SVs of the 9 different classes obtained in the first classification round by SVM in the QuickBird image. From these SVs, we generate the three rightmost patches (called virtual support vectors) by rotations and reflections.

different number of labeled examples *per* class, and show results in an independent test set. In particular, we compare the mean and standard deviation of random training sample selections, and test for statistical significance of the differences between models. Depending on the invariance to be encoded, specific parameters are varied, as discussed in the following subsections. Also note that in multiclass scenarios, the virtual support vectors should be generated for the class that encodes the invariance only.

Invariance to rotations

In the first experiment, we used a QuickBird image of a residential neighborhood of Zürich, Switzerland [3]. The image was acquired in August 2002, its size is 329 × 347 pixels, and has a spatial resolution of 2.4 m. A total of 40,762 pixels were labeled by photointerpretation, and assigned to 9 different land cover classes, such as soil, buildings, parkings, meadows, vegetation, roads, etc. Figure 4.6 shows an RGB composite and the ground truth available.

To enhance the performance of classifiers in some particular classes, morphological top-hat features were computed for the four bands and stacked to the multispectral bands before training the models. We used as structural elements squares and disks of sizes 5, 7 and 9, thus yielding a total of 22 spatial features. We performed patch-based classification: the image is divided into disjoint squared windows (patches) of size w, and each block is converted into a vector containing as many features as pixels are in the window. These vectors are used for classification, and its corresponding label is that of the center pixel in the patch. This is a very effective method to impose spatial smoothness in the classifier, and it is a procedure widely used in computer vision applications. Different patch sizes were considered, $w \in \{3, 5, 7, 9\}$. In this setting, the

[3] We would like to acknowledge Dr. Tuia at the EPFL (Switzerland) for kindly providing the QuickBird image.

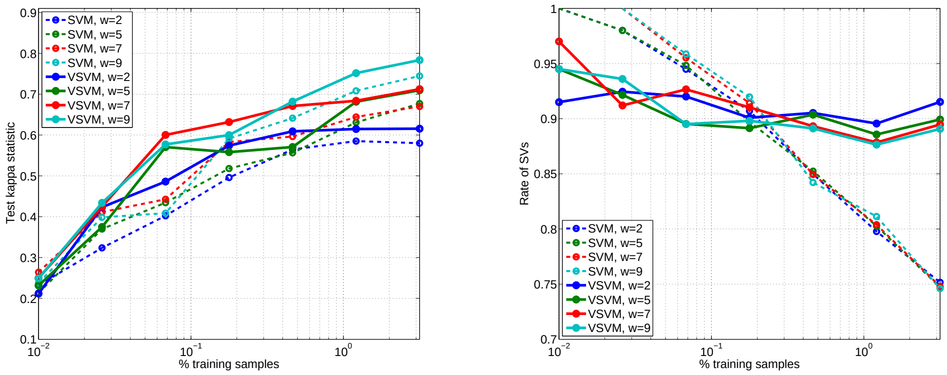


Figure 4.5: Kappa statistic κ (left) and SVs rate [%] (right) as a function of the number of training samples and window size w.

VSVs were generated by essentially rotating and reflecting the patches corresponding to the SVs, as illustrated in Fig. 4.4. We assume that the classifier should be invariant to the rotation or reflection of a patch, provided that the patch size contains enough information about the class characteristics.

Figure 4.5 shows the performance of the methods as a function of the used number of training samples and window size. Accuracy results show that in general VSVM performs better than SVM for all window sizes, but the gain is slightly higher with larger window sizes. As window size increases pixels from different classes are included as features for the classifiers (and also used for generating VSVs). This can eventually lead to decreased performance. Similar trends for different window sizes are observed for the standard SVM, but the curves of the proposed VSVM cross each other as more samples are included. Thus suggesting an optimal window size for encoding this type of invariance in this image. Another interesting observation is that the rate of SVs obtained with the VSVM is roughly constant for all training dataset sizes, suggesting that the introduced virtual vectors are rich. However, the standard SVM leads in general to

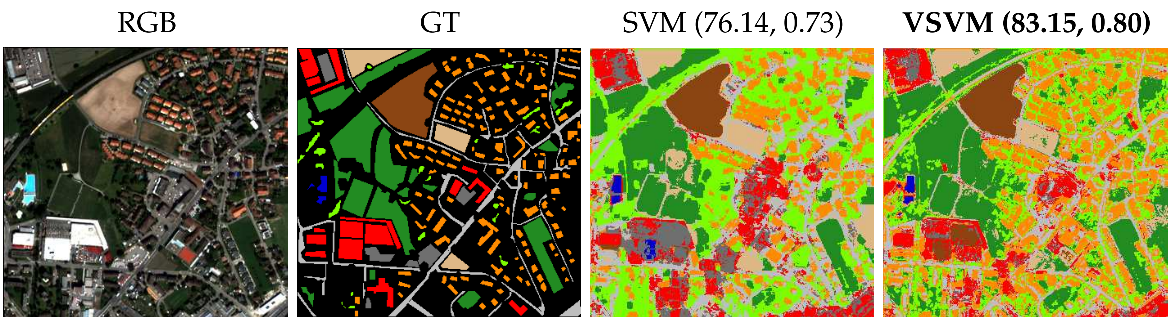

RGB GT SVM (76.14, 0.73) **VSVM (83.15, 0.80)**

Figure 4.6: True-color composite (RGB) and ground truth (GT) used in experiment 1 of patch-based classification. Classification maps of the experiments using standard SVM and VSVM with 500 training samples selected spatially disjoint. Best results are shown in parentheses in the form of (OA[%],κ).

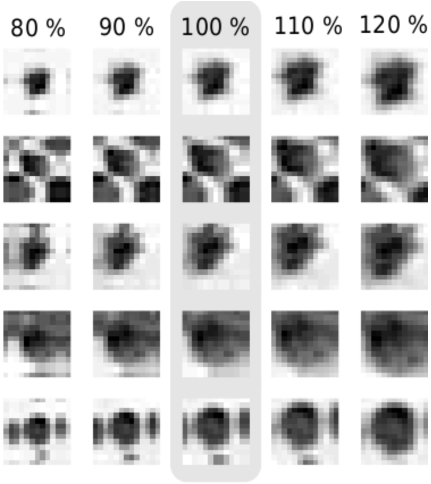


Figure 4.7: Generation of virtual support vectors by up-scaling and down-scaling 5 regular support vectors (each one in the central shaded column).

sparser models (remember that SVM uses a lower overall number of training vectors by definition). Actually, this turns to be even more noticeable with an increasing number of training samples.

Figure 4.6 shows the accuracy results and classification maps obtained with SVM and VSVM for the specific case of using a total of 500 training patches with $w = 5$. Both classifiers show high classification scores and the maps, generally, detect all major structures of the image. An improved numerical performance is obtained with the proposed VSVM (about +7% both in OA and κ). These results demonstrate the capabilities of the method to include invariances in the classifier, but also show that properly encoding the invariance is of paramount importance.

Invariance to objects scales

In this experiment, we introduce invariance to object scale in SVM: this means that the same object with different sizes should be univocally classified. This illustrative example simply focuses on the binary problem of classifying image patches as 'tree' or 'bare soil'. We used orthoimages of the Comunitat Valenciana autonomous region (Spain) provided by the Instituto Cartográfico Valenciano (ICV)[4]. The images were acquired in 2007 using an airborne Vexcel UltraCam camera (Leberl et al., 2003). The images have 0.5 m spatial resolution and 4 spectral bands (RGB and NIR). We generated a tree database with different classes (oranges, almond trees, olive trees, etc.) as well as uncultivated (bare soil) areas. Image patches of 13×13 pixels were used for classification.

[4] http://www.icv.gva.es/

Table 4.4: Number of SVs, VSVs generated and VSV finally used in the model as a function of the number of training samples.

Training samples	50	100	200	500	1000
SVs	26	36	59	112	188
Generated VSVs	81	149	268	585	759
Selected VSVs	50	87	152	341	491

The resizing operation was done by bicubic interpolation. Note that the VSVs must have same dimensions as the original samples. In order to generate VSVs, we resized a SV (image patch) and then selected the central 13 × 13 pixels. When the output size is smaller than the original one (decreasing the object size), we increase first the size of the original image by using a mirror padding. A careful padding is needed to create realistic virtual support vectors. It is also important to note that encoding the invariance can be done in this case for a particular feature (e.g. the image intensity) which alleviates the computational cost involved in the process. Figure 4.7 shows examples of different SVs (central column) that give rise to generated VSVs.

Figure 4.8 shows the results for a different number of training samples for a fixed test set of 1000 different samples. We performed 50 realizations for each number of training samples. The scale limits was set to 50% and 120%, which are realistic values for trees. The amount of VSVs generated for each SV is different in each realization, and have been tuned by cross-validation inside the training set. Results show that using the VSVs clearly improves the performance of SVM. The average improvement with VSVMs is more noticeable with a low number of training samples, which suggests that the procedure helps in describing the class distributions properly. The average gain achieved is around +5% for all situations, and statistical significant differences between methods are observed (note that error bars showing confidence intervals at 95%

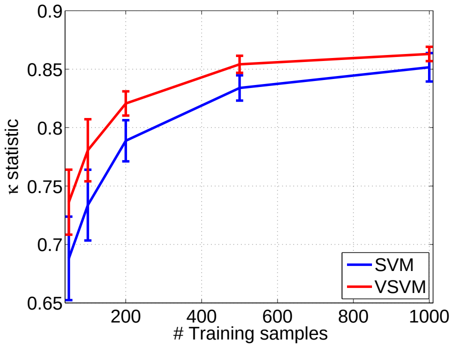
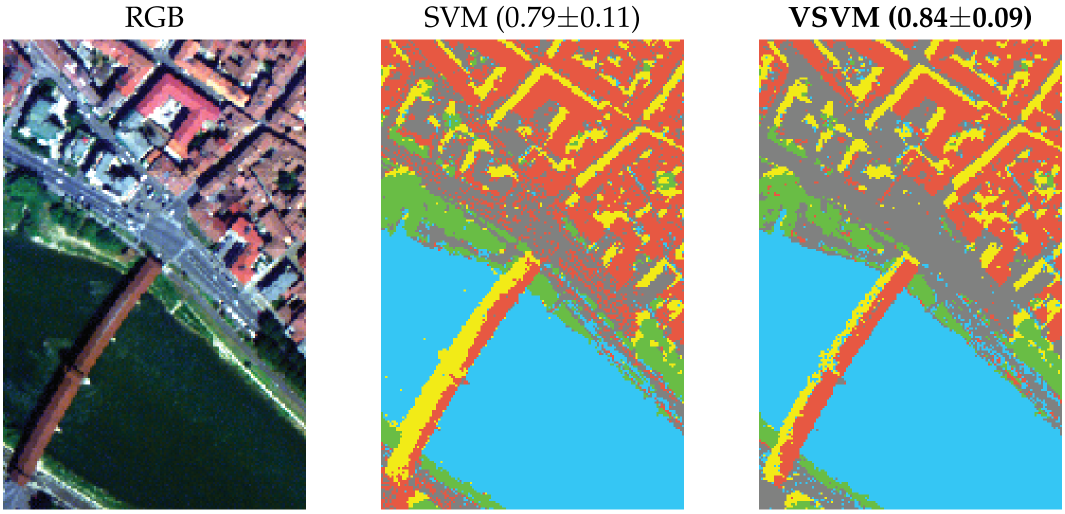


Figure 4.8: Classification results for invariance to object scale for the standard SVM (blue) and the VSVM (red) as a function of the number of training samples. Error bars indicate confidence intervals at 95% over the average accuracy computed for 50% realizations.



RGB SVM (0.79±0.11) **VSVM (0.84±0.09)**

Figure 4.9: True-color composite (RGB) used in the third experiment of patch-based classification and shadow invariance. Classification maps of the experiments using standard SVM and VSVM with 50 training pixels. Results are shown in parentheses in the form of (mean± standard deviation of κ in 20 realizations).

do not generally overlap).

Table 4.4 shows the number of SVs and VSVs for each number of training samples. Reported results are the mean of the 50 realizations. Note how, although VSVM obtains better classification performance, the solution is less sparse.

Invariance to shadows

The third experiment deals with the segmentation of hyperspectral images. We used data acquired by an airborne ROSIS-03 optical sensor of the city of Pavia (Italy) (Camps-Valls et al., 2014). The image consists of 102 spectral bands of size 1400×512 pixels with a spectral coverage ranging from 0.43 to 0.86μm. Spatial resolution of the scene is 1.3 m. 5 classes of interest (buildings, roads, water, vegetation and shadows) have been considered, and a labeled dataset of $206,009$ pixels has been extracted by visual inspection. As in the previous examples, we performed patch-based classification. For this purpose, we only used 50 training patches of size $w = 5$. This example deals with a different, but rather common, problem in remote sensing images: the presence of shadows.

The study of the presence of shadows and how to remove them before image processing (e.g. biophysical parameter estimation or classification) has been long studied (Finlayson et al., 2006). It is well-known that the radiance ratio shadow/sunlit increases as the sunlight gets weaker, thus depending on the hour of the day; and the ratio is dependent on the wavelength, due to the direct and diffuse light proportions. The intensity of the shadows is also influenced by the spatial neighborhood. In Yamazaki et al. (2009), an exponential behavior of the ratio shadow/sunlit as a function of the wavelength was observed in the visible range of Quickbird. With these observations in mind, we encoded invariance to shadows by generating VSVs from exponentially-modulated versions of the SVs, $\mathbf{x}_{vsv}(\lambda) = \mathbf{x}_{sv}(\lambda)\exp(-\gamma\lambda)$, where γ is a parameter that controls the impact of the spectral decay of the shadow/sunlit ratio as a function of the wavelength λ. We should note that only those SVs belonging to the class 'shadow' were used to generate VSVs, resembling the invariant SVM in Shivaswamy and Jebara (2006).

Numerical results and classification maps are reported in Fig. 4.9. Again, VSVM leads to more accurate results than the standard SVM in this experiment. Essentially, we observe about +5% gain in κ and overall accuracy (not shown), and slightly more stable results for different realizations, even with a reduced number of examples. Looking at the classification maps of Fig. 4.9, it is however observed that encoding shadow invariance reports some improvements, especially noticeable on the bridge and a more homogeneous classification on flat areas (see crossroads in the center of the image). The obtained numerical and visual results confirm the benefits of the invariance encoding in general. Nevertheless, we should note here that a statistical comparison between the solutions with a McNemar's test McNemar (1947) did not show significant differences ($|z| < 1.96$). The marginal homogeneity assessed by McNemar's test (and many other statistical tests) assume independence between the pairs, which might not necessarily hold in this particular case: the SVs used in SVM are also included in the VSVM. We also used Wilcoxon's rank sum tests to assessed statistical differences and results were similar to those obtained with the McNemar's test. Moreover, encoding shadow invariance in such a simple way may report some undesired effects in other classes, especially on the class 'vegetation' in this case.

4.4 Extracted features from virtual samples

In this section, we will compare the kernels and the extracted features using the original samples and also using the combination of original and virtual samples. As previously described, the Virtual SVM is a better approach than the standard SVM when the prior knowledge is properly included. Now, we will relate this new approach to the central core of this Thesis: feature extraction with kernel methods. For this, we focus on assessing two main points: 1) how the kernel (and its information content) changes with the inclusion of the virtual samples, and 2) how the extracted features are modified after adding the prior knowledge.

In order to show the variations in the kernel and the extracted features when virtual samples are added, we first use a bidimensional synthetic dataset composed of 600 samples divided in two classes. The example is generated by a Gaussian distribution (class 1) surrounded by a ring-shaped distribution (class 2). In Figure 4.10, we study a toy example distribution and the kernels generated by the training samples and the virtual samples generated by invariance to rotations, as well as the decision boundary obtained by both classifiers. We have used 30

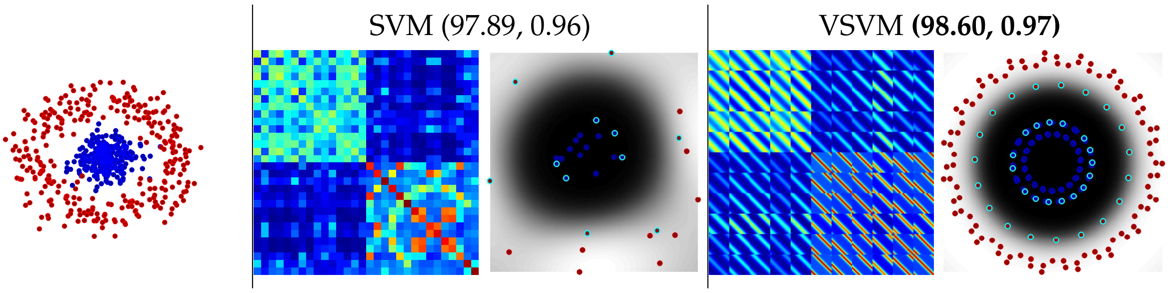
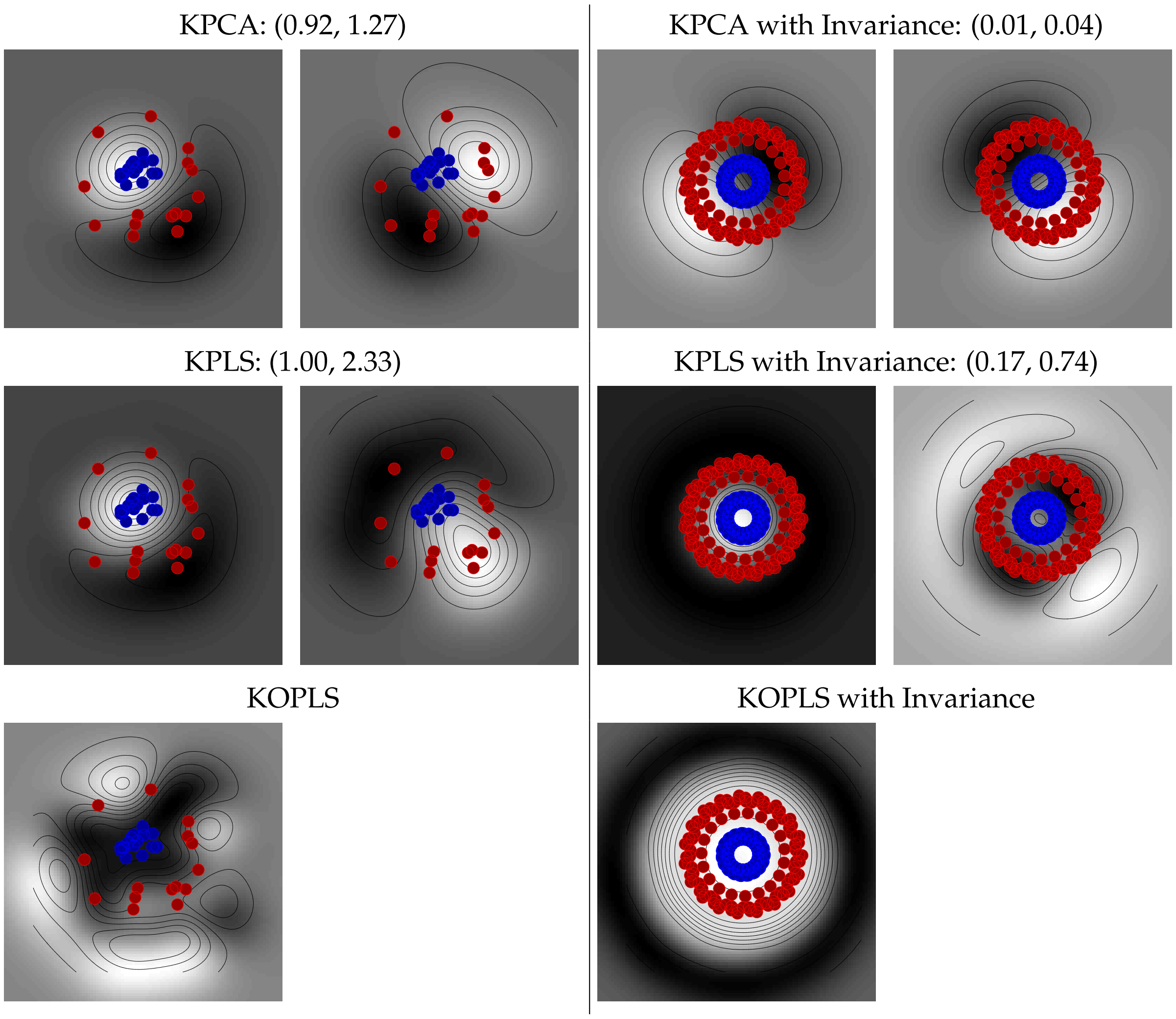


Figure 4.10: Example of differences between the kernels and the decision boundary of the standard SVM and Virtual SVM. Selected SVs are marked in cian.

samples to train the classifier and the remaining samples have been used for testing. One might think that the ''virtual kernel'' show less similarity between classes than the standard kernel, due to the fact that the first kernel is generated by the support vectors obtained by the second kernel. Nevertheless, the invariance produces a virtual kernel capable of distinguishing the classes and obtaining better accuracy than the standard RBF kernel.

We test the presented supervised feature extraction methods with the RBF kernel (using training samples) and the *virtual kernel* (using virtual samples) in order to compare the methods using both approximations. The database is bidimensional hence the number of the extracted features is maximally two, since both are limited by the rank of the centered kernel matrix. The exception is the KOPLS method that only can extract as maximum one dimension less than the number of classes in the database since is limited by the $rank(\tilde{\mathbf{K}}_x\mathbf{Y})$ (see Sec. 3.5).
Figure 4.11 shows the data projected onto the top two components. We observe that the principal features are modified when the virtual samples are added. We measured these changes using the Pearson's correlation coefficient and the mutual information between the projected data onto them. The first index is the normalized trace of the covariance matrix (Sriperumbudur et al., 2010) of the two principal components and the second one is the estimated mutual information between the data projected onto the first two principal components (Cover and Thomas, 2005).

KPCA and KPLS obtain features with both lower correlation and less mutual information using virtual samples than standard samples. That is, the two first components using virtual samples are more independent and less redundant than using the standard kernel. Note that the independence of the components does not necessary mean that the classification of the samples is better than that of dependent components. This may be particularly harmful in unsupervised feature extraction, as looking for independent features might result in not sufficiently discriminative power. In addition, note that classification in an ''independent'' domain is highly affected by the type of classifier used. Figure 4.12 shows the data test projected onto the two first components (KPCA and KPLS) and the accuracy of the classification. The KPCA, being an unsupervised method, obtains independent components but it is not capable of distinguishing



KPCA: (0.92, 1.27)

KPCA with Invariance: (0.01, 0.04)

KPLS: (1.00, 2.33)

KPLS with Invariance: (0.17, 0.74)

KOPLS

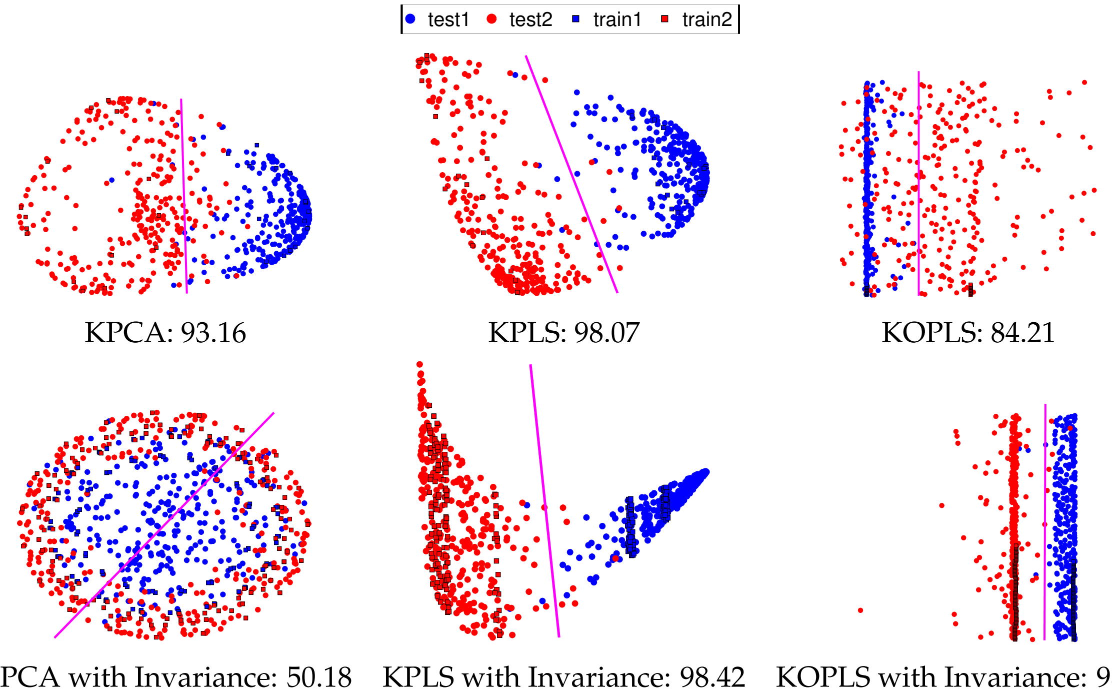
KOPLS with Invariance

Figure 4.11: Toy example: Principal components obtained by KPCA (top), KPLS (middle) and KOPLS (bottom) without invariances (left) and with invariances (right) using the training samples (red and blue dots). We also give the correlation and mutual information between features in parentheses.

both classes. Whereas the KPLS method increases the classifier accuracy using virtual samples. For the KOPLS method, we could not measure the correlation and mutual information since we can only extract one feature (number of features minus one). However, using only one feature, KOPLS method obtains similar results to KPLS method.

4.5 Summary

In this chapter, performance of supervised feature extraction methods (linear and nonlinear) has been analyzed. We have applied classification and regression algorithms using real databases to measure the quality of different FE projections. In general, KOPLS obtains better results than the other methods using a lower number of extracted features, not only in the regression but also in the classification scenario. This improvement of the results, especially in the regression



KPCA: 93.16

KPLS: 98.07

KOPLS: 84.21

KPCA with Invariance: 50.18

KPLS with Invariance: 98.42

KOPLS with Invariance: 98.42

Figure 4.12: Toy example: Decision boundary of the linear classier and the original data projected onto the two first dimensions in the case of KPCA (left) and KPLS (middle) and onto the first dimension in the case of KOPLS (right) without invariances (left) and with invariances (right).

cases, is due to the fact that KOPLS and OPLS found the projections that minimize the MSE. Whereas, KPLS and PLS projections are those projections that align with the output labels, and the KPCA and PCA projections are the directions that retain maximum variance without taking into account the labels of the samples. Consequently, much more discriminative projection vectors are typically extracted by KOPLS.

There are other ways to improve the accuracy of the classification or the error in the regression case. One of them is through the inclusion of *prior knowledge*. This knowledge was added here to the classifier by means of encoding the *invariances* into artificial generated examples. We introduced a simple method to include data invariances in SVM remote sensing image classification. We illustrated the performance in relevant remote sensing problems: invariance to rotations and reflections of image patches for contextual classification, object-scale invariances, and including prior knowledge on the way shadows affect the acquired images. Good classification accuracy was obtained in general when few labeled samples were available for training the models. Other invariances, from translation to illumination and canopy spectral invariances, and other kernel methods, from regression to clustering methods, could be explored. In all cases, the inclusion of physically-based models may lead to improved invariant statistical models.

Finally, we have introduced virtual samples into nonlinear feature extraction methods in order to study the performance of the methods using virtual samples. We have seen that the kernel constructed by virtual samples is decomposed in more independent features than the kernel constructed using available real samples, and therefore they lead to a redundant data representation. We have observed that the independence of the features does not guarantee an improvement of the classification accuracy, especially for KPCA. In general, we observed an improvement of KOPLS results using virtual samples.

The observations in this chapter confirm that, in spite of PCA and KPCA methods are the most widely used methods in remote sensing data processing, they are not always the most appropriate, so it would be advisable to analyze data before selecting the feature extraction method to apply them. Actually, accounting for labeled information guides the feature extraction better so KPLS or KOPLS would be a first choice in most of the cases. In cases of few labeled samples, an alternative would be including virtual samples in these methods to alleviate the possible overfitting issue.

Chapter 5

Advances in Unsupervised Kernel Feature Extraction

Contents

This Chapter describes our work in unsupervised kernel feature extraction by proposing two different approaches: an information theoretic kernel method and a generative kernel function. The first one is an optimized version of the KECA and seeks for the minimum number of components that maximize entropy and it is applied to pdf estimation, while the second one learns a kernel metric automatically from the data and it is used for clustering.

This Chapter is based on the works:

□ E. Izquierdo-Verdiguier, V. Laparra, R. Jenssen, L. Gómez-Chova, and G. Camps-Valls, *"Optimized Kernel Entropy Components,"* *IEEE Transactions Neural Network*, submitted (2014).

□ E. Izquierdo-Verdiguier, R. Jenssen, L. Gómez-Chova, and G. Camps-Valls, *"Spectral Clustering with the Probabilistic Cluster Kernel,"* *Neurocomputing*, submitted (2013), second review round.

5.1 Introduction

The previous chapter has introduced supervised kernel feature extraction: different feature extraction methods have been compared and the use of invariant kernels has been proposed in remote sensing. The use of supervised methods in remote sensing generates the necessity to work with labeled samples. However, this labeling process is expensive and costly as it involves terrestrial campaigns or photointerpreters in the case of classification problems or in situ measurements for biophysical parameter retrieval. Because of the difficulties in obtaining labeled samples and the associated statistical problems, unsupervised algorithms are typically preferred. Unsupervised algorithms try to find the most relevant features of the data manifold that describe the problem, either in terms of information, variance, or clustering.

The most widely used unsupervised feature extraction method in remote sensing is PCA. It is possible to find multitude of works that use PCA as a tool, for mutitemporal (Byrne et al., 1980), fusion (Pohl and Van Genderen, 1998) or dimensionality reduction (Jia and Richards, 1999). Nevertheless, this not imply that the PCA is the most appropriate feature extraction method in all cases (Cheriyadat and Bruce, 2003). Other unsupervised approaches have been recently used in remote sensing: Independent Component Analysis (ICA) and its nonlinear version for change detection (Marchesi and Bruzzone, 2009) or Kernel Entropy Component Analysis (KECA) for cloud detection (Gómez-Chova et al., 2012).

An important learning paradigm in unsupervised feature extraction is clustering, which has been applied in image fusion (Amorós-López et al., 2011) or in cloud screening (Gómez-Chova et al., 2007). Clustering is of fundamental importance in data analysis. This is reflected by the vast literature on the subject, including well-known methods such as k-means and Gaussian mixture models (GMMs) (Xu and Wunch II, 2008; Jain, 2010). Recently, very promising approaches to clustering have been proposed in the form of the interrelated kernel-based and graph-spectral techniques (Shawe-Taylor and Cristianini, 2004; von Luxburg, 2007; Filippone et al., 2008; Jain et al., 2012). These methods typically consist of two separated stages: First, features are generated based on the (top) eigenvalues and eigenvectors of a matrix that encodes similarities between pairs of data objects. Then, extracted features are globally clustered using k-means. The main advantages of such methods are their well-understood behavior in terms of linear algebra and their ability to correctly cluster both linear and nonlinear data structures.

