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

\[
E(x, y; w) = \sum_{i \in V} \theta(x_i, y_i; w) + \sum_{c \in C} \theta_c(x_c, y_c; w)
\]

\(\text{diversity}\)
Quick Overview

- **High-Order Inference**: Parsimonious Labeling
  \[
  E(x, y; w) = \sum_{i \in V} \theta(x_i, y_i; w) + \sum_{c \in C} \theta_c(x_c, y_c; w)
  \]

- **HOAP-SVM**: \(w\) very high-dimensional \(\rightarrow\) exhaustive search ??
  \[
  \min_w \frac{\lambda}{2} \|w\|^2 + L(x, y; w)
  \]

**HOAP-SVM**: Parsimonious Labeling

**AP-Based Regularization path for SSVM**: Efficiently explore the entire space of \(\lambda \in [0, \infty]\)
Quick Overview

- **High-Order Inference:** Parsimonious Labeling
  \[ E(x, y; w) = \sum_{i \in V} \theta(x_i, y_i; w) + \sum_{c \in C} \theta_c(x_c, y_c; w) \]
  
- **HOAP-SVM:** \( w \) very high-dimensional \( \rightarrow \) exhaustive search ??
  \[ \min_w \frac{\lambda}{2} \|w\|^2 + L(x, y; w) \]
  *AP-Based*

- **Regularization path for SSVM:** Efficiently explore the entire space of \( \lambda \in [0, \infty] \)
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
The Labeling Problem

Input

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

  \[
  E(y) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in \mathcal{C}} \theta_c(y_c). \tag{1}
  \]
The Labeling Problem

Input

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

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

Output

- **Labeling corresponding to the minimum energy**

$$y^* = \arg\min_{y} E(y). \quad (2)$$

- $H^N$ possible labelings

Puneet K. Dokania
Special case – Metric Labeling (Pairwise)

Pairwise Potentials $\theta(y_i, y_j) \rightarrow \text{Metric}$ 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)

$P^n$ Potts Model$^2$

\[ \theta_c(y_c) \propto \begin{cases} 
\gamma_k, & \text{if } y_i = l, \forall i \in c, \\
\gamma_{\text{max}}, & \text{otherwise} 
\end{cases} \]

Very efficient $\alpha$-expansion algorithm – Approximate solution

Special case – $P^n$ Potts Model\(^2\) (High-Order)

\[ \theta \propto \{ \gamma_k, \text{if } y_i = l, \forall i \in c, \gamma_{\text{max}}, \text{otherwise} \} \]

Special case – $P^n$ Potts Model\(^2\) (High-Order)

$P^n$ Potts Model

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

- Very efficient $\alpha$-expansion algorithm – Approximate solution

Parsimonious Labeling: Energy Function

\[ E(y) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in C} \theta_c(y_c). \]

- Unary potentials: Arbitrary

Where, \( \Gamma(y_c) \) is the set of unique labels.
Parsimonious Labeling: Energy Function

\[ E(y) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in C} \theta_c(y_c). \]

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

\[ \theta_c(y_c) \propto \delta(\Gamma(y_c)) \]

where, \( \Gamma(y_c) \) is the set of unique labels
Parsimonious Labeling: Energy Function

\[ E(y) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in C} \theta_c(y_c). \]

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

\[ \theta_c(y_c) \propto \delta(\Gamma(y_c)) \]

where, \( \Gamma(y_c) \) is the set of unique labels

\[ \delta_c(\{l_1, l_2, l_3\}) \]
Parsimonious Labeling: Energy Function

\[ E(y) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in C} \theta_c(y_c). \]

- Unary potentials: Arbitrary
- Clique potentials: Diversity

\[ \theta_c(y_c) \propto \delta(\Gamma(y_c)) \]

diversity

where, \( \Gamma(y_c) \) is the set of unique labels

- Energy function for Parsimonious Labeling

\[ E(y) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in C} w_c \delta(\Gamma(y_c)) \]

diversity
Diversity\(^3\): Metric over sets

\[ \theta_c(y_c) \propto \delta(\Gamma(y_c)) \]

\(^3\)Bryant and Tupper, Advances in Mathematics, 2012.
Diversity\(^3\): Metric over sets

\[ \theta_c(y_c) \propto \delta(\Gamma(y_c)) \]

