Reasoning and Decision in Probabilistic Graphical models – A Unified Approach

Qiang Liu
Thesis Defense

Dissertation Committee:
Alexander Ihler (chair)
Rina Dechter
Padhraic Symth
Outline

• Background:
  – Probabilistic graphical models
  – Inference & decision tasks

• Main Results:
  – A unified variational representation
  – Derive lots of efficient approximation algorithms
  – Experiments

• Conclusion & Future Directions
Probabilistic Graphical Models

• High dimensional probability:

\[ p(\mathbf{x}) = p(x_1, x_2, \ldots, x_n) = \frac{1}{Z} \prod_k \psi(x_{\alpha_k}) \]

e.g., \( x \in \{0, 1\}^n \)

- Example: \( p(x) = \frac{1}{Z} \psi_1(x_1) \psi_3(x_3) \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \)

\[
\begin{array}{c|cc}
X1 & 0 & 1 \\
\hline
0 & 9 & 1 \\
1 & 8 & 3 \\
\end{array}
\]

\[
\begin{array}{c|cc}
X1 \setminus X2 & 0 & 1 \\
\hline
0 & 1 & 9 \\
1 & 8 & 3 \\
\end{array}
\]

\( Z \): normalization constant
Probabilistic Graphical Models

• High dimensional probability:

\[
p(\mathbf{x}) = p(x_1, x_2, \ldots, x_n) = \frac{1}{Z} \prod_k \psi(x_{\alpha_k})
\]

\[e.g., x \in \{0, 1\}^n\]

\[Z: \text{normalization constant}\]

Undirected graph: (Markov random field)

Directed graph: (Bayesian Network)
Key Inference Types

- Optimization (max)
- Marginalization (sum)
- Marginal MAP (max-sum)
- Decision Making

Most probable configuration (Maximum a posteriori (MAP) estimate):

\[
x^* = \arg \max_x p(x) = \arg \max_x \prod_k \psi_k(x_{\alpha_k})
\]

\[
x \in \{0, 1\}^n
\]
Key Inference Types

- Optimization (max)
- Marginalization (sum)
- Marginal MAP (max-sum)
- Decision Making

Normalization constant (a.k.a. partition function), or marginal probabilities:

\[ Z = \sum_x \prod_k \psi_k(x_{\alpha_k}) \quad \text{or} \quad p(x_i) = \frac{1}{Z} \sum_{x\setminus x_i} \prod_k \psi_k(x_{\alpha_k}) \]
Key Inference Types

- Optimization (max)
- Marginalization (sum)
- Marginal MAP (max-sum)
- Decision Making

Maximize marginal probability, or expected objective w.r.t. “latent variables”:

\[ x_B^* = \arg \max_{x_B} \sum_{x_A} p([x_A, x_B]) = \arg \max_{x_B} \sum_{x_A} \prod_k \psi_k(x_{\alpha_k}) \]

\[ x = [x_A, x_B] \]

Latent variables:

Missing labels:

Black: missing labels
Key Inference Types

- Optimization (max)
- Marginalization (sum)
- Marginal MAP (max-sum)
- Decision Making

Sequential Decision under uncertainty (stochastic programming):

$$\max_{x_{D_1}} \sum_{x_{R_1}} \cdots \max_{x_{D_k}} \sum_{x_{R_k}} \prod_{k} \psi_k(x_{\alpha_k})$$

- Sum: average random variables
- Max: optimize decision variables
Key Inference Types

- Optimization (max)
- Marginalization (sum)
- Marginal MAP (max-sum)
- Decision Making

Represented as influence diagrams (a.k.a. decision networks):

- : Random variables $x_R$
- : Decision variables $x_D$
- : Utility function: $U([x_R, x_D])$

Maximize expected utility (MEU):

$$\max_{\delta_D} \mathbb{E}(U(x) \mid \delta_D), \quad \delta_D : \text{decision policies}$$

$$= \max_{x_{D_1}} \sum_{x_{R_1}} \cdots \max_{x_{D_k}} \sum_{x_{R_k}} U(x) p(x_R \mid x_D)$$

(Assume: single agent; perfect memory)
Key Inference Types

- Optimization (max)
- Marginalization (sum)
- Marginal MAP (max-sum)
- Decision Making

Represented as influence diagrams (a.k.a. decision networks):

〇: Random variables $x_R$
□: Decision variables $x_D$
◇: Utility function: $U([x_R, x_D])$

Maximize expected utility (MEU):

$$\max_{\delta_D} \mathbb{E}(U(x) \mid \delta_D), \quad \delta_D: \text{decision policies}$$

