Entangled Kernels

Riikka Huusari and **Hachem Kadri**
Aix-Marseille University, CNRS, LIS, Marseille, France
{riikka.huusari, hachem.kadri}@lis-lab.fr

Abstract

We consider the problem of operator-valued kernel learning and investigate the possibility of going beyond the well-known separable kernels. Borrowing tools and concepts from the field of quantum computing, such as partial trace and entanglement, we propose a new view on operator-valued kernels and define a general family of kernels that encompasses previously known operator-valued kernels, including separable and transformable kernels. Within this framework, we introduce another novel class of operator-valued kernels called *entangled kernels* that are not separable. We propose an efficient two-step algorithm for this framework, where the entangled kernel is learned based on a novel extension of kernel alignment to operator-valued kernels. The utility of the algorithm is illustrated on both artificial and real data.

1 Introduction

There is a growing body of learning problems for which each instance in the training set is naturally associated with a set of labels (discrete and/or continuous). Output kernel learning algorithms approach these problems by learning simultaneously a vector-valued function in a reproducing kernel Hilbert space (RKHS) and a positive semidefinite matrix that describes the relationships between the labels [Dinuzzo *et al.*, 2011; Dinuzzo and Fukumizu, 2011; Ciliberto *et al.*, 2015; Jawanpuria *et al.*, 2015]. The main idea of these methods is to learn a separable operator-valued kernel.

Operator-valued kernels appropriately generalize the well-known notion of reproducing kernels and provide a means for extending the theory of reproducing kernel Hilbert spaces from scalar- to vector-valued functions. They were introduced as a machine learning tool in [Micchelli and Pontil, 2005] and have since been investigated for use in various machine learning tasks, including multi-task learning [Evgeniou *et al.*, 2005], functional regression [Kadri *et al.*, 2016], structured output prediction [Brouard *et al.*, 2016], quantile learning [Sangnier *et al.*, 2016] and multi-view learning [Minh *et al.*, 2016]. The kernel function outputs then a linear operator (a matrix in the case of finite-dimensional output spaces) which encodes information about multiple output variables. A challenging question in vector-valued learning is what sort of interactions should the operator-valued kernel learn and quantify, and how should one build and design these kernels. This is the main question investigated in the paper in the context of non-separability between input and output variables.

Some classes of operator-valued kernels have been proposed in the literature [Caponnetto *et al.*, 2008; Alvarez *et al.*, 2012], with separable kernels being one of the most widely used for learning vector-valued functions due to their simplicity and computational efficiency. These kernels are formulated as a product between a kernel function for the input space alone, and a matrix that encodes the interactions among the outputs. In order to overcome the need for choosing a kernel before the learning process, output kernel learning methods learn the output matrix from data [Ciliberto *et al.*, 2015; Dinuzzo *et al.*, 2011; Jawanpuria *et al.*, 2015]. However there are limitations in using separable kernels. These kernels use only one output matrix and one input kernel function, and then cannot capture different kinds of dependencies and correlations. Moreover the kernel matrix associated to separable kernels is a rank-one kronecker product matrix (i.e, computed by only one kronecker product), which is restrictive as it assumes a strong repetitive structure in the operator-valued kernel matrix that models input and output interactions.

To go beyond separable kernels, some attempts have been made to learn a weighted sum of them in the multiple kernel learning framework [Kadri *et al.*, 2012; Sindhwani *et al.*, 2013; Gregorová *et al.*, 2017]. Another approach, proposed by [Lim *et al.*, 2015], is to learn a combination of a separable and a transformable kernel. The form of the transformable kernel is fixed in advance but allows to encode non-separable dependencies between inputs and outputs. Despite these previous investigations, the lack of knowledge about the full potential of operator-valued kernels and how to go beyond the restrictive separable kernel clearly hampers their widespread use in machine learning and other fields.

Contributions. In the present paper, we provide a novel framework for characterizing and designing inseparable kernels. By leveraging tools from the field of quantum computing (Section 2), we introduce a novel class of kernels based on the notion of partial trace which generalizes the trace operation to block matrices. This class of partial trace kernels we propose is very broad and encompasses previously known operator-valued kernels, including separable and transformable kernels (Section 3). From the new class of partial trace kernels we derive another new class of operator-valued kernels, called *entangled* kernels, that are not separable. To our knowledge, this is the first time such operator-valued kernel categorization has been performed. For this class of kernels we develop a new algorithm called EKL (Entangled Kernel Learning) that in two steps learns a partial trace kernel function (Section 4), and a vector-valued funcion. For the first step of kernel learning, we propose a novel definition of alignment between an operator-valued kernel and labels of a multi-output learning problem. To our knowledge, this is the first proposition on how to extend alignment to context of operator-valued kernels. Our algorithm offers improvements to high computational cost usually associated with learning with general operator-valued kernels. We provide an empirical evaluation of EKL performance which demonstrates its effectiveness on artificial data as well as real benchmarks (Section 5).

Notation. We denote scalars, vectors and matrices as a, $\mathbf{a}$ and $\mathbf{A}$ respectively. The notation $\mathbf{A} \geq 0$ will be used to denote a positive semidefinite (psd) matrix. Throughout the paper we use n as the number of labeled data samples and p as the number of outputs corresponding to one data sample. We denote our set of data samples by $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is a Polish space and $\mathcal{Y}$ is a separable Hilbert space. Usually, $\mathcal{X}$ and $\mathcal{Y}$ are respectively $\mathbb{R}^d$ and $\mathbb{R}^p$ equipped with the standard Euclidean metric. We use $k(\cdot,\cdot)$ as a scalar-valued, and $K(\cdot,\cdot)$ as an operator-valued kernel function; the corresponding kernel matrices are $\mathbf{K} \in \mathbb{R}^{n\times n}$ and $\mathbf{G} \in \mathbb{R}^{np\times np}$, the latter containing blocks of size $p \times p$. We denote by $\mathcal{K}$ and $\mathcal{H}$ the reproducing kernel Hilbert spaces associated to the kernels k and K, respectively.

