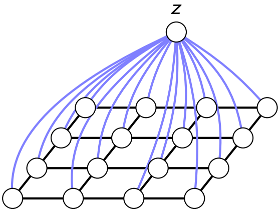
Towards Accurate Structured Output Learning and Prediction

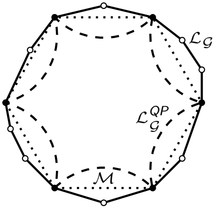


Patrick Pletscher

ETH Zurich

October 25, 2012

pat@pletscher.org



Motivation: Segmentation Problems

Object Recognition in Computer Vision

- grass
- bicycle
- road

- Given an image. Task: locate the objects in it.

Motivation: Segmentation Problems

Object Recognition in Computer Vision

- Given an image. Task: locate the objects in it.
- Strong dependencies among labels between closeby pixels.
  "pixel is likely grass if neighboring pixels are grass"

Motivation: Segmentation Problems

Object Recognition in Computer Vision

- Given an image. Task: locate the objects in it.
- Strong dependencies among labels between closeby pixels.
  "pixel is likely grass if neighboring pixels are grass"
- Key idea: Impose some structural constraints.
  Predict labels of the pixels jointly.

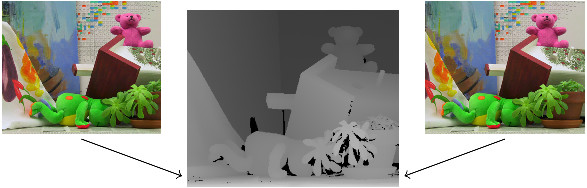
Motivation: Segmentation Problems

Object Recognition in Computer Vision

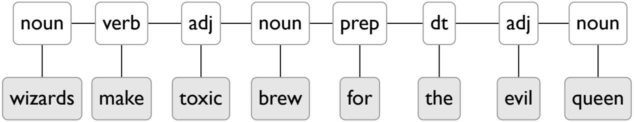
- Given an image. Task: locate the objects in it.
- Strong dependencies among labels between closeby pixels.
  "pixel is likely grass if neighboring pixels are grass"
- Key idea: Impose some structural constraints.
  Predict labels of the pixels jointly.
- This thesis: learning & prediction with structured data.

Additional Examples of Structured Data

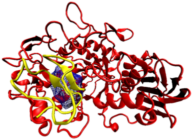
- Computer Vision: Image denoising or stereo.



- Natural language processing: Parsing or part-of-speech tagging.



- Biology: Protein side-chain prediction and design.



Structured Output Prediction

Setting

- Given observed input variables $\boldsymbol{x} \in \mathcal{X}$; usually $\mathcal{X} = \mathbb{R}^D$.
- Predict a *multivariate* discrete output variable $\boldsymbol{y} \in \mathcal{Y}$.
- Learning: Find good predictor $f_{\boldsymbol{w}}(\boldsymbol{x}) : \mathcal{X} \rightarrow \mathcal{Y}$. Parameterized by $\boldsymbol{w}$.
- Energy $E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w})$
  - Cost function to score the different outputs.
  - Models the dependencies.

Structured Output Prediction

Setting

- Given observed input variables $\boldsymbol{x} \in \mathcal{X}$; usually $\mathcal{X} = \mathbb{R}^D$.
- Predict a *multivariate* discrete output variable $\boldsymbol{y} \in \mathcal{Y}$.
- Learning: Find good predictor $f_{\boldsymbol{w}}(\boldsymbol{x}) : \mathcal{X} \rightarrow \mathcal{Y}$. Parameterized by $\boldsymbol{w}$.
- Energy $E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w})$
  - Cost function to score the different outputs.
  - Models the dependencies.

Standard Binary Classification

- Binary classification as a special case: $\mathcal{Y} = \{-1, 1\}$.
- Linear prediction function: $f_{\boldsymbol{w}}(\boldsymbol{x}) = \operatorname{sign}\langle \boldsymbol{w}, \boldsymbol{x} \rangle$.
- Energy: $E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w}) = -y\langle \boldsymbol{w}, \boldsymbol{x} \rangle$

Structured Models: Why Is It Difficult?

