Convergent message passing algorithms – a unifying view

Talya Meltzer,
Amir Globerson and Yair Weiss
Inference in Undirected Graphical Models

Protein structure to design

Connect interacting positions...

Graphical Model
Inference in Undirected Graphical Models

Probability assumed to factorize:

$$\Pr(x; \theta) = \frac{1}{Z} \cdot \exp \left( \sum_{\langle i, j \rangle \in E} \theta_{ij}(x_i, x_j) \right)$$

Two classic inference problems:

1. Finding the log-partition function:
   $$\ln Z = \ln \left( \sum_x \exp (\theta(x)) \right)$$

2. Finding the MAP:
   $$\text{MAP} = \max_x \theta(x)$$

$$\theta(x) = \sum_{\langle ij \rangle} \theta_{ij}(x_i, x_j)$$
Message passing algorithms

- Inference often approximated via message passing algorithms
  - Sum product for $\ln Z$
  - Max product for MAP
- Belief propagation (BP) often successful but may not converge
- Convex variants aim to fix this
- (Too) many different algorithms.
- How are they related?
Convergent BP Variants

- Solve a convex variational problem
- Iteratively minimize a bound

\[ \hat{x}(\theta) \]
Convergent BP Variants

- Solve a convex variational problem
- Iteratively minimize a bound
  - Heskes ’06
  - Kolmogorov ’06
  - Werner ’07
  - Globerson & Jaakkola ’07
  - Hazan & Shashua ’08
- Proofs are specific
Convergent Message Passing Algorithms – A Unifying View

- In this work:
  - A general bound for approx. $\ln Z$ and MAP
  - Show that they are very similar
  - Give framework for bound optimization.
  - Identify existing algorithms as instances
  - Obtain new algorithms “for free”
Approximate Inference in Region Graphs

**Definition:** A region graph

regions of nodes and edges

their intersections

\[
\theta_{ij}(x_i, x_j) \rightarrow \theta_\alpha(x_\alpha)
\]
Approximate Inference in Region Graphs

**Optimization problem:** \( \ln Z = \max_{q} \left( \left\langle \theta(x) \right\rangle_{q} + H(q) \right) \)

**Variational problem:** \( \ln Z \approx \ln \tilde{Z} = \max_{q \in \text{local}} \left( \left\langle \theta(x) \right\rangle_{q} + \tilde{H}(q) \right) \)

Relax (1) Marginals / beliefs only locally consistent
\[
q_{\beta}(x_{\beta}) = \sum_{x_{\alpha} \setminus x_{\beta}} q_{\alpha}(x_{\alpha})
\]

Relax (2) Approximate entropy
\[
\tilde{H}(q) = \sum_{\alpha} c_{\alpha} \cdot H_{\alpha}(q_{\alpha})
\]

Double counting number
Region’s entropy
Approximate Inference and Re-parameterization

- All algorithms we discuss use variables $\tilde{\theta}_\alpha$ which give a re-parameterization

$$\forall x: \sum_\alpha \tilde{\theta}_\alpha(x_\alpha) = \sum_\alpha \theta_\alpha(x_\alpha)$$

- Output “beliefs” (est. marginals), calculated from $\tilde{\theta}_\alpha$

$$b_\alpha(x_\alpha; \tilde{\theta}) = \frac{1}{Z_{\tilde{\theta}_\alpha}} \exp\left(\tilde{\theta}_\alpha(x_\alpha)/c_\alpha\right)$$

- $\ln Z$ / MAP approximated from beliefs
Re-parameterization

Assume we’re given these potentials for a simple 2-node graph:

<table>
<thead>
<tr>
<th>x</th>
<th>θ(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>20</td>
</tr>
<tr>
<td>(1,2)</td>
<td>10</td>
</tr>
<tr>
<td>(2,1)</td>
<td>10</td>
</tr>
<tr>
<td>(2,2)</td>
<td>20</td>
</tr>
</tbody>
</table>
Re-parameterization and Bounds

\[ \text{bound} = F(\tilde{\theta}) \]

\[ F(\tilde{\theta}^1), F(\tilde{\theta}^2) \]

\[ \begin{array}{c|c}
\theta_1 & 10 \\
0 & \\
\end{array} \]

