Eduardo Jose Castro Witting

Candidate

Electrical and Computer Engineering

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Vince D. Calhoun , Chairperson

Manel Martínez-Ramón (co-chair)

Vince P. Clark

Marios Pattichis

Application of Multiple Kernel Learning on Brain Imaging for Mental Illness Characterization

by

Eduardo Castro

B.S., Electrical Engineering, Pontificia Universidad Católica del Perú, 2003

M.S., Computer Engineering, University of New Mexico, 2008

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Engineering

The University of New Mexico

Albuquerque, New Mexico

Dec, 2013

Dedication

To God, who has always been by my side,
to my guardian angel, who has taken care of me all this time,
to my family, for their support and for being understanding,
and especially to my dear wife Carla, for her everlasting love and encouragement.

“It does not matter how slowly you go as long as you do not stop.” – Confucius

Acknowledgments

I would like to express my deepest gratitude to my advisor, Vince, for his support and guidance throughout these years in the good and in the very bad moments too. I would also like to thank him for giving me the chance to work in the wonderful field of medical imaging and bringing me the possibility of being involved in this small but genuine attempt to improve our yet incipient understanding of the human mind.

I would also like to sincerely thank my co-advisor, Manel, to whom I am indebted for his great contribution to this work, for his significant assistance that helped me acquire a broader technical knowledge, and for his kindness on the difficult moments I went through in this endeavor.

I am also obliged to Vanessa for her meticulous and unrestricted help in all aspects of the latest stage of this study. I am not only grateful to have been given the chance to work with her, but also to get to know a very nice and precious human being.

Finally, I would like to thank my wife, Carla, as well as Rogers, Lei and other members of the MIALAB group for their help during these years. I would also like to thank the members of the Imaging Genetics group for their support at the latest stage of this work.

Application of Multiple Kernel Learning on Brain Imaging for Mental Illness Characterization

by

Eduardo Castro

B.S., Electrical Engineering, Pontificia Universidad Católica del Perú, 2003

M.S., Computer Engineering, University of New Mexico, 2008

Ph.D., Engineering, University of New Mexico, 2013

Abstract

Mental disorders are diagnosed on the basis of reported symptoms and externally observed clinical signs. Nonetheless, these cannot be evaluated by means of clinical tests. This is the case for schizophrenia, a complex disease characterized by perturbations in language, perception, thinking, social relationships and will that affects about 1% of the U.S. population. Besides the absence of an objective assessment of symptoms to diagnose schizophrenia, not even a set of symptoms that uniquely characterize this disorder have been found.

Given the absence of a biologically-based diagnosis of schizophrenia, several studies have used different brain imaging techniques in an attempt to characterize the biological abnormalities found on patients. One of those techniques is functional magnetic resonance imaging (fMRI), a non-invasive technique that captures brain images that reflect neuronal activity. While fMRI studies have been able to provide significant information about schizophrenia, the acquired data present some technical challenges. FMRI characterizes the dynamics of brain activity in time for several brain volumetric elements (voxels), thus generating massive amounts of data. On the other hand, fMRI studies acquire images from tens or hundreds of subjects, the rate between the data dimensionality and the sample size being very high. One way of dealing with this issue is to use univariate approaches to analyze the data, i.e., analyze each voxel individually. However, such approaches neglect the spatial correlation in the data.

Machine learning algorithms can be used to do a multivariate analysis of fMRI data and predict a condition of interest. In addition, the contribution of the analyzed features (in this case voxels) to the learning task can be estimated. Nonetheless, these algorithms' performance is also precluded by the high dimensionality of fMRI data, making whole-brain approaches prone to poorly fit the data. For this reason, some studies restricted their analyses to sets of voxels within certain regions of interest (ROIs). While such approaches are able to solve the dimensionality problem, they do it at the expense of losing information from other potentially informative regions. Furthermore, these studies perform an interpretation of the results at a voxel level and not at a brain region level, which could potentially be richer and more meaningful. Under the assumption that activation is sparsely distributed across the brain, methods that are capable of performing a sparse region selection would be able to address the dimensionality problem and provide a better interpretation of brain activation patterns. Such functionality can be attained by using multiple kernel learning approaches.

This dissertation proposes a machine learning framework based on a multiple-kernel data representation to distinguish groups of schizophrenia patients from healthy controls using fMRI data, the activation patterns of each brain region being characterized by a kernel. This approach is capable of performing a sparse selection of informative regions and it is flexible enough to exploit linear or nonlinear relationships among the voxels within them. Two algorithms that follow this framework are presented: recursive composite kernels (RCK) and ν-multiple kernel learning (ν-MKL). This work evaluates these algorithms in terms of their prediction performance, the validity of the brain regions that are deemed relevant and their capacity to analyze diverse data sources.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Nowadays, medicine can detect several diseases by means of objective clinical tests, as certain biomarkers have been found to be associated to specific pathogenic processes. Unfortunately, this is not the case for several mental disorders, which are diagnosed on the basis of reported symptoms and externally observed clinical signs. Schizophrenia, a chronic, severe, and disabling illness, is one of such disorders. It is characterized by deficits in thought processes, perceptions, and emotional responsiveness and affects about 1% of the U.S. population [1].

Functional magnetic resonance imaging (fMRI) is a non-invasive technique that captures brain images that reflect neuronal activity. It has been extensively used on different experimental tasks and resting-state conditions to better understand the dynamics of normal and pathological brain function. Specifically, it has been widely used to study schizophrenia. Despite these efforts, the etiology of this disease remains unclear.

Machine learning has emerged as a valuable field of study that can predict a condition of interest by analyzing the interactions of different regions of the brain. However, most machine learning based studies applied to fMRI data are restricted to the analysis of subsets of voxels on regions of interest. Furthermore, these studies perform an interpretation of the results at a voxel level. Yet, a more robust understanding of cognitive processes can be obtained by parcellating whole brain data into brain regions, since different areas in the brain are specialized for different functions.

This provides the motivation of this study, which supplies the derivation and implementation of machine learning algorithms that detect the degree of abnormal activity on functional regions of the brain to achieve a better understanding of the cognitive processes involved in schizophrenia. By doing so, not only can the proposed algorithms obtain a more accurate schizophrenia detection rate, but they can also detect the most informative regions that characterize this disorder.

1.2 Thesis Statement

This PhD dissertation research presents a machine learning framework based on a multiple-kernel data representation to distinguish groups of schizophrenia patients from healthy controls using fMRI data. The activation patterns of each brain region are characterized by a kernel, which enables this approach to perform a sparse selection of informative regions and estimate the degree of abnormal activity in them. In addition, this framework is capable of achieving a better characterization of schizophrenia by analyzing diverse fMRI data sources.

1.3 Innovations and Contributions

A list of the primary innovations and contributions of this dissertation includes:

- The development of algorithms capable of detecting linear/nonlinear relationships between voxels within brain regions. These algorithms are capable of detecting a sparse set of informative regions for schizophrenia characterization, being less sensitive to the noise inherent to fMRI data.
- The intrinsic capability of these algorithms to analyze data from diverse sources, such as information retrieved from different fMRI data analysis methods.
- The capacity of these algorithms to better characterize schizophrenia by incorporating the phase of the fMRI signal in the classification task.

1.4 Organization

This dissertation is organized as follows:

Chapter 2 provides fMRI background and an overview of machine learning. Later, it provides a list of feature selection and classification approaches applied to fMRI and a review of multiple kernel learning algorithms.

Chapter 3 introduces the proposed machine learning framework and explains the rationale and the formulation of the algorithms devised under this structure. These algorithms are recursive composite kernels (RCK) and ν-multiple kernel learning (ν-MKL).

Chapter 4 presents the results obtained by RCK and ν-MKL on a simulated fMRI dataset where the amount of information present on different brain regions is known beforehand. By knowing this ground truth, the performance of both algorithms can be properly evaluated.

Chapter 5 presents the results of RCK and ν-MKL on the classification of healthy controls and schizophrenia patients on two different fMRI datasets acquired from an auditory task experiment. The first dataset is composed of data generated using different analysis methods. The study that applied RCK on this dataset is available in the following publication:

- E. Castro, M. Martínez-Ramón, G. Pearlson, J. Sui, and V. D. Calhoun, "Characterization of groups using composite kernels and multi-source fMRI analysis data: Application to schizophrenia," *NeuroImage*, vol. 58, no. 2, pp. 526–536, 2011.

The second dataset incorporates phase information in addition to fMRI magnitude data. A preliminary analysis based on RCK was applied to this dataset. Then, subjects from the same dataset were selected to better match controls and patients in terms of age and a more solid framework based on ν-MKL was applied to this dataset to further improve schizophrenia characterization. The studies which used the second dataset are available on the following publications:

- E. Castro, M. Martínez-Ramón, A. Caprihan, K. Kiehl, and V. D. Calhoun, "Complex fMRI data classification using composite kernels: Application to schizophrenia," Organization of Human Brain Mapping, 17th Annual Meeting, Canada, 2011.

- E. Castro, M. Martínez-Ramón, K. Kiehl, and V. D. Calhoun, "A multiple kernel learning approach for schizophrenia classification from complex-valued fMRI data," Organization of Human Brain Mapping, 19th Annual Meeting, Seattle, 2013.

- E. Castro, V. Gómez-Verdejo, M. Martínez-Ramón, K. A. Kiehl, and V. D. Calhoun, “A multiple kernel learning approach to perform classification of groups from complex-valued fMRI data analysis: Application to schizophrenia,” *NeuroImage* (in press).

Finally, conclusions, future work, and recommendations are presented on Chapter 6.

Chapter 2

Literature Review

This chapter provides a discussion of feature selection and classification approaches applied to fMRI as well as a review of multiple kernel learning algorithms, which can potentially be applied to fMRI. Prior to the discussion, an overview of fMRI and machine learning is presented.

2.1 fMRI Background[1]

This section gives an introduction to the fMRI, as well as an overview of fMRI data processing.

[1] The information provided in this section has been mainly retrieved from [2]

2.1.1 Introduction to fMRI

BOLD fMRI

FMRI is an MRI procedure that detects changes in blood flow or oxygenation in response to task activation. The most popular fMRI technique uses blood oxygenation level dependent (BOLD) contrast, which is based on the differing magnetic properties of oxygenated (diamagnetic) and deoxygenated (paramagnetic) blood. This approach takes advantage of the phenomenon that increases in neuronal activity are accompanied by local increases in perfusion.

The BOLD response dynamics can be explained as follows: Following an increase in neuronal activity, local blood flow increases. The increase in perfusion, in excess of that needed to support the increased oxygen consumption due to neuronal activation, results in a local decrease in the concentration of deoxyhemoglobin. As deoxyhemoglobin is paramagnetic, a reduction in its concentration results in an increase in the homogeneity of the static magnetic field, which yields an increase in the MRI signal (see Fig. 2.1).

However, the detected signal changes are small. Relatively low image signal-to-noise ratio (SNR) of the BOLD effect, head movement, and undesired physiological sources of variability (cardiac, pulmonary) make detection of the activation-related signal changes difficult.

The change in the MR signal from neuronal activity is called the hemodynamic response. This hemodynamic response is temporally delayed relative to neuronal activation by about 1 to 2 seconds.

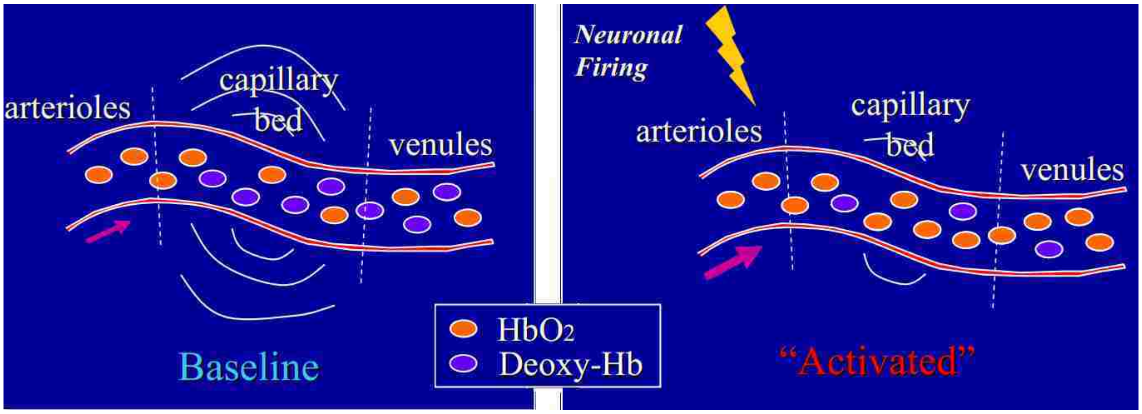


Figure 2.1: The BOLD response dynamics. As neural activity increases, local blood flow increases too. The increase of perfusion, in excess to what is needed to support increased neuronal oxygen consumption, results on a decrease in the concentration of deoxyhemoglobin. This, in turn, increases the homogeneity of the static magnetic field, yielding an increase in the MRI signal. (Extracted from [3])

Acquisition

The MRI signal is acquired as a quadrature signal. That is, two orthogonal "detectors" are used to capture the MRI signal. The two outputs from such a system are often put in a complex form, with one output being treated as the real part and the other one as the imaginary part. These are located in the frequency space, the data on the complex image space being obtained by means of the inverse Fourier transform.

Even though this complex-valued spatiotemporal data has been shown to contain physiologic information [4], virtually all fMRI studies analyze the magnitude images only. The analysis of complex fMRI data is discussed later on chapter 5. For the moment, only fMRI magnitude data will be discussed.

Noise

There are several types of signals that can be encoded within the hemodynamic signals measured by fMRI. They may be grouped into signals of interest and signals not of interest. The signals of interest include task-related and some others.

The signals not of interest (noise) include physiology-related, motion-related, and scanner-related signals. Physiology-related signals such as breathing and heart rate tend to come from the brain ventricles (fluid filled regions of the brain) and areas of the brain with large blood vessels present, respectively.

Motion-related signals can also be present and tend to be changes across large regions of the image (particularly at the edges of images). Head motion is a problem for fMRI acquisition since images are acquired at millimeter scale on absolute spatial locations. For this reason, even if the subject makes slight head movements of a few millimeters, this can have drastic effects upon the data.

Finally, there are scanner-related signals that can be varying in time (such as scanner drift and system noise) or varying in space (such as susceptibility and radio-frequency artifacts).

2.1.2 fMRI Data Processing

fMRI data processing is performed in several stages, which can be divided in two main blocks: preprocessing and data analysis. The upcoming sections discuss these two processing steps.

Preprocessing

The main preprocessing stages are: slice-timing correction, realignment, coregistration and normalization. Timing correction is necessary because each slice is typically acquired sequentially, rather than acquiring all slices simultaneously. This results in a slight phase shift between the slices and within each volume.

Realignment is required to correct for motion correction. A successful realignment ensures that the source of the signal in one voxel originates from the same location within each scan. This is usually done by applying rigid-body motion correction. The goal of coregistration is to obtain an overlap between functional images and the anatomical image, so that the activation areas are located at their correlation anatomical positions. Finally, since the brain of every individual is different, normalization is of extreme importance for group analyses. In addition, it helps to use standardized atlases to identify particular brain regions for comparisons between studies. Often a spatial smoothing stage is introduced following the normalization stage to reduce high frequency noise and increase the signal-to-noise ratio.

Data Analysis

FMRI data analysis methods can be broadly classified in two categories: model-based analysis and non model-based analysis methods. In this case, model-based refers to an explicit a priori model for the hemodynamic response.

Model-based Model-based methods assume a fixed hemodynamic model over time for the fMRI data. The most widely used method is an implementation of the general linear model (GLM), the simplest of which reduces to a simple correlation with a predicted temporal waveform. Often there is a hypothesized task-related waveform that may be convolved with an estimate of the hemodynamic point spread function prior to the analysis. This estimate is known as the hemodynamic response function (HRF). The primary limitation of this method is that the HRF and other regressors that can be included in the analysis must be specified a priori. Non model-based approaches provide additional flexibility and can potentially reveal new information in the fMRI data.

Non-model based[2] This section will focus on a specific non model-based approach: independent component analysis (ICA).

ICA is an application of blind source separation that attempts to decompose a data set into statistically independent components. For fMRI, it is usually used to extract spatial brain networks that are assumed to be systematically non-overlapping. Furthermore, temporal coherence of brain networks is usually assumed.

ICA is used in fMRI modeling to understand the spatio-temporal structure of the signal. Most applications of ICA to fMRI look for spatially independent components that are maximally independent. Given an observation data matrix, the aim of fMRI component analysis is to factor the data matrix into a product of a set of time courses and a set of spatial patterns, where the latter are assumed to be independent. Contrary to GLM, ICA does not attempt to explicitly parameterize the fMRI time course, which is estimated implicitly in the source separation algorithm (see Fig. 2.2).

Machine Learning and fMRI

The interpretation of fMRI requires analysis of high-dimensional, multivariate data. The inherent challenges of fMRI data gave rise to the application of machine learning algorithms to train classifiers to decode stimuli, mental states, behaviors and other variables of interest from it [6]. In order to provide a better understanding of the

[2] The information provided in this section has been mainly extracted from [5]

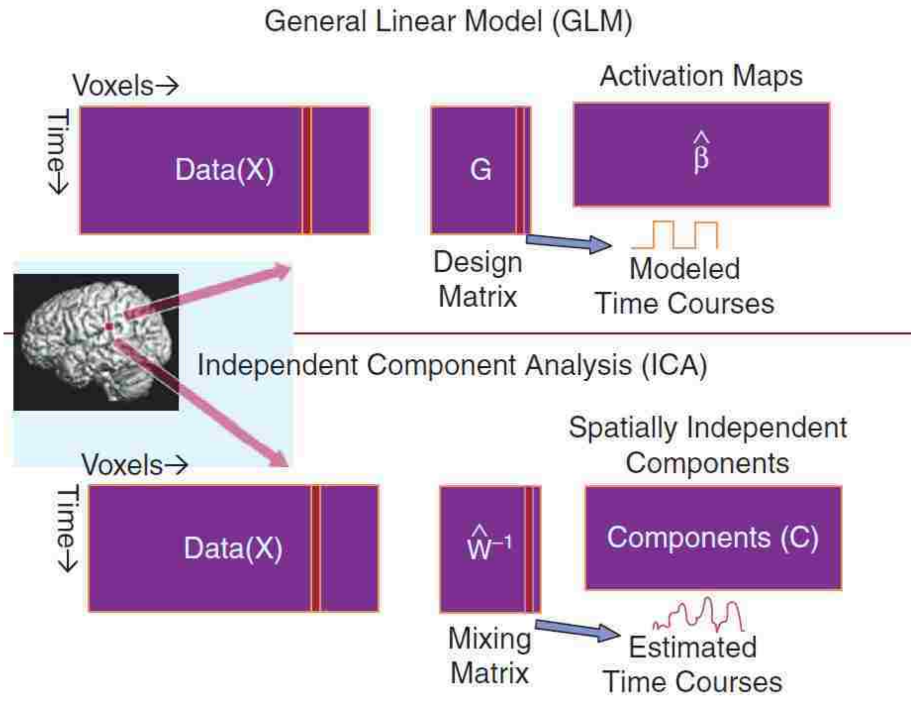


Figure 2.2: A comparison of GLM and ICA. To apply GLM (top) one needs a model for the fMRI time course, whereas in spatial ICA (bottom), there is no explicit temporal model for the fMRI time course (this is estimated along with the hemodynamic source locations). (Extracted from [5])

analyzed problem, machine learning is usually applied along with feature selection. The next section provides an overview of machine learning.

2.2 Introduction to Machine Learning[3]

Our world can be characterized by very diverse kinds of data, where any entity on it can be represented as a datum. More specifically, a datum (or data point) is a set of

[3] The information provided in this section has been mainly retrieved from the following textbooks: [7], [8] and [9]

numerical and/or categorical features that characterizes an object, a subject or an observation of a physical phenomenon.

This concept can be better visualized by means of an example. Let us assume that we need to characterize motor wheeled vehicles. To characterize a single vehicle, which would represent a data point, certain features would need to be extracted from it. These could be its color, average speed, number and size of wheels, number of occupants, size, weight, engine noise level, horsepower, etc.

Let us further assume that the task of interest is to identify the category to which the observed vehicle belongs to based on its data representation. In other words, classify the vehicle as a motorcycle, a car, a bus or a truck, being this an arbitrary categorization. Such a task would be trivial for an individual living on the city, who would do a one-to-one association between the vehicle and its category at first glance by instantly processing the information embedded on these features. However, this task would be much more complex if it had to be done for all the vehicles commuting at a given location of a highway on certain time periods to estimate its traffic congestion and the suitability of its pavement material. As the number of vehicles to be analyzed increases, the problem becomes less tractable for a human being. It would become even more intricate if this had to be done at multiple locations of this highway. Employing several thousands of people to perform such an assignment would be out of the question, as it would be cost-prohibitive and inaccurate. Machine learning would be an effective and sensible solution to this problem.

So what exactly is machine learning? By definition, it is a discipline related to the construction and study of systems that can learn data, where learning involves retrieving information from the data to generate knowledge. Let us revisit our example to better understand these concepts.

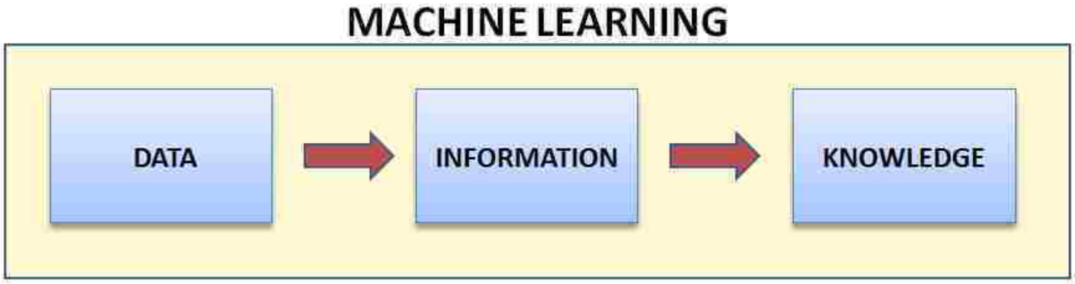


Figure 2.3: Machine learning generates knowledge from the input data.

Recall the aforementioned features that would be used to represent a vehicle. These were: color, average speed, number and size of wheels, number of occupants, size, weight, engine noise level and horsepower. Assuming an automated system could estimate a vehicle's horsepower, this feature would most probably be highly correlated to the engine noise level and its average speed. Similarly, the size and weight of the vehicle would be closely related to its wheel's size and its number of wheels. In fact, one or more of these features may be redundant, and therefore unnecessary to increase the *amount of available information* to characterize the vehicle properly. On the other hand, the number of occupants is an irrelevant feature as it is not strictly related to the loading capacity of the vehicle, thus giving *no helpful information* to determine the vehicle's category. Likewise, the color provides no information whatsoever about the class of the vehicle.

The next concept is knowledge. What kind of knowledge is it obtained with this task? This knowledge can be estimated using two criteria:

- The capacity of the trained machine to determine the category of a *new*, *unseen* vehicle.

- The capacity of the machine to determine which features are relevant to properly identify the different vehicle categories.

2.2.1 Types of Machine Learning

There are several branches of machine learning, out of which two will be discussed here. In *supervised learning*, the goal is to learn a mapping from inputs $\mathbf{x}$ to outputs y, given a labeled set of input-output pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1\dots N}$. An instance of supervised learning is the previously discussed vehicle classification task. In this setup, $\mathcal{D}$ is called the *(labeled) training set*, *training data* or *training sample*, and N is the number of *training observations*. In addition, $\mathbf{x}_i$ lies in some *input space* $\mathcal{X}$ and $y_i \in \mathcal{Y}$, the *output space*.

In the simplest setting, each *training example* $\mathbf{x}_i$ (also called *pattern* or *input vector*) is a d-dimensional vector of features that represent an object ($\mathbf{x}_i \in \mathbb{R}^d$). In general, however, $\mathbf{x}_i$ could be a complex structured object, such as a sentence, an email message, a molecular shape, etc.

Similarly the form of the output or response variable can in principle be anything, but most methods assume that y_i is a categorical or nominal variable from some finite set, $\mathcal{Y} = \{1, \dots, C\}$, or that y_i is a real-valued scalar ($\mathcal{Y} = \mathbb{R}$). When y_i is categorical, the problem is known as *classification* or *pattern recognition*, and when y_i is real-valued, the problem is known as *regression*. In the case of classification, the elements of $\mathcal{Y}$ are called *class labels*, or *classes*, for short. In fact, y_i is usually referred to as the class associated to $\mathbf{x}_i$.

The second main type of machine learning is *unsupervised learning*. Here we are only given inputs, $\mathcal{D} = \{\mathbf{x}_i\}_{i=1\dots N}$, and the goal is to find hidden structure in unlabeled data. A slight change in the proposed vehicle classification task can provide an instance of unsupervised learning. Let us assume that instead of capturing information at different locations of the highway, the vehicle identification approach would use satellite images of the highway to represent the vehicles. The training set would not provide any information about the types of vehicles on the road, so the machine would need to *discover* the four types of vehicles (motorcycles, cars, buses or trucks) by generating four data clusters, the vehicles within a cluster being more similar to those on the same cluster than the rest of the vehicles in the training set. Such an approach is known as clustering.

In what follows, this section will only discuss supervised learning. More precisely, it will be focused in classification.

2.2.2 Classification Problem

In classification the goal is to learn a mapping from inputs $\mathbf{x}$ to outputs y, where $\mathcal{Y} = \{1, \ldots, C\}$, with C being the number of classes. If $C = 2$, this is called binary classification (in which case we assume $\mathcal{Y} = \{\pm 1\}$); if $C > 2$, this is called multi-class classification.

One way to formalize the problem is by using a function approximation. Given a class of parametric functions $f(\mathbf{x}, \boldsymbol{\theta})$ we have

$$y = f(\mathbf{x}, \boldsymbol{\theta}) + \varepsilon, \tag{2.1}$$

where ε is the approximation error. The goal of learning is to estimate the optimal parameters $\boldsymbol{\theta}^*$ that minimize a loss function $l(\boldsymbol{\theta}, \varepsilon)$ on the training sample, i.e.,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} l(\varepsilon_i, \boldsymbol{\theta}), \tag{2.2}$$

where the best function approximation is defined by $f(\mathbf{x}, \boldsymbol{\theta}^*)$.

A good classifier is capable of achieving good *generalization*, which is the ability of the estimator to perform accurately on new, unseen examples after being trained.

Example: a simple classifier

Let us assume that the training data is linearly separable in input space $\mathcal{X}$, which for simplicity is assumed to be $\mathbb{R}^d$. Then, a linear classifier, which predicts the class of an input vector based on the value of a linear combination of its features, would give the following estimate:

$$\hat{y} = \operatorname{sgn}(f(\mathbf{x}, \mathbf{w}, b)) = \operatorname{sgn}(\mathbf{w}^T\mathbf{x} + b). \tag{2.3}$$

Since this is a non-differentiable function, an approximation of the form (2.1) is used

$$\begin{aligned} y &= f(\mathbf{x}) + \varepsilon \\ &= \mathbf{w}^T\mathbf{x} + b + \varepsilon, \end{aligned} \tag{2.4}$$

where ε is the residual of the fitted value of $\mathbf{x}$. One way of solving this problem is to find the least-squares solution, i.e., the solution that minimizes the expected value of the residual, which is estimated in the training sample as the the sum of the squared residuals for all the observations in it. Equivalently,

$$\begin{aligned} \min \mathbb{E}\left[\varepsilon^2\right] &= \min \mathbb{E}\left[(y - \mathbf{w}^T\mathbf{x} - b)^2\right] \\ &= \min \sum_{i=1}^{N} (y_i - \mathbf{w}^T\mathbf{x}_i - b)^2. \end{aligned} \tag{2.5}$$

A more compact representation of (2.4) can be obtained by rewriting it as

$$\begin{aligned} y &= f(\mathbf{x}) + \varepsilon \\ &= \tilde{\mathbf{w}}^T\tilde{\mathbf{x}} + \varepsilon, \end{aligned} \tag{2.6}$$

where $\tilde{\mathbf{w}} = [\mathbf{w}^T b]^T$ and $\tilde{\mathbf{x}} = [\mathbf{x}^T 1]^T$. Let $\tilde{\mathbf{X}}$ be a $(d+1) \times N$ matrix where each column is an input vector of the training set, the last row being a vector of elements equal to 1. Then (2.6) can be expressed as

$$\mathbf{y} = \tilde{\mathbf{X}}^T\tilde{\mathbf{w}} + \boldsymbol{\varepsilon}, \tag{2.7}$$

where $\mathbf{y} = \{y_i\}_{i=1}^N$ is the vector with the classes of each data point in the training set and $\boldsymbol{\varepsilon}$ is the residuals vector. Then, the least squares estimate of the coefficients of the linear classifier is

$$\tilde{\mathbf{w}} = \left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\right)^{-1}\tilde{\mathbf{X}}\mathbf{y} \tag{2.8}$$

By virtue of the representer theorem [10], $\tilde{\mathbf{w}}$ can be expressed as a function of the training examples

$$\tilde{\mathbf{w}} = \sum_{i=1}^{N} \alpha_i \tilde{\mathbf{x}}_i = \tilde{\mathbf{X}}\boldsymbol{\alpha}, \tag{2.9}$$

where $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^N$. By incorporating (2.9) in (2.8), the coefficients α_i can be estimated by

$$\begin{aligned} \tilde{\mathbf{X}}\boldsymbol{\alpha} &= \left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\right)^{-1}\tilde{\mathbf{X}}\mathbf{y} \\ \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\boldsymbol{\alpha} &= \tilde{\mathbf{X}}\mathbf{y} \\ \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\boldsymbol{\alpha} &= \mathbf{y} \\ \boldsymbol{\alpha} &= \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\mathbf{y}. \end{aligned} \tag{2.10}$$

If (2.9) is replaced in (2.6) we get

$$f(\mathbf{x}) = \boldsymbol{\alpha}^T\tilde{\mathbf{X}}^T\tilde{\mathbf{x}} = \sum_{i=1}^{N} \alpha_i \tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i^T\mathbf{x} + \sum_{i=1}^{N} \alpha_i, \tag{2.11}$$

so the linear classifier can be expressed in terms of a dot product between the training examples $\mathbf{x}_i$ and the test example $\mathbf{x}$.

2.2.3 Kernel Functions and Hilbert Spaces

It has been shown in the previous section that the predicted class of an unseen example $\mathbf{x}$ using a least-squares classifier can be estimated by computing the dot product between $\mathbf{x}$ and the the training examples $\mathbf{x}_i$. A geometrical interpretation of the dot product is that it computes the cosine of the angle between the vectors. In that sense, the dot product estimates the *similarity* between two vectors. By the same token, it can be said that the least-squares classifier estimates the class associated to $\mathbf{x}$ based on its similarity to the training examples.

Broadly speaking, a kernel function is analogous to the dot product in the sense that given two vectors $\mathbf{x}$, $\mathbf{x}^{'}$, it outputs a scalar characterizing their similarity [11]. That is the intuitive idea behind kernels used in the context of machine learning. The remainder of this section provides some definitions required to define Hilbert spaces. Then, an overview of kernels is presented, after which a formal description of kernels and related concepts is provided. Finally, these concepts are incorporated in the previously introduced least-squares classifier example.

Hilbert spaces[4]

Definition 1. *(Norms and Normed spaces). Let H be a vector space over the field $\mathbb{R}$ of real scalars. Then H is a* normed vector space *if for every $f \in H$ there is a real number $\|f\|$, called the* norm *of f, such that:*

(a) $\|f\| \geq 0$,

(b) $\|f\| = 0$ *if and only if* $f = 0$,

(c) $\|cf\| = |c| \, \|f\|$ *for every scalar c, and*

(d) $\|f + g\| \leq \|f\| + \|g\|$

Definition 2. *(Convergent and Cauchy sequences). Let H be a normed space, and let $\{f_n\}_{n \in \mathbb{N}}$ be a sequence of elements of H.*

[4] The information provided in this section has been retrieved from [12]

(a) $\{f_n\}_{n \in \mathbf{N}}$ *converges to* $f \in H$ *if* $\lim_{n\to\infty} \|f - f_n\| = 0$, *i.e., if*

$$\forall \epsilon > 0, \quad \exists N > 0, \quad \forall n \geq N, \quad \|f - f_n\| < \epsilon.$$

(b) $\{f_n\}_{n \in \mathbf{N}}$ *is Cauchy if*

$$\forall \epsilon > 0, \quad \exists N > 0, \quad \forall m, n \geq N, \quad \|f_m - f_n\| < \epsilon.$$

Definition 3. *(Completeness). A normed vector space H which does have the property that all Cauchy sequences are convergent is said to be* complete*.*

Definition 4. *(Inner products and inner product spaces). Let H be a vector space. Then H is an* inner product space *if for every* $f, g \in H$ *there exists a real number* $\langle f, g\rangle$ *called the* inner product *of f and g, such that:*

(a) $\langle f, f\rangle$ *is real and* $\langle f, f\rangle \geq 0$.

(b) $\langle f, f\rangle = 0$ *if and only if* $f = 0$.

(c) $\langle g, f\rangle = \langle f, g\rangle$,

(d) $\langle af_1 + bf_2, g\rangle = a\langle f_1, g\rangle + b\langle f_2, g\rangle$.

Each inner product determines a norm by the formula $\|f\| = \langle f, f\rangle^{1/2}$. *Hence every inner product space is a normed vector space.*

Definition 5. *(Hilbert space). A complete inner product space is called* Hilbert space.

Overview of kernels

Generally speaking, a Mercer's kernel in a Hilbert space $\mathcal{H}$ is a function that determines the inner product between vectors in $\mathcal{H}$. These vectors are maps of input vectors in $\mathcal{X}$, where the mapping function can be nonlinear. The possibility of using nonlinear transformations of the data gives the analytical power to kernel methods.

Kernel methods for machine learning have become an attractive alternative to traditional methods in machine learning for three main reasons: First, if the feature space is rich enough, then simple linear estimators with decision functions such as hyperplanes and half-spaces in feature space may be sufficient. For instance, to classify the examples in Fig. 2.4, a nonlinear decision boundary is needed, but once the points are mapped to a 3-dimensional space a hyperplane suffices. Second, kernels allow us to construct machine learning algorithms in Hilbert space $\mathcal{H}$ without explicitly computing the mapping of the input vectors. This makes it possible to *kernelize* linear algorithms provided that they can be expressed in terms of dot products between the data. Third, there is no need to make any assumptions about the input space $\mathcal{X}$ other than for it to be a set. This makes it possible to compute similarity between discrete objects such as strings, trees and graphs.

Kernel functions and mappings

Kernel methods rely on the properties of kernel functions, which are inner products of vectors mapped to Hilbert spaces through implicit (not necessarily known) mapping functions. The conditions that need to be met by kernel functions for this setting to hold are described in this section.

Reproducing kernel Hilbert spaces The notion of reproducing kernel Hilbert spaces (RKHS) through the reproducing property is explained here. Kernel properties following from RKHS will then be discussed.

Let $\mathcal{H}$ be a Hilbert space, whose elements are functions, that is provided with an inner product $< \cdot, \cdot >$. Let $f(\cdot)$ be an element of this space and $f(\mathbf{x})$ its value at a particular argument. We will assume that the arguments belong to the Euclidean space, i.e., $\mathbf{x} \in \mathbb{R}^d$

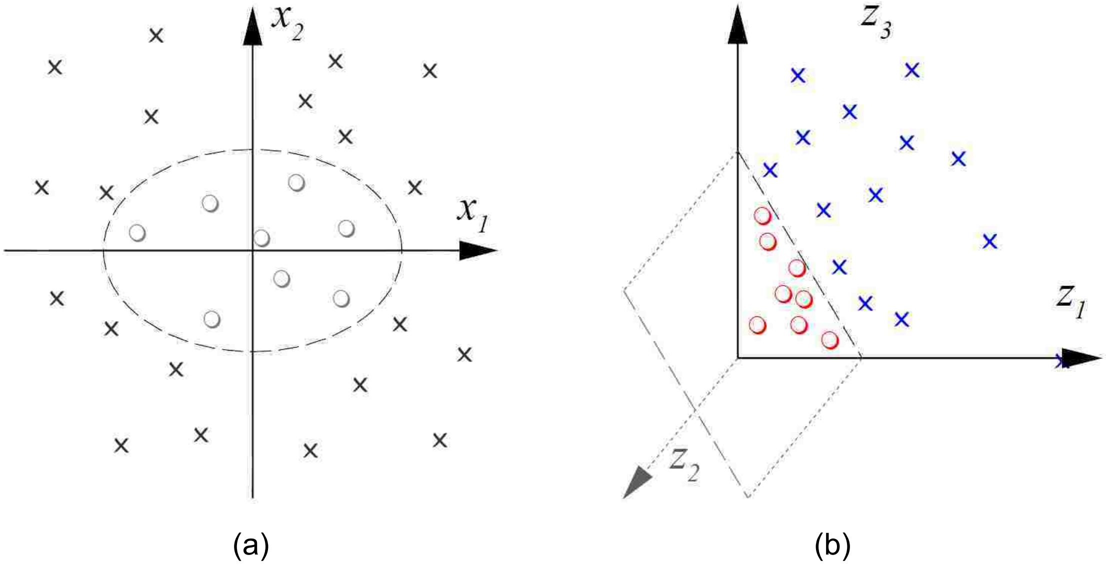


Figure 2.4: Nonlinear mapping from input space to feature space. On the input space (a) the examples are not linearly separable. By choosing a map $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ such that $\mathbf{z} = \varphi(\mathbf{x}) = [x_1^2 \; \sqrt{2}x_1x_2 \; x_2^2]^T$, a linear decision boundary in feature space (b) splits patterns from both classes. It can be shown that this feature space possesses the structure of an inner product that can be characterized by a kernel function $k(\mathbf{x}, \mathbf{x}^{'}) = \langle \mathbf{x}, \mathbf{x}^{'} \rangle^2 = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}^{'}) \rangle$. (Extracted from [13])

Definition 6. *(Reproducing Kernel Hilbert Space). A Hilbert space $\mathcal{H}$ is said to be an RKHS if [14]:*

1. *The elements of $\mathcal{H}$ are real valued functions $f(\cdot)$ defined on any set of elements $\mathbf{x}$.*
2. *For every element $\mathbf{x}$, $f(\cdot)$ is bounded.*

The name of these spaces come from the so called reproducing property. Indeed, in an RKHS $\mathcal{H}$, there exists a function $k(\cdot, \cdot)$ such that

$$f(\mathbf{x}) =< f(\cdot), k(\cdot, \mathbf{x}) >, f \in \mathcal{H} \tag{2.12}$$

by virtue of the Riesz Representation theorem [15]. The function $k(\cdot,\cdot)$ is called kernel.

