Practical matters

• Course website
  – http://thoth.inrialpes.fr/~alahari/disinflearn
  – (linked from my webpage)

• Mailing list
• Questions ?
Project suggestions

• Implement BP on trees, then graph, extend to TRW, compare
• Implement graph cut + extension (Ishikawa, other multi-label) or variation of implementation + small application
• Complex application of graph cut, requiring modelling (e.g., sequence of images)
• Geometric scene labelling with graph cuts
• Joint modelling of two labelling problems (e.g., segmentation + detection)
• Implement fast primal-dual algorithm + evaluate
• Implement deformable parts model for object detection
• ...

• Or your own (but check with us first)
Recap: Lecture 1

• Graphical Models
  – Making **global** predictions from **local** observations
  – Learning from large quantities of data

• Two types of models studied in the class
  – Bayesian nets
  – Markov nets
Recap: Lecture 1
Recap: Lecture 1

• Question: What is the core of these models?

• Question: Can you compute probabilities in Markov nets? If yes, how and if no, why?

• Question: What is the difference between Markov and Conditional random fields?
A Computer Vision Application

Binary Image Segmentation

How?

Cost function Models our knowledge about natural images

Optimize cost function to obtain the segmentation
A Computer Vision Application

Binary Image Segmentation

Object - white, Background - green/grey

Each vertex corresponds to a pixel

Edges define a 4-neighbourhood grid graph

Assign a label to each vertex from $L = \{\text{obj}, \text{bkg}\}$

Graph $G = (V,E)$
A Computer Vision Application

Binary Image Segmentation

Object - white, Background - green/grey

Cost of a labelling $f : V \rightarrow L$

Cost of label ‘obj’ low Cost of label ‘bkg’ high

Graph $G = (V,E)$

Per Vertex Cost
A Computer Vision Application

Binary Image Segmentation

Graph $G = (V,E)$

Cost of a labelling $f : V \rightarrow L$

Object - white, Background - green/grey

Cost of label ‘obj’ high Cost of label ‘bkg’ low

Per Vertex Cost

UNARY COST
A Computer Vision Application

Binary Image Segmentation

Graph $G = (V,E)$

Cost of a labelling $f : V \rightarrow L$

Object - white, Background - green/grey

Cost of same label low

Cost of different labels high

Per Edge Cost
A Computer Vision Application

Binary Image Segmentation

Object - white, Background - green/grey

Cost of a labelling $f : V \rightarrow L$

Cost of same label high
Cost of different labels low

Graph $G = (V, E)$

Per Edge Cost

PAIRWISE COST
A Computer Vision Application

Binary Image Segmentation

Problem: Find the labelling with minimum cost $f^*$
Another Computer Vision Application

Stereo Correspondence

Disparity Map

How?

Minimizing a cost function
Another Computer Vision Application

Stereo Correspondence

Graph $G = (V, E)$

- Vertex corresponds to a pixel
- Edges define grid graph

$L = \{\text{disparities}\}$
Another Computer Vision Application

Stereo Correspondence

Cost of labelling $f$:

Unary cost + Pairwise Cost

Find minimum cost $f^*$
The General Problem

Graph $G = (V, E)$

Discrete label set $L = \{1, 2, \ldots, h\}$

Assign a label to each vertex $f : V \rightarrow L$

Cost of a labelling $Q(f)$

Unary Cost Pairwise Cost

Find $f^* = \text{arg min } Q(f)$
Overview

• Basics: problem formulation
  – Energy Function
  – MAP Estimation
  – Computing min-marginals
  – Reparameterization

• Solutions
  – Belief Propagation and related methods [Lecture 2]
  – Graph cuts [Lecture 3]
Energy Function

Random Variables $V = \{V_a, V_b, \ldots\}$

Labels $L = \{l_0, l_1, \ldots\}$

Data $D$

Labelling $f: \{a, b, \ldots\} \rightarrow \{0,1, \ldots\}$
Energy Function

\[ Q(f) = \sum_{a} \theta_{a;f(a)} \]

Unary Potential

Easy to minimize

Neighbourhood
Energy Function

\[ E : (a,b) \in E \text{ iff } V_a \text{ and } V_b \text{ are neighbours} \]

\[ E = \{ (a,b) , (b,c) , (c,d) \} \]
Energy Function

\[ Q(f) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)} \]
\[
Q(f; \theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}
\]
Overview

• Basics: problem formulation
  – Energy Function
  – MAP Estimation
  – Computing min-marginals
  – Reparameterization

• Solutions
  – Belief Propagation and related methods [Lecture 2]
  – Graph cuts [Lecture 3]
$Q(f; \theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$

