Discrete Inference and Learning
Lecture 3

MVA
2017 – 2018

http://thoth.inrialpes.fr/~alahari/disinflearn

Slides based on material from Pushmeet Kohli, Nikos Komodakis, M. Pawan Kumar
Recap

- The st-mincut problem
- Connection between st-mincut and energy minimization?
- What problems can we solve using st-mincut?
- st-mincut based Move algorithms
St-mincut based Move algorithms

Commonly used for solving non-submodular multi-label problems

Extremely efficient and produce good solutions

Not Exact: Produce local optima
Move Making Algorithms

- Energy
- Solution Space
- Current Solution
- Search Neighbourhood
- Optimal Move
Move Making Algorithms

![Graph showing energy vs. solution space with key points labeled: Current Solution, Search Neighbourhood, and Optimal Move.](image-url)
Computing the Optimal Move

Key Property

Move Space

- Bigger move space
  - Better solutions
  - Finding the optimal move hard
Moves using Graph Cuts

Expansion and Swap move algorithms
[Boykov Veksler and Zabih, PAMI 2001]

- Makes a series of changes to the solution (moves)
- Each move results in a solution with smaller energy

Space of Solutions (y) : \( L^N \)
Move Space (t) : \( 2^N \)
Expansion and Swap move algorithms
[Boykov Veksler and Zabih, PAMI 2001]

- Makes a series of changes to the solution (moves)
- Each move results in a solution with smaller energy

How to minimize move functions?
General Binary Moves

Minimize over move variables $t$ to get the optimal move

$$y = t y^1 + (1- t) y^2$$

New solution  Current Solution  Second solution

$$E_m(t) = E(t y^1 + (1- t) y^2)$$

Minimize over move variables $t$ to get the optimal move

Move energy is a submodular QPBF  
(Exact Minimization Possible)

Boykov, Veksler and Zabih, PAMI 2001
Expansion Move

- Variables take label $\alpha$ or retain current label

Status: Initialize with Tree

[Boykov, Veksler, Zabih]
Expansion Move

- Variables take label $\alpha$ or retain current label

Status: Expand Ground

[Boykov, Veksler, Zabih]
Expansion Move

- Variables take label $\alpha$ or retain current label

Status: Expand House

[Boykov, Veksler, Zabih]
Expansion Move

• Variables take label $\alpha$ or retain current label

Status: Expand Sky

[Boykov, Veksler, Zabih]
Expansion Move

• Variables take label $\alpha$ or retain current label

• Move energy is submodular if:
  – Unary Potentials: Arbitrary
  – Pairwise potentials: Metric

\[
\theta_{ij}(l_a, l_b) \geq 0
\]
\[
\theta_{ij}(l_a, l_b) = 0 \text{ iff } a = b
\]

Examples: Potts model, Truncated linear

Cannot solve truncated quadratic

[Boykov, Veksler, Zabih]
Expansion Move

- Variables take label $\alpha$ or retain current label

- Move energy is submodular if:
  - Unary Potentials: Arbitrary
  - Pairwise potentials: Metric

\[ \theta_{ij}(l_a, l_b) + \theta_{ij}(l_b, l_c) \geq \theta_{ij}(l_a, l_c) \]

Triangle Inequality

Examples: Potts model, Truncated linear

Cannot solve truncated quadratic

[Boykov, Veksler, Zabih]
Swap Move

- Variables labeled $\alpha$, $\beta$ can swap their labels

Swap Sky, House

[Boykov, Veksler, Zabih]
Swap Move

• Variables labeled $\alpha$, $\beta$ can swap their labels

Swap Sky, House

[Boykov, Veksler, Zabih]
Swap Move

• Variables labeled $\alpha$, $\beta$ can swap their labels

• Move energy is submodular if:
  – Unary Potentials: Arbitrary
  – Pairwise potentials: Semimetric

\[
\theta_{ij}(l_a, l_b) \geq 0 \\
\theta_{ij}(l_a, l_b) = 0 \quad a = b
\]

Examples: Potts model, Truncated Convex

[Boykov, Veksler, Zabih]
Summary

Labelling Problem

Exact Transformation (global optimum)

Or Relaxed transformation (partially optimal)

Submodular Quadratic Pseudoboolean Function

Move making algorithms
Where do we stand?

