A Kernel-based Framework to Tensorial Data Analysis

Marco Signoretto[a], Lieven De Lathauwer[b], Johan A. K. Suykens[a]

[a]*Katholieke Universiteit Leuven, ESAT-SCD/SISTA Kasteelpark Arenberg 10, B-3001 Leuven (BELGIUM)*
{marco.signoretto,johan.suykens}@esat.kuleuven.be
[b]*Group Science, Engineering and Technology Katholieke Universiteit Leuven, Campus Kortrijk E. Sabbelaan 53, 8500 Kortrijk (BELGIUM)*
lieven.delathauwer@kuleuven-kortrijk.be

Abstract

Tensor-based techniques for learning allow one to exploit the structure of carefully chosen representations of data. This is a desirable feature in particular when the number of training patterns is small which is often the case in areas such as biosignal processing and chemometrics. However, the class of tensor based models is somewhat restricted and might suffer from limited discriminative power. On a different track, kernel methods lead to flexible nonlinear models that have been proven successful in many different contexts. Nonetheless, a naïve application of kernel methods does not exploit structural properties possessed by the given tensorial representations. The goal of this work is to go beyond this limitation by introducing non-parametric tensor based models. The proposed framework aims at improving the discriminative power of supervised tensor-based models while still exploiting the structural information embodied in the data. We begin by introducing a feature space formed by multilinear functionals. The latter can be considered as the infinite dimensional analogue of tensors. Successively we show how to implicitly map input patterns in such a feature space by means of kernels that exploit the algebraic structure of data tensors. The proposed tensorial kernel links to the MLSVD and features an interesting invariance property; the approach leads to convex optimization and fits into the same primal-dual framework underlying SVM-like algorithms.

Keywords: multilinear algebra, reproducing kernel Hilbert spaces, tensorial kernels, subspace angles

1. Introduction

Tensors [30] are higher order arrays that generalize the notions of *vectors* (first-order tensors) and *matrices* (second-order tensors). The use of these data structures has been advocated in virtue of certain favorable properties. Additionally, tensor representations naturally result from the experiments performed in a number of domains, see Table 1 for some examples.

An alternative representation prescribes to *flatten* the different dimensions namely to represent the data as high dimensional vectors. In this way, however, important structure might be lost. Exploiting a natural 2−way representation, for example, retains the relationship between the row space and the column space and allows one to find structure preserving projections more efficiently [23]. Still, a main drawback of tensor-based learning is that it allows the user to construct models which are affine in the data (in a sense that we clarify later) and hence fail in the presence of nonlinearities. On a different track kernel methods [40],[48] lead

Table 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:	fluoresce excitation-emission data (samples × emission × excitation)

NOTICE: this is the authors version of a work that was accepted for publication in Neural Network Journal. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Neural Networks, Volume 24(8), October 2011, Pages 861-874; doi:10.1016/j.neunet.2011.05.011.

to flexible models that have been proven successful in many different contexts. The core idea in this case consists of mapping input points represented as vectors $\{X^1, \ldots, X^M\} \subset \mathbb{R}^I$ into a high dimensional inner-product space $(\mathfrak{F}, \langle \cdot, \cdot \rangle_{\mathfrak{F}})$ by means of a *feature map* $\phi : \mathbb{R}^I \to \mathfrak{F}$. Since the feature map is normally chosen to be nonlinear, a linear model in the feature space corresponds to a nonlinear rule in $\mathbb{R}^I$. On the other hand, the so-called *kernel trick* allows one to develop computationally feasible approaches regardless of the dimensionality of $\mathfrak{F}$ as soon as we know $k : \mathbb{R}^I \times \mathbb{R}^I \to \mathbb{R}$ satisfying $k(X, Y) = \langle \phi(X), \phi(Y) \rangle_{\mathfrak{F}}$. When input data are $N-$th order arrays, nonetheless, a naïve application of kernel methods amounts to perform flattening first, with a consequent loss of potentially useful structural information.

1.1. Main Contributions

In this paper we elaborate on a possible framework to extend the flexibility of tensor-based models by kernel-based techniques. We make several contributions:

- We give a constructive definition of the (feature) space of infinite dimensional tensors and show the link with finite dimensional tensors that are used in multilinear algebra. The formalism gives rise to product kernels which comprise, as a special case, the popular Gaussian-RBF kernel.
- The Gaussian-RBF kernel and the linear kernel are based on the Euclidean distance. However the latter does not capture the topological structure underlying a number of objects of interests, such as videos. In turn, such objects often admit a very natural tensorial representation. We then introduce a class of structure-preserving product kernels for tensors that fully exploits the tensorial representation. This relies on the assumption that the latter is useful for the learning task of interest.
- We study an invariance property fulfilled by the proposed kernels and introduce the concept of congruence sets. We highlight the relevance of this formalism for pattern recognition and explicitly discuss a class of problems that takes advantage of the new similarity measure.
- We elaborate on the primal-dual framework used in Support Vector Machines (SVMs) and related algorithms and discuss implications of the tensor-like primal representation. As an additional contribution we detail the rigorous derivation of Least-Squares SVM (LS-SVM) for classification based upon results in infinite dimensional optimization.

1.2. Relation with Existing Literature

Tensor-based techniques are mostly based on decompositions that to some extent generalize the matrix SVD [31],[9]. As such, the largest part of the existing approaches relates to unsupervised methods. Recently, machine learning and related communities got interested in tensors and their use for supervised techniques have also been explored [51],[43]. However with the exception of very specialized attempts [22], the existing proposals deal with linear tensor-based models and a systematic approach to the construction of non-parametric tensor-based models is still missing. A first attempt in this direction [42] focused on second order tensors (matrices) and led to non-convex and computationally demanding problem formulations. The proposed ideas can be extended to higher order tensors at the price of an even higher computational complexity. Here we consider tensors of any order and elaborate on a different formalism that leads to convex optimization. The approach fits into the same primal-dual framework underlying SVM-like algorithms while exploiting algebraic properties of tensors in a convenient way.

1.3. Outline

In the next Section we introduce the notation and some basic facts about finite dimensional tensors and spaces of functions admitting a reproducing kernel. In Section 3 we study spaces of infinite dimensional tensors which give rise to product kernels. Successively in Section 4 we introduce a novel family of structure-preserving factor kernels for tensors. Section 5 is dedicated to the study of an invariance property possessed by the new kernels. Special attention is devoted to the case where input data are temporal or spatial signals represented via Hankel tensors. In Section 6 we then discuss estimation of nonparametric tensor-based models in the framework of primal-dual techniques. Successively we validate our finding by presenting experimental results in Section 7. We end the paper by drawing our concluding remarks in Section 8.

2. Notation and Background Material

We denote scalars by lower-case letters $(a, b, c, \ldots)$, vectors as capitals $(A, B, C, \ldots)$ and matrices as boldface capitals $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \ldots)$. We also use lower-case letters i, j in the meaning of indices and with some abuse of notation 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 write a_i to mean the $i-$th entry of a vector

A. Similarly we write a_{ij} to mean the entry with row index i and column index j in a matrix $\boldsymbol{A}$. Finally we will often use gothic letters ($\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \ldots$) to denote general sets or spaces, regardless of their specific nature.

2.1. Basic Facts about Finite Dimensional Tensors

In this paper we deal with input data observations represented as real-valued $N-$th order tensors, which we denote by calligraphic letters ($\mathcal{A}, \mathcal{B}, C, \ldots$). They are higher order generalizations of vectors ($1-$st order tensors) and matrices ($2-$nd order tensors). Scalars can be seen as tensors of order zero. We write $a_{i_1,\ldots,i_N}$ to denote $(\mathcal{A})_{i_1,\ldots,i_N}$. An $N-$th order tensor $\mathcal{A}$ has 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

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

for all values of the indices. In this case we write $\mathcal{A} = U^{(1)} \otimes U^{(2)} \otimes \cdots \otimes U^{(N)}$. The linear span of such elements forms a vector space, denoted by $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$, which is endowed with the inner product

$$\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} a_{i_1 i_2 \cdots i_N} b_{i_1 i_2 \cdots i_N} \tag{2}$$

and with the Hilbert-Frobenius norm $\|\mathcal{A}\|_F := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. The latter is a straightforward extension of the usual Hilbert-Frobenius norm for matrices and of the l_2 norm for vectors, denoted simply by $\|\cdot\|$. In the following we will use $\langle \cdot, \cdot \rangle$ for any $N \geq 1$ and $\|\cdot\|_F$ for any $N > 1$, regardless of the specific tuple $(I_1, I_2, \ldots, I_N)$. Additionally, notice that for rank-1 tensors $a_{i_1 i_2 \ldots i_N} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(N)}_{i_N}$ and $b_{i_1 i_2 \ldots i_N} = v^{(1)}_{i_1} v^{(2)}_{i_2} \cdots v^{(N)}_{i_N}$ it holds that

$$\langle \mathcal{A}, \mathcal{B} \rangle = \left\langle U^{(1)}, V^{(1)} \right\rangle \left\langle U^{(2)}, V^{(2)} \right\rangle \cdots \left\langle U^{(N)}, V^{(N)} \right\rangle . \tag{3}$$

It is often convenient to rearrange the elements of a tensor so that they form a matrix. This operation is referred to as *matricization* or *unfolding*.

Definition 1 ($n-$mode matricization [28]). Assume a $N-$th order tensor $\mathcal{A} \in \mathbb{R}^{I_1} \otimes \cdots \otimes \mathbb{R}^{I_N}$. The $n-$th mode matrix unfolding, denoted as $\mathcal{A}_{\langle n \rangle}$, is the matrix $\mathbb{R}^{I_n} \otimes \mathbb{R}^{J} \ni \mathcal{A}_{\langle n \rangle} : a^{(n)}_{i_n j} := a_{i_1 i_2 \ldots i_N}$ where $J := I_{n+1} I_{n+2} \cdots I_N I_1 I_2 I_3 \cdots I_{n-1}$ and for $R := [n+1\, n+2 \cdots N\, 1\, 2\, 3 \cdots n-1]$ we have: $j = 1 + \sum_{l \in \mathbb{N}_{N-1}} \left[(i_{r_l} - 1) \prod_{\hat{l} \in \mathbb{N}_{l-1}} I_{r_{\hat{l}}} \right]$.

We conclude this quick excursion on tensors by recalling the multilinear singular value decomposition (MLSVD) [53],[54],[15] that shares many properties with the matrix singular value decomposition (SVD). First we introduce $n-$mode products.

