High-Order Inference, Ranking, and Regularization Path for Structured SVM

Puneet Kumar Dokania

Supervisors: Prof. M. Pawan Kumar & Prof. Nikos Paragios

CentraleSupélec and INRIA Saclay

May 30, 2016

Presentation Outline

1. Thesis Overview
2. Parsimonious Labeling
3. Learning to Rank Using High-Order Information
4. Regularization Path for SSVM
5. Future Work
6. Publications

Quick Overview

- **High-Order Inference**: Parsimonious Labeling

$$E(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \sum_{i \in V} \theta(x_i, y_i; \mathbf{w}) + \sum_{c \in \mathcal{C}} \underbrace{\theta_c(\mathbf{x}_c, \mathbf{y}_c; \mathbf{w})}_{diversity}$$

Quick Overview

- **High-Order Inference**: Parsimonious Labeling

$$E(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \sum_{i \in V} \theta(x_i, y_i; \mathbf{w}) + \sum_{c \in \mathcal{C}} \underbrace{\theta_c(\mathbf{x}_c, \mathbf{y}_c; \mathbf{w})}_{\textit{diversity}}$$

- **HOAP-SVM**: $\mathbf{w}$ very high-dimensional $\rightarrow$ exhaustive search ??

$$\min_{\mathbf{w}} \quad \frac{\lambda}{2} \|\mathbf{w}\|^2 + \underbrace{L(\mathbf{x}, \mathbf{y}; \mathbf{w})}_{\textit{AP-Based}}$$

Quick Overview

- **High-Order Inference**: Parsimonious Labeling

$$E(\mathbf{x},\mathbf{y};\mathbf{w}) = \sum_{i \in V} \theta(x_i, y_i; \mathbf{w}) + \sum_{c \in \mathcal{C}} \underbrace{\theta_c(\mathbf{x}_c, \mathbf{y}_c; \mathbf{w})}_{diversity}$$

- **HOAP-SVM**: $\mathbf{w}$ very high-dimensional $\rightarrow$ exhaustive search ??

$$\min_{\mathbf{w}} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \underbrace{L(\mathbf{x},\mathbf{y};\mathbf{w})}_{AP\text{-}Based}$$

- **Regularization path for SSVM**: Efficiently explore the entire space of $\lambda \in [0, \infty]$

Presentation Outline

The Labeling Problem

Input

- *Lattice* $V = \{1, \cdots, N\}$, *Random variables* $\mathbf{y} = \{y_1, \cdots, y_N\}$
- *A discrete label set* $\mathcal{L} = \{l_1, \cdots, l_H\}$
- *Energy functional to assess the quality of each labeling* $\mathbf{y}$*:*

$$E(\mathbf{y}) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in \mathcal{C}} \theta_c(\mathbf{y}_c). \tag{1}$$

The Labeling Problem

Input

- *Lattice* $V = \{1, \cdots, N\}$, *Random variables* $\mathbf{y} = \{y_1, \cdots, y_N\}$
- *A discrete label set* $\mathcal{L} = \{l_1, \cdots, l_H\}$
- *Energy functional to assess the quality of each labeling* $\mathbf{y}$*:*

$$E(\mathbf{y}) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in \mathcal{C}} \theta_c(\mathbf{y}_c). \quad (1)$$

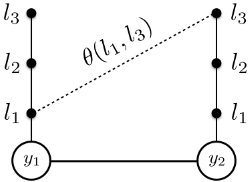
Output

- *Labeling corresponding to the minimum energy*

$$\boxed{\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmin}}\, E(\mathbf{y}).} \quad (2)$$

- H^N *possible labelings*

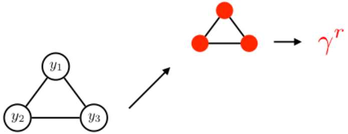
Special case – Metric Labeling (Pairwise)



- Pairwise Potentials $\theta(y_i, y_j) \rightarrow$ Metric over the labels
- Recall, distance function $\theta(y_i, y_j) : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}_+$ is metric if:
    - Non Negative
    - Symmetric
    - Triangular Inequality
- $\alpha-$expansion[1] – Very Efficient – Approximate solution

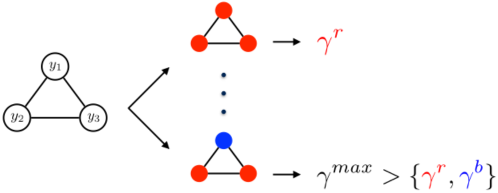
[1] **Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, 2001.**

Special case – P^n Potts Model[2] (High-Order)



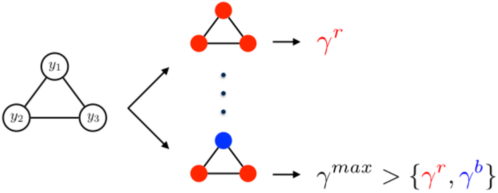
[2] **Kohli et al.**, **P3 & Beyond: Solving Energies with Higher Order Cliques**, **2007**.

Special case – P^n Potts Model[2] (High-Order)



[2] **Kohli et al., P3 & Beyond: Solving Energies with Higher Order Cliques, 2007.**

Special case – P^n Potts Model[2] (High-Order)



- P^n **Potts Model**

$$\theta_c(\mathbf{y}_c) \propto \begin{cases} \gamma^k, & \text{if } y_i = l_k, \forall i \in c, \\ \gamma^{max}, & \text{otherwise,} \end{cases}$$

- Very efficient α-expansion algorithm – Approximate solution

[2] **Kohli et al., P3 & Beyond: Solving Energies with Higher Order Cliques, 2007.**

Parsimonious Labeling: Energy Function

$$E(\mathbf{y}) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in \mathcal{C}} \theta_c(\mathbf{y}_c).$$

- Unary potentials: Arbitrary

Parsimonious Labeling: Energy Function

$$E(\mathbf{y}) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in \mathcal{C}} \theta_c(\mathbf{y}_c).$$

- Unary potentials: Arbitrary
- Clique potentials: **Diversity**

$$\theta_c(\mathbf{y}_c) \propto \underbrace{\delta(\Gamma(\mathbf{y}_c))}_{\text{Diversity}}$$

where, $\Gamma(\mathbf{y}_c)$ is the set of unique labels

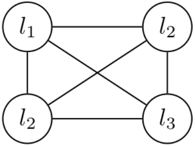
Parsimonious Labeling: Energy Function

$$E(\mathbf{y}) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in \mathcal{C}} \theta_c(\mathbf{y}_c).$$

- Unary potentials: Arbitrary
- Clique potentials: **Diversity**

$$\theta_c(\mathbf{y}_c) \propto \underbrace{\delta(\Gamma(\mathbf{y}_c))}_{\text{Diversity}}$$

where, $\Gamma(\mathbf{y}_c)$ is the set of unique labels



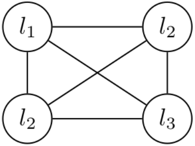
$\delta_c(\{l_1, l_2, l_3\})$

Parsimonious Labeling: Energy Function

$$E(\mathbf{y}) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in \mathcal{C}} \theta_c(\mathbf{y}_c).$$

- Unary potentials: Arbitrary
- Clique potentials: **Diversity**

$$\theta_c(\mathbf{y}_c) \propto \underbrace{\delta(\Gamma(\mathbf{y}_c))}_{\text{Diversity}}$$



where, $\Gamma(\mathbf{y}_c)$ is the set of unique labels

$$\delta_c(\{l_1, l_2, l_3\})$$

- Energy function for *Parsimonious Labeling*

$$E(\mathbf{y}) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in \mathcal{C}} w_c \underbrace{\delta(\Gamma(\mathbf{y}_c))}_{\text{Diversity}}$$

Diversity[3]: Metric over sets

$$\theta_c(\mathbf{y}_c) \propto \underbrace{\delta(\Gamma(\mathbf{y}_c))}_{\text{Diversity}}$$

[3] **Bryant and Tupper**, **Advances in Mathematics**, **2012.**

Diversity[3]: Metric over sets

$$\theta_c(\mathbf{y}_c) \propto \underbrace{\delta(\Gamma(\mathbf{y}_c))}_{\text{Diversity}}$$

- Metric over sets $\delta : \bar{\mathcal{L}} \rightarrow \mathbb{R}, \forall \bar{\mathcal{L}} \subseteq \mathcal{L}$, satisfying
    - Non Negativity
    - Triangular Inequality
    - Monotonicity: $\mathcal{L}_1 \subseteq \mathcal{L}_2$ implies $\delta(\mathcal{L}_1) \leq \delta(\mathcal{L}_2) \rightarrow$ Parsimony

[3] **Bryant and Tupper**, **Advances in Mathematics**, **2012.**

Diversity[3]: Metric over sets

$$\theta_c(\mathbf{y}_c) \propto \underbrace{\delta(\Gamma(\mathbf{y}_c))}_{\text{Diversity}}$$

