A Consistent Regularization Approach for Structured Prediction

Carlo Ciliberto, Alessandro Rudi, Lorenzo Rosasco

University of Genova
Istituto Italiano di Tecnologia - Massachusetts Institute of Technology
lcsl.mit.edu

Dec 9th, NIPS 2016

Structured Prediction

x $\xrightarrow{f}$ y

Image Captioning
(also Localization
Segmentation
Classification)

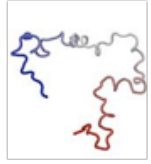
Movie Ranking

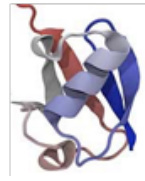
NETFLIX
user:127

Speech Recognition

"Ok Google"

Protein Folding





Outline

Standard Supervised Learning

Structured Prediction with SELF
- Algorithm
- Theory
- Experiments

Conclusions

Outline

Scalar Learning

Goal: given $(x_i, y_i)_{i=1}^n$, find $f_n : \mathcal{X} \to \mathcal{Y}$

Let $\mathcal{Y} = \mathbb{R}$

- **Parametrize**

$$f(x) = w^\top \varphi(x) \qquad w \in \mathbb{R}^P \qquad \varphi : \mathcal{X} \to \mathbb{R}^P$$

- **Learn**

$$f_n = w_n^\top \varphi(x) \qquad w_n = \operatorname*{argmin}_{w \in \mathbb{R}^P} \frac{1}{n} \sum_{i=1}^n L(w^\top \varphi(x_i), y_i)$$

Multi-variate Learning

Goal: given $(x_i, y_i)_{i=1}^n$, find $f_n : \mathcal{X} \to \mathcal{Y}$

Let $\mathcal{Y} = \mathbb{R}^M$

- **Parametrize**

$$f(x) = W\varphi(x) \qquad W \in \mathbb{R}^{M \times P} \qquad \varphi : \mathcal{X} \to \mathbb{R}^P$$

- **Learn**

$$f_n(x) = W_n\, \varphi(x) \qquad W_n = \underset{W \in \mathbb{R}^{M \times P}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(W\varphi(x_i), y_i)$$

Learning Theory

Expected Risk

$$\mathcal{E}(f) = \int_{\mathcal{X}\times\mathcal{Y}} L(f(x), y)\ d\rho(x, y)$$

- **Consistency**

$$\lim_{n\to+\infty} \mathcal{E}(f_n) \ = \ \inf_f \ \mathcal{E}(f) \qquad (in \ \ probability)$$

- **Excess Risk Bounds**

$$\mathcal{E}(f_n) - \inf_{f\in\mathcal{H}} \ \mathcal{E}(f) \ \lesssim \ \epsilon(n, \rho, \mathcal{H}) \qquad (w.h.p.)$$

Outline

(Un)Structured prediction

What if $\mathcal{Y}$ is **not** a vector space?
(e.g. strings, graphs, histograms, etc.)

Q. How do we:

- **Parametrize**
- **Learn**

a function $f : \mathcal{X} \to \mathcal{Y}$?

Possible Approaches

- Score-Learning Methods
  - \+ General algorithmic framework (e.g. StructSVM [Tsochandaridis et al '05])
  - − Limited Theory ([McAllester '06])

- Surrogate/Relaxation approaches:
  - \+ Clear theory
  - − Only for special cases
    (e.g. classification, ranking, multi-labeling etc.)
    [Bartlett et al '06, Duchi et al '10, Mroueh et al '12, Gao et al. '13]

Relaxation Approaches

1. **Encoding**
   choose $c : \mathcal{Y} \to \mathbb{R}^M$

2. **Learning**
   Given $(x_i, c(y_i))_{i=1}^n$, find $g_n : \mathcal{X} \to \mathbb{R}^M$

3. **Decoding**
   choose $d : \mathbb{R}^M \to \mathcal{Y}$ and let $f_n(x) = (d \circ g_n)(x)$

Example I: Binary Classification

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

1. $c : \{-1, 1\} \to \mathbb{R}$ identity
2. Scalar learning $\quad g_n : \mathcal{X} \to \mathbb{R}$
3. $d = sign : \mathbb{R} \to \{-1, 1\}$

$$f_n(x) = sign(g_n(x))$$

Example II: Multi-class Classification

Let $\mathcal{Y} = \{1, \ldots, M\}$

1. $c : \mathcal{Y} \to \{e_1, \ldots, e_M\} \subset \mathbb{R}^M$ canonical basis, $\quad c(j) = e_j \in \mathbb{R}^M$

2. Multi-variate learning $\quad g_n : \mathcal{X} \to \mathbb{R}^M$

3. $d : \mathbb{R}^M \to \{1, \ldots, M\}$

$$f_n(x) = \underset{j=1,\ldots,M}{\operatorname{argmax}} \underbrace{e_j^\top g_n(x)}_{j-\text{th value of } g_n(x)}$$