It can be shown that the kernel $k(\cdot,\cdot)$ that generates the RKHS is a symmetric and positive definite function. In addition, this kernel fully generates the space. Furthermore, for a given kernel there is a unique RKHS; conversely, every RKHS contains a single kernel.

The Mercer's theorem The Mercer's theorem is of crucial importance because it expresses the analytical power of kernel methods. It embeds the idea behind the so called *kernel trick*, which makes it feasible to solve several nonlinear optimization problems through the construction of kernelized counterparts of linear algorithms.

Assume that $k(\cdot,\cdot)$ is a continuous kernel function that satisfies the properties of an RKHS, which have been recently discussed. Assume further that the kernel belongs to the family of square integrable functions. Also, let us define the following integral operator:

$$L_k f(\mathbf{x}) = \int_{\mathbf{x}} K(\mathbf{x},\mathbf{z}) f(\mathbf{z}) d\mu(\mathbf{z}), \tag{2.13}$$

μ being any Borel measure. The eigenfunctions φ_k of the operator form an orthonormal basis such that the corresponding eigenvalues λ_k form a nonnegative sequence.

Theorem 1. *(Mercer's) The aforementioned kernel can be expressed as*

$$k(\mathbf{x}_1,\mathbf{x}_2) = \sum_{k=1}^{\infty} \lambda_k \varphi_k(\mathbf{x}_1)\varphi_k(\mathbf{x}_2) \tag{2.14}$$

where the series converges uniformly for each pair $\mathbf{x}_1, \mathbf{x}_2$.

It follows from the Mercer's theorem that a mapping $\varphi : \mathcal{X} \rightarrow \mathcal{H}$ can be expressed as a (possibly infinite dimension) row vector

$$\varphi(\mathbf{x}) = \{\lambda_i^{1/2}\varphi_i(\mathbf{x})\}_{i=1}^{\infty} \tag{2.15}$$

From now on, we will only consider the case where $\mathcal{X} = \mathbb{R}^d$. The inner product between two of these maps $\varphi(\mathbf{x}_1)$ and $\varphi(\mathbf{x}_2)$ is defined then as the kernel function of vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ as

$$\varphi(\mathbf{x}_1)^T\varphi(\mathbf{x}_2) = \sum_{i=1}^{\infty} \lambda_i \varphi_i(\mathbf{x}_1)\varphi_i(\mathbf{x}_2) = k(\mathbf{x}_1, \mathbf{x}_2). \tag{2.16}$$

The Mercer's theorem shows that a mapping function into an RKHS and an inner product $k(\cdot, \cdot)$ exist if and only if $k(\cdot, \cdot)$ is a positive definite function. Therefore, if a function $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is proven to be positive definite, then it is the kernel of a given RKHS.

Regularization properties of kernels

Tikhonov regularization Regularization methods are designed to turn an ill-posed problem into a well-posed one such that a stable solution exists. A problem is well-posed if it meets 3 conditions: a solution to the problem exists, this solution is unique and it is stable to perturbations.

In particular, if a parameter estimation problem does not meet those conditions it is said to be unstable. Tikhonov minimization [16] is a regularization method that assures that such problems are well-posed.

Assume that we want to find a predictor $f(\mathbf{x})$ according to (2.4). Then the regularization procedure consists of constructing the functional

$$\mathcal{L} = \sum_{i=1}^{N} V\left(f(\mathbf{x}_i), y_i\right) + CR\left(f\right), \tag{2.17}$$

where V is a cost function over the empirical error of the estimation procedure, and R plays the role of a regularizer over the parameters of the predictor f. While the first term of the functional is used to choose the parameters that minimize the training error, the regularizer is used to account for a smooth solution.

The next section shows the extension of regularization to the field of RKHS.

The representer theorem The representer theorem [10] generalizes the idea of regularization to RKHS. This theorem provides two important findings. First, that given a training set, a solution of the regularization functional exists, which can be expressed as a linear combination of the maps of the training data points on the RKHS. Second, the smoothness of the solution is carried out by the particular kernel used to solve the estimation problem.

Theorem 2. *Assume an RKHS $\mathcal{H}$ provided with a kernel $k(\cdot,\cdot)$, and a dataset $\{\mathbf{x}_i, y_i\}_{i=1}^N$. Let us define an estimation problem as the selection of the function f^* over a given family of functions $\mathcal{F}$*

$$\mathcal{F} = \left\{ f(\cdot) = \sum_{i=1}^{\infty} \beta_i k(\cdot, \mathbf{x}_i), \beta_i \in \mathbb{R} \right\} \tag{2.18}$$

that better approximates the data. Assume an arbitrary convex function $V(f(\mathbf{x}_i), y_i)$ and a non-decreasing function R. Then, the function f^* that minimizes the functional

$$f^* = \arg\min_f \sum_{i=1}^{N} V(f(\mathbf{x}_i, y_i)) + R(\|f\|) \tag{2.19}$$

has a representation of the form

$$f^*(\cdot) = \sum_{i=1}^{N} \alpha_i k(\cdot, \mathbf{x}_i) \tag{2.20}$$

This is, the function that minimizes functional (2.19) *is a linear function of inner products between training data points mapped into the RKHS.*

A simple classifier revisited

Recall the least-squares classifier presented in section 2.2.2. The estimator can be expressed as

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle + \sum_{i=1}^{N} \alpha_i, \tag{2.21}$$

which is defined by a dot product between data points.

According to Mercer's theorem, if a kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is positive definite, then there exists a mapping function $\varphi : \mathbb{R}^d \to \mathcal{H}$, where $\mathcal{H}$ is an RKHS. Furthermore, $k(\cdot, \cdot)$ is the inner product of $\mathcal{H}$, such that $\varphi(\mathbf{x}_1)^T \varphi(\mathbf{x}_2) = k(\mathbf{x}_1, \mathbf{x}_2)$. By replacing the dot product in (2.21) by this kernel function we get

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + \sum_{i=1}^{N} \alpha_i. \tag{2.22}$$

This kernelized version of the linear least-squares classifier implicitly maps the training data points to $\mathcal{H}$. By doing so, the linear classifier can be extended to a nonlinear one with a nontrivial decision boundary.

2.2.4 Feature scaling

One thing to keep in mind is that features represent different properties of an object and as such, their values are most probably in different numeric ranges. Since some kernels depend on dot products of input vectors, features with big values may dominate those in smaller numeric ranges. For this reason, it is advisable to do feature scaling (normalization) prior to training the classifier.

There are several approaches to do feature scaling, among which the most widely used is *feature standardization.* This procedure makes the values of each feature in the data have zero mean and unit standard deviation.

2.2.5 Introduction to Support Vector Machines

From this point on we will change the notation used to represent the inner product for consistency purposes. Hence given $\mathbf{x}, \mathbf{z} \in \mathcal{H}$, where $\mathcal{H}$ is an inner product space, the inner product of $\mathbf{x}$ and $\mathbf{z}$ is represented as

$$\langle \mathbf{x}, \mathbf{z} \rangle := \mathbf{x}^T \mathbf{z} \tag{2.23}$$

Consider a binary classification task, where we are given N labeled training data $(\mathbf{x}_i, y_i)$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. Our goal is to find a linear decision boundary parameterized by $(\mathbf{w}, b)$ with $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that $\mathbf{w}^T\mathbf{x}_i + b > 0$ whenever $y_i = +1$ and $\mathbf{w}^T\mathbf{x}_i + b < 0$ whenever $y_i = -1$. Based on the rationale used to define the linear decision boundary, the separating hyperplane that determines it is defined by $\Pi_0 := \mathbf{w}^T\mathbf{x} + b = 0$.

Let us assume that $\mathbf{w}$ is scaled such that $\min_{i=\{1,\ldots,N\}} \left|\mathbf{w}^T\mathbf{x}_i + b\right| = 1$. If that is the case, the closest positive example to the separating hyperplane lies in $\Pi_1 := \mathbf{w}^T\mathbf{x} + b = 1$; similarly, the closest negative example lies in $\Pi_{-1} := \mathbf{w}^T\mathbf{x} + b = -1$ (see Fig. 2.5). Furthermore, every training example would satisfy $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$. Also let $d_+(d_-)$ be the distance from $\Pi_1(\Pi_{-1})$ to the separating hyperplane. Define the *margin* of a separating hyperplane to be $d_+ + d_-$. For the linearly separable case, support vector machines (SVMs) look for the separating hyperplane that achieves the maximum margin. The distances from Π_1 and Π_{-1} to the separating hyperplane equal $|1-b| / \|\mathbf{w}\|$ and $|-1-b| / \|\mathbf{w}\|$, respectively. Hence $d_+ = d_- = 1/\|\mathbf{w}\|$ and the margin is $2/\|\mathbf{w}\|$[5]. The problem of maximizing the margin therefore reduces to

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ s.t. \quad & y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1 \qquad \forall i. \end{aligned} \tag{2.24}$$

[5] Text excerpt extracted from [17]

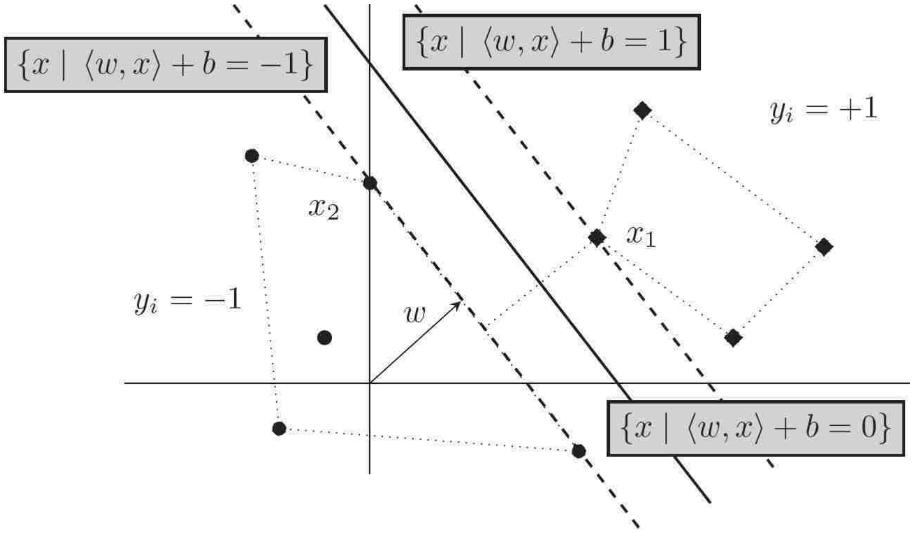

$\{x \mid \langle w,x\rangle + b = -1\}$
$\{x \mid \langle w,x\rangle + b = 1\}$
$y_i = +1$
x_2
x_1
$y_i = -1$
w
$\{x \mid \langle w,x\rangle + b = 0\}$

Figure 2.5: A linearly separable toy binary classification problem of separating the diamonds from the circles. The pair $(\mathbf{w}, b)$ is normalized to ensure $\min_{i=\{1,\ldots,N\}} \left|\mathbf{w}^T\mathbf{x}_i + b\right| = 1$. (Extracted from [9])

In deriving Eq. 2.24, we implicitly assumed that the data is linearly separable, that is, there is a hyperplane which correctly classifies the training data. Such a classifier is called a *hard margin classifier.* If the data is not linearly separable, then Eq. 2.24 does not have a solution. To deal with this situation Cortes and Vapnik [18] introduced non-negative slack variables ξ_i to relax the constraints:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i. \tag{2.25}$$

The work in [18] also required the reformulation of the optimization problem to penalize large values of ξ_i. This is done through this modified optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ s.t. \quad & y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1 - \xi_i \qquad \forall i \\ & \xi_i \geq 0 \qquad\qquad\qquad\quad \forall i, \end{aligned} \tag{2.26}$$

where $C > 0$ is a penalty parameter. The resulting classifier is said to be a *soft margin* classifier and Eq. 2.26 is called the SVM primal problem.

It can be demonstrated that the SVM dual problem can be expressed in terms of a dot product. As a consequence, an SVM can be kernelized, i.e., there exists a mapping $\varphi : \mathbb{R}^d \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space whose inner product is defined by $k(\mathbf{x}, \mathbf{x}^{'}) = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}^{'}) \rangle$. In essence, Eq. 2.26 can be extended to

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i & \\ s.t. \quad & y_i \left(\mathbf{w}^T \varphi(\mathbf{x}_i) + b \right) \geq 1 - \xi_i & \forall i \\ & \xi_i \geq 0 & \forall i. \end{aligned} \tag{2.27}$$

The objective function and the inequality constraints can be combined in the Lagrangian as

$$L_P = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \left\{ y_i \left[\mathbf{w}^T \varphi(\mathbf{x}_i) + b \right] - 1 + \xi_i \right\} - \sum_{i=1}^{N} \mu_i \xi_i. \tag{2.28}$$

Finally, the application of the Karush-Kuhn-Tucker (KKT) optimality conditions on the Lagrangian yield the following dual formulation:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{N} \alpha_i & \\ s.t. \quad & 0 \leq \alpha_i \leq C & \forall i \\ & \sum_{i=1}^{N} \alpha_i y_i = 0. & \end{aligned} \tag{2.29}$$

Different optimization strategies have been proposed for this problem. The most popular approach to solve this optimization problem is the Sequential Minimal Optimization [19]. This algorithm works on the dual formulation and breaks the overall quadratic problem (QP) of (2.29) on smallest subproblems composed of two Lagrange multipliers at each step. Solving for two Lagrange multipliers can be done analytically, so numerical QP optimization is avoided entirely.

Another optimization strategy, the iterative re-wighted least squares (IRWLS), rearranges (2.28) to incorporate a weighted least squares error term on the formulation, these weights being updated iteratively. This modified primal formulation is used to solve linear SVMs. For non-linear variants, this method uses the dual. Finally, [20] proposes a method to find the SVM solution in the primal and highlights the benefits of doing so when the machine needs to be trained with large amounts of data.

In summary, solutions in the primal are advantageous when linear kernels are used for a large scale optimization problem. In other cases, it is usually better to solve the SVM using the dual formulation.

2.3 Feature Selection and Classification

2.3.1 Feature Selection

One of the main difficulties of applying pattern recognition to certain problems is that the ratio between the number of collected features and the size of the training sample can be very high. The significant difference between the data dimensionality and the number of available observations can affect the generalization performance of the classifier or even preclude its use due to the low average information per dimension present in the data. Thus, it is desirable to reduce the data dimensionality with an algorithm that loses the least amount of information possible. This approach is consistent with the assumption that the data contains many redundant or irrelevant features.

There are two ways of performing dimensionality reduction: feature extraction and feature selection. The former approach transforms the input data into a reduced representation set of features to extract the relevant information of the data. On the other hand, feature selection picks a subset of relevant features on the input space. One advantage of feature selection is that it keeps the original features, thus providing better interpretability of the analyzed data.

Feature selection methods can be divided into three categories: filters, wrappers and embedded methods [21]. Filters select a subset of features as a preprocessing step to classification. On the other hand, wrappers and embedded methods use the classifier itself to find the optimal feature set. The difference between them is that while wrappers make use of the learning machine to select the feature set that increases its prediction accuracy, embedded methods incorporate feature selection as part of the training phase of the learning machine.

2.3.2 Application to fMRI

As it has been mentioned on section 2.1.2, fMRI studies have to deal with the high dimensionality of the data, which is retrieved from a small set of subjects. Therefore, feature selection is well-suited for the analysis of fMRI data. For this reason, different methods that incorporate feature selection have been applied on this field. We briefly mention some of them in what follows.

Mourão-Miranda et al. [22] proposed the application of temporal compression and space selection to fMRI data from a visual experiment. This space selection procedure extracted a subset of voxels with statistically significant activation for the analyzed tasks, which is a clear example of a filtering approach. Similarly, Haynes and Rees [23] applied filtering by selecting the top 100 voxels that had the strongest activation in two different visual stimuli prior to the application of classification. In both cases, these methods applied univariate strategies to perform feature selection. This is a valid strategy that also has a fast execution time, but it does not account for the multivariate relationships between voxels.

Another work [24] used a hybrid filter/wrapper approach by applying univariate voxel selection strategies prior to using recursive feature elimination SVM (RFE-SVM) [25] on both simulated and real data. RFE-SVM provides an alternative solution to univariate approaches, as it performs a multivariate ranking of features, discarding the least informative one at each iteration of the algorithm. In this case, the optimal feature subset is the one that achieves the best classification accuracy on an independent validation dataset. Nonetheless, it is a computational intensive method since it requires the SVM to be retrained M times, where M is the data dimensionality. While it is possible to remove several features at a time, this could come at the expense of classification performance degradation [25].

Another possibility is to use embedded feature selection methods such as the one presented in [26], which has a smaller execution time since it does not require to be repeatedly retrained. One class of algorithms that also fits the characterization of embedded feature selection methods is multiple kernel learning. Despite its potential to detect sets of relevant features, multiple kernel learning has not been applied to fMRI data to date. The following section provides an introduction to multiple kernel learning and reviews the evolution of this field.

2.3.3 Multiple Kernel Learning

Overview

Multiple kernel learning (MKL) algorithms aroused as developments on SVM and other kernel-based methods emphasized the need to consider multiple kernels, or parameterizations of kernels, and not a single fixed kernel. The incorporation of multiple kernels provides a flexible framework to solve practical problems that often involve multiple, heterogeneous data sources.

Another issue of single-kernel approaches is that the resulting decision function is hard to interpret, making it difficult to extract relevant knowledge about the problem at hand. If a distinct set of features is used by each kernel and a sparse weighting of the kernels is achieved, then one can quite easily interpret the resulting decision function.

Camps et al. [27] evaluated composite kernels, a combination of kernels from different data sources, as an unweighted or a convex linear combination of kernels applied to spatial and spectral information for hyperspectral image classification.

The approach presented in [28] is one of the first that tried to find an optimal combination of kernels on a real-world data set and embedded the kernel coefficients estimation on its optimization formulation. Based on a formulation presented in [29], kernels were generated using different genetic data sources such as gene expression data, known protein-protein interactions and others, casting this problem as a convex optimization one (semi-definite programming). If the kernel coefficients were constrained to be non-negative, the semi-definite program (SDP) reduced to a quadratically-constrained quadratic program (QCQP). They proposed the application of seven kernel functions to generate kernel matrices depending on the data source and claimed that their SDP-based approach performed better than a classifier trained using a naive, unweighted combination of kernels.

Later, Bach et al. [30] proposed a dual formulation of the QCQP presented in [29] as a second-order cone programming (SOCP) problem [31], which can be solved using sequential minimal optimization (SMO) techniques [32].

The rationale of the proposed algorithm is introduced through the formulation of a linear classifier that is an extension of the linear SVM. Specifically, given n labeled data $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \{-1, 1\}$, the input space $\mathcal{X} = \mathbb{R}^k$ is decomposed as the product of m blocks: $\mathbb{R}^k = \mathbb{R}^{k_1} \times \cdots \times \mathbb{R}^{k_m}$. The idea is to find a classifier of the form $y = \mathbf{w}^T\mathbf{x} + b$ such that $\mathbf{w}$ is block-sparse. In order to achieve a classifier with maximum margin and block-sparsity, the structural risk term $\|\mathbf{w}\|^2$ of the SVM formulation (see Eq. 2.26) is replaced by $(\sum_{j=1}^{m} d_j \|\mathbf{w}_j\|)^2$, which is the square of a weighted block l_1-norm of $\mathbf{w}$. From this primal problem, the authors derived the following dual problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\gamma^2 - \boldsymbol{\alpha}^T\boldsymbol{e} \\ w.r.t. \quad & \gamma \in \mathbb{R},\ \boldsymbol{\alpha} \in \mathbb{R}^n \\ s.t. \quad & 0 \leq \boldsymbol{\alpha} \leq C,\ \boldsymbol{\alpha}^T\mathbf{y} = 0 \\ & \left\| \sum_i \alpha_i y_i \mathbf{x}_{ji} \right\| \leq d_j\gamma,\ \forall j \in \{1, \ldots, m\}, \end{aligned} \tag{2.30}$$

where $e \in \mathbb{R}^n$ is a vector of all ones and the resulting formulation is an SOCP. After the analysis of the KKT optimality conditions, they demonstrated that the solution of this problem yielded that $\exists \eta_j : \mathbf{w}_j = \eta_j \sum_i \alpha_i y_i \mathbf{x}_{ji}$, where $\eta_j = 0$ for some values of j. In other words, by solving the dual problem $\mathbf{w}$ was block-sparse.

Then they removed the assumption that the classifier worked directly on the input space $\mathcal{X}$ and assumed that each input vector was mapped to a Hilbert space via a mapping $\varphi : \mathcal{X} \to \mathcal{H}$. They also assumed that, in correspondence with their block-based formulation of the classification problem, $\varphi(x)$ had m blocks $\varphi(x) = [\varphi_1(x), \ldots, \varphi_m(x)]$ and further assumed that this mapping was performed implicitly using kernel functions $k_j(\cdot, \cdot)$ for each block. Finally, they defined K_j as the Gram matrices generated by the available input vectors for each kernel function $k_j(\cdot, \cdot)$. By doing so, they generalized their approach on feature space with this formulation

$$\begin{aligned} \min \quad & \frac{1}{2}\gamma^2 - \boldsymbol{\alpha}^T\boldsymbol{e} \\ w.r.t. \quad & \gamma \in \mathbb{R},\ \boldsymbol{\alpha} \in \mathbb{R}^n \\ s.t. \quad & 0 \leq \boldsymbol{\alpha} \leq C,\ \boldsymbol{\alpha}^T\mathbf{y} = 0 \\ & (\boldsymbol{\alpha}^T D(\mathbf{y}) K_j D(\mathbf{y}) \boldsymbol{\alpha})^{1/2} \leq d_j\gamma,\ \forall j, \end{aligned} \tag{2.31}$$

where $D(\mathbf{y})$ is the diagonal matrix with diagonal $\mathbf{y}$.

The resulting classifier has the same structure as an SVM, but based on the sparse kernel matrix combination $K = \sum_j \eta_j K_j$. Later they demonstrated that by taking $d_j = \sqrt{\frac{trK_j}{c}}$, where $tr \sum_{j=1}^m \eta_j K_j = c$ and $c > 0$ is fixed, this formulation is in fact the dual of the QCQP formulation in [29].

Some years later, Sonnenburg et al. [33] reformulated the binary classification MKL problem in [28] as a semi-infinite linear program (SILP), which can be solved using a linear program (LP) solver and a standard SVM implementation by means of a wrapper method. Recall the formulation in [30], where the sparse kernel matrix is represented as $K = \sum_j \eta_j K_j$. Sonnenburg et al. reformulated Eq. 2.31 for $d_j = 1\ \forall j$, thus generating the following SILP:

$$\begin{aligned} \max \quad & \theta \\ w.r.t. \quad & \theta \in \mathbb{R},\ \boldsymbol{\eta} \in \mathbb{R}^m \\ s.t. \quad & \boldsymbol{\eta} \geq 0,\ \sum_{j=1}^{m} \eta_j = 1,\ \sum_{j=1}^{m} \eta_j S_j(\boldsymbol{\alpha}) \geq \theta \\ & \forall \boldsymbol{\alpha} \in \mathbb{R}^n :\ 0 \leq \boldsymbol{\alpha} \leq C,\ \boldsymbol{\alpha}^T \mathbf{y} = 0, \end{aligned} \tag{2.32}$$

where $S_j(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T D(\mathbf{y}) K_j D(\mathbf{y}) \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{e}$.

This is an LP in θ and $\boldsymbol{\eta}$, but it has infinitely many constraints, one for each $\boldsymbol{\alpha}$ satisfying the constraints in Eq. 2.32. The proposed algorithm is solved using a wrapper algorithm that finds $\boldsymbol{\alpha}$ with an SVM solver for an initialized set of values of $\boldsymbol{\eta}$ that generate the single kernel matrix $K = \sum_j \eta_j K_j$, after which the values of $\boldsymbol{\eta}$ and θ are updated. This procedure is iteratively repeated until a certain convergence criterion is satisfied, yielding an approximate solution as no convergence rates for this algorithm are known.

This work further generalized this algorithm to arbitrary strictly convex and differentiable loss functions, for which their MKL SILP formulation were derived.

This extension made it possible for this algorithm to solve regression problems too.

The algorithms mentioned so far attempted to achieve sparsity by using l_1-norm regularization terms, an approach that has exhibited certain disadvantages for linear SVM. As it is pointed out in [34, 35], l_1-norm SVM presents two limitations: first, when there are highly correlated features, it usually removes some of them; and second, the maximum number of selected features is limited by the number of available training data. Some approaches [36, 37, 38] have attempted to address these shortcomings. These are discussed later in this section.

The work presented in [39] provides another MKL framework. However, its theory was not conceived with the same motivation of the aforementioned MKL publications. Its main motivation was to solve the problem of choosing a kernel function suitable for estimation with a support vector machine. This was done by defining a RKHS on the space of kernels itself and minimizing a risk functional to select the optimal kernel: a regularized quality functional, which measured the 'badness' of the kernel function.

The convergence point between this approach and the previous ones is that the optimal kernel is not a single kernel, but a linear combination of them. The introduced optimal kernel, which is a kernel on the space of kernels itself, is called hyperkernel. The positive definiteness of the generated kernel function is ensured using the positive definiteness of the kernel matrix, and the resulting optimization problem is an SDP.

An extension of the work in [39] was presented in [40]. This work showed that the hyperkernel method presented in [39] could be equivalently formulated as an SOCP. They also mentioned that while the work in [29] was cast as an SDP too, it differed from [39] in this sense: the former work looks for the optimal kernel matrix, whereas the hyperkernel approach looks for the optimal kernel function from a given family of kernel functions. However, the problem setting is essentially the same as previous approaches, i.e., learning from hyperkernels involves optimizing two sets of variables: the set of coefficients of the training examples ($\boldsymbol{\alpha}$) and the set of coefficients on the hyperkernel expansion ($\boldsymbol{\eta}$).

The disadvantage of using hyperkernels is that learning arbitrary kernel combinations is a problem too general to allow for a general optimal solution. If the problem is further restricted, it is possible to achieve guaranteed optimality.

A work that tried to analyze the shortcomings of l_1-norm MKL is presented in [36]. The authors empirically investigated the best tradeoff between sparse and uniformly-weighted MKL on real and simulated data sets due to the evidence of sparse MKL being frequently outperformed by the latter method [41]. This tradeoff was evaluated using an elastic-net type regularization term, which is a smooth interpolation between the sparse (l_1-norm) MKL and the uniformly-weighted MKL. They discovered that the best accuracy rate was obtained in between the sparse and uniformly weighted MKL.

Some years later, Orabona and Jie [37] discussed several MKL algorithms, such as those that suggest an alternating optimization of SVM parameters and kernel parameters, as it is proposed in [33]. They appraised the fact that these algorithms can use existing SVM solvers for the SVM optimization step. However, they criticized that these algorithms not always guarantee convergence, thus estimating approximate solutions provided an error tolerance. Furthermore, they criticized the usage of such approaches on dual algorithms, as the obtained solution may be far from the optimal one.

This work also reviewed the algorithms based on l_p-norm constraints, such as [38]. They emphasized that these algorithms are not designed to achieve sparsity. In addition, they claimed that another limitation of such algorithms is that they rely on particular loss functions, and the entire algorithm has to be changed if the loss function is changed.

They proposed an MKL algorithm that achieves a tunable level of sparsity and a fast convergence rate that is independent on the particular convex loss function used. They adapted [36], replacing the l_2-norm regularization term by an l_p-norm term, p being selected to improve the convergence rate of the algorithm. In addition, they multiplied the l_1-norm term by their sparsity coefficient α. While this coefficient certainly adjusts the sparsity level of their solution, it does not provide the actual achieved sparsity, or at least a bound for it.

On the same year, a paper by Kloft et al. [38] presented three main contributions to MKL: First, it proposed a general framework which, theoretically, consolidated the MKL formulations proposed to date. Second, it proposed a non-sparse l_p-norm MKL approach with arbitrary $p \geq 1$, which they claimed achieves accuracies that surpass the state-of-the-art. Finally, they proposed interleaved optimization strategies for l_p-norm MKL that are faster than commonly used wrapper approaches such as the one presented in [33].

Kloft et al. supported their proposed l_p-norm MKL based on evidence that sparse MKL implementations usually achieve accuracy rates smaller than that of a regular SVM trained using an unweighted-sum kernel $K = \sum_j K_j$, as highlighted in [41]; they actually adjusted the value of p in order to tune the level of 'sparsity', with $p = 1$ achieving actual sparsity and $p \to \infty$ being equivalent to performing a uniformly-weighted kernel combination. The problem with this statement is based on the fact that the authors generalize sparse MKL implementations as being represented by l_1-norm MKL. In fact, this paper lists what would be a comprehensive list of publications on MKL, but they do not mention implementations based on elastic-net type regularization terms, such as the ones presented in [36] and [37].

Despite the aforementioned omissions, we agree with the authors that multiple kernel learning research in the past years has been focused almost only on accelerating algorithms for learning convex combinations of kernels, and that they provide an approach that can provide better accuracies than previous works. Finally, they conclude that both the correlation amongst the kernels with each other and their correlation with the target (i.e., the amount of discriminative information that they carry) play a role in the distinction of sparse from non-sparse scenarios.

Kernel Normalization

On section 2.2.4, we highlighted the importance of feature scaling when using single-kernel models. Likewise, kernel normalization is key for MKL. As it has been seen on the review of MKL approaches, the norms of feature spaces' weight vectors are required to be small. This can be done more easily for those features that are on a smaller magnitude scale. In order to have a choice of kernels that is unbiased by data scaling factors, kernel normalization is required.

This section presents two steps for kernel standardization, which is analogous to feature standardization. These steps are mean removal and variance normalization. Both steps use the following notation: $k_l(\cdot, \cdot)$ is the kernel from block l prior to the application of either normalization procedure, its corresponding feature map being represented as $\varphi_l(\cdot)$. On the other hand, $\tilde{k}_l(\cdot, \cdot)$ is the normalized kernel with associated feature map $\tilde{\varphi}_l(\cdot)$.

Mean Removal This step adjusts kernel $k_l(\cdot, \cdot)$ so that the N training examples have mean zero on feature space. More specifically, given N labeled training data $(\mathbf{x}_i, y_i)$ with $\mathbf{x}_i \in \mathcal{X}$ we require that

$$\sum_{i=1}^{N} \tilde{\varphi}_l(\mathbf{x}_i) = 0. \tag{2.33}$$

Mean removal in feature space l can be obtained implicitly by manipulating kernel $k_l(\cdot,\cdot)$ as shown below:

$$\begin{aligned}
\tilde{k}_l(\mathbf{x}_i,\mathbf{x}_j) &= \tilde{\varphi}_l^T(\mathbf{x}_i)\tilde{\varphi}_l(\mathbf{x}_j) \\
&= \left(\varphi_l(\mathbf{x}_i) - \frac{1}{N}\sum_{m=1}^{N}\varphi_l(\mathbf{x}_m)\right)^T \left(\varphi_l(\mathbf{x}_j) - \frac{1}{N}\sum_{n=1}^{N}\varphi_l(\mathbf{x}_n)\right) \\
&= \varphi_l^T(\mathbf{x}_i)\varphi_l(\mathbf{x}_j) - \frac{1}{N}\sum_{n=1}^{N}\varphi_l^T(\mathbf{x}_i)\varphi_l(\mathbf{x}_n) - \frac{1}{N}\sum_{m=1}^{N}\varphi_l^T(\mathbf{x}_m)\varphi_l(\mathbf{x}_j) \\
&\quad + \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\varphi_l^T(\mathbf{x}_m)\varphi_l(\mathbf{x}_n) \\
&= k_l(\mathbf{x}_i,\mathbf{x}_j) - \frac{1}{N}\sum_{n=1}^{N}k_l(\mathbf{x}_i,\mathbf{x}_n) - \frac{1}{N}\sum_{m=1}^{N}k_l(\mathbf{x}_m,\mathbf{x}_j) \\
&\quad + \frac{1}{N^2}\sum_{m,n=1}^{N}k_l(\mathbf{x}_m,\mathbf{x}_n)
\end{aligned} \tag{2.34}$$

Variance Normalization Kernels are normalized to have unit uniform variance in feature space, a procedure that should be applied after mean removal to achieve kernel standardization. For this condition to hold for feature space l, the following condition must be met:

$$\frac{1}{N}\sum_{i=1}^{N}\left\|\hat{\varphi}_l(\mathbf{x}_i) - \overline{\hat{\varphi}_l(\mathbf{x})}\right\|^2 = 1, \tag{2.35}$$

where $\overline{\hat{\varphi}_l(\mathbf{x})} = \frac{1}{N}\sum_k \hat{\varphi}(\mathbf{x}_k)$ is the mean of the training examples on feature space l for the rescaled feature map $\hat{\varphi}_l(\cdot)$. To achieve sample unit variance on feature space l, the rescaled feature map must satisfy

$$\hat{\varphi}_l(\mathbf{x}) = \frac{\varphi_l(\mathbf{x})}{\sqrt{Var}}, \tag{2.36}$$

where Var is the sample variance on feature space for feature map $\varphi_l(\cdot)$. Var is estimated as follows:

$$\begin{aligned}
Var &= \frac{1}{N}\sum_{i=1}^{N}\left[\varphi_l(\mathbf{x}_i) - \frac{1}{N}\sum_{k=1}^{N}\varphi_l(\mathbf{x}_k)\right]^T\left[\varphi_l(\mathbf{x}_i) - \frac{1}{N}\sum_{j=1}^{N}\varphi_l(\mathbf{x}_j)\right] \\
&= \frac{1}{N}\sum_{i=1}^{N}\Bigg\{\varphi_l^T(\mathbf{x}_i)\varphi_l(\mathbf{x}_i) - \frac{1}{N}\sum_{j=1}^{N}\varphi_l^T(\mathbf{x}_i)\varphi_l(\mathbf{x}_j) - \frac{1}{N}\sum_{k=1}^{N}\varphi_l^T(\mathbf{x}_k)\varphi_l(\mathbf{x}_i) \\
&\quad + \left(\frac{1}{N}\sum_{k=1}^{N}\varphi_l^T(\mathbf{x}_k)\right)\left(\frac{1}{N}\sum_{j=1}^{N}\varphi_l(\mathbf{x}_j)\right)\Bigg\} \\
&= \frac{1}{N}\sum_{i=1}^{N}\Bigg\{\varphi_l^T(\mathbf{x}_i)\varphi_l(\mathbf{x}_i) - \frac{1}{N}\sum_{j=1}^{N}\varphi_l^T(\mathbf{x}_i)\varphi_l(\mathbf{x}_j) \\
&\quad - \overline{\varphi_l(\mathbf{x})}^T\left[\varphi_l(\mathbf{x}_i) - \overline{\varphi_l(\mathbf{x})}\right]\Bigg\} \\
&= \frac{1}{N}\sum_{i=1}^{N}\varphi_l^T(\mathbf{x}_i)\varphi_l(\mathbf{x}_i) - \frac{1}{N^2}\sum_{i,j=1}^{N}\varphi_l^T(\mathbf{x}_i)\varphi_l(\mathbf{x}_j) \\
&\quad - \frac{1}{N}\,\overline{\varphi_l(\mathbf{x})}^T\underbrace{\left[\sum_{i=1}^{N}\varphi_l(\mathbf{x}_i) - N\,\overline{\varphi_l(\mathbf{x})}\right]}_{=0} \\
&= \frac{1}{N}\sum_{i=1}^{N}k_l(\mathbf{x}_i,\mathbf{x}_i) - \frac{1}{N^2}\sum_{i,j=1}^{N}k_l(\mathbf{x}_i,\mathbf{x}_j).
\end{aligned} \tag{2.37}$$

By using the expression (2.36) to estimate the inner product in feature space we get

$$\hat{\varphi}_l{}^T(\mathbf{x})\hat{\varphi}_l(\mathbf{x}^{'}) = \hat{k}_l(\mathbf{x},\mathbf{x}^{'}) = \frac{\varphi_l^T(\mathbf{x})\varphi_l(\mathbf{x}^{'})}{Var} = \frac{k_l(\mathbf{x},\mathbf{x}^{'})}{Var}. \tag{2.38}$$

Therefore, the normalization rule is given by

$$\hat{k}_l(\mathbf{x},\mathbf{x}^{'}) = \frac{k_l(\mathbf{x},\mathbf{x}^{'})}{\frac{1}{N}\sum_{i=1}^{N}k_l(\mathbf{x}_i,\mathbf{x}_i) - \frac{1}{N^2}\sum_{i,j=1}^{N}k_l(\mathbf{x}_i,\mathbf{x}_j)}. \tag{2.39}$$

Chapter 3

Proposed Classification Algorithms

3.1 Structure of the classifier

Let us assume that we are given N labeled training data $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. We also assume that features are divided in L blocks (subspaces) such that $\mathbb{R}^d = \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_L}$, so that each example $\mathbf{x}_i$ can be decomposed into these L blocks, i.e., $\mathbf{x}_i = [\mathbf{x}_{i,1}^T, \ldots, \mathbf{x}_{i,L}^T]^T$. Furthermore, let us assume that each vector $\mathbf{x}_{i,l}$ is mapped from its input space into a Hilbert space (feature space) via a mapping $\varphi_l : \mathbb{R}^{d_l} \to \mathcal{H}_l$. Thus,

$$\varphi(\mathbf{x}_i) = [\varphi_1^T(\mathbf{x}_{i,1}), \ldots, \varphi_L^T(\mathbf{x}_{i,L})]^T. \tag{3.1}$$

The goal of the proposed classification framework is to find a classifier that is a linear combination of a subset of blocks $\mathcal{I}_L$, that is,

$$f(\mathbf{x}_*) = \sum_{l \in \mathcal{I}_L} \mathbf{w}_l^T \varphi_l(\mathbf{x}_{*,l}) + b, \tag{3.2}$$

where $\mathbf{x}_*$ is a given test pattern and $\mathbf{w}_l \in \mathcal{H}_l$.

