Supervised Learning (COMP0078)

Structured Prediction

Carlo Ciliberto

University College London
Department of Computer Science

In this Class

1. Surrogate Methods
2. Implicit Loss Embeddings
3. A General Surrogate Algorithm
4. Theoretical Properties

Part 1: Surrogate Methods

Structured Prediction

x $\xrightarrow{f}$ y

Image Captioning
(also Localization
Segmentation
Classification)

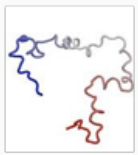
Movie Ranking

NETFLIX
user:127

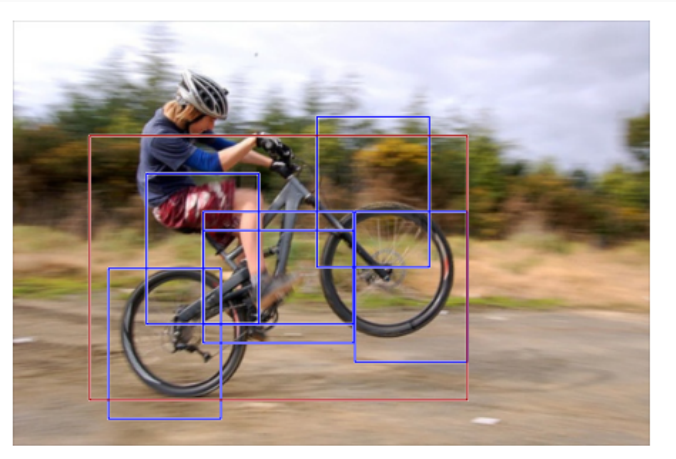
Speech Recognition

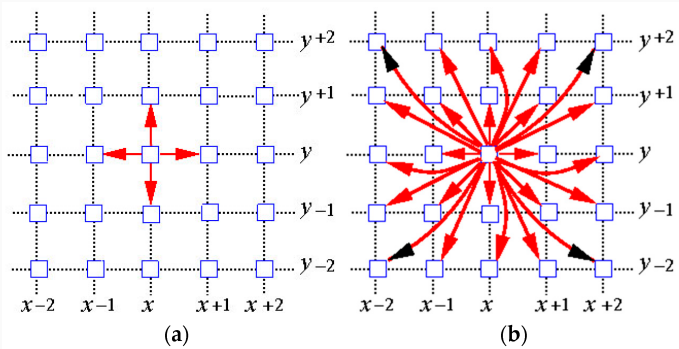
"Ok Google"

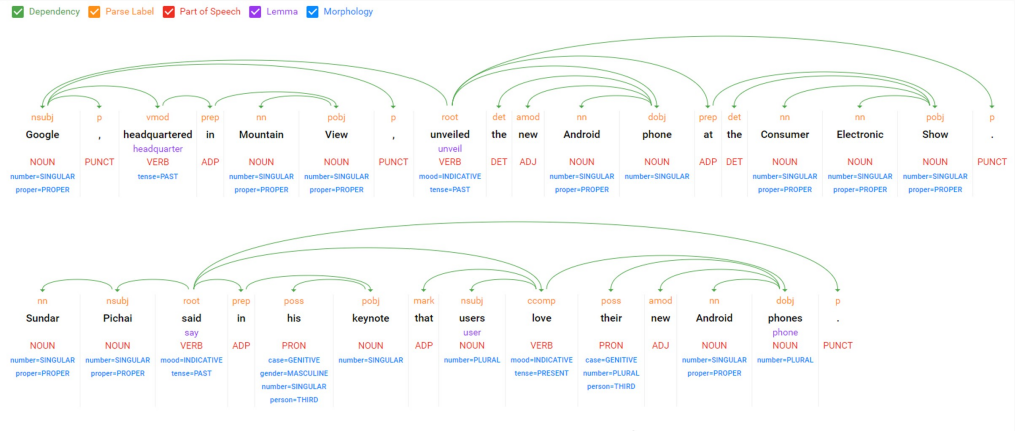
Protein Folding

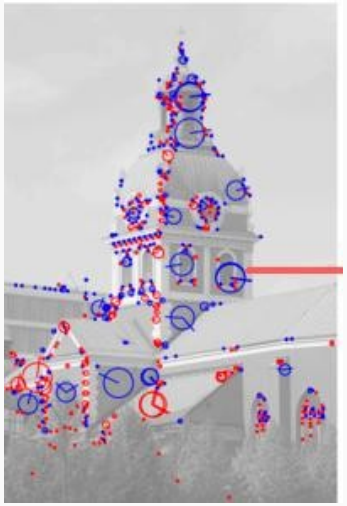


Structured Prediction + Local Information









Structured Prediction Vs. Supervised Learning

Q: This seems "just" **standard supervised learning**, doesn't it?

- Learn $f : \mathcal{X} \to \mathcal{Y}$,
- Given some training examples $(x_i, y_i)_{i=1}^n$.

A: Indeed **it is** supervised learning!

However, standard learning methods **do not apply here...**

What changes is **what we do to learn f.**

Supervised Learning 101

- $\mathcal{X}$ input space, $\mathcal{Y}$ output space,
- $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ loss function,
- ρ **unknown** probability on $\mathcal{X} \times \mathcal{Y}$.

Goal: find a $f^\star : \mathcal{X} \to \mathcal{Y}$ minimizing the **expected risk**...

$$f^\star = \operatorname*{argmin}_{f:\mathcal{X}\to\mathcal{Y}} \mathcal{E}(f), \qquad \mathcal{E}(f) = \mathbb{E}[\ell(f(x), y)],$$

...given **only** the dataset $(x_i, y_i)_{i=1}^n$ sampled independently from ρ.

Supervised Learning 101 - A Wishlist

In practice, we learn an estimator f_n from data...

...and are interested in controlling its **Excess Risk**

$$\mathcal{E}(f_n) - \mathcal{E}(f^\star)$$

Wish list:

- **Consistency:**

$$\lim_{n \to +\infty} \mathcal{E}(f_n) - \mathcal{E}(f^\star) = 0$$

- **Learning Rates:**

$$\mathcal{E}(f_n) - \mathcal{E}(f^\star) \leq O(n^{-\gamma})$$

$\gamma > 0$ (the larger the better).

Protypical Approach: Empirical Risk Minimization

Solve $$\widehat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \quad \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

Where $\mathcal{F} \subseteq \{f : \mathcal{X} \to \mathcal{Y}\}$ (usually a convex function space)

Protypical Approach: Empirical Risk Minimization

Solve $$\widehat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \quad \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

Where $\mathcal{F} \subseteq \{f : \mathcal{X} \to \mathcal{Y}\}$ (usually a convex function space)

If $\mathcal{Y}$ is a **vector space** (e.g. $\mathcal{Y} = \mathbb{R}$):

- $\mathcal{F}$ is "easy" to choose/optimize over: linear models, Kernel methods, Neural Networks, etc.

Protypical Approach: Empirical Risk Minimization

Solve $$\widehat{f} = \operatorname*{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

Where $\mathcal{F} \subseteq \{f : \mathcal{X} \to \mathcal{Y}\}$ (usually a convex function space)

If $\mathcal{Y}$ is a **vector space** (e.g. $\mathcal{Y} = \mathbb{R}$):

- $\mathcal{F}$ is "easy" to choose/optimize over: linear models, Kernel methods, Neural Networks, etc.

If $\mathcal{Y}$ is a "structured" space:

- How to choose $\mathcal{F}$?
- How to perform optimization over it?
- How to study the statistics of f_n over $\mathcal{F}$?

Prototypical Results: Empirical Risk Minimization

Several results allow to study ERM's *consistency* and *rates* when:

- $\mathcal{Y} = \mathbb{R}^d$ and,
- $\mathcal{F}$ is a "standard" space of functions (e.g. a reproducing kernel Hilbert space).

