An Introduction to Structured Prediction

Carlo Ciliberto

Computer Science
University College London

Outline

Structured prediction: what & why?

Structured Prediction

$$\mathcal{X} \xrightarrow{f} \mathcal{Y}$$

Image Captioning
(also Localization
Segmentation
Classification)

→

Movie Ranking

NETFLIX
user:127

→

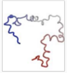
$> \cdots >$

$> \cdots >$

Speech Recognition

→ "Ok Google"

Protein Folding



→

Structured Prediction Vs. Supervised Learning

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

- Learn $f : \mathcal{X} \to \mathcal{Y}$,
- Given many 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 $f^\star : \mathcal{X} \to \mathcal{Y}$

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

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

Protypical Approach: Empirical Risk Minimization

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

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

Protypical Approach: Empirical Risk Minimization

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

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

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

- $\mathcal{F}$ easy to choose/optimize over: (generalized) 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. $\mathcal{Y} = \mathbb{R}$):

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

Example: **Linear models**. $\mathcal{X} = \mathbb{R}^d$

- $f(x) = w^\top x$ for some $w \in \mathbb{R}^d$.

Empirical Risk Minimization (ERM)

We are interested in controlling the **Excess Risk** of $\widehat{f}$

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

Wish list:

- **Consistency:**

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

- **Learning Rates:**

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

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

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,
- ....

Protypical Approach: Empirical Risk Minimization

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

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

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

- $\mathcal{F}$ easy to choose/optimize over: (generalized) 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 $\widehat{f}$ over $\mathcal{F}$?

Structured Prediction Methods

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

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)

Score learning techniques

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

Surrogate Frameworks

Example: Binary Classification setting

Binary Classification:

- "any" input space $\mathcal{X}$
- output space $\mathcal{Y} = \{-1, 1\}$
- 0-1 loss function, i.e $\ell(y, y') = \mathbf{1}_{\{y \neq y'\}} = \begin{cases} 0 & \text{if } y = y \\ 1 & \text{otherwise} \end{cases}$

Example: Binary Classification Problem

- A classification rule is a map $f : \mathcal{X} \to \mathcal{Y}$
- The risk of a rule f is $\mathcal{E}(f) = \mathbb{E}_{(x,y)\sim\rho}[\mathbf{1}_{\{f(x)\neq y\}}]$.
- The classification rule that *minimizes* $\mathcal{E}$ is

$$f^* : \mathcal{X} \to \mathcal{Y}, \qquad f^*(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} \ \rho(y \mid x).$$

- Why? Exercise :)

Example: Binary Classification

Goal: approximate f^* given a training set $(x_i, y_i)_{i=1}^n$.

Issues:

i) $\mathcal{Y}$ is **not** linear! $\Rightarrow \mathcal{H} = \{$classification rules$\}$ is **not** linear!

i) $\ell(y, y') = \mathbf{1}_{\{y \neq y'\}}$ is **not** convex $\Rightarrow$ very **hard** to minimize!

Example: Binary Classification

Idea:

i) Rephrase the problem using a **linear** output space,

ii) Find a good **convex** “replacement” for ℓ.

Example: Binary Classification

Idea:

i) Rephrase the problem using a **linear** output space,

ii) Find a good **convex** "replacement" for ℓ.

i) Replace $\mathcal{Y} = \{-1, 1\}$ to $\mathbb{R}$ and consider functions $g : \mathcal{X} \to \mathbb{R}$ (*surrogate* classification rule)

Example: Binary Classification

Idea:

✓ Rephrase the problem using a **linear** output space,

ii) Find a good **convex** "replacement" for ℓ.

i) Replace $\mathcal{Y} = \{-1, 1\}$ to $\mathbb{R}$ and consider functions $g : \mathcal{X} \to \mathbb{R}$ (*surrogate* classification rule)

Example: Binary Classification

Idea:

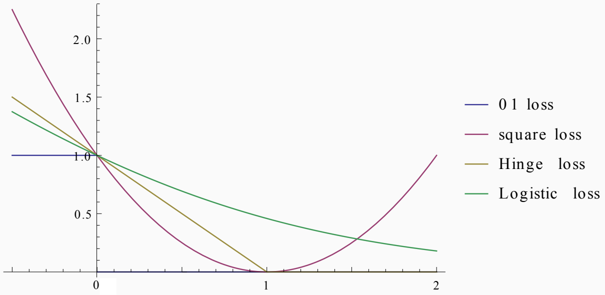
- ✓ Rephrase the problem using a **linear** output space,
- ✓ Find a good **convex** "replacement" for ℓ.

i) Replace $\mathcal{Y} = \{-1, 1\}$ to $\mathbb{R}$ and consider functions $g : \mathcal{X} \to \mathbb{R}$ ("surrogate" classification rule)

ii) Replace $\ell(y, y') = \mathbf{1}_{\{y \neq y'\}}$ with $\mathcal{L}$: $\mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ non-negative convex "surrogate" loss: e.g. logistic, least squares, hinge.