While the remainder of this chapter will continue the analysis of this algorithmic structure at an abstract level, we provide an illustration of the application of this

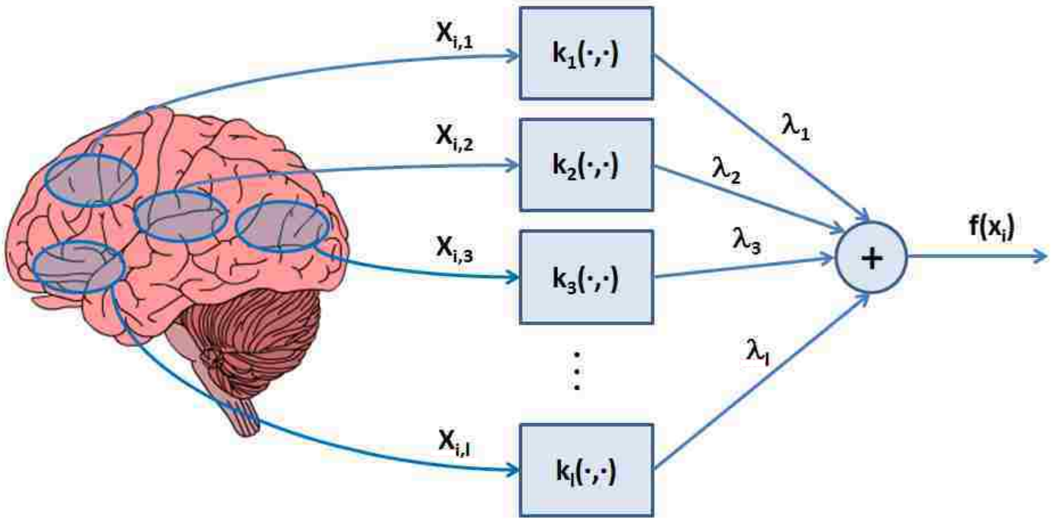


Figure 3.1: Structure of multiple-kernel based approach. Kernels are applied to data from different regions of the brain to estimate if a subject is either a control or a patient and to estimate the contribution of each region for this task.

framework to fMRI data for schizophrenia classification on Fig. 3.1. Let us assume that we are given spatial activation maps of the brain for a set of healthy controls and schizophrenia patients. The proposed framework would divide these maps into L brain regions, each region l being transformed to another representation by means of φ_l, or equivalently, by using a kernel function $k_l(\cdot,\cdot)$. The class associated to each subject is defined by a weighted combination of these kernels, where the coefficients assigned to these kernels indicate the amount of information provided by their associated regions to better discriminate both groups. A sparse selection of these regions is performed under the assumption that only some of them are actually informative to characterize schizophrenia.

In this chapter we present two methods that combine data from multiple kernels by using this framework. The first one, which is called recursive composite kernels, iteratively eliminates uninformative kernels by using a wrapper method based on SVM. Since this is a greedy algorithm, it is relatively fast. In addition, its implementation is relatively simple. The second method is ν-multiple kernel learning (ν-MKL), which is an SVM formulation that incorporates a sparse selection of kernels. While this algorithm is slower in terms of computation time, it is supposed to achieve a better performance than recursive composite kernels.

3.2 Optimization through a recursive composite kernels approach

3.2.1 Introduction

Recursive composite kernels (RCK) proposes to iteratively eliminate uninformative blocks based on the evaluation of the projection of weight vector $\mathbf{w}$ on each block. On its first iteration it uses the data from the whole set of blocks, i.e., $\mathcal{I}_L = \{1, 2, \ldots, L\}$. For the SVM case, the Representer Theorem [42, 8] states that the solution vector $\mathbf{w} = [\mathbf{w}_1^T, \ldots, \mathbf{w}_L^T]^T$ lies in the subspace spanned by training examples $\mathbf{x}_i$ in the feature space. Briefly,

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \varphi(\mathbf{x}_i). \tag{3.3}$$

By plugging Eq. 3.1 in Eq. 3.3, the following expression is obtained:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i [\varphi_1^T(\mathbf{x}_{i,1}), \ldots, \varphi_L^T(\mathbf{x}_{i,L})]^T. \tag{3.4}$$

From this equation, it can be seen that the projection of $\mathbf{w}$ over block l is $\mathbf{w}_l = \sum_{i=1}^{N} \alpha_i \varphi_l(\mathbf{x}_{i,l})$. By replacing $\mathbf{w}_l$ by this expression in Eq. 3.2 and including the entire set of blocks we get

$$\begin{aligned} f(\mathbf{x}_*) =& \sum_{l=1}^{L} \sum_{i=1}^{N} \alpha_i \varphi_l^T(\mathbf{x}_{i,l}) \varphi_l(\mathbf{x}_{*,l}) + b \\ =& \sum_{i=1}^{N} \alpha_i \sum_{l=1}^{L} k_l(\mathbf{x}_{i,l}, \mathbf{x}_{*,l}) + b, \end{aligned} \tag{3.5}$$

where $k_l(\mathbf{x}_{i,l}, \mathbf{x}_{j,l}) = \varphi_l^T(\mathbf{x}_{i,l})\varphi_l(\mathbf{x}_{j,l})$ is the kernel inner product of Hilbert space $\mathcal{H}_l$, as introduced in chapter 2.

Composite kernels, which have been used in [43, 44], usually apply kernels to different subspaces of the data input space that are linearly recombined in the feature space. RCK attempts to get rid of the blocks that show the least differential pattern activation between class groups, i.e., the least informative blocks. To do so, it proposes to iteratively eliminate one block at a time using a wrapper method based on SVM by analyzing the projection of $\mathbf{w}$ on each block. This approach is an extension of the method presented in [25]. The difference between both methods relies on the fact that the latter approach eliminates one feature and not one block at a time. For RCK, the SVM is trained with the information provided by the sum of kernels shown in Eq. 3.5.

3.2.2 Weight Vector Block Projection

Usually there is no inverse transformation to nonlinear transformations $\varphi(\cdot)$. Then, the spatial information that vector $\mathbf{w}$ may have cannot be retrieved. But by using composite kernels, each Hilbert space will hold all the properties of its associated block of the input space. That way, a straightforward analysis can provide information about that block. If a particular block of the input space contains no information relevant to the classification task, then vector $\mathbf{w}$ will tend to be orthogonal to this subspace. On the contrary, if it contains relevant information, then the weight vector

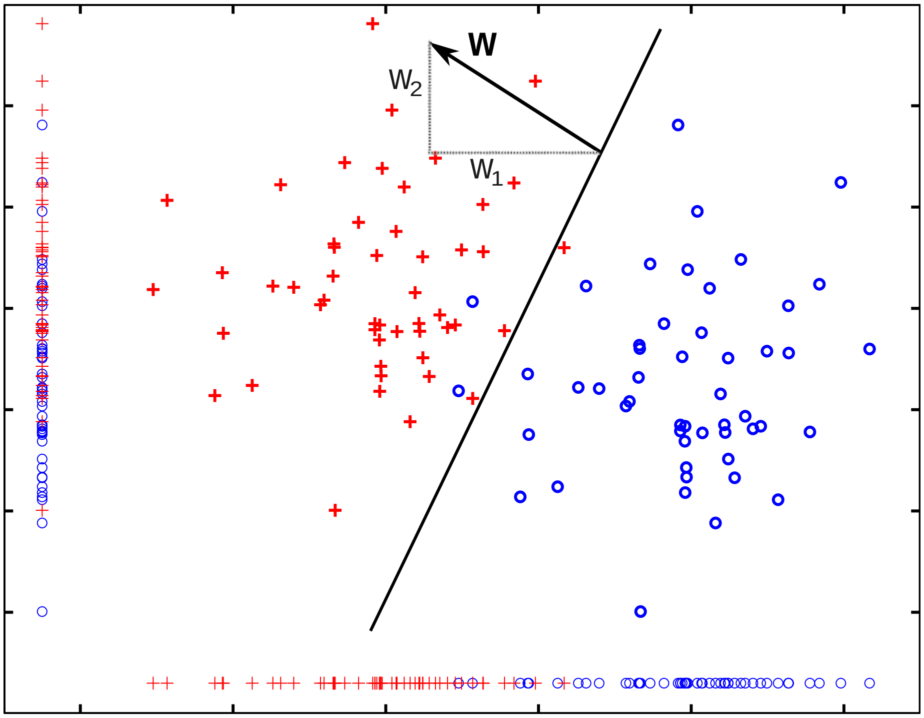


Figure 3.2: The projections of the weight vector on dimension (block) 1 (x-axis) and dimension 2 (y-axis) on a 2-dimensional classification problem. Block 1 offers more information to discriminate both classes as seen in the projections of the examples on each block, which is the reason why $\|\mathbf{w}_1\| > \|\mathbf{w}_2\|$. Nonetheless, block 2 is also informative, thus explaining why $\mathbf{w}$ is not orthogonal to it.

will tend to be parallel to this subspace. Fig. 3.2 shows a 2-dimensional classification problem that illustrates this point. Dimension 1 of the input space, which in this case represents a block, is more informative than block 2 to discriminate both class labels. For this reason, $\|\mathbf{w}_1\|$ is greater than $\|\mathbf{w}_2\|$. However, since block 2 also provides relevant information, $\mathbf{w}$ is not perpendicular to this block.

Eq. 3.4 specifies how to estimate the projection of $\mathbf{w}$ on all blocks. However, these vectors may not be accessible. Nonetheless, the quadratic norms of vectors $\mathbf{w}_l$

can be computed as follows:

$$\begin{aligned} ||\mathbf{w}_l||^2 &= \mathbf{w}_l^T \mathbf{w}_l = \\ &= \sum_{i=1}^{N} \alpha_i \varphi_l^T(\mathbf{x}_{i,l}) \sum_{j=1}^{N} \varphi_l(\mathbf{x}_{j,l}) \alpha_j \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i k_l(\mathbf{x}_{i,l}, \mathbf{x}_{j,l}) \alpha_j . \end{aligned} \tag{3.6}$$

Let $\mathbf{K}_l$ be an $N \times N$ matrix whose component i, j is computed as $\mathbf{K}_l(i, j) = k_l(\mathbf{x}_{i,l}, \mathbf{x}_{j,l})$ and $\boldsymbol{\alpha}$ be a vector composed of all parameters α_i. Then the quadratic norm of $\mathbf{w}_l$ can be expressed in matrix form as

$$\|\mathbf{w}_l\|^2 = \boldsymbol{\alpha}^T \mathbf{K}_l \boldsymbol{\alpha}. \tag{3.7}$$

This procedure can be viewed as the projection of vector $\boldsymbol{\alpha}$ into the principal components of matrix $\mathbf{K}_l$. The relevance of space l is then approximated by the similarity of $\boldsymbol{\alpha}$ to these vectors.

The previous equation provides a metric to evaluate the relevance of a certain block in the classification task, which will be called discriminative weight from now on. Furthermore, this metric makes it possible to detect the least relevant block, thus providing RCK information of which block should be removed at each iteration.

3.2.3 Recursive Algorithm

There is only one piece of information missing to completely define an RCK algorithm, and this is how to find the optimal block set. The most informative block set is the one that achieves the minimum error rate across the RCK iterations on a validation set, i.e., a dataset that is independent from the one used for training purposes.

With the above elements, the RCK algorithm can be constructed. Initially, an SVM is trained using the training set with the sum of the kernels from all the blocks as explained on section 3.2.1, after which the block with smallest associated discrimination weight is removed from the initial block set. At the next iteration, the SVM is trained with the data from all the blocks but the previously removed one and their discriminative weights are recalculated, eliminating the block with current minimum weight. This procedure is applied iteratively until a single block remains in the analyzed set of blocks, with the optimal block set $\mathcal{I}_L$ being the one that achieves the lowest validation error rate across the iterations of the recursive algorithm. Algorithm 1 summarizes the described procedure.

Algorithm 1 RCK Algorithm

```
**Inputs**: $TrainSet$, $ValidSet$
**Outputs**: $I_L$
**Define** $I(1)$: indexes for all blocks
**Define** $P$: number of blocks
**for** $p = 1$ to $P - 1$ **do**
    **TrainSVM**(**SumKernels**($TrainSet$,$I(p)$)) $\Rightarrow TrainedSVM$
    Compute discriminative weights
    **TestSVM**($TrainedSVM$,$ValidSet$) $\Rightarrow E(p)$
    Remove area with lowest weight
    Store indexes of remaining blocks $\Rightarrow I(p+1)$
**end for**
Find $p$ that minimizes $E(p) \Rightarrow p_{min}$
$I(p_{min}) \Rightarrow I_L$
```

3.3 Optimization through a sparse MKL approach

3.3.1 Introduction

The proposed MKL algorithm generates a sparse selection of features' subsets (block-sparse selection) by using a ν-SVM formulation [45], where ν defines the upper bound of the fraction of blocks to be selected.

Gómez-Verdejo et al. [46] proposed a ν-SVM formulation for the linear case that forced a sparse selection of features. ν-MKL is an extension of this work, whose aim is to attain block sparsity and generate a classifier that linearly combines feature subspaces, the difference being that these blocks can be mapped into arbitrarily higher dimensional spaces, i.e., ν-MKL is not restricted to be a linear classifier.

3.3.2 SVM with Block-Sparsity Constraints

Recall from chapter 2 that the SVM optimization problem can be expressed by

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\left\|\mathbf{w}\right\|^2 + C\sum_{i=1}^{N}\xi_i & \\ s.t. \quad & y_i\left(\mathbf{w}^T\varphi(\mathbf{x}_i)+b\right) \geq 1-\xi_i & \forall i \\ & \xi_i \geq 0 & \forall i, \end{aligned} \tag{3.8}$$

If features are partitioned in L blocks as it has been specified at the beginning of the chapter, then the weight vector could also be split into these blocks such that $\mathbf{w} = [\mathbf{w}_1^T, \ldots, \mathbf{w}_L^T]^T$, thus satisfying $\left\|\mathbf{w}\right\|^2 = \sum_{l=1}^{L}\left\|\mathbf{w}_l\right\|^2$. In order to attain block sparsity, additional constraints that upper bound the l_2-norm of $\mathbf{w}_l$ would need to be included in the formulation. By adding these constraints, we get this modified formulation:

$$
\begin{aligned}
\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\gamma}} \quad & \frac{1}{2}\sum_{l=1}^{L}\|\mathbf{w}_l\|^2 + C\sum_{i=1}^{N}\xi_i + \frac{C'}{L}\sum_{l=1}^{L}\gamma_l \\
s.t. \quad & y_i\left(\sum_{l=1}^{L}\mathbf{w}_l^T\varphi_l(\mathbf{x}_{i,l}) + b\right) \geq 1-\xi_i && \forall i \\
& \xi_i \geq 0 && \forall i \\
& \|\mathbf{w}_l\| \leq \epsilon + \gamma_l && \forall l \\
& \gamma_l \geq 0 && \forall l.
\end{aligned}
\tag{3.9}
$$

This optimization problem includes a small parameter ϵ and slack variables γ_l in the upper bound terms of the inequalities associated to $\|\mathbf{w}_l\|$, where the ones being strictly greater than zero are associated with relevant feature blocks after the functional optimization. Conversely, blocks l such that $\|\mathbf{w}_l\| \leq \epsilon$ are deemed irrelevant and are discarded. A new cost term that is composed of the summation of slack variables γ_l weighted by a tradeoff parameter C' is included in the formulation, a larger C' corresponding to assigning a higher penalty to relevant blocks.

A final modification is introduced to the proposed formulation in Eq. 3.9 to automatically adjust the value of ϵ by following the ν-SVM proposed in [45], where $\nu \in (0,1]$. The resulting optimization problem is given by:

$$
\begin{aligned}
\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\gamma},\epsilon} \quad & \frac{1}{2}\sum_{l=1}^{L}\|\mathbf{w}_l\|^2 + C\sum_{i=1}^{N}\xi_i + C'\left[\nu\epsilon + \frac{1}{L}\sum_{l=1}^{L}\gamma_l\right] \\
s.t. \quad & y_i\left(\sum_{l=1}^{L}\mathbf{w}_l^T\varphi_l(\mathbf{x}_{i,l}) + b\right) \geq 1-\xi_i && \forall i \\
& \xi_i \geq 0 && \forall i \\
& \|\mathbf{w}_l\| \leq \epsilon + \gamma_l && \forall l \\
& \gamma_l \geq 0 && \forall l \\
& \epsilon \geq 0.
\end{aligned}
\tag{3.10}
$$

It will be demonstrated later on this chapter that ν defines the upper bound of the fraction of relevant blocks.

The dual formulation of this algorithm is analyzed under the assumption that the data is limited and significantly smaller than its dimensionality. If the assumption is met, a reduced execution time could be achieved to optimize this problem, as it would depend on the number of training observations. This formulation is then reduced to a second-order cone program (SOCP), which requires the validation of the parameters C, C' and ν.

3.3.3 Dual Problem

In this section, we show how a dual problem of the proposed formulation can lead to a SOCP. In order to be able to do so, a definition of a SOC on a composite Hilbert space of interest is provided first.

Second-order cone on a composite Hilbert space

Given a Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle)$, a SOC $\mathcal{K} \subset V$, where V is a composite Hilbert space such that $V = \mathbb{R} \times \mathcal{H}$, can be defined as follows:

$$\mathcal{K} = \{(t, \mathbf{x}) \in \mathbb{R} \times \mathcal{H} : t \geq \|x\|\}, \tag{3.11}$$

where $\|x\| = \sqrt{\langle x, x \rangle}$ and $\mathcal{K}$ is self-dual, i.e., the dual $\mathcal{K}^*$ of $\mathcal{K}$ coincides with $\mathcal{K}$ [47].

Dual Lagrangian derivation

As defined before, $\mathbf{w}_l \in \mathcal{H}_l$. Let $t_l \in \mathbb{R}$ and $\|\mathbf{w}_l\| \leq t_l \leq \epsilon + \gamma_l$. Then, $(t_l, \mathbf{w}_l) \in \mathcal{K}_l$, where $\mathcal{K}_l \subset V_l = \mathbb{R} \times \mathcal{H}_l$ is a SOC in the composite Hilbert space V_l. Thus, Eq. 3.10

can be restated as follows:

$$\begin{aligned}
\min_{\mathbf{w},\mathbf{t},b,\boldsymbol{\xi},\boldsymbol{\gamma},\epsilon} \quad & \frac{1}{2}\sum_{l=1}^{L} t_l^2 + C\sum_{i=1}^{N}\xi_i + C^{'}\left[\nu\epsilon + \frac{1}{L}\sum_{l=1}^{L}\gamma_l\right] & \\
s.t. \quad & y_i\left(\sum_{l=1}^{L}\mathbf{w}_l^T\varphi_l(\mathbf{x}_{i,l}) + b\right) \geq 1 - \xi_i & \forall i \\
& \xi_i \geq 0 & \forall i \\
& t_l \leq \epsilon + \gamma_l & \forall l \\
& (t_l, \mathbf{w}_l) \in \mathcal{K}_l & \forall l \\
& \gamma_l \geq 0 & \forall l \\
& \epsilon \geq 0. &
\end{aligned} \tag{3.12}$$

Since $\mathcal{K}_l$ is self-dual, the primal Lagrangian corresponding to the problem is

$$\begin{aligned}
L_P \equiv \; & \frac{1}{2}\sum_{l=1}^{L} t_l^2 + C\sum_{i=1}^{N}\xi_i + C^{'}\nu\epsilon + \frac{C^{'}}{L}\sum_{l=1}^{L}\gamma_l \\
& - \sum_{i=1}^{N}\alpha_i\left[y_i\sum_{l=1}^{L}\mathbf{w}_l^T\varphi_l(\mathbf{x}_{i,l}) + y_i b - 1 + \xi_i\right] - \sum_{i=1}^{N}\mu_i\xi_i \\
& - \sum_{l=1}^{L}\beta_l\left(\epsilon + \gamma_l - t_l\right) - \sum_{l=1}^{L}\left(\mathbf{w}_l^T\boldsymbol{\sigma}_l + \theta_l t_l\right) - \sum_{l=1}^{L}\tau_l\gamma_l - \delta\epsilon \\
with \quad & \alpha_i \geq 0 \qquad \forall i \\
& \mu_i \geq 0 \qquad \forall i \\
& (\theta_l, \boldsymbol{\sigma}_l) \in \mathcal{K}_l \qquad \forall l \\
& \beta_l \geq 0 \qquad \forall l \\
& \tau_l \geq 0 \qquad \forall l \\
& \delta \geq 0
\end{aligned} \tag{3.13}$$

where $\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\beta}, \boldsymbol{\tau}$ and δ are Lagrange multipliers (dual variables). Next, the partial derivatives with respect to the primal variables are computed and set to zero.

$$\begin{aligned}
\frac{\partial L_P}{\partial t_l} &: \quad t_l + \beta_l - \theta_l = 0 \quad \Leftrightarrow \quad \theta_l = t_l + \beta_l \\
\frac{\partial L_P}{\partial \mathbf{w}_l} &: \quad -\sum_{i=1}^{N} \alpha_i y_i \varphi_l(\mathbf{x}_{i,l}) - \boldsymbol{\sigma}_l = 0 \quad \Leftrightarrow \quad \boldsymbol{\sigma}_l = -\sum_{i=1}^{N} \alpha_i y_i \varphi_l(\mathbf{x}_{i,l}) \\
\frac{\partial L_P}{\partial \xi_i} &: \quad C - \mu_i - \alpha_i = 0. \ \textit{Since } \mu_i, \alpha_i \geq 0 \ \Rightarrow \ 0 \leq \alpha_i \leq C \\
\frac{\partial L_P}{\partial b} &: \quad -\sum_{i=1}^{N} \alpha_i y_i = 0 \\
\frac{\partial L_P}{\partial \epsilon} &: \quad C^{'}\nu - \delta - \sum_{l=1}^{L} \beta_l = 0. \ \textit{Since } \delta, \beta_l \geq 0 \ \Rightarrow \ 0 \leq \sum_{l=1}^{L} \beta_l \leq C^{'}\nu \\
\frac{\partial L_P}{\partial \gamma_l} &: \quad \frac{C^{'}}{L} - \tau_l - \beta_l = 0. \ \textit{Since } \tau_l, \beta_l \geq 0 \ \Rightarrow \ 0 \leq \beta_l \leq \frac{C^{'}}{L}.
\end{aligned} \tag{3.14}$$

By replacing in Eq. 3.13 the expressions obtained in Eq. 3.14 the following dual Lagrangian function is obtained:

$$\begin{aligned}
L_D \ \equiv \ & -\frac{1}{2}\sum_{l=1}^{L} t_l^2 + \sum_{i=1}^{N} \alpha_i \\
\textit{with} \quad & 0 \leq \alpha_i \leq C && \forall i \\
& t_l \geq 0 && \forall l \\
& 0 \leq \beta_l \leq \frac{C^{'}}{L} && \forall l \\
& \left\| \sum_{i=1}^{N} \alpha_i y_i \varphi_l(\mathbf{x}_{i,l}) \right\| \leq t_l + \beta_l && \forall l \\
& \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& 0 \leq \sum_{l=1}^{L} \beta_l \leq C^{'}\nu,
\end{aligned} \tag{3.15}$$

where maximizing L_D with respect to the dual variables is equivalent to minimizing L_P with respect to the primal variables.

Proof of block sparsity and upper bound enforcement

Proposition 1. *Let L be the number of feature subspaces on which the input space is partitioned. If ν-MKL is provided with N labeled training data $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, it achieves block sparsity, that is, $\gamma_l > 0 \;\; \forall l \in \mathcal{I}_L \subseteq \{1, 2, \ldots, L\}$.*

Proposition 2. *Let n be the number of relevant blocks detected by ν-MKL. Then ν is an upper bound of the fraction of blocks that are deemed relevant. In brief, $n/L \leq \nu$.*

Proof. In order to verify that the presented formulation achieves block sparsity and it is capable of defining an upper bound to the number of relevant blocks through the parameter ν, it is necessary to examine some of its KKT complementarity conditions. They are the following:

$$\begin{aligned} &\beta_l(\epsilon + \gamma_l - t_l) = 0 && \forall l \\ &\left(\frac{C'}{L} - \beta_l\right)\gamma_l = 0 && \forall l \\ &\begin{pmatrix} t_l \\ \mathbf{w}_l \end{pmatrix}^T \begin{pmatrix} t_l + \beta_l \\ -\sum_i \alpha_i y_i \varphi_l(\mathbf{x}_{i,l}) \end{pmatrix} = 0 && \forall l. \end{aligned} \tag{3.16}$$

Recall that $(t_l, \mathbf{w}_l) \in \mathcal{K}_l$ ($\|\mathbf{w}_l\| \leq t_l$). Before performing an analysis of the previous equations, it is necessary to know under which conditions the product of two elements of a SOC equal zero, as specified in Eq. 3.16.

Let $(t, \mathbf{x}),\ (t', \mathbf{x}') \in \mathcal{K}$. The product $\binom{t}{\mathbf{x}}^T \binom{t'}{\mathbf{x}'} = 0$ holds if and only if either of these two conditions is met:

(a) One or both factors of the product are zero.

(b) Both factors are nonzero, belong to the boundary of $\mathcal{K}$, and are anti-proportional; i.e., $\exists \eta > 0$ such that $\|\mathbf{x}\| = t,\ \|\mathbf{x}'\| = t'$, and $(t, \mathbf{x}) = \eta(t', -\mathbf{x}')$ [30].

By analyzing the complementarity conditions, it is possible to know what the values of $\|\mathbf{w}_l\|$ are for different values of the variables β_l. The values of $\|\mathbf{w}_l\|$ (more specifically the values of γ_l) indicate which blocks l are deemed relevant by the classifier. In addition, variables β_l are important on their own right since their summation is upper bounded by a multiple of ν, as it is specified in Eq. 3.15. The conditions in Eq. 3.16 are analyzed as follows:

i. If $\beta_l = 0 \quad \Rightarrow \gamma_l = 0 \ \Rightarrow \epsilon - t_l > 0 \ \Rightarrow t_l < \epsilon \ \Rightarrow \|\mathbf{w}_l\| \leq \epsilon$

ii. If $0 < \beta_l < \frac{C^{'}}{L} \Rightarrow \gamma_l = 0 \ \Rightarrow \epsilon - t_l = 0 \ \Rightarrow t_l = \epsilon$

- If $\epsilon > 0 \quad \Rightarrow$ since $\epsilon = t_l > 0$ and $\beta_l + t_l > 0 \ \Rightarrow \|\mathbf{w}_l\| = t_l = \epsilon$
- If $\epsilon = 0 \quad \Rightarrow \epsilon = t_l = 0 \ \Rightarrow \|\mathbf{w}_l\| = t_l = 0$

iii. If $\beta_l = \frac{C^{'}}{L} \quad \Rightarrow \gamma_l > 0 \ \Rightarrow \epsilon + \gamma_l - t_l = 0 \ \Rightarrow t_l = \gamma_l + \epsilon$. Since $t_l \geq \gamma_l > 0$ and $\beta_l + t_l > 0 \ \Rightarrow \|\mathbf{w}_l\| = t_l = \gamma_l + \epsilon$.

As it has been mentioned before, only the blocks $l : \gamma_l > 0$ are relevant and these ones, in turn, have an associated $\beta_l = \frac{C^{'}}{L}$. It has been shown that $\mathcal{I}_L = \{l : \gamma_l > 0\} \subseteq \{1, 2, \ldots, L\}$, which proves that the algorithm achieves block sparsity. In addition, if n blocks are relevant, $\exists p \geq 0$ such that:

$$\sum_{l=1}^{L} \beta_l = \frac{nC^{'}}{L} + p \leq C^{'}\nu \Rightarrow \frac{nC^{'}}{L} \leq C^{'}\nu \Rightarrow \frac{n}{L} \leq \nu. \tag{3.17}$$

Thus, it has also been proven that ν is an upper bound of the fraction of blocks that are relevant. □

Conic linear program formulation

A conic linear program (LP) is an LP with the additional constraint that the solution needs to lie in a convex cone. A conic LP has the form

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T\mathbf{x} \\ s.t. \quad & \mathbf{l}_c \leq \mathbf{A}\mathbf{x} \leq \mathbf{u}_c \\ & \mathbf{l}_x \leq \mathbf{x} \leq \mathbf{u}_x \\ & \mathbf{x} \in \mathcal{C}, \end{aligned} \tag{3.18}$$

where $\mathcal{C}$ is a convex cone. This cone can be expressed as the Cartesian product of p convex cones as $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_p$, in which case $\mathbf{x} \in \mathcal{C}$ could be written as $\mathbf{x} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_p^T]^T$, $\mathbf{x}_1 \in \mathcal{C}_1, \ldots, \mathbf{x}_p \in \mathcal{C}_p$. It should be highlighted that the d-dimensional Euclidean space $\mathbb{R}^d$ is a cone itself, so linear variables also comply with the added constraint [48].

A SOCP is a conic LP where the cone constraints are defined by SOCs. It can be seen that the problem of maximizing Eq. 3.15 is not a SOCP since there are quadratic terms in both the objective function and the constraints. The problem needs some algebraic manipulation for it to become a SOCP.

The term $\left\|\sum_{i=1}^N \alpha_i y_i \varphi_l(\mathbf{x}_{i,l})\right\|$, which is quadratic on $\boldsymbol{\alpha}$, needs to be rearranged in order to make the proposed problem a SOCP. This term can be expressed as

$$\begin{aligned} \left\|\sum_{i=1}^N \alpha_i y_i \varphi_l(\mathbf{x}_{i,l})\right\| &= \sqrt{\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \varphi_l^T(\mathbf{x}_{i,l}) \varphi_l(\mathbf{x}_{j,l})} \\ &= \sqrt{\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k_l(\mathbf{x}_{i,l}, \mathbf{x}_{j,l})}, \end{aligned} \tag{3.19}$$

where $k_l(\mathbf{x}_{i,l}, \mathbf{x}_{j,l}) = \varphi_l^T(\mathbf{x}_{i,l})\varphi_l(\mathbf{x}_{j,l})$ is the (symmetric) kernel inner product of Hilbert space $\mathcal{H}_l$. Let $\mathbf{K}_l$ be an $N \times N$ matrix whose component i, j is computed as $\mathbf{K}_l(i, j) =$

$k_l(\mathbf{x}_{i,l}, \mathbf{x}_{j,l})$. Then the quadratic term on $\boldsymbol{\alpha}$ can be expressed in matrix notation as

$$\left\| \sum_{i=1}^{N} \alpha_i y_i \varphi_l(\mathbf{x}_{i,l}) \right\| = \sqrt{\boldsymbol{\alpha}^T \mathbf{Y}\mathbf{K}_l\mathbf{Y}\boldsymbol{\alpha}} = \sqrt{\boldsymbol{\alpha}^T \mathbf{H}_l \boldsymbol{\alpha}}, \tag{3.20}$$

where $\mathbf{H}_l = \mathbf{Y}\mathbf{K}_l\mathbf{Y}$ and $\mathbf{Y}$ is an $N \times N$ diagonal matrix such that $\mathbf{Y}(i,i) = y_i$. Since $\mathbf{K}_l$ is a Gram matrix, it is positive semidefinite. In addition, it is symmetric. As a consequence, $\mathbf{H}_l$ is symmetric positive semidefinite, so there $\exists \mathbf{F}_l : \mathbf{F}_l^T \mathbf{F}_l = \mathbf{H}_l$.[1] Thus,

$$\left\| \sum_{i=1}^{N} \alpha_i y_i \varphi_l(\mathbf{x}_{i,l}) \right\| = \sqrt{(\boldsymbol{\alpha}^T \mathbf{F}_l^T)(\mathbf{F}_l \boldsymbol{\alpha})} = \left\| \mathbf{F}_l \boldsymbol{\alpha} \right\|. \tag{3.21}$$

By replacing the obtained expression on Eq. 3.15 and writing the formulation in matrix notation we get

$$\begin{aligned}
\min_{\mathbf{t},\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & \frac{1}{2}\left\|\mathbf{t}\right\|^2 - \mathbf{1}^T\boldsymbol{\alpha} \\
s.t. \quad & \left\|\mathbf{F}_l\boldsymbol{\alpha}\right\| \leq t_l + \beta_l \qquad \forall l \\
& 0 \leq \boldsymbol{\alpha} \leq C \\
& \boldsymbol{\alpha}^T\mathbf{y} = 0 \\
& 0 \leq \boldsymbol{\beta} \leq \frac{C'}{L} \\
& 0 \leq \mathbf{1}^T\boldsymbol{\beta} \leq C'\nu \\
& \mathbf{t} \geq 0.
\end{aligned} \tag{3.22}$$

It can be seen that the quadratic constraint is now defined by a SOC. However, the unknowns (and not a linear transformation of them) are the ones that must be members of a cone, as defined by Eq. 3.18. Let $u_l = t_l + \beta_l$ and $\mathbf{z}_l = \mathbf{F}_l\boldsymbol{\alpha}$. Then the

[1] The details of the estimation of $\mathbf{F}_l$ are provided in Appendix A.

problem could be restated as

$$\begin{aligned}
\min_{\mathbf{t},\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{u},\mathbf{z}} \quad & \frac{1}{2}\left\|\mathbf{t}\right\|^2 - \mathbf{1}^T\boldsymbol{\alpha} \\
s.t. \quad & \|\mathbf{z}_l\| \leq u_l \qquad \forall l \\
& u_l - t_l - \beta_l = 0 \quad \forall l \\
& \mathbf{F}_l\boldsymbol{\alpha} - \mathbf{z}_l = 0 \qquad \forall l \\
& 0 \leq \boldsymbol{\alpha} \leq C \\
& \boldsymbol{\alpha}^T\mathbf{y} = 0 \\
& 0 \leq \boldsymbol{\beta} \leq \frac{C^{'}}{L} \\
& 0 \leq \mathbf{1}^T\boldsymbol{\beta} \leq C^{'}\nu \\
& \mathbf{t} \geq 0.
\end{aligned} \tag{3.23}$$

At this point, the problem has been restated so that all the unknowns lie in convex cones. All that remains to be done are algebraic manipulations so that the objective function becomes linear, thus meeting all the requirements of a conic LP.

