Entangled Kernels

Riikka Huusari, Hachem Kadri

QARMA - LIS, CNRS, Aix-Marseille University, France

Context

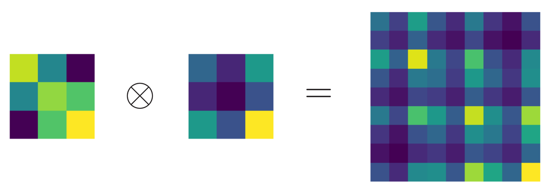
We consider the problem of multi-output learning in context of kernel methods and operator-valued kernel learning.

Multi-output learning

- Outputs of learning task are vectors, $\mathbf{y}_i \in \mathbb{R}^p$
- Operator-valued kernels learn vector-valued functions, and offer a natural solution to multi-output learning.

Kernel learning

- Motivation: how to choose a good kernels? Kernel learning tries to find suitable kernel based on data instead of fixing it in advance.
- Learning separable operator-valued kernels is common but restrictive:
  - all similarities share the structure

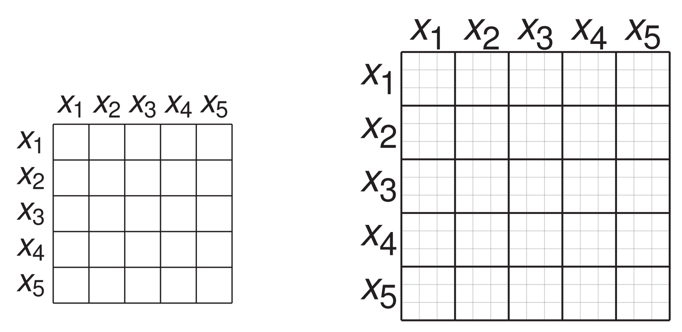


  - only symmetric interactions allowed

- Is there a way to learn unseparable kernels that model more complex dependencies between input and output variables?

Operator-valued kernels

Comparison of scalar- and operator-valued kernels

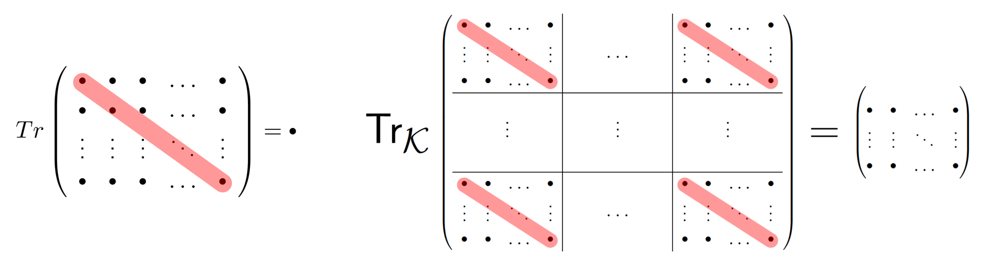


	scalar-valued	operator-valued
target function	$\mathcal{K} \ni f : \mathcal{X} \to \mathcal{Y} \in \mathbb{R}$	$\mathcal{H} \ni f : \mathcal{X} \to \mathcal{Y} \in \mathbb{R}^p$
kernel function	$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{p\times p}$
kernel trick	$k(x, x') = \langle \phi(x), \phi(x')\rangle_{\mathcal{K}}$	$\langle K(x, x')z, z'\rangle_{\mathcal{Y}} = \langle \phi(x)z, \phi(x')z'\rangle_{\mathcal{H}}\ \forall z, z' \in \mathcal{Y}$
representer theorem	$f(x) = \sum_i \alpha_i k(x_i, x)$ $\forall \alpha_i \in \mathbb{R}$	$f(x) = \sum_i K(x_i, x)c_i$ $\forall c_i \in \mathcal{Y}$

