Learning to predict complex outputs

A kernel view

Florence d'Alché-Buc

June 13, 2022

LTCI, Télécom Paris, Institut Polytechnique de Paris
`florence.dalche@telecom-paris.fr`
LSE, Statistics Seminar

Contributors

Pierre Geurt

Céline Brouard

Marie Szafranski

Romain Brault

Pierre Laforgue

Stephan Clémençon

Alex Lambert

Zoltan Szabo

Luc Brogat-Motte

Juho Rousu

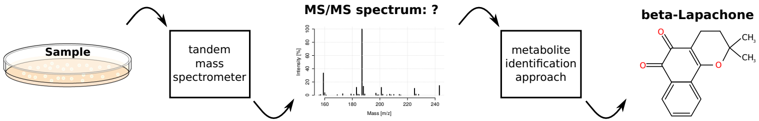
Alessandro Rudi

Sanjeel Parekh

Outline

Motivation: Metabolite identification using mass spectrometry

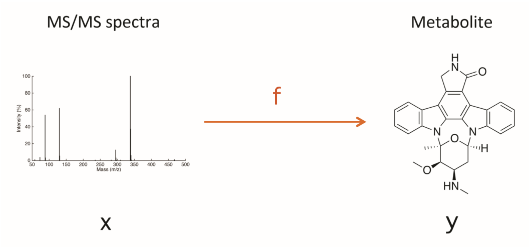
Goal: help chemists to identify metabolites in a biological sample using mass spectra.



(Dührkop et al., 2015, Nguyen et al. 2018)

Supervised metabolite prediction from mass spectra

Assume we observe pairs of mass spectra and graphs, use them to train a labeled graph prediction model



(Brouard et al. 2016, Brouard et al. 2019)

Structured prediction

Learning problem
Given some loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, the structured prediction problem writes as:

$$\min_{f \in \mathcal{F}(\mathcal{X},\mathcal{Y})} \mathbb{E}_{X,Y}[\Delta(Y, f(X))]. \tag{1}$$

In supervised learning, we aim at finding a good estimator f_n in some hypothesis space $\mathcal{H}$ of a minimizer of this problem using a given sample i.i.d. $\{(x_i, y_i)_{i=1}^n\})$.

Structured prediction: challenges

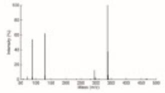
- the space $\mathcal{Y}$ is finite and huge !
- how to make this problem amenable to continuous optimization ?
- in the literature, different relaxations of the problem: energy-based learning, end-to-end learning, surrogate approaches (this talk)

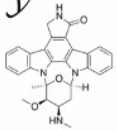
Proposition of a generic framework

- Choose an appropriate representation vector space $\mathcal{Z}$ for complex outputs
- Regress the output $Z = \psi(Y)$ in this representation space $\mathcal{Z}$ especially by leveraging regularization and get $h : \mathcal{X} \rightarrow \mathcal{Z}$
- Structured prediction: at testing time, solve a pre-image problem and get $f : \mathcal{Z} \rightarrow \mathcal{Y}$ by decoding $f = d \circ h$

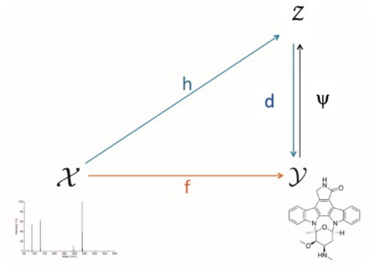
Get the intuition with molecule identification from mass spectra

$$\begin{tikzcd}
 & \mathcal{Z} \arrow[d, shift right, "d"'] \\
\mathcal{X} \arrow[ur, "h"] \arrow[r, "f"'] & \mathcal{Y} \arrow[u, shift right, "\psi"']
\end{tikzcd}$$





Novel problems to solve



1. Define $\mathcal{Z}$ and $\psi : \mathcal{Y} \rightarrow \mathcal{Z}$
2. Learn $h : \mathcal{X} \rightarrow \mathcal{H}_{k_\mathcal{Y}}$ to predict $\psi(y)$ given x
3. Solve a pre-image problem : compute $f(x) = d \circ h(x)$ where d is a "decoding function".

Leveraging the kernel trick

In this talk, focus on:

- Learning functions with values in a Hilbert space $\mathcal{Z}$
- $\mathcal{Z}$ is chosen to be a Reproducing Kernel Hilbert Space associated to a so-called output kernel, i.e. a similarity between outputs.

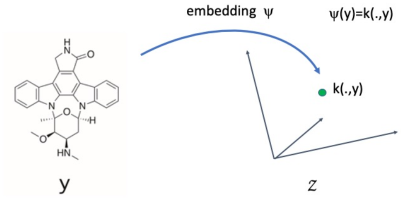
We called the corresponding family of regression tasks: output kernel regression.

Output Kernel Regression: use kernel trick in the output space

Choose a kernel $k_y : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ that encodes the similarity between structured objects

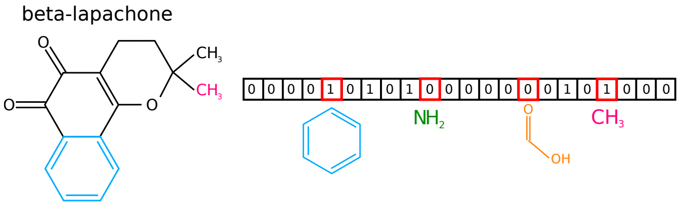
Take $\psi(y) = k(\cdot, y)$

$\mathcal{Z} := \mathcal{H}_{k_\mathcal{Y}}$



Example: kernel between molecules

Based on FingerID [Heinonen et al., 2012; Dührkop et al., 2015; Nguyen et al., 2018]



- Use **molecular fingerprint** $c(y) \in \mathbb{R}^d$ to encode the structure of a molecule as a (very large) binary vector
- Each entry indicates the existence or the frequency of a certain molecular property: atom or bond type, substructure (e.g. aromatic ring).