In this Chapter, we propose two unsupervised kernel feature extraction methods: First, we propose the optimization of the kernel decompositionin KECA method (section 3.5.4), which is based on the ICA framework (Hyvärinen et al., 2001). With this optimization we will obtain features that are more efficient than KECA features for density estimation. Additionally, the selection of the kernel parameter critically affects the performance of both the KECA and the proposed method (Izquierdo-Verdiguier et al., 2014b) . Therefore, we also analyze the most common kernel length-scale selection criteria. Second, we present the Probabilistic Cluster kernel (Izquierdo-Verdiguier et al., 2013a), which is not only proposing a new kernel based on probabilistic models (section 2.3.5), but the idea of using this kind of kernels for data clustering, hence finalize the data clustering through clustering-based kernels. The Probabilistic Cluster Kernel (PCK) for data clustering is proposed as an unsupervised approach. In this setting, we have the need to generate a parameter-free kernel due to the lack of labeled samples (supervised information) that help in tuning parametrized kernel functions. Hence, our PCK is a parameter-free kernel learned directly from the data. Furthermore, the PCK captures the data manifold structure at different scales and, therefore, we can better cover data manifolds than with other kernel types, such as the RBF.

5.2 Optimized Kernel Entropy Component Analysis (OKECA)

Kernel entropy component analysis was proposed in pattern analysis and machine intelligence (Jenssen (2010), section 3.5.4). It has proven useful in different applications e.g. remote sensing data analysis (Gómez-Chova et al., 2012; Luo and Wu, 2012; Luo et al., 2013), face recognition (Shekar et al., 2011), chemical processes modelling (Jiang et al., 2013), high-dimensional celestial spectra reduction (Hu et al., 2013) and audio processing (Xie and Guan, 2012). Several extensions have been proposed for feature selection (Luo et al., 2012), class-dependent feature extraction (Cheng et al., 2011) and semisupervised learning as well (Myhre and Jenssen, 2012).

One distinguishing feature of KECA is that the method originates from kernel density estimation (KDE) (Silverman, 1986; Girolami, 2002b; Duin, 1976), as do principal curves estimation (Ozertem and Erdogmus, 2011) and the family of information theoretic learning methods (Principe, 2010). In KDE, the key is the kernel function, locally approximating the underlying probability density function (pdf). This in turn enables estimation of entropy, a quantity that describes the shape of the pdf (Cover and Thomas, 2005). The KDE kernel must be a non-negative function that integrates to one (i.e. a density) but needs not be positive semi-definite (PSD). The KDE kernel is versatile since it is not limited to PSD. However, many KDE kernels are PSD, well-known examples include the Gaussian kernel, the Student kernel, and the Laplacian kernel (Kim and Scott, 2012). If the KDE kernel used in KECA is PSD, then there are close relations to the aforementioned *kernel* signal processing methods, in the sense that the kernel computes an inner-product in a reproducing kernel Hilbert space (RKHS). In this situation, KPCA, KCCA and KPLS are based on RKHS learning algorithms to maximize e.g. the feature space variance, correlation or alignment with the output variables. PSD KECA hence bridges KDE, information theoretic learning and RKHS learning.

Although both KDE and kernel methods have experienced great success, all kernel-based methods, including the one proposed in this section, are sensitive to the kernel function used. For instance, many kernel methods depend heavily on a bandwidth, or length scale, parameter. In addition, all the aforementioned spectral methods may need a considerable number of components (eigenvalues and eigenvectors) in order to properly describe the data. This may be undesirable e.g. in compression and data visualization contexts.

Here, we take advantage of the KDE foundation of KECA (see also Girolami (2002b) for further details), and introduce an optimization procedure aiming at compressing the entropy information into optimal directions in feature space. To accomplish this goal, we introduce a rotation procedure that resembles the one in Independent Component Analysis (ICA) (Hyvärinen et al., 2001). The resulting Optimized KECA (OKECA) employs a gradient descent method for searching the new features. Two major benefits stand out with respect to the extracted OKECA components:

1. OKECA shows great robustness to the kernel bandwidth parameter. This is important, as there is no universally accepted kernel size selection procedure for unsupervised KDE-based kernel methods.

2. We use OKECA in order to improve the KDE. This is achieved based on far fewer components compared to KECA.

The rest of the section is divided as follows. Section 5.2.1 presents the OKECA formulation and proposes a density estimation that exploits kernel feature characteristics. Section 5.3.3 is devoted to the analysis of the results. We use OKECA as a feature-extraction method and analyze the retained entropy, show the estimated pdf, and perform data classification.

5.2.1 Proposed Optimized Kernel Entropy Components (OKECA)

As mentioned before, if the KDE kernel is PSD, then there is a close connection between KECA and un-centered KPCA since the kernel function in that case reproduces the dot product between two samples mapped to a RKHS $\mathcal{H}$ via $\boldsymbol{\phi}(\cdot)$, i.e. $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_j)$. Note that centering of the kernel matrix $\mathbf{K}$ makes no sense in the KDE and entropy context of Eq. (3.22), as this would correspond to $v = 0$, i.e. infinite entropy. Hence, $\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{U}^\top$ is the uncentered projection of the feature space data $\mathcal{D}_\mathcal{H} = \{\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)\}$ onto all the principal axes in the feature space (Jenssen, 2010, 2013a). These projections may be sorted according to their contribution to the input space entropy as measured by the information potential (the entropy values), constituting the KECA procedure (see Section 3.5.4).
However, the projections and their entropy content are fully dependent on the quality of the KDE performed via the kernel function. Moreover, using the eigendecomposition procedure may not be optimal to find the best projections from an entropy perspective.

Optimized KECA (OKECA)

The novel approach proposed in this Thesis searches for a basis that maximizes the information potential in as few components as possible. The procedure corresponds to optimally capturing in these components the low entropy part of the data, which typically corresponds to the structure of the data in terms of class or cluster information.

To that end, we present a solution motivated by the classical Independent Component Analysis (ICA) formulation (Hyvärinen and Oja, 2000) in which, after the whitening step (applying $\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^\top$), there is an extra rotation (applying $\mathbf{W}^\top$) that maximizes the independence between components. Note that $\mathbf{W}$ is an orthonormal linear transformation, i.e. $\mathbf{W}\mathbf{W}^\top = \mathbf{I}$. Similar ideas have been applied in kernel-based component analysis (see for instance Pan and Yang (2011)). Following the ICA rationale, we now aim at a new kernel matrix decomposition:

$$\mathbf{K} = \mathbf{C}\mathbf{C}^\top = (\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{W})(\mathbf{W}^\top\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^\top). \tag{5.1}$$

Note that the kernel matrix does not change (in relation to Eq. (3.21)), but the modification allows us to directly find the basis that maximize the information potential with respect to the number of retained components. Therefore, for each column vector $\mathbf{w}_k$ in $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_n]$, we maximize:

$$\mathcal{L} = \left(\mathbf{1}_n^\top\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{w}_k\right)^2, \tag{5.2}$$

where each $\mathbf{w}_k$ is restricted to be normal $\|\mathbf{w}_k\|_2 = 1$ and to be orthonormal to the previous $\mathbf{w}_l$, $\forall l < k$. This deflationary procedure ensures that the obtained solution retains more (or equal) information potential than the one obtained by the standard KECA in fewer components.

In order to solve the OKECA optimization problem in Eq. (5.2), a gradient-descent approach can be followed:

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \tau\frac{\partial\mathcal{L}}{\partial\mathbf{w}_k(t)}, \tag{5.3}$$

where τ is the step size and the gradient is:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}_k} = 2(\mathbf{1}_n^\top\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{w}_k)(\mathbf{1}_n^\top\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}})^\top. \tag{5.4}$$

A Matlab implementation of the algorithm is available at `http://isp.uv.es/code/okeca.htm` for the interested reader. While other more sophisticated optimization algorithms could be deployed here, in our experiments we observed that this simple gradient-descent strategy performed consistently even in the presence of noise.

Kernel decomposition on density estimation

This section illustrates the benefits of using the proposed decomposition for KDE (Parzen, 1962). KDE is a classical method for estimating a pdf in a non-parametric way. Essentially, KDE defines the pdf as a sum of kernel functions, $K(\cdot, \mathbf{x}_i)$, defined over the training dataset $\mathcal{D}$ as follows:

$$\hat{p}(\mathbf{x}_*) = \frac{1}{n}\sum_{i=1}^{n} K(\mathbf{x}_*, \mathbf{x}_i). \tag{5.5}$$

As mentioned before, KDE kernel functions do not need in general to be PSD but have to be nonnegative and integrate to one to ensure that $\hat{p}$ is a valid probability density function. A classical example of such a kernel function is the Gaussian distribution, $K(\mathbf{x}_*, \mathbf{x}_i) = (2\pi\sigma^d)^{-1/2}$ $\exp(-\|\mathbf{x}_* - \mathbf{x}_i\|^2/(2\sigma^2))$, but as mentioned, other choices exist. Then, the corresponding kernel matrix can be used for KDE

$$\hat{p}(\mathbf{x}_*) = \frac{1}{n}\sum_{i=1}^{n} K(\mathbf{x}_*, \mathbf{x}_i) = \frac{1}{n}\mathbf{1}_n^\top \mathbf{k}_*, \tag{5.6}$$

where $\mathbf{k}_*$ is the vector of kernel evaluations between the point of interest $\mathbf{x}_*$ and all samples in the dataset $\mathcal{D}$. As explained in Girolami (2002b), if the decomposition of the uncentered kernel matrix follows the form $\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{U}$ is orthonormal and $\mathbf{\Lambda}$ is a diagonal matrix, then the kernel-based density estimation may be expressed as

$$\hat{p}(\mathbf{x}_*) = \mathbf{1}_n^\top \mathbf{U}_r \mathbf{U}_r^\top \mathbf{k}_*, \tag{5.7}$$

where $\mathbf{U}_r$ is the reduced version of $\mathbf{U}$ by keeping columns for $r < n$. Note that when using $\mathbf{U}$ instead of $\mathbf{U}_r$, Eq. (5.7) reduces to Eq. (5.6). This shows that the retained KECA components may be used for KDE (Girolami, 2002b) by selecting the dimensions that maximize the information potential in Eq. (3.22).

A novel aspect of the proposal is to use the OKECA components for KDE in a similar manner. Note that the KECA decomposition in Eq. (3.21) is not exactly the same as the proposed OKECA in Eq. (5.1). Nevertheless, it is easy to find a basis that fulfills the same decomposition form, i.e. $\widetilde{\mathbf{U}}\widetilde{\mathbf{\Lambda}} = \mathbf{C}$, where $\widetilde{\mathbf{U}}$ is the $\mathbf{C}$ matrix with normalized column vectors and $\widetilde{\mathbf{\Lambda}}$ is diagonal matrix containing the norms of each column in $\mathbf{C}$. Therefore, Eq. (5.7) for OKECA reduces to $\hat{p}(\mathbf{x}_*) = \mathbf{1}_n^\top \widetilde{\mathbf{U}}_r \widetilde{\mathbf{U}}_r^\top \mathbf{k}_*$.

Model estimation

In this work, we consider the Gaussian RBF kernel since it is the most common and versatile in both RKHS kernel methods and KDE (Parzen, 1962). This kernel induces a probabilistic Gaussian mixture model, and it only introduces one scalar free parameter, σ. Note that more complicated models could be taken into account in both frameworks. However, a recurrent and unsolved problem in both approaches is the estimation of the length-scale parameter σ.

A plethora of heuristics and rules for estimating the length-scale have been proposed in the machine learning and statistics literatures. Roughly speaking, one finds two main approximations. The first approach considers maximizing a particular objective function through a cross-validation procedure. The objective function may be optimized using unsupervised (e.g. maximum likelihood (Duin, 1976), denoted by σ_{ML} in the experiments) or supervised (e.g. a classification accuracy score, denoted by σ_{class} in the experiments) approaches (Section 2.3.6). The second approach resorts to empirical rules of performance or theoretical bounds. Good

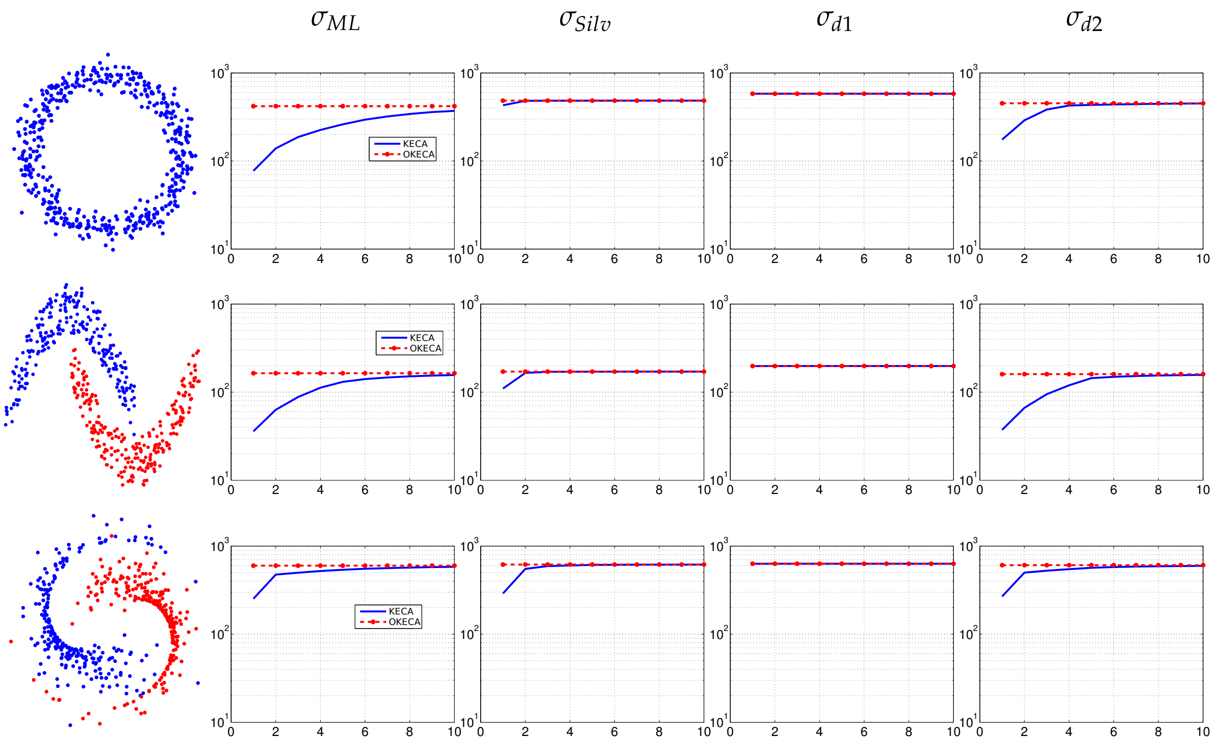


Figure 5.1: Cumulative estimated information potential versus the number of dimension components for different toy datasets, using KECA and OKECA and different σ estimation approaches.

examples of this second approach, which are considered in this section, are: 1) the Silverman's rule (Silverman, 1986), which is the classical rule of thumb in KDE (see Section 2.3.6), σ_{Silv} in the experiments; 2) the mean distance between training points, which is a common approach in kernel methods for classification, σ_{d1} in the experiments; and 3) the 15% of the median distance between points, which is the classical employed in KECA, σ_{d2} in the experiments.

5.2.2 Experiments

We compare the performance of the standard KECA and the proposed OKECA for both density estimation and data classification. We analyze the methods in terms of the retained information potential as a function of the extracted features, the impact of the model selection criteria, and the classification accuracies in synthetic and real datasets.

OKECA for optimally entropic representations

The first experiment considers three well-known 2D toy examples for analyzing the methods: a *ring*-shaped distribution consisting of one class only, and the binary *two-moons* and *pinwheel* datasets. In this section, we illustrate the ability of the proposed method to obtain projections that maximize the information potential, hence minimizing the squared Rényi entropy. In the results, we used 80, 20 and 45 training samples for each problem, respectively.

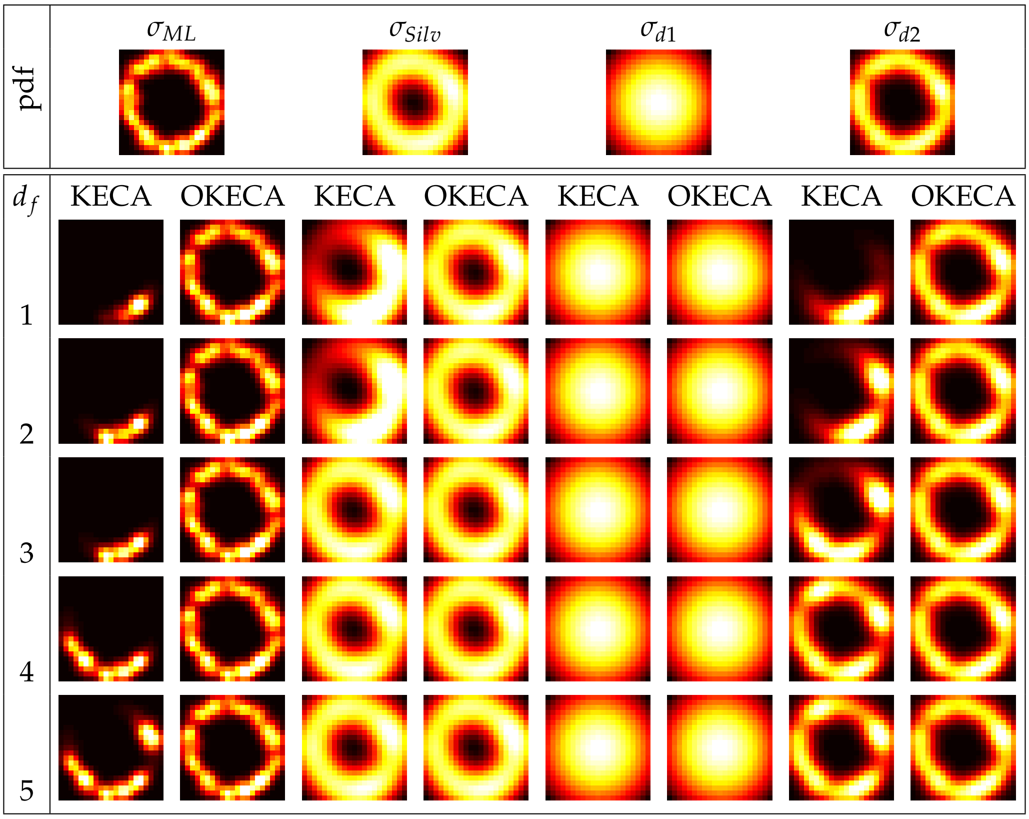


Figure 5.2: Density estimation for the ring dataset by KECA and OKECA using different number of extracted features d_f and approaches to estimate the kernel lengthscale parameter σ. Black color represents low pdf values and yellow color high pdf values.

Figure 5.1 shows the original data distributions and the estimated cumulative information potential (v and $\mathcal{L}$ defined in (3.22) and (5.2), respectively) attained by KECA and OKECA as a function of the 10 top components and all the considered kernel length-scale selection criteria. For all datasets and for all σ values, OKECA reaches almost the maximum entropy value with just one feature; whereas KECA cumulative entropy values need five or more components to saturate. This effect is almost independent of the chosen criterion to set the σ parameter. The higher information content may translate into more informative features potentially useful for density estimation and classification, as we illustrate in the next sections.

OKECA for pdf estimation

Figure 5.2 illustrates the ability of KECA and OKECA for density estimation in the ring dataset. We merely applied Eq. (5.7) for a different number of components r in $\mathbf{U}_r$. This figure should be analyzed together with Fig. 5.1[top]. Note that, for the proposed OKECA, the first projection concentrates most of the entropy information. This agrees with the fact that just one dimension is needed to obtain a good pdf estimation. On the contrary, KECA cannot estimate correctly the pdf using only the first component and actually needs at least five components. This issue

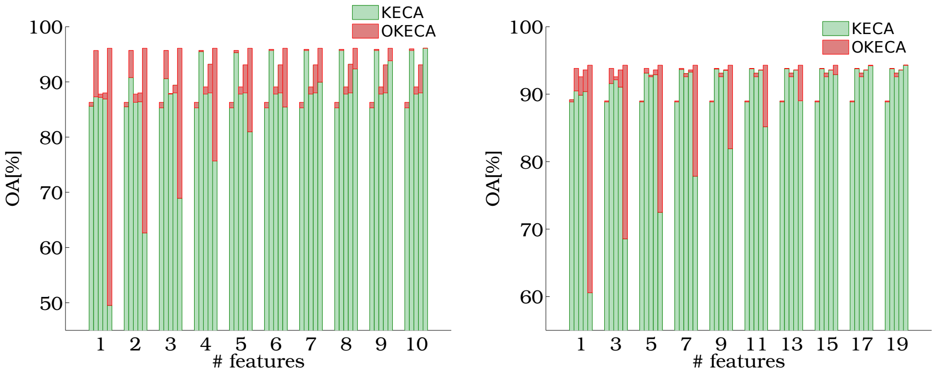


Figure 5.3: Overall Accuracy obtained with *two-moons* (left) and *pinwheel* (right) datasets. The five bars for every number of retained features are, from left to right: σ_{d1}, σ_{d2}, σ_{Silv}, σ_{ML} and σ_{class}.

is even more dramatic when using σ_{ML}, see Fig. 5.2. It is worth noting that σ_{ML} and σ_{d2} give rise the best pdf estimates in OKECA.

OKECA components for data classification

We here illustrate the capabilities of OKECA for data classification. The experiments are conducted on a wide range of synthetic and real problems: 1) The *two moons* and the *pinwheel* datasets considered previously in Sec. 5.2.2; 2) Six real datasets from the University California Irvine (UCI) Machine Learning Repository[1]; and 3) A real satellite multispectral image classification problem. In order to evaluate the classification performance, we have used the overall classification accuracy (OA) which is obtained as the average of samples correctly predicted in percentage terms. While one could classify on top of the extracted features, we here rely intentionally on the class-dependent estimated densities and perform maximum a posteriori (MAP) classification.

Synthetic datasets

Figure 5.3 shows the OA test obtained with different σ values and different number of retained dimensions with KECA and the proposed OKECA on the *two-moons* and *pinwheel* datasets. The five bars for every number of retained features are, from left to right: σ_{d1}, σ_{d2}, σ_{Silv}, σ_{ML} and σ_{class}. The value of σ_{class} has been optimized for classification using all features in a 5-fold cross-validation scheme. We used 20 samples and 45 samples *per* class for training *two-moons* and *pinwheel* respectively, and 500 *per* class for testing the models and computing the OA test in both datasets. Note that the OKECA method achieves better classification results than KECA for all σ values, confirming that to seek for optimally entropic data descriptors may benefit classification. Smaller differences between methods are observed as the number of components increases. When all n features are used, OKECA and KECA are trivially equivalent.

[1] `http://archive.ics.uci.edu/ml/datasets.html`

Table 5.1: Overall accuracy, OA[%], obtained by KECA and OKECA methods using different values of noise. In bold the highest OA for each number of features.

Noise, σ_n	0.001		0.051		0.091	
# dim	*KECA*	*OKECA*	*KECA*	*OKECA*	*KECA*	*OKECA*
1	57.1	**93.8**	73.2	**90.7**	79.0	**85.0**
2	60.3	**93.8**	76.9	**90.7**	78.6	**85.0**
3	63.0	**93.8**	79.3	**90.7**	80.0	**85.0**
4	66.6	**93.8**	81.5	**90.7**	81.1	**85.0**
5	69.2	**93.8**	83.6	**90.7**	82.1	**85.0**
6	71.5	**93.8**	85.1	**90.7**	82.4	**85.0**
7	74.3	**93.8**	86.9	**90.7**	83.2	**85.0**
8	77.0	**93.8**	88.1	**90.7**	83.6	**85.0**
9	79.1	**93.8**	89.4	**90.7**	84.6	**85.0**
10	80.6	**93.8**	90.3	**90.7**	84.8	**85.0**
11	82.5	**93.8**	90.5	**90.7**	84.9	**85.0**
12	84.9	**93.8**	**90.7**	**90.7**	84.9	**85.0**
13	86.8	**93.8**	**90.7**	**90.7**	84.9	**85.0**
14	89.4	**93.8**	**90.7**	**90.7**	84.9	**85.0**
15	91.2	**93.8**	**90.7**	**90.7**	84.9	**85.0**
16	92.7	**93.8**	**90.7**	**90.7**	84.9	**85.0**
17	93.4	**93.8**	**90.7**	**90.8**	84.9	**85.0**
18	93.7	**93.8**	**90.7**	**90.7**	84.9	**85.0**
19	**93.8**	**93.8**	**90.7**	**90.7**	84.9	**85.0**
20	**93.8**	**93.8**	**90.7**	**90.7**	84.9	**85.0**

In the rest of the section, we discuss the capabilities of OKECA in the presence of distorted distributions. The question raised is how sensitive the optimization algorithm is to the presence of noise. To this end, a toy example of the KECA and OKECA projections in presence of noise is considered. We used 50 samples of *two-moons* dataset to training and 500 samples to test the classifier. Gaussian noise was added to the original data distributions by varying the

Table 5.2: UCI database description (d: number of dimensions, n_c: number of classes, N_{train}: number of training samples, and N_{test}: number of test samples.

Database	m	d	n_c	N_{train}	N_{test}
Ionosphere	351	33	2	80	172
Letter	20000	16	26	1014	3874
Pendigits	10992	16	9	540	3498
Pima-Indians	768	8	2	200	330
Vowel	990	12	10	150	330
wdbc	569	30	2	80	344

Table 5.3: Overall accuracy obtained with UCI database using KECA and OKECA methods with different number of dimensions.

	ionosphere		Letter		Pendigits	
# dim	*KECA*	*OKECA*	*KECA*	*OKECA*	*KECA*	*OKECA*
1	76.9	**78.9**	35.8	**69.9**	74.1	**93.4**
2	76.8	**78.9**	45.1	**69.9**	80.0	**93.4**
3	76.4	**78.9**	50.9	**69.9**	82.4	**93.4**
4	76.6	**78.9**	55.1	**69.9**	84.1	**93.4**
5	76.8	**78.9**	58.3	**69.9**	85.5	**93.4**
6	77.0	**78.9**	61.0	**69.9**	86.9	**93.4**
7	77.1	**78.9**	63.3	**69.9**	87.9	**93.4**
8	77.4	**78.9**	65.3	**69.9**	89.3	**93.4**
9	77.2	**78.9**	66.8	**69.9**	90.4	**93.4**
10	77.3	**78.9**	67.9	**69.9**	91.7	**93.4**

	pima		Vowel		wdbc	
# dim	*KECA*	*OKECA*	*KECA*	*OKECA*	*KECA*	*OKECA*
1	55.3	**62.9**	33.1	**83.6**	79.8	**87.1**
2	58.5	**62.9**	41.5	**83.6**	82.8	**87.1**
3	61.2	**62.9**	47.0	**83.6**	86.1	**88.5**
4	62.2	**62.9**	59.0	**83.6**	87.7	**89.3**
5	62.4	**62.9**	64.4	**83.6**	88.3	**89.9**
6	62.3	**62.9**	72.6	**83.6**	88.8	**90.2**
7	62.3	**62.9**	80.3	**83.6**	89.0	**90.4**
8	62.3	**62.9**	83.3	**83.6**	89.0	**90.5**
9	62.3	**62.9**	83.3	**83.6**	89.3	**90.5**
10	62.6	**62.9**	83.3	**83.6**	89.8	**90.5**

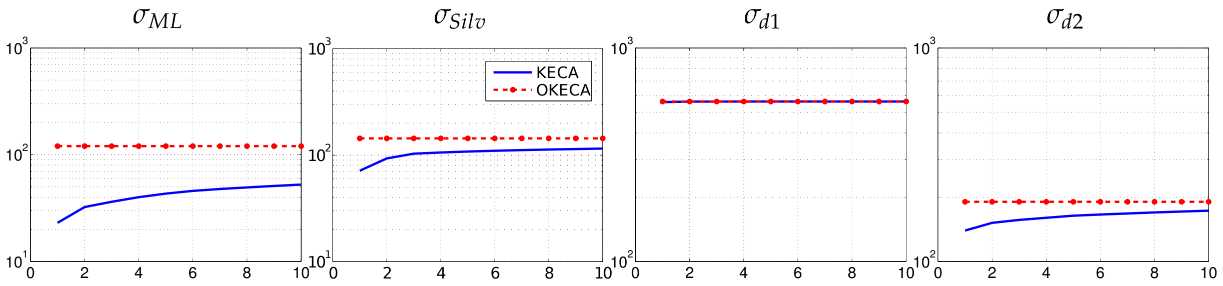


Figure 5.4: The cumulative information potential for the multispectral image using KECA and OKECA and different σ estimation approaches.

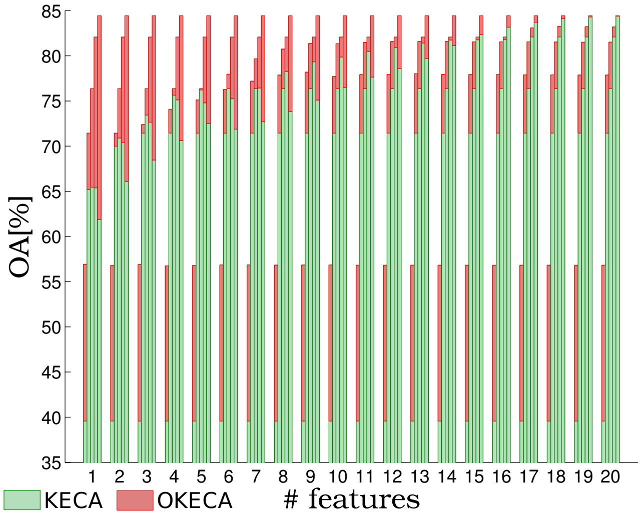


Figure 5.5: Classification results for the Zürich QuickBird satellite image for different σ values and number of retained dimensions by KECA and OKECA. The five bars for every number of retained features are, from left to right: σ_{d1}, σ_{d2}, σ_{Silv}, σ_{ML} and σ_{class}.

dimension-wise standard deviation of the Gaussian noise σ_n from 0.001 to 0.091. Numerical results are shown in Table 5.1. Note that the difference of OA between methods is reduced when the noise increases, but even in the high-noise regime, OKECA needs far less features to outperform standard KECA.

UCI benchmark datasets

We used the six datasets from the UCI machine learning repository of different sizes and dimensionality described in Section 4.2.1. Table 5.2 gives details on the dimensionality, number of classes and training and test samples used in the experiments that follow.