A General Relaxation Approach

A General Relaxation Approach

Main Assumption. Structure Encoding Loss Function (SELF)

Given $\triangle : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, there exist:

- $\mathcal{H}_\mathcal{Y}$ RKHS with $c : \mathcal{Y} \to \mathcal{H}_\mathcal{Y}$ feature map
- $V : \mathcal{H}_\mathcal{Y} \to \mathcal{H}_\mathcal{Y}$ bounded linear operator

such that:

$$\triangle(y, y') = \langle c(y), V c(y') \rangle_{\mathcal{H}_\mathcal{Y}} \qquad \forall y, y' \in \mathcal{Y}$$

Note. If V is Positive Semidefinite $\implies \triangle$ is a kernel.

SELF: Examples

- Binary classification: $c : \{-1, 1\} \to \mathbb{R}$ and $V = 1$.
- Multi-class classification: $c(j) = e_j \in \mathbb{R}^M$ and $V = \mathbf{1} - I \in \mathbb{R}^{M \times M}$.
- Kernel Dependency Estimation (KDE) [Weston et al. '02, Cortes et al. '05]: $\triangle(y, y') = 1 - h(y, y'), \quad h : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ kernel on $\mathcal{Y}$.

SELF: Finite $\mathcal{Y}$

All $\triangle$ on discrete $\mathcal{Y}$ are SELF

Examples:

- **Strings**: edit distance, KL divergence, word error rate, ...
- **Ordered sequences**: rank loss, ...
- **Graphs/Trees**: graph/trees edit distance, subgraph matching ...
- **Discrete subsets**: weighted overlap loss, ...
- ...

SELF: More examples

- **Histograms/Probabilities**: e.g. χ^2, Hellinger, ...
- **Manifolds**: Diffusion distances
- ...

Relaxation with SELF

1. **Encoding**. $c : \mathcal{Y} \to \mathcal{H}_{\mathcal{Y}}$ canonical feature map of $\mathcal{H}_{\mathcal{Y}}$

2. **Surrogate Learning**. Multi-variate regression $g_n : \mathcal{X} \to \mathcal{H}_{\mathcal{Y}}$

3. **Decoding**. $f_n(x) = \underset{y \in \mathcal{Y}}{\operatorname{argmin}} \ \langle c(y), V g_n(x) \rangle_{\mathcal{H}_{\mathcal{Y}}}$

Surrogate Learning

Multi-variate learning with ridge regression

- **Parametrize**

$$g(x) = W\varphi(x) \quad W \in \mathbb{R}^{M\times P} \quad \varphi : \mathcal{X} \to \mathbb{R}^P$$

- **Learn**

$$g_n = W_n\,\varphi(x) \qquad W_n = \operatorname*{argmin}_{W\in\mathbb{R}^{M\times P}} \frac{1}{n}\sum_{i=1}^{n} \underbrace{\|W\varphi(x_i) - c(y_i)\|}_{\text{least-squares}}^2_{\mathcal{H}_\mathcal{Y}}$$

Learning (cont.)

Solution[1]

$g_n(x) = W_n \ \varphi(x)$

$$W_n = C \underbrace{(\Phi^\top \Phi)^{-1} \Phi^\top}_{A \in \mathbb{R}^{n \times n}} = CA$$

- $\Phi = [\varphi(x_1), \ldots, \varphi(x_n)] \in \mathbb{R}^{P \times n}$ input features
- $C = [c(y_1), \ldots, c(y_n)] \in \mathbb{R}^{M \times n}$ output features

[1] In practice add a regularizer!

Decoding

Lemma (Ciliberto, Rudi, Rosasco '16)

Let $g_n(x) = CA\ \varphi(x)$ solution the surrogate problem. Then

$$f_n(x) = \operatorname*{argmin}_{y \in \mathcal{Y}} \ \langle c(y), V g_n(x) \rangle_{\mathcal{H}_\mathcal{Y}}$$

can be written as

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

where

$$(\alpha_1(x), \dots, \alpha_n(x))^\top = A\ \varphi(x) \in \mathbb{R}^n$$

Decoding

Sketch of the proof:

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

  with $(\alpha_1(x), \ldots, \alpha_n(x))^\top = A\ \varphi(x) \in \mathbb{R}^n$

- Plugging $g_n(x)$ in

$$\begin{aligned} \langle c(y), V g_n(x) \rangle_{\mathcal{H}_\mathcal{Y}} &= \langle c(y), V \sum_{i=1} \alpha_i(x) c(y_i) \rangle_{\mathcal{H}_\mathcal{Y}} \\ &= \textstyle\sum_{i=1} \alpha_i(x)\ \boldsymbol{\langle c(y), V c(y_i) \rangle_{\mathcal{H}_\mathcal{Y}}} \\ &= \textstyle\sum_{i=1}^{n} \alpha_i(x)\ \boldsymbol{\triangle\,(y, y_i)} \end{aligned} \qquad \textbf{(SELF)}$$