Use a Gaussian kernel on $c(y)$: $k_{\mathcal{Y}}(y, y') = \exp(-\gamma \|c(y) - c(y')\|^2)$

Advantages of defining a kernel $k(y, y')$ i?

- Allowing **infinite dimensional embeddings** while leveraging the kernel trick
- One principle to rule them all: kernels for various structured objects (See Gaertner 2006), opening the door to many structured tasks
    - label ranking (see Korba et al. 2018)
    - link prediction (Geurts et al. 2006, 2007)
    - image completion (Cortes et al. 2005, ...)
    - graph prediction (Brouard et al. 2020, Brogat-Motte et al. 2021)

A constraint however: to benefit from the kernel trick, not all the losses are suitable !

Structured prediction with Output Kernel Regression

Now take $\Delta(y, y') = \ell(\psi(y), \psi(y'))$ and replace the target problem in Eq.1 by the surrogate problem:

$$\min_{h:\mathcal{X}\to\mathcal{Z}} \mathbb{E}_{X,Y}[\ell(\psi(Y), h(X))].$$

Empirical (regularized) counterpart: with $\Omega : \mathcal{H} \to \mathbb{R}^+$ and $\lambda > 0$ given some hypothesis space $\mathcal{H}$,

$$\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(\psi(y_i), h(x_i)) + \lambda\Omega(h),$$

using a given dataset $\{(x_i, y_i)_{i=1}^n\}$.
Once we get h_n, define
$f_n(x) = d \circ h_n(x) = \arg\min_{y\in\mathcal{Y}} \ell(\psi(y), h_n(x))$

From a practical point of view

One wishes to use the kernel trick...

- Condition 1: ℓ must be computed by using inner products $\langle \psi(y), \psi(y') \rangle_{\mathcal{H}_{k_\mathcal{Y}}} = k(y, y')$.
- Condition 2: if an estimated model h_n writes as:

$$h_n(x) = \sum_{i=1}^{n} \beta_i(x) \psi(y_i)$$

$\beta : \mathcal{X} \to \mathbb{R}^d$, then if ℓ satisfies Condition 1, one can compute $f_n(x) = \arg\min_{y \in \mathcal{Y}} \ell(\psi(y), h_n(x))$.

Non-parametric models come to the place: trees-based approaches, k-nearest-neighbors, ... , kernel methods

Kernel-based approach

- Leverage convex optimization, govern regularization
- Allow for structured data in the input space as well.
- Structured Encoding Loss Framework (Ciliberto et al. 2016) / Implicit Loss Embedding (Ciliberto et al. 2020): Fisher Consistency, and the excess risk of f governed by the excess risk of h.

OK, but to get functions with values in Hilbert space $\mathcal{Z}$: *we need Operator-Valued Kernels (OVK) !*

Outline

Operator-valued Kernels and vector-valued Reproducing Kernel Hilbert Spaces

- (Pedrycs, 1957) theory of vv-RKHS
- (Senkene and Tempel'man, 1973) theory of vv-RKHS
- (Hein and Bousquet, 2004) survey on positive definite kernels, including a short introduction to OVK
- (Micchelli and Pontil, 2005) learning vector-valued functions with OVK
- (Carmeli et al., 2006) theory of vv-RKHS
- (Carmeli et al. 2010) vv-RKHS and universality

From scalar-valued kernels to operator-valued kernels

Notations: if $\mathcal{Z}$ is a Hilbert Space, $\mathcal{L}(\mathcal{Z})$ is the space of bounded linear operators on $\mathcal{Z}$.

Scalar kernel	Operator-valued kernel
$k(x, x') \in \mathbb{R}$	$\mathcal{K}(x, x') \in \mathcal{L}(\mathcal{Z})$
$k(x, x') = k(x', x)$	$\mathcal{K}(x, x') = \mathcal{K}(x', x)^*$
$\forall (x_i, z_i)_{i=1}^m \in (\mathcal{X} \times \mathbb{R})^m,$	$\forall (x_i, z_i)_{i=1}^m \in (\mathcal{X} \times \mathcal{Z})^m,$
$\sum_{i,j=1}^m z_i z_j k(x_i, x_j) \geq 0$	$\sum_{i,j=1}^m \langle z_i, \mathcal{K}(x_i, x_j) z_j \rangle_{\mathcal{Z}} \geq 0$
$\mathcal{H}_k = \overline{Span\{k(\cdot, x), x \in \mathcal{X}\}}$	$\mathcal{H}_{\mathcal{K}} = \overline{Span\{\mathcal{K}(\cdot, x)z : x, z \in \mathcal{X} \times \mathcal{Z}\}}$
$\langle f, k(\cdot, x) \rangle_{\mathcal{H}_k} = f(x)$	$\langle f, \mathcal{K}(\cdot, x)z \rangle_{\mathcal{H}_{\mathcal{K}}} = \langle f(x), z \rangle_{\mathcal{Z}}$

Hint: think about the matrix-valued case

$\mathcal{Z} = \mathbb{R}^d$

- A trivial kernel : $\mathcal{K}(x, x') = I_{\mathcal{Z}}.k(x, x')$, where $I_{\mathcal{Z}}$ is the $d \times d$ identity matrix (independent outputs)
- A separable kernel: $\mathcal{K}(x, x') = A.k(x, x')$ where A is positive semi-definite matrix $d \times d$ (dependencies between outputs)

Important! As in scalar kernel methods, choosing $\mathcal{K}$ implies choosing the way you want to regularize when using $\|\cdot\|_{\mathcal{H}_{\mathcal{K}}}$

Separable Operator-valued kernels

In particular, we will make use of a special separable operator-valued kernel:

$$K(x, x') = I_{\mathcal{H}_{\mathcal{Y}}} k(x, x'),$$

which allows us to work with outputs in $\mathcal{H}_{\mathcal{Y}}$.

