Exploiting Problem Structure for Efficient Discrete Optimization

Collaborators

Karteek Alahari  Dhruv Batra  Vladimir Kolmogorov
M Pawan Kumar  Lubor Ladicky  Sebastian Nowozin
Carsten Rother  Danny Tarlow  Philip Torr
and many others ....

Pushmeet Kohli

Microsoft Research
Image Segmentation

$E(X)$

$E: \{0, 1\}^n \rightarrow R$

$0 (Fg) \quad 1 (Bg)$

Image (D)
Image Segmentation

\[ E(X) = \sum c_i x_i \]

Unary Cost (\(c_i\))

Dark (negative)  Bright (positive)
Image Segmentation

$$E(X) = \sum c_i x_i$$

$$x^* = \arg \min E(x)$$

Unary Cost ($c_i$)

Dark (negative)  Bright (positive)
Image Segmentation

\[ E(X) = \sum c_i x_i \]
Image Segmentation

\[ E(X) = \sum c_i x_i + \sum d_{ij} |x_i - x_j| \]

Discontinuity Cost \((d_{ij})\)

Ising Prior

\[ \sum c_i x_i \]

\[ \sum c_i x_i \]

[Boykov and Jolly '01] [Blake et al. '04] [Rother, Kolmogorov and Blake '04]
**Image Segmentation**

$$E(X) = \sum c_i x_i + \sum d_{ij} |x_i - x_j|$$

How to minimize $E(x)$?

$$x^* = \text{arg min} E(x)$$
Energy Minimization Problems

n = Number of Variables

- Segmentation Energy

Space of Problems
Submodular Functions: Definition

Pseudo-boolean function \( f: \{0,1\}^n \rightarrow \mathbb{R} \) is submodular if

\[
f(A) + f(B) \geq f(A \lor B) + f(A \land B)
\]

(OR) \hspace{1cm} (AND)

for all \( A, B \in \{0,1\}^n \)
Submodular Functions: Definition

Pseudo-boolean function $f: \{0,1\}^n \rightarrow \mathbb{R}$ is submodular if

$$f(A) + f(B) \geq f(A \lor B) + f(A \land B)$$

for all $A, B \in \{0,1\}^n$

(OR) \hspace{1cm} (AND)

**Example:** $n = 2$, $A = [1,0]$, $B = [0,1]$

$$f([1,0]) + f([0,1]) \geq f([1,1]) + f([0,0])$$

$$f([1,0]) - f([0,0]) \geq f([1,1]) - f([0,1])$$

The benefit of adding an element to a smaller set is larger than the benefit of adding it to a larger one
**Submodular Functions: Definition**

Pseudo-boolean function \( f: \{0,1\}^n \rightarrow \mathbb{R} \) is submodular if

\[
f(A) + f(B) \geq f(A \lor B) + f(A \land B) \quad \text{for all } A, B \in \{0,1\}^n
\]

**Example:** \( n = 2, \ A = [1,0], \ B = [0,1] \)

\[
f([1,0]) + f([0,1]) \geq f([1,1]) + f([0,0])
\]

**Property:** Sum of submodular functions is submodular

**Binary Image Segmentation Energy is submodular**

\[
E(x) = \sum_i c_i x_i + \sum_{i,j} d_{ij} |x_i - x_j|
\]
Submodular Functions

- **Discrete Analogues of Concave Functions**
  [Lovasz, ’83]

- **Widely applied in Operations Research**

- **Applications in Machine Learning**
  - MAP Inference in Markov Random Fields
  - Clustering [Narasimhan, Jojic, & Bilmes, NIPS 2005]
  - Structure Learning [Narasimhan & Bilmes, NIPS 2006]

- **Maximizing the spread of influence through a social network** [Kempe, Kleinberg & Tardos, KDD 2003]
Minimizing Submodular Functions

- Polynomial time algorithms
  - Current Best: $O(n^5 Q + n^6)$  [Q is function evaluation time] [Orlin ‘07]

