Localized Structured Prediction

Carlo Ciliberto[1], Francis Bach[2,3], Alessandro Rudi[2,3]

[1] Department of Electrical and Electronic Engineering, Imperial College London, London
[2] Département d'informatique, Ecole normale supérieure, PSL Research University.
[3] INRIA, Paris, France

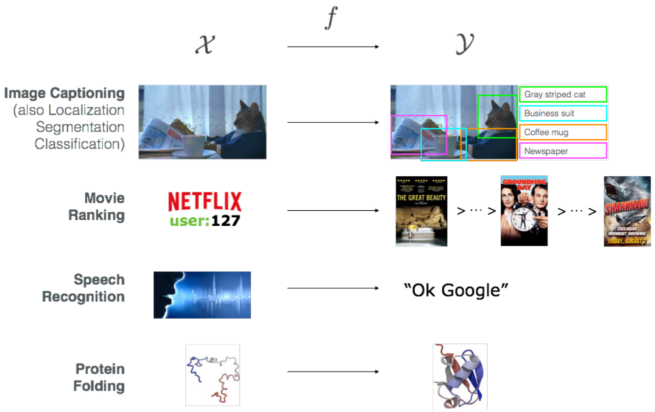
Supervised Learning 101

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

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

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

Structured Prediction



Protypical Approach: Empirical Risk Minimization

Solve the problem:

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

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

Protypical Approach: Empirical Risk Minimization

Solve the problem:

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

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

If $\mathcal{Y}$ is a vector space

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

Protypical Approach: Empirical Risk Minimization

Solve the problem:

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

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

If $\mathcal{Y}$ is a vector space

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

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

- How to choose $\mathcal{G}$? How to optimize over it?

State of the art: Structured case

$\mathcal{Y}$ arbitrary: how do we parametrize $\mathcal{G}$ 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])

Is it possible to have best of both worlds?

general algorithmic framework

+

clear theory

Table of contents

1. A General Framework for Structured Prediction

   [Ciliberto et al., 2016]

2. Leveraging Local Structure

   [This Work]

A General Framework for Structured Prediction

Characterizing the target function

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

Characterizing the target function

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

Pointwise characterization in terms of the **conditional expectation**:

$$f^\star(x) = \operatorname*{argmin}_{z\in\mathcal{Y}} \; \mathbb{E}_y[\ell(z, y) \mid x].$$

Deriving an Estimator

Idea: approximate

$$f^\star(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \; E(z,x) \qquad E(z,x) = \mathbb{E}_y[\ell(z,y) \mid x]$$

by means of an estimator $\widehat{E}(z,x)$ of the ideal $E(z,x)$

$$\widehat{f}(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \; \widehat{E}(z,x) \qquad \widehat{E}(z,x) \approx E(z,x)$$

Question: How to choose $\widehat{E}(z,x)$ given the dataset $(x_i, y_i)_{i=1}^n$?

Estimating the Conditional Expectation

Idea: for every z perform "regression" over the $\ell(z,\cdot)$.

$$\widehat{g}_z = \operatorname*{argmin}_{g:\mathcal{X}\to\mathbb{R}} \frac{1}{n}\sum_{i=1}^{n} L(g(x_i), \ell(z, y_i)) + \lambda R(g)$$

Then we take $\widehat{E}(z,x) = \widehat{g}_z(x)$.

Questions:

- **Models:** How to choose L?
- **Computations:** Do we need to compute $\widehat{g}_z$ for *every* $z \in \mathcal{Y}$?
- **Theory:** Does $\widehat{E}(z,x) \to E(z,x)$? More generally, does $\widehat{f} \to f^\star$?

Square Loss!

Let L be the *square loss*. Then:

$$\widehat{g}_z = \underset{g}{\operatorname{argmin}} \ \frac{1}{n}\sum_{i=1}^{n}(g(x_i) - \ell(z, y_i))^2 + \lambda\|g\|^2$$

In particular, for linear models $g(x) = \phi(x)^\top w$

$$\widehat{g}_z(x) = \phi(x)^\top \widehat{w}_z \qquad \widehat{w}_z = \underset{w}{\operatorname{argmin}} \ \left\|Aw - b\right\|^2 + \lambda\left\|w\right\|^2$$

With

$$A = [\phi(x_1), \dots, \phi(x_n)]^\top \qquad \text{and} \qquad b = [\ell(z, y_1), \dots, \ell(z, y_n)]^\top$$