Grid graph -
submodular, 2-label: Use graphcuts
“metric”:
Use expansion
otherwise: Use TRW,
dual decomposition,
relaxation

Chain/Tree, 2/multi-label: Use BP
Dynamic Energy Minimization
Image Segmentation in Video

\[ E(x) \]

\[ s = 0 \]

\[ t = 1 \]

Flow

Global Optimum

Video frame
Image Segmentation in Video

Video frame

Flow

Global Optimum
Dynamic Energy Minimization

$E_A$ minimize $S_A$

Can we do better?

$E_B$ S_B

computationally expensive operation

Recycling Solutions

Boykov & Jolly ICCV’01, Kohli & Torr (ICCV05, PAMI07)
Dynamic Energy Minimization

$E_A$  

minimize  

$S_A$  

Reuse flow  

$E_{B^*}$  

Simpler energy  

cheaper operation  

$S_B$  

Reparametrization  

computationally expensive operation  

A and B similar

$E_B$

Kohli & Torr (ICCV05, PAMI07)

Boykov & Jolly ICCV’01, Kohli & Torr (ICCV05, PAMI07)
Dynamic Energy Minimization

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

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

New Energy
\[ E(a_1, a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 7a_1\bar{a}_2 + \bar{a}_1a_2 \]

New Reparameterized Energy
\[ E(a_1, a_2) = 8 + \bar{a}_1 + 3a_2 + 3\bar{a}_1a_2 + 5a_1\bar{a}_2 \]
Outline

• Reparameterization (lecture 1)

• Belief Propagation (lecture 1)

• Tree-reweighted Message Passing
  – Integer Programming Formulation
  – Linear Programming Relaxation and its Dual
  – Convergent Solution for Dual
  – Computational Issues and Theoretical Properties
First...

Recap of Integer Linear Program
Integer Linear Program

\[
\max_x c^T x \\
\text{s.t. } A x \leq b \\
x \text{ is an integer vector}
\]

Every element of \( x \) is an integer
Integer Linear Program

\[ \max_x c^T x \]

s.t. \( A x \leq b \)

\( x \in \mathbb{Z}^n \)

Every element of \( x \) is an integer
Example

$$\max_x x_1 + x_2$$

s.t. \hspace{1cm} x_1 \geq 0

\hspace{1cm} x_2 \geq 0

\hspace{1cm} 4x_1 - x_2 \leq 8

\hspace{1cm} 2x_1 + x_2 \leq 10

\hspace{1cm} 5x_1 - 2x_2 \geq -2

x \in \mathbb{Z}^n
Example

\[ x_1 \geq 0 \]
\[ x_2 \geq 0 \]
Example

\[ 4x_1 - x_2 = 8 \]

\[ x_1 \geq 0 \]

\[ x_2 \geq 0 \]
Example

\begin{align*}
x_1 & \geq 0 \\
x_2 & \geq 0 \\
4x_1 - x_2 & \leq 8
\end{align*}
Example

\[\begin{align*}
4x_1 - x_2 & \leq 8 \\
2x_1 + x_2 & \leq 10 \\
5x_1 - 2x_2 & \geq -2
\end{align*}\]

\[x_1 \geq 0\]
\[x_2 \geq 0\]
Example

\[ \begin{align*}
4x_1 - x_2 & \leq 8 \\
2x_1 + x_2 & \leq 10 \\
5x_1 - 2x_2 & \geq -2
\end{align*} \]

\[ x_1 \geq 0 \quad \text{and} \quad x_2 \geq 0 \]

\[ \mathbf{x} \in \mathbb{Z}^n \]

\[ \max_{\mathbf{x}} c^T \mathbf{x} \]
Example

\[ 4x_1 - x_2 \leq 8 \]
\[ 2x_1 + x_2 \leq 10 \]
\[ 5x_1 - 2x_2 \geq -2 \]
\[ x_1 \geq 0 \]
\[ x_2 \geq 0 \]
\[ x \in \mathbb{Z}^n \]

Why? True in general?

\[ \max_x c^T x \]
Why is the solution at a vertex?