2 Background

In this section we give some background about quantum entanglement and review the basics of learning with operator-valued kernels.

2.1 Background on Quantum Entanglement

Quantum entanglement is a fundamental feature of quantum mechanics and quantum information. This section is not intended to provide a broad overview or exhaustive survey of the literature on quantum etanglement, but gives some notions on entanglement as quantum property of mixed composite quantum systems that inspired our entangled kernel design. We refer the reader to [Horodecki *et al.*, 2009], [Bengtsson and Życzkowski, 2017], and [Rieffel and Polak, 2011, chap. 10] for more background information.

Composite quantum systems are systems that naturally decompose into two or more subsystems, where each subsystem itself is a proper quantum system. We focus here only on *bipartite* quantum systems, i.e., systems composed of two distinct subsystems. The Hilbert space $\mathcal{F}$ associated with a bipartite quantum system is given by the tensor product $\mathcal{F}_1 \otimes \mathcal{F}_2$ of the spaces corresponding to each of the subsystems. In quantum mechanics, the state of a quantum system is represented by a state vector $\psi \in \mathcal{F}$. However, it is also possible for a system to be in a statistical ensemble of different state vectors. The sate of the quantum system in this case is called a *mixed* state. It is characterized by a density matrix ρ which in general takes the following form

$$\rho = \sum_j p_j \psi_j \psi_j^\top,$$

where the coefficients p_j are non-negative and sum to one. For a composite quantum system of two subsytems with a density matrix ρ, the state of, say, the first subsystem is described by a reduced density matrix, given by taking the *partial trace* of ρ over $\mathcal{F}_2$. In the following we review the notions of partial trace, separability and entanglement of bipartite quantum systems.

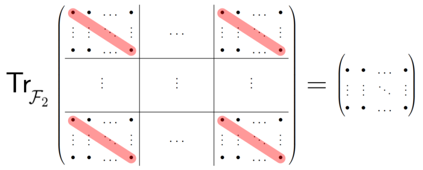


Figure 1: Illustration of partial trace operation. The partial trace operation applied to $N \times N$-blocks of a $pN \times pN$ matrix gives a $p \times p$ matrix as an output.

We denote the set of linear operators from a Hilbert space $\mathcal{B}$ to $\mathcal{B}$ as $\mathcal{L}(\mathcal{B})$. Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be separable Hilbert spaces.

Definition 1. *(partial trace)*
Partial trace operator on $\mathcal{L}(\mathcal{F}_1 \otimes \mathcal{F}_2)$ is the unique linear operator $tr_{\mathcal{F}_2} : \mathcal{L}(\mathcal{F}_1 \otimes \mathcal{F}_2) \to \mathcal{L}(\mathcal{F}_1)$ such that $tr_{\mathcal{F}_2}(\mathbf{A} \otimes \mathbf{B}) = \mathbf{A}\, tr(\mathbf{B})$, $\forall\, \mathbf{A} \in \mathcal{L}(\mathcal{F}_1)$, $\mathbf{B} \in \mathcal{L}(\mathcal{F}_2)$.

In finite dimensions, elements in $\mathcal{L}(\mathcal{A})$ and $\mathcal{L}(\mathcal{A} \otimes \mathcal{B})$ are simply matrices and block matrices of some sizes $p \times p$ and $pN \times pN$, and the partial trace is obtained by computing the trace of each block in the input matrix (see Figure 1). The notion of *partial trace* is a generalization of the trace operation to block structured matrices [Rieffel and Polak, 2011, chap. 10]. Note that there are two ways of generalizing trace to block matrices. Another possiblity would be so-called block trace [Filipiak *et al.*, 2018] where the result is sum of diagonal blocks, but in this work we only consider the "blockwise trace" definition.

In the case where the density matrix ρ of a mixed bipartite state can be written as $\rho = \rho_1 \otimes \rho_2$, where ρ_1 and ρ_2 are subsystems' density matrices on $\mathcal{F}_1$ and $\mathcal{F}_2$, the partial trace of ρ with respect to $\mathcal{F}_2$ is ρ_1. This form of mixed product states is restrictive and does not exhibit correlations between the two subsystems. A convex sum of different product states,

$$\rho = \sum_i p_i \rho_1^i \otimes \rho_2^i, \tag{1}$$

with $p_i \geq 0$ and $\sum_i p_i = 1$, however, will in general represent certain types of correlations between the subsystems of the composite quantum system. These correlations can be described in terms of the classical probabilities p_i, and are therefore considered classical. States of the form (1) thus are called *separable* mixed states. In contrast, a mixed state is *entangled* if it cannot be written as a convex combination of product states, i.e.,

$$\nexists\, \rho_1^i, \rho_2^i, p_i \geq 0 \quad \text{such that} \quad \rho = \sum_i p_i \rho_1^i \otimes \rho_2^i. \qquad (2)$$

Entangled states are one of the most commonly encountered class of bipartite states possessing quantum correlations [Mintert *et al.*, 2009]. A challenging problem in quantum computing is to identify necessary and suffcient conditions for quantum separability. There is no practically efficient necessary and suffcient criteria for identifying whether a given ρ is entangled or separable [Horodecki *et al.*, 2009].

2.2 Learning with Operator-Valued Kernels

We now review the basics of operator-valued kernels (OvKs) and their associated vector-valued reproducing kernel Hilbert spaces (RKHSs) in the setting of supervised learning. Vector-valued RKHSs were introduced to the machine learning community by [Micchelli and Pontil, 2005] as a way to extend kernel machines from scalar to vector outputs. Given a set of training samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ on $\mathcal{X} \times \mathcal{Y}$, optimization problem

$$\arg\min_{f \in \mathcal{H}} \sum_{i=1}^n V(f, \mathbf{x}_i, \mathbf{y}_i) + \lambda \|f\|_{\mathcal{H}}^2, \qquad (3)$$

where f is a vector-valued function and V is a loss function, can be solved in a vector-valued RKHS $\mathcal{H}$ by the means of a vector-valued extension of the representer theorem.