We run KECA and OKECA for all datasets for different numbers of extracted components. The average of the OA for the ten first dimensions is shown in Table 5.3. In this case we restrict ourselves to σ_{ML} because of the good performance in the previous experiments and for the sake of simplicity. In general, the OKECA method outperforms the KECA method and, as observed before, OKECA saturates its performance with just the first extracted dimension.

Multispectral VHR image classification

In this experiment, we apply KECA and OKECA to the segmentation of remotely-sensed multispectral images. We consider a real multispectral image acquired over a residential neighbourhood of the city of Zürich by the QuickBird satellite in 2002 (see Section 4.2.1).

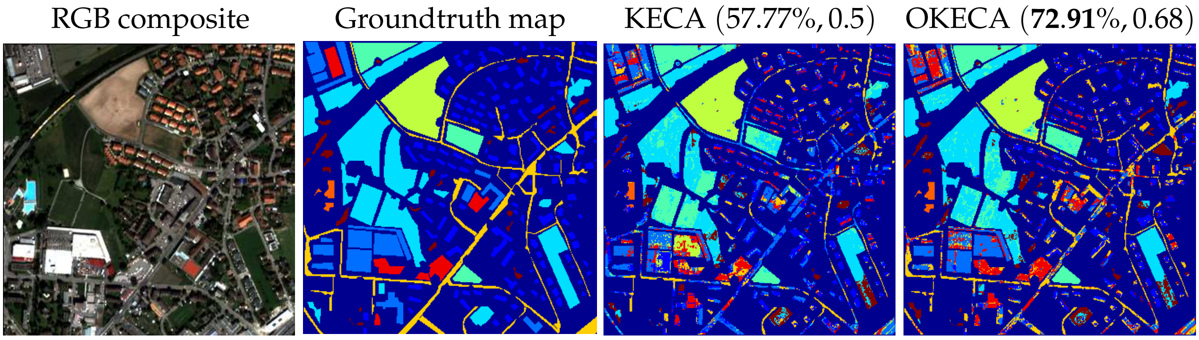


Figure 5.6: Classification maps for the Zürich QuickBird satellite image using three features and σ_{class} in KECA and OKECA. Top-left: RBG version of the original image; top-right: groundtruth classification map (each color represents a different landcover class); bottom-left: classification map obtained with KECA; bottom-right: classification map obtained with OKECA.

Table 5.4: Confusion Matrix yield by the three retained features and σ_{class} in the test set (whole scene Fig. 5.6, u: user's accuracy [%] and p: producer's accuracy [%]).

KECA

	True classes									u	classes
Predicted classes	**3884**	808	75	2	0	1741	70	186	384	54.32	residential buildings
	1088	**2492**	46	7	20	2018	4	416	134	40.03	commercial buildings
	46	6	**8397**	430	1	367	0	0	100	89.84	meadows
	16	28	2901	**2067**	21	91	0	9	0	40.27	harvest vegetation
	0	721	0	5	**3775**	53	0	6	0	82.79	bare soil
	854	218	25	0	1	**1554**	11	388	23	50.55	asphalt
	1	0	0	0	0	0	**180**	0	0	99.45	pools
	747	989	2	0	4	288	4	**744**	0	26.78	parkings
	110	15	1677	12	0	46	0	0	**454**	19.62	trees
p	57.57	47.22	63.99	81.93	98.77	25.24	66.91	42.54	41.46		

OKECA

	True classes									u	classes
Predicted classes	**4238**	190	29	0	0	639	0	26	23	82.37	residential buildings
	366	**3130**	6	1	42	676	0	145	1	71.67	commercial buildings
	26	8	**9840**	71	0	168	0	0	72	96.61	meadows
	4	49	1834	**2432**	117	15	0	7	4	54.50	harvest vegetation
	0	47	0	2	**3657**	15	0	1	0	98.25	bare soil
	1356	610	8	1	1	**3838**	3	239	1	63.36	asphalt
	0	4	0	0	0	2	**262**	1	0	97.40	pools
	331	1208	16	0	5	695	4	**1327**	0	37.01	parkings
	425	31	1390	16	0	110	0	3	**994**	33.48	trees
p	62.82	59.31	74.98	96.39	95.68	62.33	97.40	75.87	90.78		

The KECA and OKECA cumulative information potential values follow similar trends to the toy examples (see Fig. 5.4). OKECA reaches the maximum with just one feature, while KECA needs much more components to achieve similar informative content, especially noticeable for the σ_{ML} and σ_{d2} criteria. Such dependence with the criterion is not shared by OKECA. These results suggest that the sharpness in the component selection made by OKECA is relevant in cases of high feature redundancy as well.

Figure 5.5 shows the classification results obtained using different σ values and different number of retained dimensions. In this case, we use 22 and 200 samples *per* class for training and testing the models, respectively. Both methods achieve the best results using the σ_{ML} and σ_{class} criteria. Finally, note that σ_{d1}, which is a common choice in unsupervised kernel methods, provides very poor results for both methods. Figure 5.6 shows the classification maps obtained using three retained features and σ_{class} for both methods. Note how OKECA outperforms KECA in general for all the classes (Table 5.4).

5.3 Probabilistic cluster kernel (PCK)

In the previous section, we have proposed an optimized version of the unsupervised KECA feature extraction method and studied procedures for the selection of the kernel scale size for unsupervised KDE. Actually, the bottleneck of KECA but also of most kernel feature extraction methods is the selection of the kernel parameters. This section presents a novel approach that aims to mitigate this relevant machine learning problem.

5.3.1 Introduction to generative kernels

The similarity (kernel) matrix is commonly based on a parameterized function such as the radial basis function (RBF). As mencioned before (Section 5.2), the important parameter in RBFs is the width, which basically determines a fixed scale of analysis, and the choice of this parameter is of paramount importance. Lately some probabilistic (often referred such as generative) approaches have been introduced to design kernel functions that capture the signal characteristics. Among them, we stand out the Fisher kernel (Jaakkola et al., 1999), other generative approaches (Bicego et al., 2013), kernels that accommodate particular characteristics of the expected signal distribution (Campbell et al., 2006; Carli et al., 2014), and kernels based on GMM (You et al., 2010). All these kernel functions have shown very good results, but three main shortcomings still arise: 1) they all require first assuming a data generative model (e.g. Gaussian (You et al., 2010), Riccian (Carli et al., 2014), etc.) for which explicit metaparameter-dependent feature extractors need to be derived; 2) they have all been specifically designed and applied to supervised problems, mainly through the Support Vector Machine (SVM); and 3) they need *a priori* knowledge about the data to fix some parameters. These problems prevent using such kernels for data clustering, as no prior knowledge (besides the number of clusters) is assumed. In this Thesis, we address these issues by presenting a parameter-free kernel function based on *clustering for data clustering*.

In particular, we take a different approach to spectral clustering, wherein the feature generation process is obtained not separately from the clustering, but as a part of an integrated process. The idea is to encode similarity between objects using their probability of being grouped together at different scales, which is obtained from multiple "weak" learners based on GMM clustering. These local linear clusterings are then combined to build a global multiscale kernel that is used for spectral decomposition. As a result, an ensemble of linear clusterings enables nonlinear clustering.

The key quantity we introduce is a generative probabilistic cluster kernel function that is learned directly from the data by looking at local-to-global similarities along the manifold. This entails no parameter tuning, which is especially beneficial in the current context of unsupervised clustering. We analyze the main properties of the kernel and compare it to the standard RBF kernel and other kernel clustering approaches. The structure, informative content, optimality and spectral decomposition are studied. Analysis and performance are illustrated in several real problems.

5.3.2 Proposed Probabilistic Cluster Kernel (PCK)

Given n data points $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$, the proposed generative kernel, $K_c(\mathbf{x}_i, \mathbf{x}_j)$, is directly learned by clustering the available data. Building K_c requires first running a clustering algorithm, such as Expectation-Maximization (EM) assuming a Gaussian mixture model (GMM) with different initializations, $q = 1, \ldots, Q$, and with different number of clusters, $g = 2, \ldots, G+1$. These results in $m = Q \cdot G$ cluster assignments where each sample $\mathbf{x}_i$ has its corresponding posterior probability vector $\boldsymbol{\pi}_{i,g}(q) \in \mathbb{R}^g$. The *probabilistic cluster kernel* K_c is then computed as a composite kernel by averaging all the dot products between the posterior probability vectors (Jebara et al., 2004)

$$K_c(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{Z} \sum_{q=1}^{Q} \sum_{g=2}^{G+1} \boldsymbol{\pi}_{i,g}(q)^\top \boldsymbol{\pi}_{j,g}(q), \tag{5.8}$$

where Z is a normalization factor. After kernel construction with a sufficiently large number of clusters G and realizations Q, we proceed as in the standard spectral clustering approach, described above.

An illustrative toy example of the multiscale cluster kernel construction is shown in Fig. 5.7. Intuitively, the probabilistic cluster kernel accounts for *probabilistic* similarities at small and large scales (which are related to the number of clusters, since a higher number of clusters implies local scales and *vice versa*) between all samples along the data manifold. Actually, the

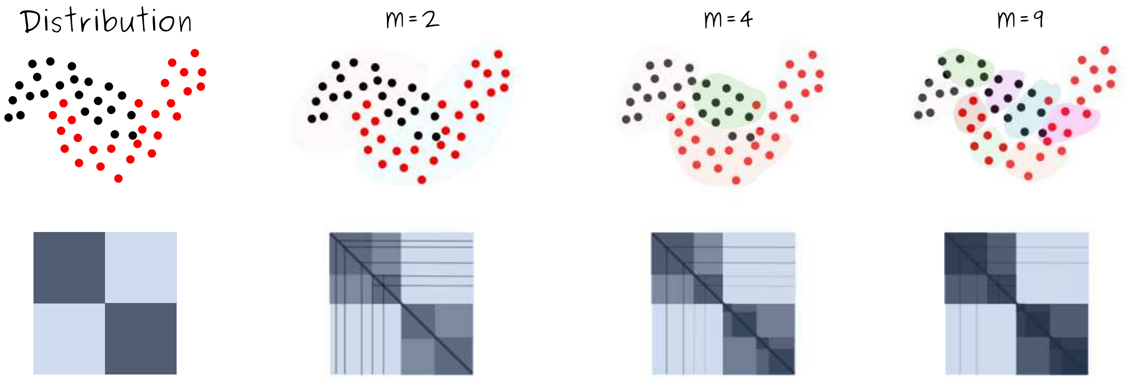


Figure 5.7: Illustration of the construction of the probabilistic cluster kernel. The method clusters data with EM-GMM clustering for $m = \{2, 4, 9\}$, the posterior probability vectors are used to compute the dot products leading to the cluster kernel explicitly, and after repeating the process for a number of clusters, it accumulates the similarities in a multi-scale way. Samples with similar probabilities of membership to a group should belong to the same class. The multiscale cluster kernel (right kernel) is a better estimation of the optimal ideal kernel $\mathbf{K}_{ideal} = \mathbf{Y}\mathbf{Y}^\top$ (left kernel). Based on Tuia and Camps-Valls (2011)

proposed kernel has the very important advantage that it does not assume an *ad hoc* parametric form or sophisticated priors and thus is more flexible and general. Moreover, the method does not require computationally demanding procedures (e.g. EM-GMM clustering algorithms scale linearly with n). Finally, note that the proposed kernel generalizes previous (semi) supervised approaches based on cluster kernels, e.g. the approach in Weston et al. (2005) is obtained when solely the cluster assignment with maximum posterior probability is considered. Moreover, it is worth noting that the proposed multiscale approach might also be applied to other generative kernels such as the Fisher kernel (Jaakkola et al., 1999).

Properties

Here we will describe the main theoretical properties of the proposed cluster kernel in a Hilbert space.

Property 1. *The probabilistic cluster kernel performs a linear kernel in a posterior probability space.*

Proof. From Eq. (5.8), an arbitrary kernel function that forms the probabilistic cluster kernel is $K_c(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle = \langle \boldsymbol{\pi}_i, \boldsymbol{\pi}_j \rangle$, and then the explicit feature mapping is $\boldsymbol{\phi}(\mathbf{x}_i) = \boldsymbol{\pi}_i$. Therefore, the probabilistic cluster kernel computes second-order statistics in a probability space. □

Property 2. *The probabilistic cluster kernel K_c is a positive definite (p.d.) kernel.*

Proof. The function $K_c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a p.d. kernel *if and only if* there exists a Hilbert space $\mathcal{H}$ and a feature map $\phi : \mathcal{X} \to \mathcal{H}$ such that for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ we have $K_c(\mathbf{x}, \mathbf{x}') = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') \rangle_{\mathcal{H}}$. Using standard properties of kernel functions and property 1, and as a simple consequence of the bilinearity of the dot product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, then $\forall c_i \in \mathbb{R}$:

$$\sum_{i,j=1}^{n} c_i c_j K_c(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i,j=1}^{n} c_i c_j \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle_{\mathcal{H}} = \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \left\langle \boldsymbol{\pi}_i, \boldsymbol{\pi}_j \right\rangle_{\mathcal{H}}$$

$$= \left\langle \sum_{i=1}^{n} c_i \boldsymbol{\pi}_i, \sum_{j=1}^{n} c_j \boldsymbol{\pi}_j \right\rangle_{\mathcal{H}} = \|c\boldsymbol{\pi}\|_{\mathcal{H}}^2 > 0.$$

since $K_c(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\pi}_i, \boldsymbol{\pi}_j \rangle$ is symmetric positive definite, where $\boldsymbol{\pi}_i \in \mathbb{R}^g$. □

Property 3. *The probabilistic cluster kernel K_c in* (5.8) *is a valid Mercer's kernel.*

Proof. The kernel is a weighted summation of valid kernels (see property 2), which can be shown to be a valid kernel (Reed and Simon, 1980). The corresponding mapping to a summation of inner products in QG-dimensional spaces, $\boldsymbol{\phi}(\mathbf{x}_i) = \bigcup_{q=1,g=2}^{Q,G} \boldsymbol{\pi}_{i,g}(q)$, where operator $\bigcup$ represents vector concatenation. The mapping induced by the sum can be expressed as a concatenation of different multiscale mappings ($\boldsymbol{\phi}_{q,g}$), $\boldsymbol{\phi}(\mathbf{x}_i) = \bigcup_{q=1,g=2}^{Q,G} \boldsymbol{\phi}_{q,g}$. □

5.3.3 Experimental results

Data collection

We analyze the proposed kernel and illustrate its capabilities for data clustering in four challenging, high dimensional real problems: two UCI machine learning repository datasets[2], and the segmentation of two satellite images using their reflectance spectral bands as inputs. The Pendigits dataset is composed of 10,992 samples with 16 dimensions and 9 classes. The *wdbc* dataset consists of 569 samples, 30 dimensions and 2 classes (Table 5.2). We used 200 randomly selected samples in wdbc dataset and 500 in Pendigits dataset for illustration purposes. The third dataset considers a multispectral image acquired over a residential neighborhood of the city of Zürich by the QuickBird satellite in 2002 and the fourth dataset refers to an image acquired by the DAIS7915 sensor over the city of Pavia (Italy) (see Section 4.2.1).

Analysis of the kernel structure

First, we analyze the structure of both the standard RBF kernel and K_c. The RBF kernel was obtained by fixing the width parameter to the average Euclidean distance between all samples. The probabilistic cluster kernel was generated for a maximum of $G = 20$ clusters with $Q = 20$ realizations, i.e. K_c is an average of 400 kernels. It is worth noting that the selected maximum number of clusters G might be different depending on the number of classes, samples and dimensions of the dataset. Figure 5.8 shows these kernels for the considered datasets. We also include the HSIC (see Section 2.3.4) and the Frobenius norm error of K_c and K_{RBF} with the ideal kernel matrix $\mathbf{K}_{ideal} = \mathbf{Y}\mathbf{Y}^\top$, where $\mathbf{Y}$ stores the binary assignment of the samples to the classes (which are of course unknown for the clustering algorithm). It is worth noting that K_c gives

[2] *http://archive.ics.uci.edu/ml/datasets.html*

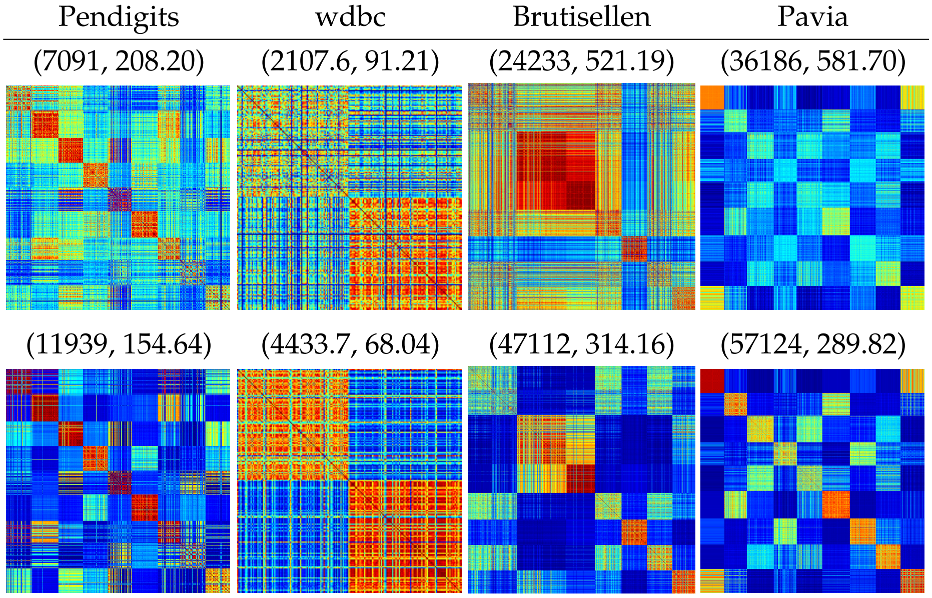


Figure 5.8: RBF kernel (top) and probabilistic cluster kernel (bottom) matrices for different datasets. We show in parenthesis the quality of the similarity measure as (HSIC($\mathbf{K}, \mathbf{YY}^\top$), $\|\mathbf{K} - \mathbf{YY}^\top\|_F$).

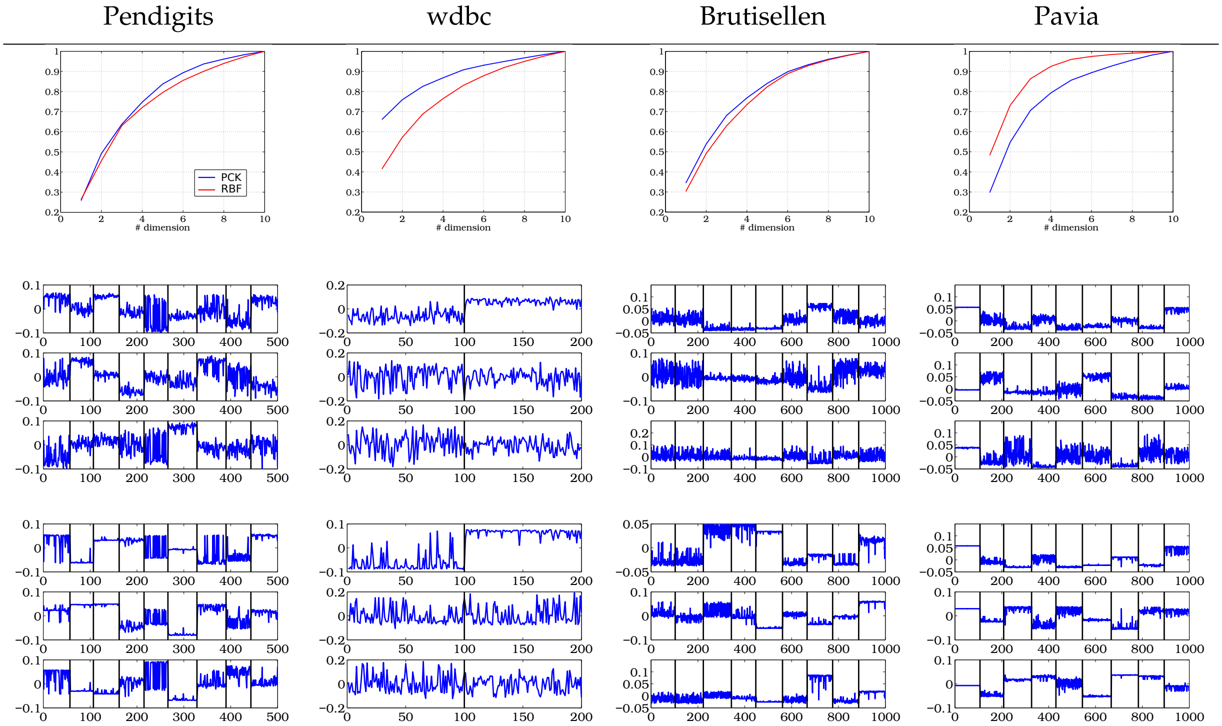


Figure 5.9: Cumulative normalized eigenvalues (top), and the three leading eigenvectors for the RBF (middle) and the proposed K_c (bottom) for the considered datasets obtained in the kernel eigendecomposition.

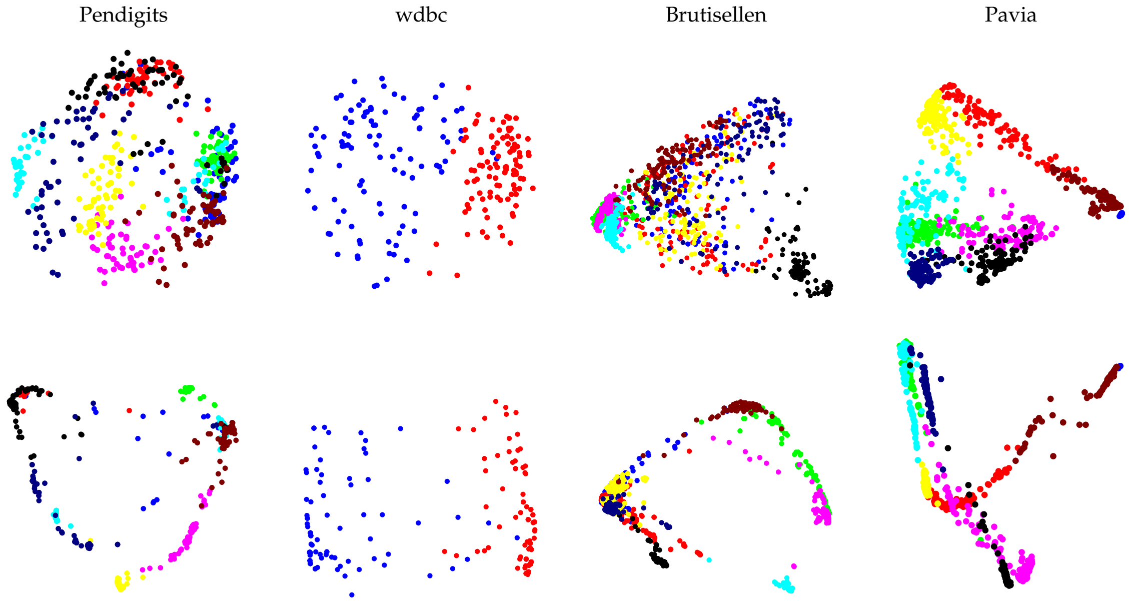


Figure 5.10: Data projected on the two top eigenvectors for the RBF (top) and the proposed K_c (bottom). Different colors indicate classes.

rise to a more blocky structure which is consistent with the numeric quantitative results (higher dependency, lower error).

In order to further analyze the information content of the kernel, we perform the spectral decomposition on the kernel matrices obtained from K_c and the RBF kernel, respectively. The spectral decomposition seeks the directions in the kernel Hilbert space that preserve most of the data variance. Figure 5.9 presents the cumulative normalized eigenvalues obtained with both kernels as well as the first three eigenvectors. On the one hand, we observe that the energy contained in each kernel per dimension is different since the cumulative eigenvalues obtain distinct values. On the other hand, the three top extracted eigenvectors reveal a more clear class structure in the case of the K_c kernel. Figure 5.10 shows the projected data onto the first two components. Results show important differences between the two kernels. Interestingly, the K_c better reveals class structure compared to the RBF kernel. This suggests that the adaptive scale encoded in K_c may be useful for visualization purposes.

So far we showed that the K_c obtains favorable eigenvectors and reveals blockier than the RBF kernel because K_c captures different scales of information along the manifold unlike the RBF. Figure 5.11 shows an illustrative toy example in which are presented the differences between the local properties using four kernels: RBF, PCK, Fisher (Jaakkola et al., 1999) and Jensen-Shannon (Bicego et al., 2013) kernels. The data was generated by the composition of two normal distributions, $\mathcal{N}(3, 0.5)$ and $\mathcal{N}(5, 0.5)$. We look at the structure of the kernel matrix through the first eigenvectors, and compute the Frobenius norm of the residuals with the ideal kernel. The

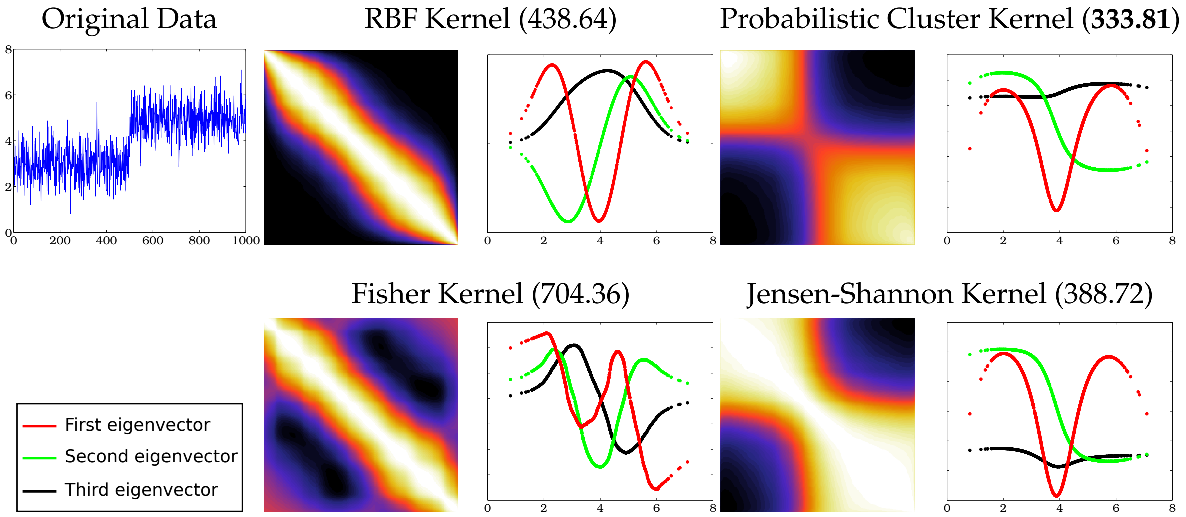


Figure 5.11: Example of differences between the local properties of the RBF, the proposed probabilistic cluster kernel (PCK), Fisher's, and Jensen-Shannon kernels. We indicate in parentheses the Frobenius norm of the residuals with the ideal kernel, $||\mathbf{K}_{ideal} - \mathbf{K}||_F^2$, where $\mathbf{K}_{ideal} = \mathbf{y}\mathbf{y}^\top$.

lengthscale of the RBF was fixed to the average distance of all examples while we used $Q = 10$ and $G = 25$ for K_c. The Fisher kernel here built from feature vectors extracted from a Gaussian mixture model using the same number of clusters as for the PCK, and the Jensen-Shannon kernel was built from divergence Jensen-Shannon obtained from Shannon entropy. Since the Fisher and Jensen-Shannon kernels are not intrinsically multiscale, here we implement a multiscale version of the Fisher and Jensen-Shannon kernels for the sake of a fair comparison. The figure shows that the PCK and the Jensen-Shannon return more discriminative eigenvectors and substantially different from the somewhat Fourier-like basis obtained by the RBF and the Fisher kernel. The PCK better captures the local structure than the Jensen-Shannon kernel. This becomes clear through the visual comparison of the kernel matrices, and is also supported by the Frobenius norm of the differences to the ideal kernel (in parentheses).

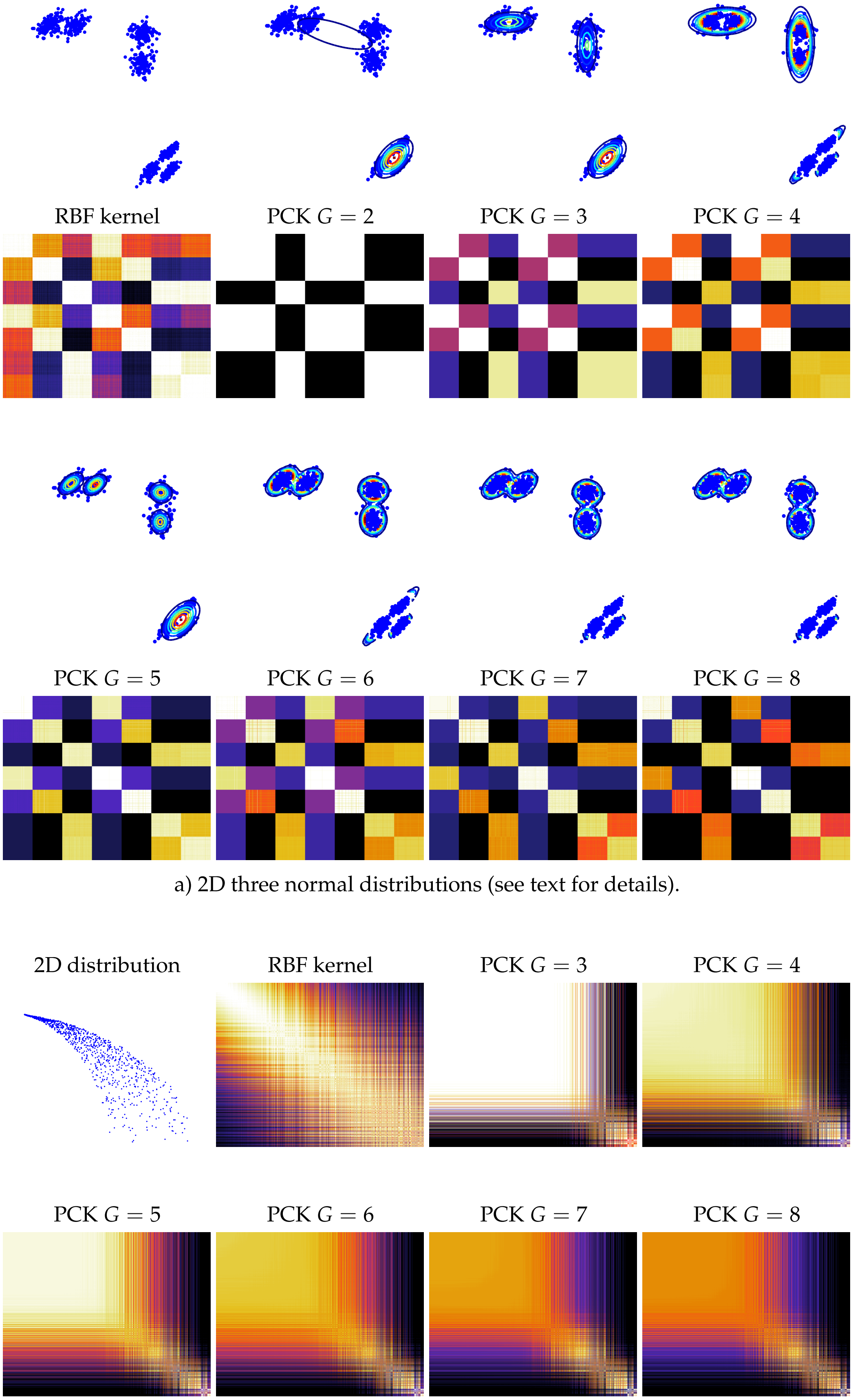
Figure 5.12 presents two additional examples that show how K_c captures different scales of information along the manifold. The first one shows seven 2D normal distributions forming a clear hierarchical cluster structure. Solid lines in the figure represent the pdf contours of the distributions obtained by the EM-GMM for different number of clusters (from 2 to 8). The RBF kernel captures global similarities that only help to distinguish samples from well-separated clusters in terms of Euclidean distance. On the other hand, the PCK correctly captures the data manifold structure (which is given by the cluster distribution) at different scales (which are determined by the selected number of clusters).