Definition 2 (n-mode product [15]). The $n-$mode product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ by a matrix $\boldsymbol{U} \in \mathbb{R}^{J_n} \otimes \mathbb{R}^{I_n}$, denoted by $\mathcal{A} \times_n \boldsymbol{U}$, is a $(I_1 \times I_2 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N)-$tensor with entries $(\mathcal{A} \times_n \boldsymbol{U})_{i_1 i_2 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} := \sum_{i_n \in \mathbb{N}_{I_n}} a_{i_1 i_2 \cdots i_{n-1} i_n i_{n+1} \cdots i_N} u_{j_n i_n}$.

2.2. Multilinear Singular Value Decomposition

Theorem 1 (MLSVD[15])**.** *Any $\mathcal{A} \in \mathbb{R}^{I_1} \otimes \cdots \otimes \mathbb{R}^{I_N}$ can be written as the product*

$$\mathcal{A} = \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \cdots \times_N \boldsymbol{U}^{(N)} \tag{4}$$

in which $\boldsymbol{U}^{(n)} = \left[U^{(n)}_1 U^{(n)}_2 \cdots U^{(n)}_{I_n} \right] \in \mathbb{R}^{I_n} \otimes \mathbb{R}^{I_n}$ is an orthogonal matrix and $\mathcal{S} \in \mathbb{R}^{I_1} \otimes \cdots \otimes \mathbb{R}^{I_N}$ is called core tensor.

Notably, as shown in [15], the core tensor features a number of properties. In practice the matrix $\boldsymbol{U}^{(n)}$ can be directly found from the SVD decomposition of the $n-$th unfolding $\mathcal{A}_{\langle n \rangle} = \boldsymbol{U}^{(n)} \boldsymbol{S}^{(n)} \boldsymbol{V}^{(n)\top}$. The core tensor $\mathcal{S}$ then satisfies: $\mathcal{S} = \mathcal{A} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \times_3 \cdots \times_N \boldsymbol{U}^{(N)\top}$.

2.3. Reproducing Kernel Hilbert Spaces of Functions

An important role in this paper is played by (infinite dimensional) spaces of real-valued functions. We denote such a space by $\mathfrak{H}$ and we will often write $(\mathfrak{H}, \langle \cdot, \cdot \rangle_{\mathfrak{H}})$ to indicate that $\mathfrak{H}$ is endowed with the Hilbert space (HS) structure defined according to some inner product $\langle \cdot, \cdot \rangle_{\mathfrak{H}}$. The theory of reproducing kernel Hilbert spaces (RKHSs) [1],[56] is concerned with HSs of functions defined on an arbitrary abstract set $\mathfrak{X}$. We consider the case where $\mathfrak{X} \subseteq \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ and denote by $\mathcal{X}$ a generic element of $\mathfrak{X}$. We stress at this point that $\mathfrak{X}$ might equally well denote a subset of scalars, vectors, matrices or — more generally — tensors of any order. We recall that a HS $(\mathfrak{H}, \langle \cdot, \cdot \rangle_{\mathfrak{H}})$ of functions $f : \mathfrak{X} \to \mathbb{R}$ is a reproducing kernel Hilbert space (RKHS) if for any $\mathcal{X} \in \mathfrak{X}$ the *evaluation functional* $L_{\mathcal{X}} : f \mapsto f(\mathcal{X})$ is bounded. A function $k : \mathfrak{X} \times \mathfrak{X} \to \mathbb{R}$ is called *reproducing kernel* of $\mathfrak{H}$ if *(i)* $k_{\mathcal{X}} := k(\cdot, \mathcal{X}) \in \mathfrak{H}$ for any $\mathcal{X} \in \mathfrak{X}$ *(ii)* $f(\mathcal{X}) = \langle f, k_{\mathcal{X}} \rangle_{\mathfrak{H}}$ holds for any $\mathcal{X} \in \mathfrak{X}$ and $f \in \mathfrak{H}$. We write $\mathfrak{H}_k$ instead of $\mathfrak{H}$ whenever we want to stress that k acts as a reproducing kernel for $\mathfrak{H}$. Point *(ii)* is the same as saying that $k_{\mathcal{X}}$ is the *Riesz representer* [38] of $L_{\mathcal{X}}$. From points *(i)* and *(ii)* it is clear that $k(\mathcal{X}, \mathcal{Y}) = \langle k_{\mathcal{X}}, k_{\mathcal{Y}} \rangle_{\mathfrak{H}}$ $\forall (\mathcal{X}, \mathcal{Y}) \in \mathfrak{X} \times \mathfrak{X}$. If we now let $\phi(\mathcal{X}) := k(\mathcal{X}, \cdot)$, we can see $\mathfrak{H}$ as an instance of the feature space $\mathfrak{F}$ discussed in the Introduction. Alternative feature space representations can be stated. Recall

that given a countable set $\mathfrak{A}$, the space of $\mathbb{K}-$valued square summable sequences is defined as $l_2^{\mathbb{K}}(\mathfrak{A}) := \left\{(x_i)_{i\in\mathfrak{A}} \text{ s.t. } x_i \in \mathbb{K}\ \forall\ i \in \mathfrak{A} \text{ and } \sum_{i\in\mathfrak{A}} |x_i|^2 < \infty\right\}$.

Theorem 2 ($l_2^{\mathbb{K}}(\mathfrak{A})$ feature space, [4])**.** *A function k defined on* $: \mathfrak{X} \times \mathfrak{X}$ *is a reproducing kernel if and only if there exists* $\mathfrak{A}$ *and* $\phi : \mathfrak{X} \mapsto l_2^{\mathbb{K}}(\mathfrak{A})$ *such that*

$$k(\mathcal{X}, \mathcal{Y}) = \langle \phi(\mathcal{X}), \phi(\mathcal{Y}) \rangle_{l_2^{\mathbb{K}}(\mathfrak{A})} \tag{5}$$

for any $(\mathcal{X}, \mathcal{Y}) \in \mathfrak{X} \times \mathfrak{X}$.

3. Non-parametric Tensor-based Models

We can now turn to the problem of interest, namely the definition of non-parametric tensor-based models. By tensor-based we mean that the input of our model will be a tensor $\mathcal{X}$. We will refer to $\mathcal{X}$ as the *data tensor*. On the other hand we call "non-parametric" a model that is not affine in the data tensor. Affine models are those of the type

$$f_{\mathcal{F},b}(\mathcal{X}) = \langle \mathcal{F}, \mathcal{X} \rangle + b \tag{6}$$

that are considered e.g. in [43]. A related approach found e.g. in [51] considers affine models with a predefined 1-rank parametrization for $\mathcal{F}$: $f_{i_1 i_2 \dots i_N} = v^{(1)}_{i_1} v^{(2)}_{i_2} \cdots v^{(N)}_{i_N}$. The corresponding supervised technique is non-convex and results into an alternating scheme to find b and vectors $\left\{V^{(n)} \in \mathbb{R}^{I_n} \ :\ n \in \mathbb{N}_N\right\}$. We will compare to this approach later on in the experimental Section.

In the next Sections we will discuss a framework to overcome the limitation entailed by the restrictive model class in (6). This is achieved by leveraging the flexibility of kernel methods on the one hand and the structure of data tensors on the other. Next we discuss the integration with kernel methods starting from the simplest cases.

3.1. Naïve Kernels for Data Tensors

Notice that Theorem 2 implies that

$$k(\mathcal{X}, \mathcal{Y}) = \langle \mathcal{X}, \mathcal{Y} \rangle \tag{7}$$

defined upon (2), is a valid reproducing kernel. Indeed (5) reads here $k(\mathcal{X}, \mathcal{Y}) = \langle \text{vec}(\mathcal{X}), \text{vec}(\mathcal{Y}) \rangle$ where $\text{vec}(\cdot)$ denotes vector unfolding and the inner product in the right hand-side is defined on $\mathbb{R}^{I_1 I_2 \cdots I_N}$. Equation (7) is an elementary generalization of the linear kernel defined on $\mathbb{R}^I$. This choice of kernel function is precisely what leads to model of the type (6). In a similar way other kernel functions admit a straightforward generalization to the case where input data are tensors. For instance, a natural way to generalize the popular Gaussian-RBF kernel [40] to data tensors is

$$k(\mathcal{X}, \mathcal{Y}) = \exp\left(-\frac{1}{2\sigma^2}\|\mathcal{X} - \mathcal{Y}\|_F^2\right) \tag{8}$$

where σ is used to set an appropriate bandwitch. However observe that both (7) and (8) treat tensor data as mere collections of entries without keeping into account the underlying structure. In particular notice that (8) can be equivalently restated as:

$$k(\mathcal{X}, \mathcal{Y}) = \prod_{p \in \mathbb{N}_{I_1} \times \mathbb{N}_{I_2} \times \cdots \times \mathbb{N}_{I_N}} \exp\left(-\frac{1}{2\sigma^2}(x_p - y_p)^2\right) \tag{9}$$

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 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 , namely third order tensors, it would additionally neglects the temporal structure.

Notice that (8) is a special case of a more general class of *product kernels*. Later we will introduce a different choice of product kernel that conveniently exploits the algebraic structure of data tensors. First we show in the next Section that product kernels can be seen to arise from a space of infinite dimensional tensors. This fact is relevant on its own as 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 6.2.