Definition 2. *(vector-valued RKHS)*

A Hilbert space $\mathcal{H}$ of functions from $\mathcal{X}$ to $\mathcal{Y}$ is called a reproducing kernel Hilbert space if there is a positive definite $\mathcal{L}(\mathcal{Y})$-valued kernel K on $\mathcal{X} \times \mathcal{X}$ such that:

i. the function $z \mapsto K(\mathbf{x}, \mathbf{z})\mathbf{y}$ belongs to $\mathcal{H}$, $\forall\ \mathbf{z}, \mathbf{x} \in \mathcal{X},\ \mathbf{y} \in \mathcal{Y}$,

ii. $\forall f \in \mathcal{H}, \mathbf{x} \in \mathcal{X},\ \mathbf{y} \in \mathcal{Y},\ \langle f, K(\mathbf{x}, \cdot)\mathbf{y}\rangle_{\mathcal{H}} = \langle f(\mathbf{x}), \mathbf{y}\rangle_{\mathcal{Y}}$ (reproducing property).

Definition 3. *(operator-valued kernel)*

A $\mathcal{L}(\mathcal{Y})$-valued kernel K on $\mathcal{X} \times \mathcal{X}$ is a function $K(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$; it is positive semidefinite if:

i. $K(\mathbf{x}, \mathbf{z}) = K(\mathbf{z}, \mathbf{x})^*$, where superscript * denotes the adjoint operator,

ii. and, for every $r \in \mathbb{N}$ and all $\{(\mathbf{x}_i, \mathbf{y}_i)_{i=1,\ldots,r}\} \in \mathcal{X} \times \mathcal{Y}$, $\sum_{i,j} \langle \mathbf{y}_i, K(\mathbf{x}_i, \mathbf{x}_j)\mathbf{y}_j\rangle_{\mathcal{Y}} \geq 0$.

Theorem 1. *(bijection between vector-valued RKHS and operator-valued kernel) An $\mathcal{L}(\mathcal{Y})$-valued kernel K on $\mathcal{X} \times \mathcal{X}$ is the reproducing kernel of some Hilbert space $\mathcal{H}$, if and only if it is positive semidefinite.*

Theorem 2. *(representer theorem)*

Let K be a positive semidefinite operator-valued kernel and $\mathcal{H}$ its corresponding vector-valued RKHS. The solution $\hat{f} \in \mathcal{H}$ of the regularized optimization problem (3) has the form

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^n K(\mathbf{x}, \mathbf{x}_i)\mathbf{c}_i, \quad \textit{with} \quad \mathbf{c}_i \in \mathcal{Y}. \qquad (4)$$

With regard to the classical representer theorem, here the kernel K outputs a matrix and the “weights” $\mathbf{c}_i$ are vectors. The proofs of Theorem 1 and 2 can be found in [Micchelli and Pontil, 2005; Kadri *et al.*, 2016]. For further reading on operator-valued kernels and their associated RKHSs, see, e.g., [Caponnetto *et al.*, 2008; Carmeli *et al.*, 2010].

3 Partial Trace and Entangled Kernels

Some well-known classes of operator-valued kernels include separable and transformable kernels. Separable kernels are defined by

$$K(\mathbf{x}, \mathbf{z}) = k(\mathbf{x}, \mathbf{z})\mathbf{T}, \quad \forall\, \mathbf{x}, \mathbf{z} \in \mathcal{X}, \qquad (5)$$

where $\mathbf{T}$ is a psd matrix in $\mathbb{R}^{p \times p}$ (an operator in $\mathcal{L}(\mathcal{Y})$ in the general case of any separable Hilbert space $\mathcal{Y}$) and k is a scalar-valued kernel. This class of kernels is very attractive in terms of computational time, as it is easily decomposable. However the matrix $\mathbf{T}$ acts only on the outputs independently of the input data, which makes it difficult for these kernels to capture input-output relations. In the same spirit a more general class, sum of separable kernels, is given by

$$K(\mathbf{x}, \mathbf{z}) = \sum_l k_l(\mathbf{x}, \mathbf{z})\mathbf{T}_l, \quad \forall\, \mathbf{x}, \mathbf{z} \in \mathcal{X}, \qquad (6)$$

where k_l are a scalar-valued kernels and $\mathbf{T}_l \in \mathbb{R}^{p \times p}$ are psd. It can capture different kinds of similarities but still assumes that the unknown input-output dependencies can be decomposed into a product of two separate kernel functions that encode interactions among inputs and outputs independently.

Transformable kernels are defined by

$$K(\mathbf{x}, \mathbf{z}) = \Big[\widetilde{k}(S_m\mathbf{x}, S_l\mathbf{z})\Big]_{l,m=1}^p, \quad \forall\, \mathbf{x}, \mathbf{z} \in \mathcal{X}. \qquad (7)$$

Here m and l are indices of the output matrix and $\{S_t\}_{t=1}^p$ are mappings which transform the data from $\mathcal{X}$ to another space $\widetilde{\mathcal{X}}$ in which a scalar-valued kernel $\widetilde{k} : \widetilde{\mathcal{X}} \times \widetilde{\mathcal{X}} \to \mathbb{R}$ is defined. In contrast to separable kernels, here the mappings S_t operate on input data while dependening on outputs; however they are not intuitive nor easy to interpret and determine.

While it is straightforward to see that separable kernels belong to the larger class of sum of separable, the picture is less clear for transformable kernels. The following examples clarify this situation.