More about operator-valued-kernels

Again general case: $\mathcal{Z}$ Hilbert Space

Scalar kernel	Operator-valued kernel
Representer theorem:	Representer Theorem:
$\min_{h\in\mathcal{H}_k} L(h(x_1),\ldots,h(x_n)) + \lambda\|h\|^2_{\mathcal{H}_k}$	$\min_{h\in\mathcal{H}_\mathcal{K}} L(h(x_1),\ldots,h(x_n)) + \lambda\|h\|^2_{\mathcal{H}_\mathcal{K}}$
$h_n(x) = \sum_{i=1}^n k(x,x_i)\alpha_i \in \mathbb{R}$	$h_n(x) = \sum_{i=1}^n \mathcal{K}(x,x_i)\alpha_i \in \mathcal{Z}$

N.B. A representer theorem for OVK but still we do not know how to compute $\alpha_i \in \mathcal{Z}$

A simple case: kernel ridge regression 1/2

Assume we observe $(x_i, z_i)_{i=1}^n$, define an operator-valued kernel $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Z})$ such that: $K(x, x') = Id_{\mathcal{Z}} k_X(x, x')$

Let us consider, for $\lambda > 0$:

$$\min_{h \in \mathcal{H}_{\mathcal{K}}} \frac{1}{n} \sum_{i=1}^{n} \|z_i - h(x_i)\|_{\mathcal{Z}}^2 + \lambda \|h\|_{\mathcal{H}_{\mathcal{K}}}^2 \tag{2}$$

A simple case: kernel ridge regression 2/2

- The representer theorem (Micchelli and Pontil, 2005) applies
- The unique minimizer h_n writes: $h_{ridge}(x) = \sum_{i=1}^{n} \mathcal{K}(x, x_i)\hat{\alpha}_i$

where $\hat{\alpha}_i$'s enjoy a closed form, yielding to the following expression:

$$h_{ridge}(x) = \sum_{j=1}^{n} \beta_j(x) z_j, \tag{3}$$

with: $\beta(x) = (K_x + n\lambda I)^{-1}\kappa_X^x$
and $\kappa_X^x = [k_X(x_1, x), \ldots, k_X(x_n, x)]^T$.

Back to structured prediction: Input Output Kernel Ridge Regression (ridge-IOKR)

Now the feature space $\mathcal{Z} := \mathcal{H}_{k_\mathcal{Y}}$ is the RKHS associated to $k_\mathcal{Y}$, a kernel on $\mathcal{Y}$.
Define the OVK $K(x, x') = Id_{\mathcal{H}_{k_\mathcal{Y}}} k_X(x, x')$
Denote $\psi(y) = k_\mathcal{Y}(\cdot, y)$.

$$h_n(x) = \sum_{i=1}^{n} \beta_i(x)\psi(y_i), \tag{4}$$

with: $\beta(x) = (K_x + n\lambda I)^{-1}\kappa_X^x$
and $\kappa_X^x = [k_X(x_1, x), \ldots, k_X(x_n, x)]^T$ and $\lambda > 0$.
Then, we are able to compute

$$f_n(x) = \arg\min_{y \in \mathcal{Y}} \|\psi(y) - h(x_i)\|^2_{\mathcal{H}_{k_\mathcal{Y}}}, \tag{5}$$

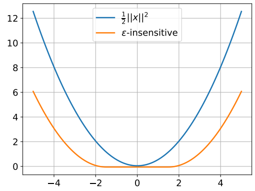
using only inner products of $\psi(y_i)$s.
NB. We retrieve Kernel Dependency Estimation of Cortes et al. as well.

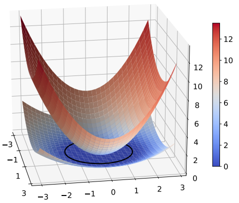
More about kernel ridge regression with input and output kernels

- Leveraging unlabeled input data: semi-supervised IOKR (ridge or not) - Brouard et al. 2011,16 with nice applications to link prediction.
- Leveraging structure in the output feature space: reduced-rank approach *Work of Luc Brogat-Motte et al., submitted*

Now more interesting loss functions: sparsity and robustness

ϵ-Ridge

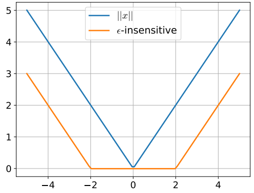


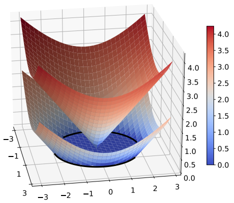


$$\frac{1}{2}\|\cdot\|^2 \,\square\, \chi_{\mathcal{B}_\epsilon}$$

(Sparsity)