$2 + 1 + 2 + 1 + 3 + 1 + 3 = 13$
MAP Estimation

\[ Q(f; \theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)} \]

\[ 5 + 1 + 4 + 0 + 6 + 4 + 7 = 27 \]
\[ q^* = \min Q(f; \theta) = Q(f^*; \theta) \]

\[ Q(f; \theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)} \]

\[ f^* = \arg \min Q(f; \theta) \]

Equivalent to maximizing the associated probability
**MAP Estimation**

16 possible labellings

\[ f^* = \{1, 0, 0, 1\} \]

\[ q^* = 13 \]

<table>
<thead>
<tr>
<th>f(a)</th>
<th>f(b)</th>
<th>f(c)</th>
<th>f(d)</th>
<th>Q(f; \theta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>f(a)</th>
<th>f(b)</th>
<th>f(c)</th>
<th>f(d)</th>
<th>Q(f; \theta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>16</td>
</tr>
</tbody>
</table>
Computational Complexity

Segmentation

$2^{|V|}$

$|V| = \text{number of pixels} \approx 153600$

Can we do better than brute-force?

MAP Estimation is NP-hard!!
MAP Inference / Energy Minimization

• Computing the assignment minimizing the energy in NP-hard in general
  \[
  \arg\min_{\mathbf{y}} E(\mathbf{y}; \mathbf{x}) = \arg\max_{\mathbf{y}} P(\mathbf{y} \mid \mathbf{x})
  \]

• Exact inference is possible in some cases, e.g.,
  – Low treewidth graphs \(\rightarrow\) message-passing
  – Submodular potentials \(\rightarrow\) graph cuts

• Efficient approximate inference algorithms exist
  – Message passing on general graphs
  – Move-making algorithms
  – Relaxation algorithms
Overview

• Basics: problem formulation
  – Energy Function
  – MAP Estimation
  – Computing min-marginals
  – Reparameterization

• Solutions
  – Belief Propagation and related methods [Lecture 2]
  – Graph cuts [Lecture 3]
\[ f^* = \arg \min Q(f; \theta) \quad \text{such that } f(a) = i \]

Min-marginal \( q_{a;i} \)

Not a marginal (no summation)
## Min-Marginals

16 possible labellings

\[ q_{a;0} = 15 \]

<table>
<thead>
<tr>
<th>f(a)</th>
<th>f(b)</th>
<th>f(c)</th>
<th>f(d)</th>
<th>Q(f; \theta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>f(a)</th>
<th>f(b)</th>
<th>f(c)</th>
<th>f(d)</th>
<th>Q(f; \theta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>16</td>
</tr>
</tbody>
</table>
Min-Marginals

16 possible labellings

$q_{a;1} = 13$

<table>
<thead>
<tr>
<th>f(a)</th>
<th>f(b)</th>
<th>f(c)</th>
<th>f(d)</th>
<th>Q(f; θ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>f(a)</th>
<th>f(b)</th>
<th>f(c)</th>
<th>f(d)</th>
<th>Q(f; θ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>16</td>
</tr>
</tbody>
</table>
Min-Marginals and MAP

• Minimum min-marginal of any variable = energy of MAP labelling

\[
\min_i q_{a;i} \\
\min_i \left( \min_f Q(f; \theta) \text{ such that } f(a) = i \right)
\]

\(V_a\) has to take one label

\[
\min_f Q(f; \theta)
\]
Summary

Energy Function

\[ Q(f; \theta) = \sum_a \theta_a;f(a) + \sum_{(a,b)} \theta_{ab};f(a)f(b) \]

MAP Estimation

\[ f^* = \arg \min Q(f; \theta) \]

Min-marginals

\[ q_{a,i} = \min Q(f; \theta) \quad \text{s.t.} \quad f(a) = i \]
Overview

• Basics: problem formulation
  – Energy Function
  – MAP Estimation
  – Computing min-marginals
  – Reparameterization

• Solutions
  – Belief Propagation and related methods [Lecture 2]
  – Graph cuts [Lecture 3]
Reparameterization

Add a constant to all $\theta_a;i$

Subtract that constant from all $\theta_b;k$

$$Q(f; \theta') = Q(f; \theta)$$
Reparameterization

Add a constant to one $\theta_{b;k}$

Subtract that constant from $\theta_{ab;ik}$ for all ‘i’

$$Q(f; \theta') = Q(f; \theta)$$
Reparameterization

\( \theta' \) is a reparameterization of \( \theta \), iff

\[
Q(f; \theta') = Q(f; \theta), \text{ for all } f \quad \theta' \equiv \theta
\]

Equivalently

\[
\begin{align*}
\theta'_{a;i} &= \theta_{a;i} + M_{ba;i} \\
\theta'_{b;k} &= \theta_{b;k} + M_{ab;k} \\
\theta'_{ab;ik} &= \theta_{ab;ik} - M_{ab;k} - M_{ba;i}
\end{align*}
\]