Prediction for an Input $\boldsymbol{x}$

Choose the best output: $\boldsymbol{y}^\star = f_{\boldsymbol{w}}(\boldsymbol{x}) = \operatorname{argmin}_{\boldsymbol{y}\in\mathcal{Y}} E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w})$.

Due to dependencies no obvious way how to do this

Structured Models: Why Is It Difficult?

Prediction for an Input $\boldsymbol{x}$

Choose the best output: $\boldsymbol{y}^\star = f_{\boldsymbol{w}}(\boldsymbol{x}) = \operatorname{argmin}_{\boldsymbol{y}\in\mathcal{Y}} E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w})$.

Due to dependencies no obvious way how to do this

Exhaustive Enumeration? Here: Object Recognition

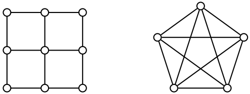
- M pixels, K different object classes. $|\mathcal{Y}| = K^M$ possibilities.
- Usually: $K > 10$, $M \gg 10 \times 10$.
- Compare to: 10^{80} atoms in the universe.
- Prediction as a computational problem.

Need for Clever Algorithms and Approximations

- Some problems exactly tractable. But often only approximate.
- Learning the energy: Even harder as prediction a subprocedure.

SOP Overview – Energy

- "Blueprint of a model" $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.



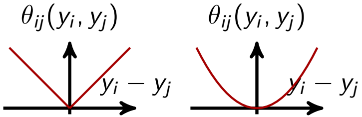
- Linear dependence on parameters:

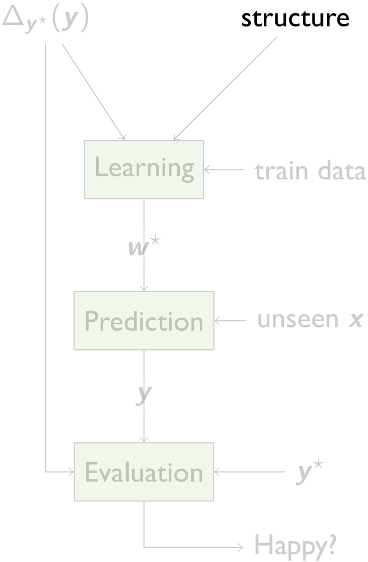
$$E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w}) = -\langle \boldsymbol{w}, \phi(\boldsymbol{x}, \boldsymbol{y}) \rangle$$

- Explicit way to write the energy:

$$E(\boldsymbol{y}) = \sum_{i \in \mathcal{V}} \theta_i(y_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(y_i, y_j)$$

construct θ from $\boldsymbol{w}$ and $\boldsymbol{x}$.





SOP Overview – Loss

- Loss: the smaller the better!

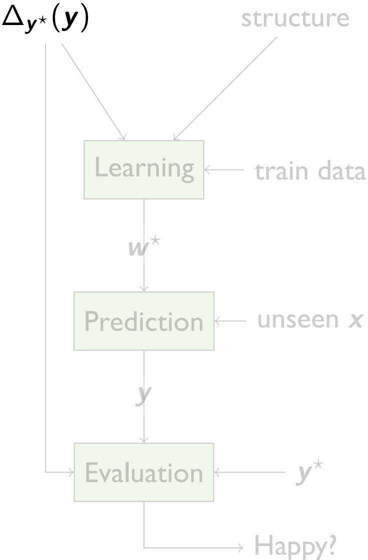
Δ  () $= 0$

Δ  () $= 0.5$

- Example: Missclassified pixels.

$$\Delta_{\boldsymbol{y}^\star}(\boldsymbol{y}) = \sum_{i \in \mathcal{V}} y_i \neq y_i^\star$$

- More complex losses (wrong direction):
  - Overlap of bounding boxes
  - F_1 score
  - Area-under-curve



SOP Overview – Learning

- Goal: Learn a good energy.
  ground-truth has small energy
- Learning = Estimation of $\boldsymbol{w}^\star$.