3.2. Space of Multilinear Functionals

Assume RKHSs $(\mathfrak{H}_1, \langle\cdot,\cdot\rangle_{\mathfrak{H}_1})$, $(\mathfrak{H}_2, \langle\cdot,\cdot\rangle_{\mathfrak{H}_2}), \dots,$ $(\mathfrak{H}_P, \langle\cdot,\cdot\rangle_{\mathfrak{H}_P})$ of functions on $\mathfrak{X}$ and for any $p \in \mathbb{N}_P$ let $k^p : \mathfrak{X} \times \mathfrak{X} \to \mathbb{R}$ be the reproducing kernel of $\mathfrak{H}_p$. We recall that

$$\psi : \mathfrak{H}_1 \times \mathfrak{H}_2 \times \cdots \times \mathfrak{H}_P \to \mathbb{R} \tag{10}$$

is a bounded (equivalently continuous) multilinear functional [27], if it is linear in each argument and there exists $c \in [0, \infty)$ such that

$$|\psi(h_1, h_2, \dots, h_P)| \leq c\|h_1\|_{\mathfrak{H}_1}\|h_2\|_{\mathfrak{H}_2} \cdots \|h_p\|_{\mathfrak{H}_P}$$

for all $h_i \in \mathfrak{H}_i$, $i \in \mathbb{N}_P$. It is said to be *Hilbert-Schmidt* if it further satisfies

$$\sum_{e_1 \in \mathfrak{E}_1} \sum_{e_2 \in \mathfrak{E}_2} \cdots \sum_{e_P \in \mathfrak{E}_P} |\psi(e_1, e_2, \ldots, e_P)|^2 < \infty$$

for one (equivalently each) orthonormal basis $\mathfrak{E}_p$ of $\mathfrak{H}_p$, $p \in \mathbb{N}_P$. It can be shown [27] that the collections of such well behaved Hilbert-Schmidt functionals endowed with the inner product

$$\langle \psi, \xi \rangle_{\text{HSF}} := \sum_{e_1 \in \mathfrak{E}_1} \sum_{e_2 \in \mathfrak{E}_2} \cdots \sum_{e_P \in \mathfrak{E}_P} \psi(e_1, e_2, \ldots, e_P) \xi(e_1, e_2, \ldots, e_P) \quad (11)$$

forms — by completion — a HS that we denote by HSF.

Proposition 1. *The multilinear functional associated to any $P-$tuple $(h_1, h_2, \ldots, h_P) \in \mathfrak{H}_1 \times \mathfrak{H}_2 \times \cdots \times \mathfrak{H}_P$ and defined by*

$$\psi_{h_1, h_2, \ldots, h_P}(f_1, f_2, \ldots, f_P) := \langle h_1, f_1 \rangle_{\mathfrak{H}_1} \langle h_2, f_2 \rangle_{\mathfrak{H}_2} \cdots \langle h_P, f_P \rangle_{\mathfrak{H}_P} \quad (12)$$

belongs to HSF. *Furthermore it holds that*

$$\langle \psi_{h_1, h_2, \ldots, h_P}, \psi_{g_1, g_2, \ldots, g_P} \rangle_{HSF} = \langle h_1, g_1 \rangle_{\mathfrak{H}_1} \langle h_2, g_2 \rangle_{\mathfrak{H}_2} \cdots \langle h_P, g_P \rangle_{\mathfrak{H}_P} . \quad (13)$$

In particular for any $\mathcal{X} \in \mathfrak{X}$ the multilinear functional

$$\psi_{k^1_{\mathcal{X}}, k^2_{\mathcal{X}}, \ldots, k^P_{\mathcal{X}}}(f_1, f_2, \ldots, f_P) := \langle k^1_{\mathcal{X}}, f_1 \rangle_{\mathfrak{H}_1} \langle k^2_{\mathcal{X}}, f_2 \rangle_{\mathfrak{H}_2} \cdots \langle k^P_{\mathcal{X}}, f_P \rangle_{\mathfrak{H}_P} = f_1(\mathcal{X}) f_2(\mathcal{X}) \cdots f_P(\mathcal{X}) \quad (14)$$

belongs to HSF. *Finally we have for any $\mathcal{X} \in \mathfrak{X}$ and $\mathcal{Y} \in \mathfrak{X}$,*

$$\langle \psi_{k^1_{\mathcal{X}}, k^2_{\mathcal{X}}, \ldots, k^P_{\mathcal{X}}}, \psi_{k^1_{\mathcal{Y}}, k^2_{\mathcal{Y}}, \ldots, k^P_{\mathcal{Y}}} \rangle_{\text{HSF}} = k^1(\mathcal{X}, \mathcal{Y}) k^2(\mathcal{X}, \mathcal{Y}) \cdots k^P(\mathcal{X}, \mathcal{Y}) . \quad (15)$$

Proof. See Appendix Appendix A. □

3.3. Link with Finite Dimensional Tensors

A comparison between rank-1 elements (1) and (12) and between (13) and (3) clarifies the relation between the finite dimensional case and its infinite dimensional extension. Notice that starting from (12) one can let

$$h_1 \otimes h_2 \cdots \otimes h_P := \psi_{h_1, h_2, \ldots, h_P} \quad (16)$$

and define the tensor product space $\mathfrak{H}_1 \otimes \mathfrak{H}_2 \otimes \cdots \otimes \mathfrak{H}_P$ as the completion of the linear span

$$\text{span}\{h_1 \otimes h_2 \otimes \cdots \otimes h_P \ : \ h_i \in \mathfrak{H}_i, \ i \in \mathbb{N}_P\} .$$

This approach gives rise to a space of infinite dimensional $P-$th order tensors. The construction mimics the way $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ was constructed based upon elements (1). However in the next Subsection we give a different derivation which emphasizes the role of reproducing kernels, a key feature to construct practical algorithms.

3.4. Reproducing Kernel Hilbert Space Induced by Multilinear Functionals

Recall from (14) the definition of the multilinear functional $\psi_{k^1_{\mathcal{X}}, k^2_{\mathcal{X}}, \ldots, k^P_{\mathcal{X}}}$. Let

$$\begin{array}{rccc} \tilde{\phi}: & \mathfrak{X} & \to & \text{HSF} \\ & \mathcal{X} & \mapsto & \psi_{k^1_{\mathcal{X}}, k^2_{\mathcal{X}}, \ldots, k^P_{\mathcal{X}}} \end{array} \quad (17)$$

and define $k : \mathfrak{X} \times \mathfrak{X} \to \mathbb{R}$ by

$$k(\mathcal{X}, \mathcal{Y}) := \langle \tilde{\phi}(\mathcal{X}), \tilde{\phi}(\mathcal{Y}) \rangle_{\text{HSF}} . \quad (18)$$

Notice that according to (15), k can be equivalently stated as the product kernel

$$k(\mathcal{X}, \mathcal{Y}) = k^1(\mathcal{X}, \mathcal{Y}) k^2(\mathcal{X}, \mathcal{Y}) \cdots k^P(\mathcal{X}, \mathcal{Y}) \quad (19)$$

where for $p \in \mathbb{N}_P$, k^p denotes the reproducing kernel of $\mathfrak{H}_p$. In the following, in light of (18), we call k the *tensorial kernel*. Notice that k is positive definite since it arises from the well-defined inner product $\langle \cdot, \cdot \rangle_{\text{HSF}}$ and inner products define positive kernels [4]. As well known, a key feature of kernel methods is that it is not needed to define the feature map — which is now $\tilde{\phi}$ — explicitly. Rather, one can choose a positive kernel k and exploit the so-called *kernel trick*. In turn, since by (19) the tensorial kernel k is obtained by the product of the factor kernels $\{k^p\}_{p \in \mathbb{N}_P}$, choosing k amounts to choose the factors.

4. Factor Kernels for Data Tensors

It is important to stress at this point that, as equation (9) shows, the Gaussian-RBF kernel is also a tensorial kernel with factors that depend upon the entry-wise evaluation of data tensors. However, as discussed

See e.g. [40, Definition 2.5] for a formal definition of positive definite kernel.

in Section 3.1, this tensorial kernel does not take advantage of the additional structure of $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$. More generally, the naïve kernels that were considered in Subsection 3.1 act on the data tensors as if they were vectors of $\mathbb{R}^{I_1 I_2 \cdots I_N}$. In this way one defines the distance between two tensors $\mathcal{X}$ and $\mathcal{Y}$ as the length $\|\mathcal{X} - \mathcal{Y}\|_F$ of the straight line segment connecting them. It is well known that many objects of interest live in low dimensional manifolds embedded in high dimensional vector spaces. 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. As we will show this links to the MLSVD and possesses an interesting invariance property. In general the choice of a kernel function should be addressed case by case depending on the specific aspects of the problem of interest. Nonetheless we will show in Section 5 that, in virtue of its properties, the proposed family of kernels especially suits certain tasks involving the analysis of temporal (or spatial) signals.

4.1. 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 1). Assume for now that $I_p < I_{p+1} I_{p+2} \cdots I_N I_1 I_2 I_3 \cdots I_{p-1}$ and denote by $R(\boldsymbol{W})$ the row space of a matrix $\boldsymbol{W} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2}$, $R(\boldsymbol{W}) := \left\{\boldsymbol{W}^\top A \,:\, A \in \mathbb{R}^{I_1}\right\} \subseteq \mathbb{R}^{I_2}$. More precisely we can define for some $\sigma \in \mathbb{R}^+$

$$k^p(\mathcal{X}, \mathcal{Y}) := \exp\left(-\frac{1}{2\sigma^2} d(\mathcal{X}_{<p>}, \mathcal{Y}_{<p>})^2\right) \qquad (20)$$

where $d(\mathcal{X}_{<p>}, \mathcal{Y}_{<p>})$ denotes a suitable distance between $R(\mathcal{X}_{<p>})$ and $R(\mathcal{Y}_{<p>})$ on the *Grassmann manifold* corresponding to the set of I_p dimensional subspaces in a $(I_{p+2} \cdots I_N I_1 I_2 I_3 \cdots I_{p-1})-$dimensional vector space.

The idea of using subspaces has already been exploited to establish a similarity between matrices [21]. This choice has been shown to be relevant in a number of tasks such as face recognition, see e.g. [3] and reference therein. The choice of using an exponential in (20) 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 (19) is a valid reproducing kernel. This, in particular, imposes restrictions on the choice of the distance function d. Notably, however, the definition in (20) implies that the product kernel k in (19) 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 (8) but differs in that the Euclidean norm is replaced by the non-Euclidean distance function defined as:

$$d_T(\mathcal{X}, \mathcal{Y}) = \sqrt{\sum_{n \in \mathbb{N}_N} d(\mathcal{X}_{<n>}, \mathcal{Y}_{<n>})^2}\,. \qquad (21)$$

In (20) we have used p to index a generic matrix unfolding — and not n — to stress that we can consider, as factors, kernels based on matricizations indexed by any subset $\mathfrak{P} \subseteq \mathbb{N}_N$. The choice of factors to be retained can be guided by suitable information criteria such as the kernel-target alignment [12]. In the following we will assume for simplicity that $\mathfrak{P} = \mathbb{N}_N$ and use n instead of p. Later we will show that this case enjoys a special invariance property.