Examples of techniques/notions involved to obtain these results:

- VC dimension,
- Rademacher & Gaussian complexity,
- Covering numbers,
- Stability,
- Empirical processes,
- ....

Classification as Structured Prediction

What about **classification**?

In binary classification $\mathcal{Y} = \{-1, 1\}$ and linear models **do not** satisfy the requirement $f : \mathcal{X} \to \mathcal{Y}$!

Classification **IS** a structured prediction setting.

Example. Even if we had $f_1, f_2 : \mathcal{X} \to \mathcal{Y}$ linear models with values in $\mathcal{Y}$ of the form $f_1(x) = w_1^\top \phi(x)$ and $f_2(x) = w_2^\top \phi(x)$, there is **no guarantee** that a linear (or even convex) combination

$$f_n(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) = (\alpha_1 w_1 + \alpha_2 w_2)^\top \phi(x) \in \mathcal{Y}$$

for some $\alpha_1, \alpha_2 \in \mathbb{R}$.

So, what to do?

Classification as Structured Prediction

A standard approach in classification is to:

1. Learn a **linear model** $f : \mathcal{X} \to \mathbb{R}$ such that $f(x) = w^\top \phi(x)$,
2. Send $f(x)$ **"back"** to $\mathcal{Y} = \{-1, 1\}$ by taking $\operatorname{sign} f(x)$.

However, this raises two main questions about this:

- Does it... **work?** (you touched upon this question in CW2)
- How to choose an **"analogous"** to sign when $\mathcal{Y}$ is a set of more complicated objects? (for examples graphs)

Structured Prediction Methods

$\mathcal{Y}$ arbitrary: how do we parametrize $\mathcal{F}$ and learn f_n?

Surrogate approaches

+ Clear theory (e.g. convergence and learning rates)
− Only for special cases (classification, ranking, multi-labeling etc.)
[Bartlett et al., 2006, Duchi et al., 2010, Mroueh et al., 2012]

Likelihood estimation methods

+ General algorithmic framework
(e.g. StructSVM [Tsochantaridis et al., 2005])
− Limited Theory (no consistency, see e.g. [Bakir et al., 2007])

Surrogate Approaches

Surrogate approaches generalize classification by defining:

- An **encoding** $\mathsf{c} : \mathcal{Y} \to \mathcal{H}$ into a suitable **linear** (output) feature space $\mathcal{H}$ (e.g. $\mathcal{H} = \mathbb{R}$).
- A **Surrogate loss** $L : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ defining a problem to learn a $g_n : \mathcal{X} \to \mathcal{H}$ over a suitable (linear) **hypotheses space** $\mathcal{G}$

$$g_n = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \ \frac{1}{n} \sum_{i=1}^{n} L(g(x_i), c(y_i)).$$

- A **decoding** $\mathsf{d} : \mathcal{H} \to \mathcal{Y}$, such that the final model is written

$$f_n = \mathsf{d} \circ g_n : \mathcal{X} \to \mathcal{Y} \qquad \text{such that} \qquad f_n(x) = d\big(g_n(x)\big)$$

Binary Classification

Let $\mathcal{Y} = \{-1, 1\}$

- **Encoding** $\mathsf{c} : \mathcal{Y} \to \mathbb{R}$ is the identity map.
- **Surrogate loss** L**:** squared, hinge, logistic, exponential, etc. $g_n : \mathcal{X} \to \mathbb{R}$ is learned by solving a regression problem.
- **Decoding** $\mathsf{d} : \mathbb{R} \to \mathcal{Y}$ is the sign function.

Multi-class Classification

Let $\mathcal{Y} = \{1, 2, \ldots, T\}$ for T classes.

- **Encoding** $\mathsf{c} : \mathcal{Y} \to \mathbb{R}^T$ the one-hot encoding $\mathsf{c}(t) = e_t \in \mathbb{R}^T$
  ($e_t = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$ vector of all zeros but 1 in t-th entry)

- **Surrogate loss** L**:** So-called One-vs-All (OVA) loss

$$L(g(x), e_y) = \sum_{t=1}^{T} \ell(g(x)^{(t)}, e_y^{(t)})$$

  with ℓ squared, hinge, logistic, exponential, etc.

- **Decoding** $\mathsf{d} : \mathbb{R}^T \to \mathcal{Y}$ such that

$$\mathsf{d}(g(x)) = \underset{t=1,\ldots,T}{\operatorname{argmax}} \; e_t^\top g(x).$$

Structured Prediction Methods

$\mathcal{Y}$ arbitrary: how do we parametrize $\mathcal{F}$ and learn f_n?

Surrogate approaches

+ Clear theory (e.g. convergence and learning rates)
− Only for special cases (classification, ranking, multi-labeling etc.)
  [Bartlett et al., 2006, Duchi et al., 2010, Mroueh et al., 2012]

Likelihood estimation methods

+ General algorithmic framework
  (e.g. StructSVM [Tsochantaridis et al., 2005])
− Limited Theory (no consistency, see e.g. [Bakir et al., 2007])

Likelihood Estimation Methods

Likelihood Estimation Methods do what it says on the tin...

Given a dataset of joint $(x_i, y_i)_{i=1}^n$ samples, they aim to:

- First **approximate** $\mathbb{P}(y|x)$ with some $P_n : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$.
  The advantage being that now P_n is **real-valued!** So we can use again linear models:

$$P_n(y, x) = w_n^\top \psi(y, x) \qquad \text{with} \qquad \psi : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$$

- Then, **predict** the most likely output as

$$f_n(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} \; P_n(y, x)$$

Structured Prediction Methods

$\mathcal{Y}$ arbitrary: how do we parametrize $\mathcal{F}$ and learn f_n?

Surrogate approaches

+ Clear theory (e.g. convergence and learning rates)
− Only for special cases (classification, ranking, multi-labeling etc.)
  [Bartlett et al., 2006, Duchi et al., 2010, Mroueh et al., 2012]

Likelihood estimation methods

+ General algorithmic framework
  (e.g. StructSVM [Tsochantaridis et al., 2005])
− Limited Theory (no consistency, see e.g. [Bakir et al., 2007])

Is it possible to have best of both worlds?

general algorithmic framework

$+$

clear theory

Part 2: Implicit Loss Embeddings

Wish List

We would like a method that:

- Is **flexible**: can be applied to (m)any $\mathcal{Y}$ and ℓ.
- Leads to efficient **computations**.
- Has strong **theoretical** guarantees (i.e. consistency, rates)

Angle of Attack

We will try to define a Surrogate framework that can be applied in **general** and not on an ad-hoc basis.

We will try to work backwards by:

- Starting from the characterization of the **ideal solution** $f^\star$ (Since it does not require a choice of $\mathcal{F}$)
- Hopefully find out that $f^\star$ exhibits some **"interesting" structures/patterns**.
- See if we can exploit such pattern to derive some rules to define a **general surrogate framework**.

Ideal solution

Let's study the expected risk of our problem

$$\begin{aligned}\mathcal{E}(f) &= \int \ell(f(x), y)\ d\rho(x, y) \\ &= \int \left(\int \ell(f(x), y)\ d\rho(y|x) \right)\ d\rho_{\mathcal{X}}(x)\end{aligned}$$

We can minimize it pointwise. Then, the best $f^\star : \mathcal{X} \to \mathcal{Y}$ is:

$$f^\star(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \int \ell(z, y)\ d\rho(y|x)$$

$f^\star$ is the point-wise minimizer of the expectation $\mathbb{E}_{y|x}\ \ell(\cdot, y)$ conditioned w.r.t. x

Finite Dimensional Intuition

Consider the finite case, where $\mathcal{Y} = \{1, \ldots, T\}$.

This generalizes the multi-class classification setting:
(since we are not assuming a $0-1$ loss).