Computing the $\widehat{g}_z$ All in Once

Closed form solution

$$\widehat{g}_z(x) = \phi(x)^\top \widehat{w}_z = \underbrace{\phi(x)^\top (A^\top A + \lambda n I)^{-1} A^\top}_{\alpha(x)} b = \alpha(x)^\top b$$

In particular, we can compute

$$\alpha_i(x) = \phi(x)^\top (A^\top A + \lambda n I)^{-1} \phi(x_i)$$

only once (independently of z). Then, for any z

$$\widehat{g}_z(x) = \sum_{i=1}^n \alpha_i(x) b_i = \sum_{i=1}^n \alpha_i(x) \ell(z, y_i)$$

Structured Prediction Algorithm

Input: dataset $(x_i, y_i)_{i=1}^n$.

Training: for $i = 1, \dots, n$, compute

$$v_i = (A^\top A + \lambda n I)^{-1} \phi(x_i)$$

Prediction: given a new test point x compute

$$\alpha_i(x) = \phi(x)^\top v_i$$

Then,

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

The Proposed Structured Prediction Algorithm

Questions:

- **Models:** How to choose L?
  Square loss!
- **Computations:** Do we need to compute $\widehat{g}_z$ for *every* $z \in \mathcal{Y}$?
  No need, Compute them all in once!
- **Theory:** Does $\widehat{f} \to f^\star$?
  Yes!

The Proposed Structured Prediction Algorithm

Questions:

- **Models:** How to choose L?
  Square loss!
- **Computations:** Do we need to compute $\widehat{g}_z$ for *every* $z \in \mathcal{Y}$?
  No need, Compute them all in once!
- **Theory:** Does $\widehat{f} \to f^\star$?
  Yes!

Theorem (Rates - [Ciliberto et al., 2016])
Under mild assumption on ℓ. Let $\lambda = n^{-1/2}$, then

$$\mathbb{E}[\ell(\widehat{f}(x), y) - \ell(f^\star(x), y)] \;\; \leq \;\; O(n^{-1/4}), \qquad w.h.p.$$

(General Algorithm + Theory)
Is it possible to have best of both worlds?

Yes!

We introduced an algorithmic framework for structured prediction:

- Directly applicable on a wide family of problems $(\mathcal{Y}, \ell)$.
- With strong theoretical guarantees.
- Recovering many existing algorithms (not seen here).

What Am I Hiding?

- **Theory.** The key assumption to achieve consistency and rates is that ℓ is a **Structure Encoding Loss Function (SELF)**.

$$\ell(z, y) = \langle \psi(z), \varphi(y) \rangle_{\mathcal{H}} \qquad\qquad \forall z, y \in \mathcal{Y}$$

  With $\psi, \varphi : \mathcal{Y} \to \mathcal{H}$ continuous maps into $\mathcal{H}$ Hilbert.

  - Similar to the characterization of reproducing kernels.
  - In principle hard to verify. However lots of ML losses satisfy it!

- **Computations.** We need to solve an optimization problem at prediction time!

Prediction: The Inference Problem

Solving an optimization problem at prediction time is a standard practice in structured prediction. Known as **Inference Problem**

$$\widehat{f}(x) = \operatorname*{argmin}_{z \in \mathcal{Y}} \ \widehat{E}(x, z)$$

In our case it is reminiscient of a weighted barycenter.

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

It is *very* problem dependent

Example: Learning to Rank

Goal: given a query x, order a set of documents $d_1, \dots, d_k$ according to their relevance scores $y_1, \dots, y_k$ w.r.t. x.

Pair-wise Loss: $$\ell rank(f(x), \mathbf{y}) = \sum_{i,j=1}^{k} (y_i - y_j)\, \mathrm{sign}(f(x)_i - f(x)_j)$$

It can be shown that $\widehat{f}(x) = \mathrm{argmin}_{z \in \mathcal{Y}} \; \sum_{i=1}^{n} \alpha_i(x) \ell(z, y_i)$ is a **Minimum Feedback Arc Set** problem on DAGs (NP Hard!)

Still, approximate solutions can improve upon non-consistent approaches.