4.2. Relation with Principal Angles

It turns out that any unitarily invariant metric on a Grassmannian manifold connects to the notion of *principal angles*. Let us recall that for $R = \min\{\dim(R(\mathcal{X}_{<n>})), \dim(R(\mathcal{Y}_{<n>}))\}$ the *principal angles* $\theta_1^{(n)}, \theta_2^{(n)}, \ldots, \theta_R^{(n)}$ between $R(\mathcal{X}_{<n>})$ and $R(\mathcal{Y}_{<n>})$ can be defined recursively by $\cos(\theta_r^{(n)}) := \max_{X \in R(\mathcal{X}_{<n>}), Y \in R(\mathcal{Y}_{<n>})} \langle X, Y\rangle = \langle X^{(r)}, Y^{(r)}\rangle$ subject to $\|X\| = \|Y\| = 1$ and $\langle X, X^{(i)}\rangle = \langle Y, Y^{(i)}\rangle = 0$ for $i \in \mathbb{N}_{r-1}$. Among the various distance measures arising from the principal angles [17] a suitable distance between $R(\mathcal{X}_{<n>})$ and $R(\mathcal{Y}_{<n>})$ is the *projection Frobenius norm* (also known as *chordal distance* [7]). It relies on the one-to-one correspondence between a subspace $\mathfrak{A}$ and the associated orthogonal projection $\Pi_{\mathfrak{A}}$ and is defined by:

$$d_{pF}(\mathcal{X}_{<n>}, \mathcal{Y}_{<n>}) := \left\|\Pi_{R(\mathcal{X}_{<n>})} - \Pi_{R(\mathcal{Y}_{<n>})}\right\|_F = \sqrt{2}\left\|\sin\theta^{(n)}\right\|_2 \qquad (22)$$

where $\sin\theta^{(n)}$ is the vector obtained taking the sine of each one of the principal angles between the $n-$th matrix unfoldings $\mathcal{X}_{<n>}$ and $\mathcal{Y}_{<n>}$. This specific choice of distance gives rise to positive definite kernels.

Theorem 3. *If the distance function d corresponds to the projection Frobenius norm* (22) *then the tensorial*

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.

kernel k obtained from the product of factors (20) *is positive definite.*

Proof. The proof is given in Appendix □

4.3. Factors, Tensor Dimensions and Degeneracy

At the beginning of Subsection 4.1 for ease of presentation we made a precise assumption on the dimensions of the $n-$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:** $I_n < I_{n+1}\, I_{n+2} \cdots I_N\, I_1\, I_2\, I_3 \cdots I_{n-1}$. This is the case that we considered above. It holds that $d_{pF}(\mathcal{X}_{<n>}, \mathcal{Y}_{<n>}) > 0$ and hence $k^n(\mathcal{X}, \mathcal{Y}) < 1$ unless $\mathcal{X}_{<n>}$ and $\mathcal{Y}_{<n>}$ span the same row space.
>
> **case 2:** $I_n > I_{n+1}\, I_{n+2} \cdots I_N\, I_1\, I_2\, I_3 \cdots I_{n-1}$. In this case we define k^n in (20) based upon a distance between column spaces instead of row spaces.
>
> **case 3:** $I_n = I_{n+1}\, I_{n+2} \cdots I_N\, I_1\, I_2\, I_3 \cdots I_{n-1}$. Under this condition we have that $k^n(\mathcal{X}, \mathcal{Y}) = 1$ unless *both* $\mathcal{X}_{<n>}$ and $\mathcal{Y}_{<n>}$ are rank deficient. In practice when dealing with real-life noisy data this event does not occur. Thus, in general, the $n-$th matricization is uninformative and we can avoid computing k^n since it does not contribute to the product kernel (19). Notice, however, that the case of square matrix unfolding can occur at most for a single running index $n \in \mathbb{N}_N$: the remaining $N-1$ 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 $R(\mathcal{X}_{<2>})$ and $R(\mathcal{X}_{<3>})$ 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 $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ where $I_1 = I_2 = \cdots = I_N = I$. In practice, as in Subsection 5.3, 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 of section 2.2.

4.4. Link with the MLSVD

Recall that, at a matrix level, the MLSVD of $\mathcal{X}$ boils down to the SVD of the matrix unfoldings $\mathcal{X}_{\langle n \rangle}$, where

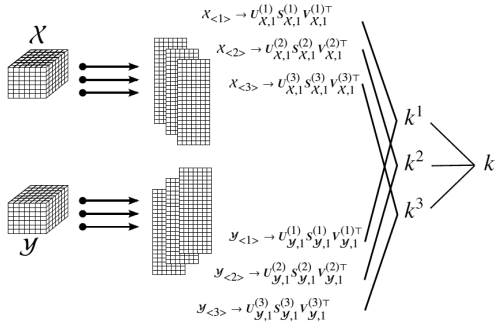


Figure 1: An illustration of the tensorial kernel k based upon factors (24). For 3−rd order tensors $\mathcal{X}$ and $\mathcal{Y}$ it requires to compute the SVD of the matrix unfoldings $\mathcal{X}_{<n>}$ and $\mathcal{Y}_{<n>}$.

$n \in \mathbb{N}_N$. The latter can be stated in block-partitioned form as:

$$\mathcal{X}_{\langle n \rangle} = \begin{pmatrix} \boldsymbol{U}^{(n)}_{\mathcal{X},1} & \boldsymbol{U}^{(n)}_{\mathcal{X},2} \end{pmatrix} \begin{pmatrix} \boldsymbol{S}^{(n)}_{\mathcal{X},1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{V}^{(n)\top}_{\mathcal{X},1} \\ \boldsymbol{V}^{(n)\top}_{\mathcal{X},2} \end{pmatrix} \quad (23)$$

where entries on the diagonal of $\boldsymbol{S}^{(n)}_{\mathcal{X},1}$ are assumed to be ordered in a decreasing manner. A well known property of the SVD decomposition states now that the orthogonal projection operator onto $R(\mathcal{X}_{<n>})$ is given by

$$\Pi_{R(\mathcal{X}_{<n>})} = \boldsymbol{V}^{(n)}_{\mathcal{X},1} \boldsymbol{V}^{(n)\top}_{\mathcal{X},1} \ .$$

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

$$k^n(\mathcal{X}, \mathcal{Y}) = \exp\left(-\frac{1}{2\sigma^2} \left\| \boldsymbol{V}^{(n)}_{\mathcal{X},1} \boldsymbol{V}^{(n)\top}_{\mathcal{X},1} - \boldsymbol{V}^{(n)}_{\mathcal{Y},1} \boldsymbol{V}^{(n)\top}_{\mathcal{Y},1} \right\|^2_F \right) . \quad (24)$$

Figure 1 illustrates the computation of the tensorial kernel based on the SVD's of the matrix unfoldings. Simple matrix algebra shows that (24) is equivalent to $k^n(\mathcal{X}, \mathcal{Y}) = \exp\left(-\frac{1}{\sigma^2}(I_n - \mathrm{trace}(\boldsymbol{Z}^\top \boldsymbol{Z})\right)$ where $\boldsymbol{Z} = \boldsymbol{V}^{(n)\top}_{\mathcal{X},1} \boldsymbol{V}^{(n)}_{\mathcal{Y},1}$. This formula is more efficiently computed than the right hand-side of (24).

5. 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* [5]. 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}: $k(\mathcal{X}, \mathcal{Y}) = \exp(-1/(2\sigma^2) \sum_{n \in \mathbb{N}_N} d_{pF}(\mathcal{X}_{<n>}, \mathcal{Y}_{<n>})^2)$.

5.1. 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 $\mathcal{X}, \mathcal{Y} \in \mathfrak{X}$, then we call $\mathfrak{X}$ a *congruence set*. A characterization of tensors by means of subspaces [14] shows that congruence sets arise, in particular, in the following case.

Theorem 4 (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* $\mathfrak{X} \subset \mathbb{R}^I \otimes \mathbb{R}^I \otimes \mathbb{R}^I$ *is a congruence set if for any* $\mathcal{X} \in \mathfrak{X}$

$$\mathcal{X} = \sum_{r \in \mathbb{N}_R} d_r \, A_r \otimes B_r \otimes C_r \tag{25}$$

for some $D = (d_1, \ldots, d_R) \in \mathbb{C}^R$.

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 (25). Notice that the result holds also for the case where elements of $\mathfrak{X}$ are general complex-valued tensors. A formal proof of Theorem 4 requires additional technical material and is beyond the scope of this manuscript. Further details are found in [14] that actually deals with a broader specification of equivalence classes. Our next goal is to highlight the significance of this result for pattern recognition.

5.2. Implications for Pattern Recognition

A first important remark pertains the nature of congruence sets.

Remark 1. If $\mathfrak{X}_1$ and $\mathfrak{X}_2$ 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 ($\mathfrak{X}_1 \cap \mathfrak{X}_2 = \emptyset$).

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 (24) 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 [12]. Assume we are given a training set of M input-output pairs $\left\{\left(\mathcal{X}^{(m)}, y_m\right) \in \mathfrak{X} \times \mathfrak{Y} \; : \; m \in \mathbb{N}_M\right\}$. Recall the definition of inner product (2) for tensors of arbitrary order. Then the (empirical) kernel-target alignment $A(\boldsymbol{K}, Y)$ is

$$A(\boldsymbol{K}, Y) = \frac{\langle \boldsymbol{K}, YY^\top \rangle}{M \sqrt{\langle \boldsymbol{K}, \boldsymbol{K} \rangle}} \tag{26}$$

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 [11].

Equation (26) 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 (24) 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.