Let $\frac{1}{2}\left\|\mathbf{t}\right\|^2 \leq s$, where $s \geq 0$. If we define $r = 1$, then $\left\|\mathbf{t}\right\|^2 \leq 2rs$. By substituting this expression on Eq. 3.23 we get

$$\begin{aligned}
\min_{\mathbf{t},\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{u},\mathbf{z},s,r} \quad & s - \mathbf{1}^T\boldsymbol{\alpha} & \\
s.t. \quad & \|\mathbf{z}_l\| \leq u_l & \forall l \\
& u_l - t_l - \beta_l = 0 & \forall l \\
& \mathbf{F}_l\boldsymbol{\alpha} - \mathbf{z}_l = 0 & \forall l \\
& \boldsymbol{\alpha}^T\mathbf{y} = 0 & \\
& 0 \leq \boldsymbol{\alpha} \leq C & \\
& 0 \leq \boldsymbol{\beta} \leq \frac{C^{'}}{L} & \\
& 0 \leq \mathbf{1}^T\boldsymbol{\beta} \leq C^{'}\nu & \\
& \|\mathbf{t}\|^2 \leq 2rs & \\
& r = 1 & \\
& s \geq 0 & \\
& \mathbf{t} \geq 0, &
\end{aligned} \tag{3.24}$$

where expression $\|\mathbf{t}\|^2 \leq 2rs$ defines a rotated SOC [48]. The problem defined on Eq. 3.24 characterizes the problem as a SOCP, having the same form as the canonical conic LP formulation shown in Eq. 3.18.

3.3.4 Class Prediction

On Section 3.3.3 it was stated that two nonzero elements $(t, \mathbf{x}), (t^{'}, \mathbf{x}^{'}) \in \mathcal{K}$ are perpendicular to each other if and only if $\|\mathbf{x}\| = t$, $\left\|\mathbf{x}^{'}\right\| = t^{'}$, and $(t, \mathbf{x}) = \eta(t^{'}, -\mathbf{x}^{'})$, where $\eta > 0$. In addition, it was also demonstrated that relevant blocks l have an associated parameter $\beta_l = C^{'}/L$. If the first proposition is applied to Eq. 3.16

$\forall l \in \mathcal{I}_\beta = \{l : \beta_l = C^{'}/L\}$, then

$$t_l = \eta_l(t_l + \beta_l) \tag{3.25a}$$

$$\mathbf{w}_l = \eta_l \sum_{i=1}^{N} \alpha_i y_i \varphi_l(\mathbf{x}_{i,l}). \tag{3.25b}$$

The estimated class of an unknown example $\mathbf{x}_*$, as specified at the beginning of this chapter, should be defined by a subset of blocks $\mathcal{I}_L$ such that $f(\mathbf{x}_*) = \sum_{l \in \mathcal{I}_L} \mathbf{w}_l^T \varphi_l(\mathbf{x}_{*,l}) + b$. Since the relevant subset of blocks is defined by $\mathcal{I}_\beta$, then $f(\mathbf{x}_*) = \sum_{l \in \mathcal{I}_\beta} \mathbf{w}_l^T \varphi_l(\mathbf{x}_{*,l}) + b$. By replacing Eq. 3.25b on this equation we get

$$\begin{aligned} f(\mathbf{x}_*) &= \sum_{i=1}^{N} \alpha_i y_i \sum_{l \in \mathcal{I}_\beta} \eta_l \varphi_l^T(\mathbf{x}_{i,l}) \varphi_l(\mathbf{x}_{*,l}) + b \\ &= \sum_{i=1}^{N} \alpha_i y_i \sum_{l \in \mathcal{I}_\beta} \eta_l k_l(\mathbf{x}_{i,l}, \mathbf{x}_{*,l}) + b. \end{aligned} \tag{3.26}$$

Once the SOCP is solved, $\eta_l : l \in \mathcal{I}_\beta$ can be calculated directly from Eq. 3.25a. The only variable that needs to be found to fully define Eq. 3.26 is b.

Let $\mathcal{I}_\alpha = \{i : 0 < \alpha_i < C\}$. It can be proven from the KKT conditions of the primal problem (Eq. 3.13) along with Eq. 3.26 that $\forall i \in \mathcal{I}_\alpha$ the following equality holds:

$$\begin{aligned} 1 &= y_i f(\mathbf{x}_i) \\ &= y_i \sum_{j=1}^{N} \alpha_j y_j \sum_{l \in \mathcal{I}_\beta} \eta_l k_l(\mathbf{x}_{j,l}, \mathbf{x}_{*,l}) + b. \end{aligned} \tag{3.27}$$

After some algebraic manipulation we get

$$b = y_i - \sum_{l \in \mathcal{I}_\beta} \eta_l \sum_{j=1}^{N} k_l(\mathbf{x}_{i,l}, \mathbf{x}_{j,l}) \alpha_j y_j \qquad \forall i \in \mathcal{I}_\alpha. \tag{3.28}$$

While b can be estimated by using Eq. 3.28 for any $i \in \mathcal{I}_\alpha$, it is numerically safer to take the mean value of b across all such values of i [17].

Chapter 4

Application of RCK and ν-MKL to simulated data

As it was mentioned on the previous chapter, by integrating a sparse selection of kernels on its formulation, ν-MKL is supposed to achieve a better performance than RCK at the expense of having a slower execution time. In order to verify the validity of this statement, both algorithms have to be tested on a dataset whose ground truth is already known, i.e., one where the information present on each block for classification purposes is well characterized.

This chapter presents the results obtained by RCK and ν-MKL on a simulated fMRI data set that mimics the BOLD response of two groups of subjects to an auditory oddball discriminant (AOD) task. These two groups are characterized so that their fMRI responses represent (to a certain extent) that of healthy controls and schizophrenia patients. To do so, differential activity between both groups is generated on brain regions where there is evidence of abnormal activation patterns on schizophrenia.

The organization of this chapter is explained as follows. Section 4.1 gives an overview of the toolbox used to generate fMRI simulated data. Section 4.2 explains the criteria used to generate data from an AOD task and to simulate the differential activity between groups for certain brain regions. Section 4.3 explains the data analysis applied to the simulated fMRI data, whose output is provided to RCK and ν-MKL. Section 4.4 shows the results obtained by both classification algorithms. Finally, section 4.5 provides a brief discussion of these results.

4.1 SimTB Toolbox

SimTB [49] is a simulation toolbox that runs on MATLAB (The Mathworks, Inc.) and allows for flexible generation of fMRI data under the model of spatiotemporal separability, which is consistent with the assumptions of ICA. This toolbox provides the user control over data generation including the creation and manipulation of spatial sources, implementation of block- and event-related experimental designs, inclusion of tissue-specific baselines, simulated head movement, and more.

Under the assumptions of spatiotemporal separability, data can be expressed as the product of time courses (TCs) and spatial maps (SMs), as shown in Fig. 4.1. Specifically, for each subject $i = 1, \ldots, M$, it is assumed that there are up to C components, each consisting of a SM, a TC of activation and an amplitude. The no-noise (nn) data is a linear combination of amplitude-scaled and baseline-shifted TC and SM components, which yields a time-by-voxel ($T \times V$) no-noise data for subject i.

A template of the 30 default SMs is shown in Fig. 4.2(a) on a square image of $V = \sqrt{V} \times \sqrt{V}$ voxels, where side length $\sqrt{V}$ is specified by the user. Default SMs are modeled after components commonly seen in axial slices of real fMRI data, being most of them created by combinations of Gaussian distributions. Users can vary the location and orientation of these activation blobs across subjects. The spatial extent of the SMs can be varied with the "spread" parameter ρ. SMs are normalized to have a maximum intensity of 1 and are transformed as $S^{'}_{ic} = S^{1/\rho}_{ic}$, where ρ describes the expansion ($\rho > 1$) or contraction ($\rho < 1$) of the component and $S^{'}_{ic}$ is the modified SM for subject i. Finally, a little Gaussian noise distributed as $\mathcal{N}(0, 2.5 \times 10^{-5})$ is added so that each subject SM is unique.

Each component TC is T time points in length, where the user specifies the repetition time (TR) in seconds per sample. TCs are constructed under the assumption that component activations result from underlying neural events as well as noise. Neural events can follow block- or event-related experimental designs, or can represent unexplained, random deviations from baseline. An underlying event time series is referred to as TS to distinguish it from the subsequent TC that is created with a hemodynamic model.

Experimental paradigms are designed with task blocks and task events that can be assigned to several components and can be identical across subjects. Unique events refer to unexplained deviations that are unique to each component and subject. These three types of TS inputs are controlled independently. Each task block is described by a block length and an inter-stimulus interval. Task events and unique events are defined by a probability of occurrence at each TR. For a given component, the TS is created by adding together coefficient-modulated task blocks, task events and unique events, as it is shown in Fig. 4.1. Coefficients for task inputs can be negative or positive (indicating suppression or activation with the task), or can be zero (indicating that component activation does not follow the task).

Generating the fMRI BOLD-like TCs from the event TS may be done in several ways, including linear convolution with a canonical hemodynamic response function (HRF) (difference of two gamma functions) [50] and the Windkessel balloon model [51]. Users may vary hemodynamic parameters between components and subjects, and define their own TC source models. After creation of the TCs, each component

TC is scaled to have a peak-to-peak range of one. As with the SMs, Gaussian noise distributed $\mathcal{N}(0, 2.5 \times 10^{-5})$ is added to ensure non-zero TCs.

A baseline intensity b_i is specified for each subject and an optional tissue-type modifier scales the baseline for each voxel. Tissue types with different intensity levels are assigned to each component. For example, Figure 4.2(b) displays the baseline intensity map where four tissue types are defined: sinus signal dropout, cerebrospinal fluid (CSF), white matter and gray matter.

Finally, motion (translation in the plane and rotation) and noise can be added for each subject. Rician noise is added to the no-noise simulated fMRI signal relative to a specified contrast-to-noise ratio (CNR). The CNR is defined as $\hat{\sigma}_s/\hat{\sigma}_n$, where $\hat{\sigma}_s$ is the temporal standard deviation of the true signal and $\hat{\sigma}_n$ is the temporal standard deviation of the noise.

4.2 fMRI Data for Controls and Patients

The AOD experimental design, which consists of detecting an infrequent sound within a series of regular and different sounds, is generated based on an example simulation from [52]. This simulation is explained in [49] and has been slightly modified to generate differential activation between two groups of subjects. In addition, the CNR and the probability of unique events were also modified to make the classification task more challenging.

This event-related paradigm task consists of a single run of three stimuli presented to each participant in random order. The standard stimulus is a baseline tone, the target stimulus is a distinct tone that subjects should press a button upon hearing, and the novel stimulus is a random digital noise. These stimuli occur at each TR with probability 0.6, 0.075, and 0.075, respectively.

The experimental task is simulated for two groups of $M = 50$ subjects, each subject with up to $C = 27$ components in a data set with $V = 148 \times 148$ voxels and $T = 150$ time points collected at TR=2 seconds. Some of the selected sources are task-related, while the other ones are “not of interest”, being present with probability 0.9, i.e., some sources may be absent for each subject. To mimic between-subject spatial variability, the sources for each subject are given a small amount of translation, rotation, and spread via normal deviates. Translation in the horizontal and vertical directions of each source have a standard deviation of 0.1 voxels, rotation has a standard deviation of 1 degree, and spread has a center of 1 and standard deviation of 0.03.

The TCs are defined by task and unique events, the timing of the task events being the same for all subjects to simulate a unique session of the AOD task. Four task event types are defined. At each TR, in addition to the three task event types that have already been mentioned, a spike event occurs with probability 0.05. Components are separately modulated by each event type and spike events are mapped only to CSF sources with amplitude 1. The details of the modulation coefficients of each task event for all components can be found in [49]. We only provide a list of the task-related brain regions of this simulation on Table 4.1. This section will only give specific details of the modulation coefficients of the task events for the components that are differentially activated between groups. But first, let us discuss which these components are and the rationale used for their selection.

Some publications [53, 54] have demonstrated that both the temporal lobe and the default mode network show an abnormal activation in schizophrenia patients. In addition, a resting-state study [55] found evidence of reduced connectivity in the dorsal attention and executive control networks on schizophrenia patients. The AOD task is designed in such a way that subjects have to make a quick button-press response upon the presentation of target stimuli. Since it has been suggested that

Table 4.1: List of task-related brain regions. These regions have a one-to-one association to the default SMs defined by SimTB. The second column of this table indicates the component number assigned to these regions when ICA was applied to the simulated data (section 4.3).

Component Label	Number
Left Auditory	5
Left Frontal	7
Bilateral Frontal	8
Right Frontal	9
Left Hippocampus	12
Right SensoriMotor	15
Right Hippocampus	18
Dorsal Attention Network	19
Precuneus	21
Default Mode Network	24
Left SensoriMotor	29
Right Auditory	30

together with the posterior cingulate the precuneus is “pivotal for conscious information processing” [56], it was considered in the set of differentially activated brain regions. Finally, due to the fact that the AOD task is designed to activate the sensorimotor cortex upon target stimuli and given the evidence of impaired attention on schizophrenia, these brain regions also had a different activation pattern for patients.

Table 4.2 shows the modulation coefficients of the components that comprise the aforementioned brain regions for both controls and patients. This table also provides an estimate of the fractional increment/decrement of the absolute values of the coefficients assigned to controls used to generate the ones used for the patients group.

All sources have unique events that occur with a probability of 0.4 at each TR. For sources not of interest (no task modulation), the unique event amplitude is 1. For task-modulated sources, unique events are added with small amplitudes (0.2 to

Table 4.2: Modulation coefficients of components with differential activity between controls and patients. This table lists the modulation coefficients of the three AOD task events of different components for healthy subjects and the fractional increment/decrement of their absolute values on the patients group.

Source	Task Event	HC	SZ	Coeff. Inc.
Auditory	Standard	1.00	1.05	↑ 5%
	Target	1.20	1.08	↓ 10%
	Novel	1.50	1.35	↓ 10%
SensoriMotor	Standard	–	–	–
	Target	1.00	1.15	↑ 15%
	Novel	0.50	0.45	↓ 10%
Default Mode Network	Standard	-0.30	-0.27	↓ 10%
	Target	-0.30	-0.40	↑ 30%
	Novel	-0.30	-0.33	↑ 10%
Dorsal Attention Network	Standard	0.70	0.70	0
	Target	0.80	0.65	↓ 20%
	Novel	1.20	1.30	↑ 10%
Precuneus	Standard	–	–	–
	Target	0.50	0.35	↓ 30%
	Novel	–	–	–

0.5) so that components responding to the same events have similar but not identical activation. CSF sources have smaller unique events (amplitude of 0.05).

TCs are generated from the event time series using the convolution with a canonical HRF, except for CSF components, which use a spike model with faster dynamics than the canonical HRF (peak at 2 seconds). To avoid the application of motion correction to the simulated data prior to using data analysis approaches on it, no motion is considered in the simulation. Finally, Rician noise is added to the data of each subject to reach the appropriate CNR level, which is uniformly distributed over subjects from 0.4 to 1.2.

4.3 Simulated Data Analysis

4.3.1 Group spatial ICA

The simulated fMRI data set is generated under the model of spatiotemporal separability so that specific brain regions can exhibit differential activation between controls and patients. The example simulation presented in [52] not only provides a setting that is flexible enough to generate data that satisfies this condition, but it also generates fMRI activation distributed across several brain regions that have a similar response across groups. Such a data set is well suited to test the performance of classifiers that look for a sparse set of regions that present dissimilar activation patterns between groups. In order to extract the brain activity present in the regions that are modelled by the simulation, ICA is applied to the data.

Group spatial ICA [57] was used to decompose the data into independent components as follows. First, each subject's functional data was applied dimensionality reduction by using PCA. The time domain was reduced from 150 time points to 40 dimensions. Next, the reduced data from all subjects was temporally concatenated into a group matrix and a second data reduction step was applied to it, reducing this aggregate data set to 30 temporal dimensions. Then, ICA was applied to these reduced aggregate data set using the infomax algorithm [58] and 30 components were extracted. Finally, individual subject components were back-reconstructed from the group ICA analysis. Fig. 4.3 shows a subset of the extracted components, which includes the task-related components as well as white matter (WM) regions and the lateral ventricles containing cerebrospinal fluid (CSF). More information can be found in Table 4.1, which displays the labels of the task-related components and their associated component numbers.

The TCs of the components were used to characterize the data. Therefore, 100 labeled data (one observation per subject) composed of 30 feature subspaces (the number of components) is used to classify controls and patients (50 subjects on each group). Each feature subspace (block) has a dimensionality of 150 (the number of fMRI time points).

4.3.2 Degree of differential activation of the components

Since the components specified on Table 4.2 were modelled on SimTB to be differentially activated between controls and patients, it is hypothesized that the ICA components associated to those brain regions exhibit a similar activation pattern between both groups. However, a metric that measures the degree of differential activation of the components' TCs is required. To do so, the multivariate extension of the t-test, Hotelling's T-squared test, was applied to the TCs.

Hotelling's T-squared test is capable of measuring differences between multivariate means of two populations, in this case the population distributions of controls and patients. Since this test is computed based on the sample covariance of the data and the number of subjects per group (50) was less than the number of fMRI time points (150), a two-sample Hotteling's T-squared test was run on three windows of 40 time points each for all the ICA components, taking the mean of the T-squared values as the metric of interest. Similarly, the p-values associated to each test were retrieved, generating a mean p-value for each component. Then the averaged T-squared values across components were normalized so that they added to 100 to achieve a better interpretation of the results.

Table 4.3 shows the normalized T-squared values (and their averaged p-values) sorted in decreasing order for the top-ranked components. It can be seen that the ICA components that show high activation levels on the brain regions modelled to be differentially activated between groups achieve high T^2 values. While some of these components have a mean p-value that is not low enough to reject the null hypothesis, which assumes that multivariate means of the TCs of the tested components are equal for both groups, they still achieve a high metric value. In addition, the proposed metric is not a rigorous estimate of the differential activity of the components between groups. Nonetheless, it provides a rough estimate that allows us to rank the components based on their dissimilarity between groups. Furthermore, it makes it possible to distinguish two groups of components in the table: one of clearly informative ones (down to the CSF component) and a group of components that is statistically equivalent for controls and patients. It is interesting to find that the CSF component shows a nontrivial metric value, although the data of this component was modeled the same for both groups. This finding might be explained by the fact that this component was activated by a spike event, which had a different onset timing than the other events (standard, target and novel) and was modeled using a time response that is completely different from the canonical HRF.

4.4 Application of RCK and ν-MKL to Simulated Data

After extracting the data from the TCs, the features from the generated input vectors were standardized. Then, linear kernel matrices were generated with the normalized input vectors, after which the kernels were subtracted their mean and were scaled to have unitary standard deviation on the feature space (refer to chapter 2 for details on feature and kernel normalization).

Table 4.3: Normalized two-sample Hotelling's T-squared coefficients of the top-ranked components' TCs for controls and patients and their associated mean p-values. The ICA components that show high activation levels on the brain regions modelled to be differentially activated between groups achieve high T^2 values, as it was expected.

Component	Normalized T^2 value	p-value
Left Auditory	38.04	0
Left SensoriMotor	14.74	3.92e-14
Right SensoriMotor	14.53	3.49e-07
Right Auditory	7.93	2.09e-05
Default Mode Network	7.87	1.85e-05
Dorsal Attention Network	6.99	0.12
Precuneus	2.76	0.41
Lateral Ventricles (CSF)	1.63	0.99
White matter tracts - posterior	0.69	1.00
Left Frontal	0.68	1.00
Right Frontal	0.57	1.00
White matter tracts - anterior	0.55	1.00
Left Hippocampus	0.42	1.00
Bilateral Visual - more posterior	0.39	1.00
Bilateral Post-central	0.33	1.00

4.4.1 Parameter selection, optimal component set selection and prediction accuracy estimation

The mean accuracy rate of the analyzed classification algorithms is estimated. In addition, model selection (the optimal component selection) needs to be performed by both algorithms, not to mention the selection of the optimal parameters of the learning machines. In order to avoid getting a biased estimate of the classification accuracy rates achieved by both algorithms, two-layer 10-fold cross-validation [59] was used.

Accuracy rate calculation and model/parameter validation were performed as follows. First, the labeled data (100 observations) was divided into 10 stratified folds, i.e., each fold contained the same proportions of the two class labels. In addition, parameter C (refer to chapter 2) was fixed to 10 for both algorithms. Similar procedures are followed for both algorithms, but they are explained separately for clarity purposes.

In the case of RCK, 1 fold was set aside for test purposes only. The remaining data, which is called $TrainValSet$ in Algorithm 2, was further divided into training and validation sets, the latter one being composed of data from one fold of $TrainValSet$, as shown in Algorithm 3. The classifier was trained by using all the components and the validation error rates were estimated as shown in Algorithms 2 and 3. The above process was repeated for all folds. Then, the algorithm was retrained and the the discriminative weights were estimated, eliminating the component with minimum associated value. This procedure was repeated until a single component remained.

Afterwards, the component set I_L that achieved the minimum validation error was selected for $TrainValSet$ and the test error rate was estimated using the previously reserved test set. Then, another fold was selected as the new test set and the entire procedure was repeated for each of these test sets. The test accuracy rate was then estimated by averaging the accuracy rates achieved by each test set.

Similar to the procedure followed by RCK, a fold was set aside for test purposes for ν-MKL, the remaining data being called $TrainValSet$. As specified in Algorithm 4, the optimal values of parameters $C^{'}$ and ν were selected for $TrainValSet$. $C^{'}$ was selected from a pool of 10 logarithmically spaced values betweeen 10 and 100. On the other hand, the pool of values of ν was selected such that the number of selected components was at most 1, 2, 3 and so forth up to 15, a value that is considerably higher than the number of relevant components defined by the ground truth presented in Table 4.3. Since the value of ν defines a strict upper bound of the

Algorithm 2 Estimate optimal component set for RCK

```
**Inputs**: $TrainValSet$
**Outputs**: $I_L$, $W_L$
**Define** $I(1)$: indexes for all components
**Define** $P$: number of components
**for** $p = 1$ to $P - 1$ **do**
    **Validate component set error RCK**($TrainValSet$,$I(p)$) $\Rightarrow E(p)$
    Train with $TrainValSet$ and $I(p)$
    Compute discriminative weights $W(p)$
    Remove component with lowest weight
    Store indexes of remaining components $\Rightarrow I(p+1)$
**end for**
Find $p$ that minimizes $E(p) \Rightarrow p_{min}$
$I(p_{min}) \Rightarrow I_L$, $W(p_{min}) \Rightarrow W_L$
```

Algorithm 3 Validate component set error RCK

```
**Inputs**: $TrainValSet$ and $I(p)$
**Outputs**: $E(p)$
**Define** $N$: number of folds in $TrainValSet$
**for** $j = 1$ to $N$ **do**
    Extract $Train(j)$ from $TrainValSet$
    Extract $Val(j)$ from $TrainValSet$
    Train with $Train(j)$ and $I(p) \Rightarrow SVMparameters$
    Test with $Val(j)$, $I(p)$ and $SVMparameters$
    Store error $\Rightarrow e(j)$
**end for**
Average $e(j) \Rightarrow E(p)$
```

number of selected components (see section 3.3.3), their values were set to guarantee that the aforementioned criterion was satisfied. Therefore, $\nu \in \{1.8, 2.8, \ldots, 15.8\}$.

Algorithm 4 Train and Validate ν-MKL

```
1: Inputs: $TrainValSet$, $\nu_{vals}$, $C^{'}_{vals}$, $C$
2: Outputs: $I_L$, $\gamma_L$
3: Validate parameters $\nu$ − MKL ($TrainValSet, \nu_{vals}, C^{'}_{vals}, C$) $\Rightarrow C^{'}, \nu$
4: Train with $TrainValSet$, $C^{'}$, $\nu$ and $C \Rightarrow \gamma_L$, $I_L$
```

$TrainValSet$ was subdivided into training and validation sets, as it is specified in Algorithm 5. ν-MKL was trained with all possible ($C^{'}$,ν) pairs, the validation error being estimated for each of them. This process was repeated for all folds, being the optimal pair the one that achieved the minimum mean validation error. Then, the optimal pair ($C^{'}$,ν) was used to retrain ν-MKL, thus finding the optimal component set I_L for $TrainValSet$. Next, the test error rate was estimated in the reserved test set, with the test accuracy rate being estimated by averaging the accuracy rates achieved for all test sets, just as it was done for RCK.

4.4.2 Estimation of informative components' statistics

As it has been explained in the previous section, the proposed algorithms compute optimal component sets I_L for every possible $TrainValSet$. Since 10-fold cross-validation is applied, the number of possible $TrainValSet$ is 10, each of them having a unique associated optimal component set.

The overall optimal component set found by these classifiers is composed of the union of the 10-folds' optimal component sets. The most informative components in this set should be the ones that are selected by most of the folds. By the same token, those components that are scarcely selected should be less informative than the other ones. For this reason, the selection frequency of the components across folds is estimated as an informative statistic to measure the relevance of the components in the optimal set.

Both classification algorithms provide another metric to estimate the relevance of the components in the optimal set. On the one hand, RCK reports the discriminative weights of the components. On the other hand, ν-MKL presents their γ values. While both of them measure the degree of information provided by the components, their values are not in the same numeric scale. To be able to compare them, these values were normalized by their maximum values at each fold. By doing so, the most relevant component for a given fold would achieve a normalized score of 1, regardless of it being detected by RCK or ν-MKL.

After normalizing both the discriminative weights and the γ values, their mean and standard deviation for each component were computed, being reported along with their selection frequency scores.

4.4.3 RCK and ν-MKL results

Both RCK and ν-MKL achieved very similar classification accuracy rates (0.90 and 0.92, respectively), the latter algorithm attaining a slightly better performance than RCK. While their accuracy rates are similar, their selected sets of the most discriminative components for both groups present some important differences.

RCK is capable of detecting the components that are actually differentially activated between groups, as shown in Table 4.4. However, it also includes components that show an equivalent activation pattern on both controls and patients. In fact, regions that are not even part of the set of top-15 ranked components presented in Table 4.3, such as the Bilateral Visual and Right Hippocampus components, are deemed relevant by RCK. In contrast, ν-MKL only includes informative components on its optimal component set, which is displayed in Table 4.5. It can also be seen that

Table 4.4: Optimal block set detected by RCK, which is composed of the components included on the block sets selected for each $TrainValSet$ on the 10-fold cross-validation procedure. The components' selection frequency, normalized mean discriminative weight value and their standard deviation across the 10 folds are reported.

Relevant Components	Selection Frequency	Norm. Discr. Weight Mean Val.	Norm. Discr. Weight Std. Dev.
Left Auditory	1.00	1.00	0.00
Right Auditory	1.00	0.49	0.02
Right SensoriMotor	1.00	0.27	0.01
Left SensoriMotor	1.00	0.26	0.01
Default Mode Network	1.00	0.22	0.03
Dorsal Attention Network	1.00	0.16	0.02
Precuneus	0.90	0.12	0.01
Lateral Ventricles (CSF)	0.90	0.10	0.01
White matter tracts (posterior)	0.50	0.05	0.01
Bilateral Visual	0.40	0.05	0.01
Right Hippocampus	0.40	0.05	0.01
Left Frontal	0.10	0.04	0.00
Bilateral Post-central	0.10	0.04	0.00

the CSF component is selected only 3 times by the different $TrainValSet$ generated by the 10-fold cross-validation procedure.

4.5 Discussion of the Results

RCK applies an iterative approach, particularly backward elimination, to find the optimal block set. Backward elimination is a greedy algorithm and it is known that greedy algorithms are usually suboptimal [60]. Thus it is highly probable that RCK finds a suboptimal component set. This statement may explain the fact that components that are not significantly different across groups are still selected by RCK. On the other hand, ν-MKL analyzes the whole set of available components on

Table 4.5: Optimal block set detected by ν-MKL which is composed of the components included on the block sets selected for each $TrainValSet$ on the 10-fold cross-validation procedure. The components' selection frequency, normalized mean gamma value and their standard deviation across the 10 folds are reported.

Relevant Components	Selection Frequency	Norm. Gamma Mean Value	Norm. Gamma Std. Dev.
Left Auditory	1.00	1.00	0.00
Right Auditory	1.00	0.57	0.04
Right SensoriMotor	1.00	0.31	0.07
Left SensoriMotor	1.00	0.31	0.07
Default Mode Network	0.90	0.25	0.07
Dorsal Attention Network	0.70	0.15	0.05
Precuneus	0.70	0.09	0.05
Lateral Ventricles (CSF)	0.30	0.06	0.03

its formulation to select the most relevant ones, thus being less prone to selecting non-differentially activated components.

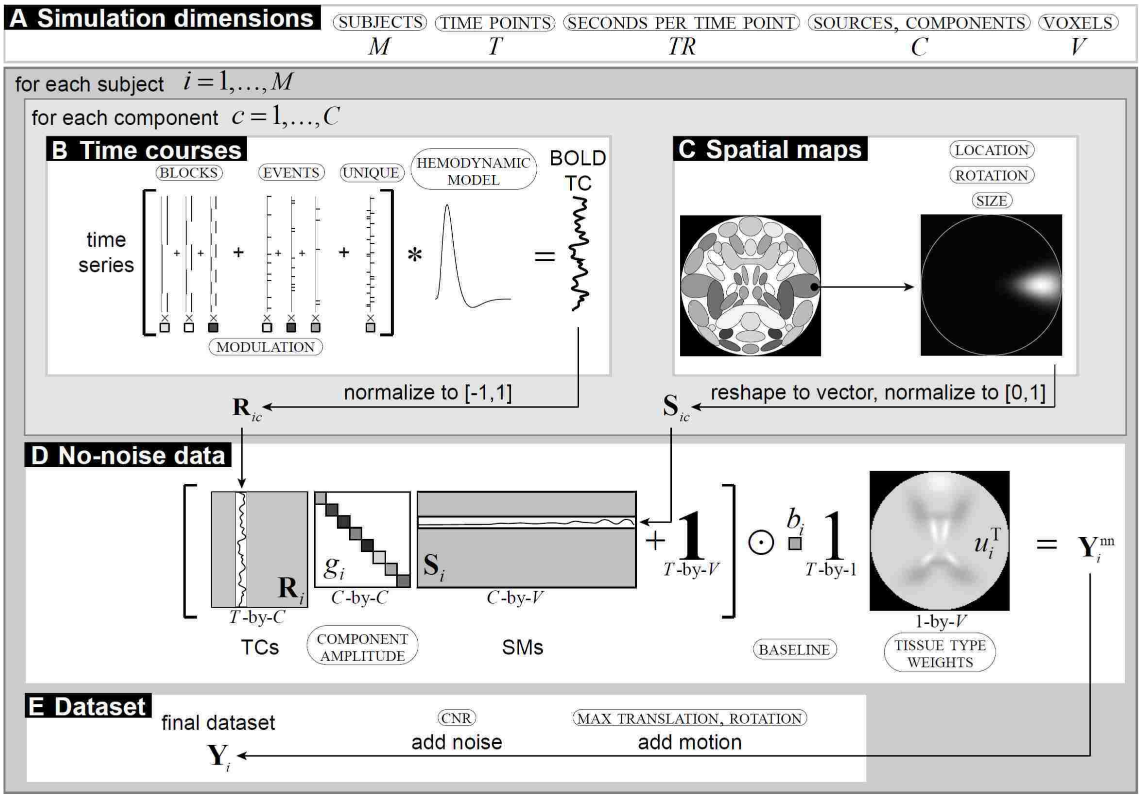


Figure 4.1: SimTB flowchart of data generation. (A) Simulation dimension is determined by the number of subjects, time points (and seconds per time point), and voxels (representing a number of selected sources). (B) Time courses are the sum of coefficient-modulated task block, task event, and unique event time series modeled into a BOLD TC and normalized. (C) Spatial maps are selected, translated, rotated, resized, and normalized. (D) The "no-noise" data combines the TCs and SMs scaled by component amplitudes, and scaled to a tissue type weighted baseline. (E) The final data set includes motion and noise. (Extracted from [49])

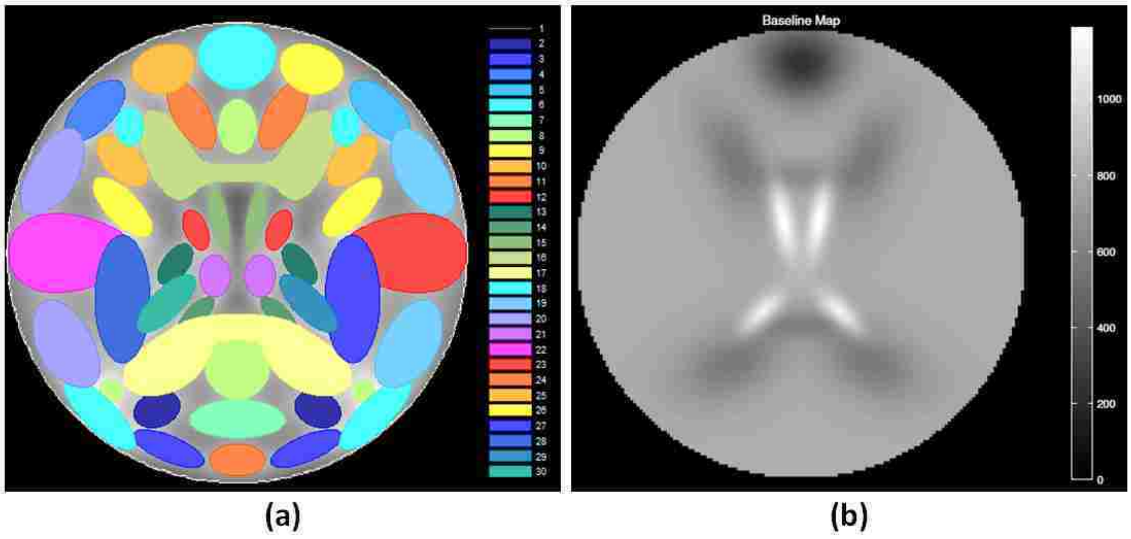


Figure 4.2: Configuration of (a) default sources and (b) default tissue baseline. Spatial maps are designed to represent components observed in axial slices of real fMRI data.

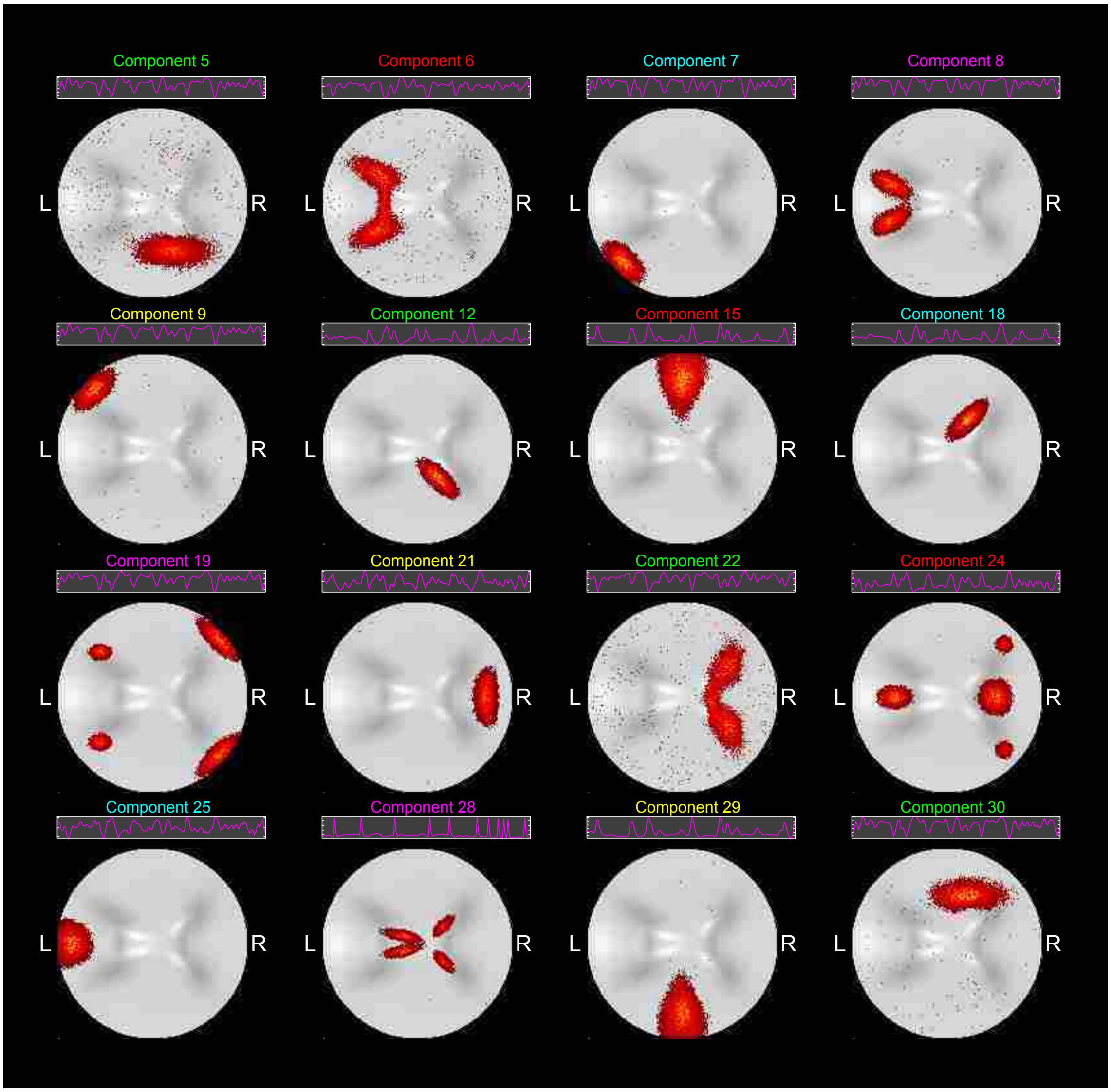


Figure 4.3: ICA components: task-related and medial frontal regions, white matter (WM) regions and lateral ventricles containing cerebrospinal fluid (CSF). The medial frontal region (component 25) and CSF (component 28) present a baseline greater than the primary tissue-type (TT), which is the gray matter (GM), while WM (components 6 and 22) has a baseline less than the primary TT. Refer to Table 4.1 to find the list of the task-related components and their associated labels.

