

KATHOLIEKE UNIVERSITEIT
LEUVEN

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Electrical Engineering

Kernels and Tensors for Structured Data Modelling

Marco SIGNORETTO

Dissertation presented in partial
fulfillment of the requirements for
the degree of Doctor
in Engineering

December 2011

Kernels and Tensors for Structured Data Modelling

Marco SIGNORETTO

Jury:
Prof. dr. P. Sas, chair
Prof. dr. J. Suykens, promotor
Prof. dr. J. Vandewalle, co-promotor
Prof. dr. L. De Lathauwer, co-promotor
Prof. dr. G. Gielen
Prof. dr. B. De Moor
Prof. dr. M. Van Barel
Prof. dr. R. Vandebril
Prof. dr. A. Verri
(DISI, Università degli Studi di Genova)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

December 2011

© Katholieke Universiteit Leuven – Faculty of Engineering
Kasteelpark Arenberg 10, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

D/2011/7515/162
ISBN 978-94-6018-463-5

A Giorgia.

Foreword

This Thesis is based on the research carried out during the years I spent as a PhD student at the SISTA research group. It is the result of an enriching experience that made me learn important things about science and life.

I wish to thank my promotor Johan Suykens who gave me the opportunity to undertake this journey. Above all I would like to thank him for his trust and his guidance during these years; this work owes much to his constructive comments and his valuable insight.

I would also like to express my deepest gratitude to my co-promotors Joos Vandewalle and Lieven De Lathauwer. Lieven brought tensors on my way and since then I benefited from very insightful and productive discussions. My appreciation is extended to my committee members who provided me with helpful comments: Georges Gielen, Bart De Moor, Marc Van Barel, Raf Vandebril, Alessandro Verri and Paul Sas.

During my doctoral studies I had the opportunity to enjoy the international and interdisciplinary atmosphere at SISTA. I interacted with great people from all over the world. In particular, I would like to acknowledge the researchers with whom I had the pleasure to collaborate with more closely: Kristiaan Pelckmans, Anneleen Daemen, Fabian Ojeda, Carlo Savorgnan, Tillmann Falck, Quoc Tran Dinh, Bori Hunyadi, Emanuele Olivetti and Geert Gins. I owe my sincere gratitude to the whole administrative staff; my appreciation goes especially to Ida Tassens, Ilse Pardon and John Vos for the help I received during these years.

I am also grateful to a number of other colleagues and friends who enriched my Belgian experience with cheerful moments. Special thanks go to my journey mates Denis and Jenny Marcon with whom I shared precious memories.

Finally, I would like to thank my whole family, especially my parents, for their continuous support and care. Without their inspiring education this work

would have never been possible. My most special thanks go to Giorgia for her endless love and for Mattia, our greatest achievement. My appreciation is also extended to Giorgia's family for keeping me confident along my path.

The support that I received up to now was a great incentive and I hope it will continue throughout the challenges and endeavors yet to come.

Marco Signoretto

Leuven, December 2011.

Abstract

A key ingredient to improve the generalization of machine learning algorithms is to convey prior information, either by choosing appropriate input representations or by tailored regularization schemes. This becomes of paramount importance in all the applications where the number of available observations for training is limited. In many of such cases, data are structured and can be conveniently represented as higher order arrays (tensors). The scope of this thesis is the development of learning algorithms that exploit the structural information of these arrays to improve generalization. This is achieved by combining tensor-based methods with kernels, convex optimization, sparsity and statistical learning principles.

As a first contribution we present a parametric framework based on convex optimization and spectral regularization. We give a mathematical characterization of spectral penalties for tensors and analyze a unifying class of convex optimization problems for which we present a new, provably convergent and scalable template algorithm. We then specialize this class of problems to perform learning both in a transductive as well as in an inductive setting. In the transductive case one has an input data tensor with missing features and, possibly, a partially observed matrix of labels. The goal is to both infer the missing input features as well as predict the missing labels. For induction, the goal is to determine a parametric model for each learning task to be used for out of sample prediction. Each training pair consists of a multidimensional array and a set of labels each of which corresponds to related but distinct tasks. As a by-product of using a tensor-based formalism, our approach enables one to tackle multiple tasks simultaneously in a natural way. Empirical studies demonstrate the merits of the proposed methods.

Parametric tensor-based techniques present a number of advantages; in particular, they often lead to interpretable models which is a desirable feature in a number of applications of interest. However they constitute a somewhat restricted class that might suffer from limited predictive power. A second

contribution of this thesis is to go beyond this limitation by introducing nonparametric tensor-based models. To this end we discuss two different ideas. The first approach is based on an explicit multi-way feature representation. The latter is found as the minimum norm solution of an operatorial equation and carries structural information from the input data representation. A main drawback is that estimation within this feature space results into non-convex and non-scalable problems. The second approach fits into the same primal-dual framework underlying SVM-like algorithms and allows the efficient estimation of nonparametric tensor-based models. Although specialized kernels exist for certain classes of structured data, no existing approach exploits the (algebraic) structure of tensorial representations. We go beyond this limitation by proposing a class of tensorial kernels that links to the multilinear singular value decomposition (MLSVD) and study properties of the proposed similarity measure.

The tensorial kernel is a special case of a more general class of product kernels. Product kernels, including the widely used Gaussian RBF kernel, play a special role in nonparametric statistics and machine learning. At a more fundamental level, we elaborate on the link between tensors and kernels. We show that, on the one hand, spaces of finite dimensional tensors can be regarded as RKHSs associated to product kernels. On the other hand, the Hilbert space of multilinear functionals associated to general product kernels can be regarded as a space of infinite dimensional tensors.

Many objects of interest, such as videos and colored images, admit a natural tensorial representation. Additionally, tensor representations naturally result from the experiments performed in a number of fields. On top of this, there are cases where one can explicitly carry on tensor transformations with the purpose of exploiting the spectral content of these new representations. We show that one of such transformations can be used for learning when input data are multivariate time series. We represent these objects by cumulant tensors and train classifiers based on tensorial kernels. Contrary to existing approaches the arising procedure does not require an (often nontrivial) blind identification step. Nonetheless, insightful connections with the dynamics of the generating systems can be drawn under specific modeling assumptions. The approach is illustrated on a brain decoding task where the direction, either left of right, towards where the subject modulates attention is predicted from magnetoencephalography (MEG) signals.

Abbreviations

ALS	Alternating Least Squares
ARMA	Auto-Regressive Moving Average
AUC	Area Under the Curve
CP	canonical polyadic
CV	Cross-Validation
EEG	Electroencephalography
EG	Encephalography
ERM	Empirical Risk Minimization
HMM	Hidden Markov Model
HS	Hilbert Space
HSF	HS of Hilbert-Schmidt multilinear functionals
LASSO	Least Absolute Shrinkage and Selection Operator
LIBRAS	Língua Brasileira de Sinais (Brasilian Sign Language)
LOO	Leave-One Out
LS-SVM	Least Squares Support Vector Machine
MALDI	Matrix-assisted laser desorption/ionization
MEG	Magnetoencephalography
MIMO	Multi-Input Multi-Output
MKL	Multiple Kernel Learning
ML	Maximum Likelihood
MLSVD	Multilinear Singular Value Decomposition
MSE	Mean Square Error
MSI	Mass Spectral Imaging

NP	non-polynomial
NRMSE	Normalized Root Mean Squared Error
RBF	Radial Basis Function
RKHS	Reproducing Kernel Hilbert Space
ROC	Receiver Operating Characteristic
SRM	Structural Risk Minimization
SVD	Singular Value Decomposition
SVM	Support Vector Machine
VC	Vapnik-Chervonenkis

List of Symbols

$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \cdots$	Matrices
$\boldsymbol{A}^{\langle n \rangle}$	$n-$th mode tensorization of a matrix $\boldsymbol{A}$
$\mathscr{H}_1 \otimes \mathscr{H}_2$	Space of Hilbert-Schmidt operators from $\mathscr{H}_2$ to $\mathscr{H}_1$
$\langle \cdot, \cdot \rangle$	Inner product
$\{x(t)\}$	Stochastic vector process
$\mathbb{N}_N$	Set $\{1, 2, \ldots, N\}$
$\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$	Higher-order tensors
$\mathcal{A} \times_n \boldsymbol{U}$	$n-$th mode product between $\mathcal{A}$ and $\boldsymbol{U}$
$\mathcal{A}_{\langle n \rangle}$	$n-$th mode matricization of a tensor $\mathcal{A}$
$\mathcal{C}^x(l_1, l_2, \ldots, l_{J-1})$	$(l_1, l_2, \ldots, l_{J-1})$-cumulant tensor of $\{x(t)\}$
$\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}}$	$4-$th order sample cumulant tensor of $\{x(t)\}$ with reference signal r
$\mathcal{C}^x_{\mathscr{L}}$	$\mathscr{L}$-cumulant tensor of $\{x(t)\}$
$\mathscr{A}, \mathscr{B}, \mathscr{C}, \ldots$	Sets and Spaces
$\mathscr{D}_N$	Dataset composed of N input-target pairs
$\mathscr{N}(A)$	Null space of A
$\mathscr{R}(A)$	Range of A
$\mathscr{X} \times \mathscr{Y}$	2-fold Cartesian Product of vector spaces $\mathscr{X}$ and $\mathscr{Y}$

$\text{prox}_h(x)$	Proximity operator of a function h evaluated at x
$\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$	Space of $N-$th order tensors
$\dim(\mathscr{A})$	Vector space dimension of $\mathscr{A}$
$\times_{m \in \mathbb{N}_M} \mathscr{A}_m$	M-fold Cartesian product $\mathscr{A}_1 \times \mathscr{A}_2 \times \cdots \times \mathscr{A}_M$
vec	Vectorization operator
$\|\cdot\|$	Hilbert-Frobenius norm ($\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$)
$\|\cdot\|_*$	Nuclear norm (a.k.a trace-norm, Shatten-1 norm)
$\|\cdot\|_1$	l_1 norm
$A, B, C, \cdots$	Abstract operators
$a, b, c, \ldots$	Elements of $\mathbb{R}^D$ ($D \geq 1$), functions and abstract vectors
$A \otimes B$	Kronecker product of A and B
$I, J, K, M, \cdots$	Upper indices
$i, j, k, m, \cdots$	Running indices
$i_1 i_2 \cdots i_N$	Generic element of $\mathbb{N}_{I_1} \times \mathbb{N}_{I_2} \times \cdots \times \mathbb{N}_N$
R_T	Resolvent of T
$S_{\mathscr{S}}$	Sampling operator defined according to the ordered set $\mathscr{S}$
$v_1 \otimes v_2 \otimes \cdots \otimes v_N$	$N-$th order rank-1 tensor

Contents

Foreword iii

Abstract v

Abbreviations viii

List of Symbols x

Contents xi

1 Introduction 1

1.1 The Three Cornerstones 1

1.1.1 Tensor-based Methods 2

1.1.2 Kernel Methods 2

1.1.3 Convexity 3

1.2 Challenges and Objectives 3

1.3 Contributions 7

1.4 Thesis Structure 12

2 Kernels, Tensors and Learning 15

2.1 Foundations of Learning 15

2.1.1 General Setting for Statistical Learning Problems 16
2.1.2 Supervised and Unsupervised Learning 16
2.1.3 Semi-supervised Learning and Transduction 17
2.1.4 Discriminative Versus Generative Methods 18
2.1.5 The SRM Principle for Induction and Transduction . . 19
2.2 Learning through Regularization 24
2.2.1 Tikhonov Theory 24
2.2.2 SRM and Regularization in RKHSs 25
2.2.3 Alternative forms of Regularization 26
2.3 Aspects of Learning with Tensors 28
2.3.1 The Need for Informative Data Representations 28
2.3.2 Tensors and Higher-order Data Representations 31
2.3.3 Parametric and Non-parametric Tensor-based Models . 36
2.4 Chapter Summary 37

3 Spectral Learning based on Parametric Models 39

3.1 General Problem Setting 41
3.1.1 Abstract Vector Spaces 42
3.1.2 Some Illustrative Examples 43
3.2 Template Algorithm 45
3.2.1 Overview of Proximal Point Algorithms 45
3.2.2 Douglas-Rachford Splitting Technique 47
3.2.3 Limits of Two-level Strategies 47
3.2.4 Template Based on Inexact Splitting Technique 48
3.3 Spectral Regularization for Higher Order Arrays 50
3.3.1 Spectral Penalties for Higher Order Tensors 50
3.3.2 Relation With Multilinear Rank 51

3.3.3 Proximity Operators 52
3.3.4 Multiple Module Spaces 53
3.4 Multi-task Transductive Learning 54
3.4.1 Modelling Assumptions 55
3.4.2 Sampling Operator and its Adjoint 57
3.4.3 Soft-completion of Heterogeneous Data 58
3.4.4 Hard-completion without Target Labels 61
3.5 Multi-task Inductive Learning 63
3.5.1 Modelling Assumptions 64
3.5.2 Problem Formulation 65
3.5.3 Algorithm 66
3.6 Experiments 66
3.6.1 Transductive Learning 66
3.6.2 Hard completion of Hyperspectral Data 76
3.6.3 Inductive Learning 80
3.7 Chapter Summary 84

4 Kernel-based Learning for Data Tensors 89

4.1 Learning with Explicit Feature Representations 89
4.1.1 Input Data Matrices 90
4.1.2 Auxiliary Operators and Operatorial Equation 90
4.1.3 Feature Representation of Data Matrices 91
4.1.4 Model Estimation 94
4.1.5 Experiments 97
4.2 Learning with Kernels for Data Tensors 98
4.2.1 Naive Kernels for Tensors 99
4.2.2 Structure Exploiting Kernels 100

4.2.3 Congruent Data Tensors and Invariance Property 104
4.2.4 Model Estimation 107
4.2.5 Experiments 111
4.3 Chapter Summary 119

5 Classification of Signals with Cumulant-based Kernels 121
5.1 Problem Statement 121
5.1.1 Discriminative Versus Generative Methods 122
5.1.2 Existing Kernels for Vector Processes 123
5.2 Tensors and Cumulants 124
5.2.1 Cumulants of Vector Processes 124
5.2.2 Cumulant Tensors 124
5.2.3 Sample versions 125
5.2.4 Cumulants of State-Space Processes 126
5.3 Cumulant-based Kernels 127
5.3.1 Cumulant-based Kernel Functions 128
5.3.2 Computational Aspects 128
5.3.3 Connection with the System Dynamics 130
5.4 The Case of EG Signals 132
5.4.1 Classification Problems Involving EG Signals 132
5.4.2 Interpretation of Theorem 5.3.1 133
5.5 Experiments 133
5.5.1 Synthetic Examples: 3-class problems 134
5.5.2 Biomag 2010 Data 137
5.6 Chapter Summary 138

6 Conclusions 141
6.1 Concluding Remarks 141

6.2 Perspectives for Future Research 143

Mathematical Concepts and Tools **148**

A Hilbert Spaces **149**

A.1 Normed Vector Spaces 149

A.2 Linear Mappings 155

A.3 Inner Products and Hilbert Spaces 159

A.4 Operators on Hilbert Spaces 165

B Kernels and Spaces of Functions **173**

B.1 Notions of Kernels 174

B.2 Properties of Kernels 178

B.3 Data-dependent Kernels 181

B.4 Kernels and Tensors 182

C Optimization in Hilbert Spaces and LS-SVM **189**

C.1 Generalized Differential and Gradient 189

C.2 Lagrange Multipliers Theorem 191

C.3 Derivation of LS-SVM for Classification 192

Bibliography **195**

Publications by the Author **217**

Chapter 1

Introduction

The discipline of machine learning lies at the interface of a number of different fields including statistics, biology and computational neuroscience. Machine learning studies date back at least to the 1950's, when the perceptron was invented. In the past decades, machine learning algorithms have helped to analyze data, supporting humans in facing both scientific as well as technical challenges.

In the early applications people were performing experiments in controlled environments; in a subsequent step, the outcome of these experiments, a collection of measurements, were mined by early computers in order to extract useful information concerning the process under study. Meanwhile, the progressive availability of computational power allowed practical applications to flourish in the most diverse domains. As a result, a new trend is nowadays emerging and it is understood that machine learning has a much broader role to play. The challenge consists of being able to process data coming from the most diverse sources of information on a continuous base. In some cases these data are *structured* and can be conveniently represented as higher order arrays. Exploiting the structural information of these arrays to devise better learning algorithms is the scope of this thesis.

1.1 The Three Cornerstones

From a methodological perspective a major role in this work is played by three interplaying mathematical tools that we discuss next. Individually, these

tools have been investigated thoroughly in the past century. However, only recently different research communities have started exploring them *jointly*. Even more recent is the attempt to combine seemingly distinct ideas from the three domains with the purpose of devising better practical algorithms.

1.1.1 Tensor-based Methods

Tensors are a generalization of vectors and matrices to higher dimensions. Existing tensor techniques [56, 125, 127] are mostly based on decompositions that to some extent generalize the matrix singular value decomposition (SVD). Various tensor (multi-way) decompositions have been proposed in the last decades for unsupervised learning/exploratory data analysis. There are several advantages over (two-way) matrix factorizations; this includes for instance the uniqueness of optimal solutions obtained via certain tensor techniques. Furthermore, tensor decompositions are often particularly effective in low signal to noise ratios and when the number of observations is small in comparison with the dimensionality of the data. Tensor-based methods explicitly exploit the multi-way structure which is lost when collapsing the modes of the tensor in order to analyze the data by matrix factorizations. Applications include the analysis of image ensembles (tensor faces) and the analysis of electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) data to extract activation patterns across multiple medical trials and the identification of dynamical systems.

1.1.2 Kernel Methods

Kernel methods [110, 177, 178] have a long tradition in mathematics and statistics and, later, in machine learning and other fields. The exploration of their mathematical foundation dates back to the beginning of the previous century and received a new impulse in the last two decades by the development of kernel-based techniques within machine learning. The use of kernel methods is systematic and properly motivated by statistical principles. In practical applications, kernel methods lead to flexible predictive models that often outperform competing approaches in terms of generalization performance. The core idea consists of mapping data into a high dimensional space by means of a feature map. Since the feature map is normally chosen to be nonlinear, a linear model in the feature space corresponds to a nonlinear rule in the original domain. This fact suits many real world data analysis problems that often require nonlinear models to describe their structure. On the other hand, the so-called kernel trick allows one to develop computationally feasible

approaches regardless of the high-dimensionality of feature space representation. Perhaps the most popular techniques within kernel methods are Support Vector Machines (SVMs). The latter formulate learning as an optimization problem within a suitably defined vector space (the feature space mentioned above). The estimation is designed based on the geometry of the problems at hand.

1.1.3 Convexity

In general, finding a model by minimization of the *empirical risk* (an empirical proxy of the theoretical prediction error) is NP-hard [144], see also [110] and references therein. In spite of this, the main lesson taught by SVMs to the machine learning community is that one can still devise practical optimization problems that enjoy good computational and statistical properties. A fundamental aspect is represented by the *convexity* of these optimization problems. This property defines a subclass of nonlinear optimization problems for which all solutions are globally optimal: no improvement is admissible by any other feasible model. Convex problems, on the one hand, are amenable to efficient solution strategies. On the other hand, the study of the optimality of the solution with respect to the original aim, the minimization of the theoretical prediction error, can benefit from tools of convex analysis.

On a different track, in recent years exciting new developments in compressed sensing have shown that a number of combinatorial problems, under certain conditions, can be solved exactly by convex programming techniques [14,15,62, 72]. A key assumption is *sparsity*. For instance it was shown that certain convex programs based on the l_1 norm can find the solution of an underdetermined system of linear equations. This is ensured under the condition that the true solution is sparse. These findings stimulated the development of novel sparse techniques for inverse problems. In learning a number of *structure inducing penalties* have been proposed [118, 169, 237, 239]. These new forms of regularization amount at incorporating different types of prior knowledge and structural assumptions over the problem under study. From a theoretical standpoint, various forms of consistencies can be usually proved based on how well the underlying structure is captured by the penalty function (see, e.g., [148] and references therein).

1.2 Challenges and Objectives

This research was carried out with the long term aim — in line with the recent interest of the machine learning community [1] — of combining

the algebraic know-how developed along with tensor-based methods with kernels, convex optimization, sparsity and statistical learning principles. This encompasses both theoretical and algorithmical studies and is expected to impact significantly in all those contexts where data are structured and the number of available observations is limited. This is commonly encountered, for instance, in domains such as bioinformatics, biosignal processing and mass spectral imaging. Two main objectives are then:

A Develop and analyze a systematic kernel-based framework to tensorial data analysis.

B Based on convex optimization, design algorithms that combine tensors and kernels.

Next we elaborate on a number of research challenges that emerge within these two general goals.

A - Kernel-based Framework to Tensorial Data Analysis

The scientific and technical literature has shown that tensor representations of structured data are particularly effective for improving generalization of learning machines. On the other hand kernels lead to flexible models that have been proven successful in many different contexts. Recent research shows that it is possible to bring together these desirable properties. One approach that we discuss in this thesis is to define kernels that exploit the spectral content of tensors thereby obtaining the best of the two worlds. This opens up new interesting directions.

Understanding Learning Principles through Tensors and Manifolds Empirical observations show improved learning rates for tensor models in small sample problems. This grounds on the well established principle that exploiting prior knowledge and structural information is often a good idea. However, until now it is still far from clear in the general case how to characterize this principle from a mathematical perspective. The fact that tensors come with a wealth of (algebraic) structure suggests a practical way to tackle the investigation. The kernel proposed in [181], for instance, establishes a similarity measure between the different subspaces spanned by higher-order tensors [65]. In turn this allows one to demonstrate certain invariances that could serve useful to better understand the properties of learning algorithms combining tensors and kernels. Generally speaking, keeping into account the manifold structure of data [5, 25] has become a common light motif in machine learning and many other domains

in the last few years. It has been shown that for certain learning problems the convergence rates depend on the intrinsic dimension of the manifold and not on the dimension of the ambient space [147]. This sheds some light on the nature of the curse of dimensionality showing that learning is still feasible in spite of the seemingly high dimensionality of the data. The importance of this lies in the fact that many objects of interest are known to live in a low dimensional manifold embedded in a high dimensional space. In turn, such non-linear objects often admit a natural tensorial representation. This fact allows one to exploit useful instrumental results from the (algebraic) geometry of spaces of tensors. This field is being actively studied [129] and is already rich of useful instrumental results that can support additional research on the role of tensorial kernels in learning.

Kernels for Dynamical Systems and Networks In particular, the space of dynamical models is known to be strongly nonlinear. For instance, the space of linear dynamical systems, which are determined only up to a change of basis, has the structure of a *Stiefel manifold* [219]. A characterization of its geometry is fundamental to carry on recognitions and decisions for tasks involving dynamical systems [64]. Recently [30, 233] provided useful frameworks to establish a similarity measure between systems by means of kernels. However, these kernels require the knowledge of a model and hence demand for the foregoing identification of the generating systems. This step poses many challenges: most of the identification techniques are multistage iterative procedures that can suffer from the problem of multiple local minima. In contrast, one can define tensorial kernels that are entirely data-driven (they do not require to estimate models) and are still capturing the essential features of the dynamics of the underlying systems. This is of particular importance given that the same ideas can be used in the context of networks. Graphs and networks are widely used nowadays to study the relationship between structured objects. In this context nodes represent objects whereas edges represent the interaction among them. A natural way to define similarity measures between graphs and networks is based upon random walks [232, 233]. In turn, the evolution of random walks can be conveniently described by means of dynamical systems. Therefore tensorial kernels can be defined and are likely to be effective also for the analysis of graphs.

B - Kernel-tensor Learning via Convex Optimization

The second objective is to develop cutting-edge algorithms for data analysis by thoroughly combining optimization-based modeling with tensors and kernels. As mentioned above, most of the classical tensor data analysis techniques are unsupervised and to some extent can be considered as a generalization

of the matrix SVD. However recent research took a different perspective and introduced structure-preserving representations for supervised learning tasks [102, 204]. Other than classical machine learning problems, such as classifications, other formulations are possible, such as collaborative filtering and multi-task learning [2, 8] and there is room for substantial contributions. Additionally, several papers (such as [204]) make use of rank-1 tensors. However one could envisage the use of tensors parametrized in canonical polyadic (CP) form, with low rank, or in Tucker form, with low multilinear rank.

Tensor-based Models based on Non-smooth Convex Optimization A different approach is considered in [182]. In this case one does not start the modeling from a fixed parametrized structure but rather uses a convex penalty to enforce a solution with approximately low multilinear ranks. From a methodological perspective, the considered general formulation can be specialized to accomplish different (and possibly supervised) data-driven modeling tasks that demand for additional studies. So far this includes for instance completion problems [86, 135, 189] as well as recognition of patterns represented as tensors. Notice that the tensor representation could naturally result from the structure of the data to be analyzed. Alternatively one could deliberately arrange vectors or matrices in multi-way arrays, so that powerful tensor techniques can be used. Additionally, the approach in [182] has the advantage of bringing in the well established domain of convex optimization. In fact it seems that only very simple methods are normally being used within tensor algorithms, like Alternating Least Squares (ALS). However it is known that ALS can be very slow for ill-conditioned problems and can even fail to converge. Alternatively greedy algorithms are often considered with the price of obtaining suboptimal solutions. Convex problems do not suffer from this limitation. Moreover, optimal first order schemes — that recently attracted renewed interest — ensure that each iteration is reasonably cheap even for large scale problems.

Design of Structure Inducing Penalties and the Role of Sparsity Recent years witnessed exciting new developments in sparse techniques and compressed sensing. Even when no extra a priori knowledge is available, using the fact that features and structures of interest are typically concentrated on subspaces or manifolds of small(er) dimensions has been shown to be a key feature for the success of algorithms. The convex penalty used in [86, 135, 182] extends the nuclear norm for matrices. In turn the latter is the matrix analogue of the l_1 norm for vectors that underlies the LASSO and many compressed sensing algorithms. For the case of matrices it has been shown that other norms and penalties are of interest other than the nuclear norm [39]. In general the design of a penalty function should be tailored to the problem of interest [238]. This of

course regards tensor representations too and generates a new exiting frontier of research.

Design of Proximity Mappings for Tensors The convex techniques for tensors proposed so far work in connection with the Tucker decomposition and use matricization as a main tool. An important topic for future research deals with broadening this approach. In particular, an important direction of future research deals with proximity mappings. Proximity mappings are fundamental for deriving scalable first order methods like the Douglas-Rachford or the backward-forward splitting technique [19, 51]. For the case of vectors and matrices, for instance, proximity operators are known for a number of functions. This area is largely unexplored for tensors. For example a proximity mapping for the nuclear norm [86, 182, 212] is not known[1]. In general, one could explicitly design proximal problems so that a closed form (or a simple multi-step procedure) in the tensor unknown exists. These proximal problems would then serve as building blocks within iterative algorithms for solving a number of convex problems of interest.

1.3 Contributions

The starting point of this work is the realization that many different types of structured data such as images, videos and dynamical systems admit a natural tensorial representation. The latter comes with a wealth of algebraic structure that can be exploited for the purpose of learning better models.

The contribution of this thesis goes mainly in two directions.

Spectral Learning based on Parametric Models The first direction consists of an algorithmic framework based on convex optimization and spectral regularization to perform learning. This includes in particular the cases where observations are vectors or matrices. In addition, it enables to deal appropriately with data that have a natural representation as higher-order arrays. We begin by presenting a unifying class of convex optimization problems for which we present new, provably convergent and scalable algorithms based on an operator splitting technique. This class of problems is specialized to perform single as well as multi-task learning both in a transductive as well as

[1] For matrices, the proximity mapping of the nuclear norm is the singular value shrinkage operator, see [38].

in an inductive setting. To this end we develop new tools extending to higher order tensors the concept of spectral regularization for matrices.

In the transductive case one has an input data tensor with missing features and, possibly, a partially observed matrix of labels. The goal is to both infer the missing entries in the data tensors as well as predict the missing labels. Notably, the special case when there is no labeling information, corresponds to tensor completion that was considered for the first time in [135] and can be regarded as a single learning task. For the case where input patterns are represented as vectors our approach boils down to the formulation in [92]. In this sense the transductive formulation that we propose can be regarded as a generalization to the case when input data admit a higher order representation. As in that case the essential idea consists of regularizing the collection of input features and labels directly without learning a model. For the second family of problems that we consider, within the setting of inductive learning, the goal is to determine a model for each learning task to be used for out of sample prediction. Each training pair consists of an input tensor data observation and a vector of labels that corresponds to related but distinct tasks. This setting extends the standard penalized empirical risk minimization problem to allow for both multiple tasks and higher order observational data.

Both for the transductive and inductive cases, regularization is based on composite spectral penalties and connects to the concept of Tucker decomposition. The generality of the proposed methods ensures the applicability in a large variety of problems involving structured data. As a by-product of using a tensor-based formalism, our general approach allows one to tackle the multi-task case (dealing with multiple learning tasks simultaneously) in a natural way.

[188] M. Signoretto, Q. Tran Dinh, L. De Lathauwer and J.A.K. Suykens. Learning with tensors: a framework based on convex optimization and spectral regularization. *Submitted*, Internal Report 11-129, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 43 pages, 2011.

[189] M. Signoretto, R. Van de Plas, B. De Moor and J. A. K. Suykens. Tensor versus matrix completion: a comparison with application to spectral data. *IEEE Signal Processing Letters*, 18(7):403—406, 2011.

[114] B. Hunyadi, M. De Vos, M. Signoretto, J. A. K. Suykens, W. Van Paesschen and S. Van Huffel. Automatic Seizure Detection Incorporating Structural Information. *ICANN 2010, Part II, LNCS 6791*,233—240, 2011.

[180] M. Signoretto, L. De Lathauwer and J.A.K. Suykens. Convex multilinear estimation and operatorial representations. In *NIPS Workshop on Tensors, Kernels and Machine Learning*, 6 pages, 2010.

Kernel-based Learning with Data Tensors Most of tensor-based models (including those arising within the aforementioned framework) are linear with respect to the data. This sometimes results into limited discriminative power. In contrast, kernel models proved to be very accurate thanks to their flexibility. Specialized kernels exist for certain classes of structured data. However no existing approach exploits the (algebraic) structure of tensorial representations. In particular, using kernel functions for vectors does not exploit structural properties possessed by the given tensorial representations. We discuss two approaches to go beyond this limitation. The first idea is based on an explicit multi-way feature representation and it is shown to outperform competing approaches on small scale problems. This confirms that keeping into account the higher-order nature of certain data does help in improving generalization. This is in line with previous findings on tensor-based parametric models. Unfortunately the idea leads to non-convex and non-scalable problem formulations. The second approach consists of a class of tensorial kernels that links to the MLSVD and features an interesting invariance property; the approach fits into the same primal-dual framework underlying SVM-like algorithms and allows one to efficiently estimate nonparametric tensor-based models.

Tensorial kernels are a special case of a more general class of product kernels. This large class of kernel functions includes the widely used Gaussian RBF kernel and plays a central role to construct nonparametric multivariate models. An additional contribution of this thesis, at a more fundamental level, is to draw the link between these type of kernels and tensors. We show that spaces of finite dimensional tensors can be regarded as RKHSs associated to certain product kernels. On the other hand, the feature space associated to general product kernels can be regarded as a space of infinite dimensional tensors. This fact addresses the recent interest shown for the interplay between these two seemingly separate mathematical concepts. We discuss the consequences of this insight.

A specific class of structured data are multivariate time series (multichannel signals). For this case we proposed a cumulant-based kernel function. The latter measures similarity based upon the spectral information of tensors of higher order cross-cumulants associated to each multichannel signal. Alternative approaches either neglect the dynamical nature of time series or are generative in nature. Contrary to this latter class of techniques the use of

the cumulant-based kernel does not require the estimation of a model for each observed multivariate sequence. In fact, the method is entirely data-driven and does not require the assumption of any specific model class. Nonetheless, we show that insightful connections with the dynamics of the generating systems can be drawn under specific modeling assumptions. We illustrated the method on a brain decoding task where the direction, either left or right, towards where the subject modulates attention is predicted from magneto-encephalography (MEG) signals.

[181] M. Signoretto, L. De Lathauwer and J. A. K. Suykens. Kernel-based learning from infinite dimensional 2-way tensors. In *ICANN 2010, Part II, LNCS 6353*,59—69, 2010.

[183] M. Signoretto, L. De Lathauwer and J. A. K. Suykens. A kernel-based framework to tensorial data analysis. *Neural Networks*, 24(8):861—874, 2011.

[184] M. Signoretto, E. Olivetti, L. De Lathauwer and J. A. K. Suykens. Classification of multichannel signals with cumulant-based kernels. *Submitted*, Internal Report 10-251, ESAT-SISTA, K.U. Leuven (Leuven, Belgium), 10 pages, 2010.

Although related to the topic of this thesis, mainly from a methodological perspective, the following contributions are not covered in the next chapters for reasons of consistency.

Data-dependent Penalties and Kernels for Nonparametric Learning It is known [45, Chapter 24] that a possible way to improve generalization of learning algorithms is to adaptively shape the *hypothesis space* (i.e., the set of candidate models) by incorporating structural information of the problem under study. One way to do so is devising regularization schemes that take into account the geometry of the empirical sample [25,190]. In [185,186] we consider the problem of learning sparse, nonparametric models from observations drawn from an arbitrary, unknown distribution. This specific problem leads us to an algorithm extending techniques for Multiple Kernel Learning (MKL), functional ANOVA models and the Component Selection and Smoothing Operator (COSSO). The key element is to use a data-dependent penalty that adapts to the specific distribution underlying the data. Equivalently, this corresponds to adaptively modifying the set of kernel functions that are associated to the functional components of a candidate additive model. A related methodology emerges in the problem of learning partial linear models. These models are formed by a parametric part, defined based upon a prescribed set of functions, and a

nonparametric part, associated to a kernel. In [79] a novel estimation algorithm for these type of models is proposed within the context of system identification; the approach aims at decoupling the estimation of the parametric part from the nonparametric one. The main idea is to consider a certain orthogonal constraint in the optimization problem used to estimate the model. From a functional perspective, as shown by the author of this thesis, this corresponds to modifying the kernel function used for the nonparametric part based upon the training set and the set of parametric functions. The approach makes use of certain results concerning reproducing kernel Hilbert spaces, see also Appendix B.3 of this thesis.

[187] M. Signoretto, K. Pelckmans and J. A. K. Suykens. Quadratically Constrained Quadratic Programming for Subspace Selection in Kernel Regression Estimation. In *ICANN 2008, Part I, LNCS 5163*,175—184, 2008.

[186] M. Signoretto, K. Pelckmans, L. De Lathauwer and J. A. K. Suykens, Improved nonparametric sparse recovery with data matched penalties. In *The 2nd International Workshop on Cognitive Information Processing (CIP)*, (2010).

[185] M. Signoretto, K. Pelckmans, L. De Lathauwer and J. A. K. Suykens. Data-dependent Norm Adaptation for Sparse Recovery in Kernel Ensembles Learning. *Submitted*, 32 pages, 2010.

[79] T. Falck, M. Signoretto, J. A. K. Suykens and B. De Moor. A two stage algorithm for kernel based partially linear modeling with orthogonality constraints. *Submitted*, 15 pages, 2011.

Graph-based Penalties for Learning High Dimensional Sparse Models High dimensional linear modeling deals with the problem of estimating a $p-$dimensional coefficient vector based on a sample of size n when p is large and, possibly, $n \ll p$. A general approach prescribes to use penalized empirical risk minimization to find an estimate. Theoretical as well as empirical results show that, loosely speaking, the "goodness" of the estimated model depends on how well the underlying structure is captured by the penalty function used in estimating the model, see [148] and references therein. In the high dimensional setting, a popular example of structural assumption is *sparsity* and the corresponding choice of the penalty is the l_1 norm. Although appealing for many problems, the l_1 penalty has been shown to suffer from some drawbacks [242]. In particular, when groups of correlated variables are present, the LASSO tends to select only one variable out of the group without paying attention to which one is selected. In random design, namely when the n input-output

pairs are i.i.d. according to some unknown probability distribution, this leads to an unstable behavior in the sense that different realizations of the dataset might produce very different results. This fact stimulated the design of new regularization approaches [118, 237, 239, 242]. The general idea is to convey structural assumptions on the problem, by suitably crafting the penalty.

We focused on a method that requires to endow the set of covariates with a graph structure. In this case nodes are variables, edges represent interactions and groups naturally emerge as connected components of the graph. Our work extends the elastic-net [242] and recent research on network constrained selection [130]. We proposed an approach to learn predictive models with grouping effect induced by multiple noisy estimates of some latent graph, relevant to the task of interest. The penalty enforces smooth profile of coefficients associated to neighboring nodes and ensures that coefficients with small contribution would shrink to exact zero. The aim is both to discover and model macro-variables that are highly predictive. Such problems commonly arise for example in bioinformatics, where markers for a certain disease are often found to coincide with entire groups of variables and multliple graphs are available to model the interactions. We also considered an extension of the approach to devise semi-supervised methods. In this case, to overcome the lack of labeled data, we additionally pose a connectivity graph over the set of predicted labels.

[179] M. Signoretto, A. Daemen, C. Savorgnan and J. A. K. Suykens, Variable selection and grouping with multiple graph priors. In *NIPS Workshop on Optimization for Machine Learning*, (2009).

[61] A. Daemen, M. Signoretto, O. Gevaert, J. A. K. Suykens and B. De Moor, Improved microarray-based decision support with graph encoded interactome data. *PLoS ONE*, 5(4):1—16, 2010.

[153] F. Ojeda, M. Signoretto, R. Van de Plas, E. Waelkens, B. De Moor and J. A. K. Suykens, Semi-supervised Learning of Sparse Linear Models in Mass Spectral Imaging. In *Pattern Recognition in Bioinformatics (PRIB), vol. 6282 of Lecture Notes in Bioinformatics*, 325—334, 2010.

1.4 Thesis Structure

In the next Chapter we introduce the general setting of learning from examples via regularization; we then recall basic facts about tensors and set up the tensor-based tools used within the rest of the thesis. Chapter 3 deals with a general parametric framework for tensor-based models. This comprises both

transductive as well as inductive learning algorithms. In Chapter 4 we discuss kernel based approaches for nonparametric tensor-based models. Chapter 5 deals with nonparametric classification of multivariate time series via cumulant based kernels. In Chapter 6 we draw our concluding remarks.

The main body of this thesis is accompanied with a self-contained account on some mathematical foundations. The material is not included elsewhere to improve readability; the relation with the main text is illustrated in Figure 1.1. Appendix A deals with Hilbert spaces; it contains those basic concepts which are prerequisites to deal with reproducing kernels. These are the subject of Appendix B. The content of these appendices is originally and organically assembled from scattered sources; it does not contain, however, new results from the author. There are three important exceptions to this. Appendix B.4 deals with data dependent kernels which arise from taking a functional viewpoint in the estimation of partial linear models [79]. Appendix B.3 bridges the gap between kernels and finite dimensional tensors. It is shown that spaces of finite dimensional tensors can be interpreted as reproducing kernel Hilbert spaces associated to certain product kernels; to the best of our knowledge this interpretation can not be found in the literature. Finally, Appendix C deals with a formal derivation of LS-SVM for classification starting from results on optimization in Hilbert Spaces. This is also not found elsewhere.

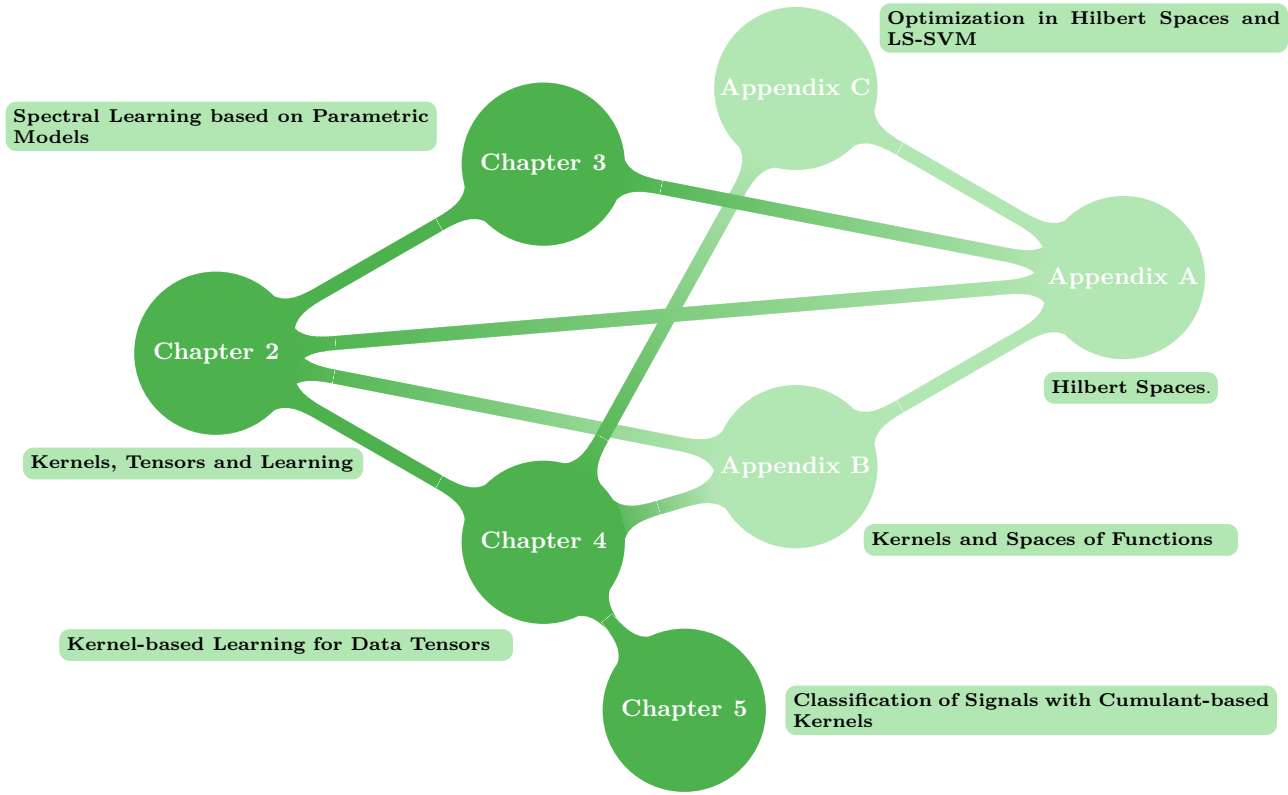


Figure 1.1: The dependence structure of the thesis. In light green the appendices.

Chapter 2

Kernels, Tensors and Learning

The scope of this Chapter is twofold. A first goal, that we accomplish in the next Section, is to introduce the general setting of learning from examples. A second goal, that we deal with in Section 2.3, is to introduce tensors in the picture. We begin by motivating the use of tensorial representations in learning. Successively we recall basic definitions and tensor methods; these methods constitute the main building blocks used within the proposed algorithms in the next Chapters.

2.1 Foundations of Learning

Discussing a setting for (artificial) learning problems is not an obvious task since different paradigms are possible. For instance in [106] we find "the learning problem can be described as finding a general rule that explains data given only a sample of limited size". However, an objection is that this description suits the case of *inductive learning* only. Indeed in this case one is concerned with the general picture. Nonetheless, non-inductive inference is also possible. This approach is followed in *transductive* learning methods; one of such methods will be presented in the next Chapter. In our exposition we mainly follow [227] and give a (not formal) account of the theoretical foundations behind these alternative learning problems.

2.1.1 General Setting for Statistical Learning Problems

The setting of learning from examples comprises three components [227]:

1. A *generator* of input data. We shall assume that data can be represented as vectors[1] of $\mathbb{R}^D$. These vectors are independently and identically distributed (i.i.d.) according to a fixed but unknown probability distribution $p(x)$.
2. A *supervisor* that, given x, returns an output value y according to a conditional distribution $p(y|x)$ also fixed and unknown. Note that the supervisor might be or might not be present.
3. A *learning machine* (or *learning algorithm*) able to choose a function, or hypothesis, from a given set called *hypothesis space*.

The function chosen by the learning machine is denoted by $f(x;\theta)$ where θ is a parameter in a certain set Θ.

2.1.2 Supervised and Unsupervised Learning

When the supervisor is present the learning problem is called *supervised*. When it is not present, the learning problem is called *unsupervised*. In this latter case the aim is to find a concise representation of the data based upon training data $x_n, n \in \mathbb{N}_N$ produced by the generator. In contrast, the goal of supervised learning is to find an approximation of the supervisor response. The N training data are i.i.d. pairs $(x_n, y_n), n \in \mathbb{N}_N$ each of which is assumed to be drawn according to $p(x,y) = p(y|x)p(x)$.

Three common learning tasks are found within this categorization: *regression*, *classification* and *density estimation*.

Regression In *regression* the supervisor's response, as well as the output of $f(x;\theta)$, take values in the real numbers.

Classification In *classification* (a.k.a. *pattern recognition*) the supervisor's output takes values in the discrete finite set of possible labels $\mathcal{Y}$. The function chosen by the learning machine has also range $\mathcal{Y}$. In particular, in the *binary*

[1] Note that this case includes arrays of any orders. The latter can be envisioned as vectors with additional structure conveyed by the generator.

classification problem $\mathcal{Y}$ consists of two elements, typically $\mathcal{Y} = \{0, 1\}$. The function $f(x; \theta)$ is an indicator function.

Density Estimation This is an instance of unsupervised learning: there is no supervisor. The functional relation to be learned from examples is the probability density $p(x)$ (the generator).

2.1.3 Semi-supervised Learning and Transduction

Supervised and unsupervised learning are concerned with estimating a function over the whole input domain — say $\mathbb{R}^D$ — based upon a finite set of points. Therefore they are inductive approaches aiming at the general picture. There is yet another inductive approach that we shall mention, namely *semi-supervised learning*. In semi-supervised learning one has a set of labelled pairs

$$(x_1, y_1), \quad (x_2, y_2), \quad \cdots, \quad (x_N, y_N) \ , \tag{2.1}$$

as in supervised learning, as well as a set of unlabeled data

$$x_{N+1} \quad x_{N+2}, \quad \cdots, \quad x_{N+T} \ , \tag{2.2}$$

as in unsupervised learning. The purpose is the same as in supervised learning: to find an approximation of the supervisor response. However this goal is achieved by a learning algorithm that keeps into account the additional information coming from the unlabelled data. According to [45], semi-supervised learning was popularized for the first time in the mid-1970's although similar ideas appeared earlier. Alternative semi-supervised learning machines differ in the way they exploit the information from the unlabelled set. One popular idea is to assume that the (possibly high-dimensional) input data lie (roughly) on a low-dimensional manifold [23–25, 190].

In induction one seeks for the general picture with the purpose of making out-of-sample prediction. This is an ambitious goal that might be unmotivated in certain settings. What if all the (unlabeled) data are given in advance? Suppose that one is only interested in prediction at *finitely many* points. It is expected that this less ambitious task results in simple inference problems. This expectation is supported by existing theoretical insights, as we should mention in a moment. These ideas are reflected in the approach found in *transductive learning* formulations. As in semi-supervised learning in transductive learning one has training pairs (2.1) as well as test data (2.2). However, differently than in semi-supervised learning one is only interested in making predictions at the test data (2.2).

Remark. Notice that transduction might still go through finding a functional relationship. In other words operatively a transductive algorithm might still choose a function $f(x;\theta)$ where θ is a parameter in a certain set. However in this case $f(x;\theta)$ is only of interest for making predictions at the test data (2.2). Alternatively one might find directly the labels $y_{N+1}, y_{N+2}, \cdots, y_{N+T}$ corresponding to (2.2). Notice, however, that this latter approach results into mixed-integer problems; these problems are harder to solve than convex optimization programs solved in inductive learning [119].

2.1.4 Discriminative Versus Generative Methods

Traditionally the *perceptron* due to Rosenblatt [170] is considered as one of the first contributions to machine learning research. The idea was to provide mathematical models of the organization and the functioning of the brain. At the same time, these mathematical models revealed themselves useful to solve artificial learning tasks. It was the beginning of *Cybernetics* with its dream of developing an artificial intelligence.

The Emergence of a New Paradigm

The new paradigm was extremely powerful: the perceptron was successfully trained for the ten-class digit classification problem in $D = 400$ dimensions with only $N = 512$ training examples [226]. The same pattern recognition problem was tackled by Ronald Fisher with a conceptually different approach in the 1930s under the name of *discriminant analysis*[2]. The solution proposed by Fisher followed the classical methodology of parametric statistics. The main idea was:

1. Find the parameters of a *generative model*.
2. Construct the decision rule based upon the estimated parameters.

Following this strategy requires the estimation of about $0.5D^2$ parameters where D is the dimension of the space. The perceptron required only $D \approx N$. Assuming that estimating one parameter requires, roughly speaking, C examples, it is clear that for problems in high dimensions like the ten-class digit classification the perceptron is feasible whereas Fisher's method is not.

[2]This name is universally adopted although it is pruned to confusion: in fact, as discussed below, discriminant analysis is a *generative* method and not a *discriminative* approach, as the name might suggest.

The approach consisting of point 1 and 2 is the general methodology followed by *generative methods* (of induction). In contrast an approach based on minimizing different types of *empirical losses*, like the perceptron, were called *discriminative methods* (of induction). Within generative methods one finds a model based upon a notion of closeness to the *true function*. In contrast within the discriminative approach one uses a notion of closeness based on the accuracy of prediction. This is a notion of closeness based on *evaluation functionals* (see Appendix B).

Philosophical Implications

In the discriminative approach one gives up the ambitious aim of finding a model of how the generating mechanism works. The idea is to focus only on the goal of performing accurate predictions. As seen above in the inductive case one is interested in prediction at *any* point of the input domain. In transduction one only cares about a finite predefined set of sites. As pointed out by Vapnik [226] these different ideas are reflected by two corresponding views in the philosophy of science. Within the first view, called *realism*, scientific discoveries are regarded as a faithful representation of the real laws of nature. Within the second view, called *instrumentalism*, scientific discoveries are only seen as a way to make good predictions. The two views are not antithetic: instrumentalism is not specifically anti-realist. It only emphasizes that theories can offer explanations of how the world works. They should be regarded as approximations of the world, rather than an ultimate reality.

2.1.5 The SRM Principle for Induction and Transduction

Transductive and inductive inference share the common goal of achieving the lowest possible error on test data. In contrast with induction, transduction assumes that test data are given in advance and consist of a finite discrete set of patterns drawn from the same distribution as the training set. From this perspective, it is clear that both transductive and inductive learning are concerned with *generalization*. In turn, one of the most effective frameworks to study the problem of generalization is the *Structural Risk Minimization* (SRM) principle.

Expected and Empirical Risk

The starting point is the definition of a *loss* $l(y, f(x;\theta))$, or discrepancy, between the response y of the supervisor to a given input x and the response $f(x;\theta)$ of

the learning machine (that can be transductive or inductive[3]). The *expected risk* is defined as:

$$R(\theta) = \int l(y, f(x;\theta)) dP(x,y) \ . \tag{2.3}$$

From a mathematical perspective the goal of learning is the minimization of this quantity. However, $p(x,y)$ is unknown and one can rely only on the sample version of (2.4) namely the *empirical risk*:

$$R^N_{\text{emp}}(\theta) = \sum_{n \in \mathbb{N}_N} l(y_n, f(x_n, \theta)) \ . \tag{2.4}$$

A possible learning approach is based on *Empirical Risk Minimization* (ERM) and encompasses Maximum Likelihood (ML) inference [227]. It consists of finding:

$$\hat{\theta}_N := \arg\min_{\theta \in \Theta} R^N_{\text{emp}}(\theta) \tag{2.5}$$

where we denote by Θ the parameter space that defines the set of functions $\{f(x;\theta) \ : \ \theta \in \Theta\}$.

Definition 2.1.1. The ERM approach is said to be *consistent* if

$$\begin{aligned} R^N_{\text{emp}}(\hat{\theta}_N) &\xrightarrow{N} \min_{\theta\in\Theta} R(\theta) \\ R(\hat{\theta}_N) &\xrightarrow{N} \min_{\theta\in\Theta} R(\theta) \end{aligned}$$

where $\xrightarrow{N}$ denotes convergence in probability for $N \to \infty$.

In words: the ERM is consistent if, as the number of training patterns N increases, *both* the expected risk $R(\hat{\theta}_N)$ and the empirical risk $R^N_{\text{emp}}(\hat{\theta}_N)$ converge to the minimal possible risk $\min_{\theta\in\Theta} R(\theta)$. It was shown in [230] that the necessary and sufficient condition for consistency is that:

$$P\left\{\sup_{\theta\in\Theta} |R(\theta) - R^N_{\text{emp}}(\theta)| \geq \epsilon\right\} \xrightarrow{N} 0, \quad \forall \epsilon > 0 \ . \tag{2.6}$$

In turn, the necessary and sufficient condition for (2.6) to hold true were discovered in 1968 by Vapnik [228, 229] and are based on *capacity factors*.

Capacity Factors

Consistency is one of the main theoretical questions in Statistics. From a learning perspective, however, it does not address the most important aspect.

[3]Without loss of generality in our notation we imply that, as for an inductive algorithm, also the transductive learning machine goes through the estimation of a functional relation f, see the Remark at the end of Subsection 2.1.3.

The aspect that one should be mostly concerned with is how to control the generalization of a certain learning algorithm. Whereas consistency is an asymptotic result, we want to minimize the expected risk given that we have available only finitely many observations to train the learning algorithm. It turns out, however, that consistency is central to address also this aspect [227]. Additionally, a crucial role for answering this question is played by capacity factors that, roughly speaking, are all measures of how well the set of functions $\{f(x;\theta) \ : \ \theta \in \Theta\}$ can separate data. A more precise description is given in the following[4].

VC Entropy The first capacity factor[5] relates to the expected number of *equivalence classes*[6] according to which the training patterns factorize $\{f(x;\theta) \ : \ \theta \in \Theta\}$. We denote it by $\mathrm{En}(p,N)$ where the symbols emphasize the dependence of the VC Entropy on the underlying joint probability p and the number of training patterns N. The condition

$$\lim_N \frac{\mathrm{En}(p,N)}{N} = 0$$

forms the necessary and sufficient condition for (2.6) to hold true *with respect to the fixed probability density p.*

Growth Function It corresponds to the maximal number of *equivalence classes* with respect to all the possible training samples of cardinality N. As such, it is a distribution-independent version of the VC Entropy obtained via a worst-case approach. We denote it by $\mathrm{Gr}(N)$. The condition

$$\lim_N \frac{\ln \mathrm{Gr}(N)}{N} = 0$$

forms the necessary and sufficient condition for (2.6) to hold true *for all the probability densities p.*

VC dimension This is the cardinality of the largest set of points that the algorithm can shatter; we denote it by $\dim_{VC}$. Note that $\dim_{VC}$ is a property of $\{f(x;\theta) \ : \ \theta \in \Theta\}$ that does not depend neither on N nor on p. Roughly speaking it tells how flexible is the set of functions. A finite value of $\dim_{VC}$

[4]Precise definitions and formulas can be found in [227, Chapter 2].

[5]Here and below VC is used as an abbreviation for Vapnik-Chervonenkis.

[6]An equivalence class is a subset of $\{f(x;\theta) \ : \ \theta \in \Theta\}$ consisting of functions that attribute the same labels to the input pattern in the training set.

forms the necessary and sufficient condition for (2.6) to hold true *for all the probability densities* p.

The three capacities are related by the chain of inequalities [228, 229]:

$$\mathrm{En}(p,N) \leq \ln \mathrm{Gr}(N) \leq \dim_{VC} \left(\ln \frac{N}{\dim_{VC}} + 1 \right) . \tag{2.7}$$

VC Bounds

One of the key results of the theory developed by Vapnik and Chervonenkis is the following probabilistic bound. With probability $1-\eta$ simultaneously for all $\theta \in \Theta$ it holds that [227]:

$$R(\theta) \leq R^N_{\mathrm{emp}}(\theta) + \sqrt{\frac{\mathrm{En}(p,2N) - \ln \eta}{N}} . \tag{2.8}$$

Note that the latter depends on p. The result says that, for a fixed set of functions $\{f(x;\theta) \ : \ \theta \in \Theta\}$, one can pick that $\theta \in \Theta$ that minimizes $R^N_{\mathrm{emp}}(\theta)$ and obtain in this way the best guarantee on $R(\theta)$[7]. Now, taking into account (2.7) one can formulate the following bound based on the growth function:

$$R(\theta) \leq R^N_{\mathrm{emp}}(\theta) + \sqrt{\frac{\ln \mathrm{Gr}(2N) - \ln \eta}{N}} . \tag{2.9}$$

In the same way one has:

$$R(\theta) \leq R^N_{\mathrm{emp}}(\theta) + \sqrt{\frac{\dim_{VC} \left(\ln \frac{2N}{\dim_{VC}} + 1 \right) - \ln \eta}{N}} . \tag{2.10}$$

Note that both (2.9) and (2.10) are distribution-independent. Additionally (2.10) only depends upon the VC dimension (which, contrary to Gr, is independent from N). Unfortunately there is no free lunch: (2.9) is less tight than (2.8) and (2.10) is less tight than (2.9).

The Role of Transduction

It turns out that a key step in obtaining the bound (2.8) is based upon the *symmetrization lemma*:

$$P \left\{ \sup_{\theta} |R(\theta) - R^N_{\mathrm{emp}}(\theta)| \geq \epsilon \right\} \leq 2P \left\{ \sup_{\theta} |R^{N_1}_{\mathrm{emp}_1}(\theta) - R^{N_2}_{\mathrm{emp}_2}(\theta)| \geq \frac{\epsilon}{2} \right\} \tag{2.11}$$

[7] This is because the bound holds simultaneously for all $\theta \in \Theta$.

where $R^{N_1}_{\text{emp}_1}$ and $R^{N_2}_{\text{emp}_2}$ are constructed upon two different i.i.d. samples, precisely as in transduction. More specifically, (2.8) comes from upper-bounding the right hand-side of (2.11) [45, Chapter 22, "Transductive Inference and Semi-Supervised Learning"]. More generally it is apparent that for obtaining all bounds of this type the key element remains the symmetrization lemma [227]. Notably starting from the latters one can derive bounds explicitly designed for the transductive case where one of the two samples plays the role of the training set and the other of the test set. In light of this, Vapnik argues that transductive inference is a fundamental step in machine learning. Additionally, since the bounds for transduction are tighter than those for induction, the theory suggests that, whenever possible, transductive inference should be preferred over inductive inference. Practical algorithms can take advantage of this fact by implementing the *adaptive* version of the Structural Risk Minimization (SRM) principle that we discuss next.

Structural Risk Minimization Principle

The bounds seen above are hardly useful. In fact they are normally too loose to lead to practical model selection techniques. Although tighter bounds exist, the study of better bounds remains a challenge for future research. However, the theory does provide guiding lines for the implementation of algorithms. Indeed the structure of the bounds above suggests that one should minimize the empirical risk while controlling some measure of capacity. The idea behind the SRM, introduced in the 1970's, is to construct nested subsets of functions:

$$\mathscr{S}_1 \subset \mathscr{S}_1 \subset \cdots \subset \mathscr{S}_K = \mathscr{S} = \{f(x,\theta) \ : \ \theta \in \Theta\} \tag{2.12}$$

where each subset $\mathscr{S}_k$ has capacity h_k (VC entropy, Growth function or VC dimension) with $h_1 < h_2 < \cdots < h_K$. Then one chooses an element of the nested subsets so that the second term in the right hand side of the bounds is kept under control; within that subset one then picks that specific function that minimizes the empirical risk. As Vapnik points out [45, Chapter 22, "Transductive Inference and Semi-Supervised Learning"]:

"to find a good solution using a finite (limited) number of training examples one has to construct a (smart) structure which reflects prior knowledge about the problem of interest."

In practice one can use the information coming from the unlabeled data to define a smart structure to improve the learning. This opportunity is recognized as the key advantage of SRM for transductive inference over that for the inductive case. Existing techniques relying on this idea are found in [45].

Remark. In other words, the side information coming from unlabeled data can serve the purpose of devising an *adaptive* set of functions. Whenever available, however, one should use additional side information over the structure of the problem. Indeed using *informative representations* for the input data is also a way to construct smart set of functions. In fact, representing the data in a suitable form implies a mapping from the input space to a more convenient set of features. We will discuss this aspect more extensively in Section 2.3.1.

2.2 Learning through Regularization

So far we addressed the theory but we have not talked about how to practically implement it. It is understood that the essential idea of SRM is to find the best trade-off between the empirical risk and some measure of complexity of the hypothesis space. This ensures that the left hand side of VC bounds — the expected risk that we are interested in to achieve generalization — is minimized. In practice there are different ways to define the sets in the sequence (2.12). The generic set $\mathscr{S}_k$ could be the set of polynomials of degree k or a set of splines with k nodes. However, it is in connection to *regularization theory* that practical implementations of the SRM principle find their natural domain.

2.2.1 Tikhonov Theory

Regularization theory was introduced by Andrey Tikhonov [208–210] as a way to solve ill-posed problems. Ill-posed problems are problems that are not well-posed in the sense of Hadamard [98]. Consider solving in f a linear equation of the type[8]:

$$Af = b \ . \tag{2.13}$$

Even if a solution exists, it is often observed that a slight perturbation of the right hand side b causes large deviations in the solution f. Tikhonov proposed to solve this problem by minimizing a functional of the type:

$$\|Af - b\|^2 + \gamma\Omega(f)$$

where $\|\cdot\|$ is a suitable norm on the range of A, γ is some parameter and Ω is a regularization functional (sometimes called *stabilizer*). The theory of such an approach was developed by Tikhonov and Ivanov; in particular it was shown that there exists a strategy to choose γ depending on the accuracy of

[8] In the general case, f is an element of some Hilbert space, A is a compact operator and b is an element of its range.

b that asymptotically leads to the desired solution $f^\star$. This was shown under the assumption that there exists $c^\star$ such that $f^\star \in \{f \ : \ \Omega(f) \leq c^\star\}$.

In general, the theory of Tikhonov regularization differs from Statistical Learning Theory in a number of ways. To begin with Tikhonov regularization considers specific structures in the nested sequence (2.12) (depending on the way Ω is defined); secondly it requires the solution to be in the hypothesis space; finally the theory developed by Tikhonov and Ivanov was not concerned with guarantees for a finite number of observations.

In the simplest case, where f lives in a Euclidean space and $\Omega(f) = \|f\|^2$, Tikhonov regularization is known as *Ridge Regression* [109,138]. When f is an element of a Reproducing Kernel Hilbert Space (RKHS) $\mathscr{H}$ (see Appendix B), the theory is best known through the work of Wahba [124,235].

2.2.2 SRM and Regularization in RKHSs

The setting of a RKHS provides a convenient way to define the sequence (2.12). In fact, a nested sequence can be constructed by bounding the norm in $\mathscr{H}$:

$$\mathscr{S}_k = \{f \in \mathscr{H} \ : \ \|f\| \leq a_k\} \ . \tag{2.14}$$

It turns out that there is a measure of capacity of $\mathscr{S}_k$ which is an increasing function of a_k [77]. This capacity measure can be used to derive probabilistic bounds in line with (2.8), (2.9) and (2.10). In practice, instead of solving the *constrained problem*:

$$\begin{array}{ll} \min_{f \in \mathscr{H}} & R^N_{\text{emp}}(f) \\ \text{subject to} & \|f\| \leq a_k \end{array} \tag{2.15}$$

for any k, one normally solves the provably equivalent *penalized problem*:

$$\min_{f \in \mathscr{H}} R^N_{\text{emp}}(f) + \lambda_k \|f\|^2 \tag{2.16}$$

and pick the optimal λ_k appropriately. As we mentioned already, existing bounds do not usually provide practical recipes for the choice of the optimal λ_k; therefore one usually resorts to cross-validation or similar strategies.

Note that in (2.15) and (2.16) we wrote $R^N_{\text{emp}}(f)$ instead of $R^N_{\text{emp}}(\theta)$, as before. In fact, in this case the solution of the learning problem is found by formulating a *convex variational problem* where f itself plays the role of the optimization variable θ. In practice, however, the *representer theorem* [124,176] shows that a representation of the optimal f only depends upon an expansion of kernel functions centered at the training patterns. More specifically it holds that $f(x) = \sum_{n \in \mathbb{N}_N} \alpha_n k(x_n, x)$ where $\alpha \in \mathbb{R}^N$ is found solving a finite dimensional

optimization problem. The latter is convex provided that l is a convex loss function. An alternative derivation is based on a primal dual argument and can be obtained based on known results of convex variational analysis, see Appendix C.

The great majority of kernel methods for supervised learning minimize a *penalized empirical risk* functional in line with (2.16). They are either based on the representer theorem [77, 78, 176], or consist of primal-dual formulations with emphasis on the geometrical aspects of the learning problem [26, 33, 57, 74, 199, 236].

2.2.3 Alternative forms of Regularization

The penalized empirical risk minimization problem was introduced in (2.16) in the setting of RKHS of functions. However it shall be noted that it is a very general idea. Ultimately this can be related to the generality of equation (2.13). The latter can either refer to infinite dimensional problems or to a finite system of linear equations involving objects living in some finite dimensional space. Therefore for the sake of generality one can restate (2.16) as

$$\min_{\theta\in\Theta} R^N_{\text{emp}}(\theta) + \lambda\|\theta\|^2 \tag{2.17}$$

where $(\Theta, \langle\cdot,\cdot\rangle)$ is some Hilbert Space (HS) with inner product $\langle\cdot,\cdot\rangle$ (see Appendix A), R^N_{emp} is the empirical risk and $\|\theta\|$ is the *Hilbertian norm* defined upon the space's inner product

$$\|\theta\| = \sqrt{\langle\theta,\theta\rangle}\ . \tag{2.18}$$

This format drives the attention away from the nature of the unknown (which is just an abstract vector in some inner-product space). Problem (2.17) comprises the case where θ is a function, a vector, a matrix, a higher-order tensor or even an abstract operator between Hilbert spaces, as we shall see in the following. The notation emphasizes that there is no real distinction between *parametric* and *nonparametric* methods (although we retain this nomenclature for convenience, see also the Remark at the end of Section 2.3.3).