- Metric over sets $\delta : \bar{\mathcal{L}} \rightarrow \mathbb{R}, \forall \bar{\mathcal{L}} \subseteq \mathcal{L}$, satisfying
  - Non Negativity
  - Triangular Inequality
  - Monotonicity: $\mathcal{L}_1 \subseteq \mathcal{L}_2$ implies $\delta(\mathcal{L}_1) \leq \delta(\mathcal{L}_2) \rightarrow$ Parsimony
- **Induced Metric**: Every diversity induces a metric:

$$d(l_i, l_j) = \delta(\{l_i, l_j\})$$

[3] **Bryant and Tupper**, **Advances in Mathematics**, **2012.**

Diversity[3]: Metric over sets

$$\theta_c(\mathbf{y}_c) \propto \underbrace{\delta(\Gamma(\mathbf{y}_c))}_{\text{Diversity}}$$

- Metric over sets $\delta : \bar{\mathcal{L}} \rightarrow \mathbb{R}, \forall \bar{\mathcal{L}} \subseteq \mathcal{L}$, satisfying
  - Non Negativity
  - Triangular Inequality
  - Monotonicity: $\mathcal{L}_1 \subseteq \mathcal{L}_2$ implies $\delta(\mathcal{L}_1) \leq \delta(\mathcal{L}_2) \rightarrow$ Parsimony
- **Induced Metric**: Every diversity induces a metric:

$$d(l_i, l_j) = \delta(\{l_i, l_j\})$$

- **Diameter Diversity**: $\delta^{dia}(\mathcal{L}) = \max_{l_i, l_j \in \mathcal{L}} d(l_i, l_j)$

[3] **Bryant and Tupper**, **Advances in Mathematics**, **2012**.

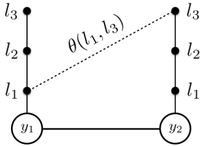
Special Case 1: Metric Labeling

- If cliques are of size 2 $\rightarrow$ diversity $\rightarrow$ metric

[4] **Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, 2001.**

Special Case 1: Metric Labeling

- If cliques are of size 2 $\rightarrow$ diversity $\rightarrow$ metric
- Parsimonious Labeling $\rightarrow$ Metric Labeling[4]



- Many applications in low level vision tasks: Stereo matching, Inpainting, Denoising, Image stitching.

[4] **Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, 2001.**

Special Case 2: P^n-Potts Model[5]

- Uniform Metric

$$d(l_i, l_j) = \min(|l_i - l_j|, 1), \forall l_i, l_j \in \mathcal{L}$$

[5] **Kohli et al.**, **P3 "& Beyond: Solving Energies with Higher Order Cliques**, **2007**.

Special Case 2: P^n-Potts Model[5]

- Uniform Metric

$$d(l_i, l_j) = \min(|l_i - l_j|, 1), \forall l_i, l_j \in \mathcal{L}$$

- Diversity → Diameter diversity over uniform metric
- Parsimonious Labeling → P^n-Potts Model

[5] **Kohli et al.**, **P3 "& Beyond: Solving Energies with Higher Order Cliques**, **2007**.

Special Case 2: P^n-Potts Model[5]

- Uniform Metric

$$d(l_i, l_j) = \min(|l_i - l_j|, 1), \forall l_i, l_j \in \mathcal{L}$$

- Diversity → Diameter diversity over uniform metric
- Parsimonious Labeling → P^n-Potts Model

Labels	l_1	l_2	l_3
l_1	0	1	1
l_2	1	0	1
l_3	1	1	0

Table: Uniform Metric

[5] **Kohli et al., P3 "& Beyond: Solving Energies with Higher Order Cliques, 2007.**

Special Case 2: P^n-Potts Model[5]

- Uniform Metric

$$d(l_i, l_j) = \min(|l_i - l_j|, 1), \forall l_i, l_j \in \mathcal{L}$$

- Diversity $\rightarrow$ Diameter diversity over uniform metric
- Parsimonious Labeling $\rightarrow$ P^n-Potts Model

Labels	l_1	l_2	l_3
l_1	0	1	1
l_2	1	0	1
l_3	1	1	0

Table: Uniform Metric

$$\begin{aligned}\theta_c(\{l_1, l_2, l_3\}) &= \max(d(l_1, l_2), d(l_1, l_3), d(l_2, l_3)) \\ &= 1\end{aligned}$$

$$\theta_c(\mathbf{y}_c) \propto \begin{cases} 0, & \text{if } y_i = l_k, \forall i \in c, \\ 1, & \text{otherwise,} \end{cases}$$

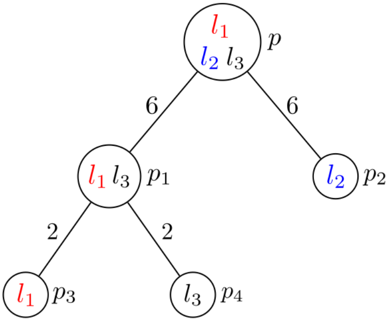
[5] **Kohli et al., P3 "& Beyond: Solving Energies with Higher Order Cliques, 2007.**

So far ...

$$E(\mathbf{y}) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in \mathcal{C}} w_c \underbrace{\delta(\Gamma(\mathbf{y}_c))}_{\text{Diversity}}$$

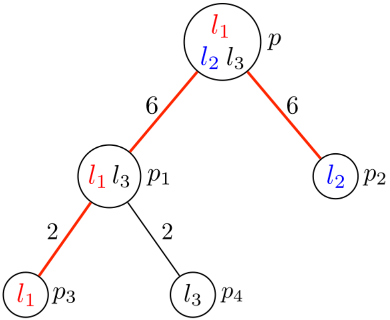
Hierarchical P^n Potts Model

- Given tree metric



Hierarchical P^n Potts Model

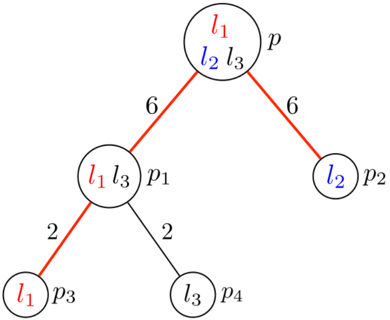
- Given tree metric



- $d^t(l_1, l_2) = 14$, $d^t(l_1, l_3) = 4$, $d^t(l_1, l_1) = 0$

Hierarchical P^n Potts Model

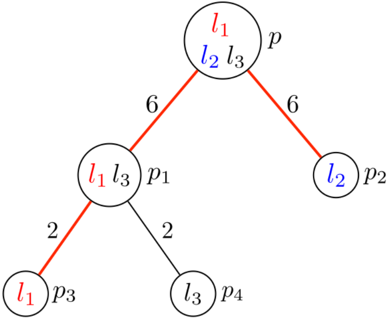
- Given tree metric



- $d^t(l_1, l_2) = 14$, $d^t(l_1, l_3) = 4$, $d^t(l_1, l_1) = 0$
- Hierarchical P^n Potts Model $\rightarrow$ diameter diversity over tree metric

Hierarchical P^n Potts Model

- Given tree metric



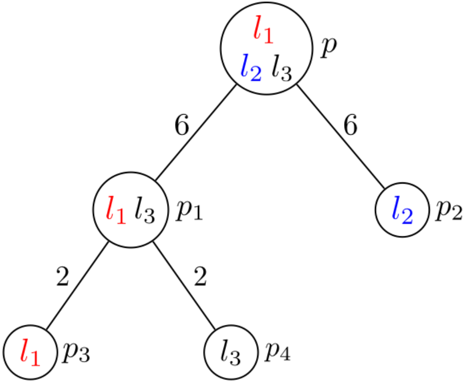
- $d^t(l_1, l_2) = 14$, $d^t(l_1, l_3) = 4$, $d^t(l_1, l_1) = 0$
- Hierarchical P^n Potts Model $\rightarrow$ diameter diversity over tree metric
- Diameter diversity at cluster p is $\max_{\{l_i, l_j\}} d^t(l_i, l_j) = 14$.