Example 1. *On the space $\mathcal{X} = \mathbb{R}$, consider the kernel*

$$K(\mathbf{x}, \mathbf{z}) = \begin{pmatrix} \mathbf{xz} & \mathbf{xz}^2 \\ \mathbf{x}^2\mathbf{z} & \mathbf{x}^2\mathbf{z}^2 \end{pmatrix}, \quad \forall\, \mathbf{x}, \mathbf{z} \in \mathcal{X}.$$

K is a transformable kernel, but not a (sum of) separable kernel. We obtain that K is transformable simply by choosing $\widetilde{k}(\mathbf{x}, \mathbf{z}) = \mathbf{xz}$, $S_1(\mathbf{x}) = \mathbf{x}$, and $S_2(\mathbf{x}) = \mathbf{x}^2$ in Eq. 7. From the property of positive definiteness of the operator-valued kernel, it is easy to see that the matrix $\mathbf{T}$ of a separable kernel is symmetric (see Eq. 5), and since the matrix $K(\mathbf{x}, \mathbf{z})$ is not, K is not a separable kernel.

Example 2. *Let K be the kernel function defined as*

$$K(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z}\rangle \mathbf{T}, \quad \forall\, \mathbf{x}, \mathbf{z} \in \mathcal{X},$$

where $\mathbf{T} \in \mathbb{R}^{p \times p}$ is a rank one positive semidefinite matrix. K is both separable and transformable kernel. Since $\mathbf{T}$ is of rank one, it follows that $\big(K(\mathbf{x}, \mathbf{z})\big)_{lm} = \mathbf{u}_l\mathbf{u}_m\langle \mathbf{x}, \mathbf{z}\rangle$, with $\mathbf{T} = \mathbf{u}\mathbf{u}^\top$. We can see that K is transformable by replacing in Eq. 7 $\widetilde{k}(\mathbf{x}, \mathbf{z})$ by $\langle \mathbf{x}, \mathbf{z}\rangle$ and $S_t(\mathbf{x})$ by $\mathbf{u}_t\mathbf{x}$, $t = 1, \ldots, p$. K is separable by construction.

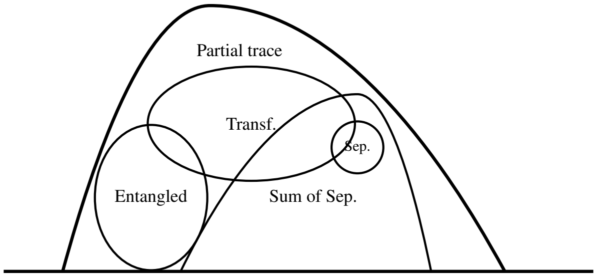


Figure 2: Illustration of inclusions among various operator-valued kernel classes.

It is worth noting that separable kernels are not limited to finite-dimensional output spaces, while transformable kernels are. Figure 2 depicts inclusions among kernel classes discussed here and the two new families of operator-valued kernels we propose: partial trace kernels and entangled kernels.

We now define two novel classes of operator-valued kernels. The first one, class of partial trace kernels, encompasses both (sum of) separable and transformable kernels, while the second, entangled kernels, is a class of non-separable kernels. We start by introducing the more general class of partial trace kernels. The intuition behind this class of kernels is that in the scalar-valued case any kernel function k can be written as the trace of an operator in $\mathcal{L}(\mathcal{K})$. It is easy to show that $k(\mathbf{x},\mathbf{z}) = tr\big(\phi(\mathbf{x}) \otimes \phi(\mathbf{z})^\top\big)$.

Definition 4. *(Partial trace kernel)*
A partial trace kernel is an operator-valued kernel function K having the following form

$$K(\mathbf{x},\mathbf{z}) = tr_{\mathcal{K}}(\mathbf{P}_{\phi(\mathbf{x}),\phi(\mathbf{z})}), \tag{8}$$

where $\mathbf{P}_{\phi(\mathbf{x}),\phi(\mathbf{z})}$ is an operator on $\mathcal{L}(\mathcal{Y}\otimes\mathcal{K})$, and $tr_{\mathcal{K}}$ is the partial trace on $\mathcal{K}$ (i.e., over the inputs).

The class of partial trace kernels is very broad and encompasses the classes of separable and transformable kernels (see Figure 2). From the definition of the partial trace operation, we can see that if we choose $\mathbf{P}_{\phi(\mathbf{x}),\phi(\mathbf{z})} = \sum_l \mathbf{T}_l \otimes \big(\phi_l(\mathbf{x}) \otimes \phi_l(\mathbf{z})^\top\big)$, we recover the case of sum of separable kernels. In the same way, if we fix $[\mathbf{P}_{\widetilde{\phi}(\mathbf{x}),\widetilde{\phi}(\mathbf{z})}]_{l,m=1}^{p} = \big(\widetilde{\phi}\circ S_l(\mathbf{x})\big) \otimes \big(\widetilde{\phi}\circ S_m(\mathbf{z})\big)^\top$ in Eq. 8, computing the trace of each block using the partial trace will give the transformable kernel. With this in mind, we can use the partial trace kernel formulation to induce a novel class for operator-valued kernels which are not separable, with the goal to characterize inseparable correlations between inputs and outputs.

Definition 5. *(Entangled kernel)*
An entangled operator-valued kernel K is defined as

$$K(\mathbf{x},\mathbf{z}) = tr_{\mathcal{K}}\left(\mathbf{U}\big(\mathbf{T}\otimes(\phi(\mathbf{x})\otimes\phi(\mathbf{z})^\top)\big)\mathbf{U}^\top\right), \tag{9}$$

where $\mathbf{T}$ is of size $p\times p$, and $\mathbf{U}\in\mathbb{R}^{pN\times pN}$ is not separable.