The second example is a nonuniform distribution with increasing variance in a curved manifold. The RBF and the PCK Kernels are sorted with respect to the vertical axis of the original data figure. We observe that the RBF kernel is not capable of recognizing the differences of the data density whereas the PCK can identify it, depending on the scale in both examples. Using $G = 4$, the PCK retains better the structure in areas of the manifold with higher density than in areas of lower density. While the PCK built from 2 to 7 clusters keeps the structure of the higher density (structure obtained with $G = 4$), and besides, the PCK manages to obtain the manifold structure in areas with lower density.

Probabilistic cluster kernel clustering

The good properties of the probabilistic cluster kernel are here exploited in spectral clustering. Essentially, a number of features are extracted through the eigendecomposition of the selected kernel and the scores (projections onto top eigenvectors) are then used for canonical k-means clustering. Interestingly, using K_c for clustering involves performing clustering twice: An ensemble of clusterings is used to build the kernel matrix and a second spectral clustering algorithm is used to assign labels to the observations. This may seem counterintuitive but we should stress here that the first clustering operation is not intended to assign centroids but to learn a proper (in our case, probabilistic) similarity metric in the feature space that allow us to extract nonlinear features capturing the manifold structure. We compare clustering results of the standard k-means and EM-GMM algorithms in the input space and the kernel k-means (K_{RBF}) (Girolami, 2002a), bagged kernel (K_{bag}) (Weston et al., 2005), and the proposed probabilistic cluster kernel (K_c) in feature space. Clustering is assessed using six validation indices (Wu et al., 2009): the Overall Accuracy, estimated Cohen's Kappa statistic, entropy, F-measure, cluster's purity, and the Fowlkes and Mallows validation index. All measures are based on the confusion matrix, which in this unsupervised scenario is constructed assigning the clusters to the corresponding most repeated class labels. Average results and confidence intervals are computed for 10 realizations.

Figure 5.13 shows the results of the four datasets. For all the examples, the use of the probabilistic cluster kernel improves the results compared to the RBF kernel. This matches the previous observations on the structure of the kernels and the obtained eigenvectors, hence we can confirm that the probabilistic cluster kernel represents better the cluster structure along the manifold than the RBF kernel.



a) 2D three normal distributions (see text for details).

b) 2D non uniform distribution with changing variance and curved manifold.

Figure 5.12: Example showing a hierarchical cluster structure at differents scales.

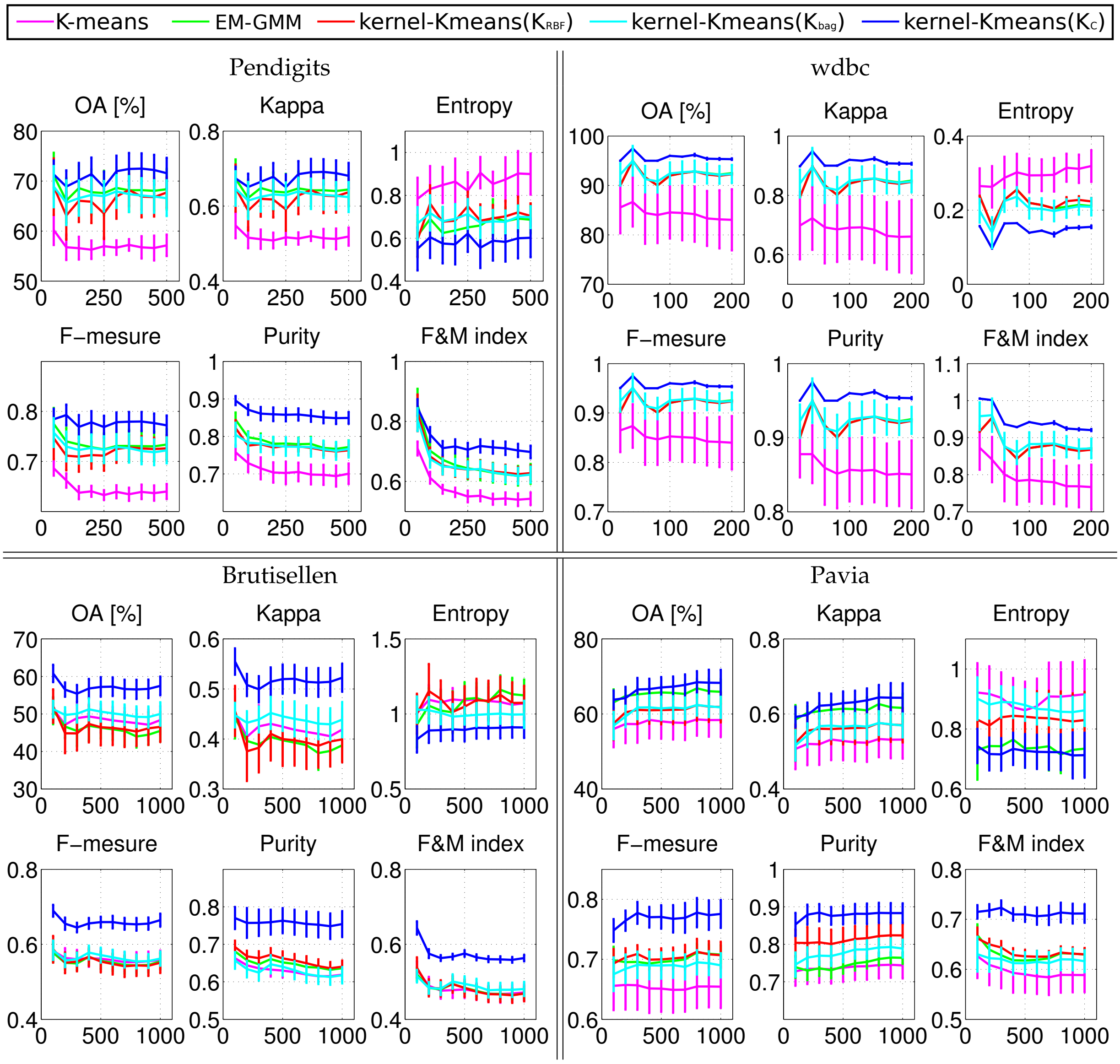


Figure 5.13: Results obtained by clustering validation with six different scores and for a number of test samples in different datasets.

5.4 Summary

In this Chapter, we have focused on unsupervised kernel feature extraction methods, that have roots on information theory and on the cluster assumption.

First, we proposed a simple yet highly efficient modification of the KECA algorithm for optimal extraction of entropic kernel components. While KECA reduces to sort the kernel eigenvectors by entropy, OKECA explicitly searches for the features that retain most informative content. We have illustrated the ability of OKECA to retain more information in pdf estimation and classification in both synthetic and real examples. Results consistently showed that OKECA outperforms KECA in terms of information content and robustness. In fact, in all experiments, a single OKECA feature retained almost all the relevant information. Furthermore we have analyzed the effect of using different unsupervised rules to fit the RBF kernel lengthscale parameter on KECA and OKECA performances. In general, the maximum likelihood approach showed the best performance.

The Chapter also introduced a very simple yet efficient generative cluster kernel that also avoids the RBF kernel bandwidth to be tuned. Comparison to the standard RBF kernel function revealed very good capabilities for data description and adaptation to the local and global structure of the manifold. We studied the spectral decomposition and explored the cluster structure of eigenvalues and eigenvectors. The kernel structure revealed sharper and more blocky, and better aligned with the ideal kernel. After projection onto the kernel eigenvectors, the use of canonical k-means for nonlinear clustering substantially improved the results obtained with other approaches in several synthetic and real examples.

Chapter 6

Semisupervised Kernel Feature Extraction with Generative Cluster Kernels

Contents

In the previous chapters, supervised (Chapter 4) and unsupervised (Chapter 5) methods have been analyzed. Now, we will study the semisupervised methodologies. We will start by introducing the semisupervised learning approach, its main advantages, and how to apply it in feature extraction methods for remote sensing data processing. We will propose the adaptation of feature extraction methods to the semisupervised framework. In particular, we focus on the well-known kernel PLS and OPLS methods that have been described in Chapter 3. Finally, we will finish the chapter with the conclusions.

This Chapter is based on the works:

□ E. Izquierdo-Verdiguier, L. Gómez-Chova, L. Bruzzone and G. Camps-Valls, *"Semisupervised Kernel Feature Extraction for Remote Sensing Image Analysis," IEEE Trans. on Geosc. and Rem. Sens.*, vol. 52 (9), pp. 5567–5578, 2014.

□ E. Izquierdo-Verdiguier, L. Gómez-Chova, L. Bruzzone and G. Camps-Valls, *"Semisupervised nonlinear feature extraction for image classification," 2012 IEEE Inter. Geosc. and Rem. Sens. Symp. (IGARSS)*, pp. 1525–1528, 2012.

□ E. Izquierdo-Verdiguier, J. Arenas-García, S. Muñoz-Romero, L. Gómez-Chova and G. Camps-Valls, *"Semisupervised kernel orthonormalized partial least squares," IEEE Inter. Workshop on Machine Learning for Sig. Process.*, pp. 1–6, 2012.

6.1 Introduction to semisupervised learning

Learning the best feature projections for data representation from a representative set of labeled samples by supervised methods allows to obtain the best decision function for the task at hand, e.g. classification or regression (Gómez-Chova et al., 2008). Nevertheless, supervised methods present problems when the sample data do not cover sufficiently well the manifold, generating an estimated probability that does not represent the model of the true underlying distribution correctly (Camps-Valls and Bruzzone, 2009). On the other hand, unsupervised methods do not present this problem since they do not rely on a limited set of labeled samples. Unfortunately, unsupervised methods have to confront other kinds of problems. The fundamental problem is to estimate a pdf which has likely generated the data distribution so selecting both a plausible model and its parameters is the key problem. This problem is extremely difficult even in moderate dimensional spaces and is even more complicated with high dimensional space and heavy-tailed data distributions.

The learning halfway between unsupervised and supervised is known as *Semisupervised learning* (SSL) (Chapelle et al., 2006). The field consists of using the available supervised information together with the contribution of the unlabeled information to generate a model that pays attention to both the manifold structure and the class specificities. From the probabilistic point of view, SSL approaches typically adopt either discriminative or generative models:

- *Generative models* involve estimating the conditional distribution by means of modeling the class-conditional distribution explicitly, such EM algorithms which have been extensively applied in the context of remote sensing data classification (Maulik and Chakraborty, 2011; Gómez-Chova et al., 2010).

- *Discriminative models* estimate the conditional distribution directly and there is no need to explicitly specify the class-conditional distribution. Within this kind of semisupervised models, we distinguish two subgroups: *Graph-based methods* and *The low density separation* *algorithms* principle. In graph methods, each sample extends its label information to its neighbors until stable state is achieved on the whole dataset. These techniques have been adapted to remote sensing data processing (Camps-Valls et al., 2007) and used in different applications (Gómez-Chova et al., 2008; Camps-Valls et al., 2009). Alternatively, methods implementing *Low density separation*, try to maximize the margin for labeled and unlabeled samples simultaneously, such a Transductive SVM (Vapnik, 1998), which has been applied to remote sensing as well (Bruzzone et al., 2006). A nice classifying discussion about the differences and similarities between transductive and semisupervised learning can be found in Chapelle et al. (2006).

The main difference between these two approaches resides in that transductive algorithms do not provide a function for predicting out-of-the-sample, only predictions for the used unlabeled examples. This raises computational problems. In this thesis, we focus on generative models, which take into account the "cluster assumption" that states that when points in the same cluster are likely to belong to the same class. Several works have been clearly based on this assumption (Weston et al., 2005; Tuia and Camps-Valls, 2009b). We will show some of these works in this section considering that they have been the starting point of our proposal. Furthermore, this section presents the proposed semisupervised kernel feature extraction method. The underlying idea is to construct a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ measuring the similarity among labeled samples, taking into account the distribution of all available pixels, i.e. labeled ℓ and unlabeled u. The constructed kernel has two contributions, one using all available $\ell + u$ samples and the other computed with the ℓ labeled samples. The summation of the kernels is a valid kernel, and can be used in any kernel method for classification or regression, such as the standard support vector machine (SVM). Nevertheless, in this thesis, we plug this kernel into kernel PLS feature extraction to extract a desired number of nonlinear features, which are then used for linear classification and regression. The method is easy to apply and relies on our recent developments presented in Chapter 5.

6.1.1 Bagged kernel for support vector machine

In (Tuia and Camps-Valls, 2009b), the authors exploited the general idea of developing a kernel directly learned from data. The *bagged kernel* (Chapelle et al., 2006) was defined by counting the occurrences of two pixels in the same cluster over several runs of an unsupervised algorithm. The algorithm consists of different steps. First, it computes the standard RBF kernel K_s using labeled samples only (supervised). Second, it runs q times the k-means algorithm (Duda and Hart, 1973) with different initializations but with the same number of clusters g, which results in $q = 1, \dots, Q$ cluster assignments $c_q(\mathbf{x}_i)$ for each sample x_i. Third, we build a bagged kernel K_{bag} based upon the fraction of number of times that $\mathbf{x}_i$ and $\mathbf{x}_j$ are assigned to the same cluster:

$$K_{bag}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{Q} \sum_{q=1}^{Q} [c_q(\mathbf{x}_i) = c_q(\mathbf{x}_j)] \tag{6.1}$$

where $i, j = 1, \ldots, (\ell + u)$ and operator $[c_q(\mathbf{x}_i) = c_q(\mathbf{x}_j)]$ returns '1' if samples $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster according to the Qth realization of the clustering, $c_q(\cdot)$, and '0' otherwise. Finally, train a SVM with the weighted sum (or the product) between the standard and the bagged kernels (Chapelle et al., 2006):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \beta K_s(\mathbf{x}_i, \mathbf{x}_j) + (1 - \beta) K_{bag}(\mathbf{x}_i, \mathbf{x}_j), \tag{6.2}$$

where $i, j = 1, \ldots, \ell$ and the weighting parameter $\beta \in [0, 1]$ provides a trade-off between the supervised and the unsupervised information.

6.1.2 Multiscale bagged kernel support vector machine

The previous kernel implements the *cluster assumption* in the sense that samples that repeatedly fall in the same cluster should belong to the same class. However, this quite intuitive idea should hold *independently* of the scale of the relations we look at. Noting that the notion of similarity can be particularly distinctive at different scales, (Tuia and Camps-Valls, 2011) developed a *multiscale bagged kernel* for urban very high resolution (VHR) images. The kernel of Eq. (6.1) was replaced by a kernel using G clusters of Q runs of the standard k-means. This new averaged kernel accounts for similarities at different scales across the manifold between the pixels. The final kernel is the averaging of the q single-k bagged kernels and encodes multiscale (MS) similarities:

$$K_{bag}^{MS}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{G} \sum_{g=2}^{G+1} K_{bag}^{g}(\mathbf{x}_i, \mathbf{x}_j). \tag{6.3}$$

This kernel was then linearly combined to the standard supervised kernel K_s (Chapelle et al., 2006), as in (Tuia and Camps-Valls, 2009b):

$$K_{C}^{MS}(\mathbf{x}_i, \mathbf{x}_j) = \beta K_s(\mathbf{x}_i, \mathbf{x}_j) + (1 - \beta) K_{bag}^{MS}(\mathbf{x}_i, \mathbf{x}_j). \tag{6.4}$$

6.1.3 Proposed semisupervised kernel feature extraction

The two previous developments share in common the use of the k-means algorithm. Note the similarity of the multiscale bagged kernel and the probabilistic cluster kernel (PCK) proposed in Chapter 5. The bag kernels use k-means clustering while the PCK uses a EM-GMM clustering to decide whether two samples fall into the same cluster. The difference is that while the k-means gives us hard-decisions (the pixels belong or not to the same clustering), the EM-GMM yields a soft-decision by means of membership probabilities. Using hard decisions leads to generating too blocky kernels. This problem motivates our two modifications of the previous algorithms:

1. The probabilistic cluster kernel presented in chapter 5 replaces the bag/cluster kernel. The EM-GMM is a probabilistic model to group the data in different subgroups focused on mixture Gaussian densities. Using the general Bayes' rule, it is possible to obtain the posterior probabilities, $\pi_{i,g}$, of the sample $\mathbf{x}_i$ belonging to cluster g as:

$$\pi_{i,g} = \frac{p(\mathbf{x}_i|g)p(g)}{p(\mathbf{x}_i)}, \tag{6.5}$$

   where $p(g)$ is the prior probability and $p(\mathbf{x}_i|g)$ is the conditional probability of sample $\mathbf{x}_i$ given the cluster g. In the case of GMM, $p(\mathbf{x}_i|g)$ is a linear combination of Gaussian probability functions. The mixture parameters can be estimated by the classical expectation-maximization method, and the maximum posterior probability is computed. The GMM clustering is almost as fast as k-means, but it also provides posterior membership probabilities. By using these probabilities instead of the hard memberships in k-means, smoother kernels are obtained. Including GMM in the construction of cluster kernels leads to the interesting notion of *probabilistic kernel functions* that account for the local structure of the data manifold, whose excellent performance has been assessed in the previous chapter.

2. We replace the standard SVM of (Weston et al., 2005; Tuia and Camps-Valls, 2009b) with any supervised kernel feature extraction algorithm plus linear classification or regression. This has several benefits: i) A supervised kernel feature extraction method allows us to extract nonlinear features maximally aligned with the target variables, ii) it allows us to control the number of features easily, which has a direct impact on the compactness of the solution, and iii) in turn it allows us to describe the data complexity indirectly with the number of needed features to achieve a given level of classification or regression error.

With these two modifications in mind, the proposed *semisupervised kernel feature extraction* will consist of

1. Compute the kernel function using labeled samples:

$$K_s(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}_s(\mathbf{x}_i), \boldsymbol{\phi}_s(\mathbf{x}_j) \rangle, \quad i,j = 1,\ldots,\ell \tag{6.6}$$

2. Build a probabilistic cluster kernel K_c based upon the probability that $\mathbf{x}_i$ and $\mathbf{x}_j$ are assigned to the same cluster:

$$K_c(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{Z} \sum_{q=1}^{Q} \sum_{g=2}^{G+1} \boldsymbol{\pi}_{i,g}(q)^\top \boldsymbol{\pi}_{j,g}(q), \tag{6.7}$$

   where $i,j = 1,\ldots,(\ell+u)$ and Z is a normalized factor (maximum value of K_c).

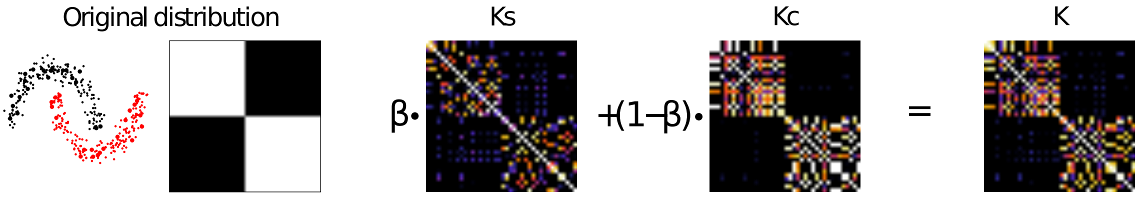


Figure 6.1: Illustration of the different kernels. The original data and the ideal kernel (left). Ks corresponds to the RBF kernel constructed with labeled samples, Kc is the probabilistic cluster kernel constructed with both labeled and unlabeled samples, and K is the final kernel constructed by a linear combination of the previous kernels (right).

3. Define the final kernel function K as the weighted sum (see also properties of kernel methods in Chapter 2) of the standard and the probabilistic cluster kernels (Camps-Valls et al., 2006b):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \beta K_s(\mathbf{x}_i, \mathbf{x}_j) + (1 - \beta) K_c(\mathbf{x}_i, \mathbf{x}_j), \tag{6.8}$$

   where $i, j = 1, \ldots, \ell$ and $\beta \in [0, 1]$ is a scalar parameter.

4. Plug K into the kernel feature extraction method solver (e.g. KPCA, KPLS or KOPLS). The kernel feature extraction method returns the requested number of features d_f, which are used to project data onto them. These (nonlinear) projected data (scores) are then used as inputs to a *linear* classifier or regression method. The application of a *linear* model to the projected data is not incidental: note that all the features are extracted with a nonlinear method so this is the proper scheme to evaluate the effectiveness of the extracted variables.

The probabilistic cluster kernel accounts for *probabilistic* similarities at small and large scales between all labeled samples along the data manifold. Note that finding a proper kernel is equivalent to learn metric relations in the manifold, which are defined through a generative model learned from the data. The PCK kernel generalizes previous approaches based on multiscale cluster kernels. For example, the kernel in Eq. (6.7) reduces to the approach in (Tuia and Camps-Valls, 2011) when only the cluster assignment with maximum posterior probability is considered (hard or crisp clustering). As we have explained before (Section 5.3), the PCK can be related to the family of Fisher's kernels (Jaakkola and Haussler, 1998; Chapelle et al., 2003). Nonetheless, the kernel has the very important advantage that it does not assume an *ad hoc* parametric form or sophisticated priors and thus is more flexible and general.

A toy example of the three kernels involved in the semisupervised proposal is shown in Fig. 6.1 for a two-dimensional binary classification problem. One could think that the probabilistic cluster kernel alone constitute a good enough metric to find better projections. However, this issue strongly depends on the number of both labeled and unlabeled samples. Figure 6.2[left] shows the results in this toy example for a fixed number of labeled samples and varying

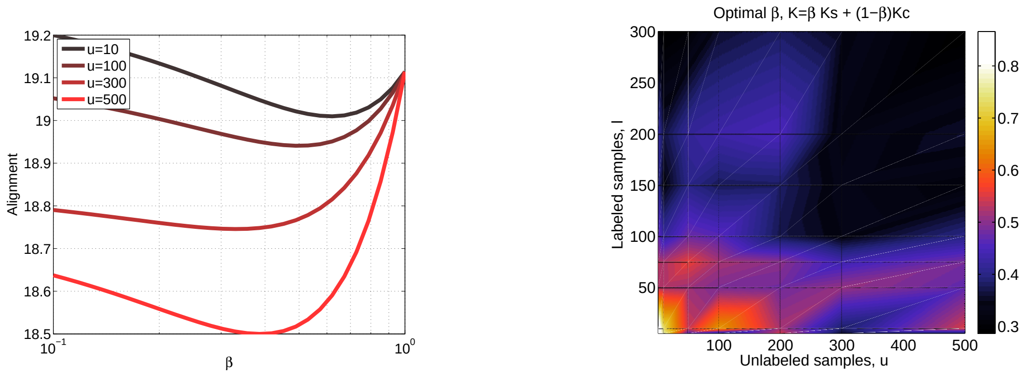
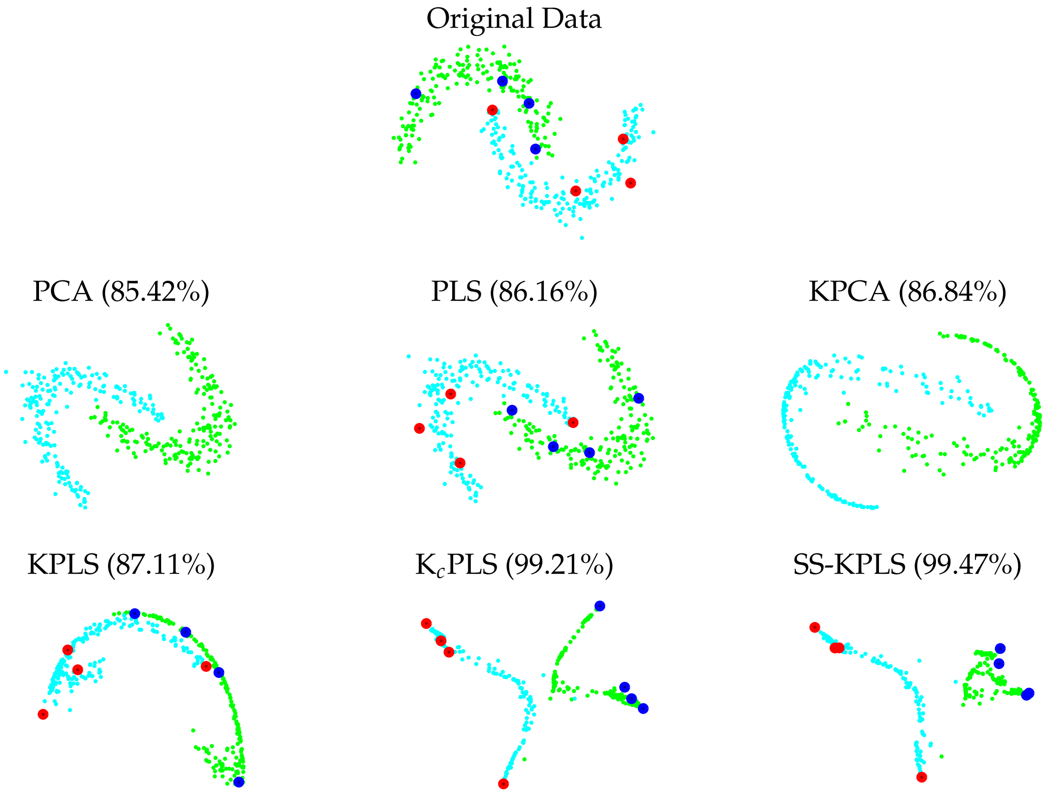


Figure 6.2: Left: The alignment obtained with several values of β (weight of linear combination kernels) for a fixed number of labeled samples $\ell = 100$ and different unlabeled samples, $u = \{10, 100, 300, 500\}$. Right: Surface of optimal values of β for different number of labeled and unlabeled samples.

number of unlabeled samples, u: as u is increased the optimal β becomes lower, and hence the probabilistic cluster kernel becomes relatively more important. Furthermore, we show in Fig. 6.2[right] the surface of optimal β values for different numbers of labeled and unlabeled samples. It is worth noting that the RBF kernel dominates the linear combination (high β values) when few data (less than 100 labeled and less than 200 unlabeled samples) are available, while for many data available, the PCK kernel becomes more important (low β values). This is due to the fact that the PCK kernel is not able to capture well information of the manifold data using low number of samples (labeled and unlabeled) since the clusters obtained by GMM are not representative of the data manifold.

6.2 Semisupervised Kernel Partial Least Squares (SS-KPLS)

This section presents the experimental setup used in the proposed SS-KPLS (Izquierdo-Verdiguier et al., 2014a, 2012b) applied to remote sensing image classification and biophysical parameter retrieval problems. For the classification setting, we show results in three multispectral and hyperspectral images acquired by different sensors and involving the identification of different numbers and types of land cover classes. For the biophysical parameter retrieval, we consider two particularly relevant problems for land and ocean monitoring: the estimation of oceanic chlorophyll concentration, and of chlorophyll, LAI and fPAR for vegetation monitoring. The method is compared against standard linear and nonlinear feature extraction approaches in terms of accuracy and robustness, and expressive power (compactness of the information). Matlab code and demos are available for the interested reader in `http://isp.uv.es/code/sskpls.html`.



Original Data

PCA (85.42%)

PLS (86.16%)

KPCA (86.84%)

KPLS (87.11%)

K$_c$PLS (99.21%)

SS-KPLS (99.47%)

Figure 6.3: Projections extracted by different linear and nonlinear feature extraction methods in a binary problem. We indicate the overall accuracy in the test set for comparison. Note that the SS-KPLS method reduces to KPLS for $\beta = 1$ and K$_c$PLS method for $\beta = 0$.

For all experiments, we used ℓ labeled samples and u unlabeled samples in order to define the $(G \cdot Q)$ cluster centers and the pixel posterior probabilities for each example $\mathbf{x}_i$, i.e. $\boldsymbol{\pi}_i$. In all cases, we used $G = Q = 20$ and the parameters β and σ were optimized by N-fold cross-validation. Given the low number of examples, a common prescription in machine learning is to use a low number of folds; in our case we optimized β and σ with $N = 3$ folds. The parameter β was tuned between $(0, 1)$ in steps of 0.05 and σ was varied between $[0.05, 2] \times s$ (s here represents the mean distance between all labeled data) for each number of extracted features. Once the mixture models are obtained and stored, the posterior probabilities or membership of the samples to each cluster are computed and K_c is constructed following (6.7). The same assignment is used for predicting the output (class membership for classification or estimated output variable for regression) of an unknown test pixel.

The projections in feature space for new data $\mathbf{X}_*$ involve the two operations described in Section 3.3. We used Eq. (3.2) to obtain them. We used this projected data (*scores* in the statistics literature) in a simple linear regression model, $\hat{\mathbf{Y}} = \mathcal{P}(\tilde{\boldsymbol{\Phi}}_*)\mathbf{W}$. The weight vectors are obtained through the normal equations, $\mathbf{W} = \mathcal{P}(\tilde{\boldsymbol{\Phi}}_*)^\dagger\tilde{\mathbf{Y}}$, where † is the Moore-Penrose pseudoinverse. This solution is valid for multioutput regression problems. For the particular case of classification, the linear model is followed by a "winner-takes-all" activation function. We used different quality measures to test model's accuracy. In all cases, the accuracy values were computed over a total of u unlabeled samples for each number of extracted features. For classification, we used the overall accuracy OA[%] and the estimated Cohen's kappa statistic κ. For regression problems, we evaluated the accuracy of the estimations through the RMSE and the MAE; the bias through the ME; and the goodness-of- fit through R.