5.3. The Special Case of Hankel Tensors

In this section we consider a specific class of tensorial representations. We focus of 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 the we can write

$$s_t = \sum_{k=0}^{T-1} \xi_k z_k^t \tag{27}$$

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

$z_k = \exp((i2\pi f_k - d_k)\Delta t)$, the k-th pole of the signal. One specific situation arise when $d_k = 0$, $f_k = k$ and finally $\Delta t = \frac{1}{T}$ in which case (27) is the Inverse Discrete Fourier Transform (IDFT) [8]. 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 [35] can be defined entry-wise by

$$x_{i_1 i_2 i_3} := s_{i_1+i_2+i_3-3} \; . \tag{28}$$

In light of (27) 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:

$$\mathcal{X} = \sum_{k \in \mathbb{N}_T} \xi_{k-1} \begin{pmatrix} z_{k-1}^0 \\ z_{k-1}^1 \\ \vdots \\ z_{k-1}^{I_1-1} \end{pmatrix} \otimes \begin{pmatrix} z_{k-1}^0 \\ z_{k-1}^1 \\ \vdots \\ z_{k-1}^{I_2-1} \end{pmatrix} \otimes \begin{pmatrix} z_{k-1}^0 \\ z_{k-1}^1 \\ \vdots \\ z_{k-1}^{I_3-1} \end{pmatrix} . \tag{29}$$

When $\mathcal{X}$ is cubic the latter is seen to be a special case of (25). Theorem 4 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 (28) 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 [36] and also the multichannel case can be dealt with [35]. 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 (24) can be simplified to

$$k(\mathcal{X}, \mathcal{Y}) = \exp\left(-\frac{1}{2\sigma^2} \left\| \boldsymbol{V}_{\mathcal{X},1}^{(1)} \boldsymbol{V}_{\mathcal{X},1}^{(1)\top} - \boldsymbol{V}_{\mathcal{Y},1}^{(n)} \boldsymbol{V}_{\mathcal{Y},1}^{(1)\top} \right\|_F^2 \right) \tag{30}$$

where we considered only the first matricization. In Section 7 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* [44].

We denoted by i the imaginary unit $i = \sqrt{-1}$.

6. Model Estimation

We now turn to the general learning problem of interest. We want to estimate a model f to predict a target variable $y \in \mathcal{Y} \subseteq \mathbb{R}$ from an input pattern $\mathcal{X} \in \mathfrak{X}$ given a training set of M input-output pairs

$$\left\{ \left(\mathcal{X}^{(m)}, y_m \right) \in \mathfrak{X} \times \mathcal{Y} \; : \; m \in \mathbb{N}_M \right\} .$$

Since k in (19) is of positive type, the Moore-Aronszajn theorem [1],[4] ensures that there exists only one Hilbert space $\mathfrak{H}_k$ of functions on $\mathfrak{X}$ with k as reproducing kernel. The estimation of a non-parametric model of $\mathcal{X}$ can then be formulated as a variational problem in the function space $\mathfrak{H}_k$. In spite of the infinite dimensionality of the latter a solution can be found based on finite dimensional optimization as ensured by representer theorems, see [29], [39].

6.1. Primal-Dual Techniques

An alternative approach relies on primal-dual techniques that underlies Support Vector Machines (SVM) and related estimators [48],[47],[49]. In this case one starts from a primal model representation of the type:

$$f_{(\Psi,b)}(\mathcal{X}) := \langle \Psi, \tilde{\phi}(\mathcal{X}) \rangle_{\mathrm{HSF}} + b \; . \tag{31}$$

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, (31) is affine in $\tilde{\phi}(\mathcal{X})$ as much as (6) 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.

Relying on Lagrangian duality arguments the problem is re-parametrized in terms of dual variables $\{\alpha_m\}_{m \in \mathbb{N}_M}$ and solved in $(\alpha, b) \in \mathbb{R}^{M+1}$. 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 $\mathcal{Y} = \{+1, -1\}$ and (31) 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 [10] translates into convex quadratic programs. By contrast, in least-squares SVM (LS-SVM) [48], a modification of the SVM primal problem leads to a considerably simpler estimation problem. In particular, the primal

formulation for classification [50] reads in our setting:

$$\min_{(\Psi,E,b)\in \mathrm{HSF}\times\mathbb{R}^M\times\mathbb{R}} \ \tfrac{1}{2}\langle\Psi,\Psi\rangle_{\mathrm{HSF}} + \gamma\tfrac{1}{2}\sum_{m\in\mathbb{N}_M} e_m^2$$
$$\text{subject to } \ y_m(\langle\Psi,\tilde{\phi}(\mathcal{X}^{(m)})\rangle_{\mathrm{HSF}} + b) = 1 - e_m,\ m\in\mathbb{N}_M \tag{32}$$

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:

$$\begin{bmatrix} 0 & Y^\top \\ Y & \boldsymbol{\Omega} + \frac{1}{\gamma}\boldsymbol{I}_M \end{bmatrix}\begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_M \end{bmatrix} \tag{33}$$

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

$$(\boldsymbol{\Omega})_{ij} = y_i y_j\langle\tilde{\phi}(\mathcal{X}^{(i)}),\tilde{\phi}(\mathcal{X}^{(j)})\rangle_{\mathrm{HSF}} = y_i y_j k(\mathcal{X}^{(i)},\mathcal{X}^{(j)})\ .$$

Finally to evaluate $f_{(\Psi^\star,b^\star)}$ at a given test point $\mathcal{X}$, the dual model representation is exploited:

$$f_{(\Psi^\star,b^\star)}(\mathcal{X}) = \sum_{m\in\mathbb{N}_M} y_m\alpha_m^\star k(\mathcal{X}^{(m)},\mathcal{X}) + b^\star\ . \tag{34}$$

Notice that problem (32) 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 [34],[18],[20],[2] and [26], among others. For Vapnik's SVM formulation a rigorous primal-dual derivation is discussed in [32]. Similar results for LS-SVM have not been reported, to the best of our knowledge. As an additional contribution we then give a formal derivation in Appendix C.

The procedure to compute a model with the tensorial kernel is summarized in Table 2. It is assumed that both the parameter γ in (33) and σ in (24) are given. In practical applications the choice of these parameter is performed according to some model selection criterion often based on cross-validation.

6.2. Structure-inducing Penalties

It is worth noticing that the optimality conditions of (32) (see (C.8)) yields

$$\Psi^\star = \sum_{m\in\mathbb{N}_M}\alpha_m^\star y_m\tilde{\phi}(X^{(m)}) \tag{35}$$

which — given the nature of HSF — shows that the optimal multilinear functional $\Psi^\star$ has at most rank M where M is the cardinality of the training set. In SVM-like algorithms the complexity of the model is usually controlled by a notion of margin [55] which is here attached to $\langle\Psi,\Psi\rangle_{\mathrm{HSF}}$, the squared Frobenius norm of Ψ. In the present context the interpretation of equation (35) suggests that an additional complexity measure might be based on some generalized notion of rank [25],[24]. Recently the use of the nuclear norm was proposed to define convex relaxation for rank constrained matrix problem [37]. This approach parallels the use of the l_1 norm in sparse approximation and cardinality minimization [52],[16]. Extension of the nuclear norm to higher order tensors has been considered in [43], [33]. Hence we remark that an interesting extension, that we do not approach here, might be to consider a penalty of this type in the infinite dimensional setting of problem (32).

Table 2: Model estimation with factor kernels (24)

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**
    - $\boldsymbol{V}^{(n)}_{\mathcal{X}^{(m_1)},1} \leftarrow \mathrm{SVD}(\mathcal{X}^{(m_1)}_{<n>})$
    - $\boldsymbol{V}^{(n)}_{\mathcal{X}^{(m_2)},1} \leftarrow \mathrm{SVD}(\mathcal{X}^{(m_2)}_{<n>})$
    - $\boldsymbol{Z}_{(n)} \leftarrow \boldsymbol{V}^{(n)\top}_{\mathcal{X}^{(m_1)},1}\boldsymbol{V}^{(n)}_{\mathcal{X}^{(m_2)},1}$
    - $a_n \leftarrow I_n - \mathrm{trace}(\boldsymbol{Z}^\top_{(n)}\boldsymbol{Z}_{(n)})$
- $(\boldsymbol{\Omega})_{m_1m_2} \leftarrow y_{m_1}y_{m_2}\exp\left(-\frac{1}{\sigma^2}(a_1+a_2+\cdots+a_N)\right)$

$\boldsymbol{\Omega}\leftarrow\boldsymbol{\Omega}+\boldsymbol{\Omega}^\top+\mathbf{I}_\mathrm{M}$
comment: Find model parameters
Solve (33) for given $\boldsymbol{\Omega}$, Y and parameter γ .

7. Experimental Results

7.1. Classification of Sparsity Patterns

The purpose of this experiment is to test the impact of the invariance property studied in Section 5 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:

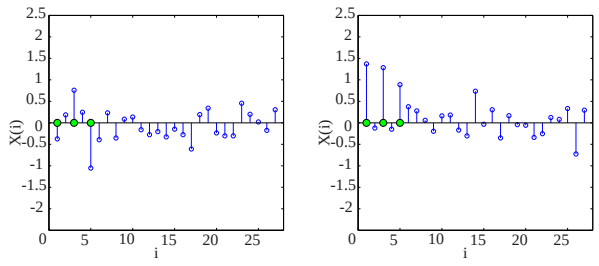
$$\Delta_j := E^j\otimes E^j\otimes E^j\ .$$

We generated data tensors in $\mathbb{R}^I\otimes\mathbb{R}^I\otimes\mathbb{R}^I$ according to the following model:

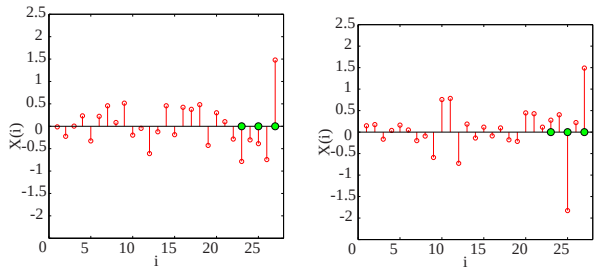
$$\mathcal{X}^{(m)} = \begin{cases} a_m\Delta_1 + b_m\Delta_2 + c_m\Delta_3 + \mathcal{E}^{(m)}, & \text{if } y_m = +1 \\ a_m\Delta_4 + b_m\Delta_5 + c_m\Delta_6 + \mathcal{E}^{(m)}, & \text{if } y_m = -1 \end{cases} \tag{36}$$

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 (36) and (25) reveals that for $\beta^2 = 0$ (noiseless case) the two classes of tensors correspond to separate congruence sets, see also Remark 1. 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 2 for the case where $I = 3$. We use the LS-SVMlab tool-



(a) class 1



(b) class 2

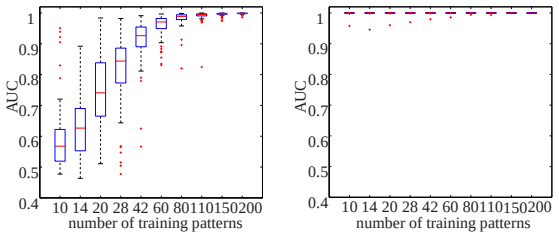
Figure 2: By vector unfolding the experiment of Section 7.2 can be interpreted as the classification of sparsity patterns of (noisy) vectors. As an example we take here $I = 3$ and plot the 27 elements of the vectorized version of data tensors generated according to (36). The solid green dots in plots 2(a) and 2(b) represent two hypothetical index sets of non-zero entries before corruption by gaussian noise with variance β^2.