- 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 \( \delta(\mathcal{L}_1) \leq \delta(\mathcal{L}_2) \) → Parsimony

\(^3\)Bryant and Tupper, Advances in Mathematics, 2012.
Diversity\(^3\): Metric over sets

\[ \theta_c(y_c) \propto \delta(\Gamma(y_c)) \]

- Metric over sets \( \delta : \bar{L} \rightarrow \mathbb{R}, \forall \bar{L} \subseteq L \), satisfying
  - Non Negativity
  - Triangular Inequality
  - Monotonicity: \( L_1 \subseteq L_2 \) implies \( \delta(L_1) \leq \delta(L_2) \) \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

\[ \theta_c(y_c) \propto \delta(\Gamma(y_c)) \]

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

Puneet K. Dokania
Special Case 1: Metric Labeling

- If cliques are of size 2 → diversity → metric

---

Special Case 1: Metric Labeling

- If cliques are of size 2 $\rightarrow$ diversity $\rightarrow$ metric
- Parsimonious Labeling $\rightarrow$ Metric Labeling\(^4\)


---

Special Case 2: $P^n$-Potts Model\(^5\)

- **Uniform Metric**

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

---

Special Case 2: $P^n$-Potts Model\textsuperscript{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

\textsuperscript{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**

<table>
<thead>
<tr>
<th>Labels</th>
<th>$l_1$</th>
<th>$l_2$</th>
<th>$l_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$l_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$l_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table: Uniform Metric**

Special Case 2: $P^n$-Potts Model\textsuperscript{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**

<table>
<thead>
<tr>
<th>Labels</th>
<th>$l_1$</th>
<th>$l_2$</th>
<th>$l_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$l_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$l_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table: Uniform Metric

\[
\theta_c(\{l_1, l_2, l_3\}) = \max(d(l_1, l_2), d(l_1, l_3), d(l_2, l_3)) = 1
\]

\[
\theta_c(y_c) \propto \begin{cases} 
0, & \text{if } y_i = l_k, \forall i \in c, \\
1, & \text{otherwise}
\end{cases}
\]

\textsuperscript{5}Kohli et al., P3 "& Beyond: Solving Energies with Higher Order Cliques, 2007.
So far ...

\[ E(y) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in C} w_c \delta(\Gamma(y_c)) \]

Diversity
Hierarchical $P^n$ Potts Model

- Given tree metric

![Diagram showing a tree structure with labels and distances]

Given the tree metric $d_t(l_1, l_2) = 14$, $d_t(l_1, l_3) = 4$, $d_t(l_1, l_1) = 0$. Diameter diversity at cluster $p$ is max $\{l_i, l_j\}$ where $d_t(l_i, l_j) = 14$. 

Puneet K. Dokania
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

\[
\begin{align*}
d^t(l_1, l_2) &= 14, \\
d^t(l_1, l_3) &= 4, \\
d^t(l_1, l_1) &= 0
\end{align*}
\]

- Hierarchical $P^n$ Potts Model $\rightarrow$ diameter diversity over tree metric

- Diameter diversity at cluster $p$ is $\max\{d^t(l_i, l_j)\} = 14$. 
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**

![Diagram of a hierarchical Potts model]

- $l_1$, $l_2$, $l_3$ at the root node
- $p_1$, $p_2$ at the second level
- $p_3$, $p_4$ at the third level
- Numbers represent weights or penalties
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**

![Diagram of a hierarchical Potts model with labels $l_1$, $l_2$, $l_3$, $P_1$, $P_2$, $P_3$, and $P_4$.](image)
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 $\rightarrow$ Get the induced metric

---

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

Puneet K. Dokania
Move Making for Parsimonious Labeling

- Given any general diversity $\rightarrow$ Get the induced metric
- Induced Metric $\rightarrow$ Mixture of tree metrics ($r$-hst)$^6$

---

$^6$Fakcharoenphol et al., In STOC 2003.
Move Making for Parsimonious Labeling

- Given any general diversity $\rightarrow$ Get the induced metric
- Induced Metric $\rightarrow$ Mixture of tree metrics $(r\text{-}\text{HST})^6$

Hierarchical $P^n$-Potts model over each tree metric $\rightarrow$ diameter diversity over each tree metric $(r\text{-}\text{HST})$