Here we have abused the notation by introducing N as the dimensionality of feature representation $\phi(\mathbf{x})$. However we do not restrict ourselves to finite dimensions and N in this notation can also be infinite. In this definition, $\mathbf{U}$ not being separable means that it cannot be written as $\mathbf{U} = \mathbf{A}\otimes\mathbf{B}$, with $\mathbf{A}\in\mathbb{R}^{p\times p}$ and $\mathbf{B}\in\mathbb{R}^{N\times N}$. The term $\mathbf{T}\otimes(\phi(\mathbf{x})\otimes\phi(\mathbf{z})^\top)$ represents a separable kernel function over inputs and outputs, while $\mathbf{U}$ characterizes the entanglement shared between them. Some intuition to $\mathbf{U}$ can be seen from its role of an "entangled" similarity in the joint feature space. It is entangled in the sense that cannot be decomposed into two "sub"-matrices of similarity between inputs and between outputs independently. The partial trace is the operation used to recover the sub-similarity matrix between the outputs from the entangled joint similarity matrix. In the particular case of separability, the partial trace will give the output metric. Choice of $\mathbf{U}$ is crucial to this class of kernels. In the next section we develop an algorithm that learns an entangled kernel from data.

4 Entangled Kernel Learning

In general, there is no knowing whether input and output data are or are not entangled. In this sense, learning the entangled K in Eq. 9 by imposing that $\mathbf{U}$ is inseparable can sometimes be restrictive. In our entangled kernel learning approach we do not impose any separability restriction, with the hope that our learning algorithm can automatically detect the lack or presence of entanglement. Key to our method is a reformulation of the entangled kernel K (Eq. 9) via Choi-Kraus representation. The infinite case is less treated in literature, and for it we refer reader to [Attal, 2015].

Theorem 3. *(Choi-Kraus representation [Choi, 1975; Kraus, 1983; Rieffel and Polak, 2011])*
The map $K(\mathbf{x},\mathbf{z}) = tr_{\mathcal{K}}\left(\mathbf{U}\big(\mathbf{T}\otimes(\phi(\mathbf{x})\otimes\phi(\mathbf{z})^\top)\big)\mathbf{U}^\top\right)$ can be generated by an operator sum representation containing at most pN elements,

$$K(\mathbf{x},\mathbf{z}) = \sum_{i=1}^{r} \mathbf{M}_i\phi(\mathbf{x})\phi(\mathbf{z})^\top\mathbf{M}_i^\top, \tag{10}$$

where $\mathbf{M}_i\in\mathbb{R}^{p\times N}$ and $1\le r\le pN$.

It is easy to see that our kernel is positive. Using this formulation, entangled kernel learning consists of finding a low-rank decomposition of the kernel by learning the matrices $\mathbf{M}_i$, $i=1,\ldots,r$ ($r\ll pN$), which "merge" the matrices $\mathbf{T}$ and $\mathbf{U}$. While every entangled kernel can be represented like this, the representation is not restricted only to entangled kernels. Thus by learning the $\mathbf{M}_i$ we expect to learn the meaningful relationships in the data, be they entangled or not.

Because the feature space can easily be of very large dimensionality (or infinite-dimensional), we consider an approximation to speed up the computation. For example random Fourier features or Nyström approximation [Rahimi and Recht, 2008; Williams and Seeger, 2001] give us $\hat{\phi}$ such that $k(x,z) = \langle\phi(x),\phi(z)\rangle \approx \langle\hat{\phi}(x),\hat{\phi}(z)\rangle$. We note that our approximation is on scalar-valued kernel, not operator-valued, although there are methods for approximating them, too, directly [Brault *et al.*, 2016; Minh, 2016]. Our approximated kernel is thus

$$\hat{K}(\mathbf{x},\mathbf{z}) = \sum_{i=1}^{r} \hat{\mathbf{M}}_i\hat{\phi}(\mathbf{x})\hat{\phi}(\mathbf{z})^\top\hat{\mathbf{M}}_i^\top, \tag{11}$$

where $\hat{\phi}(\mathbf{x}) \in \mathbb{R}^m$ and $\hat{\mathbf{M}}_i \in \mathbb{R}^{p\times m}$, from where our goal is to learn the $\hat{\mathbf{M}}_i$. We can write our $np \times np$ kernel matrix as

$$\begin{aligned}\hat{\mathbf{G}} &= \sum_{i=1}^{r} \operatorname{vec}(\hat{\mathbf{M}}_i\hat{\mathbf{\Phi}}) \operatorname{vec}(\hat{\mathbf{M}}_i\hat{\mathbf{\Phi}})^\top \qquad (12)\\ &= \sum_{i=1}^{r} (\hat{\mathbf{\Phi}}^\top \otimes \mathbf{I}_p) \operatorname{vec}(\hat{\mathbf{M}}_i) \operatorname{vec}(\hat{\mathbf{M}}_i)^\top (\hat{\mathbf{\Phi}} \otimes \mathbf{I}_p)\end{aligned}$$ [1]

where $\hat{\mathbf{\Phi}} = [\hat{\phi}(\mathbf{x}_1), \cdots, \hat{\phi}(\mathbf{x}_n)]$ is of size $m \times n$. Further, if we denote $\mathbf{D} = \sum_{i=1}^{r} \operatorname{vec}(\hat{\mathbf{M}}_i) \operatorname{vec}(\hat{\mathbf{M}}_i)^\top$, we can write

$$\hat{\mathbf{G}} = (\hat{\mathbf{\Phi}}^\top \otimes \mathbf{I}_p)\mathbf{D}(\hat{\mathbf{\Phi}} \otimes \mathbf{I}_p).$$