box (`www.esat.kuleuven.be/sista/lssvmlab`, [13]) and perform training with M input-output pairs $\left\{\left(\mathcal{X}^{(m)}, y_m\right) : m \in \mathbb{N}_M\right\}$. We compared the naïve Gaussian-RBF kernel function (8) (Gauss-RBF in the tables) — which corresponds to vectorizing the tensors — with the tensorial kernel based on factors (24) (tensorial in the tables) for increasing values of M. We also compared with affine tensor-based models (6) 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 [51]. 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 3 refers to the

Table 3: Accuracy on test data for $I = 7$, $\beta^2 = 0.05$

AUC performance: mean (and standard deviation)

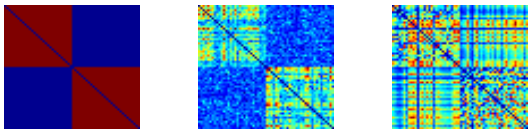
M	tensorial (19)-(24)	Gauss-RBF (8)	linear rank-1 [51]
10	**0.86(0.04)**	0.53(0.07)	0.50(0.04)
14	**0.88(0.03)**	0.53(0.05)	0.51(0.03)
20	**0.88(0.09)**	0.61(0.10)	0.50(0.02)
28	**0.92(0.02)**	0.60(0.10)	0.50(0.02)
42	**0.94(0.02)**	0.63(0.10)	0.50(0.02)
60	**0.95(0.02)**	0.69(0.08)	0.50(0.01)
80	**0.96(0.02)**	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)



(a) (b)

Figure 3: Synthetic example, $I = 10$, $\beta^2 = 0.005$ and increasing number of training examples. Boxplots of AUC obtained over the same 200 test patterns for for the Gaussian-RBF kernel 3(a) and for the tensorial kernel 3(b).

case of increasing values of M, $I = 7$ 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 3 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 (7) (not reported in Table 3). 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 [12] can reveal before training how well different kernel functions capture the structures of the problem. A good alignment often results in visually detectable patterns,



(a) (b) (c)

Figure 4: Classification of sparsity patterns ($\beta^2 = 0.05$ and $I = 10$). 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.

see Figure 4. In general we observed that models based on the Gaussian-RBF kernel (which is *universal* [46]) also reach perfect classification accuracy when M is sufficiently large. This shows that exploiting the underlying invariance property is relevant especially for small sample size problems.

7.2. Recognition of Signals

We now present a simple example to illustrate Subsection 5.3. 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 variable drawn from a normal distribution. Notice that Δ_y in the previous is defined upon the signal's label. In turn, 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 M signal-label pairs where generated for training. The 57-dimensional vector corresponding to the m-th training signal $S^{(m)}$ was either fed directly into kernels for vectors:

$$k\left(S^{(m_1)}, S^{(m_2)}\right) = \exp\left(-\sigma^2 \left\|S^{(m_1)} - S^{(m_2)}\right\|^2\right) \quad (37)$$

$$k\left(S^{(m_1)}, S^{(m_2)}\right) = \left\langle S^{(m_1)}, S^{(m_2)} \right\rangle \quad (38)$$

called respectively Gauss-RBF vec and linear vec, or first converted into an Hankel tensor $\mathcal{X}^{(m)} \in \mathbb{R}^{20} \times \mathbb{R}^{20} \times \mathbb{R}^{20}$ as explained in Section 5.3. For this latter tensorial representations we then used the Gaussian kernel (8) (Gauss-RBF), the linear kernel (6) (linear) and the simplified version of tensorial kernel that holds for Hankel tensors (30) (tensorial). We also considered affine tensor-based models (6) 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. As in the previous example the tensorial kernel leads to far more accurate predictions in the low range of M. 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.

Table 4: Accuracy for the signals example

AUC performance: mean (and standard deviation)

M	tensorial (30)	Gauss-RBF (8)	linear rank-1 [51]
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)

M	Gauss-RBF vec (37)	linear vec (38)	linear (7)
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)

7.3. Libras Movement Data

Next we consider the Libras Movement Data Set [19] 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 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 structures [36] and features similar properties as the Hankel tensor we discussed in Section 5.3 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 5.

Table 5: Accuracy on test data for Libras

AUC performance: mean (and standard deviation)

task	tensorial (19)-(24)	Gauss-RBF (8)	linear rank-1 [51]
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 (7)	Gauss-RBF vec (37)	linear vec (38)
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)

7.4. Aerial Views

Table 6: Accuracy on test data for Aerial Views

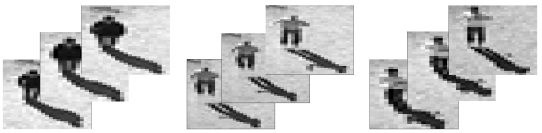
AUC performance: mean (and standard deviation)

task	tensorial (19)-(24)	Gauss-RBF (8)
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 (7)	linear rank-1 [51]
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)

These experiments are about the Aerial View Activity Classification Dataset [6]. 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 5. As a preprocessing step we normalize the videos in the datasets. Each frame 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 $10 \times 13 \times M$ tensor consisting of M frames. Denote by $\tilde{\mathcal{X}}'_{<3>}$ the matrix

(a) class 3 (digging)



(b) class 9 (jumping)

Figure 5: Examples of frames taken from low-resolution videos of human activities.

obtained centering the columns of the $130 \times M$ matrix $\tilde{\mathcal{X}}'_{<3>}$. We compute from the $M \times M$ empirical covariance matrix $1/129\bar{\mathcal{X}}_{<3>}\bar{\mathcal{X}}'_{<3>}$ the 4 principal eigenvectors $\boldsymbol{E} = [E_1, \cdots, E_4] \in \mathbb{R}^M \otimes \mathbb{R}^4$ 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 (19)-(24), linear kernel (7) and rank-1 models [51]. As before we averaged the performances over 100 replicates obtained from random splitting of training and test set. Results for different pairs of classes are reported in Table 6.

8. Conclusion

In this paper we have introduced a new framework to go beyond the class of affine models considered in the existing supervised tensor-based methods. This was achieved by exploiting the flexibility of kernel methods on the one hand and the structure of data tensors on the other. We began by showing that product kernels, among which the popular Gaussian-RBF kernel, arise from the space HSF of infinite dimensional analogue of finite dimensional tensors. This realization is important on its own as it shows that kernels are closely connected with the seemingly distinct domain of tensor-based techniques. We then turned to the problem of implicitly mapping data tensor into HSF by defining suitable factor kernels. Contrary to naïve kernels, the tensorial kernel we proposed keeps into account the intrinsic geometry of data tensors by leveraging the Grassman-

nian nature of matrix unfoldings. 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.

Acknowledgements

Research supported by Research Council KUL: GOA Ambiorics, GOA MaNet, CoE EF/05/006 Optimization in Engineering(OPTEC), CIF1 and STRT1/08/023 IOF-SCORES4CHEM. Flemish Government: FWO: PhD/ postdoc grants, projects: G0226.06 (cooperative systems and optimization), G0321.06 (Tensors), G.0427.10N, G.0302.07 (SVM/Kernel), G.0588.09 (Brain-machine) research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011); EU: ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940).

References

[1] Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404.

[2] Barbu, V. and Precupanu, T. (1986). *Convexity and optimization in Banach spaces*. Springer.

[3] Basri, R., Hassner, T., and Zelnik-Manor, L. (2010). Approximate Nearest Subspace Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):266–278.

[4] Berlinet, A. and Thomas-Agnan, C. (2004). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers.

[5] Burges, C. (1999). *Advances in Kernel Methods: Support Vector Learning*, chapter Geometry and invariance in kernel based methods, pages 89–116. MIT Press Cambridge, MA, USA.

[6] Chen, C., Ryoo, M., and Aggarwal, J. (2010). UT-Tower Dataset: Aerial View Activity Classification Challenge. http://cvrc.ece.utexas.edu/SDHA2010/Aerial_View_Activity.html.

[7] Conway, J., Hardin, R., and Sloane, N. (1996). Packing lines, planes, etc.: Packings in Grassmannian spaces. *Experimental Mathematics*, 5:139–159.

[8] Cooley, J. and Tukey, J. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

[9] Coppi, R. and Bolasco, S. (1989). *Multiway data analysis*. North-Holland Publishing Co. Amsterdam, The Netherlands.

[10] Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20:273–297.

[11] Cristianini, N., Kandola, J., Elisseeff, A., and Shawe-Taylor, J. (2006). On kernel target alignment. In Holmes, D. and Jain, L., editors, *Innovations in Machine Learning*, volume 194 of *Studies in Fuzziness and Soft Computing*, pages 205–256. Springer Berlin / Heidelberg.

[12] Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J. (2002). On kernel-target alignment. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 367–373.

[13] De Brabanter, K., Karsmakers, P., Ojeda, F., Alzate, C., De Brabanter, J., Pelckmans, K., De Moor, B., Vandewalle, J., and Suykens, J. A. K. (2010). LS-SVMlab toolbox user's guide version 1.8. *Internal Report 10-146, ESAT-SISTA, K.U.Leuven (Leuven, Belgium)*.

[14] De Lathauwer, L. (2011). Characterizing higher-order tensors by means of subspaces. *Internal Report 11-32, ESAT-SISTA, K.U. Leuven (Leuven, Belgium)*.

[15] De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278.

[16] Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306.

[17] Edelman, A., Arias, T. A., and Smith, S. T. (1999). The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353.

[18] Ekeland, I. and Temam, R. (1976). *Convex Analysis and Variational Problems*. North-Holland Publishing Co.

[19] Frank, A. and Asuncion, A. (2010). UCI machine learning repository, University of California, Irvine, School of Information and Computer Sciences, http://archive.ics.uci.edu/ml.

[20] Girsanov, I., Poljak, B., and Louvish, D. (1972). *Lectures on mathematical theory of extremum problems*. Springer Berlin-Heidelberg-New York.

[21] Hamm, J. and Lee, D. (2008). Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th international conference on Machine learning*, pages 376–383. ACM.

[22] 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.

[23] He, X., Cai, D., and Niyogi, P. Tensor subspace analysis. In *Advances in Neural Information Processing Systems (NIPS), 2006*, pages 499–506.