- Symmetric functions: $E(x) = E(1-x)$
  - Can be minimized in $O(n^3)$

- Minimizing Pairwise submodular functions
  - Can be transformed to st-mincut/max-flow [Hammer, 1965]
  - Very low empirical running time ~ $O(n)$

\[ E(X) = \sum_{i} f_i(x_i) + \sum_{ij} g_{ij}(x_i, x_j) \]
So how does this work?

\[ E(x) \]

Pairwise Submodular

[Hammer, 1965] [Kolmogorov and Zabih, 2002]
Flow and Reparametrization

\[ E(a_1, a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1a_2 \]
Flow and Reparametrization

\[ E(a_1, a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1a_2 \]

Sink (1)  
Source (0)

\[ 2a_1 + 5\bar{a}_1 = 2(a_1 + \bar{a}_1) + 3\bar{a}_1 = 2 + 3\bar{a}_1 \]
Flow and Reparametrization

\[ E(a_1, a_2) = 2 + 3\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1a_2 \]

\[ 2a_1 + 5\bar{a}_1 = 2(a_1 + \bar{a}_1) + 3\bar{a}_1 = 2 + 3\bar{a}_1 \]
Flow and Reparametrization

\[ E(a_1, a_2) = 2 + 3\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1a_2 \]

\[ 9a_2 + 4\bar{a}_2 = 4(a_2 + \bar{a}_2) + 5\bar{a}_2 \]

\[ = 4 + 5\bar{a}_2 \]
Flow and Reparametrization

\[ E(a_1, a_2) = 2 + 3\bar{a}_1 + 5a_2 + 4 + 2a_1\bar{a}_2 + \bar{a}_1a_2 \]

9\(a_2 + 4\bar{a}_2\)

= 4(\(a_2 + \bar{a}_2\)) + 5\(\bar{a}_2\)

= 4 + 5\(\bar{a}_2\)
Flow and Reparametrization

\[ E(a_1, a_2) = 6 + 3\bar{a}_1 + 5a_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2 \]
Flow and Reparametrization

\[ E(a_1, a_2) = 6 + 3\bar{a}_1 + 5a_2 + 2a_1\bar{a}_2 + \bar{a}_1a_2 \]
Flow and Reparametrization

$E(a_1,a_2) = 6 + 3\tilde{a}_1 + 5a_2 + 2a_1\tilde{a}_2 + \tilde{a}_1a_2$

$$3\tilde{a}_1 + 5a_2 + 2a_1\tilde{a}_2$$
$$= 2(\tilde{a}_1 + a_2 + a_1\tilde{a}_2) + \tilde{a}_1 + 3a_2$$
$$= 2(1 + \tilde{a}_1a_2) + \tilde{a}_1 + 3a_2$$

$F_1 = \tilde{a}_1 + a_2 + a_1\tilde{a}_2$
$F_2 = 1 + \tilde{a}_1a_2$

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$F_1$</th>
<th>$F_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Flow and Reparametrization

\[ E(a_1, a_2) = 8 + \bar{a}_1 + 3a_2 + 3\bar{a}_1a_2 \]

F1 = \bar{a}_1 + a_2 + a_1\bar{a}_2

F2 = 1 + a_1\bar{a}_2

<table>
<thead>
<tr>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( F1 )</th>
<th>( F2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Flow and Reparametrization

\[ E(a_1, a_2) = 8 + \bar{a}_1 + 3a_2 + 3\bar{a}_1a_2 \]

No more augmenting paths possible.
$E(a_1, a_2) = 8 + \tilde{a}_1 + 3a_2 + 3\tilde{a}_1a_2$

Residual Graph (positive coefficients)

Total Flow

bound on the energy of the optimal solution

Tight Bound --> Inference of the optimal solution becomes trivial
Flow and Reparametrization

\[ E(a_1, a_2) = 8 + \bar{a}_1 + 3a_2 + 3\bar{a}_1a_2 \]

Residual Graph (positive coefficients)