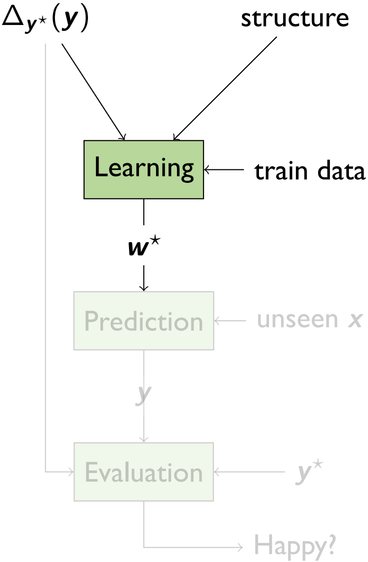
$$\min_{\boldsymbol{w}} \underbrace{\frac{\lambda}{2}\|\boldsymbol{w}\|^2}_{\text{regularizer}} + \frac{1}{N}\sum_{n=1}^{N} \underbrace{\ell(\boldsymbol{w}, \boldsymbol{x}^n, \boldsymbol{y}^n)}_{\text{surrogate loss}}$$

1. Max-margin loss for $(\boldsymbol{x}, \boldsymbol{y}^\star)$:

$$E(\boldsymbol{y}^\star, \boldsymbol{x}, \boldsymbol{w}) - \min_{\boldsymbol{y}\in\mathcal{Y}} [E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w}) - \Delta_{\boldsymbol{y}^\star}(\boldsymbol{y})]$$

2. Log-loss for $(\boldsymbol{x}, \boldsymbol{y}^\star)$:

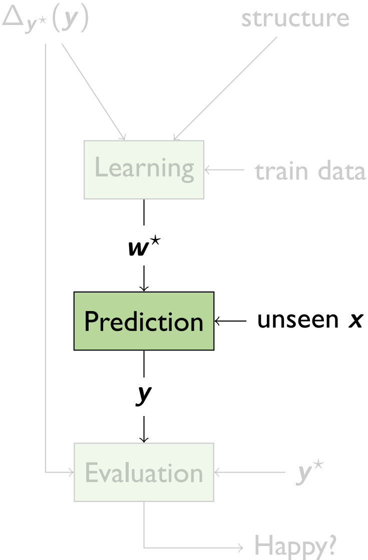
$$E(\boldsymbol{y}^\star, \boldsymbol{x}, \boldsymbol{w}) + \log \sum_{\boldsymbol{y}\in\mathcal{Y}} \exp(-E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w}))$$



SOP Overview – Prediction

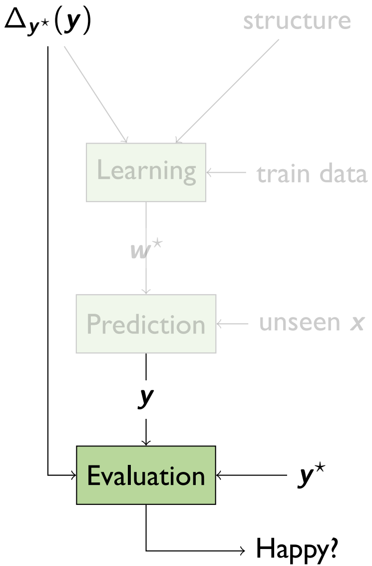
- Given input $\boldsymbol{x}$ and weights $\boldsymbol{w}^\star$.
- Construct potentials: $\boldsymbol{w}^\star, \boldsymbol{x} \mapsto \boldsymbol{\theta}$.
- Minimize energy.

$$\min_{\boldsymbol{y}\in\mathcal{Y}} \sum_{i\in\mathcal{V}} \theta_i(y_i) + \sum_{(i,j)\in\mathcal{E}} \theta_{ij}(y_i, y_j)$$



SOP Overview – Evaluation

Compare prediction $\boldsymbol{y}$ to ground-truth $\boldsymbol{y}^\star$ using the loss function.