	Rank Loss
Linear [7]	0.430 ± 0.004
Hinge [27]	0.432 ± 0.008
Logistic [28]	0.432 ± 0.012
SVM Struct [4]	0.451 ± 0.008
Ours	$\mathbf{0.396 \pm 0.003}$

Table 1: Normalized ℓ_{rank} for ranking methods on the MovieLens dataset

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]

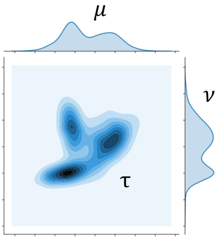
Extensions:

- 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]

Predicting Probability Distributions

[Luise, Rudi, Pontil, Ciliberto '18]

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	$\mathbf{3.7 \pm 0.6}$	4.9 ± 0.9	8.0 ± 2.4	12.0 ± 4.1
4	$\mathbf{22.2 \pm 0.9}$	31.8 ± 1.1	29.2 ± 0.8	40.8 ± 4.2
10	$\mathbf{38.9 \pm 0.9}$	44.9 ± 2.5	48.3 ± 2.4	64.9 ± 1.4

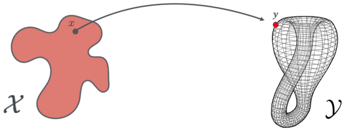
Manifold Regression

[Rudi, Ciliberto, Marconi, Rosasco '18]

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

Loss: (squared) geodesic distance.

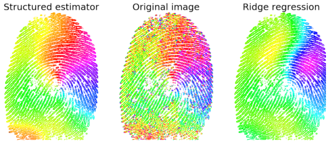
Optimization: Riemannian GD.



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**

[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.).

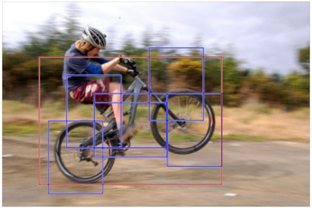
MTL+Structured Prediction

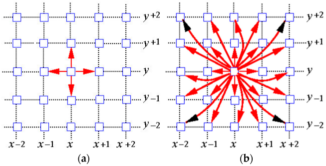
- Interpret multiple tasks as separate outputs.
- Impose constraints as structure on the joint output.

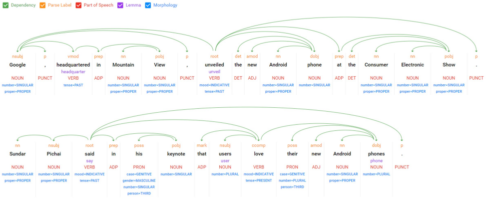
	ml100k	**sushi**
MART	0.499 (±0.050)	0.477 (±0.100)
RankNet	0.525 (±0.007)	0.588 (±0.005)
RankBoost	0.576 (±0.043)	0.589 (±0.010)
AdaRank	0.509 (±0.007)	0.588 (±0.051)
Coordinate Ascent	0.477 (±0.108)	0.473 (±0.103)
LambdaMART	0.564 (±0.045)	0.571 (±0.076)
ListNet	0.532 (±0.030)	0.588 (±0.005)
Random 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)**

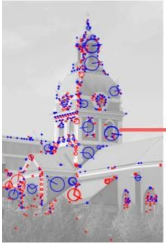
Leveraging local structure

Local Structure



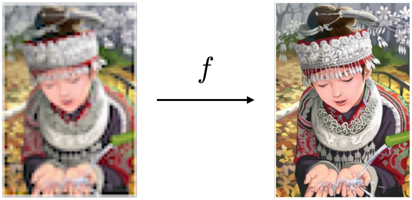






Motivating Example (Between-Locality)

Super-Resolution:
Learn $f : Low_{res} \rightarrow High_{res}$.



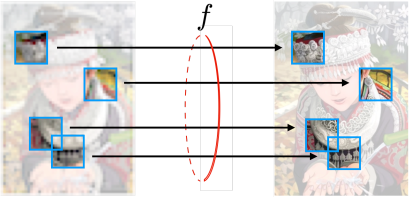
However...

- Very large output sets (high sample complexity).
- Local info might be sufficient to predict output.

Motivating Example (Between-Locality)

Idea: learn local input-output maps under structural constraints
(i.e. overlapping output patches should line up)

Super-Resolution:
Learn $f : Low_{res} \to High_{res}$.