Algorithm 5 Validate parameters ν-MKL

```
**Inputs**: $TrainValSet$, $\nu_{vals}$, $C^{'}_{vals}$, $C$
**Outputs**: $C^{'}$, $\nu$
**Define** $N$: number of folds in $TrainValSet$
**for** $i = 1$ to $N$ **do**
   Extract $Train(i)$ from $TrainValSet$
   Extract $Val(i)$ from $TrainValSet$
   **for** $j = 1$ to $\#C^{'}_{vals}$ **do**
      $C^{'}_{sel} = C^{'}_{vals}(j)$
      **for** $k = 1$ to $\#\nu_{vals}$ **do**
         $\nu_{sel} = \nu_{vals}(k)$
         Train with $Train(i)$, $C^{'}_{sel}$, $\nu_{sel}$ and $C \Rightarrow Trained\ \nu - MKL$
         Test with $Val(i)$ and $Trained\ \nu - MKL$
         Store error $\Rightarrow e(i, j, k)$
      **end for**
   **end for**
**end for**
Average $e(i, j, k)$ over $i \Rightarrow e(j, k)$
Find $(j, k)$ that minimizes $e(j, k) \Rightarrow (J, K)$
$C^{'}_{vals}(J) \Rightarrow C^{'}$
$\nu_{vals}(K) \Rightarrow \nu$
```

Chapter 5

Application of RCK and ν-MKL to fMRI data

This chapter presents the results of RCK and ν-MKL on the classification of healthy controls and schizophrenia patients on two different fMRI data sets acquired from an auditory task experiment. The first work, which is described in 5.1 and has been published in [61], applies RCK to combine fMRI data that is processed with two data analysis methods (GLM and ICA), showing that this algorithm takes advantage of the complementary nature of these analysis methods. The second one (section 5.2) analyzes fMRI data with RCK by taking into account both its magnitude and phase information. This preliminary analysis, which has been published in [62], provides evidence that phase information is useful to better discriminate controls from patients when used along with magnitude data. The last section of this chapter provides another analysis using data from the same study, but using a different set of subjects to better match controls and patients in terms of age. It presents a more solid framework for complex-valued fMRI data analysis using ν-MKL [63, 64], which in turn renders an improved characterization of schizophrenia.

5.1 Characterization of schizophrenia using RCK and multi-source fMRI analysis data

5.1.1 Introduction

As it has been specified in section 2.1.2, fMRI data can be characterized by model-based analysis such as GLM, which emphasize task-related activity in each voxel separately, or by non-model based ones such as ICA, which looks for different components of voxels that have temporally coherent neural activity. GLM and ICA approaches are complementary to each other. For this reason, it would be sensible to devise a method that could gain more insight of the underlying processes of brain activity by combining data from both approaches.

ICA has been extensively applied to fMRI data to identify differences among healthy controls and schizophrenia patients [65, 66, 67]. Calhoun et al. [53] showed that the temporal lobe and the default mode components could reliably be used together to identify patients with bipolar disorder and schizophrenia from each other and from healthy controls. Furthermore, Garrity et al. [54] demonstrated that the default mode component showed abnormal activation and connectivity patterns in schizophrenia patients. Therefore, there is evidence that suggest that the default mode and temporal lobe components are disturbed in schizophrenia. Based on the reported importance of the temporal lobe in the characterization of schizophrenia we used data from an auditory oddball discrimination (AOD) task, which provides a consistent activation of this part of the brain. Three sources were extracted from fMRI data using two analysis methods: model-based information via the GLM and functional connectivity information retrieved by ICA. The first source is a set of β-maps generated by the GLM. The other two sources come from an ICA analysis and include a temporal lobe component and the default mode network component.

As it has been discussed on section 2.3.2, one of the most commonly used approaches to reduce the dimensionality of fMRI data is feature selection. However, most models assume that there is an intrinsic linear relationship between voxels, as multivariate, nonlinear feature selection is computationally intensive. A convenient tradeoff consists on assuming that there are nonlinear relationships between voxels that are close to each other and that are part of the same anatomical brain region, and that voxels in different brain regions are linearly related. To do so, we propose the application of RCK using nonlinear kernels to fMRI data for schizophrenia detection.

Once the sources are extracted, volumes from both the GLM and ICA sources are segmented into anatomical regions. Each of these areas is mapped into a different space using RCK.

5.1.2 Materials and Methods

Participants

Data were collected at the Olin Neuropsychiatric Research Center (Hartford, CT) from healthy controls and patients with schizophrenia. All subjects gave written, informed, Hartford hospital IRB approved consent. Schizophrenia was diagnosed according to DSM-IV-TR criteria [68] on the basis of both a structured clinical interview (SCID) [69] administered by a research nurse and the review of the medical file. All patients were on stable medication prior to the scan session. Healthy participants were screened to ensure they were free from DSM-IV Axis I or Axis II psychopathology using the SCID for non-patients [70] and were also interviewed to determine that there was no history of psychosis in any first-degree relatives. All participants had normal hearing, and were able to perform the AOD task (see Section 5.1.2) successfully during practice prior to the scanning session.

Data from 106 right-handed subjects were used, 54 controls aged 17 to 82 years (mean=37.1, SD=16.0) and 52 patients aged 19 to 59 years (mean=36.7, SD=12.0). A two-sample t-test on age yielded $t = 0.13$ ($p = 0.90$). There were 29 male controls (M:F ratio=1.16) and 32 male patients (M:F ratio=1.60). A Pearson's chi-square test yielded $\chi^2 = 0.67$ ($p = 0.41$).

Experimental Design

The AOD task involved subjects that were presented with three frequencies of sounds: target (1200 Hz with probability, $p = 0.09$), novel (computer generated complex tones, $p = 0.09$), and standard (1000 Hz, $p = 0.82$) presented through a computer system via sound insulated, MR-compatible earphones. Stimuli were presented sequentially in pseudorandom order for 200 ms each with inter-stimulus interval varying randomly from 500 to 2050 ms. Subjects were asked to make a quick button-press response with their right index finger upon each presentation of each target stimulus; no response was required for the other two stimuli. There were two runs, each comprising 90 stimuli (3.2 minutes) [71].

Image Acquisition

Scans were acquired at the Institute of Living, Hartford, CT on a 3T dedicated head scanner (Siemens Allegra) equipped with 40mT/m gradients and a standard quadrature head coil. The functional scans were acquired using gradient-echo echo planar imaging (EPI) with the following parameters: repeat time (TR) = 1.5 sec, echo time (TE) = 27 ms, field of view = 24 cm, acquisition matrix = 64×64, flip angle = $70\,^{\circ}$, voxel size = $3.75 \times 3.75 \times 4$ mm^3, slice thickness = 4 mm, gap = 1 mm, number of slices = 29; ascending acquisition. Six dummy scans were carried out at the beginning to allow for longitudinal equilibrium, after which the paradigm was automatically triggered to start by the scanner.

Preprocessing

fMRI data were preprocessed using the SPM5 software package (`http://www.fil.ion.ucl.ac.uk/spm/software/spm5/`). Images were realigned using INRIalign, a motion correction algorithm unbiased by local signal changes [72]. Data were spatially normalized into the standard Montreal Neurological Institute (MNI) space [73], spatially smoothed with a $9 \times 9 \times 9-\text{mm}^3$ full width at half-maximum Gaussian kernel. The data (originally acquired at $3.75 \times 3.75 \times 4$ mm^3) were slightly upsampled to $3 \times 3 \times 3$ mm^3, resulting in $53 \times 63 \times 46$ voxels.

Creation of Spatial Maps

The GLM analysis performs a univariate multiple regression of each voxel's timecourse with an experimental design matrix, which is generated by doing the convolution of pulse train functions (built based on the task onset times of the fMRI experiment) with the hemodynamic response function [51]. This results in a set of β-weight maps (or β-maps) associated with each parametric regressor. The β-maps associated with the target versus standard contrast were used in our analysis. The final target versus standard contrast images were averaged over two runs.

In addition, group spatial ICA [57] was used to decompose all the data into 20 components using the GIFT software (`http://icatb.sourceforge.net/`) as follows. Dimension estimation, which was used to determine the number of components, was performed using the minimum description length criteria, modified to account for spatial correlation [74]. Data from all subjects were then concatenated and this aggregate data set reduced to 20 temporal dimensions using PCA, followed by an independent component estimation using the infomax algorithm [58]. Individual subject components were back-reconstructed from the group ICA analysis to generate their associated spatial maps (ICA maps). Component maps from the two runs were averaged together resulting in a single spatial map of each ICA component for each subject. It is important to mention that this averaging was performed after the spatial ICA components were estimated. The two components of interest (temporal lobe and default mode) were identified in a fully automated manner using different approaches. The temporal lobe component was detected by temporally sorting the components in GIFT based on their similarity with the SPM design regressors and retrieving the component whose ICA timecourse had the best fit. By contrast, the default mode network was identified by spatially sorting the components in GIFT using a mask derived from the Wake Forest University pick atlas (WFU-PickAtlas) [75, 76, 77], (`http://www.fmri.wfubmc.edu/download.htm`). For the default mode mask we used precuneus, posterior cingulate, and Brodmann areas 7, 10, and 39 [78, 79]. A spatial multiple regression of this mask with each of the networks was performed, and the network which had the best fit was automatically selected as the default mode component.

Data Segmentation and Normalization

The spatial maps obtained from the three available sources were segmented into 116 regions according to the automated anatomical labeling (AAL) brain parcellation defined in [80] by using the WFU-PickAtlas. In addition, the spatial maps were normalized by subtracting from each voxel its mean value across subjects and dividing it by its standard deviation. Multiple kernel learning methods such as composite kernels and RCK further required each kernel matrix to be scaled such that the variance of the training vectors in its associated feature space were equal to 1. This procedure is explained in more detail in the next section.

Composite Kernels Method

Structure of the learning machine based on composite kernels Each area from example i is placed in a vector $\mathbf{x}_{i,l}$ where $i, 1 \leq i \leq N$ is the example index and $l, 1 \leq l \leq L$ is the area index. An example is defined as either a single-source spatial map or the combination of multiple sources spatial maps of a specific subject. In the particular case of our study $N = 106$. For single-source analysis, composite kernels map each example i into $L = 116$ vectors $\mathbf{x}_{i,l}$; for two-source analysis, composite kernels map each example into $L = 2 \times 116 = 232$ vectors $\mathbf{x}_{i,l}$, and so on. Then, each vector is mapped through a nonlinear transformation $\varphi_l(\cdot)$, following the classification structure defined on section 3.1. In this work, kernels $k_l(\cdot, \cdot)$ are defined to be Gaussian kernels with the same parameter σ.

When the kernel function $k_l(\cdot, \cdot)$ is applied to the training vectors in the data set, matrix $\mathbf{K}_l$ is generated. Component i, j of this matrix is computed as $\mathbf{K}_l(i, j) = k_l(\mathbf{x}_{i,l}, \mathbf{x}_{j,l})$. Variance normalization is applied to these kernel matrices as mentioned in section 2.3.3 by using the following transformation:

$$K_l \mapsto \frac{K_l}{\frac{1}{N}\sum_{i=1}^{N} K_l(i,i) - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} K_l(i,j)}, \tag{5.1}$$

where the denominator of Eq. 5.1 is the variance of the examples in the feature space [38].

Let example $\mathbf{x}_i$ be nonlinearly mapped to a Hilbert space such that $\varphi(\mathbf{x}_i) = [\varphi_1^T(\mathbf{x}_{i,1}) \cdots \varphi_L^T(\mathbf{x}_{i,L})]^T$. Then, as it has been shown in chapter 3, the predicted value of a given test pattern $\mathbf{x}_*$ can be expressed by Eq. 3.5, which is displayed below

$$\begin{aligned} y =& \sum_{l=1}^{L}\sum_{i=1}^{N} \alpha_i \varphi_l^T(\mathbf{x}_{i,l})\varphi_l(\mathbf{x}_{*,l}) + b \\ =& \sum_{i=1}^{N} \alpha_i \sum_{l=1}^{L} k_l(\mathbf{x}_{i,l}, \mathbf{x}_{*,l}) + b, \end{aligned} \tag{5.2}$$

where α_i are the machine parameters that have to be optimized using a simple least squares approach or SVMs. In this work, SVMs are used by means of the LIBSVM software package [81] (`http://www.csie.ntu.edu.tw/~cjlin/libsvm`). Note that the output is a linear combination of kernels, which is used to train RCK as shown in section 3.2.3, the discriminative weights of the brain regions being estimated as it is explained in section 3.2.2.

Parameter selection, optimal area set selection and prediction accuracy estimation RCK (see section 3.2) is run for both single-source and multi-source data. There are two parameters that need to be tuned in order to achieve the best performance of the learning machine. These parameters are the SVM error penalty parameter C [17] and the Gaussian kernel parameter σ. Based on preliminary experimentation, it was discovered that the problem under study was rather insensitive to the value of C, so it was fixed to $C = 100$. In order to select σ, a set of 10 logarithmically spaced values between 1 and 100 were provided to the classifier.

The validation procedure consists of finding the optimal parameter pair $\{\sigma, I_{areas}\}$, where I_{areas} specifies a subset of the areas indexes. If a brute-force approach were used, then the validation error rates obtained for all possible values of σ and all combinations of areas would need to be calculated.

The previously mentioned exhaustive approach is unaffordable. For this reason, we propose a recursive algorithm based on the calculation of discriminative weights (please refer to section 3.2). Based on this method, a grid search could be performed by calculating the validation error and the training discriminative weights for each value of σ and each remaining subset of areas at each iteration of the recursive algorithm. The algorithm would start with all brain regions, calculate the discriminative weights for each value of σ and eliminate at each iteration the regions with least discriminative weight in the area sets associated to each σ value. After executing the whole grid search, the pair $\{\sigma, I_{areas}\}$ that yielded the minimum validation error rate would be selected.

The aforementioned method could be further simplified by calculating only the training discriminative weights associated to the optimal value of σ at each iteration of RCK. This procedure is suboptimal compared to the previous one, but it reduces its computational time. The following paragraphs provide more details of the previously discussed validation procedure and the test accuracy rate calculation.

First of all, a pair of observations (one from a patient and one from a control) is set aside to be used for test purposes and not included in the validation procedure. The remaining data, which is called $TrainValidSet$ in algorithm 6, is further divided in training and validation sets, the latter one being conformed by another control/patient data pair, as shown in algorithm 7.

Algorithm 6 Train and Validate

```
**Inputs**: $TrainValSet$
**Outputs**: $SigmaOpt$, $Iopt$ and $SVMparameters$
**Define** $I(1)$: indexes for all areas
**Define** $P$: number of areas
**for** $p = 1$ to $P - 1$ **do**
    **Validate sigma with LTO**($TrainValSet$,$I(p)$) $\Rightarrow$ $Sigma(p)$ and $E(p)$
    Train with $TrainValSet$, $Sigma(p)$ and $I(p)$
    Compute discriminative weights
    Remove area with lowest weight
    Store indexes of remaining areas $\Rightarrow I(p+1)$
**end for**
Find $p$ that minimizes $E(p) \Rightarrow p_{min}$
$Sigma(p_{min}) \Rightarrow SigmaOpt$, $I(p_{min}) \Rightarrow Iopt$
Train with $TrainValSet$, $SigmaOpt$ and $Iopt \Rightarrow SVMparameters$
```

Algorithm 7 Validate sigma with LTO

```
**Inputs**: $TrainValSet$ and $I(p)$
**Outputs**: $Sigma(p)$ and $E(p)$
**Define** $N$: number of subject pairs in $TrainValSet$
**Define** $L$: Number of possible values for sigma
**for** $j = 1$ to $N$ **do**
    Extract $Train(j)$ from $TrainValSet$
    Extract $Val(j)$ from $TrainValSet$
    **for** $k = 1$ to $L$ **do**
        Train with $Train(j)$, $sigma(k)$ and $I(p) \Rightarrow SVMparameters$
        Test with $Val(j)$, $sigma(k)$, $I(p)$ and $SVMparameters$
        Store error $\Rightarrow e(j,k)$
    **end for**
**end for**
Average $e(j,k)$ over $j \Rightarrow e(k)$
Find $k$ that minimizes $e(k) \Rightarrow E(p)$
$sigma(k) \Rightarrow Sigma(p)$
```

The classifier is trained by using all the brain regions and all possible σ values and the validation error rates are estimated as shown in algorithm 7. The above process is repeated for all control/patient pairs. Next, the value of σ that yields the minimum validation error is selected and this error is stored. Next, the algorithm is retrained with this value of σ and the discriminative weights are estimated, eliminating the area with minimum associated value. This procedure is then repeated until a single brain region is analyzed.

Afterwards, the pair $\{\sigma, I_{areas}\}$ that achieves minimum validation error is selected and the test error rate is estimated using the previously reserved test set. Then, another control/patient pair is selected as the new test set and the entire procedure is repeated for each of these test set pairs. The test accuracy rate is then estimated by averaging the accuracy rates achieved by each test set.

Comparison of composite kernels and RCK with other methods The composite kernels algorithm allows the analysis of non-linear relationships between voxels within a brain region and captures linear relationships between those regions. We compare the performance of the proposed algorithm for single-source and multi-source analyses with both a linear SVM, which assumes linear relationships between voxels, and a Gaussian SVM, which analyzes all possible non-linear relationships between voxels. The data from each area, which is extracted by the segmentation process, is input to the aforementioned conventional kernel-based methods after been concatenated.

Besides analyzing the classification accuracy rate obtained by our proposed feature selection approach (RCK) compared to the previously mentioned algorithms, we are interested in evaluating the performance of RCK by comparing it against another RFE-based procedure: RFE-SVM applied to linear SVMs (which will be hereafter referred to as RFE-SVM).

Parameter selection for the aforementioned algorithms is performed as follows. As stated before, the problem under study is rather insensitive to the value of C. Therefore, its value is fixed to 100 for linear SVM, Gaussian SVM and RFE-SVM. In addition, the Gaussian kernel parameter σ values are retrieved from a set of 100 logarithmically spaced values between 1 and 1000.

5.1.3 Results

RCK Applied to Single Sources

This section presents the sets of most relevant areas and the test results of RCK applied to each source.

The mean test accuracy achieved by using ICA default-mode component data is 90%. The list of overall 40 brain regions that were selected by RCK for the ICA default mode component data are listed in Table 5.1, alongside the statistics of their discriminative weights. These regions are grouped in macro regions to better identify their location in the brain. Furthermore, the rate of training sets that selected each region (selection frequency) is also specified.

When RCK is applied to the ICA temporal lobe component data, it achieves a mean test accuracy rate of 85%. The optimal area set obtained by using ICA temporal lobe data is reported in Table 5.2.

Finally, RCK achieves a mean test accuracy rate of 86% when it is applied to GLM data. The list of areas selected by RCK in this case is displayed in Table 5.3.

RCK Applied to Multiple Sources

All possible combinations of data sources were analyzed by RCK, and we report the obtained results for each of them (please refer to Table 5.6). It can be seen that RCK achieves its peak performance when it is applied to all of the provided sources (95%). Due to this fact, we think that special attention should be given to the areas retrieved by this multi-source analysis and its characterization by means of their discriminative weights. Therefore, we present Table 5.4, which displays this information. In addition, a graphical representation of the coefficients associated

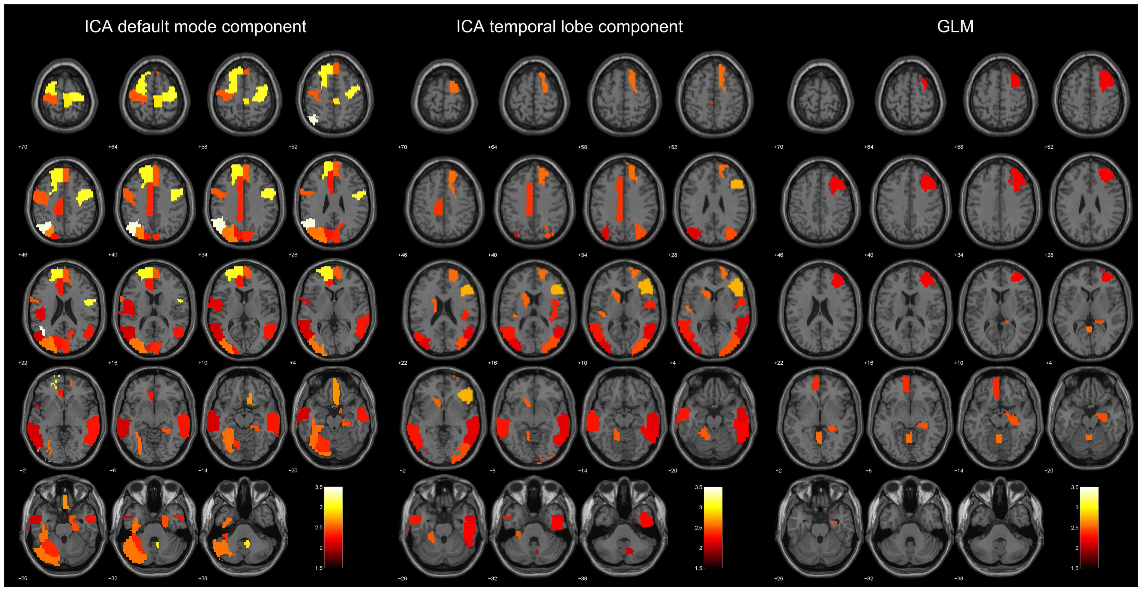


Figure 5.1: Discriminative weights brain maps for multi-source analysis. The brain maps of each of these sources highlight the brain regions associated to each of them that were present in the optimal area set for this multi-source data classification. These areas are color-coded according to their associated discriminative weights.

to those areas is presented in Fig. 5.1, which overlay colored regions on top of a structural brain map for each of the three analyzed sources.

Comparison of the Performance of Composite Kernels and RCK with Other Methods

For single-source data analysis, Table 5.5 shows that both Gaussian SVMs and composite kernels exhibit an equivalent performance for all sources, while the classification accuracy achieved by linear SVMs for both ICA temporal lobe and GLM sources are smaller than the ones attained by the aforementioned algorithms. It can also be seen that there is a moderate difference between the classification accuracy rates obtained by RCK and RFE-SVM when they are applied to all data sources, except

ICA default mode.

The results of multi-source analysis are shown in Table 5.6. In this case, linear SVMs and Gaussian SVMs reach a similar prediction accuracy for all multi-source analyses, except for the case when they are provided with data from ICA temporal lobe and GLM sources. While composite kernels achieve almost the same classification accuracy as linear and Gaussian SVMs when provided with three-sources data, its performance is reduced on the other multi-source analyses. The differences between classification rates for RFE-based methods are small for multi-source data analyses, with RCK achieving slightly better results in some cases.

5.1.4 Discussion

A classification algorithm based on composite kernels that is applicable to fMRI data has been introduced. This algorithm analyzes nonlinear relationships across voxels within anatomical brain regions and combines the information from these areas linearly, thus assuming underlying linear relationships between them. By using composite kernels, the regions from segmented whole-brain data can be ranked multivariately, thus capturing the spatially distributed multivariate nature of fMRI data. The fact that whole-brain data is used by the composite kernels algorithm is a feature of special importance, since the data within each region does not require any feature extraction preprocessing procedure in order to reduce their dimensionality. The application of RFE to composite kernels enables this approach to discard the least informative brain regions and hence retrieve the brain regions that are more relevant for class discrimination for both single-source and multi-source data analyses. The discriminative coefficients of each brain region indicate the degree of differential activity between controls and patients. Despite the fact that composite kernels cannot indicate which of the analyzed groups of interest is more activated for a specific brain region like linear SVMs could potentially do, the proposed method is still capable of measuring the degree of differential activity between groups on that region. Furthermore, RCK enables the use of a nonlinear kernel within a RFE procedure, a task that can become barely tractable with conventional SVM implementations. Another advantage of RCK over other RFE-based procedures such as RFE-SVM is its faster execution time; while the former takes 12 hours to be executed, the latter takes 157 hours, achieving a 13-fold improvement. Finally, this paper shows that the proposed algorithm is capable of taking advantage of the complementarity of GLM and ICA by combining them to better characterize groups of healthy controls and schizophrenia patients; the fact that the classification accuracy achieved by using data from three sources surpasses that reached by using single-source data supports this claim.

The set of assumptions upon which the proposed approach is based are the linear relationships between brain regions, the nonlinear relationships between voxels in the same brain region and the sparsity of information in the brain. These assumptions seem to be reasonable enough to analyze the experimental data based on the obtained classification results. This does not imply that cognitive processes actually work in the same way as it is stated in our assumptions, but that the complexity assumed by our method is sensible enough to produce good results with the available data. While composite kernels achieve classification accuracy rates that are greater than or equal to those reached by both linear and Gaussian SVMs when applied to single-source whole-brain data, the same does not hold for multi-source analysis. It may be possible that composite kernels performance is precluded when it is provided with too many areas, making it prone to overfitting.

The presented results suggest that for a given number of training data, the trade-off of our proposed algorithm between the low complexity of the linear assumption, which provides the rationale of linear SVMs, and the high complexity of the fully nonlinear approach, which motivates the application of Gaussian SVMs, is convenient. In the case of composite kernels, they assume linear relationships between brain regions but are flexible enough to analyze nonlinearities within them. Nevertheless, their results are similar to the ones of the previously mentioned approaches for single-source analysis and inferior for multi-source analysis since they do not take advantage of information sparsity in the brain, thus not significantly reducing the classifier complexity. However, the accuracy rates attained by RCK are significantly better than the ones achieved by composite kernels. These results reinforce the validity of two hypotheses: first, that indeed there are brain regions that are irrelevant for the characterization of schizophrenia (information sparsity); and second, that RCK is capable of detecting such regions, therefore being capable of finding the set of most informative regions for schizophrenia detection given a specific data source.

Table 5.6 shows the results achieved by different classifiers using multi-source data. It is important to notice that the results obtained by all the classifiers when all of the sources are combined are greater than those obtained by these algorithms when they are provided with data from the ICA default mode component and either the ICA temporal lobe component or GLM data. The only method for which the previous statement does not hold is RFE-SVM. This finding may seem counterintuitive as one may think that both ICA temporal lobe component and GLM data are redundant, since they are detected based on their similarity to the stimuli of the fMRI task. However, the fact that ICA and GLM characterize fMRI data in different ways (the former analyzes task-related activity, while the latter detects groups of voxels with temporally coherent activity) might provide some insight of why the combination of these two sources proves to be important together with ICA default mode data.

In addition to the accuracy improvement achieved by applying feature selection to whole-brain data classification, RCK allows us to better identify the brain regions that characterize schizophrenia. The fact that several brain regions in the ICA temporal lobe component are present in the optimal area set is consistent with the findings that highlight the importance of the temporal lobe for schizophrenia detection. It is also important to note the presence of the anterior cingulate gyrus of the ICA default mode component in the optimal area set, for it has been proposed that error-related activity in the anterior cingulate cortex is impaired in patients with schizophrenia [82]. The participants of the study are subject to making errors since the AOD task is designed in such a way that subjects have to make a quick button-press response upon the presentation of target stimuli. Since attention plays an important role in this fMRI task, it is sensible to think that consistent differential activation of the dorsolateral prefrontal cortex (DLPFC) for controls and patients will be present [83]. That may be the reason why the right middle frontal gyrus of the GLM is included in the optimal area set.

Brain aging effects being more pronounced in individuals after age 60 [84] raised a concern that our results may have been influenced by the data collected from four healthy controls who exceeded this age cutoff in our sample. Thus, we re-ran our analysis excluding these four subjects. Both the resulting classification accuracy rates and the optimal area sets were consistent with the previously found ones. These findings seem to indicate that the algorithm proposed in this paper is robust enough not to be affected by the presence of potential outliers when provided with consistent features within the groups of interest.

To summarize, this work extends previous studies like [85, 53, 54] by introducing new elements. First, the method allows the usage of multi-source fMRI data, making it possible to combine ICA and GLM data. And second, it can automatically identify and retrieve regions which are relevant for the classification task by using whole-brain data without the need of selecting a subset of voxels or a set of ROIs prior to classification. Based on the aforementioned capabilities of the presented method, it is reasonable to think that it can be applied not only to multi-source fMRI data, but also to data from multiple imaging modalities (such as fMRI, EEG or MEG data) for schizophrenia detection and identify the regions within each of the sources which differentiate controls and patients better. Further work includes the modification of the composite kernels formulation to include scalar coefficients associated to each kernel. By applying new improved strategies based on optimizers that provide sparse solutions to this formulation, a direct sparse selection of kernels would be attainable. Such approaches are attractive because they would enable the selection of the optimal area set without the need of using a recursive algorithm, significantly improving the execution time of the learning phase of the classifier. Moreover, it is possible to analyze nonlinear relationships between groups of brain regions by using those methods, thus providing a more general setting to characterize schizophrenia. Finally, it should be stated that even though this approach is useful in schizophrenia detection and characterization, it is not restricted to this disease detection and can be utilized to detect other mental diseases.

5.2 Characterization of schizophrenia using RCK and complex fMRI data

5.2.1 Introduction

Functional magnetic resonance imaging (fMRI) data are acquired at each scan as a bivariate complex image pair for single-channel coil acquisition, containing both the magnitude and the phase of the signal. This complex-valued spatiotemporal data have been shown to contain physiologic information [4]. In fact, it has been shown that there are activation-dependent differences in the phase images as a function of blood flow, especially for voxels with larger venous blood fractions [86]. Based on these findings and on results of some models that showed that phase changes arise only from large non-randomly oriented blood vessels, previous work has focused on filtering voxels with large phase changes [87, 88, 89]. Nonetheless, more recent studies provide evidence that the randomly oriented microvasculature can also produce non-zero blood-oxygen-level-dependent (BOLD)-related phase changes [90, 89], suggesting that the phase information contains useful physiologic information. Furthermore, previous studies have reported task-related fMRI phase changes [4, 88]. The previously discussed findings on the literature provide evidence that phase incorporates information that may help us better understand brain function. For this reason, the present study explores whether phase could improve the detection of functional changes in the brain when combined with magnitude data.

While both magnitude and phase effects are generated by the blood-oxygen-level-dependent mechanism and they both depend on the underlying vascular geometry and the susceptibility change, they primarily depend on different magnetic field characteristics [91]. To first order, the magnitude attenuation depends on the intra-voxel magnetic field inhomogeneity and the phase depends on the mean magnetic field at the voxel. For this reason, it makes sense to think that the inclusion of the phase along with the magnitude could increment the sensitivity to detect informative regions and better discriminate control and patient subjects. Although phase could potentially provide complementary information to magnitude data, most studies discard the phase data. The phase images are usually discarded since their noisy nature poses a challenge for a successful study of fMRI when the processing is performed in the complex domain [92].

Nonetheless, some studies, such as [93, 92], have tried to incorporate phase data on fMRI analyses, but neither of these papers evaluated phase changes at group level. The work in [94] presents a group analysis to evaluate task-related phase changes compared to the task-related magnitude changes in both block-design and event-related tasks. The detection of phase activation in the regions expected to be activated by the task in this study provides further motivation to implement new methods that focus on combining magnitude and phase data to achieve better group inferences.

This study proposes a pattern recognition methodology based on RCK that is capable of attaining a better classification accuracy to differentiate groups of healthy controls and schizophrenia patients by combining fMRI magnitude and phase data. The fMRI data was acquired through an AOD task. In order to overcome the noisy nature of phase data, RFE-SVM [25] is applied to phase data prior to merging it with whole-brain magnitude data. After this preprocessing step, the data is input to RCK.

5.2.2 Materials and Methods

Participants and experimental design

Data from 52 subjects were used, 21 healthy controls aged 18 to 40 years (mean=26.2, SD=7.5) and 31 schizophrenia patients aged 19 to 54 years (mean=30.5, SD=9.2). The experimental design was a three-stimulus AOD task; two runs of auditory stimuli consisting of standard, target, and novel stimuli were presented to the subject. The standard stimulus was a 1000-Hz tone, the target stimulus was a 1500-Hz tone, and the novel stimuli consisted of non-repeating random digital noises. The target and novel stimuli each was presented at a probability of 0.10, and the standard stimuli with a probability of 0.80. The stimulus duration was 200 ms with a 2000-ms stimulus onset asynchrony. Both the target and novel stimuli were always followed by at least 3 standard stimuli. Steps were taken to make sure that all participants could hear the stimuli and discriminate them from the background scanner noise. Subjects were instructed to respond to the target tone with their right index finger and not to respond to the standard tones or the novel stimuli.

Image acquisition

FMRI imaging was performed on a 1.5 T Siemens Avanto TIM system with a 12-channel radio frequency coil. Conventional spin-echo T1-weighted sagittal localizers were acquired for use in prescribing the functional image volumes. Echo planar images were collected with a gradient-echo sequence, modified so that it stored real and imaginary data separately, with the following parameters: FOV = 24 cm, voxel size = $3.75 \times 3.75 \times 4.0$ mm^3, slice gap = 1 mm, number of slices = 27, matrix size = 64×64, TE = 39 ms, TR = 2 s, flip angle = 75 °. The participant's head was firmly secured using a custom head holder. The two stimulus runs consisted of 189 time points each, the first 6 images of each run being discarded to allow for T1 effects to stabilize.

Preprocessing

The magnitude and phase images were written out as 4D NIfTI (Neuroimaging Informatics Technology Initiative) files using a custom reconstruction program on the scanner. Preprocessing of the data was done using the SPM5 software package[1]. The phase images were unwrapped by creating a time series of complex images (real and imaginary) and dividing each time point by the first time point, and then recalculating the phase images. Further phase unwrapping was not required. Magnitude data were co-registered using INRIAlign [95, 72] to compensate for movement in the fMRI time series images. Images were then spatially normalized into the standard Montreal Neurological Institute (MNI) space [73]. Following spatial normalization, the data (originally acquired at $3.75 \times 3.75 \times 4$ mm^3) were slightly upsampled to $3 \times 3 \times 3$ mm^3, resulting in $53 \times 63 \times 46$ voxels. Motion correction and spatial normalization parameters were computed from the magnitude data and then applied

[1] Available at `http://www.fil.ion.ucl.ac.uk/spm/software/spm5/`

to the phase data. The magnitude and phase data were both spatially smoothed with a $10 \times 10 \times 10 - \text{mm}^3$ full-width at half-maximum Gaussian filter. Phase and magnitude data were masked to exclude non-brain voxels.

Creation of spatial maps

A standard general linear model (GLM) analysis on each individual subject was performed using the SPM5 software. Activation maps were computed for magnitude and phase data separately using the multiple regression framework within SPM5, in which regressors are created from the stimulus onset times and convolved with a standard hemodynamic response function in SPM (a combination of two gamma functions which has a peak at 6 s).

Three regressors modeling the target, novel, and standard stimuli were used for each run. Two contrasts for the difference of the target and standard regressors of each run were computed. The resulting contrast images are simply referred to as GLM maps.

Proposed Method

GLM maps from magnitude and phase data were normalized by subtracting from each voxel its mean value across subjects and dividing it by its standard deviation. Next, the GLM maps generated from magnitude data were segmented into 116 regions according to the automated anatomical labeling (AAL) brain parcellation [80] by using the the Wake Forest University pick atlas (WFU-PickAtlas) [75, 76, 77], (`http://www.fmri.wfubmc.edu/download.htm`). If magnitude-only data was analyzed, the information from these brain regions was directly provided to RCK. This algorithm fixed parameter C to 100 and used linear kernels; these kernels were applied variance normalization as explained in 2.3.3. Parameter selection as well as classification accuracy estimation were performed by applying a similar methodology as the one described in 5.1.

In order to incorporate phase data in the analysis, a feature selection procedure (RFE-SVM) [25] was applied to it to get rid of noisy voxels. In this work, 10% of the lowest ranked voxels were discarded at each iteration of RFE-SVM. Next, the activation values of the selected phase voxels were mapped to their corresponding brain regions in the AAL atlas and were combined with the magnitude data present in those regions. Finally, the combination of magnitude and phase data from each region was input to RCK.

Comparison of the proposed method with other algorithms

The proposed algorithm is compared with RFE-SVM and linear SVM for both magnitude-only and magnitude and phase data analyses. The latter algorithms were provided with whole-brain GLM maps, i.e., these maps were not segmented into brain regions for these two methods. In addition, both RFE-SVM and linear SVM were trained with $C = 100$.

5.2.3 Results

Table 5.7 shows the results attained by the classification algorithms for both magnitude only and magnitude and phase data. It can be seen that the results obtained by linear SVM are significantly lower than those achieved by RFE-SVM and the proposed method for both magnitude and complex data. It can also be seen that an improvement is achieved by these methods when phase data is included. Conversely, the performance of linear SVM decays when it is provided with phase data.