ϵ-SVR

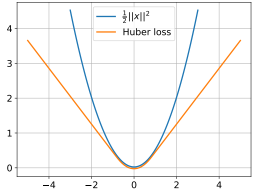


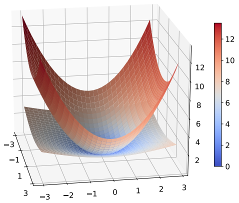


$$\|\cdot\| \,\square\, \chi_{\mathcal{B}_\epsilon}$$

(Sparsity, Robustness)

κ-Huber





$$\kappa\|\cdot\| \,\square\, \frac{1}{2}\|\cdot\|^2$$

(Robustness)

Data-sparse and robust loss: the example of ϵ-insensitive loss

Data-sparse and Robust losses [Sangnier et al. 2017, Laforgue et al. 2020]:
With a slight abuse of notation
Let $\ell : \mathcal{Z} \to \mathbb{R}$ be a convex loss with unique minimum
at 0, and $\epsilon > 0$. The ϵ-insensitive version of ℓ, denoted ℓ_ϵ, is defined by:

$$\ell_\epsilon(z) = \left(\ell \;\square\; \chi_{\mathcal{B}_\epsilon}\right)(z) = \begin{cases} \ell(0) & \text{if } \|z\|_{\mathcal{Z}} \leq \epsilon \\ \inf\limits_{\|d\|_{\mathcal{Z}} \leq 1} \ell(z - \epsilon d) & \text{otherwise} \end{cases},$$

Infimal convolution: $(f \square g)(x) = \inf_{x'} f(x') + g(x - x')$. (Bauschke et al. 2011)

Reminder: representer theorem and convex losses

General case: the output space is $\mathcal{Z}$: Hilbert Space and output training data are denoted z_i. Let $\ell : \mathcal{Z} \to \mathbb{R}$ a convex loss.

Theorem (Micchelli et Pontil 2005)

The solution to the learning problem is given by

$$h_n = \frac{1}{\lambda n} \sum_{i=1}^{n} \mathcal{K}(\cdot, x_i)\hat{\alpha}_i, \tag{6}$$

with $(\hat{\alpha}_i)_{i=1}^n \in \mathcal{Z}^n$ the solutions to the dual problem:

Problem

(Brouard et al. 2016, Sangnier et al. 2017)

$\min_{(\alpha_i)_{i=1}^n \in \mathcal{Z}^n} \ \sum_{i=1}^n \ell_i^\star(-\alpha_i) + \frac{1}{2\lambda n} \sum_{i,j=1}^n \langle \alpha_i, \mathcal{K}(x_i, x_j)\alpha_j \rangle_{\mathcal{Z}}$,

where $g^\star : \alpha \in \mathcal{Z} \mapsto \sup_{z \in \mathcal{Z}} \langle \alpha, z \rangle_{\mathcal{Z}} - g(z)$ denotes the Fenchel-Legendre transform of a function $g : \mathcal{Z} \to \mathbb{R}$.

with $\ell_i(y) = \ell(y_i - y)$.

Some limitations

- **1st limitation:** the Fenchel-Legendre transform $\ell^\star$ needs to be computable ($\rightarrow$ assumption)
- **2nd limitation :** the dual variables $(\alpha_i)_{i=1}^n$ are still **infinite dimensional!**

Some limitations

- **1st limitation:** the Fenchel-Legendre transform $\ell^\star$ needs to be computable ($\rightarrow$ assumption)
- **2nd limitation :** the dual variables $(\alpha_i)_{i=1}^n$ are still **infinite dimensional!**

If $\mathbf{Z} = \mathrm{Span}\{z_j,\ j \leq n\}$ invariant by $\mathcal{K}$, *i.e.*
$\forall (x, x'),\quad z \in \mathbf{Z} \ \Rightarrow\ \mathcal{K}(x, x')z \in \mathbf{Z}$

$$\hat{\alpha}_i \in \mathbf{Z} \quad \rightarrow \quad \text{possible reparametrization}$$

The double representer theorem

Laforgue et al. ICML 2020.

Theorem (Double representer theorem)

Assume that OVK $\mathcal{K}$ and loss ℓ satisfy the appropriate assumptions (see paper for details, verified by standard kernels and our losses), then

$$\hat{h} = \underset{\mathcal{H}_{\mathcal{K}}}{\operatorname{argmin}} \; \frac{1}{n} \sum_i \ell(h(x_i) - z_i) + \frac{\lambda}{2} \|h\|^2_{\mathcal{H}_{\mathcal{K}}} \quad \textit{is given by}$$

$$\hat{h} = \frac{1}{\lambda n} \sum_{i,j=1}^{n} \mathcal{K}(\cdot, x_i) \; \hat{\omega}_{ij} \; z_j,$$

*with $\hat{\Omega} = [\hat{\omega}_{ij}] \in \mathbb{R}^{n \times n}$ the solution to the **finite dimensional** problem*

$$\min_{\Omega \in \mathbb{R}^{n\times n}} \; \sum_{i=1}^{n} L_i \left(\Omega_{i:}, K^Z\right) + \frac{1}{2\lambda n} \mathbf{Tr}\left(\tilde{M}^\top (\Omega \otimes \Omega)\right),$$

with $\tilde{M}$ the $n^2 \times n^2$ matrix writing of M s.t. $M_{ijkl} = \langle z_k, \mathcal{K}(x_i, x_j) z_l \rangle_{\mathcal{Z}}$.

