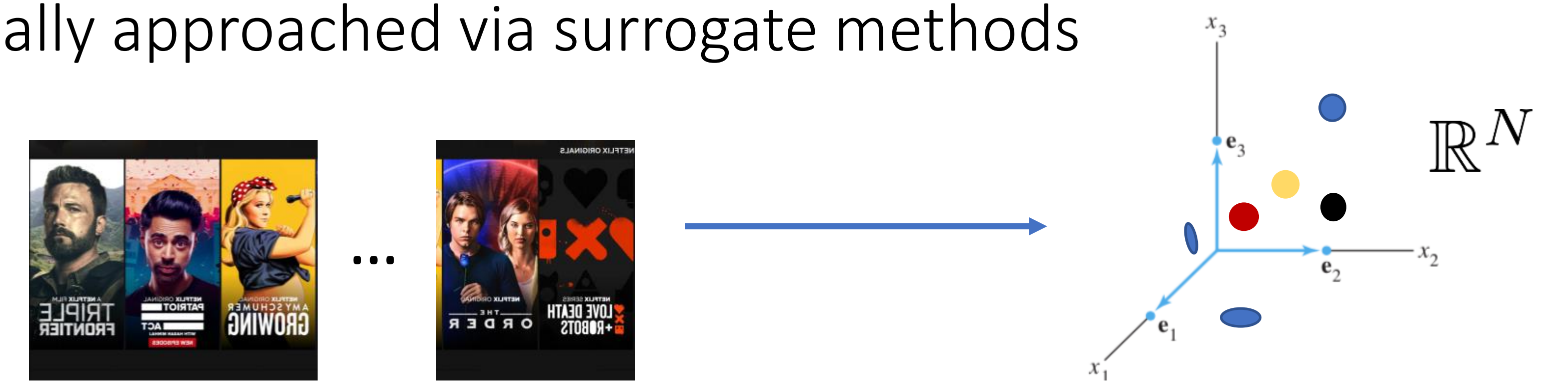
Leveraging Low-Rank Relations between Surrogate Tasks in Structured Prediction

Giulia Luise, Dimitris Stamos, Massimiliano Pontil, Carlo Ciliberto

ISTITUTO ITALIANO
DI TECNOLOGIA

Introduction

- structured problems: multilabelling, muticlass classification, ranking etc.
- usually approached via surrogate methods



- Multitask techniques leveraging similarity of outputs are used to lower sample complexity of the problem

- GOAL: combining the two and study when and how there is an advantage

Surrogate Methods for Structured Prediction

Structured Problem

$\mathcal{X}$ input space ,

$\mathcal{Y}$ structured output space

Find $f : \mathcal{X} \to \mathcal{Y}$ minimizing the expected risk

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

given examples $(x_i, y_i)_{i=1}^n$

?

Surrogate Methods for Structured Prediction

Structured Problem

$\mathcal{X}$ input space ,

$\mathcal{Y}$ structured output space

Find $f : \mathcal{X} \to \mathcal{Y}$ minimizing the expected risk

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

given examples $(x_i, y_i)_{i=1}^n$

?

coding

$c : \mathcal{Y} \to \mathcal{H}$

Surrogate Problem

$\mathcal{X}$ input space ,

$\mathcal{H}$ surrogate output space
(Hilbert space, possibly infinite dimensional)

$\| \cdot - \cdot \|_{\mathcal{H}}^2$ surrogate loss

Surrogate Methods for Structured Prediction

Structured Problem

$\mathcal{X}$ input space ,

$\mathcal{Y}$ structured output space

Find $f : \mathcal{X} \to \mathcal{Y}$ minimizing the expected risk

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

given examples $(x_i, y_i)_{i=1}^n$

?

coding

$\mathsf{c} : \mathcal{Y} \to \mathcal{H}$

Surrogate Problem

$\mathcal{X}$ input space ,

$\mathcal{H}$ surrogate output space
(Hilbert space, possibly infinite dimensional)