SELF Learning

Two steps:

1. **Surrogate Learning**

$$(\alpha_1(x), \ldots, \alpha_n(x))^\top = A\ \varphi(x) \qquad A = (\Phi^\top \Phi + \lambda)^{-1} \Phi^\top$$

2. **Decoding**

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

Note:

- **Implicit encoding**: no need to know $\mathcal{H}_\mathcal{Y}$, V (extends kernel trick)!
- Optimization over $\mathcal{Y}$ is problem specific and can be a challenge.

Connections with Previous Work

- Score-Learning approaches (e.g. StructSVM [Tsochandaridis et al '05])
  In StructSVM is possible to choose any feature map on the output...
  ... here we show that this choice must be compatible with $\triangle$

- Kernel dependency estimation, $\triangle$ is (one minus) a kernel

- Conditional mean embeddings ?
  [Smola et al '07]

Relaxation Analysis

Relaxation Analysis

Consider

$$\mathcal{E}(f) = \int_{\mathcal{X}\times\mathcal{Y}} \triangle(f(x), y)\ d\rho(x, y)$$

and

$$\mathcal{R}(g) = \int_{\mathcal{X}\times\mathcal{Y}} \|g(x) - c(y)\|^2\ d\rho(x, y)$$

How are $\mathcal{R}(g_n)$ and $\mathcal{E}(f_n)$ related?

Relaxation Analysis

$$f_* = \operatorname*{argmin}_{f:\mathcal{X}\to\mathcal{Y}} \mathcal{E}(f) \qquad \text{and} \qquad g_* = \operatorname*{argmin}_{g:\mathcal{X}\to\mathcal{H}_\mathcal{Y}} \mathcal{R}(g)$$

Key properties:

- **Fisher Consistency (FC)**

$$\mathcal{E}(d \circ g_*) = \mathcal{E}(f_*)$$

- **Comparison Inequality (CI)**
  $\exists\ \theta : \mathbb{R} \to \mathbb{R}$ such that $\theta(r) \to 0$ when $r \to 0$ and

$$\mathcal{E}(d \circ g) - \mathcal{E}(f_*) \ \leq\ \theta(\mathcal{R}(g) - \mathcal{R}(g_*)) \qquad \forall g : \mathcal{X} \to \mathcal{H}_\mathcal{Y}$$

SELF Relaxation Analysis

Theorem (Ciliberto, Rudi, Rosasco '16)

$\triangle : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ SELF loss, $g_* : \mathcal{X} \to \mathcal{H}_\mathcal{Y}$ least-square "relaxed" solution.

Then

- **Fisher Consistency**

$$\mathcal{E}(d \circ g_*) = \mathcal{E}(f_*)$$

- **Comparison Inequality** $\forall g : \mathcal{X} \to \mathcal{H}_\mathcal{Y}$

$$\mathcal{E}(d \circ g) - \mathcal{E}(f_*) \lesssim \sqrt{\mathcal{R}(g) - \mathcal{R}(g_*)}$$

SELF Relaxation Analysis (cont.)

Lemma (Ciliberto, Rudi, Rosasco '16)

$\triangle : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ SELF loss. Then

$$\mathcal{E}(f) = \int_{\mathcal{X}} \langle c(f(x)), V g_*(x) \rangle_{\mathcal{H}_\mathcal{Y}} \; d\rho_\mathcal{X}(x)$$

where $g_* : \mathcal{X} \to \mathcal{H}_\mathcal{Y}$ minimizes

$$\mathcal{R}(g) = \int_{\mathcal{X} \times \mathcal{Y}} \|g(x) - c(y)\|^2_{\mathcal{H}_\mathcal{Y}} \; d\rho(x, y)$$

Least-squares on $\mathcal{H}_\mathcal{Y}$ is a good surrogate loss

Consistency and Generalization Bounds

Theorem (Ciliberto, Rudi, Rosasco '16)

If we consider a universal feature map and $\lambda = 1/\sqrt{n}$*, then,*

$$\lim_{n\to\infty} \mathcal{E}(f_n) = \mathcal{E}(f_*), \quad \textit{almost surely}$$

Moreover, under mild assumptions

$$\mathcal{E}(f_n) - \mathcal{E}(f_*) \lesssim n^{-1/4} \qquad (w.h.p.)$$

Proof.