The LASSO and non-Hilbertian Norms

There is a number of ways according to which problem (2.17) has been further generalized in the last decade. In particular, a lot of effort has been put on the design of alternative penalties. This arises from the realization that using a certain penalty is also a way to convey prior knowledge. This

fact is best understood within a Bayesian setting, in light of a *Maximum a Posteriori* (MAP) interpretation of (2.17), see, e.g., [77]. In particular $\|\cdot\|$ has been replaced with various type of *non-Hilbertian norms*. These are norms that, contrary to (2.18), do not arise from inner products. The LASSO (Least Absolute Shrinkage and Selection Operator, [206]) is perhaps the most prominent example of such cases. In the LASSO one considers linear functions

$$f(x;\theta) = \langle x, \theta \rangle = \sum_{d \in \mathbb{N}_D} x_d \theta_d$$

and uses the the l_1 norm $\|\theta\|_1 = \sum_{i \in \mathbb{N}_I} |\theta_i|$ to promote the sparsity of the parameter vector θ. Note that this corresponds to define the structure (2.12) according to:

$$\mathscr{S}_k = \{ f(x;\theta) \ : \ \|\theta\|_1 \leq a_k \} \ . \tag{2.19}$$

Like Ridge Regression, the LASSO is a continuous shrinkage method that achieves good prediction performance via a bias-variance trade-off. Since usually the estimated coefficient vector has many entries equal to zero, the approach has the further advantage over Ridge Regression of giving rise to interpretable models.

More recently, different structure-inducing penalties have been proposed as a promising alternative, see [118, 237, 239]. The general idea is to convey structural assumption on the problem, such as grouping or hierarchies over the set of input variables, by suitably crafting the penalty. In this way the users are permitted to customize the regularization approach according to their subjective knowledge on the task. Correspondingly, as in (2.19), one (implicitly) forms a smart structure of nested subsets of functions, in agreement with the SRM principle.

These ideas have been generalized to the case where Θ is infinite dimensional, in particular in the framework of the Multiple Kernel Learning (MKL) problem. This was investigated both from a functional viewpoint [142,143], and from the more pragmatic point of view of optimization [12,13,128].

Spectral Regularization and Multi-task Learning

Yet another generalization of (2.17) arises in the context of *Multi-task learning* [20, 42, 205]. In this setting one approaches simultaneously different learning tasks under some common constraint(s). The general idea, sometimes also known as *collaborative filtering*, is that one can take advantage of shared features across tasks. In practical applications it was shown that one can significantly gain in terms of generalization performance from exploiting such prior knowledge. From the point of view of learning through regularization, a

sensible approach is given in [6, 7]. Suppose one has T datasets, one for each task; the $t-$th dataset has N_t observations. Note that, in general, $N_1 \neq N_2 \neq \cdots \neq N_T$. In this setting one has to learn vectors[9] $\theta_t,\ t = 1, 2, \ldots, T$, one per task. The idea translates into penalized empirical risk minimization problems of the type:

$$\min_{\theta=[\theta_1,\theta_2,\cdots,\theta_T]\in\Theta} \sum_{t\in\mathbb{N}_T} R^{N_t}_{\text{emp}}(\theta_t) + \lambda\|\theta\|_* \tag{2.20}$$

where $\|\theta\|_*$ is the *nuclear norm*:

$$\|\theta\|_* = \sum_{r\in\mathbb{R}_R} s_r(\theta) \tag{2.21}$$

and $s_1(\theta),\ s_2(\theta), \cdots, s_R(\theta)$ are the $R \leq \min(T, D)$ non-zero singular values of the $D \times T$ matrix[10] θ. The nuclear norm is the convex envelope of the rank function on the spectral-norm unit ball [81]; roughly speaking, it represents the best convex relaxation of the rank function. Its use in (2.20) is motivated by the assumption that the parameter vectors of related tasks should be approximately linear dependent. More generally, one can consider *spectral penalties* [2,8] that include the nuclear norm as a special case. We will be back to spectral penalties in Chapter 3 where we extend this concept to higher order arrays.

2.3 Aspects of Learning with Tensors

2.3.1 The Need for Informative Data Representations

Discriminative methods have been shown to outperform generative methods in many applications; after all, as we discussed in the previous Section, they are designed to accomplish their task in a much more direct way. Their performance is superior in particular for those cases where input data are naturally embedded in real vector spaces. This is because, usually, discriminative methods rely on Euclidean metrics.

Embedding of Structured Data

The situation is less clear in those domains where input patterns carry on additional structure, for instance inherited by the dynamical nature of the

[9] As before we use vectors in a generalized sense. A vector can be a function as well as an element of $\mathbb{R}^D$ for some finite natural number D.

[10] Note that (2.21) corresponds to the l_1 norm of the vector of singular values. The definition remains valid also in the infinite dimensional case under some regularity assumptions, see [70].

data. As a concrete example, suppose one collects $D-$variate measurements from some industrial plant at T successive time instants; the purpose is to discriminate between different working modes. Note that each measurement is in this case a $T \times D$ matrix. One could assume that, around the working modes, the dynamics is approximately linear. In turn, linear models — which are determined up to a change of basis — are elements of the Stiefel manifold. It is expected that neglecting this structure by treating each $T \times D$ matrix simply as a vector with dimension TD leads to suboptimal learning strategies. In fact, one observes that the learning rates of algorithms that follow this approach are considerably worse than those observed when the structure is kept into account. In general, it is known to be of paramount importance to keep characteristic properties of the data into account; in particular, one should use an appropriate input representation especially when few observations are available. In this respect Poggio and Smale write [160]:

"One could argue that the most important aspect of intelligence and of the amazing performance of real brains is the ability to learn. How then do the learning machines we have described in the theory compare with brains? There are of course many aspects of biological learning that are not captured by the theory and several difficulties in making any comparison. One of the most obvious differences, however, is the ability of people and animals to learn from very few examples. The algorithms we have described can learn an object recognition task from a few thousand labeled images. This is a small number compared with the apparent dimensionality of the problem (thousands of pixels) but a child, or even a monkey, can learn the same task from just a few examples. Of course, evolution has probably done a part of the learning but so have we, when we choose for any given task an appropriate input representation for our learning machine. From this point of view, as Donald Geman has argued, the interesting limit is not *m goes to infinity*, but rather *m goes to zero*."

It should be noted that certain classes of neural networks are universal approximators [112]. Likewise, the RBF-Gaussian kernel is *universal*: kernel methods that use such a kernel can approximate any smooth function [196]. This means that, no matter the structure of the input data, when the number of training points N approaches infinity the best possible performance is achieved. Therefore there is no need to keep into account the specific nature of the data when one has enough observations. However, there is a number of areas where data are intrinsically scarce. This is the case for instance in biosignal processing or chemometrics. Additionally one often wants to process information coming from *rare* events (earthquakes, economical crisis etc.). In this cases to achieve generalization one needs to convey as much information as possible concerning the nature of the phenomena of interest.

Why Tensors?

The algorithms that we discuss in this Thesis make use of *tensorial representations* that is, they represent input data as multidimensional arrays; the structure of these objects is taken into account to improve generalization. The choice of tensors is motivated as follows. On the one hand many objects of interest, such as videos and colored images, admit a natural tensorial representation. Additionally, tensor representations naturally result from the experiments performed in a number of fields. Table 2.1 gives some examples. On the other hand there are cases where one can explicitly carry on tensor

Table 2.1: Some examples of tensorial representations in real-life applications

neuroscience:	EEG data (time × frequency × electrodes) fMRI data (time × x − axis × y − axis × z − axis)
vision:	image (/video) recognition (pixel × illumination × expression × ⋯)
chemistry:	fluorescence excitation-emission data (samples × emission × excitation)

transformations with the purpose of exploiting the spectral content of these new representations. This is the case, for instance, of the cumulant tensors that we consider in Chapter 5.

Note that in real-world applications one is more and more faced with the task of integrating different sources of information. In many applications involving multimedia data, for instance, it is of interest to integrate different modalities such as acoustic and visual content of video [139]. Likewise, it is desirable to combine different views in diagnostic systems [61]. Due to their multi-way nature tensors are naturally suited for this. Additionally it is often the case that input data are graphs. Consider the situation where one wants to test the activity of chemical compounds against an illness. A natural way to represent the chemical structure of compounds is by means of graphs. In turn, as mentioned earlier, graphs and dynamical systems are strongly connected. Therefore it is arguable that one could conveniently use tensorial representations also in these cases.

Next we recall some basic facts about tensors and introduce the tools for tensors that we use in the next Chapters.

2.3.2 Tensors and Higher-order Data Representations

Basic Facts about Tensors

We denote both scalars and vectors as lower case letters $(a, b, c, \ldots)$ and matrices as bold-face capitals $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \ldots)$. We write 1_N to denote $[1, 1, \ldots, 1] \in \mathbb{R}^N$ and $\boldsymbol{I}_N$ to indicate the $N \times N$ identity matrix. We also use subscript lower-case letters i, j in the meaning of indices and we will use I, J to denote the index upper bounds. Additionally we write $\mathbb{N}_I$ to denote the set $\{1, \ldots, I\}$. We recall that $N-$th order tensors, which we denote by calligraphic letters $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots)$, are higher-order generalizations of vectors (first order tensors) and matrices (second order tensors). More generally, the order N of a tensor is the number of dimensions, also known as ways or modes. We write $a_{i_1,\ldots,i_N}$ to denote the entry $(\mathcal{A})_{i_1,\ldots,i_N}$. Likewise we write a_i to mean $(a)_i$ and a_{ij} to mean $(\boldsymbol{A})_{ij}$.

An $N-$th order tensor $\mathcal{A}$ is rank-1 if it consists of the outer product of N nonzero vectors $u^{(1)} \in \mathbb{R}^{I_1},\ u^{(2)} \in \mathbb{R}^{I_2}, \ldots,\ u^{(N)} \in \mathbb{R}^{I_N}$, that is, if

$$a_{i_1 i_2 \ldots i_N} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(N)}_{i_N}$$

for all values of the indices. In this case we write

$$\mathcal{A} = u^{(1)} \otimes u^{(2)} \otimes \cdots \otimes u^{(N)} \ . \tag{2.22}$$

The linear span of such elements forms a vector space, which once endowed with the inner product

$$\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} a_{i_1 i_2 \cdots i_N} b_{i_1 i_2 \cdots i_N}, \tag{2.23}$$

is denoted by $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$.

Remark. We note at this point that $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ can be regarded as a reproducing kernel Hilbert space associated to a certain product kernel. This fact shows a link between tensors and kernels; it addresses the recent interest shown for the interplay between these two seemingly separate mathematical concepts [1]. The precise argument requires to first introduce notions of kernels; for this reason it is reported in Appendix B.4.

This link also gives us the opportunity to motivate the use of the symbol $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ in place of the more compact $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ which generalizes the

common notation $\mathbb{R}^{I_1 \times I_2}$ used for matrices. We adopt this convention, common in multilinear algebra [95], because it emphasizes that $\mathbb{R}^{I_i}$ plays the role of a *factor space* and it makes explicit the fact that the very same construction holds for more general spaces. For instance in Section 4.1 we will denote by

$$\mathscr{H}_1 \otimes \mathscr{H}_2 \tag{2.24}$$

the space of Hilbert-Schmidt operators of the type $A : \mathscr{H}_1 \to \mathscr{H}_2$ where $\mathscr{H}_1$ and $\mathscr{H}_2$ are HSs. This space is a generalization of the space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2}$ of $I_1 \times I_2$ matrices; the latter represent linear operators from $\mathbb{R}^{I_2}$ to $\mathbb{R}^{I_1}$. However if we were to write $\mathbb{R}^{I_1 \times I_2}$ the relation with (2.24) would not be apparent.

The Hilbert-Frobenius norm corresponding to (2.23) is

$$\|\mathcal{A}\| := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} \ . \tag{2.25}$$

Remark. We use $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ for any $N \geq 1$, regardless of the specific tuple $(I_1, I_2, \ldots, I_N)$.

Definition 2.3.1 ($n-$mode vector)**.** An *$n-$mode vector* of $\mathcal{A} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ is an element of $\mathbb{R}^{I_n}$ obtained from $\mathcal{A}$ by varying the index i_n and keeping the other indices fixed.

Definition 2.3.2 ($n-$rank)**.** The *$n-$rank* of $\mathcal{A}$, indicated by $\text{rank}_n(\mathcal{A})$, is the dimension of the space spanned by the $n-$mode vectors.

A tensor for which $r_n = \text{rank}_n(\mathcal{A})$ for $n \in \mathbb{N}_N$ is called a rank-$(r_1, r_2, \ldots, r_N)$ tensor; the $N-$tuple $(r_1, r_2, \ldots, r_N)$ is called the *multilinear rank* of $\mathcal{A}$.

Remark. For higher-order tensors an alternative notion of rank exists. This reads

$$\text{rank}(\mathcal{A}) := \arg\min \Big\{ R \in \mathbb{N} \ : \ \mathcal{A} = \sum_{r \in \mathbb{N}_R} u_r^{(1)} \otimes u_r^{(2)} \otimes \cdots \otimes u_r^{(N)} \ : \\ u_r^{(n)} \in \mathbb{R}^{I_n} \ \forall \ r \in \mathbb{N}_R, \ n \in \mathbb{N}_N \Big\} \tag{2.26}$$

and lies at the heart of the canonical polyadic decomposition (CPD) (a.k.a. PARAFAC or CANDECOMP) [41, 66, 103, 108]. However, for reasons that we will clarify later the methodologies that we discuss in this thesis do not rely on this notion of rank.

Whereas for second order tensors

$$\text{rank}_1(\mathcal{A}) = \text{rank}_2(\mathcal{A}) = \text{rank}(\mathcal{A})$$

for the general case we can only establish that $\text{rank}_n(\mathcal{A}) \leq \text{rank}(\mathcal{A})$ for any $n \in \mathbb{N}_N$. Additionally the $n-$ranks differ from each other in the general $N-$th order case.

Definition 2.3.3 ($n-$mode unfolding)**.** Let $\mathcal{A} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ and set

$$J := \prod_{j \in \mathbb{N}_N \setminus \{n\}} I_j \ . \tag{2.27}$$

The *$n-$mode unfolding* (also called matricization or flattening) of $\mathcal{A}$ is the matrix $\mathcal{A}_{\langle n \rangle} \in \mathbb{R}^{I_n} \otimes \mathbb{R}^J$ whose columns are the $n-$mode vectors.

Note that we have not specified the ordering according to which the vectors are arranged to form $\mathcal{A}_{\langle n \rangle}$. In fact this aspect does not matter for our purposes; it suffices that one sticks to a chosen ordering rule.

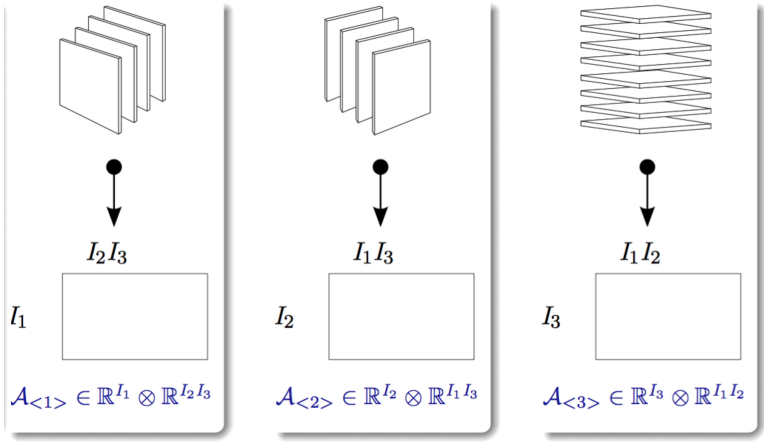


Figure 2.1: An illustration of the matrix unfoldings of a $\mathcal{A} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \mathbb{R}^{I_3}$.

Remark. Assume a second order tensor $\boldsymbol{A} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2}$. Then if the 2−mode unfolding is defined upon the lexicographic ordering, we have

$$\boldsymbol{A}_{\langle 2 \rangle} = \boldsymbol{A}^\top$$

where $\cdot^\top$ denotes matrix transposition.

In our setting the use of unfoldings is motivated by the elementary fact that[11]

$$\text{rank}_n(\mathcal{A}) = \text{rank}(\mathcal{A}_{\langle n \rangle}) \ . \tag{2.28}$$

Note that the $n-$mode unfolding as introduced above defines a linear operator that we explicitly introduce in the following, together with its *adjoint*[12].

Definition 2.3.4 ($n-$mode matricization/tensorization operators)**.** We call $n-$mode matricization operator the linear operator

$$\begin{array}{rccc} \cdot_{\langle n \rangle} : & \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N} & \to & \mathbb{R}^{I_n} \otimes \mathbb{R}^{J} \\ & \mathscr{A} & \mapsto & \mathscr{A}_{\langle n \rangle} \end{array}$$

where J is as in (2.27). The $n-$mode tensorization operator, denoted as $\cdot^{\langle n \rangle}$, is defined as its unique adjoint $\cdot^{\langle n \rangle} : \mathbb{R}^{I_n} \otimes \mathbb{R}^{J} \to \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ satisfying

$$\langle \boldsymbol{A}^{\langle n \rangle}, \mathcal{B} \rangle = \langle \boldsymbol{A}, \mathcal{B}_{\langle n \rangle} \rangle \ .$$

Remark. Note that, although seemingly abstract, this characterization through operators serves a very practical purpose. In fact, it allows one to rely on the chain rule for subdifferentials [76] that we extensively exploit in the next Chapter. In general we find that tools from functional analysis are very useful to deal with tensors. Notably the very same tools are used for symbolic calculus within modern optimization toolboxes [22].

Definition 2.3.5 ($n-$mode product)**.** The *$n-$mode product* of a tensor $\mathcal{A} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_{n-1}} \otimes \mathbb{R}^{I_n} \times \mathbb{R}^{I_{n+1}} \otimes \cdots \otimes \mathbb{R}^{I_N}$ by a matrix $\boldsymbol{U} \in \mathbb{R}^{J_n \times I_n}$, denoted by $\mathcal{A} \times_n \boldsymbol{U}$, is that element of $\mathbb{R}^{I}_1 \otimes \mathbb{R}^{I}_2 \otimes \cdots \otimes \mathbb{R}^{I_{n-1}} \otimes \mathbb{R}^{J_n} \times \mathbb{R}^{I_{n+1}} \otimes \cdots \otimes \mathbb{R}^{I_N}$ defined by

$$\mathcal{A} \times_n \boldsymbol{U} := (\boldsymbol{U} \mathcal{A}_{\langle n \rangle})^{\langle n \rangle} \ .$$

Multilinear Singular Value Decomposition

In [67,216,217] the following generalization of the singular value decomposition (SVD) for matrices was developed. It was termed in [67] higher-order singular value decomposition in view of the many analogies with the matrix SVD. Nowadays the name multilinear singular value decomposition (MLSVD) seems to be more appropriate. It is not the only possible generalization though: see [67, Section 7] for alternative definitions.

[11] We mentioned above that the specific ordering according to which the vectors are arranged to form $\mathcal{A}_{\langle n \rangle}$ does not matter for us. This is because the right hand side of (2.28) is invariant to permutations of the columns of $\mathcal{A}_{\langle n \rangle}$.

[12] See Appendix A.4 for a definition of Adjoint.

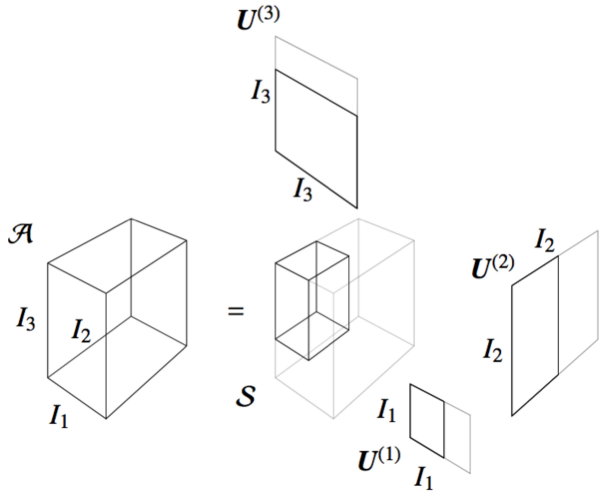


Figure 2.2: An illustration of the truncated MLSVD

Theorem 2.3.1 (MLSVD [67] [13]). *Any* $\mathcal{A} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ *can be written as the product*

$$\mathcal{A} = \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \cdots \times_N \boldsymbol{U}^{(N)}$$

in which:

- $\mathbb{R}^{I_n} \otimes \mathbb{R}^{I_n} \ni \boldsymbol{U}^{(n)} = \left[U_1^{(n)}\, U_2^{(n)}\, \cdots\, U_{I_n}^{(n)}\right]$ *is an orthogonal matrix*
- *the* core tensor $\mathcal{S} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ *is such that the* $(N-1)-$*order subtensor* $\mathcal{S}^{i_n=\theta}$ *defined by*

$$s^{i_n=\theta}_{i_1 i_2 \cdots i_{n-1} i_{n+1} \cdots i_N} := s_{i_1 i_2 \cdots i_{n-1}\, \theta\, i_{n+1} \cdots i_N}$$

  satisfies:

  1. all-orthogonality*:*

$$\langle \mathcal{S}^{i_n=\theta}, \mathcal{S}^{i_n=\beta} \rangle = 0, \ \forall \theta \neq \beta, \theta, \beta \in \mathbb{N}_n, \ n \in \mathbb{N}_N$$

  2. ordering*:*

$$\|\mathcal{S}^{i_n=1}\|_F \geq \|\mathcal{S}^{i_n=2}\|_F \geq \cdots \geq \|\mathcal{S}^{i_n=I_N}\|_F \geq 0 \text{ for any } n \in \mathbb{N}_N.$$

[13] In the present setting we state the theorem for the real valued case whereas the MLSVD theorem in [67] refers to the more general complex valued case.

The Hilbert-Frobenius norms $\|\mathcal{S}^{i_n=i}\|_F$*, symbolized by* $\sigma_i^{(n)}$*, are the* $n-$*mode singular values of* $\mathcal{A}$ *and the vector* $U_i^{(n)}$ *is the* $i-th$ $n-$*mode singular vector.*

In practice the matrix $\boldsymbol{U}^{(n)}$ of $n-$mode singular vectors can directly be found from the left singular vectors of the $n-$mode unfolding $\mathcal{A}_{\langle n \rangle}$ with SVD

$$\mathcal{A}_{\langle n \rangle} = \boldsymbol{U}^{(n)} \boldsymbol{S}^{(n)} \boldsymbol{V}^{(n)\top} \ . \tag{2.29}$$

The core tensor $\mathcal{S}$ then satisfies:

$$\mathcal{S} = \mathcal{A} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \cdots \times_N \boldsymbol{U}^{(N)\top} \ .$$

It is well known for matrices that the best rank-R approximation in least squares sense can be readily obtained from the truncated SVD. Although useful, the MLSVD is not as powerful: for general tensors it does not deliver the best low rank approximation, no matter what concept of rank is considered.

2.3.3 Parametric and Non-parametric Tensor-based Models

We can now turn to the problem of interest, namely *learning with tensors*. In the following Chapters we will assume that the generic input observation is represented as a tensor $\mathcal{X} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M}$ that we call *input data tensor*. We will focus on transductive and inductive learning algorithms that are based on a model of the type $f(\mathcal{X}; \theta)$ where θ is a parameter in a certain set Θ. In particular we can distinguish two situations depending on the way f is parametrized. The first case, which we encounter in the next Chapter, is the situation where $\theta = \{\mathcal{W}, b\}$. Note that $\mathcal{W}$ is also an element of $\mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M}$ while $b \in \mathbb{R}$. More specifically we focus on *affine* models of the type:

$$f(\mathcal{X}; \mathcal{W}, b) = \langle \mathcal{X}, \mathcal{W} \rangle + b \ . \tag{2.30}$$

By definition of $\langle \cdot, \cdot \rangle$ (see (B.5)) we have

$$\langle \mathcal{X}, \mathcal{W} \rangle = \mathrm{vec}(\mathcal{X})^\top \mathrm{vec}(\mathcal{W}) \tag{2.31}$$

where vec : $\mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \rightarrow \mathbb{R}^{D_1 D_2 \cdots D_M}$ is the *vectorization operator* mapping entries of an $M-$th order tensor $\mathcal{X}$ into entries of a $(D_1 D_2 \cdots D_M)-$dimensional vector according to some predefined ordering. This seems in contrast with what we discussed earlier concerning Euclidean embeddings. However, due to the specific regularization approach that we will use in learning the model's parameters, the approach does not neglect the structure of the input space. In turn, the linearity of (2.31) ensures interpretability, at least in those cases where one can explicitly relate entries of

the parameter space to the process under study. This is the case, for instance, for affine models of fMRI scans. In this case entries within a certain region are associated to voxels and therefore refer to a specific area of the brain.

Still, the class of models (4.1) is inadequate to represent many situations of interest. In particular it is often necessary to enforce a notion of *locality*. For these situations kernel methods seem the most appropriate choice. However, as we discuss in Chapter 4, a naive kernel approach again neglects the structural information conveyed by tensorial representations. Specialized kernels exist for certain classes of structured data [87, 88]. However, no existing approach exploits the (algebraic) structure of tensorial representations. To overcome this limitation in Chapter 4 we will propose to define *nonparametric* tensor-based models upon a novel class of tensorial kernels.

Remark. Here we follow the traditional nomenclature used within Statistics and related disciplines. In this spirit we call models of the type (4.1) *parametric*. Note that the term nonparametric is not meant to imply that such models lack parameters; it only means that the nature of the parameters implies the greater flexibility of such models. As we pointed out in Subsection 2.2.2, from an optimization perspective, the parameter to be learned in nonparametric methods is a function in a certain Hilbert Space (HS).

2.4 Chapter Summary

One of the most effective frameworks for statistical learning problems is represented by the SRM principle. The latter prescribes to define a (smart) structure of nested subsets of functions of increasing complexity. It is understood that a major role in the success of both transductive and inductive learning algorithms is played by the design of such structure; the latter should reflect prior knowledge on the problem at hand. A practical way of implementing this principle is represented by regularization schemes. In kernel methods, in particular, one tackles the learning task by solving a penalized empirical risk minimization problem; in this problem the RKHS norm is used as a way to penalize complex functions. More generally, one can convey structural information by suitably crafting the penalty term. This is the case for problems like the LASSO, that use the l_1 norm as a penalty function to impose sparsity. Other problem formulations use more involved regularization schemes that leverage common features between different learning tasks. This is the case, in particular, for the nuclear norm penalty used within multi-task learning. An alternative approach to convey prior knowledge consists of using structure preserving input representations. This gives to learning algorithm the ability to learn from small samples, which constitutes one of the key aspects

of (human) intelligence. Tensors naturally arise from a wide class of real-life applications; they allow one to capture the multi-modal structure of many types of data. A number of tools can be used to deal with them; in particular, the MLSVD represent a possible generalization to tensors of the matrix SVD. The use of tensors to represent input data give rise to two classes of models: affine models, useful to ensure a-posteriori interpretability; non-parametric tensors-based models that allow one to overcome the lack of flexibility of the aforementioned class.

Chapter 3

Spectral Learning based on Parametric Models

In this Chapter we present a general framework for learning based on convex optimization. The proposed approach allows one to deal appropriately with data that have a natural representation as higher-order arrays.

We begin by presenting a unifying class of convex optimization problems for which we present new, provably convergent and scalable algorithms. We then specialize this class of problems to perform single as well as multi-task learning both in a transductive as well as in an inductive setting. To this end we develop new tools extending the existing concept of *spectral regularization* for matrices (see Subsection (2.2.3)) to higher order tensors. For transduction, as well as for induction, the proposed technique leverages precise low multilinear rank assumptions over unknown multidimensional arrays and connects to the MLSVD. As a by-product of using a tensor-based formalism, our approach allows one to tackle the multi-task case in a natural way.

In the transductive case one has an input data tensor with missing features and, possibly, a partially observed matrix of labels. The goal is to both infer the missing entries in the data tensors as well as predict the missing labels. The essential idea consists of regularizing the collection of input features and labels directly, *without learning a model*. This approach is in contrast with the second family of problems that we consider, within the setting of inductive learning. In this case we explicitly estimate a parametric model of the type (2.31) for each learning task. This is needed to perform out of sample prediction. Each training pair consists of an input data tensor and a vector of labels that

corresponds to related but distinct tasks. This setting extends the standard penalized empirical risk minimization problem to allow for both multiple tasks and higher-order observational data. The taxonomy of learning problems we consider is reported in Table 3.1.

Table 3.1: A schematic representation of the learning problems that we formulate within this Chapter

transductive learning with tensors

soft-completion (tensor-sc)

data:	partially specified input data tensor and matrix of target labels
output:	latent features and missing labels

hard-completion (tensor-hc)

data:	partially specified input data tensor
output:	missing input data

inductive learning with tensors

data:	pairs of fully specified input features and vectors of target labels
output:	models (2.31) for out-of-sample evaluations of multiple tasks

The following characteristics of the novel tensor-based formulations can be deduced from the experimental evidence reported at the end of the Chapter:

- tensor-sc generally performs comparably or better to matrix-sc (its matrix analogue applied to the matrix obtained unfolding the data tensor along a fixed mode) in terms of misclassification errors; however the experiments show that tensor-sc leads to more favorable results for the reconstruction of the underlying input data tensor.

- The experimental evidence suggests that tensor-hc performs better than matrix-hc (performed along the third mode) when either no n-rank dominates the others or when the 3-rank is higher. This observation holds across different noise levels and fractions of entries used in the reconstruction.
- The computational complexity of tensor-hc is roughly M times that of matrix-hc (assuming the latter is applied on a fixed unfolding of the original tensor) where M is the order of the tensor.
- Our experiments on spectral data show that, when the number of spectral bands is limited, using tensor-hc, thereby exploiting the structure of all the modes, improves the imputation of missing entries. In contrast, if the number of bands is sufficiently high, the experiments show that matrix completion of the unfolded data tensor is more advantageous. Contrary to alternative approaches, tensor completion exploits redundancy along all the modes of spectral data. When no correlation along the spectral mode is present (Table 3.8) tensor completion performs worse than matrix completion.
- For inductive learning, the experiments show that exploiting relations between tasks significantly improves results over the case where learning tasks are approached independently from each other. Additionally our experiments reveal that exploiting structural assumptions over the input features is a good idea; this is seen to be the case especially when the number of tasks is small or the features dimensionality is higher.

3.1 General Problem Setting

The learning problems that we consider in this Chapter can be regarded as special instances of the penalized empirical risk minimization approach discussed in Section 2.2.3. From the optimization viewpoint they can all be cast in terms of a convex optimization problem of the type[1]:

$$\begin{array}{ll} \underset{w \in \mathscr{W}}{\text{minimize}} & \bar{J}(w) := \bar{f}(w) + \bar{g}(w) \\ \text{subject to} & w \in \bar{\mathscr{C}} \ . \end{array} \tag{3.1}$$

In this problem $\mathscr{W}$ is an *abstract vector space*, the nature of which we should clarify in a moment; $\bar{f}$ plays the role of an error functional which is convex and

[1] As we shall clarify by examples in Subsection 3.1.2, (3.1) does not represent, directly, the problem of interest; the latter will be stated in terms of the error functional f and regularization g (in contrast with $\bar{f}$ and $\bar{g}$ used in (3.1)).

differentiable; it is assumed that $\nabla \bar{f}$ is $L_{\bar{f}}$-Lipschitz, namely that:

$$\|\nabla \bar{f}(w) - \nabla \bar{f}(v)\|_{\mathscr{W}} \leq L_{\bar{f}} \|w - v\|_{\mathscr{W}} \quad \forall\, w, v \in \mathscr{W}; \tag{3.2}$$

$\bar{g}$ is a convex but possibly non-differentiable functional playing the role of a penalty. Finally $\bar{\mathscr{C}} \subseteq \mathscr{W}$ is a set which is non-empty, closed and convex.

3.1.1 Abstract Vector Spaces

We shall consider optimization problems on abstract finite dimensional inner product spaces that represent a generalization of $\mathbb{R}^P$, see, e.g., [136]. We are especially interested in the case where such an abstract space, denoted by $\mathscr{W}$, is obtained by endowing the cartesian product of Q module spaces of tensors of different orders:

$$(\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_{N_1}}) \times (\mathbb{R}^{J_1} \otimes \mathbb{R}^{J_2} \otimes \cdots \otimes \mathbb{R}^{J_{N_2}}) \times \cdots \times (\mathbb{R}^{K_1} \otimes \mathbb{R}^{K_2} \otimes \cdots \otimes \mathbb{R}^{K_{N_Q}}) \tag{3.3}$$

with the *canonical inner product* formed upon the uniform sum of the module spaces' inner products:

$$\langle (\mathcal{W}_1, \mathcal{W}_2, \cdots, \mathcal{W}_Q), (\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_Q) \rangle_{\mathscr{W}} := \langle \mathcal{W}_1, \mathcal{V}_1 \rangle + \langle \mathcal{W}_2, \mathcal{V}_2 \rangle + \cdots + \langle \mathcal{W}_Q, \mathcal{V}_Q \rangle \,. \tag{3.4}$$

Note that we denoted the $q-$th component using the notation introduced in Section 2.3.2 for higher order tensors. When $N_q = 2$ (second order case) we stick with the notation for matrices and finally we denote it as a vector if $N_q = 1$. We denote $(\mathcal{W}_1, \mathcal{W}_2, \cdots, \mathcal{W}_Q)$ by $\mathcal{W}$. The norm associated to (3.4) is $\|\mathcal{W}\|_{\mathscr{W}} = \sqrt{\langle \mathcal{W}, \mathcal{W} \rangle_{\mathscr{W}}}$.

Remark. Assume $\mathscr{W}$ is formed upon the module spaces $\mathbb{R}^2 \otimes \mathbb{R}^3 \otimes \mathbb{R}^3$, $\mathbb{R}^4 \otimes \mathbb{R}^4$ and $\mathbb{R}^5$. A generic element of $\mathscr{W}$ will be denoted then by $(\mathcal{A}, \boldsymbol{B}, c)$ where we use different letters to emphasize the different role played by the corresponding elements.

Alternatively we will denote elements of $\mathscr{W}$, i.e., abstract vectors, by lower case letters $(w, v, \ldots)$ like ordinary vectors, i.e., elements of $\mathbb{R}^P$. We use this convention whenever we do not want to specify the structure of $\mathscr{W}$. We note that this is consistent with the fact that elements of $\mathscr{W}$ can always be considered as "long vectors" avoiding involved notation. Additionally we denote by capital letters $(A, B, C, \ldots)$ general operators between abstract spaces such as $F : \mathscr{W} \to \mathscr{V}$ and use lower case letters $(f, g, \ldots)$ to denote functionals on $\mathscr{W}$ namely operators of the type $f : \mathscr{W} \to \mathbb{R}$. Next we introduce the general family of optimization problems of interest.

3.1.2 Some Illustrative Examples

Problem (3.1) is very general and covers a wide range of learning problems where one faces single as well as *composite penalties* (see, e.g., [239]). To show this and illustrate the formalism introduced above we begin by the simplest problem that can be cast as (3.1), namely *Ridge Regression* [109].

Ridge Regression

Unlike the original proposal [109] we considered an additional bias term in the model, as common in machine learning. In this case the ambient space is defined upon two module spaces; equations (3.3) and (3.4) read:

$$\mathscr{W} = \mathbb{R}^D \times \mathbb{R}, \quad \langle (w,b),(v,c)\rangle_{\mathscr{W}} := \sum_{d\in\mathbb{N}_D} w_d v_d + bc \tag{3.5}$$

and one has

$$\bar{f}(w,b) = \frac{1}{2N}\sum_{n\in\mathbb{N}_N}(y_n - \sum_{d\in\mathbb{N}_D} w_d x_{nd} - b)^2, \quad \bar{g}(w) = \frac{1}{2}\sum_{d\in\mathbb{N}_D} w_d^2 \quad \text{and} \quad \bar{\mathscr{C}} = \mathscr{W}\,. \tag{3.6}$$

This formulation uses a single penalty, assumes input measurements in $\mathbb{R}^D$ and solves a single inductive regression task. The affine model $m(z) = \sum_{d\in\mathbb{N}_D} \hat{w}_d z_d + \hat{b}$, corresponding to the unique solution $(\hat{w},\hat{b})$, is estimated based upon input data collected in the *design matrix* $\boldsymbol{X} \in \mathbb{R}^N \otimes \mathbb{R}^D$ and a vector of measured responses $y \in \mathbb{R}^N$.

Group Lasso

As a second example, consider the more involved situation where the l_2 penalty used in Ridge Regression is replaced by the group lasso penalty with overlapping groups, see [117]. Let 2^{N_D} denote the *power set*[2] of N_D and consider some collection of M ordered sets $\{\mathscr{S}_1,\mathscr{S}_2,\ldots,\mathscr{S}_M\} \subseteq 2^{N_D}$. For any $w \in \mathbb{R}^D$ let $w|_{\mathscr{S}}$ be defined entry-wise by

$$(w|_{\mathscr{S}})_s = \begin{cases} w_s, & \text{if } s \in \mathscr{S} \\ 0, & \text{otherwise.} \end{cases}$$

[2]The power set of a set $\mathscr{A}$, denoted as $2^{\mathscr{A}}$, is the set of all subsets of $\mathscr{A}$, including the empty set, denoted as $\emptyset$, and $\mathscr{A}$ itself.

The group lasso problem with overlapping groups and an unpenalized bias term can be expressed as

$$\min_{w\in\mathbb{R}^D\times\mathbb{R}} f(w,b) + \sum_{m\in\mathbb{N}_m} g_m(w) \tag{3.7}$$

where $g_m(w) = \|w|_{\mathscr{S}_m}\|$ for any $m \in \mathbb{N}_M$. Consider now the abstract vector space

$$\mathscr{W} = \underbrace{\mathbb{R}^D \times \mathbb{R}^D \times \cdots \times \mathbb{R}^D}_{\text{M times}} \times \mathbb{R} \tag{3.8}$$

endowed with the canonical inner product

$$\langle (w_{[1]}, w_{[2]}, \ldots, w_{[M]}, b), (v_{[1]}, v_{[2]}, \ldots, v_{[M]}, c)\rangle_{\mathscr{W}} := \sum_{m\in\mathbb{N}_M}\sum_{d\in\mathbb{N}_D} (w_{[m]})_d (v_{[m]})_d + bc\ . \tag{3.9}$$

Define the set

$$\bar{\mathscr{C}} := \left\{ (w_{[1]}, w_{[2]}, \ldots, w_{[M]}, b) \in \mathscr{W} \ :\ w_{[1]} = w_{[2]} = \ldots = w_{[M]}\ ; \right\} \tag{3.10}$$

we can solve (3.7) by means of the problem

$$\begin{array}{rl} \underset{(w_{[1]},w_{[2]},\ldots,w_{[M]},b)\in\mathscr{W}}{\text{minimize}} & \bar{f}(w_{[1]}, w_{[2]}, \ldots, w_{[M]}, b) + \bar{g}(w_{[1]}, w_{[2]}, \ldots, w_{[M]}) \\ \text{subject to} & (w_{[1]}, w_{[2]}, \ldots, w_{[M]}, b) \in \bar{\mathscr{C}} \end{array} \tag{3.11}$$

where

$$\bar{f}(w_{[1]}, w_{[2]}, \ldots, w_{[M]}, b) = \frac{1}{M}\sum_{m\in\mathbb{N}_M} f(w_{[m]}, b) \tag{3.12}$$

$$g(w_{[1]}, w_{[2]}, \ldots, w_{[M]}) = \sum_{m\in\mathbb{N}_M} g_m(w_{[m]})\ . \tag{3.13}$$

Indeed it is clear that if $(\hat{w}_{[1]}, \hat{w}_{[2]}, \ldots, \hat{w}_{[M]}, \hat{b})$ is a solution of (3.11) then for any $m \in \mathbb{N}_M$ $(\hat{w}_{[m]}, \hat{b})$ is a solution of (3.7).

In the next Sections we will deal with both inductive and transductive multi-task learning problems where it is assumed that input features are collected within higher order tensors. Multiple module spaces will be used to account for tensor unknowns with different orders. A strategy similar to the one considered above for the group lasso will be used to recast our learning problems in terms of (3.1).

3.2 Template Algorithm

The fact that the rather distinct classes of problems discussed above can all be seen as special instances of (3.1) allows us to develop a unified algorithmical strategy to find their solutions. A central tool is the fact that $\mathscr{W}$ is a metric space (with the metric induced by an inner product as in (3.5) and (3.9)). Next we describe a provably convergent algorithm that is suitable for the situation where $\mathscr{W}$ is high dimensional. In the next Sections we will show how this general approach can be adapted to our different purposes.

For certain closed forms of $\bar{f}$ and $\bar{g}$, (3.1) can be restated as a semi-definite programming (SDP) problem [225] and solved via some SDP solvers such as SeDuMi [197], or SDPT3 [221]. However there is an increasing interest in the case where $\mathscr{W}$ is high dimensional in which case this approach is not satisfactory. Alternative scalable techniques that can be adapted to the solution of (3.1) are either based on the Alternating Direction Method of Multipliers (see [32] for a recent review) or are based on related proximal point algorithms designed to find a zero of the sum of maximal monotone operators. Classical references include [134, 168] and [193]. A modern and comprehensive review with application to signal processing is found in [51]. Here we propose an algorithm in the family of the Douglas-Rachford splitting methods. More specifically, our general approach can be regarded as a variant of the proximal decomposition method proposed in [50] and [49] by which it was inspired. Notably, as Subsection (3.1.2) shows, the algorithm we develop is not relevant for our tensor-based framework only.

3.2.1 Overview of Proximal Point Algorithms

In order to illustrate our scalable solution strategy we begin by equivalently restating (3.1) as the unconstrained problem:

$$\underset{w\in\mathscr{W}}{\text{minimize}} \quad \bar{f}(w) + \bar{g}(w) + \delta_{\bar{\mathscr{C}}}(w) \tag{3.14}$$

where $\delta_{\bar{\mathscr{C}}}$ is the indicator function:

$$\delta_{\bar{\mathscr{C}}} : \bar{w} \mapsto \begin{cases} 0, & \text{if } \bar{w} \in \bar{\mathscr{C}} \\ \infty, & \text{otherwise} . \end{cases}$$

Note that $\hat{w}$ is a solution to (3.14) if and only if [165]

$$0 \in \nabla \bar{f}(\hat{w}) + \partial \bar{g}(\hat{w}) + N_{\bar{\mathscr{C}}}(\hat{w}) \tag{3.15}$$

where $\nabla \bar{f}$ denotes the gradient of $\bar{f}$, $\partial \bar{g}$ is the *subdifferential* of $\bar{g}$ and finally $N_{\bar{\mathscr{C}}}$ is the *normal cone* of $\bar{\mathscr{C}}$, see, e.g., [19]:

$$N_{\bar{\mathscr{C}}}(w) := \begin{cases} \left\{ x \in \mathscr{W} \ : \ \langle x, y - w \rangle_{\mathscr{W}} \leq 0, \ \forall y \in \bar{\mathscr{C}} \right\}, & \text{if } w \in \bar{\mathscr{C}} \\ \emptyset, & \text{otherwise .} \end{cases}$$

Letting now

$$A := \nabla \bar{f} + N_{\bar{\mathscr{C}}} \quad \text{and} \quad B := \partial \bar{g} \tag{3.16}$$

equation (3.15) can be restated as

$$0 \in A(\hat{w}) + B(\hat{w}) \tag{3.17}$$

where A and B, as well as their sum $A + B$, are set-valued operators (for each w their image is a subset of $\mathscr{W}$) and they all qualify as *maximal monotone*. Maximal monotone operators, of which subdifferentials are a special instance, have been studied extensively in the literature, see, e.g., [35,145,166] and [158]. A recent account on the argument can be found in [19].

It is well known that, for any $\tau > 0$ and a given maximal monotone operator T on $\mathscr{W}$, $\hat{x} \in T^{-1}(0)$ if and only if $\hat{x}$ satisfies $\hat{x} \in R_{\tau T}\hat{x}$, i.e., if $\hat{x}$ is a fixed point of the single-valued *resolvent* of τT, defined as

$$R_{\tau T} := (I + \tau T)^{-1} \tag{3.18}$$

see, e.g., [19]. Proximal point algorithms are based on this fundamental fact and are a variation on the theme of the basic proximal iteration:

$$x^{(t+1)} = (I + \tau T)^{-1} x^{(t)} \ . \tag{3.19}$$

Note that, for the special case where T is the subdifferential of some convex function h, the latter can be restated as $x^{(t)} \in x^{(t+1)} + \tau \partial h(x^{(t+1)})$ and it is therefore equivalent to

$$0 \in \partial_x \left(h\left(x\right) + \frac{1}{2\tau} \left\| x - x^{(t)} \right\|_{\mathscr{W}}^2 \right) \Bigg|_{x = x^{(t+1)}} \tag{3.20}$$

which corresponds to:

$$x^{(t+1)} = \arg\min_x h\left(x\right) + \frac{1}{2\tau} \left\| x - x^{(t)} \right\|_{\mathscr{W}}^2 \ . \tag{3.21}$$

This illustrates the well known fact that the resolvent of the subdifferential ∂h of a convex function $h : \mathscr{W} \to \mathbb{R}$ corresponds to the *proximity operator* prox_h defined as [146]

$$\mathrm{prox}_h : x \mapsto \arg\min_{w \in \mathscr{W}} h(w) + \frac{1}{2}\|w - x\|^2 \ . \tag{3.22}$$

3.2.2 Douglas-Rachford Splitting Technique

When applied to $T = A + B$, the proximal iteration (3.19) is numerically viable only in those cases in which it is easy to compute $R_{\tau(A+B)}$. A more widely applicable alternative is to devise an operator splitting approach. In particular, a classical method to solve (3.17) is the Douglas-Rachford splitting technique that was initially developed in [134] based upon an idea found in [73]. The Douglas-Rachford splitting technique allows one to solve the inclusion problem (3.17) where A and B are assumed to be maximal monotone operators and consists of the main iteration G_{DR}:

$$G_{DR}(w^{(k)}; A, B, \gamma^{(k)}, \tau) \begin{cases} y^{(k)} & = R_{\tau A}(w^{(k)}), \quad (3.23a) \\ r^{(k)} & = R_{\tau B}(2y^{(k)} - w^{(k)}), \quad (3.23b) \\ w^{(k+1)} & = w^{(k)} + \gamma^{(k)} \left(r^{(k)} - y^{(k)}\right) \quad (3.23c) \end{cases}$$

In the latter τ is a positive proximity parameter and $(\gamma^{(k)})_k$ is a sequence of parameters that, once chosen appropriately, ensures convergence. With reference to (3.16), equation (3.23a) reads in our context

$$y^{(k)} = \arg\min_{x \in \mathscr{C}} \bar{q}\,(x) := \bar{f}\,(x) + \frac{1}{2\tau} \left\| x - w^{(k)} \right\|_{\mathscr{W}}^2 \tag{3.24}$$

whereas (3.23b) reads

$$r^{(k)} = \mathrm{prox}_{\tau \bar{g}}(2y^{(k)} - w^{(k)}). \tag{3.25}$$

3.2.3 Limits of Two-level Strategies

Next we present our algorithm based on an *inexact* variant of the Douglas-Rachford iteration. Our interest is in those situations where (3.25) can be computed exactly whereas the inner problem (3.24) requires an iterative procedure. As it turns out, in fact, in many situations one can design equivalent problems (and, contextually, the abstract space $\mathscr{W}$) in such a way that (3.25) can be computed easily and with high precision. Nonetheless, for general $\bar{f}$ in the inner problem (3.24), using the Douglas-Rachford iteration to solve (3.1) requires a procedure consisting of two nested iterative schemes. In general, the convergence of such a two-level strategy is ensured only upon exact solution of the inner problem. On the other hand, practical implementations require to specify a termination criterion and a corresponding accuracy. Notably [86] proposes different algorithms for an instance of the general problem in (3.1) similar to the formulations we will consider in Section 3.4. In particular the

authors also devise an inexact algorithm in Section 5.4 of [86] but they do not provide any convergence guarantee. Motivated by this we propose an adaptive termination criterion for the inner problem and prove the convergence of the outer scheme to a solution of (3.1).

3.2.4 Template Based on Inexact Splitting Technique

Denote by $P_{\bar{\mathscr{C}}}$ the projection onto $\bar{\mathscr{C}}$:

$$P_{\bar{\mathscr{C}}} : x \mapsto \arg\min_{w \in \bar{\mathscr{C}}} \|x - w\|_{\mathscr{W}}^2 \ . \tag{3.26}$$

The approach that we propose here for solving (3.1), termed *Inexact Splitting Method* (ISM), is presented in Algorithm 1. The idea is sketched as follows.

1. We apply an inexact version of G_{DR} to solve problem (3.1), where we only require to compute $y^{(k)}$ in (3.24) up to a given precision $\epsilon(k)$. Since, in our setting, (3.23b) can be computed in a closed form, we do not require any inexactness at this step.

2. Problem (3.24) is strongly convex for any $\tau > 0$. One can apply a gradient method that converges in this situation at a linear rate [149].

Notice that step 2 in the Main procedure consists of solving the optimization subproblem (3.24) with a precision $\epsilon(k)$ that depends upon the iteration index k. In practice this is achieved via the Goldstein-Levitin-Polyak gradient projection method, see [28, 29]. In the first main iterations a solution for (3.24) is found with low accuracy (from which the term *inexact*); as the estimate is refined along the iterations of MAIN the precision within the inner problem is increased; this ensures that the sequence $(y^{(k)})_k$ produced by Algorithm 1 converges to a solution of problem (3.1), as the following result shows.

Theorem 3.2.1. *Assume the solution set $\hat{\mathscr{S}}$ of problem* (3.1) *is not empty; In Algorithm 1 let $\epsilon_0 > 0,\ \sigma > 1$ and $\tau > 0$ be arbitrarily fixed parameters. Then $\{y^{(k)}\}_k$ converges to $\hat{w} \in \hat{\mathscr{S}}$.*

Proof. A proof of this fact can be found in [188]. □

Remark (Unknown Lipschitz constant). Notice that in the procedure that computes the proximity operator with adaptive precision we assumed known $L_{\bar{f}}$ as defined in (3.2); based upon the latter, $L_{\bar{q}}$ is immediately computed since $L_{\bar{q}} = L_{\bar{f}} + 1/\tau$, see [188, Lemma 24]. In practical applications, however, $L_{\bar{f}}$ is often unknown or hard to compute. In this situation an upper bound for $L_{\bar{q}}$

Algorithm 1: ISM()

procedure INEXACTPROXY($z, \epsilon, \tau, L_{\bar{f}}$)
comment: $z \in \mathscr{W}$

1. $L_{\bar{q}} \leftarrow L_{\bar{f}} + 1/\tau$
2. $w^{(0)} \leftarrow z$

repeat

3. $w^{(t+1)} \leftarrow P_{\bar{\mathscr{C}}}\left(w^{(t)} - \frac{1}{L_{\bar{q}}}\left(\nabla \bar{f}(w^{(t)}) + \frac{1}{\tau}\left(w^{(t)} - z\right)\right)\right)$

until $\|w^{(t)} - w^{(t-1)}\|_{\mathscr{W}} \leq \epsilon$
return $(w^{(t)})$

procedure MAIN()
comment: $\epsilon_0 > 0, \sigma > 1, \tau > 0$ arbitrarily fixed.

1. $w^{(0)} \in \mathscr{W}$
2. $\kappa \leftarrow \sqrt{\tau L_{\bar{f}}(\tau L_{\bar{f}} + 1)} + \tau L_{\bar{f}}$

repeat

3. $\epsilon^{(k)} \leftarrow \epsilon_0 / \left(\kappa (k+1)^{\sigma}\right)$
4. $y^{(k)} \leftarrow$ INEXACTPROXY$(w^{(k)}, \epsilon^{(k)}, \tau, L_{\bar{f}})$
5. $r^{(k)} \leftarrow \mathrm{prox}_{\tau \bar{g}}(2y^{(k)} - w^{(k)})$
6. $w^{(k+1)} \leftarrow w^{(k)} + r^{(k)} - y^{(k)}$

until convergence criterion met
return $(y^{(k)})$

can be found according to a backtracking strategy, see [21,150] for details. The constant step-size $L_{\bar{q}}$ in step 3 of INEXACTPROXY is replaced by an adaptive step-size $h \in (0, \frac{1}{L_{\bar{q}}}]$ as appropriately chosen by the backtracking procedure.

Remark (Termination of the outer loop). Since, as we proved, the sequence $\{y^{(k)}\}_k$ converges to the solution of problem (3.1), one can use the condition

$$\frac{\|y^{(k+1)} - y^{(k)}\|_{\mathscr{W}}}{\|y^{(k)}\|_{\mathscr{W}}} \leq \eta \tag{3.27}$$

to terminate the loop in the procedure MAIN, where $\eta > 0$ is a desired accuracy. However, we prefer to use the objective value to measure the approximate

solution. Typically, we terminate the outer loop if [3]

$$\frac{\left|\bar{J}(y^{(k+1)}) - \bar{J}(y^{(k)})\right|}{\left|\bar{J}(y^{(k)})\right|} \leq \eta \ . \tag{3.28}$$

3.3 Spectral Regularization for Higher Order Arrays

So far we have elaborated on the general formulation in (3.1); in this Section we specify the nature of the penalty functions that we are concerned with in our tensor-based framework. The concept of spectral penalty for matrices was introduced in Section 2.2.3 in connection to multi-task learning. Here we extend the idea both to (single) higher order tensors and to multiple tensor unknowns that one wants to jointly regularize. Additionally we elaborate on the corresponding proximity operators that we use to derive concrete algorithms.

3.3.1 Spectral Penalties for Higher Order Tensors

We recall that a *symmetric gauge function* $h : \mathbb{R}^P \to \mathbb{R}$ is a norm which is both absolute and invariant under permutations [234], see also [111, Definition 3.5.17]. Symmetric gauge functions are for instance all the l_p norms. The following definition generalizes the concept of spectral regularizer, studied in [2] and [8], to higher-order tensors.

Definition 3.3.1 ($n-$mode Spectral Penalty for Higher Order Tensors)**.** For $n \in \mathbb{N}_N$ a function $\Omega : \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N} \to \mathbb{R}$ is called an $n-$mode spectral penalty if it can be written as:

$$\Omega(\mathcal{W}) = h(\sigma(\mathcal{W}_{\langle n \rangle}))$$

where, for $R = \min\left\{I_n, \prod_{j \in \mathbb{N}_N \setminus \{n\}} I_j\right\}$, $h : \mathbb{R}^R \to \mathbb{R}$ is some symmetric gauge function and $\sigma(\mathcal{W}_{\langle n \rangle}) \in [0, \infty)^R$ is the vector of singular values of the matrix $\mathcal{W}_{\langle n \rangle}$ in non-increasing order.

We are especially interested in composite spectral regularizers obtained from the (weighted) sum of different $n-$mode spectral penalties. The earliest

[3] Note that in (3.28) and before in (3.27) we implicitly assumed that the denominator in the left hand-side is never exactly zero. In all the problems we will consider later, in particular, one can verify that this is always the case unless $0 \in \mathscr{W}$ is a solution, which never occurs in practical applications. For those specifications of $\bar{J}$ where this condition might arise one can replace $|\bar{J}(y^{(k)})|$ by $|\bar{J}(y^{(k)})| + 1$.

example of such a situation is found in [135]. Recall the definition of nuclear norm for matrices given in (2.21); [135] considers the penalty

$$g(\mathcal{W}) = \sum_{n \in \mathbb{N}_N} \frac{1}{N} \|\mathcal{W}_{\langle n \rangle}\|_* \tag{3.29}$$

with the purpose of performing completion of a partially observed tensor. It is clear that since $\|\mathcal{W}_{\langle n \rangle}\|_* = \|\sigma(\mathcal{W}_{\langle n \rangle})\|_1$ and $\|\cdot\|_1$ is a symmetric gauge function, (3.29) qualifies as a composite spectral regularizer.

The nuclear norm has been used to devise convex relaxation for rank constrained matrix problems [39, 40, 162]; this parallels the use of the l_1-norm in sparse approximation and cardinality minimization [47, 72, 207]. Likewise, minimizing (3.29) can be seen as a convex proxy for the minimization of the multilinear ranks.

3.3.2 Relation With Multilinear Rank

By Theorem 2.3.1, a tensor $\mathcal{W} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ can be written as:

$$\mathcal{W} = \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \cdots \times_N \boldsymbol{U}^{(N)} \tag{3.30}$$

where $\mathcal{S} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ is the core tensor and for any $n \in \mathbb{N}_N$, $\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n} \otimes \mathbb{R}^{I_n}$ consists of the left singular vectors of the $n-$mode unfolding $\mathcal{W}_{\langle n \rangle}$ with SVD [4]

$$\mathcal{W}_{\langle n \rangle} = \boldsymbol{U}^{(n)} \operatorname{diag}(\sigma(\mathcal{W}_{\langle n \rangle})) {\boldsymbol{V}^{(n)}}^\top . \tag{3.31}$$

Equation (3.30) is also known as the Multilinear Singular Value Decomposition (MLSVD). It has some striking similarities with the matrix SVD, see [67]. Since penalizing the nuclear norm of $\mathcal{W}_{\langle n \rangle}$ enforces the sparsity of $\sigma(\mathcal{W}_{\langle n \rangle})$, (3.29) favors low multilinear rank tensors. Notably for $N = 2$ (second order case) it is easy to see that (3.29) is consistent with the definition of nuclear norm for matrices. Similar results do not exist for (3.29) when $N > 2$, no matter what definition of tensorial rank one considers (see Subsection 2.3.2). It is therefore arguable whether or not it is appropriate to call it *nuclear norm for tensors*, as done in [135]. Nonetheless this penalty provides a viable way to compute low complexity estimates in the spirit of the Tucker decomposition. In contrast tensorial rank problems are know to be intractable [104,107]. To the best of our knowledge it remains an open problem to devise an appropriate convexification for this type of rank function. The problem was recently discussed in [43] but no solution was actually found.

[4] Assume the unfolding is performed according to the ordering rule in [67]. Then one has $\boldsymbol{V}^{(n)} = \left(\boldsymbol{U}^{(n+1)} \otimes \cdots \otimes \boldsymbol{U}^{(N-1)} \otimes \boldsymbol{U}^{(N)} \otimes \boldsymbol{U}^{(1)} \otimes \cdots \otimes \boldsymbol{U}^{(n-1)}\right)^\top$.

3.3.3 Proximity Operators

The numerical feasibility of proximal point algorithms largely depends upon the simplicity of computing the proximal operator introduced in (3.22). For the class of $n-$mode spectral penalties we can establish the following.

Proposition 3.3.1 (Proximity Operator of an $n-$mode Spectral Penalty)**.** *Assume* $\mathcal{W} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ *and let* (3.31) *be the SVD of its* $n-$*mode unfolding* $\mathcal{W}_{\langle n \rangle}$*. Then the evaluation at* $\mathcal{W}$ *of the proximity operator of* $\Omega(\mathcal{W}) = h(\sigma(\mathcal{W}_{\langle n \rangle}))$ *is*

$$\operatorname{prox}_{\Omega}(\mathcal{W}) = \left(\boldsymbol{U}^{(n)} \operatorname{diag}(\operatorname{prox}_{h}(\sigma(\mathcal{W}_{\langle n \rangle})))\boldsymbol{V}^{(n)\top}\right)^{\langle n \rangle} . \tag{3.32}$$

Proof. For a matrix $\boldsymbol{A}$ with SVD $\boldsymbol{A} = \boldsymbol{U} \operatorname{diag}(\sigma(\boldsymbol{A}))\boldsymbol{V}^{\top}$, [9, Proposition 3.1] established that

$$\operatorname{prox}_{h\circ\sigma}(\boldsymbol{A}) = \boldsymbol{U} \operatorname{diag}(\operatorname{prox}_{h}(\sigma(\boldsymbol{A})))\boldsymbol{V}^{\top} .$$

In order to prove Proposition 3.3.1 it remains to show that $\operatorname{prox}_{\Omega}(\mathcal{W}) = \left(\operatorname{prox}_{h\circ\sigma}(\mathcal{W}_{\langle n \rangle})\right)^{\langle n \rangle}$. Note that $\cdot_{\langle n \rangle}$ is a linear one-to-one (invertible) operator and that $(\mathcal{W}_{\langle n \rangle})^{\langle n \rangle} = \mathcal{W}$ namely, the composition between the folding operator and its adjoint yields the identity[5] on $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$. Additionally by the chain rule for the subdifferential (see, e.g., [149, Lemma 3.18]) and by the definition of Ω one has $\partial\Omega(\mathcal{V}) = (\partial(h \circ \sigma)(\mathcal{V}_{\langle n \rangle}))^{\langle n \rangle}$. We now have:

$$\begin{aligned} \mathcal{V} = \operatorname{prox}_{\Omega}(\mathcal{W}) \quad &\Leftrightarrow \quad \mathcal{V} - \mathcal{W} \in \partial\Omega(\mathcal{V}) = (\partial(h \circ \sigma)(\mathcal{V}_{\langle n \rangle}))^{\langle n \rangle} \\ &\Leftrightarrow \quad (\mathcal{V} - \mathcal{W})_{\langle n \rangle} \in \left((\partial(h \circ \sigma)(\mathcal{V}_{\langle n \rangle}))^{\langle n \rangle}\right)_{\langle n \rangle} \\ &\Leftrightarrow \quad \mathcal{V}_{\langle n \rangle} = \operatorname{prox}_{h\circ\sigma}(\mathcal{W}_{\langle n \rangle}) \\ &\Leftrightarrow \quad \mathcal{V} = \left(\operatorname{prox}_{h\circ\sigma}(\mathcal{W}_{\langle n \rangle})\right)^{\langle n \rangle} . \end{aligned} \tag{3.33}$$

□

In particular for the case where $\Omega(\mathcal{W}) = \lambda \left\|\sigma(\mathcal{W}_{\langle n \rangle})\right\|_1$ one has

$$\operatorname{prox}_{\lambda\|\sigma(\cdot_{\langle n \rangle})\|_1}(\mathcal{W}) = \left(\boldsymbol{U}^{(n)} \operatorname{diag}(d_\lambda)\boldsymbol{V}^{(n)\top}\right)^{\langle n \rangle} \tag{3.34}$$

where $(d_\lambda)_i := \max(\sigma_i(\mathcal{W}_{\langle n \rangle}) - \lambda, 0)$. The latter corresponds to refolding the matrix obtained applying to $\mathcal{W}_{\langle n \rangle}$ the *matrix shrinkage operator* [38].

[5] Equivalently, $\cdot_{\langle n \rangle}$ is unitary.

3.3.4 Multiple Module Spaces

So far we considered the case where $\mathscr{W}$ consisted solely of the module space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$. Next we focus on the case where $\mathscr{W}$ is given by 2 modules, see Subsection 3.1.1. The following definition will turn out to be useful in the next Section where we deal with two distinct types of unknowns that are jointly regularized.

Definition 3.3.2 ($(n_1, n_2)-$mode Spectral Penalty)**.** Assume a vector space $\mathscr{W}$ obtained endowing $\left(\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_{N_1}}\right) \times \left(\mathbb{R}^{J_1} \otimes \mathbb{R}^{J_2} \otimes \cdots \otimes \mathbb{R}^{J_{N_2}}\right)$ with the canonical inner product

$$\langle \mathcal{W}, \mathcal{V} \rangle_{\mathscr{W}} = \langle \mathcal{W}_1, \mathcal{V}_1 \rangle + \langle \mathcal{W}_2, \mathcal{V}_2 \rangle \tag{3.35}$$

and norm $\|\mathcal{W}\|_{\mathscr{W}} = \sqrt{\langle \mathcal{W}, \mathcal{W} \rangle_{\mathscr{W}}}$. Suppose that for $n_1 \in \mathbb{N}_{N_1}$ and $n_2 \in \mathbb{N}_{N_2}$ $I_{n_1} = J_{n_2} = K$ and let $S_1 := \prod_{p \in \mathbb{N}_{N_1} \setminus \{n_1\}} I_p$, $S_2 := \prod_{p \in \mathbb{N}_{N_2} \setminus \{n_2\}} J_p$. A function $\Omega : \mathscr{W} \to \mathbb{R}$ is called an $(n_1, n_2)-$mode spectral penalty if it can be written as:

$$\Omega(\mathcal{W}) = h\left(\sigma\left(\left[\mathcal{W}_{1\langle n_1 \rangle}, \mathcal{W}_{2\langle n_2 \rangle}\right]\right)\right) \tag{3.36}$$

where, for $R = \min\{K, S_1 S_2\}$, $h : \mathbb{R}^R \to \mathbb{R}$ is some symmetric gauge function and $\sigma(\left[\mathcal{W}_{1\langle n_1 \rangle}, \mathcal{W}_{2\langle n_2 \rangle}\right]) \in [0, \infty)^R$ is the vector of singular values of the matrix $\left[\mathcal{W}_{\langle n_1 \rangle}, \mathcal{W}_{\langle n_2 \rangle}\right]$ in non-increasing order.

Note that we required that $I_{n_1} = J_{n_2} = K$ since otherwise $\mathcal{W}_{1\langle n_1 \rangle}$ and $\mathcal{W}_{2\langle n_2 \rangle}$ cannot be concatenated.

Proposition 3.3.2 (Proximity Operator of an $(n_1, n_2)-$mode Spectral Penalty)**.** *Let $\mathscr{W}$, Ω, S_1 and S_2 be defined as in Definition 3.3.2 and assume the SVD $\left[\mathcal{W}_{1\langle n_1 \rangle}, \mathcal{W}_{2\langle n_2 \rangle}\right] = \boldsymbol{U}\sigma\left(\left[\mathcal{W}_{1\langle n_1 \rangle}, \mathcal{W}_{2\langle n_2 \rangle}\right]\right)\boldsymbol{V}^\top$. Then we have*

$$\operatorname{prox}_{\Omega}(\mathcal{W}) = \left(\boldsymbol{Z}_1^{\langle n_1 \rangle}, \boldsymbol{Z}_2^{\langle n_2 \rangle}\right) \tag{3.37}$$

where

$$\boldsymbol{Z} = \boldsymbol{U} \operatorname{diag}\left(\operatorname{prox}_h\left(\sigma\left(\left[\mathcal{W}_{1\langle n_1 \rangle}, \mathcal{W}_{2\langle n_2 \rangle}\right]\right)\right)\right)\boldsymbol{V}^\top \tag{3.38}$$

is partitioned into $[\boldsymbol{Z}_1, \boldsymbol{Z}_2]$ where $\boldsymbol{Z}_1$ is a $(K \times S_1)-$matrix and $\boldsymbol{Z}_2$ is a $(K \times S_2)-$matrix.