To learn an entangled kernel, we need to learn the psd matrix $\mathbf{D}$. We adopt kernel alignment -based kernel learning strategy introduced in [Cristianini *et al.*, 2002; Cortes *et al.*, 2010] for scalar-valued kernels in the setting when every input was associated with only one output, or label. Alignment between two matrices $\mathbf{M}$ and $\mathbf{N}$ is defined as

$$A(\mathbf{M}, \mathbf{N}) = \frac{\langle \mathbf{M}_c, \mathbf{N}_c\rangle_F}{\|\mathbf{M}_c\|_F \|\mathbf{N}_c\|_F} \qquad (13)$$

where subscript c refers to centered matrices. We extend the concept of alignment into case of multiple outputs and consider a convex combination of two such alignments. Namely our optimization problem is

$$\max_{\mathbf{D}} \,(1-\gamma)\, A\left(tr_p(\hat{\mathbf{G}}), \mathbf{Y}^\top\mathbf{Y}\right) + \gamma\, A\left(\hat{\mathbf{G}}, \mathbf{y}\mathbf{y}^\top\right) \qquad (14)$$

where $\gamma \in [0, 1]$, $\mathbf{y} = \operatorname{vec}(\mathbf{Y})$, and $\mathbf{Y}$ is of size $p \times n$, containing the labels associated to data sample i on its ith column. We note that by applying Lemma 2.11 from [Filipiak *et al.*, 2018], we can write $tr_p(\hat{\mathbf{G}}) = \hat{\mathbf{\Phi}}^\top tr_p(\mathbf{D})\hat{\mathbf{\Phi}}$.

Intuitively the first alignment learns a scalar-valued kernel matrix that can be obtained via partial trace applied to the more complex operator-valued kernel, while the second term focuses on the possibly entangled relationships in the data. Indeed, one possibility for using the entangled kernel framework is to learn a scalar-valued kernel for multi-output problem and to use this kernel in machine learning algorithms.

The optimization problem is solved with gradient-based approach. To make sure that the resulting kernel is valid (psd), we write $\mathbf{D} = \mathbf{Q}\mathbf{Q}^\top$ with $\mathbf{Q}$ of size $mp \times r$ with r at most mp, and perform the optimization over $\mathbf{Q}$. The gradients for alignment terms are straightforward to calculate. The optimization is performed on sphere manifold as a way to regularize $\mathbf{D}$.[2] After we have learned the entangled kernel, we solve the learning problem by choosing the squared loss

$$\min_{\mathbf{c}} \|\operatorname{vec}(\mathbf{Y}) - \hat{\mathbf{G}}\mathbf{c}\|^2 + \lambda\langle\hat{\mathbf{G}}\mathbf{c}, \mathbf{c}\rangle. \qquad (15)$$

For this $\mathbf{c}$ update we can find the classical closed-form solution, $\mathbf{c} = (\hat{\mathbf{G}} + \lambda\mathbf{I})^{-1}\operatorname{vec}(\mathbf{Y})$. Note that this computation is, by considering the entangled structure of $\hat{\mathbf{G}}$, more computationally efficient than a general (say, some transfomable) operator-valued kernel. Generally we can say that the complexity of predicting with nonseparable operator-valued kernels is $\mathcal{O}(n^3p^3)$. In our proposed network, however, we can apply Woodbury formula for the matrix inversion and only invert a $r \times r$ matrix, giving total complexity of $\mathcal{O}(r^3 + mnp^2)$. Moreover it is possible in our kernel learning framework to extract a scalar-valued kernel, $\mathbf{K} = tr_p(\hat{\mathbf{G}})$, and use that in predicting with traditional cost of $\mathcal{O}(n^3)$, or $\mathcal{O}(m^3 + mnp)$ if taken advantage of the form of the entangled kernel. The parameters γ and λ can be set with cross-validation, first caculating the kernel learning step with various γ parameters and then considering the various λ for each of them.

Algorithm 1 Entangled Kernel Learning (EKL)

```
Input: matrix of features $\mathbf{\Phi}$, labels $\mathbf{Y}$
// 1) Kernel learning: ($\mathbf{D} = \mathbf{Q}\mathbf{Q}^\top$)
Solve for $\mathbf{Q}$ in eq.14 within a sphere manifold
// 2) Learning the predictive function:
if Predict with scalar-valued kernel then
    $\mathbf{c}_K = (tr_p(\hat{\mathbf{G}}) + \lambda\mathbf{I})^{-1}\mathbf{Y}^\top$        $\mathcal{O}(m^3 + mnp)$
else
    $\mathbf{c}_G = (\hat{\mathbf{G}} + \lambda\mathbf{I})^{-1}\operatorname{vec}(\mathbf{Y})$        $\mathcal{O}(r^3 + mnp^2)$
Return $\mathbf{D} = \mathbf{Q}\mathbf{Q}^\top$, $\mathbf{c}$
```

5 Experiments

In this section the performance of our algorithm is illustrated with artificial and real datasets. The algorithms compared in these settings are: **EKL**[3]; our proposed Entangled Kernel Learning algorithm, **OKL** [Dinuzzo *et al.*, 2011]; a kernel learning method for separable kernels (we use the code provided by the authors[4]), and **KRR**; kernel ridge regression. Furhtermore, we investigate performance of predicting with scalar-valued kernel extracted from the operator-valued kernel matrix EKL learns, and call this **ptrEKL**. In all the experiments we cross-validate over various regularization parameters λ, and for EKL also γs controlling the combination of alignments. In the experiments we consider (normalized) mean squared error (nMSE) and normalized improvement to KRR (nI) [Ciliberto *et al.*, 2015] as error measures.