Figure 6.3 illustrates the features extracted by linear and kernel feature extraction methods in the nonlinear toy classification problem in a two-dimensional space. Linear methods fail in finding good projections since they cannot cope with the nonlinear nature of the data distribution. Kernel methods find nonlinear projections that better separate the data. The solution of KPCA does not allow to linearly separate the data. This is due to the fact that it becomes very difficult to tune the kernel parameter without labeled data, as previously studied in (Braun et al., 2008). Such problem should be alleviated with KPLS but tuning the parameter is hampered by the low number of labeled data. The PCK kernel K_c included in the KPLS method projects the original data such that they become linearly separable. The combination of the supervised and unsupervised kernels in KPLS refines the decision boundaries.

6.2.1 Semisupervised feature extraction for classification

This subsection presents the results obtained by applying the proposed SS-KPLS technique to remote sensing multispectral and hyperspectral image classification. The next subsection details the data used in the experiments. Then, we focus our attention on the accuracy and robustness of the proposed algorithm in terms of the number of extracted features. Finally, we analyze the eigenspectrum, structure, and information content of the derived kernels.

Data

The first image dataset consists of 4 spectral bands acquired on a residential neighborhood of the city of Zürich by the QuickBird satellite in 2002. The portion of the image analyzed has a size of 329 × 347 pixels. The original image has been pansharpened using a Bayesian data fusion method (for more information see (Fasbender et al., 2008)) to attain a spatial resolution of 0.6 m. Nine classes of interest have been defined by photointerpretation. According to the good results obtained in previous studies (Tuia et al., 2010), a total of 18 spatial features extracted using morphological opening and closing (Serra, 1988) have been added to the spectral bands, resulting in a final 22-dimensional vector.

The second image was acquired by the DAIS7915 sensor over the city of Pavia (Italy), and constitutes a challenging 9-class urban classification problem dominated by structural features and relatively high spatial resolution (5-meter pixels). Following previous works on classification of this image, we took into account only 40 spectral bands in the range [0.5, 1.76] μm, and thus skipped thermal and middle infrared bands above 1958 nm.

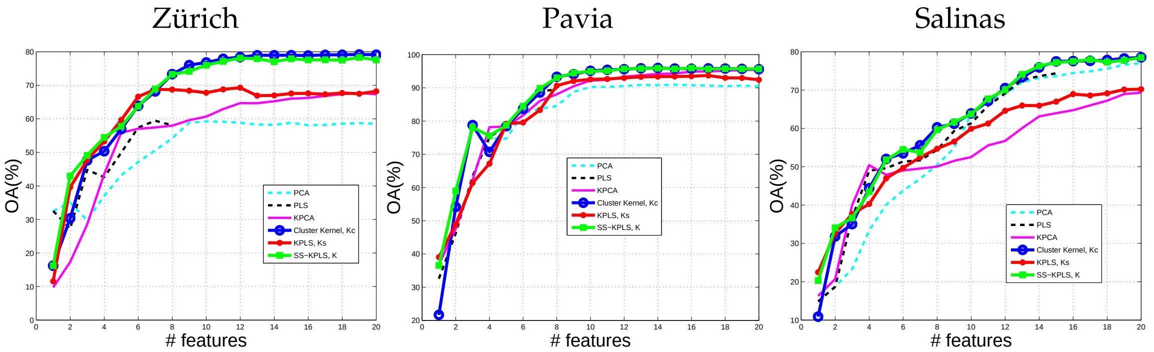


Figure 6.4: Comparison between different feature extraction methods (linear and nonlinear) using the overall accuracy versus the number of extracted features for the Zürich image (left), Pavia image (center), Salinas image (left).

The third image is an AVIRIS hyperspectral image acquired over Salinas valley, an agricultural area of California (USA). A total of 16 crop classes were labeled and 224 spectral bands were used. This is a high-resolution scene with pixels of 3.7 meters. The high number of spectrally similar subclasses makes the classification problem very complex.

Results and discussion

For all experiments, we use ℓ labeled samples *per* class and u unlabeled samples, being $\ell = 10$ and $u = 1710$ for all images. In order to avoid biased results, a total number of 10 realizations is carried out, and the averaged results are shown. We also provide the classification maps and the accuracies obtained in the whole scenes with the optimal parameters and fixing the number of extracted features.

We evaluated the accuracy of several methods for a varying number of extracted features: 1) unsupervised linear, PCA, and its nonlinear version, KPCA; 2) supervised feature extraction algorithms (PLS and its nonlinear version KPLS); and 3) the different kernels involved in SS-KPLS. Note that the proposed SS-KPLS generalizes the standard KPLS (when $\beta = 1$).

Mean and standard deviation accuracies are shown in Fig. 6.4. In general, nonlinear kernel methods (KPCA, KPLS and variants) outperform linear approaches (PCA and PLS). The proposed SS-KPLS improves the results of the standard KPLS and the cluster kernel. The generative cluster kernel yields higher accuracies than the RBF kernel when increasing the number of features. When a higher number of nonlinear features is extracted, all curves become stable but the proposed SS-KPLS clearly outperforms the standard PCA in a range between +5-15%, the more advanced KPCA in a range between +4-15% and KPLS in a range between +3-10%. The behavior of PCA and KPCA in the Zürich and Salinas images should be analyzed because higher accuracy is not obtained with higher number of extracted features, revealing a kind of overfitting problem. This effect has been recently reported in the literature (Braun et al.,

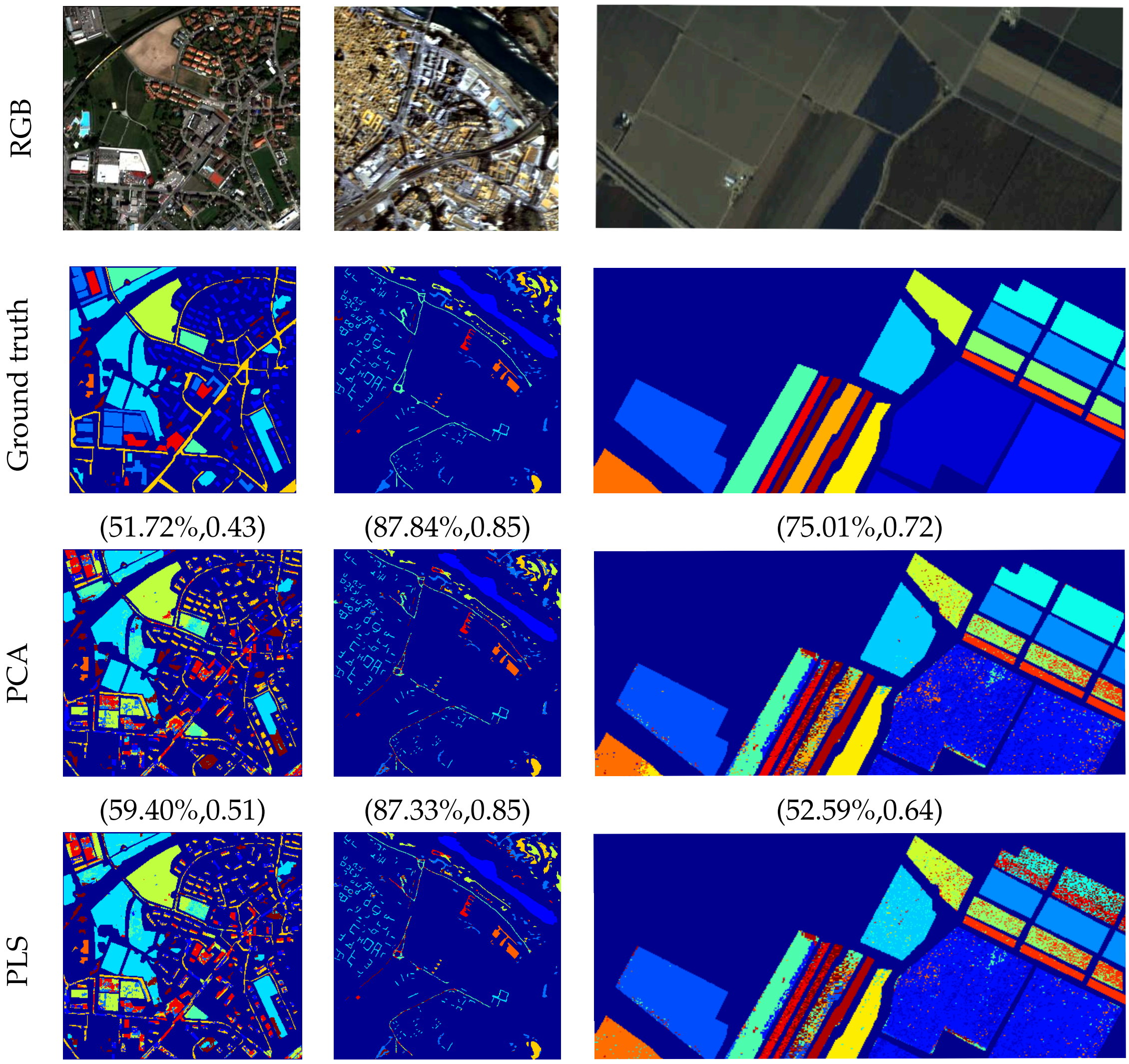


Figure 6.5: Top to bottom: RGB composite, ground truth and three classification maps along with the overall accuracy and kappa statistic for the Zürich image (left) for 11 extracted features, Pavia image (middle) for 16 extracted features and Salinas image (right) for 20 extracted features by linear methods.

2008). This is not the case of the probabilistic cluster kernel K_c. These results are confirmed by the visual inspection of the classification maps shown in Fig. 6.5 (linear methods) and Fig. 6.6 (nonlinear methods), which confirm qualitatively the quantitative results in which the SS-KPLS shows a clear and consistent gain over KPLS of about +7% (Zürich), +3% (Pavia), +13% (Salinas). Nevertheless, in some cases, SS-KPLS does not achieve the K_c accuracy since only with unsupervised information is enough to obtain the higher accuracy.

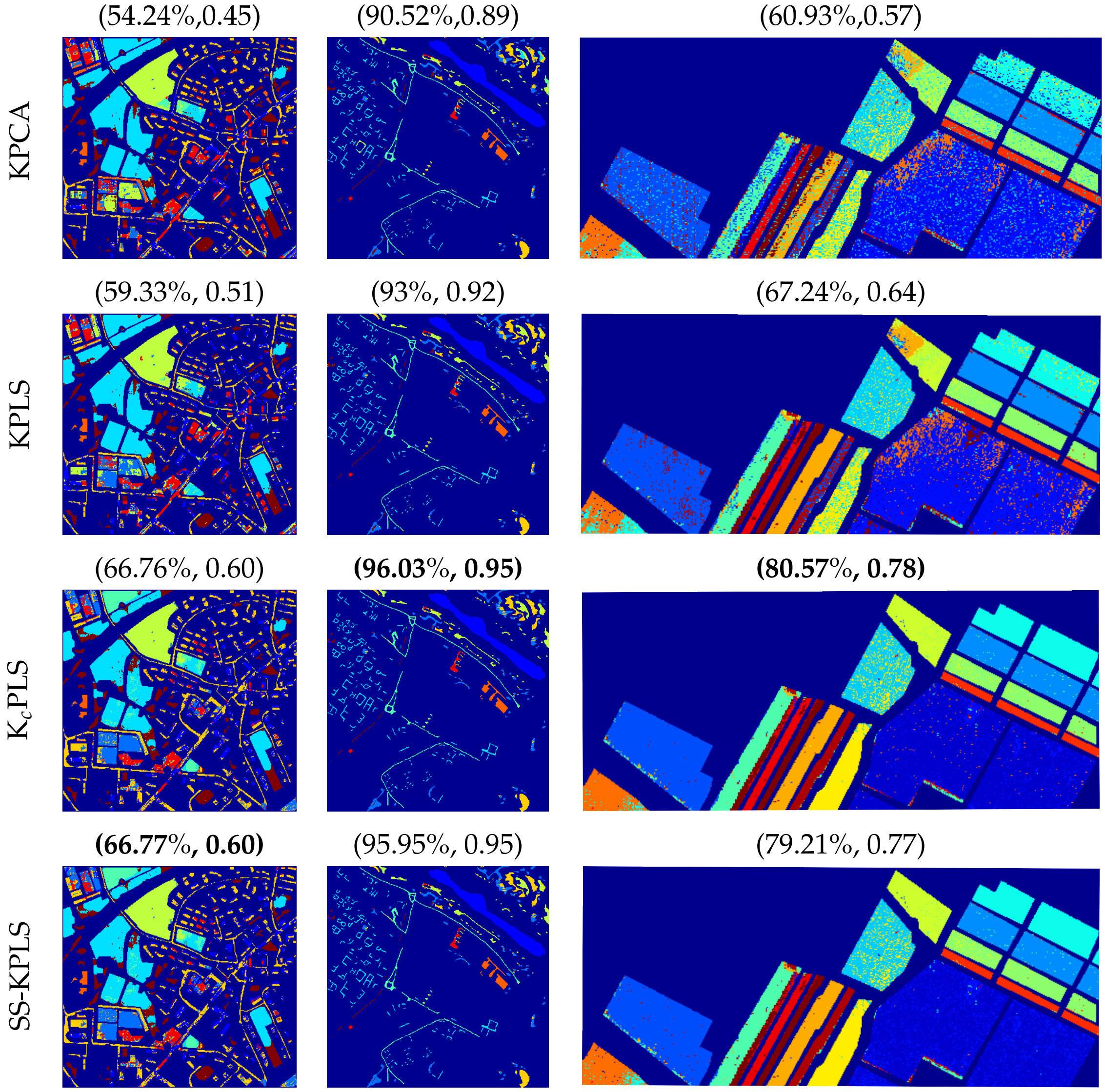


Figure 6.6: Top to bottom: three classification maps along with the overall accuracy and kappa statistic for the Zürich image (left) for 11 extracted features, Pavia image (middle) for 16 extracted features and Salinas image (right) for 20 extracted features by nonlinear methods.

Figure 6.7 shows the false color composite obtained by the data projections with the four, five and six features for the Zürich image. As we see, in PLS and PCA we cannot differentiate the classes. With KPCA and KPLS it is possible to distinguish some classes, as roads, although with KPCA, they are confused with roofs. We show that the features obtained by K_cPLS and SS-KPLS are visually more discriminant than the obtained by the other methods and we are capable to distinguish several classes within the image.

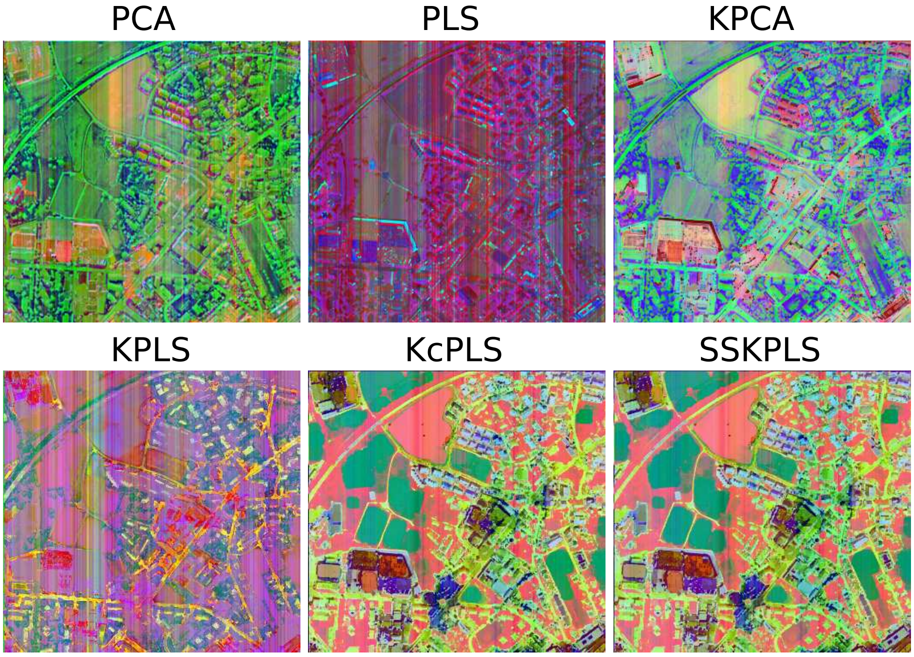


Figure 6.7: False color composite using features 4,5 and 6 for the Zürich image obtained by several feature extraction methods.

6.2.2 Semisupervised feature extraction for biophysical parameter retrieval

We focus now on two challenging problems of biophysical parameter estimation. In particular, we first tackle the estimation of oceanic chlorophyll concentration from multispectral MERIS measurements, and second the retrieval of land-cover biophysical parameters –leaf chlorophyll content (Chl), leaf area index (LAI), and fractional vegetation cover (fCover)– from CHRIS hyperspectral images. In both cases, satellite-derived data and *in situ* measurements are subjected to high levels of uncertainty, as well as collinearity between the input features (channels) and the output target variables. In these difficult scenarios, a proper (robust) feature extraction is necessary, particularly when their relationship is believed to be nonlinear or the target data are scarce thus leading to poorly conditioned problems.

Oceanic chlorophyll concentration

The first dataset simulates data acquired by the Medium Resolution Imaging Spectrometer (MERIS) on board the Envisat satellite (MERIS dataset), and in particular the spectral behavior of chlorophyll concentration in the subsurface waters. We selected the eight channels in the visible range (412-681 nm) to be used for retrieval. The range of variation of chlorophyll concentration in this dataset is $0.02 - 25 mg/m^3$.

Table 6.1: Estimated results for the oceanic chlorophyll concentration retrieval problem as a function of the number of extracted features.

Model	**RMSE**	**MAE**	**|ME|**	**R**
PCA ($d_f = 1$)	0.484	0.385	**0.005**	0.221
PCA ($d_f = 2$)	0.352	0.279	**0.007**	0.704
PCA ($d_f = 3$)	0.294	0.228	**0.006**	0.809
PCA ($d_f = 4$)	0.235	0.170	0.006	0.882
PLS ($d_f = 1$)	0.429	0.339	0.008	0.502
OPLS ($d_f = 1$)	0.153	0.109	**0.005**	0.951
KPCA ($d_f = 1$)	0.486	0.390	0.008	0.194
KPCA ($d_f = 2$)	0.480	0.383	0.015	0.250
KPCA ($d_f = 3$)	0.368	0.292	0.014	0.673
KPCA ($d_f = 4$)	0.363	0.280	0.015	0.682
KPLS ($d_f = 1$)	0.401	0.317	0.022	0.589
KPLS ($d_f = 2$)	0.350	0.278	0.022	0.709
KPLS ($d_f = 3$)	0.339	0.269	0.008	0.730
KPLS ($d_f = 4$)	0.312	0.238	**0.005**	0.785
KOPLS ($d_f = 1$)	**0.143**	**0.066**	0.037	**0.961**
K$_c$PLS ($d_f = 1$)	0.269	0.198	0.029	0.842
K$_c$PLS ($d_f = 2$)	0.233	0.180	0.044	0.890
K$_c$PLS ($d_f = 3$)	**0.225**	**0.169**	0.029	**0.901**
K$_c$PLS ($d_f = 4$)	0.228	0.171	0.021	**0.901**
SS-KPLS ($d_f = 1$)	0.296	0.226	0.030	0.804
SS-KPLS ($d_f = 2$)	**0.232**	**0.176**	0.048	**0.892**
SS-KPLS ($d_f = 3$)	0.248	0.182	0.034	0.873
SS-KPLS ($d_f = 4$)	**0.225**	**0.164**	0.050	**0.901**

In this experiment, we evaluate different quantitative measures of accuracy, bias and goodness-of-fit for a varying number of extracted features. We compare the results obtained by 1) unsupervised linear PCA and its nonlinear kernel version, KPCA; 2) supervised feature extraction algorithms (PLS and its nonlinear version KPLS); and 3) the different kernels involved in SS-KPLS. Table 6.1 shows the obtained results with $\ell = 135$ labeled samples and $u = 865$ unlabeled samples to construct the cluster kernel K_c. The models have been tested with 4000 labeled samples. In general, the nonlinear methods obtain better results than linear approaches. The proposed SS-KPLS reduces the prediction error around 35% with respect to linear PLS and PCA, and KPCA method. In addition, the proposed semisupervised KPLS reduces the error about 25% for a given number of extracted features. Note that, the good results obtained with semisupervised KPLS are mainly due to the cluster kernel function (β values are small) which in many cases yields very high accuracies working alone ($\beta = 0$).

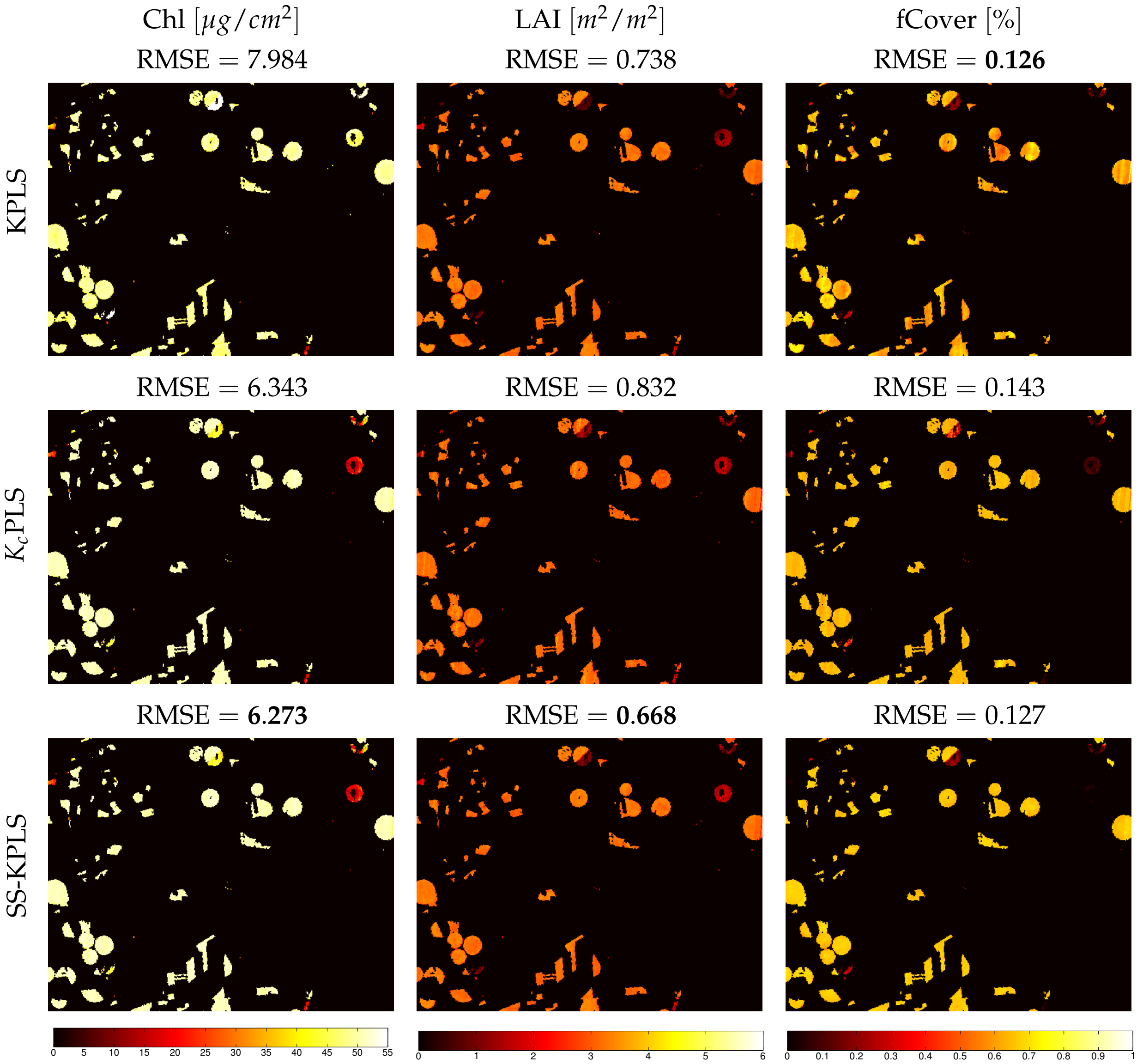


Figure 6.8: Estimation maps for Chl, LAI and FCV, for the KPLS, K_cPLS and SS-KPLS feature extraction methods with the RMSE for the small area of CHRIS/PROBA image with 4 features.

Biophysical parameter retrieval

For the second dataset, we considered data obtained in the SPectra bARrax Campaign (SPARC) in 2003 and 2004 in Barrax, Spain. The test area is an agricultural research facility with an extent of 5 × $10km$. It is characterized by a flat landscape and large uniform land-use units of irrigated and dry lands. The vegetation biophysical parameters were measured among different crops where a large number of samples on an elementary sampling unit (ESU) were taken and averaged for different parameters, obtaining a local characterization of the crops. The Chl was measured with a calibrated Minolta CCM-200 from 50 samples per ESU. The LAI was derived from canopy measurements made with a LiCor LAI-2000 at 24 locations per ESU. The fCover

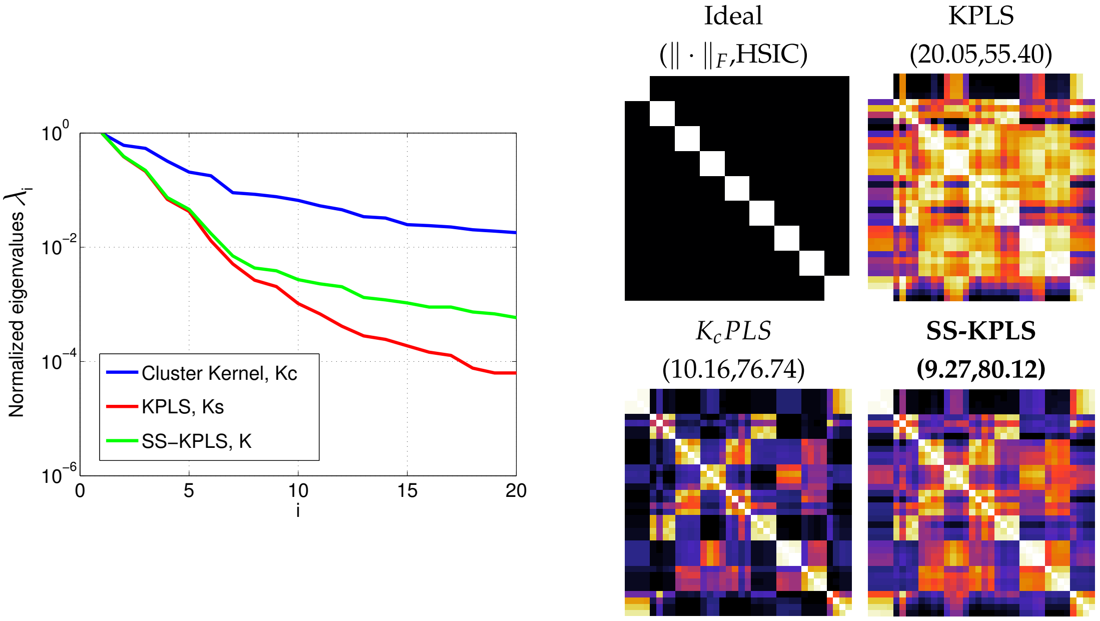


Figure 6.9: Left: Normalized eigenvalues for all kernels used in the Pavia dataset. Right: ideal and used kernels, along quantitative measures of error $\|\cdot\|_F$ and dependence (HSIC).

was derived from hemispherical photographs taken at the same locations as the LAI measurements. All parameters present standard errors between 3% and 10%. For both years, we have a total of nine crop types (garlic, alfalfa, onion, sunflower, corn, potato, sugar beet, vineyard, and wheat), with field-measured values of LAI that vary between 0.4 and 6.3, Chl between 2 and 55 $\mu g/cm^2$, and fCover between 0 and 1. This makes the dataset representative and well-suited to multioutput regression studies. Simultaneously to the ground sampling, hyperspectral images were collected by the CHRIS/PROBA spaceborne sensor. The data provided have 62 bands in the visible and near-infrared (NIR) region ($400 - 1000$ nm) at a spatial resolution of $34m$. The images selected for this experiment were those acquired from the nadir view sharing similar observation configuration in order to minimize angular and atmospheric effects. The images were geometrically and atmospherically corrected using the official CHRIS/PROBA Toolbox for BEAM (Alonso et al., 2009). Finally, the database consists of 135 labeled pixels of Chl, LAI, and fCover measurements and their associated 62 CHRIS reflectance channels. We used $\ell = 30$ and $u = 2437$ pixels to construct the cluster kernel and 105 pixels to test the models.

The obtained maps of vegetation area and RMSE for the three considered biophysical parameters are shown in Fig. 6.8. In the three cases, the use of the kernel combination reports slightly better results. Even if the gain is not very high with regard the standard KPLS approach (about $+2\%$), we should note that 1) the built K_c could be used directly for retrieval without the need of tuning kernel parameters; 2) the probabilistic cluster kernel leads to higher RMSEs than KPLS for Chl and fCover but, since the solutions are complementary, the SS-KPLS benefits from the combination, and 3) the combination makes the final model more robust for LAI as well.

6.2.3 Analysis of the kernels

Figure 6.9 shows the eigenvalues of the best kernels for the Pavia image. The eigendecomposition of the proposed semisupervised kernel K shows a trade off between the RBF and the cluster kernel, as expected. It is worth noting that the eigenvalues of cluster kernel (blue line) show a slower decay because the kernel is indeed quite blocky and sparse. On the other hand, the RBF kernel shows a heavier tail. The introduction of the cluster kernel can be casted as an extra regularization of the RBF kernel. The right plots present the used kernels and their similarity to the ideal one, $\mathbf{K}_{\text{ideal}} = \mathbf{Y}\mathbf{Y}^\top$. Two quantitative measures are given: the Frobenius norm of the difference of these two kernels, $\|\cdot\|_F$, and the HSIC between them (Camps-Valls et al., 2010). The proposed kernel K aligns well with the ideal kernel (lower error, higher dependence), and takes advantage of the sharper structure learned by the PCK.

6.3 Semisupervised Kernel Orthonormal Partial Least Squares (SS-KOPLS)

In the previous section, we proposed the SS-KPLS method. Nevertheless, in the view of the results obtained in supervised kernel feature extraction in chapter 4, we decided to extend the SS-KPLS proposal to a semisupervised kernel orthonormalized partial least squares (SS-KOPLS) algorithm (Izquierdo-Verdiguier et al., 2012a) for nonlinear feature extraction. This is in principle intended to improve results with a reduce number of components, but also to study the impact of unlabeled samples to alleviate the eventual overfitting observed in KOPLS. The method finds projections that minimize the least squares regression error in Hilbert spaces and incorporates the wealth of unlabeled information to deal with low-sized labeled datasets. The method relies on combining a standard RBF kernel using labeled information, and a generative kernel learned by clustering all available data. The structure and information content of the derived kernels will be studied. The effectiveness of the method will be illustrated in classification and biophysical parameter estimation tasks using standard UCI databases and high-dimensional hyperspectral satellite images. We will study performances in terms of expressive power of the extracted nonlinear features.