Proof. Consider the unfolding operator on $\mathscr{W}$,

$$\cdot_{\langle n_1 n_2 \rangle} : (\mathcal{W}_1, \mathcal{W}_1) \mapsto \left[\mathcal{W}_{1\langle n_1 \rangle}, \mathcal{W}_{2\langle n_2 \rangle}\right] .$$

Based on (3.35) it is not difficult to see that its adjoint corresponds to the operator $.^{\langle n_1 n_2 \rangle} : \mathbb{R}^K \otimes \mathbb{R}^{S_1+S_2} \to \mathscr{W}$ given by

$$.^{\langle n_1 n_2 \rangle} : [\boldsymbol{W}_1, \boldsymbol{W}_2] \mapsto \left(\boldsymbol{W}_1^{\langle n_1 \rangle}, \boldsymbol{W}_2^{\langle n_2 \rangle} \right) .$$

The chain rule for the subdifferential reads now

$$\partial\Omega(\mathcal{V}) = \left(\partial(h \circ \sigma) \left(\mathcal{V}_{\langle n_1 n_2 \rangle} \right) \right)^{\langle n_1 n_2 \rangle} .$$

In the same fashion as in (3.33) we now have:

$$\begin{aligned} \mathcal{V} = \operatorname{prox}_{\Omega}(\mathcal{W}) \quad &\Leftrightarrow \quad \mathcal{V} - \mathcal{W} \in \partial\Omega(\mathcal{V}) = (\partial(h \circ \sigma)(\mathcal{V}_{\langle n_1 n_2 \rangle}))^{\langle n_1 n_2 \rangle} \\ &\Leftrightarrow \quad (\mathcal{V} - \mathcal{W})_{\langle n_1 n_2 \rangle} \in \left((\partial(h \circ \sigma)(\mathcal{V}_{\langle n_1 n_2 \rangle}))^{\langle n_1 n_2 \rangle} \right)_{\langle n_1 n_2 \rangle} \\ &\Leftrightarrow \quad \mathcal{V}_{\langle n_1 n_2 \rangle} = \operatorname{prox}_{h \circ \sigma}(\mathcal{W}_{\langle n_1 n_2 \rangle}) \\ &\Leftrightarrow \quad \mathcal{V} = \left(\operatorname{prox}_{h \circ \sigma}(\mathcal{W}_{\langle n_1 n_2 \rangle}) \right)^{\langle n_1 n_2 \rangle} . \end{aligned} \tag{3.39}$$

□

We note that definition 3.3.2 and the result above can be easily generalized to more than two module spaces at the price of a more involved notation. Finally we recall the following known result which simple proof can be found in the literature, see, e.g., [19, Proposition 23.30].

Proposition 3.3.3. *For $i \in \mathbb{N}_I$ let $\mathscr{W}_i$ be some vector space with inner product $\langle \cdot, \cdot \rangle_i$. Let $\mathscr{W}$ be the space obtained endowing the cartesian product $\mathscr{W}_1 \times \mathscr{W}_2 \times \cdots \times \mathscr{W}_I$ with the inner product $\langle x, y \rangle = \sum_{i \in \mathbb{N}_I} \langle x_i, y_i \rangle_i$. Assume a function $f : \mathscr{W} \to \mathbb{R}$ defined by*

$$f : (x_1, x_2, \ldots, x_I) \mapsto \sum_{i \in \mathbb{N}_I} f_i(x_i)$$

where for any $i \in \mathbb{N}_I$, $f_i : \mathscr{W}_i \to \mathbb{R}$ is convex. Then we have

$$\operatorname{prox}_f(x) = \left(\operatorname{prox}_{f_1}(x_1), \operatorname{prox}_{f_2}(x_2), \cdots, \operatorname{prox}_{f_I}(x_I) \right) .$$

3.4 Multi-task Transductive Learning

In this section we specialize problem (3.1) in order to perform transductive learning with partially observed higher-order data. It is assumed one has a set of N items with higher-order representation

$$\mathcal{X}^{(n)} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M}, \ n \in \mathbb{N}_N .$$

These items are gathered in the input dataset $\mathcal{X} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \otimes \mathbb{R}^N$ defined entry-wise by

$$x_{d_1 d_2 \cdots d_M n} = x^{(n)}_{d_1 d_2 \cdots d_M} \ .$$

Associated to the $n-$th item there is a target vector $y^{(n)} \in \mathscr{Y}^T$. In particular we shall focus on the case where $\mathscr{Y} = \{-1, 1\}$ so that $\boldsymbol{Y} = [y^{(1)}, y^{(2)}, \cdots, y^{(N)}]$ is a $(T \times N)-$matrix of binary labels. Entries of $\mathcal{X}$ and $\boldsymbol{Y}$ can be missing with

$$\mathscr{S}_{\mathcal{X}} = \{(d_1^p, \ldots, d_M^p, n^p) \in \mathbb{N}_{D_1} \times \mathbb{N}_{D_2} \times \cdots \times \mathbb{N}_{D_M} \times \mathbb{N}_N \ : \ p \in \mathbb{N}_P\} \quad (3.40)$$

and

$$\mathscr{S}_{\boldsymbol{Y}} = \{(t^q, n^q) \in \mathbb{N}_T \times \mathbb{N}_N \ : \ q \in \mathbb{N}_Q\} \quad (3.41)$$

being the index set of the observed entries in, respectively, $\mathcal{X}$ and $\boldsymbol{Y}$. The goal is to infer the missing entries in $\mathcal{X}$ and $\boldsymbol{Y}$ simultaneously. We refer to this task as *heterogeneous* data completion to emphasize that the nature of $\mathcal{X}$ and $\boldsymbol{Y}$ is different. Note that this reduces to standard transductive learning (see Section 2.1.3) as soon as $T = 1$, $M = 1$ and finally $\mathscr{S}_{\mathcal{X}} = \emptyset$ (no missing entries in the input dataset). The paper [92] considers the more general situation where $T \geq 1$ and $\mathscr{S}_{\mathcal{X}} \neq \emptyset$. Here we further generalize this to the case where $M \geq 1$, that is, items admit a higher-order representation. We also point out that the special case where $T = 1$ and there is no labeling task defined (in particular, $\mathscr{S}_{\boldsymbol{Y}} = \emptyset$) corresponds to tensor completion as considered for the first time in [135]. Next we clarify our modelling assumptions.

3.4.1 Modelling Assumptions

The heterogeneous data completion task is ill-posed in the sense that there are infinitely many ways to fully specify the entries of $\mathcal{X}$ and $\boldsymbol{Y}$ [6]. Making the inference process feasible requires to formulate assumptions for both the input dataset as well as for the matrix of labels.

In this Section we consider the following generative model. It is assumed that the input dataset $\mathcal{X} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \otimes \mathbb{R}^N$ can be decomposed into

$$\mathcal{X} = \tilde{\mathcal{X}} + \mathcal{E} \quad (3.42)$$

where $\tilde{\mathcal{X}}$ is a rank-$(r_1, r_2, \ldots, r_M, r_{M+1})$ tensor and $\mathcal{E}$ is a remainder. In our setting the assumption considered in [92] is solely that

$$r_{M+1} \ll \min(N, J) \quad (3.43)$$

[6] This is the case since entries of $\mathcal{X}$ are in $\mathbb{R}$.

where

$$J = \prod_{j \in \mathbb{N}_M} D_j \ .$$

This amounts at regarding items as elements of $\mathbb{R}^J$ hereby neglecting their multimodal structure. Additionally, we further assume that

$$r_m \ll \min(D_m, NJ/D_m) \text{ for some } m \in \mathbb{N}_M \ . \tag{3.44}$$

This type of assumption is generally fulfilled in a number of cases where multimodal dependence arises; this occurs for instance when dealing with spectral images [189]. Additionally we suppose that

$$\tilde{y}_t^{(n)} = \langle \tilde{\mathcal{X}}^{(n)}, \mathcal{W}^{(t)} \rangle + b_t \tag{3.45}$$

where $\mathcal{W}^{(t)}$ and b_t are task-specific unknowns. Let $\mathcal{W} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \otimes \mathbb{R}^T$ be such that $w_{d_1 d_2 \cdots d_M t} = w^{(t)}_{d_1 d_2 \cdots d_M}$. Note that $\mathcal{W}$ gathers the representers of the linear functionals associated to the T tasks. We now have that

$$\tilde{\boldsymbol{Y}} = \mathcal{W}_{\langle M+1 \rangle} \tilde{\mathcal{X}}^{\top}_{\langle M+1 \rangle} + b \otimes 1_N \tag{3.46}$$

and it follows from (3.43) that

$$\operatorname{rank}\left(\left[\tilde{\mathcal{X}}_{\langle M+1 \rangle}, \tilde{\boldsymbol{Y}}^{\top} \right] \right) \leq r_{M+1} + 1 \ll \min(N, J+T) \ . \tag{3.47}$$

Finally it is assumed that the actual matrix $\boldsymbol{Y}$ is produced by assigning at random each binary entry with alphabet $\{-1, 1\}$ following the probability model

$$p(y_{tn} | \tilde{y}_{tn}) = 1/(1 + \exp(-y_{tn} \tilde{y}_{tn})) \ . \tag{3.48}$$

Remark. Notice that we have deliberately refrained from specifying the nature of $\mathcal{E}$ in (3.42). Central to our approach is the way input features and target labels are linked together; this is specified by the functional relation (3.45). One could interpret $\mathcal{E}$ as noise in which case $\tilde{\mathcal{X}}$ can be regarded as the underlying true representation of the input observation. This is in line with error-in-variables models [93, 224]. Alternatively one might regard $\mathcal{X}$ as the true representation and assume that the target variable depends only upon the latent tensor $\tilde{\mathcal{X}}$, having low multilinear rank $(r_1, r_2, \ldots, r_M, r_{M+1})$.

Before proceeding we introduce the sampling operator and its adjoint. These operators play a central role both in the problem formulation and in the practical algorithms that we discuss in Section 3.4.3 and 3.4.4.

3.4.2 Sampling Operator and its Adjoint

Assume $\mathcal{A} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ and consider the ordered set

$$\mathscr{S} = \{s^p := (i_1^p, i_2^p, \ldots, i_N^p) \in \mathbb{N}_{I_1} \times \mathbb{N}_{I_2} \times \cdots \times \mathbb{N}_{I_N} \ : \ p \in \mathbb{N}_P\}$$

identifying P entries of the $N-$th order tensor $\mathcal{A}$. In the following we denote by $S_{\mathscr{S}} : \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N} \to \mathbb{R}^P$ the *sampling operator* defined by

$$(S_{\mathscr{S}}\mathcal{X})_p := x_{i_1^p i_2^p \ldots i_N^p} \text{ for any } p \in \mathbb{N}_P \ .$$

Note that $S_{\mathscr{S}}$ is linear and it can be equivalently restated as $(S_{\mathscr{S}}\mathcal{X})_p = \langle \mathcal{X}, \mathcal{E}_{s^p} \rangle$ where $\mathcal{E}_{s^p}$ is that element of the canonical basis of $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ defined as

$$(\mathcal{E}_{s^p})_{i_1 i_2 \cdots i_N} := \begin{cases} 1, & \text{if } (i_1, i_2, \ldots, i_N) = s^p \\ 0, & \text{otherwise .} \end{cases}$$

Based upon this fact one can show that the adjoint of $S_{\mathscr{S}}\mathcal{X}$, namely that unique operator $S^*_{\mathscr{S}} : \mathbb{R}^P \to \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ satisfying $\langle b, S_{\mathscr{S}}\mathcal{X} \rangle = \langle S^*_{\mathscr{S}} b, \mathcal{X} \rangle$, is:

$$S^*_{\mathscr{S}} \ : \ b \mapsto \sum_{p \in \mathbb{N}_P} b_p \mathcal{E}_{s^p} \ .$$

It is immediate to check that $S_{\mathscr{S}} S^*_{\mathscr{S}} = I_P$ and hence, $S_{\mathscr{S}}$ is a *co-isometry* in the sense of [99, Section 127, page 69].

Remark. From this fact it follows that any solution of $S_{\mathscr{S}}\mathcal{X} = b$ can be written as $S^*_{\mathscr{S}} b + \mathcal{Z}$ where $\mathcal{Z} \in \ker(S_{\mathscr{S}}) := \left\{\mathcal{X} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N} \ : \ S_{\mathscr{S}}\mathcal{X} = 0\right\}$.

Remark. Sampling operators in line with $S_{\mathscr{S}}$ abound in learning theory and algorithms. For instance [191] considers a sampling operator on a *reproducing kernel Hilbert space* of functions $\mathscr{H}$ [10] based on a set of *evaluation functionals* of the type

$$L_x : f \mapsto f(x) \tag{3.49}$$

where $f \in \mathscr{H}$ is a function on a certain domain $\mathscr{X}$ and $x \in \mathscr{X}$. It is worth remarking that an $N-$th order array $\mathcal{A} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ can be regarded as a function from $\mathbb{N}_{I_2} \times \cdots \times \mathbb{N}_{I_N}$ to $\mathbb{R}$. Correspondingly, $S_{\mathscr{S}}$ can be restated in terms of evaluation functionals of the same type as (3.49), namely

$$L_{i_1^p i_2^p \ldots i_N^p} : \mathcal{A} \mapsto a_{i_1^p i_2^p \ldots i_N^p} \ .$$

This is no surprise as any finite dimensional space (such as $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$) is isomorphic to a reproducing kernel Hilbert space of functions, see, e.g., [27, Chapter 1].

3.4.3 Soft-completion of Heterogeneous Data

We denote by $S_{\mathscr{S}_\mathcal{X}}$ and $S_{\mathscr{S}_\mathbf{Y}}$ the sampling operators defined, respectively, upon (3.40) and upon (3.41). Let $m^x \in \mathbb{R}^P$ and $m^y \in \mathbb{R}^Q$ be the corresponding measurement vectors. Let $l_x, l_y : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ be some predefined convex loss functions respectively for the input data and the target labels. The empirical error functional we consider, namely

$$f_{\lambda_0}(\mathcal{X}, \mathbf{Y}, b) := f^x(\mathcal{X}) + \lambda_0 f^y(\mathbf{Y}, b) \tag{3.50}$$

is composed by an error for the inputs,

$$f^x : \mathcal{X} \mapsto \sum_{p \in \mathbb{N}_P} l_x((S_{\mathcal{S}_\mathcal{X}}\mathcal{X})_p, m^x_p) \tag{3.51}$$

and one for the target labels,

$$f^y : (\mathbf{Y}, b) \mapsto \sum_{q \in \mathbb{N}_Q} l_y((S_{\mathcal{S}_\mathbf{Y}}(\mathbf{Y} + b \otimes 1_J))_q, m^y_q) \ . \tag{3.52}$$

Problem Formulation

The heterogeneous completion task is then solved by means of the optimization problem

$$\min_{(\mathcal{X},\mathbf{Y},b)\in\mathscr{V}} f_{\lambda_0}(\mathcal{X}, \mathbf{Y}, b) + \textstyle\sum_{m\in\mathbb{N}_M} \lambda_m \|\mathcal{X}_{\langle m\rangle}\|_* + \lambda_{M+1} \left\| \left[\mathcal{X}_{\langle M+1\rangle}, \tilde{\mathbf{Y}}^\top\right] \right\|_* \tag{3.53}$$

where $\mathscr{V}$ is obtained endowing the cartesian product:

$$\left(\mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \otimes \mathbb{R}^{D_N}\right) \times \left(\mathbb{R}^T \otimes \mathbb{R}^N\right) \times \mathbb{R}^T \tag{3.54}$$

with the inner product

$$\langle (\mathcal{X}_1, \mathbf{Y}_1, b_1), (\mathcal{X}_2, \mathbf{Y}_2, b_2)\rangle_\mathscr{V} = \langle \mathcal{X}_1, \mathcal{X}_2\rangle + \langle \mathbf{Y}_1, \mathbf{Y}_2\rangle + \langle b_1, b_2\rangle \ ; \tag{3.55}$$

for any $m \in \{0\} \cup \mathbb{N}_{M+1}$, $\lambda_m > 0$ is a user-defined parameter and $\sum_{m\in\mathbb{N}_{M+1}} \lambda_m = 1$. Problem (3.53) is convex since its objective is the sum of convex functions. It is a form of penalized empirical risk minimization with a composite penalty. The first M penalty terms

$$\Omega_m : (\mathcal{X}) \mapsto \lambda_m \|\mathcal{X}_{\langle m\rangle}\|_*, \ m \in \mathbb{N}_M \tag{3.56}$$

reflect the modelling assumption (3.44). The $(M+1)-$th penalty

$$T : (\mathcal{X}, \boldsymbol{Y}) \mapsto \lambda_{M+1} \left\| \left[\mathcal{X}_{\langle M+1 \rangle}, \tilde{\boldsymbol{Y}}^\top \right] \right\|_*$$

ensures that the recovered matrix $\left[\hat{\mathcal{X}}_{\langle M+1 \rangle}, \hat{\boldsymbol{Y}}^\top \right]$ is approximately low rank, in line with equation (3.47). The contribution of the different terms is trimmed by the associated regularization parameters which are either preselected or chosen according to some model selection criterion.

In principle any meaningful pair of convex penalties can be used to define the error functional. Here we follow [92] and consider in (3.51) and (3.52)

$$l_x \quad : \ (u, v) \mapsto \quad \frac{1}{2}(u-v)^2 \qquad \text{(quadratic loss)} \tag{3.57a}$$

$$l_y \quad : \ (u, v) \mapsto \quad \log(1 + \exp(-uv)) \qquad \text{(logistic loss)} \ . \tag{3.57b}$$

Note that (3.57a) is fully justified by assuming that $\mathcal{E}$ in (3.42) has Gaussian entries. These losses ensure that the overall error functional (3.50) is smooth. This fact, along with the tools developed in Section 3.3, allows us to use Algorithm 1 as a template to devise a solution strategy.

Algorithm for Soft-completion

In order to rely on Algorithm 1, we need to suitably design $\mathscr{W}$, $\bar{f}$, $\bar{g}$ as well as $\mathscr{C}$. That is, we have to cast (3.53) into the prototypical formulation in (3.1). Consider the abstract vector space $\mathscr{W}$ obtained endowing [7]

$$\left(\times_{m \in \mathbb{N}_{M+1}} \left\{ \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \otimes \mathbb{R}^{D_N} \right\} \right) \times \left(\mathbb{R}^T \otimes \mathbb{R}^N \right) \times \mathbb{R}^T \tag{3.58}$$

with the canonical inner product

$$\langle (\mathcal{X}_{[1]}, \ldots, \mathcal{X}_{[M+1]}, \boldsymbol{Y}, b), (\mathcal{W}_{[1]}, \ldots, \mathcal{W}_{[M+1]}, \boldsymbol{U}, c) \rangle_{\mathscr{W}} := \sum_{m \in \mathbb{N}_{M+1}} \langle \mathcal{X}_{[m]}, \mathcal{W}_{[m]} \rangle + \langle \boldsymbol{Y}, \boldsymbol{U} \rangle + \langle b, c \rangle \ . \tag{3.59}$$

[7] We adopt the short-hand $\times_{m \in \mathbb{N}_M} \mathscr{A}$ to indicate the iterated Cartesian product: $\underbrace{\mathscr{A} \times \mathscr{A} \times \cdots \times \mathscr{A}}_{\text{M times}}$.

Once defined the set

$$\bar{\mathscr{C}} := \big\{(\mathcal{X}_{[1]}, \mathcal{X}_{[2]}, \ldots, \mathcal{X}_{[M]}, \mathcal{X}_{[M+1]}, \boldsymbol{Y}, b) \in \mathscr{W} \ : \\ \mathcal{X}_{[1]} = \mathcal{X}_{[2]} = \ldots = \mathcal{X}_{[M+1]}\big\} \quad (3.60)$$

we can solve (3.53) by means of the problem

$$\begin{aligned} \underset{(\mathcal{X}_{[1]},\ldots,\mathcal{X}_{[M+1]},\boldsymbol{Y},b)\in\mathscr{W}}{\text{minimize}} \quad & \bar{f}(\mathcal{X}_{[1]}, \ldots, \mathcal{X}_{[M+1]}, \boldsymbol{Y}, b) + \bar{g}(\mathcal{X}_{[1]}, \ldots, \mathcal{X}_{[M+1]}, \boldsymbol{Y}) \\ \text{subject to} \quad & (\mathcal{X}_{[1]}, \ldots, \mathcal{X}_{[M+1]}, \boldsymbol{Y}, b) \in \bar{\mathscr{C}} \end{aligned} \quad (3.61)$$

where

$$\bar{f}(\mathcal{X}_{[1]}, \ldots, \mathcal{X}_{[M+1]}, \boldsymbol{Y}, b) := \quad \frac{1}{M+1} \sum_{m \in \mathbb{N}_{M+1}} f^x(\mathcal{X}_{[m]}) + \lambda_0 f^y(\boldsymbol{Y}, b) \quad (3.62)$$

$$\bar{g}(\mathcal{X}_{[1]}, \ldots, \mathcal{X}_{[M+1]}, \boldsymbol{Y}) := \quad \sum_{m \in \mathbb{N}_M} \Omega_m(\mathcal{X}_{[m]}) + T(\mathcal{X}, \boldsymbol{Y}) \ . \quad (3.63)$$

Application of propositions 3.3.1, 3.3.2 and 3.3.3 shows now that $\text{prox}_{\tau\bar{g}}$ in Step 3 of MAIN (Algorithm 1) reads:

$$\text{prox}_{\tau\bar{g}}(\mathcal{X}_{[1]}, \ldots, \mathcal{X}_{[M+1]}, \boldsymbol{Y}) = \\ \Big(\text{prox}_{\tau\lambda_1\|\sigma(\cdot\langle 1\rangle)\|_1}(\mathcal{X}_{[1]}), \cdots, \text{prox}_{\tau\lambda_M\|\sigma(\cdot\langle M\rangle)\|_1}(\mathcal{X}_{[M]}), \boldsymbol{Z}_1, \boldsymbol{Z}_2\Big) \quad (3.64)$$

where $[\boldsymbol{Z}_1, \boldsymbol{Z}_2]$ is a partitioning of

$$\boldsymbol{Z} = \boldsymbol{U}\,\text{diag}\left(\text{prox}_{\tau\lambda_{M+1}\|\sigma(\cdot)\|_1}\left(\left[\mathcal{X}_{\langle M+1\rangle}, \boldsymbol{Y}^\top\right]\right)\right)\boldsymbol{V}^\top \quad (3.65)$$

consistent with the dimensions of $\mathcal{X}_{\langle M+1\rangle}$ and $\boldsymbol{Y}^\top$, the operator $\text{prox}_{\lambda\|\sigma(\cdot)\|_1}$ is defined as in (3.34) and finally $\boldsymbol{U}$ and $\boldsymbol{V}$ are respectively left and right singular vectors of the matrix $\left[\mathcal{X}_{\langle M+1\rangle}, \boldsymbol{Y}^\top\right]$. The closed form of $\nabla\bar{f}$ can be found in [188]. Finally note that (3.26) reads here:

$$P_{\bar{\mathscr{C}}}(\mathcal{X}_{[1]}, \cdots, \mathcal{X}_{[M+1]}, \boldsymbol{Y}, b) = \left(\sum_{m \in \mathbb{N}_{M+1}} \mathcal{X}_{[m]}, \cdots, \sum_{m \in \mathbb{N}_{M+1}} \mathcal{X}_{[m]}, \boldsymbol{Y}, b\right) . \quad (3.66)$$

3.4.4 Hard-completion without Target Labels

Problem (3.53) assumes that data consist of both input and target measurements. In turn, the situation were we do not consider target measurements can be dealt with by the following special instance of (3.53):

$$\hat{\mathcal{X}} = \arg \min_{\mathcal{X} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \otimes \mathbb{R}^{D_N}} f^x(\mathcal{X}) + \sum_{m \in \mathbb{N}_{M+1}} \Omega_m(\mathcal{X}) \ . \tag{3.67}$$

As before, f^x penalizes the misfit of $\mathcal{X}$ to the partially observed input data tensor; the composite penalty term favors solutions with small multilinear rank. The solution strategy illustrated in the previous Subsection can be easily adjusted to deal with this situation.

Problem Formulation

For certain practical problems, however, it is more desirable to require the exact adherence to the data. Let us use $\mathscr{V}$ as a shorthand notation for $\mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \otimes \mathbb{R}^{D_N}$. Strict adherence to observables can be accomplished by means of the following constrained formulation of *tensor completion* [86,135,189,213]:

$$\begin{array}{ll} \underset{w \in \mathscr{V}}{\text{minimize}} & \sum_{m \in \mathbb{N}_{M+1}} \Omega_m(\mathcal{X}) \\ \text{subject to} & S_{\mathscr{S}_\mathcal{X}} \mathcal{X} = m^x \end{array} \tag{3.68}$$

where $\mathscr{S}_\mathcal{X}$ is the sampling set (3.40).

Algorithm for Hard-completion

As before in order to devise a solution strategy for problem (3.68) we accommodate Algorithm 1. Consider the abstract space $\mathscr{W}$ obtained endowing $\mathscr{V}^{M+1}$ with the canonical inner product. Let us introduce the constraint set:

$$\begin{aligned} \bar{\mathscr{C}} := \big\{ (\mathcal{X}_{[1]}, \mathcal{X}_{[2]}, \ldots, \mathcal{X}_{[M+1]}) \in \mathscr{W} \ : \\ \mathcal{X}_{[1]} = \mathcal{X}_{[2]} = \ldots = \mathcal{X}_{[M+1]}, \ \Omega_{\mathscr{S}_\mathcal{X}} \mathcal{X}_{[m]} = m^x \ \forall \ m \in \mathbb{N}_{M+1} \big\} \ . \end{aligned} \tag{3.69}$$

It is clear that a solution of (3.68) is readily obtained from a solution of the following problem:

$$\begin{array}{ll} \underset{(\mathcal{X}_{[1]}, \mathcal{X}_{[2]}, \ldots, \mathcal{X}_{[M+1]}) \in \mathscr{W}}{\text{minimize}} & \sum_{m \in \mathbb{N}_{M+1}} \Omega_m(\mathcal{X}_{[m]}) \\ \text{subject to} & (\mathcal{X}_{[1]}, \mathcal{X}_{[2]}, \ldots, \mathcal{X}_{[M+1]}) \in \bar{\mathscr{C}} \ . \end{array} \tag{3.70}$$

Note that, with respect to the prototypical problem (3.1), we now have that $\bar{f}$ is identically zero and

$$\bar{g} : (\mathcal{X}_{[1]}, \mathcal{X}_{[2]}, \ldots, \mathcal{X}_{[M+1]}) \mapsto \sum_{m \in \mathbb{N}_{M+1}} \Omega_m(\mathcal{X}_{[m]}) \ . \tag{3.71}$$

Additionally, the projection of elements of $\mathscr{W}$ onto $\bar{\mathscr{C}}$ can be computed in closed form. To see this let $\mathcal{X}|_{\mathcal{B}}$ denote the tensor obtained from $\mathcal{X} \in \mathbb{R}^{D_1} \otimes \cdots \otimes \mathbb{R}^{D_M} \otimes \mathbb{R}^{D_N}$ setting to zero those entries that are not indexed by $\mathcal{B} \subset \mathbb{N}_{D_1} \times \mathbb{N}_{D_2} \times \cdots \times \mathbb{N}_{D_M} \times \mathbb{N}_N$:

$$x|_{\mathcal{B}b_1b_2\cdots b_Mc} := \begin{cases} 0 & \text{if } (b_1, b_2, \cdots, b_M, c) \notin \mathcal{B} \\ x_{b_1b_2\cdots b_Mc} & \text{otherwise} \ . \end{cases}$$

We have the following.

Proposition 3.4.1 (Projection onto $\bar{\mathscr{C}}$)**.** *Let $\mathscr{W}$ and $\bar{\mathscr{C}}$ be defined as above. Then for any $(\mathcal{X}_{[1]}, \cdots, \mathcal{X}_{[M+1]}) \in \mathscr{W}$, it holds that*

$$P_{\bar{\mathscr{C}}}(\mathcal{X}_{[1]}, \mathcal{X}_{[2]}, \cdots, \mathcal{X}_{[M+1]}) = (\mathcal{Z}, \mathcal{Z}, \cdots, \mathcal{Z})$$

where

$$\mathcal{Z} = \left(\frac{1}{M+1} \sum_{m \in \mathbb{N}_{M+1}} \mathcal{X}_{[m]} \right) \Bigg|_{\mathscr{S}_{\mathcal{X}}^c} + S^*_{\mathscr{S}_{\mathcal{X}}} m^x,$$

and we denoted by $S^*_{\mathscr{S}_{\mathcal{X}}}$ the adjoint of the sampling operator $S_{\mathscr{S}_{\mathcal{X}}}$.

Proof. See [188]. □

Finally by propositions 3.3.1 and 3.3.3 it follows that $\mathrm{prox}_{\tau\bar{g}}$ in Step 3 of MAIN (Algorithm 1) reads:

$$\mathrm{prox}_{\tau\bar{g}}(\mathcal{X}_{[1]}, \ldots, \mathcal{X}_{[M+1]}) = \\ \left(\mathrm{prox}_{\tau\lambda_1 \|\sigma(\cdot_{\langle 1 \rangle})\|_1}(\mathcal{X}_{[1]}), \cdots, \mathrm{prox}_{\tau\lambda_M \|\sigma(\cdot_{\langle M \rangle})\|_1}(\mathcal{X}_{[M]}) \right) \ . \tag{3.72}$$

The steps needed to compute a solution are reported in Algorithm 2, which is obtained by adapting Algorithm 1 to the present setting. With reference to the latter, note that INEXACTPROXY is no longer needed. Indeed, since $\bar{f}$ is identically zero, (3.24) boils down to computing the projection onto $\bar{\mathscr{C}}$. By Proposition 3.4.1, this can be done in closed form.

Remark. Algorithm 2 is explicitly designed for the higher order case ($M \geq 2$). However it can be easily simplified to perform hard completion of matrices. In this case it is not difficult to see that one needs to evaluate only one proximity operator; consequently, multiple copies of the matrix unknown can be avoided.

Algorithm 2: HARDCOMPLETION()

procedure MAIN()
comment: $\tau > 0$ arbitrarily fixed.

1. $\left(\mathcal{W}_{[1]}^{(0)}, \cdots, \mathcal{W}_{[M+1]}^{(0)}\right) \in \mathscr{W}$

repeat

$$\left\{\begin{array}{lrcl} 4. & \mathcal{Y}^{(k)} & \leftarrow & \left(\frac{1}{M+1}\sum_{m\in\mathbb{N}_{M+1}} \mathcal{W}_{[m]}^{(k)}\right)\Big|_{\mathscr{I}_{\mathcal{X}}^{c}} + S_{\mathscr{I}_{\mathcal{X}}}^{*} m^{x} \\ 5. & \mathcal{R}_{[m]}^{(k)} & \leftarrow & \operatorname{prox}_{\tau\lambda_m \|\sigma(\cdot_{\langle m\rangle})\|_1}(2\mathcal{Y}^{(k)} - \mathcal{W}_{[m]}^{(k)}) \;\forall\, m \in \mathbb{N}_{M+1} \\ 6. & \mathcal{W}_{[m]}^{(k+1)} & \leftarrow & \mathcal{W}_{[m]}^{(k)} + \mathcal{R}_{[m]}^{(k)} - \mathcal{Y}^{(k)} \;\forall\, m \in \mathbb{N}_{M+1} \end{array}\right.$$

until convergence criterion met
return $(\mathcal{Y}^{(k)})$

3.5 Multi-task Inductive Learning

For the inductive case the goal is to learn an affine predictive model based upon a dataset $\mathscr{D}_N$ of N input-target training pairs

$$\mathscr{D}_N := \left\{\left(\mathcal{X}^{(n)}, y^{(n)}\right) \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \times \mathscr{Y}^T \;:\; n \in \mathbb{N}_N\right\} . \quad (3.73)$$

Each item is represented by an $M-$th order tensor and is associated with a vector of T labels. As before, we focus on the case where $\mathscr{Y} = \{-1, 1\}$. For ease of notation we assumed that we have the same input data across the tasks; in general, however, this needs not to be the case.

To understand the rationale behind the regularization approach we are about to propose, consider the following generative mechanism.

3.5.1 Modelling Assumptions

For a generic item, represented by the tensor $\mathcal{X}$, assume the decomposition $\mathcal{X} = \tilde{\mathcal{X}} + \mathcal{E}$ where

$$\tilde{\mathcal{X}} = \mathcal{S}_{\tilde{\mathcal{X}}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times \cdots \times_M \boldsymbol{U}_M \ . \tag{3.74}$$

Note that the core tensor $\mathcal{S}_{\tilde{\mathcal{X}}} \in \mathbb{R}^{R_1} \otimes \mathbb{R}^{R_2} \otimes \cdots \otimes \mathbb{R}^{R_M}$ and $\mathcal{E} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M}$ are item-specific; on the other hand for any $m \in \mathbb{N}_M$ and $R_m \ll D_m$, the full rank matrix $\boldsymbol{U}_m \in \mathbb{R}^{D_m} \otimes \mathbb{R}^{R_m}$ spans a latent space relevant to the tasks at hand and common to all the input data. To be precise we assume that the target labels y_t were generated according to the probability model $p(y_t|\tilde{y}_t) = 1/(1 + \exp(-y_t\tilde{y}_t))$, where $\tilde{y}_t$ depends upon the core tensor $\mathcal{S}_{\tilde{\mathcal{X}}}$:

$$\tilde{y}_t = \langle \mathcal{S}_{\tilde{\mathcal{X}}}, \mathcal{S}_{\tilde{\mathcal{W}}^{(t)}} \rangle + b_t$$

where $\mathcal{S}_{\tilde{\mathcal{W}}^{(t)}} \in \mathbb{R}^{R_1} \otimes \mathbb{R}^{R_2} \otimes \cdots \otimes \mathbb{R}^{R_M}$ and b_t are task-specific unknowns. It is important to remark that $\mathcal{S}_{\mathcal{W}^{(t)}}$ comprises $R_1R_2 \cdots R_M \ll D_1D_2 \cdots D_M$ parameters. In practice the common latent spaces as well as the core tensor $\mathcal{S}_{\tilde{\mathcal{X}}}$ are both unknowns so that $\mathcal{S}_{\tilde{\mathcal{W}}^{(t)}}$ cannot be estimated directly. However if we further assume that

$$\mathscr{R}(\mathcal{E}_{\langle m \rangle}) \perp \mathscr{R}(\boldsymbol{U}_m) \text{ for some } m \in \mathbb{N}_M \ , \tag{3.75}$$

where we denote by $\mathscr{R}(\boldsymbol{A})$ the range of a matrix $\boldsymbol{A}$, one has:

$$\langle \mathcal{S}_{\tilde{\mathcal{X}}}, \mathcal{S}_{\tilde{\mathcal{W}}^{(t)}} \rangle = \langle \mathcal{X}, \tilde{\mathcal{W}}^{(t)} \rangle \tag{3.76}$$

where $\tilde{\mathcal{W}}^{(t)} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M}$ is the low multilinear rank tensor

$$\tilde{\mathcal{W}}^{(t)} = \mathcal{S}_{\tilde{\mathcal{W}}^{(t)}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times \cdots \times_M \boldsymbol{U}_M \tag{3.77}$$

and the right-hand side of (3.76) is an affine function of the given higher order representation $\mathcal{X}$ of the item. This fact follows immediately from properties of the $n-$mode product and the assumptions in (3.74).

Remark. Equation (3.75) implies in particular that $\langle \mathcal{E}, \tilde{\mathcal{X}} \rangle = 0$, namely, $\tilde{\mathcal{X}} + \mathcal{E}$ is an orthogonal decomposition. In general, if this condition is not satisfied, one can have, at most, that $\langle \mathcal{S}_{\tilde{\mathcal{X}}}, \mathcal{S}_{\tilde{\mathcal{W}}^{(t)}} \rangle \approx \langle \mathcal{X}, \tilde{\mathcal{W}}^{(t)} \rangle$ and therefore $\tilde{y}_t \approx \langle \mathcal{X}, \tilde{\mathcal{W}}^{(t)} \rangle + b_t$ where $\tilde{\mathcal{W}}^{(t)}$ refers to (3.77).

Let now $\tilde{\mathcal{W}} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \otimes \mathbb{R}^T$ be the tensor that gathers all the T tasks:

$$\tilde{w}_{d_1 d_2 \cdots d_M t} := \tilde{w}^{(t)}_{d_1 d_2 \cdots d_M} \text{ for any } t \in \mathbb{N}_T \ . \tag{3.78}$$

Additionally we consider the case where the tasks are related, as common in the literature of multi-task learning, see, e.g., [6]. In our context this is

accomplished by assuming that $\tilde{\mathcal{W}}_{\langle M+1\rangle}$ can be explained by a limited number of factors, namely that $\tilde{\mathcal{W}}_{\langle M+1\rangle}$ admits a thin SVD [94]:

$$\tilde{\mathcal{W}}_{\langle M+1\rangle} = \boldsymbol{U}_{M+1}\boldsymbol{S}_{M+1}\boldsymbol{V}_{M+1}^{\top}$$

where for $R_m \ll D_m$ one has $\boldsymbol{U}_{M+1} \in \mathbb{R}^{D_m} \otimes \mathbb{R}^{R_m}$, $\boldsymbol{S}_{M+1} \in \mathbb{R}^{R_m} \otimes \mathbb{R}^{R_m}$ and finally $\boldsymbol{V}_{M+1} \in \mathbb{R}^{R_m} \otimes \mathbb{R}^{D_1 D_2 \cdots D_{m-1} D_{m+1} \cdots D_M}$. Note that $\tilde{\mathcal{W}}$ can now be equivalently restated as the low multilinear rank tensor

$$\tilde{\mathcal{W}} = \mathcal{S}_{\tilde{\mathcal{W}}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \cdots \times_{M+1} \boldsymbol{U}_{M+1} \tag{3.79}$$

for some core tensor $\mathcal{S}_{\tilde{\mathcal{W}}} \in \mathbb{R}^{R_1} \otimes \mathbb{R}^{R_2} \otimes \cdots \otimes \mathbb{R}^{R_M} \otimes \mathbb{R}^{R_{M+1}}$ and latent matrices $\boldsymbol{U}_m$, $m \in \mathbb{N}_{M+1}$ that define subspaces that concentrate the discriminative relationship.

We conclude by pointing out that a supervised learning problem where data observations are represented as matrices, namely second order tensors, is a special case of our setting. Single classification tasks in this situation were studied in [211]. Similarly to the present setting, the latter proposes a spectral regularization as a principled way to perform complexity control over the space of matrix-shaped models. Before discussing a solution strategy we point out that the method can be easily generalized to regression problems by changing the loss function.

3.5.2 Problem Formulation

As for transduction we evaluate misclassification errors via the logistic loss; we therefore measure the empirical risk associated to the different tasks on the dataset (5.3) via:

$$f_{\mathscr{D}_N} : (\mathcal{W}, b) \mapsto \sum_{n\in\mathbb{N}_N} \sum_{t\in\mathbb{N}_T} \log\left(1 + \exp\left(-y_t^{(n)}\left(\langle \mathcal{X}^{(n)}, \mathcal{W}^{(t)}\rangle + b_t\right)\right)\right) \ . \tag{3.80}$$

The pair $(\mathcal{W}, b)$ is estimated based upon the following penalized empirical risk minimization problem:

$$\underset{(\mathcal{W},b)\in\mathscr{V}}{\text{minimize}} \quad f_{\mathscr{D}_N}(\mathcal{W}, b) + \textstyle\sum_{m\in\mathbb{N}_{M+1}} \lambda_m \|\mathcal{W}_{\langle m\rangle}\|_* \tag{3.81}$$

where $\mathscr{V}$ is formed upon the module spaces $\mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \otimes \mathbb{R}^T$ and $\mathbb{R}^T$. The composite spectral penalty in (3.81) is designed to match the assumption that $\mathcal{W}$ has low multilinear rank, as discussed above. Note that, in line with (3.79), other than performing complexity control, the regularization allows one to determine subspaces that concentrate discriminative information, without any additional feature extraction step.

3.5.3 Algorithm

Consider the abstract vector space $\mathscr{W}$ obtained endowing

$$\left(\times_{m\in\mathbb{N}_{M+1}}\left\{\mathbb{R}^{D_1}\otimes\mathbb{R}^{D_2}\otimes\cdots\otimes\mathbb{R}^{D_M}\otimes\mathbb{R}^{T}\right\}\right)\times\mathbb{R}^{T} \tag{3.82}$$

with the canonical inner product. Additionally let

$$\bar{\mathscr{C}}:=\left\{(\mathcal{W}_{[1]},\mathcal{W}_{[2]},\ldots,\mathcal{W}_{[M]},\mathcal{W}_{[M+1]},b)\in\mathscr{W}\ :\ \mathcal{W}_{[1]}=\mathcal{W}_{[2]}=\ldots=\mathcal{W}_{[M+1]}\right\}. \tag{3.83}$$

We solve (3.81) based upon the following problem:

$$\begin{aligned}\underset{(\mathcal{W}_{[1]},\mathcal{W}_{[2]},\ldots,\mathcal{W}_{[M]},\mathcal{W}_{[M+1]},b)\in\mathscr{W}}{\text{minimize}}\quad & \bar{f}(\mathcal{W}_{[1]},\ldots,\mathcal{W}_{[M+1]},b)+\bar{g}(\mathcal{W}_{[1]},\ldots,\mathcal{W}_{[M+1]})\\ \text{subject to}\quad & (\mathcal{W}_{[1]},\ldots,\mathcal{W}_{[M+1]},b)\in\bar{\mathscr{C}}\end{aligned} \tag{3.84}$$

where

$$\bar{f}(\mathcal{W}_{[1]},\ldots,\mathcal{W}_{[M+1]},b):=\frac{1}{M+1}\sum_{m\in\mathbb{N}_{M+1}}f_{\mathscr{D}_N}(\mathcal{W}_{[m]},b) \tag{3.85}$$

and $\bar{g}$ is the same as in (3.71). Its proximity operator is found in (3.72). The formula for the gradient of $\bar{f}$, which we also need in order to rely on Algorithm 1, can be found in [188].

3.6 Experiments

In this Section we empirically assess the effectiveness of the proposed approaches. To do so we run different tests where we compared our tensor-based techniques with alternative approaches.

3.6.1 Transductive Learning

We begin by presenting experiments on transductive learning with multiple tasks, see Section 3.4.

Evaluation Criterion and Choice of Parameters

As performance indicators we considered: 1) the capability of each procedure to predict the correct test labels; 2) the capability to interpolate the missing

entries in the input data tensor. We measure the latter based upon the *normalized root mean square error* (NRMSE) on the complementary set $\mathscr{S}^{\mathrm{c}}_{\tilde{\mathcal{X}}}$:

$$\mathrm{NRMSE}(\tilde{\mathcal{X}}, \hat{\mathcal{X}}) := \frac{\left\| S_{\mathscr{S}^{\mathrm{c}}_{\tilde{\mathcal{X}}}} \tilde{\mathcal{X}} - S_{\mathscr{S}^{\mathrm{c}}_{\tilde{\mathcal{X}}}} \hat{\mathcal{X}} \right\|}{\left(\max\left(S_{\mathscr{S}^{\mathrm{c}}_{\tilde{\mathcal{X}}}} \right) - \min\left(S_{\mathscr{S}^{\mathrm{c}}_{\tilde{\mathcal{X}}}} \right) \right) \sqrt{J^c}} \tag{3.86}$$

where we denoted by J^c the cardinality of $\mathscr{S}^{\mathrm{c}}_{\tilde{\mathcal{X}}}$, $\tilde{\mathcal{X}}$ is as in (3.42), and $\hat{\mathcal{X}}$ denotes the estimated tensor. For both *tensor soft-completion* (tensor-sc) and *matrix soft-completion* (matrix-sc) we solve the optimization problem in (3.53) via the approach presented in Subsection 3.4.3. The parameter λ_0 is chosen in the set $\left\{10^{-5}, 10^{-4}, 10^{-3}, \ 10^{-2}, 10^{-1}, 1\right\}$. For tensor-sc we set the parameters in the composite spectral penalty as

$$\lambda_m = \frac{1}{M+1} \bar{\lambda} \text{ for any } m \in \mathbb{N}_{M+1} \tag{3.87}$$

where $M+1$ is the order of the input data tensor and $\bar{\lambda}$ is a varying parameter. For matrix-sc we take

$$\lambda_m = \begin{cases} 0 & \text{if } m \in \mathbb{N}_M \\ \bar{\lambda} & \text{if } m = M+1 \ . \end{cases} \tag{3.88}$$

Notice that by doing so we essentially recover the matrix-based approach proposed in [92, Formulation 1]. We let $\bar{\lambda}$ in both (3.87) and (3.88) vary on a wide range. More precisely we follow [92] and [137]; for each value of λ_0 we compute the regularization path with respect to $\bar{\lambda}$ beginning with a large value $\bar{\lambda}^{(0)}$ and solving a sequence of problems with $\bar{\lambda}^{(t)} = \eta_{\bar{\lambda}} \bar{\lambda}^{(t-1)}$ where as in [92] we consider as decay parameter $\eta_{\bar{\lambda}} = 0.25$. At each step t we perform warm-starting, that is, we take as initial point the solution obtained at step $t-1$. We stop when $\bar{\lambda} \leq 10^{-6}$. For both tensor and matrix soft-completion we choose the values of parameters corresponding to the minimum fraction of mis-predicted labels of a hold-out validation set.

Remark. Note that in our experiments we either considered a single spectral penalty (3.88) or a composite regularizer where all the penalties were equally enforced (3.87). The same strategy is followed also later for the inductive case. Although uniform weights are shown to work in practice, this black and white setting might be restrictive: alternatively one might perform model selection to search for the optimal combination of parameters. Unfortunately this comes at the price of increasing substantially the computation burden. Additionally, when the model selection is performed based upon a small-sample, tuning too many parameters may lead to overfitting.

Implementation of the Optimization Algorithm

As termination criterion for the algorithm that finds a solution of (3.53) we use the relative increment (3.28) where we set $\eta = 10^{-4}$. With reference to Algorithm 1 (see also Subsection 3.4.3) we let $\epsilon_0 = 10^{-2}$ and set $\sigma = 1.1$. We use a backtracking procedure to find an upper bound for the Lipschitz constant $L_{\bar{q}}$ (see Remark 3.2.4 and references therein). Finally we let $\tau = 0.02/\tilde{L}_{\bar{f}}$ where $\tilde{L}_{\bar{f}}$ is an upper bound for the Lipschitz constant $L_{\bar{f}}$, also found via backtracking. As explained above we compute the entire path with respect to the penalty parameter and use warm-starting at each step. At step $t = 0$ the initialization of the algorithm is performed as follows. For both matrix as well as tensor-sc we set $b^{(0)}$ to be a vector of zeros. For what concerns $\mathcal{X}^{(0)}$ and $\boldsymbol{Y}^{(0)}$ we do as follows. Let $\mathcal{X}^*$ and $\boldsymbol{Y}^*$ be obtained by setting to zero unobserved entries of $\mathcal{X}$ and $\boldsymbol{Y}$ respectively. Consider a partitioning $[\boldsymbol{Z}_{M+1,1}, \boldsymbol{Z}_{M+1,2}]$ of the rank-1 approximation of the matrix $[\mathcal{X}^*_{<M+1>}, \boldsymbol{Y}^{*\top}]$ consistent with the dimension of $\mathcal{X}^*_{<M+1>}$ and $\boldsymbol{Y}^{*\top}$. Both matrix-sc and tensor-sc are then initialized according to

$$\mathcal{X}^{(0)} = \boldsymbol{Z}^{\langle M+1\rangle}_{M+1,1} \quad \text{and} \quad \boldsymbol{Y}^{(0)} = \boldsymbol{Z}^{\top}_{M+1,2} \ .$$

This approach is adapted from the method suggested in [92] for their matrix soft completion procedure.

Alternative Approach

We also report results obtained using linear kernel within LS-SVM classifiers applied to vectorized input data, see [200]. We find models via the LS-SVMlab toolbox [63]. A classifier is built for each task independently since these models do not handle vector-valued labels simultaneously. Although the presence of missing values in the context of LS-SVM has been studied [157] the toolbox does not implement any strategy to handle this situation. For this reason we considered the vectorized version of $\mathcal{X}^* + \boldsymbol{Z}^{\langle M+1\rangle}_{M+1,1}|_{\mathscr{S}^c_{\mathcal{X}}}$ as input data, where $\mathcal{X}^*$ and $\boldsymbol{Z}^{\langle M+1\rangle}_{M+1,1}$ are as in the previous paragraph. We denote this approach as imp+ls-svm where imp is a shorthand for imputation.

Soft Completion: Toy Problems on Multi-labeled Data

For the first set of experiments we considered a family of synthetic datasets following the generative mechanism illustrated in Subsection 3.4.1. For each experiment we generated a core tensor $\mathcal{S}$ in $\mathbb{R}^{r_1} \otimes \mathbb{R}^{r_2} \otimes \mathbb{R}^{r_3}$ with entries i.i.d. from a normal distribution; for $i \in \{1, 2\}$ a matrix $\boldsymbol{U}_i \in \mathbb{R}^D \otimes \mathbb{R}^{r_i}$ with entries i.i.d. from a normal distribution. Finally $\boldsymbol{U}_3 \in \mathbb{R}^N \otimes \mathbb{R}^{r_3}$ was generated

according to the same distribution. The input data tensor in $\mathbb{R}^D \otimes \mathbb{R}^D \otimes \mathbb{R}^N$ was taken to be

$$\mathcal{X} = \mathcal{S} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 + \sigma \mathcal{E}.$$

Next, for each task $t \in \mathbb{N}_T$ we created a weight tensor $\mathcal{W}_t$ and a bias b_t, again with independent and identically normally distributed entries; successively, we produced $\tilde{\boldsymbol{Y}}$ and $\boldsymbol{Y}$ according to (3.46) and the probability model (3.48)[8]. Finally, the sampling sets $\mathscr{S}_{\mathcal{X}}$ and $\mathscr{S}_{\boldsymbol{Y}}$ in (3.40) and (3.41) were created by picking uniformly at random a fraction ω of entries of the data tensor and the target matrix respectively. For matrix and tensor soft-completion we performed model selection by using 70% of these entries for training; we measure the performance corresponding to each parameter pair on the hold-out validation set constituted by the remaining entries. We finally use the optimal values of parameters and train with the whole set of labelled data; we then measure performance on the hold-out test set. Model selection for the linear LS-SVM classifiers was based on 10−fold cross-validation.

The procedure above was considered for $D = 30,\ T = 10,\ \sigma = 0.1$ and different values of the multilinear rank $(r_1, r_2, r_3),\ N$ and ω. Table 3.2 concerns the fraction of unobserved labels (that is, test data) predicted incorrectly by the different procedures as well as NRMSEs. Note that the latter is not reported for the linear LS-SVM models as these approaches do not have an embedded imputation strategy. We report the mean (and standard deviation) over 10 independent trials where each trial deals with independently generated data and sample sets.

[8]Note that each input observation possesses multiple (binary) labels. This situation differs from the multi-class paradigm that we consider later on. In here any possible binary vector is admissible; by contrast, in a multi-class setting only those vectors belonging to the codebook are admissible.

Table 3.2: Fractions of unobserved labels (test data) predicted incorrectly and NRMSEs. tensor-sc exploits low rank assumptions along all the modes; in contrast, matrix-sc works only with the third mode unfolding (hereby ignoring the two-way nature of each data observation). tensor-sc generally performs comparably or better than matrix-sc and imp+ls-svm in terms of misclassification errors; tensor-sc generally outperforms matrix-sc on the reconstruction of the underlying input data tensor $\tilde{\mathcal{X}}$, see (3.42).

ml-rank	N	ω	tensor-sc		matrix-sc		imp+ls-svm
			label error	NRMSE ($\times 10^{-2}$)	label error	NRMSE ($\times 10^{-2}$)	label error
(3, 3, 3)	30	0.2	**0.28**(0.09)	**6.81**(1.24)	0.31(0.08)	7.63(1.42)	0.32(0.03)
		0.3	**0.20**(0.07)	**3.29**(2.39)	0.21(0.07)	5.98(2.24)	**0.20**(0.05)
		0.4	**0.13**(0.04)	**1.68**(1.29)	**0.13**(0.04)	3.87(1.12)	0.14(0.06)
	90	0.2	**0.11**(0.03)	**2.75**(1.00)	**0.11**(0.02)	3.74(1.26)	0.16(0.03)
		0.3	**0.08**(0.02)	**0.87**(0.36)	0.09(0.02)	1.68(0.67)	0.10(0.02)
		0.4	**0.05**(0.01)	**1.87**(1.56)	0.06(0.01)	2.55(1.67)	0.07(0.02)
(3, 3, 9)	30	0.2	0.44(0.09)	**11.13**(2.52)	0.42(0.09)	11.57(1.96)	**0.41**(0.05)
		0.3	**0.29**(0.04)	**5.57**(2.53)	0.33(0.06)	10.19(2.51)	0.33(0.03)
		0.4	**0.25**(0.04)	**4.91**(5.31)	0.27(0.04)	8.82(2.88)	0.27(0.04)
	90	0.2	**0.17**(0.02)	**4.71**(0.56)	0.18(0.02)	6.63(1.16)	0.27(0.02)
		0.3	**0.13**(0.02)	**3.97**(2.67)	0.14(0.02)	4.07(1.62)	0.17(0.03)
		0.4	**0.10**(0.01)	**2.81**(3.15)	0.11(0.02)	3.44(2.10)	0.14(0.02)

Multi-class Categorization via Soft-completion: Olivetti Faces

In this experiment we deal with the classification of pictures of faces taken from different poses; we compared tensor-sc with matrix-sc and imp+ls-svm as before. We considered the first five persons of the Olivetti database [9]. For each person, ten 56×46 [10] grayscale pictures with different poses are available; the input dataset consists therefore of a $(56 \times 46 \times 50)-$tensor of which 65% of the entries were artificially removed. For each input image a vector-valued target label was created with one-vs-one encoding. That is, if $c_i \in \{1, 2, \cdots, 5\}$ denotes the class (person) indicator for the $i-$th image we set $y^{(i)} \in \{-1, 1\}^5$ to be

$$(y^{(i)})_j = \begin{cases} -1, & \text{if } j \neq c_i \\ 1, & \text{otherwise .} \end{cases}$$

This type of encoding was also used within imp+ls-svms. For these classifiers we considered as input the vector unfoldings of the images. Likewise matrix-sc, obtained from (3.53) setting all but the last regularization parameter equal to zero, treats each image as a vector by considering only the last matrix unfolding. In all the cases we use 25 images for training and validation and the remaining for testing. As for the toy problems above, the spectral penalties parameters within matrix-sc and tensor-sc were set according to (3.88) and (3.87). We compute the regularization path corresponding to the free parameter $\bar{\lambda}$. In all the cases the selection of parameters is driven by the misclassification error. For the LS-SVM models we used ten-fold CV. For matrix-sc and tensor-sc we chose parameters according to a hold-out set. More precisely, of the 25 images we use 17 for actual training and consider the remaining 8 for validation and then use all of the 25 images once the set of optimal parameters has been found. Following this procedure we performed five trials each of which obtained from a random splitting of training and test data and random mask of input missing entries. For each method we report in Table 3.3 the cumulative confusion matrix obtained summing up the confusion matrices associated to the test (unlabelled) data found in the different trials. Table 3.4 summarizes the performance of the different methods in terms of classification accuracy and feature imputation.

Remark. Note that, since the choice of parameters was driven by misclassification errors, the objective of the approach is the correct completion of the labeling. Therefore the estimated input features $\hat{\mathcal{X}}$ in (3.53) should be interpreted as carriers of latent discriminative information rather than a reconstruction of the underlying images. With reference to Remark 3.4.1 we interpret here $\mathcal{X}^{(i)}$ as the true representation of the $i-$th image, $\tilde{\mathcal{X}}^{(i)}$ as latent discriminative features and $\hat{\mathcal{X}}^{(i)}$ as their estimates.

[9] Publicly available at `http://cs.nyu.edu/~roweis/data.html`.

[10] Sizes refer to the images obtained after removing borders.

Table 3.3: Cumulative confusion matrices for the different procedures. tensor-sc leads to better classification accuracy in comparison to the alternative techniques

		y				
		1	2	3	4	5
$\hat{y}$	1	**19**	3	0	0	3
	2	0	**29**	0	0	0
	3	0	0	**20**	0	0
	4	0	0	1	**23**	0
	5	0	0	0	0	**27**

tensor-sc

		y				
		1	2	3	4	5
$\hat{y}$	1	17	3	1	0	4
	2	2	27	0	0	0
	3	0	0	**20**	0	0
	4	0	0	3	21	0
	5	1	0	2	0	24

matrix-sc

		y				
		1	2	3	4	5
$\hat{y}$	1	14	5	0	1	5
	2	0	**29**	0	0	0
	3	0	0	19	0	1
	4	0	0	2	22	0
	5	0	0	0	0	**27**

imp+ls-svm

Table 3.4: Mean and standard deviation of misclassification error rates and NRMSE of features imputation for the Olivetti dataset

tensor-sc		matrix-sc		imp+ls-svm
label err	NRMSE ($\times 10^{-2}$)	label err	NRMSE ($\times 10^{-2}$)	label err
0.05(0.10)	19.90(9.75)	0.11(0.14)	20.90(10.24)	0.09(0.13)

Remark. Unlike in the toy problems above, for which $\tilde{\mathcal{X}}$ was available, the NRMSEs in Table 3.4 are computed upon the actual set of images $\mathcal{X}$.

Figure 3.1 illustrates the retrieval of latent features for some unlabelled (test) pictures. Notably the latent features obtained by tensor-sc look as over-smoothed images whereas those obtained by matrix-sc generally look more noisy. In particular, the cases for which matrix-sc incorrectly assigns labels often correspond to situations where latent features do not capture person-specific traits (first and second rows). Wrongly assigned labels also correspond to cases where latent features are close to those corresponding to a different person (last two rows).

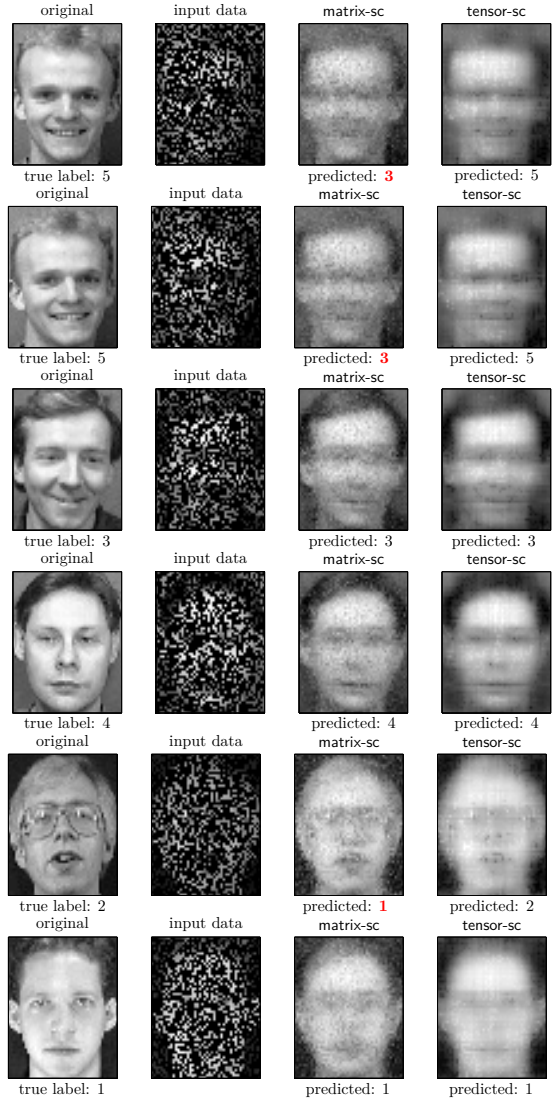


Figure 3.1: Olivetti faces: reconstructed features and assigned labels for a few examples; see also Tables 3.3 and 3.4. Wrong labels are reported in red. Estimated input features should be interpreted as carriers of latent discriminative information rather than a reconstruction of the underlying images, see the Remark at page 71. In particular, the cases for which matrix-sc incorrectly assigns labels often correspond to situations where latent features do not capture person-specific traits (first and second rows). Wrongly assigned labels also correspond to cases where latent features are close to those corresponding to a different person (last two rows).

Table 3.5: NRMSEs and execution times for tensor-hc and matrix-hc; the latter completes the given tensor based upon low rank assumption on the third mode only. Note that the computational complexity of tensor-hc is roughly 3 times that of matrix-hc, in line with the last remark at page 75.

σ	multilinear rank	ω	tensor-hc		matrix-hc	
			NMRSE ($\times 10^{-2}$)	time (s)	NMRSE ($\times 10^{-2}$)	time (s)
0.02	(3, 3, 3) (SNR $\approx$ 3)	0.3	**4.41**(0.48)	42.04(2.15)	5.51(0.65)	13.93(1.01)
		0.6	**3.10**(0.43)	23.24(1.14)	3.86(0.55)	8.05(1.09)
		0.9	**3.07**(0.60)	16.13(1.26)	3.71(0.73)	5.24(0.58)
	(3, 3, 9) (SNR $\approx$ 9.5)	0.3	**3.47**(0.36)	54.65(2.03)	6.15(0.63)	21.09(0.89)
		0.6	**2.36**(0.26)	29.06(1.10)	4.04(0.44)	11.98(1.04)
		0.9	**2.11**(0.22)	18.27(0.96)	3.44(0.36)	5.99(0.58)
	(9, 9, 3) (SNR $\approx$ 30)	0.3	3.32(0.45)	104.21(7.00)	**2.26**(0.35)	28.10(1.59)
		0.6	1.96(0.31)	41.01(2.25)	**1.57**(0.24)	13.12(0.67)
		0.9	1.77(0.22)	21.59(1.05)	**1.40**(0.18)	6.64(0.44)
0.04	(3, 3, 3) (SNR $\approx$ 0.8)	0.3	**7.88**(0.87)	50.34(3.92)	8.55(0.99)	17.53(1.64)
		0.6	**5.75**(0.82)	35.03(1.51)	6.68(1.00)	11.65(0.94)
		0.9	**5.74**(1.13)	25.67(1.20)	6.68(1.13)	8.17(0.58)
	(9, 9, 3) (SNR $\approx$ 2.36)	0.3	**6.28**(0.65)	77.31(3.54)	8.53(0.86)	26.74(1.88)
		0.6	**4.42**(0.50)	46.08(1.58)	6.54(0.70)	15.60(1.02)
		0.9	**3.99**(0.41)	29.97(1.10)	5.90(0.59)	9.71(0.55)
	(9, 9, 3) (SNR $\approx$ 7.5)	0.3	5.95(0.78)	118.89(6.74)	**4.55**(0.60)	35.62(2.09)
		0.6	3.72(0.59)	59.66(2.54)	**3.00**(0.47)	19.39(0.88)
		0.9	3.40(0.43)	36.20(1.80)	**2.71**(0.34)	10.61(0.72)

Hard Completion: Toy Problems

Here we test the capability of hard completion (Section 3.4.4), denoted by tensor-hc, to recover missing entries of a partially specified input data tensor of order 3. We compare to the case where the higher order structure is neglected; in this case the input data tensor is flattened into its third matrix unfolding and one performs matrix completion (matrix-hc). Note that, with reference to (3.68), this is equivalent to retain the tensor structure and set all but the last of the regularization parameters $\lambda_m,\ m \in \{1,2,3\}$ to zero. In either case we compute solutions via Algorithm 2 keeping into account the simplifications that occur in the second order case. For each experiment we generated a core tensor $\mathcal{S}$ in $\mathbb{R}^{r_1} \otimes \mathbb{R}^{r_2} \otimes \mathbb{R}^{r_3}$ with entries i.i.d. according to the uniform distribution on the interval $[-0.5, 0.5]$, denoted as $U([-0.5, 0.5])$; for $i \in \{1,2,3\}$ a matrix $\boldsymbol{U}_i \in \mathbb{R}^D \otimes \mathbb{R}^{r_i}$ was generated also with entries i.i.d. from $U([-0.5, 0.5])$. The input data tensor in $\mathbb{R}^D \otimes \mathbb{R}^D \otimes \mathbb{R}^D$ was taken to be $\mathcal{X} = \tilde{\mathcal{X}} + \sigma\,\mathcal{E}$ where

$$\tilde{\mathcal{X}} = \mathcal{S} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3$$

and $\mathcal{E}$ is a $(D \times D \times D)-$tensor with independent normally distributed entries. Finally the sampling sets $\mathscr{S}_{\mathcal{X}}$ were created by picking uniformly at random a fraction ω of entries of the data tensor. We took $D = 50$ and considered different values of σ, the multilinear rank (r_1, r_2, r_3) and ω. In Table 3.5 we report the mean (and standard deviation) of NRMSEs and execution times over 10 independent trials where each trial deals with independently generated data and sample sets. Note that keeping σ fixed across different values of multilinear rank gives a different noise level. We therefore report in the table the approximate *signal-to-noise ratio* (SNR) obtained on the different trials. The latter is defined as $SNR := \mathrm{var}(\tilde{x})/(\sigma^2)$ where $\tilde{x}$ denotes the generic entry of $\tilde{\mathcal{X}}$ and var denotes the empirical variance.

Remark. The experimental evidence suggests that tensor completion performs better than matrix completion (performed along the third mode) when either no $n-$rank dominates the others or when the $3-$rank is higher. This observation holds across different noise levels and fractions of entries used in the reconstruction.

Remark. As Table 3.5 shows, for the third order case, the computational complexity of our implementation of tensor completion is roughly three times that of matrix completion. This is expected since the computational load is determined by Step 5 and 6 of Algorithm 2 which, in turn, involve a number of iterations equal to the order of the tensor $(M+1)$. This is no longer needed for the second order case, see the Remark at page 63.