Move Making Algorithm for Hierarchical P^n Potts Model

- Optimizing directly at the root node is non-trivial
- We propose divide and conquer based bottom-up approach

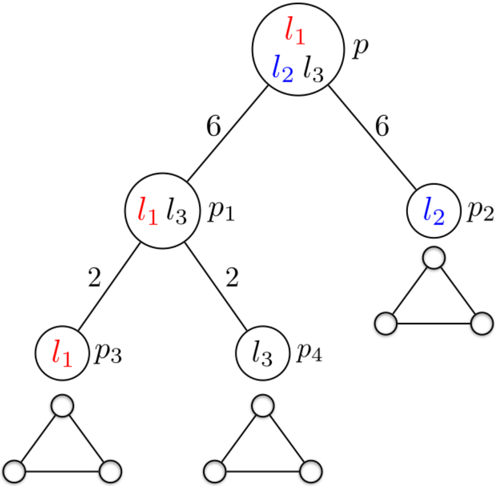
Move Making Algorithm for Hierarchical P^n Potts Model

- Optimizing directly at the root node is non-trivial
- We propose divide and conquer based bottom-up approach



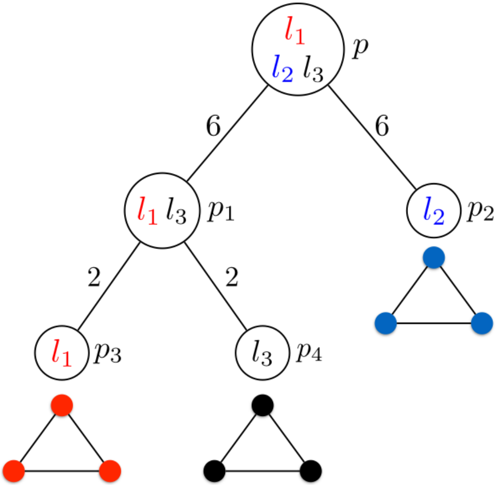
Move Making Algorithm for Hierarchical P^n Potts Model

- Optimizing directly at the root node is non-trivial
- We propose divide and conquer based bottom-up approach



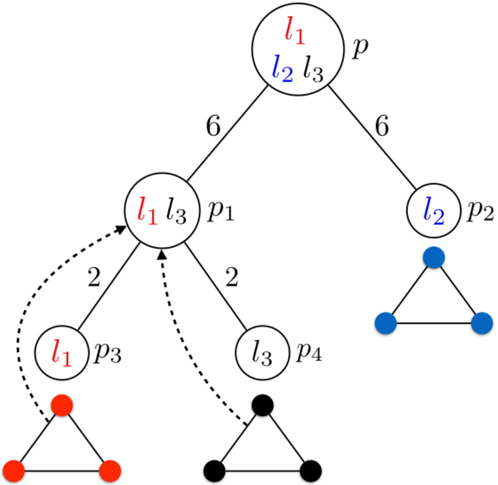
Move Making Algorithm for Hierarchical P^n Potts Model

- Optimizing directly at the root node is non-trivial
- We propose divide and conquer based bottom-up approach



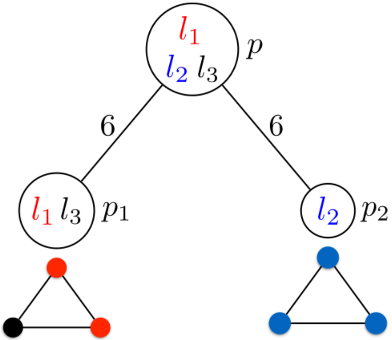
Move Making Algorithm for Hierarchical P^n Potts Model

- Optimizing directly at the root node is non-trivial
- We propose divide and conquer based bottom-up approach



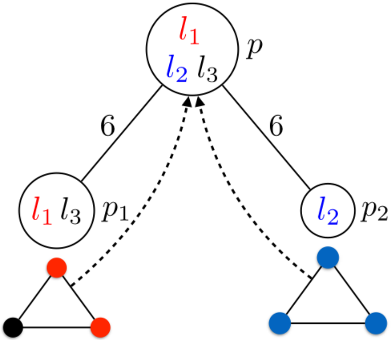
Move Making Algorithm for Hierarchical P^n Potts Model

- Optimizing directly at the root node is non-trivial
- We propose divide and conquer based bottom-up approach



Move Making Algorithm for Hierarchical P^n Potts Model

- Optimizing directly at the root node is non-trivial
- We propose divide and conquer based bottom-up approach

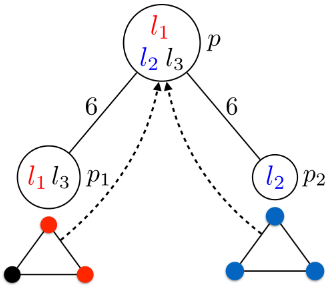


Move Making Algorithm for Hierarchical P^n Potts Model

- Solving the problem at leaf node $\rightarrow$ Trivial

Move Making Algorithm for Hierarchical P^n Potts Model

- Solving the problem at leaf node $\rightarrow$ Trivial
- Fusing at non-leaf node $\rightarrow$ P^n-Potts Model



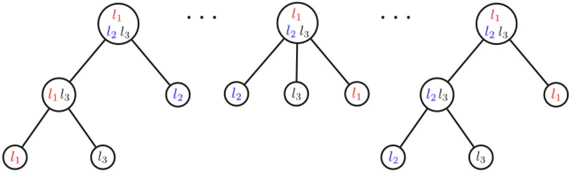
Move Making for Parsimonious Labeling

- Given any general diversity → Get the induced metric

[6] **Fakcharoenphol et al., In STOC 2003**.

Move Making for Parsimonious Labeling

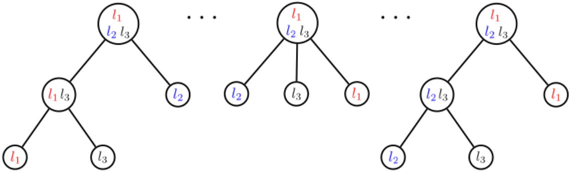
- Given any general diversity → Get the induced metric
- Induced Metric → Mixture of tree metrics (r-HST)[6]



[6] **Fakcharoenphol et al., In STOC 2003.**

Move Making for Parsimonious Labeling

- Given any general diversity → Get the induced metric
- Induced Metric → Mixture of tree metrics (r-HST)[6]

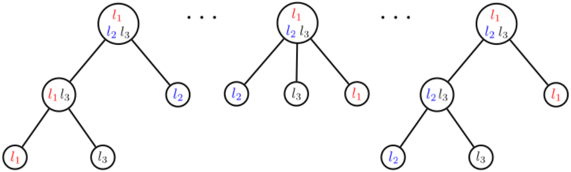


- Hierarchical P^n-Potts model over each tree metric → diameter diversity over each tree metric (r-HST)

[6] **Fakcharoenphol et al., In STOC 2003**.

Move Making for Parsimonious Labeling

- Given any general diversity → Get the induced metric
- Induced Metric → Mixture of tree metrics (r-HST)[6]

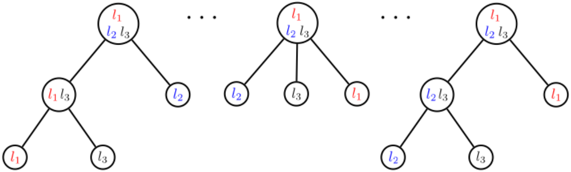


- Hierarchical P^n-Potts model over each tree metric → diameter diversity over each tree metric (r-HST)
- Optimize each Hierarchical P^n-Potts model using proposed move making algorithm

[6] **Fakcharoenphol et al., In STOC 2003.**

Move Making for Parsimonious Labeling

- Given any general diversity → Get the induced metric
- Induced Metric → Mixture of tree metrics (r-HST)[6]



- Hierarchical P^n-Potts model over each tree metric → diameter diversity over each tree metric (r-HST)
- Optimize each Hierarchical P^n-Potts model using proposed move making algorithm
- Fuse solutions or choose the one with minimum energy

[6] **Fakcharoenphol et al., In STOC 2003**.

Comparison

- Co-oc[7]:
  - Clique potentials $\rightarrow$ Monotonic
  - Very fast optimization algorithm
  - No theoretical guarantees

[7] **Ladicky, Russell, Kohli, and Torr, ECCV 2010.**
[8] **Fix, Wang, and Zabih, CVPR 2014.**
[9] **Dokania and Kumar, ICCV 2015.**

Comparison

- Co-oc[7]:
  - Clique potentials $\rightarrow$ Monotonic
  - Very fast optimization algorithm
  - No theoretical guarantees
- SoSPD[8]:
  - Clique potentials $\rightarrow$ Arbitrary $\rightarrow$ Upperbound as SoS functions
  - Slow. Practically, can not go beyond the clique of size 9
  - Loose multiplicative bound

[7] **Ladicky, Russell, Kohli, and Torr, ECCV 2010.**
[8] **Fix, Wang, and Zabih, CVPR 2014.**
[9] **Dokania and Kumar, ICCV 2015.**

Comparison

- Co-oc[7]:
  - Clique potentials $\rightarrow$ Monotonic
  - Very fast optimization algorithm
  - No theoretical guarantees
- SoSPD[8]:
  - Clique potentials $\rightarrow$ Arbitrary $\rightarrow$ Upperbound as SoS functions
  - Slow. Practically, can not go beyond the clique of size 9
  - Loose multiplicative bound
- Parsimonious Labeling[9]:
  - Clique potentials $\rightarrow$ Diversities
  - Very fast. We experimented with cliques of size $\approx$ 1200.
  - Can be parallelized over the trees and over the levels.
  - Very tight multiplicative bound.