Figure 6.10 illustrates the projected data by KPLS and KOPLS in a nonlinear toy classification problem in supervised and semisupervised settings. For both methods, the standard supervised approach fails in unfolding the data distributions, mainly because of the low number of labeled samples. On the contrary, the probabilistic cluster kernels K_c offer better projections, and the semisupervised approach optimizes the combination leading to better data separability. This feature is more noticeable in the case of the proposed SS-KOPLS over its SS-KPLS counterpart.

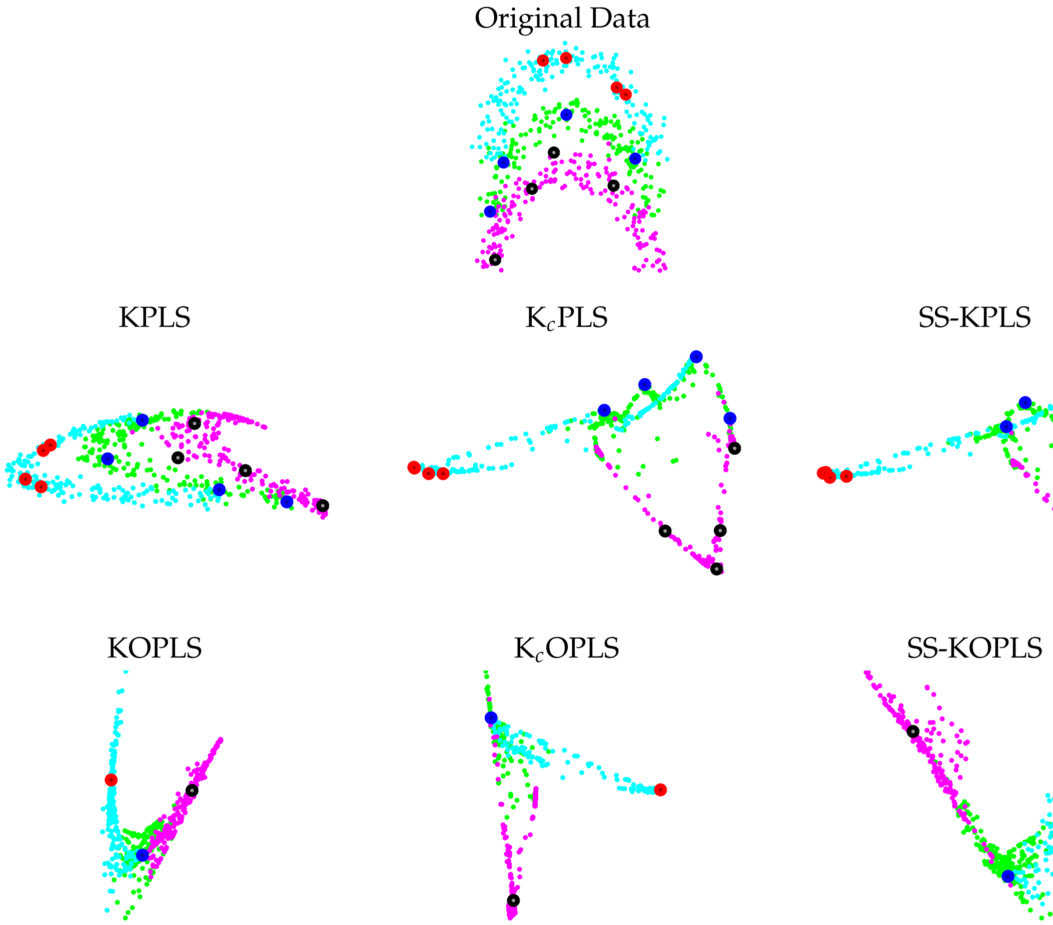


Figure 6.10: Projections extracted by different nonlinear feature extraction methods in a toy problem.

6.3.1 SS-KOPLS for image classification

We apply the SS-KOPLS technique to the six UCI databases[1] (see Table 6.2), a remote sensing multispectral image classification problem. The image was acquired by the DAIS7915 sensor over the city of Pavia (Italy), the same that has been described in the data section of SS-KPLS (Section 6.2.1). And a RGB composite is shown in Fig. 6.11[right].

For our experiments, we compare KPLS and KOPLS families in the previous datasets. We used different number of labeled, ℓ, samples *per* class and unlabeled, u, samples to illustrate the robustness to challenging poorly sampled classification problems (see Table 6.2). In the case of the Pavia image, we only used $\ell = 4$ samples *per* class, and $u = 1710$ samples to define the $(Q \cdot G)$ clusters. In all problems, we used $Q = G = 20$. Once the mixture models are computed and stored, their sample posterior probabilities $\boldsymbol{\pi}_i$ are estimated and K_c is constructed accordingly. The same assignment is used for predicting the class membership of an unknown test sample. For K_s we used in all experiments an RBF kernel of width σ. A 3-fold cross-validation

[1] http://archive.ics.uci.edu/ml/datasets.html

Table 6.2: UCI database description (d: number of dimensions, n_c: number of class, l: number of labeled samples, and u: number of unlabeled samples).

Database	n	d	n_c	l	u
Ionosphere	351	33	2	10	241
Letter	20,000	16	26	130	4,940
Pendigits	10,992	16	9	45	1,710
Pima-Indians	768	8	2	10	380
Vowel	990	12	10	50	430
wdbc	569	30	2	10	414

Table 6.3: Kappa statistic for different UCI databases.

Database				**KPLS**				**KOPLS**	
	d_f	β	K_s	K_c	K	β	K_s	K_c	K
Ionosphere	1	0.80	0.51	0.53	0.58	0.90	0.55	0.55	0.58
	2	0.80	0.55	0.56	**0.59**	–	–	–	–
Letter	1	0.95	0.03	0.03	0.03	0.60	0.02	0.02	0.01
	10	0.90	0.36	0.31	0.35	0.80	0.32	0.38	0.34
	20	0.50	0.44	0.46	0.46	0.25	0.47	0.52	**0.54**
Pendigits	1	0.05	0.11	0.11	0.11	0.05	0.12	0.12	0.11
	8	0.15	0.78	0.86	0.86	0.25	0.87	**0.92**	**0.92**
	20	0.20	0.85	**0.92**	**0.92**	–	–	–	–
Pima	1	0.70	-0.09	0.26	-0.05	0.85	0.24	0.07	**0.25**
	2	0.70	0.10	0.24	0.12	–	–	–	–
	3	0.75	0.16	0.24	0.19	–	–	–	–
Vowel	1	0.20	0.10	0.09	0.09	0.35	0.11	0.07	0.04
	9	0.75	0.51	0.30	0.50	0.65	**0.58**	0.39	0.57
	20	0.80	0.58	0.34	0.54	–	–	–	–
wdbc	1	0.05	0.84	0.88	0.88	0.05	0.75	**0.89**	**0.89**
	2	0.05	0.75	0.89	0.89	–	–	–	–
	3	0.05	0.75	0.89	0.89	–	–	–	–

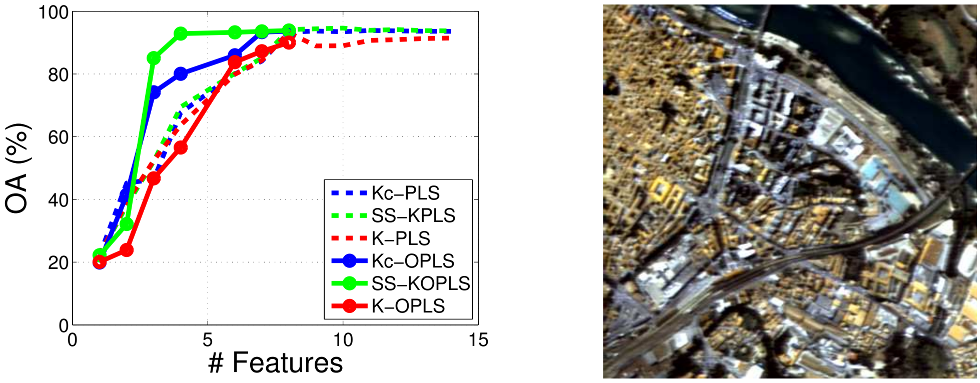


Figure 6.11: Overall accuracy (left) as a function of extracted nonlinear features for the Pavia image (right).

procedure was used to find the optimal σ and β parameters; σ was varied between $[0.5, 2] \times s$, where s represents the median distance between all labeled data, and β between $[0, 1]$ in steps of 0.05 for each number of extracted features. We evaluated the accuracy of KPLS and KOPLS methods for a varying number of extracted features, and for scenarios involving purely supervised, unsupervised and semisupervised feature extraction learning, which is controlled by the value of $\beta \in [0, 1]$. We must remember that KOPLS can extract a maximum number of features given by the rank of $\mathbf{K}_x\tilde{\mathbf{Y}}$, while KPLS is limited by the rank of $\mathbf{K}_x$, which can be low for reduced-sized datasets. Since cross-validation requires splitting the ℓ-samples datasets into smaller subgroups, it may happen that $d_f < \ell$ for KPLS and $d_f < min(n_c, \ell)$ for KOPLS.

Results for the UCI databases are shown in Table 6.3. The proposed semisupervised approach provides the best results in all databases except for *Vowel*, and SS-KOPLS outperforms KPLS in five out of six databases. Actually, inferior yet quite competitive results to KPLS are obtained only for the Ionosphere dataset. In general, the optimal β parameters for SS-KPLS and SS-KOPLS follow similar trends, but no clear dependence is observed with the dimensionality or the number of classes. The average gain of the proposed SS-KOPLS over its counterpart SS-KPLS is about $+5\%$.

Figure 6.11 shows the results for the hyperspectral Pavia image for different number of extracted features. The proposed KOPLS methods generally lead to higher accuracy with lower number of features than KPLS methods, e.g. SS-KOPLS achieves 80% accuracy with three features only, while SS-KPLS needs at least six features. It is worth noting that the cluster kernel K_c-OPLS alone yields very good results for low number of features, and the combination with the supervised kernel gives rise to an improved semisupervised feature extractor.

Table 6.4: Estimated results for the oceanic chlorophyll concentration retrieval problem as a function of the number of extracted features.

Model		KPLS				KOPLS			
	d_f	**RMSE**	**MAE**	**|ME|**	**R**	**RMSE**	**MAE**	**|ME|**	**R**
K_s	1	0.401	0.317	0.022	0.589	0.143	**0.066**	0.037	**0.961**
	2	0.350	0.278	0.022	0.709	–	–	–	–
	3	0.339	0.269	0.008	0.730	–	–	–	–
	4	0.312	0.238	0.005	0.785	–	–	–	–
K_c	1	0.269	0.198	0.029	0.842	0.214	0.157	0.012	0.916
	2	0.233	0.180	0.044	0.890	–	–	–	–
	3	0.225	0.169	0.029	0.901	–	–	–	–
	4	0.228	0.171	0.021	0.901	–	–	–	–
K	1	0.296	0.226	0.030	0.804	**0.141**	0.093	**0.009**	0.959
	2	0.232	0.176	0.048	0.892	–	–	–	–
	3	0.248	0.182	0.034	0.873	–	–	–	–
	4	0.225	0.164	0.050	0.901	–	–	–	–

6.3.2 SS-KOPLS for biophysical parameter retrieval

This subsection evaluates the biophysical parameter retrieval task. We compare the results obtained by KOPLS methods using 1) the standard RBF (K_s), 2) the PCK (K_c) and, 3) semisupervised combination (K) to two different datasets: data acquired by MERIS, (MERIS dataset) and data obtained in SPARC campaign (see Section 6.2.2). In both cases, we used $G = Q = 20$ and the parameters β and σ were optimized by cross-validation, as the SS-KPLS case. We measured the error of estimations (RMSE and MAE), the bias (ME) and the goodness-of- fit (R).

Table 6.4 shows the MERIS dataset results obtained with $\ell = 135$ labeled samples $u = 865$ to construct the K_c kernel and 4000 labeled samples to test the models. The KOPLS methods outperform their respective KPLS methods. Note that the SS-KOPLS method obtains the best results reducing the SS-KPLS error to nearly 50% with lower number of extracted features. In contrast to SS-KPLS, the goods results obtained with semisupervised KOPLS are mainly due to the RBF kernel function ($\beta = 0.95$) and not due to the PCK.

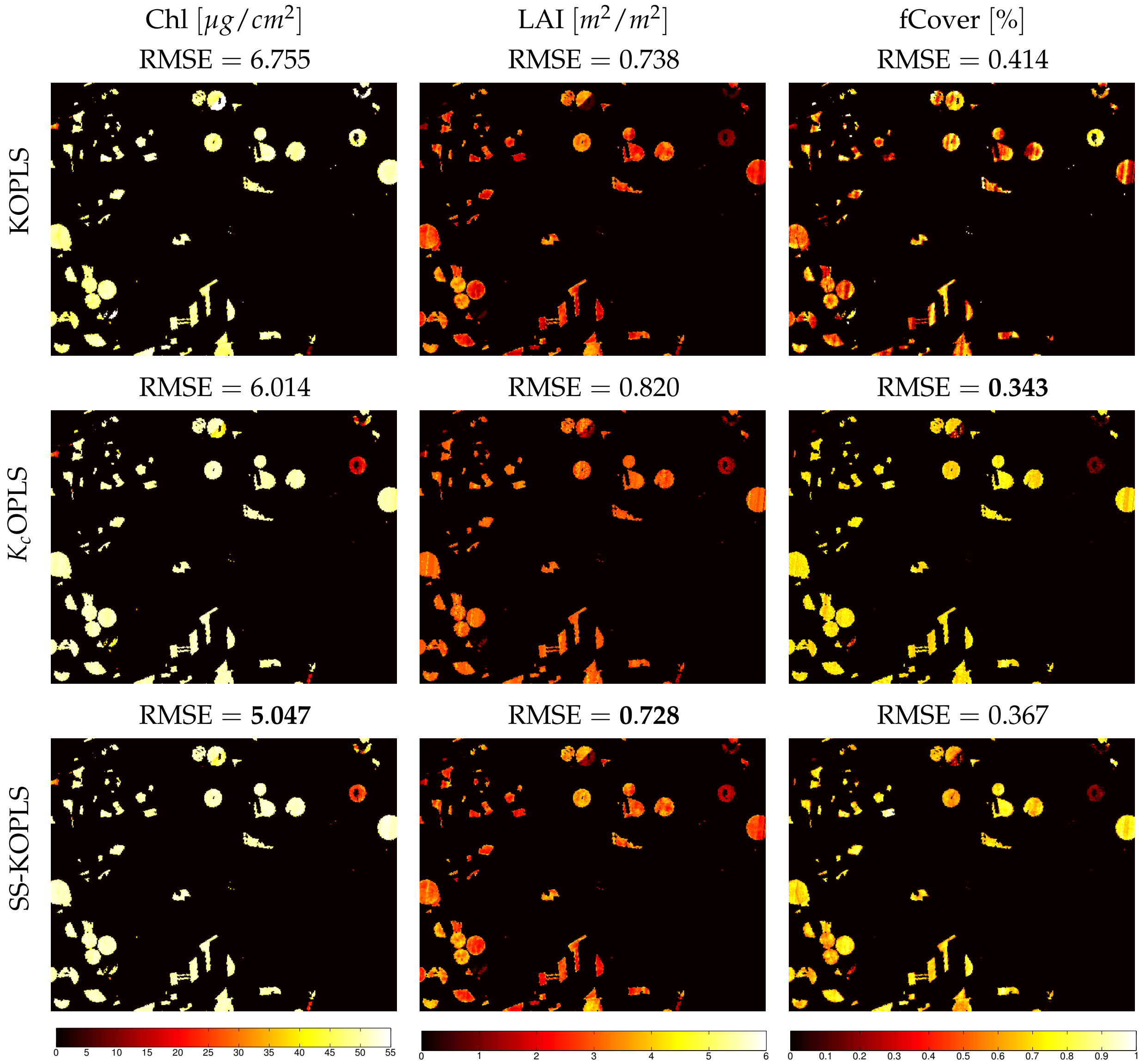


Figure 6.12: Estimation maps for Chl, LAI and FCV, for the KOPLS, K_cOPLS and SS-KOPLS feature extractor methods with the RMSE for the small area of CHRIS/PROBA image with just one feature.

The comparison among the KOPLS methods using the SPARC dataset is shown in figure 6.12. The figure illustrates the estimation maps for Chl, LAI and FVC and the RMSE with 1 feature of vegetation area. We have used the same experimental setup, labeled and unlabeled pixels to build the Probabilistic Cluster Kernel and number of training and test samples to obtain the accuracy in the case of SS-KPLS (6.8). Comparing the RMSE values of different KOPLS methods, the results obtained with semisupervised KOPLS reduce the prediction error around 13% with respect to standard KOPLS and 15% with respect to K_cOPLS. The use of the kernel combination reports slightly better results excluding the fCover results.

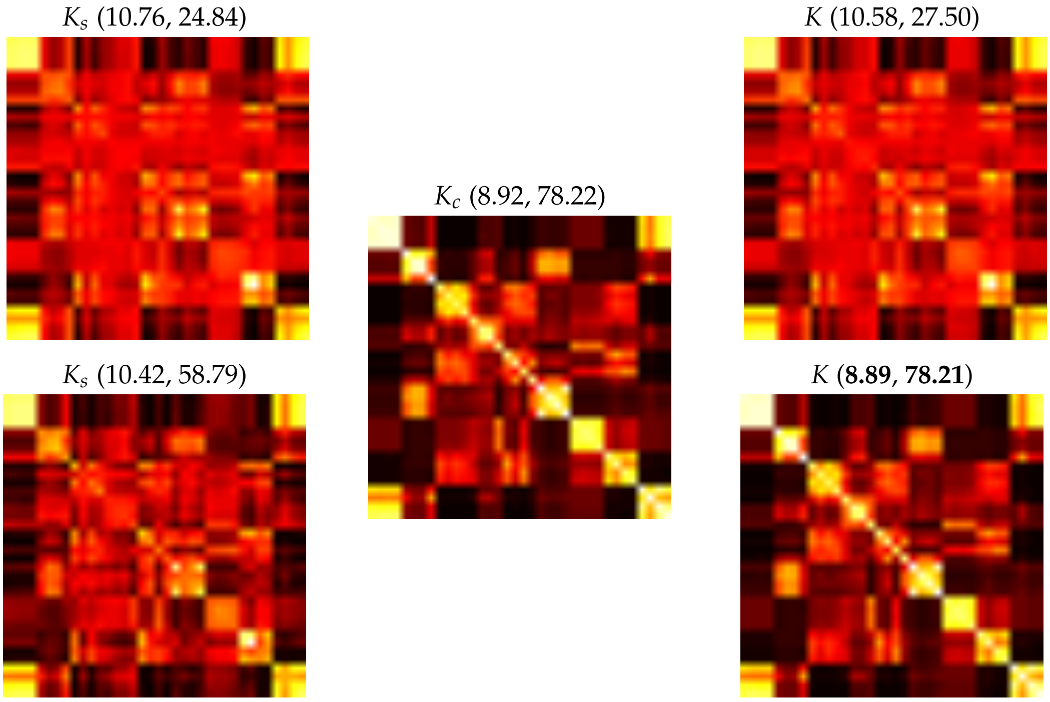


Figure 6.13: Kernel matrices along with quantitative measures of error $\| \cdot \|_F$ and dependence (HSIC) with the ideal kernel used in KPLS (top) and KOPLS (bottom) methods.

6.3.3 Analysis of the kernels

Figure 6.13 shows the used kernels and their similarity to the ideal one, $\mathbf{K}_{\text{ideal}} = \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\top$, for the case of $d_f = 4$, where the maximum difference in accuracy between SS-KOPLS and SS-KPLS appears for the Pavia image. Two quantitative measures are given: the Frobenius norm of the difference, $\| \cdot \|_F$, and the Hilbert-Schmidt Independence Criterion (HSIC) between them (Camps-Valls et al., 2010). The proposed semisupervised KOPLS aligns well with the ideal kernel (lower error, higher dependence), and takes advantage of the sharper structure learned by the cluster kernel.

Finally, note that KOPLS maximizes the covariance between the projected data in $\mathcal{H}$ and the labels. This can be easily shown to be equivalent to maximize statistical dependence with HSIC working with projected data. HSIC corresponds to estimate the norm of the input-output cross-covariance in $\mathcal{H}$, whose empirical (biased) estimator is $\|\mathcal{C}_{xy}\|^2_{\mathcal{H}} = \frac{1}{l^2}\mathrm{Tr}(\mathbf{K}_x\mathbf{K}_y) = \frac{1}{l^2}\mathrm{Tr}(\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Phi}}^\top\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\top)$. Now, by projecting mapped data to a subspace of $\mathcal{H}$, $\tilde{\boldsymbol{\Phi}}' = \tilde{\boldsymbol{\Phi}}\mathbf{U}$, the equivalent *subspace projected* HSIC estimator, $\|\mathcal{C}_{x'y}\|_{\mathcal{H}} = \frac{1}{l^2}\mathrm{Tr}(\tilde{\boldsymbol{\Phi}}'\tilde{\boldsymbol{\Phi}}'^\top\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\top)$, whose maximization reduces to solve the problem in (3.17), but without the orthogonality constraint.

6.4 Summary

This chapter proposed a novel semisupervised kernel feature extraction techniques for remote sensing image classification and retrieval of biophysical parameters. We used a convex combination of two kernels: one dedicated to labeled samples and the other a multiscale probabilistic cluster kernel. The kernel can be plugged in any kernel method for feature extraction, and is specifically devised to address problems where the number of training samples available is relatively small. Note that these problems are common in operational applications of remote sensing data processing. In such situations, the combination of labeled and unlabeled samples in a semisupervised framework can significantly improve the representation of data. The main limitation is that the number of unlabeled samples used to estimate the probabilistic cluster structure via the EM-GMM algorithm should be high enough, which is usually the case in remote sensing applications.

The first part of the chapter presented the SS-KPLS method. In classification tasks, SS-KPLS was more influenced by PCK generated from unlabeled samples than RBF kernel using only labeled samples. Even, in many cases, PCK yielded very high accuracies working alone without supervision at all. In regression tasks, the results showed an improvement of SS-KPLS results for certain values of d_f over those obtained with probabilistic and RBF kernels. Nevertheless, neither SS-KPLS nor K_cPLS have managed to overcome the standard KOPLS results. This made us think of the possibility of extending the semisupervised proposal to KOPLS method. Using SS-KOPLS, good results were obtained on both standard databases, and in hyperspectral image processing. In classification problems, KOPLS methods obtained in general better results than KPLS and their variants. In regression tasks, KOPLS methods were applied to retrieval of oceanic chlorophyll using MERIS database and to estimate Chl, LAI and fCover of SPARC campaign. In both cases, SS-KOPLS obtained the lower RMSE in test data, except for fCover estimation of SPARC in which K_cOPLS method obtained the best prediction value.

Chapter 7

Conclusion and Discussion

This Thesis has presented different kernel feature extraction methods for remote sensing analysis. We have organized the Thesis in three parts: supervised, unsupervised and semisupervised approaches. In the following paragraphs, several remarks and conclusions are drawn from the different developments in the Thesis.

One of the main conclusions is that, in order to choose the most suitable feature extraction method among the available ones, it is very important to first study the main characteristics of the data distribution (linear or nonlinear, Gaussian or not, dimensionality and number of labeled and prior information) to choose the most suitable method. Through toy examples, standard databases and in many real remote sensing classification and regression problems, we have observed that, in general, the KOPLS projections stood out among all methods in terms of accuracy and reduced number of discriminant features. This main conclusion has to be further clarified, as it strongly depends on the amount and characteristics of the labeled data and a priori information.

Supervised kernel feature extraction. We have studied classification and regression problems using real and standard databases to measure the quality of different supervised feature extraction methods. We have compared these methods with classical unsupervised remote sensing methods, such as PCA and KPCA. We have observed that the KOPLS method usually obtains better results than the other methods using a lower number of extracted features, not only in the regression but also in the classification scenario. This improvement of the results, especially in the regression cases, is due to the fact that KOPLS and OPLS find the projections that minimize the Mean Square Error (MSE). Consequently, a much more discriminative projection vectors are extracted by the KOPLS method over unsupervised methods as KPCA, and over other supervised kernel methods optimizing alignment measures such as KPLS. However, we want to rise a concern about the OPLS and KOPLS considering that they may incur in overfitting problems when dealing with low-sized datasets and few classes.

We have studied some alternatives to improve the accuracy and robustness of the models. One of them is to reinforce invariance to factors affecting the performance by means of including artificial examples. We have introduced a simple method to include data invariances in SVM remote sensing image classification and extended this analysis to kernel feature extraction. Good classification accuracy was obtained in general when few labeled samples were available to train the models. Interestingly, despite of containing more support vectors, we can confirm that the obtained classifiers revealed in some cases enhanced sparsity and robustness properties. We consider that the promising results obtained here open a new and interesting research line for the future.

The inclusion of the virtual samples into the kernel nonlinear feature extraction has allowed us to study the performance of the feature extraction methods using virtual samples. We observed that the kernel constructed using virtual samples was decomposed into more independent features than the kernel constructed by standard samples and therefore carried less redundant information. We observed that independence between features does not guarantee an improvement of the classification accuracy. KPCA did not improve the classification, being the method that obtained the most independent features. Finally, an important conclusion is noticeable improvement of KOPLS encoding invariances, which suggests that including prior knowledge in KOPLS may help to reduce the overfitting issue when when few training samples are used.

The observed facts in the analysis of supervised methods and their comparison with the PCA and KPCA methods confirm that, although they both are the most widely used methods in remote sensing, they are not the most appropriate methods when labeled samples are available. In this regard, we evaluated several supervised kernel feature extraction alternatives in different databases and applications, and we showed the benefits of the supervised KPLS and KOPLS methods.

Unsupervised kernel feature extraction. Two approximations to unsupervised learning with kernel feature extraction have been induced in this Thesis: an information theoretic kernel method for pdf estimation, and a generative kernel for clustering. The first proposal was based on KECA, a kernel feature extraction that seeks for projections that maximize the entropy description. An optimization for searching the optimal components was presented called OKECA. While KECA reduces to sort the kernel eigenvectors by entropy, OKECA explicitly searches for the features that retain most informative content. We have illustrated the ability of OKECA to retain more information in pdf estimation and classification on both synthetic and real examples. Results consistently showed that OKECA outperforms KECA in terms of information content and robustness. In fact, in all experiments, a single OKECA feature retains almost all the relevant information. Furthermore we have analyzed the effect of using different unsupervised rules to fit the RBF kernel lengthscale parameter on KECA and OKECA performances. And in general, the maximum likelihood approach showed the best performance.

In order to avoid the unsupervised methods problem of adjusting the kernel parameters, we introduced a very simple yet efficient probabilistic cluster kernel. Comparison to the standard RBF kernel function (and other kernels such as Fisher's and Jensen-Shannon's) revealed very good capabilities for data description and adaptation to the local and global structure of the manifold. We analyzed the spectral decomposition and explored the cluster structure of eigenvalues and eigenvectors. The kernel structure revealed sharper and more blocky, and better aligned with the ideal kernel. Finally, we studied the proposed kernel for nonlinear clustering. The use of canonical k-means substantially improved the results obtained with other approaches in several synthetic and real remote sensing examples.

Semisupervised kernel feature extraction. The last approach combined labeled and unlabeled data. Based on the previous studies and the use of the probabilistic cluster kernel, we proposed novel semisupervised kernel feature extraction techniques. The methods are specifically devised to address problems where the number of training samples available is relatively small. In such situations, the combination of labeled and unlabeled samples in a semisupervised framework can significantly improve the representation of data. The main limitation is that the number of unlabeled samples used to estimate the probabilistic cluster structure via the EM-GMM algorithm should be high enough.

We applied the developed SS-KPLS and SS-KOPLS methods in remote sensing image classification and retrieval of biophysical parameters, obtaining good results on multispectral and hyperspectral datasets. The proposed methods perform better than supervised and unsupervised linear and nonlinear approaches. In both cases SS-KOPLS performed better than SS-KPLS in general but performance depends again on the problem characteristics (mainly on the number of samples and classes). The main advantage of KOPLS over KPLS relates to the fact that fewer (but more informative) features can be extracted, which is the final objective in feature extraction. But this is at the same time an important limitation in problems with few output variables.

Summarizing, different kernel feature extraction algorithms have been developed for the different approaches (unsupervised, supervised and semisupervised). All the proposed algorithms have been tested in several remote sensing scenarios: classification, regression, clustering, pdf estimation or in presence of distorted distributions, obtaining most of the times better results than the classical methods.

Finally, we would like to close the discussion by looking back to the hypotheses raised at the beginning of the Thesis. According to the developments and experimental evidences observed, we can conclude that most of the hypotheses are correct:

1. Remote sensing data live in low-dimensional manifolds that can be learned by kernel feature extraction methods, either looking for a cluster (multiscale) structure or for the shape of the pdf (entropy).
2. Prior knowledge either by invariance encoding or by the information contained by unlabeled samples act as an efficient regularization way and generally improve the results.

7.1 Achievements and relevance

The conclusions of this work have been presented on several conferences and published as research papers on different international journals. The following list summarizes the achievements and relevance directly related to this Thesis:

International journal papers

- *Encoding Invariances in Remote Sensing Image Classification With SVM*, E. Izquierdo-Verdiguier, V. Laparra, L. Gómez-Chova and G. Camps-Valls, IEEE Geos. and Rem. Sens. Letters 10(5): 981-985 (2013). DOI: 10.1109/LGRS.2012.2227297.
- *Semisupervised Kernel Feature Extraction for Remote Sensing Image Analysis*, E. Izquierdo-Verdiguier, L. Gómez-Chova, L. Bruzzone and G. Camps-Valls, IEEE Trans. on Geos. and Rem. Sens. (2014). DOI: 10.1109/TGRS.2013.2290372.
- *Spectral Clustering with the Probabilistic Cluster Kernel*, E. Izquierdo-Verdiguier, R. Jessen, L. Gómez-Chova and G. Camps-Valls, Neourocomputing Letter (Submitted, 2013, 2nd review round).
- *Optimized Kernel Entropy Components*, E. Izquierdo-Verdiguier, V. Laparra, R. Jessen, L. Gómez-Chova and G. Camps-Valls, IEEE Trans. on Neural Networks and Learning Systems (Submitted, 2014).

Conference papers

- *Including invariances in SVM remote sensing image classification*, E. Izquierdo-Verdiguier, V. Laparra, L. Gómez-Chova and G. Camps-Valls, 2012 IEEE Inter. Geos. and Rem. Sens. Symp. (IGARSS): pp. 7353-7356. DOI: 10.1109/IGARSS.2012.6351931.
- *Semisupervised Nonlinear Feature Extraction for Image Classification*, E. Izquierdo-Verdiguier, L. Gómez-Chova, L. Bruzzone and G. Camps-Valls: 2012 IEEE Inter. Geos. and Rem. Sens. Symp. (IGARSS): pp. 1525-1528. DOI: 10.1109/IGARSS.2012.6351244.