[24] Hitchcock, F. (1927). Multiple invariants and generalized rank of a p-way matrix or tensor. *J. Math. Phys*, 7(1):39–79.

[25] Ishteva, M., Absil, P., Huffel, S., and Lathauwer, L. (2010). On the Best Low Multilinear Rank Approximation of Higher-order Tensors. *Recent Advances in Optimization and its Applications in Engineering, Part 3*, pages 145–164.

[26] Ito, K. and Kunisch, K. (2008). *Lagrange multiplier approach to variational problems and applications*. Advances in Design and Control. SIAM.

[27] Kadison, R. V. and Ringrose, J. R. (1983). *Fundamentals of the theory of operator algebras*, volume 15 of *Graduate Studies in Mathematics*. American Mathematical Society.

[28] Kiers, H. A. L. (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics*, 14(3):105–122.

[29] Kimeldorf, G. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95.

[30] Kolda, T. and Bader, B. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

[31] Kroonenberg, P. (2008). *Applied multiway data analysis*. Wiley-Interscience.

[32] Lin, C. (2001). Formulations of support vector machines: a note from an optimization point of view. *Neural Computation*, 13(2):307–317.

[33] Liu, J., Musialski, P., Wonka, P., and Ye, J. (2009). Tensor completion for estimating missing values in visual data. In *IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan*, pages 2114–2121.

[34] Luenberger, D. (1969). *Optimization by Vector Space Methods*. John Wiley and Sons, Inc., New York.

[35] Papy, J., De Lathauwer, L., and Van Huffel, S. (2005). Exponen-

tial data fitting using multilinear algebra: the single-channel and multi-channel case. *Numerical linear algebra with applications*, 12(8):809–826.

[36] Papy, J., De Lathauwer, L., and Van Huffel, S. (2009). Exponential data fitting using multilinear algebra: the decimative case. *J. Chemometrics*, 23(7-8):341–351.

[37] Recht, B., Fazel, M., and Parrilo, P. (2007). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52:471–501.

[38] Riesz, F. and Sz.-Nagy, B. (1955). *Functional Analysis*. Frederick Ungar Publishing Co., New York.

[39] Schälkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. *Proceedings of the Annual Conference on Computational Learning Theory (COLT)*, pages 416–426.

[40] Schölkopf, B. and Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press.

[41] Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

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

[43] Signoretto, M., De Lathauwer, L., and Suykens, J. A. K. (2010b). Nuclear Norms for Tensors and Their Use for Convex Multilinear Estimation. *Internal Report 10-186, ESAT-SISTA, K.U.Leuven (Leuven, Belgium)*.

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

[45] Smilde, A., Bro, R., and Geladi, P. (2004). *Multi-way analysis with applications in the chemical sciences*. Wiley.

[46] Steinwart, I. (2002). On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93.

[47] Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Verlag.

[48] Suykens, J., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J. (2002). *Least squares support vector machines*. World Scientific.

[49] Suykens, J. A. K., Alzate, C., and Pelckmans, K. (2010). Primal and dual model representations in kernel-based learning. *Statistics Surveys*, 4:148–183 (electronic). DOI: 10.1214/09–SS052.

[50] Suykens, J. A. K. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300.

[51] Tao, D., Li, X., Wu, X., Hu, W., and Maybank, S. (2007). Supervised tensor learning. *Knowledge and Information Systems*, 13(1):1–42.

[52] Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

[53] Tucker, L. R. (1964). *The extension of factor analysis to three-dimensional matrices*, volume Contributions to Mathematical Psychology, Holt, Rinehart Winston, NY, pages 109–127.

[54] Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.

[55] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer Verlag, New York.

[56] Wahba, G. (1990). *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia.

Appendix A. Proof of Proposition 1

The reader is referred to [27, Proposition 2.6.2] for a proof of the first two statements. Here we proof the remaining assertions that are specific to our context. First of all notice that the multilinear functional defined in (14) is clearly bounded as it follows from the definition of RKHS. In order to prove that $\psi_{k^1_{\mathcal{X}},k^2_{\mathcal{X}},\ldots,k^P_{\mathcal{X}}}$ indeed belongs to HSF we need to show that it is Hilbert-Schmidt. This is the case since we have:

$$\sum_{e_1\in\mathfrak{E}_1}\sum_{e_2\in\mathfrak{E}_2}\cdots\sum_{e_P\in\mathfrak{E}_P}|\psi_{k^1_{\mathcal{X}},k^2_{\mathcal{X}},\ldots,k^P_{\mathcal{X}}}(e_1,e_2,\ldots,e_P)|^2 = \sum_{e_1\in\mathfrak{E}_1}\sum_{e_2\in\mathfrak{E}_2}\cdots\sum_{e_P\in\mathfrak{E}_P}|\langle k^1_{\mathcal{X}},e_1\rangle_{\mathfrak{H}_1}\langle k^2_{\mathcal{X}},e_2\rangle_{\mathfrak{H}_2}\cdots\langle k^P_{\mathcal{X}},e_P\rangle_{\mathfrak{H}_P}|^2 = \|k^1_{\mathcal{X}}\|^2_{\mathfrak{H}_1}\cdots\|k^P_{\mathcal{X}}\|^2_{\mathfrak{H}_P}<\infty\,. \quad \text{(A.1)}$$

By the definition of inner product in (11) we now have:

$$\begin{aligned}&\langle\psi_{k^1_{\mathcal{X}},k^2_{\mathcal{X}},\ldots,k^P_{\mathcal{X}}},\psi_{k^1_{\mathcal{Y}},k^2_{\mathcal{Y}},\ldots,k^P_{\mathcal{Y}}}\rangle_{\text{HSF}} = \\ &\qquad\sum_{e_1\in\mathfrak{E}_1}\sum_{e_2\in\mathfrak{E}_2}\cdots\sum_{e_P\in\mathfrak{E}_P}\langle k^1_{\mathcal{X}},e_1\rangle_{\mathfrak{H}_1}\langle k^2_{\mathcal{X}},e_2\rangle_{\mathfrak{H}_2}\cdots \\ &\qquad\langle k^P_{\mathcal{X}},e_P\rangle_{\mathfrak{H}_P}\langle k^1_{\mathcal{Y}},e_1\rangle_{\mathfrak{H}_1}\langle k^2_{\mathcal{Y}},e_2\rangle_{\mathfrak{H}_2}\cdots\langle k^P_{\mathcal{Y}},e_P\rangle_{\mathfrak{H}_P} = \\ &\sum_{e_1\in\mathfrak{E}_1}\left(\langle k^1_{\mathcal{X}},e_1\rangle_{\mathfrak{H}_1}\langle k^1_{\mathcal{Y}},e_1\rangle_{\mathfrak{H}_1}\right)\cdots\sum_{e_P\in\mathfrak{E}_P}\left(\langle k^P_{\mathcal{X}},e_P\rangle_{\mathfrak{H}_P}\langle k^P_{\mathcal{Y}},e_P\rangle_{\mathfrak{H}_P}\right) = \\ &\qquad\langle k^1_{\mathcal{X}},k^1_{\mathcal{Y}}\rangle_{\mathfrak{H}_1}\cdots\langle k^P_{\mathcal{X}},k^P_{\mathcal{Y}}\rangle_{\mathfrak{H}_P} = k^1(\mathcal{X},\mathcal{Y})\cdots k^P(\mathcal{X},\mathcal{Y}) \quad \text{(A.2)}\end{aligned}$$

that proves (15).

Appendix B. Proof of Theorem 3

To show that k is positive definite it is enough to show that the factors are positive definite [4]. Let

$$\begin{aligned}\psi_n:\quad \mathbb{R}^{I_1}\otimes\cdots\otimes\mathbb{R}^{I_N} &\to \mathbb{R}^{(I_1I_2\cdots I_N)^2}\\ \mathcal{X} &\mapsto \text{vec}(\Pi_{R(\mathcal{X}_{<n>})})\end{aligned}$$

and introduce the kernel function

$$\begin{aligned}g:\quad \mathbb{R}^{(I_1I_2\cdots I_N)^2}\times\mathbb{R}^{(I_1I_2\cdots I_N)^2} &\to \mathbb{R}\\ (X,Y) &\mapsto \exp\left(\langle X,Y\rangle/\sigma^2\right).\end{aligned} \quad \text{(B.1)}$$

We first show that the latter is positive definite. To see this, notice that the exponential function can be arbitrarily well approximated by polynomials with positive coefficients and hence is a limit of kernels. Since the positive definiteness is closed under taking pointwise limit, the result follows (see e.g. [41, Proposition 3.24, point ii]). Additionally also

$$g^n(\mathcal{X},\mathcal{Y}) := g(\psi_n(\mathcal{X}),\psi_n(\mathcal{Y})) \quad \text{(B.2)}$$

is positive definite since the kernel matrix G^n arising from evaluating g at any arbitrary $T-$tuple

$\left(\psi_n\left(\mathcal{X}^{(1)}\right), \psi_n\left(\mathcal{X}^{(2)}\right), \cdots, \psi_n\left(\mathcal{X}^{(T)}\right)\right)$ is such. Now observe that for $\mathfrak{H}_{g^n} \ni g^n(\mathcal{X}) := g^n(\mathcal{X}, \cdot)$ the normalized evaluation functional $\overline{g}^n(\mathcal{X}) := 1/(\|g^n(\mathcal{X})\|_{\mathfrak{H}_{g^n}} g^n(\mathcal{X}))$ gives rise to the positive definite kernel $\overline{g}^n(\mathcal{X}, \mathcal{Y}) := \langle \overline{g}^n(\mathcal{X}), \overline{g}^n(\mathcal{Y})\rangle_{\mathfrak{H}_{g^n}} = \frac{g^n(\mathcal{X},\mathcal{Y})}{\sqrt{g^n(\mathcal{X},\mathcal{X})}\sqrt{g^n(\mathcal{Y},\mathcal{Y})}}$. Replacing (B.2) into the latter and keeping into account (B.1) we obtain

$$\frac{g^n(\mathcal{X},\mathcal{Y})}{\sqrt{g^n(\mathcal{X},\mathcal{X})}\sqrt{g^n(\mathcal{Y},\mathcal{Y})}} = \frac{\exp\left(\langle\psi_n(\mathcal{X}), \psi_n(\mathcal{Y})\rangle/\sigma^2\right)}{\sqrt{\exp\left(\langle\psi_n(\mathcal{X}), \psi_n(\mathcal{X})\rangle/\sigma^2\right)}\sqrt{\exp\left(\langle\psi_n(\mathcal{Y}), \psi_n(\mathcal{Y})\rangle/\sigma^2\right)}} = \exp\left(\frac{1}{\sigma^2}\langle\psi_n(\mathcal{X}), \psi_n(\mathcal{Y})\rangle - \frac{1}{2\sigma^2}\langle\psi_n(\mathcal{X}), \psi_n(\mathcal{X})\rangle - \frac{1}{2\sigma^2}\langle\psi_n(\mathcal{Y}), \psi_n(\mathcal{Y})\rangle\right) = \exp\left(-\frac{1}{2\sigma^2}\|\psi_n(\mathcal{X}) - \psi_n(\mathcal{Y})\|^2\right) .$$

By definition of ψ_n the last member corresponds now to $\exp\left(-\frac{1}{2\sigma^2}\left\|\Pi_{R(\mathcal{X}_{<n>})} - \Pi_{R(\mathcal{Y}_{<n>})}\right\|_F^2\right)$ which concludes the proof.