The sets of most relevant regions detected by the proposed method when it is provided with either magnitude or complex data are shown in Table 5.8. The upper part of this table shows those regions that are deemed relevant by RCK for both magnitude and complex data, while the lower one displays the relevant regions that are uniquely detected for each type of data. In addition, a graphical representation of the coefficients associated to those regions is presented in Fig. 5.2, which overlay colored regions on top of a structural brain map for each of the two types of data.

5.2.4 Discussion

A classification method that achieves a better classification of fMRI data of healthy controls and schizophrenia patients by combining magnitude and phase data is presented. This work fulfills the need for a methodology that combines magnitude and phase data to achieve better within-group inferences by demonstrating its capacity to improve between-group inferences with the inclusion of phase data.

The classification results obtained by linear SVM presented in Table 5.7, which decay considerably when phase data is included in the analysis, provide evidence that the noisy nature of phase data can preclude a group analysis if this data source is not filtered. Based on the results achieved by the other classification approaches, RFE-SVM proves to be a sensible choice for phase data filtering, being able to extract informative voxels from phase data and making it possible to get better classification results with the inclusion of phase information.

RCK results provide evidence of disturbances in the temporal lobe in schizophrenia, since this region is detected in both the magnitude and the complex data. In addition, the complex data analysis reveals that phase data shows group discriminating activity in other brain regions, such as the cingulate gyri, which is informative for schizophrenia detection. In fact, the presence of the anterior cingulate gyrus among the relevant regions is important, for it has been proposed that error-related activity in the anterior cingulate cortex is impaired in patients with schizophrenia [82].

Table 5.1: Optimal area set and associated discriminative weights for RCK analysis applied to ICA default mode data. The most informative anatomical regions retrieved by RCK when applied to ICA default mode data are grouped in macro brain regions to give a better idea of their location in the brain. The mean and the standard deviation of the discriminative weights of each area are listed in this table. In addition the rate of training sets in the cross-validation procedure that selected each area (selection frequency) is also reported in order to measure the validity of the inclusion of each region in the optimal area set.

Source	Areas and Discriminative Weights				
	Macro Regions	Regions	Discriminative Weights		
			Mean	Std. Dev.	Sel. Freq.
ICA DMN	Central Region	Right Precentral Gyrus	2.32	0.06	1.00
		Left Precentral Gyrus	2.31	0.04	1.00
		Left Postcentral Gyrus	2.22	0.03	1.00
		Right Postcentral Gyrus	2.21	0.02	1.00
	Frontal lobe	Right Paracentral Lobule	3.44	0.16	1.00
		Left Superior Frontal Gyrus, Medial	2.97	0.15	1.00
		Left Middle Frontal Gyrus, Orbital Part 1	2.52	0.15	1.00
		Right Superior Frontal Gyrus, Medial	2.51	0.10	1.00
		Left Superior Frontal Gyrus	2.28	0.09	1.00
		Right Superior Frontal Gyrus	2.27	0.06	1.00
		Left Inferior Frontal Gyrus, Triangular Part	2.24	0.04	1.00
		Right Middle Frontal Gyrus	2.21	0.04	0.94
		Right Inferior Frontal Gyrus, Opercular Part	2.19	0.08	0.79
		Left Inferior Frontal Gyrus, Orbital Part	2.16	0.08	0.55
		Right Gyrus Rectus	2.38	0.21	0.94
	Temporal lobe	Left Middle Temporal Gyrus	2.27	0.03	1.00
		Right Middle Temporal Gyrus	2.22	0.05	1.00
	Parietal lobe	Left Angular Gyrus	2.72	0.11	1.00
		Left Supramarginal Gyrus	2.45	0.11	1.00
		Right Cuneus	2.72	0.08	1.00
		Right Superior Parietal Gyrus	2.31	0.06	1.00
		Left Superior Parietal Gyrus	2.25	0.08	0.96
	Occipital lobe	Right Superior Occipital Gyrus	2.94	0.13	1.00
		Left Superior Occipital Gyrus	2.88	0.09	1.00
		Left Middle Occipital Gyrus	2.58	0.07	1.00
		Right Inferior Occipital Gyrus	2.50	0.14	1.00
		Left Cuneus	2.38	0.07	1.00
		Left Fusiform Gyrus	2.31	0.05	1.00
	Limbic lobe	Left Anterior Cingulate Gyrus	3.33	0.10	1.00
		Right Anterior Cingulate Gyrus	2.71	0.09	1.00
		Right Middle Cingulate Gyrus	2.46	0.06	1.00
		Left Middle Cingulate Gyrus	2.41	0.06	1.00
		Left Temporal Pole: Middle Temporal Gyrus	2.40	0.13	1.00
		Right Temporal Pole: Superior Temporal Gyrus	2.36	0.10	0.96
		Left Parahippocampal Gyrus	2.27	0.11	0.87

Table 5.1: (Cont'd) Optimal area set and associated discriminative weights for RCK analysis applied to ICA default mode data. The most informative anatomical regions retrieved by RCK when applied to ICA default mode data are grouped in macro brain regions to give a better idea of their location in the brain. The mean and the standard deviation of the discriminative weights of each area are listed in this table. In addition the rate of training sets in the cross-validation procedure that selected each area (selection frequency) is also reported in order to measure the validity of the inclusion of each region in the optimal area set.

Source	Areas and Discriminative Weights				
	Macro Regions	Regions	Discriminative Weights		
			Mean	Std. Dev.	Sel. Freq.
ICA DMN	Insula	Right Insular Cortex	2.25	0.07	0.98
	Sub cortical gray cortex	Left Thalamus	2.53	0.12	1.00
	Cerebellum	Right Inferior Posterior Lobe of Cerebellum	3.83	0.19	1.00
		Left Anterior Lobe of Cerebellum	2.35	0.07	1.00
		Left Superior Posterior Lobe of Cerebellum	2.32	0.07	1.00

Table 5.2: Optimal area set and associated discriminative weights for RCK analysis applied to ICA temporal lobe data. The most informative anatomical regions retrieved by RCK when applied to ICA temporal lobe data are grouped in macro brain regions to give a better idea of their location in the brain. The mean and the standard deviation of the discriminative weights of each area are listed in this table. In addition the rate of training sets in the cross-validation procedure that selected each area (selection frequency) is also reported in order to measure the validity of the inclusion of each region in the optimal area set.

Source	Areas and Discriminative Weights				
	Macro Regions	Regions	Discriminative Weights		
			Mean	Std. Dev.	Sel. Freq.
ICA TL	Central region	Right Rolandic Operculum	8.63	0.25	1.00
		Left Precentral Gyrus	7.70	0.09	1.00
	Frontal lobe	Left Inferior Frontal Gyrus, Orbital Part	7.79	0.21	1.00
		Right Superior Frontal Gyrus, Medial	7.58	0.10	0.96
		Right Superior Frontal Gyrus	7.56	0.05	1.00
	Temporal lobe	Right Middle Temporal Gyrus	7.39	0.04	0.81
	Occipital lobe	Right Middle Occipital Gyrus	7.97	0.09	1.00
		Left Middle Occipital Gyrus	7.67	0.15	1.00
		Right Fusiform Gyrus	7.57	0.12	0.98
		Right Calcarine Fissure	7.46	0.11	0.83
	Limbic lobe	Left Middle Cingulate Gyrus	7.67	0.11	1.00
	Insula	Left Insular Cortex	7.64	0.12	1.00
	Cerebellum	Right Inferior Posterior Lobe of Cerebellum	7.36	0.25	0.42

Table 5.3: Optimal area set and associated discriminative weights for RCK analysis applied to GLM data. The most informative anatomical regions retrieved by RCK when applied to GLM data are grouped in macro brain regions to give a better idea of their location in the brain. The mean and the standard deviation of the discriminative weights of each area are listed in this table. In addition the rate of training sets in the cross-validation procedure that selected each area (selection frequency) is also reported in order to measure the validity of the inclusion of each region in the optimal area set.

Source	Areas and Discriminative Weights				
	Macro Regions	Regions	Discriminative Weights		
			Mean	Std. Dev.	Sel. Freq.
GLM	Central region	Left Postcentral Gyrus	3.12	0.16	1.00
		Right Precentral Gyrus	2.78	0.12	1.00
		Left Precentral Gyrus	2.67	0.09	1.00
		Right Postcentral Gyrus	2.64	0.12	1.00
	Frontal lobe	Left Superior Frontal Gyrus	4.12	0.12	1.00
		Right Middle Frontal Gyrus	4.02	0.14	1.00
		Left Inferior Frontal Gyrus, Triangular Part	3.64	0.19	1.00
		Left Middle Frontal Gyrus	3.45	0.12	1.00
		Left Middle Frontal Gyrus, Orbital Part 2	3.15	0.17	1.00
		Right Superior Frontal Gyrus	2.71	0.10	1.00
		Left Middle Frontal Gyrus, Orbital Part 1	2.59	0.17	1.00
		Left Supplementary Motor Area	2.48	0.12	1.00
		Left Superior Frontal Gyrus, Medial	2.43	0.10	1.00
		Right Inferior Frontal Gyrus, Orbital Part	2.31	0.16	0.96
		Right Superior Frontal Gyrus, Medial	2.23	0.11	1.00
		Left Inferior Frontal Gyrus, Opercular Part	2.15	0.12	0.98
		Left Inferior Frontal Gyrus, Orbital Part	2.10	0.11	0.92
		Right Paracentral Lobule	2.07	0.16	0.83

Table 5.3: (Cont'd) Optimal area set and associated discriminative weights for RCK analysis applied to GLM data. The most informative anatomical regions retrieved by RCK when applied to GLM data are grouped in macro brain regions to give a better idea of their location in the brain. The mean and the standard deviation of the discriminative weights of each area are listed in this table. In addition the rate of training sets in the cross-validation procedure that selected each area (selection frequency) is also reported in order to measure the validity of the inclusion of each region in the optimal area set.

Source	Areas and Discriminative Weights				
	Macro Regions	Regions	Discriminative Weights		
			Mean	Std. Dev.	Sel. Freq.
GLM	Temporal lobe	Right Middle Temporal Gyrus	3.87	0.13	1.00
		Left Superior Temporal Gyrus	2.79	0.15	1.00
		Right Superior Temporal Gyrus	2.37	0.12	1.00
		Left Middle Temporal Gyrus	2.30	0.07	1.00
		Left Inferior Temporal Gyrus	2.28	0.14	1.00
		Right Inferior Temporal Gyrus	2.14	0.08	0.98
	Parietal lobe	Right Precuneus	2.35	0.10	1.00
		Left Inferior Parietal Gyrus	2.18	0.17	0.96
	Occipital lobe	Left Calcarine Fissure	3.00	0.19	1.00
		Right Fusiform Gyrus	2.55	0.13	1.00
		Right Middle Occipital Gyrus	2.50	0.11	1.00
	Limbic lobe	Right Hippocampus	2.27	0.12	1.00
		Right Middle Cingulate Gyrus	2.24	0.08	1.00
		Right Anterior Cingulate Gyrus	2.21	0.12	0.98
	Insula	Left Insular Cortex	1.96	0.07	0.42
	Sub cortical gray nuclei	Right Caudate Nucleus	2.30	0.14	1.00
		Right Amygdala	2.26	0.15	0.98
	Cerebellum	Anterior Lobe of Vermis	2.83	0.21	1.00
		Posterior Lobe of Vermis	2.67	0.22	1.00
		Right Inferior Posterior Lobe of Cerebellum	2.30	0.16	0.98

Table 5.4: Optimal area set and associated discriminative weights for RCK analysis applied to multi-source data. The most informative anatomical regions retrieved by RCK when applied to 3 data sources are grouped in macro brain regions to give a better idea of their location in the brain. The mean and the standard deviation of the discriminative weights of each area are listed in this table. In addition the rate of training sets in the cross-validation procedure that selected each area (selection frequency) is also reported in order to measure the validity of the inclusion of each region in the optimal area set.

Source	Areas and Discriminative Weights				
	Macro Regions	Regions	Discriminative Weights		
			Mean	Std. Dev.	Sel. Freq.
ICA DMN	Central region	Right Precentral Gyrus	3.10	0.13	1.00
		Left Precentral Gyrus	2.49	0.08	1.00
		Left Rolandic Operculum	2.18	0.15	0.89
	Frontal lobe	Left Superior Frontal Gyrus	3.06	0.11	1.00
		Left Superior Frontal Gyrus, Medial	3.05	0.15	1.00
		Right Paracentral Lobule	2.94	0.16	1.00
		Right Gyrus Rectus	2.66	0.20	1.00
		Right Superior Frontal Gyrus, Medial	2.50	0.10	1.00
	Temporal lobe	Right Middle Temporal Gyrus	2.30	0.08	1.00
		Left Middle Temporal Gyrus	2.09	0.11	0.74
	Parietal lobe	Left Angular Gyrus	3.44	0.22	1.00
	Occipital lobe	Left Superior Occipital Gyrus	2.62	0.15	1.00
		Left Middle Occipital Gyrus	2.59	0.15	1.00
		Left Fusiform Gyrus	2.55	0.12	1.00
		Right Cuneus	2.35	0.14	0.98
		Left Cuneus	2.30	0.12	1.00
	Limbic lobe	Parahippocampal Gyrus	2.45	0.14	0.98
		Left Middle Cingulate Gyrus	2.36	0.11	1.00
		Left Anterior Cingulate Gyrus	2.29	0.11	1.00
	Cerebellum	Right Inferior Posterior Lobe of Cerebellum	2.93	0.20	1.00
		Left Superior Posterior Lobe of Cerebellum	2.58	0.13	1.00
		Left Anterior Lobe of Cerebellum	2.37	0.14	0.98
ICA TL	Central region	Right Rolandic Operculum	2.33	0.13	0.98
	Frontal lobe	Right Inferior Frontal Gyrus Triangular Part	2.77	0.13	1.00
		Right Superior Frontal Gyrus	2.55	0.11	1.00
	Temporal lobe	Left Heschl gyrus	2.54	0.17	1.00
		Left Middle Temporal Gyrus	2.28	0.12	1.00
		Right Inferior Temporal Gyrus	2.24	0.11	0.98
		Right Middle Temporal Gyrus	2.18	0.09	0.98

Table 5.4: (Cont'd) Optimal area set and associated discriminative weights for RCK analysis applied to multi-source data. The most informative anatomical regions retrieved by RCK when applied to 3 data sources are grouped in macro brain regions to give a better idea of their location in the brain. The mean and the standard deviation of the discriminative weights of each area are listed in this table. In addition the rate of training sets in the cross-validation procedure that selected each area (selection frequency) is also reported in order to measure the validity of the inclusion of each region in the optimal area set.

Source	Areas and Discriminative Weights				
	Macro Regions	Regions	Discriminative Weights		
			Mean	Std. Dev.	Sel. Freq.
ICA TL	Occipital lobe	Right Middle Occipital Gyrus	2.44	0.11	1.00
		Left Middle Occipital Gyrus	2.16	0.11	0.94
	Limbic lobe	Left Middle Cingulate Gyrus	2.38	0.13	1.00
	Sub cortical gray nuclei	Left Caudate Nucleus	2.52	0.13	1.00
	Cerebellum	Left Anterior Lobe of Cerebellum	2.47	0.16	1.00
		Right Cerebellar Tonsil	2.25	0.19	0.98
		Right Posterior Lobe of Cerebellum	2.08	0.15	0.58
GLM	Frontal lobe	Left Middle Frontal Gyrus, Orbital Part	2.36	0.16	1.00
		Right Middle Frontal Gyrus	2.23	0.13	0.98
	Limbic lobe	Right Hippocampus	2.44	0.14	1.00
	Cerebellum	Posterior Lobe of Vermis	2.56	0.18	1.00

Table 5.5: Mean classification accuracy achieved by different algorithms using single-source data. The reported results indicate the mean classification rate attained by different algorithms for each data source using the data from all the brain regions included in the AAL brain parcellation.

	Default Mode	Temporal Lobe	GLM
Composite Kernels	0.75	0.64	0.74
Linear SVM	0.75	0.54	0.67
Gaussian SVM	0.75	0.62	0.75
RFE-SVM	0.87	0.75	0.71
RCK	0.90	0.85	0.86

Table 5.6: Mean classification accuracy achieved by different algorithms using multi-source data. The reported results indicate the mean classification rate attained by different algorithms provided with all possible combinations of data sources. The analysis is performed using all brain regions included in the AAL brain parcellation.

	Two Sources			All Sources
	Default & Temp	Default & GLM	Temp & GLM	
Composite Kernels	0.70	0.70	0.69	0.79
Linear SVM	0.79	0.78	0.62	0.80
Gaussian SVM	0.76	0.77	0.70	0.80
RFE-SVM	0.92	0.90	0.84	0.90
RCK	0.92	0.93	0.85	0.95

Table 5.7: Mean classification accuracy and sensitivity/specificity achieved by different algorithms using magnitude only and magnitude and phase data. In the case of RCK, whole-brain magnitude data is used for the first analysis, while RFE-SVM filtered phase data is combined with whole-brain magnitude data for the second analysis.

	Magnitude Data		Magnitude and Phase Data	
	Accuracy	Sens/Spec	Accuracy	Sens/Spec
Linear SVM	0.64	0.77/0.45	0.54	0.64/0.38
RFE-SVM	0.72	0.77/0.64	0.77	0.82/0.69
Proposed Method	0.70	0.82/0.55	0.81	0.87/0.71

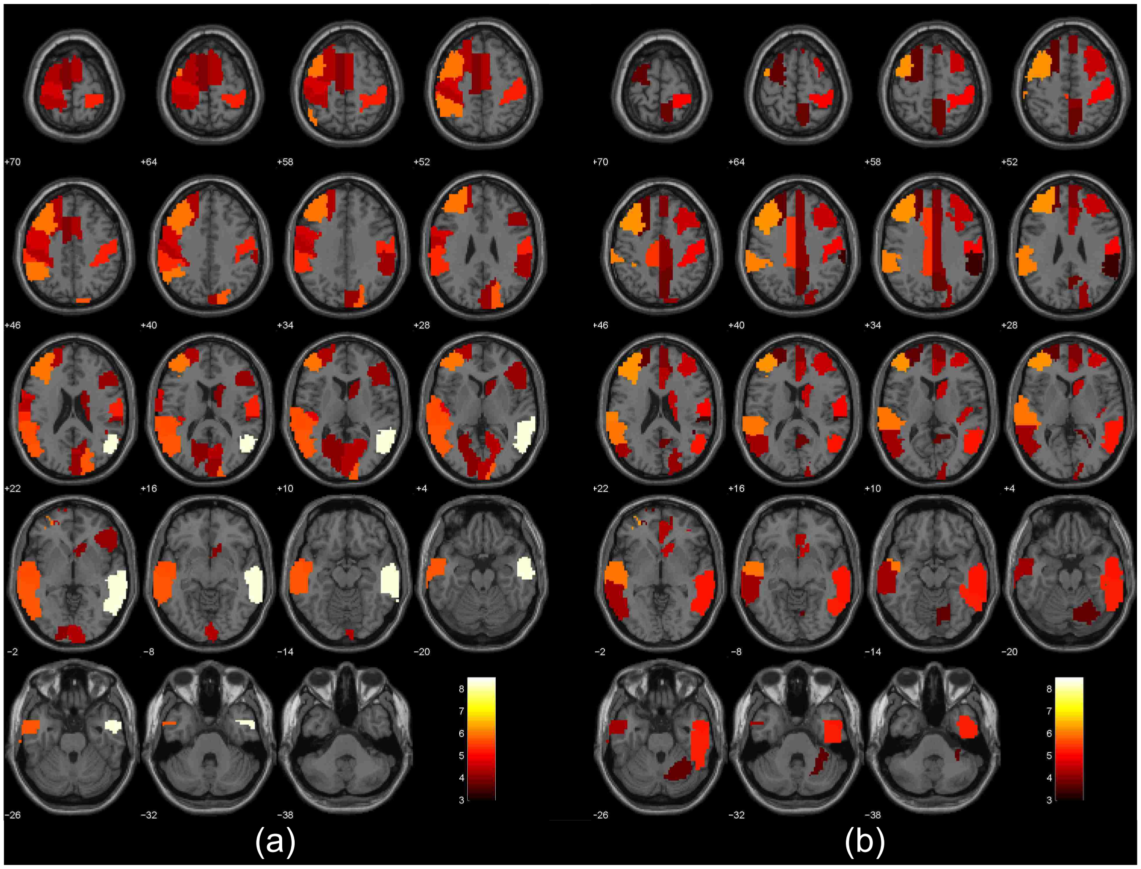


Figure 5.2: Discriminative weights brain maps for (a) magnitude data and (b) complex data. The brain maps generated for these two analyses highlight the brain regions that achieved a selection frequency greater than 0.50 for both experiments. These areas are color-coded according to their associated discriminative weights.

Table 5.8: Optimal area set and associated mean discriminative weights and selection frequencies for RCK analysis applied to magnitude only and magnitude and phase data. The top of the table displays the brain regions that are deemed relevant by both analyses, followed by the regions that are relevant for either magnitude or magnitude and phase data, respectively.

Brain Regions	Magnitude Data		Magnitude and Phase Data	
	Discr. Weight	Sel. Freq.	Discr. Weight	Sel. Freq.
Left middle frontal gyrus	5.96	0.87	6.19	1.00
Left supramarginal gyrus	5.37	0.92	6.09	1.00
Left superior temporal gyrus	5.60	0.83	5.98	1.00
Right middle temporal gyrus	8.32	0.62	5.14	0.96
Right postcentral gyrus	5.25	0.73	4.99	0.98
Right Caudate Nucleus	4.05	0.69	4.46	0.69
Left Heschl gyrus	4.87	0.58	4.46	0.90
Left middle temporal gyrus	5.68	0.52	4.20	0.62
Right superior occipital gyrus	5.73	0.62	4.10	0.58
Left superior frontal gyrus	4.27	0.58	3.66	0.56
Right supramarginal gyrus	4.20	0.60	3.41	0.56
Left inferior parietal lobule	5.97	0.62	-	-
Right calcarine fissure	4.39	0.73	-	-
Left postcentral gyrus	4.36	0.52	-	-
Right supplementary motor area	4.27	0.56	-	-
Right inferior frontal gyrus	4.17	0.58	-	-
Right cuneus	4.16	0.56	-	-
Left calcarine fissure	4.01	0.54	-	-
Left supplementary motor area	3.97	0.58	-	-
Left middle cingulate gyrus	-	-	5.43	0.98
Right inferior temporal gyrus	-	-	5.24	0.85
Right middle frontal gyrus	-	-	4.56	0.90
Right anterior cingulate gyrus	-	-	4.39	0.75
Right middle cingulate gyrus	-	-	4.09	0.71
Right superior frontal gyrus, med	-	-	3.97	0.65
Right precuneus	-	-	3.77	0.56
Right posterior lobe of cerebellum	-	-	3.71	0.54

5.3 Characterization of schizophrenia using ν-MKL and complex-valued fMRI data

5.3.1 Introduction

This work uses data from the same dataset analyzed by the approach presented on 5.2. However, both analyses use different sets of subjects. In particular, this work retrieves data from controls and patients that are better matched in terms of age.

Methods that are capable of combining different data sources can be applied to fMRI in order to efficiently use the information present in the magnitude and phase of the data. Such methods should also consider that fMRI data, though high dimensional, show sparsely distributed activation in the brain. In other words, a significant number of voxels will not convey information of brain activity. Moreover, informative voxels are likely to be distributed in clusters or brain regions. For these reasons, an adequate method to combine magnitude and phase fMRI data should also be able to automatically select the regions that characterize the condition under study.

Among the various approaches that are well-suited to solve this problem, group least angle shrinkage and selection operator (Group LASSO) [96] or nonlinear approaches such as multiple kernel learning (MKL) methods [97] are the most commonly used methods to carry out group or kernel selection. In particular, MKL algorithms can be used to do group selection if a kernel is defined on each group. There are two advantages of applying kernels to different groups on fMRI data. On the one hand, one can exploit linear or nonlinear relationships among the voxels of the same group just by using linear (Euclidean dot product) or nonlinear kernels. On the other hand, MKL admits a dual formulation, in such a way that the computational complexity of the problem is defined by the number of samples rather than the number of voxels per sample. For fMRI data, this translates into a dramatic complexity reduction with respect to the primal formulation.

Several MKL algorithms have been devised in the last decade. The optimization of a weighted linear combination of kernels for the support vector machine (SVM) was proposed in [28]. Their formulation reduces to a convex optimization problem, namely a quadratically-constrained quadratic program (QCQP). Later, [30] proposed a dual formulation of this QCQP as a second-order cone programming problem, which improved the running time of the algorithm. Afterwards, [33] reformulated the algorithm proposed by Bach et al. as a semi-infinite linear program, which amounts to repeatedly training an SVM on a mixture kernel while iteratively refining the kernel coefficients. The above mentioned algorithms attempt to achieve sparsity by promoting sparse solutions in terms of the kernel coefficients. Specifically, both [30] and [33] enforced sparsity by using l_1-norm regularization terms on these coefficients, an approach that has exhibited certain limitations for linear SVM [34, 35]. Alternative solutions can be found in [38], where a non-sparse MKL formulation based on an l_p-norm regularization term on the kernel coefficients (with $p \geq 1$) is introduced, or in [37], which mixes elements of l_p-norm and elastic net regularization.

Keeping in mind the aforementioned reasoning, the aim of the present work is to differentiate groups of healthy controls and schizophrenia patients from an auditory oddball discrimination (AOD) task by efficiently combining magnitude and phase information. To do so, we propose a novel MKL formulation that automatically selects the regions that are relevant for the classification task. First, we apply group independent component analysis (ICA) [57] separately to both magnitude and phase data to extract activation patterns from both sources. Next, given the local-oriented nature of the proposed MKL methodology, local (per-region) recursive feature elimination SVM (RFE-SVM) [25] is applied to magnitude and phase data to extract only their relevant information. Then, following the recursive composite kernels scheme presented in [61], each one of the defined brain regions is used to construct a kernel, after which our proposed MKL formulation is applied to select the most informative ones. The novelty of this formulation, which is based on the work presented in [46], relies on the addition of a parameter (ν) that allows the user to preset an upper bound of the number of kernels to be included in the final classifier. We call this algorithm ν-MKL.

Based on this procedure, we present three possible variants of the algorithm. In the first one, the assumption of magnitude and phase data belonging to a joint distribution is adopted. Therefore, they are concatenated, RFE-SVM is applied to each region, and the selected voxels of each of them are used to construct the kernels. In the second one, RFE-SVM is applied independently to magnitude and phase for each region, after which the selected voxels are concatenated to construct kernels. In the third approach, we assume that magnitude and phase come from independent distributions, so RFE-SVM is applied independently to both of them and kernels are constructed from magnitude and phase data without concatenation. The second and third approaches are significantly different for nonlinear kernels. Concatenating the data prior to kernel computation assumes nonlinear dependencies between magnitude and phase, whereas computing separate kernels assumes linear dependence. For the case of linear kernels, the difference relies on the fact that separate kernels allow the algorithm to assign different weights (and thus different importance) to the magnitude and phase data representations of the regions.

The proposed approach is tested using linear and Gaussian kernels. In addition, the performance of ν-MKL is further evaluated by comparing its results in terms of classification accuracy with those obtained by applying l_p-norm MKL [38] and SVM. Furthermore, the estimates of the sparsity of the problem of both MKL algorithms are also used for comparison purposes. However, both the actual degree of sparsity of the real dataset and the degree of differential activity present on each region are unknown. For this reason, a synthetic dataset where this information can be estimated a priori is generated to verify the capacity of ν-MKL to detect both the sparsity of the problem and the amount of information present in the analyzed brain regions, which is then compared to the one attained by l_p-norm MKL.

5.3.2 Materials and Methods

Synthetic dataset for preliminary test

Since both the sparsity and the degree of differential activity of each brain region of the real data are unknown, the performance of ν-MKL and l_p-norm MKL cannot be fully assessed. To compensate for that, a synthetic dataset that properly matches the real data for analysis purposes is analyzed. This dataset, which is generated using the simulation toolbox for fMRI data (SimTB), mimics the BOLD response of two groups of subjects with different brain activation patterns.

SimTB generates data under the assumption of spatiotemporal separability, i.e., that data can be expressed as the product of time courses and spatial maps. Default spatial maps are modeled after components commonly seen in axial slices of real fMRI data and most are created by combinations of simple Gaussian distributions, while time courses are constructed under the assumption that component activations result from underlying neural events as well as noise. Neural events can follow block or event-related experimental designs, or can represent unexplained deviations from baseline; these are referred to as unique events. The time course of each component is created by adding together amplitude-scaled task blocks, task events and unique events by means of modulation coefficients, as shown in Fig. 4.1.

The generated experimental design is characterized by the absence of task events, the BOLD response being characterized by unique events only, thus being similar to a resting-state experiment. The spatial maps generated for all components did not exhibit any consistent changes among groups, the exception being the default mode network. For this specific component, changes in the activation coefficients between groups were induced by slightly shifting them in the vertical axis. By doing so, it is expected that differential activation is generated in the voxels within the Gaussian blobs representing the anterior and posterior cingulate cortex as well as the left and right angular gyri.

The experimental design is simulated for two groups of $M = 200$ subjects, each subject with $C = 20$ components in a data set with $V = 100 \times 100$ voxels and $T = 150$ time points collected at TR = 2 seconds. Among the 30 components available by default on SimTB, we did not include in the simulation those associated with the visual cortex, the precentral and postcentral gyri, the subcortical nuclei and the hippocampus. To mimic between-subject spatial variability, the components for each subject are given a small amount of translation, rotation, and spread via normal deviates.

Translation in the horizontal and vertical directions of each source have a standard deviation of 0.1 voxels, except for the default mode network. This component has different vertical translation between groups. Both of them have a standard deviation of 0.5 voxels, but different means (0.7 and -0.7 for groups 1 and 2, respectively). In addition, rotation has a standard deviation of 1 degree, and spread has a mean of 1 and standard deviation of 0.03.

All components have unique events that occur with a probability of 0.5 at each TR and unique event modulation coefficients equal to 1. At the last stage of the data generation pipeline, Rician noise is added to the data of each subject to reach the appropriate CNR level, which is equal to 0.3 for all subjects.

Complex-valued real dataset

Participants The set of subjects is composed of 21 controls and 31 patients. Controls aged 19 to 40 years (mean=26.6, SD=7.4) and patients aged 18 to 49 years (mean=27.7, SD=8.2). A two-sample t-test on age yielded $t = 0.52$ (p-value = 0.60). There were 8 male controls and 21 male patients.

Experimental Design The subjects followed a three-stimulus AOD task; two runs of 244 auditory stimuli consisting of standard, target, and novel stimuli were presented to the subject. The standard stimulus was a 1000-Hz tone, the target stimulus was a 1500-Hz tone, and the novel stimuli consisted of non-repeating random digital noises. The target and novel stimuli each was presented with a probability of 0.10, and the standard stimuli with a probability of 0.80. The stimulus duration was 200 ms with a 2000-ms stimulus onset asynchrony. Both the target and novel stimuli were always followed by at least 3 standard stimuli. Steps were taken to make sure that all participants could hear the stimuli and discriminate them from the background scanner noise. Subjects were instructed to respond to the target tone with their right index finger and not to respond to the standard tones or the novel stimuli.

Data processing

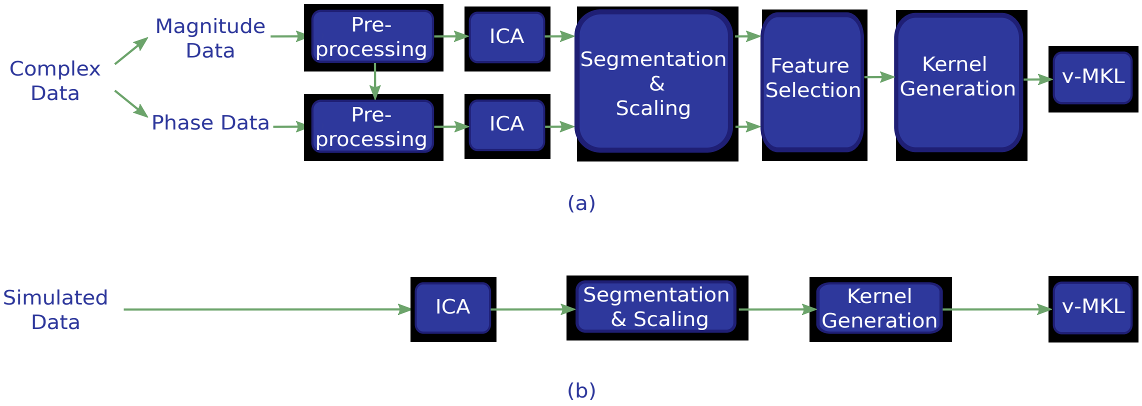
The analysis pipelines of both the simulated and the complex-valued fMRI datasets are shown in Fig. 5.3. The processing stages that are applied to these datasets are explained in what follows.

Group spatial ICA As shown in Fig. 5.3, group spatial ICA [57] is applied to both the simulated and the complex-valued fMRI datasets to decompose the data into independent components using the GIFT software[2]. Group ICA is used due to

[2] Available at `http://mialab.mrn.org/software/`

its extensive application to fMRI data for schizophrenia characterization [65, 66, 67]. We also attempted to train the proposed method with activation maps retrieved by the general linear model, but it performed better when provided with ICA data.

ICA was applied to magnitude and phase data separately for the complex-valued fMRI dataset. Dimension estimation, which was used to determine the number of components, was performed using the minimum description length criteria, modified to account for spatial correlation [74]. For both data sources, the estimated number of components was 20. Data from all subjects were then concatenated and this aggregate data set reduced to 20 temporal dimensions using principal component analysis (PCA), followed by an independent component estimation using the infomax algorithm [58]. Individual subject components were then back-reconstructed from the group ICA analyses to retrieve the spatial maps (ICA maps) of each run (2 AOD task runs) for each data source.



(a)

(b)

Figure 5.3: Data processing stages of (a) the complex-valued fMRI dataset and (b) the simulated dataset. On the preprocessing stage of the complex-valued fMRI data, motion correction and spatial normalization parameters were computed from the magnitude data and then applied to the phase data. Next, ICA was applied to magnitude and phase data separately, a single component being selected for each data source. Individual subject components were then back-reconstructed from the group ICA maps of each run (2 ICA maps per subject for each data source).

To reduce the complexity of the analysis of magnitude and phase data, a single component was selected for each data source. These components were selected as follows. For magnitude data, we found three task-related components: the temporal lobe component (t-value=13.8, p-value=5.88×10^{-19}), the default mode network (t-value=-11.0, p-value=4.57×10^{-15}) and the motor lobe component (t-value=8.0, p-value=1.47×10^{-10}). Among these three candidates, the most-discriminative task-related component was selected within a nested cross-validation (CV) procedure; this is explained on detail later on *Parameter validation, feature selection and prediction accuracy estimation*. For phase data, we only found one task-related component: the posterior temporal lobe component (t-value=-2.29, p-value=0.02). While phase data does not show as strong a task response as magnitude data, it appears to be useful for discriminative purposes.

On the other hand, the simulated dataset was decomposed into 20 components as follows. First, data from all subjects were temporally concatenated into a group matrix, being reduced to 20 temporal dimensions by using PCA. Then, an independent component estimation was applied to these reduced aggregate dataset using the infomax algorithm. Finally, individual subject components were back-reconstructed from the group ICA analysis.

To make the analysis of the simulated data resemble that of the complex-valued data as much as possible, the subjects' ICA maps associated to a single component were analyzed for this dataset. This component was the default mode network, which was modeled to present differential activity between groups, as explained on *Simulated dataset*.

Data segmentation and scaling As shown in Fig. 5.3, data segmentation is applied to both datasets. For the complex-valued one, this is applied to the individual ICA maps associated to the magnitude component and the posterior temporal lobe component for phase data. One of the objectives of the proposed approach is to locate the regions that better characterize schizophrenia through a multivariate analysis. To do so, an appropriate brain segmentation needs to be used. An adequate segmentation would properly capture functional regions in the brain and cover it entirely, as spatial smoothing may spread brain activation across neighboring regions. Unfortunately, anatomical templates such as the automated anatomical labeling (AAL) brain parcellation [80] may not capture functional regions given their large spatial extent. In fact, these regions are defined by brain structure. Furthermore, they do not cover the entire brain.

One way of solving the problem of properly representing functional regions is to use a more granular segmentation of the brain. This could be attained by using a relatively simple cubical parcellation approach. We divided the brain into $9 \times 9 \times 9$-voxel cubical regions; the first cube is located at the center of the 3-D array were brain data is stored and the rest of them are generated outwards, increasingly further from the center. A total number of 158 cubical regions containing brain voxels were generated by using a whole-brain mask together with the cubical parcellation. It should be highlighted that by applying this approach the data has not been downsampled, as the original voxels are preserved for posterior analysis. Another advantage of using the cubical regions instead of an anatomical atlas is that we do not incorporate prior knowledge of the segmentation of functional regions in the brain, letting the algorithm figure out automatically which regions are informative.