- *Semisupervised kernel orthonormalized partial least squares*, E. Izquierdo-Verdiguier, J. Arenas-García, S. Muñoz-Romero, L. Gomez-Chova, and G. Camps-Valls, 2012 IEEE Inter. Workshop on Machine Learning for Signal Processing (MLSP), 2012: 1-6. DOI:10.1109/ MLSP.2012.6349718.

- *Kernel change discriminant analysis for multitemporal cloud masking*, L. Gómez-Chova, E. Izquierdo-Verdiguier, J. Amorós-López, J. Muñoz-Marí and G. Camps-Valls, 2013 IEEE Inter. Geos. and Rem. Sens. Symposium, IGARSS, Melbourne, Australia, July 21-26, 2013: 2974-2977. DOI: .10.1109/IGARSS.2013.6723450.

- *Multiset Kernel CCA for Multitemporal Image Classification*, J. Muñoz-Marí, L. Gómez-Chova, J. Amorós-López, E. Izquierdo-Verdiguier and G. Camps-Valls, 7th International Workshop on the Analysis of Multi-temp. Rem. Sens. Images, Banff, Alberta, Canada (2013).

International book chapters

- *Kernels for remote sensing image classification*, E. Izquierdo-Verdiguier, L. Gómez-Chova and G. Camps-Valls, Wiley Encyclopedia of EEE (Submitted, 2014).

Competitions and awards

The paper *Semisupervised Nonlinear Feature Extraction for Image Classification* was preselected by the *Student Prize Paper Award Committee* to *Student Paper Competition* at IGARSS 2012 in Munich. The paper was presented in a special competition session and awarded with the *Second Student Prize Paper Award*.

Related projects

The outcomes of this work are relevant to the research carried out by the author and her colleagues at the Universitat de València in the context of different research projects in which they are involved. Following a list of the related projects in which the author of this Thesis has participated is provided:

- Spanish Ministry for Education and Science project:
  ′′Metodologías avanzadas en observación de la Tierra: Calibración de datos ópticos y extracción de la información′′ Ministerio de Educación y Ciencia (Spain) (EODIX, AYA2008-05965-C04-04/ESP), [2009-2011].

- ′′Learning image features to encode visual information (LIFE-VISION)′′, Inter-ministerial Commission for Science and Technology (Spain), TIN2012-38102-C03-01 [2013].

- ′′Métodos Avanzados Multitemporales de Extracción y Detección de Nubes en Series Temporales de Satélite′′, Generalitat Valenciana (Spain), (GV/2013/079), [2013].

7.2 Visits to national and international research centers

This thesis was carried out with the collaboration of national and international researchers. During the PhD period, the author had the chance to visit three research centers:

- Four months in the *Remote Sensing Laboratory* at University of Trento (Italy) headed by Prof. Bruzzone. Tasks involved feature extraction methods and image classification.
- Two months in the *Department of Teoría de la Señal y Comunicaciones* at University of Carlos III of Madrid (Spain) under the supervision of Prof. Arenas-García. Tasks involved extension to feature extraction methods, especifically the KOPLS method.
- Three months in the *Department of Physics and Technology* at University of Tromsø (Norway) under supervision of Prof. Jenssen. During this period, the PhD candidate was introduced into Information Theoretic Learning.

7.3 Acknowledgements

This Thesis has been possible thanks to the support of the following grants and projects:

- PhD fellowship: Beca de Formación de Personal Investigador (FPI), Ministerio de Educación y Ciencia, [2009-2013].
- European Space Agency project: 4000101773/10/I-LG. Option "Improvement and integration of cloud screening products, (CPI-13-271).

And finally, the author would like to acknowledge:

- *Prof. Paolo Gamba* for kindly providing the images of University of Pavia and Pavia center.
- *Dr. Tuia* for kindly providing the QuickBird image of a residential neighborhood of Zürich, Switzerland.

Appendix A

Linear Algebra Tools

Our aim here is to present the main matrix factorization fundamentals required to understand better the thesis due to the large number of existing matrix factorization methods.

A.1 Eigenvalue-vector decomposition

Let $\mathbf{A}$ be a square ($N \times N$) matrix, it is possible to compute its spectral decomposition into eigenvalues and eigenvectors ($\mathbf{A} = \mathbf{U}^{-1}\mathbf{\Lambda}\mathbf{U}$) if $\mathbf{A}$ satisfies the linear equation:

$$\mathbf{A}\mathbf{U} = \mathbf{\Lambda}\mathbf{U} \tag{A.1}$$

where $\mathbf{U}$ is the matrix whose columns are the eigenvectors ($\mathbf{u}_i$) of $\mathbf{A}$ and $\mathbf{\Lambda}$ is a diagonal matrix whose principal diagonal is formed by the eigenvalues (λ_i) of $\mathbf{A}$. The equation (A.1) is known as the *standard eigenvalue problem*. If $\mathbf{U}$ is an orthonormal matrix ($\mathbf{U}^\top\mathbf{U} = 1$) then $\mathbf{A}$ is orthogonally diagonalizable: $\mathbf{A} = \mathbf{U}^\top\mathbf{\Lambda}\mathbf{U}$. There exist many ways of solving the equation (A.1) depending on the matrix size and the matrix rank. If the matrix is small, the standard way is by using the *characteristic polynomial*. Since the matrix sizes in the present thesis are not small, we will focus on the approaches to solve the eigenvalue problem with high matrix sizes. The algorithm used to solve the PCA and KPCA problems has been the QR decomposition which is the one by *MATLAB* software.

A.2 Generalized eigenvalue problem

Let $\mathbf{A}$ and $\mathbf{B}$ be two square ($N \times N$) matrices, then it is possible to make their spectral decomposition into eigenvalues and eigenvectors if $\mathbf{A}$ and $\mathbf{B}$ satisfy the *general eigenvectors equation*:

$$\mathbf{A}\mathbf{U} = \mathbf{\Lambda}\mathbf{B}\mathbf{U} \tag{A.2}$$

where $\mathbf{U}$ is the eigenvectors matrix whose columns are the eigenvectors ($\mathbf{u}_i$), and $\mathbf{\Lambda}$ is a diagonal matrix whose principal diagonal is formed by the eigenvalues (λ_i). The equation (A.2) is equivalent to:

$$\begin{aligned} \mathbf{u} = \arg\max_{\mathbf{u}} \quad & \mathbf{u}^\top \mathbf{A}\mathbf{u} \\ \text{subject to: } & \mathbf{u}^\top \mathbf{B}\mathbf{u} = \mathbf{I}, \end{aligned} \tag{A.3}$$

and equivalent to:

$$\begin{aligned} (\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{1/2})\mathbf{w} &= \lambda \mathbf{w} \\ \text{being} \quad \mathbf{u} &= \mathbf{B}^{1/2}\mathbf{w} \end{aligned} \tag{A.4}$$

If matrix **B** is not a full rank matrix, we use *the decomposition theorem* to calculate the eigendecomposition of matrix $\mathbf{B}^{-1}\mathbf{A}$ (Demmel et al., 2000).

A matrix is similar to other if and only if exists a matrix **Q** such that $\mathbf{A}' = \mathbf{Q}^\top \mathbf{A}\mathbf{Q}$. If two matrices ($\mathbf{A}'$ and **A**) are similar then the eigenvalues and eigenvectors are the same for both. Let λ_i be an eigenvalue of **A**, then:

$$\begin{aligned} \left|\mathbf{A}' - \lambda_i \mathbf{I}\right| = 0 \rightarrow \left|\mathbf{Q}^\top \mathbf{A}\mathbf{Q} - \lambda_i \mathbf{I}\right| = 0 \rightarrow \left|\mathbf{Q}^\top \mathbf{A}\mathbf{Q} - \lambda_i \mathbf{Q}^\top \mathbf{I}\mathbf{Q}\right| = 0 \\ \left|\mathbf{Q}^\top\right| |\mathbf{A} - \lambda_i \mathbf{I}| \, |\mathbf{Q}| = 0 \rightarrow \left|\mathbf{Q}^\top\right| |\mathbf{Q}| \, |\mathbf{A} - \lambda_i \mathbf{I}| = 0 \rightarrow |\mathbf{A} - \lambda_i \mathbf{I}| = 0 \end{aligned} \tag{A.5}$$

Therefore, in the case of the *general eigenvectors equation*, we can use a similar matrix to convert it into a *standard eigenvalue problem* without problems using the inverse of **B**. Let $\mathbf{B}' = \mathbf{U}^\top \mathbf{B}\mathbf{U}$ be a similar matrix of **B**, where **U** is the eigenvectors matrix of **B**. And let $\mathbf{A}' = \mathbf{U}^\top \mathbf{A}\mathbf{U}$ be a similar matrix of **A**. It is fulfilled that:

$$\mathbf{B}'^{-1}\mathbf{A}' = (\mathbf{U}^\top \mathbf{B}\mathbf{U})^{-1}(\mathbf{U}^\top \mathbf{A}\mathbf{U}) = \mathbf{U}^{-1}\mathbf{B}^{-1}(\mathbf{U}^\top)^{-1}\mathbf{U}^\top \mathbf{A}\mathbf{U} = \mathbf{U}^\top \mathbf{B}^{-1}\mathbf{U}\mathbf{U}^\top \mathbf{A}\mathbf{U} = \mathbf{U}^\top \mathbf{B}^{-1}\mathbf{A}\mathbf{U},$$

that is, the eigenvalues and eigenvectors of $\mathbf{B}^{-1}\mathbf{A}$ are the same than the eigenvalues and eigenvectors of $\mathbf{B}'^{-1}\mathbf{A}'$. The last expression does not raise any problem with the inverse of $\mathbf{B}'$ since it is a non-singular matrix.

A.3 Singular value decomposition

Let **A** be a matrix ($N \times M$), the *Singular Value Decomposition* (SVD) is a factorization of the matrix in singular values and singular vectors that consists on: $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}$ where $\mathbf{U} \in \mathbb{R}^{N\times N}$ is made up of unit eigenvectors associated with non-zero eigenvalues of $\mathbf{A}\mathbf{A}^\top$, $\mathbf{V} \in \mathbb{R}^{M\times M}$ is made up of unit eigenvectors associated with non-zero eigenvalues of $\mathbf{A}^\top\mathbf{A}$, and $\boldsymbol{\Sigma}$ is a diagonal matrix that contains the Singular Values of **A** sorted in descending order. The Singular Values are the square root eigenvalues of $\mathbf{A}\mathbf{A}^\top$. We can express also the SVD in vectorial from: $\mathbf{A} = \sum_{i=1}^{d_f} \lambda_i \mathbf{u}_i \mathbf{v}_i^\top$.

A.4 Deflation

One way of solving either the *standard eigenvalue problem*, the *Generalized eigenvalue problem* or the SVD, is using iterative methods. These methods consist in approximating successively the solution by means of random initializations. There exist different iterative methods but here we focus on the deflation method. The deflation is a transformation of a symmetric matrix by means of extracting the information that the matrix contains itself about the corresponding eigenvector. The deflation consists of:

$$\mathbf{A} \leftarrow \mathbf{A} - \lambda \mathbf{u}\mathbf{u}^{\top} \tag{A.6}$$

If we apply the eigenvector over the deflation matrix: $\mathbf{Au} = \mathbf{Au} - \lambda\mathbf{u}\mathbf{u}^{\top}\mathbf{u} = \mathbf{Au} - \mathbf{Au}\mathbf{u}^{\top}\mathbf{u} = \mathbf{Au} - \mathbf{Au} = 0$ since $\mathbf{u}$ is a normalized vector. Therefore, the deflation is a transformation that keeps the eigenvector of the matrix while reduces the corresponding eigenvalue to zero without modifying the remainder.

Let $\mathbf{u}_1$ and $\mathbf{u}_2$ be eigenvectors of the matrix $\mathbf{A}$ associated to eigenvalues λ_1 and λ_2, then:

$$\begin{aligned} \mathbf{A}\mathbf{u}_1 &= \lambda_1 \mathbf{u}_1 \\ \mathbf{A}\mathbf{u}_2 &= \lambda_2 \mathbf{u}_2 \end{aligned} \tag{A.7}$$

as $\mathbf{u}_1$ and $\mathbf{u}_2$ have associated different eigenvalues. According to the eigenvalues properties, $\mathbf{u}_1$ and $\mathbf{u}_2$ are orthogonal. Repeating this procedure, we find the largest eigenvalue (λ_1) of $\mathbf{A}$.

Let $\mathbf{B} = \mathbf{A} - \lambda_1 \mathbf{u}_1 \mathbf{u}_1^{\top}$ be the matrix deflation of matrix $\mathbf{A}$. Then,

$$\begin{aligned} \mathbf{B}\mathbf{u}_2 = \lambda_2\mathbf{u}_2 \rightarrow (\mathbf{A} - \lambda_1\mathbf{u}_1\mathbf{u}_1^{\top})\mathbf{u}_2 &= \lambda_2\mathbf{u}_2 \\ \mathbf{A}\mathbf{u}_2 - \lambda_1\mathbf{u}_1\mathbf{u}_1^{\top}\mathbf{u}_2 &= \lambda_2\mathbf{u}_2 \\ \text{as: } \mathbf{u}_1^{\top}\mathbf{u}_2 = 0 \rightarrow \mathbf{A}\mathbf{u}_2 &= \lambda_2\mathbf{u}_2 \end{aligned} \tag{A.8}$$

i.e., the second eigenvalue of $\mathbf{A}$ (λ_2) is also an eigenvalue of $\mathbf{B}$ in spite of we make the deflation transform.

In the case of PCA and KPCA methods, both methods need to solve the *standard eigenvalue problem*, and the deflation transform is:

$$\begin{aligned} &\text{PCA: } \mathbf{C}_{xx} \leftarrow \mathbf{C}_{xx} - \lambda_i\mathbf{u}_i\mathbf{u}_i^{\top}, \qquad i = 1, \ldots, d_f \\ &\text{KPCA: } \mathbf{K}_{xx} \leftarrow \mathbf{K}_{xx} - \lambda_i\mathbf{u}_i\mathbf{u}_i^{\top}, \qquad i = 1, \ldots, n \end{aligned} \tag{A.9}$$

where d_f is number of dimensions and n is number of samples.

When we find a *Generalized eigenvalue problem*, we know that matrix ($\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{1/2}$) (Eq. (A.4)) can be decomposed in eigenvalues and eigenvectors as:

$$(\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{1/2}) = \sum_{i=1}^{d_f} \lambda_i \mathbf{u}_i\mathbf{u}_i^{\top} = \sum_{i=1}^{d_f} \lambda_i \mathbf{B}^{1/2}\mathbf{w}_i\mathbf{w}_i^{\top}\mathbf{B}^{1/2}$$

so

$$\mathbf{A} = \sum_{i=1}^{d_f} \lambda_i (\mathbf{B}\mathbf{w}_i)(\mathbf{w}_i\mathbf{B})^\top,$$

then, the deflation transform is:

$$\mathbf{A} \leftarrow \mathbf{A} - \lambda_i (\mathbf{B}\mathbf{w}_i)(\mathbf{w}_i\mathbf{B})^\top. \tag{A.10}$$

In the case of the methods that need to solve the *generalized eigenvalue problem* (OPLS and its kernel version approach), the deflation transform is:

$$\begin{aligned} &\text{OPLS:} \quad \mathbf{A} = \mathbf{C}_{xy}\mathbf{C}_{xy}^\top \quad \text{and} \quad \mathbf{B} = \mathbf{C}_{xx} \quad \text{then,} \\ &\mathbf{C}_{xy}\mathbf{C}_{xy}^\top \leftarrow \mathbf{C}_{xy}\mathbf{C}_{xy}^\top - \lambda_i \mathbf{C}_{xx}\mathbf{u}_i\mathbf{u}_i^\top \mathbf{C}_{xx}^\top \\ &\text{KOPLS:} \quad \mathbf{A} = \tilde{\mathbf{K}}_x\tilde{\mathbf{K}}_y\tilde{\mathbf{K}}_x \quad \text{and} \quad \mathbf{B} = \tilde{\mathbf{K}}_x\tilde{\mathbf{K}}_x \quad \text{then,} \\ &\tilde{\mathbf{K}}_x\tilde{\mathbf{K}}_y\tilde{\mathbf{K}}_x \leftarrow \tilde{\mathbf{K}}_x\tilde{\mathbf{K}}_y\tilde{\mathbf{K}}_x - \lambda_i \tilde{\mathbf{K}}_x\tilde{\mathbf{K}}_x\mathbf{u}_i\mathbf{u}_i^\top \tilde{\mathbf{K}}_x\tilde{\mathbf{K}}_x. \end{aligned} \tag{A.11}$$

A.5 Iterative power method

Alternatively, the eigenvectors and eigenvalues can be obtained using the iterative power method. This method a finds convergent sequence by means of we can solve the *eigendecomposition problem* or SVD equation.

Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a diagonalizable matrix and its eigenvalues $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$. Let $\mathbf{B} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^n$ a vectors bases that are the eigenvectors associated to $|\lambda_1|, |\lambda_2|, \ldots, |\lambda_n|$. It verifies that $\mathbf{A}^2\mathbf{x}_i = \mathbf{A}(\mathbf{A}\mathbf{x}_i) = \mathbf{A}\lambda_i\mathbf{x}_i = \lambda_i\mathbf{A}\mathbf{x}_i = \lambda_i\lambda_i\mathbf{x}_i = \lambda_i^2\mathbf{x}_i$ and it is easy to proof that $\mathbf{A}^k\mathbf{x}_i = \lambda^k\mathbf{x}_i$.

Proof. Select a random vector $\mathbf{z} \in \mathbb{R}^n$ defined as

$$\mathbf{z}_n = \mathbf{A}\mathbf{z}_{n-1} = \cdots = \mathbf{A}^n\mathbf{z}_0. \tag{A.12}$$

If the coordinates of $\mathbf{z}_0$ into the base $\mathbf{B}$ are $(\alpha_1, \ldots, \alpha_n)$, we can define $\mathbf{z}_0 = \alpha_1\mathbf{x}_1 + \cdots + \alpha_n\mathbf{x}_n$, then:

$$\begin{aligned} \mathbf{z}_n = \mathbf{A}^k\mathbf{z}_0 = \mathbf{A}^k(\alpha_1\mathbf{x}_1 + \cdots + \alpha_n\mathbf{x}_n) = \lambda_1^k\alpha_1\mathbf{x}_1 + \cdots + \lambda_n^k\alpha_n\mathbf{x}_n = \\ \lambda_1^k\left(\alpha_1\mathbf{x}_1 + \cdots + \frac{\lambda_n^k}{\lambda_1^k}\alpha_n\mathbf{x}_n\right) = \lambda_1^k\left(\alpha_1\mathbf{x}_1 + \sum_{i=1}^{n}(\frac{\lambda_i}{\lambda_1})^k\alpha_i\mathbf{x}_i\right) \end{aligned} \tag{A.13}$$

As $\lambda_1 \gg \lambda_i$ is fulfilled that $\lim\limits_{k\to+\infty}(\frac{\lambda_i}{\lambda_1})^k = 0$, and then eq. (A.13) is equivalent to $\lim\limits_{k\to+\infty}(\frac{\mathbf{z}_k}{\lambda_1^k}) = \alpha_1\mathbf{x}_1$. If k is very low, equation A.12 reduces to:

$$\mathbf{A}\mathbf{z}_k = \mathbf{z}_{k+1} \approx \lambda_1^{k+1}\alpha_1\mathbf{x}_1 = \lambda_1(\lambda_1^k\alpha_1\mathbf{x}_1) = \lambda_1\mathbf{z}_k.$$

Then, we can conclude that $\mathbf{z}$ converges and therefore $\mathbf{z}$ is a random vector which is the first eigenvector associated to λ_1 of $\mathbf{A}$. □

We used the deflation and the iterative power method to obtain the principal components of KPLS method presented in this thesis (Chapter 3) while in the case of PLS the singular value decomposition was used due to higher computational burden.

Bibliography

Alonso, L., Gómez-Chova, L., Moreno, J., Guanter, L., Brockmann, C., Fomferra, N., Quast, R., and Regner, P. (2009). CHRIS/PROBA toolbox for hyperspectral and multiangular data exploitations. In *IEEE Geosc. Rem. Sens. Symp. (IGARSS)*, volume II, pages 202–205.

Amorós-López, J., Gómez-Chova, L., Alonso, L., Guanter, L., Moreno, J., and Camps-Valls, G. (2011). Regularized multiresolution spatial unmixing for ENVISAT/MERIS and landsat/TM image fusion. *IEEE Geosc. and Rem. Sens. Lett.*, 8(5):844–848.

Amorós-López, J., Izquierdo-Verdiguier, E., Gómez-Chova, L., Muñoz-Marí, J., Rodríguez-Barreiro, J., Camps-Valls, G., and Maravilla, J. C. (2011). Land cover classification of VHR airborne images for citrus grove identification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(1):115 – 123.

Arenas-García, J. and Camps-Valls, G. (2008). Efficient kernel orthonormalized PLS for remote sensing applications. *IEEE Trans. Geosc. Rem. Sens.*, 46:2872 –2881.

Arenas-García, J., Petersen, K., Camps-Valls, G., and Hansen, L. (2013). Kernel multivariate analysis framework for supervised subspace learning: A tutorial on linear and kernel multivariate methods. *IEEE Signal Processing Magazine*, 30(4):16–29.

Barnard, E. (2010). Maximum leave-one-out likelihood for kernel density estimation. In *Pattern Recognition Association South Africa, PRASA*.

Bellman, R. E. (1961). *Adaptive control processes - A guided tour*. Princeton University Press, Princeton, New Jersey, U.S.A.

Bezy, J.-L., Delwart, S., Gourmelon, G., Baudin, G., Bessudo, R., and Sontag, H. (1997). Medium-resolution imaging spectrometer (MERIS). *Proc. SPIE*, 2957:31–41.

Bicego, M., Ulaş, A., Castellani, U., Perina, A., Murino, V., Martins, A. F. T., Aguiar, P. M. Q., and Figueiredo, M. A. T. (2013). Combining information theoretic kernels with generative embeddings for classification. *Neurocomput.*, 101:161–169.

Bioucas-Dias, J., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N., and Chanussot, J. (2013). Hyperspectral remote sensing data analysis and future challenges. *Geoscience and Remote Sensing Magazine, IEEE*, 1(2):6–36.

Blackwell, W. (2005). A neural-network technique for the retrieval of atmospheric temperature and moisture profiles from high spectral resolution sounding data. *IEEE Trans. Geosc. Rem. Sens.*, 43(11).

Bolón-Canedo, V., Sánchez-Maroño, N., and Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowledge and Information Systems*, 34(3):483–519.

Brand, M. (2003). Charting a manifold. In *NIPS 15*, pages 961–968. MIT Press.

Braun, M. L., Buhmann, J., and Müller, K.-R. (2008). On relevant dimensions in kernel feature spaces. *Journal of Machine Learning Research*, 9:1875–1908.

Bruzzone, L., Chi, M., and Marconcini, M. (2005). Transductive svms for semisupervised classification of hyperspectral data. In *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International*, volume 1, pages 4 pp.–.

Bruzzone, L., Chi, M., and Marconcini, M. (2006). A novel transductive SVM for semisupervised classification of remote sensing images. *IEEE Trans. Geosc. Rem. Sens.*, 44(11):3363–3373.

Bruzzone, L. and Demir, B. (2014). A review of modern approaches to classification of remote sensing data. In Manakos, I. and Braun, M., editors, *Land Use and Land Cover Mapping in Europe*, volume 18 of *Remote Sensing and Digital Image Processing*, pages 127–143. Springer Netherlands.

Bruzzone, L. and Marconcini, M. (2009). Toward the automatic updating of land-cover maps by a domain-adaptation SVM classifier and a circular validation strategy. *IEEE Trans. on Geosc. Rem. Sens.*, 47(4):1108–1122.

Byrne, G., Crapper, P., and Mayo, K. (1980). Monitoring land-cover change by principal component analysis of multitemporal landsat data. *Remote Sensing of Environment*, 10(3):175 – 184.

Cadima, J. and Jolliffe, I. T. (2009). On relationships between uncentred and column centred principal component analysis. *Pakistan Journal of Statistics*, 25(4):473–503.

Campbell, W., Sturim, D., and Reynolds, D. (2006). Support vector machines using GMM supervectors for speaker verification. *Signal Processing Letters, IEEE*, 13(5):308–311.

Camps-Valls, G. (2009). Machine learning in remote sensing data processing. In *Machine Learning for Signal Processing, 2009. MLSP 2009. IEEE Intr. Workshop on*, pages 1–6.

Camps-Valls, G., Bandos Marsheva, T., and Zhou, D. (2007). Semi-supervised graph-based hyperspectral image classification. *IEEE Trans. Geosci. Rem. Sens.*, 45(10):3044–3054.

Camps-Valls, G. and Bruzzone, L. (2005). Kernel-based methods for hyperspectral image classification. *IEEE Trans. Geosc. Rem. Sens.*, 43(6):1351–1362.

Camps-Valls, G. and Bruzzone, L. (2009). *Kernel Methods for Remote Sensing Data Analysis*. John Wiley and Sons.

Camps-Valls, G., Bruzzone, L., Rojo-Álvarez, J. L., and Melgani, F. (2006a). Robust support vector regression for biophysical variable estimation from remotely sensed images. *IEEE Geos. Rem. Sens. Letters*, 3(3):339–343.

Camps-Valls, G., Gómez-Chova, L., Muñoz-Marí, J., Rojo-Álvarez, J. L., and Martínez-Ramón, M. (2008). Kernel-based framework for multi-temporal and multi-source remote sensing data classification and change detection. *IEEE Trans. Geosc. Rem. Sens.*, 46(6):1822–1835.

Camps-Valls, G., Gómez-Chova, L., Muñoz-Marí, J., Vila-Francés, J., and Calpe-Maravilla, J. (2006b). Composite kernels for hyperspectral image classification. *IEEE Geosc. Rem. Sens. Lett.*, 3(1):93–97.

Camps-Valls, G., Mooij, J., and Schölkopf, B. (2010). Remote sensing feature selection by kernel dependence measures. *IEEE Geosc. Rem. Sens. Letters*, 7(3):587–591.

Camps-Valls, G., Muñoz Marí, J., Gómez-Chova, L., Richter, K., and Calpe-Maravilla, J. (2009). Biophysical parameter estimation with a semisupervised support vector machine. *IEEE Geosc. and Rem. Sens. Lett.*, 6(2):248–252.

Camps-Valls, G., Tuia, D., Bruzzone, L., and Benediktsson, J. A. (2014). Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. *IEEE Signal Processing Magazine*, 31(1):45–54.

Camps-Valls, G., Tuia, D., Gómez-Chova, L., Jimenez, S., and Malo, J. (2011). *Remote Sensing Image Processing*. Synthesis Lectures on Image, Video, and Multimedia Processing. Morgan and Claypool.

Carli, A., Figueiredo, M. A. T., Bicego, M., and Murino, V. (2014). Generative embeddings based on rician mixtures for kernel-based classification. *Neurocomputing*, 123:49–59.

Chang, C. and Lin, C. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

Chapelle, O. and Schölkopf, B. (2002). Incorporating invariances in nonlinear support vector machines. In *NIPS 14, MIT-Press*, pages 609–616.

Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, 1st edition.

Chapelle, O., Weston, J., and Schölkopf, B. (2003). Cluster Kernels for Semi-Supervised Learning. In Becker, editor, *NIPS 2002*, volume 15, pages 585–592, Cambridge, MA, USA. MIT Press.

Chen, G. and Qian, S.-E. (2009). Denoising and dimensionality reduction of hyperspectral imagery using wavelet packets, neighbour shrinking and principal component analysis. *International Journal of Remote Sensing*, 30(18):4889–4895.

Chen, G. and Qian, S.-E. (2011). Denoising of hyperspectral imagery using principal component analysis and wavelet shrinkage. *IEEE Trans. on Geos. Rem. Sens.*, 49(3):973–980.

Chen, S.-Y., Ouyang, Y. C., and Chang, C.-I. (2012). Weighted radial basis function kernels-based support vector machines for multispectral image classification. In *2012 IEEE Inter. Geosc. and Rem. Sens. Symp. (IGARSS)*, pages 4339–4342.

Chen, Y., Nasrabadi, N., and Tran, T. (2013). Hyperspectral image classification via kernel sparse representation. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(1):217–231.

Cheng, M., Pun, C.-M., and Tang, Y. Y. (2011). Nonnegative Class-Specific Entropy Component Analysis with Adaptive Step Search Criterion. *Pattern Analysis and Applications*.

Cheriyadat, A. and Bruce, L. (2003). Why principal component analysis is not an appropriate feature extraction method for hyperspectral data. In *2003 IEEE Intr. Geos. Rem. Sens. Symp., 2003. IGARSS '03. Proceedings.*, volume 6, pages 3420–3422 vol.6.

Chi, M. and Bruzzone, L. (2005). A novel transductive SVM for semisupervised classification of remote sensing images. volume 5982, pages 59820G–59820G–12.

Cho, M. A., Skidmore, A., Corsi, F., van Wieren, S. E., and Sobhan, I. (2007). Estimation of green grass/herb biomass from airborne hyperspectral imagery using spectral indices and partial least squares regression. *Intr. Journal of Applied Earth Observation and Geoinf.*, 9(4):414 – 424.

Congalton, R. and Green, K. (1999). *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*. CRC Press.

Coops, N., Smith, M.-L., Martin, M., and Ollinger, S. (2003). Prediction of eucalypt foliage nitrogen content from satellite-derived hyperspectral data. *IEEE Trans. on Geos. Rem. Sens.*, 41(6):1338–1346.

Cover, T. M. and Thomas, J. A. (2005). *Entropy, Relative Entropy, and Mutual Information*. John Wiley & Sons, Inc.

Crawford, M. M., Ma, L., and Kim, W. (2011). Exploring nonlinear manifold learning for classification of hyperspectral data. In *Optical Remote Sensing*, pages 207–234. Springer.

Dalponte, M., Bruzzone, L., and Gianelle, D. (2012). Tree species classification in the southern alps based on the fusion of very high geometrical resolution multispectral/hyperspectral images and lidar data. *Remote Sensing of Environment*, 123(0):258 – 270.

Darvishzadeh, R., Atzberger, C., Skidmore, A., and Schlerf, M. (2011). Mapping grassland leaf area index with airborne hyperspectral imagery: A comparison study of statistical approaches and inversion of radiative transfer models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6):894 – 906.

DeCoste, D. and Schölkopf, B. (2002). Training invariant support vector machines. *Machine Learning*, 46(1–3):161–190.