Total Flow bound on the energy of the optimal solution

\[ a_1 = 1 \quad a_2 = 0 \]

\[ E(1, 0) = 8 \]

\[ \text{st-mincut cost} = 8 \]

Tight Bound --> Inference of the optimal solution becomes trivial
.. So what are the challenges?
Need Richer Models
3,600,000,000 Pixels
Created from about 800 8 MegaPixel Images

[Kopf et al. (MSR Redmond) SIGGRAPH 2007]
Speed and Scalability

[Kopf et al. (MSR Redmond) SIGGRAPH 2007]
Speed and Scalability

- Processing Videos
  - 1 minute video of 1M pixel resolution
    - 3.6 B pixels

- 3D reconstruction
  - $500 \times 500 \times 500 = 0.125B$ voxels
Part I
Exploiting Problem Structure for Efficiency
Image Segmentation in Videos
Image Segmentation in Videos

Image

Flow

Segmentation

[Kohli & Torr, ICCV05 PAMI07]
Image Segmentation in Videos

Can we do better?

First Frame

Second Frame

[Kohli & Torr, ICCV05 PAMI07]
Dynamic Energy Minimization

Frame 1: \( E_A \)

Frame 2: \( E_B \)

minimize differences between \( A \) and \( B \)

\( S_A \)

Reuse Computation

\( E_B^* \)

Simpler

fast

3-100000

Reparametrization

\( S_B \)

[Kohli & Torr, ICCV05 PAMI07]  [Komodakis & Paragios, CVPR07]
Hybrid Algorithms

Original Problem (Large) → Approximation algorithm (Slow) → Approximate Solution

[Alahari Kohli & Torr CVPR ‘08]
Hybrid Algorithms
Hybrid Algorithms

Original Problem (Large) → Reduced Problem

Fast partially optimal algorithm
[Kovtun '03] [Kohli et al. '09]

Reduced Problem → Solved Problem (Global Optima)

Approximation algorithm (Slow)

Approximation algorithm (Fast)

Approximate Solution

Approximate Solution

[Alahari Kohli & Torr CVPR '08]
Hybrid Algorithms

Fast partially optimal algorithm
[Kovtun '03] [Kohli et al. '09]

Tree Reweighted Message Passing
(9.89 sec)

3-100 Times Speed up

Reduced Problem
Solved Problem (Global Optima)
Total Time (0.30 sec)

[Alahari Kohli & Torr CVPR '08]
Making Inference algorithms adapt to the problem
General Labeling Problem

\[ \min_y E(y) = \sum_i f_i(y_i) + \sum_{i,j} g_{ij}(y_i, y_j) \]

\[ y \in \text{Labels } L = \{ l_1, l_2, \ldots, l_k \} \]
General Labeling Problem

\[
\min_y \ E(y) = \sum_i f_i(y_i) + \sum_{i,j} g_{ij}(y_i,y_j)
\]

\[y \in \text{Labels } L = \{l_1, l_2, \ldots, l_k\}\]

NP–Hard!
General Labeling Problem

$$\text{Min}_y \ E(y) = \sum_{i} f_i(y_i) + \sum_{i,j} g_{ij}(y_i, y_j)$$

$$y \in \text{Labels } L = \{l_1, l_2, \ldots, l_k\}$$

- LP Relaxations
  
  [Schlesinger ‘76, Koster ‘98, Wainwright ‘05, Sontag ‘08]
MAP Inference as an IP

$$\min \left[ \sum_{a \in L} V_p(a)x_{p,a} + \sum_{a,b \in L} V_{pq}(a,b)x_{pq,ab} \right]$$

Integer Program
LP Relaxation of MAP Inference

\[
\min \left[ \sum_{a \in L} V_p(a) x_{p,a} + \sum_{a, b \in L} V_{pq}(a, b) x_{pq,ab} \right]
\]

s.t. \[
\sum_{a \in L} x_{p,a} = 1
\]
\[
\sum_{a \in L} x_{pq,ab} = x_{q,b}
\]
\[
\sum_{b \in L} x_{pq,ab} = x_{p,a}
\]
\[
x_{p,a}, x_{pq,ab} \in \{0, 1\}
\]
\[
x_{p,a} \geq 0, \quad x_{pq,ab} \geq 0
\]