\[ \begin{array}{c|c}
\theta_2 & 0 \\
10 & \\
\end{array} \]

\[ \begin{array}{c|c|c}
\theta_{12} & 10 & -10 \\
10 & 10 & \\
\end{array} \]

\[ \begin{array}{c|c|c}
\tilde{\theta}_1 & 10 & \\
-10 & \\
\end{array} \]

\[ \begin{array}{c|c|c}
\tilde{\theta}_2 & 0 & \\
0 & \\
\end{array} \]

\[ \begin{array}{c|c|c}
\tilde{\theta}_{12} & 10 & 0 \\
20 & 30 & \\
\end{array} \]
Bounds on the Variational Problem

Assume non-negative counting numbers $c_{\alpha}$

Local norm.

$$Z_{\tilde{\theta}_{\alpha}} = \sum_{x_{\alpha}} \exp\left(\tilde{\theta}_{\alpha}(x_{\alpha}) / c_{\alpha}\right)$$

Any re-parameterization gives bound:

Bound on approx. log-partition

$$\ln \tilde{Z} \leq \text{bound}_{\text{sum}}\left(\tilde{\theta}\right) = \sum_{\alpha} c_{\alpha} \ln Z_{\tilde{\theta}_{\alpha}}$$

Bound on MAP

$$\text{MAP} \leq \text{bound}_{\text{max}}\left(\tilde{\theta}\right) = \sum_{\alpha} \max_{x_{\alpha}} \tilde{\theta}_{\alpha}(x_{\alpha})$$

Decompose to local terms!
Re-parameterization and Bounds

Assume we’re given these potentials for a simple 2-node graph:

\[
\begin{align*}
\theta_1 & = \begin{bmatrix} 10 \\ 0 \end{bmatrix} \\
\theta_1 \theta_2 & = \begin{bmatrix} 10 & -10 \\ 10 & 10 \end{bmatrix} \\
\theta_2 & = \begin{bmatrix} 0 \\ 10 \end{bmatrix}
\end{align*}
\]

<table>
<thead>
<tr>
<th>x</th>
<th>( \theta(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>20</td>
</tr>
<tr>
<td>(1,2)</td>
<td>10</td>
</tr>
<tr>
<td>(2,1)</td>
<td>10</td>
</tr>
<tr>
<td>(2,2)</td>
<td>20</td>
</tr>
</tbody>
</table>

\[\text{MAP}=10+0+10=20\]
Re-parameterization and Bounds

Assume we’re given these potentials for a simple 2-node graph:

<table>
<thead>
<tr>
<th>x</th>
<th>θ(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>20</td>
</tr>
<tr>
<td>(1,2)</td>
<td>10</td>
</tr>
<tr>
<td>(2,1)</td>
<td>10</td>
</tr>
<tr>
<td>(2,2)</td>
<td>20</td>
</tr>
</tbody>
</table>

$$\theta_1 = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

$$\theta_2 = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$$

$$\theta_{1,2} = \begin{bmatrix} 10 & -10 \\ 10 & 10 \end{bmatrix}$$

$$\text{MAP} = 10 + 0 + 10 = 20$$

$$\text{Bound-MAP} = 10 + 10 + 10 = 30$$
Assume we’re given these potentials for a simple 2-node graph:

<table>
<thead>
<tr>
<th>x</th>
<th>θ(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>20</td>
</tr>
<tr>
<td>(1,2)</td>
<td>10</td>
</tr>
<tr>
<td>(2,1)</td>
<td>10</td>
</tr>
<tr>
<td>(2,2)</td>
<td>20</td>
</tr>
</tbody>
</table>
Re-parameterization and Bounds

Assume we’re given these potentials for a simple 2-node graph:

\[ \theta_1 \]
\[ \begin{array}{c}
10 \\
0 
\end{array} \]

\[ \theta_2 \]
\[ \begin{array}{c}
0 \\
10 
\end{array} \]

\[ \theta_{12} \]
\[ \begin{array}{cc}
10 & -10 \\
10 & 10 
\end{array} \]

<table>
<thead>
<tr>
<th>x</th>
<th>\theta(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>20</td>
</tr>
<tr>
<td>(1,2)</td>
<td>10</td>
</tr>
<tr>
<td>(2,1)</td>
<td>10</td>
</tr>
<tr>
<td>(2,2)</td>
<td>20</td>
</tr>
</tbody>
</table>
Re-parameterization and Bounds