Examples of operator-valued kernels

- Separable:
$$K(\mathbf{x}, \mathbf{z}) = k(\mathbf{x}, \mathbf{z})\mathbf{T} \quad \forall\, \mathbf{x}, \mathbf{z} \in \mathcal{X}$$
where k is a scalar-valued kernel and $\mathbf{T} \in \mathbb{R}^{p\times p}$ is symmetric
- Sum of separable:
$$K(\mathbf{x}, \mathbf{z}) = \sum_l k_l(\mathbf{x}, \mathbf{z})\mathbf{T}_l, \quad \forall\, \mathbf{x}, \mathbf{z} \in \mathcal{X},$$
k_l are scalar-valued kernels, $\mathbf{T}_l \in \mathbb{R}^{p\times p}$ are symmetric
- Transformable:
$$K(\mathbf{x}, \mathbf{z}) = \left[\tilde{k}(S_m\mathbf{x}, S_l\mathbf{z})\right]_{l,m=1}^{p}, \quad \forall\, \mathbf{x}, \mathbf{z} \in \mathcal{X}$$
where $\{S_t\}_{t=1}^p$ are mappings which transform the data from $\mathcal{X}$ to another space $\tilde{\mathcal{X}}$ where $\tilde{k}$ is defined.

Partial trace kernels



Trace Partial trace

Definition 1. (Partial trace kernel)
A partial trace kernel is an operator-valued kernel function K having the follwing form
$$K(\mathbf{x}, \mathbf{z}) = tr_{\mathcal{K}}(\mathbf{P}_{\phi(\mathbf{x}),\phi(\mathbf{z})}), \tag{1}$$
where $\mathbf{P}_{\mathbf{x},\mathbf{z}}$ 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).

Generalization of the kernel trick:
$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z})\rangle = tr(\phi(\mathbf{x})\phi(\mathbf{z})^\top)$$

Operator-valued kernel classes

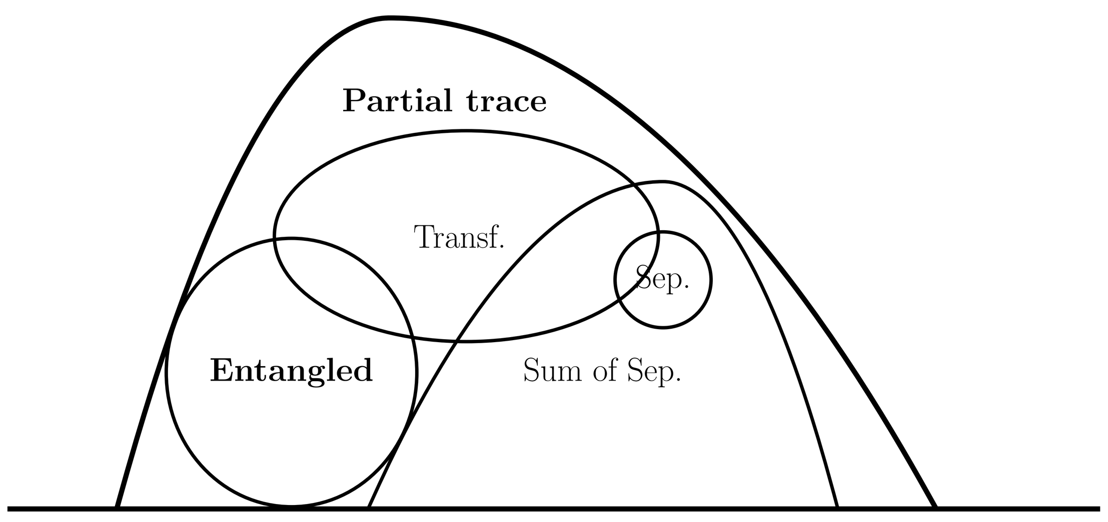


Illustration of inclusions among various operator-valued kernel classes.