Our MKL-based methodology evaluates the information within regions under the assumption that active voxels are clustered, an inactive voxel being one with coefficients equal to zero across ICA maps for all subjects. This assumption would not hold for regions composed of few scattered voxels. To avoid such cases, those regions containing less than 10 active voxels were not considered valid and were not included in our analysis. Nonetheless, a post-hoc analysis of this threshold value showed that it does not significantly change the results of the proposed approach.

A similar segmentation procedure was used for the simulated dataset, where the analyzed spatial maps where divided into 9×9-voxel square regions. These data parcellation generated a total number of 109 square regions. Furthermore, each voxel activation level was normalized for both datasets. This was done by subtracting its mean value across subjects and dividing it by its standard deviation.

Region representation For the complex-valued fMRI dataset, the ICA maps associated to magnitude and phase sources are segmented in cubical regions, while the ICA maps extracted from the simulated dataset are segmented in square regions, as stated in the previous section. The term region will be used hereafter to refer to either of these to be able to explain the following processing stages regardless of the analyzed dataset. Nonetheless, the procedure described on this section is applicable to the complex-valued dataset only.

Per-region feature selection is applied to magnitude and phase data either for single-source analysis or for data source combination. For the former case, local (per-region) RFE-SVM is directly applied to the analyzed data source, while for the combination of both sources local RFE-SVM (hereafter referred to simply as RFE-SVM) is applied to the data using two strategies:

- The data from both magnitude and phase are concatenated prior to the application of RFE-SVM, under the assumption that both magnitude and phase data come from a joint distribution. We refer to this approach as joint feature selection.
- RFE-SVM is applied independently to each data source. In this case, we assume that magnitude and phase come from independent distributions. We refer to this approach as independent feature selection.

Region characterization The information within each region is characterized by means of a dot product matrix (Gram matrix in Euclidean space), which provides a pairwise measure of similarity between subjects for that region. This representation enables the selection of informative regions via an MKL formulation, which is explained later on this chapter.

As mentioned in the previous section, magnitude and phase are analyzed either separately or together. For single-source analysis, the generation of a Gram matrix for each region is straightforward. Conversely, three combination approaches are proposed to combine magnitude and phase data based on the used region representation. The first one computes the Gram matrix of each region right after joint feature selection is applied. The second one concatenates the outputs of independent feature selection for the computation of the Gram matrix, while the third one generates a Gram matrix from each output of the independent feature selection. This is graphically summarized on Fig. 5.4 and their rationale has already been discussed on the introduction.

We now provide a brief explanation of the application of dot products on regions' data in the context of our proposed methodology. Let us assume that we are given N labeled training data $(\mathbf{x}_i, y_i)$, where the examples $\mathbf{x}_i$ are represented as vectors of d features and $y_i \in \{-1, 1\}$. In this case, the examples lie on $\mathcal{X} = \mathbb{R}^d$, which is called input space. Let us further assume that features are divided in L blocks such that $\mathbb{R}^d = \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_L}$, so that each example $\mathbf{x}_i$ can be decomposed into these L blocks, i.e., $\mathbf{x}_i = [\mathbf{x}_{i,1}^T, \ldots, \mathbf{x}_{i,L}^T]^T$. In the case of our study, these blocks represent brain regions. Given two examples $\mathbf{x}_i$, $\mathbf{x}_j$, their data representations for region l are $\mathbf{x}_{i,l} = [x_{i,l}^1, \ldots, x_{i,l}^{d_l}]^T$ and $\mathbf{x}_{j,l} = [x_{j,l}^1, \ldots, x_{j,l}^{d_l}]^T$, respectively. The dot product of these two examples for region l is defined by

$$\langle \mathbf{x}_{i,l}, \mathbf{x}_{j,l} \rangle = \mathbf{x}_{i,l}^T \mathbf{x}_{j,l} = \sum_{k=1}^{d_l} x_{i,l}^k x_{j,l}^k,$$

which outputs a scalar value that equals 0 if both vectors are orthogonal.

Our proposed MKL approach is initially cast as a linear formulation to be optimized in dual space, although it is possible to solve its primal problem too. The reasons why we solve the dual problem are twofold. First, by working with the dual formulation the computational complexity of the problem is defined by the number of available data points instead of the number of features per data point. For fMRI data this amounts to a significant reduction in computational complexity with respect to the primal formulation. Second, the dual formulation can be easily extended to account for nonlinear relationships among voxels of a given region, as it will be explained later. However, increasing the model complexity is not guaranteed to be advantageous, due to the limited amount of data and their high dimensionality.

Normalization of kernels is very important for MKL as feature sets can be scaled differently for diverse data sources. In our framework, the evaluation of dot products on areas composed of different numbers of active voxels yields values in different scales. To compensate for that, unit variance normalization is applied to the computed Gram matrices, as specified on section 2.3.3.

More formally, let l be a region index and $\mathbf{K}_l$ be the Gram matrix associated to region l, i.e., $\mathbf{K}_l(i,j) = \mathbf{x}_{i,l}^T \mathbf{x}_{j,l}$. This matrix is normalized using the following transformation [38]:

$$\mathbf{K}_l \mapsto \frac{\mathbf{K}_l}{\frac{1}{N}\sum_{i=1}^{N}\mathbf{K}_l(i,i) - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{K}_l(i,j)} \tag{5.3}$$

Region selection based on a sparse MKL formulation

MKL problem As it has been discussed on section 3.1 and illustrated in Fig. 3.1, MKL represents the data as a linear combination of kernels, the parameters of this combination being learned by solving an optimization problem. The decision function of this problem is defined in the primal by

$$f(\mathbf{x}_*) = \sum_{l=1}^{L} \mathbf{w}_l^T \mathbf{x}_{*,l} + b, \tag{5.4}$$

where $\mathbf{x}_*$ is a given test pattern and $\mathbf{w}_l$ are the parameters to be optimized.

Non-sparse MKL formulation Several MKL approaches explicitly incorporate the coefficients of the linear combination of kernels in their primal formulations. In general, they include coefficients η_l such that $\mathbf{K} = \sum_l \eta_l \mathbf{K}_l$ and add an l_1-norm regularization constraint on $\boldsymbol{\eta}$. The work presented in [38], which has been outlined on section 2.3.3, proposes a non-sparse combination of kernels by using an l_p-norm constraint with $p > 1$. For the specific case of the classification task introduced on

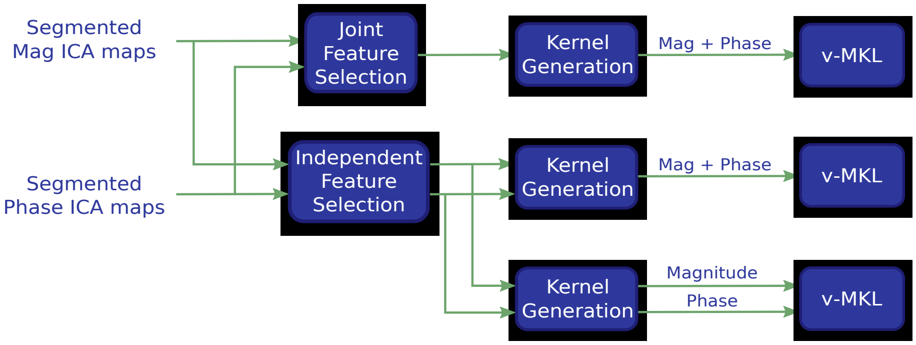


Figure 5.4: Strategies for complex-valued fMRI data feature selection and data sources combination. (Top row) First approach: Generation of a single kernel per brain region after the application of feature selection to the concatenation of the magnitude and phase brain region's feature sets. (Middle row) Second approach: Feature selection is applied separately to the magnitude and phase brain region's feature sets, after which they are concatenated and a single kernel per brain region is generated. (Bottom row) Third approach: Generation of one kernel per brain region for each data source after the independent application of feature selection to the magnitude and phase brain region's feature sets.

Region characterization this is their primal formulation:

$$\begin{aligned}
\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\eta}} \quad & \frac{1}{2}\sum_{l=1}^{L}\frac{\|\mathbf{w}_l\|_2^2}{\eta_l} + C\sum_{i=1}^{N}\xi_i \\
s.t. \quad & y_i\left(\sum_{l=1}^{L}\mathbf{w}_l^T\mathbf{x}_{i,l} + b\right) \geq 1 - \xi_i \qquad \forall i \\
& \xi_i \geq 0 \qquad \forall i \\
& \eta_l \geq 0 \qquad \forall l \\
& \|\boldsymbol{\eta}\|_p^2 \leq 1,
\end{aligned} \tag{5.5}$$

and its dual formulation is given by

$$\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\left\|\left(\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{K}_l(i,j)\right)_{l=1}^{L}\right\|_{p^*} - \sum_{i=1}^{N}\alpha_i \\
s.t. \quad & 0 \leq \alpha_i \leq C \qquad \forall i \\
& \sum_{i=1}^{N}\alpha_i y_i = 0,
\end{aligned} \tag{5.6}$$

where $p^* = \frac{p}{p-1}$ and the notation $(s_l)_{l=1}^{L}$ is used as an alternative representation of $s = [s_1, \ldots, s_L]^T$ for $s \in \mathbb{R}^L$.

An MKL formulation with block-sparsity constraints The proposed MKL algorithm generates a block-sparse selection of features based on the idea of introducing primal variable sparsity constraints in the SVM formulation presented in [46]. Please refer to section 3.3 for a detailed explanation of this algorithm.

Since the algorithm has been described so far using a dual formulation that only uses dot products between data points, a nonlinear version of this algorithm can be directly constructed as follows. By applying a nonlinear transformation function $\varphi_l(\cdot)$ to the data points $\mathbf{x}_{i,l}$ on region l, they can be mapped into a higher (possibly infinite) dimensional reproducing kernel Hilbert space [14] provided with an inner product of the form $\mathbf{K}_l(i,j) = \varphi_l^T(\mathbf{x}_{i,l})\varphi_l(\mathbf{x}_{j,l})$. By virtue of the reproducing property, the dot product is a (scalar) expression depending only on the input data $\mathbf{x}_{i,l}, \mathbf{x}_{j,l}$, and it fits the Mercer's theorem (see section 2.2.3). Such a function is called Mercer's kernel. Thus, the formulation remains exactly the same, the only difference being the substitution of the scalar dot product by a Mercer's kernel. One of the most popular Mercer's kernels is the Gaussian kernel, with the expression $\mathbf{K}_l(i,j) = \exp(-\frac{\|\mathbf{x}_{i,l}-\mathbf{x}_{j,l}\|^2}{2\sigma^2})$.

Note that the use of Mercer's kernels in the ν-MKL formulation exploits the nonlinear properties inside each region, while keeping linear combinations between them. ν-MKL is tested with both linear and Gaussian kernels for the complex-valued fMRI dataset, whereas linear kernels are used for the simulated dataset.

Parameter validation, feature selection and prediction accuracy estimation Accuracy rate calculation, feature selection and parameter validation were performed by means of a nested K-fold CV, the latter two procedures being performed sequentially in the external CV. For the complex-valued dataset, K was set to 52 (leave-one-subject-out CV), while for the simulated dataset $K = 10$.

The external CV is used to estimate the accuracy rate of the classifier and the γ values associated to the informative regions as follows. At each round of the external CV, a subset of the data composed of a single fold is reserved as a test set (*TestAll*), the remaining data being used to train and validate the algorithm (labeled *TrainValAll* in Algorithm 8). Next, the most discriminative magnitude component of the three task-related ones is selected based on the error rate attained by each of them on an internal CV using a linear SVM, as shown in Algorithm 10. The component that achieves the minimum validation error is the one used to represent the magnitude source. It should be noted that lines 7 through 9 of Algorithm 8 are applied exclusively when magnitude-only or magnitude and phase data are analyzed. After doing so, feature selection is applied to the data using RFE-SVM. While this procedure is applied to the complex-valued dataset only as stated on *Region representation*, we have incorporated it in Algorithm 8 as this is the only step that differs between both datasets in the nested K-fold CV.

It can be seen that RFE-SVM is applied at each round of the external CV to *TrainValSel*, i.e., the test set is never incorporated in this procedure, as it is a supervised algorithm. RFE-SVM then performs an internal CV to validate the selection of informative features. Within this validation procedure, a linear SVM is initially trained with all of the features of a given region. At each iteration of RFE-SVM, 20% of the lowest ranked features are removed, the last iteration being the one where the analyzed voxel set is reduced to 10% of its initial size.

After applying feature selection to the data, which yields the reduced sets *TrainValRed* and *TestRed*, *TrainValRed* is further divided into training and validation sets (see Algorithm 9), the latter one being composed of data from a single fold of *TrainValRed*. The classifier is then trained with a pool of parameter values for C, $C^{'}$ and ν, the validation error being estimated for each parameter combination as shown in Algorithm 9. The above process was repeated for all folds in *TrainValRed*, being the optimal tuple the one that achieved the minimum mean validation error. Then, the optimal tuple $(C, C^{'}, \nu)$ was used to retrain ν-MKL (see Algorithm 8) and retrieve the γ values associated to each region for the current CV round.

Next, the test error rate is estimated in the reserved test set. After doing so, another fold is selected as the new test set and the entire procedure is repeated for each of them. The test accuracy rate is then estimated by averaging the accuracy rates achieved by each test set and the γ values associated to each region across CV rounds are retrieved.

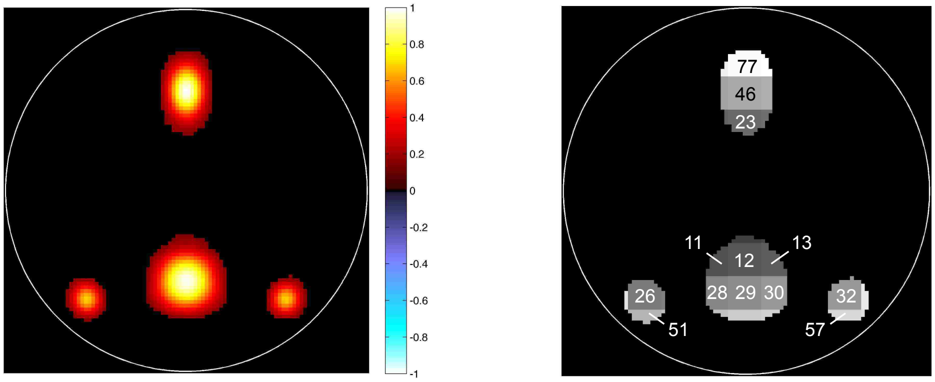
The criteria used to define the pool of values used for ν-MKL parameter selection was the following. The error penalty parameter C was selected from the set of values $\{0.01, 0.1, 1, 10, 100\}$, while the the sparsity tradeoff parameter C' was selected from a set of 4 values in the range $[0.1C, 10C]$, thus being at least one order of magnitude smaller than C but at most one order of magnitude higher. On the other hand, the set of values of the sparsity parameter ν were defined differently according to the analyzed dataset.

Since we had no prior knowledge of the degree of sparsity of the complex-valued dataset, ν was selected from the set of values $\{0.3, 0.5, 0.7, 0.9\}$. We also evaluated nonlinear relationships in each region by using Gaussian kernels, which additionally required the validation of σ. For each iteration of Algorithm 8, the median of the distances between examples of *TrainValSet* (σ_{med}) was estimated. This value was then multiplied by different scaling factors to select the optimal value of σ on Algorithm 9, the scaling factor being validated from a set of three logarithmically spaced values between 1 and 10.

To get a better idea of the sparsity of the simulated data classification task, the mean of the spatial maps across subjects was generated and thresholded, as shown in Fig. 5.5(a). As stated on *Simulated dataset*, differential activation should be generated in the voxels within the Gaussian blobs of the default mode component, thus generating a sparse problem. However, the actual sparsity of this problem cannot be fully characterized mainly due to the high variance (compared to the mean) of the within-group vertical translation and the spread introduced on this component, which changes the location and the extent of these blobs. Nonetheless, by analyzing the regions that overlap with the map in Fig.5.5(a), we can get a coarse estimate of its sparsity. It can be seen from Fig. 5.5(b) that the sparsity is higher than 10%. Based on this observation, we selected ν from the set of values $\{0.2, 0.4, 0.6, 0.8, 1\}$.

Estimation of informative regions The value of γ associated to a given region indicates its degree of differential activity between groups. However, γ does not take values on a fixed numeric scale. Specifically, γ values of informative regions across rounds of CV could be scaled differently, preventing us from directly comparing them. To correct for this, γ values at each CV round were normalized by the maximum value attained at that round. By doing so, the most relevant region for a given CV round would achieve a normalized score of 1 and the mean of the normalized γ values across CV rounds could be estimated.

The degree of differential activity of a region can also be assessed by estimating the number of times this region is deemed relevant across CV rounds (selection frequency). One way of taking into account both the selection frequency and the mean of the normalized γ to estimate the degree of information carried by a region is to generate a ranking coefficient that is the product of both estimates. These three estimates are used to evaluate the relevance of the analyzed regions for both the



(a)

(b)

Figure 5.5: Mean spatial map of the default mode component and indexes of overlapping square regions. This figure shows (a) the default mode component's thresholded mean spatial map across subjects and (b) the square regions that overlap with this mean map and the indexes of the overlapping regions.

complex-valued and the simulated datasets.

For the specific case of the simulated dataset, the incorporation of a small vertical translation between groups allows us to identify the location of certain regions that are differentially activated. However, numeric a priori estimates of the degree of differential activation of all the regions were needed to test how well ν-MKL detected the most informative ones. These estimates were generated by calculating their classification accuracy by means of a 10-fold CV using a linear SVM.

As it has been previously mentioned, brain data was segmented in cubical regions for the complex-valued dataset in order to be capable of performing a multivariate analysis that included all of the regions in the brain. However, it is difficult to interpret our results based on the relevance of cubical regions. One way of solving this problem was to map cubical regions and their associated γ values to anatomical regions defined by the automated anatomical labeling (AAL) brain parcellation [80] using the Wake Forest University pick atlas (WFU-PickAtlas)[3] [75, 76, 77, 98].

The mapping criterion is explained as follows. A cubical region was assumed to have an effective contribution to an anatomical one if the number of overlapping voxels between them was greater than or equal to 10% of the number of voxels of that cubical region. If this condition was satisfied, then the cube was mapped to this anatomical region. After generating the correspondence between cubical and anatomical regions, a weighted average of the γ values of the cubes associated to an anatomical region was computed and assigned to this region for each CV round.

Proposed data processing with l_p-norm MKL and SVM As it has been previously discussed, one of the goals of this work is to compare the performance of ν-MKL with other classifiers and MKL algorithms, such as SVMs and l_p-norm MKL. To do so, the same data processing applied in the proposed approach was

[3] Available at `http://www.fmri.wfubmc.edu/cms/software`

used for these two cases, thus simply replacing ν-MKL by either an SVM or l_p-norm MKL. The only difference in the processing pipeline for SVM was that the generated kernels were concatenated prior to being input to the classifier. As it will be seen in the results section, ν-MKL with Gaussian kernels does not provide better results than those obtained using linear kernels. These results were predictable based on the limited number of available subjects on our dataset. For this reason, we considered it appropriate to evaluate l_p-norm MKL and SVM using linear kernels only.

The SVM was trained using the LIBSVM software package[4] [81], and the error penalty parameter C was selected from a pool of 10 logarithmically spaced points between 1 and 100. Additionally, the l_p-norm MKL implementation code was retrieved from the supplementary material of [38], which is available at `http://doc.ml.tu-berlin.de/nonsparse_mkl/`, and was run under the SHOGUN machine learning toolbox[5] [99]. For both the simulated and complex-valued dataset we considered norms $p \in \{1, 4/3, 2, 4, \infty\}$ and $C \in [1, 100]$ (5 values, logarithmically spaced).

For the simulated dataset, the mean of the kernel weights of l_p-norm MKL across CV rounds for each region were also retrieved to evaluate how well this algorithm detected the amount of information provided by them, as well as to compare it against ν-MKL based on this criterion.

Data analysis with global approaches We also wanted to evaluate the performance of our local-oriented MKL methodology on the complex-valued dataset by comparing it against global approaches, which analyze activation patterns on the brain as a whole. Linear kernels were applied to the data for these approaches.

One straightforward global approach is the direct application of an SVM to the

[4] Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

[5] Available at `http://www.shogun-toolbox.org`

data without the application of per-region feature selection. Its performance was used as a benchmark for other approaches and was applied to either magnitude data, phase data or the concatenation of both. We refer to the concatenation of of whole-brain data from both sources as whole data. Another used approach was the application of global (whole-brain) RFE-SVM to the data. This algorithm was implemented such that 10% of the lowest ranked voxels were removed at each iteration of RFE-SVM.

In addition, global RFE-SVM was used to combine magnitude and phase data using two strategies. The first one concatenated data from magnitude and phase sources prior to the application of global RFE-SVM. On the other hand, the second one applied global RFE-SVM to each source independently for feature selection purposes, after which an SVM was trained with the output of feature selection. The concatenation of the data from both sources after the application of this feature selection procedure is referred to as filtered data.

Statistical assessment of the contribution of phase data If an improvement in the classification accuracy rate were obtained by combining both magnitude and phase data, further analysis would be required to confirm that this increment was indeed statistically significant. The statistic to be analyzed would be the accuracy rate obtained by using both data sources.

Since the underlying probability distribution of this statistic is unknown, a non-parametric statistical test such as a permutation test [100] would enable us to test the validity of the null hypothesis. In this case, the null hypothesis would state that the accuracy rate obtained by using magnitude and phase data should be the same as the one attained by working with these two data sources regardless of the permutation (over the subjects) of the phase signal.

Let $\mathcal{D}^m$ and $\mathcal{D}^f$ be the labeled magnitude and phase data samples, respectively, and let $\mathrm{CR}(\mathcal{D}^m, \mathcal{D}^f)$ be the classification accuracy rate obtained with these two data sources using one of the combination approaches described on *Region characterization* and the prediction accuracy estimation presented on *Parameter validation, feature selection and prediction accuracy estimation.* The permutation test generates all possible permutation sets of the phase data sample $\mathcal{D}^f_{perm}(k)$, $1 \leq k \leq N!$, doing no permutation of the magnitude data sample $\mathcal{D}^m$. Next, it computes the accuracy rates $\mathrm{CR}(\mathcal{D}^m, \mathcal{D}^f_{perm}(k))$. The p-value associated to $\mathrm{CR}(\mathcal{D}^m, \mathcal{D}^f)$ under the null hypothesis is defined as

$$p = \frac{\sum_{k=1}^{N!} \mathcal{I}(\mathrm{CR}(\mathcal{D}^m, \mathcal{D}^f_{perm}(k)) > \mathrm{CR}(\mathcal{D}^m, \mathcal{D}^f))}{N!}, \tag{5.7}$$

where $\mathcal{I}(\cdot)$ is the indicator function.

Due to the high computational burden of computing all possible permutations in the elements of $\mathcal{D}^f_{perm}(k)$, in practice only tens or hundreds of them are used in a random fashion. The observed p-value is defined as

$$\hat{p} = \frac{\sum_{k=1}^{M} \mathcal{I}(\mathrm{CR}(\mathcal{D}^m, \mathcal{D}^f_{perm}(k)) > \mathrm{CR}(\mathcal{D}^m, \mathcal{D}^f))}{M}, \tag{5.8}$$

where M is the number of used permutations. In this case, the exact p-value cannot be known but a 95% confidence interval (CI) around $\hat{p}$ can be estimated [101]

$$\mathrm{CI}_{95\%}(p) = \hat{p} \pm 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{M}}. \tag{5.9}$$

5.3.3 Results

Simulated dataset

The prior estimates of the degree of differential activation present on a subset of regions are shown on the first column of Table 5.9, these regions being sorted from most to least discriminative. It can be seen that 11 out of the 15 reported regions are consistent with the assumption that most of the differential activity would be focused on those squares overlapping with the default mode network activation blobs, as shown in Fig. 5.5.

This table also shows the selection frequency and the relevance estimates of these regions using ν-MKL (normalized γ) and l_p-norm MKL (kernel weights). A classification accuracy rate of 0.90 and 0.85 is attained by ν-MKL and l_p-norm MKL, respectively. In addition, the fraction of selected regions was 0.14 for ν-MKL and 0.50 for l_p-norm MKL.

Table 5.9: Estimation of the information of a subset of regions using linear kernels along with ν-MKL and l_p-norm MKL for the simulated dataset. The metrics used to determine the amount of information of the regions by means of ν-MKL (mean of the normalized γ values) and l_p-norm MKL (kernel weights' mean) as well as their selection frequencies for each algorithm are reported. Both the normalized γ values and the kernel weights have been scaled so that their maximum values equal 1 to make the comparison easier. These coefficients are contrasted against the accuracy rates achieved by these regions using a linear SVM.

Region	Linear SVM	ν-MKL		l_p-norm MKL	
	Acc. Rate	Sel. Freq.	Normalized γ	Sel. Freq.	Kernel Weights
Square 26	0.81	1	1.00	1	0.91
Square 46	0.78	1	0.95	1	0.91
Square 32	0.77	1	0.99	1	1.00
Square 77	0.76	1	0.91	1	0.72
Square 29	0.76	1	0.76	1	0.67
Square 23	0.76	1	0.71	1	0.81
Square 12	0.75	1	0.75	1	0.53
Square 57	0.69	1	0.54	0.50	0.58
Square 51	0.68	1	0.52	1	0.34
Square 30	0.67	1	0.24	0.50	0.34
Square 107	0.63	0.60	0.08	0.60	0.30
Square 13	0.60	0.60	0.09	0.50	0.38
Square 44	0.57	0.30	0.13	0.90	0.29
Square 37	0.56	0.10	0.09	0.90	0.24
Square 20	0.54	0.10	0.07	0.80	0.22

Complex-valued dataset

We present the results of both local-oriented and global approaches on Table 5.10. Accuracy rates of the proposed methodology using ν-MKL, l_p-norm MKL and SVM for single-source analysis and different source combination approaches are listed along with the results obtained by the global approaches introduced on *Data analysis with global approaches*.

It can be seen that by applying linear ν-MKL to magnitude and phase data using the third combination approach, an increment of 5% with respect to the magnitude-only data analysis is obtained. In this case, $\mathrm{CR}(\mathcal{D}^m, \mathcal{D}^f) = 0.85$. After generating 100 permutations we get $\hat{p} = 0.01$ and a 95% CI $[0, 0.03]$ according to (5.8) and (5.9), respectively. Since $p < \alpha = 0.05$, we can reject the null hypothesis at a significance

Table 5.10: Performance of the proposed methodology and global approaches on the complex-valued fMRI dataset. This table presents the classification accuracy (first row) and the sensitivity/specificity rates (second row) of our local-oriented methodology using ν-MKL l_p-norm MKL and SVM for single-source data (magnitude or phase) and different source combination approaches. It also shows the results obtained by global approaches. Notice that SVM is applied to both the proposed approach and global approaches. The reported values are attained by these algorithms using linear kernels, except where noted.

Classifier	Single Sources				Combined Sources				
	Prop. Approach		Global Approach		Proposed Approach			Global Approaches	
	Magn	Phase	Magn	Phase	Comb 1	Comb 2	Comb 3	Whole	Filt.
SVM	0.77	0.64	0.62	0.58	0.80	0.79	0.79	0.63	0.80
	0.84/0.67	0.65/0.64	0.71/0.48	0.55/0.62	0.85/0.71	0.82/0.74	0.82/0.74	0.71/0.50	0.82/0.76
Global	–	–	0.76	0.61	–	–	–	0.80	–
RFE-SVM	–	–	0.81/0.69	0.63/0.57	–	–	–	0.92/0.62	–
ν-MKL	0.80	0.70	–	–	0.76	0.76	**0.85**	–	–
(linear)	0.85/0.71	0.69/0.71	–	–	0.82/0.67	0.84/0.64	0.90/0.76	–	–
ν-MKL	0.78	0.68	–	–	0.68	0.77	**0.85**	–	–
(Gaussian)	0.84/0.69	0.71/0.64	–	–	0.77/0.55	0.87/0.62	0.92/0.74	–	–
l_p-norm	0.78	0.64	–	–	0.76	0.72	0.84	–	–
MKL	0.84/0.69	0.66/0.62	–	–	0.82/0.67	0.73/0.71	0.90/0.74	–	–

level of 0.05. Consequently, the improvement in classification accuracy rate obtained by including phase data is statistically significant with 95% confidence level.

Table 5.11 shows the cubical regions' selection sparsity achieved by ν-MKL and l_p-norm MKL. It can be seen that a higher selection sparsity is attained by classifying the data with ν-MKL for single-source analysis and the third source combination approach.

The most informative regions and their associated relevance estimates detected by ν-MKL using linear kernels are reported as follows. The ranking coefficients of a subset of the top 40% ranked regions for magnitude-only and magnitude and phase data analyses (combination approach 3) are color-coded and displayed on top of a structural brain map in Fig. 5.6. This figure provides a graphical representation of the spatial distribution of these regions. In addition, Table 5.12 provides the differential activity estimates of some of these regions, such as selection frequency and normalized γ. This table also reports ranking indexes, which enables the analysis of changes on the relative contribution of these regions across single-source and combined-source analyses.

Table 5.11: Selection sparsity achieved by ν-MKL and l_p-norm MKL on the complex-valued dataset. This table shows the fraction of valid selected regions (according to the criterion discussed in section 5.3.2) for both ν-MKL and l_p-norm MKL for single-source analysis (magnitude or phase) and the third combination approach of both sources. The presented values are achieved by both algorithms using linear kernels, except where noted.

Source	Fraction of valid selected regions			# of valid regions
	ν-MKL		l_p-norm MKL	
	Linear	Gaussian		
Magnitude	0.69	0.71	0.90	135 (of 158)
Phase	0.70	0.69	0.85	108 (of 158)
Mag + Phase	0.74	0.75	0.95	243 (of 316)

5.3.4 Discussion

This work presents an MKL-based methodology that combines magnitude and phase data to better differentiate groups of healthy controls and schizophrenia patients from an AOD task. In contrast, previous approaches devised methods that incorporated magnitude and phase data, but did not perform between-group inferences. In addition, the presented methodology is capable of detecting the most informative regions for schizophrenia detection.

Table 5.10 shows the results obtained by our MKL-based methodology using ν-MKL for single-source analysis, as well as the combination of magnitude and phase.

Table 5.12: Reduced set of the top 40% ranked regions for magnitude-only and magnitude and phase analyses and their differential activity estimates. This table lists a set of informative regions and their associated relevance estimates, such as selection frequency and normalized γ values. In addition, ranking indexes are reported to analyze changes on the relative contribution of these areas across single-source and combined-source analyses.

Region	Single Source			Combined Sources					
	Magnitude			Magnitude			Phase		
	Rank	Sel. Freq.	γ	Rank	Sel. Freq.	γ	Rank	Sel. Freq.	γ
Right Caudate Nucleus	1	1.00	0.82	1	1.00	0.80	–	–	–
Right Precuneus	2	1.00	0.51	2	1.00	0.57	–	–	–
Right Superior Occipital Gyrus	3	1.00	0.49	3	1.00	0.53	–	–	–
Right Middle Cingulate Gyrus	4	0.98	0.49	15	1.00	0.43	–	–	–
Right Superior Parietal Lobe	5	1.00	0.48	8	1.00	0.48	–	–	–
Left Gyrus Rectus	6	0.96	0.49	12	0.98	0.44	–	–	–
Right Angular Gyrus	7	1.00	0.46	11	1.00	0.43	–	–	–
Left Precuneus	8	1.00	0.46	6	1.00	0.52	–	–	–
Left Middle Temporal Gyrus	9	1.00	0.45	7	1.00	0.50	–	–	–
Left Superior Temporal Gyrus	10	1.00	0.45	4	1.00	0.53	–	–	–
Left Angular Gyrus	11	1.00	0.44	20	1.00	0.40	–	–	–
Left Parahippocampal Gyrus	12	1.00	0.44	10	1.00	0.44	–	–	–
Left Paracentral Lobule	13	1.00	0.43	18	0.98	0.42	–	–	–
Right Gyrus Rectus	14	0.96	0.44	39	0.98	0.37	–	–	–
Right Cuneus	15	1.00	0.41	13	1.00	0.43	–	–	–
Right Anterior Cingulate Gyrus	23	0.96	0.39	35	0.98	0.38	–	–	–
Left Hippocampus	–	–	–	16	0.98	0.43	–	–	–
Right Superior Temporal Gyrus	–	–	–	23	1.00	0.39	88	0.96	0.23
Left Superior Frontal Gyrus	–	–	–	34	0.98	0.38	–	–	–
Left Anterior Cingulate Gyrus	–	–	–	36	0.98	0.38	–	–	–
Left Middle Frontal Gyrus	–	–	–	42	0.98	0.37	–	–	–
Right Posterior Cingulate Gyrus	–	–	–	50	0.98	0.34	–	–	–
Left Posterior Cingulate Gyrus	–	–	–	51	0.98	0.34	–	–	–
Right Middle Temporal Gyrus	–	–	–	62	0.98	0.31	72	0.98	0.29
Right Inferior Temporal Gyrus	–	–	–	–	–	–	56	0.98	0.33
Left Temporal Pole: Middle Temporal Gyrus	–	–	–	–	–	–	83	0.92	0.27
Left Lingual Gyrus	–	–	–	–	–	–	91	0.88	0.25
Right Temporal Pole: Superior Temporal Gyrus	–	–	–	–	–	–	92	0.94	0.23

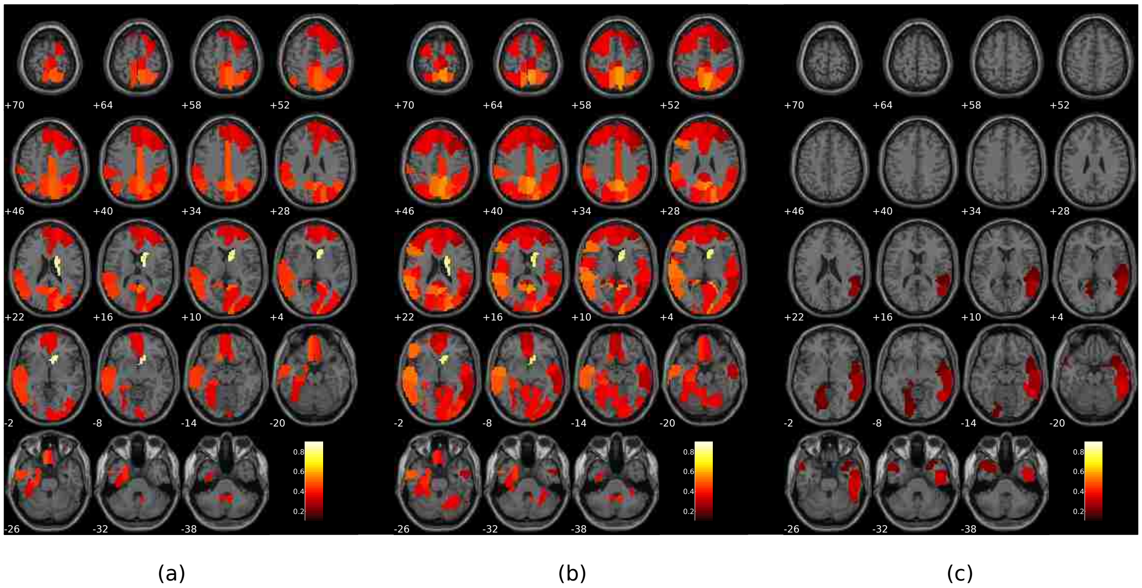


Figure 5.6: Ranking coefficients of a subset of the of the top 40% ranked regions for magnitude-only and magnitude and phase analyses. This figure shows (a) informative regions for the magnitude-only analysis, (b) informative regions of the magnitude source for the magnitude and phase analysis, and (c) informative regions of the phase source for the magnitude and phase analysis. Each of the displayed blobs are color-coded according to their associated ranking coefficients. As expected, magnitude is the most informative source, but several regions in phase, including the temporal lobe, are also informative.

It can be seen that, when linear kernels are used, the first and the second combination approaches obtain a smaller classification accuracy rate compared to the magnitude-only analysis. On the contrary, the third approach achieves an increment of 5% with respect to the magnitude data analysis. The probability of this value being obtained by chance is in the range $[0, 0.03]$, being statistically significant at the 95% confidence level. These results support the validity of the rationale behind the third combination approach, which assumed that magnitude and phase are dissimilar data, thus requiring a kernel mapping to be applied independently for each source.

The performance of ν-MKL was also evaluated using Gaussian kernels. These results are comparable to those obtained using linear kernels, except for combination 1. A detailed analysis of the parameter validation procedure revealed that the values of σ were usually 10 times σ_{med}. Such a large value of σ makes the Gaussian kernel similar to a linear one, which is consistent with the reported results. In addition, these results suggest that adding complexity to the classification model is not helpful on this dataset. This finding comes as no surprise since our dataset is composed of data from a small number of subjects. However, it is expected that nonlinear kernels would better characterize schizophrenia if a bigger dataset were analyzed, In fact, the work presented in [61] supports this postulate.