$\|\cdot - \cdot\|_{\mathcal{H}}^2$ surrogate loss

$$\min_{g:\mathcal{X}\to\mathcal{H}} \; n^{-1} \sum_{i=1}^{n} \|g(x_i) - \mathsf{c}(y_i)\|_{\mathcal{H}}^2 + \lambda \|g\|_{\mathsf{HS}}^2$$

$\hat{g}$ vector valued surrogate estimator

Surrogate Methods for Structured Prediction

Structured Problem

$\mathcal{X}$ input space ,

$\mathcal{Y}$ structured output space

Find $f : \mathcal{X} \to \mathcal{Y}$ minimizing the expected risk

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

given examples $(x_i, y_i)_{i=1}^n$

?

coding

$\mathsf{c} : \mathcal{Y} \to \mathcal{H}$

decoding

$\mathsf{d} : \mathcal{H} \to \mathcal{Y}$

Surrogate Problem

$\mathcal{X}$ input space ,

$\mathcal{H}$ surrogate output space

(Hilbert space, possibly infinite dimensional)

$\| \cdot - \cdot \|_{\mathcal{H}}^2$ surrogate loss

$$\min_{g:\mathcal{X}\to\mathcal{H}} n^{-1} \sum_{i=1}^n \|g(x_i) - \mathsf{c}(y_i)\|_{\mathcal{H}}^2 + \lambda \|g\|_{\mathsf{HS}}^2$$

$\hat{g}$ vector valued surrogate estimator

Surrogate Methods for Structured Prediction

Structured Problem

$\mathcal{X}$ input space ,

$\mathcal{Y}$ structured output space

Find $f : \mathcal{X} \to \mathcal{Y}$ minimizing the expected risk

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

given examples $(x_i, y_i)_{i=1}^n$

$$\hat{f} = \mathsf{d} \circ \hat{g}$$

coding

$\mathsf{c} : \mathcal{Y} \to \mathcal{H}$

decoding

$\mathsf{d} : \mathcal{H} \to \mathcal{Y}$

Surrogate Problem

$\mathcal{X}$ input space ,

$\mathcal{H}$ surrogate output space

(Hilbert space, possibly infinite dimensional)

$\| \cdot - \cdot \|_{\mathcal{H}}^2$ surrogate loss

$$\min_{g : \mathcal{X} \to \mathcal{H}} n^{-1} \sum_{i=1}^{n} \| g(x_i) - \mathsf{c}(y_i) \|_{\mathcal{H}}^2 + \lambda \| g \|_{\mathsf{HS}}^2$$

$\hat{g}$ vector valued surrogate estimator

Low-Rank Structured Prediction

$$\min_{g:\mathcal{X}\to\mathcal{H}} n^{-1}\sum_{i=1}^{n}\|g(x_i)-\mathsf{c}(y_i)\|_{\mathcal{H}}^2+\lambda\|g\|_{\mathsf{HS}}^2$$

Single/independent task learning

Is it possible to apply multitask learning techniques to enforce similarity?

Low-Rank Structured Prediction

$$\min_{g:\mathcal{X}\to\mathcal{H}} \; n^{-1}\sum_{i=1}^{n}\|g(x_i)-\mathsf{c}(y_i)\|_{\mathcal{H}}^2 + \lambda\|g\|_*^2$$

Trace norm regularization

Challenges: can we deal with infinite dimensional surrogate spaces?

Low-Rank Structured Prediction

$$\min_{g:\mathcal{X}\to\mathcal{H}} \; n^{-1}\sum_{i=1}^{n} \|g(x_i) - \mathsf{c}(y_i)\|_{\mathcal{H}}^2 + \lambda\|g\|_*^2$$

Trace norm regularization

Challenges: can we deal with infinite dimensional surrogate spaces?

Using the variational formulation of trace norm:

$$\hat{g}(x) = \sum_{i=1}^{n} \alpha_{\mathbf{i}}(x)\mathsf{c}(y_i)$$

The decoding of the setting yields the following estimator for the original problem:

$$\hat{f}(x) = \operatorname{argmin}_{y\in\mathcal{Y}} \sum_{i=1}^{n} \alpha_{\mathbf{i}}(x)\ell(y, y_i)$$

Algorithm 1 - Low-rank Structured Prediction

Input: $K_x, K_y \in \mathbb{R}^{n\times n}$ empirical kernel matrices for input and output data, λ regularizer, r rank, ν step size, k number of iterations.