Consider a re-parameterization:

\[ \forall x \quad \theta(x) = \tilde{\theta}(x) \]

<table>
<thead>
<tr>
<th>( x )</th>
<th>( \theta(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>20</td>
</tr>
<tr>
<td>(1,2)</td>
<td>10</td>
</tr>
<tr>
<td>(2,1)</td>
<td>10</td>
</tr>
<tr>
<td>(2,2)</td>
<td>20</td>
</tr>
</tbody>
</table>
Consider a re-parameterization:

\( \forall x \ \theta(x) = \tilde{\theta}(x) \)

<table>
<thead>
<tr>
<th></th>
<th>( \tilde{\theta}_1 )</th>
<th>( \tilde{\theta}_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>(1,2)</td>
<td>-10</td>
<td>0</td>
</tr>
<tr>
<td>(2,1)</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>(2,2)</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

Bound

- 40
- 30
- 20

MAP
Re-parameterization and Bounds

Search for re-parameterization with tightest bound

$$\forall x \theta(x) = \widetilde{\theta}(x)$$

<table>
<thead>
<tr>
<th>x</th>
<th>$\theta(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>20</td>
</tr>
<tr>
<td>(1,2)</td>
<td>10</td>
</tr>
<tr>
<td>(2,1)</td>
<td>10</td>
</tr>
<tr>
<td>(2,2)</td>
<td>20</td>
</tr>
</tbody>
</table>

Bound

40
30
20

MAP

1

2

$\tilde{\theta}_1$

10
-10

$\tilde{\theta}_2$

0
0

$\tilde{\theta}_{1,2}$

10
0

20
30
Re-parameterization and Bounds

Search for re-parameterization with tightest bound

\[ \forall x \quad \theta(x) = \tilde{\theta}(x) \]

<table>
<thead>
<tr>
<th>x</th>
<th>( \theta(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>20</td>
</tr>
<tr>
<td>(1,2)</td>
<td>10</td>
</tr>
<tr>
<td>(2,1)</td>
<td>10</td>
</tr>
<tr>
<td>(2,2)</td>
<td>20</td>
</tr>
</tbody>
</table>

Bound

40
30
20

MAP

\[ \tilde{\theta}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \]

\[ \tilde{\theta}_2 = \begin{pmatrix} 10 \\ 10 \end{pmatrix} \]

\[ \tilde{\theta}_{12} = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \]
Re-parameterization and Bounds

Search for re-parameterization with tightest bound

∀x θ(x) = \tilde{θ}(x)

<table>
<thead>
<tr>
<th>x</th>
<th>θ(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>20</td>
</tr>
<tr>
<td>(1,2)</td>
<td>10</td>
</tr>
<tr>
<td>(2,1)</td>
<td>10</td>
</tr>
<tr>
<td>(2,2)</td>
<td>20</td>
</tr>
</tbody>
</table>
Optimizing the Bounds

- Want to update many \( \tilde{\theta}_\alpha \) coordinates at once such that bound decreases.
- Updates on trees at once can be more easily characterized.
  - Sontag & Jaakkola ’09

![Diagram showing bounds labeled 20, 30, and 40, with a MAP marker at the end.]
Theorem: Bound Tightness in Trees

- Re-parameterization with max-consistent beliefs gives the tightest bound on MAP in tree region graph

\[ b_\alpha(x_\alpha; \tilde{\theta}) = \frac{1}{Z_{\tilde{\theta}_\alpha}} \exp\left(\tilde{\theta}_\alpha(x_\alpha) / c_\alpha\right) \]

max-consistency: \( \forall \alpha, \beta \in T \)

\[ \max_{x_\alpha \setminus x_\beta} b_\alpha(x_\alpha; \tilde{\theta}_\alpha) = b_\beta(x_\beta; \tilde{\theta}_\beta) \]
Theorem: Bound Tightness in Trees