Then, **any** $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is represented by a *matrix* $V \in \mathbb{R}^{T \times T}$:

$$\ell(y, z) = V_{yz} = e_y^\top V e_z \qquad \forall y, z \in \mathcal{Y}$$

where e_y is the **one-hot** encoding of y or, more formally, it is the y-th element of the canonical basis of $\mathbb{R}^T$.

The "(bi)linearity" of the loss is very useful...

Finite Dimensional Intuition (cont.)

Going back to the point-wise characterization of $f^\star$

$$\begin{aligned} f^\star(x) &= \operatorname*{argmin}_{z \in \mathcal{Y}} \int \ell(z, y) \, d\rho(y|x) \\ &= \operatorname*{argmin}_{z \in \mathcal{Y}} \int e_z^\top V e_y \, d\rho(y|x) \\ &= \operatorname*{argmin}_{z \in \mathcal{Y}} \; e_z^\top V \underbrace{\int e_y \, d\rho(y|x)}_{g_*(x)}. \end{aligned}$$

Now, let $g_* : \mathcal{X} \to \mathbb{R}^T$ be the conditional expectation of e_y given x,

$$g_*(x) = \mathbb{E}[e_y|x] = \int e_y \, d\rho(y|x)$$

Then:

$$f^\star(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \; e_z^\top V g_*(x)$$

Angle of Attack

We will try to define a Surrogate framework that can be applied in general and not on an ad-hoc basis.

We will try to work backwards by:

- Starting from the characterization of the **ideal solution** $f^\star$ (Since it does not require a choice of $\mathcal{F}$)
- Hopefully find out that $f^\star$ exhibits some **"interesting" structures/patterns**.
- See if we can exploit such pattern to derive some rules to define a general surrogate framework.

Finite Dimensional Intuition (cont.)

Idea: replace $g_* : \mathcal{X} \to \mathbb{R}^T$ in

$$f^\star(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \; e_z^\top V \; g_*(x)$$

Finite Dimensional Intuition (cont.)

Idea: replace $g_* : \mathcal{X} \to \mathbb{R}^T$ in

$$f^\star(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \; e_z^\top V \; g_*(x)$$

. . . with an estimator $g_n : \mathcal{X} \to \mathbb{R}^T$

$$f_n(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \; e_z^\top V \; g_n(x)$$

Finite Dimensional Intuition (cont.)

Idea: replace $g_* : \mathcal{X} \to \mathbb{R}^T$ in

$$f^\star(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \; e_z^\top V \; g_*(x)$$

. . . with an estimator $g_n : \mathcal{X} \to \mathbb{R}^T$

$$f_n(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \; e_z^\top V \; g_n(x)$$

Questions:

- How to **generalize** this approach to any (not only finite) $\mathcal{Y}$?
- How to **learn** g_n?

General Case: Implicit Embeddings

Goal: generalize the intuition from the finite case to any $\mathcal{Y}$.

Definition. *A continuous* $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ *admits an* **Implicit Embedding (IE)** *if there exists a map* $\mathsf{c} : \mathcal{Y} \to \mathcal{H}$ *into a separable Hilbert space* $\mathcal{H}$ *and a linear operator* $V : \mathcal{H} \to \mathcal{H}$ *such that*

$$\ell(z, y) \;=\; \left\langle\, \mathsf{c}(z) \;, V\; \mathsf{c}(y) \,\right\rangle_{\mathcal{H}}.$$

General Case: Implicit Embeddings

Goal: generalize the intuition from the finite case to any $\mathcal{Y}$.

Definition. *A continuous* $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ *admits an* **Implicit Embedding (IE)** *if there exists a map* $\mathsf{c} : \mathcal{Y} \to \mathcal{H}$ *into a separable Hilbert space* $\mathcal{H}$ *and a linear operator* $V : \mathcal{H} \to \mathcal{H}$ *such that*

$$\ell(z, y) \;=\; \left\langle\, \mathsf{c}(z) \;, V\; \mathsf{c}(y) \,\right\rangle_{\mathcal{H}}.$$

- For $V = I$, we recover the notion of *reproducing kernel* !
- Accounts for non positive definite, non-symmetric functions,
- Holds also for infinite dimensional spaces $\mathcal{H}$!

Quite technical definition however. . . when does it hold in practice?

IE Example: Squared Loss

Let $\ell(y, y') = (y - y')^2$ with $\mathcal{Y} = \mathbb{R}$.

(Not really a structured prediction problem, but instructive nevertheless)

Consider the polynomial map $\mathsf{c} : \mathcal{Y} \to \mathbb{R}^3$ such that

$$\mathsf{c}(y) = \begin{pmatrix} y^2 \\ \sqrt{2}y \\ 1 \end{pmatrix} \qquad \text{and take} \qquad V = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Then one can verify that $\ell(y, y') = \mathsf{c}(y)^\top V \mathsf{c}(y')$.

IE Example: Kernel Dependency Estimation (KDE)

KDE losses follow the intuition that reproducing kernels are a "sort" of **measure of similarity** between points (see Lecture 2).

Hence, given a kernel $k : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ on the output space $\mathcal{Y}$...

The KDE Loss of k is defined as $\triangle(y, y') = 1 - k(y, y')$, namely a **measure of "dissimilarity"** between output points.

Exercise: find an explicit choice for c and V. Assume k normalized, i.e. $k(y, y') \equiv 1$ for all $y \in \mathcal{Y}$.

Which loss functions have an IE? [Not Examinable]

- **All Losses** on **discrete** $\mathcal{Y}$ (strings, graphs, orderings, subsets, etc.)
- Typical **Regression** & **Classification** loss:
  least-squares, logistic, hinge, e-insensitive, pinball, etc.
- **Robust estimation** loss:
  absolute value, Huber, Cauchy, German-McLure, "Fair" an L2– L1.
- Distances on **Histograms/Probabilities**:
  The χ2 and the Hellinger distances, Sinkhorn Divergence.
- **KDE**. Loss functions $\triangle(y, y') = 1 - k(y, y')$ k reproducing kernel
- **Diffusion** distances on **Manifolds**:
  The squared diffusion distance induced by the heat kernel (at time t > 0) on a compact Reimannian manifold without boundary.

A few useful sufficient conditions... [Not Examinable]

Theorem 19. *Let $\mathcal{Y}$ be a set. A function $\triangle : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ satisfy Asm. 1 when at least one of the following conditions hold:*

1. *$\mathcal{Y}$ is a finite set, with discrete topology.*
2. *$\mathcal{Y} = [0,1]^d$ with $d \in \mathbb{N}$, and the mixed partial derivative $L(y, y') = \frac{\partial^{2d} \triangle(y_1,\dots,y_d,y'_1,\dots,y'_d)}{\partial y_1,\dots,\partial y_d,\partial y'_1,\dots,\partial y'_d}$ exists almost everywhere, where $y = (y_i)_{i=1}^d, y' = (y'_i)_{i=1}^d \in \mathcal{Y}$, and satisfies*
$$\int_{\mathcal{Y}\times\mathcal{Y}} |L(y,y')|^{1+\epsilon} dy\, dy' < \infty, \quad \text{with} \quad \epsilon > 0. \tag{149}$$
3. *$\mathcal{Y}$ is compact and $\triangle$ is a continuous kernel, or $\triangle$ is a function in the RKHS induced by a kernel K. Here K is a continuous kernel on $\mathcal{Y} \times \mathcal{Y}$, of the form*
$$K((y_1,y_2),(y'_1,y'_2)) = K_0(y_1,y'_1)K_0(y_2,y'_2), \quad \forall y_i, y'_i \in \mathcal{Y}, i = 1,2,$$
with K_0 a bounded and continuous kernel on $\mathcal{Y}$.
4. *$\mathcal{Y}$ is compact and*
$$\mathcal{Y} \subseteq \mathcal{Y}_0, \quad \triangle = \triangle_0|_{\mathcal{Y}},$$
that is the restriction of $\triangle_0 : \mathcal{Y}_0 \times \mathcal{Y}_0 \to \mathbb{R}$ on $\mathcal{Y}$, and $\triangle_0$ satisfies Asm. 1 on $\mathcal{Y}_0$,
5. *$\mathcal{Y}$ is compact and*
$$\triangle(y,y') = f(y)\, \triangle_0\, (F(y), G(y'))g(y'),$$
with F, G continuous maps from $\mathcal{Y}$ to a set $\mathcal{Z}$ with $\triangle_0 : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ satisfying Asm. 1 and $f, g : \mathcal{Y} \to \mathbb{R}$, bounded and continuous.
6. *$\mathcal{Y}$ compact and*
$$\triangle = f(\triangle_1, \dots, \triangle_p),$$
where $f : [-M, M]^d \to \mathbb{R}$ is an analytic function (e.g. a polynomial), $p \in \mathbb{N}$ and $\triangle_1, \dots, \triangle_p$ satisfy Asm. 1 on $\mathcal{Y}$. Here $M \geq \sup_{1\leq i\leq p} \|V_i\| C_i$ where V_i is the operator associated to the loss $\triangle_i$ and C_i is the value that bounds the norm of the feature map ψ_i associated to $\triangle_i$, with $i \in \{1, \dots, p\}$.