Initalize: Sample $M_0, N_0 \in \mathbb{R}^{n\times r}$ randomly.

For $j = 0, \ldots, k$:

$$M_{j+1} = (1-\lambda\nu)M_j - \nu(K_x M_j N_j - I)K_y N_j$$
$$N_{j+1} = (1-\lambda\nu)N_j - \nu(N_j M_j^\top K_x - I)K_x M_j$$

Return: The weighting function $\alpha : \mathcal{X} \to \mathbb{R}^n$ with $\alpha(x) = N_k M_k^\top v_x$ for any $x \in \mathcal{X}$

Low-Rank Structured Prediction

Excess risk bounds for the surrogate:

$$\mathcal{R}(\hat{g}) - \min_{g:\mathcal{X}\to\mathcal{H}} \mathcal{R}(g) \leq \mathsf{M}\ n^{-\frac{1}{2}} \quad \text{w.h.p}$$

Surrogate expected risk:

$$\mathcal{R}(g) := \int \|(g(x), \mathsf{c}(y))\|_{\mathcal{H}}^2\, d\rho$$

and structured problem:

$$\mathcal{E}(\hat{f}) - \min_{f:\mathcal{X}\to\mathcal{Y}} \mathcal{E}(f) \leq \sqrt{\mathsf{M}}\ n^{-\frac{1}{4}} \quad \text{w.h.p}$$

Expected risk:

$$\mathcal{E}(f) := \int \ell(f(x), y)\, d\rho$$

Studying the constant M, we identified regimes where the **proposed** MTL **estimator** exhibits **better generalization performance** than its ``independent task'' counterpart.

Experiments

Experiments on ranking problems on a range of datasets:

	ml100k	jester1	jester2	jester3	sushi
MART	0.499 (±0.050)	0.441 (±0.002)	0.442 (±0.003)	0.443 (±0.020)	0.477 (±0.100)
RankNet	0.525 (±0.007)	0.535 (±0.004)	0.531 (±0.008)	0.511 (±0.017)	0.588 (±0.005)
RankBoost	0.576 (±0.043)	0.531 (±0.002)	0.485 (±0.061)	0.496 (±0.010)	0.589 (±0.010)
AdaRank	0.509 (±0.007)	0.534 (±0.009)	0.526 (±0.001)	0.528 (±0.015)	0.588 (±0.051)
Coordinate Ascent	0.477 (±0.108)	0.492 (±0.004)	0.502 (±0.011)	0.503 (±0.023)	0.473 (±0.103)
LambdaMART	0.564 (±0.045)	0.535 (±0.005)	0.520 (±0.013)	0.587 (±0.001)	0.571 (±0.076)
ListNet	0.532 (±0.030)	0.441 (±0.002)	0.442 (±0.003)	0.456 (±0.059)	0.588 (±0.005)
Random Forests	0.526 (±0.022)	0.548 (±0.001)	0.549 (±0.001)	0.581 (±0.002)	0.566 (±0.010)
SVMrank	0.513 (±0.009)	0.507 (±0.007)	0.506 (±0.001)	0.514 (±0.009)	0.541 (±0.005)
SELF + $\lVert \cdot \rVert_{HS}$	0.312 (±0.005)	0.386 (±0.005)	0.366 (±0.002)	0.375 (±0.005)	0.391 (±0.003)
(Ours) SELF + $\lVert \cdot \rVert_{*}$	**0.156** (±**0.005**)	**0.247** (±**0.002**)	**0.340** (±**0.003**)	**0.343** (±**0.003**)	**0.313** (±**0.003**)

Surrogate Low-Rank regularizer is beneficial in practice!

Poster: h 6.30 pm

Poster: Pacific Ballroom #202

Thank you!