Loss Functions for Binary Classification

Loss functions of the form $\mathcal{L}(y, y') = \tilde{\mathcal{L}}(y \cdot y')$



Surrogate ERM

The loss $\mathcal{L}$ induces a *surrogate* risk

$$\mathcal{R}(g) = \mathbb{E}_{(x,y)\sim\rho}\ \mathcal{L}(g(x), y).$$

and can define the surrogate ERM estimator

$$\widehat{g} = \operatorname*{argmin}_{g\in\mathcal{G}}\ \mathcal{R}_n(g) \qquad \mathcal{R}_n(g) = \frac{1}{n}\sum_{i=1}^{n}\ \mathcal{L}(g(x_i), y_i).$$

Modeling. The output space is linear $\Rightarrow$ many options for $\mathcal{G}$!

Optimization. The loss is convex $\Rightarrow$ we can efficiently find $\widehat{g}$!

Statistics. Standard results $\Rightarrow$ generalization properties of $\widehat{g}$!

$$\mathcal{R}(\widehat{g}) - \mathcal{R}(g^\star) \to 0$$

Surrogate Vs. Original Problems

This is all nice and well, but ...

- How can we go from $\widehat{g} : \mathcal{X} \to \mathbb{R}$ to some $\widehat{f} : \mathcal{X} \to \mathcal{Y}$?

- How is $g^\star$ related to $f^\star$?

- Are surrogate learning rates for $\widehat{g}$ of any use?

Surrogate Vs. Original Problems

This is all nice and well, but ...

- How can we go from $\widehat{g}: \mathcal{X} \to \mathbb{R}$ to some $\widehat{f}: \mathcal{X} \to \mathcal{Y}$?
  Standard approach: $\widehat{f}(x) = \mathrm{sign}(\widehat{g}(x))$
- How is $g^\star$ related to $f^\star$?
- Are surrogate learning rates for $\widehat{g}$ of any use?

Surrogate Vs. Original Problems

This is all nice and well, but ...

- How can we go from $\widehat{g} : \mathcal{X} \to \mathbb{R}$ to some $\widehat{f} : \mathcal{X} \to \mathcal{Y}$?
  Standard approach: $\widehat{f}(x) = \operatorname{sign}(\widehat{g}(x))$
- How is $g^\star$ related to $f^\star$?
  Exercise. $f^\star(x) = \operatorname{sign}(g^\star(x))$!
- Are surrogate learning rates for $\widehat{g}$ of any use?

Surrogate Vs. Original Problems

This is all nice and well, but ...

- How can we go from $\widehat{g} : \mathcal{X} \to \mathbb{R}$ to some $\widehat{f} : \mathcal{X} \to \mathcal{Y}$?
  Standard approach: $\widehat{f}(x) = \mathrm{sign}(\widehat{g}(x))$
- How is $g^\star$ related to $f^\star$?
  Exercise. $f^\star(x) = \mathrm{sign}(g^\star(x))$!
- Are surrogate learning rates for $\widehat{g}$ of any use?
  Theorem.
  $$\mathcal{E}(\widehat{f}) - \mathcal{E}(f^\star) \ \leq\ \varphi\Big(\mathcal{R}(\widehat{g}) - \mathcal{R}(g^\star)\Big).$$
  (where $\varphi : \mathbb{R} \to \mathbb{R}_+$ depends on the surrogate loss $\mathcal{L}$).

Example: Multiclass Classification setting

Multiclass Classification:

- input space $\mathcal{X}$
- output space $\mathcal{Y} = \{1, 2, \ldots, T\}$
- 0-1 loss function, i.e $\ell(y, y') = \mathbf{1}_{\{y \neq y'\}}$

Issues:

- Can we still map $\mathcal{Y}$ in $\mathbb{R}$?
- What surrogate $\mathcal{L}$ can replace ℓ?

Example: Multiclass Classification

- **Attempt 1:** $\mathcal{Y} = \{1, 2, \ldots, T\} \subset \mathbb{R}$. Could replace $\mathcal{Y}$ with $\mathbb{R}$

Example: Multiclass Classification

- **Attempt 1:** $\mathcal{Y} = \{1, 2, \ldots, T\} \subset \mathbb{R}$. Could replace $\mathcal{Y}$ with $\mathbb{R}$
  Not a good choice: induces an arbitrary distance on classes.
  (i.e. 1 is closer to 2 than 3 and so on $\ldots$)

Example: Multiclass Classification

- **Attempt 1:** $\mathcal{Y} = \{1, 2, \ldots, T\} \subset \mathbb{R}$. Could replace $\mathcal{Y}$ with $\mathbb{R}$
  Not a good choice: induces an arbitrary distance on classes.
  (i.e. 1 is closer to 2 than 3 and so on . . .)

- **Attempt 2:** replace $\mathcal{Y} = \{1, 2, \ldots, T\}$ with $\mathbb{R}^T$.
  "Replace" means "embed" $\mathcal{Y}$ into $\mathbb{R}^T$ using an **encoding** $\mathsf{c} : \mathcal{Y} \to \mathbb{R}^T$ defined by

$$\mathsf{c}(i) = e_i \qquad\qquad i = 1, \ldots Y$$

  where e_i is the i^{th} vector of the canonical basis of $\mathbb{R}^T$.