(Multi-agent; limited communication: no sequential elimination form)
# Key Inference Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Maximization</strong></td>
<td>$\max_x \prod_k \psi_k(x_{\alpha_k})$</td>
</tr>
<tr>
<td><strong>Sum/Counting</strong></td>
<td>$\sum_x \prod_k \psi_k(x_{\alpha_k})$</td>
</tr>
<tr>
<td><strong>Max-Sum</strong></td>
<td>$\max_{x_B} \sum_{x_A} \prod_k \psi_k(x_{\alpha_k})$</td>
</tr>
<tr>
<td><strong>Decision Making (sequential)</strong></td>
<td>$\max_{x_{B_1}} \sum_{x_{A_1}} \cdots \max_{x_{B_k}} \sum_{x_{A_k}} \prod_k \psi_k(x_{\alpha_k})$</td>
</tr>
<tr>
<td><strong>Decision Making (multi-agent)</strong></td>
<td>![Image] (No sequential elimination form)</td>
</tr>
</tbody>
</table>
Pure sum or max are (relatively) “easy”

- Dynamic programming.

\[
Z = \sum_{x_1, x_2, x_3} \psi_3(x_3) \psi_{23}(x_2, x_3) \psi_{12}(x_1, x_2)
\]

\[
= \sum_{x_2, x_3} \psi_3(x_3) \psi_{23}(x_2, x_3) \sum_{x_1} \psi_{12}(x_1, x_2)
\]

\[
= \sum_{x_2, x_3} \psi_3(x_3) \psi_{23}(x_2, x_3) m_{1\rightarrow 2}(x_2)
\]

\[
= \sum_{x_3} \psi_3(x_3) \sum_{x_2} \psi_{23}(x_2, x_3) m_{1\rightarrow 2}(x_2)
\]

\[
= \sum_{x_3} \psi_3(x_3) m_{2\rightarrow 3}(x_3)
\]

“message passing”
Pure sum or max are (relatively) easy!

- On general trees
  - Linear complexity

- Sum or max-product Belief Propagation (BP):

\[
m_{i \rightarrow j}(x_{j}) \propto \max_{x} \left( \psi_{ij}(x_{i}, x_{j}) \prod_{k \neq j} m_{k \rightarrow i}(x_{i}) \right) \quad \text{or} \quad \sum_{x} m_{i \rightarrow j}(x_{j}) \]

“Message passing”
Marginal MAP & Decision are challenging!

- NP-hard even on trees:

\[
\begin{align*}
\text{max: } & x_B \\
\text{sum: } & x_A
\end{align*}
\]

\[
\max_{x_B} \sum_{x_A} p([x_A, x_B])
\]

- Can not exchange max and sum (\(\max \sum \neq \sum \max\) )
Marginal MAP & Decision are challenging!

- NP-hard even on trees:

\[
\text{max: } x_B \\
\text{sum: } x_A
\]

- Can not exchange max and sum (\(\max \sum \neq \sum \max\))

\[
\begin{align*}
\max_{x_B} \sum_{x_A} p([x_A, x_B]) &= \max_{x_B} \hat{\psi}(x_B)
\end{align*}
\]
Use Joint Optimization instead?

$$[x_A^*, x_B^*] = \arg \max_x p([x_A, x_B])$$
Use Joint Optimization instead?

\[
[x^*_A, x^*_B] = \arg \max_x p([x_A, x_B])
\]

- \(x \in \{\text{rainy, sunny}\}: \text{the weather of Irvine}\)
- \(y \in \{\text{walk, drive}\}: \text{whether I walk or drive to office}\)
- Query: is Irvine more likely to be sunny?

\[
p(x, y)
\]

<table>
<thead>
<tr>
<th></th>
<th>walk</th>
<th>drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>rainy</td>
<td>0.05</td>
<td>0.35</td>
</tr>
<tr>
<td>sunny</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>
Use Joint Optimization instead?

\[ [x_A^*, x_B^*] = \arg \max_x p([x_A, x_B]) \]

- \( x \in \{\text{rainy, sunny}\} \): the weather of Irvine
- \( y \in \{\text{walk, drive}\} \): whether I walk or drive to office
- Query: is Irvine more likely to be sunny?

\[ p(x, y) \]

<table>
<thead>
<tr>
<th></th>
<th>walk</th>
<th>drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>rainy</td>
<td>0.05</td>
<td>0.35</td>
</tr>
<tr>
<td>sunny</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>

**Marginal MAP:** \( x^* = \arg \max_x p(x) \)

\[ x^* = \text{sunny} \]

\[ \square \]

Paradox: My behavior influence the weather!
Use Joint Optimization instead?

\[ [x_A^*, x_B^*] = \arg \max_x p([x_A, x_B]) \]

- \( x \in \{\text{rainy, sunny}\} \): the weather of Irvine
- \( y \in \{\text{walk, drive}\} \): whether I walk or drive to office
- Query: is Irvine more likely to be sunny?