---

[7] **Ladicky, Russell, Kohli, and Torr, ECCV 2010.**

[8] **Fix, Wang, and Zabih, CVPR 2014.**

[9] **Dokania and Kumar, ICCV 2015.**

Experimental Setting

- **Energy Function:**

$$E(\mathbf{y}) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in \mathcal{C}} w_c \underbrace{\delta(\Gamma(\mathbf{y}_c))}_{\text{Diversity}} \quad (3)$$

- **Clique Potential**: Diameter diversity over truncated Linear Metric:

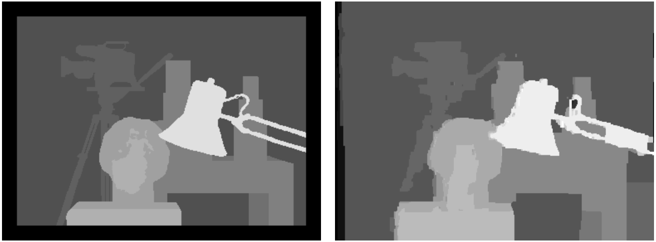
$$\theta_{i,j}(l_a, l_b) = \lambda \min(|l_a - l_b|, M), \forall l_a, l_b \in \mathcal{L}$$

- **Cliques**: Superpixels generate using Mean Shift.
- **Clique Weights**:

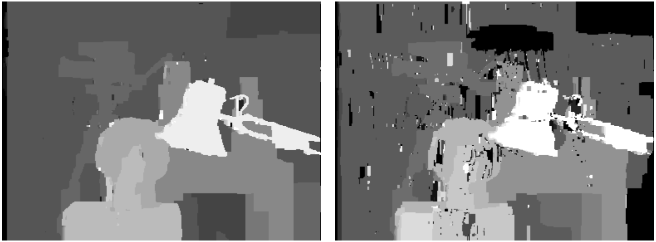
$$w_c = exp\left(\frac{-\rho(\mathbf{y}_c)}{\sigma^2}\right)$$

where, $\rho(\mathbf{y}_c)$ is the variance of intensities of pixels in clique $\mathbf{y}_c$.

Stereo Matching Results – Visually



(a) Ground Truth (b) Our



(c) $\alpha-$Exp (d) Co-oc

Stereo Matching Results – Energy and Time

(a) Our
$(E = \mathbf{1.4 \times 10^6}, 773\ sec)$

(b) Co-oc
$(E = 2.1 \times 10^6, \mathbf{306}\ sec)$

Image denoising and Inpainting Results – Visually

(a) Original

(b) Our

(c) $\alpha-$Exp

(d) Co-oc

Image denoising and Inpainting Results – Energy and Time

(a) Our
($E = \mathbf{1.2 \times 10^7}$, 1964 *sec*)

(b) Co-occ
($E = 1.4 \times 10^7$, **358** *sec*)

Presentation Outline

Ranking?

Ranking?

0.5 0.3 0.1 -0.2 -0.7 -0.9

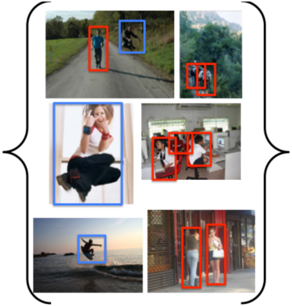
Ranking?

- Get the feature vector $\phi(x_i)$
- Learn $\mathbf{w}$
- Sort using $s_i(\mathbf{w}) = \mathbf{w}^\top \phi(x_i)$

Ranking?

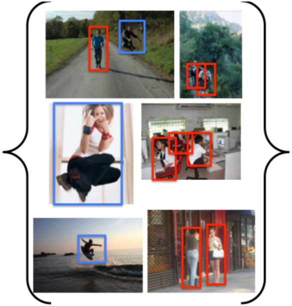
- Get the feature vector $\phi(x_i)$
- Learn $\mathbf{w}$
- Sort using $s_i(\mathbf{w}) = \mathbf{w}^\top \phi(x_i)$
- SVM $\rightarrow$ Optimizes accuracy
- Accuracy $\neq$ Average Precision

AP-SVM[10]: Problem Formulation



[10] **Yue et al., A support vector method for optimizing average precision, 2007.**

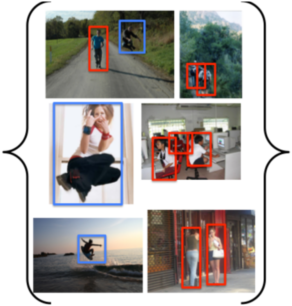
AP-SVM[10]: Problem Formulation



- Single input $\mathbf{x}$, Positive Set $\mathcal{P}$, Negative Set $\mathcal{N}$
- $\phi(x_i), \forall i \in \mathcal{P}, \phi(x_j), \forall j \in \mathcal{N}$

[10] Yue et al., A support vector method for optimizing average precision, 2007.

AP-SVM[10]: Problem Formulation



- Single input $\mathbf{x}$, Positive Set $\mathcal{P}$, Negative Set $\mathcal{N}$
- $\phi(x_i), \forall i \in \mathcal{P}, \phi(x_j), \forall j \in \mathcal{N}$
- Rank Matrix

$$\mathbf{R}_{ij} = \begin{cases} +1, & \text{if i is better ranked than j} \\ -1, & \text{if j is better ranked than i} \end{cases}$$

- Define Joint Score:

$$S(\mathbf{x}, \mathbf{R}; \mathbf{w}) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \mathbf{R}_{ij}(s_i(\mathbf{w}) - s_j(\mathbf{w}))$$

Encodes Ranking

[10] **Yue et al., A support vector method for optimizing average precision, 2007.**

AP-SVM: Objective Function

- Loss function $\Delta(\mathbf{R}, \mathbf{R}^*) = 1 - AP(\mathbf{R}, \mathbf{R}^*)$
- Objective Function

$$\min_{\mathbf{w}, \xi} \quad \frac{\lambda}{2} \|\mathbf{w}\|^2 + \xi \tag{4}$$

$$\text{s.t.} \quad S(\mathbf{x}, \mathbf{R}^*; \mathbf{w}) \geq S(\mathbf{x}, \mathbf{R}; \mathbf{w}) + \Delta(\mathbf{R}, \mathbf{R}^*) - \xi, \quad \forall \mathbf{R}. \tag{5}$$

- Loss augmented inference: $\bar{\mathbf{R}} = argmax_{\mathbf{R}} \{S(\mathbf{x}, \bar{\mathbf{R}}; \mathbf{w}) + \Delta(\mathbf{R}, \mathbf{R}^*)\}$, greedy algorithm $\mathcal{O}(|\mathcal{P}||\mathcal{N}|)$ by Yue et.al.

AP-SVM: Joint Score

- Joint Score:

$$S(\mathbf{x}, \mathbf{R}; \mathbf{w}) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \mathbf{R}_{ij} (s_i(\mathbf{w}) - s_j(\mathbf{w}))$$

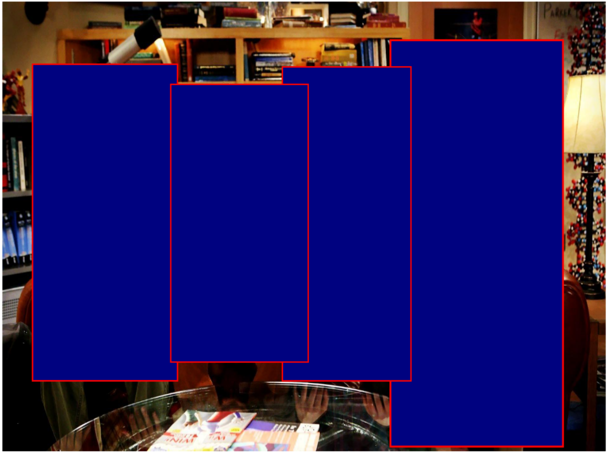
Encodes Ranking

- Sample Score:

$$s_i(\mathbf{w}) = \mathbf{w}^\top \phi(x_i)$$

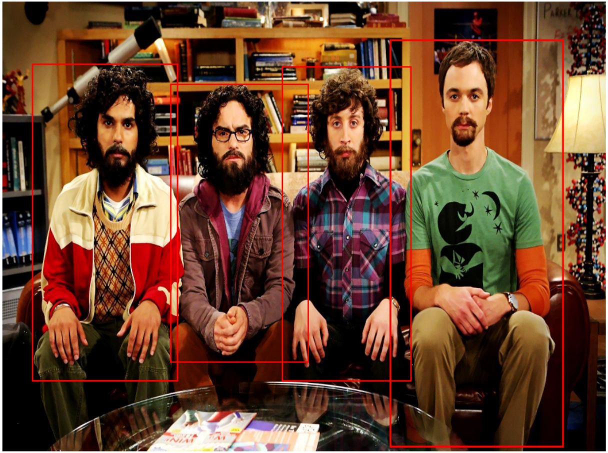
No High-Order Information

Why High-Order Information?