---

$^6$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 $\rightarrow$ Get the induced metric
- Induced Metric $\rightarrow$ Mixture of tree metrics ($r$-$\text{HST}$)

Hierarchical $P^n$-Potts model over each tree metric $\rightarrow$ diameter diversity over each tree metric ($r$-$\text{HST}$)

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 → Monotonic
  - Very fast optimization algorithm
  - No theoretical guarantees

- **SoSPD**\(^8\):
  - Clique potentials → Arbitrary → 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:**

\[ E(y) = \sum_{i \in V} \theta_i(y_i) + \sum_{c \in C} w_c \delta(\Gamma(y_c)) \]  

(Diversity)

- **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:**

\[ w_c = \exp\left(-\frac{\rho(y_c)}{\sigma^2}\right) \]

where, \( \rho(y_c) \) is the variance of intensities of pixels in clique \( y_c \).
Stereo Matching Results – Visually

(a) Ground Truth

(b) Our

(c) $\alpha$-Exp

(d) Co-oc
Stereo Matching Results – Energy and Time

(a) Our
\[ E = 1.4 \times 10^6, \ 773 \ sec \]

(b) Co-oc
\[ E = 2.1 \times 10^6, \ 306 \ sec \]
Image denoising and Inpainting Results – Visually

(a) Original

(b) Our

(c) α–Exp

(d) Co-oc
Image denoising and Inpainting Results – Energy and Time

(a) Our
\[ E = 1.2 \times 10^7, \ 1964 \ \text{sec} \]

(b) Co-occ
\[ E = 1.4 \times 10^7, \ 358 \ \text{sec} \]
Learning to Rank Using High-Order Information

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
Ranking?

Get the feature vector $\phi(x_i)$

Learn $w$

Sort using $s_i(w) = w^\top \phi(x_i)$

SVM $\rightarrow$ Optimizes accuracy

Accuracy $\neq$ Average Precision
Ranking?

Get the feature vector $\phi(x_i)$

Learn $w$

Sort using $s_i(w) = w^T \phi(x_i)$

SVM $\rightarrow$ Optimizes accuracy

Accuracy $\neq$ Average Precision
Learning to Rank Using High-Order Information

Ranking?

- Get the feature vector $\phi(x_i)$
- Learn $w$
- Sort using $s_i(w) = w^\top \phi(x_i)$
Ranking?

- Get the feature vector $\phi(x_i)$
- Learn $w$
- Sort using $s_i(w) = w^\top \phi(x_i)$
- SVM $\rightarrow$ Optimizes accuracy
- Accuracy $\neq$ Average Precision
AP-SVM\textsuperscript{10}: Problem Formulation

\textsuperscript{10} Yue et al., A support vector method for optimizing average precision, 2007
AP-SVM\textsuperscript{10} : Problem Formulation

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

\textsuperscript{10} Yue et al., A support vector method for optimizing average precision, 2007


**AP-SVM\(^\text{10}\): 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

\[
R_{ij} = \begin{cases} 
+1, & \text{if } i \text{ is better ranked than } j \\
-1, & \text{if } j \text{ is better ranked than } i 
\end{cases}
\]

- Define Joint Score:

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

Encodes Ranking

---

\(^{10}\) Yue et al., A support vector method for optimizing average precision, 2007.

Puneet K. Dokania
AP-SVM: Objective Function

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

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

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

- Loss augmented inference: $\tilde{R} = \arg\max_R \{S(x, \tilde{R}; w) + \Delta(R, R^*)\}$, greedy algorithm $O(|\mathcal{P}|\|\mathcal{N}|)$ by Yue et.al.
**AP-SVM: Joint Score**

- **Joint Score:**

  \[
  S(x, R; w) = \frac{1}{|P||N|} \sum_{i \in P} \sum_{j \in N} R_{ij}(s_i(w) - s_j(w))
  \]

  Encodes Ranking

- **Sample Score:**

  \[
  s_i(w) = w^T \phi(x_i)
  \]

  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)
  
  $$
  \Phi(x, y) = \left( \frac{\sum_i \Phi_1(x_i, y_i)}{\sum_{i,j} \Phi_2(x_i, y_i, x_j, y_j)} \right)
  $$