Example 1. *(Transformable and not separable kernel) On the space $\mathcal{X} = \mathbb{R}$, consider the kernel*
$$K(\mathbf{x}, \mathbf{z}) = \begin{pmatrix} \mathbf{xz} & \mathbf{xz}^2 \\ \mathbf{x}^2\mathbf{z} & \mathbf{x}^2\mathbf{z}^2 \end{pmatrix}, \quad \forall\, \mathbf{x}, \mathbf{z} \in \mathcal{X}.$$

- *Transformable: choose $\tilde{k}(\mathbf{x}, \mathbf{z}) = \mathbf{xz}$, $S_1(\mathbf{x}) = \mathbf{x}$, and $S_2(\mathbf{x}) = \mathbf{x}^2$*
- *For a separable kernel the matrix $\mathbf{T}$ is always symmetric and since the matrix $K(\mathbf{x}, \mathbf{z})$ is not, K is not a separable kernel*

Example 2. *(Transformable and separable kernel) 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 separable by construction.*
- *Since $\mathbf{T}$ is of rank one, it follows that $\mathbf{T} = \mathbf{uu}^\top$ and $\big(K(\mathbf{x}, \mathbf{z})\big)_{lm} = \mathbf{u}_l\mathbf{u}_m\langle \mathbf{x}, \mathbf{z}\rangle$.*
- *We can see that K is transformable by replacing $\tilde{k}(\mathbf{x}, \mathbf{z})$ by $\langle \mathbf{x}, \mathbf{z}\rangle$ and $S_t(\mathbf{x})$ by $\mathbf{u}_t\mathbf{x}$, $t = 1, \dots, p$.*

Example 3. *(Partial trace contains sum of separable) Choose*
$$\mathbf{P}_{\phi(\mathbf{x}),\phi(\mathbf{z})} = \sum_l \mathbf{T}_l \otimes \big(\phi_l(\mathbf{x})\phi_l(\mathbf{z})^\top\big).$$

Example 4. *(Partial trace contains transformable) Choose*
$$[\mathbf{P}_{\tilde{\phi}(\mathbf{x}),\tilde{\phi}(\mathbf{z})}]_{l,m} = \big(\tilde{\phi} \circ S_l(\mathbf{x})\big)\big(\tilde{\phi} \circ S_m(\mathbf{z})\big)^\top.$$

Entangled kernels

Definition 2. (Entangled kernel)
An entangled operator-valued kernel K is defined as
$$K(\mathbf{x}, \mathbf{z}) = tr_{\mathcal{K}}\Big(\mathbf{U}\big(\mathbf{T} \otimes (\phi(\mathbf{x})\phi(\mathbf{z})^\top)\big)\mathbf{U}^\top\Big), \tag{2}$$
where $\mathbf{U} \in \mathbb{R}^{pN\times pN}$ is not separable (i.e. it cannot be written as product $\mathbf{A} \otimes \mathbf{B}$).

Remark. *Entangled kernels are subclass of partial trace kernels*

Example 5. *(Entangled and transformable) Choose linear kernel ($\phi(\mathbf{x}) = \mathbf{x}$) and mappings S_m to be linear, such that we can write matrix $\mathbf{U} = diag([\mathbf{S}_1 \cdots \mathbf{S}_p])$. Now the entangled kernel with operator*
$$\begin{aligned}\mathbf{P}_{\phi(\mathbf{x}),\phi(\mathbf{z})} &= \mathbf{U}([\mathbb{1}_p\mathbb{1}_p^\top] \otimes (\mathbf{xz}^\top))\mathbf{U}^\top \\ &= \mathbf{U}([\mathbf{xz}^\top]_{l,m=1}^{p})\mathbf{U}^\top \\ &= [\mathbf{S}_l\mathbf{xz}^\top\mathbf{S}_m^\top]_{l,m=1}^{p}\end{aligned}$$
is clearly also transformable with $\tilde{k}$ a linear kernel.

Theorem 1. *(Choi-Kraus representation)*
The map $K(\mathbf{x}, \mathbf{z}) = tr_{\mathcal{K}}\Big(\mathbf{U}\big(\mathbf{T} \otimes (\phi(\mathbf{x})\phi(\mathbf{z})^\top)\big)\mathbf{U}^\top\Big)$ can be generated by an operator sum representation containing at most pN elements,
$$K(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{r} \mathbf{M}_i\phi(\mathbf{x})\phi(\mathbf{z})^\top\mathbf{M}_i^\top, \tag{3}$$
where $\mathbf{M}_i \in \mathbb{R}^{p\times N}$ and $1 \le r \le pN$.