Note. Here *Asm. 1* refers to the loss admitting an IE.

Structured Prediction with Implicit Embeddings

Going back to our learning problem...

If ℓ has an **implicit embedding**, then[1]

$$f^\star(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \ \langle \mathsf{c}(z), V \ g_*(x) \rangle_{\mathcal{H}} \,,$$

with $g_* : \mathcal{X} \to \mathcal{H}$ such that

$$g_*(x) = \int \mathsf{c}(y) \ d\rho(y|x),$$

the conditional mean embedding of $\rho(\cdot|x)$ with respect to the *output kernel* $k(z, y) = \langle \mathsf{c}(z), \mathsf{c}(y) \rangle_{\mathcal{H}}$. (see [Song et al., 2009])

[1] by repeating the exact same steps as for finite $\mathcal{Y}$

Learning f_n in the general case

Idea: replace $g_* : \mathcal{X} \to \mathcal{H}$ in

$$f^\star(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \ \langle \mathsf{c}(z), V \ g_*(x) \rangle$$

. . . with an estimator $g_n : \mathcal{X} \to \mathcal{H}$

$$f_n(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \ \langle \mathsf{c}(z), V \ g_n(x) \rangle$$

Questions:

- How to **generalize** this approach to any $\mathcal{Y}$?
- How to **learn** g_n?

Part 3: A General Surrogate Algorithm

Learning f_n in the general case

Idea: replace $g_* : \mathcal{X} \to \mathcal{H}$ in

$$f^\star(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \ \langle \mathsf{c}(z), V \ g_*(x) \rangle$$

. . . with an estimator $g_n : \mathcal{X} \to \mathcal{H}$

$$f_n(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \ \langle \mathsf{c}(z), V \ g_n(x) \rangle$$

Questions:

- How to **generalize** this approach to any $\mathcal{Y}$?
- How to **learn** g_n?

Approximating g_*

What is a good algorithm to learn g_n?

Note that $g_*(x) = \int \mathsf{c}(y)\ d\rho(y|x) = \mathbb{E}_{y|x}[\mathsf{c}(y)]$ is a conditional expectation...

...namely the minimizer of the least-squares risk

$$g_* = \operatorname*{argmin}_{g:\mathcal{X}\to\mathcal{H}} \mathcal{R}(g) \qquad \mathcal{R}(g) = \int \left\| g(x) - \mathsf{c}(y) \right\|^2 d\rho(x,y)$$

Therefore we can take g_n to be the least-squares ERM estimator!

Surrogate Approaches

Going back to our quest to find a general Surrogate approach:

- An **encoding** $\mathsf{c} : \mathcal{Y} \to \mathcal{H}$ into a suitable **linear** (output) feature space $\mathcal{H}$ (e.g. $\mathcal{H} = \mathbb{R}$).
- A **Surrogate loss** $L : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ defining a problem to learn a $g_n : \mathcal{X} \to \mathcal{H}$ over a suitable (linear) **hypotheses space** $\mathcal{G}$

$$g_n = \operatorname*{argmin}_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} \|g(x_i) - c(y_i)\|_{\mathcal{H}}^2.$$

- A **decoding** $\mathsf{d} : \mathcal{H} \to \mathcal{Y}$, such that the final model is written

$$f_n = \mathsf{d} \circ g_n : \mathcal{X} \to \mathcal{Y} \qquad \text{such that} \qquad f_n(x) = d\big(g_n(x)\big)$$

Structured Prediction with Implicit Embeddings (Cont.)

Let $\mathcal{X} = \mathbb{R}^d$. We approximate g_* with $g_n(x) = W_n\ x$

$$W_n \quad = \quad \underset{W \in \mathcal{H} \otimes \mathbb{R}^d}{\operatorname{argmin}} \ \frac{1}{n} \sum_{i=1}^{n} \|\mathsf{c}(y_i) - W x_i\|^2 + \lambda \|W\|_F^2 \ ,$$

- If $\mathcal{H} = \mathbb{R}^T$ we have $W \in \mathbb{R}^T \otimes \mathbb{R}^d = \mathbb{R}^{T \times d}$ is a matrix,
- If $\mathcal{H}$ is infinite dimensional, $W \in \mathcal{H} \otimes \mathbb{R}^d$ is an operator.

Still... the solution is

$$W_n = Y^\top X \ (X^\top X + n\lambda I)^{-1}$$

$X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^n \otimes \mathcal{H}$ the matrices/operators with i-th "row" corresponding to x_i and $\mathsf{c}(y_i)$ respectively.

Aside: Tensor Products (Not Examinable)

Technically speaking, given two linear spaces $\mathcal{H}$ and $\mathcal{F}$, the tensor product $\mathcal{H} \otimes \mathcal{F}$ corresponds to the space of Hilbert-Schmidt operators, namely linear functions $W : \mathcal{F} \to \mathcal{H}$ with norm

$$\|W\|_{HS} = \sup_{\|f\|_{\mathcal{F}} \leq 1, \|h\|_{\mathcal{H}} \leq 1} \langle h, Wf \rangle_{\mathcal{H}} < +\infty$$

However, for what we need in the following, it is sufficient to keep in mind that they essentially behave like "infinite" matrices. Indeed, if $\mathcal{H} = \mathbb{R}^T$ and $\mathcal{H} = \mathbb{R}^d$:

- The space $\mathbb{R}^T \otimes \mathbb{R}^d = \mathbb{R}^{T \times d}$ (just matrices!)
- The Hilbert-Schmidt norm, corresponds to the Frobenius norm $\|W\|_{HS}^2 = \|W\|_F^2 = \sum_{i,j=1}^{T,d} W_{ij}^2$.

Structured Prediction with Implicit Embeddings (Cont.)

Let $\mathcal{X} = \mathbb{R}^d$. We approximate g_* with $g_n(x) = W_n\ x$

$$W_n \quad = \quad \underset{W \in \mathcal{H} \otimes \mathbb{R}^d}{\operatorname{argmin}} \ \frac{1}{n} \sum_{i=1}^{n} \|\mathsf{c}(y_i) - W x_i\|^2 + \lambda \|W\|_F^2 \ ,$$

- If $\mathcal{H} = \mathbb{R}^T$ we have $W \in \mathbb{R}^T \otimes \mathbb{R}^d = \mathbb{R}^{T \times d}$ is a matrix,
- If $\mathcal{H}$ is infinite dimensional, $W \in \mathcal{H} \otimes \mathbb{R}^d$ is an operator.

Structured Prediction with Implicit Embeddings (Cont.)