Impainting of Colored Images via Hard Completion

Here we apply hard-completion to impainting of 8-bit RGB colored images, each of which is represented as a third order tensor. The first two modes span the pixels space; the third mode contains information from the three channels. For each image we remove entries in all the channels simultaneously (first three rows of Figure 3.2), or consider the case where entries are missing at random (last two rows of Figure 3.2). We then solve the problem in equation (3.68) with the sampling set $\mathscr{S}_{\mathcal{X}}$ indexing non-missing pixels. A solution is found via Algorithm 2. As termination criterion we use the relative increment (3.28) where we set $\eta = 10^{-7}$. With reference to Algorithm 2 we let $\tau = 10^4$ and $\lambda_m = 1$ for $m \in \{1, 2, 3\}$. Figure 3.2 reports the original pictures, the input data tensor and the outcome of our algorithm.

3.6.2 Hard completion of Hyperspectral Data

Spectral data consist of collections of measurements sensed from contiguous spectral bands. We call *frame* the measurements corresponding to a single band that are organized within a matrix of size $D_1 \times D_2$. The complete dataset results from arranging the D_3 frames along the third dimension of a tensor $\mathcal{X}$ which, as a result, has dimensions $D_1 \times D_2 \times D_3$. Hyperspectral Remote Sensing (HRS) [44] is used for the detection and identification of minerals, terrestrial vegetation, and man-made materials. In contrast Mass Spectral Imaging (MSI) studies the distribution in tissues of biochemical compounds such as metabolites, peptides, or proteins by their molecular masses [203], [122].

Completion and decay of spectra

A common approach when dealing with an ensemble of images corresponds to vectorizing each image [220]. For the case of hyperspectral data this corresponds to regarding the $D_1 \times D_2 \times D_3$ data tensor as the third $D_3 \times D_1 D_2$ matrix unfolding $\mathcal{X}_{<3>}$. In practice, intensities measured at nearby wavelengths are normally highly correlated. It is then expected that each row of $\mathcal{X}_{<3>}$ can be approximately restated as a linear combination of a small number of orthogonal vectors that play the role of *eigenimages*. This corresponds to a low rank assumption and ultimately motivates the completion of $\mathcal{X}_{<3>}$ based upon the nuclear norm (2.21). For tensors, the idea of relying on the third mode unfolding to perform completion is found in [212]. However, we note that — especially when the number of spectral bands is limited — it might be not possible to interpolate the values of missing entries at a certain frequency

based upon the intensity on the same locations in the other frequencies. In these cases relying solely on spectral redundancy might not be ideal.

A possible workaround is based on the realization that the aforementioned approach discards the matrix representation of each frame. Different algorithms for images, such as [105], successfully rely on this additional structural information. For a given frame, the well known Eckart-Young-Mirsky theorem states that the best low rank approximation in least squares sense is obtained via its truncated SVD. In particular it is often the case that a good approximation is achieved by retaining a small number of singular values/vectors. Matrix completion performed independently on each frame exploits this prior belief only. However, since the entire data tensor is available, it makes sense to exploit this property as well as the redundancy along the spectral mode. It is reasonable to expect that the spectrum of the different matrix unfoldings has a fast decay (Figure 3.3). Since minimizing using the penalty (3.29) enforces low multilinear ranks, tensor completion is seen to leverage this composite assumption.

Tensor versus Matrix Completion

Next we assess these different approaches by experiments. For each trial we draw uniformly at random the sampling set $\mathscr{S}_{\mathcal{X}}$ (see (3.40)) containing a prescribed fraction of the total number of entries of the data tensor. The estimation error incurred by each procedure is measured in terms of the NRMSE function (see (3.86)) $\mathrm{NRMSE}(\mathcal{X}, \hat{\mathcal{X}})$. In the following we consider the cases where the approximation $\hat{\mathcal{X}}$ is obtained by *a*) tensor completion (indicated as *tensor* in the tables) *b*) combining the solutions of frame completion problems (indicated as *frame*) *c*) third mode matrix completion (indicated as *mode-3*). Notice that, since matrix completion is a special case of tensor completion, the same algorithm can be used to deal with the three cases (with certain simplification for the two matrix problems).

Mass Spectral Imaging

As a first case study we considered Matrix-assisted laser desorption/ionization (MALDI) mass spectral measurements that were performed on a sagittal section of the brain of a mouse at the University Hospital Leuven. Details about the experimental setting can be found in [153, 222]. Here we focus on a grid consisting of 22×22 measurement locations (pixels) situated at the center of the tissue section. For each location N equispaced mass-over-charge (m/z) bins within the range from 2500 to about 25000 Dalton where considered to form

the $22 \times 22 \times N$ MSI data tensor. Figure 3.4 reports the intensity plots for two different values of m/z overlaid on the registered sectioned tissue. The (normalized) intensity of each pixel represents the number of molecules (or ions) collected for the specified m/z bin. In order to assess the three completion procedures, entries were artificially removed from the tensor data. The NRMSE results are reported in Table 3.6 for increasing values of the fraction of sampled entries and two different values of N.

Table 3.6: MSI case study - NRMSE

	N=50			
fraction of entries	0.1	0.3	0.5	0.7
frame	0.123	0.110	0.108	0.106
tensor	0.129	0.071	0.052	0.036
mode-3	**0.058**	**0.021**	**0.016**	**0.014**

	N=4			
fraction of entries	0.1	0.3	0.5	0.7
frame	0.172	0.147	0.149	0.148
tensor	**0.171**	**0.137**	**0.131**	**0.125**
mode-3	0.180	0.160	0.149	0.137

Hyperspectral Images of Natural Scenes

This example considers natural images that were previously used in [84]. In a first part we focused on scene 6 (Braga) and 7 (Ribeira). The Matlab reflectance files for these scenes can be downloaded from:

`http://personalpages.manchester.ac.uk/staff/david.foster/`.

Details on the hyperspectral system and processing are given at the referred webpage and in [84].

First Part Each scene originally consisted of 33 frames obtained varying the peak-transmission wavelength from 400nm to 720nm. We down-sampled each frame independently to 203×268 for Ribeira and 204×267 for Braga. Successively we performed completion for increasing fraction of missing entries. On one hand we considered the data tensor consisting of the whole set of frames. On the other hand, we used the ensemble consisting of the first 5 wavelengths only as the data tensor. This case is considered to simulate the situation where the information along the spectral mode is limited. With reference to this second case Figure 3.5 shows the plot of a data frame (in black the missing

Table 3.7: First part of the HRS case study - NRMSE

Ribeira (33 frames)				
fraction of entries	0.1	0.3	0.5	0.7
frame	0.092	0.061	0.0455	0.035
tensor	0.072	0.031	0.020	0.013
mode-3	**0.068**	**0.018**	**0.012**	**0.009**

Ribeira (5 frames)				
fraction of entries	0.1	0.3	0.5	0.7
frame	0.071	0.046	0.034	0.026
tensor	**0.067**	**0.034**	**0.023**	**0.016**
mode-3	0.191	0.119	0.070	0.036

Braga (33 frames)				
fraction of entries	0.1	0.3	0.5	0.7
frame	0.093	0.064	0.052	0.040
tensor	0.069	0.033	0.022	0.013
mode-3	**0.05**	**0.014**	**0.010**	**0.008**

Braga (5 frames)				
fraction of entries	0.1	0.3	0.5	0.7
frame	0.091	0.066	0.051	0.040
tensor	**0.083**	**0.048**	**0.033**	**0.023**
mode-3	0.187	0.114	0.062	0.031

entries) and the corresponding reconstructed images. The performance of the different completion techniques is reported in Table 3.7.

Table 3.8: Mixture of images - NRMSE

	fraction of entries			
fraction of entries	0.1	0.3	0.5	0.7
frame	**0.038**	**0.029**	**0.023**	**0.025**
tensor	0.046	0.031	0.024	0.026
mode-3	0.091	0.077	0.067	0.077

Second Part As a second experiment we looked into the case where the spectral mode fails to satisfy the redundancy assumption. Specifically we

considered the whole set of 8 spectral images available at the referred webpage. The first frame of each image was down-sampled to 102×134 and stacked along the third mode of a new data tensor. The results for this case are reported in Table 3.8. Contrary to alternative approaches, tensor completion exploits redundancy along all the modes of spectral data. Our experiments (Table 3.6 and 3.7) show that, when the number of spectral bands is limited, exploiting the structure of all the modes improves the imputation of missing entries. In contrast, if the number of bands is sufficiently high, the experiments show that working in the third mode only is more advantageous. Finally when no correlation along the spectral mode is present relying solely on low ranks assumptions of the frames performs best (Table 3.8).

Remark. Contrary to alternative approaches, tensor completion exploits redundancy along all the modes of spectral data. Our experiments (Table 3.6 and 3.7) show that, when the number of spectral bands is limited, exploiting the structure of all the modes improves the imputation of missing entries. In contrast, if the number of bands is sufficiently high, the experiments show that working in the third mode only is more advantageous. Finally when no correlation along the spectral mode is present relying solely on low ranks assumptions of the frames performs best (Table 3.8).

3.6.3 Inductive Learning

We report here experiments on inductive learning with multiple tasks, see Section 3.5. As performance indicator we considered the capability of each procedure to predict the correct test labels. We compared LS-SVM models with linear kernel (lin ls-svm) and naive Bayes classifiers (naive Bayes) [71] [11] with models obtained solving (3.81). With reference to the latter we set the parameters in the composite spectral penalty as follows. In one case, referred to as log mlrank[12], we set

$$\lambda_m = \frac{1}{M+1}\bar{\lambda} \text{ for any } m \in \mathbb{N}_{M+1} \tag{3.89}$$

where M is the order of the input data and $\bar{\lambda}$ is a varying parameter. Alternatively we take

$$\lambda_m = \begin{cases} 0 & \text{if } m \in \mathbb{N}_M \\ \bar{\lambda} & \text{if } m = M+1 \ . \end{cases} \tag{3.90}$$

[11] Specifically we considered the routine NaiveBayes contained in the Statistics Toolbox of Matlab.

[12] We use log as a short-hand for *logistic* since (3.81) is based on the logistic loss; we use mlrank as a short-hand for *multilinear rank* since in this case we penalize high multilinear rank of the tensor corresponding to the ensemble of models.

This latter approach, referred to as log rank, corresponds to leverage only the interdependence between tasks; structural assumptions over the input features are not exploited. In either case, as in Section 3.6.1, we compute the entire regularization path with respect to $\bar{\lambda}$. In our experiments the choice of this parameter was driven by the misclassification rate. For log mlrank and log rank we use a hold-out validation set. For LS-SVM we found models via the LS-SVMlab toolbox [63] and considered ten-fold CV for model selection.

Multi-labeled Data Toy Problems

We begin by a set of artificially generated problems. For each trial we considered an ensemble of T models represented by a $(D \times D \times T)$-tensor $\tilde{\mathcal{W}}$ with multilinear rank (r, r, r) and a vector of bias terms $b \in \mathbb{R}^T$ with normal entries; $\tilde{\mathcal{W}}$ was obtained by generating a core tensor $\mathcal{S}_{\tilde{\mathcal{W}}}$ in $\mathbb{R}^r \otimes \mathbb{R}^r \otimes \mathbb{R}^r$ with entries i.i.d. from a normal distribution; for $i \in \{1, 2\}$ a matrix $\boldsymbol{U}_i \in \mathbb{R}^D \otimes \mathbb{R}^r$ and $\boldsymbol{U}_3 \in \mathbb{R}^T \otimes \mathbb{R}^r$ were generated also with entries i.i.d. from a normal distribution. Finally we set

$$\tilde{\mathcal{W}} = \mathcal{S}_{\tilde{\mathcal{W}}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times \boldsymbol{U}_3 \ . \tag{3.91}$$

We then created a dataset of N input-output pairs as follows. For $n \in \mathbb{N}_N$ we let $\mathcal{X}^{(n)}$ be a $(D \times D)-$matrix with normal entries; for any $t \in \mathbb{N}_T$ we let the corresponding label $y_t^{(n)}$ be a Bernoulli random variable with alphabet $\{1, -1\}$ and success probability $1/(1 + \exp(-\tilde{y}_t^{(n)}))$; the latent variable $\tilde{y}_t^{(n)}$ was taken to be

$$\tilde{y}_t^{(n)} = \langle \tilde{\mathcal{W}}^{(t)}, \mathcal{X}^{(n)} \rangle + b_t$$

where for any $t \in \mathbb{N}_T$, $\mathcal{W}^{(t)}$ is as in (3.78).

We set $r = 2$ and considered the procedure above for different values of T and D. For a fixed value of T we use $N = 100T$ pairs for training. Note that N refers to the whole set of tasks; in turn, the N pairs were distributed uniformly at random across different tasks. As such, there are on average 100 input-output pairs per task; in this way, the amount of training information is kept constant as T varies. For each setting we perform 10 trials. For log mlrank and log rank we chose $\bar{\lambda}$ based upon a validation set of 30% pairs selected at random within the N observations. Results in terms of misclassification rate on a test set are reported in Table 3.9. The training times required by the different procedures for the case where $T = 8$ and $N = 800$ are reported in Table 3.10.

Remark. The experiments show that exploiting relations between tasks (log mlrank and log rank) significantly improves results; the performance of ls-svm models, which are trained independently, is the same as T is increased. This is to be expected since the amount of training data per task is kept approximately the same across different values of T.

Table 3.9: Fractions of unobserved labels (test data) predicted incorrectly

D	T	N	log mlrank	log rank	lin ls-svm	naive Bayes
5	2	200	**0.08**(0.03)	0.11(0.03)	0.13(0.02)	0.23(0.02)
	4	400	**0.07**(0.02)	0.08(0.01)	0.13(0.01)	0.22(0.01)
	8	800	**0.06**(0.01)	0.07(0.01)	0.14(0.01)	0.22(0.01)
15	2	200	**0.22**(0.03)	0.31(0.02)	0.34(0.01)	0.39(0.01)
	4	400	**0.19**(0.05)	0.28(0.03)	0.35(0.01)	0.39(0.01)
	8	800	**0.17**(0.02)	0.22(0.01)	0.35(0.01)	0.39(0.01)

Table 3.10: Mean (and standard deviation) of training times (including model selection, values in seconds) required by the different procedures for $T = 8$ and $N = 800$. Note that naive Bayes and lin ls-svm actually solve 8 small and separate problems, each of which based on a training dataset of about 100 patterns. In contrast, log mlrank and log rank solve simultaneously all the tasks at once using the whole set of 800 patterns.

D	log mlrank	log rank	lin ls-svm	naive Bayes
5	428.46(49.30)	138.52(21.65)	**0.04**(0.01)	0.08(0.03)
15	734.19(69.48)	174.05(15.57)	0.29(0.02)	**0.17**(0.03)

Remark. A comparison between log mlrank and log rank reveals that exploiting structural assumptions over the input features is a good idea; this is seen to be the case especially when the number of tasks is small or the features dimensionality is higher.

Multiclass classification of Binary Alphadigits

In this experiment we considered discrimination between handwritten digits. We focused on the Binary Alphadigits dataset[13] made up of digits from "0" through "9" followed by capital letters from "A" through "Z" (English alphabet). Each digit is represented by 39 examples each of which consists of a binary 20×16 matrix. In log mlrank the matrix shape of each digit is retained; log rank and lin ls-svm treat each input pattern as a vector of dimension 320. We consider problems with different numbers of classes; each class corresponds to a digit. As before we used one-vs-one encoding (see Subsection 3.6.1);

[13]Publicly available at `http://cs.nyu.edu/~roweis/data.html`.

Table 3.11: Fractions of misclassified test digits, multiple problems, $T = 2$

log mlrank	log rank	lin ls-svm
0.05(0.04)	0.07(0.04)	0.06(0.04)

log mlrank	log rank	lin ls-svm
0.12(0.05)	0.13(0.05)	0.14(0.04)

log mlrank	log rank	lin ls-svm
0.10(0.05)	0.11(0.04)	**0.10**(0.04)

log mlrank	log rank	lin ls-svm
0.07(0.05)	0.08(0.05)	**0.07**(0.03)

log mlrank	log rank	lin ls-svm
0.06(0.05)	0.07(0.05)	0.07(0.05)

log mlrank	log rank	lin ls-svm
0.15(0.05)	0.16(0.06)	**0.14**(0.07)

Table 3.12: Fractions of misclassified test digits, $T > 2$.

T	N	log mlrank	log rank	lin ls-svm
4	40	**0.07**(0.03)	0.08(0.03)	0.08(0.03)
8	80	**0.17**(0.04)	**0.17**(0.04)	0.22(0.04)
12	120	**0.24**(0.03)	**0.24**(0.03)	0.38(0.04)
16	160	**0.27**(0.02)	**0.27**(0.03)	0.51(0.02)

correspondingly, the number of tasks T is equal to the number of classes. In each case we train models upon N training examples uniformly distributed across the considered classes; we chose N so that approximately 10 examples per class are used for training (and model selection) whereas the remaining examples are retained for testing. For each setting we average results over 20 trials each of which is obtained from a random splitting of training and test data. Due to the scarcity of training patterns an error occurs when running NaiveBayes; therefore we could not obtain results for this approach. Tables 3.11 and 3.12 report results for different values of T. For $T = 2$ we considered a subset of arbitrary binary problems; for $T > 2$ we considered classes of digits in their given order. In general, for $T \leq 4$ log mlrank seems to perform

slightly better than log rank (Tables 3.11 and 3.12). As for the multi-labels example above, there is a strong evidence that enforcing task relationships via the regularization mechanism in log mlrank and log rank improves over the case where tasks are considered independently (Table 3.12).

3.7 Chapter Summary

In this Chapter we have presented novel tensor-based formulations for transductive and inductive learning. These formulations allow one to exploit the multimodal nature of data as well as common features among learning tasks. We have worked with affine models that ensure a-posteriori interpretability. A common template algorithm based on an inexact splitting technique was used to compute solutions; this was achieved by recasting each problem as a special instance of a general convex problem. Regularization was based on extending to tensors the concept of spectral penalties for matrices. The experiments show that exploiting the structure of data with natural tensorial representation leads to better learning rates in comparison with techniques that neglect multimodal structures.

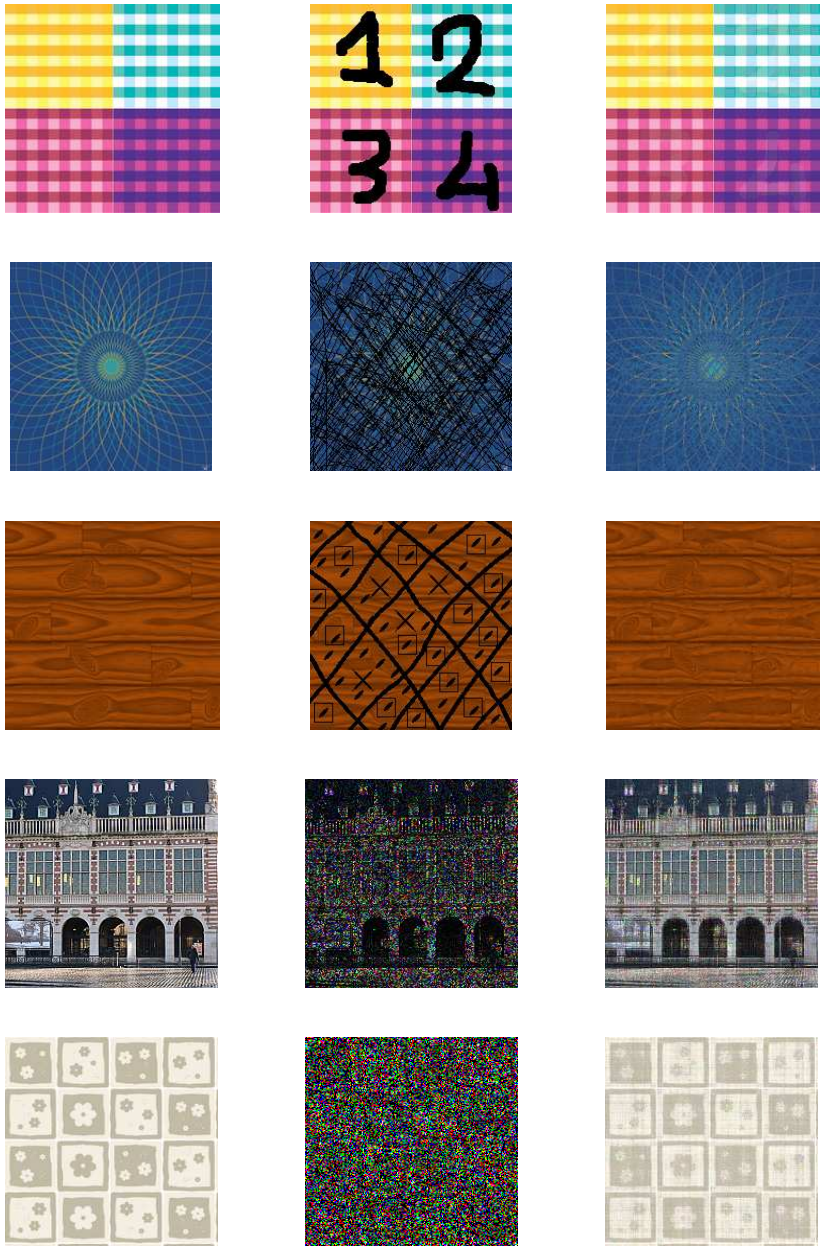


Figure 3.2: Original images (left column); given pixels (middle column); reconstruction by Algorithm 2 (right column).

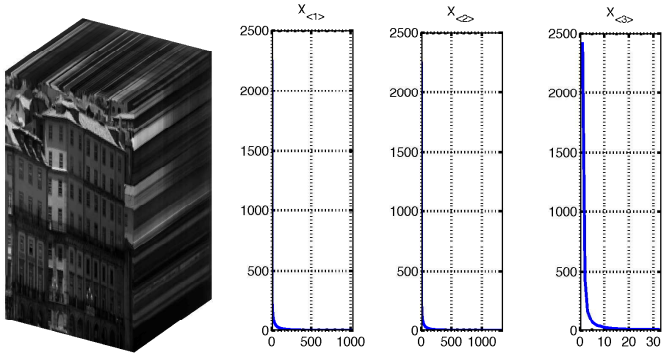
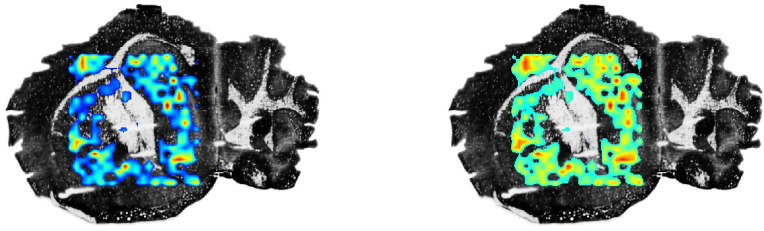
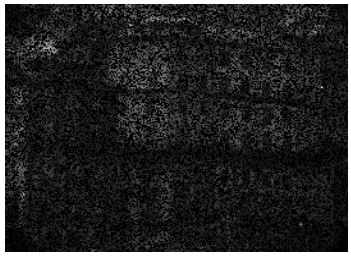


Figure 3.3: An hyperspectral image represented as a $1017 \times 1340 \times 33$ tensor $\mathcal{X}$ along with the spectrum of the corresponding matrix unfoldings.



(a) m/z=2.80e3 (b) m/z=3.03e3

Figure 3.4: 4.5(a) and 4.5(b): intensity plots corresponding to two different m/z bins. The square box of 22×22 measurement locations (pixels) is overlaid on the registered tissue section. Note that the the two intensity plots are not too dissimilar; in particular, the peaks of intensities (identified by warm colors) are localized in the same areas.



(a) Data — Ribeira @ 400nm



(b) Tensor



(c) Frame



(d) mode-3

Figure 3.5: Completion of the first frame in the ensemble given by the first 5 wavelengths of Ribeira. Approximately 50% of entries are missing.

Chapter 4

Kernel-based Learning for Data Tensors

Tensor-based techniques for learning allow one to exploit the structure of carefully chosen representations of data. As we discussed in Chapter 2, this is a desirable feature in particular when the number of training patterns is small. Parametric tensor-based techniques, such as those discussed in the last Chapter, are based on affine models of the type:

$$f(\mathcal{X}; \mathcal{W}, b) = \langle \mathcal{W}, \mathcal{X} \rangle + b \ . \tag{4.1}$$

As we have seen, the use of spectral penalties enforce the parsimony of the estimated models while preserving convexity of the optimization problems used for training. However, affine functions constitute a somewhat restricted class, which might suffer from limited predictive power. This is true, in particular, when the task at hand requires to introduce a notion of locality. The goal of this Chapter is therefore to go beyond the limitation of affine models by introducing nonparametric tensor-based models. To this end we briefly discuss in Section 4.1 a first idea, based on an explicit multi-way feature representation, which application is hindered by non-convexity; we then elaborate in Section 4.2 on an effective framework based on a suitably defined class of kernel functions.

4.1 Learning with Explicit Feature Representations

The first approach that we discuss was introduced in [181] and aims at relaxing linearity of standard tensor-based models while still exploiting the

structural information embodied in the data. Recall that standard kernel methods correspond to (implicitly) map data represented as vectors into high dimensional feature vectors. In comparison, the methodology detailed in this Section corresponds to mapping input data tensors into high dimensional feature tensors that carry information embodied in the original structure. We first explain how the feature representation of higher order data is obtained; successively we elaborate on how to use this feature representation to estimate nonparametric models based on penalized empirical risk minimization.

4.1.1 Input Data Matrices

As in [181], for simplicity of presentation, we consider the case of second order tensors. Therefore we shall assume that each input observation is actually an input data matrix $\boldsymbol{X} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2}$. We are interested in defining predictive models that depends nonlinearly from $\boldsymbol{X}$ and take advantage of its spectral information. Recall that for $D = \min\{D_1, D_2\}$ the thin SVD [94] of a point $\boldsymbol{X}$ is defined as the factorization

$$\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^\top \tag{4.2}$$

where $\boldsymbol{U} \in \mathbb{R}^{D_1 \times D} = [u_1, u_2, \ldots, u_D]$ and $\boldsymbol{V} \in \mathbb{R}^{D_2 \times D} = [v_1, v_2, \ldots, v_D]$ satisfy $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_D$ and $\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}_D$ respectively and $\Sigma \in \mathbb{R}^D \otimes \mathbb{R}^D$ has its only nonzero elements on the first $r = \text{rank}(\boldsymbol{X})$ entries along the main diagonal. These elements are the ordered singular values

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

whereas columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ are called respectively left and right singular vectors. Equivalently

$$\boldsymbol{X} = \sum_{i \in \mathbb{N}_r} \sigma_i u_i \otimes v_i \tag{4.3}$$

where $u_i \otimes v_i$ are rank-1 operators of the type (2.22) and the set $\{u_i : i \in \mathbb{N}_r\}$ and $\{v_i : i \in \mathbb{N}_r\}$ span respectively the column space $\mathscr{R}(\boldsymbol{X})$ and the row space $\mathscr{R}(\boldsymbol{X}^\top)$.

4.1.2 Auxiliary Operators and Operatorial Equation

Let $\phi_1 : \mathbb{R}^{D_1} \to \mathscr{H}_1$ and $\phi_2 : \mathbb{R}^{D_2} \to \mathscr{H}_2$ be *feature maps* associated to reproducing kernels (see Appendix B). Based upon $\{u_i : i \in \mathbb{N}_r\}$ and $\{v_i : i \in \mathbb{N}_r\}$ we introduce the *mode*-1 *auxiliary operator* $\Gamma_U : \mathscr{H}_1 \to \mathbb{R}^{D_1}$

and the *mode−2 auxiliary operator* $\Gamma_V : \mathscr{H}_2 \to \mathbb{R}^{D_2}$ defined, respectively, by

$$\Gamma_U h = \sum_{i \in \mathbb{N}_r} \langle \phi_1(u_i), h \rangle_{\mathscr{H}_1} u_i \quad \text{and} \quad \Gamma_V h = \sum_{i \in \mathbb{N}_r} \langle \phi_2(v_i), h \rangle_{\mathscr{H}_1} v_i \ . \tag{4.4}$$

In order to proceed we need the concepts of Hilbert-Schmidt operators and Kronecker products. Recall from Appendix A.4, that by $\mathcal{H}_1 \otimes \mathcal{H}_2$ we denote the space of *Hilbert-Schmidt*[1] operators $X : \mathcal{H}_2 \to \mathcal{H}_1$. Let $A : \mathcal{H}_1 \to \mathcal{G}_1$ and $B : \mathcal{H}_2 \to \mathcal{G}_2$ be bounded Hilbert-Schmidt operators between HSs and suppose $X \in \mathcal{H}_1 \otimes \mathcal{H}_2$. The linear operator $X \to AXB^*$ is a mapping from $\mathcal{H}_1 \otimes \mathcal{H}_2$ into $\mathcal{G}_1 \otimes \mathcal{G}_2$. It is called Kronecker product between the *factors* A and B and denoted as $A \otimes B$.

Denote now by $\Gamma_U \otimes \Gamma_V$ the Kronecker product between Γ_U and Γ_V, $\Gamma_U \otimes \Gamma_V : \mathscr{H}_1 \otimes \mathscr{H}_2 \to \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2}$. Under the assumption that $\boldsymbol{X} \in \mathscr{R}(\Gamma_U \otimes \Gamma_V)$ we define $\Phi_X \in \mathscr{H}_1 \otimes \mathscr{H}_2$ by

$$\Phi_X := \arg\min \left\{ \|\Psi_X\|^2_{\mathscr{H}_1 \otimes \mathscr{H}_2} \ : \ (\Gamma_U \otimes \Gamma_V)\Psi_X = \boldsymbol{X}, \ \Psi_X \in \mathscr{H}_1 \otimes \mathscr{H}_2 \right\} \ . \tag{4.5}$$

This way $\boldsymbol{X}$ is associated to a minimum norm solution of an operatorial equation. The idea is illustrated in Figure 4.1. Notice that the range $\mathscr{R}(\Gamma_U \otimes \Gamma_V)$ is closed in the finite dimensional space $\mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2}$. Hence a solution of Φ_X exists as soon as $\boldsymbol{X} \in \mathscr{R}(\Gamma_U \otimes \Gamma_V)$. Before continuing, we recall a fundamental Theorem that we report here for completeness.

Theorem 4.1.1. *[136, Theorem 1, Chapter 6.1] Let $(\mathscr{C}, \langle \cdot, \cdot \rangle_{\mathscr{C}})$ and $(\mathscr{D}, \langle \cdot, \cdot \rangle_{\mathscr{D}})$ be HS and suppose $A : \mathscr{C} \to \mathscr{D}$ is linear and bounded with $\mathscr{R}(A)$ closed in $\mathscr{D}$. Then*

$$x = \arg\min \left\{ \|x\|^2_{\mathscr{C}} \ : \ Ax = y, \ x \in \mathscr{C} \right\}$$

is unique and given by $x = A^* z$ where z is any solution (possibly not unique) of $AA^* z = y$.

4.1.3 Feature Representation of Data Matrices

Based upon this result we now specify how to obtain Φ_X, the feature representation of an input data matrix $\boldsymbol{X}$.

Theorem 4.1.2. *Let $A_U : \mathscr{H}_1 \to \mathbb{R}^r$ and $B_V : \mathscr{H}_2 \to \mathbb{R}^r$ be defined entry-wise as $(A_U h)_i = \langle \phi_1(u_i), h \rangle$ and $(B_V h)_i = \langle \phi_2(v_i), h \rangle$ respectively. The unique solution Φ_X of (4.5) is then given by*

$$\Phi_X = A_U^* \boldsymbol{Z} B_V \tag{4.6}$$

[1] In a nutshell, well behaved infinite dimensional operators, see A.4 for technical details.

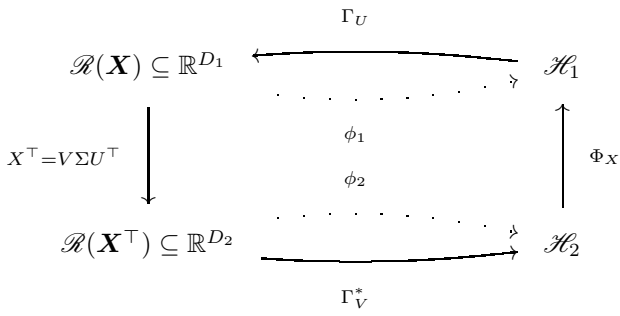


Figure 4.1: A diagram illustrating the different spaces and mappings we have introduced. The operator $\Phi_X \in \mathscr{H}_1 \otimes \mathscr{H}_2$ is the feature representation of interest.

where $\boldsymbol{Z} \in \mathbb{R}^r \otimes \mathbb{R}^r$ *is any solution of*

$$\boldsymbol{K}_U \boldsymbol{Z} \boldsymbol{K}_V = \Sigma \tag{4.7}$$

where $(\boldsymbol{K}_U)_{ij} = \langle \phi_1(u_i), \phi_1(u_j) \rangle$ *and* $(\boldsymbol{K}_V)_{ij} = \langle \phi_2(v_i), \phi_2(v_j) \rangle$.

Proof. Recall from Appendix A.4 that $\mathscr{H}_1 \otimes \mathscr{H}_2$ is an Hilbert space with the inner-product (A.23). Notice that $(\Gamma_U \otimes \Gamma_V)\Phi_X = \boldsymbol{X}$ can be restated as

$$(A_U \otimes B_V)\Phi_X = \Sigma \tag{4.8}$$

where $A_U = \boldsymbol{U}^\top \Gamma_U$ is defined entry-wise as $(A_U h)_i = \langle h, \phi(u_i) \rangle$ and similarly we have $B_V = \Gamma_V \boldsymbol{V}$ where $(B_V h)_i = \langle h, \phi(v_i) \rangle$. Further observe that $(A_U \otimes B_V)$ is linear, bounded and being finite dimensional $\mathscr{R}(A_U \otimes B_V)$ is closed. Theorem 4.1.1 reads now

$$\Phi_X = (A_U \otimes B_V)^* \boldsymbol{Z}$$

for any $\boldsymbol{Z}$ such that

$$(A_U \otimes B_V)(A_U \otimes B_V)^* \boldsymbol{Z} = \Sigma \ .$$

Now by properties of the Kronecker product

$$(A_U \otimes B_V)^* = A_U^* \otimes B_V^* \text{ and } (A_U \otimes B_V)(A_U^* \otimes B_V^*) = A_U A_U^* \otimes B_V B_V^*$$

where $A_U A_U^*,\ B_V B_V^* : \mathbb{R}^r \otimes \mathbb{R}^r \to \mathbb{R}^r \otimes \mathbb{R}^r$ have matrix representation

$$(\boldsymbol{K}_U)_{ij} = \langle \phi_1(u_i), \phi_1(u_j) \rangle_{\mathscr{H}_1} \text{ and } (\boldsymbol{K}_V)_{ij} = \langle \phi_1(v_i), \phi_1(v_j) \rangle_{\mathscr{H}_2}$$

respectively. Hence $(A_U \otimes B_V)(A_U \otimes B_V)^* \boldsymbol{Z} = \Sigma$ is equivalent to (4.7). □

Corollary 4.1.3. *Suppose* $\boldsymbol{Z} = \sum_{i \in \mathbb{N}_r} \rho_i q^i \otimes w^i$ *is a SVD factorization*[2] *of a solution of* (4.7). *Then*

$$\Phi_X = \sum_{i,j,l \in \mathbb{N}^r} \rho_i q_j^i w_l^i \phi_1(u_j) \otimes \phi_2(v_l) \tag{4.9}$$

and hence Φ_X *is a finite-rank operator (see Appendix A.2.7) with rank at most equal to* r.

Proof. The adjoint operators $A_U^* : \mathbb{R}^r \to \mathscr{H}_1$ can be written as

$$A_U^* q = \sum_{j \in \mathbb{N}^r} q_j \phi_1(u_j) \ .$$

Indeed we see that

$$\langle A_U^* q, h \rangle_{\mathscr{H}_1} = \langle q, A_U h \rangle = \sum_{i \in \mathbb{N}^r} q_j \langle \phi_1(u_j), h \rangle_{\mathscr{H}_1} \ .$$

Similarly we can obtained the closed form of B_V^*. The result is now a consequence of the previous Theorem. □

Remark. When the *polynomial kernel* [192] is used, namely we have:

$$k_1(x, y) = k_2(x, y) = k(x, y) = (\langle x, y \rangle)^d$$

for some integer $d \geq 1$, one can gain further insight. Suppose this type of kernel is employed and let ϕ denote the corresponding feature maps in (4.4). Then it is not difficult to see that $\boldsymbol{K}_U$ and $\boldsymbol{K}_V$ are identity matrices, $\boldsymbol{Z} = \Sigma$ and

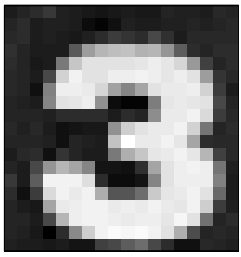
$$\Phi_X = \sum_{i \in \mathbb{N}_r} \sigma_i \phi(u_i) \otimes \phi(v_i) \ .$$

In particular when $d = 1$, $\phi(x) = x$ and the latter corresponds to the usual SVD of $\boldsymbol{X}$. Figure 4.2 shows an example for $d = 2$.

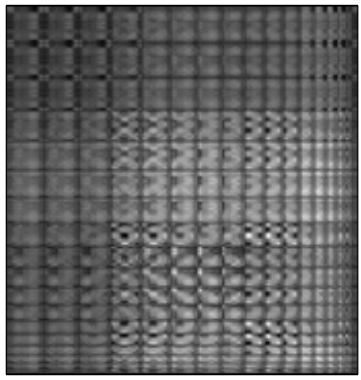
Remark. The rank-1 elements in (4.9) are not mutually orthogonal. In spite of this we can have a rough idea of the energy of Φ_X in the case the Gaussian-RBF kernels are used. Indeed recall that in this case $\|\phi(x)\| = 1$ for any x. Then by simply applying the triangular inequality we have

$$\|\Phi_X\| = \| \sum_{i,j,l \in \mathbb{N}^r} \rho_i q_j^i w_l^i \phi(u_j) \otimes \phi(v_l) \| \leq \sum_{i,j,l \in \mathbb{N}^r} |\rho_i q_j^i w_l^i| \ .$$

[2]Note that the result actually holds for any rank-1 decomposition.



(a) A 19×18 grayscale image X of a character taken from a natural scene.



(b) Its 190×171 feature representation Φ_X.

Figure 4.2: An image (a) and its feature representation (b). Here 2−degree polynomial kernels were used (see Remark 4.1.3) to generate the operators in (4.4) and successively Φ_X was found based upon (4.6) and (4.7).

4.1.4 Model Estimation

The idea that we presented above amounts at representing an input data matrix as a finite rank operator acting between infinite dimensional Hilbert spaces. This goes in the same spirit as in standard kernel methods where one implicitly maps a finite dimensional vector (or vectorized object) into a possibly infinite dimensional Hilbert space. However, our approach additionally retains the *operatorial nature* of the original input observation[3]. We now discuss how this feature representation can be used to define nonparametric models estimated via penalized empirical risk minimization. The idea follows the same general regularization approach introduced in Section 2.2. Unfortunately, however, it results into a non-convex procedure which suffers from local minima.

[3]Two-way tensors are indeed representers of linear operators. Higher order tensors are associated to multiple operators via unfolding.

Learning Based on Explicit Feature Representation

Assume that we are given a dataset consisting of input-target pairs

$$\mathscr{D}_N = \left\{ \left(\boldsymbol{X}^{(n)}, \boldsymbol{Y}^{(n)} \right) \in \mathscr{X} \times \mathscr{Y} \ : \ n \in \mathbb{N}_N \right\} \tag{4.10}$$

where $\mathscr{X} \subset \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2}$ and $\mathscr{Y} \subset \mathbb{R}^{L_1} \otimes \mathbb{R}^{L_2}$. The situation where $\mathscr{Y} \subset \mathbb{R}^L$ or simply $\mathscr{Y} \subset \mathbb{R}$ is clearly a special case of this framework. Our goal is to find a predictive operator

$$F : \mathscr{X} \to \mathscr{Y} \ . \tag{4.11}$$

This objective defines different type of problems. For the case of linear tensor models, we have $F = \boldsymbol{A} \otimes \boldsymbol{B}$ namely

$$F : \boldsymbol{X} \mapsto \boldsymbol{A}\boldsymbol{X}\boldsymbol{B}^\top \ . \tag{4.12}$$

This is the type of models considered, for example, in [105]. Their problem is unsupervised and amounts to find a pair of matrices $\boldsymbol{A} \in \mathbb{R}^{L_1 \times D_1}$ and $\boldsymbol{B} \in \mathbb{R}^{L_2 \times D_2}$ such that the mapping $\boldsymbol{X} \mapsto \boldsymbol{A}\boldsymbol{X}\boldsymbol{B}^\top$ constitutes a structure-preserving projection onto a lower dimensional space $\mathbb{R}^{L_1 \times L_2}$.

Remark. The case where $\mathscr{Y} \subset \mathbb{R}$, comprises standard classification ($\mathscr{Y}$ is a countable set of labels) or regression ($\mathscr{Y}$ is a non-countable subset of $\mathbb{R}$) problems. In this case (4.12) actually reads

$$F : \boldsymbol{X} \mapsto a^\top \boldsymbol{X} b$$

where $a \in \mathbb{R}^{D_1}$ and $b \in \mathbb{R}^{D_2}$. Note that we can equivalently write the latter as

$$F(\boldsymbol{X}) = \langle a \otimes b, \boldsymbol{X} \rangle \ ,$$

which reveals that F is indeed a special instance of (4.1).

On the other hand, for the nonlinear case we have

$$F = (A \otimes B) \circ \Phi \tag{4.13}$$

where $\circ$ denotes composition, $\Phi : \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \to \mathscr{H}_1 \otimes \mathscr{H}_2$ is a fixed $2-$way feature map of the type introduced in the previous section and $A \otimes B : \mathscr{H}_1 \otimes \mathscr{H}_2 \to \mathbb{R}^{L_1} \otimes \mathbb{R}^{L_2}$ is the actual unknown. Since in this case

$$F : \boldsymbol{X} \mapsto A\Phi(\boldsymbol{X})B^* \tag{4.14}$$

defines, in general, a nonlinear model in $\boldsymbol{X}$, we can account for possible nonlinearities.

Remark. Note that (4.14) includes (4.12) as a special case.

Penalized Empirical Risk Functional

The problem of finding $A \otimes B$ can be tackled by minimizing the penalized empirical risk functional of the type [181]:

$$\min_{\substack{A : \mathscr{H}_1 \to \mathbb{R}^{L_1} \\ B : \mathscr{H}_2 \to \mathbb{R}^{L_2}}} \left\{ \sum_{n \in \mathbb{N}_N} l(\boldsymbol{Y}^{(n)}, (A \otimes B)\Phi(\boldsymbol{X}^{(n)})) + \frac{\lambda}{2}(\|A\|\|B\|)^2 \right\} \quad (4.15)$$

where $l : (\mathbb{R}^{L_1} \otimes \mathbb{R}^{L_2}) \times (\mathbb{R}^{L_1} \otimes \mathbb{R}^{L_2}) \to \mathbb{R}^+$ is a loss function and λ is a user defined parameter.

Problem (4.15) is non-convex. It can be shown that the estimation of A and B amounts at finding $L_1 + L_2$ functions. Practical algorithms can be derived based on a Representer Theorem [181]. These algorithms require to alternatively fix one of the unknown operators, say A (B), and find the L_2 (L_1) functions that correspond to the other. At each step, computing a solution in the free factor can be done relying on finite dimensional optimization.

Steps Required for Model Estimation

In practice the steps required to derive nonparametric models based upon an explicit 2-way feature representation are as follows:

1. Given the training dataset (4.10) for pattern indices $n_1, n_2 \in \mathbb{N}_N$ compute the kernel matrices defined entry-wise by

   $$(\boldsymbol{K}_U^{n_1,n_2})_{ij} = \langle \phi_1(u_i^{n_1}), \phi_1(u_j^{n_2}) \rangle \ , \quad (\boldsymbol{K}_V^{n_1,n_2})_{ij} = \langle \phi_2(v_i^{n_1}), \phi_2(v_j^{n_2}) \rangle$$

   where

   $$\boldsymbol{X}^{(n)} = \boldsymbol{V}^{(n)} \Sigma^{(n)} \boldsymbol{U}^{(n)}$$

   is the SVD for the $n-$pattern and for $D = \min\{D_1, D_2\}$,

   $$\boldsymbol{U}^{(n)} = [u_1^n, u_2^n, \ldots, u_D^n] \text{ and } \boldsymbol{V}^{(n)} = [v_1^n, v_2^n, \ldots, v_D^n] \ .$$

2. For pattern index $n \in \mathbb{N}_N$ solve in $\boldsymbol{Z}^{(n)}$ (see (4.7))

   $$\boldsymbol{K}_U^{n,n} \boldsymbol{Z}^{(n)} \boldsymbol{K}_V^{n,n} = \Sigma^{(n)} \ .$$

3. Solve (4.15) via an alternating scheme that uses $\boldsymbol{K}_U^{n_1,n_2}$, $n_1, n_2 \in \mathbb{N}_N$ and $\boldsymbol{Z}^{(n)}$, $n \in \mathbb{N}_N$ found at step 1 and 2 [4].

[4] A complete algorithm for the case where $L_1 = L_2 = 1$ is given in [181].

4.1.5 Experiments

In performing experiments we focused on small scale problems. We compared a standard (vectorized) kernel approach versus the nonlinear tensor method presented above. Both the type of kernel matrices computed at step 2 (see page 96) were constructed upon the Gaussian RBF-kernel with the same value of width parameter. As standard kernel method we consider LS-SVM [199] also trained with a Gaussian RBF-kernel. We do not consider a bias term as this is not present in our tensor-based model either. In both cases we took a 20×20 grid of kernel width and regularization parameter and performed model selection via leave-one-out cross-validation (LOO-CV).

Robot Execution Failures

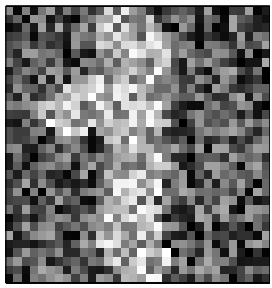
Each input data point is here a 15×6 multivariate time-series where columns represent a force or a torque. The task we considered was to discriminate between two operating states of the robot, namely *normal* and *collision_in_part*. Within the 91 observations available, n were used for training and the remaining $n-91$ for testing. We repeated the procedure over 20 random splittings of training and test set. Averages (with standard deviation in parenthesis) of Correct classification rates (CCR) of models selected via LOO-CV are reported in Table 4.1 for different numbers n of training points. Best results are highlighted.

Table 4.1: Test results for the Robot Execution Failures Data Set.

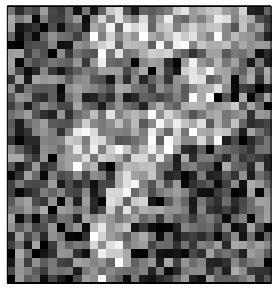
Correct Classification rates				
	n=5	n=10	n=15	n=20
RBF-LS-SVM	0.55(0.06)	0.64 (0.08)	0.66 (0.08)	0.70(0.06)
RBF-Tensor	**0.62(0.07)**	**0.66(0.08)**	**0.68(0.10)**	**0.71(0.11)**

Optical Recognition of Handwritten Digits

Here we considered recognition of handwritten digits. We took 50 bitmaps of size 32×32 of handwritten 7s and the same number of 1s and added noise to make the task of discriminating between the two classes more difficult (Figure 4.3(a) and 4.3(b)). We followed the same procedure as for the previous example and report results in Table 4.2.



(a) A noisy 1



(b) A noisy 7

Correct Classification rates			
RBF-LS-SVM		RBF-Tensor	
n=5	n=10	n=5	n=10
0.71(0.20)	0.85 (0.14)	**0.84 (0.12)**	**0.88(0.09)**

Figure 4.3 & Table 4.2: Instances of handwritten digits with high level of noise ((a) and (b)) and CCR on test for different number n of training points.

4.2 Learning with Kernels for Data Tensors

Exploiting tensor models in linear tensor-based learning has been shown to be particularly helpful when the number of training points is limited [105]. Our experiments on small scale problems seem to suggest that this still holds for nonparametric 2-ways models. However the practical application of the approach described in Subsection 4.1 is hindered by its computational complexity. Besides, non-convexity of the training problem means that we do not have any guarantee of global optimality of the solution found. These features harm performance as soon as the number of training patterns increases. In contrast, any "well-behaved" learning algorithm is expected to show better performance when the number of training samples increases. In the following we present an alternative approach that does not present these drawbacks. The idea relies on a kernel function that effectively exploits the algebraic content of data tensors. In recent years specialized kernels have been designed for a number of structured data [87, 88]. However, no existing approach exploits the (algebraic) structure of tensorial representations. We go beyond this limitation by proposing a class of tensorial kernels that links to the MLSVD and features an interesting invariance property.

4.2.1 Naive Kernels for Tensors

Notice that Proposition B.1.1 and Theorem B.1.5 in Appendix B.1 implies that the function

$$k(\mathcal{X}, \mathcal{Y}) = \langle \mathcal{X}, \mathcal{Y} \rangle \tag{4.16}$$

where $\langle \cdot, \cdot \rangle$ is the inner product introduced in (B.5), defines a valid reproducing kernel. Indeed (4.16) reads here

$$k(\mathcal{X}, \mathcal{Y}) = \langle \mathrm{vec}(\mathcal{X}), \mathrm{vec}(\mathcal{Y}) \rangle \tag{4.17}$$

where $\mathrm{vec}(\cdot)$ denotes vector unfolding and the inner product in the right hand-side is defined on $\mathbb{R}^{D_1 D_2 \cdots D_M}$. Equation (4.16) is an elementary generalization of the linear kernel defined on $\mathbb{R}^I$. In a similar way other kernel functions admit a straightforward generalization to tensors. For instance, a natural way to generalize the popular Gaussian-RBF kernel [177] reads

$$k(\mathcal{X}, \mathcal{Y}) = \exp\left(-\frac{1}{2\sigma^2}\|\mathcal{X} - \mathcal{Y}\|^2\right) \tag{4.18}$$

where $\|\cdot\|$ denotes the Frobenius norm introduced in (2.25) and σ is used to set an appropriate bandwidth. However observe that both (4.16) and (4.18) treat tensor data as mere collections of entries without keeping into account the underlying structure. In particular notice that (4.18) can be equivalently restated as:

$$k(\mathcal{X}, \mathcal{Y}) = \prod_{p \in \mathbb{N}_{D_1} \times \mathbb{N}_{D_2} \times \cdots \times \mathbb{N}_{D_M}} \exp\left(-\frac{1}{2\sigma^2}(x_p - y_p)^2\right) \tag{4.19}$$

namely as the product of Gaussian-RBF kernels each of which is defined on the entries of data tensors.

Suppose now that P denotes an operator that acts on data tensors by permuting their entries according to some fixed rule. Then, when k is either (4.16) or (4.18), we clearly have

$$k(\mathcal{X}, \mathcal{Y}) = k(P(\mathcal{X}), P(\mathcal{Y})) \ .$$

This type of invariance is not desirable in many practical situations. For the case of grayscale images, namely second order tensors, the use of this kernel leads to ignoring the relation between each pixel and its neighbors. For videos [218, 219], namely third order tensors, it would additionally neglect the temporal structure.

4.2.2 Structure Exploiting Kernels

Notice that (4.18) is a special case of a more general class of *product kernels*

$$k(\mathcal{X},\mathcal{Y}) = k^1(\mathcal{X},\mathcal{Y})k^2(\mathcal{X},\mathcal{Y})\cdots k^P(\mathcal{X},\mathcal{Y}) \tag{4.20}$$

which are obtained from the product of P *factor kernels* k^p, $p \in \mathbb{N}_P$. Note that the latter is positive definite as soon as the factor kernels are positive, see Appendix B.1. Product kernels can be seen to arise from a space of *multilinear functionals* which are, loosely speaking, infinite dimensional tensors, see Appendix B.4. This fact is relevant on its own: it shows that these kernels are strictly connected to the notion of finite dimensional tensors on which tensor-based techniques are grounded. The consequences of this fact will be discussed in Section 4.2.4.

The naïve kernels discussed in the previous Subsection act on the data tensors as if they were vectors of $\mathbb{R}^{D_1 D_2 \cdots D_M}$. In this way one defines the distance between two tensors $\mathcal{X}$ and $\mathcal{Y}$ as the length $\|\mathcal{X}-\mathcal{Y}\|$ of the straight line segment connecting them. It is known, however, that many objects of interest live in low dimensional manifolds embedded in high dimensional vector spaces[5]. In all these cases the Euclidean metric is suboptimal to capture the topology of the input patterns. To cope with such cases we will now introduce, as factors, a new class of kernel functions based upon the chordal distance on the Grassmannian manifolds of matrix unfoldings.

Distance Between Matrix Unfoldings

Next we address the problem of defining a similarity measure taking advantage of the algebraic structure of input tensors. This can be achieved regarding tensors as the collection of linear subspaces coming from each matricization (see Definition 2.3.3). Assume for now that

$$D_m < D_{m+1}\, D_{m+2} \cdots D_M\, D_1\, D_2\, D_3 \cdots D_{m-1}\ . \tag{4.21}$$

Recall that by $\mathscr{R}(\boldsymbol{W}^\top)$ we mean the row space of a matrix $\boldsymbol{W} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2}$, $\mathscr{R}(\boldsymbol{W}^\top) := \left\{\boldsymbol{W}^\top a \ :\ a \in \mathbb{R}^{D_1}\right\} \subseteq \mathbb{R}^{D_2}$. We can now define the $m-$th factor kernel in (4.20) as:

$$k^m(\mathcal{X},\mathcal{Y}) := \exp\left(-\frac{1}{2\sigma^2} d(\mathcal{X}_{<m>},\mathcal{Y}_{<m>})^2\right) \tag{4.22}$$

[5]For instance, the space of linear dynamical systems, which are determined only up to a change of basis, has the structure of a Stiefel manifold.

where $\sigma \in \mathbb{R}^+$ and $d(\mathcal{X}_{<m>}, \mathcal{Y}_{<m>})$ denotes a suitable distance between $\mathscr{R}(\mathcal{X}_{<m>}^\top)$ and $\mathscr{R}(\mathcal{Y}_{<m>}^\top)$ on the *Grassmann manifold* corresponding to the set of D_m dimensional subspaces in a vector space of dimension $D_{m+2} \cdots D_M D_1 D_2 D_3 \cdots D_{m-1}$.

The idea of using subspaces has already been exploited to establish a similarity between matrices [100]. This choice has been shown to be relevant in a number of tasks such as face recognition, see, e.g., [18] and references therein. The choice of using an exponential in (4.22) is to a large extent arbitrary. In fact, one has only to ensure that the factor kernels are positive definite which, in turn, guarantees that (4.20) is a valid reproducing kernel. This, in particular, imposes restrictions on the choice of the distance function d. Notably, however, the definition in (4.22) implies that the product kernel k in (4.20) can be equivalently restated as the RBF kernel $k(\mathcal{X}, \mathcal{Y}) = \exp(-1/(2\sigma^2) d_T(\mathcal{X}, \mathcal{Y})^2)$ that closely resembles (4.18) but differs in that the Euclidean norm is replaced by the non-Euclidean distance function defined as:

$$d_T(\mathcal{X}, \mathcal{Y}) = \sqrt{\sum_{m \in \mathbb{N}_M} d(\mathcal{X}_{<m>}, \mathcal{Y}_{<m>})^2} \,. \tag{4.23}$$

Note that we can consider, as factors, kernels based on matricizations indexed by any subset $\mathscr{P} \subseteq \mathbb{N}_M$. The choice of factors to be retained can be guided by suitable information criteria such as the kernel-target alignment [59]. In the following we will assume for simplicity that $\mathscr{P} = \mathbb{N}_M$. Later we will show that this case enjoys a special invariance property.

Relation with Principal Angles and Choice of Distance

It turns out that any unitarily invariant metric on a Grassmannian manifold connects to the notion of *principal angles*. Let us recall that for

$$R = \min\left\{\dim(\mathscr{R}(\mathcal{X}_{<m>}^\top)), \dim(\mathscr{R}(\mathcal{Y}_{<m>}^\top))\right\}$$

the *principal angles* $\theta_1^{(m)}, \theta_2^{(m)}, \ldots, \theta_R^{(m)}$ between $\mathscr{R}(\mathcal{X}_{<m>}^\top)$ and $\mathscr{R}(\mathcal{Y}_{<m>}^\top)$ can be defined recursively by

$$\cos(\theta_r^{(m)}) := \max_{x \in \mathscr{R}(\mathcal{X}_{<m>}^\top), y \in \mathscr{R}(\mathcal{Y}_{<m>}^\top)} \langle x, y \rangle = \langle x^{(r)}, y^{(r)} \rangle$$

$$\text{subject to } \|x\| = \|y\| = 1 \text{ and } \langle x, x^{(i)} \rangle = \langle y, y^{(i)} \rangle = 0, \; i \in \mathbb{N}_{r-1}$$

for any $r \in \mathbb{N}_R$. Among the various distance measures arising from the principal angles [75] a suitable distance between $\mathscr{R}(\mathcal{X}_{<m>}^\top)$ and $\mathscr{R}(\mathcal{Y}_{<m>}^\top)$ is

the *projection Frobenius norm* (also known as *chordal distance* [54]). It relies on the one-to-one correspondence between a subspace $\mathscr{A}$ and the associated orthogonal projection $\Pi_{\mathscr{A}}$ and is defined by:

$$d_{pF}(\mathcal{X}_{<m>}, \mathcal{Y}_{<m>}) \;:=\; \left\| \Pi_{\mathscr{R}(\mathcal{X}_{<m>}^{\top})} - \Pi_{\mathscr{R}(\mathcal{Y}_{<m>}^{\top})} \right\| \;=\; \sqrt{2} \left\| \sin \theta^{(m)} \right\| \quad (4.24)$$

where $\sin \theta^{(m)}$ is the vector obtained taking the sine of each one of the principal angles between $\mathcal{X}_{<m>}^{\top}$ and $\mathcal{Y}_{<m>}^{\top}$. This specific choice of distance gives rise to positive definite kernels.

Theorem 4.2.1. *If the distance function d corresponds to the projection Frobenius norm* (4.24) *then the tensorial kernel k obtained from the product of factors* (4.22) *is positive definite.*

Proof. A proof can be found in [183]. □

Factors, Tensor Dimensions and Degeneracy

In (4.21) for ease of presentation we made a precise assumption on the dimensions of the $m-$th matrix unfolding. We shall now discuss all the three possible situations for the case where factors are defined upon the chordal distance d_{pF}:

> **case 1:** $D_m < D_{m+1}\, D_{m+2} \cdots D_M\, D_1\, D_2\, D_3 \cdots D_{m-1}$. This is the case that we considered above. It holds that $d_{pF}(\mathcal{X}_{<m>}, \mathcal{Y}_{<m>}) > 0$ and hence $k^m(\mathcal{X}, \mathcal{Y}) < 1$ unless $\mathcal{X}_{<m>}$ and $\mathcal{Y}_{<m>}$ span the same row space.
>
> **case 2:** $D_m > D_{m+1}\, D_{m+2} \cdots D_M\, D_1\, D_2\, D_3 \cdots D_{m-1}$. In this case we define k^m in (4.22) based upon a distance between column spaces instead of row spaces.
>
> **case 3:** $D_m = D_{m+1}\, D_{m+2} \cdots D_M\, D_1\, D_2\, D_3 \cdots D_{m-1}$. Under this condition we have that $k^m(\mathcal{X}, \mathcal{Y}) = 1$ unless *both* $\mathcal{X}_{<m>}$ and $\mathcal{Y}_{<m>}$ are rank deficient. In practice when dealing with real-life noisy data this event does not occur. Thus, in general, the $m-$th matricization is uninformative and we can avoid computing k^m since it does not contribute to the product kernel (4.20). Notice, however, that the case of square matrix unfolding can occur at most for a single running index $m \in \mathbb{N}_M$: the remaining $M-1$ are guaranteed to be non-square and informative.

As a concrete example of the third case let $\mathcal{X} \in \mathbb{R}^9 \otimes \mathbb{R}^3 \otimes \mathbb{R}^3$. The first matrix unfolding is square and hence in general uninformative whereas $\mathscr{R}(\mathcal{X}_{<2>}^{\top})$ and

$\mathscr{R}(\mathcal{X}^\top_{<3>})$ are both 3-dimensional subspaces of $\mathbb{R}^{27}$ and we can conveniently compute their similarity based upon the information they share.

We conclude noticing that, in particular, case 3 never arises for *cubic* tensors namely for elements of $\mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M}$ where $D_1 = D_2 = \cdots = D_M = D$. In practice, as in Subsection 4.2.3, the tensor representation is often enforced by the user for instance to take advantage of certain characteristics of data, such as their dynamical nature. In these situations the dimensions of the tensor representation can be chosen and hence one can avoid degenerate cases. Next we clarify the relation with the MLSVD.

Link with the MLSVD

Recall that, at a matrix level, the MLSVD (Section 2.3.2) of a data tensor $\mathcal{X}$ boils down to the SVD of the matrix unfoldings $\mathcal{X}_{\langle m \rangle}$, where $m \in \mathbb{N}_M$. The latter can be stated in block-partitioned form as:

$$\mathcal{X}_{\langle m \rangle} = \begin{pmatrix} \boldsymbol{U}^{(m)}_{\mathcal{X},1} & \boldsymbol{U}^{(m)}_{\mathcal{X},2} \end{pmatrix} \begin{pmatrix} \boldsymbol{S}^{(m)}_{\mathcal{X},1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{V}^{(m)\top}_{\mathcal{X},1} \\ \boldsymbol{V}^{(m)\top}_{\mathcal{X},2} \end{pmatrix} \tag{4.25}$$

where entries on the diagonal of $\boldsymbol{S}^{(m)}_{\mathcal{X},1}$ are assumed to be ordered in a decreasing manner. A well known property of the SVD states now that the orthogonal projection operator onto $\mathscr{R}(\mathcal{X}^\top_{<m>})$ is given by

$$\Pi_{\mathscr{R}(\mathcal{X}^\top_{<m>})} = \boldsymbol{V}^{(m)}_{\mathcal{X},1} \boldsymbol{V}^{(m)\top}_{\mathcal{X},1} \ .$$

Hence computing the tensorial kernel based on the projection Frobenius norm, corresponds to computing the MLSVD (equivalently finding the SVD of the matrix unfoldings) and let the factor kernel be

$$k^m(\mathcal{X}, \mathcal{Y}) = \exp\left(-\frac{1}{2\sigma^2} \left\| \boldsymbol{V}^{(m)}_{\mathcal{X},1} \boldsymbol{V}^{(m)\top}_{\mathcal{X},1} - \boldsymbol{V}^{(m)}_{\mathcal{Y},1} \boldsymbol{V}^{(m)\top}_{\mathcal{Y},1} \right\|^2 \right) \ . \tag{4.26}$$

Figure 4.4 illustrates the computation of the tensorial kernel based on the SVD's of the matrix unfoldings. Simple matrix algebra shows that (4.26) is equivalent to

$$k^m(\mathcal{X}, \mathcal{Y}) = \exp\left(-\frac{1}{\sigma^2} (D_m - \mathrm{trace}(\boldsymbol{Z}^\top \boldsymbol{Z}) \right) \tag{4.27}$$

where $\boldsymbol{Z} = \boldsymbol{V}^{(m)\top}_{\mathcal{X},1} \boldsymbol{V}^{(m)}_{\mathcal{Y},1} \in \mathbb{R}^{D_m} \otimes \mathbb{R}^{D_m}$. This formula is more efficiently computed than the right hand-side of (4.26).

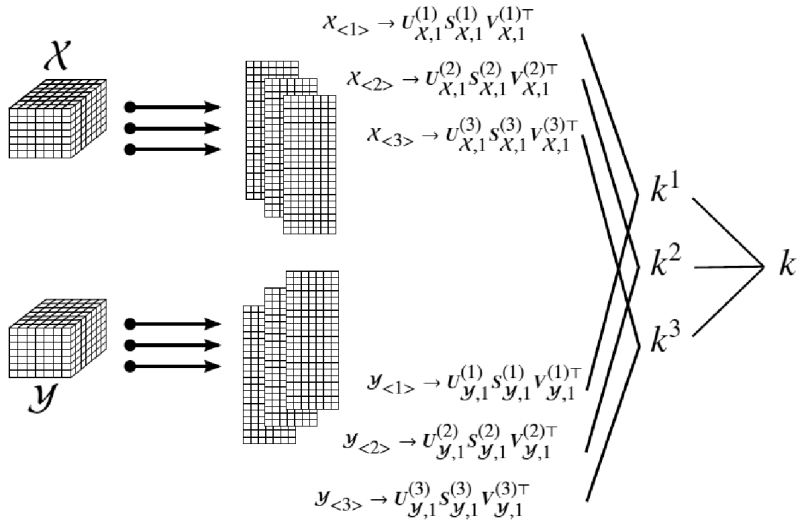


Figure 4.4: An illustration of the tensorial kernel k based upon factors (4.26). For 3−rd order tensors $\mathcal{X}$ and $\mathcal{Y}$ it requires to compute the SVD of the matrix unfoldings $\mathcal{X}_{<m>}$ and $\mathcal{Y}_{<m>}$.

4.2.3 Congruent Data Tensors and Invariance Property

How to describe the intrinsic geometry of manifolds in learning problems is an important issue that involves the understanding of certain *invariance properties* [37]. In this Section we consider cubic data tensors and study the invariance property that follows from regarding tensors as the collection of linear subspaces spanned by each matricization. As in the previous Sections we shall assume that the tensorial kernel is defined upon the projection Frobenius norm d_{pF}: $k(\mathcal{X}, \mathcal{Y}) = \exp(-1/(2\sigma^2) \sum_{n \in \mathbb{N}_N} d_{pF}(\mathcal{X}_{<n>}, \mathcal{Y}_{<n>})^2)$.

Congruence Sets and Invariance

In the following two data tensors $\mathcal{X}$ and $\mathcal{Y}$ are called *congruent* if $k(\mathcal{X}, \mathcal{Y}) = 1$. Additionally if $k(\mathcal{X}, \mathcal{Y}) = 1$ for any pair $\mathcal{X}, \mathcal{Y} \in \mathscr{X}$, then we call $\mathscr{X}$ a *congruence set*. A characterization of tensors by means of subspaces [65] shows that congruence sets arise, in particular, in the following case.