For computational feasibility (ϕ can be infinite-dimensional) we need to use an approximation $\hat{\phi}$ such that
$$k(x, z) = \langle \phi(x), \phi(z)\rangle \approx \langle \hat{\phi}(x), \hat{\phi}(z)\rangle$$

Our approximated kernel will thus be
$$\hat{K}(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{r} \hat{\mathbf{M}}_i\hat{\phi}(\mathbf{x})\hat{\phi}(\mathbf{z})^\top\hat{\mathbf{M}}_i^\top,$$
where $\hat{\phi}(\mathbf{x}) \in \mathbb{R}^m$ and $\hat{\mathbf{M}}_i \in \mathbb{R}^{p\times m}$.
Approximated kernel matrix is
$$\begin{aligned}\hat{\mathbf{G}} &= \sum_{i=1}^{r} \mathrm{vec}(\hat{\mathbf{M}}_i\hat{\mathbf{\Phi}})\,\mathrm{vec}(\hat{\mathbf{M}}_i\hat{\mathbf{\Phi}})^\top \\ &= \sum_{i=1}^{r} (\hat{\mathbf{\Phi}}^\top \otimes \mathbf{I}_p)\underbrace{\mathrm{vec}(\mathbf{M}_i)\,\mathrm{vec}(\mathbf{M}_i)^\top}_{\mathbf{D}_i}(\hat{\mathbf{\Phi}} \otimes \mathbf{I}_p) \\ &= (\hat{\mathbf{\Phi}}^\top \otimes \mathbf{I}_p)\mathbf{D}(\hat{\mathbf{\Phi}} \otimes \mathbf{I}_p)\end{aligned}$$

$$\hat{\mathbf{G}} = (\hat{\mathbf{\Phi}}^\top \otimes \mathbf{I}_p)\mathbf{QQ}^\top(\hat{\mathbf{\Phi}} \otimes \mathbf{I}_p) \tag{4}$$

Entangled Kernel Learning

We extend alignment between two matrices $\mathbf{M}$ and $\mathbf{N}$ is defined as
$$A(\mathbf{M}, \mathbf{N}) = \frac{\langle \mathbf{M}_c, \mathbf{N}_c\rangle_F}{\|\mathbf{M}_c\|_F\|\mathbf{N}_c\|_F} \tag{5}$$
to be our closeness criterion for learning the entangled kernel. Here subscript c refers to centered matrices, that is, $\mathbf{M}_c = \mathbf{HMH}$ where $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{11}^\top$, if $\mathbf{M}$ is a $n \times n$ matrix.

The optimization problem:

$$\max_{\mathbf{Q}} \quad (1-\gamma)\,A\left(tr_p(\hat{\mathbf{G}}), \mathbf{Y}^\top\mathbf{Y}\right) + \gamma\,A\left(\hat{\mathbf{G}}, \mathbf{yy}^\top\right) \tag{6}$$

with $\gamma \in [0, 1]$.

- First term learns a scalar-valued kernel $tr_p(\hat{\mathbf{G}})$ with alignment to linear kernel on outputs, $\mathbf{Y}^\top\mathbf{Y}$.
- Second term learns full operator-valued kernel $\mathbf{G}$ by aligning it to outer product of the outputs, promoting entanglement.

This can be solved with gradient-based approach.

Scalar-valued partial trace entangled kernel

After learning $\mathbf{G}$, we can extract $tr_p(\hat{\mathbf{G}})$ which can be used in scalar-valued framework, denoted ptrEKL.