Linear Program
Solving the LP relaxation

Primal-Dual Methods

\[ \min \ c^T x \]
\[ \text{s.t. } A x = b, \ x \in \mathbb{N} \]  

(NP-hard problem)

Relaxation approach:

primal LP: \[ \min \ c^T x \]
\[ \text{s.t. } A x = b, \ x \geq 0 \]  
dual LP: \[ \max \ b^T y \]
\[ \text{s.t. } A^T y \leq c \]  

dual cost of solution \( y \)

\( b^T y \)

cost of optimal integral solution \( x^* \)

\( c^T x^* \)

primal cost of solution \( x \)

\( c^T x \)
Solving the LP Relaxations

Primal LP

Dual LP

Objective

Computation
Solving the LP Relaxations

Primal LP

Dual LP

Objective

Computation
Solving the LP relaxation

- **Message Passing**
  [Pearl, ‘88] [Kolmogorov, ‘06] [Ravikumar et al. ‘10]
  - Dual variables: Messages

- **Graph cut based move algorithms (Expansion)**
  [Boykov ‘01, Komodakis ‘05]
  - Dual variables: Flow
Can we make inference algorithms adapt to the `current` problem?
Local Primal-Dual Gaps

- Can be seen as distributed Primal-Dual Gap
- Primal contribution minus Dual Contribution of a set of variables
- Generalization of Complimentary Slackness Conditions
- Used to isolate which dual variables have the most potential to increase the dual objective
Focused Inference

Energy-Aware Message-Passing
ICML 2011
Message Passing

Standard Message Scheduling
Intelligent Message Scheduling

TBCA with Static Schedule:
630 messages needed

TBCA with Dynamic Schedule:
276 messages needed
Intelligent Message Scheduling

(a) Iteration 1  (b) Iteration 100  (c) Iteration 100
Intelligent Message Scheduling

![Graphs showing energy vs. number of distance transforms for different algorithms.](image)
Focused Inference

Label Re-ordering in $\alpha$-Expansion
CVPR 2011
Graph Cut based Move Making Algorithms

Loop over $\alpha$

Current Solution

$\alpha$-Expansion

Current Label

2-Label Problem + GC

Label ($\alpha$)

New Solution
Graph Cut based Move Making Algorithms

Status: Expand Sky in Tree

Ascent on all parameters corresponding to one label

How to order labels for expansion?
Dynamic Re-ordering of Labels

For each label, compute a score that is high when the potential decrease in the primal dual gap is high.
Dynamic Re-ordering of Labels

Local Primal-Dual Gap
\[
lpdg(\alpha, i) = h_i(x^P_i) - h_i(\alpha)
\]

Proposed Label Scores

- LPDG-crisp:
  \[
  \omega_1(\alpha) = \sum_{i \in \mathcal{V}} \mathbb{I}_{[0, \infty)} (lpdg(\alpha, i))
  \]

- LPDG-deficit:
  \[
  \omega_2(\alpha) = \sum_{i \in \mathcal{V}} lpdg(\alpha, i) \cdot \mathbb{I}_{[0, \infty)} (lpdg(\alpha, i))
  \]

- LPDG-tradeoff:
  \[
  \omega_3(\alpha) = \sum_{i \in \mathcal{V}} |lpdg(\alpha, i)|
  \]

\[
\mathbb{I}_S(y) = \begin{cases} 
  1 & y \in S \\
  0 & \text{else}
\end{cases}
\]
For each label, compute a score that is high when the potential decrease in the primal dual gap is high.
Dynamic Re-ordering of Labels