Appendix C. LS-SVM and Optimization in Infinite Dimensional Spaces

We first recall the results that we need in a general HS setting. Successively, we detail the derivation of LS-SVM for classification starting from (32).

Appendix C.1. Generalized Differential and Gradient

In the following $(\mathfrak{H}, \langle\cdot,\cdot\rangle_{\mathfrak{H}})$ will denote a HS and f a functional on $\mathfrak{H}$, namely a mapping of the type $f : \mathfrak{H} \to \mathbb{R}$. We recall that f is convex if $\mathrm{dom}(f) := \{h \in \mathfrak{H} : |f(h)| < \infty\}$ 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 implied in particular if f is linear or affine.

Definition 3 (Subgradient and Subdifferential [18]). Let $f : \mathfrak{H} \to \mathbb{R}$ be a convex functional. An element $g \in \mathfrak{H}$ is called *subgradient* of f at $h_0 \in \mathrm{dom}(f)$ if for any $h \in \mathrm{dom}(f)$ we have $f(h) \geq f(h_0) + \langle g, h - h_0\rangle_{\mathfrak{H}}$. 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 2. 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 [34],[18],[2]. In particular, the fact that HS's are reflexive implies that subgradients of functionals can be considered as elements of the same space and the use of more general duality pairings can be avoided.

Definition 4 (Gateaux Differential). Let $f : \mathfrak{H} \to \mathbb{R}$ 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:

$$f'(h; s) = \lim_{\alpha \downarrow 0} \frac{1}{\alpha}\left(f(h + \alpha s) - f(h)\right) . \tag{C.1}$$

If there exists $h^\star \in \mathfrak{H}$ such that

$$f'(h; s) = \left\langle s, h^\star\right\rangle_{\mathfrak{H}} \ \forall s \in \mathfrak{H} \tag{C.2}$$

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. [18]) that if f is Gateaux-differentiable at $h \in \mathfrak{H}$ 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 3. If f is a continuous linear functional, then by the Riesz theorem there exists $h^\star$ such that $f(h) = \langle h, h^\star\rangle_{\mathfrak{H}}$ for any $h \in \mathfrak{H}$. It is immediate to see now that $f'(h; s) = \lim_{\alpha\downarrow 0}\frac{1}{\alpha}\left(f(h + \alpha s) - f(h)\right) = \langle s, h^\star\rangle_{\mathfrak{H}}$ and hence that $h^\star$ is the Gateaux-differential at h for any $h \in \mathfrak{H}$. Similarly if f is a continuous affine functional: $f(h) = \langle h, h^\star\rangle_{\mathfrak{H}} + b$ then again $h^\star$ is the Gateaux-differential at h for any $h \in \mathfrak{H}$.

Remark 4. If $f(h) = \langle h, h\rangle_{\mathfrak{H}}$ simple calculus shows that equation (C.1) reads $f'(h; s) = 2\langle s, h\rangle_{\mathfrak{H}}$. Hence by equation (C.2) $f'(h) = 2h$.

Appendix C.2. The Case of Composite Spaces

Given two HS's $(\mathfrak{H}_1, \langle\cdot,\cdot\rangle_{\mathfrak{H}_1})$ and $(\mathfrak{H}_2, \langle\cdot,\cdot\rangle_{\mathfrak{H}_2})$ we can consider the product space $\mathfrak{H}_1 \times \mathfrak{H}_2$ consisting of ordered pairs (h_1, h_2). Such a space can be turned into a HS $\mathfrak{H}$ based upon the inner product $\langle(h_1, h_2), (g_1, g_2)\rangle_{\mathfrak{H}} := \langle h_1, g_1\rangle_{\mathfrak{H}_1} + \langle h_2, g_2\rangle_{\mathfrak{H}_2}$. A *separable* functional on $\mathfrak{H}$ is now a functional of the type $f((h_1, h_2)) = f_1(h_1) + f_2(h_2)$. 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

$$f'((h_1, h_2); (s_1, s_2)) = \left\langle s_1, h_1^\star\right\rangle_{\mathfrak{H}_1} + \left\langle s_2, h_2^\star\right\rangle_{\mathfrak{H}_2} \ \forall (s_1, s_2) \in \mathfrak{H} \tag{C.3}$$

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 $\mathfrak{H}_1 \times \mathfrak{H}_2 \times \cdots \times \mathfrak{H}_T$ in a straightforward manner.

Appendix C.3. Lagrange Multipliers Theorem

In here we recall the Lagrange multiplier theorem that we need in deriving the set of linear equations corresponding to the LS-SVM primal problem. More general results of this type are found in [2] and [34]. For $m \in \mathbb{N}_M$ and $a_m \in \mathfrak{H}$ consider the affine functional $r_m : \mathfrak{H} \to \mathbb{R}$ defined by $r_m(h) = \langle h, a_m \rangle_{\mathfrak{H}} + b_m$for some $B \in \mathbb{R}^M$. Let f and g_s, for $s \in \mathbb{N}_S$, denote convex and continuous functionals on $\mathfrak{H}$. Consider the following constrained problem:

$$\begin{array}{ll} \min_{h\in\mathfrak{H}} & f(h) \\ \text{such that} & r_m(h) = 0, \quad m \in \mathbb{N}_M \\ & g_s(h) \leq 0, \quad s \in \mathbb{N}_S \ . \end{array} \tag{C.4}$$

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 $\mathfrak{F} := \mathrm{dom}(f) \bigcap_{s\in\mathbb{N}_S} \mathrm{dom}(g_s)$ and $\mathfrak{A} := \{h \in \mathfrak{H} \ : \ r_m(h) = 0 \ \forall \ m \in \mathbb{N}_M, \ g_s(h) \leq 0 \ \forall \ s \in \mathbb{N}_S\}$. The next Theorem is a restatement of [2, Theorem 1.2 and Theorem 1.3].

Theorem 5 (Lagrange Multiplier Theorem [2])**.** *Suppose that*
1.) $g_s(h) < 0 \ \forall \ s \in \mathbb{N}_S$ *for some point* $h \in \mathfrak{A}$
2.) $0 \in \mathrm{int}\{(r_1(h), r_2(h), \ldots, r_M(h)) \ : \ h \in \mathfrak{F}\}$.
Then $h^\star \in \mathfrak{A}$ *is an optimal solution to* (C.4) *if there exist 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.) $\lambda_s^\star g_s(h^\star) = 0$.

Appendix C.4. Derivation of LS-SVM for Classification

We now base ourselves upon Theorem 5 in order to derive the optimality condition of the equality constrained problem (32):

$$\begin{array}{ll} \min\limits_{(\Psi,E,b)\in\mathrm{HSF}\times\mathbb{R}^M\times\mathbb{R}} & \frac{1}{2}\langle\Psi,\Psi\rangle_{\mathrm{HSF}} + \gamma\frac{1}{2}\sum_{m\in\mathbb{N}_M} e_m^2 \\ \text{such that} & y_m(\langle\Psi, \tilde{\phi}(X^{(m)})\rangle_{\mathrm{HSF}} + b) = 1 - e_m, \ m \in \mathbb{N}_M \ . \end{array}$$

The problem involves finding an optimal ordered pair $(\Psi^\star, E^\star, b^\star)$ in the product space $\mathrm{HSF} \times \mathbb{R}^M \times \mathbb{R}$. This space, denoted by $\mathfrak{H}$ for convenience of notation, can be turned into a HS by means of the inner product

$$\langle(\Psi, E, b), (\Xi, F, c)\rangle_{\mathfrak{H}} = \langle\Psi, \Xi\rangle_{\mathrm{HSF}} + \langle E, F\rangle + bc \ .$$

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

$$r_m((\Psi, E, b)) := \langle(\Psi, E, b), (y_m\tilde{\phi}(X^{(m)}), E^{(m)}, y_m)\rangle_{\mathfrak{H}} - 1 \tag{C.5}$$

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 (32) can be restated as

$$\min_{(\Psi,E,b)\in\mathfrak{H}} \{f((\Psi, E, b)) \ : \ r_m((\Psi, E, b)) = 0, \ m \in \mathbb{N}_M\} \ .$$

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 Appendix C.2 on composite spaces and Remark 4. By equation (C.5), Remark 3 and Appendix C.2 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 5 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 $\mathfrak{A}$ of Theorem 5 reads here $\mathfrak{A} = \{r_m((\Psi, E, b)) = 0, \ m \in \mathbb{N}_M\}$. Making $r_m((\Psi, E, b)) = 0$ explicit for $m \in \mathbb{N}_M$, we obtain the additional set of conditions:

$$y_m(\langle\Psi^\star, \tilde{\phi}(X^{(m)})\rangle_{\mathrm{HSF}} + b^\star) = 1 - e_m^\star, \ m \in \mathbb{N}_M \ . \tag{C.10}$$

Replacing (C.7) and (C.8) into the latter to eliminate the primal variable $\Psi^\star$ and $E^\star$, and keeping into account (C.9), one obtains the system of linear equations (33) 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

$$(\boldsymbol{\Omega})_{ij} = y_i y_j\langle\tilde{\phi}(X^{(i)}), \tilde{\phi}(X^{(j)})\rangle_{\mathrm{HSF}} = y_i y_j k(X^{(i)}, X^{(j)}) \ .$$