Specific dual problems for our losses 1

If $\mathcal{K} = k\ \mathbf{I}_{\mathcal{Z}}$, the solutions to the ϵ-Ridge regression, κ-Huber regression, and ϵ-SVR primal problems

$$
\begin{aligned}
(P1) \quad & \min_{h \in \mathcal{H}_{\mathcal{K}}} \quad \frac{1}{2n} \sum_{i=1}^{n} \|h(x_i) - z_i\|_{\mathcal{Z},\epsilon}^2 + \frac{\Lambda}{2}\|h\|_{\mathcal{H}_{\mathcal{K}}}^2, \\
(P2) \quad & \min_{h \in \mathcal{H}_{\mathcal{K}}} \quad \frac{1}{n} \sum_{i=1}^{n} \ell_{H,\kappa}(h(x_i) - z_i) + \frac{\Lambda}{2}\|h\|_{\mathcal{H}_{\mathcal{K}}}^2, \\
(P3) \quad & \min_{h \in \mathcal{H}_{\mathcal{K}}} \quad \frac{1}{n} \sum_{i=1}^{n} \|h(x_i) - z_i\|_{\mathcal{Z},\epsilon} + \frac{\Lambda}{2}\|h\|_{\mathcal{H}_{\mathcal{K}}}^2,
\end{aligned}
$$

are given in next slide, with $\hat{\Omega} = \hat{W} V^{-1}$, and $\hat{W}$ the solution to the respective finite dimensional dual problems

Specific dual problems for our losses 2

For the ϵ-ridge, ϵ-SVR and κ-Huber, it holds $\hat{\Omega} = \hat{W} V^{-1}$, with $\hat{W}$ the solution to these finite dimensional dual problems:

$$(D1) \quad \min_{W \in \mathbb{R}^{n \times n}} \quad \frac{1}{2} \left\| AW - B \right\|_{\text{Fro}}^2 + \epsilon \left\| W \right\|_{2,1},$$

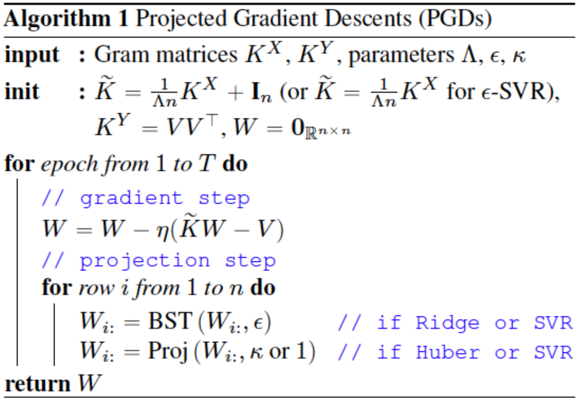
$$(D2) \quad \begin{array}{ll} \displaystyle\min_{W \in \mathbb{R}^{n \times n}} & \displaystyle\frac{1}{2} \left\| AW - B \right\|_{\text{Fro}}^2 + \epsilon \left\| W \right\|_{2,1}, \\ \text{s.t.} & \left\| W \right\|_{2,\infty} \leq 1, \end{array}$$

$$(D3) \quad \begin{array}{ll} \displaystyle\min_{W \in \mathbb{R}^{n \times n}} & \displaystyle\frac{1}{2} \left\| AW - B \right\|_{\text{Fro}}^2, \\ \text{s.t.} & \left\| W \right\|_{2,\infty} \leq \kappa, \end{array}$$

with V, A, B such that: $VV^\top = K^Y$, $A^\top A = K^X/(\lambda n) + \mathbf{I}_n$ (or $A^\top A = K^X/(\lambda n)$ for the ϵ-SVR), and $A^\top B = V$.

Projected Gradient Algorithm

Projected Gradient Descent algorithms with appropriate projection operator. For instance, (D1) is a multi-task lasso problem (See Obozinski et al. 2010)



Algorithm 1 Projected Gradient Descents (PGDs)

```
input : Gram matrices $K^X$, $K^Y$, parameters $\Lambda$, $\epsilon$, $\kappa$
init  : $\widetilde{K} = \frac{1}{\Lambda n} K^X + \mathbf{I}_n$ (or $\widetilde{K} = \frac{1}{\Lambda n} K^X$ for $\epsilon$-SVR),
        $K^Y = VV^\top$, $W = \mathbf{0}_{\mathbb{R}^{n \times n}}$
for epoch from 1 to $T$ do
    // gradient step
    $W = W - \eta(\widetilde{K}W - V)$
    // projection step
    for row $i$ from 1 to $n$ do
        $W_{i:} = \mathrm{BST}\left(W_{i:}, \epsilon\right)$          // if Ridge or SVR
        $W_{i:} = \mathrm{Proj}\left(W_{i:}, \kappa \text{ or } 1\right)$   // if Huber or SVR
return $W$
```

Proximal operators

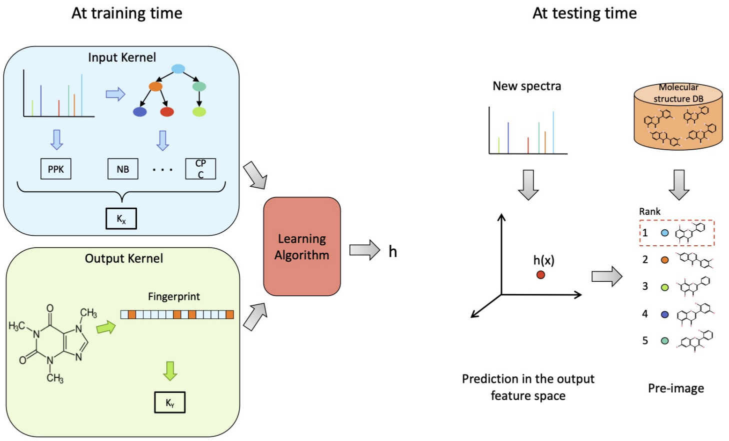
Block Soft Thresholding operator: $\mathrm{BST}(x, \tau) = \left(1 - \tau / \|x\|\right)_+ x$.
Projection operator for (D2) such that $\mathrm{Proj}(x, \tau) = \min\left(\tau / \|x\|, 1\right) x$.