Let \( x^* \) be the minimum. Since each point in a polytope is a convex combination of the vertices \( v_1, \ldots, v_t \), we have

\[
x^* = \lambda_1 v_1 + \ldots + \lambda_t v_t
\]

and the objective value at optimality can be expressed as

\[
cx^* = \lambda_1 * (cv_1) + \ldots + \lambda_t (cv_t).
\]

Assume that the minimum is not at a vertex, i.e.,

\[
 cx^* < cv_i \quad \forall i : 1 \leq i \leq t.
\]

It follows that

\[
 cx^* = \lambda_1 * (cx^*) + \ldots + \lambda_t (cx^*) \\
 > \lambda_1 * (cx^*) + \ldots + \lambda_t (cx^*) \\
 > (\lambda_1 + \ldots + \lambda_t)(cx^*) \\
 > cx^*.
\]

Hence, it must be the case that \( x^* = v_i \) for some \( 1 \leq i \leq t \).
Unary Potentials

\[ \theta_{a;0} = 5 \quad \theta_{b;0} = 2 \]
\[ \theta_{a;1} = 2 \quad \theta_{b;1} = 4 \]

Labeling

\( f(a) = 1 \quad y_{a;0} = 0 \quad y_{a;1} = 1 \)
\( f(b) = 0 \quad y_{b;0} = 1 \quad y_{b;1} = 0 \)

Any \( f(.) \) has equivalent boolean variables \( y_{a;i} \)
**Integer Programming Formulation**

**Unary Potentials**

\[
\begin{align*}
\theta_{a;0} &= 5 & \theta_{b;0} &= 2 \\
\theta_{a;1} &= 2 & \theta_{b;1} &= 4
\end{align*}
\]

**Labeling**

\[
\begin{align*}
f(a) &= 1 & y_{a;0} &= 0 & y_{a;1} &= 1 \\
f(b) &= 0 & y_{b;0} &= 1 & y_{b;1} &= 0
\end{align*}
\]

Find the optimal variables \(y_{a;i}\)
Integer Programming Formulation

Unary Potentials

\[ \theta_{a;0} = 5 \quad \theta_{b;0} = 2 \]
\[ \theta_{a;1} = 2 \quad \theta_{b;1} = 4 \]

Sum of Unary Potentials

\[ \sum_a \sum_i \theta_{a;i} \ y_{a;i} \]
\[ y_{a;i} \in \{0,1\}, \text{ for all } V_a, l_i \]
\[ \sum_i y_{a;i} = 1, \text{ for all } V_a \]
Integer Programming Formulation

Pairwise Potentials

\[ \theta_{ab;00} = 0 \quad \theta_{ab;01} = 1 \]
\[ \theta_{ab;10} = 1 \quad \theta_{ab;11} = 0 \]

Sum of Pairwise Potentials

\[ \sum_{(a,b)} \sum_{ik} \theta_{ab;ik} y_{a;i} y_{b;k} \]

\[ y_{a;i} \in \{0,1\} \]
\[ \sum_i y_{a;i} = 1 \]
Integer Programming Formulation

**Pairwise Potentials**

\[
\begin{align*}
\theta_{ab;00} &= 0 & \theta_{ab;01} &= 1 \\
\theta_{ab;10} &= 1 & \theta_{ab;11} &= 0
\end{align*}
\]