Theorem 4.2.2 (Congruence Classes of Data Tensors)**.** *Assume matrices* $\boldsymbol{A} = [A_1, A_2, \cdots, A_R]$, $\boldsymbol{B} = [B_1, B_2, \cdots, B_R]$, $\boldsymbol{C} = [C_1, C_2, \cdots, C_R] \in \mathbb{R}^I \otimes \mathbb{R}^R$ *with*

full rank R. A set $\mathscr{X} \subset \mathbb{R}^I \otimes \mathbb{R}^I \otimes \mathbb{R}^I$ *is a congruence set if for any* $\mathcal{X} \in \mathscr{X}$

$$\mathcal{X} = \sum_{r \in \mathbb{N}_R} d_r \; A_r \otimes B_r \otimes C_r \tag{4.28}$$

for some $D = (d_1, \ldots, d_R) \in \mathbb{C}^R$ *with* $d_r \neq 0$ *for all* $r \in \mathbb{N}_R$.

Before proceeding it is important to stress that congruence set membership of a data tensor $\mathcal{X}$ is *invariant* with respect to the specific value of the multiplier vector D in (4.28). Notice that the result holds also for the case where elements of $\mathscr{X}$ are general complex-valued tensors. Further details are found in [65] that actually deals with a broader specification of equivalence classes. Our next goal is to highlight the significance of this result for pattern recognition.

Implications for Pattern Recognition

A first important remark pertains the nature of congruence sets.

Remark. If $\mathscr{X}_1$ and $\mathscr{X}_2$ are congruence sets corresponding to matrices $\{\boldsymbol{A}_1, \boldsymbol{B}_1, \boldsymbol{C}_1\}$ and $\{\boldsymbol{A}_2, \boldsymbol{B}_2, \boldsymbol{C}_2\}$ respectively, then $\{\boldsymbol{A}_1, \boldsymbol{B}_1, \boldsymbol{C}_1\} \neq \{\boldsymbol{A}_2, \boldsymbol{B}_2, \boldsymbol{C}_2\}$ implies that the two sets do not intersect ($\mathscr{X}_1 \cap \mathscr{X}_2 = \emptyset$).

In light of this, the machinery of congruence sets is seen to have an immediate application for pattern recognition. In fact, suppose that we want to discriminate between classes that are known to coincide with separate congruence sets. In this limiting case we are guaranteed that the within class distance is exactly zero and the between class distance is strictly positive. The use of factor kernels (4.26) ensures that perfect class separation is achieved. For practical problems, however, one does not know in advance if classes are well approximated by congruence sets. The question is then if the embedding implied by factor kernels still captures the structure of the learning tasks of interest. In fact, in the statistical learning literature several results exist showing that generalization takes place if this is the case. This type of insight can be achieved, for instance, based upon kernel-target alignment [59]. Assume we are given a training set of N input-output pairs:

$$\mathscr{D}_N := \left\{ \left(\mathcal{X}^{(n)}, y_n \right) \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M} \times \mathscr{Y} \; : \; n \in \mathbb{N}_N \right\} , \tag{4.29}$$

where $\mathscr{Y} \subset \mathbb{R}$. Recall the definition of inner product (B.5) for tensors of arbitrary order. Then the (empirical) kernel-target alignment $A(\boldsymbol{K}, y)$ is

$$A(\boldsymbol{K}, y) = \frac{\langle \boldsymbol{K}, yy^\top \rangle}{N \sqrt{\langle \boldsymbol{K}, \boldsymbol{K} \rangle}} \tag{4.30}$$

and represents the agreement between the kernel matrix $(\boldsymbol{K})_{ij} = k(\mathcal{X}^{(i)}, \mathcal{X}^{(j)})$ and the set of labels y. A concentration bound shows that this empirical quantity is concentrated around its population counterpart; in turn it can be shown that if the population alignment is high then there always exists a good classification hyperplane [58].

Equation (4.30) only depends upon the kernel matrix $\boldsymbol{K}$ and the training labels. Hence the alignment can be used as a criterion to compare different similarity measures before training the corresponding models. Finally it is important to remark that the alignment is clearly task dependent: for the general case it is hard to grasp before computing the kernel matrix if the similarity measure does capture the structure of the problem. In practice it is expected that the factor kernels (4.26) outperform general purpose similarity as soon as classes are well approximated by congruence sets. The purpose of the next Subsection is then to illustrate a special case where this situation arises.

The Special Case of Hankel Tensors

In this section we consider a specific class of tensorial representations. We focus on the case where input tensors with Hankel structure were constructed based upon univariate signals. Let $\{s_0, s_1, \cdots, s_{T-1}\}$ be a sequence of T real-valued numbers that represent a signal S on a time (or space) domain. We shall assume that we can write

$$s_t = \sum_{k=0}^{T-1} \xi_k z_k^t \tag{4.31}$$

where $\{\xi_0, \xi_1, \cdots, \xi_{T-1}\}$ is a sequence of T complex-valued numbers that represent weights and $\left\{z_k^0, z_k^1, \cdots, z_k^{T-1}\right\}$ are powers of $z_k = \exp\left((i2\pi f_k - d_k)\Delta t\right)$, the k-th pole of the signal[6]. One specific situation arises when $d_k = 0,\ f_k = k$ and finally $\Delta t = \frac{1}{T}$ in which case (4.31) is the Inverse Discrete Fourier Transform (IDFT) [55]. The weights collectively form the *spectrum* of the original signal S. Assume now integers I_1, I_2 and I_3 satisfying $I_1+I_2+I_3 = T+2$. The *Hankel tensor* $\mathcal{X} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \mathbb{R}^{I_3}$ of the signal S [155] can be defined entry-wise by

$$x_{i_1 i_2 i_3} := s_{i_1+i_2+i_3-3}\ . \tag{4.32}$$

[6]We denoted by i the imaginary unit $i = \sqrt{-1}$.

In light of (4.31) and a fundamental property of the (complex) exponential we now have that $\mathcal{X}$ can be equivalently restated in terms of rank-1 tensors as:

$$\mathcal{X} = \sum_{k \in \mathbb{N}_T} \xi_{k-1} \begin{pmatrix} z_{k-1}^0 \\ z_{k-1}^1 \\ \vdots \\ z_{k-1}^{I_1-1} \end{pmatrix} \otimes \begin{pmatrix} z_{k-1}^0 \\ z_{k-1}^1 \\ \vdots \\ z_{k-1}^{I_2-1} \end{pmatrix} \otimes \begin{pmatrix} z_{k-1}^0 \\ z_{k-1}^1 \\ \vdots \\ z_{k-1}^{I_3-1} \end{pmatrix} . \tag{4.33}$$

When $\mathcal{X}$ is cubic the latter is seen to be a special case of (4.28). Theorem 4.2.2 means, in this context, that two cubic Hankel tensors are congruent if the corresponding signals decompose into the same poles. For the IDFT case this means that the two cubic Hankel tensors are equivalent if the spectra of the corresponding signals share the same support. Hence the proposed kernel in combination with Hankel tensors is well suited for the case where, within the same class, signals have approximately the same spectral content.

For ease of exposition, in (4.32) we have chosen to deal with the simplest notion of Hankel tensors. An alternative and more powerful definition of Hankel tensors exists for univariate signals [156] and also the multi-channel case can be dealt with [155]. Due to its symmetrical nature, the Hankel tensor $\mathcal{X}$ as defined above satisfies $\mathcal{X}_{<1>} = \mathcal{X}_{<2>} = \mathcal{X}_{<3>}$ which is not the case for the alternative definitions. In practice this means that when applied to this type of Hankel tensors the tensorial kernel k based on factors (4.26) can be simplified to

$$k(\mathcal{X}, \mathcal{Y}) = \exp\left(-\frac{1}{2\sigma^2} \left\| \boldsymbol{V}_{\mathcal{X},1}^{(1)} \boldsymbol{V}_{\mathcal{X},1}^{(1)\top} - \boldsymbol{V}_{\mathcal{Y},1}^{(n)} \boldsymbol{V}_{\mathcal{Y},1}^{(1)\top} \right\|^2\right) \tag{4.34}$$

where we considered only the first matricization. In Section 5.20 we will provide explicit examples both for univariate and multichannel signals. Finally we remark that a different approach for the classification of signals can be based on *cumulant tensors* [184].

4.2.4 Model Estimation

We now turn to the general learning problem of interest. We want to estimate a model f to predict a target variable $y \in \mathscr{Y}$ from an input pattern $\mathcal{X} \in \mathbb{R}^{D_1} \otimes \mathbb{R}^{D_2} \otimes \cdots \otimes \mathbb{R}^{D_M}$ given a training set of N input-output pairs (4.29).

Estimation via Representer Theorems

For positive kernels, the Moore-Aronszajn theorem (see Theorem B.1.5 in Appendix B.1) ensures that there exists only one Hilbert space $\mathscr{H}_k$ of functions

on $\mathscr{X}$ with k as reproducing kernel. The estimation of a nonparametric model of $\mathcal{X}$ can then be formulated as a variational problem in the function space $\mathscr{H}_k$, as we discussed in Section 2.2.2. In its most general form this problem reads:

$$\min_{f \in \mathscr{H}_k} \sum_{m \in \mathbb{N}_M} l\left(y_m, f(\mathcal{X}^{(m)})\right) + \gamma g\left(\|f\|_{\mathscr{H}_k}\right) \tag{4.35}$$

where $l : \mathscr{Y} \times \mathbb{R} \to \mathbb{R}^+$ is a *loss function* used to measure goodness-of-fit, g is a strictly monotonically increasing real-valued function on $[0, \infty)$ and finally $\|f\|_{\mathscr{H}_k} := \sqrt{\langle f, f \rangle_{\mathscr{H}_k}}$. Despite the infinite dimensionality of the RKHS $\mathscr{H}_k$ a solution can be found based on finite dimensional optimization as ensured by representer theorems, see Section 2.2.2 and references therein.

Primal-Dual Techniques

An alternative approach relies on primal-dual techniques that underlies Support Vector Machines (SVM) and related estimators [195, 198, 199]. In this case one starts from a primal model representation of the type:

$$f_{(\Psi,b)}(\mathcal{X}) := \langle \Psi, \tilde{\phi}(\mathcal{X}) \rangle_{\mathrm{HSF}} + b \tag{4.36}$$

where HSF is the feature space of multilinear functionals associated to a product kernel (4.20), with

$$k(\mathcal{X}, \mathcal{Y}) = \langle \tilde{\phi}(\mathcal{X}), \tilde{\phi}(\mathcal{Y}) \rangle_{\mathrm{HSF}} \ ,$$

see Appendix B.4. The primal problem formulation is then aimed at finding an optimal $(\Psi^\star, b^\star) \in \mathrm{HSF} \times \mathbb{R}$. Notice that the latter defines an affine hyperplane in HSF. Remarkably, (4.36) is affine in $\tilde{\phi}(\mathcal{X})$ as much as (4.1) is affine in $\mathcal{X}$. However since $\tilde{\phi}$ is in general a nonlinear mapping, $f_{(\Psi,b)}$ does not depend linearly on $\mathcal{X}$ which provides the improved flexibility of the model with respect to the affine class (4.1).

Relying on Lagrangian duality arguments the problem is re-parametrized in terms of dual variables $\{\alpha_n\}_{n \in \mathbb{N}_N}$ and solved in $(\alpha, b) \in \mathbb{R}^{N+1}$. In comparison with the methodology based on representer theorems the primal-dual approach emphasizes the geometrical aspects of the problem and it is particularly insightful when $\mathscr{Y} = \{+1, -1\}$ and (4.36) is used to define a discriminative rule of the type $\hat{y} = \mathrm{sign}(f_{(\Psi^\star, b^\star)}(\mathcal{X}))$. Additionally, primal-dual techniques are best suited to deal with supplementary constraints that might be used to encode prior knowledge. Vapnik's original SVM formulation [57] translates into convex quadratic programs. In contrast, in Least-Squares SVM (LS-SVM) [199], a modification of the SVM primal problem leads to a considerably simpler estimation. In particular, the primal formulation for classification [200]

reads in our setting:

$$\begin{array}{ll} \min\limits_{(\Psi,E,b)\in \mathrm{HSF}\times\mathbb{R}^N\times\mathbb{R}} & \frac{1}{2}\langle\Psi,\Psi\rangle_{\mathrm{HSF}} + \gamma\frac{1}{2}\sum_{m\in\mathbb{N}_M} e_m^2 \\ \text{subject to} & y_n(\langle\Psi,\tilde{\phi}(\mathcal{X}^{(n)})\rangle_{\mathrm{HSF}} + b) = 1 - e_n,\ n\in\mathbb{N}_N \end{array} \tag{4.37}$$

where $\gamma > 0$ is a user-defined trade-off parameter. It is possible to show that the estimation can be performed solving the following system of linear equations:

$$\begin{bmatrix} 0 & Y^\top \\ Y & \boldsymbol{\Omega} + \frac{1}{\gamma}\boldsymbol{I}_M \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_N \end{bmatrix} \tag{4.38}$$

where $1_N = (1, 1, \ldots, 1) \in \mathbb{R}^N$, $\boldsymbol{I}_N = \mathrm{diag}(1_N)$ and $\boldsymbol{\Omega} \in \mathbb{R}^M \otimes \mathbb{R}^M$ is defined entry-wise by

$$(\boldsymbol{\Omega})_{ij} = y_i y_j \langle\tilde{\phi}(\mathcal{X}^{(i)}), \tilde{\phi}(\mathcal{X}^{(j)})\rangle_{\mathrm{HSF}} = y_i y_j k(\mathcal{X}^{(i)}, \mathcal{X}^{(j)}) \ .$$

Finally to evaluate $f_{(\Psi^\star,b^\star)}$ at a given test point $\mathcal{X}$, the dual model representation is exploited:

$$f_{(\Psi^\star,b^\star)}(\mathcal{X}) = \sum_{n\in\mathbb{N}_N} y_n\alpha_n^\star k(\mathcal{X}^{(n)}, \mathcal{X}) + b^\star \ . \tag{4.39}$$

Notice that problem (4.37) involves the infinite dimensional multilinear functional $\Psi \in \mathrm{HSF}$ and the results of finite dimensional optimization do not apply rigorously. Theories of optimization in abstract vector spaces are the subject of [17, 76, 91, 136] and [115], among others. For Vapnik's SVM formulation a rigorous primal-dual derivation is discussed in [132]. Similar results for LS-SVM have not been reported, to the best of our knowledge. A formal derivation is given in Appendix C.

The procedure to compute a model with the tensorial kernel is summarized in Table 4.3. It is assumed that both the parameter γ in (4.38) and σ in (4.26) are given. In practical applications the choice of these parameters is performed according to some model selection criterion often based on cross-validation.

Structure-inducing Penalties

It is worth noticing that the optimality conditions of (4.37) (see Appendix C) yield

$$\Psi^\star = \sum_{n\in\mathbb{N}_N} \alpha_n^\star y_n \tilde{\phi}(X^{(n)}) \tag{4.40}$$

which — given the nature of HSF — shows that the optimal multilinear functional $\Psi^\star$ has at most rank N where N is the cardinality of the training set.

Table 4.3: Model estimation with factor kernels (4.26)

input: γ, σ, training pairs $\left\{\left(\mathcal{X}^{(\mathrm{m})}, \mathrm{y_m}\right) \ : \ \mathrm{m} \in \mathbb{N}_\mathrm{M}\right\}$.
comment: Compute $\boldsymbol{\Omega}$

for each $\mathrm{m}_1, \mathrm{m}_2 \in \mathbb{N}_\mathrm{M}$ and $\mathrm{m}_2 > \mathrm{m}_1$

do
- **for each** $\mathrm{n} \in \mathbb{N}_N$
- **do**
  - $\boldsymbol{V}^{(n)}_{\mathcal{X}^{(m_1)},1} \leftarrow \mathrm{SVD}(\mathcal{X}^{(m_1)}_{<n>})$
  - $\boldsymbol{V}^{(n)}_{\mathcal{X}^{(m_2)},1} \leftarrow \mathrm{SVD}(\mathcal{X}^{(m_2)}_{<n>})$
  - $\boldsymbol{Z}_{(n)} \leftarrow \boldsymbol{V}^{(n)\top}_{\mathcal{X}^{(m_1)},1} \boldsymbol{V}^{(n)}_{\mathcal{X}^{(m_2)},1}$
  - $a_n \leftarrow I_n - \mathrm{trace}(\boldsymbol{Z}^\top_{(n)} \boldsymbol{Z}_{(n)})$
- $(\boldsymbol{\Omega})_{m_1 m_2} \leftarrow y_{m_1} y_{m_2} \exp\left(-\frac{1}{\sigma^2}(a_1 + a_2 + \cdots + a_N)\right)$

$\boldsymbol{\Omega} \leftarrow \boldsymbol{\Omega} + \boldsymbol{\Omega}^\top + \mathbf{I}_\mathrm{M}$
comment: Find model parameters

Solve (4.38) for given $\boldsymbol{\Omega}$, Y and parameter γ .

As seen in Section 2.2.2, in kernel algorithms the complexity of the model class is usually controlled by Hilbertian norms in connection to the SRM principle. In the present context capacity control is performed by $\langle \Psi, \Psi \rangle_{\mathrm{HSF}}$, the squared Frobenius norm of Ψ. On the other hand, the interpretation of equation (4.40) suggests that a different complexity measure might be based on some generalized notion of rank. In Section (3) we effectively used spectral regularization to control the complexity in finite dimensional spaces of tensors. An interesting line of future research consists of studying kernel extensions of this approach.

Computational Complexity

The approach illustrated above consists of two main steps: first the computation of the kernel matrix and successively solving a system of linear equations. Whereas this second step depends upon the chosen kernel technique (LS-SVM, in our case), the first step is required by most of the algorithms making use of the kernel trick. With reference to Table 4.3, assume that the M training patterns are $D_1 \times D_2 \times \cdots \times D_N$ tensors (see also Figure 4.4 for the case where $M = 3$). Then the computation of the kernel matrix requires $M(M-1)$ MLSVDs. In turn, the MLSVD of the m-th pattern $\mathcal{X}^{(m)}$ actually consists of the SVDs of N matrices $\mathcal{X}^{(m)}_{\langle 1 \rangle}, \mathcal{X}^{(m)}_{\langle 2 \rangle}, \cdots, \mathcal{X}^{(m)}_{\langle N \rangle}$ where the $n-$th matrix $\mathcal{X}^{(m)}_{\langle n \rangle}$ has size $D_n \times J_n$, $J_n = (D_1 D_2 \cdots D_{n-1} D_{n+1} \cdots D_N)$. Each SVD is required to evaluate one argument of a factor kernel. Assuming $\mathcal{X}^{(m)}_{\langle n \rangle}$ has full

rank D_n, the cost of the $n-$th SVD using Chan's algorithm is approximately $O(6J_nD_n^2 + 11D_n^3)$ operations [94]. As a concrete example, consider cubic tensors with random Gaussian entries and $D_1 = D_2 = D_3 = 50$. In this case the evaluation of the tensorial kernel for a pair of random patterns $\mathcal{X}$ and $\mathcal{Y}$ takes approximately 0.25 seconds on a standard laptop using MATLAB. Consequently computing a 100×100 kernel matrix takes about 40 minutes. In comparison, the evaluation of the RBF-Gaussian kernel on the vectors $\text{vec}(\mathcal{X})$ and $\text{vec}(\mathcal{Y})$ with dimension $1.25 \cdot 10^5$ takes only about $2.5 \cdot 10^{-3}$ seconds and therefore computing a 100×100 kernel matrix requires only about 25 seconds. Finally note that, when computing kernel matrices corresponding to different values of σ, it is not needed to recompute the MLSVDs. Therefore performing cross-validation to choose the kernel and regularization parameters can be done at the price of computing a single kernel matrix. Finally note that the computation of MLSVD is amenable to parallelization; this becomes of particular relevance when the order of the tensors N is high.

4.2.5 Experiments

Classification of Sparsity Patterns

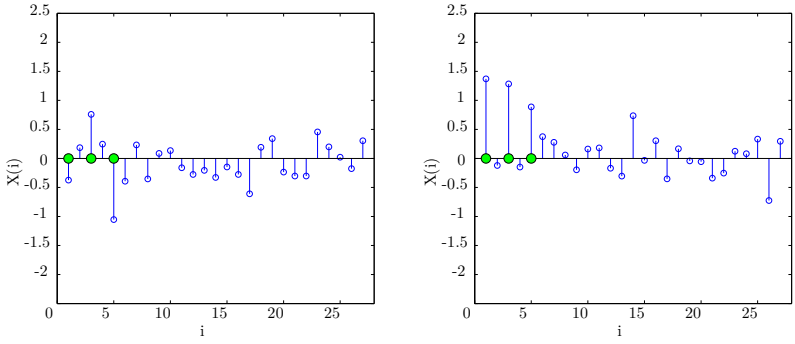
The purpose of this experiment is to test the impact of the invariance property studied in Section 4.2.3 on a classification problem. Let $e^j \in \mathbb{R}^I$ be the j-th canonical basis vector defined as $e^j_i := 1$ if $i = j$ and $e^j_i := 0$ otherwise and let $\Delta_j \in \mathbb{R}^I \otimes \mathbb{R}^I \otimes \mathbb{R}^I$ be the rank-1 tensor defined as:

$$\Delta_j := e^j \otimes e^j \otimes e^j \ .$$

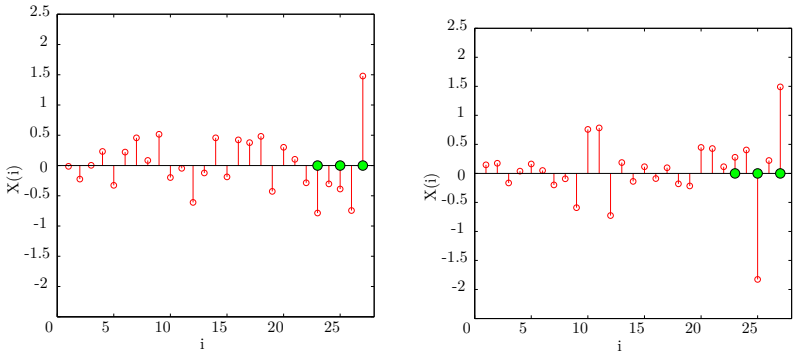
We generated data tensors in $\mathbb{R}^I \otimes \mathbb{R}^I \otimes \mathbb{R}^I$ according to the following model:

$$\mathcal{X}^{(m)} = \begin{cases} a_m\Delta_1 + b_m\Delta_2 + c_m\Delta_3 + \mathcal{E}^{(m)}, & \text{if } y_m = +1 \\ a_m\Delta_4 + b_m\Delta_5 + c_m\Delta_6 + \mathcal{E}^{(m)}, & \text{if } y_m = -1 \end{cases} \tag{4.41}$$

where a_m, b_m and c_m are i.i.d. from a zero-mean Gaussian distribution with variance $1 - \beta^2$ and the entries of the noise tensor $\mathcal{E}^{(m)}$ are i.i.d. from a zero-mean Gaussian distribution with variance β^2. We then consider the binary classification problem that consists of estimating the underlying label of a given test data tensor. A comparison between (4.41) and (4.28) reveals that for $\beta^2 = 0$ (noiseless case) the two classes of tensors correspond to separate congruence sets, see also Remark 4.2.3. Additionally, this task can be regarded as the classification of vectors of $\mathbb{R}^{I^3}$ having two different types of sparsity patterns, see Figure 4.5 for the case where $I = 3$. We use the LS-SVMlab toolbox (`www.esat.kuleuven.be/sista/lssvmlab`, [63]) and perform training with N input-output pairs. We compared the Gaussian-RBF kernel



(a) class 1



(b) class 2

Figure 4.5: By vector unfolding the experiment of Section 4.2.5 can be interpreted as the classification of sparsity patterns of (noisy) vectors. As an example we take here $I = 3$ and plot the 27 elements of the vectorized version of data tensors generated according to (4.41). The solid green dots in plots 4.5(a) and 4.5(b) represent two hypothetical index sets of non-zero entries (sparsity patterns) before corruption by gaussian noise with variance β^2. Vectors belonging to the same class have the same sparsity patterns. Note that the class membership only depends on the sparsity pattern; it is invariant from the actual values of the entries indexed by the sparsity pattern.

function (4.18) (Gauss-RBF in the tables) — which corresponds to vectorizing the tensors — with the tensorial kernel based on factors (4.26) (tensorial in the tables) for increasing values of N. We also compared with affine tensor-based models (4.1) with fixed rank-1 parametrization (linear rank-1). We use

quadratic loss as for the kernel-based classifiers and find the model via the alternating approach proposed in [204]. For the kernel-based procedures we tune the kernel parameter σ and regularization parameter γ based upon 10-fold cross-validated misclassification error. The same approach is used for the regularization parameter needed for linear rank-1 models.

Table 4.4: Accuracy on test data for $I = 7,\ \beta^2 = 0.05$

AUC performance: mean (and standard deviation)

N	tensorial (4.20)-(4.26)	Gauss-RBF (4.18)	linear rank-1 [204]
10	**0.86(0.04)**	0.53(0.07)	0.50(0.04)
14	**0.88(0.03)**	0.53(0.05)	0.51(0.03)
20	**0.88(0.09)**	0.61(0.10)	0.50(0.02)
28	**0.92(0.02)**	0.60(0.10)	0.50(0.02)
42	**0.94(0.02)**	0.63(0.10)	0.50(0.02)
60	**0.95(0.02)**	0.69(0.08)	0.50(0.01)
80	**0.96(0.02)**	0.73(0.07)	0.50(0.01)
110	**0.96(0.01)**	0.80(0.05)	0.50(0.01)
150	**0.97(0.01)**	0.84(0.04)	0.50(0.01)
200	**0.97(0.01)**	0.88(0.03)	0.50(0.01)

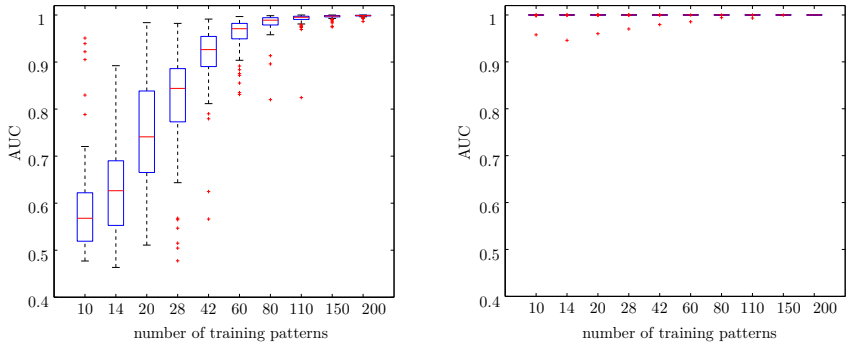
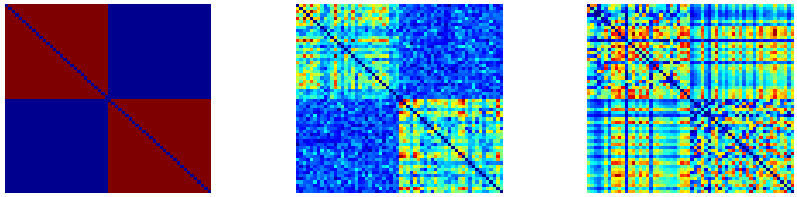


Figure 4.6: Synthetic example (classification of sparsity patterns, see also Figure 4.5), $I = 10,\ \beta^2 = 0.005$ and increasing number of training examples. Boxplots of AUC obtained over the same 200 test patterns for the Gaussian-RBF kernel (a) and for the tensorial kernel (b).



(a) (b) (c)

Figure 4.7: Synthetic example (classification of sparsity patterns, see also Figure 4.5), $I = 10$, $\beta^2 = 0.05$. Here kernel-target alignment appears from the pattern of off-diagonal entries of kernel matrices. (a) the 1-rank matrix $yy^\top$ obtained from training labels y. (b) the tensorial kernel matrix leading to superior classification accuracy (c) the Gaussian-RBF kernel.

Table 5.1 refers to the case of increasing values of N, $I = 7$ and $\beta^2 = 0.05$. We reported the mean value and standard deviation of the Area Under the receiver operating characteristic Curve (AUC) obtained across 100 random experiments. Each AUC was computed based upon the predicted labels of the same 200 test patterns. Similar results were obtained for the case where $I = 10$ and $\beta^2 = 0.005$. For this case Figure 4.6 reports the box plots of AUCs for the two RBF-type kernels. In all our experiments the linear rank-1 models consistently achieved random guessing performance. The same behavior was observed for the linear kernel (4.16) (not reported in Table 5.1). The tensorial kernel outperforms the Gaussian-RBF kernel showing that the proposed approach is useful even when the classes are only approximated by congruence sets (due to the fact that $\beta^2 \neq 0$). In general, the quantitative measure of kernel-target alignment proposed in [59] can reveal before training how well different kernel functions capture the structure of the problem. A good alignment often results in visually detectable patterns, see Figure 4.7. In general we observed that models based on the Gaussian-RBF kernel (which is *universal* [194]) also reach perfect classification accuracy when N is sufficiently large. This shows that exploiting the underlying invariance property is relevant especially for small sample size problems.

Recognition of Signals

We now present a simple example to illustrate Subsection 4.2.3. We generated two classes of real-valued signals corrupted by noise. Each class consisted of signals with different spectral content. Specifically, each signal s was a sequence

of the type $\{s_0, s_1, \cdots, s_{57}\}$ where

$$s_t = \sum_{k \in \mathbb{N}_{10}} \alpha_k \cos(2\Delta_y \pi t k/10) + 0.5\epsilon_t,\ \Delta_y = \begin{cases} 1 & \text{if } y = +1 \\ 1.01 & \text{if } y = -1 \end{cases}$$

and $\alpha \in \mathbb{R}^{10}$ was a vector of i.i.d. random variables drawn from a normal distribution. Notice that Δ_y in the previous is defined upon the signal's label. The latter was taken to be i.i.d. from a Bernoulli distribution with probability 0.5. Finally ϵ was a white noise sequence with normal distribution. Following this approach N signal-label pairs where generated for training. The 57-dimensional vector corresponding to the n-th training signal $s^{(n)}$ was either fed directly into kernels for vectors:

$$k\left(s^{(m_1)}, s^{(m_2)}\right) = \exp\left(-\sigma^2 \left\| s^{(m_1)} - s^{(m_2)} \right\|^2\right) \tag{4.42}$$

$$k\left(s^{(m_1)}, s^{(m_2)}\right) = \left\langle s^{(m_1)}, s^{(m_2)} \right\rangle \tag{4.43}$$

called respectively Gauss-RBF vec and linear vec, or first converted into an Hankel tensor $\mathcal{X}^{(n)} \in \mathbb{R}^{20} \times \mathbb{R}^{20} \times \mathbb{R}^{20}$ as explained in Section 4.2.3. For this latter tensorial representations we then used the Gaussian kernel (4.18) (Gauss-RBF), the linear kernel (4.16) (linear) and the simplified version of tensorial kernel that holds for Hankel tensors (4.34) (tensorial). We also considered affine tensor-based models (4.1) with fixed rank-1 parametrization (linear rank-1). The accuracy of the corresponding models, measured on the same set of 200 test patterns, were reported in Table 4.5. As in the previous example the tensorial kernel leads to far more accurate predictions in the low range of N. All the affine models (linear, linear vec, linear rank-1) achieve random guessing performance. Finally notice that Gauss-RBF vec outperforms Gauss-RBF. This is expected since vectorized Hankel tensors contain the same information as the vectors they are generated upon. In turn their dimensionality is considerably higher.

Libras Movement Data

Next we consider the Libras Movement Data Set [85] that contains different classes of hand movement type of LIBRAS (the Brazilian sign language). Each class consists of 24 bidimensional trajectories performed by the hand in a period of time (45 time instants for each hand movement). So each input pattern is a 45×2 matrix. We considered binary discrimination between different pairs of hand movement types. On the one hand each matrix was vectorized and fed into the same kernels for vectors considered in the previous Subsection (Gauss-RBF vec and linear vec). On the other hand based upon each row of the input

Table 4.5: Accuracy for the signals example

AUC performance: mean (and standard deviation)

N	tensorial (4.34)	Gauss-RBF (4.18)	linear rank-1 [204]
10	**0.88(0.04)**	0.54(0.06)	0.50(0.02)
14	**0.91(0.03)**	0.55(0.07)	0.50(0.03)
20	**0.93(0.05)**	0.64(0.09)	0.50(0.02)
28	**0.94(0.09)**	0.71(0.10)	0.50(0.02)
42	**0.97(0.01)**	0.77(0.12)	0.50(0.02)
60	**0.98(0.01)**	0.86(0.09)	0.50(0.02)
80	**0.98(0.01)**	0.73(0.07)	0.50(0.01)
110	**0.99(0.01)**	0.81(0.20)	0.50(0.01)
150	**0.99(0.01)**	0.83(0.20)	0.50(0.02)
200	0.99(0.01)	0.90(0.18)	0.50(0.02)

N	Gauss-RBF vec (4.42)	linear vec (4.43)	linear (4.16)
10	0.57(0.07)	0.50(0.03)	0.50(0.03)
14	0.64(0.08)	0.50(0.03)	0.50(0.03)
20	0.69(0.09)	0.50(0.03)	0.50(0.03)
28	0.75(0.09)	0.50(0.03)	0.50(0.04)
42	0.87(0.05)	0.50(0.03)	0.50(0.04)
60	0.93(0.03)	0.50(0.04)	0.50(0.05)
80	0.96(0.02)	0.50(0.04)	0.50(0.04)
110	0.98(0.01)	0.50(0.04)	0.50(0.04)
150	**0.99(0.01)**	0.50(0.04)	0.50(0.04)
200	**1.00(0.00)**	0.50(0.03)	0.50(0.04)

matrix, a 6×40 Hankel matrix was formed. The $6 \times 40 \times 2$ tensor obtained stacking together these 2 matrices has a partial Hankel structure [156] and features similar properties as the Hankel tensor we discussed in Section 4.2.3 for the case of univariate signals. This tensor representation was then used within kernels Gauss-RBF, linear and tensorial. Also rank-1 affine models were considered. For each binary classification task we compared the AUC curve obtained over 100 runs of LS-SVMlab. For each run we considered a different splitting into training and test set of the 48 time series available. In particular we take 8 for training and 40 for testing. Results for different pairs of classes are reported in Table 4.6.

Table 4.6: Accuracy on test data for Libras

AUC performance: mean (and standard deviation)

task	tensorial (4.20)-(4.26)	Gauss-RBF (4.18)	linear rank-1 [204]
1 vs 2	**0.83**(**0.07**)	0.76(0.11)	0.68(0.16)
1 vs 3	0.92(0.04)	**0.98**(**0.05**)	0.94(0.13)
1 vs 4	**1**(**0**)	0.98(0.05)	0.86(0.15)
1 vs 5	**1**(**0**)	0.97(0.06)	0.87(0.12)
1 vs 6	**1**(**0**)	0.95(0.07)	0.85(0.13)
task	linear (4.16)	Gauss-RBF vec (4.42)	linear vec (4.43)
1 vs 2	0.77(0.12)	0.75(0.11)	0.77(0.12)
1 vs 3	0.94(0.09)	**0.98**(**0.05**)	0.95(0.08)
1 vs 4	0.94(0.08)	0.98(0.03)	0.95(0.07)
1 vs 5	0.91(0.11)	0.97(0.06)	0.92(0.09)
1 vs 6	0.88(0.10)	0.95(0.06)	0.86(0.10)

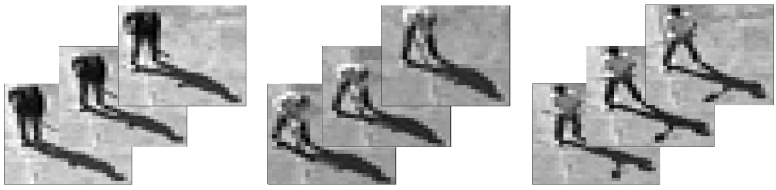
Table 4.7: Accuracy on test data for Aerial Views

AUC performance: mean (and standard deviation)

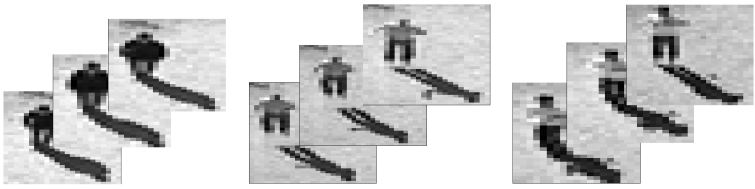
task	tensorial (4.20)-(4.26)	Gauss-RBF (4.18)
1 vs 2	**0.95**(**0.03**)	0.71(0.20)
3 vs 9	**1**(**0**)	0.70(0.25)
5 vs 6	**0.99**(**0.02**)	0.61(0.18)
7 vs 8	**0.95**(**0.05**)	0.58(0.17)
task	linear (4.16)	linear rank-1 [204]
1 vs 2	**0.95**(**0.06**)	0.79(0.20)
3 vs 9	0.99(0.04)	0.99(0.05)
5 vs 6	0.86(0.12)	0.82(0.14)
7 vs 8	0.92(0.09)	0.70(0.19)

Aerial Views

These experiments are about the Aerial View Activity Classification Dataset [46]. The goal is to discriminate between pairs of human actions from the given low-resolution grayscale videos, 12 per action. Each video is a 3−rd order tensor where the first two dimensions represent number of pixels of each frame and the third dimension is the number of frames, see Figure 4.8. As a preprocessing step we normalize the videos in the datasets. Each frame



(a) class 3 (digging)



(b) class 9 (jumping)

Figure 4.8: Examples of frames taken from low-resolution videos of human activities.

of each video is resampled to match the common size of 10×13 pixels. To cope with the different number of frames per video, we perform dimensionality reduction along the time mode and extract 4 eigen-images separately for all the videos. More precisely let $\tilde{\mathcal{X}}$ denotes the $10 \times 13 \times N$ tensor consisting of N frames. Denote by $\bar{\mathcal{X}}'_{<3>}$ the matrix obtained centering the columns of the $130 \times N$ matrix $\tilde{\mathcal{X}}'_{<3>}$. We compute from the $N \times N$ empirical covariance matrix $1/129\bar{\mathcal{X}}_{<3>}\bar{\mathcal{X}}'_{<3>}$ the 4 principal eigenvectors $\boldsymbol{E} = [e_1, \cdots, e_4] \in \mathbb{R}^N \otimes \mathbb{R}^4$ and finally obtain the $10 \times 13 \times 4$ data tensor $\mathcal{X}$ from reshaping $\bar{\mathcal{X}}'_{<3>}\boldsymbol{E}$. As a result of this normalization procedure for each binary classification task we are left with 24 $10 \times 13 \times 4$ input tensors and corresponding target labels. For each task we considered 8 tensors for training and the remaining 16 for testing. We compared the linear and Gaussian-RBF kernel with the tensorial kernel (4.20)-(4.26), linear kernel (4.16) and rank-1 models [204]. As before we averaged the results over 100 replicates obtained from random splittings of training and test set. Results for different pairs of classes are reported in Table 4.7.

4.3 Chapter Summary

In this Chapter we elaborated on non-parametric tensor-based models. We introduced an explicit multi-way feature representation, the application of which is hindered by non-convexity; we then elaborate on an effective framework based on a suitably define kernel function termed *tensorial kernel*. Contrary to naive kernels, the tensorial kernel takes into account the intrinsic geometry of data tensors by incorporating the Grassmannian nature of matrix unfoldings. The idea relates to the concepts of principal angles and the MLSVD. We have elaborated on an invariance property possessed by the proposed factor kernels and introduced the concept of congruence sets. From a pattern recognition viewpoint this is important because as soon classes are well approximated by congruence sets, improved classification accuracy is to be expected. This is in line with statistical learning results showing that good generalization takes place if similarity measures do capture the structure of the learning tasks of interest. In applications we estimated models via LS-SVMs; alternatively, the tensorial kernel could be used in combination with other kernel-based techniques.

Chapter 5

Classification of Signals with Cumulant-based Kernels

In the previous Chapter we developed the tensorial kernel in order to define a suitable similarity measure between objects represented through tensors. In this Chapter we focus on a specific class of such objects, namely multivariate time-series. Existing kernels that deal with these objects require the explicit estimation of models within a user-defined generative class. In contrast, the cumulant based kernel that we study here does not require an (often non-trivial) identification step. Although the applicability of the proposed kernel does not require postulating any specific model class, insightful properties can be derived as soon as certain generating models are assumed. In particular, we will draw the link between the proposed similarity measure and the latent state process in the case of multi-input multi-output (MIMO) linear and time-invariant state-space models. This provides useful insights on certain processes of interest as for the case of EG signals that we consider.

5.1 Problem Statement

Here we are concerned with classifying time-frames of vector processes. This task consists of correctly assigning a class-membership symbol $y \in \mathscr{Y}$ to an input pattern

$$\boldsymbol{X} = [x(T_0),\ x(1+T_0),\ \cdots,\ x(T-1+T_0)] \in \mathbb{R}^D \otimes \mathbb{R}^T \tag{5.1}$$

that represents the evolution along successive time instants of a real $D-$variate discrete-time stationary vector process $\{x(t)\}$. This is accomplished by means of a *classifier*, namely a function

$$\bar{f} : \mathbb{R}^D \otimes \mathbb{R}^T \to \mathscr{Y} \tag{5.2}$$

inferred based upon a dataset of N input-target[1] pairs

$$\mathcal{D}_N = \left\{ \left(\boldsymbol{X}^{(n)}, y_n \right) \in \mathbb{R}^D \otimes \mathbb{R}^T \times \mathscr{Y} \ : \ n \in \mathbb{N}_N \right\} \ . \tag{5.3}$$

It is assumed that elements of $\mathcal{D}_N$ are drawn in a i.i.d. manner from the same unknown joint distribution $p(\boldsymbol{X}, y)$ that generates the test pair $(\boldsymbol{X}, y)$. Note that the marginal distribution $p(\boldsymbol{X})$ (as well as the conditional distribution $p(\boldsymbol{X}|y)$ on $\mathbb{R}^D \otimes \mathbb{R}^T$) is induced by the underlying mechanism generating $\{x(t)\}$.

Next we discuss existing techniques for dealing with this task.

5.1.1 Discriminative Versus Generative Methods

Consider for simplicity a binary classification problem for which[2] $\mathscr{Y} = \{-1, 1\}$. In order to build a classifier (5.2) one approach is to directly find a function

$$f : \mathbb{R}^D \otimes \mathbb{R}^T \to \mathbb{R} \tag{5.4}$$

so that $\bar{f} = \operatorname{sign} \circ f$ is as close as possible to some minimizer $\bar{h}$ of a suitable performance criterion such as the expected risk based on the $0-1$ *loss*:

$$\bar{h} \in \arg\min \left\{ E_{XY} \left[\delta(h(\boldsymbol{X}), y) \right] \ \text{s.t.} \ h : \mathbb{R}^D \otimes \mathbb{R}^T \to \mathscr{Y} \right\}$$

where δ denotes the Kronecker delta. Techniques following this strategy are called discriminative methods, as we have seen in Section 2.1.4. In contrast, generative methods such as Hidden Markov models (HMMs) first model the underlying probability distribution $P(\boldsymbol{X}, y)$ and then use the Bayes rule for determining the class membership. In this situation the classification rule (5.2) is therefore the result of a multi-step process.

Discriminative methods have been shown to outperform generative methods in many applications; this applies in particular for the case where input data

[1] As it will be manifested later an input pattern $\boldsymbol{X}$ should be considered as formed upon a time frame extracted from what corresponds to — in the standard state-space nomenclature — the output process x (compare equation (5.1) with equation (5.11)). Therefore we reserve the word *output* to refer to the output process in a state-space representation whereas we use *target* to address the target label.

[2] Multiclass classification problems ($\mathscr{Y}$ is a discrete set with more than two elements) are generally more involved but similar strategies apply.

are naturally embedded in real vector spaces. The situation is less clear in those domains where input patterns carry on additional structure, for instance inherited by the dynamical nature of the data, as in (5.1). In this case, in fact, it is generally difficult to extract features or metric relations to be used within discriminative approaches [116].

5.1.2 Existing Kernels for Vector Processes

In the context of kernel methods the difficulty of discriminative methods to deal with non-Euclidean data is overcome by defining suitable kernels [87] that serve as a similarity measure between pairs of input data. In particular, certain kernels that have been proposed for structured data can be adapted to the case where input patterns are time frames of vector processes, as in (5.1). As we have seen in Section 4.2.4, within kernel methods (5.4) is a superposition of kernel functions. Within the present context it can be written as

$$f(\boldsymbol{X}) = \sum_{n \in \mathbb{N}_N} y_n \alpha_n k(p(\boldsymbol{X}^{(n)}), p(\boldsymbol{X})) + b \tag{5.5}$$

where, for any $n \in \mathbb{N}_N$, α_n weights the contribution of the $n-$th training data, $p : \mathbb{R}^D \otimes \mathbb{R}^T \to \mathscr{F}$ is a *processing step* that maps $\boldsymbol{X}$ into some metric space $\mathscr{F}$, k is a kernel function over $\mathscr{F}$ and finally b is a bias term. The approach proposed in [30], assumes that (5.1) is the output process of a linear non-Gaussian dynamical system. The processing step p corresponds to the explicit identification of a model as well as the recovery of the input process and the initial state associated to each output process. Remarkably, this deconvolution is a nontrivial step given that the generating systems might have non-minimal phase (as observed for the case of the human motion [30]). Finally k is a similarity measure over the parameter space of linear dynamical systems $\mathscr{F}$.

Related approaches are based on suitably defined instances of the *Fisher Kernel*. The Fisher Kernel represents a convenient way to incorporate generative models into discriminative classifiers [116, 215]. The key intuition behind it is that similar structured objects should induce similar log-likelihood gradients in the parameters of a predefined class of generative models [116]. Notably, as in [30], the use of Fisher kernels requires explicit estimation within a user-defined generative class.

Remark. In applications the choice of the generative class is often guided by computational tractability more than by principled insights. A sensible choice for multivariate time series is to assume *linear gaussian models* [171]. For this case, practical expectation maximization (EM) algorithms exist for the estimation of the model parameters [89, 171]. A derivation of a Fisher kernel within this framework can be found in [184].

5.2 Tensors and Cumulants

Next we give the definition of (tensor) cumulants and recall instrumental facts that we need in developing our cumulant-based kernel function.

5.2.1 Cumulants of Vector Processes

Cumulants are higher order statistics [151,152] that so far have received interest especially for the identification of linear and non-linear systems [80,90] as well as for blind source separation [52]. For a real-valued discrete-time and zero-mean stationary scalar process [101], the $J-$th order cumulant function $c^x : \mathbb{Z}^{J-1} \to \mathbb{R}$ is defined in terms of the so called *cumulant generating function* [161]. For a real $D-$variate discrete-time stationary vector process[3] $\{x(t)\}$ one needs the notion of *cross-cumulants* between scalar entries of the vector. Suppose $E[x(t)] = 0$. Then the second order cross-cumulant between the d_1-th and d_2-th entries of the vector process x is defined as:

$$c^x_{d_1 d_2}(l_1) := E[x_{d_1}(t) x_{d_2}(t + l_1)] \ . \tag{5.6}$$

Higher order cross-cumulants are defined in a similar fashion. In particular, simple formula exist up to the fourth order, see [184].

5.2.2 Cumulant Tensors

We now bring in the tensor representations of cumulants that we consider in the following. To begin with, for a given finite $J \in \mathbb{N}$ and any $i \in \mathbb{N}_{J-1}$, let $\mathscr{L}_i \subset \mathbb{Z}$ be an arbitrary finite set with cardinality S^i and denote by l^i_s its $s-$th element. Consider now the collection of $\prod_{i \in J-1} S^i$ integer tuples obtained as

$$\mathscr{L} := \mathscr{L}_1 \times \mathscr{L}_2 \times \cdots \times \mathscr{L}_{J-1} \ . \tag{5.7}$$

Definition 5.2.1 ($\mathscr{L}-$Cumulant Tensor)**.** We call $\mathscr{L}-$cumulant tensor, denoted as $\mathcal{C}^x_{\mathscr{L}}$, the
$(2J-1)-$th order tensor

$$\mathcal{C}^x_{\mathscr{L}} \in \underbrace{\mathbb{R}^D \otimes \mathbb{R}^D \otimes \cdots \otimes \mathbb{R}^D}_{J \text{ times}} \otimes \mathbb{R}^{S^1} \otimes \mathbb{R}^{S^2} \otimes \cdots \otimes \mathbb{R}^{S^{J-1}}$$

[3] Sometimes we simply indicate it as x.

defined entry-wise by

$$(\mathcal{C}^x_{\mathscr{L}})_{d_1 d_2 \cdots d_J s_1 s_2 \cdots s_{J-1}} := c^x_{d_1 d_2 \cdots d_J}(l^1_{s_1}, l^2_{s_2}, \ldots, l^{J-1}_{s_{J-1}}) = \tag{5.8}$$

$$E\left[x_{d_1}(t) x_{d_2}(t + l^1_{s_1}) \cdots x_{d_J}(t + l^{J-1}_{s_{J-1}})\right] \tag{5.9}$$

for any tuple $[d_1,\ d_2,\ \cdots,\ d_J,\ s_1,\ s_2, \cdots, s_{J-1}] \in \mathbb{N}_D \times \mathbb{N}_D \cdots \times \mathbb{N}_D \times \mathbb{N}_{S^1} \times \mathbb{N}_{S^2} \times \cdots \times \mathbb{N}_{S^{J-1}}$.

Example 5.2.1. As a concrete example suppose that x is a zero-mean $4-$variate ($D = 4$) stationary process. Take $J = 3$ and let $\mathscr{L}_1 = \{-3, 1\}$ and $\mathscr{L}_2 = \{-3, 0, 2\}$. In this case $\mathscr{L} = \{(-3,-3), (-3,0),\ (-3,2), (1,-3), (1,0), (1,2)\}$, and each entry of the cumulant tensor $\mathcal{C}_{\mathscr{L}}$, indexed by a tuple $[d_1,\ d_2,\ d_3,\ s_1, s_2] \in \mathbb{N}_4 \times \mathbb{N}_4 \times \mathbb{N}_4 \times \mathbb{N}_2 \times \mathbb{N}_3$, is a third order cross-cumulant between the d_1-th, d_2-th and d_3-th entries of the vector process x.

A special case that arises from the previous definition corresponds to the situation where for any $i \in \mathbb{N}_{J-1}$, the set $\mathscr{L}_i$ is a singleton. In this case $\mathscr{L}$ is also a singleton, i.e., we have $\mathscr{L} = \left\{[l^1_1,\ l^2_1,\ \ldots,\ l^{J-1}_1]\right\}$ for a single fixed tuple of integers $[l^1_1,\ l^2_1,\ \ldots,\ l^{J-1}_1]$. Additionally $J-1$ dimensions of $\mathcal{C}^x_{\mathscr{L}}$ are degenerate, that is, $\mathcal{C}^x_{\mathscr{L}}$ can be equivalently represented as a $J-$th order tensor. This is formalized in the next Definition.

Definition 5.2.2 (($l_1,\ l_2,\ \ldots,\ l_{J-1}$)-Cumulant Tensor)**.** We call ($l_1,\ l_2,\ \ldots,\ l_{J-1}$)-cumulant tensor, denoted as $\mathcal{C}^x(l_1, l_2, \ldots, l_{J-1})$, the $J-$th order tensor

$$\mathcal{C}^x(l_1, l_2, \ldots, l_{J-1}) \in \underbrace{\mathbb{R}^D \otimes \mathbb{R}^D \otimes \cdots \otimes \mathbb{R}^D}_{J \text{ times}}$$

for any tuple

$$(\mathcal{C}^x(l_1, l_2, \ldots, l_{J-1}))_{d_1 d_2 \cdots d_J} := c^x_{d_1 d_2 \cdots d_J}(l_1, l_2, \ldots, l_{J-1})\ . \tag{5.10}$$

In the remainder of the Chapter we simply write *cumulant tensor* when the specific notion that we are referring to is clear from the context.

5.2.3 Sample versions

We write, respectively, $\hat{\mathcal{C}}^x_{\mathscr{L}}$ and $\hat{\mathcal{C}}^x(l_1, l_2, \ldots, l_{J-1})$ to mean the empirical (a.k.a. sample) versions of the cumulant tensors introduced in the previous definitions.

These are simply obtained by replacing the expectation operator by the *time average* in the definitions of cross-cumulants (see equation (5.6) and [184]) [83, 201]. For instance, suppose that $\{x(1), x(2), \cdots, x(T)\}$ is a given sample from the process x. Then if we let $T_0 := \max\{1, 1 - l_1\}$ and $T' := \min\{T, T - l_1\}$, the empirical version of (5.6) is obtained as:

$$\hat{c}^x_{d_1 d_2}(l_1) := \frac{1}{T' - T_0 + 1} \sum_{t=T_0}^{T'} x_{d_1}(t) x_{d_2}(t + l_1) \ .$$

We also assume (as done everywhere else, at least implicitly) that sample versions of cumulants do converge towards their theoretical counterpart for $T \to \infty$. This is true, in particular, whenever x is *strictly stationary* and *ergodic*[4] and $E[x_d^2(t)] < \infty$ for all $d \in \mathbb{N}_D$ [101, Section IV.2, Theorem 2]. However, as known, the property of ergodicity is not observationally verifiable from a single realization of a time series [101].

We now link a class of time series with their generating dynamical system. This will allow us to derive useful properties of the cumulant-based similarity measure that we introduce in the next Section.

5.2.4 Cumulants of State-Space Processes

Suppose that, for the given stationary process x, the *z-spectrum*[5] $S(z) := \sum_{l \in \mathbb{Z}} \mathcal{C}^x(l) z^{-l}$ is a rational function and has maximal normal rank everywhere on the complex unit circle. In this case a fundamental result [121] shows that there are $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{D_z} \otimes \mathbb{R}^{D_z}, \boldsymbol{C} \in \mathbb{R}^D \otimes \mathbb{R}^{D_z}$ and white processes[6] w (input process) and v (noise process) such that x can be interpreted as the output process of the MIMO stable system $\Sigma = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ in state-space form:

$$z(t+1) = \boldsymbol{A} z(t) + \boldsymbol{B} w(t) \tag{5.11}$$

$$x(t) = \boldsymbol{C} z(t) + v(t) \ . \tag{5.12}$$

Conversely starting from the generating dynamical system (5.11)-(5.12) we have the following basic result.

[4] Different is the situation of signals driven by *cyclostationary* sources that we do not consider here. In this situation, that arise for instance in the context of radiocommunications, it is known that the empirical estimators arising from time averages are *biased*. See, e.g., [82] and [52, Section 17.2.2.2].

[5] Notice that $\mathcal{C}^x(l)$, corresponds to $E[x(t)x(t+l)^\top]$.

[6] We recall that a stationary stochastic process $x(t)$ is *white* if $E[x(i)x^\top(j)] = \boldsymbol{Q}\delta_{ij}$ where δ is the Kronecker delta function and $\boldsymbol{Q}$ denotes some covariance matrix.

Proposition 5.2.1 (State-space Model: Cumulant of the Output Process)**.** *Assume a stable state-space model* (5.11)-(5.12) *specified as above and suppose additionally that the processes w and v are mutually independent. For any tuple $[l_1,\ l_2,\ \ldots,\ l_{J-1}] \in \mathbb{Z}^{J-1}$ we have, in steady state*

1. $$\mathcal{C}^x(l_1, l_2, \ldots, l_{J-1}) = \mathcal{C}^z(l_1, l_2, \ldots, l_{J-1}) \times_1 \boldsymbol{C} \times_2 \boldsymbol{C} \times_3 \cdots \times_{J-1} \boldsymbol{C} + \mathcal{C}^v(l_1, l_2, \ldots, l_{J-1}) \quad . \quad (5.13)$$

2. *If the noise process v is Gaussian and $J > 2$ then $\mathcal{C}^v(l_1, l_2, \ldots, l_{J-1})$ vanishes in* (5.13)*.*

Proof. In steady state the state vector results from a linear filtering of the input process w and hence is statistically independent of the noise process v. Point 1 follows immediately from basic properties of cumulants, namely multilinearity and their additivity under the independence assumption (see, e.g., [68, Sec. 2.2 properties 2 and 5]). Point 2 is a well known property of cumulants of a white Gaussian process [152]. □

Results in line with Proposition B.4.2 can be found elsewhere. In particular, cumulants of the output process of a stable single-input single-output linear time-invariant system were studied in [36]. Cumulants of a broader class of linear MIMO systems (time-varing with possibly non-stationary inputs) were the subject of [202]. It is worth recalling that if a real random variable has symmetric distribution about the origin, then all its odd cumulants vanish [152]. In many cases this last condition is naturally verified for the entries of the state vector; in light of this fact and of point 2 of Proposition B.4.2 one then considers $J \geq 4$ in applications.

Generally speaking, it becomes harder to estimate cumulants from sample data as the order increases, i.e. longer data sets are required to obtain the same accuracy [31, 83, 201]. In the following we stick to $J = 4$, also used in the context of system identification [131]. This choice ensures that the output cumulant is is independent of the noise process.

5.3 Cumulant-based Kernels

Recall from Section 4.2.2, that the expression for the tensorial kernel between two generic tensors $\mathcal{X}$ and $\mathcal{Y}$ is $k(\mathcal{X}, \mathcal{Y}) = \Pi_{m \in \mathbb{N}_M} k^m(\mathcal{X}, \mathcal{Y})$ where the factor

k^m is defined upon the $m-$th matrix unfoldings:

$$k^m(\mathcal{X},\mathcal{Y}) = \exp\left(-\frac{1}{2\sigma^2}\left\|\boldsymbol{V}_{\mathcal{X},1}^{(m)}\boldsymbol{V}_{\mathcal{X},1}^{(m)\top} - \boldsymbol{V}_{\mathcal{Y},1}^{(m)}\boldsymbol{V}_{\mathcal{Y},1}^{(m)\top}\right\|^2\right) . \tag{5.14}$$

5.3.1 Cumulant-based Kernel Functions

We consider now specific types of tensors, i.e., $\mathscr{L}-$cumulant tensors as introduced above. Notice that, since we stick to the case of fourth order cumulants, these tensors have in general order 7. One has $\mathscr{L} := \mathscr{L}_1 \times \mathscr{L}_2 \times \mathscr{L}_3$ where, for any $i \in \{1,2,3\}$, $\mathscr{L}_i$ is an arbitrary finite subset of $\mathbb{Z}$. Given (5.14) it is now straightforward to introduce a cumulant-based kernel $k_{\mathscr{L}} : \mathbb{R}^D \otimes \mathbb{R}^T \times \mathbb{R}^D \otimes \mathbb{R}^T \to \mathbb{R}$ according to:

$$k_{\mathscr{L}}(\boldsymbol{X}_1,\boldsymbol{X}_2) := \prod_{p\in\mathbb{N}_P} k^p(\hat{\mathcal{C}}_{\mathscr{L}}^{x_1},\hat{\mathcal{C}}_{\mathscr{L}}^{x_2}) \tag{5.15}$$

where the empirical versions of the cumulants are computed based upon the observed time frames $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$.

Remark. In (5.5) we introduced the general kernel-based model with p playing the role of a processing step. We note here that in (5.15) the processing step p corresponds to the evaluation of empirical expectations (see Subsection 5.2.3). Contrary to the kernels for structured data discussed in Subsection 5.1.2, no additional estimation steps are required.

Remark. A careful look at equations (5.8) and (5.10) reveals that entries of the cumulant tensors depend only on time lags and channel indices. Therefore (5.15) does not require that $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are aligned in time nor that the two sequences have the same length [7].

5.3.2 Computational Aspects

The freedom in constructing $\mathscr{L}$ gives rise to a whole family of cumulant-based kernel functions. For general $\mathscr{L}$, however, the evaluation of $k_{\mathscr{L}}$ at $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ requires computing the SVD of 14 matrices (7 for both arguments). The sizes of these matrices depend upon D and the cardinalities of $\mathscr{L}_i, i \in \mathbb{N}_3$. Since in many practical applications D is of the order of tens or even hundreds, $k_{\mathscr{L}}$ might be prohibitive to evaluate. We refer to Section 4.2.4 for an analysis on

[7] This is because we consider stationary and ergodic signals. Note that in Section 5.1 we have required for simplicity that time-frames of multichannel signals refer to the same time instants (see equation (5.1)). A more general problem formulation amounts at relaxing this condition.

the computation complexity for the general tensorial kernel. In the current setting we can make the following observations.

- The case where $\mathscr{L}$ is singleton (Definition 5.2.2) implies a significant reduction in the order of the tensor: in this case $\mathcal{C}^{x_1}_{\mathscr{L}}$ and $\mathcal{C}^{x_2}_{\mathscr{L}}$ can be equivalently represented as 4-th order tensors. For this reason in applications we consider $\mathcal{C}^x(0,0,0)$ namely, we take only the zero-lags tuple ($\mathscr{L} = \{[0,0,0]\}$)[8]. Note that in this case all the tensor matricizations have the same dimensions which depend upon D.

- An additional reduction can be obtained by fixing a *reference signal*, indexed by $r \in \mathbb{N}_D$. This is achieved by replacing $\hat{\mathcal{C}}^x(0,0,0)$ with $\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}}$ defined entry-wise by:

$$(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}})_{d_1 d_2 d_3} := \left(\hat{\mathcal{C}}^x(0,0,0)\right)_{d_1 d_2 d_3 r} . \tag{5.16}$$

  Notice that $\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}}$ is a third order tensor whereas $\hat{\mathcal{C}}^x(0,0,0)$ is a fourth order tensor.

- $\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}}$ is a *symmetric* tensor[9]. This implies that $(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}})_{\langle n \rangle} = (\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}})_{\langle m \rangle}$ for any $n, m \in \mathbb{N}_3$. Therefore the expression for the tensorial kernel (5.15) when $\hat{\mathcal{C}}^x_{\mathscr{L}}$ is replaced by $\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}}$ can be simplified to:

$$k_{\mathscr{L}}(\boldsymbol{X}_1, \boldsymbol{X}_2) = \exp\left(-\tfrac{1}{2\sigma^2}\left\|\boldsymbol{V}^{(1)}_{\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_1},1}\boldsymbol{V}^{(1)\top}_{\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_1},1} - \boldsymbol{V}^{(1)}_{\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_2},1}\boldsymbol{V}^{(1)\top}_{\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_2},1}\right\|^2\right) \tag{5.17}$$

  that is, only the first (or any other) factor needs to be computed.

- Let $\boldsymbol{Z} = \boldsymbol{V}^{(1)\top}_{\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_1},1}\boldsymbol{V}^{(1)}_{\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_2},1}$. Simple matrix algebra shows that (5.17) is equivalent to

$$k_{\mathscr{L}}(\boldsymbol{X}_1, \boldsymbol{X}_2) = \exp\left(-\frac{1}{\sigma^2}(I_D - \mathrm{trace}(\boldsymbol{Z}^\top \boldsymbol{Z})\right) \tag{5.18}$$

  which is more efficiently computed.

The procedure in Algorithm 1 summarizes the steps required to evaluate the tensorial kernel for the input patterns in $\mathcal{D}_N$ when $\mathcal{L}$ is the zero-lags tuple and a reference signal is used. The order reduction entailed by using a reference signal directly impacts on the computation at line (i) and at line (ii) of Algorithm

[8] A similar restriction is generally considered in the context of blind source separation [82].

[9] This simply means that for all $[d_1,\ d_2,\ d_3] \in \mathbb{N}^3_D$, $(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}})_{d_1 d_2 d_3} = (\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}})_{d_{P(1)} d_{P(2)} d_{P(3)}}$ where P is any permutation of the set $\{1,2,3\}$.

Algorithm 1: KERNELMATRIX$(\mathcal{D}_N, \sigma, r)$

for each $n \in \mathbb{N}_N$

comment: Precompute cumulant tensors and SVD's

$$\textbf{do} \begin{cases} \mathcal{A}^{(n)} & \leftarrow \mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}^{(n)}} (\text{see (5.16), (5.8) and [185]}) \ (i) \\ \boldsymbol{V}_{\mathcal{A}^{(n)},1} & \leftarrow \mathrm{SVD}(\mathcal{A}^{(n)}_{<1>}) \qquad\qquad (ii) \end{cases}$$

for each $n, m \in \mathbb{N}_N$ and $m > n$

comment: Compute entries of the kernel matrix $\boldsymbol{K}$

$$\textbf{do} \begin{cases} \boldsymbol{Z} & \leftarrow \boldsymbol{V}^{\top}_{\mathcal{A}^{(n)},1} \boldsymbol{V}_{\mathcal{A}^{(m)},1} \\ (\boldsymbol{K}_r)_{nm} & \leftarrow \exp\left(-\frac{1}{\sigma^2}(I_D - \mathrm{trace}(\boldsymbol{Z}^{\top}\boldsymbol{Z})\right) \end{cases}$$

$\boldsymbol{K}_r \leftarrow \boldsymbol{K}_r + \boldsymbol{K}_r^{\top} + \boldsymbol{I}_N$

return $(\boldsymbol{K}_r)$

1. Finally note that alternative kernel methods differ in the way they use the kernel matrix $\boldsymbol{K}_r$ to carry on the estimation of the parameters $\{\alpha_n \ : \ n \in \mathbb{N}_N\}$ and b in (5.5). In experiments we use Least Squares Support Vector Machines (LS-SVM's) that we briefly recalled in Subsection 4.2.4.

Remark. To give a concrete example of the time complexity, consider the situation where there are $D = 50$ channels; in this case each signal is represented as a $50 \times 50 \times 50$ cumulant tensor. Computing a 100×100 kernel matrix takes approximately 30 minutes with a standard laptop and a MATLAB implementation of the aforementioned approach. As computing the kernel matrix dominates the whole process, this is also a rough estimate of the full time required by training an LS-SVM classifier (including model selection) for the case where there are 100 training patterns.