Kolmogorov, PAMI, 2006

\[
\begin{array}{c}
\begin{array}{c}
2 + 2 \\
5 \\
2 + 2
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
0 \\
1 \\
0 \\
2 \\
4 - 2
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
1 \\
1
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
V_a \\
V_b
\end{array}
\end{array}
\]
Recap

MAP Estimation

\[ f^* = \arg \min Q(f; \theta) \]

\[ Q(f; \theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)} \]

Min-marginals

\[ q_{a;i} = \min Q(f; \theta) \quad \text{s.t.} \quad f(a) = i \]

Reparameterization

\[ Q(f; \theta') = Q(f; \theta), \text{ for all } f \quad \theta' \equiv \theta \]
Overview

• Basics: problem formulation
  – Energy Function
  – MAP Estimation
  – Computing min-marginals
  – Reparameterization

• Solutions
  – Belief Propagation and related methods [Lecture 2]
  – Graph cuts [Lecture 3]
Belief Propagation

- Remember, some MAP problems are easy
- Belief Propagation gives exact MAP for chains
- Exact MAP for trees
- Clever Reparameterization
Two Variables

Add a constant to one $\theta_{b;k}$

Subtract that constant from $\theta_{ab;ik}$ for all ‘$i$’

Choose the *right* constant $\theta’_{b;k} = q_{b;k}$
Choose the \textit{right} constant $\theta'_{b;k} = q_{b;k}$
Choose the right constant $\theta'_{b;k} = q_{b;k}$
Choose the right constant \( \theta'_{b;k} = q_{b;k} \)
Minimum of min-marginals = MAP estimate

Choose the **right** constant $\theta'_{b;k} = q_{b;k}$
Choose the right constant $	heta'_{b;k} = q_{b;k}$
Choose the right constant 

\[ \theta'_{b;k} = q_{b;k} \]
Recap

We only need to know two sets of equations

General form of Reparameterization

\[ \theta'_{a;i} = \theta_{a;i} + M_{ba;i} \quad \theta'_{b;k} = \theta_{b;k} + M_{ab;k} \]

\[ \theta'_{ab;ik} = \theta_{ab;ik} - M_{ab;k} - M_{ba;i} \]

Reparameterization of (a,b) in Belief Propagation

\[ M_{ab;k} = \min_i \{ \theta_{a;i} + \theta_{ab;ik} \} \]

\[ M_{ba;i} = 0 \]
Reparameterize the edge (a, b) as before
Reparameterize the edge \((a,b)\) as before

Potentials along the red path add up to 0
Reparameterize the edge \((b,c)\) as before

Potentials along the red path add up to 0
Reparameterize the edge (b,c) as before

Potentials along the red path add up to 0
Three Variables

f(a) = 1  f(b) = 1

f*(c) = 0  f*(b) = 0  f*(a) = 1

Generalizes to any length chain
Three Variables

Three Variables

f(a) = 1  f(b) = 1

f*(c) = 0  f*(b) = 0  f*(a) = 1

Only Dynamic Programming
Why Dynamic Programming?

3 variables $\equiv$ 2 variables + book-keeping
n variables $\equiv$ (n-1) variables + book-keeping

Start from left, go to right

Reparameterize current edge (a,b)

$$M_{ab;k} = \min_i \{ \theta_{a;i} + \theta_{ab;ik} \}$$

$$\theta'_{b;k} = \theta_{b;k} + M_{ab;k}$$

$$\theta'_{ab;ik} = \theta_{ab;ik} - M_{ab;k}$$

Repeat
Why Dynamic Programming?

Messages  Message Passing

Why stop at dynamic programming?

Start from left, go to right

Reparameterize current edge \((a,b)\)

\[
M_{ab;k} = \min_i \{ \theta_{a;i} + \theta_{ab;ik} \}
\]

\[
\theta'_{b;k} = \theta_{b;k} + M_{ab;k} \quad \theta'_{ab;ik} = \theta_{ab;ik} - M_{ab;k}
\]

Repeat
Reparameterize the edge \((c,b)\) as before
Reparameterize the edge (c,b) as before

\[ \theta'_{b;i} = q_{b;i} \]
Reparameterize the edge (b,a) as before

Three Variables
Reparameterize the edge \((b,a)\) as before

\[ \theta'_{a;i} = q_{a;i} \]
Three Variables

Forward Pass ➔ Backward Pass

All min-marginals are computed
Reparameterize the edge (1, 2)
Chains

Reparameterize the edge (1,2)
Chains

Reparameterize the edge (2,3)
Chains

Reparameterize the edge (n-1,n)