**Sum of Pairwise Potentials**

\[
\sum_{(a,b)} \sum_{ik} \theta_{ab;ik} y_{ab;ik}
\]

\[
y_{a;i} \in \{0, 1\} \quad y_{ab;ik} = y_{a;i} y_{b;k}
\]

\[
\sum_i y_{a;i} = 1
\]
Integer Programming Formulation

\[
\min \sum_a \sum_i \theta_{a;i} y_{a;i} + \sum_{(a,b)} \sum_{ik} \theta_{ab;ik} y_{ab;ik}
\]

\[y_{a;i} \in \{0, 1\}\]

\[\sum_i y_{a;i} = 1\]

\[y_{ab;ik} = y_{a;i} y_{b;k}\]
Integer Programming Formulation

\[
\min \theta^T y
\]

\[y_{a;i} \in \{0,1\}\]

\[\sum_i y_{a;i} = 1\]

\[y_{ab;ik} = y_{a;i} \cdot y_{b;k}\]

\[\theta = [ \ldots \theta_{a;i} \ldots ; \ldots \theta_{ab;ik} \ldots ]\]

\[y = [ \ldots y_{a;i} \ldots ; \ldots y_{ab;ik} \ldots ]\]
Integer Programming Formulation

\[
\min \theta^T y \\
\sum_i y_{a; i} = 1 \\
y_{ab; ik} = y_{a; i} y_{b; k}
\]

Solve to obtain MAP labeling \( y^* \)
Integer Programming Formulation

\[ \text{min } \theta^T y \]

\[ y_{a;i} \in \{0,1\} \]

\[ \sum_i y_{a;i} = 1 \]

\[ y_{ab;ik} = y_{a;i} \cdot y_{b;k} \]

But we can't solve it in general
Outline

• Reparameterization (lecture 1)

• Belief Propagation (lecture 1)

• Tree-reweighted Message Passing
  – Integer Programming Formulation
  – Linear Programming Relaxation and its Dual
  – Convergent Solution for Dual
  – Computational Issues and Theoretical Properties
Linear Programming Relaxation

\[ \min \theta^T y \]

\[ y_{a;i} \in \{0,1\} \]

\[ \sum_i y_{a;i} = 1 \]

\[ y_{ab;ik} = y_{a;i} y_{b;k} \]

Two reasons why we can’t solve this
Linear Programming Relaxation

\[
\begin{align*}
\min & \quad \theta^T y \\
y_{a;i} & \in [0,1] \\
\sum_i y_{a;i} & = 1 \\
y_{a;i} y_{b;k} & = y_{ab;ik}
\end{align*}
\]

One reason why we can’t solve this
Linear Programming Relaxation

\[ \min \theta^T y \]

\[ y_{a;i} \in [0,1] \]

\[ \sum_i y_{a;i} = 1 \]

\[ \sum_k y_{ab;ik} = \sum_k y_{a;i} y_{b;k} \]

One reason why we can’t solve this
Linear Programming Relaxation

\[
\min \theta^T y
\]

\[
y_{a;i} \in [0,1]
\]

\[
\sum_i y_{a;i} = 1
\]

\[
\sum_k y_{ab;ik} = y_{a;i} \sum_k y_{b;k} = 1
\]

One reason why we can’t solve this
Linear Programming Relaxation

\[
\min \theta^T y
\]

\[y_{a;i} \in [0,1]\]

\[\sum_i y_{a;i} = 1\]

\[
\sum_k y_{ab;ik} = y_{a;i}
\]

One reason why we can’t solve this
Linear Programming Relaxation

\[
\begin{align*}
\min & \quad \theta^T y \\
\text{s.t.} & \quad y_{a;i} \in [0,1] \\
& \quad \sum_i y_{a;i} = 1 \\
& \quad \sum_k y_{ab;ik} = y_{a;i}
\end{align*}
\]