- $\Phi_1$ - first-order information
- $\Phi_2$ - high-order information

- Joint labeling: $y \in \{-1, +1\}^n$
Encoding High-Order Information

Define Joint Feature Map (encodes the structure)

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

- \( \Phi_1\) - first-order information
- \( \Phi_2\) - high-order information

Joint labeling: \( y \in \{-1, +1\}^n \)

Define Score \( S(x, y; w) = w^\top \Phi(x, y) \)
Joint Score: Closer look

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

Encodes High-Order Information

(6)
Joint Score: Closer look

$$w^\top \Phi(x, y) = \left( \begin{array}{c} w_1 \\ w_2 \end{array} \right)^\top \left( \begin{array}{c} \sum_i \Phi_1(x_i, y_i) \\ \sum_{i,j} \Phi_2(x_i, y_i, x_j, y_j) \end{array} \right)$$

$$= \sum_i w_1^\top \Phi_1(x_i, y_i) + \sum_{i,j} w_2^\top \Phi_2(x_i, y_i, x_j, y_j)$$

(6)

Encodes High-Order Information

* Single score for the entire dataset → Ranking?
Ranking Using Max-Marginals

We propose to use difference of max-marginals
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.

\[ m_i^+(\mathbf{w}) = \arg\max_{\mathbf{y}, y_i=+1} \mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y}) \]

- Dynamic Graph Cuts\(^1\) — Very Efficient

---
\(^1\) 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

\[
S(x, R; w) = \frac{1}{|P||N|} \sum_{i \in P} \sum_{j \in N} R_{ij}(s_i(w) - s_j(w))
\]

Encodes Ranking

Puneet K. Dokania
HOAP-SVM: Score

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

- **Joint Score for the given ranking**

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

  **Encodes Ranking**

- **Sample score** \(s_i\) **as difference of max-marginals**

  \[
  s_i(w) = m_i^+(w) - m_i^-(w)
  \]

  **Encodes High-Order Information**
HOAP-SVM: Objective Function

Objective Function

\[
\min_{\mathbf{w}, \xi} \quad \frac{\lambda}{2} \|\mathbf{w}\|^2 + \xi \\
\text{s.t.} \quad S(\mathbf{x}, \mathbf{R}^*; \mathbf{w}) \geq S(\mathbf{x}, \mathbf{R}; \mathbf{w}) + \Delta(\mathbf{R}, \mathbf{R}^*) - \xi, \quad \forall \mathbf{R}, \\
\mathbf{w}_2 \leq 0, \quad \xi \geq 0.
\]
**HOAP-SVM: Objective Function**

**Objective Function**

\[
\begin{align*}
\min_{w, \xi} & \quad \frac{\lambda}{2} \|w\|^2 + \xi \\
\text{s.t.} & \quad S(x, R^*; w) \geq S(x, R; w) + \Delta(R, R^*) - \xi, \quad \forall R, \\
& \quad w_2 \leq 0, \xi \geq 0.
\end{align*}
\]

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

\[
m^+_i(w) = \arg\max_{y, y_i=+1} w^\top \Phi(x, y)
\]
**HOAP-SVM: Objective Function**

- **Objective Function**

\[
\begin{align*}
\min_{w, \xi} \quad & \frac{\lambda}{2} \|w\|^2 + \xi \\
\text{s.t.} \quad & S(x, R^*; w) \geq S(x, R; w) + \Delta(R, R^*) - \xi, \quad \forall R, \\
& w_2 \leq 0, \xi \geq 0.
\end{align*}
\]

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

\[
m_i^+(w) = \arg\max_{y, y_i=+1} w^\top \Phi(x, y)
\]

- The objective function is a **difference of convex program**
Difference of convex functions can be optimized using CCCP algorithm

---

Yuille et al., The concave-convex procedure, 2003.
Difference of convex functions can be optimized using CCCP algorithm

\[ \text{Difference of convex functions} \]

\[ + \]

\[ \text{Yuille et al., The concave-convex procedure, 2003.} \]
Difference of convex functions can be optimized using CCCP algorithm

\[ \text{Yuille et al., The concave-convex procedure, 2003.} \]
Difference of convex functions can be optimized using CCCP algorithm