<table>
<thead>
<tr>
<th>Move Number</th>
<th>Classical Expansions</th>
<th>Our Guided Expansions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image 1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Airplane</td>
<td>Car</td>
</tr>
<tr>
<td>2</td>
<td>Bicycle</td>
<td>Person</td>
</tr>
<tr>
<td>3</td>
<td>Bird</td>
<td>Motorbike</td>
</tr>
<tr>
<td>4</td>
<td>Boat</td>
<td>Train</td>
</tr>
<tr>
<td>5</td>
<td>Bottle</td>
<td>Airplane</td>
</tr>
<tr>
<td><strong>Image 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Airplane</td>
<td>Sheep</td>
</tr>
<tr>
<td>2</td>
<td>Bicycle</td>
<td>Dog</td>
</tr>
<tr>
<td>3</td>
<td>Bird</td>
<td>Bird</td>
</tr>
<tr>
<td>4</td>
<td>Boat</td>
<td>Cow</td>
</tr>
<tr>
<td>5</td>
<td>Bottle</td>
<td>Cat</td>
</tr>
</tbody>
</table>
Experiments

(a) Cones: Image.
(b) Cones: Ground Truth Disparity.
(c) Cones: Energy vs. Iterations.
(d) Cones: Energy vs. Time.
(e) Teddy: Energy vs. Iterations.
(f) Teddy: Energy vs. Time.
(g) Venus: Energy vs. Iterations.
(h) Venus: Energy vs. Time.
Focused Inference

- Energy-Aware Message-Passing
  ICML 2011
- Label Re-ordering in $\alpha$-Expansion
  CVPR 2011
- Tightening LP Relaxations
  AISTATS 2011
Focused Inference

Tightening LP Relaxations
AISTATS 2011
Hierarchy of LPs

Increasingly Complex Sub-problems

Edge-Consistent LP

Triplet-Clique Consistent LP

Use LPDG to
-- Score Clusters / Constraints
-- Add high scoring ones
Tightening Relaxation

<table>
<thead>
<tr>
<th>Primal LP</th>
<th>Dual LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \min_{\mu} \bar{\theta} \cdot \mu )</td>
<td>( \max_{h,y} 1 \cdot h )</td>
</tr>
<tr>
<td>( \sum_{x_A} \mu_A(x_A) = 1 )</td>
<td>( h_A \in \mathbb{R} )</td>
</tr>
<tr>
<td>( \sum_{x_{A \setminus B}} \mu_A(x_A) = \mu_B(x_B) )</td>
<td>( y_{A \rightarrow B}(x_B) \in \mathcal{R} )</td>
</tr>
<tr>
<td>( \mu_A(x_A) \geq 0 )</td>
<td>( h_A \leq \tilde{\theta}_A(x_A) )</td>
</tr>
<tr>
<td>( \forall (A, B, x_B) \in J )</td>
<td></td>
</tr>
</tbody>
</table>

**Local Primal-Dual Gap**

\[ \text{lpdg}(A) = \tilde{\theta}_A(x_A^p) - \min_{x_A} \tilde{\theta}_A(x_A) \]
Experiments

![Graph 1: Lower Bound vs. Time](image1.png)

- Sontag08-alone
- LPDGcrisp+Sontag08
- LPDG-alone
- LPDG+Sontag08
- Random

![Graph 2: Primal-Dual Gap vs. Time](image2.png)

- Sontag08-alone
- LPDGcrisp+Sontag08
- LPDG-alone
- LPDG+Sontag08
- Random

(a) Original Image. (b) Blurry Noisy Image. (c) Schlesinger’s LP solution. (d) MAP from Triplet LP.
Part II
Exploiting Problem Structure for Handling Richer Models
Image Segmentation

\[ E(X) = \sum_i c_i x_i + \sum_{i,j} d_{ij} |x_i - x_j| \]
P^n Potts Model for label consistency

Patch Dictionary (Tree)

\[ h(X_p) = \begin{cases} 
C_1 & \text{if } x_i = 0, i \in p \\
C_{\text{max}} & \text{otherwise} 
\end{cases} \]

\[ C_{\text{max}} \geq C_1 \]