5.3.3 Connection with the System Dynamics

The properties of cumulants recalled in the previous Section enable us to bridge the gap between the kernel constructed on the observed outputs and the dynamics of the generating systems. In particular, we have the following *invariance properties.*

Theorem 5.3.1 (Asymptotical invariances)**.** *Assume systems* $\Sigma_1 = (\boldsymbol{A}_1, \boldsymbol{B}_1, \boldsymbol{C}_1)$ *and* $\Sigma_2 = (\boldsymbol{A}_2, \boldsymbol{B}_2, \boldsymbol{C}_2)$ *driven by, respectively, mutually independent white ergodic processes* (w_1, v_1) *and mutually independent white ergodic processes* (w_2, v_2)*. Additionally assume that* v_1 *and* v_2 *are Gaussian. Let* $\boldsymbol{X}_1 \in \mathbb{R}^D \otimes \mathbb{R}^T$ *and* $\boldsymbol{X}_2 \in \mathbb{R}^D \otimes \mathbb{R}^T$ *be given realizations from time-frames of the output processes taken from the steady-state behavior of the two systems. Then for* $T \to \infty$ *and any* $i \in \mathbb{N}_3$

1. *if* $\boldsymbol{C}_1 = \boldsymbol{C}_2$ *the evaluation of the* $p-$*th factor can be restated as:*

$$k^p(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_1}, \mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_2}) = \exp\left(-\tfrac{1}{2\sigma^2} g\left(\boldsymbol{V}^{(p)}_{\mathcal{C}^{z_1}(0,0,0),1}, \boldsymbol{V}^{(p)}_{\mathcal{C}^{z_2}(0,0,0),1}\right)\right) \tag{5.19}$$

   and hence depends only on the fourth order cross-cumulants of the state processes.

2. *if* $\boldsymbol{X}'_1, \boldsymbol{X}'_2 \in \mathbb{R}^D \otimes \mathbb{R}^T$ *are also output realizations obtained from systems* Σ_1 *and* Σ_2 *respectively, then we have:*

   (a) $k^p(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}'_1}, \mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}'_2}) = k^p(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_1}, \mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_2})$

   (b) $k^p(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}'_1}, \mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_1}) = 1$ *and* $k^p(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}'_2}, \mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_2}) = 1.$

Proof. We indicate here by $\boldsymbol{X}$ (resp. $\boldsymbol{X}'$) the generic output process, either $\boldsymbol{X}_1$ or $\boldsymbol{X}_2$ (resp. $\boldsymbol{X}'_1$ or $\boldsymbol{X}'_2$). First of all notice that, since the output processes are ergodic, for $T \to \infty$ we have:

$$(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}})_{d_1 d_2 d_3} \to (\mathcal{C}^x(0,0,0))_{d_1 d_2 d_3 r}$$

for any $[d_1,\ d_2,\ d_3] \in \mathbb{N}^3_D$. In particular, this implies that $\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}} \equiv \mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}'}$ from which the equations in 2.a and 2.b immediately follow. In order to proof the first part notice now that, without loss of generality[10], we can consider $\boldsymbol{C}$ to be orthogonal. By Proposition B.4.2 we have:

$$\mathcal{C}^x(0,0,0) = \mathcal{C}^z(0,0,0) \times_1 \boldsymbol{C} \times_2 \boldsymbol{C} \times_3 \cdots \times_{J-1} \boldsymbol{C} + \mathcal{C}^v(0,0,0)$$

where $\mathcal{C}^v(0,0,0)$ vanishes by point 2 in Proposition B.4.2. Denote by $\boldsymbol{A} \otimes \boldsymbol{B}$ the Kronecker product between any two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The $i-$th mode unfolding $(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}})_{<i>}$ corresponds now to

$$(\mathcal{C}^x(0,0,0))_{<i>} = \boldsymbol{C}\,(\mathcal{C}^z(0,0,0))_{<i>}\,(\boldsymbol{C} \otimes \boldsymbol{C} \otimes \boldsymbol{C})^\top$$

[10] In fact, as well known, there is an entire class of state-space representations that are equivalent up to similarity transformations.

where we used a basic property on the matricization operation [67,163]. Notice that $\boldsymbol{C} \otimes \boldsymbol{C} \otimes \boldsymbol{C}$ is an orthogonal matrix and hence if

$$\boldsymbol{U}^{(i)}_{\mathcal{C}^z(0,0,0),1} \boldsymbol{S}^{(i)}_{\mathcal{C}^z(0,0,0),1} \boldsymbol{V}^{(i)\top}_{\mathcal{C}^z(0,0,0),1}$$

is the thin SVD [94] factorization of $(\mathcal{C}^z(0,0,0))_{<i>}$ then for the right singular vectors $\boldsymbol{V}^{(i)}_{\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}},1}$ we have

$$\boldsymbol{V}^{(i)}_{\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}},1} = (\boldsymbol{C} \otimes \boldsymbol{C} \otimes \boldsymbol{C})\, \boldsymbol{V}^{(i)}_{\mathcal{C}^z(0,0,0),1}. \tag{5.20}$$

Replacing (5.20) in the formula that gives $k^i(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_1}, \mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_2})$ and exploiting the orthogonality of $\boldsymbol{C} \otimes \boldsymbol{C} \otimes \boldsymbol{C}$ now proofs (5.19). □

The former result is best understood in light of a concrete application that we consider next.

5.4 The Case of EG Signals

State-space methods have proven a valuable tool in the wide domain of neural data analysis [154]. In particular, a state-space framework for multichannel electroencephalography (EEG) signals was proposed in [34, 48] with the purpose of modeling cortical connectivity. We argue that a similar state-space representation is plausible also to describe magnetoencephalography (MEG) signals; we call *EG signals* the general class of encephalography signals. For EG signals, (5.11)-(5.12) is assumed to describe the physics behind the cortical dynamics. Specifically, the state process z represents the latent cortical signals. The output process x represents the evolution in time of the electric potentials (the multichannel EEG signals) measured at the scalp. Equation (5.12) expresses them as a noisy linear combination of the cortical signals.

5.4.1 Classification Problems Involving EG Signals

There are multiple classification problems within this context. For instance, the N training input patterns might refer to multichannel EG signals measured from N different subjects on certain time windows. In this case, each input (a time frame of the output process) is labelled according to the class to which the corresponding subject belongs. For experiments on a single subject, on the other hand, one might assume that the subject-specific system remains unchanged whereas what changes is the cortical activity driven by external stimuli belonging to different categories (different stochastic processes).

5.4.2 Interpretation of Theorem 5.3.1

For the case of EG signals Theorem 5.3.1 reads as follows. The first part means that, although the factor kernel $k^p(\mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_1}, \mathcal{C}^{\mathrm{r}}_{4\boldsymbol{X}_2})$ (and, as a consequence, the kernel function (5.17)) is defined on the measured EG signals, under the given assumptions it depends uniquely on (cumulants of) the cortical signals. Therefore it works as a similarity measure between the latent mental states. The second part of Theorem 5.3.1 is a consequence of the fact that the kernel is defined upon a statistic (in fact, the cumulant function) and is therefore invariant with respect to the sample signal which is observed (see also Remark 5.3.1). In particular, the equations in 2.b state that two time frames (for $T \to \infty$) are *maximally similar* if they are generated by the same systems (e.g., healthy brains), possibly driven by different realizations of the same stochastic process.

5.5 Experiments

In this Section we compare the results obtained with alternative classification approaches. Our primary goal is to evaluate the performance of discriminative classifiers based on the cumulant-based kernel function (5.18) (denoted as sub$-\mathcal{C}$) against those obtained based on a number of alternative similarity measures. As (5.18), the first group of alternative kernels we considered is also based on zero-lags cumulant tensors with a reference signal. These are:

$$k_{\mathrm{RBF}-\mathcal{C}}(\boldsymbol{X}, \boldsymbol{Y}) := \exp\left(-\frac{1}{2\sigma^2}\|\mathcal{C}^r_{4\boldsymbol{X}} - \mathcal{C}^{\mathrm{r}}_{4\boldsymbol{Y}}\|^2\right) \tag{5.21}$$

$$k_{\mathrm{lin}-\mathcal{C}}(\boldsymbol{X}, \boldsymbol{Y}) := \langle \mathcal{C}^r_{4\boldsymbol{X}}, \mathcal{C}^{\mathrm{r}}_{4\boldsymbol{Y}} \rangle \, . \tag{5.22}$$

Contrary to sub$-\mathcal{C}$, these kernels do not exploit the algebraic structure of tensors [183]. We also tested an instance of the Fisher kernel based upon linear Gaussian system with state dimension d (denoted as Fisher$-d$; see [184]). This kernel requires the identification of models. We used the implementation of the EM algorithm for linear Gaussian systems due to Zoubin Ghahramani[11]. Additionally, we considered kernels that simply act on the concatenation of the

[11] Available at `http://www.gatsby.ucl.ac.uk/~zoubin/software.html`.

vectors corresponding to each channel:

$$k_{\mathrm{RBF}}(\boldsymbol{X},\boldsymbol{Y}) := \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{X}-\boldsymbol{Y}\|^2\right) \tag{5.23}$$

$$k_{\mathrm{lin}}(\boldsymbol{X},\boldsymbol{Y}) := \langle \boldsymbol{X},\boldsymbol{Y}\rangle\ . \tag{5.24}$$

Notice that the latter kernel functions do not exploit the dynamical structure of the data. Nevertheless they are widely used in the context of machine learning to deal with multivariate time series. As discriminative classifiers in all our experiments we used Least Squares Support Vector Machines (LS-SVM's, Subsection 4.2.4). Finally we tested generative classifiers and considered the same model class as for the Fisher kernel. With these methods classification was obtained as follows. Assume class-membership symbols $\mathscr{Y} = \{s_1, s_2, \ldots, s_K\}$. To classify a sequence into one of the K classes, we trained K HMMs with state dimension d (denoted as HMM$-d$), one per class, and then computed the log-likelihood that each model gave to the test sequence; if the $k-$th model was the most likely, then we declared the class of the sequence to be s_k.

5.5.1 Synthetic Examples: 3-class problems

First we validate the proposed approach on synthetic examples where we generate time series according to class-specific models.

Generative Method

For $n \in \mathbb{N}_N$ we let y_n be drawn from a categorical distribution with set of symbols $\mathcal{Y} = \{1,2,3\}$ and success probabilities $p_1 = p_2 = p_3 = \frac{1}{3}$. Denote by $\mathcal{U}([a,b]^p)$ the uniform distribution on $[a,b]^p$ and by $\mathcal{N}(0_p, \boldsymbol{I}_p)$ the $p-$variate normal distribution. For any y_n we generated the corresponding training output processes $\bar{x}^{(n)}$ according to

$$\begin{aligned} z^{(n)}(t+1) &= \boldsymbol{A}_{(n)} z^{(n)}(t) + \boldsymbol{B} w^{(n)}(t), \\ \bar{x}_i^{(n)}(t) &= \left(\boldsymbol{C} z^{(n)}(t)\right)_i^{\tau} + v_i^{(n)}(t), \qquad i \in \mathbb{N}_5 \end{aligned}$$

where for any n and t, $w^{(n)}(t) \sim \mathcal{U}([0,10]^3)$ and $v^{(n)}(t) \sim \mathcal{N}(0_5, \boldsymbol{I}_5)$. We considered three distinct cases, namely $\tau = 1$ (*linear state-space models*), $\tau = 2$ and $\tau = 3$ (*nonlinear state-space models*). Additionally note that we have considered the case where each model, indexed by n, is constructed around one

of the three class prototypes but does not coincide with it[12]. Specifically, the system matrices $\boldsymbol{A}_{(n)}, \boldsymbol{B}_{(n)} \in \mathbb{R}^3 \otimes \mathbb{R}^3$ and $\boldsymbol{C} \in \mathbb{R}^5 \otimes \mathbb{R}^3$ were defined as follows:

$$\left(\tilde{\boldsymbol{A}}_{(n)}\right)_{ij} = \begin{cases} \sin(ij), & \text{if } y_n = 1 \\ \cos(ij), & \text{if } y_n = 2 \\ \sin(ij+\pi)^3, & \text{if } y_n = 3 \end{cases},$$

$$R^{(n)} = \operatorname{diag}\left(\left[-0.5 + F_1^{(n)}, -0.2 + F_2^{(n)}, 0.6 + F_3^{(n)}\right]\right),$$

$$\left(\boldsymbol{B}_{(n)}\right)_{ij} = \sin(ij) + F_4^{(n)}, \quad (\boldsymbol{C})_{ij} = \cos(ij),$$

$$\boldsymbol{A}_{(n)} = \tilde{\boldsymbol{A}}_{(n)} R^{(n)} \tilde{\boldsymbol{A}}_{(n)}^{-1},$$

where for any n, $F^{(n)} \sim \mathcal{U}([-0.4, 0.4]^4)$. In generating the output measurements we let $t \in \mathbb{N}_{2000}$ and formed $\bar{\boldsymbol{X}}^{(n)} \in \mathbb{R}^5 \otimes \mathbb{R}^{600}$ based upon the last 600 time instants. Finally, for any n we took as input pattern $\boldsymbol{X}^{(n)}$ defined column-wise as[13] $x_i^{(n)} = \bar{x}_i^{(n)} - \frac{1}{600}\bar{\boldsymbol{X}}^{(n)} 1_{600}$ for $i \in \mathbb{N}_{600}$.

This scheme was used to draw the N training pairs as well as 200 input-output pairs used across different experiments for the purpose of comparing the different procedures (test set).

Test Procedure

For the generative classifiers we trained 3 (one per class) HMMs with state dimension d using the appropriate subset of the N training patterns. For the cumulant-based kernels we let $r = 1$, that is, we used the first output channel as the reference signal. All the kernels where used within discriminative classifiers that were trained upon the N training samples via the LS-SVMlab toolbox (`www.esat.kuleuven.be/sista/lssvmlab`, [63]). We used the default one-versus-one multiclass encoding and 10-fold cross validation to perform model selection (namely the choice of tuning parameter γ in (C.11) and — where

[12] In many applications, in fact, it is reasonable to assume that measurements within the same class might still differ in the way they have been generated. Consider for instance the case of EG signals taken from different subjects. In this case one observes intra-patients variability even within the same class (e.g., healthy patients).

[13] It is worthy remarking at this point that cumulants are defined for zero-mean vector processes, see Section 5.2.2 and [184].

Table 5.1: Synthetic examples: misclassification errors on test data

$\tau = 1$ (linear state-space models)

kernel/method	$N = 80$	$N = 160$	$N = 320$	$N = 640$
sub$-\mathcal{C}$	0.57 (0.04)	0.54 (0.02)	0.51 (0.02)	0.49 (0.01)
lin$-\mathcal{C}$	0.65 (0.01)	0.64 (0.02)	0.62 (0.02)	0.61 (0.02)
RBF$-\mathcal{C}$	0.57 (0.02)	0.56 (0.02)	0.54 (0.02)	0.54 (0.01)
lin	0.67 (0.02)	0.68 (0.02)	0.68 (0.02)	0.67 (0.02)
RBF	0.64 (0.02)	0.64 (0.03)	0.61 (0.02)	0.61 (0.01)
fisher-2	0.42 (0.04)	0.59 (0.12)	0.65 (0.03)	0.66 (0.02)
fisher-3	0.38 (0.04)	0.60 (0.09)	0.65 (0.03)	0.66 (0.03)
fisher-4	0.35 (0.04)	0.56 (0.12)	0.62 (0.03)	0.66 (0.05)
HMM-2	0.38 (0.04)	0.37 (0.05)	0.38 (0.05)	0.36 (0.01)
HMM-3	**0.27** (0.03)	0.28 (0.04)	0.25 (0.02)	**0.24** (0.01)
HMM-4	**0.27** (0.03)	**0.26** (0.02)	**0.24** (0.02)	**0.24** (0.01)

$\tau = 2$ (nonlinear state-space models)

kernel/method	$N = 80$	$N = 160$	$N = 320$	$N = 640$
sub$-\mathcal{C}$	0.14 (0.03)	0.10 (0.02)	**0.07** (0.02)	**0.05** (0.01)
lin$-\mathcal{C}$	0.40 (0.12)	0.38 (0.18)	0.45 (0.18)	0.50 (0.14)
RBF$-\mathcal{C}$	0.66 (0.02)	0.67 (0.02)	0.67 (0.02)	0.67 (0.01)
lin	0.66 (0.02)	0.64 (0.03)	0.62 (0.03)	0.61 (0.03)
RBF	0.66 (0.02)	0.67 (0.02)	0.67 (0.02)	0.68 (0.01)
fisher-2	0.41 (0.24)	0.41 (0.21)	0.49 (0.05)	0.50 (0.20)
fisher-3	0.41 (0.20)	0.33 (0.17)	0.42 (0.15)	0.47 (0.14)
fisher-4	0.46 (0.13)	0.48 (0.11)	0.58 (0.15)	0.53 (0.14)
HMM-4	0.15 (0.06)	0.13 (0.05)	0.13 (0.03)	0.13 (0.01)
HMM-8	**0.10** (0.06)	**0.09** (0.04)	0.08 (0.02)	0.08 (0.01)
HMM-12	**0.10** (0.06)	**0.09** (0.04)	0.08 (0.02)	0.08 (0.01)

$\tau = 3$ (nonlinear state-space models)

kernel/method	$N = 80$	$N = 160$	$N = 320$	$N = 640$
sub$-\mathcal{C}$	**0.19** (0.03)	**0.14** (0.02)	**0.12** (0.02)	**0.10** (0.02)
lin$-\mathcal{C}$	0.54 (0.09)	0.60 (0.07)	0.63 (0.06)	0.67 (0.05)
RBF$-\mathcal{C}$	0.66 (0.02)	0.66 (0.01)	0.66 (0.02)	0.66 (0.01)
lin	0.63 (0.02)	0.62 (0.02)	0.62 (0.03)	0.61 (0.02)
RBF	0.66 (0.02)	0.66 (0.02)	0.66 (0.02)	0.66 (0.01)
fisher-2	0.42 (0.13)	0.51 (0.12)	0.58 (0.11)	0.54 (0.11)
fisher-4	0.46 (0.11)	0.48 (0.12)	0.59 (0.12)	0.59 (0.11)
fisher-8	0.32 (0.08)	0.51 (0.11)	0.50 (0.10)	0.51 (0.11)
HMM-4	0.21 (0.04)	0.21 (0.03)	0.20 (0.07)	0.20 (0.01)
HMM-8	0.21 (0.04)	0.20 (0.03)	0.19 (0.02)	0.20 (0.01)
HMM-12	0.21 (0.04)	0.20 (0.03)	0.19 (0.02)	0.20 (0.01)

it applies — the kernel parameter σ). To estimate models necessary for the Fisher$-d$ kernel we used the whole set of N training pairs.

We considered different values of N. For each setting we performed 30 experiments each of which was based upon an independent random drawn of $\mathcal{D}_N$. In Table 5.1 we reported the mean (with standard deviation in parenthesis)

of misclassification errors obtained by the different models on the test set.

Remark. According to Table 5.1 the cumulant kernel (sub$-\mathcal{C}$) outperforms all the alternative similarity measures (including those based on generative models). Notably HMMs work better for the case where the underlying models are linear ($\tau = 1$) and otherwise perform comparably or worse ($\tau = 2$ and $\tau = 3$).

5.5.2 Biomag 2010 Data

As a real life application we considered a brain decoding problem. The experiment was designed to investigate whether the brain activity of a subject modulating his/her attention towards a specific spatial direction (left or right), could be used as the control signal in a brain-computer interface (BCI) system [123, 223].

Experimental Setup and Preprocessing

Brain data were recorded by means of a magnetoencephalography (MEG) system and distributed by the Biomag 2010 data analysis competition[14]. Samples of 274 sensors were collected at 1200 Hz for 255 trials per subject and refer to 4 subjects[15]. Two standard pre-processing steps were operated on all time series, specifically: first downsampling from 1200 to 300 Hz and secondly band-pass filtering in order to remove frequency components outside the range $5 - 70$ Hz. The brain activity of interest related to attention tasks is known to fall within that range [223]. In this work we present results for subject 1 in the original study. Based on previous findings [123, 223], we restricted our analysis to a subset of 45 out of the 274 sensors/timeseries of the initial dataset corresponding to the parietal areas of the brain. As in [223], we considered the last 2000 ms of each trial to remove artifacts occurring immediately after the stimulus offset. The final dataset consisted of 255 trials, 127 corresponding to the *left* stimulus and 128 to the *right* stimulus. The input pattern corresponding to each trial consisted of a 45×600 matrix. Additionally, we constructed a second dataset in order to compare results against the state-of-the-art analysis found in [223]. In this case, each of the 45 channels of each trial was represented by a vector of 129 elements corresponding to values of the power spectral density (PSD) in the $5 - 70$ Hz range.

[14] The dataset was downloaded at `ftp://ftp.fcdonders.nl/pub/courses/biomag2010/competition1-dataset`.

[15] The Biomag dataset corresponds to subjects 1, 2, 3 and 7 respectively in [223].

Test Procedure

Out of the 255 recordings available 100 were selected at random for testing purposes. Within the remaining patterns N were extracted at random in repeated experiments to form the training set $\mathcal{D}_N$. We tested the same kernel functions (5.21)-(5.24) as for the synthetic experiment above as well as the proposed kernel function (5.18) (sub-$\mathcal{C}$) within the LS-SVMlab toolbox. Additionally we tested the kernel function (5.24) (PSD-lin) on the PSD dataset described above and the Fisher kernel. The Areas under the ROC curves (AUCs) obtained via the best performing kernels are reported in Table 5.2 along with those obtained by generative classifiers. We reported results for Fisher$-d$ and HMM$-d$ only for $d = 2$ since $d > 2$ led to worse performance.

Table 5.2: Real-life problem: AUCs measured on test data

kernel/method	$N = 90$	$N = 110$	$N = 130$	$N = 150$
sub$-\mathcal{C}$	**0.86**(0.01)	**0.87**(0.01)	**0.87**(0.01)	**0.88**(0.01)
PSD-lin	0.80(0.02)	0.81(0.01)	0.82(0.01)	0.82(0.01)
fisher-2	0.68(0.05)	0.70(0.05)	0.72(0.04)	0.73(0.05)
HMM-2	0.81 (0.02)	0.81 (0.02)	0.81 (0.02)	0.83 (0.01)

5.6 Chapter Summary

In this Chapter we considered a specific application for the tensorial kernel introduced in Chapter 4. We tackled the problem of classifying multivariate time series. Each input data is a matrix obtained from observing a given vector process along a finite number of successive time instants. Each of these matrices is transformed into a cumulant tensor and fed into a tensorial kernel to build a non-parametric tensor-based model. The resulting classifier is compared with alternative approaches (including generative methods) on both synthetic examples as well as on a real-life brain decoding task. Contrary to alternative techniques the proposed approach is entirely data-driven and does not require the blind identification of models. Being based on cumulants, the approach is asymptotically insensitive to gaussian noise affecting the processes. Additionally we show that, under certain assumptions on the generating systems, the cumulant-based kernel actually measure similarity between the state processes. This result admits a special interpretation towards brain-decoding tasks. In fact, if one assumes a state-space representation for EG

signals, the result shows that the cumulant-based kernel actually measures similarity between latent mental states, which might explain the superior performances.

Chapter 6

Conclusions

6.1 Concluding Remarks

Tensor-based techniques for learning allow one to exploit the structure of carefully chosen representations of data. As discussed in Chapter 2 this is a desirable feature in particular when the number of training patterns is small which is the case in many applications of interest. This thesis proposed different novel techniques to construct such models.

In Chapter 3 we have established a framework for learning with higher order tensors based on affine functions (what we called parametric models). We have used this to devise both transductive as well as inductive learning problems generalizing existing formulations to the case of multimodal data and/or multiple tasks. The transductive approach we considered is especially useful in the presence of missing input features. The inductive formulation, on the other hand, allows one to predict labels associated to input items unavailable during training. Both these approaches work by simultaneously identifying subspaces of highly predictive features without the need for a preliminary feature extraction step. This is accomplished both leveraging relationships across tasks and within the multimodal representation of the (possibly very high dimensional) input data. A crucial role is played by composite penalties based on the nuclear norm that promote solutions with small multilinear ranks. This idea mimics the concept of continuous shrinkage that underlies many convex statistical methods such as the LASSO or Ridge Regression. By contrast, the majority of the existing methodologies for tensors are non-convex and rely on hard multilinear rank constraints. Additionally, we note that traditional

techniques based on tensor decompositions are usually not developed within the solid framework of statistical learning. This means that the emphasis is usually posed on algebraic aspects and issues like model selection, the adequate choice of loss functions and the problem of out-of-sample extension are often overlooked. Phrasing the problem of learning with tensors within the context of regularization is desirable also to properly address these aspects.

In Chapter 4 we have discussed novel techniques to go beyond the class of affine models. This was achieved by exploiting the flexibility of kernel methods on the one hand and the structure of data tensors on the other. A first idea, that we fully developed only for the two-way case, corresponds to map matrices into abstract operators between reproducing kernel Hilbert spaces. This is performed while retaining structural information from the original input representation (that is, the operatorial nature of the input data). Practical usage of this type of feature representation requires to estimate Kronecker-type models and, as a consequence, results into non-convex techniques. Additionally the number of parameters that one needs to estimate is considerably high. This fact limits the practical applicability of the idea. In turn, the second approach we proposed allows one to efficiently estimate nonparametric tensor-based models. We began by showing that a naive application of existing kernels amounts to neglect the higher order structure of tensor patterns, hereby loosing the advantage of using informative representations. In contrast, the tensorial kernel we proposed keeps into account the intrinsic geometry of data tensors by leveraging the Grassmannian nature of matrix unfoldings. Additionally we have elaborated on an invariance property possessed by the proposed factor kernels and introduced the concept of congruence sets. From a pattern recognition viewpoint this is important because as soon as classes are well approximated by congruence sets, improved classification accuracy is to be expected. This is in line with statistical learning results showing that good generalization takes place if similarity measures do capture the structure of the learning tasks of interest.

The tensorial kernel is a special case of a more general class of product kernels. Product kernels, including the widely used Gaussian RBF kernel, play a special role in nonparametric statistics and machine learning. They have been used as a main tool to construct nonparametric multivariate models. In Appendix B.4, as an additional contribution, we gathered insight on the RKHSs associated to product kernels. We have shown that spaces of finite dimensional tensors can be regarded as RKHSs associated to product kernels. On the other hand, the Hilbert space of multilinear functionals associated to general product kernels can be regarded as a space of infinite dimensional tensors. This fact shows a link between tensors and kernels; it addresses the recent interest shown for the interplay between these two seemingly separate mathematical concepts. It has

far reaching consequences in terms of new lines of research. In fact, it means that results concerning finite dimensional tensors (e.g., algebraic geometric properties of tensors, see below) could probably affect kernel methods and, vice versa, insights for product kernels could translate into new insights for tensors.

Many types of structured data admit a natural representation as tensors. Therefore the tensorial kernel is largely applicable as opposed to kernels designed to deal with very specific data structures. Indeed the idea can be adapted to different context. In particular, in Chapter 5 we focused on multivariate time series. We adapt the tensorial kernel to define similarity between multichannel signals, based upon the spectral information of cumulant tensors. These tensors are constructed directly from observed measurements and do not require specific modeling assumptions, nor a foregoing identification step. This allows one to derive fully data-driven pattern recognition methods while still exploiting the dynamical nature of the data. We have discussed computational aspects and shown properties of the proposed cumulant kernels. The significance of these properties was discussed in connection with a specific application, namely EG data analysis. Experimental results compare favorably to those obtained by discriminative classifiers built upon alternative kernel functions or by generative methods. Alternative kernels include similarities that do not exploit the dynamical structure of the data. Additionally we have considered an instance of the Fisher kernel that, similarly to an approach based on HMMs, requires the estimation of linear Gaussian models.

6.2 Perspectives for Future Research

This thesis fits into a genuinely new line of research originated by the cross-fertilization between kernel methods and tensor-based techniques. As such, a number of research directions remain largely unexplored. This comprises theoretical inquiries as well as more algorithmical aspects.

Statistical Performance of Tensor-based Learning Algorithms

The non convexity of traditional tensor-based techniques based on decomposition largely prevents the analysis of their performance. In contrast, the new convex based approaches for tensors are amenable to performance guarantees [214]. In particular the analysis conducted within the aforementioned reference comes from recent results on matrix completion based on the nuclear norm. It naturally extends statistical bounds existing for the two-way case to higher

order tensors. The result was shown to precisely predict the scaling behavior of certain convex algorithms for tensors [214]. Asides from purely theoretical considerations, this is desirable since it goes in the direction of bound-based model selection criteria for parameter selection.

On the other hand this emerging theory is mainly concerned with completion; it says little about learning and generalization. In particular it differs substantially from the type of bounds that we discussed in Chapter 2. Therefore an important open problem is to characterize the new convex algorithms for tensors from a learning perspective, bridging the gap with existing learning bounds.

Algebraic geometry: Implication for Tensor-based Learning

As we discussed already, many types of structured data admit a natural representation as tensors, on the one hand; on the other, one can use tensor transformations to rely on the rich algebraic structure of tensors. In Chapter 4 we relied on such structure and elaborated on an invariance property possessed by the tensorial kernel. The argument partially addresses the question of why such kernel exhibits improved learning rates over kernels like the Gaussian RBF. The latter is universal but does not exploit structural information. The result is based on an algebraic characterization of cubic tensors [65]. Algebraic geometry for spaces of tensors as well as related topics are actively studied [129]. Understanding the consequences of these results for learning is an important challenge of future research. This might help in predicting the behavior of algorithms involving tensor data and ultimately suggesting when and how to use such representations.

Kernels and Spectral Regularization

A drawback of the methods proposed in Chapter 3 is their restriction to affine models. An interesting line of future research concerns the extension to a broader class of models. This could spring from the realization that tensors and kernels are deeply connected (Appendix B.4). In light of this, spectral regularization could probably be extended to models based on general product kernels. This might lead to new structure inducing penalties for kernel models. At present this area is largely dominated by the existing MKL framework based on an extension of the block-l_1 penalty to spaces of functions.

Nuclear Norm, Tensor Ranks and Proximity Operators

In Chapter 3 we extended spectral regularization from matrices to tensors and then focused on penalties based on the nuclear norm. An important open problem is to characterize the class of spectral norms (and the nuclear norm in particular) with respect to the two distinct notions of tensorial ranks (see Section 2.3.2). In fact a main criticism for what was called *nuclear norm for tensors* ([135] and, later, [182]) is that such a norm does not have the same fundamental properties possessed by the nuclear norm for matrices. Nonetheless, as a penalty function, it provides a viable way to compute low complexity estimates in the spirit of the Tucker decomposition. In contrast tensorial rank problems are known to be intractable [104, 107]. Still, there might be better convex relaxations to tackle tensorial rank problems. This aspect was recently discussed in [43] but no solution was actually found. Note that convex functions are strictly related to monotone operators [19]. Therefore the problem is strictly related to that of finding suitable proximity operators for tensors.

Tensor Transforms and Scalability

One drawback of using tensor transforms for representing patterns, is the high dimensionality of the resulting objects. In Chapter 5 we considered the case where time frames of vector processes were represented via $4-$th order sample cumulant tensors. In general this means that each of the N training patterns, originally consisting of a $D \times T$ matrix, is represented by a $D \times D \times D \times D$ tensor. Even storing the training set in this case can be unpractical as soon as N and/or D are even moderately big. A possible way to circumvent this difficulty is by considering a reference signal, as we proposed in Section 5.3.2. By doing so, in fact, input data become third order tensors. When $D = 45$, as in our BMI application (Section 5.5.2), the time complexity of the proposed approach is in line with that required by generative and hybrid classifiers. In turn, these classifiers underperform in terms of correct classification rates. Still, it is important to notice that employing a reference signal entails a loss of information and, as a consequence, a potential loss of classification accuracy. Therefore it is desirable to come up with alternative approaches to ensure computational feasibility without (or with little) loss of information. We mention that a possible approach, that demands further investigation, is the compression by means of random projections in the spirit of the *Johnson-Lindenstrauss lemma*, see [3, 16] and references therein.

Technology Transfer: Towards Domain-specific Methodologies

In this thesis we took a methodological perspective and derived a number of techniques that are broadly applicable. Combining these techniques with domain-specific expertise could lead to cutting edge approaches, especially for those tasks where little observational data are available at training. The cumulant-based approach considered in Chapter 5, for instance, leads to state-of-the art performance in a BMI application. Preliminary results show that spectral regularization can be conveniently used for the early detection of epileptic seizures [114]. Likewise, it is expected that similar results can be obtained in a number of problems of interest where tensorial representations could be exploited to convey domain knowledge. This includes in particular all those problems with dynamics or where data are in the frequency domain such as in MRI data analysis or in MSI.

Mathematical Concepts and Tools

Appendix A

Hilbert Spaces

We begin our mathematical journey from the fundamental concept of a Hilbert space. Our aim is to give a self-contained account of the fundamental mathematical facts that we need throughout the thesis. Our survey is therefore biased towards our needs and far from complete; in particular, we omit the proofs unless they are particularly relevant within the global context of this thesis.

Bibliographical Pointers Elements of Banach and Hilbert space theory can be found in all functional analysis textbooks; classical references include [53, 126, 164, 173]. An account suitable for engineers is given in [136, Chapter 2 and 3]. An exhaustive and self-contained theory is found in [70, Chapter 1 to 5]. Tensor products of HSs and abstract Kronecker products are found in [231, Supplement to Chapter 1] and [120]. A recent reference is [174] (within a Banach space setting).

A.1 Normed Vector Spaces

An *abstract vector space* (or vector space, for brevity) $\mathscr{X}$ over $\mathbb{R}$ is a collection of elements, called *vectors*, together with two operations, namely *addition* and *scalar multiplication*. The first associates any two vectors $x, y \in \mathscr{X}$ with a vector $x + y \in \mathscr{X}$, the sum of x and y. The second associates $x \in \mathscr{X}$ and $\alpha \in \mathbb{R}$ with the vector $\alpha x \in \mathscr{X}$. The two operations are assumed to satisfy the following axioms:

1. $x + y = y + x$ (commutative law)

2. $(x + y) + z = x + (y + z)$ (associative law)

3. there exist a *null element* $\theta \in \mathscr{X}$ such that $x + \theta = x$ for all $x \in \mathscr{X}$

4. $\left.\begin{array}{l} \alpha(x + y) = \alpha x + \alpha y \\ (\alpha + \beta)x = \alpha x + \beta x \end{array}\right\}$ (distributive laws)

5. $(\alpha\beta)x = \alpha(\beta x)$ (associative law)

6. $0x = \theta$, $1x = x$.

Subtraction is now defined by setting $x - y = x + (-1)y$. For convenience the vector $(-1)x$ is denoted by $-x$ and called the negative of x. We have $x + (-x) = (1 - 1)x = 0x = \theta$. A number of elementary but fundamental properties are derived from the axioms above.

Remark. Unless differently specified we will denote the zero element θ simply by 0 throughout the thesis. Note that, for the sake of generality, one usually introduces vector spaces over some general field $\mathbb{K}$. In this thesis, however, we are concerned with spaces that are obtained adding additional structure to some abstract vector space over the reals. Therefore throughout the thesis we will always consider the case $\mathbb{K} = \mathbb{R}$.

Example A.1.1. The simplest example is $\mathbb{R}$ with the ordinary definition of addition and multiplication. An obvious extension is $\mathbb{R}^P$ namely the space of vectors of the type $x = (x_1, x_2, \ldots, x_P)$ with addition and scalar multiplication defined component-wise.

Example A.1.2 ($\mathscr{C}[a, b]$ space)**.** Consider the interval $[a, b] \subset \mathbb{R}$. The collection of all real-valued continuous functions forms a vector space, denoted as $\mathscr{C}[a, b]$, with the operations introduced as follows. If f and g are continuous function on $[a, b]$ we define $f + g$ as the function point-wise defined by $(f + g)(x) := f(x) + g(x)$ and we further indicate by αf the function point-wise defined by $(\alpha f)(x) := \alpha f(x)$. The null vector 0 is the function identically zero on the interval $[a, b]$.

Remark. The reason why we followed many other authors in calling $\mathscr{X}$ an *abstract vector space*, and not simply a *vector space*, is understood by the variety of possible instances of what is, indeed, a quite powerful abstraction. We have already seen examples. We will further show in the following that the same concept can be used to describe collections of elements as diverse as finite dimensional arrays of any order, infinite dimensional sequences, functions, and even tuples of inhomogeneous objects.

A finite collection of vectors $\{x_1, x_2, \ldots, x_N\} \subset \mathscr{X}$ is called *linearly independent* if $\lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_N x_N = 0$ implies $\lambda_1 = \lambda_2 = \cdots = \lambda_N = 0$. An infinite collection of vectors is called linearly independent if every finite subcollection is linearly independent. A finite collection of vectors which is not linearly independent is called *linearly dependent*.

Definition A.1.1 ((Proper) subspace)**.** A nonempty subset $\mathscr{A}$ of an abstract vector space $\mathscr{X}$ is called a *subspace* of $\mathscr{X}$ if every vector of the form $\alpha x + \beta y$ is in $\mathscr{A}$ whenever x and y are in $\mathscr{A}$. A subspace $\mathscr{A}$ is additionally called *proper* if $\mathscr{A} \neq \mathscr{X}$ and $\mathscr{A} \neq \emptyset$.

Definition A.1.2 (Span)**.** The (linear) span of a set of vectors $\{x_1, x_2, \ldots, x_N\} \subset \mathscr{X}$ is the set formed by all the linear combinations of the elements of the set. In symbols:

$$\operatorname{span}\{x_1, x_2, \ldots, x_N\} = \{\lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_N x_N \ : \ \lambda_n \in \mathbb{R} \text{ for all } n \in \mathbb{N}_N\} \ .$$

Let $\mathscr{A} = \{x_1, x_2, \ldots, x_N\}$. By definition of span, $\operatorname{span}\mathscr{A}$ is a vector subspace of $\mathscr{X}$. This subspace is called the *space spanned by* $\mathscr{A}$.

A set of vectors $\mathscr{B} \subset \mathscr{X}$ is called a *basis* of $\mathscr{X}$ if $\mathscr{B}$ is linearly independent and $\operatorname{span}\mathscr{B} = \mathscr{X}$. If $\mathscr{X}$ admits a finite basis then $\mathscr{X}$ is called a *finite dimensional vector space*. Otherwise we say that $\mathscr{X}$ is *infinite dimensional*. It can be proved that if $\mathscr{B}$ consists of N elements, then all the bases of $\mathscr{X}$ consist of N elements. In this case N is called the *dimension* of $\mathscr{X}$ and we write $\dim \mathscr{X} = N$.

Definition A.1.3 (Cartesian Product)**.** Let $\mathscr{X}_1$ and $\mathscr{X}_2$ be abstract vector spaces. The *Cartesian product* of $\mathscr{X}_1$ and $\mathscr{X}_2$, denoted by $\mathscr{X}_1 \times \mathscr{X}_2$, is the collection of ordered pairs (x_1, x_2) with $x_1 \in \mathscr{X}_1$ and $x_2 \in \mathscr{X}_2$. It can be turned into a vector space by defining addition as $(x_1, x_2) + (x_1', x_2') = (x_1 + x_1', x_2 + x_2')$ and scalar multiplication as $\alpha(x_1, x_2) = (\alpha x_1, \alpha x_2)$.

The definition is easily generalized to the product of N spaces $\mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_N$. In particular, we write $\times_{n \in \mathbb{N}_N} \mathscr{X}_n$ to mean $\mathscr{X}_1 \times \mathscr{X}_2 \times \ldots \times \mathscr{X}_N$. We write $\mathscr{X}^N$ for the product of $\mathscr{X}$ with itself for N times.

Remark. Note that abstract vectors of $\times_{n \in \mathbb{N}_N} \mathscr{X}_n$ might be $N-$tuples of inhomogeneous objects.

Definition A.1.4 (Norm)**.** Let x and y be elements of an abstract vector space $\mathscr{X}$. A *norm* on $\mathscr{X}$, denoted by $\|\cdot\|$, is a real-valued function on $\mathscr{X}$ for which the following conditions hold:

1. $\|x + y\| \leq \|x\| + \|y\|$

2. $\|\alpha x\| = |\alpha|\|x\|$
3. $\|x\| \geq 0$ and $\|x\| = 0$ if and only if $x = 0$.

Definition A.1.5 (Normed space)**.** We call normed space the pair $(\mathscr{X}, \|\cdot\|)$.

Example A.1.3 ($\mathscr{C}[a,b]$ space)**.** The abstract vector space $\mathscr{C}[a,b]$ together with the norm $\|f\| := \max_{t\in[a,b]} |f(t)|$ forms a normed space. It is not difficult to check that $\|\cdot\|$ indeed satisfies the properties listed in the Definition A.1.5.

Example A.1.4. Assume normed spaces $(\mathscr{X}, \|\cdot\|_{\mathscr{X}})$ and $(\mathscr{Y}, \|\cdot\|_{\mathscr{Y}})$. Then $\mathscr{X} \times \mathscr{Y}$ can be normed in different ways. One is to let

$$\|(x,y)\| := \left(\|x\|^2_{\mathscr{X}} + \|y\|^2_{\mathscr{Y}}\right)^{\frac{1}{2}} .$$

In the same fashion one can norm $\mathscr{X}^N$ by letting

$$\|(x_1, x_2, \cdots, x_N)\| := \left(\|x_1\|^2_{\mathscr{X}} + \|x_2\|^2_{\mathscr{X}} + \cdots + \|x_N\|^2_{\mathscr{X}}\right)^{\frac{1}{2}} .$$

This approach generalizes to abstract vector spaces the way in which $\mathbb{R}^N$ is normed.

In order to properly introduce the next examples we need the following result the proof of which can be found in Analysis textbooks.

Proposition A.1.1 (Hölder's inequality)**.** *Assume $p > 1, q < \infty$ satisfying $\frac{1}{p} + \frac{1}{q} = 1$.*

If $x = (x_n \in \mathbb{R} : n \in \mathbb{N})$ and $y = (y_n \in \mathbb{R} : n \in \mathbb{N})$ are two arbitrary sequences, then

$$\sum_{n\in\mathbb{N}} |x_n y_n| \leq \left(\sum_{n\in\mathbb{N}} |x_n|^p\right)^{\frac{1}{p}} \left(\sum_{n\in\mathbb{N}} |y_n|^q\right)^{\frac{1}{q}} .$$

If x and y are measurable real-valued functions on $[a,b]$, then

$$\int_a^b |f(t)g(t)|dt \leq \left(\int_a^b |f(t)|^p dt\right)^{\frac{1}{p}} \left(\int_a^b |g(t)|^q dt\right)^{\frac{1}{q}} .$$

Example A.1.5 ($l_2(\mathbb{N})$ space)**.** Define:

$$\begin{cases} \|x\| := \left(\sum_{n\in\mathbb{N}} x_n^2\right)^{\frac{1}{2}} & \text{(A.1a)} \\ \\ l_2(\mathbb{N}) := \{x = (x_n \in \mathbb{R} \ : \ n \in \mathbb{N}) \text{ such that } \|x\| < \infty\} . & \text{(A.1b)} \end{cases}$$

The pair $(l_2(\mathbb{N}), \|\cdot\|)$ is a normed space with operations defined component-wise as $(x+y)_n = x_n + y_n$ and $(\alpha x)_n = \alpha x_n$. In particular, it can be checked that $\|\cdot\|$ in (A.1a) is a well-defined norm. Let $z = x + y$ where $x, y \in l_2(\mathbb{N})$. To begin with, note that $\|z\| = 0$ if and only if $y = -x$, that is, if $y_n = -x_n$ for any $n \in \mathbb{N}$. Now we have

$$\|z\|^2 = \sum_{n\in\mathbb{N}} (x_n+y_n)^2 = \sum_{n\in\mathbb{N}} x_n z_n + \sum_{n\in\mathbb{N}} y_n z_n \overset{\text{(Hölder's inequality)}}{\leq} (\|x\|+\|y\|)\|z\| \tag{A.2}$$

from which the triangular inequality follows. Note that, since $\|x+y\| \leq \|x\| + \|y\|$, by (A.1b) $x + y$ indeed belongs to $l_2(\mathbb{N})$.

Remark. The notion of $l_2(\mathbb{N})$ generalizes that of finite dimensional Euclidean spaces. Indeed, $\mathbb{R}^P$ is obtained by replacing $\mathbb{N}$ with $\mathbb{N}_P$ in (A.1). Note that the summability condition that appears in (A.1b) becomes superfluous in this latter case. In the following we will sometimes treat the two cases jointly by referring to $l_2(\mathscr{I})$ where $\mathscr{I} \subseteq \mathbb{N}$ is either a finite discrete index set or the whole set of natural numbers.

Example A.1.6 ($L_2[a,b]$ space)**.** Define:

$$\begin{cases} \|f\| := \left(\int_a^b f^2(t)dt \right)^{\frac{1}{2}} & \text{(A.3a)} \\ \\ L_2[a,b] := \{ f \text{ is a measurable function on } [a,b] \text{ and } \|f\| < \infty \}\,. & \text{(A.3b)} \end{cases}$$

Similarly as for $l_2(\mathbb{N})$, the pair $(L_2[a,b], \|\cdot\|)$ can be shown to form a normed space. Note, however, that here $\|f\| = 0$ does not imply that f is the null vector. In fact, f might still be non-zero on a subset of $[a,b]$ of measure zero (a set consisting of a countable number of points). Strictly speaking $L_2[a,b]$ does not consist of functions, but rather of equivalence classes of functions[1].

Open and Closed Sets

Before proceeding further we recall basic topological definitions.

Definition A.1.6 (Interior point)**.** Assume a normed space $(\mathscr{X}, \|\cdot\|)$ and a subset $\mathscr{P} \subset \mathscr{X}$. The vector $p \in \mathscr{P}$ is said to be an *interior point* if there is an

[1] Equivalence classes of functions are formed by functions that are equivalent almost everywhere (a.e.), that is, they are allowed to differ on a set of measure zero.

$\epsilon > 0$ such that all vectors $x \in \mathscr{X}$ satisfying $\|x - p\| < \epsilon$ are also members of $\mathscr{P}$. The collection of all interior points is called the *interior* and denoted by $\overset{\circ}{\mathscr{P}}$.

Definition A.1.7 (Open set)**.** A subset $\mathscr{P} \subset \mathscr{X}$ is said to be *open* if $\mathscr{P} = \overset{\circ}{\mathscr{P}}$.

Definition A.1.8 (Closure point)**.** A point $x \in \mathscr{X}$ is said to be a *closure point of a subset* $\mathscr{P} \subset \mathscr{X}$ if, given $\epsilon > 0$, there is a point $p \in \mathscr{P}$ such that $\|x - p\| < \epsilon$. The collection of all closure points of a subset is called the *closure* and denoted by $\bar{\mathscr{P}}$.

Definition A.1.9 (Closed set)**.** A subset $\mathscr{P} \subset \mathscr{X}$ is said to be *closed* if $\mathscr{P} = \bar{\mathscr{P}}$.

The empty set as well as the whole space $\mathscr{X}$ are both closed and open. The set $\{x \in \mathscr{X} \ : \ \|x\| < \alpha\}$ is open, the set $\{x \in \mathscr{X} \ : \ \|x\| \leq \alpha\}$ is closed.

Convergence

Definition A.1.10 (Convergent sequence)**.** A sequence $\{x_n\}_{n\in\mathbb{N}}$ in a normed space $(\mathscr{X}, \|\cdot\|)$ is said to be *convergent* to a vector x if the sequence of real numbers $\{\|x - x_n\|\}_{n\in\mathbb{N}}$ converges to zero, in symbols:

$$\|x - x_n\| \to 0 \ . \tag{A.4}$$

In this case we write

$$x_n \to x \ .$$

Note that (A.4) is the same as saying that given $\epsilon > 0$ there exists an N such that $\|x - x_n\| < \epsilon$ for any $n > N$.

Definition A.1.11 (Cauchy sequence)**.** A sequence $\{x_n\}_{n\in\mathbb{N}}$ in a normed space $(\mathscr{X}, \|\cdot\|)$ is said to be a *Cauchy sequence* if $\|x_n - x_m\| \to 0$ as $m, n \to \infty$, i.e., for an arbitrary $\epsilon > 0$ there exists an $N > 0$ such that if $m, n > N$ then $\|x_n - x_m\| < \epsilon$.

In $(\mathscr{X}, \|\cdot\|)$ every convergent sequence is a Cauchy sequence. Indeed, if $x_n \to x$, then we have:

$$\|x_n - x_m\| = \|x_n - x + x - x_m\| \leq \|x_n - x\| + \|x_m - x\| \to 0.$$

In general, however, a Cauchy sequence might not be convergent.

Definition A.1.12 (Complete space)**.** A normed space $(\mathscr{X}, \|\cdot\|)$ is said to be *complete* if every Cauchy sequence in $\mathscr{X}$ admits a limit in $\mathscr{X}$.

Definition A.1.13 (Banach space)**.** A complete normed space is called a *Banach* space.

Remark. It can be shown that every finite-dimensional normed space (over $\mathbb{R}$ and $\mathbb{C}$) is complete and therefore qualifies as a Banach space. The situation is more involved with infinite dimensional spaces.

Example A.1.7 ($\mathscr{C}[a,b]$ space)**.** Suppose $\{x_n\}_{n\in\mathbb{N}}$ is a Cauchy sequence in $\mathscr{C}[a,b]$. Then for every $t \in [a,b]$ we have

$$|x_n(t) - x_m(t)| \leq \max_{v\in[a,b]} |x_n(v) - x_m(v)| = \|x_n - x_m\| \to 0$$

so $\{x_n(t)\}_{n\in\mathbb{N}}$ is a Cauchy sequence of real numbers. Since $\mathbb{R}$ is complete there is a number in $\mathbb{R}$ to which the sequence converges: $x_n(t) \to x(t)$. Therefore the Cauchy sequence $\{x_n\}_{n\in\mathbb{N}}$ converges pointwise to the function x. Based upon this fact one can further show that x is continuous and that $\{x_n\}_{n\in\mathbb{N}}$ converges to x in the norm of $\mathscr{C}[a,b]$.

Theorem A.1.2. *Assume a normed space $(\mathscr{X}, \|\cdot\|)$ and let $\mathscr{S}$ be a finite dimensional subspace of $\mathscr{X}$. Then $\mathscr{S}$ is closed.*

Corollary A.1.3. *Assume a normed space $(\mathscr{X}, \|\cdot\|)$. Then $\{0\} \subset \mathscr{X}$ is a closed subspace of $\mathscr{X}$.*

A.2 Linear Mappings

Let $\mathscr{X}$ and $\mathscr{Y}$ be vector spaces and let L be a mapping from $\mathscr{X}$ to $\mathscr{Y}$, $L : \mathscr{X} \to \mathscr{Y}$. If $y = L(x)$, y is called the *image* of x. The image of a subset $\mathscr{A} \subset \mathscr{X}$ and the *inverse image* of a subset $\mathscr{B} \subset \mathscr{Y}$ are defined, respectively, as:

$$L(\mathscr{A}) := \{L(x) \ : \ x \in \mathscr{A}\} \quad \text{and} \quad L^{-1}(\mathscr{B}) := \{x \in \mathscr{X} \ : \ L(x) \in \mathscr{B}\} \ .$$

We sometimes consider mappings that are defined on proper subsets of vector spaces. We denote by $\mathrm{dom}(L) \subset \mathscr{X}$ the *domain* of L.

The *range* of L is defined as:

$$\mathscr{R}(L) := \{y \in \mathscr{Y} \ : \ L(x) = y \text{ for some } x \in \mathrm{dom}(L)\} \ .$$

The *null space* is:

$$\mathscr{N}(L) := \{x \in \mathrm{dom}(L) \ : \ L(x) = 0\} \ .$$

Basic Facts

Definition A.2.1 (Linear mapping)**.** A map $L : \mathscr{X} \to \mathscr{Y}$ is *linear* if $L(\alpha x + \beta y) = \alpha L(x) + \beta L(y)$ for all $x, y \in \mathscr{X}$ and scalars α, β.

Remark. It is not difficult to see that, if L is linear and $\mathrm{dom}(L)$ is a vector space, then both $\mathscr{R}(L)$ and $\mathscr{N}(L)$ are vector spaces.

Definition A.2.2 (Isometry)**.** Assume normed spaces $(\mathscr{X}, \|\cdot\|_{\mathscr{X}})$ and $(\mathscr{Y}, \|\cdot\|_{\mathscr{Y}})$. A linear operator $Q : \mathscr{X} \to \mathscr{Y}$ is called *isometry* if:

$$\|Qx\|_{\mathscr{Y}} = \|x\|_{\mathscr{X}} \ .$$

Definition A.2.3 (Bilinear mapping)**.** A map $M : \mathscr{X} \times \mathscr{Y} \to \mathscr{Z}$ is *bilinear* if the mapping $L_x : \mathscr{Y} \to \mathscr{Z}$ and the mapping $L_y : \mathscr{X} \to \mathscr{Z}$ defined, respectively, as:

$$L_x : y \mapsto M(x, y) \quad \text{and} \quad L_y : x \mapsto M(x, y)$$

are both linear.

Remark. If $\mathscr{X}$ and $\mathscr{Y}$ are general vector spaces a mapping L from $\mathscr{X}$ to $\mathscr{Y}$ is often called an *operator*. If, on the other hand, $\mathscr{Y} = \mathbb{R}$ then L is called a *functional*. For a linear operator L we usually follow the convention of writing Lx instead of $L(x)$.

The notion of continuous mapping generalizes that of continuous function and it is based on the concept of convergent sequences (Definition (A.1.10)).

Definition A.2.4 (Continuous mapping between normed spaces)**.** Assume normed spaces $(\mathscr{X}, \|\cdot\|_{\mathscr{X}})$ and $(\mathscr{Y}, \|\cdot\|_{\mathscr{Y}})$. A mapping $L : \mathscr{X} \to \mathscr{Y}$ is *continuous at* $x \in \mathscr{X}$ if, for any sequence $\{x_n\}_{n\in\mathbb{N}}$ such that $x_n \to x$, we have that the sequence $\{L(x_n)\}_{n\in\mathbb{N}}$ satisfies $L(x_n) \to L(x)$. If L is continuous at any $x \in \mathscr{X}$ then we say that L is *continuous*.

Remark. Equivalently: L is continuous if at any point $x_0 \in \mathscr{X}$ and any real number $\epsilon > 0$, there exists a real number $\delta > 0$ such that any x satisfying $\|x_0 - x\|_{\mathscr{X}} < \delta$ will also satisfy $\|Lx_0 - Lx\|_{\mathscr{Y}} < \epsilon$.

Example A.2.1 (A Non-continuous Linear Mapping)**.** Consider the vector space of polynomials over $[0, 1]$ endowed with the norm:

$$\|p\| = \left(\int_0^1 |p(x)|^2 dx \right)^{1/2} \ .$$

Observe that the sequence of functions $\{q_n\}_{n\in\mathbb{N}}$ where $q_n(x) = x^n$ converges to the null function because:

$$\|q_n - 0\| = \left(\int_0^1 x^{2n} dx \right)^{1/2} = (2n+1)^{-1/2} \ .$$

However the sequence $\{q_n(1)\}_{n\in\mathbb{N}}$ is constant with $q_n(1) = 1$ for all $n \in \mathbb{N}$. Therefore the evaluation functional $L_1 : p \mapsto p(1)$ is not-continuous in this space.

Example A.2.2 (Continuity of the norm)**.** Assume a normed space $(\mathscr{X}, \|\cdot\|)$. Then $\|\cdot\|$ is a continuous functional. Indeed if $\|x_n - x\| \to 0$ then $|\|x_n\| - \|x\|| \to 0$, that is, if $x_n \to x$, then $\|x_n\| \to \|x\|$.

Proposition A.2.1. *Assume normed spaces $(\mathscr{X}, \|\cdot\|_{\mathscr{X}})$ and $(\mathscr{Y}, \|\cdot\|_{\mathscr{Y}})$. If $L : \mathscr{X} \to \mathscr{Y}$ is continuous and $\mathscr{A} \subset \mathscr{Y}$ is open (closed), then $L^{-1}(\mathscr{A})$ is open (closed).*

For functionals, there is also a weaker notion of continuity.

Definition A.2.5 (Semi-continuous functional)**.** Assume a normed space $(\mathscr{X}, \|\cdot\|)$. A functional $L : \mathscr{X} \to \mathbb{R}$ is said to be *lower* (*upper*) *semi-continuous* if the set

$$\{x \in \mathscr{X} \ : \ L(x) \leq \alpha\} \qquad (\{x \in \mathscr{X} \ : \ L(x) \geq \alpha\})$$

is closed for all α.

Note that a continuous functional is both lower and upper semi-continuous.

Definition A.2.6 (Bounded linear mapping)**.** Assume normed spaces $(\mathscr{X}, \|\cdot\|_{\mathscr{X}})$ and $(\mathscr{Y}, \|\cdot\|_{\mathscr{Y}})$. A linear mapping $L : \mathscr{X} \to \mathscr{Y}$ is called *bounded* if there exists a scalar $\alpha > 0$ such that $\|Lx\|_{\mathscr{Y}} \leq \alpha\|x\|_{\mathscr{X}}$ for all $x \in \mathscr{X}$.

Note that the last condition is equivalent to require that $\|Lx\|_{\mathscr{Y}} \leq \alpha$ for all $x \in \mathscr{X}$ such that $\|x\|_{\mathscr{X}} = 1$. It is not difficult to see that linear mappings on finite dimensional spaces are always bounded. In fact, if the linear mapping is defined on a finite dimensional space then $\|Lx\|_{\mathscr{Y}}$ is given by the sum of finitely many numbers. It is not difficult to check that if a linear mapping $L : \mathscr{X} \to \mathscr{Y}$ is continuous at $x_0 \in \mathscr{X}$ then it is continuous. We can further state the following.

Definition A.2.7 (Finite Rank Operator)**.** Assume normed spaces $(\mathscr{X}, \|\cdot\|_{\mathscr{X}})$ and $(\mathscr{Y}, \|\cdot\|_{\mathscr{Y}})$. An operator $L : \mathscr{X} \to \mathscr{Y}$ is called a *finite rank* operator if its range $\mathscr{R}(L)$ is finite dimensional; in particular, L is called *rank−K* if its range has dimension K.

A more precise characterization will be given for the case where $\mathscr{X}$ and $\mathscr{Y}$ are Hilbert spaces.

Theorem A.2.2 (Continuous)**.** *Assume normed spaces* $(\mathscr{X}, \|\cdot\|_{\mathscr{X}})$ *and* $(\mathscr{Y}, \|\cdot\|_{\mathscr{Y}})$ *are given together with a linear mapping* $L : \mathscr{X} \to \mathscr{Y}$*. The following statements are equivalent.*

- L *is continuous at* $x_0 \in \mathscr{X}$
- L *is continuous*
- L *is bounded.*

Theorem A.2.3. *Assume normed spaces* $(\mathscr{X}, \|\cdot\|_{\mathscr{X}})$ *and* $(\mathscr{Y}, \|\cdot\|_{\mathscr{Y}})$ *and a linear mapping* $L : \mathscr{X} \to \mathscr{Y}$*. Then if* L *is continuous, the nullspace* $\mathscr{N}(L)$ *is a closed vector subspace of* $\mathscr{X}$*.*

Proof. The set $\{0\}$ is a closed subspace (Corollary A.1.3). The statement now follows from the definition of $\mathscr{N}(L)$ and by Proposition A.2.1. □

Note, in particular, that the result applies to functionals ($\mathscr{Y} = \mathbb{R}$).

Dual Spaces

Definition A.2.8 (Algebraic dual)**.** We call *algebraic dual* the abstract vector space of linear functionals on $\mathscr{X}$ with the operation defined as follows

$$(L_1 + L_2)x = L_1x + L_2x \text{ and } (\lambda L)x = \lambda(Lx) \ . \tag{A.5}$$

The last definition is independent from the topological structure induced by a norm. More practical is the case of bounded functionals and operators on a normed space. If $(\mathscr{X}, \|\cdot\|_{\mathscr{X}})$ and $(\mathscr{Y}, \|\cdot\|_{\mathscr{Y}})$ are normed spaces, then the set of all bounded linear mappings from $\mathscr{X}$ to $\mathscr{Y}$ forms an abstract vector space with the operation defined in the same fashion as in (A.5). This space is denoted throughout this thesis as $\mathscr{B}(\mathscr{X}, \mathscr{Y})$. We additionally use the shorthand notation $\mathscr{B}(\mathscr{X}) = \mathscr{B}(\mathscr{X}, \mathscr{X})$.

Theorem A.2.4. *Let* $(\mathscr{X}, \|\cdot\|_{\mathscr{X}})$ *and* $(\mathscr{Y}, \|\cdot\|_{\mathscr{Y}})$ *be normed spaces and assume* $\mathscr{Y}$ *is complete (therefore a Banach space). Then the abstract vector space* $\mathscr{B}(\mathscr{X}, \mathscr{Y})$ *equipped with the norm*

$$\|L\| := \sup_{\|x\|_{\mathscr{X}}=1} \|Lx\|_{\mathscr{Y}} \tag{A.6}$$

is a Banach space.

Remark. Since $L \in \mathscr{B}(\mathscr{X}, \mathscr{Y})$ there exists $\alpha > 0$ such that $\|L(x)\| \leq \alpha \|x\|_{\mathscr{X}}$ for any $x \in \mathscr{X}$. If we take $x \neq 0$, the latter is equivalent to $\|L(x/\|x\|_{\mathscr{X}})\| \leq \alpha$ where $x/\|x\|_{\mathscr{X}}$ is a unit vector. By the definition of $\|L\|$ in (A.6) it now follows that

$$\|L(x)\| \leq \|L\| \|x\|_{\mathscr{X}} \ . \tag{A.7}$$

Definition A.2.9 (Dual space)**.** Assume a normed space $(\mathscr{X}, \|\cdot\|_{\mathscr{X}})$. We call *dual space*, denoted by $\mathscr{X}^*$, the Banach space of bounded linear functionals on $\mathscr{X}$.

By definition, the dual space is a vector subspace of the algebraic dual.

Example A.2.3 (Dual space of $\mathbb{R}^N$)**.** In $\mathbb{R}^N$ each vector is an $N-$tuple of real numbers $x = (x_1, x_2, \ldots, x_N)$ with norm $\|x\| = \sqrt{\sum_{n \in \mathbb{N}} x_n^2}$. For any $\xi \in \mathbb{R}^N$, the function

$$f(x) = \sum_{n \in \mathbb{N}_N} \xi_n x_n \tag{A.8}$$

is linear. The Cauchy-Schwarz inequality

$$|f(x)| \leq \|\xi\| \|x\| \tag{A.9}$$

shows that f is bounded. Moreover, using the latter we have that

$$\|f\| = \sup_{\|x\|=1} |f(x)| \leq \|\xi\|$$

and therefore $\|f\| \leq \|\xi\|$. Since for $x = \xi$ the equality is attained in (A.9), we have that $\|f\| = \|\xi\|$. On the other hand, suppose f is a bounded linear functional. Denote by e_n the $n-$th vector of the canonical basis, namely the vector defined entry-wise as:

$$(e_n)_m = \begin{cases} 1, & \text{if } n = m \\ 0, & \text{otherwise} \end{cases}$$

and let $\eta_n = f(e_n)$. Note that, since f is bounded, $|\eta_n| < \infty$. Now we have:

$$f(x) = f\Big(\sum_{n \in \mathbb{N}_N} x_n e_n\Big) = \sum_{n \in \mathbb{N}_N} x_n f(e_n) = \sum_{n \in \mathbb{N}_N} \eta_n x_n$$

which shows that *any* bounded linear functional has the same form as (A.8).

A.3 Inner Products and Hilbert Spaces

A bilinear map $s : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is *symmetric* if $s(x, y) = s(y, x)$; it is said to be *positive* if $s(x, x) \geq 0$ for all $x \in \mathscr{X}$ and *positive-definite* if $s(x, x) > 0$ for all $x \in \mathscr{X}$ with $x \neq 0$.

Definition A.3.1 (Positive-definite matrix)**.** A $N \times N$ *real matrix* $\boldsymbol{A}$ is said to be *positive-definite* if the induced bilinear map

$$s(x, y) := \sum_{n \in \mathbb{N}_N} \sum_{m \in \mathbb{N}_N} a_{nm} x_n y_m \tag{A.10}$$

is positive-definite.

In this thesis we adopt the standard notation for matrix-vector multiplication and write $s(x, y) := x^\top A y$ instead of (A.10). Note that s is symmetric if $\boldsymbol{A}$ is a symmetric matrix, namely, if $a_{nm} = a_{mn}$ for any $n, m \in \mathbb{N}_N$.

Definition A.3.2 (Inner product space)**.** An *inner product space* is an abstract vector space $\mathscr{X}$ together with a symmetric and positive-definite bilinear form $s : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$. For any $x, y \in \mathscr{X}$, $s(x, y)$ is called the *inner product* between x and y and denoted by $\langle x, y \rangle$.

Example A.3.1. For $\mathbb{R}$ the multiplication between scalars is a positive-definite symmetric bilinear form; it constitutes the simplest example of inner product: $\langle x, y \rangle = xy$. On $\mathbb{R}^N$, a symmetric and positive definite matrix A defines the inner product $\langle x, y \rangle := x^\top A y$; in particular, the *canonical* inner product is $\langle x, y \rangle := x^\top y$.

It is not difficult to check that

$$\|x\| = \sqrt{\langle x, x \rangle}$$

satisfies the conditions in Definition A.1.5 and is therefore a norm. Additionally we have the following useful facts.

Proposition A.3.1. *Let* $\|x\| = \sqrt{\langle x, x \rangle}$*. Then for all* $x, y \in \mathscr{X}$ *we have:*

1. $|\langle x, y \rangle| \leq \|x\| \|y\|$ *(Cauchy-Schwarz inequality)*
2. $\langle x, y \rangle = \frac{1}{4}(\|x + y\|^2 - \|x - y\|^2)$ *(polarization identity)*
3. $\|x + y\|^2 + \|x - y\|^2 = 2(\|x\|^2 + \|y\|^2)$ *(parallelogram law)*

Definition A.3.3 (Hilbert space)**.** We call $\mathscr{X}$ a Hilbert space (HS for brevity) if $\mathscr{X}$ is an abstract vector space equipped with an inner product $\langle \cdot, \cdot \rangle$ in such a way that $\mathscr{X}$ is complete with respect to the norm $\|x\| = \sqrt{\langle x, x \rangle}$.

Remark. In the following we sometimes denote an HS $\mathscr{X}$ with inner product $\langle \cdot, \cdot \rangle$ by $(\mathscr{X}, \langle \cdot, \cdot \rangle)$.

Remark. Conditions for the existence of a completion of an incomplete space of functions is given in [10, page 347] in connection to the theory of *reproducing kernels* (see Appendix B).