Min-marginals $e_n(i)$ for all labels
Belief Propagation on Chains

Start from left, go to right

Reparameterize current edge $(a,b)$

$$M_{ab;k} = \min_i \{ \theta_{a;i} + \theta_{ab;ik} \}$$

$$\theta'_{b;k} = \theta_{b;k} + M_{ab;k} \quad \theta'_{ab;ik} = \theta_{ab;ik} - M_{ab;k}$$

Repeat till the end of the chain

Start from right, go to left

Repeat till the end of the chain
Belief Propagation on Chains

- Generalizes to chains of any length
- A way of computing reparam constants

- Forward Pass - Start to End
  - MAP estimate
  - Min-marginals of final variable

- Backward Pass - End to start
  - All other min-marginals
Computational Complexity

Number of reparameterization constants = \((n-1)h\)

Complexity for each constant = \(O(h)\)

Total complexity = \(O(nh^2)\)

Better than brute-force \(O(h^n)\)
Reparameterize the edge (4,2)
Reparameterize the edge (4,2)
Reparameterize the edge (5,2)
Reparameterize the edge (7,3)
Reparameterize the edge (2,1)
Reparameterize the edge (3,1)

Min-marginals $e_1(i)$ for all labels
Trees

Start from leaves and move towards root

Pick the minimum of min-marginals

Backtrack to find the best labeling $x$
Belief Propagation on Cycles

Where do we start? Arbitrarily
Reparameterize (a,b)
Belief Propagation on Cycles

Potentials along the red path add up to 0
Belief Propagation on Cycles

Potentials along the red path add up to 0
Belief Propagation on Cycles

Potentials along the red path add up to 0
Belief Propagation on Cycles

Potentials along the red path add up to 0
Belief Propagation on Cycles

\[ \theta'_{a;1} - \theta_{a;1} = q_{a;0} \]
\[ \theta'_{a;0} - \theta_{a;0} = q_{a;1} \]

Potentials along the red path add up to 0
Belief Propagation on Cycles

\[ \theta'_{a;0} - \theta_{a;0} = q_{a;0} \quad \theta'_{a;1} - \theta_{a;1} = q_{a;1} \]

Pick minimum min-marginal. Follow red path.
Belief Propagation on Cycles

Potentials along the red path add up to 0
Belief Propagation on Cycles

Potentials along the red path add up to 0
Belief Propagation on Cycles

Potentials along the red path add up to 0
Belief Propagation on Cycles

\[ \theta'_{a;1} - \theta_{a;1} = q_{a;1} \leq \theta'_{a;0} - \theta_{a;0} = q_{a;0} \]

Potentials along the red path add up to 0
Belief Propagation on Cycles

Problem Solved

\[ \theta'_{a;1} - \theta_{a;1} = q_{a;1} \leq \theta'_{a;0} - \theta_{a;0} = q_{a;0} \]
Belief Propagation on Cycles

\[ \theta'_{a;1} - \theta_{a;1} = q_{a;1} \geq \theta'_{a;0} - \theta_{a;0} = q_{a;0} \]

Problem Not Solved
Belief Propagation on Cycles

Reparameterize (a,b) again
Belief Propagation on Cycles

Reparameterize \((a,b)\) again

But doesn’t this overcount some potentials?
Reparameterize \((a,b)\) again

Yes. But we will do it anyway
Belief Propagation on Cycles

Keep reparameterizing edges in some order

Hope for convergence and a good solution
Belief Propagation on Cycles

Any suggestions? Fix $V_a$ to label $l_0$

Equivalent to a tree-structured problem
Belief Propagation on Cycles

Any suggestions?

Fix $V_a$ to label $l_1$

Equivalent to a tree-structured problem
Belief Propagation on Cycles

This approach quickly becomes infeasible.

Choose the minimum energy solution.
Loopy Belief Propagation

Keep reparameterizing edges in some order.

Hope for convergence and a good solution.
Belief Propagation

- Generalizes to any arbitrary random field

- Complexity per iteration: $O(|E||L|^2)$

- Memory required: $O(|E||L|)$
Summary of BP

Exact for chains

Exact for trees

Approximate MAP for general cases

Not even convergence guaranteed

So can we do something better?
Other alternatives

• Integer linear programming and relaxation

• TRW, Dual decomposition methods

• Extensively studied
  - Schlesinger, 1976
  - Koster et al., 1998, Chekuri et al., ’01, Archer et al., ’04
  - Wainwright et al., 2001, Kolmogorov, 2006
  - Globerson and Jaakkola, 2007, Komodakis et al., 2007
  - Kumar et al., 2007, Sontag et al., 2008, Werner, 2008
  - Batra et al., 2011, Werner, 2011, Zivny et al., 2014
Where do we stand?

Chain/Tree, 2-label: Use BP

Chain/Tree, multi-label: Use BP

Grid graph: Use TRW, dual decomposition, relaxation