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Trainval</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>54.7/+4.2</td>
<td>48.82/+4.93</td>
</tr>
<tr>
<td>AP-SVM</td>
<td>56.2/+2.7</td>
<td>51.42/+2.33</td>
</tr>
<tr>
<td>HOAP-SVM</td>
<td>58.9</td>
<td>53.75</td>
</tr>
</tbody>
</table>
Visualization - Reading top 4

**SVM**

**AP-SVM**

**HOAP-SVM**
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

\[
\min_{\mathbf{w}, \xi} \frac{\lambda}{2} \| \mathbf{w} \|^2 + \frac{1}{n} \sum_{i=1}^{n} \xi_i
\]

\text{s.t. set of constraints}

- \( \lambda \rightarrow \) important for good generalization \( \rightarrow \) cross validate

- \( \lambda \in [0, \infty] \rightarrow \text{cross validation over subset} \rightarrow \text{poor generalization} \)
Regularization Path: What and Why

- Optimize SSVM objective function

\[
\min_{\mathbf{w}, \xi} \quad \frac{\lambda}{2} \| \mathbf{w} \|^2 + \frac{1}{n} \sum_{i=1}^{n} \xi_i
\]

s.t.  set of constraints

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

Algorithm
Regularization Path: What and Why

- Optimize SSVM objective function

\[
\min_{w, \xi} \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^{n} \xi_i 
\]

s.t. set of constraints

- \(\lambda\) → important for good generalization → cross validate
- \(\lambda \in [0, \infty]\) → cross validation over subset → poor generalization
- \(\epsilon\)-optimal regularization path algorithm

\[\lambda \in [0, \infty]\] → Algorithm
Regularization Path: What and Why

- Optimize SSVM objective function

\[
\min_{w, \xi} \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^{n} \xi_i
\]

s.t. \text{ set of constraints}

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

\[\lambda \in [0, \infty] \quad \text{Algorithm} \quad w \quad \text{dual gap} \leq \epsilon\]
Dual Objective and Duality Gap

- **SSVM dual objective function**

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

where, \(\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^{|\mathcal{Y}_1|} \times \cdots \mathbb{R}^{|\mathcal{Y}_n|} \).
Dual Objective and Duality Gap

- SSVM dual objective function

\[
\min_{\alpha} \quad f(\alpha) \rightarrow \text{smooth convex}
\]

\[s.t. \quad \sum_{y \in \mathcal{Y}_i} \alpha_i(y) = 1, \forall i \in [n],\]

\[\alpha_i(y) \geq 0, \forall i \in [n], \forall y \in \mathcal{Y}_i.\]

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

- Duality Gap

\[
g(\alpha; \lambda) = \frac{1}{n} \sum_i \left( \max_{y \in \mathcal{Y}_i} H_i(y; w) - \sum_{y \in \mathcal{Y}_i} \alpha_i(y) H_i(y; w) \right)
\]

where, \(H_i(y; w)\) is the hinge loss.
Key Idea: $\epsilon$-Optimal Regularization Path

\[ \lambda = \infty \]

\[ \lambda_k, w_k \rightarrow (\epsilon_1)_{opt} \]

\[ \epsilon_1 < \epsilon \]
Key Idea: $\epsilon$-Optimal Regularization Path

\[ \lambda = \infty \]

$\lambda_k, w_k \rightarrow (\epsilon_1)_{opt}$

$\epsilon_1 < \epsilon$

$w_k \rightarrow fixed$

$\lambda_k \downarrow$

duality gap $\uparrow$
Key Idea: $\epsilon$-Optimal Regularization Path

\[ \lambda = \infty \]

\[ \lambda_k, w_k \rightarrow \left( \epsilon_1 \right)_{opt} \]
\[ \epsilon_1 < \epsilon \]

\[ w_k \rightarrow fixed \]
\[ \lambda_k : \downarrow \]
\[ duality\ gap : \uparrow \]

\[ w_k \rightarrow \epsilon_{opt}, \forall \lambda \in [\lambda_{k+1}, \lambda_k] \]

\[ \lambda_{k+1} \]
Key Idea: $\epsilon$-Optimal Regularization Path

\[ \lambda = \infty \]