In addition to the results obtained by ν-MKL, Table 5.10 displays the results obtained by our local-oriented methodology using l_p-norm MKL and SVM. The results obtained by ν-MKL seem to be equivalent or slightly better than those obtained by l_p-norm MKL. The differences in classification accuracy for both algorithms do not seem to be statistically significant. However, we must keep in mind that this is not the only criterion used to compare the performance of both algorithms. These algorithms are also evaluated based on their capacity to detect the degree of differential activity of the analyzed regions and their capability to detect the sparsity of the classification task. In short, we analyze the capacity of both algorithms to achieve a better interpretation of the data. This is analyzed on more detail later on this section.

It can also be seen from Table 5.10 that both ν-MKL and l_p-norm MKL appear to show a similar trend. For example, both algorithms obtain a classification accuracy rate below the one achieved by the magnitude-only analysis for the first and the second combination approaches; instead, SVM achieves a better classification result than magnitude data analysis for all combination approaches. This can be explained by the fact that SVM does not analyze the regions' information locally since the data is concatenated prior to being input to the SVM.

The results obtained by using global approaches are shown on the same table. It can be seen that the two global RFE-SVM-based strategies used to combine magnitude and phase data also improve the classification accuracy rate obtained by processing magnitude data only. Furthermore, both of them reach the same rates (0.80). However, their rates are smaller than the one achieved by combination 3 of our local-oriented approach (0.85).

Another important objective of this work is to show that ν-MKL can better identify the feature sets that show discriminative activation between groups compared to other MKL algorithms, such as l_p-norm MKL; the simulated dataset is used for this purpose. It was previously mentioned that the results in Table 5.9 indicate that 11 of the 15 reported regions do overlap with the default mode network activation blobs (Fig. 5.5). It should be noted that 10 out of those 11 regions, which show a significant differential activation according to the accuracy rates reported by SVM, are selected on all CV rounds by ν-MKL. In contrast, 2 of these regions (57 and 30) are selected by l_p-norm MKL on only half of the CV rounds. On the other hand, the last three regions (44, 37 and 20), which show weak differential activation across groups, are selected by ν-MKL on a few CV rounds, whereas they achieve a high selection frequency with l_p-norm MKL. Furthermore, it can be seen that the γ coefficients assigned by ν-MKL to these regions are approximately one order of magnitude smaller than the top ranked region (26), which is not the case for l_p-norm MKL.

On the methods section, we mention the validation of parameter p for l_p-norm MKL experiments, this parameter being the norm of the kernel coefficients on one of the constraints imposed on (5.5). When $p \approx 1$, these coefficients yield a kernel combination that is close to a sparse one, being actually sparse when $p = 1$. On the contrary, these coefficients are uniformly assigned the value 1 when $p = \infty$. We analyzed the validated values of p for each CV round in order to get a better idea of the reason why l_p-norm MKL failed to give a better estimate of the contribution of the relevant areas on the simulated dataset. We found out that on 7 out of 10 rounds, $p = 1$ or $4/3$ (close to 1). It is clear that l_p-norm attempts to do a sparse selection of the informative regions, but with $p \approx 1$ this algorithm seems to pick just some kernels when they are highly correlated, a limitation that would be consistent with the findings on l_1-norm SVM [35]. Even though l_p-norm MKL looks for a sparse solution, it still estimates that the fraction of relevant regions is 0.50, deeming half of the regions of the analyzed spatial map informative. Based on the accuracy rate estimates obtained by a linear SVM and the graphical representation provided in Fig. 5.5, it is unlikely that the sparsity of the simulated data classification task is of that order. On the contrary, ν-MKL estimates that the fraction of relevant regions is 0.14, which seems more consistent with the prior knowledge of the spatial extent of the voxels having differential activation across groups.

Based on the analysis of the performance of both MKL algorithms on the simulated dataset, it can be inferred that the l_p-norm MKL formulation based on a non-sparse combination of kernels provides a less precise estimate of the sparsity of the classification task at hand than ν-MKL. In addition, ν-MKL provides a more accurate measurement of the degree of information conveyed by each kernel.

If we analyze the results obtained for the complex-valued fMRI dataset, it can be seen that ν-MKL region selection is sparser than the l_p-norm MKL one (Table 5.11), while still achieving at least equivalent classification results. A similar trend is found on the simulated dataset, with ν-MKL better detecting the sparsity of the classification task. Based on this finding, it can be argued that ν-MKL may achieve a better detection of the most informative brain regions on the complex-valued dataset. However, this cannot be verified as the ground truth for real fMRI data is unknown.

In terms of the selection of the most discriminative magnitude component, it should be highlighted that the default mode component was consistently selected at each iteration of Algorithm 8. This is an important finding that reinforces the notion that this spatial component reliably characterizes schizophrenia [53, 54].

Table 5.12 shows a reduced set of the most informative regions for magnitude-only and magnitude and phase analyses. Among the regions deemed informative by the former analysis temporal lobe regions can be found, which is consistent with findings on schizophrenia. To better understand which regions could be informative on our study, we need to be aware that the AOD task requires the subjects to make a quick button-press response upon the presentation of target stimuli. Such an action is highly sensitive to attentional selection and evaluation of performance, as the subject needs to avoid making mistakes. For this reason we highlight the presence of the anterior cingulate gyrus among the informative regions for the magnitude-only analysis, for it has been proposed that error-related activity in the anterior cingulate cortex is impaired in patients with schizophrenia [82]. The presence of the precuneus and the middle frontal gyrus is also important, as it has been suggested that both regions are involved in disturbances in selective attention, which represents a core characteristic of schizophrenia [83].

The regions that are deemed informative for magnitude only remain being the most informative when phase data is included in the analysis. However, their relative importance changes on several of them, as it can be seen by inspecting the rank values of these regions in these two scenarios. In addition, new brain areas show up in the set of informative regions, which is the case for some other temporal lobe regions and, for phase data, for regions of the temporal pole.

The presence of phase activation in regions expected to be differentially activated across groups in the AOD task, such as the temporal lobe regions, suggests that phase indeed provides reliable information to better characterize schizophrenia. In addition, it implies that the inclusion of phase can potentially increase sensitivity within regions also showing magnitude activation.

Similarly, the fact that regions of the temporal pole show up in the set of most informative regions is appealing, as evidence has been found that the temporal pole links auditory stimuli with emotional reactions [102]. In fact, some studies report the temporal pole as a relevant component of the paralimbic circuit, and associate it with socioemotional processing [103]. Since social cognition is a key determinant of functional disability of schizophrenia, it makes sense to hypothesize that the temporal pole is activated differently in schizophrenia patients when auditory stimuli is presented.

The aforementioned results reinforce the notion that magnitude and phase may be complementary data sources that can better characterize schizophrenia when combined.

Algorithm 8 Test ν-MKL

```
**Inputs**: $DataSet, \nu_{vals}, C'_{vals}, C_{vals}$
**Outputs**: $TestAcc, \boldsymbol{\gamma}$
**Define** $N$: number of folds in $DataSet$
**for** $i = 1$ to $N$ **do**
    Extract $TrainValAll(i)$ from $DataSet$
    Extract $TestAll(i)$ from $DataSet$
    *Select Magnitude Component($TrainValAll(i)$) $\Rightarrow CompInd$
    $^{*}TrainValAll(i)(CompInd) \Rightarrow TrainValSel(i)$
    $^{*}TestAll(i)(CompInd) \Rightarrow TestSel(i)$
    *RFE-SVM($TrainValSel(i)$) $\Rightarrow SelectFeat$
    $^{*}TrainValSel(i)(SelectFeat) \Rightarrow TrainValRed(i)$
    $^{*}TestSel(i)(SelectFeat) \Rightarrow TestRed(i)$
    **Validate parameters** $\nu -$ **MKL** $(TrainValRed(i), \nu_{vals}, C'_{vals}, C_{vals}) \Rightarrow$
    $C, C', \nu$
    Train with $TrainValRed(i)$, $C'$, $\nu$ and $C \Rightarrow Trained\ \nu - MKL$, $\boldsymbol{\gamma}(i)$
    Test with $TestRed(i)$ and $Trained\ \nu - MKL$
    Store accuracy rate $\Rightarrow acc(i)$
**end for**
Average $acc(i)$ over $i \Rightarrow TestAcc$
```

Algorithm 9 Validate parameters ν-MKL

```
**Inputs**: $TrainValRed$, $\nu_{vals}$, $C^{'}_{vals}$, $C_{vals}$
**Outputs**: $C$, $C^{'}$, $\nu$
**for** $i = 1$ to $N - 1$ **do**
    Extract $Train(i)$ from $TrainValRed$
    Extract $Val(i)$ from $TrainValRed$
    **for** $j = 1$ to $\#C^{'}_{vals}$ **do**
        $C^{'}_{sel} = C^{'}_{vals}(j)$
        **for** $k = 1$ to $\#\nu_{vals}$ **do**
            $\nu_{sel} = \nu_{vals}(k)$
            **for** $l = 1$ to $\#C_{vals}$ **do**
                $C_{sel} = C_{vals}(l)$
                Train with $Train(i)$, $C^{'}_{sel}$, $\nu_{sel}$ and $C_{sel} \Rightarrow Trained\ \nu - MKL$
                Test with $Val(i)$ and $Trained\ \nu - MKL$
                Store error $\Rightarrow e(i, j, k, l)$
            **end for**
        **end for**
    **end for**
**end for**
Average $e(i, j, k, l)$ over $i \Rightarrow e(j, k, l)$
Find $(j, k, l)$ that minimizes $e(j, k, l) \Rightarrow (J, K, L)$
$C^{'}_{vals}(J) \Rightarrow C^{'}$
$\nu_{vals}(K) \Rightarrow \nu$
$C_{vals}(L) \Rightarrow C$
```

Algorithm 10 Select Magnitude Component

```
Inputs: $TrainValAll$
Outputs: $CompInd$
for $i = 1$ to $N - 1$ do
    Extract $Train(i)$ from $TrainValAll$
    Extract $Val(i)$ from $TrainValAll$
    for $j = 1$ to $3$ do
        Train with $Train(i)(j) \Rightarrow TrainedSVM$
        Test with $Val(i)(j)$ and $TrainedSVM$
        Store error $\Rightarrow e(i, j)$
    end for
end for
Average $e(i, j)$ over $i \Rightarrow e(j)$
Find $j$ that minimizes $e(j) \Rightarrow CompInd$
```

Chapter 6

Concluding Remarks, Future Work, and Recommendations

6.1 Concluding remarks

The multiple-kernel based framework presented in this dissertation proves to be useful in the characterization of schizophrenia, as it provides an intuitive interpretation of the functional regions that present different degrees of abnormal brain activation on schizophrenia, while also achieving a reasonable classification of healthy controls and schizophrenia patients. In fact, this was the first work to propose the use of multiple kernels to represent feature sets from different brain regions and analyze their contribution to the characterize a mental illness.

As stated before, the proposed methodology identifies regions that show abnormal brain activation patterns on patients. Consistent findings across the different data analysis approaches presented on Chapter 5 highlight the importance of several regions among the brain, including the temporal lobe and the anterior cingulate cortex. Since schizophrenia is typified by perturbations in perception, it makes sense to find abnormal activity on the temporal lobe, especially on experimental paradigms that stimulate the auditory cortex. Relatedly, the anterior cingulate cortex has been reported to be involved in rationale cognitive function, which is impaired in schizophrenia [82]. The concordance of results presented in this dissertation with previous findings validates its significance.

In addition, the proposed approach is capable of better characterizing schizophrenia when provided with multiple data sources, such as information retrieved from different fMRI data analysis methods. Most importantly, the results obtained by the ν-MKL based approach provide evidence that phase along with magnitude data can indeed provide a better specificity for the location of abnormal activation in schizophrenia. Likewise, it also makes it possible to detect informative brain regions that cannot be identified by using magnitude data only. To the best of my knowledge, this is the first study to do schizophrenia classification using complex-valued fMRI data.

Furthermore, the algorithm's flexibility to analyze nonlinear relationships between voxels within brain regions may improve the characterization of schizophrenia under certain conditions, as there is a risk to overfit the data if not enough observations are available. The RCK analysis presented on section 5.1 suggests that better results can be obtained by using nonlinear approximations. On the other hand, the results obtained by ν-MKL on section 5.3 show no difference between linear or nonlinear analyses. This is probably related to the number of subjects available on the second study, which is approximately half of the first one.

It is also important to remark that ν-MKL achieves a better characterization of schizophrenia than a state of the art MKL algorithm (l_p-norm MKL), while still finding a sparse set of informative regions. This not only implies that ν-MKL is well-suited for classification tasks using fMRI data, but also suggests that this approach could be applied to classification tasks in other domains, thus providing an alternative rationale for MKL formulations that look for sparse solutions.

6.2 Future work and recommendations

Based on the capacity of the proposed framework to deal with different data sources, it is reasonable to think that this approach would be useful to combine data from multiple data modalities. One potential application could be the combination of imaging and genetics data to better characterize mental disorders.

Another development that could be incorporated in this methodology is to extend it to do between-group inferences on multi-class or even non-categorical (continuous) variables of interest by expanding ν-MKL to work with other loss functions. This would generalize the proposed binary classification approach to perform multi-class classification or even regression.

In addition, ν-MKL has been formulated as a proof of concept approach. In other words, the main criterion used for its formulation was to verify its functionality. Based on the obtained results on both simulated and real data, ν-MKL achieves a reasonable performance. The next step would be to reformulate the algorithm so that it achieves better scalability with respect to sample size and number of kernels.

In order to foresee the future of machine learning for mental illness discovery, a better understanding of these disorders is imperative. There is an inherent problem in their characterization, since there is no neuroscientific evidence to support the discrete categorization defined by the Diagnostic and Statistical Manual of Mental Disorders [104]. In fact, studies suggest that mental illnesses overlap and may lay on a continuum. As it has been stated initially, machine learning is a field of study that learns from data. The potential of machine learning to provide a better characterization of mental disorders is very vast and this field should point towards this research direction.

Appendix A

Symmetric Positive Semidefinite Matrix Decomposition

Let $\mathbf{H}$ be an $n \times n$ real symmetric matrix, with rank $r < n$. This matrix can be factored into $\mathbf{H} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, with orthonormal eigenvectors in $\mathbf{Q}$ and real eigenvalues in $\mathbf{\Lambda}$ [105]. If this matrix is also positive semidefinite, then its eigenvalues are greater than or equal to zero. While eigenvalue estimates are sensitive to perturbations for some ill-conditioned matrices, the singular value problem is always well-conditioned [106]. That is the reason why this section derives a decomposition of the form $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ of $\mathbf{H}$ based on its singular value decomposition (SVD).

The SVD of $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are $n \times n$ orthogonal matrices and $\mathbf{\Sigma}$ is an $n \times n$ diagonal matrix whose diagonal entries are the singular values of $\mathbf{H}$. Let $\sigma_1, \sigma_2, \ldots, \sigma_n$ be the elements on the diagonal of $\mathbf{\Sigma}$ and assume they are ordered in descending order. If u_i and v_i, where $i \in \{1, 2, \ldots, n\}$, are the columns of matrices $\mathbf{U}$ and $\mathbf{V}$ respectively, then

$$\mathbf{H} = \sum_{i=1}^{n} u_i \sigma_i v_i^T. \tag{A.1}$$

Since $\mathbf{H}$ has rank r, it has r nonzero singular values, which are also eigenvalues of $\mathbf{H}$. In addition, singular vectors u_i and v_i such that $i \in \{1, 2, \dots, r\}$ are equal and are in fact eigenvectors of $\mathbf{H}$. Thus,

$$\mathbf{H} = \sum_{i=1}^{r} u_i \sigma_i v_i^T = \sum_{i=1}^{r} u_i \sigma_i u_i^T = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{U}_r^T, \tag{A.2}$$

where $\mathbf{\Sigma}_r$ is an $r \times r$ matrix whose diagonal entries are $\sigma_1, \sigma_2, \dots, \sigma_r$ and $\mathbf{U}_r$ is an $n \times r$ matrix whose columns are r eigenvectors of $\mathbf{H}$. Thus, $\mathbf{H}$ can be decomposed as

$$\mathbf{H} = (\mathbf{U}_r \mathbf{\Sigma}_r^{1/2})(\mathbf{\Sigma}_r^{1/2} \mathbf{U}_r^T) = \mathbf{F}^T \mathbf{F}, \tag{A.3}$$

where $\mathbf{F} = \mathbf{\Sigma}_r^{1/2} \mathbf{U}_r^T$.

$\mathbf{F}$ can be either directly determined by Eq. A.3 as an $r \times n$ matrix or it can be zero-padded in order to make it $n \times n$. If we drop the assumption that $\mathbf{H}$ is rank deficient, the presented procedure would still hold, yielding an $n \times n$ matrix $\mathbf{F}$ directly.

References

[1] National Institute of Mental Health, "What is schizophrenia?" n.d., retrieved from http://www.nimh.nih.gov/health/topics/schizophrenia/index.shtml.

[2] S. A. Huettel, A. W. Song, and G. McCarthy, *Functional Magnetic Resonance Imaging.* Sunderland, MA: Sinauer Associates, 2004.

[3] V. D. Calhoun, "Introduction to fMRI," Analysis Methods in Functional Magnetic Resonance Imaging Course, Uiversity of New Mexico, Spring Semester, 2007, available at http://ece.unm.edu/~vcalhoun/.

[4] F. Hoogenraad, P. Pouwels, M. Hofman, J. Reichenbach, M. Sprenger, and E. Haacke, "Quantitative differentiation between BOLD models in fMRI," *Magnetic Resonance in Medicine*, vol. 45, no. 2, pp. 233–246, Feb 2001.

[5] V. D. Calhoun and T. Adali, "Unmixing fmri with independent component analysis," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 25, no. 2, pp. 79–90, April 2006.

[6] F. Pereira, T. Mitchell, and M. Botvinick, "Machine learning classifiers and fmri: A tutorial overview," *NeuroImage*, vol. 45, no. 1, Supplement 1, pp. S199 – S209, 2009.

[7] K. Murphy, *Machine Learning: a Probabilistic Perspective.* Cambridge, MA: MIT Press, 2012, Incomplete Draft. Introduction chapter available at http://www.cs.ubc.ca/~murphyk/MLbook/pml-intro-22may12.pdf.

[8] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge, UK: Cambridge University Press, 2000.

[9] A. Smola and S. Vishwanathan, *Introduction To Machine Learning.* Cambridge, UK: Cambridge University Press, 2008, Incomplete Draft. Available at http://alex.smola.org/drafts/thebook.pdf.

[10] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory.* London, UK, UK: Springer-Verlag, 2001, pp. 416–426.

[11] B. Schölkopf and A. J. Smola, "A short introduction to learning with kernels," in *Advanced Lectures on Machine Learning.* Springer, 2002, pp. 41–64.

[12] C. Heil, "Banach and Hilbert Space Review," Available at http://people.math.gatech.edu/~heil/6338/summer08/, Sep 2006, Supplementary material for Real Analysis II, Georgia Institute of Technology.

[13] A. Zien, "Multiple kernel learning," Summer School on Neural Networks, Porto, Portugal, July 2008, Oral presentation.

[14] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, May 1950.

[15] F. Riesz and B. Szökefalvi-Nagy, *Functional Analysis.* New York: Courier Dover Publications, 1990.

[16] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems.* Washington, D.C.: Winston & Sons, 1977.

[17] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun 1998.

[18] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 273–297.

[19] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Advances in Kernel Methods - Support Vector Learning, Tech. Rep., 1998.

[20] O. Chapelle, "Training a support vector machine in the primal," *Neural Computation*, vol. 19, pp. 1155–1178, 2007.

[21] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[22] J. Mourão-Miranda, E. Reynaud, F. McGlone, G. Calvert, and M. Brammer, "The impact of temporal compression and space selection on svm analysis of single-subject and multi-subject fmri data," *NeuroImage*, vol. 33, no. 4, pp. 1055 – 1065, 2006.

[23] J. D. Haynes and G. Rees, "Predicting the orientation of invisible stimuli from activity in human primary visual cortex," *Nature Neuroscience*, vol. 8, no. 5, pp. 686–691, May 2005.

[24] F. De Martino, G. Valente, N. Staeren, J. Ashburner, R. Goebel, and E. Formisano, "Combining multivariate voxel selection and support vector machines for mapping and classification of fmri spatial patterns," *NeuroImage*, vol. 43, no. 1, pp. 44 – 58, 2008.

[25] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1-3, 2002.

[26] S. Ryali, K. Supekar, D. A. Abrams, and V. Menon, "Sparse logistic regression for whole-brain classification of fmri data," *Neuroimage*, 2010.

[27] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 93–97, Jan 2006.

[28] G. R. G. Lanckriet, T. D. Bie, N. Cristianini, M. I. Jordan, and W. S. Noble, "A statistical framework for genomic data fusion," *Bioinformatics*, vol. 20, pp. 2626–2635, 2004.

[29] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[30] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proceedings of the twenty-first international conference on Machine learning*, ser. ICML '04. Canada: ACM, 2004.

[31] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, pp. 193–228, Nov. 1998.

[32] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in kernel methods*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.

[33] S. Sonnenburg, G. Rätsch, B. Schölkopf, and G. Rätsch, "Large scale multiple kernel learning," *J. Mach. Learn. Res.*, vol. 7, pp. 1531–1565, Dec 2006.

[34] J. Zhu and H. Zou, "Variable selection for the linear support vector machine," in *Trends in Neural Computation*, ser. Studies in Computational Intelligence, K. Chen and L. Wang, Eds. Springer Berlin / Heidelberg, 2007, vol. 35, pp. 35–59.

[35] L. Wang, J. Zhu, and H. Zou, "The doubly regularized support vector machine," *Statistica Sinica*, vol. 16, pp. 589–615, 2009.

[36] R. Tomioka and T. Suzuki, "Sparsity-accuracy trade-off in MKL," *ArXiv e-prints*, Jan. 2010, available at arXiv.org.

[37] F. Orabona and L. Jie, "Ultra-fast optimization algorithm for sparse multi kernel learning." in *ICML*, Washington, USA, Jun. 2011.

[38] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "l_p-norm multiple kernel learning," *J. Mach. Learn. Res.*, vol. 12, pp. 953–997, March 2011.

[39] C. S. Ong, A. J. Smola, and R. C. Williamson, "Learning the kernel with hyperkernels," *J. Mach. Learn. Res.*, vol. 6, pp. 1043–1071, Dec. 2005.

[40] I. W.-H. Tsang and J. T.-Y. Kwok, "Efficient hyperkernel learning using second-order cone programming," *IEEE Trans Neural Networks*, vol. 17, no. 1, pp. 48–58, Jan. 2006.

[41] C. Cortes, A. Gretton, G. Lanckriet, M. Mohri, and A. Rostamizadeh, NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels, 2008, available at http://www.cs.nyu.edu/learning_kernels/.

[42] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* Cambridge, MA: MIT Press Series, 2001.

[43] G. Camps-Valls, L. Gómez-Chova, J. Muñoz-Marí, J. Rojo-Álvarez, and M. Martínez-Ramón, "Kernel-based framework for multitemporal and multisource remote sensing data classification and change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 6, pp. 1822–1835, Jun 2008.

[44] M. Martínez-Ramón, J. L. Rojo-Álvarez, G. Camps-Valls, J. Muñoz-Marí, A. Navia-Vázquez, E. Soria-Olivas, and A. Figueiras-Vidal, "Support vector machines for nonlinear kernel ARMA system identification," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1617–1622, Nov 2006.

[45] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Computation*, vol. 12, no. 5, pp. 1207–1245, May 2000.

[46] V. Gómez-Verdejo, M. Martínez-Ramón, J. Arenas-García, M. Lázaro-Gredilla, and H. Molina-Bulla, "Support vector machines with constraints for sparsity in the primal parameters." *IEEE Transactions on Neural Networks*, vol. 22, no. 8, pp. 1269–1283, Aug 2011.

[47] L. Faybusovich and T. Mouktonglang, "Multi-target linear-quadratic control problem and second-order cone programming," Department of Mathematics, University of Notre Dame, Tech. Rep., 2002.

[48] *The MOSEK optimization toolbox for MATLAB manual Version 5.0 (Revision 137)*, MOSEK ApS, available at http://www.mosek.com.

[49] E. A. Allen, E. B. Erhardt, Y. Wei, T. Eichele, and V. D. Calhoun, *A simulation toolbox for fMRI data: SimTB*, Medical Image Analysis Laboratory (MIALAB), The MIND Research Network, Jun 2011, available at http://mialab.mrn.org.

[50] K. J. Friston, A. P. Holmes, K. J. Worsley, J.-P. Poline, C. D. Frith, and R. S. J. Frackowiak, "Statistical parametric maps in functional imaging: A general linear approach," *Human Brain Mapping*, vol. 2, no. 4, pp. 189–210, Oct 1994.

[51] K. J. Friston, A. Mechelli, R. Turner, and C. J. Price, "Nonlinear responses in fmri: The balloon model, volterra kernels, and other hemodynamics," *NeuroImage*, vol. 12, no. 4, pp. 466–477, Oct 2000.

[52] E. B. Erhardt, E. A. Allen, Y. Wei, T. Eichele, and V. D. Calhoun, "SimTB, a simulation toolbox for fMRI data under a model of spatiotemporal separability," *NeuroImage*, vol. 59, no. 4, pp. 4160–4167, Feb 2012.

[53] V. D. Calhoun, G. D. Pearlson, P. Maciejewski, and K. A. Kiehl, "Temporal lobe and 'default' hemodynamic brain modes discriminate between schizophrenia and bipolar disorder," *Hum. Brain Map*, vol. 29, no. 11, pp. 1265–1275, Nov 2008.

[54] A. G. Garrity, G. D. Pearlson, K. McKiernan, D. Lloyd, K. A. Kiehl, and V. D. Calhoun, "Aberrant 'Default Mode' Functional Connectivity in Schizophrenia," *Am J Psychiatry*, vol. 164, no. 3, pp. 450–457, Mar 2007.

[55] N. D. Woodward, B. Rogers, and S. Heckers, "Functional resting-state networks are differentially affected in schizophrenia," *Schizophrenia Research*, vol. 130, no. 1, pp. 86–93, Aug 2011.

[56] B. A. Vogt and S. Laureys, "Posterior cingulate, precuneal and retrosplenial cortices: cytology and components of the neural network correlates of consciousness," *Progress in Brain Research*, vol. 150, pp. 205–217, 2005.

[57] V. Calhoun, T. Adali, G. Pearlson, and J. Pekar, "A method for making group inferences from functional mri data using independent component analysis," *Human Brain Mapping*, vol. 14, no. 3, pp. 140–151, Nov 2001.

[58] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, Nov 1995.

[59] I. A. Wood, P. M. Visscher, and K. L. Mengersen, "Classification based upon gene expression data: bias and precision of error rates," *Bioinformatics*, vol. 23, no. 11, pp. 1363–1370, Jun 2007.

[60] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Columbus, OH: McGraw-Hill, Dec 2003.

[61] E. Castro, M. Martínez-Ramón, G. Pearlson, J. Sui, and V. D. Calhoun, "Characterization of groups using composite kernels and multi-source fMRI analysis data: Application to schizophrenia," *NeuroImage*, vol. 58, no. 2, pp. 526–536, Jun 2011.

[62] E. Castro, M. Martínez-Ramón, A. Caprihan, K. Kiehl, and V. D. Calhoun, "Complex fMRI data classification using composite kernels: Application to schizophrenia," Organization of Human Brain Mapping, 17th Annual Meeting, Canada, 2011.

[63] E. Castro, M. Martínez-Ramón, K. Kiehl, and V. D. Calhoun, "A multiple kernel learning approach for schizophrenia classification from complex-valued fMRI data," Organization of Human Brain Mapping, 19th Annual Meeting, Seattle, 2013.

[64] E. Castro, V. Gómez-Verdejo, M. Martínez-Ramón, K. A. Kiehl, and V. D. Calhoun, "A multiple kernel learning approach to perform classification of groups from complex-valued fMRI data analysis: Application to schizophrenia," *NeuroImage*, in press.

[65] D. Kim, J. Burge, T. Lane, G. Pearlson, K. Kiehl, and V. Calhoun, "Hybrid ica-bayesian network approach reveals distinct effective connectivity differences in schizophrenia," *NeuroImage*, vol. 42, no. 4, pp. 1560–1568, Oct 2008.

[66] O. Demirci, M. C. Stevens, N. C. Andreasen, A. Michael, J. Liu, T. White, G. D. Pearlson, V. P. Clark, and V. D. Calhoun, "Investigation of relationships between fmri brain networks in the spectral domain using ica and granger causality reveals distinct differences between schizophrenia patients and healthy controls," *NeuroImage*, vol. 46, no. 2, pp. 419–431, Jun 2009.

[67] V. D. Calhoun, T. Adali, K. A. Kiehl, R. Astur, J. J. Pekar, and G. D. Pearlson, "A method for multitask fmri data fusion applied to schizophrenia," *Human Brain Mapping*, vol. 27, no. 7, pp. 598–610, Jul 2006.

[68] American Psychiatric Association, *Diagnostic and Statistical Manual of Mental Disorders DSM-IV-TR Fourth Edition (Text Revision)*, 4th ed. Arlington, VA: American Psychiatric Publishing, Inc., June 2000.

[69] M. B. First, R. L. Spitzer, M. Gibbon, and J. B. W. Williams, *Structured Clinical Interview for DSM-IV Axis I Disorders-Patient Edition (SCID-I/P, Version 2.0)*, Biometrics Research Department, New York State Psychiatric Institute, New York, 1995.

[70] R. L. Spitzer, J. B. W. Williams, and M. Gibbon, *Structured Clinical interview for DSM-IV: Non-patient edition (SCID-NP)*, Biometrics Research Department, New York State Psychiatric Institute, New York, 1996.

[71] K. A. Kiehl and P. F. Liddle, "An event-related functional magnetic resonance imaging study of an auditory oddball task in schizophrenia," *Schizophrenia Research*, vol. 48, no. 2-3, pp. 159–171, Mar 2001.

[72] L. Freire, A. Roche, and J.-F. Mangin, "What is the best similarity measure for motion correction in fmri time series?" *Medical Imaging, IEEE Transactions on*, vol. 21, no. 5, pp. 470–484, May 2002.

[73] K. Friston, J. Ashburner, C. Frith, J. Poline, J. D. Heather, and R. Frackowiak, "Spatial registration and normalization of images," *Human Brain Mapping*, vol. 3, no. 3, pp. 165–189, 1995.

[74] Y.-O. O. Li, T. Adali, and V. D. D. Calhoun, "Estimating the number of independent components for functional magnetic resonance imaging data," *Hum Brain Mapp*, vol. 28, no. 11, pp. 1251–1266, Nov 2007.

[75] J. Lancaster, J. Summerln, L. Rainey, C. Freitas, and P. Fox, "The talairach daemon, a database server for talairach atlas labels," *NeuroImage*, vol. 5, p. S633, 1997.

[76] J. Lancaster, M. Woldorff, L. Parsons, M. Liotti, C. Freitas, L. Rainey, P. Kochunov, D. Nickerson, M. S.A., and P. Fox, "Automated talairach atlas labels for functional brain mapping," *Hum. Brain Mapp*, vol. 10, pp. 120—-131, 2000.

[77] J. Maldjian, P. Laurienti, R. Kraft, and J. Burdette, "An automated method for neuroanatomic and cytoarchitectonic atlas-based interrogation of fmri data sets," *NeuroImage*, vol. 19, pp. 1233—-1239, 2003.

[78] N. Correa, T. Adali, and V. D. Calhoun, "Performance of blind source separation algorithms for fmri analysis using a group ica method," *Magnetic Resonance Imaging*, vol. 25, no. 5, pp. 684–694, June 2007.

[79] A. R. Franco, A. Pritchard, V. D. Calhoun, and A. R. Mayer, "Interrater and intermethod reliability of default mode network selection." *Hum Brain Mapp*, vol. 30, no. 7, pp. 2293–2303, 2009.

[80] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot, "Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain," *NeuroImage*, vol. 15, no. 1, pp. 273–289, January 2002.

[81] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[82] C. S. Carter, I. MacDonald, Angus W., L. L. Ross, and V. A. Stenger, "Anterior Cingulate Cortex Activity and Impaired Self-Monitoring of Performance in Patients With Schizophrenia: An Event-Related fMRI Study," *Am J Psychiatry*, vol. 158, no. 9, pp. 1423–1428, Sep 2001.

[83] L. Ungar, P. G. Nestor, M. A. Niznikiewicz, C. G. Wible, and M. Kubicki, "Color stroop and negative priming in schizophrenia: An fmri study," *Psychiatry Research: Neuroimaging*, vol. 181, no. 1, pp. 24–29, Jan 2010.

[84] A. M. Fjell and K. B. Walhovd, "Structural brain changes in aging: courses, causes and cognitive consequences." *Reviews in the neurosciences*, vol. 21, no. 3, pp. 187–221, 2010.

[85] V. D. Calhoun, K. A. Kiehl, P. F. Liddle, and G. D. Pearlson, “Aberrant localization of synchronous hemodynamic activity in auditory cortex reliably characterizes schizophrenia,” *Biological Psychiatry*, vol. 55, no. 8, pp. 842–849, Apr 2004.

[86] F. G. Hoogenraad, J. R. Reichenbach, E. M. Haacke, K. K. S. Lai, and M. Sprenger, “In vivo measurement of changes in venous blood-oxygenation with high resolution functional mri at 0.95 tesla by measuring changes in susceptibility and velocity,” *Magn.Res.Med*, vol. 39, no. 1, pp. 97–107, Jan 1998.

[87] A. S. Nencka and D. B. Rowe, “Reducing the unwanted draining vein {BOLD} contribution in fmri with statistical post-processing methods,” *NeuroImage*, vol. 37, no. 1, pp. 177 – 188, 2007.

[88] R. S. Menon, “Postacquisition suppression of large-vessel BOLD signals in high-resolution fMRI,” *Magnetic Resonance in Medicine*, vol. 47, no. 1, pp. 1–9, Jan 2002.

[89] F. Zhao, T. Jin, P. Wang, X. Hu, and S.-G. Kim, “Sources of phase changes in bold and cbv-weighted fmri,” *Magnetic Resonance in Medicine*, vol. 57, no. 3, pp. 520–527, 2007.

[90] Z. Feng, A. Caprihan, K. B. Blagoevc, and V. D. Calhoun, “Biophysical modeling of phase changes in bold fmri,” *NeuroImage*, vol. 47, no. 2, pp. 540–548, Aug 2009.

[91] V. D. Calhoun and T. Adali, “Analysis of complex-valued functional magnetic resonance imaging data: Are we just going through a “phase”?” *Polish Academy of Sciences: Technical Sciences*, vol. 60, no. 3, pp. 371–667, 2012.

[92] V. Calhoun, T. Adali, G. Pearlson, P. van Zijl, and J. Pekar, “Independent component analysis of fMRI data in the complex domain,” *Magnetic Resonance in Medicine*, vol. 48, no. 1, pp. 180–192, Jul 2002.

[93] D. B. Rowe, “Parameter estimation in the magnitude-only and complex-valued fMRI data models,” *NeuroImage*, vol. 25, no. 4, pp. 1124–1132, May 2005.

[94] S. K. Arja, Z. Feng, Z. Chen, A. Caprihan, K. A. Kiehl, T. Adali, and V. D. Calhoun, “Changes in fMRI magnitude data and phase data observed in block-design and event-related tasks,” *NeuroImage*, vol. 49, no. 4, pp. 3149–3160, Feb 2010.

[95] L. Freire and J. Mangin, “Motion correction algorithms may create spurious brain activations in the absence of subject motion,” *NeuroImage*, vol. 14, no. 3, pp. 709–722, Sep 2001.

[96] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[97] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, Jul. 2011.

[98] J. A. Maldjian, P. Laurienti, and J. Burdette, "Precentral gyrus discrepancy in electronic versions of the talairach atlas," *NeuroImage*, vol. 21, no. 1, pp. 450–455, Jan 2004.

[99] S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. d. Bona, A. Binder, C. Gehl, and V. Franc, "The shogun machine learning toolbox," *J. Mach. Learn. Res.*, vol. 11, pp. 1799–1802, Aug. 2010, available at http://www.shogun-toolbox.org.

[100] P. Good, *Permutation Tests*. New York: Springer, 1994.

[101] J. Opdyke, "Fast permutation tests that maximize power under conventional monte carlo sampling for pairwise and multiple comparisons," *J. Mod. Appl. Stat. Methods*, vol. 2, no. 1, pp. 27–49, 2003.

[102] D. L. Clark, N. N. Boutros, and M. F. Mendez, *The Brain and Behavior: An Introduction to Behavioral Neuroanatomy*, 3rd ed. Cambridge University Press, June 2010.

[103] B. Crespo-Facorro, P. C. Nopoulos, E. Chemerinski, J.-J. Kim, N. C. Andreasen, and V. Magnotta, "Temporal pole morphology and psychopathology in males with schizophrenia," *Psychiatry Research: Neuroimaging*, vol. 132, no. 2, pp. 107 – 115, 2004.

[104] D. Adam, "Mental health: On the spectrum," *Nature*, vol. 496, pp. 416–418, 2013.

[105] G. Strang, *Linear Algebra and its Applications*. Pacific Grove, CA: Brooks Cole, 1988.

[106] C. Moler, *Numerical Computing with MATLAB*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 2004.