5.1 Artificial Data

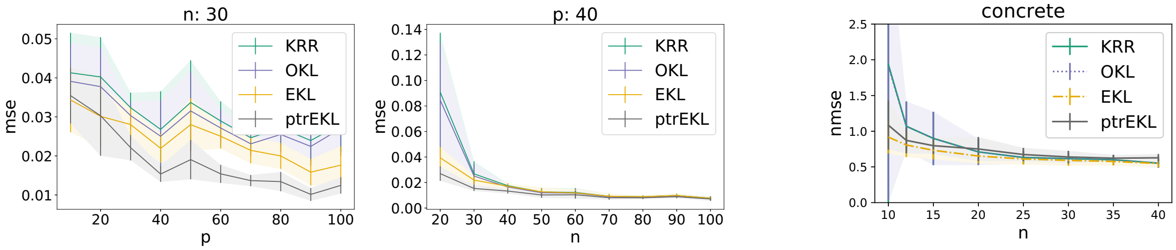
EKL is expected to learn complex relationships within the data. To illustrate this, we created data with bi-linear model $\mathbf{TCA} + \mathbf{ICK} = \mathbf{Y}$, where $\mathbf{T}$, $\mathbf{C}$ and $\mathbf{A}$ are randomly created $p \times p$, $p \times n$, and $n \times n$ matrices respectively. $\mathbf{K}$ is linear kernel calculated from randomly generated data $\mathbf{X} \in \mathbb{R}^{n\times d}$; this kernel is given to the learning algorithms along with noisy labels $\mathbf{Y}$. We can see that when p is larger than n (or comparable) the predictive capabilities of EKL are much better than for other methods. Here predicting with the scalar-valued kernel extracted form learned entangled kernel gives the best results (Figure 3(a)).

[1] [Petersen and Pedersen, 2008, Eq. 520]

[2] www.manopt.org/ ; pymanopt.github.io/

[3] The code will be made available at RH's personal webpage

[4] http://people.tuebingen.mpg.de/fdinuzzo/okl.html



(a) Results of the simulated experiments with fixed amount of inputs and varying number of outputs (left), and fixed amount of outputs and varying inputs (right).

(b) Results on Concrete data set with varying amount of training data (n) used.

Figure 3: Results on simulated (left) and Concrete dataset (right). The advantage of learning complex relationships is the biggest on small n.

	$n = 50$		$n = 100$		$n = 200$		$n = 1000$	
method	nMSE	nI	nMSE	nI	nMSE	nI	nMSE	nI
KRR	0.2418 ± 0.0281	0.0000	0.1668 ± 0.0097	0.0000	0.1441 ± 0.0037	0.0000	0.1273 ± 0.0006	0.0000
OKL	0.2445 ± 0.0296	-0.0109	0.1672 ± 0.0099	-0.0026	0.1442 ± 0.0037	-0.0009	0.1273 ± 0.0006	-0.0000
EKL/ptrEKL	0.2381 ± 0.0250	0.0139	0.1661 ± 0.0097	0.0040	0.1440 ± 0.0037	0.0003	0.1273 ± 0.0006	0.0001

Table 1: Results on Sarcos dataset with various number of training samples used, averaged over 10 data partitions. The advantage of learning complex relationships decreases with amount of data samples increasing.

	$n = 5$		$n = 10$	
method	nMSE	nI	nMSE	nI
KRR	0.951 ± 0.101	0.000	0.813 ± 0.141	0.000
OKL	1.062 ± 0.250	-0.092	0.900 ± 0.196	-0.094
EKL/ptrEKL	0.840 ± 0.084	0.124	0.722 ± 0.036	0.107

Table 2: Results on Weather data set averaged over 5 data partitions.

5.2 Real Data

We have considered the following regression data sets: **Concrete slump test** (UCI dataset repository) with 103 data samples and three output variables; **Sarcos**[5] is a dataset characterizing robot arm movements with 7 tasks; **Weather**[6] has daily weather data ($p = 365$) from 35 stations.

The main advantage of learning complex dependencies in the data lies in the setting where number of samples is relatively low; a phenomenon observed already in output kernel learning setting [Ciliberto *et al.*, 2015; Jawanpuria *et al.*, 2015]. With small amounts of data learning the complex relationships in EKL is even more beneficial than learning the output dependencies of OKL. Figure 3(b) shows this advantage on Concrete data set when number of instances used in training is small. Here, in contrast to our simulated data, EKL performs better than ptrEKL. For Sarcos data set we consider the setting in [Ciliberto *et al.*, 2015] and show the results in Table 1 (predicting is done to all 5000 test samples). Similarly, we observe that the advantages of using EKL diminish with more data samples added to the problem. This is also clearly seen in the Weather data set, where number of outputs is much larger than the number of data samples (Table 2).

6 Conclusion

In this work we shed new light on meaning of inseparable kernels by defining a general framework for constructing operator-valued kernels based on the notion of partial trace and using ideas borrowed from the field of quantum computing. Instances of our framework include entangled kernels, a new conceptually interesting class of kernels that is designed to capture more complex dependencies between input and output variables as the more restricted class of separable kernels. We have proposed a new algorithm, entangled kernel learning (EKL), that learns this entangled kernel and a vector- or scalar-valued function in two steps. The first step that learns the entangled kernel uses a definition of kernel alignment, extended here for use with operator-valued kernels with help of partial trace operator. In contrast to output kernel learning, EKL is able to learn inseparable kernels and can model a larger variety of interactions between input and output data. Moreover, the structure of the entangled kernels enables more efficient computation than that with general operator-valued kernels. Our illustration on artificial data and experiments on real data give validation to our approach.

In the future work the potential of EKL should be investigated further, especially the effect of very small number of columns in matrix $\mathbf{Q}$, in low rank kernel setting. The two-step kernel learning is proven to produce good predictors with alignment to ideal kernel [Cortes *et al.*, 2010]. It is reasonable to expect that some similar guarantees could be formulated also for our setting, as it is provably effective in practice.

Acknowledgements

We would like to thank F. Denis, P. Arrighi and G. Di Molfetta for helpful discussions. This work is granted by Lives project (ANR-15-CE23-0026).

[5] www.gaussianprocess.org/gpml/data/

[6] https://www.psych.mcgill.ca/misc/fda/

References

[Alvarez *et al.*, 2012] Mauricio A. Alvarez, Lorenzo Rosasco, Neil D. Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012.