No reason why we can’t solve this*

* memory requirements, time complexity
Dual of the LP Relaxation

Wainwright et al., 2001

\[
\begin{align*}
\min \theta^T y \\
y_{a;i} &\in [0, 1] \\
\sum_i y_{a;i} &= 1 \\
\sum_k y_{ab;ik} &= y_{a;i}
\end{align*}
\]
Wainwright et al., 2001

\[ \sum \rho^i \theta^i = \theta \]
Dual of the LP Relaxation

Wainwright et al., 2001

\[
\max \sum \rho_i q^*(\theta_i) \\
\sum \rho_i \theta_i = \theta
\]

\[
q^*(\theta^1) \quad V_a \quad V_b \quad V_c \quad \rho^1 \\
q^*(\theta^2) \quad V_d \quad V_e \quad V_f \quad \rho^2 \\
q^*(\theta^3) \quad V_g \quad V_h \quad V_i \quad \rho^3 \\
q^*(\theta^4) \quad \rho^4 \quad q^*(\theta^5) \quad \rho^5 \quad q^*(\theta^6) \quad \rho^6
\]

\[\rho^i \geq 0\]
Dual of the LP Relaxation

Wainwright et al., 2001

\[ \sum \rho_i \theta_i = \theta \]

\[ \max \sum \rho_i q^*(\theta_i) \]

\[ \rho^i \geq 0 \]
Dual of the LP Relaxation

Wainwright et al., 2001

\[
\max \sum \rho^i q^*(\theta^i)
\]

\[
\sum \rho^i \theta^i \equiv \theta
\]

I can easily compute \( q^*(\theta^i) \)

I can easily maintain reparam constraint

So can I easily solve the dual?
Outline

• Reparameterization (lecture 1)

• Belief Propagation (lecture 1)

• Tree-reweighted Message Passing
  – Integer Programming Formulation
  – Linear Programming Relaxation and its Dual
  – Convergent Solution for Dual
  – Computational Issues and Theoretical Properties
TRW Message Passing

Kolmogorov, 2006

\( \sum \rho^i q^*(\theta^i) \)

\( \sum \rho^i \theta^i \equiv \theta \)
TRW Message Passing

Kolmogorov, 2006

\[ \sum \rho^i q^*(\theta^i) \]

\[ \sum \rho^i \theta^i \equiv \theta \]
TRW Message Passing

Kolmogorov, 2006

\[ \rho^1 q^*(\theta^1) + \rho^4 q^*(\theta^4) + K \]
\[ \rho^1 \theta^1 + \rho^4 \theta^4 + \theta_{\text{rest}} = \theta \]
One pass of Belief Propagation

\[ \rho^1 q^*(\theta'1) + \rho^4 q^*(\theta'4) + K \]

\[ \rho^1 \theta'1 + \rho^4 \theta'4 + \theta_{rest} \]
TRW Message Passing

Kolmogorov, 2006

\[ \theta'_{c;1} \quad \theta'_{b;1} \quad \theta'_{a;1} \]
\[ \theta'_{c;0} \quad \theta'_{b;0} \quad \theta'_{a;0} \]
\[ V_c \quad V_b \quad V_a \]

\[ \theta'_{a;1} \quad \theta'_{d;1} \quad \theta'_{g;1} \]
\[ \theta'_{a;0} \quad \theta'_{d;0} \quad \theta'_{g;0} \]
\[ V_a \quad V_d \quad V_g \]

Remain the same

\[ \rho^1 q^*(\theta'^1) + \rho^4 q^*(\theta'^4) + K \]
\[ \rho^1 \theta'^1 + \rho^4 \theta'^4 + \theta_{\text{rest}} \equiv \theta \]
\[ \rho^1 \min\{\theta'_1 a;0, \theta'_1 a;1\} + \rho^4 \min\{\theta'_4 a;0, \theta'_4 a;1\} + \mathbf{K} \]

\[ \rho^1 \theta'_1 + \rho^4 \theta'_4 + \theta_{\text{rest}} \equiv \theta \]
TRW Message Passing

Kolmogorov, 2006

\[ \rho^1 \min\{\theta'_{a;0}, \theta'_{a;1}\} + \rho^4 \min\{\theta'_{4a;0}, \theta'_{4a;1}\} + K \]

\[ \rho^1 \theta' + \rho^4 \theta'_{4} + \theta_{\text{rest}} \equiv \theta \]
TRW Message Passing