More on IOKR

- Generalization bounds in the context of algorithm stability (extension of Elisseff, 2002; Audiffren and Kadri (2013); Laforgue et al. 2020)
- Deep IOKR: the example of KAE, kernel autoencoder (Laforgue et al. 2019), Deep structured prediction (El Ahmad et al., current work)
- Reduced-rank IOKR (a low-rank approach to IOKR-ridge with excess risk bounds, Brogat-Motte et al. submitted in 2021)

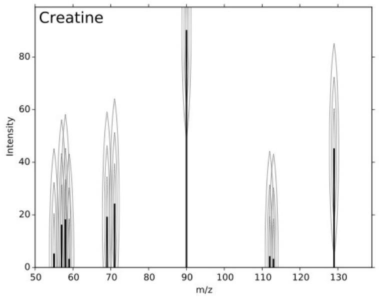
Outline

IOKR: the big picture



Input kernels: probability product kernel

- A mass spectrum is defined as a set of peaks: $x = \{x(\ell)\}_{\ell=1}^{n_x}$.
- Each peak is modeled as a 2D normal distribution centered around the observed position: $p_{x(\ell)} \sim \mathcal{N}(x(\ell), \Sigma)$.
- The covariance is shared with all peaks: $\Sigma = \begin{bmatrix} \sigma_m^2 & 0 \\ 0 & \sigma_i^2 \end{bmatrix}$.



Input kernel: probability product kernel

- A spectrum is represented as a mixture of its peak distributions:

$$p_x = \frac{1}{n_x} \sum_{\ell=1}^{n_x} p_{x(\ell)}.$$

- Probability product kernel [Jebara et al., 2004] between the peaks of two spectra x and x':

$$\begin{aligned} k(x, x') &= \int_{\mathbb{R}^2} p_x(\mathbf{z}) p_{x'}(\mathbf{z}) d\mathbf{z} \\ &= \frac{1}{n_x n_{x'}} \frac{1}{4\pi\sigma_m\sigma_i} \sum_{\ell,\ell'=1}^{n_x, n_{x'}} \exp\left(-\frac{1}{4}\left(x(\ell) - x'(\ell')\right)^T \boldsymbol{\Sigma}^{-1}\left(x(\ell) - x'(\ell')\right)\right) \end{aligned}$$

IOKR on metabolite prediction

Metabolite dataset: initially represented by 4136-size fingerprints (Brouard et al., 2016). Tanimoto kernel. Training data: 5579 molecules, Test data: 1359 molecules.

Table 1: Top 1 / 10 / 20 test accuracies (%)

λ	1*e*-6	1*e*-4
RIDGE-IOKR	35.7 | 79.9 | 86.6	38.1 | 82.0 | 88.9
HUBER-IOKR	**38.3** | **82.2** | **89.1**	37.7 | 81.9 | 88.8
ϵ-2-IOKR	37.1 | 81.7 | 88.3	36.3 | 81.2 | 87.9

More extensive results on metabolite datasets (5-CV)

Method	Tanimoto-Gaussian loss	Top-k accuracies
		$k = 1$
SPEN	0.537 ± 0.008	25.9% | 54.1% | 64.3%
ridge-IOKR	0.463 ± 0.009	29.6% | 61.1% | 71.0%
reduced-rank-IOKR	0.459 ± 0.010	30.0% | 61.5% | 71.4%

SPEN: Structured Prediction Energy Network (the best variant, structured hinge loss and feature network - Belanger et al. 2017)

Outline

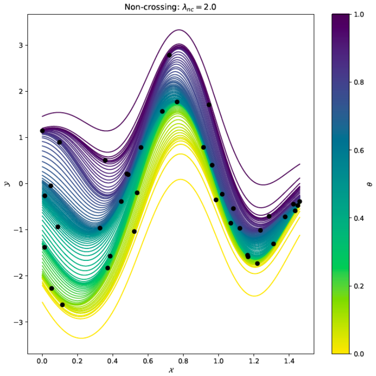
Other "complex" output variables

All these problems can be addressed by learning **functions with outputs in a Hilbert space**

Discrete structures	*Multiple Tasks*	*Functions*
Label Ranking	*Hierarchical Classification*	*Infimum of Tasks Learning*
Sequence, tree prediction	*Multi-label Classification*	*Functional Regression*
Graph prediction	*Multiple Output Regression*	*Meta-modeling*

Quantile regression for fish age prediction

Many reasons for quantile regression: outliers in the data, more robustness is asked...