Example A.3.2. Consider the abstract vector space of continuous functions $\mathscr{C}[a,b]$ discussed in Example A.1.2 and introduce the inner product:

$$\langle f,g\rangle = \int_a^b f(t)g(t)dt \ . \tag{A.11}$$

It can be shown that the normed space $(\mathscr{C}[a,b], \|\cdot\|)$ with the norm induced by (A.11) is *not* complete. Therefore $\mathscr{C}[a,b]$ endowed with the inner product (A.11) is not an HS.

Example A.3.3 (l_2 space)**.** Consider the space $l_2(\mathbb{N})$ of convergent sequences introduced in (A.1b). It can be checked that

$$\langle x,y\rangle = \sum_{n\in\mathbb{N}_N} x_n y_n$$

is a well defined inner product. Hölder's inequality (Proposition A.1.1) ensures that $\langle x,y\rangle \leq \|x\|\|y\|$; therefore $\langle x,y\rangle$ is finite for any $x,y \in l_2(\mathbb{N})$. The norm $\|\cdot\|$ induced by this inner product corresponds to (A.1a). It can be shown that $(l_2(\mathbb{N}), \|\cdot\|)$ is complete; therefore $l_2(\mathbb{N})$ is an HS.

Example A.3.4 ($L_2[a,b]$ space)**.** Likewise, for $L_2[a,b]$ introduced in (A.3b) it can be checked that

$$\langle f,g\rangle = \int_a^b f(x)g(x)dx$$

is a well defined inner product. The space $L_2[a,b]$ is an HS since, once endowed with the induced norm $\|f\| = \sqrt{\langle f,f\rangle}$, $(L_2[a,b], \|\cdot\|)$ is complete.

Example A.3.5 ($L_2(\mathscr{X},\mu)$ space)**.** A generalization of the former case goes as follows. Let μ be any *positive measure* on $\mathscr{X}$ [172]. Then the space

$$L_2(\mathscr{X},\mu) := \Big\{ f \ : \ f \text{ is a real measurable function on } \mathscr{X} \text{ and } \int_{\mathscr{X}} |f(x)|^2 d\mu(x) < \infty \Big\} \tag{A.12}$$

is a HS with inner product

$$\langle f,g\rangle := \int_{\mathscr{X}} f(x)g(x)d\mu(x) \ ,$$

see [172].

Orthogonality

Central to the concept of HS is the notion of *orthogonality*.

Definition A.3.4 (Orthogonality)**.** Let $\mathscr{X}$ be an HS and $\mathscr{A}$ a subset of $\mathscr{X}$. The *orthogonal complement* of $\mathscr{A}$ is defined as

$$\mathscr{A}^{\perp} := \{x \in \mathscr{X} \ : \ \langle x, y\rangle = 0 \text{ for all } y \in \mathscr{A}\} \ .$$

We say that two vectors $x, y \in \mathscr{X}$ are orthogonal if $\langle x, y\rangle = 0$. We denote this by $x \perp y$.

Theorem A.3.2. *For any subset $\mathscr{A}$ of $\mathscr{X}$ the set $\mathscr{A}^{\perp}$ is a closed subspace of $\mathscr{X}$.*

Two orthogonal vectors $x, y \in \mathscr{X}$ satisfy the *Pythagoras' Theorem*: $\|x+y\|^2 = \|x\|^2 + \|y\|^2$.

Theorem A.3.3. *Let $\mathscr{X}$ be an HS and $\mathscr{A}$ a closed vector subspace of $\mathscr{X}$.*

- *Every vector $x \in \mathscr{X}$ has a unique decomposition of the form $x = y + z$ with $y \in \mathscr{A}$ and $z \in \mathscr{A}^{\perp}$.*
- $(\mathscr{A}^{\perp})^{\perp} = \mathscr{A}$

Definition A.3.5. With reference to Theorem A.3.3, we call y the *orthogonal projection of x on $\mathscr{A}$* and we write $y = P_{\mathscr{A}}(x)$. The operator $P_{\mathscr{A}} : x \mapsto y$ is called the *orthogonal projection on $\mathscr{A}$*.

The operator $P_{\mathscr{A}}$ is clearly linear. For any $x \in \mathscr{X}$ we have:

$$x = (x - P_{\mathscr{A}}x) + P_{\mathscr{A}}x = (I - P_{\mathscr{A}})x + P_{\mathscr{A}}x$$

where we denoted by I the *identity operator*, $I : x \mapsto x$. The operator $(I - P_{\mathscr{A}})$ is the orthogonal projection on $\mathscr{A}^{\perp}$, denoted by $P_{\mathscr{A}^{\perp}}$.

Definition A.3.6 (Orthonormal family)**.** Denote by $\mathscr{I} \subseteq \mathbb{N}$ an index set. A family of vectors $\{e_i \ : \ i \in \mathscr{I}\}$ in an HS $\mathscr{X}$ is called an *orthonormal family* if

1. $\|e_i\| = 1$ for all $i \in \mathscr{I}$,
2. $\langle e_i, e_j\rangle = 0$ if $i \neq j$.

If $\{e_i \ : \ i \in \mathscr{I}\}$ is an orthonormal family in $\mathscr{X}$ and $\mathscr{A} = \operatorname{span}\{e_i \ : \ i \in \mathscr{I}\}$, then for any $x \in \mathscr{X}$ the orthogonal projection on $\mathscr{A}$ is

$$P_{\mathscr{A}}x = \sum_{i \in \mathscr{I}} \langle x, e_i\rangle e_i \text{ and } \|P_{\mathscr{A}}x\|^2 = \sum_{i \in \mathscr{I}} |\langle x, e_i\rangle|^2 \ .$$

Definition A.3.7 (Orthonormal basis)**.** Denote by $\mathscr{I} \subseteq \mathbb{N}$ an index set. The family $\{e_i \ : \ i \in \mathscr{I}\}$ is an orthonormal basis if

1. $\{e_i \ : \ i \in \mathscr{I}\}$ is an orthonormal family,
2. if $x \in \mathscr{X}$ and $\langle x, e_i \rangle = 0$ for all $i \in \mathscr{I}$, then $x = 0$.

We call *dimension* of $\mathscr{X}$ the cardinality of $\mathscr{I}$.

Definition A.3.8 (Separable Hilbert Space)**.** An HS is called *separable* if it contains an orthonormal basis.

Remark. It is possible to show that every orthogonal set in a separable HS is countable. This fact is often taken as a definition of separability. In what follows we will always consider separable HSs which are by far the most common. Non separable HSs sometimes arise in Quantum Physics.

Proposition A.3.4. *Let $\mathscr{X}$ be an HS with orthonormal basis $\{e_i \ : \ i \in \mathscr{I}\}$. For all $x \in \mathscr{X}$ we have:*

1. $x = \sum_{\mathrm{n} \in \mathbb{N}} \langle x, e_i \rangle e_i$ *(Plancherel formula)*
2. $\|x\|^2 = \sum_{i \in \mathscr{I}} |\langle x, e_i \rangle|^2$ *(Parseval equality)*

The following is an immediate consequence of the latter (see also Definition A.2.2).

Proposition A.3.5. *Let $\mathscr{X}$ be an HS with orthonormal basis $\{e_i \ : \ i \in \mathscr{I}\}$. The operator:*

$$\begin{array}{rccc} Q: & \mathscr{X} & \to & l_2(\mathscr{I}) \\ & x & \mapsto & \{\langle x, e_i \rangle\}_{i \in \mathscr{I}} \end{array}$$

is a linear isometry.

Riesz Representation Theorem

We noted already that a functional $s(x) = \langle x, y \rangle$ defined upon the inner product is bounded since $s(x) \leq \alpha \|x\|$ where $\alpha = \|y\|$ as ensured by the Hölder's inequality. In this Section we recall a fundamental result stating that, in an HS, *all* bounded linear functionals (or continuous functionals, see Theorem A.2.2) can be formulated in terms of inner products. A proof of this fact is based on the following Lemma.

Lemma A.3.6. *If s is a nontrivial bounded linear functional on an HS $\mathscr{X}$, then* $\dim \mathscr{N}(s)^{\perp} = 1$.

Proof. Since s is continuous by Theorem A.2.3 $\mathscr{N}(s)$ is a closed vector subspace; since s is nontrivial $\mathscr{N}(s)$ is additionally proper and hence $\mathscr{N}^{\perp}(s)$ is non-empty. Let $x_1, x_2 \in \mathscr{N}^{\perp}(s)$. Since s is linear and $s(x_1) \neq 0,\ s(x_2) \neq 0$, there exists a scalar $\alpha \neq 0$ such that

$$s(x_1) + \alpha s(x_2) = s(x_1 + \alpha x_2) = 0 \ .$$

It follows that

$$x_1 + \alpha x_2 \in \mathscr{N}(s) \ . \tag{A.13}$$

On the other hand since $\mathscr{N}^{\perp}(s)$ is a vector space and $x_1, x_2 \in \mathscr{N}^{\perp}(s)$ we must have that

$$x_1 + \alpha x_2 \in \mathscr{N}^{\perp}(s) \ . \tag{A.14}$$

From (A.13) and (A.14) we conclude that $x_1 + \alpha x_2 = 0$. Since $\alpha \neq 0$, x_1 and x_2 must be linearly dependent and hence $\dim \mathscr{N}(f)^{\perp} = 1$. □

Theorem A.3.7 (Riesz Representation Theorem)**.** *Let s be a bounded linear functional on an HS $\mathscr{X}$. There exists exactly one $x_0 \in \mathscr{X}$ such that $s(x) = \langle x, x_0 \rangle$ for all $x \in \mathscr{X}$. Moreover we have $\|s\| = \|x_0\|$.*

Proof. If s is identically zero then $x_0 = 0$ has the desired properties. Assume s is nontrivial. Then by Lemma A.3.6 $\dim \mathscr{N}(s)^{\perp} = 1$. Let $x \in \mathscr{X}$ be an arbitrary vector. Then by Theorem A.3.3 $x = y + z$ where $y \in \mathscr{N}(s)$ and $z \in \mathscr{N}(s)^{\perp}$. Furthermore, if z_0 is a unit vector in $\mathscr{N}(s)^{\perp}$ we can write:

$$x = (x - \langle x, z_0 \rangle z_0) + \langle x, z_0 \rangle z_0$$

where $(x - \langle x, z_0 \rangle z_0) \in \mathscr{N}(s)$, namely, $s(x - \langle x, z_0 \rangle z_0) = 0$. Therefore

$$s(x) = s(\langle x, z_0 \rangle z_0) = \langle x, z_0 \rangle s(z_0) = \langle x, s(z_0) z_0 \rangle$$

and if we let $x_0 = s(z_0) z_0$ then $s(x) = \langle x, x_0 \rangle$. Assume now there exists another vector x_1 such that $s(x) = \langle x, x_1 \rangle$. Then we would have $\langle x, x_0 - x_1 \rangle = 0$. If we take $x = x_0 - x_1$ we obtain $\|x_0 - x_1\|^2 = 0$ which implies $x_0 = x_1$.

Now, recall the definition of norm for a linear mapping given in (A.6). Observe that

$$\|s\| = \sup_{\|x\|=1} |s(x)| = \sup_{\|x\|=1} |\langle x, x_0 \rangle| \leq \sup_{\|x\|=1} \|x\| \|x_0\| = \|x_0\| \ .$$

and

$$\|x_0\|^2 = |\langle x_0, x_0 \rangle| = |s(x_0)| \overset{(A.7)}{\leq} \|s\| \|x_0\| \ .$$

Therefore we conclude that $\|s\| = \|x_0\|$. □

Remark. For a normed space $\mathscr{X}$, $\mathscr{B}(\mathscr{X}, \mathbb{R})$ is a Banach space (see Theorem A.2.4) that we called the dual space and denoted by $\mathscr{X}^*$. The Riesz representation Theorem further shows that when $\mathscr{X}$ is an HS, elements of $\mathscr{X}^*$ are represented by vectors of $\mathscr{X}$ (as much as in $\mathbb{R}^N$, see Example A.2.3). As such one can identify $\mathscr{X}^*$ with $\mathscr{X}$. This is not true for general Banach spaces.

A.4 Operators on Hilbert Spaces

We have already introduced linear operators and functionals between normed spaces in Section A.2. We now discuss them in relation to HSs.

Basic Facts

We begin with a simple observation.

Example A.4.1 (Operators and Finite Dimensional Matrices)**.** An $N \times M$ real matrix $\boldsymbol{A}$ defines the linear operator $L : \mathbb{R}^M \to \mathbb{R}^N$,

$$(Lx)_n := \sum_{m \in \mathbb{N}_M} a_{nm} x_m \ .$$

Vice versa, suppose $L : \mathbb{R}^M \to \mathbb{R}^N$ is a linear operator and let $\{e_1, e_2, \cdots, e_\star\}$ be the canonical basis of $\mathbb{R}^\star$ where $\star \in \{M, N\}$. Assume $\langle \cdot, \cdot \rangle$ is the Canonical inner product of $\mathbb{R}^N$ and let $a_{nm} = \langle Le_m, e_n \rangle$ for any $n \in \mathbb{N}_N$ and $m \in \mathbb{N}_M$. Then for any $x \in \mathbb{R}^M, n \in \mathbb{N}_N$ we have

$$(Lx)_n = \langle Lx, e_n \rangle = \sum_{m \in \mathbb{N}_M} x_m \langle Le_m, e_n \rangle = \sum_{m \in \mathbb{N}_M} a_{nm} x_m \ .$$

Therefore we see that there is a one-to-one relation between linear operators from $\mathbb{R}^M$ to $\mathbb{R}^N$ and $N \times M$ matrices. Additionally note that equation (A.6) reads here:

$$\|L\| = \max_{x \,:\, \sum_{m \in \mathbb{N}_M} x_m^2 = 1} \sum_{n \in \mathbb{N}_N} \left(\sum_{m \in \mathbb{N}_M} a_{nm} x_m \right)^2 \ .$$

and we can conclude that $\|L\|$ is always finite. In light of the Remark that follows Theorem A.2.4, it is now clear that linear operators from $\mathbb{R}^M$ to $\mathbb{R}^N$ are automatically bounded.

We now give a characterization of finite rank operators (Definition A.2.7) acting on HSs.

Proposition A.4.1 (Representation of finite rank operators)**.** *Assume HSs* $(\mathscr{X}, \langle\cdot,\cdot\rangle_{\mathscr{X}})$ *and* $(\mathscr{Y}, \langle\cdot,\cdot\rangle_{\mathscr{Y}})$*. Then* $L \in \mathscr{B}(\mathscr{X}, \mathscr{Y})$ *is rank-K if and only if it can be written as:*

$$Lx = \sum_{k\in\mathbb{N}_K} \langle g_k, x\rangle_{\mathscr{X}} h_k \tag{A.15}$$

for families of linearly independent vectors $\{g_k : k \in \mathbb{N}_K\} \subset \mathscr{X}$ and $\{h_k : k \in \mathbb{N}_K\} \subset \mathscr{Y}$.

Proof. Suppose that L has the aforementioned decomposition. Then it is easy to check that $L(\operatorname{span}\{g_k : k \in \mathbb{N}_K\})$ is a vector space of dimension K. Conversely, let $L : \mathscr{X} \to \mathscr{Y}$ be a linear rank$-K$ operator and let $\{h_k \in \mathscr{Y} : k \in \mathbb{N}_K\}$ be a basis for $\mathscr{R}(L)$. Then for any $x \in \mathscr{X}$ there is $\mu \in \mathbb{R}^K$ such that $Lx = \sum_{n\in\mathbb{N}_K} \mu_k h_k$; note that μ is actually a function of x, namely, for all $k \in \mathbb{N}_K$ there exists $f_k : \mathscr{X} \to \mathbb{R}$ such that $\mu_k = f_k(x)$. Since L is a bounded linear operator, f_k must be a bounded linear functional. By the Riesz representation Theorem, $f_k(x) = \langle x, g_k\rangle_{\mathscr{X}}$ for a uniquely defined $g_k \in \mathscr{X}$. Therefore

$$Lx = \sum_{k\in\mathbb{N}_K} \langle x, g_k\rangle_{\mathscr{X}} h_k$$

for a family of vectors $\mathscr{A} = \{g_k \in \mathscr{X} : k \in \mathbb{N}_K\}$. We claim that $\mathscr{A}$ is made of linearly independent vectors. Suppose this is not not the case. Then without loss of generality we can assume that $g_K = \sum_{j\in\mathbb{N}_{K-1}} \lambda_j g_j$ where $\lambda \in \mathbb{R}^{K-1}$. We now have

$$Lx = \sum_{j\in\mathbb{N}_{K-1}} \langle x, g_j\rangle_{\mathscr{X}} h_j + \sum_{j\in\mathbb{N}_{K-1}} \lambda_j \langle x, g_j\rangle_{\mathscr{X}} h_K =$$

$$\sum_{j\in\mathbb{N}_{K-1}} \langle x, g_j\rangle_{\mathscr{X}} (h_j + \lambda_j h_K) \overset{\tilde{h}_j := h_j + \lambda_j h_K}{=} \sum_{j\in\mathbb{N}_{K-1}} \langle x, g_j\rangle_{\mathscr{X}} \tilde{h}_j \tag{A.16}$$

and therefore $\mathscr{R}(A)$ cannot have dimension K. We conclude that $\mathscr{A}$ is made of linearly independent vectors. □

Theorem A.4.2 (Uniqueness of the adjoint operator)**.** *Assume HSs* $(\mathscr{X}, \langle\cdot,\cdot\rangle_{\mathscr{X}})$ *and* $(\mathscr{Y}, \langle\cdot,\cdot\rangle_{\mathscr{Y}})$ *and let* $L \in \mathscr{B}(\mathscr{X}, \mathscr{Y})$*. There exists a unique bounded operator* $L^* : \mathscr{Y} \to \mathscr{X}$ *satisfying*

$$\langle L^* y, x\rangle_{\mathscr{X}} = \langle y, Lx\rangle_{\mathscr{Y}} \text{ for all } x \in \mathscr{X}, y \in \mathscr{Y} \tag{A.17}$$

we call L^* *the adjoint of* L*. We have the following properties:*

- $(\lambda L + \mu M)^* = \lambda L^* + \mu M^*$ *for all* $L, M \in \mathscr{B}(\mathscr{X}, \mathscr{Y})$ *and* $\lambda, \mu \in \mathbb{R}$
- $(L^*)^* = L$ *for all* $L \in \mathscr{B}(\mathscr{X}, \mathscr{Y})$
- $(LM)^* = M^* L^*$ *for all* $L \in \mathscr{B}(\mathscr{X}, \mathscr{Y})$ *and* $M \in \mathscr{B}(\mathscr{Y}, \mathscr{X})$.
- $\|L\| = \|L^*\|$ *for all* $L \in \mathscr{B}(\mathscr{X}, \mathscr{Y})$.

Example A.4.2 (Adjoint of finite dimensional matrices)**.** Consider the operator L introduced in Example A.4.1. We have:

$$\langle y, Lx \rangle = \sum_{n \in \mathbb{N}_N} (Lx)_n y_n = \sum_{n \in \mathbb{N}_N} \left(\sum_{m \in \mathbb{N}_M} a_{nm} x_m \right) y_n =$$

$$\sum_{m \in \mathbb{N}_M} \left(\sum_{n \in \mathbb{N}_N} a_{mn} y_n \right) x_m = \sum_{m \in \mathbb{N}_M} (L^* y) x_m = \langle L^* y, x \rangle \ . \quad \text{(A.18)}$$

This shows that the adjoint of an operator represented by a matrix $\boldsymbol{A}$ is represented by the transpose matrix $\boldsymbol{A}^\top$.

Remark. From now on we will use $\boldsymbol{A}$ both to denote the matrix and to denote the associated operator L. The specific meaning will result clear from the context. In this spirit we may write $\boldsymbol{A}^\top$ to mean L^*.

We have already seen projection operators. Now that we have introduced the notion of adjoint we can complete their characterization.

Example A.4.3 (Adjoint of finite-rank operators)**.** Assume HSs $(\mathscr{X}, \langle \cdot, \cdot \rangle_{\mathscr{X}})$ and $(\mathscr{Y}, \langle \cdot, \cdot \rangle_{\mathscr{Y}})$ and let $L \in \mathscr{B}(\mathscr{X}, \mathscr{Y})$ be a rank-K linear operator with representation (A.15). Then we have:

$$\langle y, Lx \rangle_{\mathscr{Y}} \quad = \quad \sum_{k \in \mathbb{N}_K} \langle g_k, x \rangle_{\mathscr{X}} \langle y, h_k \rangle_{\mathscr{Y}} \quad = \quad \left\langle \sum_{k \in \mathbb{N}_K} \langle y, h_k \rangle_{\mathscr{Y}} g_k, x \right\rangle_{\mathscr{X}} \quad \text{(A.19)}$$

which shows that $L^* y = \sum_{k \in \mathbb{N}_K} \langle y, h_k \rangle_{\mathscr{Y}} g_k$.

Definition A.4.1 (Self-adjoint operator)**.** An operator $L \in \mathscr{B}(\mathscr{X})$ is called *self-adjoint* if $L = L^*$.

Proposition A.4.3 (Projection Operator)**.** *Assume an HS* $(\mathscr{X}, \langle \cdot, \cdot \rangle)$*. An operator* $P \in \mathscr{B}(\mathscr{X})$ *is an orthogonal projection onto a closed vector subspace of* $\mathscr{X}$ *if and only if* $P = P^*$ *and* $P^2 = P$.

Definition A.4.2 (Unitary Operator)**.** Assume an HSs $(\mathscr{X}, \langle\cdot,\cdot\rangle)$. An operator $L \in \mathscr{B}(\mathscr{X})$ is called *unitary* if L is a bijection and satisfies

$$\langle L(x), L(y)\rangle = \langle x, y\rangle \quad \forall x, y \in \mathscr{X}$$

It is immediate to check the following property.

Proposition A.4.4. *An operator $L \in \mathscr{B}(\mathscr{X})$ is unitary if and only if*

$$LL^* = I \text{ and } L^*L = I \ .$$

The following definition extends the notion of positive definite matrices to infinite dimensional operators.

Definition A.4.3 (Positive Operator)**.** Assume an HS $(\mathscr{X}, \langle\cdot,\cdot\rangle)$. An operator $L \in \mathscr{B}(\mathscr{X})$ is said to be *positive* (*positive-definite*) if the induced bilinear map

$$s(x, y) := \langle Lx, y\rangle \tag{A.20}$$

is positive (positive-definite).

Note that by definition of positive (positive-definite) bilinear map, the property that defines a positive (positive-definite) operator reads $\langle Lx, x\rangle \geq 0$ ($\langle Lx, x\rangle > 0$). Also note that, by the symmetry of inner products, we have:

$$\langle Lx, x\rangle = \langle x, Lx\rangle$$

and therefore L can be positive only if L is self-adjoint.

Remark. We have seen that matrices represent operators between finite dimensional spaces. In light of this it is easy to check that the definition above is consistent with Definition A.3.1.

Theorem A.4.5. *Assume an HS $(\mathscr{X}, \langle\cdot,\cdot\rangle)$ and $L \in \mathscr{B}(\mathscr{X})$. The operators L^*L and LL^* are positive.*

Remark. We write $T \geq 0$ to mean that T is positive. Additionally we write

$$T \leq S \text{ if and only if } S - T \in \mathscr{B}(\mathscr{X})^+ \ .$$

Hilbert-Schmidt Operators

We conclude this survey by showing that a subclass of bounded operators forms itself an HS. Recall that the trace of a $N \times N$ real matrix $\boldsymbol{A}$ is defined as

$$\mathrm{Tr}(\boldsymbol{A}) := \sum_{n \in \mathbb{N}_N} a_{nn} \ .$$

This definition generalizes as follows to the case of operators on arbitrary HSs.

Definition A.4.4 (Trace)**.** Denote by $\mathscr{I} \subseteq \mathbb{N}$ an index set and let $\{e_i \,:\, i \in \mathscr{I}\}$ be an orthonormal basis of an HS $(\mathscr{X}, \langle\cdot,\cdot\rangle)$. We call the map

$$\begin{array}{rccl} \mathrm{Tr}: & \mathscr{B}(\mathscr{X})^+ & \to & [0,+\infty] \\ & L & \mapsto & \sum_{i\in\mathscr{I}} \langle Le_i, e_i\rangle \end{array}$$

the *trace* on $\mathscr{B}(\mathscr{X})^+$.

It is immediate to check that Tr is linear.

Proposition A.4.6. *The trace* Tr *is independent from the choice of orthonormal basis. Moreover we have:*

$$\mathrm{Tr}(L^*L) = \mathrm{Tr}(LL^*) \text{ for all } L \in \mathscr{B}(\mathscr{X})^+ \ .$$

Proof. It is not too difficult to see that both the statements follow from application of the Parseval equality (see Proposition A.3.4). In particular, we have:

$$\mathrm{Tr}(L^*L) = \sum_{i\in\mathscr{I}} \langle L^*Le_i, e_i\rangle = \sum_{i\in\mathscr{I}} \langle Le_i, Le_i\rangle = \sum_{i\in\mathscr{I}} \|Le_i\|^2 \overset{\text{Parseval eq.}}{=}$$

$$\sum_{i\in\mathscr{I}} |\langle Le_i, e_i\rangle|^2 = \sum_{i\in\mathscr{I}} |\langle L^*e_i, e_i\rangle|^2 \overset{\text{Parseval eq.}}{=}$$

$$\sum_{i\in\mathscr{I}} \|L^*e_i\|^2 = \sum_{i\in\mathscr{I}} \langle L^*e_i, L^*e_i\rangle = \sum_{i\in\mathscr{I}} \langle LL^*e_i, e_i\rangle = \mathrm{Tr}(LL^*) \ . \quad \text{(A.21)}$$

□

Definition A.4.5 (Hilbert-Schmidt Operators)**.** The set of *Hilbert-Schmidt* operators, denoted as $\mathscr{HS}(\mathscr{X})$, is defined as

$$\mathscr{HS}(\mathscr{X}) := \left\{ L \in \mathscr{B}(\mathscr{X}) \ : \ \mathrm{Tr}(LL^*) = \sum_{i\in\mathscr{I}} \|Le_i\|^2 < \infty \right\} \ .$$

Remark. Note that, although in this Subsection we refer for simplicity to spaces of operators mapping $\mathscr{X}$ into itself, many definitions and properties generalize to the more general situation where operators are defined between two different spaces. The definition of Hilbert-Schmidt operators, in particular, can be generalized in a straightforward manner to operators between different spaces, see, e.g., [231, Supplement to Chapter 1]. The space of linear operators of Hilbert-Schmidt type from $(\mathscr{X}_2, \langle\cdot,\cdot\rangle_{\mathscr{X}_2})$ into $(\mathscr{X}_1, \langle\cdot,\cdot\rangle_{\mathscr{X}_1})$ is sometimes

denoted by $\mathscr{X}_1 \bigotimes \mathscr{X}_2$. A bounded operator $A : \mathscr{X}_2 \to \mathscr{X}_1$ is of Hilbert-Schmidt type if

$$\sum_{i\in\mathbb{N}} \|Ae_i\|^2_{\mathscr{X}_1} < \infty \ . \tag{A.22}$$

Condition (A.22) ensures that $\mathscr{X}_1 \bigotimes \mathscr{X}_2$ endowed with the inner-product

$$\langle A, B\rangle_{\mathscr{X}_1 \bigotimes \mathscr{X}_2} = \sum_{i\in\mathbb{N}} \langle Ae_i, Be_i\rangle_{\mathscr{X}_1} = \mathrm{trace}(B^*A) \tag{A.23}$$

is itself a HS. To each pair $(h_1, h_2) \in \mathscr{X}_1 \times \mathscr{X}_2$ we can associate $h_1 \otimes h_2$ defined by

$$(h_1 \otimes h_2)f = \langle h_2, f\rangle_{\mathscr{X}_2} h_1 \ . \tag{A.24}$$

One can check that (A.24) is of Hilbert-Schmidt type and hence $h_1 \otimes h_2 \in \mathscr{X}_1 \bigotimes \mathscr{X}_2$. As for the finite dimensional case, elements of $\mathscr{X}_1 \bigotimes \mathscr{X}_2$ can be represented as sum of rank-1 operators (A.24).

Example A.4.4 (All finite rank operators are Hilbert-Schmidt)**.** A rank$-K$ operator $L \in \mathscr{B}(\mathscr{X})$ admits a representation of the type $Lx = \sum_{k\in\mathbb{N}_K} \langle g_k, x\rangle_{\mathscr{X}} h_k$ (Proposition A.4.1) and we have $L^*y = \sum_{k\in\mathbb{N}_K} \langle y, h_k\rangle_{\mathscr{Y}} g_k$ (Example A.4.3).

Suppose now that $\{e_i \ : \ i \in \mathscr{I}\}$ is an orthonormal basis of $\mathscr{X}$ such that, for $\mathbb{N}_K \subset \mathscr{I}$, $\{e_j \ : \ j \in \mathbb{N}_K\}$ is an orthonormal family of

$$\mathscr{R}(L) = \mathrm{span}(\{h_k \ : \ k \in \mathbb{N}_K\}) \ .$$

Then we have:

$$\mathrm{Tr}(LL^*) = \sum_{i\in\mathscr{I}} \|L^*e_i\|^2 = \sum_{i\in\mathscr{I}} \left\| \sum_{k\in\mathbb{N}_K} \langle e_i, h_k\rangle_{\mathscr{Y}} g_k \right\|^2 \overset{\langle e_i, h_k\rangle_{\mathscr{Y}} = 0 \text{ for } i\notin\mathbb{N}_K}{=}$$

$$\sum_{i\in\mathbb{N}_K} \left\| \sum_{k\in\mathbb{N}_K} \langle e_i, h_k\rangle_{\mathscr{Y}} g_k \right\|^2 \ . \tag{A.25}$$

Since the last line involves a finite summation we conclude that $\mathrm{Tr}(LL^*) < \infty$ and therefore $L \in \mathscr{HS}(\mathscr{X})$.

Proposition A.4.7. *Assume $L \in \mathscr{B}(\mathscr{X})$. Then* $\mathrm{Tr}(L^*L) \geq \|L\|$.

Proof. As seen in equation (A.21), we have: $\mathrm{Tr}(L^*L) = \sum_{i\in\mathscr{I}} \|Le_i\|^2$. In turn,

$$\|L^*x\|^2 = \sum_{i\in\mathscr{I}} |\langle L^*x, e_i\rangle|^2 = \sum_{i\in\mathscr{I}} |\langle x, Le_i\rangle|^2 \leq \sum_{i\in\mathscr{I}} \|x\|^2 \|Le_i\|^2 \ .$$

The claim now follows from the definition of operator norm $\|\cdot\|$ (see equation (A.6)) and the fact that $\|L\| = \|L^*\|$ (Theorem A.4.2). □

Theorem A.4.8. *The set of Hilbert-Schmidt operators $\mathscr{HS}(\mathscr{X})$ forms a vector subspace of $\mathscr{B}(\mathscr{X})$. Equipped with the inner product*

$$\langle L, M\rangle = \mathrm{Tr}(LM^*) \ , \tag{A.26}$$

the vector space $\mathscr{HS}(\mathscr{X})$ becomes a HS with corresponding norm

$$\|L\|_F = \sqrt{\langle L, L\rangle} \ .$$

Appendix B

Kernels and Spaces of Functions

Kernel methods collectively define a set of methodologies that are nowadays widespread in machine learning and related disciplines. In this Chapter we present some fundamental facts related to *kernels*. Our emphasis is on the theory and in particular on the functional analytic aspects. Three different but related fundamental notions of kernels are reviewed, namely the concept of *positive definite kernel* that of *reproducing kernel* and finally that of *kernel of an operator* (not to be confused with the null space of an operator). Successively we elaborate on a specific class of data dependent kernels used in [79] to estimate partial linear models. We conclude with product kernels and discuss the link with finite dimensional tensors.

Bibliographical Pointers Early references include [141] and [10] (the latter is probably the first to give a comprehensive theoretical framework). More recent references include [175, 235] and [27]. The first use of kernels in machine learning (pattern recognition) is generally attributed to [4]. Results useful for machine learning are found in [177]. Material on multilinear functionals and related aspects is found in [120]. An interpretation of vectors as functions and the consequent view of $\mathbb{R}^P$ as a RKHS is found in [96]. The latter reference does not clarify the relation with finite dimensional tensors.

B.1 Notions of Kernels

In the following we denote by $\mathscr{X}$ a nonempty abstract set (a discrete family of elements, a normed space, a Banach or Hilbert space, etc.).

Definition B.1.1 (Gram matrix)**.** Given a function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ and $(x_1, x_2, \ldots, x_N) \in \mathscr{X}^N$, the $N \times N$ matrix $\boldsymbol{K}$:

$$k_{ij} = k(x_i, x_j)$$

is called the *Gram matrix of k with respect to* $(x_1, x_2, \ldots, x_N) \in \mathscr{X}^N$.

The following is based on Definition A.3.1 (positive definite matrix).

Positive Definite Kernels

Definition B.1.2 (Positive definite kernel)**.** A function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is a *positive definite kernel* if for any $N \in \mathbb{N}$ and for any $(x_1, x_2, \ldots, x_N) \in \mathscr{X}^N$, $\boldsymbol{K}$ namely, the Gram matrix of k with respect to $(x_1, x_2, \ldots, x_N) \in \mathscr{X}^N$, satisfies[1]:

$$s(\alpha, \alpha) = \sum_{n \in \mathbb{N}_N} \sum_{m \in \mathbb{N}_N} k_{nm} \alpha_n \alpha_m \geq 0 \quad \forall \alpha \in \mathbb{R}^N .$$

Note that, by definition, a positive definite kernel satisfies $k(x, x) \geq 0$ for any[2] $x \in \mathscr{X}$. The last definition is not very useful: it naively requires to check that infinitely many Gram matrices are positive. The next condition provides a more convenient way to check for positive definiteness of a function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$.

Proposition B.1.1. *Let $(\mathscr{H}, \langle \cdot, \cdot \rangle)$ be some Hilbert space and $\phi : \mathscr{X} \to \mathscr{H}$. Then the function*

$$\begin{array}{rccc} k: & \mathscr{X} \times \mathscr{X} & \to & \mathbb{R} \\ & (x, y) & \mapsto & \langle \phi(x), \phi(y) \rangle \end{array}$$

is a positive definite kernel.

[1] In words: if the bilinear map associated to $\boldsymbol{K}$ is positive, see the beginning of Appendix A.3.

[2] To see this it is enough to take $N = 1$.

Proof. For any $N \in \mathbb{N}$ and for any $(x_1, x_2, \ldots, x_N) \in \mathscr{X}^N$ we have

$$s(\alpha,\alpha) = \sum_{n\in\mathbb{N}_N}\sum_{m\in\mathbb{N}_N} \langle\phi(x_n),\phi(x_m)\rangle\alpha_n\alpha_m =$$

$$\sum_{n\in\mathbb{N}_N}\sum_{mx\in\mathbb{N}_N} \langle\alpha_n\phi(x_n),\alpha_m\phi(x_m)\rangle = \|\sum_{n\in\mathbb{N}_N}\alpha_n\phi(x_n)\|^2 \geq 0. \quad \text{(B.1)}$$

which proves the statement. □

Reproducing Kernels

Definition B.1.3 (proper HS of functions)**.** An HS $\mathscr{H}$ of real valued functions on $\mathscr{X}$ is called *proper* if, for each $x \in \mathscr{X}$ the evaluation functional

$$\begin{array}{rccc} L_x: & \mathscr{H} & \to & \mathbb{R} \\ & f & \mapsto & f(x) \end{array} \quad \text{(B.2)}$$

is a bounded linear functional.

Unless differently specified, in the following we assume that $\mathscr{X} \subseteq \mathbb{R}^M$. If $x \in \mathscr{X}$ we write $\|x\|$ to mean $\left(\sum_{m\in\mathbb{N}_M} x_m^2\right)^{1/2}$.

Example B.1.1 ($L_2[a,b]$ is not proper)**.** Note that, in spite of being an HS of functions, $L_2[a,b]$ is not proper since its elements are not defined point-wise (Example A.1.6).

By Theorem A.3.7 for any proper Hilbert space $\mathscr{H}$ there exists $k_x \in \mathscr{H}$ which is the representation of the evaluation functional L_x (indexed by x): $u(x) = L_x u = \langle u, k_x\rangle$.

Definition B.1.4 (reproducing kernel)**.** A function

$$\begin{array}{rccc} k: \mathscr{X}\times\mathscr{X} & \to & \mathbb{R} \\ (x,y) & \mapsto & k(x,y) \end{array} \quad \text{(B.3)}$$

is a *reproducing kernel* (r.k. for short) of the HS $\mathscr{H}$ if and only if

- $\forall x \in \mathscr{X},\ k(\cdot,x) \in \mathscr{H}$
- $\forall x \in \mathscr{X},\ \forall h \in \mathscr{H}\ \langle h, k(\cdot,x)\rangle = h(x)$.

The last condition says that $k(\cdot, x)$ is the representer of the evaluation functional L_x; it goes under the name of *reproducing property.* From B.1.4 and B.1.4 it is clear that

$$k(x, y) = \langle k(\cdot, x), k(\cdot, y) \rangle, \ \forall x, y \in \mathscr{X}; \tag{B.4}$$

since $\langle \cdot, \cdot \rangle$ is symmetric, it follows that $k(x, y) = k(y, x)$ (k is symmetric too). An HS of functions that possesses a r.k. is called a *Reproducing kernel Hilbert space* (RKHS for short). It is clear that a RKHS is proper and, vice versa, that a proper HS is a RKHS.

Remark. In the following a space with r.k. k is sometimes denoted by $\mathscr{H}_k$. From now on we use indifferently k_x and $k(\cdot, x)$ to indicate the representation of L_x in $\mathscr{H}_k$.

RKHSs are spaces of functions with the property that, if a function f is close to a function g in the sense of the distance associated to the inner product, then the value of $f(x)$ is close to the value of $g(x)$ for $x \in \mathscr{X}$. This property is desirable in a variety of applications where one wants to infer functions which are sufficiently regular. A motivation for this in the context of learning is given in Chapter 2. Additionally, the boundedness property of L_x for any $x \in \mathscr{X}$ implies that any function in $\mathscr{H}_k$ is point-wise defined.

Corollary B.1.2. *In a proper Hilbert space $\mathscr{H}$ of functions defined on $\mathscr{X}$, a sequence converging in the norm sense converges point-wise, i.e., $\phi_n \to \phi$ implies $L_x \phi_n \to L_x \phi \ \forall x \in \mathscr{X}$.*

Proof. Let's assume ϕ_n converges in norm, i.e. $\lim_n \|\phi_n - \phi\| = 0$. The completeness of $\mathscr{H}$ implies $\phi \in \mathscr{H}$. The continuity of $L_x \ \forall x \in \mathscr{X}$ implies now $\lim_n L_x \phi_n = L_x \phi$. □

Theorem B.1.3 ([27], Reproducing kernel Hilbert spaces of continuous functions)**.** *Any element of $\mathscr{H}_k$ is continuous if and only if k satisfies the following conditions:*

1. *$\forall y \in \mathscr{X}, \ k(\cdot, y)$ is continuous*
2. *$\forall x \in \mathscr{X}$ there exists $r > 0$ such that the function $y \mapsto k(y, y)$ is bounded on the open ball $B(x, r) := \{x \in \mathscr{X} \ : \ \|x\| \leq r\}$.*

As special cases of the previous Theorem the functions in $\mathscr{H}_k$ are continuous in $\mathscr{X}$ when:

- $\mathscr{X}$ is a bounded interval of $\mathbb{R}^d$ and k is continuous on $\mathscr{X} \times \mathscr{X}$.
- k is bounded on $\mathscr{X}$ and continuous as a function of one variable $k(\cdot, x)$.

Proposition B.1.4. *Any finite dimensional Hilbert space of functions $\mathscr{H}$ is a proper Hilbert space.*

Proof. In finite dimensional Hilbert spaces all linear functionals are continuous. □

It is relevant to ask when a real function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ defines a reproducing kernel. In view of equation (B.4) and Proposition B.1.1 it is immediate to see that any reproducing kernel is a positive definite kernel.

The converse result, which is more delicate, completes the equivalence between positive definite functions and reproducing kernels.

Theorem B.1.5 (Moore-Aronszajn, [27])**.** *Let k be a positive definite kernel on $\mathscr{X} \times \mathscr{X}$. There exists only one reproducing kernel Hilbert space $\mathscr{H}$ with k as reproducing kernel. The subspace $\mathscr{H}_0$ of $\mathscr{H}$ spanned by the functions $\{k(\cdot, x)\}_{x \in \mathscr{X}}$ is dense in $\mathscr{H}$ and $\mathscr{H}$ is the set of functions on $\mathscr{X}$ which are pointwise limits of Cauchy sequences in $\mathscr{H}_0$ with the inner product*

$$\langle f, g \rangle_{\mathscr{H}_0} = \sum_{i \in \mathbb{N}_I} \sum_{j \in \mathbb{N}_J} \alpha_i \beta_j k(x_i, y_j) \tag{B.5}$$

where $f = \sum_{i \in \mathbb{N}_I} \alpha_i k(\cdot, x_i)$ and $g = \sum_{j \in \mathbb{N}_J} \beta_j k(\cdot, y_j)$.

Feature Maps

In view of the last Theorem, assuming that k is a real valued positive definite kernel on $\mathscr{X} \times \mathscr{X}$ we can define the so called *feature map*:

$$\begin{array}{rccc} \Phi : & \mathscr{X} & \to & \mathscr{H}_k \\ & x & \mapsto & k(\cdot, x). \end{array} \tag{B.6}$$

Φ maps each pattern $x \in \mathscr{X}$ into a functional space defined on $\mathscr{X}$ with the properties of a reproducing kernel Hilbert space. The image of Φ is the closure of the vector space of the linear combinations $f = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i)$ equipped with the dot product (B.5) which satisfies

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle.$$

With the previous argument we showed basically that $\mathscr{H}_k$ can be seen as a possible instance of the feature space induced by the mapping Φ. A different feature space can also be given. Recall that with $l_2(\mathscr{I})$ we denote the space of l_2 sequences with respect to the index set $\mathscr{I}$ (see example A.1.5 and the successive Remark). We have the following.

Lemma B.1.6. *Let $\mathscr{X}$ denote a set and let $\mathscr{I} \subseteq \mathbb{N}$. A real function k defined on $\mathscr{X} \times \mathscr{X}$ is a reproducing kernel if and only if there exists a mapping $T : \mathscr{X} \longmapsto l_2(\mathscr{I})$ such that:*

$$k(x, y) = \langle T(x), T(y) \rangle_{l_2} \ \forall (x, y) \in \mathscr{X} \times \mathscr{X} \ .$$

Proof. Let $\mathscr{H}_k$ be a RKHS of functions on $\mathscr{X}$ with kernel k. Since $\mathscr{H}_k$ is an Hilbert space, by Proposition A.3.5, there exists a space $l_2(\mathscr{I})$ and a linear isometry $Q : x \mapsto \{\langle x, e_i \rangle\}_{i \in \mathscr{I}}$. The mapping $T = Q \circ \Phi$, with Φ defined as in (B.6), meets the requirements. Conversely, given any operator $T : \mathscr{X} \to l_2(\mathscr{I})$ the function:

$$\begin{array}{rccc} k: & \mathscr{X} \times \mathscr{X} & \to & \mathbb{R} \\ & (x, y) & \mapsto & \langle T(x), T(y) \rangle_{l_2(\mathscr{I})} \end{array}$$

by Proposition B.1.1 defines a positive definite function which is, according to Theorem B.1.5, a reproducing kernel. □

B.2 Properties of Kernels

A basic and well known property of the reproducing kernel Hilbert spaces is that, given k one can easily find the representation of any continuous linear functional, as stated below.

Lemma B.2.1. *Any continuous linear functional $A : \mathscr{H}_k \longmapsto \mathbb{R}$ has a representation given by the function $k_A : \mathscr{X} \to \mathbb{R}$ defined by:*

$$k_A(x) = A k_x.$$

Proof. By Theorem A.3.7 there exists k_A such that $Af = \langle f, k_A \rangle$. By applying it to the representation of L_x one gets $Ak_x = \langle k_x, k_A \rangle = k_A(x)$. □

Thus the kernel associated to a linear functional is simply obtained by applying A to the function $k(\cdot, x)$. As we shall see next, a similar result applies to continuous operators. First, we give another definition.

Definition B.2.1 (Kernel of an operator, [27])**.** Let $\mathscr{H}$ be a Hilbert Space of functions defined on $\mathscr{X}$ and $A : \mathscr{H} \to \mathscr{H}$ be a linear operator. A function k_A:

$$\begin{array}{rccc} k_A: & \mathscr{X} \times \mathscr{X} & \to & \mathbb{R} \\ & (x, y) & \mapsto & k_A(x, y) \end{array}$$

is said to be a kernel of A if and only if:

1. $\forall y \in \mathscr{X},\ k_A(\cdot, y) \in \mathscr{H}$
2. $\forall\ y \in \mathscr{X},\ \forall\ f \in \mathscr{H},\ [Af](y) = \langle f, k_A(\cdot, y)\rangle$.

Example B.2.1 (Integral operator)**.** Consider, in particular, an *integral operator* A defined by:

$$\begin{array}{rccc} A_k: & L_2(\mathscr{X}, \mu) & \to & L_2(\mathscr{X}, \mu) \\ & f & \mapsto & \int k(x, y) f(x) d\mu(x) \end{array} \tag{B.7}$$

where $k(\cdot, y) \in L_2(\mathscr{X}, \mu)$. Note that, since $\int k(y, x) f(x) d\mu(x) = \langle k_y, f\rangle$ where $\langle \cdot, \cdot \rangle$ denotes here the canonical inner product of $L_2(\mathscr{X}, \mu)$, k is the kernel of A_k according to the former definition.

Proposition B.2.2. *Let $\mathscr{H}$ be a HS and $A : \mathscr{H} \to \mathscr{H}$ be a linear operator. Then there is at most one kernel of A.*

Proof. Suppose k_1 and k_2 are both kernel of A. Then $\forall\ y \in \mathscr{X},\ \forall\ f \in \mathscr{H}$, $0 = [Af](y) - [Af](y) = \langle f, k_1(\cdot, y)\rangle - \langle f, k_2(\cdot, y)\rangle = \langle f, k_1(\cdot, y) - k_2(\cdot, y)\rangle$ and therefore $k_1(\cdot, y) = k_2(\cdot, y)\ \ \forall\ y \in \mathscr{X}$. □

The next proposition is an immediate consequence of definition B.2.1.

Proposition B.2.3. *Given a HS of functions $\mathscr{H}$, k is its r.k. if and only if it is the kernel of the identity operator I:*

$$\begin{array}{rccc} I: & \mathscr{H} & \to & \mathscr{H} \\ & f & \mapsto & f. \end{array}$$

Conversely, given the kernel k_A of an operator $A : \mathscr{H} \to \mathscr{H}$ where $\mathscr{H}$ is a RKHS, k_A is the r.k. for $\mathscr{F} \subset \mathscr{H}$ if the restriction of A to $\mathscr{F}$, denoted as $A|_{\mathscr{F}}$, is equal to the identity: $A|_{\mathscr{F}} = I$.

For continuous operators on a proper HS we have the following.

Theorem B.2.4 (Kernel of an operator)**.** *Let $\mathscr{H}_k$ be an Hilbert space of functions defined on $\mathscr{X}$ with r.k. k. Any linear continuous operator $D : \mathscr{H}_k \to \mathscr{H}_k$ has a kernel k_D given by:*

$$k_D(x, y) = \left[D^* k_y\right](x)$$

where D^* defines the adjoint operator of D.

Proof. Since D is bounded there exists a unique adjoint operator D^* such that, $\forall f, k_y \in \mathscr{H}_k$:

$$\langle Df, k_y \rangle = \langle f, D^* k_y \rangle.$$

Now, since k_y is the representation of L_y, the latter can be restated as:

$$[Df](y) = \langle f, D^* k_y \rangle$$

where $D^* k_y \in \mathscr{H}_k$. By Definition B.2.1, it follows that $k_D(x, y) = [D^* k_y](x)$ is the kernel of D. □

Proposition B.2.5. *Let $\mathscr{F}$ be a closed subspace of a RKHS $\mathscr{H}_k$. Then $\mathscr{F}$ is a RKHS with reproducing kernel*

$$k_{\mathscr{F}}(x, y) = [Pk_y](x) \ . \tag{B.8}$$

where P denotes the orthogonal projection onto $\mathscr{F}$.

Proof. By Theorem B.2.4 the kernel of P is $k_P(x, y) = [Pk_y](x)$ (since projection operator are self-adjoint, see A.4.3). Additionally, since we have that $P|_{\mathscr{F}} = I$, by Proposition B.2.3, k_P is the reproducing kernel of the HS $(\mathscr{F}, \langle \cdot, \cdot \rangle)$. □

Theorem B.2.6 ([60, 141])**.** *Let $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ be a positive definite continuous kernel and denote by A_k the corresponding integral operator* (B.7) *defined upon the positive measure μ. Let λ_i be the $i-$th positive eigenvalue of A_k and ϕ_i the corresponding continuous eigenfunction, $A_k \phi_i = \lambda_i \phi_i$. Then*

$$k(x, y) = \sum_{i \in \mathscr{I}} \lambda_i \phi_i(x) \phi_i(y)$$

where the convergence is absolute (for any $(x, y) \in \mathscr{X} \times \mathscr{X}$) and uniform (on $\mathscr{X} \times \mathscr{X}$).

This theorem can be used to characterize the smoothness of the functions in $\mathscr{H}_k$, see, e.g., [60, 77]. Lemma B.1.6 has already showed the existence of a mapping T corresponding to a given reproducing kernel. If k satisfies the conditions of Theorem B.2.6 we have:

$$\begin{array}{rccl} T: & \mathscr{X} & \to & l_2(\mathscr{I}) \\ & x & \mapsto & \left(\lambda_i^{1/2} \phi_i(x)\right)_{i \in \mathscr{I}} \end{array} \tag{B.9}$$

where T is a possible *feature map associated with k*.

B.3 Data-dependent Kernels

Assume that a RKHS $(\mathscr{H}_k, \langle\cdot,\cdot\rangle)$ of functions on $\mathscr{X}$, a prescribed set of functions

$$\{\psi_m \ : \ \mathscr{X} \to \mathbb{R}\}_{m\in\mathbb{N}_M} \tag{B.10}$$

and points

$$(x_1, x_2, \cdots, x_N) \in \mathscr{X}^N \ , \tag{B.11}$$

are given. In this subsection we characterize the subset $\mathscr{F}$ formed by all those functions of $\mathscr{H}_k$ with the following property: $f \in \mathscr{F}$ if and only if the vector

$$(f(x_n) \ : \ n \in \mathbb{N}_N)$$

is orthogonal to

$$(\psi_m(x_n) \ : \ n \in \mathbb{N}_N) \text{ for any } m \in \mathbb{N}_M \ .$$

In words: elements of $\mathscr{F}$ are all those functions in $\mathscr{H}_k$ that are *empirically orthogonal* to the prescribed set of functions $\{\psi_m\}_{m\in\mathbb{N}_M}$. This problem was approached in [79] that proposed an algorithm for partial linear models. These models are formed by a parametric part, defined based upon functions (B.10) and a nonparametric part. They are estimated by imposing an orthogonality constraint defined based upon the given points (B.11). This constraint aims at decoupling the parametric part from the nonparametric one. Next we show that $\mathscr{F}$ is actually a RKHS and give the closed form for its reproducing kernel.

Consider the operator

$$\begin{array}{rccl} D_\psi : & \mathscr{H}_k & \to & \mathbb{R}^M \\ & h & \mapsto & \left(\sum_{n\in\mathbb{N}_N} \psi_m(x_n)h(x_n) \ : \ m \in \mathbb{N}_M\right) \end{array}$$

with null space

$$\mathscr{N}(D_\psi) = \left\{h \in \mathcal{H}_k \ : \ D_\psi h = 0 \in \mathbb{R}^M\right\} \ .$$

It is easy to see that characterizing the desired set of functions $\mathscr{F}$, corresponds to characterizing $\mathscr{N}(D_\psi)$. To begin with, note that we have

$$\sum_{m\in\mathbb{N}_M} \alpha_m (D_\psi h)_m = \sum_{m\in\mathbb{N}_M} \sum_{n\in\mathbb{N}_N} \alpha_m \psi_m(x_n) h(x_n) =$$

$$\left\langle \sum_{m\in\mathbb{N}_M} \sum_{n\in\mathbb{N}_N} \alpha_m \psi_m(x_n) k_{x_n}, h \right\rangle \tag{B.12}$$

which shows that the adjoint operator $D_\psi^* : \mathbb{R}^M \to \mathcal{H}_k$ is given by

$$D_\psi^* \alpha = \sum_{m \in \mathbb{N}_M} \sum_{n \in \mathbb{N}_N} \alpha_m \psi_m(x_n) k_{x_n} \ .$$

The range of the latter is:

$$\mathscr{R}(D_\psi^*) = \left\{ D_\psi^* \alpha \in \mathcal{H}_k \ : \ \alpha \in \mathbb{R}^M \right\} \ .$$

Now observe that $\mathscr{R}(D_\psi^*)$ is a finite dimensional subspace of $\mathcal{H}_k$, and that

$$\left[\mathscr{R}(D_\psi^*) \right]^\perp = \mathscr{N}(D_\psi) \ .$$

Every $h \in \mathscr{H}_k$ has a unique decomposition of the form

$$h = h_1 + h_2 \text{ with } h_1 \in \mathscr{R}(D_\psi^*) \text{ and } h_2 \in \mathscr{N}(D_\psi) \ .$$

By Proposition B.2.5 $\mathscr{N}(D_\psi)$ is a RKHS and its kernel function is solely given in terms of k, the reproducing kernel of $\mathscr{H}_k$, and of $P_{\mathscr{N}(D_\psi)}$, the orthogonal projection onto $\mathscr{N}(D_\psi)$. The latter is given by (see, e.g., [136, Theorem 1, Chapter 6.9]):

$$P_{\mathscr{N}(D_\psi)} h = Ih - D_\psi^* \left(D_\psi D_\psi^* \right)^{-1} D_\psi h = h - \sum_{n \in \mathbb{N}_N} \left(T \Omega^{-1} D_\psi h \right)_n k_{x_n} \quad \text{(B.13)}$$

where $k_{ij} := k(x_i, x_j)$, $t_{im} := \psi_m(x_i)$ and $\Omega := D_\psi D_\psi^* = T^\top K T$ is assumed to be a full-rank matrix. By Proposition B.2.5 we now have that the reproducing kernel of $\mathscr{N}(D_\psi)$, simply denoted by $k_\mathscr{N}$, is given by

$$k_\mathscr{N}(z_1, z_2) = \left[P_{\mathscr{N}(D_\psi)} k(\cdot, z_2) \right](z_1) = k(z_1, z_2) - \sum_{n \in \mathbb{N}_N} \left(T \Omega^{-1} D_\psi k_{z_2} \right)_n k(x_n, z_1) \ . \quad \text{(B.14)}$$

Note that the corresponding kernel matrix $[K_\mathscr{N}]_{ij} = k_\mathscr{N}(x_j, x_j)$, is:

$$K_\mathscr{N} = K - K \left(T \Omega^{-1} T_N^\top \right) K \ . \quad \text{(B.15)}$$

To summarize, the functions of $\mathscr{H}_k$, that are empirically orthogonal to the prescribed set of functions $\{\psi_m\}_{m \in \mathbb{N}_M}$, form a subspace of $\mathscr{H}_k$ which is a RKHS with r.k. (B.14). Learning in this subspace, that is, using the kernel function (B.14), ensures the desired orthogonality property for the estimated nonparametric model.

B.4 Kernels and Tensors

Product kernels, that is, kernels of the type:

$$k(x, y) = k^1(x_1, y_1) k^2(x_2, y_2) \cdots k^P(x_P, y_P) \quad \text{(B.16)}$$

where $\{k^p\}_{p\in\mathbb{N}_P}$ is a set of prescribed positive definite kernels, play an important role in learning. They have been used in nonparametric statistics as a main tool to construct nonparametric multivariate models, see [96,97,113,133,235]. They include the widely used Gaussian RBF kernel, see also Section 4.2.1. Here we gather insight on the RKHSs associated to product kernels. We then show that spaces of finite dimensional (higher order) tensors are a special instance of this framework. This is important because it suggests that one could extend spectral regularization, proposed for finite dimensional tensor models in Chapter 3, to nonparametric multivariate models, see also Section 4.2.4.

We begin by discussing multilinear functionals in connection to general HSs and then focus on RKHSs.

Space of Multilinear Functionals

The next definition generalizes that of a linear functional.

Definition B.4.1 (Bounded multilinear functionals)**.** Assume HSs $(\mathscr{H}_1, \langle\cdot,\cdot\rangle_{\mathscr{H}_1})$, $(\mathscr{H}_2, \langle\cdot,\cdot\rangle_{\mathscr{H}_2})$, $\ldots$, $(\mathscr{H}_P, \langle\cdot,\cdot\rangle_{\mathscr{H}_P})$. The mapping

$$\psi : \mathscr{H}_1 \times \mathscr{H}_2 \times \cdots \times \mathscr{H}_P \to \mathbb{R} \tag{B.17}$$

is a *bounded* (equivalently *continuous*) *multilinear functional*, if it is linear in each argument and there exists $c \in [0,\infty)$ such that

$$|\psi(h_1, h_2, \ldots, h_P)| \leq c\|h_1\|_{\mathscr{H}_1}\|h_2\|_{\mathscr{H}_2}\cdots\|h_p\|_{\mathscr{H}_P}$$

for all $h_i \in \mathscr{H}_i$, $i \in \mathbb{N}_P$.

Definition B.4.2 (Hilbert-Schmidt multilinear functionals)**.** A multilinear mapping ψ is called *Hilbert-Schmidt*, if it is bounded and satisfies

$$\sum_{e_1\in\mathscr{E}_1}\sum_{e_2\in\mathscr{E}_2}\cdots\sum_{e_P\in\mathscr{E}_P}|\psi(e_1, e_2, \ldots, e_P)|^2 < \infty$$

for one (equivalently each) orthonormal basis $\mathscr{E}_p$ of $\mathscr{H}_p$, $p \in \mathbb{N}_P$.

We now have the following.

Proposition B.4.1 (HS of multilinear functionals)**.** *The collection of Hilbert-Schmidt multilinear functionals endowed with the inner product*

$$\langle\psi,\xi\rangle_{\text{HSF}} := \sum_{e_1\in\mathscr{E}_1}\sum_{e_2\in\mathscr{E}_2}\cdots\sum_{e_P\in\mathscr{E}_P}\psi(e_1, e_2, \ldots, e_P)\xi(e_1, e_2, \ldots, e_P) \tag{B.18}$$

forms — by completion — a HS, denoted by HSF*. The sum in* (B.18) *is absolutely convergent and does not depend on the choice of orthonormal basis* $\mathscr{E}_p$*,* $p \in \mathbb{N}_P$*.*

Proof. See [120, Proposition 2.6.2]. □

Proposition B.4.2. *The multilinear functional associated to any $P-$tuple $(h_1, h_2, \ldots, h_P) \in \mathscr{H}_1 \times \mathscr{H}_2 \times \cdots \times \mathscr{H}_P$ and defined by*

$$\psi_{h_1,h_2,\ldots,h_P}(f_1, f_2, \ldots, f_P) := \langle h_1, f_1\rangle_{\mathscr{H}_1}\langle h_2, f_2\rangle_{\mathscr{H}_2} \cdots \langle h_P, f_P\rangle_{\mathscr{H}_P} \tag{B.19}$$

belongs to HSF*. Furthermore it holds that*

$$\langle \psi_{h_1,h_2,\ldots,h_P}, \psi_{g_1,g_2,\ldots,g_P}\rangle_{HSF} = \langle h_1, g_1\rangle_{\mathscr{H}_1}\langle h_2, g_2\rangle_{\mathscr{H}_2} \cdots \langle h_P, g_P\rangle_{\mathscr{H}_P} \ . \tag{B.20}$$

Proof. The functional $\psi_{h_1,h_2,\ldots,h_P}$ defined as in (B.19), is clearly multilinear. It is further bounded since we have that:

$$\begin{aligned}
&\psi_{h_1,h_2,\ldots,h_P}(f_1, f_2, \ldots, f_P) = \\
&\qquad \langle h_1, f_1\rangle_{\mathscr{H}_1}\langle h_2, f_2\rangle_{\mathscr{H}_2} \cdots \langle h_P, f_P\rangle_{\mathscr{H}_P} \overset{\text{Cauchy-Schwarz ineq.}}{\leq} \\
&\qquad\qquad \|h_1\|_{\mathscr{H}_1}\|f_1\|_{\mathscr{H}_1}\|h_2\|_{\mathscr{H}_1}\|f_2\|_{\mathscr{H}_1} \cdots \|h_P\|_{\mathscr{H}_1}\|f_P\|_{\mathscr{H}_1} \ .
\end{aligned} \tag{B.21}$$

Additionally we have:

$$\begin{aligned}
&\|\psi_{h_1,h_2,\ldots,h_P}\| = \sum_{e_1\in\mathscr{E}_1}\sum_{e_2\in\mathscr{E}_2}\cdots\sum_{e_P\in\mathscr{E}_P} |\psi_{h_1,h_2,\ldots,h_P}(e_1, e_2, \ldots, e_P)|^2 = \\
&\qquad \sum_{e_1\in\mathscr{E}_1}\sum_{e_2\in\mathscr{E}_2}\cdots\sum_{e_P\in\mathscr{E}_P} |\langle h_1, e_1\rangle|^2|\langle h_2, e_2\rangle|^2|\langle h_P, e_P\rangle|^2 = \\
&\left(\sum_{e_1\in\mathscr{E}_1} |\langle h_1, e_1\rangle|^2\right)\left(\sum_{e_2\in\mathscr{E}_2} |\langle h_2, e_2\rangle|^2\right)\cdots\left(\sum_{e_P\in\mathscr{E}_P} |\langle h_P, e_P\rangle|^2\right) \overset{\text{Parseval eq.}}{=} \\
&\qquad\qquad \|h_1\|^2_{\mathscr{H}_1}\|h_2\|^2_{\mathscr{H}_2}\cdots\|h_P\|^2_{\mathscr{H}_P}
\end{aligned} \tag{B.22}$$

and since $h_p \in \mathscr{H}_p$ for any $p \in \mathbb{N}_P$ it follows that $\psi_{h_1,h_2,\ldots,h_P}$ is indeed Hilbert-Schmidt. Now we have:

$$\langle \psi_{h_1,h_2,\ldots,h_P}, \psi_{g_1,g_2,\ldots,g_P} \rangle_{HSF} =$$

$$\sum_{e_1 \in \mathscr{E}_1} \sum_{e_2 \in \mathscr{E}_2} \cdots \sum_{e_P \in \mathscr{E}_P} \psi_{h_1,h_2,\ldots,h_P}(e_1, e_2, \ldots, e_P) \psi_{g_1,g_2,\ldots,g_P}(e_1, e_2, \ldots, e_P) =$$

$$\left(\sum_{e_1 \in \mathscr{E}_1} \langle h_1, e_1 \rangle_{\mathscr{H}_1} \langle g_1, e_1 \rangle_{\mathscr{H}_1} \right) \cdots \left(\sum_{e_P \in \mathscr{E}_P} \langle h_P, e_P \rangle_{\mathscr{H}_P} \langle g_P, e_P \rangle_{\mathscr{H}_1} \right) =$$

$$\langle h_1, f_1 \rangle_{\mathscr{H}_1} \cdots \langle h_P, f_P \rangle_{\mathscr{H}_2} \ . \quad \text{(B.23)}$$

□

Product Kernels

Proposition B.4.3 (Product Kernels are Reproducing Kernels)**.** *Consider positive definite kernels $k^p : \mathscr{X}_p \times \mathscr{X}_p \to \mathbb{R}$, $p \in \mathbb{N}_P$. Then there exists only one RKHS $\mathscr{H}_k$ with reproducing kernel*[3]*:*

$$\begin{array}{rccl} k: & (\times_{p \in \mathbb{N}_P} \mathscr{X}_p)^2 & \to & \mathbb{R} \\ & (x, y) & \mapsto & k^1(x_1, y_1) k^2(x_2, y_2) \cdots k^P(x_P, y_P) \ . \end{array} \quad \text{(B.24)}$$

Proof. Since k^p, $p \in \mathbb{N}_P$ are positive definite kernels, by Theorem B.1.5 there are P corresponding RKHSs $(\mathscr{H}_1, \langle \cdot, \cdot \rangle_{\mathscr{H}_1})$, $(\mathscr{H}_2, \langle \cdot, \cdot \rangle_{\mathscr{H}_2})$, $\ldots$, $(\mathscr{H}_P, \langle \cdot, \cdot \rangle_{\mathscr{H}_P})$. By Proposition B.4.2, $\psi_{k^1_{x_1}, k^2_{x_2}, \ldots, k^P_{x_P}}$ is an element of the HS HSF associated to these RKHSs. We now have:

$$k(x, y) \overset{by(B.24)}{=} k^1(x_1, y_1) k^2(x_2, y_2) \cdots k^P(x_P, y_P) \overset{\text{by Proposition B.4.2}}{=}$$

$$\langle \psi_{k^1_{x_1}, k^2_{x_2}, \ldots, k^P_{x_P}}, \psi_{k^1_{y_1}, k^2_{y_2}, \ldots, k^P_{y_P}} \rangle_{HSF} \quad \text{(B.25)}$$

and therefore, by Proposition B.1.1 k is a positive definite kernel. Theorem B.1.5 now ensures that there exists only one RKHS obtained completing the subspace spanned by the functions $\left\{ k_x := k^1_{x_1} k^2_{x_2} \cdots k^P_{x_P} \right\}_{x \in \times_{p \in \mathbb{N}_P} \mathscr{X}_p}$. □

[3]Note that we use $x = (x_1, x_2, \cdots, x_P)$ and $y = (y_1, y_2, \cdots, y_P)$ to indicate elements of $\times_{p \in \mathbb{N}_P} \mathscr{X}_p$; these are vectors in an abstract sense, see Remark at page 151.