[Kohli et al. '07]
Image Segmentation

\[ E(X) = \sum_i c_i x_i + \sum_{i,j} d_{ij} |x_i - x_j| + \sum_p h_p(X_p) \]

\[ h(X_p) = \begin{cases} 
  C_1 & \text{if } x_i = 0, i \in p \\
  C_{\text{max}} & \text{otherwise}
\end{cases} \]

\[ C_{\text{max}} \geq C_1 \]

[Kohli et al. '07]
Image Segmentation

\[ E(X) = \sum_i c_i x_i + \sum_{i,j} d_{ij} |x_i - x_j| + \sum_p h_p (X_p) \]

- \( n \) = number of pixels
- \( E: \{0,1\}^n \rightarrow \mathbb{R} \)
- \( 0 \rightarrow \text{fg}, \ 1 \rightarrow \text{bg} \)

[Image Pairwise Segmentation]
[Final Segmentation]

[Kohli et al. '07]
Higher Order Potentials for Object Segmentation
Kohli Ladicky Torr [CVPR 2008]

Unary Potentials
[Shotton et al. ECCV 2006]

Colour, Location & Texture

Higher Order Potentials
(Defined using multiple Segmentations)

Higher Order Energy

Pairwise Smoothness Potentials

h(X_p) = \begin{cases} 
0 & \text{if } x_i = L, I \in p \\
C & \text{otherwise} 
\end{cases}
Higher Order Potentials for Object Segmentation

Kohli Ladicky Torr [CVPR 2008]

 Unary Potentials
[Shotton et al. ECCV 2006]

 Colour, Location & Texture

 Higher Order Energy

 Sky
 Tree
 Building
 Grass

 Pairwise Smoothness Potentials

 Higher Order Potentials
(Defined using multiple Segmentations)
Higher Order Potentials for Object Segmentation

Kohli Ladicky Torr [CVPR 2008]

Unary Potentials
[Shotton et al. ECCV 2006]

Colour, Location & Texture

Higher Order Energy

Sky
Tree
Building
Grass

+ Pairwise Smoothness Potentials

Energy Minimization

Segmentation Solution

Higher Order Potentials
(Defined using multiple Segmentations)
Unary Classifiers  
Pairwise CRF  
$P^n$ Potts  
Robust $P^n$ Potts  

Clustering 1  
Clustering 2
More Results

Image (MSRC-21) | Pairwise CRF | Higher order CRF | Ground Truth
--- | --- | --- | ---
![Image](image1.png) | ![Pairwise CRF](pairwise_crf1.png) | ![Higher order CRF](higher_order_crf1.png) | ![Ground Truth](ground_truth1.png)

[Runner-Up, PASCAL VOC 2008]
Overcoming short-boundary bias

\[ E(X) = \sum c_i x_i + \sum d_{ij} |x_i - x_j| \]
Overcoming short-boundary bias

\[ E(X) = \sum c_i x_i + \sum d_{ij} |x_i - x_j| \]

Encourages short boundaries

Image

Segmentation
Overcoming short-boundary bias

\[ E(X) = \sum c_i x_i + \sum d_{ij} |x_i - x_j| \]

Encourages short boundaries

Penalize types of boundaries not the actual number of boundaries!

[Jegelka and Bilmes, CVPR 2011]
Overcoming short-boundary bias

\[ E(X) = \sum c_i x_i + \sum d_{ij} |x_i - x_j| + \sum h_g(X_p) \quad \text{g in G} \]

- Divide edges into different types
- Incorporate a higher order consistency potential over the edges

\[ h_g(X_p) = F \left( \sum_{ij \text{ in } g} x_i - x_j \right) \quad \text{otherwise} \]

[Jegelka and Bilmes, CVPR 2011]
Identified transformable families of higher order function s.t.