- Re-parameterization with sum-consistent beliefs gives the tightest bound on approx. $\ln Z$ in any region graph
  - but easier to find in trees

sum-consistency: $\forall \alpha, \beta \in T$

$$\sum_{x_\alpha \setminus x_\beta} b_\alpha (x_\alpha ; \tilde{\theta}_\alpha) = b_\beta (x_\beta ; \tilde{\theta}_\beta)$$
1. Choose a tree subset $T$ in the region graph
Tree Consistency Bound Optimization (TCBO) Abstract Algorithm

1. Choose a tree subset $T$ in the region graph

TCBO performs coordinate descent

2. Update the variables $\tilde{\theta}_\alpha(x_\alpha)$, $\alpha \in T$ s.t.:
   - **Re-parameterization** is maintained for the whole region-graph
   - **Consistency** is enforced on beliefs of subset
Existing TCBO Algorithms

- The following maintain re-parameterization and enforce consistency on a subtree:
  - Heskes ’06 (*sum prod.*)
  - Sequential TRBP (TRW-S), Kolmogorov ’06 (*max prod.*)
  - Max Product Linear Programming (MPLP), Globerson & Jaakkola ’07 (*max prod.*)
  - Max Sum Diffusion (MSD), Werner ’07 (*max prod.*)
New TCBO Algorithms

- Exchanging \textit{max} and \textit{sum} operations replaces \textit{max} and \textit{sum} consistency
  - Heskes ’06 (\textit{sum prod.}) ➔ max
  - Sequential TRBP (TRW-S), Kolmogorov ’06 (\textit{max prod.}) ➔ sum
  - Max Product Linear Programming (MPLP), Globerson & Jaakkola ’07 (\textit{max prod.}) ➔ sum
  - Max Sum Diffusion (MSD), Werner ’07 (\textit{max prod.}) ➔ sum
A Convergent Algorithm for TRW- lnZ

- lnZ approximation:
  - Regions are trees
- Simple TRBP algorithm not guaranteed to converge
- Kolmogorov showed that in the MAP case, TRBP works for some update order
- We show: replacing max by sum will solve the lnZ cases
New TCBO Algorithms – Simple Simulation

Sum-TRBP with convergent vs. non-convergent update order applied to a 10x10 spin-glass instance
Summary

- We present an abstract algorithm (TCBO) for MAP and Log-Partition consists of
  - re-parameterization
  - consistency
- Identify existing algorithms as TCBO instances
- Obtain new instances, by exchanging sum/max operations in existing algorithms
Thank you
Appendix
Existing TCBO Algorithms – Illustrations with a 4 node graph

Will demonstrate on this graph how existing algorithms implement TCBO
Max Sum Diffusion (MSD) [Werner ‘07] as a TCBO
Max Sum Diffusion (MSD) [Werner ‘07] as a TCBO
Heskes’ Algorithm [‘04] as a TCBO
Heskes’ Algorithm ['04]
as a TCBO
Max Product Linear Programming (MPLP) [Globerson & Jaakkola ‘07]

- Originally uses region graph with negative counting numbers $c_i = 1 - d_i$
- Equivalent to Heskes’ algorithm with a different region graph

```
C=0
  1 — 2
  1 — 3
  2 — 3
  2 — 4
  3 — 4
```

```
C=-1
  1

C=-2
  2 — 3

C=-2
  2 — 4

C=-1
  3 — 4
```
MPLP [Globerson & Jaakkola ‘07] as a TCBO

C=1  C=1  C=1  C=1

C=0  C=0  C=0  C=0  C=0  C=0
MPLP [Globerson & Jaakkola ‘07] as a TCBO
Sequential Tree Reweighted BP (TRW-S) [Kolmogorov ‘06] as a TCBO
Sequential Tree Reweighted BP (TRW-S) [Kolmogorov ‘06] as a TCBO

Trees in the original graph

C=1/2

C=0

1

2

3

4

C=1/2

Trees in the original graph

C=0

1

2

3

4
Sequential Tree Reweighted BP (TRW-S) [Kolmogorov ‘06] as a TCBO
Sequential Tree Reweighted BP (TRW-S) [Kolmogorov ‘06] as a TCBO

Choose an intersection element

C=1/2

C=1/2

C=0

C=0

C=0

C=0
Sequential Tree Reweighted BP (TRW-S) [Kolmogorov ‘06] as a TCBO

Update a tree in the region graph

C=1/2

C=0