Remark. Note that, strictly speaking, the product kernel above differs from the one considered in (4.20). Indeed the latter defines the mapping $\tilde{k} : \mathscr{X}^2 \to \mathbb{R}$ rather than $k : (\times_{p\in\mathbb{N}_P} \mathscr{X}_p)^2 \to \mathbb{R}$ considered in (B.24). The latter is more general: one has $\tilde{k} : (x, y) \mapsto k(I(x), I(y))$ where I is the inclusion

$$\begin{array}{rccc} I: & \mathscr{X} & \to & \mathscr{X}^P \\ & x & \mapsto & (x, x, \cdots, x) \ . \end{array}$$

Finite Dimensional Tensors

Finite dimensional tensors (see Subsection 2.3.2) arise as a special case of the former setting. This can be seen by regarding vectors of $\mathbb{R}^I$ as functions on the discrete domain $\mathscr{X} = \mathbb{N}_I$. That is, by interpreting $x \in \mathbb{R}^I$ as a mapping[4] taking $i \in \mathbb{N}_I$ as input and giving $x(i)$ as output. In this way, $\mathbb{R}^I$ is regarded as a space of functions with the inner product $\langle f, g\rangle = \sum_{i\in\mathbb{N}_I} f(i)g(i)$; $\{e_i\}_{i\in\mathbb{N}_I}$ is the canonical basis (of functions). Being finite dimensional this space admits a reproducing kernel (Proposition B.1.4). Specifically, we have

$$\begin{array}{rccc} k^\star: & \mathbb{N}_I \times \mathbb{N}_I & \to & \mathbb{R} \\ & (i, j) & \mapsto & \langle e_i, e_j\rangle \ . \end{array}$$

Indeed observe that $k^\star_i = k^\star(i, \cdot) = e_i$ is the representer of the evaluation functional $L_i : f \mapsto f(i)$; for any $f \in \mathbb{R}^P$,

$$f(i) = \langle f, k^\star_i\rangle = \langle f, e_i\rangle$$

and also

$$k^\star(i, j) = \langle k^\star_i, k^\star_j\rangle = \langle e_i, e_j\rangle \ .$$

Take discrete sets $\mathbb{N}_{I_p}$, $p \in \mathbb{N}_P$; as in the rest of this thesis we denote by $i_1 i_2 \cdots i_P$ a generic element of $\times_{p\in\mathbb{N}_P} \mathbb{N}_{I_p}$. Consider now the product kernel:

$$\begin{array}{rccc} k: & (\times_{p\in\mathbb{N}_P} \mathbb{N}_{I_p})^2 & \to & \mathbb{R} \\ & (i_1 i_2 \cdots i_P, i'_1 i'_2 \cdots i'_P) & \mapsto & k^\star(i_1, i'_1)k^\star(i_2, i'_2) \cdots k^\star(i_P, i'_P) \ . \end{array} \tag{B.26}$$

By Proposition B.4.3, the latter is associated with the (reproducing kernel) Hilbert space of functions corresponding to

$$\operatorname{span}\{e_{i_1} e_{i_2} \cdots e_{i_P}\}_{i_1 i_2 \cdots i_P \in \times_{p\in\mathbb{N}_P} \mathbb{N}_{I_p}} \ .$$

This space is $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_P}$ in disguise, see Section 2.3.2. In fact, the function $e_{i_1} e_{i_2} \cdots e_{i_P}$ is naturally associated to the rank-1 tensor $e_{i_1} \otimes e_{i_2} \otimes$

[4]Note that, to emphasize this functional interpretation, we denote here the $i-$th component of x by $x(i)$ instead of x_i, used everywhere else in this thesis.

$\cdots \otimes e_{i_P}$ as we have:

$$(e_{i_1} \otimes e_{i_2} \otimes \cdots \otimes e_{i_P})_{i'_1 i'_2 \cdots i'_P} = e_{i_1}(i'_1) e_{i_2}(i'_2) \cdots e_{i_P}(i'_P) \ .$$

This fact shows that finite dimensional tensors are in fact a special case of RKHSs associated to product kernels. In Section 2.3.2 we have shown that, for the finite dimensional case, one can conveniently use non-Hilbertian norms to learn tensor-based models by enforcing certain structural assumptions. For kernel models, on the other hand, classical studies focused on Hilbertian norms [77,159,177,235]. An important exception is constituted by the *Multiple Kernel Learning* (MKL) framework, see in particular [69,142,143]. This setting generalizes to function spaces the use of l_1 penalties [140, 206, 240, 241], block-l_1 and generalization thereof [11, 237, 239]. Although spectral regularization has also been considered for 2-way kernel models [2, 8], mainly in connection to multi-task learning [6, 7], to the best of our knowledge the use of spectral penalties in a multi-way setting has not been explored so far. This fact suggests a new line of research, see also Section 4.2.4.

Appendix C

Optimization in Hilbert Spaces and LS-SVM

In this Appendix we first recall some elementary results concerning optimization in Hilbert Spaces; successively, we detail the derivation of LS-SVM for classification starting from (4.37). The rigorous derivation followed here considers the infinite dimensionality of feature spaces, while others are considering optimization in a finite dimensional space. In fact, the linear system (C.11) that we end up with, coincides with the one obtained in [200].

Bibliographical Pointers Among others, optimization in Banach and Hilbert spaces is discussed in [17, 76, 136, 167]. Convex optimization in HSs is the subject of [19] which also deals with monotone operator theory.

C.1 Generalized Differential and Gradient

In the following $(\mathscr{H}, \langle\cdot,\cdot\rangle_{\mathscr{H}})$ will denote a HS and f a functional on $\mathscr{H}$, namely a mapping of the type $f : \mathscr{H} \to \mathbb{R}$. We recall that f is convex if $\mathrm{dom}(f) := \{h \in \mathscr{H} \ : \ |f(h)| < \infty\}$ is a convex set and $f(\alpha h_1 + (1-\alpha)h_2) \leq \alpha f(h_1) + (1-\alpha)f(h_2)$. Notice that the latter is true in particular when f is linear or affine.

Definition C.1.1 (Subgradient and Subdifferential [76])**.** Let $f : \mathscr{H} \to \mathbb{R}$ be a convex functional. An element $g \in \mathscr{H}$ is called *subgradient* of f at $h_0 \in \mathrm{dom}(f)$

if for any $h \in \mathrm{dom}(f)$ we have

$$f(h) \geq f(h_0) + \langle g, h - h_0 \rangle_{\mathscr{H}} \ .$$

The set of all subgradients of f at h_0 is called the *subdifferential* of f at h_0 and it is denoted by $\partial f(h_0)$.

Remark. Before proceeding we remark that the HS setting we consider here translates into simpler results and definitions than those stated in terms of Banach spaces [17, 76, 136]. In particular, the fact that HS's are *reflexive*[1] implies that subgradients of functionals can be considered as elements of the same space and the use of more general duality pairings (see, e.g., [167]) can be avoided.

Definition C.1.2 (Gateaux Differential)**.** Let $f : \mathscr{H} \to \mathbb{R}$ be a convex functional. We call f differentiable in a direction s at a point $h \in \mathrm{dom}(f)$ if the following limit exists:

$$f'(h; s) = \lim_{\alpha \to 0} \frac{1}{\alpha} \left(f(h + \alpha s) - f(h) \right) \ . \tag{C.1}$$

If there exists $h^\star \in \mathscr{H}$ such that

$$f'(h; s) = \langle s, h^\star \rangle_{\mathscr{H}} \ \ \forall s \in \mathscr{H} \tag{C.2}$$

we say that f is *Gateaux-differentiable* at h, call $h^\star$ the *Gateaux-differential* of f at h and denote it by $f'(h)$.

Many properties of differentials from finite-dimensional calculus can be extended to the present generalized notion of differentials. For example it can be shown (see, e.g., [76]) that if f is Gateaux-differentiable at $h \in \mathscr{H}$ then $\partial f(h) = \{f'(h)\}$. Conversely, if f is continuous and possesses unique subgradient g at $h \in \mathrm{dom}(f)$, then f is Gateaux-differentiable at h and $f'(h) = g$.

Remark. If f is a continuous linear functional, then by the Riesz theorem there exists $h^\star$ such that $f(h) = \langle h, h^\star \rangle_{\mathscr{H}}$ for any $h \in \mathscr{H}$. It is immediate to see now that

$$f'(h; s) = \lim_{\alpha \to 0} \frac{1}{\alpha} \left(f(h + \alpha s) - f(h) \right) = \langle s, h^\star \rangle_{\mathscr{H}}$$

and hence that $h^\star$ is the Gateaux-differential at h for any $h \in \mathscr{H}$. Similarly if f is a continuous affine functional: $f(h) = \langle h, h^\star \rangle_{\mathscr{H}} + b$ then again $h^\star$ is the Gateaux-differential at h for any $h \in \mathscr{H}$.

Remark. If $f(h) = \langle h, h \rangle_{\mathscr{H}}$ simple calculus shows that equation (C.1) reads $f'(h; s) = 2\langle s, h \rangle_{\mathscr{H}}$. Hence by equation (C.2) $f'(h) = 2h$.

[1] We recall that a Banach space is said to be reflexive if it corresponds to its algebraic dual (see Definition A.2.8). All Hilbert spaces are reflexive.

Separable Functionals

Given two HS's $(\mathscr{H}_1, \langle\cdot,\cdot\rangle_{\mathscr{H}_1})$ and $(\mathscr{H}_2, \langle\cdot,\cdot\rangle_{\mathscr{H}_2})$ we can consider the product space $\mathscr{H}_1 \times \mathscr{H}_2$ consisting of ordered pairs (h_1, h_2). Such a space can be turned into a HS $\mathscr{H}$ based upon the inner product

$$\langle (h_1, h_2), (g_1, g_2) \rangle_{\mathscr{H}} := \langle h_1, g_1 \rangle_{\mathscr{H}_1} + \langle h_2, g_2 \rangle_{\mathscr{H}_2} \ .$$

Definition C.1.3. A functional on $\mathscr{H}$ is said to be *separable*, if

$$f((h_1, h_2)) = f_1(h_1) + f_2(h_2) \ .$$

If such a functional is differentiable, by (C.1) it is immediate to see that:

$$f'((h_1, h_2); (s_1, s_2)) = f_1'(h_1; s_1) + f_2'(h_1, s_1) \ .$$

Additionally, (C.2) becomes now

$$f'((h_1, h_2); (s_1, s_2)) = \langle s_1, h_1^\star \rangle_{\mathscr{H}_1} + \langle s_2, h_2^\star \rangle_{\mathscr{H}_2} \ \ \forall (s_1, s_2) \in \mathscr{H} \tag{C.3}$$

and the Gateaux-differential is then $f'((h_1, h_2)) = (h_1^\star, h_2^\star)$. These facts can be extended to the general $T-$fold product $\mathscr{H}_1 \times \mathscr{H}_2 \times \cdots \times \mathscr{H}_T$ in a straightforward manner.

C.2 Lagrange Multipliers Theorem

Here we recall the Lagrange multiplier theorem that we need in deriving the set of linear equations corresponding to the LS-SVM primal problem. Results of this type are found in [17, 19, 136]. For $m \in \mathbb{N}_M$ and $a_m \in \mathscr{H}$ consider the affine functional $r_m : \mathscr{H} \to \mathbb{R}$ defined by $r_m(h) = \langle h, a_m \rangle_{\mathscr{H}} + b_m$ for some $B \in \mathbb{R}^M$. Let f and g_s, for $s \in \mathbb{N}_S$, denote convex and continuous functionals on $\mathscr{H}$. Consider the following constrained problem:

$$\begin{array}{lrcll} \min_{h \in \mathscr{H}} & f(h) & & & \\ \text{such that} & r_m(h) & = & 0, \quad m & \in \mathbb{N}_M \\ & g_s(h) & \leq & 0, \quad s & \in \mathbb{N}_S \ . \end{array} \tag{C.4}$$

The corresponding *Lagrange functional* $L : \mathrm{dom}(f) \times \mathbb{R}^S \times \mathbb{R}^M \to \mathbb{R}$ is:

$$L(h, \lambda, \alpha) = f(h) + \sum_{s \in \mathbb{N}_S} \lambda_s g_s(h) + \sum_{m \in \mathbb{N}_M} \alpha_m r_m(h) \ .$$

Additionally, let

$$\mathscr{F} := \mathrm{dom}(f) \bigcap_{s \in \mathbb{N}_S} \mathrm{dom}(g_s)$$

and

$$\mathscr{A} := \{h \in \mathscr{H} \ : \ r_m(h) = 0 \ \forall \ m \in \mathbb{N}_M, \ g_s(h) \leq 0 \ \forall \ s \in \mathbb{N}_S\} \ .$$

The next Theorem is a restatement of [17, Theorem 1.2 and Theorem 1.3 page 175-176].

Theorem C.2.1 (Lagrange Multiplier Theorem [17])**.** *Suppose that*
1.) $g_s(h) < 0 \ \forall \ s \in \mathbb{N}_S$ *for some point* $h \in \mathscr{A}$
2.) $0 \in \operatorname{int}\{(r_1(h), r_2(h), \ldots, r_M(h)) \ : \ h \in \mathscr{F}\}$.
Then $h^\star \in \mathscr{A}$ *is an optimal solution to* (C.4) *if there exists for any* $s \in \mathbb{N}_S$ *a real number* $\lambda_s^\star$*, and for any* $m \in \mathbb{N}_M$ *a real number* $\alpha_m^\star$*, such that:*

a.) $0 \in \partial f(h^\star) + \sum_{s\in\mathbb{N}_S} \lambda_s^\star \partial g_s(h^\star) + \sum_{m\in\mathbb{N}_M} \alpha_m^\star r'_m(h^\star)$

b.) $\lambda_s^\star \geq 0$

c.) $\lambda_s^\star g_s(h^\star) = 0$.

C.3 Derivation of LS-SVM for Classification

We now base ourselves upon Theorem C.2.1 in order to derive the optimality condition of the equality constrained problem (4.37):

$$\begin{array}{ll} \min\limits_{(\Psi,e,b)\in \mathrm{HSF}\times\mathbb{R}^M\times\mathbb{R}} & \frac{1}{2}\langle\Psi,\Psi\rangle_{\mathrm{HSF}} + \gamma\frac{1}{2}\sum_{m\in\mathbb{N}_M} e_m^2 \\ \text{such that} & y_m(\langle\Psi, \tilde{\phi}(\mathcal{X}^{(m)})\rangle_{\mathrm{HSF}} + b) = 1 - e_m, \ m \in \mathbb{N}_M \ . \end{array}$$

The problem involves finding an optimal ordered pair $(\Psi^\star, e^\star, b^\star)$ in the product space[2] $\mathrm{HSF} \times \mathbb{R}^M \times \mathbb{R}$. This space, denoted by $\mathscr{H}$ for convenience of notation, can be turned into a HS by means of the inner product

$$\langle(\Psi, e, b), (\Xi, f, c)\rangle_{\mathscr{H}} = \langle\Psi, \Xi\rangle_{\mathrm{HSF}} + \langle e, f\rangle + bc \ .$$

Let us define now the separable functional

$$f((\Psi, e, b)) := \frac{1}{2}\langle\Psi,\Psi\rangle_{\mathrm{HSF}} + \gamma\frac{1}{2}\sum_{m\in\mathbb{N}_M} e_m^2$$

and for $m \in \mathbb{N}_M$ the affine functional

$$r_m((\Psi, e, b)) := \langle(\Psi, e, b), (y_m\tilde{\phi}(X^{(m)}), e^{(m)}, y_m)\rangle_{\mathscr{H}} - 1 \qquad \text{(C.5)}$$

[2]Note that, although we explicitly refer to HSF (see Section 4.2 and Appendix B.4) the argument applies to any feature space with an HS structure.

where for $m \in \mathbb{N}_M$, $e^{(m)} \in \mathbb{R}^M$ is defined in terms of the Kronecker delta by $e_j^{(m)} = \delta_{mj},\ j \in \mathbb{N}_M$. With these definitions problem (4.37) can be restated as

$$\min_{(\Psi,e,b)\in\mathscr{H}} \{f((\Psi,e,b)) \ : \ r_m((\Psi,e,b)) = 0,\ m \in \mathbb{N}_M\} \ .$$

It is easy to see that f is Gateaux-differentiable at any (Ψ, e, b). We have:

$$\partial f((\Psi,e,b)) = \{f'((\Psi,e,b))\} = \{(\Psi, \gamma e, 0)\} \tag{C.6}$$

where we used the basic facts of C.1 on composite spaces and Remark C.1. By equation (C.5), and the remarks at page 190 we have

$$r'_m((\Psi,e,b)) = (y_m\tilde{\phi}(X^{(m)}), e^{(m)}, y_m) \ .$$

Now since the subdifferential in (C.6) is a singleton, point a in Theorem C.2.1 becomes, simply:

$$(\Psi^\star, \gamma e^\star, 0) = \sum_{m\in\mathbb{N}_M} \alpha_m^\star (y_m\tilde{\phi}(X^{(m)}), e^{(m)}, y_m)$$

or, equivalently:

$$\Psi^\star = \sum_{m\in\mathbb{N}_M} \alpha_m^\star y_m \tilde{\phi}(X^{(m)}) \tag{C.7}$$

$$e_m^\star = \frac{1}{\gamma}\alpha_m^\star,\ m \in \mathbb{N}_M \tag{C.8}$$

$$\sum_{m\in\mathbb{N}_M} \alpha_m^\star y_m = 0 \ . \tag{C.9}$$

Finally, notice that the set $\mathscr{A}$ of Theorem C.2.1 reads here

$$\mathscr{A} = \{r_m((\Psi,e,b)) = 0,\ m \in \mathbb{N}_M\} \ .$$

Making $r_m((\Psi,e,b)) = 0$ explicit for $m \in \mathbb{N}_M$, we obtain the additional set of conditions:

$$y_m(\langle \Psi^\star, \tilde{\phi}(\mathcal{X}^{(m)})\rangle_{\mathrm{HSF}} + b^\star) = 1 - e_m^\star,\ m \in \mathbb{N}_M \ . \tag{C.10}$$

Replacing (C.7) and (C.8) into the latter to eliminate the primal variables $\Psi^\star$ and $e^\star$, and keeping into account (C.9), one obtains the system of linear equations:

$$\begin{bmatrix} 0 & Y^\top \\ Y & \boldsymbol{\Omega} + \frac{1}{\gamma}\boldsymbol{I}_M \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_N \end{bmatrix} \tag{C.11}$$

where $1_M = (1, 1, \ldots, 1) \in \mathbb{R}^M$, $\boldsymbol{I}_M = \mathrm{diag}(1_M)$ and $\boldsymbol{\Omega} \in \mathbb{R}^M \otimes \mathbb{R}^M$ is defined entry-wise by

$$(\boldsymbol{\Omega})_{ij} = y_i y_j \langle \tilde{\phi}(\mathcal{X}^{(i)}), \tilde{\phi}(\mathcal{X}^{(j)}) \rangle_{\mathrm{HSF}} = y_i y_j k(\mathcal{X}^{(i)}, \mathcal{X}^{(j)}) \ .$$

Bibliography

[1] NIPS Workshop on Tensors, Kernels, and Machine Learning (TKML). Whistler, BC, Canada, 2010. (Cited on pages 3 and 31)

[2] Abernethy, J., Bach, F., Evgeniou, T., and Vert, J. A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research 10* (2009), 803–826. (Cited on pages 6, 28, 50, and 187)

[3] Achlioptas, D. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences 66*, 4 (2003), 671–687. (Cited on page 145)

[4] Aizerman, M., Braverman, E. M., and Rozonoer, L. I. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control 25* (1964), 821 – 837. (Cited on page 173)

[5] Amari, S. Natural gradient works efficiently in learning. *Neural computation 10*, 2 (1998), 251–276. (Cited on page 4)

[6] Argyriou, A., Evgeniou, T., and Pontil, M. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. MIT Press, Cambridge, MA, 2007, pp. 41–48. (Cited on pages 28, 64, and 187)

[7] Argyriou, A., Evgeniou, T., and Pontil, M. Convex multi-task feature learning. *Machine Learning 73*, 3 (2008), 243–272. (Cited on pages 28 and 187)

[8] Argyriou, A., Micchelli, C., and Pontil, M. On spectral learning. *Journal of Machine Learning Research 11* (2010), 935–953. (Cited on pages 6, 28, 50, and 187)

[9] Argyriou, A., Micchelli, C., Pontil, M., Shen, L., and Xu, Y. Efficient first order methods for linear composite regularizers. *Arxiv preprint arXiv:1104.1436* (2011). (Cited on page 52)

[10] Aronszajn, N. Theory of reproducing kernels. *Transactions of the American Mathematical Society 68* (1950), 337–404. (Cited on pages 57, 160, and 173)

[11] Bach, F. Consistency of the group Lasso and multiple kernel learning. *Journal of Machine Learning Research 9* (2008), 1179–1225. (Cited on page 187)

[12] Bach, F., Lanckriet, G., and Jordan, M. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first International Conference on Machine Learning (ICML)* (2004), ACM Press New York, NY, USA. (Cited on page 27)

[13] Bach, F., Thibaux, R., and Jordan, M. Computing regularization paths for learning multiple kernels. *Advances in Neural Information Processing Systems 17: Proceedings Of The 2004 Conference* (2004). (Cited on page 27)

[14] Baraniuk, R. Compressive sensing. *Lecture notes in IEEE Signal Processing magazine 24*, 4 (2007), 118–120. (Cited on page 3)

[15] Baraniuk, R. More is less: Signal processing and the data deluge. *Science 331*, 6018 (2011), 717. (Cited on page 3)

[16] Baraniuk, R., Davenport, M., DeVore, R., and Wakin, M. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation 28*, 3 (2008), 253–263. (Cited on page 145)

[17] Barbu, V., and Precupanu, T. *Convexity and optimization in Banach spaces*. Springer, 1986. (Cited on pages 109, 189, 190, 191, and 192)

[18] Basri, R., Hassner, T., and Zelnik-Manor, L. Approximate Nearest Subspace Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 2 (2010), 266–278. (Cited on page 101)

[19] Bauschke, H., and Combettes, P. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Verlag, 2011. (Cited on pages 7, 46, 54, 145, 189, and 191)

[20] Baxter, J. Theoretical models of learning to learn. In *T. Mitchell and S. Thrun (Eds.), Learning* (1997), Kluwer, pp. 71–94. (Cited on page 27)

[21] Beck, A., and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences 2*, 1 (2009), 183–202. (Cited on page 49)

[22] Becker, S., Candès, E. J., and Grant, M. Templates for convex cone problems with applications to sparse signal recovery. Tech. rep., Stanford University, September 2010. (Cited on page 34)

[23] Belkin, M., and Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation 15*, 6 (2003), 1373–1396. (Cited on page 17)

[24] Belkin, M., and Niyogi, P. Semi-supervised learning on Riemannian manifolds. *Machine Learning 56*, 1 (2004), 209–239. (Cited on page 17)

[25] Belkin, M., Niyogi, P., and Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research 7* (2006), 2399–2434. (Cited on pages 4, 10, and 17)

[26] Bennett, K. P., and Demiriz, A. Semi-supervised support vector machines. Unpublished manuscript based on talk given at Machines That Learn Conference, Snowbird, 1998. (Cited on page 26)

[27] Berlinet, A., and Thomas-Agnan, C. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, 2004. (Cited on pages 57, 173, 176, 177, and 178)

[28] Bertsekas, D. On the Goldstein-Levitin-Polyak gradient projection method. *Automatic Control, IEEE Transactions on 21*, 2 (1976), 174–184. (Cited on page 48)

[29] Bertsekas, D. P. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995. (Cited on page 48)

[30] Bissacco, A., Chiuso, A., and Soatto, S. Classification and recognition of dynamical models: The role of phase, independent components, kernels and optimal transport. *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2007), 1958–1972. (Cited on pages 5 and 123)

[31] Bourin, C., and Bondon, P. Efficiency of high-order moment estimates. *IEEE Trans. on Signal Processing 46*, 1 (1998), 255–258. (Cited on page 127)

[32] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *To appear in Foundations and Trends in Machine Learning* (2011). (Cited on page 45)

[33] Bradley, P. S., and Mangasarian, O. L. Massive data discrimination via linear support vector machines. Tech. Rep. Mathematical Programming Technical Report 98-05, University of Wisconsin-Madison, 1998. Submitted for publication. (Cited on page 26)

[34] Bressler, S. L., Richter, C. G., Chen, Y., and Ding, M. Cortical functional network organization from autoregressive modeling of local field potential oscillations. *Statistics in Medicine 26*, 21 (2007), 3875–3885. (Cited on page 132)

[35] Brézis, H. *Opérateurs maximaux monotones*. Elsevier, 1973. (Cited on page 46)

[36] Brillinger, D. R., and Rosenblatt, M. *Computation and interpretation of k−th order spectra*. Advanced Seminar on Spectral Analysis of Time Series, University of Wisconsin–Madison, 1966. John Wiley, New York, 1967, pp. 189–232. (Cited on page 127)

[37] Burges, C. *Advances in Kernel Methods: Support Vector Learning*. MIT Press Cambridge, MA, USA, 1999, ch. Geometry and invariance in kernel based methods, pp. 89–116. (Cited on page 104)

[38] Cai, J., Candès, E., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization 20*, 4 (2010), 1956–1982. (Cited on pages 7 and 52)

[39] Candes, E., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Journal of ACM 58*, 3 (2011), Article 11, 37p. (Cited on pages 6 and 51)

[40] Candès, E., and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics 9*, 6 (2009), 717–772. (Cited on page 51)

[41] Carroll, J., and Chang, J. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika 35*, 3 (1970), 283–319. (Cited on page 32)

[42] Caruana, R. *Learning to learn*. Springer, 1998, ch. Multitask Learning, pp. 41–75. (Cited on page 27)

[43] Chandrasekaran, V., Recht, B., Parrilo, P., and Willsky, A. The convex geometry of linear inverse problems. *Arxiv preprint arXiv:1012.0621* (2010). (Cited on pages 51 and 145)

[44] Chang, C. *Hyperspectral data exploitation: theory and applications*. Wiley-Blackwell, 2007. (Cited on page 76)

[45] Chapelle, O., Schölkopf, B., and Zien, A. *Semi-supervised learning*. MIT press Cambridge, MA, 2006. (Cited on pages 10, 17, and 23)

[46] Chen, C., Ryoo, M., and Aggarwal, J. UT-Tower Dataset: Aerial View Activity Classification Challenge. http://cvrc.ece.utexas.edu/SDHA2010/Aerial_View_Activity.html, 2010. (Cited on page 117)

[47] Chen, S., Donoho, D., and Saunders, M. Atomic decomposition by basis pursuit. *SIAM Review 43* (2001), 129–159. (Cited on page 51)

[48] Cheung, B. L. P., Riedner, B. A., Tononi, G., and Van Veen, B. Estimation of cortical connectivity from EEG using state-space models. *IEEE Trans. on Biomedical Engineering 57*, 9 (2010), 2122–2134. (Cited on page 132)

[49] Combettes, P. Iterative construction of the resolvent of a sum of maximal monotone operators. *Journal of Convex Analysis 16*, 4 (2009), 727–748. (Cited on page 45)

[50] Combettes, P., and Pesquet, J. A proximal decomposition method for solving convex variational inverse problems. *Inverse problems 24* (2008), 065014. (Cited on page 45)

[51] Combettes, P., and Pesquet, J. Proximal splitting methods in signal processing. *Arxiv preprint arXiv:0912.3522* (2009). (Cited on pages 7 and 45)

[52] Comon, P., and Jutten, C. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic Press, 2010. (Cited on pages 124 and 126)

[53] Conway, J. *A Course in Functional Analysis*. Springer, 1990. (Cited on page 149)

[54] Conway, J., Hardin, R., and Sloane, N. Packing lines, planes, etc.: Packings in Grassmannian spaces. *Experimental Mathematics 5* (1996), 139–159. (Cited on page 102)

[55] Cooley, J., and Tukey, J. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation 19*, 90 (1965), 297–301. (Cited on page 106)

[56] Coppi, R., and Bolasco, S. *Multiway data analysis*. North-Holland Publishing Co. Amsterdam, The Netherlands, 1989. (Cited on page 2)

[57] Cortes, C., and Vapnik, V. Support vector networks. *Machine Learning 20* (1995), 273–297. (Cited on pages 26 and 108)

[58] Cristianini, N., Kandola, J., Elisseeff, A., and Shawe-Taylor, J. On kernel target alignment. In *Innovations in Machine Learning*, D. Holmes and L. Jain, Eds., vol. 194 of *Studies in Fuzziness and Soft Computing*. Springer Berlin / Heidelberg, 2006, pp. 205–256. (Cited on page 106)

[59] Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J. On kernel-target alignment. In *Advances in Neural Information Processing Systems (NIPS)* (2002), vol. 14, pp. 367–373. (Cited on pages 101, 105, and 114)

[60] Cucker, F., and Zhou, D. *Learning Theory: An Approximation Theory Viewpoint (Cambridge Monographs on Applied & Computational Mathematics)*. Cambridge University Press New York, NY, USA, 2007. (Cited on page 180)

[61] Daemen, A., Signoretto, M., Gevaert, O., Suykens, J., and De Moor, B. Improved microarray-based decision support with graph encoded interactome data. *PLoS ONE 5*, 4 (2010), 1–16. (Cited on pages 12 and 30)

[62] Davenport, M., Duarte, M., Eldar, Y., and Kutyniok, G. Introduction to compressed sensing. *Electrical Engineering* (2011), 1–68. (Cited on page 3)

[63] De Brabanter, K., Karsmakers, P., Ojeda, F., Alzate, C., De Brabanter, J., Pelckmans, K., De Moor, B., Vandewalle, J., and Suykens, J. A. K. LS-SVMlab toolbox user's guide version 1.8. *Internal Report 10-146, ESAT-SISTA, K.U.Leuven (Leuven, Belgium)* (2010). (Cited on pages 68, 81, 111, and 135)

[64] De Cock, K., and De Moor, B. Subspace angles between ARMA models. *Systems & Control Letters 46*, 4 (2002), 265–270. (Cited on page 5)

[65] De Lathauwer, L. Characterizing higher-order tensors by means of subspaces. *Internal Report 11-32, ESAT-SISTA, K.U. Leuven (Leuven, Belgium)* (2011). (Cited on pages 4, 104, 105, and 144)

[66] De Lathauwer, L. A short introduction to tensor-based methods for factor analysis and blind source separation. In *7th Int. Symp. on Image and Signal Processing and Analysis (ISPA), Dubrovnik, Croatia* (2011), pp. pp. 558–563. (Cited on page 32)

[67] De Lathauwer, L., De Moor, B., and Vandewalle, J. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications 21*, 4 (2000), 1253–1278. (Cited on pages 34, 35, 51, and 132)

[68] De Lathauwer, L., De Moor, B., and Vandewalle, J. An introduction to independent component analysis. *Journal of Chemometrics 14*, 3 (2000), 123–149. (Cited on page 127)

[69] De Mol, C., De Vito, E., and Rosasco, L. Elastic-net regularization in learning theory. *Journal of Complexity 25*, 2 (2009), 201–230. (Cited on page 187)

[70] Debnath, L., and Mikusiński, P. *Hilbert spaces with applications*. Academic Press, 2005. (Cited on pages 28 and 149)

[71] Domingos, P., and Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning 29*, 2 (1997), 103–130. (Cited on page 80)

[72] Donoho, D. L. Compressed sensing. *IEEE Transactions on Information Theory 52*, 4 (2006), 1289–1306. (Cited on pages 3 and 51)

[73] Douglas, J., and Rachford, H. H. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc 82*, 2 (1956), 421–439. (Cited on page 47)

[74] Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., and Vapnik, V. Support vector regression machines. In *Advances in Neural Information Processing Systems 9* (Cambridge, MA, 1997), M. Mozer, M. Jordan, and T. Petsche, Eds., MIT Press. (Cited on page 26)

[75] Edelman, A., Arias, T. A., and Smith, S. T. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications 20*, 2 (1999), 303–353. (Cited on page 101)

[76] Ekeland, I., and Temam, R. *Convex Analysis and Variational Problems*. North-Holland Publishing Co., 1976. (Cited on pages 34, 109, 189, and 190)

[77] Evgeniou, T., Pontil, M., and Poggio, T. A Unified Framework for Regularization Networks and Support Vector Machines. (Cited on pages 25, 26, 27, 180, and 187)

[78] Evgeniou, T., Pontil, M., and Poggio, T. 10 Regularization Networks and Support Vector Machines. *Advances in Large-Margin Classifiers* (2000). (Cited on page 26)

[79] Falck, T., Signoretto, M., Suykens, J. A. K., and De Moor, B. A two stage algorithm for kernel based partially linear modeling with orthogonality constraints. *Internal Report 10-03, ESAT-SISTA, K.U.Leuven* (2010). (Cited on pages 11, 13, 173, and 181)

[80] Favier, G., and Kibangou, A. Tensor-based methods for system identification. *International Journal of Sciences and Techniques of Authomatic Control Computer Engineering (IJ-STA) 3*, 1 (2009), 870–889. (Cited on page 124)

[81] Fazel, M. *Matrix rank minimization with applications*. PhD thesis, Elec. Eng. Dept., Stanford University, 2002. (Cited on page 28)

[82] Ferreol, A., and Chevalier, P. On the behavior of current second and higher order blind source separation methods for cyclostationary sources. *IEEE Trans. on Signal Processing 48*, 6 (2002), 1712–1725. (Cited on pages 126 and 129)

[83] Fonollosa, J. A. R. Sample cumulants of stationary processes: asymptotic results. *IEEE Trans. on Signal Processing 43*, 4 (2002), 967–977. (Cited on pages 126 and 127)

[84] Foster, D., Nascimento, S., and Amano, K. Information limits on neural identification of colored surfaces in natural scenes. *Visual Neuroscience 21*, 03 (2004), 331–336. (Cited on page 78)

[85] Frank, A., and Asuncion, A. UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences, http://archive.ics.uci.edu/ml, 2010. (Cited on page 115)

[86] Gandy, S., Recht, B., and Yamada, I. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems 27(2) ,19 pp., doi:10.1088/0266-5611/27/2/025010* (2011). (Cited on pages 6, 7, 47, 48, and 61)

[87] Gärtner, T. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter 5*, 1 (2003), 49–58. (Cited on pages 37, 98, and 123)

[88] Gärtner, T. *Kernels for Structured Data*, vol. 72 of *In Machine Perception and Artificial Intelligence*. World Scientific, 2008. (Cited on pages 37 and 98)

[89] Ghahramani, Z., and Hinton, G. Parameter estimation for linear dynamical systems. *University of Toronto, Technical Report CRG-TR-96-2* (1996), 6 pages. (Cited on page 123)

[90] Giannakis, G. B., Inouye, Y., and Mendel, J. M. Cumulant based identification of multichannel moving-average models. *IEEE Trans. on Automatic Control 34*, 7 (2002), 783–787. (Cited on page 124)

[91] Girsanov, I., Poljak, B., and Louvish, D. *Lectures on mathematical theory of extremum problems*. Springer Berlin-Heidelberg-New York, 1972. (Cited on page 109)

[92] Goldberg, A., Xiaojin, Z., Recht, B., Xu, J., and Nowak, R. Transduction with matrix completion: Three birds with one stone. In *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds. 2010, pp. 757–765. (Cited on pages 8, 55, 59, 67, and 68)

[93] Golub, G., and Van Loan, C. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis 17*, 6 (1980), 883–893. (Cited on page 56)

[94] Golub, G. H., and Van Loan, C. F. *Matrix Computations*, third ed. Johns Hopkins University Press, 1996. (Cited on pages 65, 90, 111, and 132)

[95] Greub, W. Multilinear algebra. *Die Grundlehren der mathematischen Wissenschaften, Berlin: Springer, 1967 1* (1967). (Cited on page 32)

[96] Gu, C. *Smoothing spline ANOVA models*. Springer Verlag, 2002. (Cited on pages 173 and 183)

[97] Gu, C., and Wahba, G. Semiparametric Analysis of Variance with Tensor Product Thin Plate Splines. *Journal of the Royal Statistical Society. Series B (Methodological) 55*, 2 (1993), 353–368. (Cited on page 183)

[98] Hadamard, J. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin 13* (1902), 49–52. (Cited on page 24)

[99] Halmos, P. *A Hilbert space problem book*, vol. 19. Springer, 1982. (Cited on page 57)

[100] Hamm, J., and Lee, D. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th international conference on Machine learning* (2008), ACM, pp. 376–383. (Cited on page 101)

[101] Hannan, E. J. *Multiple time series*. John Wiley & Sons, Inc. New York, NY, USA, 1971. (Cited on pages 124 and 126)

[102] Hardoon, D., and Shawe-Taylor, J. Decomposing the tensor kernel support vector machine for neuroscience data with structured labels. *Machine Learning 79*, 1, 1–18. (Cited on page 6)

[103] Harshman, R. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis. *UCLA working papers in phonetics 16*, 1 (1970), 84. (Cited on page 32)

[104] Hastad, J. Tensor rank is NP-complete. *Journal of Algorithms 11*, 4 (1990), 644–654. (Cited on pages 51 and 145)

[105] He, X., Cai, D., and Niyogi, P. Tensor subspace analysis. In *Advances in Neural Information Processing Systems (NIPS), 2006*, pp. 499–506. (Cited on pages 77, 95, and 98)

[106] Herbrich, R. *Learning kernel classifiers*. Mit Press, 2002. (Cited on page 15)

[107] Hillar, C., and Lim, L. Most tensor problems are NP hard. *Arxiv preprint arXiv:0911.1393* (2010). (Cited on pages 51 and 145)

[108] Hitchcock, F. Multiple invariants and generalized rank of a p-way matrix or tensor. *J. Math. Phys 7*, 1 (1927), 39–79. (Cited on page 32)

[109] Hoerl, A. E., and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 12*, 1 (1970), 55–67. (Cited on pages 25 and 43)

[110] Hofmann, T., Schölkopf, B., and Smola, A. Kernel methods in machine learning. *The annals of statistics 36*, 3 (2008), 1171–1220. (Cited on pages 2 and 3)

[111] Horn, R., and Johnson, C. *Topics in Matrix Analysis*. Cambridge University Press, 1994. (Cited on page 50)

[112] Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks 2*, 5 (1989), 359–366. (Cited on page 29)

[113] Huang, J. Functional ANOVA models for generalized regression. *J. Multivariate Analysis 67* (1998), 49–71. (Cited on page 183)

[114] Hunyadi, B., De Vos, M., Signoretto, M., Suykens, J. A. K., VanPaesschen, W., and Van Huffel, S. Automatic seizure detection incorporating structural information. In *ICANN 2011, LNCS 6791* (2011), pp. 233—240. (Cited on pages 8 and 146)

[115] Ito, K., and Kunisch, K. *Lagrange multiplier approach to variational problems and applications*. Advances in Design and Control. SIAM, 2008. (Cited on page 109)

[116] Jaakkola, T., and Haussler, D. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems* (1999), 487–493. (Cited on page 123)

[117] Jacob, L., Obozinski, G., and Vert, J. Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning* (2009), ACM New York, NY, USA. (Cited on page 43)

[118] Jenatton, R., Audibert, J., and Bach, F. Structured variable selection with sparsity-inducing norms. *Technical Report - http://arxiv.org/abs/0904.3523v1* (2009). (Cited on pages 3, 12, and 27)

[119] Joachims, T. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning (ICML)* (1999), pp. 200–209. (Cited on page 18)

[120] Kadison, R. V., and Ringrose, J. R. *Fundamentals of the theory of operator algebras*, vol. 15 of *Graduate Studies in Mathematics*. American Mathematical Society, 1983. (Cited on pages 149, 173, and 184)

[121] Kailath, T., Sayed, A. H., and Hassibi, B. *Linear estimation*. Prentice Hall Upper Saddle River, NJ, USA, 2000. (Cited on page 126)

[122] Kaufman, A., and Xiao, S. High CO2 levels in the Proterozoic atmosphere estimated from analyses of individual microfossils. *Nature 425*, 6955 (2003), 279–282. (Cited on page 76)

[123] Kelly, S. P., Lalor, E., Reilly, R. B., and Foxe, J. J. Independent brain computer interface control using visual spatial attention-dependent modulations of parieto-occipital alpha. In *Conference Proceedings of 2−nd International IEEE EMBS Conference on Neural Engineering* (2005), pp. 667–670. (Cited on page 137)

[124] Kimeldorf, G., and Wahba, G. Some results on Tchebycheffian spline functions. *J. Math. Anal. Applic. 33* (1971), 82–95. (Cited on page 25)

[125] Kolda, T., and Bader, B. Tensor decompositions and applications. *SIAM review 51*, 3 (2009), 455–500. (Cited on page 2)

[126] Kolmogorov, A., and Fomin, S. *Elements of the theory of functions and functional analysis*. Dover Publications, 1999. (Cited on page 149)

[127] Kroonenberg, P. *Applied multiway data analysis*. Wiley-Interscience, 2008. (Cited on page 2)

[128] Lanckriet, G., De Bie, T., Cristianini, N., Jordan, M., and Noble, W. A statistical framework for genomic data fusion. *Bioinformatics 20*, 16 (2004), 2626–2635. (Cited on page 27)

[129] Landsberg, J. M. *The geometry of tensors with applications*. Book in preparation. (Cited on pages 5 and 144)

[130] Li, C., and Li, H. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics 24*, 9 (2008), 1175–1182. (Cited on page 12)

[131] Liang, J., and Ding, Z. Blind MIMO system identification based on cumulant subspace decomposition. *IEEE Trans. on Signal Processing 51*, 6 (2003), 1457–1468. (Cited on page 127)

[132] Lin, C. Formulations of support vector machines: a note from an optimization point of view. *Neural Computation 13*, 2 (2001), 307–317. (Cited on page 109)

[133] Lin, Y. Tensor Product Space ANOVA Models. *The Annals of Statistics 28*, 3 (2000), 734–755. (Cited on page 183)

[134] Lions, P., and Mercier, B. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis 16*, 6 (1979), 964–979. (Cited on pages 45 and 47)

[135] Liu, J., Musialski, P., Wonka, P., and Ye, J. Tensor completion for estimating missing values in visual data. In *IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan* (2009), pp. 2114–2121. (Cited on pages 6, 8, 51, 55, 61, and 145)

[136] Luenberger, D. *Optimization by Vector Space Methods*. John Wiley and Sons, Inc., New York, 1969. (Cited on pages 42, 91, 109, 149, 182, 189, 190, and 191)

[137] Ma, S., Goldfarb, D., and Chen, L. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming 128*, 1 (2011), 321–353. (Cited on page 67)

[138] Marquardt, D. Generalized Inverses, Ridge Regression, Biased Linear Estimation, and Nonlinear Estimation. *Technometrics 12*, 3 (1970), 591–612. (Cited on page 25)

[139] McFee, B., and Lanckriet, G. Learning multi-modal similarity. *The Journal of Machine Learning Research 12* (2011), 491–523. (Cited on page 30)

[140] Meinshausen, N., and Buhlmann, P. High-dimensional graphs and variable selection with the Lasso. *Annals of statistics 34*, 3 (2006), 1436. (Cited on page 187)

[141] Mercer, J. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London A 209* (1909), 415–446. (Cited on pages 173 and 180)

[142] Micchelli, C., and Pontil, M. Learning the Kernel Function via Regularization. *Journal of Machine Learning Research 6* (2005), 1099–1125. (Cited on pages 27 and 187)

[143] Micchelli, C., and Pontil, M. Feature space perspectives for learning the kernel. *Machine Learning 66*, 2 (2007), 297–319. (Cited on pages 27 and 187)

[144] Minsky, M., and Seymour, P. Perceptrons. (Cited on page 3)

[145] Minty, G. Monotone (nonlinear) operators in Hilbert space. *Duke Mathematical Journal 29*, 3 (1962), 341–346. (Cited on page 46)

[146] Moreau, J. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Sér. A Math 255* (1962), 2897–2899. (Cited on page 46)

[147] Mukherjee, S., Wu, Q., and Zhou, D. X. Learning gradients on manifolds. *Bernoulli 16*, 1 (2010), 181–207. (Cited on page 5)

[148] Negahban, S., Ravikumar, P., Wainwright, M., and Yu, B. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems (NIPS), 2009* (2009), pp. 1348–1356. (Cited on pages 3 and 11)

[149] Nesterov, Y. *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Pub, 2003. (Cited on pages 48 and 52)

[150] Nesterov, Y. Gradient methods for minimizing composite objective function. *Center for Operations Research and Econometrics (CORE), Université catholique de Louvain, Tech. Rep.* (2007). (Cited on page 49)

[151] Nikias, C. L., and Mendel, J. M. Signal processing with higher-order spectra. *Signal Processing Magazine, IEEE 10*, 3 (2002), 10–37. (Cited on page 124)

[152] Nikias, C. L., and Petropulu, A. P. *Higher-order spectra analysis: a nonlinear signal processing framework*. Prentice Hall Upper Saddle River, NJ, USA, 1993. (Cited on pages 124 and 127)

[153] Ojeda, F., Signoretto, M., Van de Plas, R., Waelkens, E., De Moor, B., and Suykens, J. A. K. Semi-supervised learning of sparse linear models in mass spectral imaging. In *Pattern Recognition in Bioinformatics (PRIB), vol. 6282 of Lecture Notes in Bioinformatics.* (2010), pp. 325–334. (Cited on pages 12 and 77)

[154] Paninski, L., Ahmadian, Y., Ferreira, D., Koyama, S., Rahnama Rad, K., Vidne, M., Vogelstein, J., and Wu, W. A new look at state-space models for neural data. *Journal of computational neuroscience 29*, 1 (2010), 107–126. (Cited on page 132)

[155] Papy, J., De Lathauwer, L., and Van Huffel, S. Exponential data fitting using multilinear algebra: the single-channel and multi-channel case. *Numerical linear algebra with applications 12*, 8 (2005), 809–826. (Cited on pages 106 and 107)

[156] Papy, J., De Lathauwer, L., and Van Huffel, S. Exponential data fitting using multilinear algebra: the decimative case. *J. Chemometrics 23*, 7-8 (2009), 341–351. (Cited on pages 107 and 116)

[157] Pelckmans, K., De Brabanter, J., Suykens, J. A. K., and De Moor, B. Handling missing values in support vector machine classifiers. *Neural Networks 18* (2005), 684–692. (Cited on page 68)

[158] Phelps, R. *Convex functions, monotone operators, and differentiability.* Springer, 1993. (Cited on page 46)

[159] Poggio, T., and Girosi, F. Regularization algorithms for learning that are equivalent to multilayer networks. *Science 247* (1990), 978–982. (Cited on page 187)

[160] Poggio, T., and Smale, S. The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society (AMS) 50*, 5 (2003), 537–544. (Cited on page 29)

[161] Priestley, M. B. *Spectral analysis and time series*. Academic Press, 1981. (Cited on page 124)

[162] Recht, B., Fazel, M., and Parrilo, P. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev. 52* (2007), 471–501. (Cited on page 51)

[163] Regalia, P. A., and Mitra, S. K. Kronecker products, unitary matrices and signal processing applications. *SIAM review 31*, 4 (1989), 586–613. (Cited on page 132)

[164] Riesz, F., and Sz.-Nagy, B. *Functional Analysis*. Frederick Ungar Publishing Co., New York, 1955. (Cited on page 149)

[165] Rockafellar, R. *Convex analysis*. Princeton university press, 1970. (Cited on page 45)

[166] Rockafellar, R. On the maximal monotonicity of subdifferential mappings. *Pacific J. Math 33*, 1 (1970), 209–216. (Cited on page 46)

[167] Rockafellar, R. *Conjugate duality and optimization*, vol. 16. Society for Industrial Mathematics, 1974. (Cited on pages 189 and 190)

[168] Rockafellar, R. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization 14* (1976), 877. (Cited on page 45)

[169] Rosasco, L., Santoro, M., Mosci, S., Villa, S., and Verri, A. Regularization approach to nonlinear variable selection. In *Int. Conf. Artif. Intell. and Stat. (AISTATS)* (2010), vol. 9, pp. 653—660. (Cited on page 3)

[170] Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review 65*, 6 (1958), 386–408. (Cited on page 18)

[171] Roweis, S., and Ghahramani, Z. A unifying review of linear gaussian models. *Neural computation 11*, 2 (1999), 305–345. (Cited on page 123)

[172] Rudin, W. Real and complex analysis, 1987. (Cited on page 161)

[173] Rudin, W. Functional analysis. international series in pure and applied mathematics, 1991. (Cited on page 149)

[174] Ryan, R. *Introduction to tensor Product of Banach Spaces*. Springer-Verlag New York, LLC, 2002. (Cited on page 149)

[175] Saitoh, S. *Theory of Reproducing Kernels and its Applications*. Longman Scientific & Technical, Harlow, England, 1988. (Cited on page 173)

[176] Schölkopf, B., Herbrich, R., and Smola, A. J. A generalized representer theorem. *Proceedings of the Annual Conference on Computational Learning Theory (COLT)* (2001), 416–426. (Cited on pages 25 and 26)

[177] Schölkopf, B., and Smola, A. J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, 2002. (Cited on pages 2, 99, 173, and 187)

[178] Shawe-Taylor, J., and Cristianini, N. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. (Cited on page 2)

[179] Signoretto, M., Daemen, A., Savorgnan, C., and Suykens, J. Variable Selection and Grouping with Multiple Graph Priors. In *NIPS Workshop on Optimization for Machine Learning* (2009). (Cited on page 12)

[180] Signoretto, M., De Lathauwer, L., and Suykens, J. A. K. Convex multilinear estimation and operatorial representations, ESAT-SISTA, K.U.Leuven (Leuven, Belgium). In *NIPS Workshop on Tensors, Kernels, and Machine Learning.* (2010). (Cited on page 9)

[181] Signoretto, M., De Lathauwer, L., and Suykens, J. A. K. Kernel-based learning from infinite dimensional 2-way tensors. In *ICANN 2010, Part II, LNCS 6353* (2010), pp. 59—69. (Cited on pages 4, 10, 89, 90, and 96)

[182] Signoretto, M., De Lathauwer, L., and Suykens, J. A. K. Nuclear Norms for Tensors and Their Use for Convex Multilinear Estimation. *Internal Report 10-186, ESAT-SISTA, K.U.Leuven (Leuven, Belgium)* (2010). (Cited on pages 6, 7, and 145)

[183] Signoretto, M., De Lathauwer, L., and Suykens, J. A. K. A kernel-based framework to tensorial data analysis. *Neural Networks 24*, 8 (2011), 861–874. (Cited on pages 10, 102, and 133)

[184] Signoretto, M., Olivetti, E., De Lathauwer, L., and Suykens, J. A. K. Classification of multichannel signals with cumulant-based kernels. *Internal Report 10-251, ESAT-SISTA, K.U. Leuven (Leuven, Belgium)* (2010). (Cited on pages 10, 107, 123, 124, 126, 133, and 135)

[185] Signoretto, M., Pelckmans, K., De Lathauwer, L., and Suykens, J. Data-dependent Norm Adaptation for Sparse Recovery in Kernel Ensembles Learning. *Technical Report 09-97 ESAT SCD/SISTA* (2009). (Cited on pages 10 and 11)

[186] Signoretto, M., Pelckmans, K., De Lathauwer, L., and Suykens, J. A. K. Improved non-parametric sparse recovery with data matched penalties. In *The 2nd International Workshop on Cognitive Information Processing (CIP)* (2010). (Cited on pages 10 and 11)

[187] Signoretto, M., Pelckmans, K., and Suykens, J. A. K. Quadratically constrained quadratic programming for subspace selection in kernel regression estimation. In *ICANN 2008, Part I, LNCS 5163* (2008), pp. 175—184. (Cited on page 11)

[188] Signoretto, M., Tran Dinh, Q., De Lathauwer, L., and Suykens, J. Learning with tensors: a framework based on convex optimization and spectral regularization. *Internal Report 11-129, ESAT-SISTA, K.U.Leuven (Leuven, Belgium)* (2011), 43. (Cited on pages 8, 48, 60, 62, and 66)

[189] Signoretto, M., Van de Plas, R., De Moor, B., and Suykens, J. A. K. Tensor versus matrix completion: a comparison with application to spectral data. *IEEE Signal Processing Letters 18*, 7 (2011), 403–406. (Cited on pages 6, 8, 56, and 61)

[190] Sindhwani, V., Niyogi, P., and Belkin, M. Beyond the point cloud: from transductive to semi-supervised learning. In *International Conference on Machine Learning (ICML)* (2005). (Cited on pages 10 and 17)

[191] Smale, S., and Zhou, D. Shannon sampling II: Connections to learning theory. *Applied and Computational Harmonic Analysis 19*, 3 (2005), 285–302. (Cited on page 57)

[192] Smola, A. J. *Learning with Kernels*. PhD thesis, Technische Universität Berlin, 1998. (Cited on page 93)

[193] Spingarn, J. Partial inverse of a monotone operator. *Applied mathematics and optimization 10*, 1 (1983), 247–265. (Cited on page 45)

[194] Steinwart, I. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research 2* (2002), 67–93. (Cited on page 114)

[195] Steinwart, I., and Christmann, A. *Support vector machines*. Springer Verlag, 2008. (Cited on page 108)

[196] Steinwart, I., Hush, D., and Scovel, C. An explicit description of the reproducing kernel hilbert spaces of gaussian rbf kernels. *IEEE Trans. Inform. Theory 52* (2006), 4635–4643. (Cited on page 29)

[197] Sturm, J. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software 11*, 1 (1999), 625–653. (Cited on page 45)

[198] Suykens, J. A. K., Alzate, C., and Pelckmans, K. Primal and dual model representations in kernel-based learning. *Statistics Surveys 4* (2010), 148–183 (electronic). DOI: 10.1214/09–SS052. (Cited on page 108)

[199] Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J. *Least squares support vector machines*. World Scientific, 2002. (Cited on pages 26, 97, and 108)

[200] Suykens, J. A. K., and Vandewalle, J. Least squares support vector machine classifiers. *Neural Processing Letters 9*, 3 (1999), 293–300. (Cited on pages 68, 108, and 189)

[201] Swami, A., Giannakis, G., and Shamsunder, S. Multichannel ARMA processes. *IEEE Trans. on Signal Processing 42*, 4 (2002), 898–913. (Cited on pages 126 and 127)

[202] Swami, A., and Mendel, J. M. Time and lag recursive computation of cumulants from a state-space model. *IEEE Trans. on Automatic Control 35*, 1 (1990), 4–17. (Cited on page 127)

[203] Taban, I. M., Altelaar, A. F., van der Burgt, Y. E. M., McDonnell, L. A., Heeren, R., Fuchser, J., and Baykut, G. Imaging of peptides in the rat brain using MALDI-FTICR mass spectrometry. *Journal of the American Society for Mass Spectrometry 18*, 1 (2007), 145–151. (Cited on page 76)

[204] Tao, D., Li, X., Wu, X., Hu, W., and Maybank, S. Supervised tensor learning. *Knowledge and Information Systems 13*, 1 (2007), 1–42. (Cited on pages 6, 113, 116, 117, and 118)

[205] Thrun, S. *Learning to learn*. Springer, 1998, ch. Lifelong Learning Algorithms. (Cited on page 27)

[206] Tibshirani, R. Regression Shrinkage and Selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological) 58*, 1 (1996), 267–288. (Cited on pages 27 and 187)

[207] Tibshirani, R. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological) 58*, 1 (1996), 267–288. (Cited on page 51)

[208] Tikhonov, A. On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR* (1943), vol. 39, pp. 195–198. (Cited on page 24)

[209] Tikhonov, A. Solution of incorrectly formulated problems and the regularization method. In *Soviet Math. Dokl.* (1963), vol. 5, p. 1035. (Cited on page 24)

[210] Tikhonov, A. N., and Arsenin, V. Y. *Solutions of Ill-posed Problems*. W. H. Winston, Washington, D.C., 1977. (Cited on page 24)

[211] Tomioka, R., and Aihara, K. Classifying matrices with a spectral regularization. In *Proceedings of the 24th international conference on Machine learning* (2007), ACM, pp. 895–902. (Cited on page 65)

[212] Tomioka, R., Hayashi, K., and Kashima, H. On the extension of trace norm to tensors. *NIPS Workshop on Tensors, Kernels, and Machine Learning.* (2010). (Cited on pages 7 and 76)

[213] Tomioka, R., Hayashi, K., and Kashima, H. Estimation of low-rank tensors via convex optimization. *Arxiv preprint arXiv:1010.0789* (2011). (Cited on page 61)

[214] Tomioka, R., Suzuki, T., Hayashi, K., and Kashima, H. Statistical performance of convex tensor decomposition. In *Advances in Neural Information Processing Systems (NIPS)* (2011), p. to appear. (Cited on pages 143 and 144)

[215] Tsuda, K., Akaho, S., Kawanabe, M., and Müller, K. Asymptotic properties of the fisher kernel. *Neural Computation 16*, 1 (2004), 115–137. (Cited on page 123)

[216] Tucker, L. R. *The extension of factor analysis to three-dimensional matrices*, vol. Contributions to Mathematical Psychology, Holt, Rinehart Winston, NY. 1964, pp. 109–127. (Cited on page 34)

[217] Tucker, L. R. Some mathematical notes on three-mode factor analysis. *Psychometrika 31*, 3 (1966), 279–311. (Cited on page 34)

[218] Turaga, P., Veeraraghavan, A., Srivastava, A., and Chellappa, R. Statistical analysis on manifolds and its applications to video analysis. *Video Search and Mining* (2010), 115–144. (Cited on page 99)

[219] Turaga, P., Veeraraghavan, A., Srivastava, A., and Chellappa, R. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 33*, 11 (2011), 2273 – 2286. (Cited on pages 5 and 99)

[220] Turk, M., and Pentland, A. Eigenfaces for recognition. *Journal of cognitive neuroscience 3*, 1 (1991), 71–86. (Cited on page 76)

[221] Tütüncü, R., Toh, K., and Todd, M. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming 95*, 2 (2003), 189–217. (Cited on page 45)

[222] Van de Plas, R., Pelckmans, K., De Moor, B., and Waelkens, E. Spatial querying of imaging mass spectrometry data: a nonnegative least squares approach. In *Neural Information Processing Systems Workshop on Machine Learning in Computational Biology* (2007). (Cited on page 77)

[223] Van Gerven, M., and Jensen, O. Attention modulations of posterior alpha as a control signal for two-dimensional brain-computer interfaces. *Journal of Neuroscience Methods 179*, 1 (April 2009), 78–84. (Cited on page 137)

[224] Van Huffel, S., and Vandewalle, J. *The total least squares problem: computational aspects and analysis*, vol. 9. Society for Industrial Mathematics, 1991. (Cited on page 56)

[225] Vandenberghe, L., and Boyd, S. Semidefinite programming. *SIAM review 38*, 1 (1996), 49–95. (Cited on page 45)

[226] Vapnik, V. *Estimation of Dependences Based on Empirical Data [in Russian]*. Nauka, Moscow, 1979. (English translation: Springer Verlag, New York, 1982). (Cited on pages 18 and 19)

[227] Vapnik, V. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995. (Cited on pages 15, 16, 20, 21, 22, and 23)

[228] Vapnik, V., and Chervonenkis, A. Uniform convergence of frequencies of occurrence of events to their probabilities. *Doklady Akademii Nauk SSSR 181* (1968), 915—-918. (Cited on pages 20 and 22)

[229] Vapnik, V., and Chervonenkis, A. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications 16*, 2 (1971), 264—-280. (Cited on pages 20 and 22)

[230] Vapnik, V., and Chervonenkis, A. The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis 1*, 3 (1991), 283–305. (Cited on page 20)

[231] Vilenkin, N. *Special functions and the theory of group representations*. American Mathematical Society, 1968. (Cited on pages 149 and 169)

[232] Vishwanathan, S., Schraudolph, N., Kondor, R., and Borgwardt, K. Graph Kernels. *Journal of Machine Learning Research 11* (2010), 1201–1242. (Cited on page 5)

[233] Vishwanathan, S. V. N., Smola, A. J., and Vidal, R. Binet-Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes. *International Journal of Computer Vision 73*, 1 (2007), 95–119. (Cited on page 5)

[234] von Neumann, J. Some matrix inequalities and metrization of matric-space. *Tomsk University Review 1* (1937), 286–300. (Cited on page 50)

[235] Wahba, G. *Spline Models for Observational Data*, vol. 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1990. (Cited on pages 25, 173, 183, and 187)

[236] Weston, J., and Watkins, C. Multi-class support vector machines. Tech. Rep. CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998. (Cited on page 26)

[237] Yuan, M., and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B 68*, 1 (2006), 49–67. (Cited on pages 3, 12, 27, and 187)

[238] Zhao, P., Rocha, G., and Yu, B. Grouped and hierarchical model selection through composite absolute penalties. *Annals of Statistics* (2008). (Cited on page 6)

[239] Zhao, P., Rocha, G., and Yu, B. The composite absolute penalties family for grouped and hierarchical variable selection. *Annals of Statistics 37* (2009), 3468–3497. (Cited on pages 3, 12, 27, 43, and 187)

[240] Zhao, P., and Yu, B. On Model Selection Consistency of Lasso. *Journal of Machine Learning Research 7* (2006), 2541–2563. (Cited on page 187)

[241] Zou, H. The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association 101*, 476 (2006), 1418–1429. (Cited on page 187)

[242] Zou, H., and Hastie, T. Regularization and variable selection via the elastic net. *J. Roy. Stat. Soc. Ser. B 67*, 2 (2005), 301–320. (Cited on pages 11 and 12)

Publications by the Author

Journal Papers

- A. Daemen, M. Signoretto, O. Gevaert, J. A. K. Suykens and B. De Moor. Improved microarray-based decision support with graph encoded interactome data. *PLoS ONE*, 5(4):1—16, 2010.
- M. Signoretto, E. Olivetti, L. De Lathauwer, and J. A. K. Suykens. Classification of multichannel signals with cumulant-based kernels. *Submitted*, 10 pages, 2010.
- M. Signoretto, K. Pelckmans, L. De Lathauwer, and J. A. K. Suykens. Data-dependent Norm Adaptation for Sparse Recovery in Kernel Ensembles Learning. *Submitted*, 32 pages, 2010.
- M. Signoretto, L. De Lathauwer, and J. A. K. Suykens. A kernel-based framework to tensorial data analysis. *Neural Networks*, 24(8):861—874, 2011.
- M. Signoretto, R. Van de Plas, B. De Moor, and J. A. K. Suykens. Tensor versus matrix completion: a comparison with application to spectral data. *IEEE Signal Processing Letters*, 18(7):403—406, 2011.
- M. Signoretto, Q. Tran Dinh, L. De Lathauwer, and J.A.K. Suykens. Learning with tensors: a framework based on convex optimization and spectral regularization. *Submitted*, 43 pages, 2011.
- T. Falck, M. Signoretto, J. A. K. Suykens, B. De Moor. A two stage algorithm for kernel based partially linear modeling with orthogonality constraints. *Submitted*, 15 pages, 2011.
- B. Hunyadi, M. De Vos, M. Signoretto, W. Van Paesschen, J. A. K. Suykens and S. Van Huffel. Incorporating structural information from the

multichannel EEG improves patient-specific seizure detection. *Submitted*, 2011.

Conference Papers

- M.Signoretto, K. Pelckmans and J. A. K. Suykens, Quadratically constrained quadratic programming for subspace selection in kernel regression estimation. In *ICANN 2008, Part I, LNCS 5163*, 175—184, 2008.
- M. Signoretto, A. Daemen, C. Savorgnan and J. A. K. Suykens, Variable selection and grouping with multiple graph priors. In *NIPS Workshop on Optimization for Machine Learning*, 2009.
- M. Signoretto, K. Pelckmans, L. De Lathauwer and J. A. K. Suykens, Improved non-parametric sparse recovery with data matched penalties. In *The 2nd International Workshop on Cognitive Information Processing (CIP)*, 2010.
- F. Ojeda, M. Signoretto, R. Van de Plas, E. Waelkens, B. De Moor and J. A. K. Suykens, Semi-supervised Learning of Sparse Linear Models in Mass Spectral Imaging. In *Pattern Recognition in Bioinformatics (PRIB), vol. 6282 of Lecture Notes in Bioinformatics*, 325—334, 2010.
- M. Signoretto, L. De Lathauwer, and J. A. K. Suykens. Kernel-based learning from infinite dimensional 2-way tensors. In *ICANN 2010, Part II, LNCS 6353*, 59—69, 2010.
- M. Signoretto, L. De Lathauwer, and J.A.K. Suykens. Convex multilinear estimation and operatorial representations. In *NIPS Workshop on Tensors, Kernels, and Machine Learning*, 6 pages, 2010.
- B. Hunyadi, M. De Vos, M. Signoretto, J. A. K. Suykens, W. VanPaesschen and S. Van Huffel. Automatic Seizure Detection Incorporating Structural Information. *ICANN 2010, Part II, LNCS 6791*,233—240, 2011.
- M. Signoretto and J. A. K. Suykens. Convex Estimation of Cointegrated VAR Models by a Nuclear Norm Penalty, *Submitted*, 6 pages, 2011.

Abstracts

- M. Signoretto, K. Pelckmans and J. A. K. Suykens, Convex-concave problem for component selection in structured hypothesis spaces. In *ICNPAA, Special Session on Learning from Data*, Genoa (Italy), 2008.
- M. Signoretto and J. A. K. Suykens, Convex-concave selection of functional components. In *ICML Workshop on Sparse Optimization and Variable Selection*, Helsinki (Finland), 2008.
- M. Signoretto, K. Pelckmans and J. A. K. Suykens, Selection in functional ANOVA models with non-uniform data. In *Workshop on Sparsity in Machine Learning and Statistics*, Cumberland Lodge (UK), 2009.
- M. Signoretto, K. Pelckmans and J. A. K. Suykens, Optimization for feature selection with kernel models, In *Optec Workpackage Meeting*, Leuven (Belgium), 2009.
- M. Signoretto and J. A. K. Suykens, Selection and grouping with multiple graphs, In *29th Benelux Meeting on Systems and Control*, Heeze (The Netherlands), 2010.
- G. Gins, M. Signoretto, J. A. K. Suykens and J. Van Impe, Comparing Partial Least Squares with Nuclear Norm- Based Optimization for the prediction of batch-end quality, In *30th Benelux Meeting on Systems and Control*, Lommel (Belgium), 2011.

Organized Events

The Author co-organized (together with Andreas Argyriou, David F. Gleich, Tamara G. Kolda, Vicente Malave and Johan A. K. Suykens) the NIPS Workshop on Tensors, Kernels, and Machine Learning held in Whistler, BC (Canada) on December 10th 2010.

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Electrical Engineering
Systems, Models and Control
Kasteelpark Arenberg 10
B-3001 Heverlee

KATHOLIEKE UNIVERSITEIT LEUVEN

ASSOCIATIE K.U.LEUVEN