Let $\mathcal{X} = \mathbb{R}^d$. We approximate g_* with $g_n(x) = W_n\ x$

$$W_n \quad = \quad \underset{W \in \mathcal{H} \otimes \mathbb{R}^d}{\operatorname{argmin}} \ \frac{1}{n} \sum_{i=1}^{n} \|\mathsf{c}(y_i) - W x_i\|^2 + \lambda \|W\|_F^2 \ ,$$

- If $\mathcal{H} = \mathbb{R}^T$ we have $W \in \mathbb{R}^T \otimes \mathbb{R}^d = \mathbb{R}^{T \times d}$ is a matrix,
- If $\mathcal{H}$ is infinite dimensional, $W \in \mathcal{H} \otimes \mathbb{R}^d$ is an operator.

Still. . . the solution is

$$W_n \ = Y^\top X \ (X^\top X + n\lambda I)^{-1}$$

$X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^n \otimes \mathcal{H}$ the matrices/operators with i-th "row" corresponding to x_i and $\mathsf{c}(y_i)$ respectively.

Structured Prediction with Implicit Embeddings (Cont.)

W_n (might) contain infinitely many parameters. However...

$$g_n(x) \;=\; W_n\, x \;=\; Y^\top \underbrace{X\,(X^\top X + n\lambda I)^{-1} x}_{\alpha(x)} \;=\; \sum_{i=1}^{n} \alpha_i(x)\; \mathsf{c}(y_i)\;,$$

where the weights $\alpha : \mathcal{X} \to \mathbb{R}^n$ are such that

$$\alpha(x) \;=\; (\alpha_1(x), \ldots, \alpha_n(x))^\top \;=\; \underbrace{[X(X^\top X + n\lambda I)^{-1}]}_{n \times d \text{ matrix!}}\; x \;\in \mathbb{R}^n.$$

Structured Prediction with Implicit Embeddings (Cont.)

W_n (might) contain infinitely many parameters. However...

$$g_n(x) \;=\; W_n\, x \;=\; Y^\top \underbrace{X\,(X^\top X + n\lambda I)^{-1} x}_{\alpha(x)} \;=\; \sum_{i=1}^{n} \alpha_i(x)\; \mathsf{c}(y_i)\;,$$

where the weights $\alpha : \mathcal{X} \to \mathbb{R}^n$ are such that

$$\alpha(x) \;=\; (\alpha_1(x), \ldots, \alpha_n(x))^\top \;=\; \underbrace{[X(X^\top X + n\lambda I)^{-1}]}_{n \times d \text{ matrix!}}\; x \;\in \mathbb{R}^n.$$

Or, if we have a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

$$\alpha(x) \;=\; (K + n\lambda I)^{-1}\; \mathsf{v}(x) \;\in \mathbb{R}^n.$$

– $K \in \mathbb{R}^{n \times n}$ kernel matrix $K_{ij} = k(x_i, x_j)$

– $\mathsf{v}(x) \in \mathbb{R}^n$ evaluation vector $\mathsf{v}(x)_i = k(x_i, x)$.

Note. If $\mathcal{F}$ is the RKHS associated to the $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, with feature map $\phi : \mathcal{X} \to \mathcal{F}$, then

$$g_n(x) = W_n \ \phi(x) = \sum_{i=1}^{n} \alpha_i(x) \mathsf{c}(y_i), \qquad \alpha(x) = (K + n\lambda I)^{-1} \mathsf{v}(x)$$

is the solution to the generalized problem

$$W_n \ = \ \operatorname*{argmin}_{W \in \mathcal{H} \otimes \mathcal{F}} \ \frac{1}{n} \sum_{i=1}^{n} \|\mathsf{c}(y_i) - W x_i\|^2 + \lambda \|W\|_{HS}^2 \ ,$$

Nothing changes symbolically, but we automatically generalize to non-linear functions!

Surrogate Approaches

The only remaining question is how design a "practical" decoding.

- An **encoding** $\mathsf{c} : \mathcal{Y} \to \mathcal{H}$ into a suitable **linear** (output) feature space $\mathcal{H}$ (e.g. $\mathcal{H} = \mathbb{R}$).
- A **Surrogate (squared) loss** $L : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ defining a problem to learn a $g_n : \mathcal{X} \to \mathcal{H}$ over a suitable $\mathcal{G} = \mathcal{H} \otimes \mathcal{F}$

$$g_n = \operatorname*{argmin}_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} \|g(x_i) - c(y_i)\|_{\mathcal{H}}^2 + \lambda \|g\|_{\mathcal{G}}^2$$

- A **decoding** $\mathsf{d} : \mathcal{H} \to \mathcal{Y}$, such that the $f_n = \mathsf{d} \circ g_n$ with

$$\mathsf{d} \circ g = \operatorname*{argmin}_{y \in \mathcal{Y}} \langle \mathsf{c}(y), V g(x) \rangle$$

Learning f_n in the general case

Recall that our idea is to replace $g_* : \mathcal{X} \to \mathcal{H}$ in

$$f^\star(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \ \langle \mathsf{c}(z), V \ g_*(x) \rangle$$

. . . with an estimator $g_n : \mathcal{X} \to \mathcal{H}$

$$f_n(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \ \langle \mathsf{c}(z), V \ g_n(x) \rangle$$

Since we now have an explicit characterization of g_n as

$$g_n(x) = \sum_{i=1}^{n} \alpha_i(x) \mathsf{c}(y_i)$$

can we say anything about f_n?

Structured Prediction with Implicit Embeddings (Cont.)

Analogously to the case of $f^\star$ and $f^\star \ldots$

$$
\begin{aligned}
f_n(x) &= \operatorname*{argmin}_{z \in \mathcal{Y}} \ \langle \mathsf{c}(y), V \ g_n(x) \rangle \\
&= \operatorname*{argmin}_{z \in \mathcal{Y}} \ \left\langle \mathsf{c}(y), \ V \ \left(\sum_{i=1}^{n} \alpha_i(x) \ \mathsf{c}(y_i) \right) \right\rangle \\
&= \operatorname*{argmin}_{z \in \mathcal{Y}} \ \sum_{i=1}^{n} \alpha_i(x) \ \underbrace{\langle \mathsf{c}(z), V \mathsf{c}(y_i) \rangle}_{\substack{\ell(z, y_i) \\ \textbf{loss trick}}}
\end{aligned}
$$

In other words,

$$
f_n(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \ \sum_{i=1}^{n} \alpha_i(x) \ \ell(z, y_i)
$$

The “loss trick”

$$f_n(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \sum_{i=1}^{n} \alpha_i(x)\ \ell(z, y_i)$$

Analogous to the “kernel trick”, the implicit embedding enables us to find an estimator $f_n : \mathcal{X} \to \mathcal{Y} \ldots$

without need for explicit knowledge of $(\mathcal{H}, \mathsf{c}, V)$*!*

To sum up...

This approach has two phases:

- **Learning.** Where the score function $\alpha : \mathcal{X} \to \mathbb{R}^n$ is estimated.

- **Prediction.** Where we need to solve

$$f_n(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \sum_{i=1}^{n} \alpha_i(x)\ \ell(z, y_i)$$

Note. One needs to know how to minimize ℓ over $\mathcal{Y}$ (which can be hard!).

Surrogate Approaches

We have finally found our general Surrogate framework!

- An **encoding** $\mathsf{c} : \mathcal{Y} \to \mathcal{H}$ into a suitable **linear** (output) feature space $\mathcal{H}$ (e.g. $\mathcal{H} = \mathbb{R}$).
- A **Surrogate loss** $L : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ defining a problem to learn a $g_n : \mathcal{X} \to \mathcal{H}$ over a suitable (linear) **hypotheses space** $\mathcal{G}$
- A **decoding** $\mathsf{d} : \mathcal{H} \to \mathcal{Y}$, such that the $f_n = \mathsf{d} \circ g_n$ with

$$\mathsf{d} \circ g_n = \operatorname*{argmin}_{y \in \mathcal{Y}} \sum_{i=1}^{n} \alpha_i(x) \ell(y, y_i)$$

for suitable α_i.