Example: Multiclass Classification

Given a *surrogate* loss $\mathcal{L} : \mathbb{R}^T \times \mathbb{R}^T \to \mathbb{R}$ (hinge? least squares?)...

...we can train the surrogate estimator $\widehat{g} : \mathcal{X} \to \mathbb{R}^T$

$$\widehat{g} = \operatorname*{argmin}_{g \in \mathcal{G}} \; \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(g(x_i), \mathsf{c}(y_i)).$$

Example: Multiclass Classification

Given a *surrogate* loss $\mathcal{L} : \mathbb{R}^T \times \mathbb{R}^T \to \mathbb{R}$ (hinge? least squares?)...

...we can train the surrogate estimator $\widehat{g} : \mathcal{X} \to \mathbb{R}^T$

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

Question: But $\widehat{g}$ has values in $\mathbb{R}^T$... How can we go back to $\mathcal{Y}$?

Example: Multiclass Classification

Given a *surrogate* loss $\mathcal{L} : \mathbb{R}^T \times \mathbb{R}^T \to \mathbb{R}$ (hinge? least squares?)...

. . . we can train the surrogate estimator $\widehat{g} : \mathcal{X} \to \mathbb{R}^T$

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

Question: But $\widehat{g}$ has values in $\mathbb{R}^T$. . . How can we go back to $\mathcal{Y}$?

Answer: via a decoding routine!

$$\widehat{f}(x) = \underset{t=1,\ldots T}{\operatorname{argmax}} \, \widehat{g}_t(x)$$

Multiclass Classification: Surrogate Vs. Original Problems

The same questions as for binary classification ...

- How can we go from $\widehat{g} : \mathcal{X} \to \mathbb{R}^T$ to some $\widehat{f} : \mathcal{X} \to \mathcal{Y}$?
- How is $g^\star$ related to $f^\star$?
- Are surrogate learning rates for $\widehat{g}$ of any use?

Multiclass Classification: Surrogate Vs. Original Problems

The same questions as for binary classification ...

- How can we go from $\widehat{g} : \mathcal{X} \to \mathbb{R}^T$ to some $\widehat{f} : \mathcal{X} \to \mathcal{Y}$?
  Decoding: $\widehat{f}(x) = \text{argmax}_{t=1,\dots,T}\ \widehat{g}_t(x)$
- How is $g^\star$ related to $f^\star$?
- Are surrogate learning rates for $\widehat{g}$ of any use?

Multiclass Classification: Surrogate Vs. Original Problems

The same questions as for binary classification . . .

- How can we go from $\widehat{g} : \mathcal{X} \to \mathbb{R}^T$ to some $\widehat{f} : \mathcal{X} \to \mathcal{Y}$?

  Decoding: $\widehat{f}(x) = \mathrm{argmax}_{t=1,\dots,T}\ \widehat{g}_t(x)$

- How is $g^\star$ related to $f^\star$?

  Not clear: it strongly depends on $\mathcal{L}$!

- Are surrogate learning rates for $\widehat{g}$ of any use?

  Not clear: it strongly depends on $\mathcal{L}$!

The Surrogate Approach

Taking inspiration from the previous examples . . .

Surrogate Approach: Key Ingredients

A possible approach to structured prediction is to find:

1. A linear surrogate space $\mathcal{H}$,
2. An **encoding** $\mathsf{c} : \mathcal{Y} \to \mathcal{H}$,
3. A **surrogate loss** $\mathcal{L} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$,
4. A **decoding** $\mathsf{d} : \mathcal{Y} \to \mathcal{H}$.

Surrogate Approach: Recipe

Then:

1. **Encode** training set $(x_i, y_i)_{i=1}^n$ into $(x_i, \mathsf{c}(y_i))_{i=1}^n$,
2. **Learn** $\widehat{g} = \operatorname{argmin}_{g \in \mathcal{G}} \ \frac{1}{n} \sum_{i=1}^n \mathcal{L}(g(x_i), c(y_i))$
   (using standard supervised learning methods)
3. **Decode** $\widehat{f} = \mathsf{d} \circ \widehat{g}$.

Wish list

However, recall that learning $\widehat{g}$ is solving a **different** problem...

$$\mathcal{R}(g) = \int \mathcal{L}(g(x), \mathsf{c}(y)) \; d\rho(x, y).$$

Wish list

However, recall that learning $\widehat{g}$ is solving a **different** problem...

$$\mathcal{R}(g) = \int \mathcal{L}(g(x), \mathsf{c}(y))\ d\rho(x, y).$$

In order to be “useful”, a surrogate framework needs to satisfy:

- **Fischer Consistency.** $\quad \mathcal{E}(f^\star) = \mathcal{E}(\mathsf{d} \circ g^\star)$

Wish list

However, recall that learning $\widehat{g}$ is solving a **different** problem...

$$\mathcal{R}(g) = \int \mathcal{L}(g(x), \mathsf{c}(y))\; d\rho(x, y).$$

In order to be "useful", a surrogate framework needs to satisfy:

- **Fischer Consistency.** $\quad \mathcal{E}(f^\star) = \mathcal{E}(\mathsf{d} \circ g^\star)$
- **Comparison Inequality.** for any $g : \mathcal{X} \to \mathcal{H}$,

$$\mathcal{E}(\mathsf{d} \circ g) - \mathcal{E}(f^\star) \leq \varphi\left(\mathcal{R}(g) - \mathcal{R}(g^\star)\right),$$

with $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ continuous, non-decreasing and $\varphi(0) = 0$.