Kolmogorov, 2006

\[ \theta''_{a;0} = \frac{\rho^1 \theta'_{a;0} + \rho^4 \theta'_{a;0}}{\rho^1 + \rho^4} \]

\[ \theta''_{a;1} = \frac{\rho^1 \theta'_{a;1} + \rho^4 \theta'_{a;1}}{\rho^1 + \rho^4} \]

\[ \rho^1 \min\{\theta'_{a;0}, \theta'_{a;1}\} + \rho^4 \min\{\theta'_{a;0}, \theta'_{a;1}\} + \lambda \]

\[ \rho^1 \theta'_{1} + \rho^4 \theta'_{4} + \theta_{\text{rest}} \equiv \theta \]
TRW Message Passing

Kolmogorov, 2006

\[
\begin{align*}
\theta''_{a;0} &= \frac{\rho^1 \theta'_{a;0} + \rho^4 \theta'_{a;0}}{\rho^1 + \rho^4} \\
\theta''_{a;1} &= \frac{\rho^1 \theta'_{a;1} + \rho^4 \theta'_{a;1}}{\rho^1 + \rho^4} \\
\rho^1 \min\{\theta'_{a;0}, \theta'_{a;1}\} + \rho^4 \min\{\theta'_{a;0}, \theta'_{a;1}\} + K \\
\rho^1 \theta''_{a;0} + \rho^4 \theta''_{a;1} + \theta_{\text{rest}}
\end{align*}
\]
TRW Message Passing

Kolmogorov, 2006

\[ \theta''_{a;0} = \frac{\rho^1 \theta'_{a;0} + \rho^4 \theta'_{a;0}}{\rho^1 + \rho^4} \]

\[ \theta''_{a;1} = \frac{\rho^1 \theta'_{a;1} + \rho^4 \theta'_{a;1}}{\rho^1 + \rho^4} \]

\[ \rho^1 \min\{\theta'_{a;0}, \theta'_{a;1}\} + \rho^4 \min\{\theta'_{a;0}, \theta'_{a;1}\} + K \]

\[ \rho^1 \theta''_{1} + \rho^4 \theta''_{4} + \theta_{\text{rest}} = \theta \]
TRW Message Passing

Kolmogorov, 2006

\[ \theta''_{a;0} = \frac{\rho^1 \theta'_1 a;0 + \rho^4 \theta'_4 a;0}{\rho^1 + \rho^4} \]

\[ \theta''_{a;1} = \frac{\rho^1 \theta'_1 a;1 + \rho^4 \theta'_4 a;1}{\rho^1 + \rho^4} \]

\[ \rho^1 \text{min}\{\theta''_{a;0}, \theta''_{a;1}\} + \rho^4 \text{min}\{\theta''_{a;0}, \theta''_{a;1}\} + K \]

\[ \rho^1 \theta''_1 + \rho^4 \theta''_4 + \theta_{\text{rest}} \equiv \theta \]
\[ \theta''_{a;0} = \frac{\rho^1 \theta'_{c;0} + \rho^4 \theta'_{b;0}}{\rho^1 + \rho^4} \]

\[ \theta''_{a;1} = \frac{\rho^1 \theta'_{b;1} + \rho^4 \theta'_{a;1}}{\rho^1 + \rho^4} \]

\[(\rho^1 + \rho^4) \min\{\theta''_{a;0}, \theta''_{a;1}\} + K\]

\[\rho^1 \theta''_{c;0} + \rho^4 \theta''_{b;1} + \theta_{\text{rest}} \equiv \theta\]
TRW Message Passing