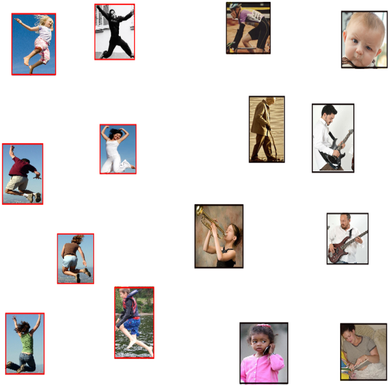


Why High-Order Information?

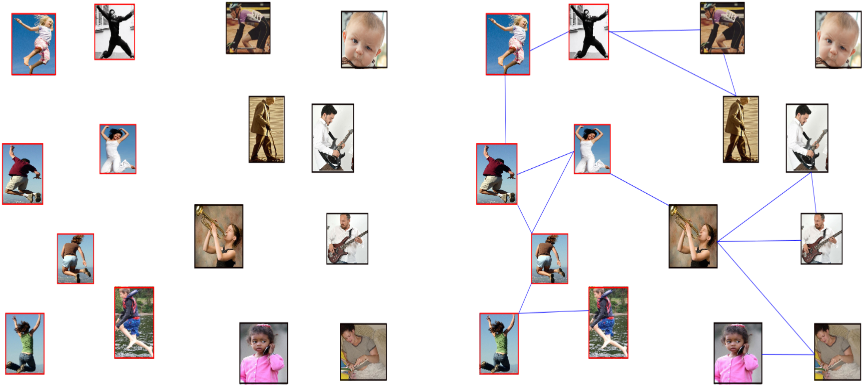
Why High-Order Information?



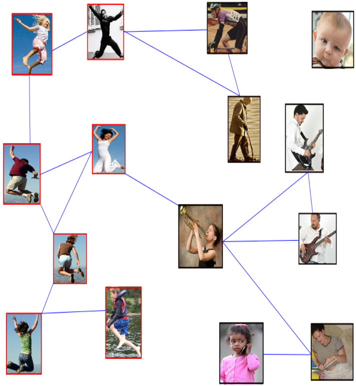
Encoding High-Order Information



Encoding High-Order Information



Encoding High-Order Information

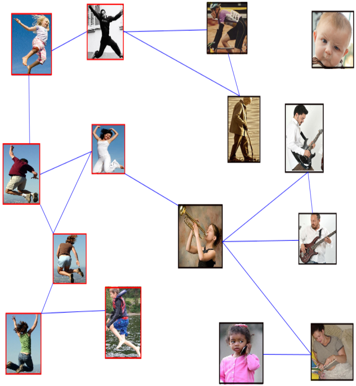


- Define Joint Feature Map (encodes the structure)

$$\Phi(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \sum_i \Phi_1(x_i, y_i) \\ \sum_{i,j} \Phi_2(x_i, y_i, x_j, y_j) \end{pmatrix}$$

  - Φ_1 - first-order information
  - Φ_2 - high-order information
- Joint labeling: $\mathbf{y} \in \{-1, +1\}^n$

Encoding High-Order Information



- Define Joint Feature Map (encodes the structure)

$$\Phi(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \sum_i \Phi_1(x_i, y_i) \\ \sum_{i,j} \Phi_2(x_i, y_i, x_j, y_j) \end{pmatrix}$$

  - Φ_1 - first-order information
  - Φ_2 - high-order information
- Joint labeling: $\mathbf{y} \in \{-1, +1\}^n$
- Define Score $S(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y})$

Joint Score: Closer look

$$\mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix}^\top \begin{pmatrix} \sum_i \Phi_1(x_i, y_i) \\ \sum_{i,j} \Phi_2(x_i, y_i, x_j, y_j) \end{pmatrix}$$

$$= \underbrace{\sum_i \mathbf{w}_1^\top \Phi_1(x_i, y_i) + \sum_{i,j} \mathbf{w}_2^\top \Phi_2(x_i, y_i, x_j, y_j)}_{\textit{Encodes High-Order Information}} \tag{6}$$

Joint Score: Closer look

$$\mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix}^\top \begin{pmatrix} \sum_i \Phi_1(x_i, y_i) \\ \sum_{i,j} \Phi_2(x_i, y_i, x_j, y_j) \end{pmatrix}$$
$$= \underbrace{\sum_i \mathbf{w}_1^\top \Phi_1(x_i, y_i) + \sum_{i,j} \mathbf{w}_2^\top \Phi_2(x_i, y_i, x_j, y_j)}_{\textit{Encodes High-Order Information}} \quad (6)$$

- Single score for the entire dataset $\rightarrow$ Ranking?

Ranking Using Max-Marginals

- We propose to use difference of max-marginals

[11] **Kohli et al., In PAMI 2007**.

Ranking Using Max-Marginals

- We propose to use difference of max-marginals
- $s(x_i; \mathbf{w}) = m_i^+(\mathbf{w}) - m_i^-(\mathbf{w})$, where, $m_i^+(\mathbf{w})$ is the max-marginal score such that sample x_i takes label of $+1$.

$$m_i^+(\mathbf{w}) = argmax_{\mathbf{y}, y_i=+1} \mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y})$$

- Dynamic Graph Cuts[11] — Very Efficient

[11] **Kohli et al., In PAMI 2007.**

HOAP-SVM: Score

Score that can **encode ranking** and high-order information

HOAP-SVM: Score

Score that can **encode ranking** and high-order information

- Joint Score for the given ranking

$$S(\mathbf{x}, \mathbf{R}; \mathbf{w}) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \mathbf{R}_{ij}(s_i(\mathbf{w}) - s_j(\mathbf{w}))$$

Encodes Ranking

HOAP-SVM: Score

Score that can **encode ranking** and high-order information

- Joint Score for the given ranking

$$S(\mathbf{x}, \mathbf{R}; \mathbf{w}) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \mathbf{R}_{ij}(s_i(\mathbf{w}) - s_j(\mathbf{w}))$$

Encodes Ranking

- Sample score s_i as difference of max-marginals

$$s_i(\mathbf{w}) = m_i^+(\mathbf{w}) - m_i^-(\mathbf{w})$$

Encodes High-Order Information

HOAP-SVM: Objective Function

- Objective Function

$$\min_{\mathbf{w},\xi} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \xi \tag{7}$$

$$\text{s.t.} \quad S(\mathbf{x},\mathbf{R}^*;\mathbf{w}) \geq S(\mathbf{x},\mathbf{R};\mathbf{w}) + \Delta(\mathbf{R},\mathbf{R}^*) - \xi, \quad \forall \mathbf{R}, \tag{8}$$

$$\mathbf{w}_2 \leq 0, \xi \geq 0.$$

HOAP-SVM: Objective Function

- Objective Function

$$\min_{\mathbf{w},\xi} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \xi \tag{7}$$

$$\text{s.t.} \quad S(\mathbf{x},\mathbf{R}^*;\mathbf{w}) \geq S(\mathbf{x},\mathbf{R};\mathbf{w}) + \Delta(\mathbf{R},\mathbf{R}^*) - \xi, \quad \forall \mathbf{R}, \tag{8}$$

$$\mathbf{w}_2 \leq 0, \xi \geq 0.$$

- Each max-marginal is a convex function (max over affine functions)

$$m_i^+(\mathbf{w}) = argmax_{\mathbf{y},y_i=+1}\mathbf{w}^\top\Phi(\mathbf{x},\mathbf{y})$$

HOAP-SVM: Objective Function

- Objective Function

$$\min_{\mathbf{w},\xi} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \xi \tag{7}$$

$$\text{s.t.} \quad S(\mathbf{x},\mathbf{R}^*;\mathbf{w}) \geq S(\mathbf{x},\mathbf{R};\mathbf{w}) + \Delta(\mathbf{R},\mathbf{R}^*) - \xi, \quad \forall \mathbf{R}, \tag{8}$$

$$\mathbf{w}_2 \leq 0, \xi \geq 0.$$

- Each max-marginal is a convex function (max over affine functions)

$$m_i^+(\mathbf{w}) = argmax_{\mathbf{y},y_i=+1}\mathbf{w}^\top\Phi(\mathbf{x},\mathbf{y})$$