Fisher consistency

Fisher consistency. We want this because we want that the surrogate problem and the decoding procedure are good ones, meaning that if we decode the best surrogate solution $\mathrm{d} \circ g^*$ we have the same risk as the best original solution f^*.

Comparison inequality

Comparison inequality. If we learn a $\widehat{g}$ which approximates g^*...

$$\mathcal{R}(\widehat{g}) - \mathcal{R}(g^*) \to 0 \quad \text{as } n \to +\infty.$$

... then the comparison inequality implies,

$$\mathcal{E}(\mathsf{d} \circ \widehat{g}) - \mathcal{E}(f^*) \to 0 \quad \text{as } n \to +\infty.$$

Therefore $\widehat{f} := \mathsf{d} \circ \widehat{g}$ is a good estimator for the original problem!

Comparison inequality

Comparison inequality. If we learn a $\widehat{g}$ which approximates g^*...

$$\mathcal{R}(\widehat{g}) - \mathcal{R}(g^*) \to 0 \quad \text{as } n \to +\infty.$$

...then the comparison inequality implies,

$$\mathcal{E}(\mathsf{d} \circ \widehat{g}) - \mathcal{E}(f^*) \to 0 \quad \text{as } n \to +\infty.$$

Therefore $\widehat{f} := \mathsf{d} \circ \widehat{g}$ is a good estimator for the original problem!

Rates. Moreover, if $\mathcal{R}(\widehat{g}) - \mathcal{R}(g^\star) \leq n^{-\alpha}$ for some $\alpha > 0$

$$\mathcal{E}(\widehat{f}) - \mathcal{E}(f^\star) \leq \varphi(n^{-\alpha}).$$

Knowledge of φ allows to derive rates for $\widehat{f}$ from the rates of $\widehat{g}$!

Going back to the examples...

Surrogate framework for **binary classification**:

- $\mathcal{Y} = \{1, -1\}$, $\mathcal{H} = \mathbb{R}$
- coding $\mathsf{c} : \{1, -1\} \to \mathbb{R}$ is the embedding $\mathcal{Y} \hookrightarrow \mathbb{R}$
- $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$: least squares ✓, hinge ✓, logistic ✓
- decoding $\mathsf{d} : \mathbb{R} \to \{1, -1\}$ is $\mathsf{d}(r) = \operatorname{sign}(r)$.

Fisher consistency? Comparison inequality?
Exercise for the reader! :)

Going back to the examples...

Surrogate framework for **multiclass classification**:

- $\mathcal{Y} = \{1, 2, \ldots, T\}$, $\mathcal{H} = \mathbb{R}^T$
- coding $\mathsf{c} : \{1, 2, \ldots, T\} \hookrightarrow \mathbb{R}^T$ with $\mathsf{c}(i) = e_i$.
- $\mathcal{L} : \mathbb{R}^T \times \mathbb{R}^T \to \mathbb{R}_+$: least squares ✓, hinge ×.
- decoding $\mathsf{d} : \mathbb{R}^T \to \{1, 2, \ldots, T\}$ is $\mathsf{d}(r) = \operatorname{argmax}_{t=1,\ldots,T} r_t$.

Fisher consistency? Comparison inequality?
Exercise for the reader! :)

To sum up

Pros

- **Modeling.** Directly borrow from ERM literature to design (surrogate) learning algorithms (vector-valued regression!)
- **Statistics.** Extend *surrogate* ERM rates for $\widehat{g}$ to $\widehat{f}$ by means of the comparison inequality.
- **Optimization.** Bypasses/Postpones dealing with the non-convex ℓ at prediction time!

Cons

- **Flexibility.** Need to design a surrogate framework $(\mathcal{H}, \mathsf{c}, \mathcal{L}, \mathsf{d})$ on a **case-by-case basis** for any $(\ell, \mathcal{Y})$.

Likelihood Estimation Approaches

A standard approach

Alternative approach to address structured prediction problems:

- **Model** the likelihood of observing y given x as a function $F^\star : \mathcal{Y} \times \mathcal{X} \to [0, 1]$ with $F^\star(y, x) = \rho(y|x)$.

- **Learn** $\hat{F} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

- Ideally $\hat{F} \to F^*$, with $F^*(x, y) = \rho(y \mid x)$.

- Then,
  - **Ideal** solution $f^*(x) = \mathrm{argmax}_{y \in \mathcal{Y}} \ F^*(x, y)$
  - **Approximate** solution $\hat{f}(x) = \mathrm{argmax}_{y \in \mathcal{Y}} \ \hat{F}(x, y)$

Struct SVM (Tsochantaridis et al., 2005)

Model:

- *joint* feature map $\Psi : \mathcal{Y} \times \mathcal{X} \to \mathcal{F}$ with $\mathcal{F}$ a Hilbert space.
- $F(y, x) = \langle w, \Psi(y, x) \rangle$ with $w \in \mathcal{F}$ a parameter vector.