[Attal, 2015] Stephane Attal. Lectures in quantum noise theory. Lecture 6: Quantum channels, 2015.

[Bengtsson and Życzkowski, 2017] Ingemar Bengtsson and Karol Życzkowski. *Geometry of quantum states: an introduction to quantum entanglement*. Cambridge university press, 2017.

[Brault *et al.*, 2016] Romain Brault, Markus Heinonen, and Florence Buc. Random Fourier features for operator-valued kernels. In *ACML*, pages 110–125, 2016.

[Brouard *et al.*, 2016] Céline Brouard, Marie Szafranski, and Florence d'Alché Buc. Input output kernel regression: Supervised and semi-supervised structured output prediction with operator-valued kernels. *JMLR*, 17(176):1–48, 2016.

[Caponnetto *et al.*, 2008] Andrea Caponnetto, Charles A. Micchelli, Massimiliano Pontil, and Yiming Ying. Universal multi-task kernels. *JMLR*, 9(Jul):1615–1646, 2008.

[Carmeli *et al.*, 2010] Claudio Carmeli, Ernesto De Vito, Alessandro Toigo, and Veronica Umanita. Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications*, 08(01):19–61, 2010.

[Choi, 1975] Man-Duen Choi. Completely positive linear maps on complex matrices. *Linear algebra and its applications*, 10(3):285–290, 1975.

[Ciliberto *et al.*, 2015] Carlo Ciliberto, Youssef Mroueh, Tomaso Poggio, and Lorenzo Rosasco. Convex learning of multiple tasks and their structure. In *ICML*, 2015.

[Cortes *et al.*, 2010] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Two-stage learning kernel algorithms. In *ICML*, pages 239–246, 2010.

[Cristianini *et al.*, 2002] Nello Cristianini, John Shawe-Taylor, Andre Elisseeff, and Jaz S Kandola. On kernel-target alignment. In *NIPS*, pages 367–373, 2002.

[Dinuzzo and Fukumizu, 2011] Francesco Dinuzzo and Kenji Fukumizu. Learning low-rank output kernels. In *ACML*, 2011.

[Dinuzzo *et al.*, 2011] Francesco Dinuzzo, Cheng S. Ong, Gianluigi Pillonetto, and Peter V Gehler. Learning output kernels with block coordinate descent. In *ICML*, 2011.

[Evgeniou *et al.*, 2005] Theodoros Evgeniou, Charles A. Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *JMLR*, 6:615–637, 2005.

[Filipiak *et al.*, 2018] Katarzyna Filipiak, Daniel Klein, and Erika Vojtková. The properties of partial trace and block trace operators of partitioned matrices. 33, 2018.

[Gregorová *et al.*, 2017] Magda Gregorová, Alexandros Kalousis, and Stéphane Marchand-Maillet. Forecasting and granger modelling with non-linear dynamical dependencies. In *ECML-PKDD*, 2017.

[Horodecki *et al.*, 2009] Ryszard Horodecki, Paweł Horodecki, Michał Horodecki, and Karol Horodecki. Quantum entanglement. *Reviews of modern physics*, 81(2):865, 2009.

[Jawanpuria *et al.*, 2015] Pratik Jawanpuria, Maksim Lapin, Matthias Hein, and Bernt Schiele. Efficient output kernel learning for multiple tasks. In *NIPS*, 2015.

[Kadri *et al.*, 2012] Hachem Kadri, Alain Rakotomamonjy, Philippe Preux, and Francis R Bach. Multiple operator-valued kernel learning. In *NIPS*, 2012.

[Kadri *et al.*, 2016] Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Alain Rakotomamonjy, and Julien Audiffren. Operator-valued kernels for learning from functional response data. *JMLR*, 16:1–54, 2016.

[Kraus, 1983] Karl Kraus. *States, effects and operations: fundamental notions of quantum theory*. Springer, 1983.

[Lim *et al.*, 2015] Néhémy Lim, Florence d'Alché Buc, Cédric Auliac, and George Michailidis. Operator-valued kernel-based vector autoregressive models for network inference. *Machine learning*, 99(3):489–513, 2015.

[Micchelli and Pontil, 2005] Charles A. Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005.

[Minh *et al.*, 2016] Ha Quang Minh, Loris Bazzani, and Vittorio Murino. A unifying framework in vector-valued reproducing kernel hilbert spaces for manifold regularization and co-regularized multi-view learning. *JMLR*, 17(25):1–72, 2016.

[Minh, 2016] Ha Quang Minh. Operator-valued Bochner theorem, Fourier feature maps for operator-valued kernels, and vector-valued learning. *arXiv preprint arXiv:1608.05639*, 2016.

[Mintert *et al.*, 2009] F Mintert, C Viviescas, and A Buchleitner. Basic concepts of entangled states. In *Entanglement and Decoherence*, pages 61–86. Springer, 2009.

[Petersen and Pedersen, 2008] Kaare Brandt Petersen and Michael Syskind Pedersen. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[Rahimi and Recht, 2008] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2008.

[Rieffel and Polak, 2011] Eleanor G. Rieffel and Wolfgang H. Polak. *Quantum computing: A gentle introduction*. MIT Press, 2011.

[Sangnier *et al.*, 2016] Maxime Sangnier, Olivier Fercoq, and Florence d'Alché Buc. Joint quantile regression in vector-valued rkhss. In *NIPS*, pages 3693–3701, 2016.

[Sindhwani *et al.*, 2013] Vikas Sindhwani, Minh Ha Quang, and Aurélie C. Lozano. Scalable matrix-valued kernel learning for high-dimensional nonlinear multivariate regression and granger causality. In *UAI*, 2013.

[Williams and Seeger, 2001] Christopher KI Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *NIPS*, pages 682–688, 2001.