Kolmogorov, 2006

\[
\theta'_{1,c;1} \theta'_{1,b;1} \theta''_{a;1} \\
\theta'_{1,c;0} \theta'_{1,b;0} \theta''_{a;0}
\]

\[
\begin{align*}
\min \{p_1 + p_2, q_1 + q_2\} & \geq \min \{p_1, q_1\} + \min \{p_2, q_2\} \\
(\rho^1 + \rho^4) & \min\{\theta''_{a;0}, \theta''_{a;1}\} + K \\
\rho^1\theta''_{1} + \rho^4\theta''_{4} + \theta_{\text{rest}} & = \theta
\end{align*}
\]
Objective function increases or remains constant

\[(\rho^1 + \rho^4) \min\{\theta''_a;0, \theta''_a;1\} + K\]

\[\rho^1 \theta''^1 + \rho^4 \theta''^4 + \theta_{\text{rest}} \equiv \theta\]
TRW Message Passing

Initialize $\theta^i$. Take care of reparam constraint

Choose random variable $V_a$

Compute min-marginals of $V_a$ for all trees

Node-average the min-marginals

REPEAT Can also do edge-averaging

Kolmogorov, 2006
Example 1

Pick variable $V_a$. Reparameterize.
Example 1

\[ \rho^1 = 1 \]

\[ \rho^2 = 1 \]

\[ \rho^3 = 1 \]

Average the min-marginals of \( V_a \)
Example 1

\[ \rho^1 = 1 \]

\[ \rho^2 = 1 \]

\[ \rho^3 = 1 \]

Pick variable \( V_b \). Reparameterize.
Example 1

\[ \rho^1 = 1 \quad \rho^2 = 1 \quad \rho^3 = 1 \]

Average the min-marginals of \( V_b \)
Example 1

\[ \rho^1 = 1 \]

\[ \rho^2 = 1 \]

\[ \rho^3 = 1 \]

Value of dual does not increase
Example 1

Maybe it will increase for $V_c$

NO
Example 1

Strong Tree Agreement

Exact MAP Estimate
Example 2

\[ \rho^1 = 1 \]

\[ \rho^2 = 1 \]

\[ \rho^3 = 1 \]

Pick variable \( V_a \). Reparameterize.
Example 2

Average the min-marginals of $V_a$
Example 2

Value of dual does not increase
Example 2

Maybe it will decrease for $V_b$ or $V_c$

NO
Example 2

\[ \rho^1 = 1 \]

\[ \rho^2 = 1 \]

\[ \rho^3 = 1 \]

Weak Tree Agreement

Not Exact MAP Estimate
Example 2

Weak Tree Agreement

Convergence point of TRW
Obtaining the Labeling

Only solves the dual. Primal solutions?

\[ \theta' = \sum \rho^i \theta^i \equiv \theta \]

Fix the label of \( V_a \)
Obtaining the Labeling

Only solves the dual. Primal solutions?

\[ \theta' = \sum \rho^i \theta^i \equiv \theta \]

Fix the label of \( V_b \)

Continue in some fixed order

Meltzer et al., 2006
Outline

• Reparameterization (lecture 1)

• Belief Propagation (lecture 1)

• Tree-rewighted Message Passing
  – Integer Programming Formulation
  – Linear Programming Relaxation and its Dual
  – Convergent Solution for Dual
  – Computational Issues and Theoretical Properties
Computational Issues of TRW

Basic Component is Belief Propagation

- Speed-ups for some pairwise potentials
  Felzenszwalb & Huttenlocher, 2004

- Memory requirements cut down by half
  Kolmogorov, 2006

- Further speed-ups using monotonic chains
  Kolmogorov, 2006
Theoretical Properties of TRW

- Always converges, unlike BP
  
  Kolmogorov, 2006

- Strong tree agreement implies exact MAP
  
  Wainwright et al., 2001

- Optimal MAP for two-label submodular problems

\[ \theta_{ab;00} + \theta_{ab;11} \leq \theta_{ab;01} + \theta_{ab;10} \]

Kolmogorov and Wainwright, 2005
Summary

• Trees can be solved exactly - BP

• No guarantee of convergence otherwise - BP

• Strong Tree Agreement - TRW-S

• Submodular energies solved exactly - TRW-S

• TRW-S solves an LP relaxation of MAP estimation