Between-Locality. Let $[x]_p, [y]_p$ denote input/output "parts" $p \in P$:

- $\mathbb{P}\big([y]_p \mid x\big) = \mathbb{P}\big([y]_p \mid [x]_p\big)$
- $\mathbb{P}\big([y]_p \mid [x]_p\big) = \mathbb{P}\big([y]_q \mid [x]_q\big)$

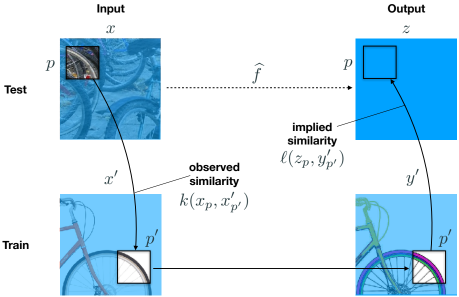
Structured Prediction + Parts

Assumption. The loss is "aware" of the parts.

$$\ell(y', y) = \sum_{p \in P} \ell_0([y']_p, [y]_p)$$

- set P indicizes the parts of $\mathcal{X}$ and $\mathcal{Y}$
- ℓ_0 loss on parts
- $[y]_p$ is the p-th part of y

Localized Structured Prediction: Inference



$$\widehat{f}(x) \;=\; \operatorname*{argmin}_{y' \in \mathcal{Y}} \sum_{p,p' \in P} \sum_{i=1}^{n} \alpha_{i,p'}(x,p)\; \ell_0([y']_p, [y]_{p'})$$

Leveraging Locality

Questions:

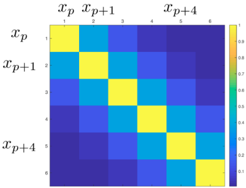
- are we really leveraging locality?
- does the parts structure help?

Problem: if two patches are too similar (i.e. correlated) they do not provide much novel information.

Within-Locality

Intuition: "far-away" parts should be uncorrelated...

More formally, let $d : P \times P \to \mathbb{R}$ be a distance on the parts.



Assumption (Within-Locality). There exists $\gamma \geq 0$ such that

$$\mathsf{C}_{pq} = \mathbb{E}\left[x_p^\top x_q - x_p^\top x_q'\right] \quad \leq \quad e^{-\gamma d(p,q)}$$

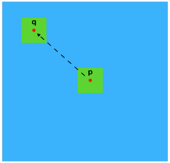
Within-Locality in the Wild

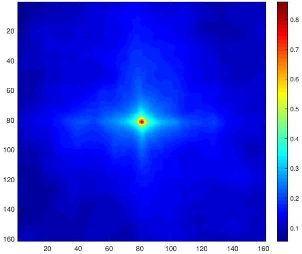
Is within-locality a sensible assumption?

Does it hold in practice on **real** datasets?

Example: (Empirical) Within-locality wrt central patch p on *ImageNet*

$$\widehat{\mathsf{C}}_{pq} = \frac{1}{m} \sum_{i,j=1}^{n} [x_{ip}^\top x_{iq} - x_{ip}^\top x_{jq}]$$





Leveraging Locality

Questions:

- are we really leveraging locality?
- does the parts structure help?

Yes!

Theorem (This work). *Under between-locality...*

- *...and no within-locality (i.e. $\gamma \approx 0$), then*

$$\mathbb{E}[\ell(\widehat{f}(x), y) - \ell(f^\star(x), y)] = O(n^{-1/4}).$$

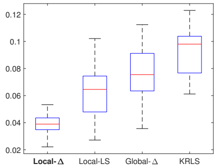
- *...and within-locality (i.e. $\gamma \gg 0$), then*

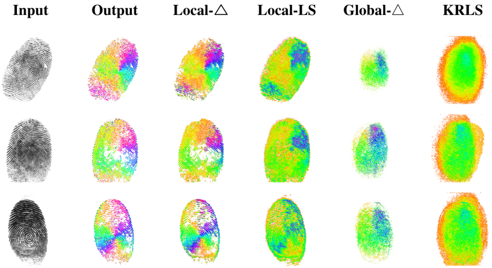
$$\mathbb{E}[\ell(\widehat{f}(x), y) - \ell(f^\star(x), y)] = O((n|P|)^{-1/4}).$$

Predicting the Direction of Ridges in Fingerprint Images

$$f : BW_{images} \rightarrow Angles_{images}$$

The output set is the manifold of ridge orientations (S^1).





Conclusions

A General Framework for Structured Prediction:

- *Algorithm:* Directly applicable on a wide family of problems.
- *Theory:* With strong theoretical guarantees.

Exploiting the local structure:

- *Algorithm:* Directly model locality between input/output parts
  (e.g. images, strings, graphs, etc.).
- *Theory:* Adaptively leverage locality to attain better rates.

Future work:

- Learning the parts (i.e. latent structured prediction).
- Integration with other models (e.g. Deep NN).

References i

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.

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.

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.