$\lambda_k, w_k \rightarrow (\epsilon_1)_{opt}$

$\epsilon_1 < \epsilon$

$w_k \rightarrow fixed$

$\lambda_k \downarrow$

\textit{duality gap} \uparrow

$w_k \rightarrow \epsilon_{opt}, \forall \lambda \in [\lambda_{k+1}, \lambda_k]$

Optimize: $w_{k+1} \rightarrow (\epsilon_1)_{opt}$
Key Idea: $\epsilon$-Optimal Regularization Path

\[
\lambda = \infty \\
\lambda_k, w_k \rightarrow (\epsilon_1)_{opt} \\
\epsilon_1 < \epsilon \\
w_k \rightarrow fixed \\
\lambda_k \downarrow \\
duality\ gap \uparrow \\
w_k \rightarrow \epsilon_{opt}, \forall \lambda \in [\lambda_{k+1}, \lambda_k] \\
\lambda_{k+1} \\
\text{optimize: } w_{k+1} \rightarrow (\epsilon_1)_{opt} \\
repeat
\]
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 \( \tilde{Y}_i = \arg\max_{y \in Y_i} \Delta(y, y_i) \) be the loss-maximizer and \( \tilde{y}_i \in \tilde{Y}_i, \forall i \).
**Challenge 1: How do we start?**

1. Let \( \tilde{Y}_i = \arg\max_{y \in Y_i} \Delta(y, y_i) \) be the loss-maximizer and \( \tilde{y}_i \in \tilde{Y}_i, \forall i \).
2. Let \( \tilde{\Psi} = \frac{1}{n} \sum_i \psi_i(\tilde{y}_i) \), where \( \psi_i(y) = \Phi(x_i, y_i) - \Phi(x_i, y) \).

\[
\lambda = \infty \\
w_k \rightarrow (\epsilon_1)_{opt}, \forall \lambda \geq \lambda_k \\
\lambda_k, w_k \rightarrow (\epsilon_1)_{opt}
\]
Challenge 1: How do we start?

- Let $\tilde{y}_i = \arg\max_{y \in \mathcal{Y}_i} \Delta(y, y_i)$ be the loss-maximizer and $\tilde{y}_i \in \tilde{\mathcal{Y}}_i$, $\forall i$.
- Let $\tilde{\Psi} = \frac{1}{n} \sum_i \Psi_i(\tilde{y}_i)$, where $\Psi_i(y) = \Phi(x_i, y_i) - \Phi(x_i, y)$.
- Then, $w_k = \frac{\tilde{\Psi}}{\lambda}$ is guaranteed to be $\epsilon_1$ optimal for any $\lambda$ satisfying the condition:

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

Inference (9)
Challenge 2: How to find the breakpoints?

Let $\lambda_{k+1} + 1 = \eta \lambda_k$, $0 \leq \eta \leq 1$. $w_k \rightarrow \epsilon_{opt}$, for all $\lambda_{k+1}$ obtained using $\eta$ satisfying the condition:

$$1 - \epsilon - g(\alpha_k; \lambda_k) \leq \eta \leq 1 \quad (10)$$

where, $\Omega(\alpha_k, \lambda_k) := \ell \alpha_k - \lambda_k w_k^\top w_k$. 

\[ \lambda_k, w_k \rightarrow (\epsilon_1)_{opt} \]

$\epsilon_1 < \epsilon$

$w_k \rightarrow \epsilon_{opt}$, $\forall \lambda \in [\lambda_{k+1}, \lambda_k]$

$duality\ gap \uparrow$

$\lambda_k \downarrow$

$w_k \rightarrow fixed$

$\lambda_{k+1}$
Challenge 2: How to find the breakpoints?

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

\[ \lambda_k, w_k \rightarrow (\epsilon_1)_{opt} \]
\[ \epsilon_1 < \epsilon \]
\[ w_k \rightarrow \epsilon_{opt}, \forall \lambda \in [\lambda_{k+1}, \lambda_k] \]

\[ \text{duality gap} \uparrow \]

\[ \text{fixed} \]

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

\[ \lambda_k \downarrow \]
**Challenge 2: How to find the breakpoints?**

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

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

where, $\Omega(\alpha^k, \lambda_k) := \ell(\alpha^k) - \lambda^k w_k^\top w_k$
Challenge 2: Proof Sketch

Keeping $w_k$ constant – from $k.sc/k.sc/t.sc$ condition $w_k = \frac{1}{n} \sum_{i \in [n]} y_i \alpha_k(y_i) \lambda_k \Psi(x_i, y_i)$.