- The objective function is a difference of convex program

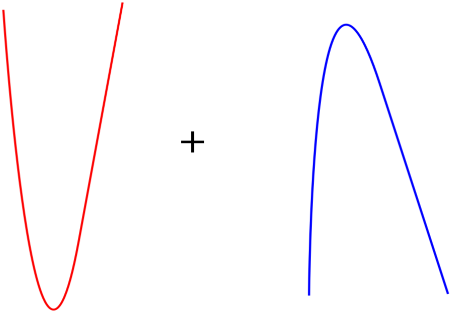
CCCP[12]

Difference of convex functions can be optimized using CCCP algorithm

[12] **Yuille et al., The concave-convex procedure, 2003.**

CCCP[12]

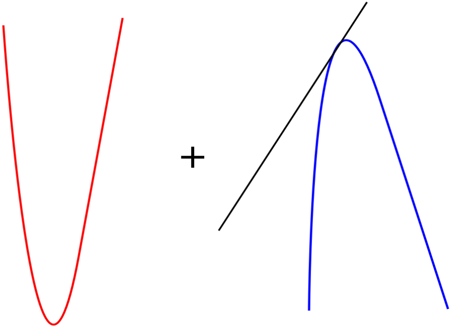
Difference of convex functions can be optimized using CCCP algorithm



[12] **Yuille et al., The concave-convex procedure, 2003.**

CCCP[12]

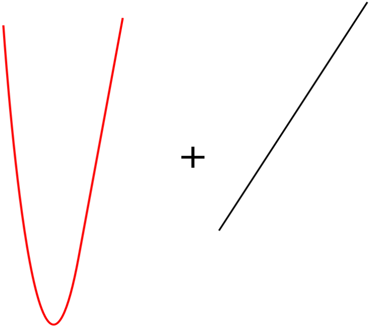
Difference of convex functions can be optimized using CCCP algorithm



[12] **Yuille et al., The concave-convex procedure, 2003.**

CCCP[12]

Difference of convex functions can be optimized using CCCP algorithm



[12] **Yuille et al., The concave-convex procedure, 2003.**

Action Recognition

- PASCAL VOC 2011 Dataset
- 10 Action Classes
- Unary Feature - POSELET and GIST concatenated
- High-Order Feature -POSELET
- High-Order Information
  - Hypothesis: Persons in the same image are more likely to perform same action
  - Connected bounding boxes coming from the same image

PASCAL VOC Results - Average AP over all 10 action classes

Method	Trainval	Test
SVM	54.7/+4.2	48.82/+4.93
AP-SVM	56.2/+2.7	51.42/+2.33
HOAP-SVM	**58.9**	**53.75**

Visualization - Reading top 4

Presentation Outline

1. Thesis Overview
2. Parsimonious Labeling
3. Learning to Rank Using High-Order Information
4. Regularization Path for SSVM
5. Future Work
6. Publications

Regularization Path: What and Why

- Optimize SSVM objective function

$$\min_{\mathbf{w},\xi} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad \textit{set of constraints}$$

- $\lambda \rightarrow$ important for good generalization $\rightarrow$ cross validate
- $\lambda \in [0, \infty] \rightarrow$ cross validation over subset $\rightarrow$ poor generalization

Regularization Path: What and Why

- Optimize SSVM objective function

$$\min_{\mathbf{w},\xi} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad \textit{set of constraints}$$

- $\lambda \rightarrow$ important for good generalization $\rightarrow$ cross validate
- $\lambda \in [0, \infty]$ $\rightarrow$ cross validation over subset $\rightarrow$ poor generalization
- ϵ-optimal regularization path algorithm

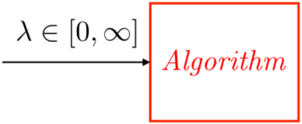
Algorithm

Regularization Path: What and Why

- Optimize SSVM objective function

$$\min_{\mathbf{w},\xi} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad \textit{set of constraints}$$

- $\lambda \rightarrow$ important for good generalization $\rightarrow$ cross validate
- $\lambda \in [0, \infty] \rightarrow$ cross validation over subset $\rightarrow$ poor generalization
- ϵ-optimal regularization path algorithm



Regularization Path: What and Why

- Optimize SSVM objective function

$$\min_{\mathbf{w},\xi} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad \textit{set of constraints}$$

- $\lambda \rightarrow$ important for good generalization $\rightarrow$ cross validate
- $\lambda \in [0, \infty] \rightarrow$ cross validation over subset $\rightarrow$ poor generalization
- ϵ-optimal regularization path algorithm

Dual Objective and Duality Gap

- SSVM dual objective function

$$\begin{aligned} \min_{\alpha} \quad & f(\alpha) \rightarrow \textit{smooth convex} \\ s.t. \quad & \sum_{\mathbf{y} \in \mathcal{Y}_i} \alpha_i(\mathbf{y}) = 1, \forall i \in [n], \\ & \alpha_i(\mathbf{y}) \geq 0, \forall i \in [n], \forall \mathbf{y} \in \mathcal{Y}_i. \end{aligned}$$

where, $\alpha = (\alpha_1, \cdots, \alpha_n) \in \mathbb{R}^{|\mathcal{Y}_1|} \times \cdots \mathbb{R}^{|\mathcal{Y}_n|}$.

Dual Objective and Duality Gap

- SSVM dual objective function

$$
\begin{aligned}
\min_{\alpha} \quad & f(\alpha) \rightarrow \textit{smooth convex} \\
\textit{s.t.} \quad & \sum_{\mathbf{y} \in \mathcal{Y}_i} \alpha_i(\mathbf{y}) = 1, \forall i \in [n], \\
& \alpha_i(\mathbf{y}) \geq 0, \forall i \in [n], \forall \mathbf{y} \in \mathcal{Y}_i.
\end{aligned}
$$

where, $\alpha = (\alpha_1, \cdots, \alpha_n) \in \mathbb{R}^{|\mathcal{Y}_1|} \times \cdots \mathbb{R}^{|\mathcal{Y}_n|}$.

- Duality Gap

$$
g(\alpha; \lambda) = \frac{1}{n} \sum_i \Big(\max_{\mathbf{y} \in \mathcal{Y}_i} H_i(\mathbf{y}; \mathbf{w}) - \sum_{\mathbf{y} \in \mathcal{Y}_i} \alpha_i(\mathbf{y}) H_i(\mathbf{y}; \mathbf{w}) \Big)
$$

where, $H_i(\mathbf{y}; \mathbf{w})$ is the hinge loss.

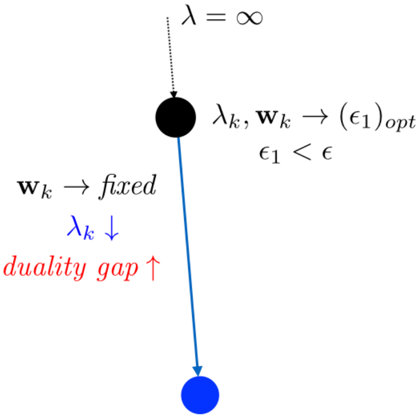
Key Idea: ϵ-Optimal Regularization Path

$\lambda = \infty$

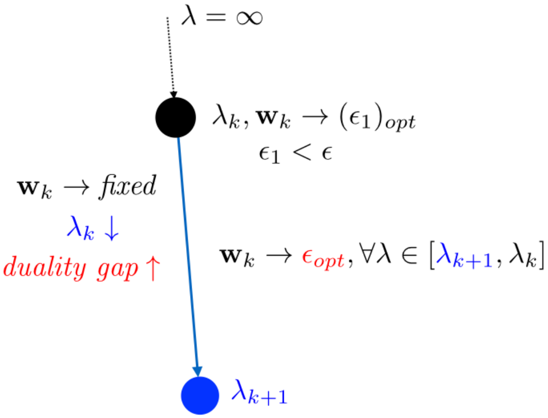
$\lambda_k, \mathbf{w}_k \rightarrow (\epsilon_1)_{opt}$