Algorithm: Find the parameters $\hat{w}$ that solve

$$\min_{w \in \mathcal{F}} \quad \|w\|^2$$

$$\langle w, \Psi(y_i, x_i) \rangle \geq \langle w, \Psi(y, x_i) \rangle + 1$$

$$\forall i = 1, \ldots, n, \forall y \in \mathcal{Y} \setminus y_i$$

Intuition: the best $y^*(x)$ must be such that $F(x, y^*(x))$ is considerably larger than any other $F(x, y)$

Including the Loss

However, things are more complicated...

we don't want to simply maximise $\rho(y \mid x)$, but we have a loss function ℓ as part of the problem:

$$\mathcal{E}(f) = \int \ell(f(x), y) \; d\rho(y \mid x) d\rho_{\mathcal{X}}(x)$$

Struct SVM Variants

Model:

- *joint* feature map $\Psi : \mathcal{Y} \times \mathcal{X} \to \mathcal{F}$ with $\mathcal{F}$ a Hilbert space.
- $F(y, x) = \langle w, \Psi(y, x) \rangle$ with $w \in \mathcal{F}$ a parameter vector.

Algorithm: Find the parameters $\hat{w}$ that solve

$$\begin{aligned} \min_{w \in \mathcal{F}} \quad & \|w\|^2 \\ & \langle w, \Psi(y_i, x_i) \rangle \geq \langle w, \Psi(y, x_i) \rangle + \ell(y_i, y) \\ & \forall i = 1, \ldots, n, \forall y \in \mathcal{Y} \setminus y_i \end{aligned}$$

Struct SVM Variants

Model:

- *joint* feature map $\Psi : \mathcal{Y} \times \mathcal{X} \to \mathcal{F}$ with $\mathcal{F}$ a Hilbert space.
- $F(y, x) = \langle w, \Psi(y, x) \rangle$ with $w \in \mathcal{F}$ a parameter vector.

Algorithm: Find the parameters $\hat{w}$ that solve

$$\min_{w \in \mathcal{F}} \quad \|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$

$$\langle w, \Psi(y_i, x_i) \rangle \geq \langle w, \Psi(y, x_i) \rangle + \ell(y_i, y) - \xi_i$$

$$\forall i = 1, \ldots, n, \forall y \in \mathcal{Y} \setminus y_i$$

Generalizing the "slack" variables in standard SVM ...

Optimization

Algorithm 1 Algorithm for solving SVM_0 and the loss re-scaling formulations $\text{SVM}_1^{\triangle s}$ and $\text{SVM}_2^{\triangle s}$

Input: $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$, C, ϵ
$S_i \leftarrow \emptyset$ for all $i = 1, \ldots, n$
repeat
  for $i = 1, \ldots, n$ **do**
    set up cost function
    $\text{SVM}_1^{\triangle s}: H(\mathbf{y}) \equiv (1 - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle) \triangle(\mathbf{y}_i, \mathbf{y})$
    $\text{SVM}_2^{\triangle s}: H(\mathbf{y}) \equiv (1 - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle) \sqrt{\triangle(\mathbf{y}_i, \mathbf{y})}$
    $\text{SVM}_1^{\triangle m}: H(\mathbf{y}) \equiv \triangle(\mathbf{y}_i, \mathbf{y}) - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle$
    $\text{SVM}_2^{\triangle m}: H(\mathbf{y}) \equiv \sqrt{\triangle(\mathbf{y}_i, \mathbf{y})} - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle$
    where $\mathbf{w} \equiv \sum_j \sum_{\mathbf{y}' \in S_j} \alpha_{j\mathbf{y}'} \delta\Psi_j(\mathbf{y}')$.
    compute $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in Y} H(\mathbf{y})$
    compute $\xi_i = \max\{0, \max_{\mathbf{y} \in S_i} H(\mathbf{y})\}$
    if $H(\hat{\mathbf{y}}) > \xi_i + \epsilon$ **then**
      $S_i \leftarrow S_i \cup \{\hat{\mathbf{y}}\}$
      $\alpha_S \leftarrow$ optimize dual over S, $S = \cup_i S_i$.
    end if
  end for
until no S_i has changed during iteration

To sum up

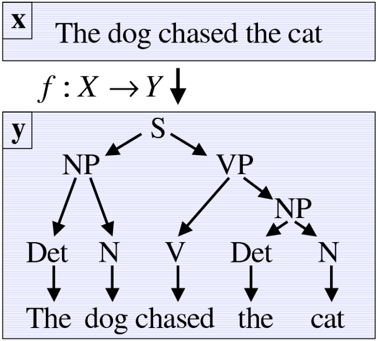
Pros

- **Flexibility.** Can be virtually applied to **any** problem.