Does this strategy have any theoretical guarantee?

Part 4: Theoretical Properties

Surrogate Approaches

We have finally found our general Surrogate framework!

- An **encoding** $\mathsf{c} : \mathcal{Y} \to \mathcal{H}$ into a suitable **linear** (output) feature space $\mathcal{H}$ (e.g. $\mathcal{H} = \mathbb{R}$).
- A **Surrogate loss** $L : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ defining a problem to learn a $g_n : \mathcal{X} \to \mathcal{H}$ over a suitable (linear) **hypotheses space** $\mathcal{G}$
- A **decoding** $\mathsf{d} : \mathcal{H} \to \mathcal{Y}$, such that the $f_n = \mathsf{d} \circ g_n$ with

$$\mathsf{d} \circ g_n = \operatorname*{argmin}_{y \in \mathcal{Y}} \sum_{i=1}^{n} \alpha_i(x) \ell(y, y_i)$$

for suitable α_i.

Does this strategy have any theoretical guarantee?

Comparison Inequality

Theorem. Let ℓ admit an IE. Let $Q = \sup_y \|\mathsf{c}(y)\|_{\mathcal{H}}$ and $\|V\| = \sup_{\|h\|_{\mathcal{H}} \leq 1} \|Vh\|_{\mathcal{H}}$ the "operator" norm[2] of V. Then,

$$\mathcal{E}(\mathsf{d} \circ g) - \mathcal{E}(f_*) \leq 2Q\|V\| \sqrt{\mathcal{R}(g) - \mathcal{R}(g_*)}$$

where

$$\mathcal{E}(f) = \int \ell(f(x), y)\, d\rho(x, y), \quad \mathcal{R}(g) = \int \|g(x) - \mathsf{c}(y)\|_{\mathcal{H}}^2\, d\rho(x, y)$$

Direct consequences:

- If $\mathcal{R}(g_n) \to \mathcal{R}(g_*)$ then $\mathcal{E}(f_n) \to \mathcal{E}(f_*)$ for $f_n = \mathsf{d} \circ g_n$!
- Learning rates for g_n translate automatically[3] to f_n!

[2] If V is a matrix, $\|V\|$ corresponds to the largest singular value of V.

[3] Admittedly slowed down by a squared root

Proof - Comparison Inequality

Denote $f = \mathsf{d} \circ g$. Since ℓ has an IE, the excess risk becomes

$$\begin{aligned}
\mathcal{E}(f) - \mathcal{E}(f^\star) &= \int_{\mathcal{X}\times\mathcal{Y}} \langle \mathsf{c}(f(x)) - \mathsf{c}(f^\star(x)), V\,\mathsf{c}(y)\rangle \; d\rho(x,y) \\
&= \int_{\mathcal{X}} \left\langle \mathsf{c}(f(x)) - \mathsf{c}(f^\star(x)), V \left(\int_Y \mathsf{c}(y)\; d\rho(y|x) \right) \right\rangle \; d\rho_{\mathcal{X}}(x) \\
&= \int_{\mathcal{X}} \langle \mathsf{c}(f(x) - \mathsf{c}(f^\star(x)), V\, g_*(x)\rangle \; d\rho_{\mathcal{X}}(x)
\end{aligned}$$

by decomposing ρ in its conditional and margin probabilities.

Proof - Comparison Inequality

We now add and remove the term $\langle \mathsf{c}(f(x)), Vg(x) \rangle$ to

$$\begin{aligned}
&\langle \mathsf{c}(f(x)) - \mathsf{c}(f^\star(x)), Vg_*(x) \rangle \pm \langle \mathsf{c}(f(x)), Vg(x) \rangle \\
&\qquad = \langle \mathsf{c}(f(x)), Vg(x) \rangle - \langle \mathsf{c}(f^\star(x)), Vg_*(x) \rangle \\
&\qquad\quad + \langle \mathsf{c}(f(x)), V(g_*(x) - g(x)) \rangle
\end{aligned}$$

Proof - Comparison Inequality

We now add and remove the term $\langle \mathsf{c}(f(x)), Vg(x)\rangle$ to

$$\begin{aligned}\langle \mathsf{c}(f(x)) - \mathsf{c}(f^\star(x)), Vg_*(x)\rangle &\pm \langle \mathsf{c}(f(x)), Vg(x)\rangle \\ &= \langle \mathsf{c}(f(x)), Vg(x)\rangle - \langle \mathsf{c}(f^\star(x)), Vg_*(x)\rangle \\ &\quad + \langle \mathsf{c}(f(x)), V(g_*(x) - g(x))\rangle\end{aligned}$$

Hence, integrating wrt $\rho_{\mathcal{X}}$, we conclude

$$\mathcal{E}(f) - \mathcal{E}(f^\star) = A + B$$

with

$$A = \int \langle \mathsf{c}(f(x)), Vg(x)\rangle - \langle \mathsf{c}(f^\star(x)), Vg_*(x)\rangle \; d\rho_{\mathcal{X}}(x)$$

$$B = \int \langle \mathsf{c}(f(x)), V(g_*(x) - g(x))\rangle \; d\rho_{\mathcal{X}}(x)$$

Proof - Comparison Inequality - Term A

To control A, recall the definitions of decoding for f and $f^\star$:

$$\mathsf{d} \circ g = \operatorname*{argmin}_{y \in \mathcal{Y}} \ \langle \mathsf{c}(y), V g(x) \rangle$$

Then,

$$\langle \mathsf{c}(f(x)), V g(x) \rangle = \min_{y \in \mathcal{Y}} \langle \mathsf{c}(y), V g(x) \rangle$$
$$\langle \mathsf{c}(f^\star(x)), V g_*(x) \rangle = \min_{y \in \mathcal{Y}} \langle \mathsf{c}(y), V g(x) \rangle$$

Implying that A is

$$A = \int \min_{y \in \mathcal{Y}} \langle \mathsf{c}(y), V g(x) \rangle - \min_{y \in \mathcal{Y}} \langle \mathsf{c}(y), V g_*(x) \rangle \ d\rho_{\mathcal{X}}(x)$$

Proof - Comparison Inequality - Term A (II)

Since for any two functions $u, v : \mathcal{Y} \to \mathbb{R}$

$$\min_y u(y) - \min_y v(y) \leq \sup_y |u(y) - v(y)|,$$

We conclude that

$$\begin{aligned} A &= \int \min_{y \in \mathcal{Y}} \langle \mathsf{c}(y), V g(x) \rangle - \min_{y \in \mathcal{Y}} \langle \mathsf{c}(y), V g_*(x) \rangle \ d\rho_{\mathcal{X}}(x) \\ &\leq \int \sup_{y \in \mathcal{Y}} \ |\langle \mathsf{c}(y), V(g_*(x) - g(x)) \rangle| \ d\rho_{\mathcal{X}}(x). \end{aligned}$$

Proof - Comparison Inequality - Term B

By the convexity of the absolute value, the term B can be controlled as

$$\begin{aligned} B &= \int \langle \mathsf{c}(y), V(g_*(x) - g(x)) \rangle \ d\rho_{\mathcal{X}}(x) \\ &\leq \left| \int \langle \mathsf{c}(y), V(g_*(x) - g(x)) \rangle \ d\rho_{\mathcal{X}}(x) \right| \\ &\leq \int \sup_{y \in \mathcal{Y}} \ |\langle \mathsf{c}(y), V(g_*(x) - g(x)) \rangle| \ d\rho_{\mathcal{X}}(x) \end{aligned}$$

Proof - Comparison Inequality - Term B

By the convexity of the absolute value, the term B can be controlled as

$$\begin{aligned} B &= \int \langle \mathsf{c}(y), V(g_*(x) - g(x)) \rangle \ d\rho_{\mathcal{X}}(x) \\ &\leq \left| \int \langle \mathsf{c}(y), V(g_*(x) - g(x)) \rangle \ d\rho_{\mathcal{X}}(x) \right| \\ &\leq \int \sup_{y \in \mathcal{Y}} \ |\langle \mathsf{c}(y), V(g_*(x) - g(x)) \rangle| \ d\rho_{\mathcal{X}}(x) \end{aligned}$$