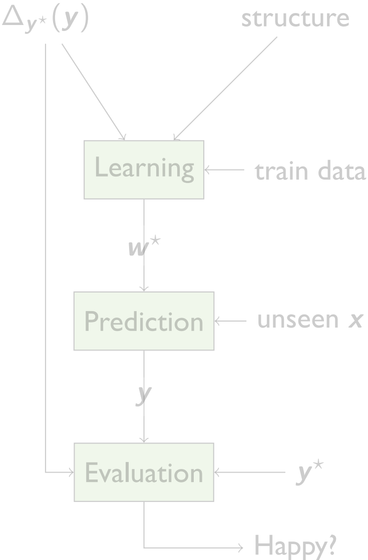
SOP Overview – Tractability

Exact algorithms for loopy graphs only if

1. $E(\boldsymbol{y})$ submodular and binary:

$$\theta_{ij}(0,0)+\theta_{ij}(1,1) \leq \theta_{ij}(0,1)+\theta_{ij}(1,0)$$

2. Loss function "easy".
3. Max-margin for learning.



My PhD Thesis

Learning:

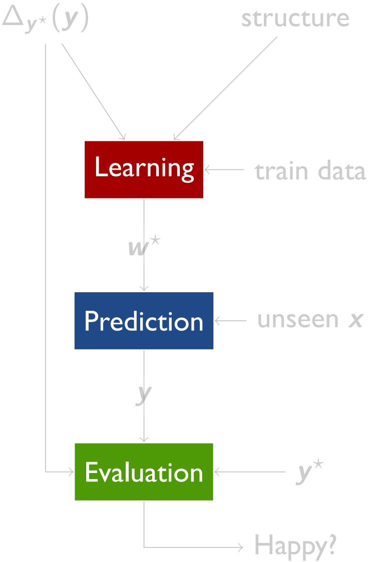
- Pletscher & Kohli, AISTATS 2012.
- Pletscher & Ong, AISTATS 2012.
- Lacoste-Julien, Jaggi, Schmidt & Pletscher, 2012.
- Pletscher, Ong & Buhmann, ECML 2010.
- Pletscher, Ong & Buhmann, AISTATS 2009.

Prediction:

- Pletscher & Wulff, ICML 2012.

Evaluation:

- Pletscher, Nowozin, Kohli & Rother, DAGM 2011.



LPQP for Energy Minimization

Pletscher & Wulff, ICML 2012

Novel Algorithm for Approximate Energy Minimization

- Compute minimum energy assignment for *general pairwise energies*:

$$\min_{\boldsymbol{y}} E(\boldsymbol{y}) = \min_{\boldsymbol{y}} \sum_{i \in \mathcal{V}} \theta_i(y_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(y_i, y_j).$$

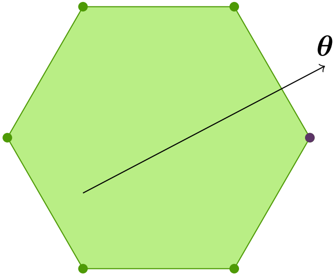
- Combination of two relaxations:
  - Linear Programming (Schlesinger 1976).
  - Quadratic Programming (Ravikumar and Lafferty 2006).
- Efficient Message Passing Algorithms.

Linear and Quadratic Programming Relaxations

LP for Marginal Polytope

$$\min_{\mu \in \mathcal{M}} \quad \sum_{i \in \mathcal{V}} \boldsymbol{\theta}_i^{\mathsf{T}} \boldsymbol{\mu}_i + \sum_{(i,j) \in \mathcal{E}} \boldsymbol{\theta}_{ij}^{\mathsf{T}} \boldsymbol{\mu}_{ij}$$

But: $\mathcal{M}$ is exponentially large!