Cons

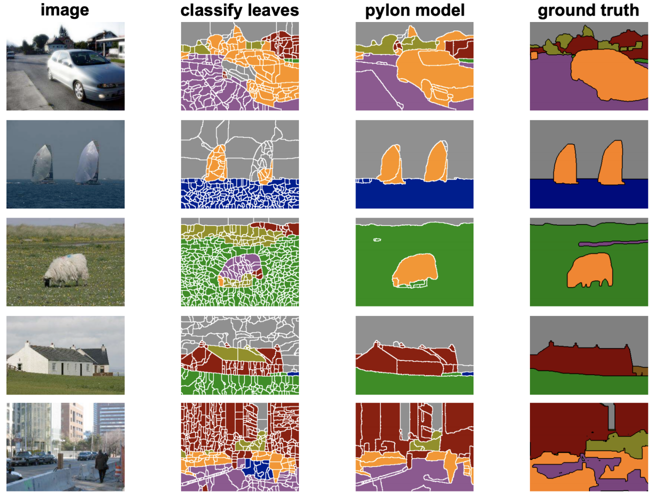
- **Optimization.** Requires solving an optimization over $\mathcal{Y}$ and with respect to ℓ at **every** iteration. It can become very expensive!
- **Statistics.** It has been shown that in some cases this approach is **not consistent** (Bakir et al., 2007).

Examples: Language Parsing



$$\Psi(\mathbf{x},\mathbf{y}) = \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \\ \vdots \\ 0 \\ 2 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{matrix} S \rightarrow NP\ VP \\ S \rightarrow NP \\ NP \rightarrow Det\ N \\ VP \rightarrow V\ NP \\ \\ Det \rightarrow dog \\ Det \rightarrow the \\ N \rightarrow dog \\ V \rightarrow chased \\ N \rightarrow cat \end{matrix}$$

Examples: Image Segmentation



E.g. [Taskar et al., 2003] (image [Lempitsky et al., 2011])

Examples: Pose Estimation

E.g. [Ramanan et al., 2005, Ramanan, 2006, Ferrari et al., 2008]

To wrap up...

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)

Score learning techniques

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

To wrap up...

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)

Score learning techniques

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

Can we get the best of both worlds?

Structured Prediction with Implicit 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)

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 best $f^\star : \mathcal{X} \to \mathcal{Y}$ is:

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

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

Finite Dimensional Intuition

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

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

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

where e_y is the y-th element of the canonical basis.

This (bi)linearity will be very useful...

Finite Dimensional Intuition (cont.)

Going back to $f^\star$...

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

Denote by $g^\star : \mathcal{X} \to \mathbb{R}^T$ the function $g^\star(x) = \int e_y \; d\rho(y|x)$. Then:

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

Finite Dimensional Intuition (cont.)

Idea: replace $g^\star : \mathcal{X} \to \mathbb{R}^T$ in

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

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

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

Finite Dimensional Intuition (cont.)

What is a good algorithm to learn $\widehat{g}$?

Recall that $g^\star(x) = \int e_y \ d\rho(y|x) = \mathbb{E}_{y|x}[e_y]$ is a conditional expectation...

It is easy to show that

$$g^\star = \underset{g:\mathcal{X}\to\mathbb{R}^T}{\operatorname{argmin}} \ \mathcal{R}(g) \qquad\qquad \mathcal{R}(g) = \int \|g(x) - e_y\|^2 \ d\rho(x,y)$$

Therefore $\widehat{g}$ can be taken to be the least-squares ERM estimator!

Going back to surrogate methods

Natural way to find a surrogate framework:

- **Encoding.** $\mathsf{c} : \mathcal{Y} \to \mathcal{H} = \mathbb{R}^T$ such that $y \mapsto e_y$,
- **Loss.** $\mathcal{L}(g(x), \mathsf{c}(y)) = \|g(x) - \mathsf{c}(y)\|^2$,
- **Decoding.** $\mathsf{d} : \mathbb{R}^T \to \mathcal{Y}$ such that for any $h \in \mathbb{R}^T$

$$\mathsf{d}(h) = \operatorname*{argmin}_{z \in \mathcal{Y}} \; e_z^\top V h$$

Very similar to the multiclass setting (but can be applied to any ℓ)!

Finite Dimensional Intuition (cont.)

We perform **vector-valued ridge-regression**.

Let $\mathcal{X} = \mathbb{R}^d$. We parametrize $\widehat{g}(x) = \widehat{W}x$, where

$$\widehat{W} \; = \; \underset{W \in \mathbb{R}^{T \times d}}{\operatorname{argmin}} \; \frac{1}{n} \sum_{i=1}^{n} \|e_{y_i} - W x_i\|^2 + \lambda \|W\|_F^2 \; ,$$

The solution is

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

$I \in \mathbb{R}^{d \times d}$ identity matrix, $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{n \times T}$ the matrices with i-th row corresponding to x_i and e_{y_i} respectively.

Finite Dimensional Intuition (cont.)

By some algebraic manipulation...

$$\widehat{g}(x) \;=\; \widehat{W}x \;=\; Y^\top \underbrace{X\,(X^\top X + n\lambda I)^{-1} x}_{\alpha(x)} \;=\; \sum_{i=1}^{n} \alpha_i(x)\, e_{y_i}\;, \qquad (1)$$

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

$$\alpha(x) \;=\; (\alpha_1(x), \dots, \alpha_n(x))^\top \;=\; [X(X^\top X + n\lambda I)^{-1}]\, x \;\in \mathbb{R}^n.$$

Finite Dimensional Intuition (cont.)