Relaxation analysis + (kernel) ridge regression results

$$\mathcal{R}(g_n) - \mathcal{R}(g_*) \lesssim n^{-1/2}$$

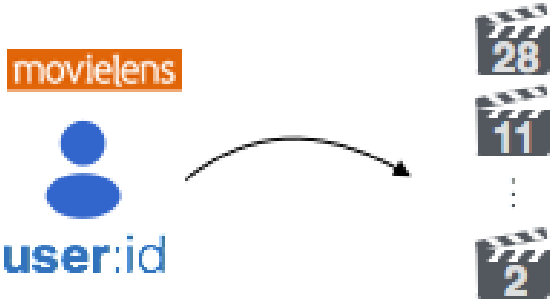
□

Remarks

- First result proving universal consistency and excess risk bounds for general structured prediction (partial results for KDE in [Gigure et al '13])
- Rates are sharp for the class of SELF loss functions $\triangle$: i.e. matching classification results.
- Faster rates under further regularity conditions.

Experiments: Ranking

$$\triangle_{rank}(f(x), y) = \sum_{i,j=1}^{M} \gamma(y)_{ij} \ (1 - \text{sign}(f(x)_i - f(x)_j))/2$$



	Rank Loss
[Herbrich et al. '99]	0.432 ± 0.008
[Dekel et al. '04]	0.432 ± 0.012
[Duchi et al. '10]	0.430 ± 0.004
[Tsochantaridis et al. '05]	0.451 ± 0.008
[**Ciliberto, Rudi, R. '16**]	$\mathbf{0.396 \pm 0.003}$

Ranking experiments on the MovieLens dataset with $\triangle_{rank}$ [Dekel et al. '04, Duchi et al. '10]. ~ 1600 Movies for ~ 900 users.

Experiments: Digit Reconstruction

Digit reconstruction on USPS dataset

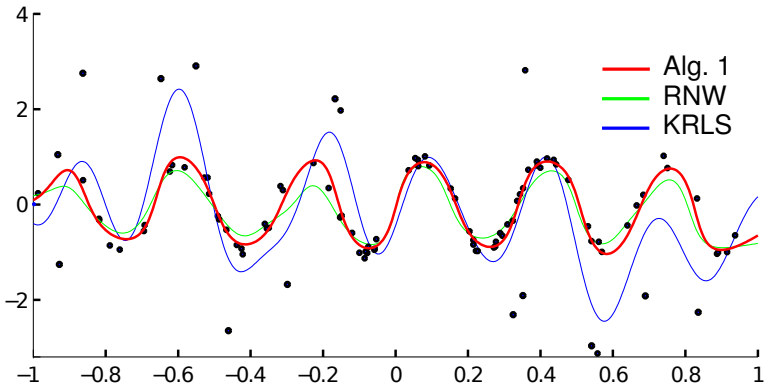
Loss	KDE $\triangle_G$	SELF $\triangle_H$
$\triangle_G$	$\mathbf{0.149 \pm 0.013}$	0.172 ± 0.011
$\triangle_H$	0.736 ± 0.032	$\mathbf{0.647 \pm 0.017}$
$\triangle_R$	0.294 ± 0.012	$\mathbf{0.193 \pm 0.015}$

- $\triangle_G(f(x), y) = 1 - k(f(x), y)$ k **Gaussian** kernel on the output.
- $\triangle_H(f(x), y) = \|\sqrt{f(x)} - \sqrt{y}\|$ **Hellinger** distance.

- $\triangle_R(f(x), y)$ **Recognition** accuracy of an SVM digit classifier.

Experiments: Robust Estimation

$$\triangle_{Cauchy}(f(x), y) = \frac{c}{2} \log(1 + \frac{\|f(x) - y\|^2}{c}) \qquad c > 0$$



n	SELF	RNW	KRR
50	**0.39** $\pm$ 0.17	0.45 $\pm$ 0.18	0.62 $\pm$ 0.13
100	**0.21** $\pm$ 0.04	0.29 $\pm$ 0.04	0.47 $\pm$ 0.09
200	**0.12** $\pm$ 0.02	0.24 $\pm$ 0.03	0.33 $\pm$ 0.04
500	**0.08** $\pm$ 0.01	0.22 $\pm$ 0.02	0.31 $\pm$ 0.03
1000	**0.07** $\pm$ 0.01	0.21 $\pm$ 0.02	0.19 $\pm$ 0.02

Outline

Wrapping Up

Contributions

1. A relaxation/regularization framework for structured prediction.
2. Theoretical guarantees: universal consistency+sharp bounds
3. Promising empirical results

Open Questions

- Surrogate loss functions beyond least-squares.
- Efficent decoding, exploit loss structure.
- Tsybakov noise "like" conditions

P.S.
I have post-doc positions! Ping me if you are interested.