1. Constant or polynomial number of auxiliary variables \((a)\) added
2. All pairwise functions \((g)\) are submodular
Minimizing Higher Order Energy Functions

Example: \( H(X) = F(\sum x_i) \)

\[ \text{concave} \]

[Kohli et al. ‘08]
Higher order to Quadratic

- Simple Example using Auxiliary variables

\[ f(x) = \begin{cases} 
0 & \text{if all } x_i = 0 \\
C_1 & \text{otherwise}
\end{cases} \]

\[ x \in L = \{0,1\}^n \]

\[
\min_x f(x) = \min_{x,a \in \{0,1\}} C_1a + C_1 \sum \bar{a} x_i
\]

Higher Order Submodular Function

Quadratic Submodular Function

\[
\sum x_i < 1 \quad \rightarrow \quad a=0 \ (\bar{a}=1) \quad \rightarrow \quad f(x) = 0
\]

\[
\sum x_i \geq 1 \quad \rightarrow \quad a=1 \ (\bar{a}=0) \quad \rightarrow \quad f(x) = C_1
\]
Higher order to Quadratic

\[ \min_{x} f(x) = \min_{x,a \in \{0,1\}} C_1 a + C_1 \sum \bar{a}_i x_i \]

Higher Order Submodular Function

Quadratic Submodular Function

[ Kohli et al. ‘08 ]
Higher order to Quadratic

\[ \min_{x} f(x) = \min_{x,a \in \{0,1\}} C_1 a + C_1 \sum \bar{a} x_i \]

Higher Order Submodular Function

Quadratic Submodular Function

Lower envelop of concave functions is concave

[Kohli et al. ’08]
Higher order to Quadratic

$$\min_x f(x) = \min_{x,a \in \{0,1\}} f_1(x)a + f_2(x)\bar{a}$$

Higher Order Submodular Function

Quadratic Submodular Function

Lower envelope of concave functions is concave

[Kohli et al. ‘08]
$$\min_x f(x) = \min_{x,a \in \{0,1\}} f_1(x)a + f_2(x)\bar{a}$$

Higher Order Submodular Function

Quadratic Submodular Function

Lower envelope of concave functions is concave

[Kohli et al. ‘08]
Upper Envelopes

\[
\min_x f(x) = \min_{x \in \{0,1\}} \max_\alpha f_1(x)\alpha + f_2(x)\bar{\alpha}
\]

Higher Order Submodular Function

Quadratic Submodular Function

Very Hard Problem!!!!

Upper envelope of linear functions is convex

[Kohli and Kumar, CVPR 2010]
Why Upper Envelopes?

SILHOUETTE CONSTRAINTS
[Sinha et al. ‘05, Cremers et al. ‘08]

Rays must touch silhouette at least once

3D RECONSTRUCTION

BINARY SEGMENTATION

Prior on size of object (say n/2)

SIZE/VOLUME PRIORS
[Woodford et al. 2009]

[Kohli and Kumar, CVPR 2010]
Pattern based representation
Encoding Image Structure for Texture Restoration

Higher Order Structure not Preserved

Pairwise Energy \( P(x) \)

Minimized using \( st\)-mincut or max-product message passing
Sparse Higher Order Potentials for Texture Restoration

Minimize: \[ E(X) = P(X) + \sum_{c} h_c (X_c) \]

Where: \( h_c: \{0, 1\}^{|c|} \rightarrow \mathbb{R} \)

Higher Order Function (\(|c| = 10 \times 10 = 100\))
Assigns cost to \(2^{100}\) possible labellings!

Exploit function structure to transform it to a Pairwise function

[Rother and Kohli, MSR Tech Report 2010]
Sparse Higher Order Potentials for Texture Restoration

Training Image

Learned Patterns

Test Image

Test Image (60% Noise)

Pairwise Result

Higher-Order Result

[Rother and Kohli, MSR Tech Report 2010]
Summary

- Exploiting structure of higher order models to transform them to pairwise functions
- Problem-aware adaptive inference algorithms
Challenges and Opportunities

- Scalability
- Average case analysis
- Parallel Architectures – GPUs/multi-core