Demmel, J., Dongarra, J., Ruhe, A., and van der Vorst, H. (2000). *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. Wiley, New York, USA.

Duin, R. P. W. (1976). On the choice of smoothing parameters for Parzen estimators of probability density functions. *IEEE Trans. on Computers*, C-25(11):1175–1179.

Duro, D. C., Franklin, S. E., and Dubé, M. G. (2012). A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using spot-5 HRG imagery. *Remote Sensing of Environment*, 118(0):259 – 272.

Ehlers, M., Klonus, S., Johan Åstrand, P., and Rosso, P. (2010). Multi-sensor image fusion for pansharpening in remote sensing. *International Journal of Image and Data Fusion*, 1(1):25–45.

Fasbender, D., Radoux, J., and Bogaert, P. (2008). Bayesian data fusion for adaptable image pansharpening. *IEEE Trans. Geosc. Rem. Sens.*, 46(6):1847 –1857.

Feilhauer, H., Faude, U., and Schmidtlein, S. (2011). Combining isomap ordination and imaging spectroscopy to map continuous floristic gradients in a heterogeneous landscape. *Remote Sensing of Environment*, 115(10):2513–2524. cited By (since 1996)17.

Filippone, M., Camastra, F., Masulli, F., and Rovetta, S. (2008). A Survey of Kernel and Spectral Methods for Clustering. *Pattern Recognition*, 41(1):176–190.

Finlayson, G., Hordley, S., Lu, C., and Drew, M. (2006). On the removal of shadows from images. *IEEE Trans. Patt. Anal. Mach. Intel.*, 28(1):59–68.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals Eugenics*, 7:179–188.

Friedl, M. and Brodley, C. (1997). Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment*, 61(3):399 – 409.

Fukunaga, K. and Hayes, R. (1989). Effects of sample size in classifier design. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(8):873–885.

Fuyi, T., Mohammed, S., Abdullah, K., Lim, H., and Ishola, K. (2013). A comparison of atmospheric correction techniques for environmental applications. In *2013 IEEE International Conference on Space Science and Communication (IconSpace)*, pages 233–237.

García-Vílchez, F., Muñoz Marí, J., Zortea, M., Blanes, I., González-Ruiz, V., Camps-Valls, G., Plaza, A., and Serra-Sagrista, J. (2011). On the impact of lossy compression on hyperspectral image classification and unmixing. *Geos. Rem. Sens. Letters, IEEE*, 8(2):253–257.

García, M., Riaño, D., Chuvieco, E., Salas, J., and Danson, F. M. (2011). Multispectral and lidar data fusion for fuel type mapping using support vector machine and decision rules. *Remote Sensing of Environment*, 115(6):1369 – 1379.

Girolami, M. (2002a). Mercer Kernel-Based Clustering in Feature Space. *IEEE Trans. Neur. Nets.*, 13(3):780–784.

Girolami, M. (2002b). Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, 14(3):669–688.

Gómez Chova, L., Camps-Valls, G., Bruzzone, L., and Calpe-Maravilla, J. (2008). Semi-supervised remote sensing image classification based on clustering and the kernel mean map. In *2008 IEEE Intr. Geos. Rem. Sens. Symp. (IGARSS)*, volume IV, pages 391–394.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)*. The Johns Hopkins University Press.

Gómez-Chova, L., Camps-Valls, G., Bruzzone, L., and Calpe-Maravilla, J. (2010). Mean map kernel methods for semisupervised cloud classification. *IEEE Trans. Geosc. Rem. Sens.*, 48(1):207–220.

Gómez-Chova, L., Camps-Valls, G., Calpe-Maravilla, J., Guanter, L., and Moreno, J. (2007). Cloud-screening algorithm for ENVISAT/MERIS multispectral images. *IEEE Trans. on Geos. Rem. Sens.*, 45(12):4105–4118.

Gómez-Chova, L., Camps-Valls, G., Muñoz Marí, J., and Calpe, J. (2008). Semisupervised image classification with laplacian support vector machines. *IEEE Geosc. and Rem. Sens. Lett.*, 5(3):336–340.

Gómez-Chova, L., Izquierdo-Verdiguier, E., Amorós-López, J., and Camps-Valls, G. (2013). Kernel change discriminant analysis for multitemporal cloud masking. In *2013 IEEE Intr. Geos. Rem. Sens. Symp. (IGARSS),*, pages 2974–2977.

Gómez-Chova, L., Jenssen, R., and Camps-Valls, G. (2012). Kernel Entropy Component Analysis for Remote Sensing Image Clustering. *IEEE Geosc. Rem. Sens. Letters*, 9(2):312–316.

Gómez-Chova, L., Muñoz Marí, J., Laparra, V., Malo-López, J., and Camps-Valls, G. (2011). A review of kernel methods in remote sensing data analysis. In Prasad, S., Bruce, L. M., and Chanussot, J., editors, *Optical Remote Sensing*, volume 3 of *Augmented Vision and Reality*, pages 171–206. Springer Berlin Heidelberg.

Gómez-Chova, L., Nielsen, A., and Camps-Valls, G. (2011). Explicit signal to noise ratio in reproducing kernel Hilbert spaces. In *IEEE Geosc. Rem. Sens. Symp. (IGARSS)*, pages 3570–3570. IEEE.

Green, A., Berman, M., Switzer, P., and Craig, M. (1988). A transformation for ordering multispectral data in terms of image quality with implications for noise removal. *IEEE Trans. Geosc. Rem. Sens.*, 26(1):65 –74.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-schmidt norms. In *Proceedings of the 16th Intr. Conference on Algorithmic Learning Theory*, ALT'05, pages 63–77, Berlin, Heidelberg. Springer-Verlag.

Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A. (2006). *Feature extraction: foundations and applications*. Springer.

Hansen, P. and Schjoerring, J. (2003). Reflectance measurement of canopy biomass and nitrogen status in wheat crops using normalized difference vegetation indices and partial least squares regression. *Remote Sensing of Environment*, 86(4):542 – 553.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Hu, C.-M. and Tang, P. (2011). Hj-1a/b ccd imagery geometric distortions and precise geometric correction accuracy analysis. In *2011 IEEE Inter. Geosc. and Rem. Sens. Symp. (IGARSS)*, pages 4050–4053.

Hu, Y. D., Pan, J. C., and Tan, X. (2013). High-Dimensional Data Dimension Reduction based on KECA. *Applied Mechanics and Materials*, 303-306.

Huber, P. (1985). Projection pursuit. *Annals of Statistics*, 13(2):435–475.

Hughes, G. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Trans. on Information Theory*, 14(1):55–63.

Hwang, J.-N., Lay, S.-R., and Lippman, A. (1994). Nonparametric multivariate density estimation: a comparative study. *IEEE Trans. on Signal Processing*, 42(10):2795–2810.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons, New York, USA.

Hyvärinen, A. and Oja, E. (2000). Independent Component Analysis: A Tutorial. *Neural Networks*, 13(4-5):411–430.

Izquierdo-Verdiguier, E., Arenas-Garcia, J., Munoz-Romero, S., Gomez-Chova, L., and Camps-Valls, G. (2012a). Semisupervised kernel orthonormalized partial least squares. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, pages 1–6.

Izquierdo-Verdiguier, E., Gómez-Chova, L., Amorós-López, J., and Camps-Valls, G. (2011). Detección automática de plantaciones de árboles de cultivo en imágenes de alta resolución espacial. In *XIV Congreso de la Asociación Española de Teledetección*, pages 13–16, Principado de Asturias, Spain.

Izquierdo-Verdiguier, E., Gomez-Chova, L., Bruzzone, L., and Camps-Valls, G. (2012b). Semisupervised nonlinear feature extraction for image classification. In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pages 1525–1528.

Izquierdo-Verdiguier, E., Gomez-Chova, L., Bruzzone, L., and Camps-Valls, G. (2014a). Semisupervised kernel feature extraction for remote sensing image analysis. *Geoscience and Remote Sensing, IEEE Transactions on*, 52(9):5567–5578.

Izquierdo-Verdiguier, E., Gómez-Chova, L., and Camps-Valls, G. (2015). *Kernels for remote sensing image classification*. Wiley.

Izquierdo-Verdiguier, E., Jenssen, R. Gomez-Chova, L., and Camps-Valls, G. (2013a). Spectral clustering with the probabilistic cluster kernel. *Neurocomputing*.

Izquierdo-Verdiguier, E., Laparra, V., Gomez-Chova, L., and Camps-Valls, G. (2012c). Including invariances in svm remote sensing image classification. In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pages 7353–7356.

Izquierdo-Verdiguier, E., Laparra, V., Gomez-Chova, L., and Camps-Valls, G. (2013b). Encoding invariances in remote sensing image classification with svm. *Geoscience and Remote Sensing Letters, IEEE*, 10(5):981–985.

Izquierdo-Verdiguier, E., Laparra, V., Jenssen, R., Gomez-Chova, L., and Camps-Valls, G. (2014b). Optimized kernel entropy components. *IEEE Transactions Neural Network*.

Jaakkola, T., Diekhans, M., and Haussler, D. (1999). Using the Fisher kernel method to detect remote protein homologies. In *In Proceedings of the Seventh Int. Conf. on Intelligent Systems for Molecular Biology*, pages 149–158. AAAI Press.

Jaakkola, T. and Haussler, D. (1998). Exploiting generative models in discriminative classifiers. In *NIPS'11*, pages 487–493.

Jain, A. K. (2010). Data Clustering: 50 Years Beyond K-Means. *Pattern Recognition Letters*, 31(8):651–666.

Jain, P., Kulis, B., Davis, J. V., and Dhillon, I. S. (2012). Metric and kernel learning using a linear transformation. *J. Mach. Learn. Res.*, 13:519–547.

Jebara, T., Kondor, R., and Howard, A. (2004). Probability product kernels. *J. Mach. Learn. Res.*, 5:819–844.

Jenssen, R. (2009). Information theoretic learning and kernel methods. In Emmert-Streib, F. and Dehmer, M., editors, *Information Theory and Statistical Learning*, pages 209–230. Springer US.

Jenssen, R. (2010). Kernel Entropy Component Analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31, Issue 9.

Jenssen, R. (2013a). Entropy-relevant dimensions in the kernel feature space: Cluster-capturing dimensionality reduction. *IEEE Signal Processing Magazine*, 30(4):30–39.

Jenssen, R. (2013b). Mean vector component analysis for visualization and clustering of non-negative data. *IEEE Trans. on Neural Networks and Learning Systems*, 24(10):1553–1564.

Jia, X., Kuo, B.-C., and Crawford, M. (2013). Feature mining for hyperspectral image classification. *Proceedings of the IEEE*, 101(3):676–697.

Jia, X. and Richards, J. (1999). Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification. *IEEE Trans. on Geos. Rem. Sens.,*, 37(1):538–542.

Jiang, Q., Yan, X., Lv, Z., and Guo, M. (2013). Fault Detection in Nonlinear Chemical Processes based on Kernel Entropy Component Analysis and an Angular Structure. *Korean Journal of Chemical Engineering*, 30(6).

Jolliffe, I. T. (2010). *Principal Component Analysis*. Springer, 2nd edition.

Kamusoko, C., Gamba, J., and Murakami, H. (2013). Monitoring urban spatial growth in harare metropolitan province, zimbabwe. *Advances in Remote Sensing*, 2013.

Karami, A., Yazdi, M., and Mercier, G. (2012). Compression of hyperspectral images using discerete wavelet transform and tucker decomposition. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(2):444–450.

Karimi, Y., Prasher, S., Madani, A., and Kim, S. (2008). Application of support vector machine technology for the estimation of crop biophysical parameters using aerial hyperspectral observations. *Canadian Biosystems Engineering / Le Genie des biosystems au Canada*, 50:7.13–7.20.

Keerthana, P. and Sivasankar, A. (2013). The Impact of Lossy Compression on Hyperspectral Data Adaptive Spectral Unmixing and PCA Classification. In *Intr. Journal of Science and Modern Engineering (IJISME)*, volume 1(7).

Kim, J. and Scott, C. D. (2012). Robust kernel density estimation. *J. Mach. Learn. Res.*, 13(1):2529–2565.

Kimeldorf, G. S. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95.

Kiyasu, S., Uraguchi, Y., Sonoda, K., and Sakai, T. (2011). Semi-supervised method for land cover classification of remotely sensed image considering spatial arrangement of the pixels. In *SICE Annual Conference (SICE), 2011 Proceedings of*, pages 2402–2405.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243.

Kuo, B.-C., Li, C.-H., and Yang, J.-M. (2009). Kernel nonparametric weighted feature extraction for hyperspectral image classification. *IEEE Trans. Geosc. Rem. Sens.*, 47(4):1139 –1155.

Kwon, H. and Nasrabadi, N. (2005). Kernel orthogonal subspace projection for hyperspectral signal classification. *IEEE Trans. on Geos. Rem. Sens.*, 43(12):2952–2962.

Laparra, V., Camps, G., and Malo, J. (2011). Iterative gaussianization: from ICA to random rotations. *IEEE Trans. Neur. Nets.*, 22(4):537–549.

Leberl, F., Gruber, M., Ponticelli, M., Bernoegger, S., and Perko, R. (2003). The ultracam large format aerial camera system. *Proceedings of the American Society for Photogrammetry and Remote Sensing*.

Lee, J. A. and Verleysen, M. (2007). *Nonlinear dimensionality reduction*. Springer.

Leiva-Murillo, J., Gomez-Chova, L., and Camps-Valls, G. (2013). Multitask remote sensing data classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(1):151–161.

Li, J., Bioucas-Dias, J., and Plaza, A. (2010). Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(11):4085–4098.

Li, J., Bioucas-Dias, J., and Plaza, A. (2011a). A new subspace discriminant analysis approach for supervised hyperspectral image classification. In *2011 IEEE Inter. Geosc. and Rem. Sens. Symp. (IGARSS)*, pages 3911–3914.

Li, L., Ustin, L., and Riano, D. (2007). Retrieval of fresh leaf fuel moisture content using genetic algorithm partial least squares (GA-PLS) modeling. *IEEE Geosc. and Rem. Sens. Lett.*, 4(2):216–220.

Li, W., Prasad, S., Fowler, J., and Bruce, L. (2011b). Locality-preserving discriminant analysis in kernel-induced feature spaces for hyperspectral image classification. *IEEE Geosc. and Rem. Sens. Lett.*, 8(5):894–898.

Lillesand, T. M., Kiefer, R. W., and Chipman, J. (2008). *Remote Sensing and Image Interpretation*. J. Wiley & Sons, NY.

Luo, X. Q. and Wu, X. J. (2012). Fusing Remote Sensing Images using a Statistical Model. *Applied Mechanics and Materials*, 263-266.

Luo, X. Q., Wu, X. J., and Zhang, Z. (2012). Kernel Entropy-Based Unsupervised Spectral Feature Selection. *Intr. Journal of Pattern Recognition and Artificial Intelligence*, 26(5).

Luo, X. Q., Wu, X. J., and Zhang, Z. (2013). Regional and Entropy Component Analysis based Remote Sensing Images Fusion. *Journal of Intelligent and Fuzzy Systems*.

Mantero, P., Moser, G., and Serpico, S. (2005). Partially supervised classification of remote sensing images through SVM-based probability density estimation. *IEEE Trans. on Geos. Rem. Sens.*, 43(3):559–570.

Marchesi, S. and Bruzzone, L. (2009). ICA and kernel ICA for change detection in multispectral remote sensing images. In *2009 IEEE Intr. Geos. Rem. Sens. Symp.,IGARSS 2009*, volume 2, pages II–980–II–983.

Maulik, U. and Chakraborty, D. (2011). A self-trained ensemble with semisupervised SVM: An application to pixel classification of remote sensing imagery. *Pattern Recognition*, 44(3):615 – 623.

Maulik, U. and Chakraborty, D. (2012). A novel semisupervised svm for pixel classification of remote sensing imagery. *International Journal of Machine Learning and Cybernetics*, 3(3):247–258.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Mendez-Rial, R. and Martín-Herrero, J. (2012). Efficiency of semi-implicit schemes for anisotropic diffusion in the hypercube. *IEEE Trans. on Image Processing*, 21(5):2389–2398.

Mercier, G. and Girard-Ardhuin, F. (2006). Partially supervised oil-slick detection by SAR imagery using kernel expansion. *IEEE Trans. Geosc. Remote Sensing*, 44(10(1)):2839–2846.

Mitra, P., Shankar, B. U., and Pal, S. K. (2004). Segmentation of multispectral remote sensing images using active support vector machines. *Pattern Recognition Letters*, 25(9):1067 – 1074.

Mountrakis, G., Im, J., and Ogole, C. (2011). Support vector machines in remote sensing: A review. *{ISPRS} Journal of Photogrammetry and Remote Sensing*, 66(3):247 – 259.

Muñoz Marí, J., Bovolo, F., Gómez-Chova, L., Bruzzone, L., and Camp-Valls, G. (2010). Semisupervised one-class support vector machines for classification of remote sensing data. *IEEE Trans. on Geos. and Rem. Sens.*, 48(8):3188–3197.

Muñoz Marí, J., Bruzzone, L., and Camps-Valls, G. (2007). A support vector domain description approach to supervised classification of remote sensing images. *IEEE Trans. Geosc. Remote Sensing*, 45(8):2683–2692.

Muñoz Marí, J., Gómez-Chova, L., Amorós-López, J., Izquierdo-Verdiguier, E., and Camps-Valls, G. (2013). Multiset kernel CCA for multitemporal image classification. In *7th Intr. Workshop on the Analysis of Multitemporal Remote Sensing Images*.

Muñoz Marí, J., Tuia, D., and Camps-Valls, G. (2012). Semisupervised classification of remote sensing images with active queries. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(10):3751–3763.

Myhre, J. and Jenssen, R. (2012). Mixture Weight Influence on Kernel Entropy Component Analysis and Semi-Supervised Learning using the LASSO. In *IEEE Intr. Workshop in Machine Learning for Signal Processing*, Santander, Spain.

N. Indhumadhi, G. P. (2011). Enhanced Image Fusion Algorithm Using Laplacian Pyramid and Spatial frequency Based Wavelet Algorithm. In *Intr. Journal of Soft Computing and Engineering (IJSCE)*, volume 1(5).

Naesset, E., Bollandsas, O. M., and Gobakken, T. (2005). Comparing regression methods in estimation of biophysical properties of forest stands from two different inventories using laser scanner data. *Remote Sensing of Environment*, 94(4):541 – 553.

Nielsen, A. (2007). The regularized iteratively reweighted MAD method for change detection in multi- and hyperspectral data. *IEEE Trans. on Image Processing*, 16(2):463–478.

Niu, X. and Ban, Y. (2013). Multi-temporal radarsat-2 polarimetric sar data for urban land-cover classification using an object-based support vector machine and a rule-based approach. *International Journal of Remote Sensing*, 34(1):1–26.

Okujeni, A., van der Linden, S., Tits, L., Somers, B., and Hostert, P. (2013). Support vector regression and synthetically mixed training data for quantifying urban land cover. *Remote Sensing of Environment*, 137(0):184 – 197.

Ozertem, U. and Erdogmus, D. (2011). Locally defined principal curves and surfaces. *J. Mach. Learn. Res.*, 12:1249–1286.

Pacifici, F. and Del Frate, F. (2010). Automatic change detection in very high resolution images with pulse-coupled neural networks. *Geos. Rem. Sens. Letters, IEEE*, 7(1):58–62.

Pal, M. and Foody, G. (2010). Feature selection for classification of hyperspectral data by svm. *IEEE Trans. on Geos. Rem. Sens.*, 48(5):2297–2307.

Pan, S. J. and Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Trans. Neural Networks*, 22:199–210.

Paola, J. D. and Schowengerdt, R. A. (1995). A review and analysis of backpropagation neural networks for classification of remotely-sensed multi-spectral imagery. *Intr. Journal of Remote Sensing*, 16(16):3033–3058.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.

Pearson, K. (1895). Contributions to the mathematical theory of evolution. II. skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London. (A.)*, 186:343–414.

Petropoulos, G. P., Kalaitzidis, C., and Vadrevu, K. P. (2012). Support vector machines and object-based classification for obtaining land-use/cover cartography from hyperion hyperspectral imagery. *Computers and Geosciences*, 41(0):99 – 107.

Pohl, C. and Van Genderen, J. L. (1998). Review article multisensor image fusion in remote sensing: Concepts, methods and applications. *Intr. Journal of Remote Sensing*, 19(5):823–854.

Principe, J. C. (2010). *Information Theoretic Learning: Renyi Entropy and Kernel Perspectives*. Springer.

Rakotomamonjy, A., Rouen, U. D., Bach, F., Canu, S., and Grandvalet, Y. (2008). Y.: Simplemkl. *Journal of Machine Learning Research 9*.

Rast, M. and Agency, E. S. (1999). *ESA Medium Resolution Imaging Spectrometer (MERIS)*. Intr. journal of remote sensing. Taylor & Francis.

Reed, M. C. and Simon, B. (1980). *Functional Analysis*, volume I of *Methods of Modern Mathematical Physics*. Academic Press.

Richards, J. A. and Jia, X. (1999). *Remote Sensing Digital Image Analysis. An Introduction*. Springer-Verlag, Berlin, Heidenberg, 3rd edition.

Rosipal, R. and Krämer, N. (2006). Overview and recent advances in partial least squares. In *Subspace, Latent Structure and Feature Selection*, volume 3940 of *LNCS*, pages 34–51. Springer.

Roweis, S. and Brody, C. (1999). Linear heteroencoders. Technical report, Gatsby Computational Neuroscience Unit, Alexandra House.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Roweis, S. T., Saul, L. K., and Hinton, G. E. (2002). Global coordination of local linear models. In *Advances in Neural Information Processing Systems 14*, pages 889–896. MIT Press.

Sampson, P. D., Streissguth, A. P., Barr, H. M., and Bookstein, F. L. (1989). Neurobehavioral effects of prenatal alcohol: Partial Least Squares analysis. *Neurotoxicology and teratology*, 11:477–491.

Schlapfer, D., Richter, R., and Kellenberger, T. (2012). Aspects of atmospheric and topographic correction of high spatial resolution imagery. In *2012 IEEE Inter. Geosc. and Rem. Sens. Symp. (IGARSS)*, pages 4291–4294.

Schölkopf, B., Burges, C., and Vapnik, V. (1996). Incorporating invariances in support vector learning machines. In *Artificial neural networks, ICANN'96. Lecture Notes in Comp. Science (Vol. 1112)*, pages 47–52, Berlin. Springer.

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*, COLT '01/EuroCOLT '01, pages 416–426, London, UK, UK. Springer-Verlag.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press Series, Cambridge, MA, USA.

Schölkopf, B., Smola, A. J., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319.

Scholz, M., Fraunholz, M., and Selbig, J. (2007). *Nonlinear principal component analysis: neural networks models and applications*, chapter 2, pages 44–67. Springer.

Schwert, B., Rogan, J., Giner, N. M., Ogneva-Himmelberger, Y., Blanchard, S. D., and Woodcock, C. (2013). A comparison of support vector machines and manual change detection for land-cover map updating in massachusetts, usa. *Remote Sensing Letters*, 4(9):882–890.

Serpico, S. and Moser, G. (2007). Extraction of spectral channels from hyperspectral images for classification purposes. *IEEE Trans. on Geos. Rem. Sens.*, 45(2):484–495.

Serra, J. (1988). *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic press.

Shaw, G. and Manolakis, D. (2002). Signal processing for hyperspectral image exploitation. *IEEE Signal Processing Magazine*, 50:12–16.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, MA, USA.

Shekar, B. H., Kumari, M. S., Mestetskiy, L. M., and Dyshkant, N. F. (2011). Face Recognition using Kernel Entropy Component Analysis. *Neurocomputing*, 74(6).

Shi, L., Zhang, L., Zhao, L., Yang, J., Li, P., and Zhang, L. (2013). The potential of linear discriminative laplacian eigenmaps dimensionality reduction in polarimetric sar classification for agricultural areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 86:124–135.

Shivaswamy, P. K. and Jebara, T. (2006). Permutation invariant SVMs. In *Proceedings of the 23rd ICML*, ICML'06, pages 817–824, New York, NY, USA. ACM.

Silva, G., Mello, M., Shimabukuro, Y., Rudorff, B., and de Castro Victoria, D. (2011). Multitemporal classification of natural vegetation cover in brazilian cerrado. In *Analysis of Multitemporal Remote Sensing Images (Multi-Temp), 2011 6th International Workshop on the*, pages 117–120.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14:199–222.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. R. (2010). Hilbert space embeddings and metrics on probability measures. *J. Mach. Learn. Res.*, 11:1517–1561.

Sun, Z., Wang, C., Li, D., and Li, J. (2014). Semisupervised classification for hyperspectral imagery with transductive multiple-kernel learning. *Geoscience and Remote Sensing Letters, IEEE*, 11(11):1991–1995.

Tarabalka, Y., Fauvel, M., Chanussot, J., and Benediktsson, J. (2010). Svm- and mrf-based method for accurate classification of hyperspectral images. *Geoscience and Remote Sensing Letters, IEEE*, 7(4):736–740.

Teh, Y. W. and Roweis, S. (2003). Automatic alignment of local representations. In *NIPS 15*, pages 841–848. MIT Press.

Tenenbaum, J. B., Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Townsend, P., Foster, J., Chastain, R.A., J., and Currie, W. (2003). Application of imaging spectroscopy to mapping canopy nitrogen in the forests of the central appalachian mountains using hyperion and AVIRIS. *IEEE Trans. on Geos. Rem. Sens.*, 41(6):1347–1354.

Tuia, D. and Camps-Valls, G. (2009a). Recent advances in remote sensing image processing. In *Image Processing (ICIP), 2009 16th IEEE Intr. Conference on*, pages 3705–3708.

Tuia, D. and Camps-Valls, G. (2009b). Semisupervised remote sensing image classification with cluster kernels. *IEEE Geosc. Rem. Sens. Letters*, 6(2):224–228.

Tuia, D. and Camps-Valls, G. (2011). Urban image classification with semisupervised multiscale cluster kernels. *IEEE JSTARS*, 4:65–74.

Tuia, D., Ratle, F., Pozdnoukhov, A., and Camps-Valls, G. (2010). Multisource composite kernels for urban-image classification. *IEEE Geosc. Rem. Sens. Lett.*, 7:88–92.

Tuia, D., Volpi, M., Copa, L., Kanevski, M., and Munoz-Mari, J. (2011a). A survey of active learning algorithms for supervised remote sensing image classification. *Selected Topics in Signal Processing, IEEE Journal of*, 5(3):606–617.

Tuia, D., Volpi, M., Copa, L., Kanevski, M., and Muñoz-Marí, J. (2011b). A survey of active learning algorithms for supervised remote sensing image classifications. *IEEE J. Sel. Topics Signal Proc.*, 5(3):606–617.

Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley & Sons, New York, NY, USA.

Vatsavai, R., Shekhar, S., and Burk, T. (2005). A semi-supervised learning method for remote sensing data mining. In *17th IEEE Inter. Conf. on Tools with Artificial Intelligence, 2005. ICTAI 05.*, pages 5 pp.–211.

Verbeek, J. J., Vlassis, N., and Krose, B. (2002). Coordinating principal component analyzers. In *In Proc. Intr. Conference on Artificial Neural Networks*, pages 914–919. Springer.

Verrelst, J., Alonso, L., Camps-Valls, G., Delegido, J., and Moreno, J. (2012a). Retrieval of vegetation biophysical parameters using gaussian process techniques. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(5):1832–1843.

Verrelst, J., Muñoz, J., Alonso, L., Delegido, J., Rivera, J. P., Camps-Valls, G., and Moreno, J. (2012b). Machine learning regression algorithms for biophysical parameter retrieval: Opportunities for sentinel-2 and -3. *Remote Sensing of Environment*, 118(0):127 – 139.

Vilas, L. G., Spyrakos, E., and Palenzuela, J. M. T. (2011). Neural network estimation of chlorophyll a from MERIS full resolution data for the coastal waters of galician rias (NW spain). *Remote Sensing of Environment*, 115(2):524 – 535.

Volpi, M., Tuia, D., Camps-Valls, G., and Kanevski, M. (2012). Unsupervised change detection with kernels. *IEEE Geosc. and Rem. Sens. Lett.*, 9(6):1026–1030.

von Luxburg, U. (2007). A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416.

Walder, C. and Lovell, B. (2005). Homogenized virtual support vector machines. In *Dig. Image Comp.: Tech. and Applic., 2005. DICTA '05. Proceedings 2005*, pages 57–63.

Wang, F., Huang, J., and Lou, Z. (2011). A comparison of three methods for estimating leaf area index of paddy rice from optimal hyperspectral bands. *Precision Agriculture*, 12(3):439–447.

Weinberger, K. Q. and Saul, L. K. (2004). Unsupervised learning of image manifolds by semidefinite programming. In *Proc. IEEE CVPR*, pages 988–995.

Weston, J., Leslie, C. S., Ie, E., Zhou, D., Elisseeff, A., and Noble, W. S. (2005). Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247.

Wilson, M., Ustin, L., and Rocke, D. (2004). Classification of contamination in salt marsh plants using hyperspectral reflectance. *IEEE Trans. on Geos. Rem. Sens.*, 42(5):1088–1095.

Wold, H. (1966). Estimation of principal components and related models by iterative least squares. *Multivariate Analysis*, pages 391–420.

Wu, J., Xiong, H., and Chen, J. (2009). Adapting the right measures for *k*-means clustering. In *Proc. of the 15th ACM SIGKDD inter. conf. on Knowledge discovery and data mining*, KDD '09, pages 877–886, New York, NY, USA. ACM.

Xie, Z. and Guan, L. (2012). Multimodal information fusion of audio emotion recognition based on kernel entropy component analysis. In *IEEE Intr. Symp. on Multimedia*.

Xu, R. and Wunch II, D. C. (2008). *Clustering*. Wiley and Sons.

Yamazaki, F., Liu, W., and Takasaki, M. (2009). Characteristics of shadow and removal of its effects for remote sensing imagery. In *IEEE Geosc. and Rem. Sens. Symp.*, volume 4, pages 426–429.

Yan, W. and Shaker, A. (2014). Radiometric correction and normalization of airborne lidar intensity data for improving land-cover classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 52(12):7658–7673.

Yang, X. b., Huang, J., Wu, Y., Wang, J., Wang, P., Wang, X., and Huete, A. (2011). Estimating biophysical parameters of rice with remote sensing data using support vector machines. *Science China Life Sciences*, 54(3):272–281.

You, C. H., Lee, K.-A., and Li, H. (2010). GMM-SVM kernel with a Bhattacharyya-based distance for speaker recognition. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(6):1300–1312.

Zolfaghari, K., Shang, J., McNairn, H., Li, J., and Homyouni, S. (2013). Using support vector machine (svm) for agriculture land use mapping with sar data: Preliminary results from western canada. In *Agro-Geoinformatics (Agro-Geoinformatics), 2013 Second International Conference on*, pages 126–130.