Linear and Quadratic Programming Relaxations

LP for Marginal Polytope

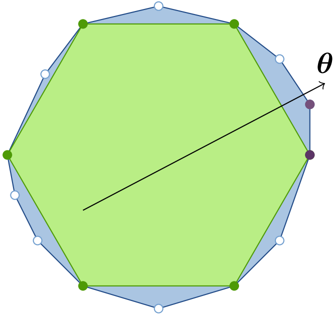
$$\min_{\mu \in \mathcal{M}} \quad \sum_{i \in \mathcal{V}} \boldsymbol{\theta}_i^\mathsf{T} \boldsymbol{\mu}_i + \sum_{(i,j) \in \mathcal{E}} \boldsymbol{\theta}_{ij}^\mathsf{T} \boldsymbol{\mu}_{ij}$$

But: $\mathcal{M}$ is exponentially large!

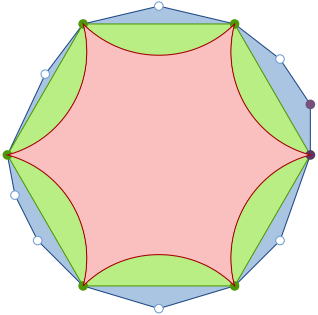
outer approximation

LP for Local Marginal Polytope

$$\min_{\mu \in \mathcal{L}_\mathcal{G}} \sum_{i \in \mathcal{V}} \boldsymbol{\theta}_i^\mathsf{T} \boldsymbol{\mu}_i + \sum_{(i,j) \in \mathcal{E}} \boldsymbol{\theta}_{ij}^\mathsf{T} \boldsymbol{\mu}_{ij}$$



Linear and Quadratic Programming Relaxations



LP for Marginal Polytope

$$\min_{\boldsymbol{\mu}\in\mathcal{M}} \sum_{i\in\mathcal{V}} \boldsymbol{\theta}_i^{\mathsf{T}}\boldsymbol{\mu}_i + \sum_{(i,j)\in\mathcal{E}} \boldsymbol{\theta}_{ij}^{\mathsf{T}}\boldsymbol{\mu}_{ij}$$

But: $\mathcal{M}$ is exponentially large!

outer approximation inner approximation

LP for Local Marginal Polytope

$$\min_{\boldsymbol{\mu}\in\mathcal{L}_{\mathcal{G}}} \sum_{i\in\mathcal{V}} \boldsymbol{\theta}_i^{\mathsf{T}}\boldsymbol{\mu}_i + \sum_{(i,j)\in\mathcal{E}} \boldsymbol{\theta}_{ij}^{\mathsf{T}}\boldsymbol{\mu}_{ij}$$

Quadratic Programming

$$\min_{\boldsymbol{\mu}\in\mathcal{L}_{\mathcal{G}}} \sum_{i\in\mathcal{V}} \boldsymbol{\theta}_i^{\mathsf{T}}\boldsymbol{\mu}_i + \sum_{(i,j)\in\mathcal{E}} \boldsymbol{\theta}_{ij}^{\mathsf{T}}\boldsymbol{\mu}_{ij}$$

$$\text{s.t.} \quad \boldsymbol{\mu}_{ij} = \mathrm{vec}(\boldsymbol{\mu}_i\boldsymbol{\mu}_j^{\mathsf{T}}) \quad \forall (i,j)\in\mathcal{E}$$