Therefore, using $\alpha_{k+1}(y_i) \lambda_{k+1} = \alpha_k(y_i) \lambda_k$, $\forall y_i \neq y_i$;

$\sum_{y \in Y} \alpha_i(y) = 1$, $\forall i \in [n]$;

$\lambda_{k+1} = \eta \lambda_k$.

New duality gap $g(\alpha_{k+1}; \lambda_{k+1}) = g(\alpha_k; \lambda_k) \leq \epsilon$. 

Puneet K. Dokania
Challenge 2: Proof Sketch

- Keeping $w_k$ constant – from KKT condition

$$w_k = \frac{1}{n} \sum_{i \in [n], y \in \mathcal{Y}_i} \frac{\alpha_i^k(y)}{\lambda_k} \psi(x_i, y).$$
Challenge 2: Proof Sketch

- Keeping $w_k$ constant – from KKT condition

$$w_k = \frac{1}{n} \sum_{i \in [n], y \in \mathcal{Y}_i} \frac{\alpha_i^k(y)}{\lambda_k} \psi(x_i, y).$$

- Therefore, using

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

- New duality gap

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

$$\leq \epsilon$$
Challenge 3: How to optimize efficiently?

Notice that, $\mathbf{w}_k$ is already $\epsilon$-optimal at $\lambda_{k+1}$

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\textsuperscript{13} for the optimization.

Lagrange duality gap is the by product

\textsuperscript{13} Lacoste-Julien et al., In ICML 2013.
Effects of $\epsilon_1$

$\epsilon_1$ decreases — big jumps — number of breakpoints decreases (see below)

$\lambda_{k+1} = \eta \lambda_k$ ;

$\epsilon - g(\alpha_k;\lambda_k) \Omega(\alpha_k,\lambda_k) \leq \eta \leq 1$
Effects of $\epsilon_1$

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

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

- **Increase $\epsilon_1$ — Similar arguments**
OCR dataset\textsuperscript{14} with 6251 train and 626 test samples.
\( \epsilon = 0.1 \)
20 different values of \( \lambda \) equally spaced between \([10^{-4}, 10^3]\)

\textsuperscript{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 \(\lambda\) 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 with 6251 train and 626 test samples.
- $\epsilon = 0.1$
- 20 different values of $\lambda$ 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.

---

Effect of $\epsilon_1$ for $\epsilon = 0.1$

Number of breakpoints in the regularization path

<table>
<thead>
<tr>
<th>$\epsilon_1$</th>
<th>RP-BCFW-HEU-G</th>
<th>RP-BCFW-STD-G</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>142</td>
<td>133</td>
</tr>
<tr>
<td>0.05</td>
<td>225</td>
<td>153</td>
</tr>
<tr>
<td>0.09</td>
<td>1060</td>
<td>349</td>
</tr>
</tbody>
</table>
Effect of $\epsilon_1$ for $\epsilon = 0.1$

Number of breakpoints in the regularization path

<table>
<thead>
<tr>
<th>$\epsilon_1$</th>
<th>RP-BCFW-HEU-G</th>
<th>RP-BCFW-STD-G</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>142</td>
<td>133</td>
</tr>
<tr>
<td>0.05</td>
<td>225</td>
<td>153</td>
</tr>
<tr>
<td>0.09</td>
<td>1060</td>
<td>349</td>
</tr>
</tbody>
</table>

Number of passes through the data for optimization

<table>
<thead>
<tr>
<th>$\epsilon_1$</th>
<th>RP-BCFW-HEU-G</th>
<th>RP-BCFW-STD-G</th>
<th>BCFW-STD-G</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>2711.946</td>
<td>4405.881</td>
<td>1138.872</td>
</tr>
<tr>
<td>0.05</td>
<td>1301.869</td>
<td>2120.969</td>
<td>1138.872</td>
</tr>
<tr>
<td>0.09</td>
<td><strong>1076.005</strong></td>
<td>2100.304</td>
<td>1138.872</td>
</tr>
</tbody>
</table>
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

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

\[
\min_{\mathbf{w}} \quad \frac{\lambda}{2} \|\mathbf{w}\|^2 + L(x, y; \mathbf{w})
\]
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
List of publications

Thank you