The same quantity we obtained for term A! Hence,

$$\begin{aligned} \mathcal{E}(f) - \mathcal{E}(f^\star) &= A + B \\ &\leq 2 \int \sup_{y \in \mathcal{Y}} \ |\langle \mathsf{c}(y), V(g_*(x) - g(x)) \rangle| \ d\rho_{\mathcal{X}}(x) \end{aligned}$$

Proof - Comparison Inequality (Continued)

Now, by Cauchy-Schwartz inequality

$$\begin{aligned}
&\int \sup_{y\in\mathcal{Y}} \ |\langle \mathsf{c}(y), V(g_*(x) - g(x))\rangle| \ d\rho_{\mathcal{X}}(x) \\
&\qquad \leq \int \sup_{y\in\mathcal{Y}} \|V^*\mathsf{c}(y)\|_{\mathcal{H}} \|g_*(x) - g(x)\|_{\mathcal{H}} \ d\rho_{\mathcal{X}}(x) \\
&\qquad \leq Q\|V\| \int \|g_*(x) - g(x)\| \ d\rho_{\mathcal{X}}(x)
\end{aligned}$$

Proof - Comparison Inequality (Continued)

Now, by Cauchy-Schwartz inequality

$$\begin{aligned}
&\int \sup_{y\in\mathcal{Y}} \ |\langle \mathsf{c}(y), V(g_*(x) - g(x))\rangle| \ d\rho_{\mathcal{X}}(x) \\
&\qquad \leq \int \sup_{y\in\mathcal{Y}} \|V^*\mathsf{c}(y)\|_{\mathcal{H}} \|g_*(x) - g(x)\|_{\mathcal{H}} \ d\rho_{\mathcal{X}}(x) \\
&\qquad \leq Q\|V\| \int \|g_*(x) - g(x)\| \ d\rho_{\mathcal{X}}(x)
\end{aligned}$$

since, by dividing and multiplying by $\|\mathsf{c}(y)\|_{\mathcal{H}}$,

$$\sup_{y\in\mathcal{Y}} \|V^*\mathsf{c}(y)\|_{\mathcal{H}} = \underbrace{\sup_{y\in\mathcal{Y}} \|\mathsf{c}(y)\|_{\mathcal{H}}}_{=Q} \qquad \underbrace{\left\| V^* \frac{\mathsf{c}(y)}{\|\mathsf{c}(y)\|_{\mathcal{H}}} \right\|_{\mathcal{H}}}_{\leq \|V\|}$$

since $\mathsf{c}(y)/\|\mathsf{c}(y)\|_{\mathcal{H}}$ has norm 1

Proof - Comparison Inequality (Continued)

Finally, by Jensen's inequality[4],

$$\begin{aligned}\int \|g_*(x) - g(x)\| \, d\rho_{\mathcal{X}}(x) &= \int \sqrt{\|g_*(x) - g(x)\|^2} \, d\rho_{\mathcal{X}}(x) \\ &\leq \sqrt{\int \|g_*(x) - g(x)\|^2 \, d\rho_{\mathcal{X}}(x)} \\ &= \sqrt{\mathcal{R}(g(x)) - \mathcal{R}(g_*(x))}\end{aligned}$$

[4] for any $u : \mathcal{X} \to \mathbb{R}$, we have $\mathbb{E}[\sqrt{f(x)}] \leq \sqrt{\mathbb{E}[f(x)]}$

Proof - Comparison Inequality - Conclusion

We conclude that

$$\mathcal{E}(f) - \mathcal{E}(f^\star) \leq 2Q\|V\|\sqrt{\mathcal{R}(g(x)) - \mathcal{R}(g_*(x))}$$

As desired.

Comparison Inequality

Theorem. Let ℓ admit an IE. Let $Q = \sup_y \|\mathsf{c}(y)\|_{\mathcal{H}}$ and $\|V\| = \sup_{\|h\|_{\mathcal{H}} \leq 1} \|Vh\|_{\mathcal{H}}$. Then,

$$\mathcal{E}(\mathsf{d} \circ g) - \mathcal{E}(f_*) \leq 2Q\|V\| \sqrt{\mathcal{R}(g) - \mathcal{R}(g_*)}$$

Direct consequences:

- If $\mathcal{R}(g_n) \to \mathcal{R}(g_*)$ then $\mathcal{E}(f_n) \to \mathcal{E}(f_*)$ for $f_n = \mathsf{d} \circ g_n$!
- Learning rates for g_n translate automatically to f_n!
  (Admittedly slowed down by a squared root)

Universal consistency

Theorem (Universal Consistency) Let $\mathcal{X}, \mathcal{Y}$ compact ℓ admit an implicit embedding and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a universal kernel[5]. Choose $\lambda = n^{-1/2}$ to train f_n. Then,

$$\lim_{n \to +\infty} \mathcal{E}(f_n) - \mathcal{E}(f^\star) = 0,$$

with probability 1.

[5]Technical requirement. Use e.g. the Gaussian kernel $k(x, x') = e^{-\left\|x - x'\right\|^2 / \sigma}$.

Learning Rates

Theorem (Learning Rates). Let $\mathcal{X}, \mathcal{Y}$ compact ℓ admit an implicit embedding. Use $\lambda = n^{-1/2}$ to train f_n. Then, $\forall \delta \in (0, 1)$

$$\mathcal{E}(f_n) - \mathcal{E}(f^\star) \leq Q\|V\| \ \log(1/\delta) \ \frac{1}{n^{1/4}},$$

hold with probability at least $1 - \delta$.

Learning Rates

Theorem (Learning Rates). Let $\mathcal{X}, \mathcal{Y}$ compact ℓ admit an implicit embedding. Use $\lambda = n^{-1/2}$ to train f_n. Then, $\forall \delta \in (0,1)$

$$\mathcal{E}(f_n) - \mathcal{E}(f^\star) \leq Q\|V\| \ \log(1/\delta) \ \frac{1}{n^{1/4}},$$

hold with probability at least $1 - \delta$.

Comments.

- Same rates as worst-case binary classification (better rates with Tsibakov-like noise assumptions [Nowak-Vila et al., 2018]).

Wish List

Going back to our wishlist. This strategy:

- Is **flexible**: can be applied to (m)any $\mathcal{Y}$ and ℓ. ✓
- Leads to efficient **computations**.
  - Yes in training, ✓
  - It depends on $\mathcal{Y}$ and ℓ at test time.
- Has strong **theoretical** guarantees (i.e. consistency, rates) ✓

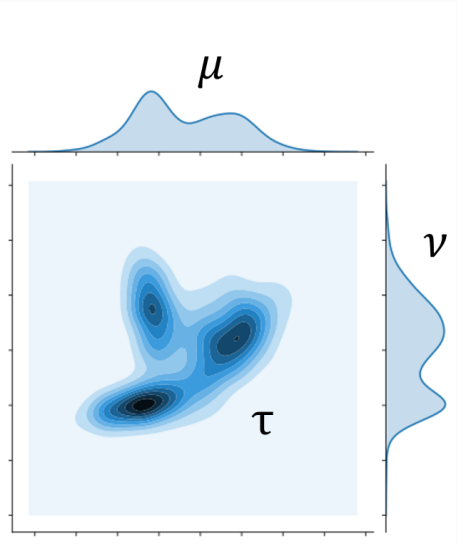
Example Applications

Predicting Probability Distributions

Setting: $\mathcal{Y} = \mathcal{P}(\mathbb{R}^d)$ probability distributions on $\mathbb{R}^d$.