<span style="color:red">Question:</span> Predict any θ-quantile of Y given x, for $\theta \in (0,1)$ [Brault et al. 2019]

Comparison with structured output regression

$\mathcal{X}$: input space

Structured prediction

$\mathcal{Y}$: finite set of structured objects

$k_{\mathcal{Y}}$: kernel over $\mathcal{Y}$

$\mathcal{Z} := \mathcal{H}_{k_{\mathcal{Y}}}$: RKHS associated to $k_{\mathcal{Y}}$

$$\mathcal{X} \xrightarrow{h} \mathcal{Z} \xrightarrow{d} \mathcal{Y}$$

Goal: obtain $f(x) = d \circ h(x)$

Infinite Task learning

$\mathcal{Y}$: output (observation) space

Θ: task parameter space

k_{Θ}: kernel over Θ

$\mathcal{Z} := \mathcal{H}_{k_{\Theta}}$: RKHS associated to k_{Θ}

$$\mathcal{X} \xrightarrow{h} \underbrace{(\Theta \to \mathcal{Y})}_{\mathcal{Z}}$$

Goal: obtain $h(x)(\theta)$

Conclusion

- The kernel trick used in the output space
- Leveraging vv-RKHS for learning output in infinite dimensional embedding space
- Practical algorithms even for losses more involved than the squared loss
- Other results: generalization bounds within the algorithm stability context, excess risk beyond SELF framework

Perspectives

- Scaling up the approaches:
    - Exploit approximations (Random Fourier features: Brault et al. 2017; Projection Learning: Bouche et al. 2020, Sketching, current work of El Ahmad et al.)
- Kernel Learning:
    - Exploiting approximations for both input and output kernel
    - Deep hybrid architecture (learning $\mathcal{K}$) - see for instance (Laforgue et al. 2019, Giffon et al. 2019, Li et al. 2019, Lambert 2021)

Additional thoughts ...

- Handling output features is not exclusive of kernel methods: see label embedding in one-shot/few-shot learning (Lampert et al. 2015, Djerrab et al. 2018), work of Lerouge et al. (2015) around IODA and Belharbi et al. (2017), for neural networks.
- Leveraging distances like those in Optimal Transport (see Luise, Rudi et al. 2018) yields to other non-parametric models: see Brogat-Motte et al. 's work on graph prediction with Fused-Gromov-Wasserstein barycenters (ICML 2022).

Codes

- Dualization and Robust losses (`https://github.com/plaforgue/dual_exp`), Pierre Laforgue
- Infinite task Learning: torch-itl (`https://github.com/allambert/torch_itl`), Alex Lambert, Sanjeel Parekh, Dimitri Bouche.
- Reduced-Rank IOKR (not yet public, Luc Brogat-Motte)
- Operalib (`https://github.com/operalib/operalib`) (Romain Brault) RFF for OVK, KRR, IOKR, ITL
- Currently tested : release of a general scikit-learn compatible library with **Hi!Paris** engineering group: if interested to test it, please send me an email.

Outline

References i

- Álvarez, M. A. and Rosasco, L. and Lawrence, N. D., Kernels for vector-valued functions: a review, Foundations and Trends in Machine Learning, 4:3,2012.
- R. Brault, M. Heinonen, F. d'Alché-Buc, Random Fourier Features for Operator-valued Kernels, ACML 2016, (2016)
- R. Brault, A. Lambert, Z. Szabó, M. Sangnier, F. d'Alché-Buc: Infinite Task Learning in RKHSs. AISTATS 2019: 1294-1302
- L. Brogat-Motte, R. Flamary, F. d'Alché-Buc:Learning to Predict Graphs with Fused Gromov-Wasserstein Barycenters. To appear in Proc. of ICML 2022.
- C. Brouard, F. d'Alché-Buc, M. Szafranski: Semi-supervised Penalized Output Kernel Regression for Link Prediction. ICML 2011: 593-600.
- C. Brouard, F. d'Alché-Buc, M. Szafranski, Input Output Kernel Regression for supervised and semi-supervised structured output learning, JMLR, oct. 2016
- C. Brouard, H. Shen, K. Dührkop, F. d'Alché-Buc, S. Böcker, J. Rousu: Fast metabolite identification with Input Output Kernel Regression. Bioinformatics 32(12): 28-36 (2016)
- C. Ciliberto, L. Rosasco, A. Rudi: A General Framework for Consistent Structured Prediction with Implicit Loss Embeddings. J. Mach. Learn. Res. 21: 98:1-98:67 (2020)
- C. Ciliberto, L. Rosasco, A. Rudi: A Consistent Regularization Approach for Structured Prediction. NIPS 2016: 4412-4420

References ii

- T. Evgeniou, C. A. Micchelli, M. Pontil: Learning Multiple Tasks with Kernel Methods. Journal of Machine Learning Research 6: 615-637 (2005)
- P. Geurts, L. Wehenkel, F. d'Alché-Buc: Kernelizing the output of tree-based methods. ICML 2006: 345-352
- Pierre Geurts, Nizar Touleimat, Marie Dutreix, F. d'Alché-Buc: Inferring biological networks with output kernel trees. BMC Bioinform. 8(S-2) (2007)
- Pierre Geurts, Louis Wehenkel, Florence d'Alché-Buc: Gradient boosting for kernelized output spaces. ICML 2007: 289-296
- Luc Giffon, Stéphane Ayache, Thierry Artières, Hachem Kadri: Deep Networks with Adaptive Nyström Approximation. IJCNN 2019: 1-8
- Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Manuel Davy: Nonlinear functional regression: a functional RKHS approach. AISTATS 2010: 374-380
- Hachem Kadri, Mohammad Ghavamzadeh, Philippe Preux: A Generalized Kernel Approach to Structured Output Learning. ICML (1) 2013: 471-479
- Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Alain Rakotomamonjy, Julien Audiffren: Operator-valued Kernels for Learning from Functional Response Data. J. Mach. Learn. Res. 17: 20:1-20:54 (2016)
- A. Lambert, D. Bouche, Z. Szabo, F. d'Alché-Buc:Functional Output Regression with Infimal Convolution: Exploring the Huber and ϵ-insensitive loss. To appear in Proc. of ICML 2022.