Algorithm 1 Entangled Kernel Learning (EKL)

Input: matrix of features ˘, labels $\mathbf{Y}$
// 1) Kernel learning:
Solve for $\mathbf{Q}$ in eq.6 ($\mathbf{D} = \mathbf{QQ}^\top$) within a sphere manifold
// 2) Learning the predictive function:
if Predict with scalar-valued kernel **then**
$\mathbf{c}_K = (tr_p(\hat{\mathbf{G}}) + \lambda\mathbf{I})^{-1}\mathbf{Y}^\top$ $\quad\mathcal{O}(m^3 + mnp)$
else
$\mathbf{c}_G = (\hat{\mathbf{G}} + \lambda\mathbf{I})^{-1}\,\mathrm{vec}(\mathbf{Y})$ $\quad\mathcal{O}(r^3 + mnp^2)$
Return $\mathbf{D} = \mathbf{QQ}^\top$, $\mathbf{c}$

Experiments

We compare our **EKL** to Output Kernel Learning (**OKL**) for separable kernels and kernel ridge regression(**KRR**) to learn outputs independently.

Simulated data created with bi-linear model $\mathbf{TCA} + \mathbf{ICK} = \mathbf{Y}$, algorithms are given $\mathbf{K}$ to solve for $\mathbf{Y}$.

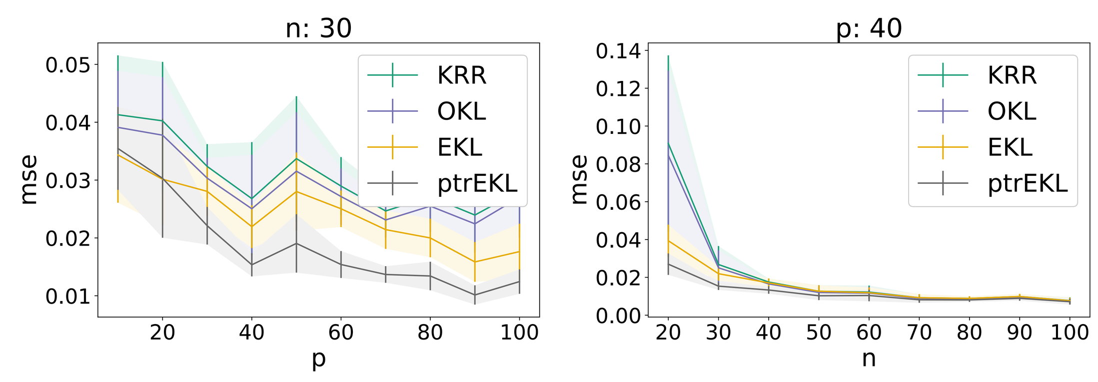


Figure: Results (mean squared error) of the simulated experiments with fixed amount of inputs and varying number of outputs (top), and fixed amount of outputs and varying inputs (bottom).The advantage of learning complex relationships is the biggest with small n.

Real Weather-dataset with $p = 365$ and $n = 35$ (`www.psych.mcgill.ca/misc/fda`).

	$n = 5$		$n = 10$		$n = 15$	
method	nMSE	nI	nMSE	nI	nMSE	nI
KRR	0.951 ± 0.101	0.000	0.813 ± 0.141	0.000	0.761 ± 0.037	0.000
OKL	1.062 ± 0.250	-0.092	0.900 ± 0.196	-0.094	0.788 ± 0.058	-0.034
EKL/ptrEKL	0.840 ± 0.084	0.124	0.722 ± 0.036	0.107	0.728 ± 0.033	0.044

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

Contributions

We have

- Connected the fields of quantum computing and machine learning by using the notion of entanglement and the Choi-Kraus quantum separability theorem in context of kernel learning.
- Defined a general class of kernels, partial trace kernels, that encompasses many OvK classes
- Defined smaller class of entangled kernels that are not separable
- Derived algorithm for Entangled Kernel Learning
- Demonstrated the effectiveness of learning non-separable kernels

References

- C. Ciliberto, Y. Mroueh, T. Poggio, and L. Rosasco. Convex learning of multiple tasks and their structure. ICML 2015.
- M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. Journal of machine learning research 2011.
- Néhémy Lim et al. Operator-valued kernel-based vector autoregressive models for network inference. Machine learning 99.3 (2015), pp. 489–51
- C. Micchelli and M. Pontil. On Learning Vector-Valued Functions. Neural Computation, 2005.
- E. Rieffel and W. Polak. Quantum computing: A gentle introduction. MIT Press, 2011.