Therefore, by replacing the definition of $\widehat{f}$...

$$\begin{aligned}\widehat{f}(x) &= \operatorname*{argmin}_{z \in \mathcal{Y}} \; e_z^\top V \widehat{g}(x) \\ &= \operatorname*{argmin}_{z \in \mathcal{Y}} \; \sum_{i=1}^{n} \alpha_i(x) \; \underbrace{e_z^\top V e_{y_i}}_{\ell(z, y_i)}\end{aligned}$$

In other words,

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

To sum up...

This approach alternates between two phases:

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

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

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

Note. similarly to likelihood estimation methods one needs to know how to optimize over $\mathcal{Y}$ (but only needs to do it once!).

Wish List

Going back to our wishlist:

- Is **flexible**: can be applied to (m)any $\mathcal{Y}$ and ℓ. ✓
- Leads to efficient **computations**.
  - No optimization over $\mathcal{Y}$ during training,
  - Recovers many previous surrogate approaches.
- Has strong **theoretical** guarantees (i.e. consistency, rates)
  In a minute...

General Case: Implicit Embeddings

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

Definition. *A continuous* $\ell : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$ *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* $\ell : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$ *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*

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

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

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

Which loss functions have an IE?

- **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 $\chi 2$ 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...

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*
$$\int_{\mathcal{Y}\times\mathcal{Y}} |L(y,y')|^{1+\epsilon} dy\,dy' < \infty, \quad \textit{with} \quad \epsilon > 0. \qquad (149)$$
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 $i \in \{1, \dots, p\}$.

Structured Prediction with Implicit Embeddings

If ℓ has an implicit embedding:

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

with $g^\star : \mathcal{X} \to \mathcal{H}$ such that

$$g^\star(x) = \int \mathsf{c}(y) \ d\rho(y|x),$$

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

Structured Prediction with Implicit Embeddings (Cont.)

We approximate $g^\star$ with $\widehat{g}(x) = \widehat{W}x$

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

- 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.)

We approximate $g^\star$ with $\widehat{g}(x) = \widehat{W}x$

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

- 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

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

$X \in \mathbb{R}^{n \times d}$ 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.)

$\widehat{W}$ contains infinitely many parameters. However...

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

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

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

Structured Prediction with Implicit Embeddings (Cont.)

$\widehat{W}$ contains infinitely many parameters. However...

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

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

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

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

Structured Prediction with Implicit Embeddings (Cont.)

Therefore, analogously to the finite case...

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

$$= \operatorname*{argmin}_{z \in \mathcal{Y}} \ \sum_{i=1}^{n} \alpha_i(x) \ \underbrace{\langle \mathsf{c}(z), V \mathsf{c}(y_i) \rangle}_{\substack{\ell(z, y_i) \\ \textbf{loss trick}}}$$

In other words,

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

The "loss trick"

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

Analogous to the "kernel trick", the implicit embedding enables us to find an estimator $\widehat{f} : \mathcal{X} \to \mathcal{Y}$...

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

Implicit Embeddings and Surrogate Methods

Implicit embeddings naturally induce a surrogate framework:

- **Encoding.** $\mathsf{c} : \mathcal{Y} \to \mathcal{H}$,

- **Loss.** $\mathcal{L}(g(x), \mathsf{c}(y)) = \|g(x) - \mathsf{c}(y)\|_{\mathcal{H}}^2$,

- **Decoding.** $\mathsf{d} : \mathcal{H} \to \mathcal{Y}$ such that for any $h \in \mathcal{H}$

$$\mathsf{d}(h) = \underset{z \in \mathcal{Y}}{\operatorname{argmin}} \ \langle \mathsf{c}(z), V h \rangle_{\mathcal{H}}$$

Q: *do Fischer consistency and a comparison inequality hold?*

Fischer Consistency & Comparison Inequality

Fischer Coinsistency. We get it for free...

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

Fischer Consistency & Comparison Inequality

Fischer Coinsistency. We get it for free...

$$f^\star(x) = \mathsf{d}(g^\star(x)) = \underset{z \in \mathcal{Y}}{\operatorname{argmin}} \;\; \langle \mathsf{c}(z), V\; g^\star(x) \rangle_{\mathcal{H}}$$

Comparison Inequality. We have the following...

Theorem (Ciliberto et al., 2016) *Let ℓ admit an implicit embedding $(\mathcal{H}, \mathsf{c}, V)$. Then, for any measurable $g : \mathcal{X} \to \mathcal{H}$*

$$\mathcal{E}(\mathsf{d} \circ g) - \mathcal{E}(f^\star) \leq \mathsf{q}_\ell \; \sqrt{\mathcal{R}(g) - \mathcal{R}(g^\star)}$$

with $\mathsf{q}_\ell = 2 \; \sup_{y \in \mathcal{Y}} \;\; \|V \mathsf{c}(y)\|_{\mathcal{H}}$.

Universal consistency

We can borrow from the literature on vector-valued regression (Caponnetto and De Vito, 2007) to study $\widehat{g}$.

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[1]. Choose $\lambda = n^{-1/2}$ to train $\widehat{f}$. Then,

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

with probability 1.