$\epsilon_1 < \epsilon$

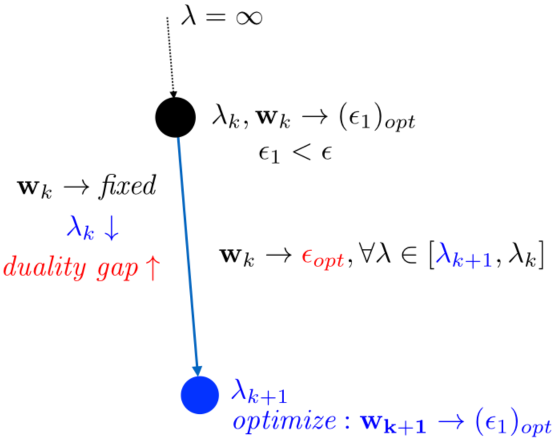
Key Idea: ϵ-Optimal Regularization Path



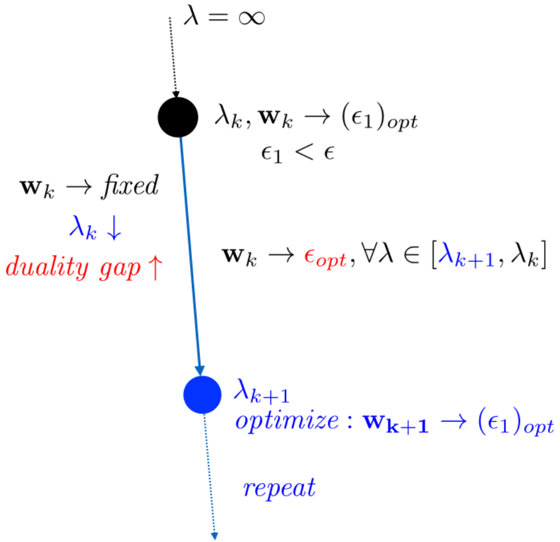
Key Idea: ϵ-Optimal Regularization Path



Key Idea: ϵ-Optimal Regularization Path



Key Idea: ϵ-Optimal Regularization Path



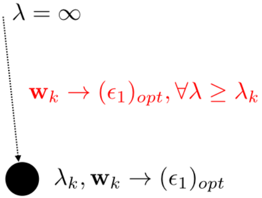
Challenge 1: How do we start?

$\lambda = \infty$

$\mathbf{w}_k \rightarrow (\epsilon_1)_{opt}, \forall \lambda \geq \lambda_k$

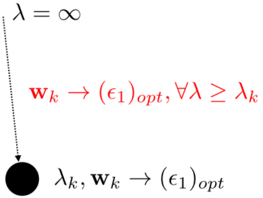
$\lambda_k, \mathbf{w}_k \rightarrow (\epsilon_1)_{opt}$

Challenge 1: How do we start?



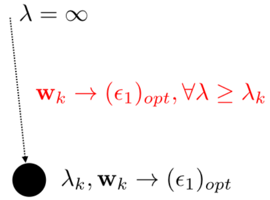
- Let $\tilde{\mathcal{Y}}_i = \text{argmax}_{\mathbf{y} \in \mathcal{Y}_i} \Delta(\mathbf{y}, \mathbf{y}_i)$ be the loss-maximizer and $\tilde{\mathbf{y}}_i \in \tilde{\mathcal{Y}}_i, \forall i$.

Challenge 1: How do we start?



- Let $\tilde{\mathcal{Y}}_i = \text{argmax}_{\mathbf{y} \in \mathcal{Y}_i} \Delta(\mathbf{y}, \mathbf{y}_i)$ be the loss-maximizer and $\tilde{\mathbf{y}}_i \in \tilde{\mathcal{Y}}_i, \forall i$.
- Let $\tilde{\Psi} = \frac{1}{n} \sum_i \Psi_i(\tilde{\mathbf{y}}_i)$, where $\Psi_i(\mathbf{y}) = \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \mathbf{y})$.

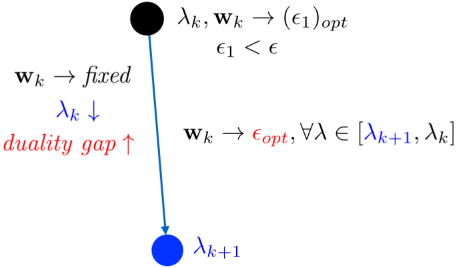
Challenge 1: How do we start?



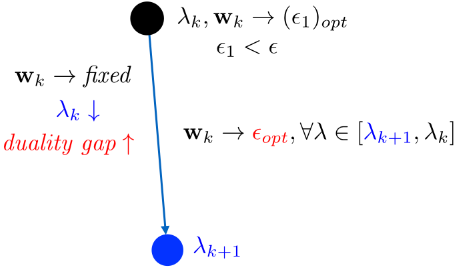
- Let $\tilde{\mathcal{Y}}_i = \text{argmax}_{\mathbf{y} \in \mathcal{Y}_i} \Delta(\mathbf{y}, \mathbf{y}_i)$ be the loss-maximizer and $\tilde{\mathbf{y}}_i \in \tilde{\mathcal{Y}}_i, \forall i$.
- Let $\tilde{\Psi} = \frac{1}{n} \sum_i \Psi_i(\tilde{\mathbf{y}}_i)$, where $\Psi_i(\mathbf{y}) = \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \mathbf{y})$.
- Then, $\mathbf{w}_k = \frac{\tilde{\Psi}}{\lambda}$ is guaranteed to be ϵ_1 optimal for any λ satisfying the condition:

$$\lambda \geq \frac{\left\|\tilde{\Psi}\right\|^2 + \frac{1}{n} \sum_i \underbrace{\max_{\mathbf{y} \in \mathcal{Y}_i} \left(- \tilde{\Psi}^\top \Psi(\mathbf{y}) \right)}_{\textit{Inference}}}{\epsilon_1} \tag{9}$$

Challenge 2: How to find the breakpoints?

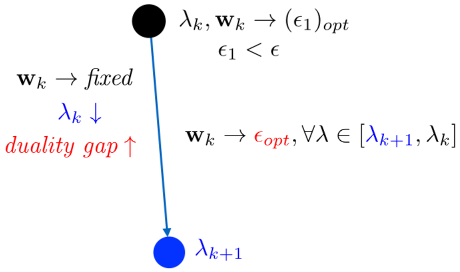


Challenge 2: How to find the breakpoints?



- Let $\lambda_{k+1} = \eta\lambda_k$, $0 \leq \eta \leq 1$.

Challenge 2: How to find the breakpoints?



- Let $\lambda_{k+1} = \eta\lambda_k$, $0 \leq \eta \leq 1$.
- $\mathbf{w}_k \rightarrow \epsilon_{opt}$, for all λ_{k+1} obtained using η satisfying the condition:.

$$1 - \frac{\epsilon - g(\alpha^k; \lambda_k)}{\Omega(\alpha^k, \lambda_k)} \leq \eta \leq 1 \tag{10}$$

where, $\Omega(\alpha^k, \lambda_k) := \ell^{\alpha^k} - \lambda^k \mathbf{w}_k^\top \mathbf{w}_k$

Challenge 2: Proof Sketch

Challenge 2: Proof Sketch

- Keeping $\mathbf{w}_k$ constant – from KKT condition

$$\mathbf{w}_k = \frac{1}{n} \sum_{i \in [n], \mathbf{y} \in \mathcal{Y}_i} \frac{\alpha_i^k(\mathbf{y})}{\lambda_k} \Psi(\mathbf{x}_i, \mathbf{y}).$$

Challenge 2: Proof Sketch

- Keeping $\mathbf{w}_k$ constant – from KKT condition

$$\mathbf{w}_k = \frac{1}{n} \sum_{i\in[n],\, \mathbf{y}\in\mathcal{Y}_i} \frac{\alpha_i^k(\mathbf{y})}{\lambda_k} \Psi(\mathbf{x}_i, \mathbf{y}).$$

- Therefore, using

$$\frac{\alpha_i^{k+1}(\mathbf{y})}{\lambda_{k+1}} = \frac{\alpha_i^k(\mathbf{y})}{\lambda_k}, \forall \mathbf{y} \neq \mathbf{y}_i; \quad \sum_{\mathbf{y}\in\mathcal{Y}_i} \alpha_i(\mathbf{y}) = 1, \forall i \in [n]; \quad \lambda_{k+1} = \eta\lambda_k$$

- New duality gap

$$g(\alpha^{k+1}; \lambda_{k+1}) = \underbrace{g(\alpha^k; \lambda_k)}_{\textit{Old gap}} + (1-\eta)\Omega(\alpha^k, \lambda_k)$$

$$\leq \epsilon$$

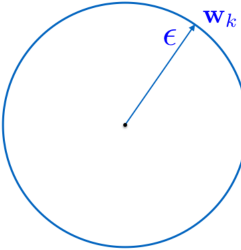
Challenge 3: How to optimize efficiently?