References iii

- P. Laforgue, S.Clémencon, Florence d'Alché-Buc: Autoencoding any Data through Kernel Autoencoders. AISTATS 2019: 1061-1069
- P. Laforgue, A. Lambert, L. Brogat-Motte, F. d'Alché-Buc: Duality in RKHSs with Infinite Dimensional Outputs: application to robustness, ICML 2020.
- Micchelli, C. A. and Pontil, M. and Ying, Y., Universal MultiTask Kernels, Journal of Machine Learning Research, 9,2008.
- M. Moussab, A. Garcia, M. Sangnier, F. d'Alché-Buc, Output Fisher Embedding Regression, Machine Learning Journal, (2018)
- M. Sangnier, O. Fercoq, F. d'Alché-Buc, Joint quantile regression in vector-valued RKHSs, NIPS 2016:3693-3701, (2016)
- M. Sangnier, O. Fercoq, F. d'Alché-Buc, Data sparse nonparametric regression with ϵ-insensitive losses, ACML 2017, (2017)
- L. Wenliang, D. J. Sutherland, H. Strathmann, A. Gretton. Proceedings of the 36th International Conference on Machine Learning (ICML), PMLR 97:6737-6746

Outline

Output Kernel Regression fits the SELF and ILE framework

Structured Encoding Loss Function (SELF, Ciliberto et al. 2016), Nowak-Villa (2018, 2019), Luige et al. 2019, and Consistent Structured prediction with Implicit Loss Embeddings (2020):

- general conditions on $\mathcal{Y}$ and losses to get Fisher consistency and excess risk bounds

SELF property and consequences

Definition (SELF loss - Ciliberto et al. 2016)
$\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is said to be SELF if it exists a separable Hilbert space $\mathcal{F}$, a feature map $\phi : \mathcal{Y} \to \mathcal{F}$ and a bounded linear operator A on $\mathcal{F}$ such that:

$$\Delta(y, y') = \langle \phi(y), A\phi(y') \rangle_{\mathcal{F}}$$

Theorem (Ciliberto et al. 2016)
Let Δ satisfy the SELF property with $\mathcal{Y}$ compact then, for every measurable function $h : \mathcal{X} \to \mathcal{F}$ and $d : \mathcal{F} \to \mathcal{Y}$, satisfying $d(z) = \arg\min_{y \in \mathcal{Y}} \langle \phi(y), Az \rangle_{\mathcal{F}}$, we have:

$$\epsilon(d \circ h^*) = \epsilon(f^*)$$
$$\epsilon(d \circ h) - \epsilon(f^*) \leq 2c_\Delta \sqrt{R(h) - R(h^*)},$$

with $\epsilon(f) = \mathbb{E}[\Delta(Y, f(X))] = \mathbb{E}[\langle \phi(y), A\phi(y') \rangle_{\mathcal{F}}]$ and $R(h) = \mathbb{E}[\|h(X) - \phi(Y)\|^2_{\mathcal{F}}]$

Output Kernel Regression - squared loss - fits the SELF framework

Trivial case: $k(y, y) = 1$ and $\ell(\psi(y), h(x)) = \|\psi(y) - h(x)\|^2_{\mathcal{H}_{k_\mathcal{Y}}}$.
Then :

$$\begin{aligned} f(x) &= d \circ h(x) \\ &= \arg\min_y \|\psi(y) - h(x)\|^2_{\mathcal{H}_{k_\mathcal{Y}}} \\ &= \arg\min_y - < \psi(y), h(x) > \end{aligned}$$

More about regularized least-squares regression: a reduced rank approach

Let $\lambda_1, \lambda_2 > 0$ and $p \in \mathbb{N}^*$. Let $\mathcal{P}_p$ be the set of the orthogonal projections from $\mathcal{Z}$ to $\mathcal{Z}$ of rank p.

We consider the estimator $x \to P\hat{h}_{\lambda_2}(x)$ where P is defined as

$$P := \text{argmin }_{P \in \mathcal{P}_p} \mathbb{E}[\|Ph^*(x) - h^*(x)\|_{\mathcal{Z}}^2]. \tag{7}$$

Nevertheless, P is unknown, thus we estimate it with $\hat{P}$ defined by

$$\hat{P} := \text{argmin }_{P \in \mathcal{P}_p} \frac{1}{n} \sum_{i=1}^{n} \|P\hat{h}_{\lambda_1}(x_i) - \hat{h}_{\lambda_1}(x_i)\|_{\mathcal{Z}}^2. \tag{8}$$

and we propose the estimator

$$\boxed{\hat{h}_{\lambda_1,\lambda_2,p}(x) = \hat{P}\hat{h}_{\lambda_2}(x)} \tag{9}$$

Reduced-rank regression in Structured Prediction

Novel estimator for IOKR in structured prediction ($\mathcal{Z} := \mathcal{H}_{k_{\mathcal{Y}}}$)

$$\hat{f}(x) = \text{argmin }_{y \in \mathcal{Y}} \|\hat{P}\hat{h}(x) - \psi(y)\|_{\mathcal{Z}}^2. \tag{10}$$

Algorithm	ridge-IOKR	Reduced-rank IOKR
Training	$\mathcal{O}(n^3)$	$\mathcal{O}(2n^3)$
Decoding	$\mathcal{O}(n_{test} n \lvert\mathcal{Y}\rvert)$	$\mathcal{O}(n_{test} p \lvert\mathcal{Y}\rvert)$

Table 2: Time complexity of IOKR versus reduced-rank IOKR.