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

Learning Rates

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

$$\mathcal{E}(\widehat{f}) - \mathcal{E}(f^\star) \leq \mathsf{q}_\ell \, \log(1/\delta) \, \frac{1}{n^{1/4}},$$

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

Learning Rates

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

$$\mathcal{E}(\widehat{f}) - \mathcal{E}(f^\star) \leq \mathsf{q}_\ell \ \log(1/\delta) \ \frac{1}{n^{1/4}},$$

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

Comments.

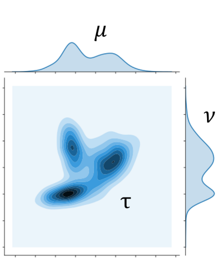
- Same rates as worst-case binary classification (better rates with Tsibakov-like noise assumptions (Nowak-Vila et al., 2018)).
- Adaptive w.r.t. q_ℓ (it automatically chooses the "best" surrogate framework).

Example Applications

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

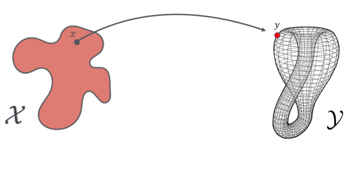
	Reconstruction Error (%)			
# Classes	**Ours**	$\widetilde{S}_\lambda$	Hell	KDE
2	**3.7 ± 0.6**	4.9 ± 0.9	8.0 ± 2.4	12.0 ± 4.1
4	**22.2 ± 0.9**	31.8 ± 1.1	29.2 ± 0.8	40.8 ± 4.2
10	**38.9 ± 0.9**	44.9 ± 2.5	48.3 ± 2.4	64.9 ± 1.4

Manifold Regression
[Rudi, Ciliberto, Marconi, Rosasco '18]

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

Loss: (squared) geodesic distance.

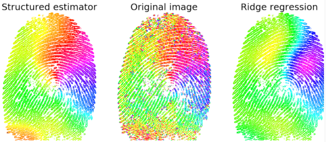
Optimization: Riemannian GD.



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)

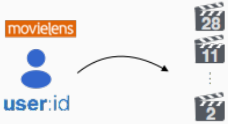
	KRLS	SP (Ours)
Emotions	0.63	**0.73**
CAL500	**0.92**	**0.92**
Scene	0.62	**0.73**

Nonlinear Multi-task Learning
[Ciliberto, Rudi, Rosasco, Pontil ’17, Luise, Stamos, Pontil, Ciliberto ’19]

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)**

Wrapping up...

Structured prediction poses hard optimization/modeling/statistical challenges. We have seen two main strategies:

- **Likelihood Estimation.** Flexible yet lacking theory.
- **Surrogate Methods.** Theoretically sound but not flexible.

By leveraging the concept of *Implicit Embeddings* we found a synthesis between these two strategies:

- **Flexible.** Can be applied to any ℓ admitting an implicit embedding.
- **Optimization.** Requires a minimization over $\mathcal{Y}$ *only at test time.*
- **Sound.** We have consistency and learning rates.

Additional Work

Case studies:

- Learning to rank (Korba et al., 2018)
- Output Fisher Embeddings (Djerrab et al., 2018)
- $\mathcal{Y}$ = manifolds, ℓ = geodesic distance (Rudi et al., 2018)
- $\mathcal{Y}$ = probability space, ℓ = 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)

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

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

Caponnetto, A. and De Vito, E. (2007). Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368.

Ciliberto, C., Rosasco, L., and Rudi, A. (2016). A consistent regularization approach for structured prediction. *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 4412–4420.

Ciliberto, C., Rudi, A., Rosasco, L., and Pontil, M. (2017). Consistent multitask learning with nonlinear output relations. In *Advances in Neural Information Processing Systems*, pages 1983–1993.

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

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

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

Luise, G., Rudi, A., Pontil, M., and Ciliberto, C. (2018). Differential properties of sinkhorn approximation for learning with wasserstein distance. In *Advances in Neural Information Processing Systems*, pages 5859–5870.

Luise, G., Stamos, D., Pontil, M., and Ciliberto, C. (2019). Leveraging low-rank relations between surrogate tasks in structured prediction. *International Conference on Machine Learning (ICML)*.

Mroueh, Y., Poggio, T., Rosasco, L., and Slotine, J.-J. (2012). Multiclass learning with simplex coding. In *Advances in Neural Information Processing Systems (NIPS) 25*, pages 2798–2806.

Nowak-Vila, A., Bach, F., and Rudi, A. (2018). Sharp analysis of learning with discrete losses. *AISTATS*.

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

Osokin, A., Bach, F., and Lacoste-Julien, S. (2017). On structured prediction theory with calibrated convex surrogate losses. In *Advances in Neural Information Processing Systems*, pages 302–313.

Rudi, A., Ciliberto, C., Marconi, G., and Rosasco, L. (2018). Manifold structured prediction. In *Advances in Neural Information Processing Systems*, pages 5610–5621.

Song, L., Huang, J., Smola, A., and Fukumizu, K. (2009). 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.

Struminsky, K., Lacoste-Julien, S., and Osokin, A. (2018). Quantifying learning guarantees for convex but inconsistent surrogates. In *Advances in Neural Information Processing Systems*, pages 669–677.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. volume 6, pages 1453–1484.