λ_{k+1}
$optimize : \mathbf{w_{k+1}} \rightarrow (\epsilon_1)_{opt}$

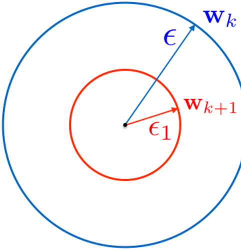
- Notice that, $\mathbf{w}_k$ is already ϵ-optimal at λ_{k+1}
- Warm starting with $\mathbf{w}_k$ requires us to reduce the duality gap only by $(\epsilon - \epsilon_1) \rightarrow$ very fast convergence
- We use Block-Coordinate Frank-Wolfe algorithm[13] for the optimization.
  - Lagrange duality gap is the by product

[13] **Lacoste-Julien et al., In ICML 2013.**

Effects of ϵ_1



Effects of ϵ_1



- Decrease ϵ_1:
  - $(\epsilon - \epsilon_1)$ increases — More passes through the data to get $(\epsilon_1)_{opt}$ solution.
  - η decreases — big jumps — number of breakpoints decreases (see below)

$$\lambda_{k+1} = \eta\lambda_k; \qquad 1 - \frac{\epsilon - g(\alpha^k; \lambda_k)}{\Omega(\alpha^k, \lambda_k)} \leq \eta \leq 1$$

- Increase ϵ_1 — Similar arguments

Dataset and BCFW Variants

- OCR dataset[14] with 6251 train and 626 test samples.
- $\epsilon = 0.1$
- 20 different values of λ equally spaced between $[10^{-4},\ 10^{3}]$

[14] **Taskar et al., *Max-margin Markov networks*, NIPS 2003.**

Dataset and BCFW Variants

- OCR dataset[14] with 6251 train and 626 test samples.
- $\epsilon = 0.1$
- 20 different values of λ equally spaced between $[10^{-4},\ 10^{3}]$
- BCFW variants
  - BCFW-HEU-G: Heuristic convergence with gap based sampling
  - BCFW-STD-G: Exact convergence with gap based sampling

[14] **Taskar et al.**, ***Max-margin Markov networks***, **NIPS 2003**.

Dataset and BCFW Variants

- OCR dataset[14] with 6251 train and 626 test samples.
- $\epsilon = 0.1$
- 20 different values of λ equally spaced between $[10^{-4},\ 10^{3}]$
- BCFW variants
  - BCFW-HEU-G: Heuristic convergence with gap based sampling
  - BCFW-STD-G: Exact convergence with gap based sampling
- RP-BCFW-HEU-G: Regularization Path with BCFW-HEU-G.

[14] **Taskar et al., *Max-margin Markov networks*, NIPS 2003.**

Effect of ϵ_1 for $\epsilon = 0.1$

Number of breakpoints in the regularization path

ϵ_1	**RP-BCFW-HEU-G**	**RP-BCFW-STD-G**
0.01	142	133
0.05	225	153
0.09	1060	349

Effect of ϵ_1 for $\epsilon = 0.1$

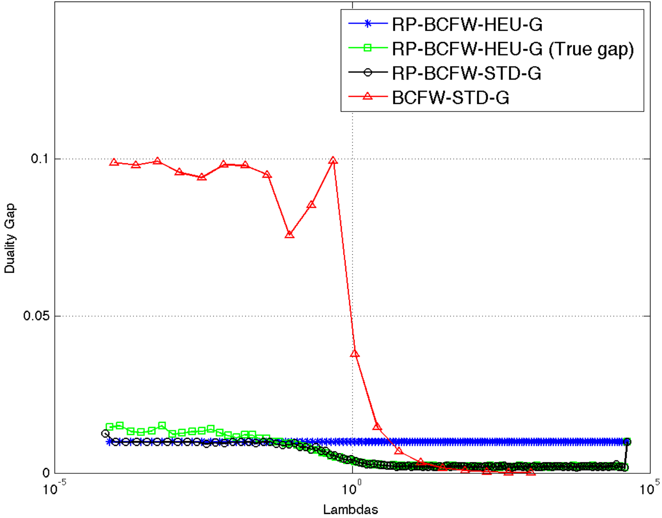
Number of breakpoints in the regularization path

ϵ_1	**RP-BCFW-HEU-G**	**RP-BCFW-STD-G**
0.01	142	133
0.05	225	153
0.09	1060	349

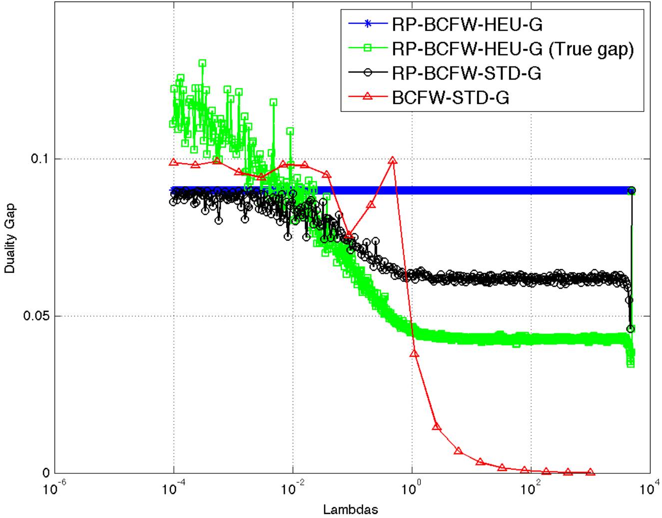
Number of passes through the data for optimization

ϵ_1	**RP-BCFW-HEU-G**	**RP-BCFW-STD-G**	**BCFW-STD-G**
0.01	2711.946	4405.881	1138.872
0.05	1301.869	2120.969	1138.872
0.09	**1076.005**	2100.304	1138.872

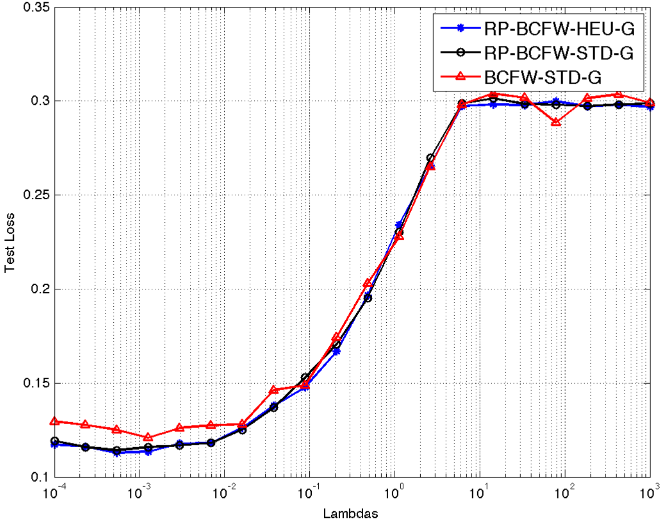
Duality gap for $\epsilon_1 = 0.01$



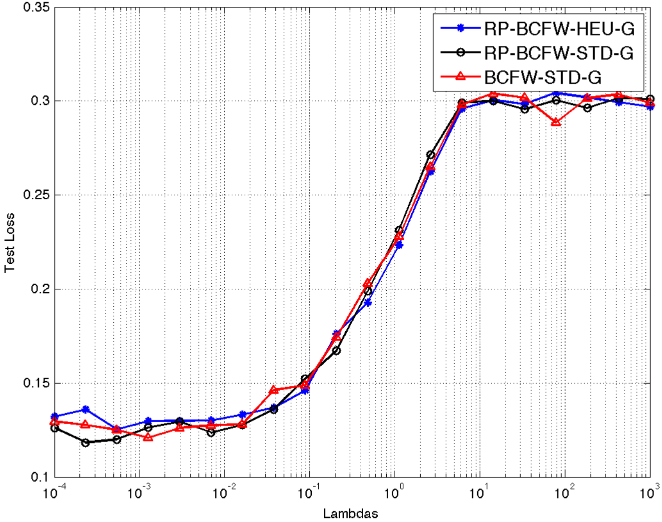
Duality gap for $\epsilon_1 = 0.09$



Test loss for $\epsilon_1 = 0.01$



Test loss for $\epsilon_1 = 0.09$



Presentation Outline

Possible future directions...

- High-Order
  - Parsimonious labeling for semantic labels
- SSVM
  - Latent HOAP-SVM
  - Discovering label dependence structure
  - Latent SSVM: Interaction between latent variables?
- Regularization path

$$\min_{\mathbf{w}} \quad \frac{\lambda}{2}\|\mathbf{w}\|^2 + L(\mathbf{x}, \mathbf{y}; \mathbf{w})$$

Presentation Outline

List of publications

1. *Discriminative parameter estimation for random walks segmentation*, In MICCAI 2013.
2. *Learning to Rank using High-Order Information*, In ECCV 2014.
3. *Parsimonious Labeling*, In ICCV 2015.
4. *Minding the Gaps for Block Frank-Wolfe Optimization of Structured SVM*, In ICML 2016.
5. *Rounding-based Combinatorial Algorithms for Metric Labeling*, In JMLR 2016.
6. *Deformable Registration through Learning of Context-Specific Metric Aggregation*, Under submission, ECCV 2016.
7. *Partial Linearization based Optimization for Multi-class SVM*, Under submission, ECCV 2016.