Loss: Wasserstein distance

$$\ell(\mu, \nu) = \min_{\tau \in \Pi(\mu,\nu)} \int \|z - y\|^2 \, d\tau(x, y)$$



Digit Reconstruction

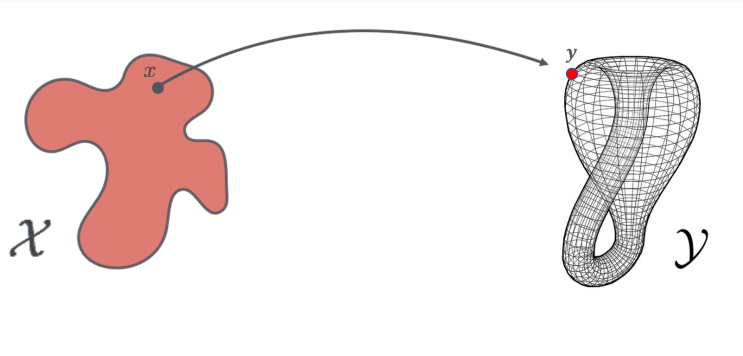
	Reconstruction Error (%)			
# Classes	**Ours**	$\widetilde{S}_\lambda$	Hell	KDE
2	**3.7 ± 0.6**	4.9 ± 0.9	8.0 ± 2.4	12.0 ± 4.1
4	**22.2 ± 0.9**	31.8 ± 1.1	29.2 ± 0.8	40.8 ± 4.2
10	**38.9 ± 0.9**	44.9 ± 2.5	48.3 ± 2.4	64.9 ± 1.4

Manifold Regression

Setting: $\mathcal{Y}$ Riemmanian manifold.

Loss: (squared) geodesic distance.

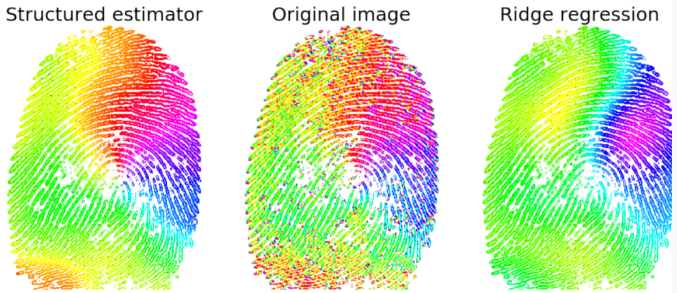
Optimization: Riemannian GD.



Fingerprint Reconstruction

($\mathcal{Y} = S^1$ sphere)

	Δ Deg.
KRLS	26.9 ± 5.4
MR [33]	22 ± 6
SP (ours)	$\mathbf{18.8 \pm 3.9}$



Multi-labeling

($\mathcal{Y}$ statistical manifold)

	KRLS	SP (Ours)
Emotions	0.63	**0.73**
CAL500	**0.92**	**0.92**
Scene	0.62	**0.73**

Nonlinear Multi-task Learning

Idea: instead of solving multiple learning problems (tasks) separately, *leverage the potential relations among them.*

Previous Methods: only imposing/learning **linear** tasks relations.

Unable to cope with non-linear constraints (e.g. ranking, robotics, etc.).

MTL+Structured Prediction

- Interpret multiple tasks as separate outputs.
- Impose constraints as structure on the joint output.

	ml100k	sushi
MART	0.499 (±0.050)	0.477 (±0.100)
RankNet	0.525 (±0.007)	0.588 (±0.005)
RankBoost	0.576 (±0.043)	0.589 (±0.010)
AdaRank	0.509 (±0.007)	0.588 (±0.051)
Coordinate Ascent	0.477 (±0.108)	0.473 (±0.103)
LambdaMART	0.564 (±0.045)	0.571 (±0.076)
ListNet	0.532 (±0.030)	0.588 (±0.005)
Random Forests	0.526 (±0.022)	0.566 (±0.010)
SVMrank	0.513 (±0.008)	0.541 (±0.005)
Ours	**0.333 (±0.005)**	**0.286 (±0.006)**

Additional Work

Case studies:

- Learning to rank [Korba et al., 2018]
- Output Fisher Embeddings [Djerrab et al., 2018]
- $\mathcal{Y} =$ manifolds, $\ell =$ geodesic distance [Rudi et al., 2018]
- $\mathcal{Y} =$ probability space, $\ell =$ wasserstein distance [Luise et al., 2018]

Refinements of the analysis:

- Alternative derivations [Osokin et al., 2017]
- Discrete loss [Nowak-Vila et al., 2018, Struminsky et al., 2018]

Extensions:

- Application to multitask-learning [Ciliberto et al., 2017]
- Beyond least squares surrogate [Nowak-Vila et al., 2019]
- Regularizing with trace norm [Luise et al., 2019]

Wrapping Up

- Structrued prediction problems pose new challenges than standard Supervised Learning.
- Inspired by classification settings this example, we tried to design a general Surrogate framework.
- We introduced the notion of Implicit Embedding and observed that it leads automatically to a choice of surrogate embedding and loss.
- We derived a practical algorithm to solve the surrogate problem and then go "back" to the structured space.
- We reported a few results that guarantee consistency and rates of the estimator (in particular a comparison inequality).
- We discussed a few sample applications.

References i

G. H. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. V. N. Vishwanathan. *Predicting Structured Data*. MIT Press, 2007.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Carlo Ciliberto, Alessandro Rudi, Lorenzo Rosasco, and Massimiliano Pontil. Consistent multitask learning with nonlinear output relations. In *Advances in Neural Information Processing Systems*, pages 1983–1993, 2017.

Moussab Djerrab, Alexandre Garcia, Maxime Sangnier, and Florence d'Alché Buc. Output fisher embedding regression. *Machine Learning*, 107(8-10):1229–1256, 2018.

John C. Duchi, Lester W. Mackey, and Michael I. Jordan. On the consistency of ranking algorithms. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 327–334, 2010.

Anna Korba, Alexandre Garcia, and Florence d'Alché Buc. A structured prediction approach for label ranking. In *Advances in Neural Information Processing Systems*, pages 8994–9004, 2018.

References ii

Giulia Luise, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. Differential properties of sinkhorn approximation for learning with wasserstein distance. In *Advances in Neural Information Processing Systems*, pages 5859–5870, 2018.

Giulia Luise, Dimitris Stamos, Massimiliano Pontil, and Carlo Ciliberto. Leveraging low-rank relations between surrogate tasks in structured prediction. *International Conference on Machine Learning (ICML)*, 2019.

Youssef Mroueh, Tomaso Poggio, Lorenzo Rosasco, and Jean-Jacques Slotine. Multiclass learning with simplex coding. In *Advances in Neural Information Processing Systems (NIPS) 25*, pages 2798–2806, 2012.

Alex Nowak-Vila, Francis Bach, and Alessandro Rudi. Sharp analysis of learning with discrete losses. *AISTATS*, 2018.

Alex Nowak-Vila, Francis Bach, and Alessandro Rudi. A general theory for structured prediction with smooth convex surrogates. *arXiv preprint arXiv:1902.01958*, 2019.

Anton Osokin, Francis Bach, and Simon Lacoste-Julien. On structured prediction theory with calibrated convex surrogate losses. In *Advances in Neural Information Processing Systems*, pages 302–313, 2017.

References iii

Alessandro Rudi, Carlo Ciliberto, GianMaria Marconi, and Lorenzo Rosasco. Manifold structured prediction. In *Advances in Neural Information Processing Systems*, pages 5610–5621, 2018.

Le Song, Jonathan Huang, Alex Smola, and Kenji Fukumizu. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 961–968. ACM, 2009.

Kirill Struminsky, Simon Lacoste-Julien, and Anton Osokin. Quantifying learning guarantees for convex but inconsistent surrogates. In *Advances in Neural Information Processing Systems*, pages 669–677, 2018.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. volume 6, pages 1453–1484, 2005.