LPQP: Combine LP and QP relaxations

Joint LP and QP Objective

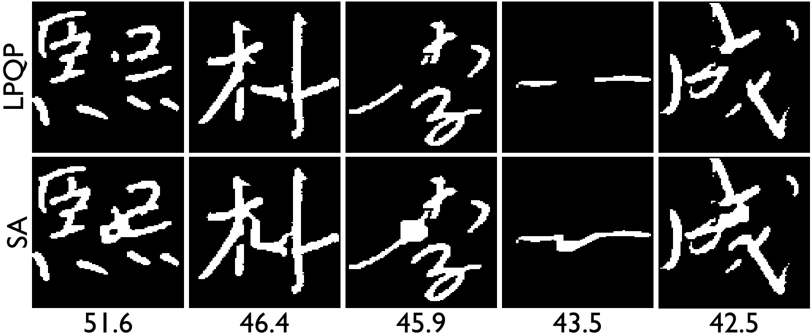
$$\min_{\mu \in \mathcal{L}_{\mathcal{G}}} \boldsymbol{\theta}^{\mathsf{T}} \boldsymbol{\mu} + \underbrace{\rho \sum_{(i,j) \in \mathcal{E}} D_{KL}(\boldsymbol{\mu}_{ij}, \boldsymbol{\mu}_i \boldsymbol{\mu}_j^{\mathsf{T}})}_{\text{encourages consistency}} .$$

Numerical Solution

- Non-convex KL divergence: use the Concave-Convex Procedure.
- Iteratively solve convex optimization problems.
- Efficient message-passing algorithms.
- Gradual increase of ρ.

Decision Tree Fields

LPQP Results in Low Energy Solutions



High-order Loss Functions

Pletscher & Kohli, AISTATS 2012

Exact Max-margin Learning for High-order Losses

- High-order loss $\Delta_{\boldsymbol{y}^\star}(\boldsymbol{y})$: does not factorize into unaries

$$\underbrace{\left|\sum_{i\in\mathcal{V}} y_i - \sum_{i\in\mathcal{V}} y_i^\star\right|}_{\text{Label-count loss}} \quad \text{vs.} \quad \underbrace{\sum_{i\in\mathcal{V}} y_i \neq y_i^\star}_{\text{Hamming loss}}$$

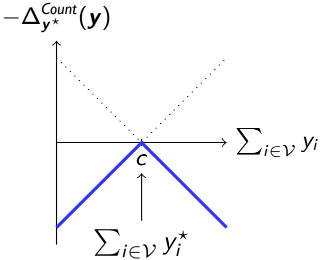
- Maximum margin $\Leftrightarrow$ Energy Minimization:

$$\ell_{mm}(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y}^\star) = E(\boldsymbol{y}^\star, \boldsymbol{x}, \boldsymbol{w}) - \min_{\boldsymbol{y}\in\mathcal{Y}} [E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w}) - \Delta_{\boldsymbol{y}^\star}(\boldsymbol{y})]$$

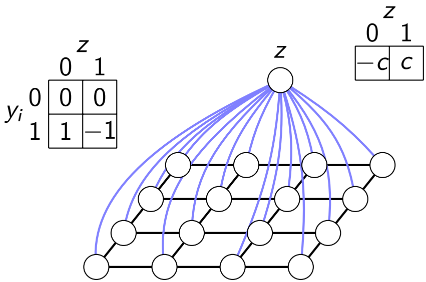
- We characterize a family of tractable high-order loss functions.

Label-count Loss: Introduce Auxiliary Variable

$$\min_{\boldsymbol{y}, z \in \{0,1\}} E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{w}) + \underbrace{\left(2z\left(\sum_{i \in \mathcal{V}} y_i^\star - \sum_{i \in \mathcal{V}} y_i\right) + \sum_{i \in \mathcal{V}} y_i - \sum_{i \in \mathcal{V}} y_i^\star\right)}_{\text{Label-count loss: pairwise representation}}$$



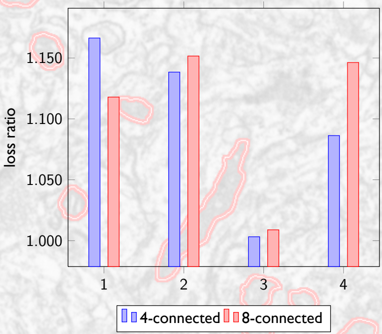
Label-count loss



Pairwise graphical model

Label-count Loss: Results

$$\text{loss ratio} = \frac{\Delta_{\mathbf{y}^\star}(f_{\mathbf{w}}^{Hamming}(\mathbf{x}))}{\Delta_{\mathbf{y}^\star}(f_{\mathbf{w}}^{Count}(\mathbf{x}))}$$



Label-count loss evaluation

Summary & Future Work

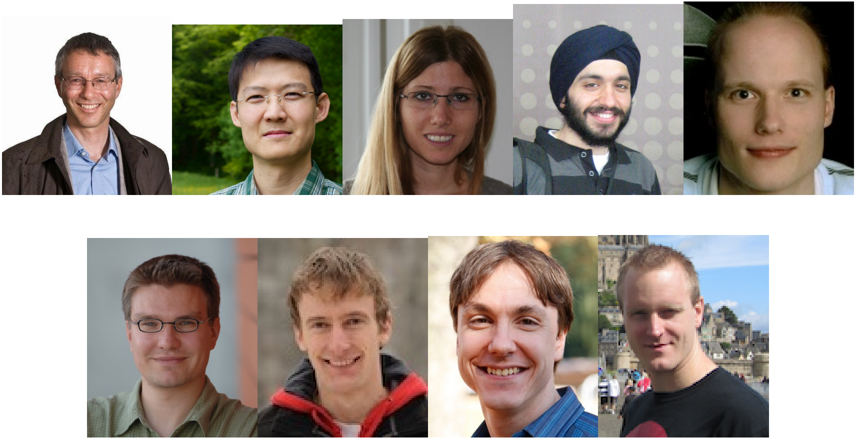
Summary

- Work in most aspects of structured output prediction and learning.
- Goal: fast and accurate approximations for training and prediction.

Future Work

- Smooth-max for direct loss minimization.
- Max-margin learning on a relaxed polytope. Use similar ideas as in LPQP for enforcing consistency.
- Applications. Currently working with ImageNet, a 2 TB dataset.

Thanks To My Collaborators



thanks also to:
Committee, ETHZ ML Groups (JB+AK), Friends/Family

Discussion

References I

Lacoste-Julien, Simon et al. (2012). "Stochastic Block-Coordinate Frank-Wolfe Optimization for Structural SVMs".

Pletscher, Patrick and Pushmeet Kohli (2012). "Learning low-order models for enforcing high-order statistics". In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*. JMLR: W&CP 22, pp. 886–894.

Pletscher, Patrick and Cheng Soon Ong (2012). "Part & Clamp: An efficient algorithm for structured output learning". In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*. JMLR: W&CP 22, pp. 877–885.

Pletscher, Patrick, Cheng Soon Ong, and Joachim M. Buhmann (2009). "Spanning Tree Approximations for Conditional Random Fields". In: *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*. JMLR: W&CP 5, pp. 408–415.

References II

Pletscher, Patrick, Cheng Soon Ong, and Joachim M. Buhmann (2010). "Entropy and Margin Maximization for Structured Output Learning". In: *Proceedings of the 20th European Conference on Machine Learning (ECML)*. Vol. 6321. Lecture Notes in Computer Science, pp. 83–98.

Pletscher, Patrick and Sharon Wulff (2012). "LPQP for MAP: Putting LP Solvers to Better Use". In: *Proceedings of the 29th International Conference on Machine Learning (ICML)*.

Pletscher, Patrick et al. (2011). "Putting MAP back on the Map". In: *33rd Annual Symposium of the German Association for Pattern Recognition (DAGM)*. Vol. 6835. Lecture Notes in Computer Science. Springer, pp. 111–121.

Ravikumar, Pradeep and John Lafferty (2006). "Quadratic Programming Relaxations for Metric Labeling and Markov Random Field MAP Estimation". In: *Proceedings of the 23rd international conference on Machine learning (ICML)*, pp. 737–744.

References III

Schlesinger, M I (1976). “Syntactic analysis of two-dimensional visual signals in noisy conditions”. In: *Kibernetika* 4, pp. 113–130.