<table>
<thead>
<tr>
<th></th>
<th>walk</th>
<th>drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>rainy</td>
<td>0.05</td>
<td>0.35</td>
</tr>
<tr>
<td>sunny</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>

**Marginal MAP:** \( x^* = \arg \max_x p(x) \)

\( x^* = \text{sunny} \) ✗

**Joint MAP:** \( (x^*, y^*) = \arg \max_{x,y} p(x, y) \)

\( (x^*, y^*) = [\text{rainy, drive}] \) ✗

Paradox: My behavior influence the weather!
Two Types of Approximate Methods

• Elimination-based approximation:
  – Directly approximate the sum & max operators
  – Mini-bucket, weighted mini-bucket (ICML11)

• Variational approximation:
  – Reframe into continuous (hopefully linear or convex) optimization
  – Approximating the continuous optimization
Outline

• Background:
  – Probabilistic graphical models
  – Inference & decision tasks

• Main Results:
  – A unified variational representation
  – Derive lots of efficient approximation algorithms
  – Experiments

• Conclusion & Future Directions
Variational Algorithms

- Reframe into continuous (hopefully linear or convex) optimization
  
  - Notation:

  \[
  p(x) = \frac{1}{Z} \prod_k \psi_k(x_{\alpha_k}) = \frac{1}{Z} \exp(\theta(x))
  \]

  \[\Rightarrow \theta(x) = \log \prod_k \psi_k(x_{\alpha_k})\]
Variational Algorithms

• Reframe into continuous (hopefully linear or convex) optimization
  
  – Maximization

  \[
  \max_x \theta(x) = \max_{q \in \mathcal{M}} \mathbb{E}_q[\theta(x)] \quad \text{(maximum: } q = 1(x^*))
  \]

  \(\mathcal{M}\): The set of joint distributions on \(x\):

  \[
  \theta(x) = \log \prod_k \psi_k(x_{\alpha_k})
  \]

  – Summation:

  \[
  \log \sum_x \exp(\theta(x)) = \max_{q \in \mathcal{M}} \left\{ \mathbb{E}_q[\theta(x)] + H(x ; q) \right\} \quad \text{(maximum: } q = p)\]

  Entropy: \(- \sum_x q(x) \log q(x)\)

  – \(\mathcal{M}\) and \(H(x ; q)\) are intractable!
Variational Approximation

• Approximate $M$ and $H(x ; q)$
• Derives lots of “message passing” algorithms (e.g., Wainwright & Jordan 08)
  – e.g., Bethe approximation yields loopy BP (Yedidia et al 2000)

$$m_{i \rightarrow j}(x_j) \propto \max_x \psi_{i,j}(x_i, x_j) \prod_{k \neq j} m_{k \rightarrow i}(x_i)$$

  – Exact on trees
  – Approximate on loopy graphs (Pearl 88)
<table>
<thead>
<tr>
<th></th>
<th>Primal Form</th>
<th>Dual Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>$\max_x \exp(\theta(x))$</td>
<td>$\max_{q \in \mathcal{M}} \mathbb{E}_q[\theta(x)]$</td>
</tr>
<tr>
<td>Sum</td>
<td>$\sum_x \exp(\theta(x))$</td>
<td>$\max_{q \in \mathcal{M}} {\mathbb{E}_q[\theta(x)] + H(x; q)}$</td>
</tr>
<tr>
<td>Max-sum</td>
<td>$\max_{x_B} \sum_{x_A} \exp(\theta(x))$</td>
<td></td>
</tr>
<tr>
<td>Decision Making (perfect recall)</td>
<td>$\max_{x_{B_1}} \sum_{x_{A_1}} \cdots \max_{x_{B_k}} \sum_{x_{A_k}} \exp(\theta(x))$</td>
<td></td>
</tr>
<tr>
<td>Decision Making (imperfect recall)</td>
<td><img src="image" alt="Decision Making" /></td>
<td></td>
</tr>
</tbody>
</table>
Variational Representations (UAI11, JMLR13)

For marginal MAP: \( Z_{AB} = \max_{x_B} \sum_{x_A} \exp(\theta(x)) \)

\[
\log Z_{AB} = \max_{q \in \mathcal{M}} \left\{ \mathbb{E}_q[\theta(x)] + H(x_A|x_B ; q) \right\}
\]

\[
= \max_{q \in \mathcal{M}} \left\{ \mathbb{E}_q[\theta(x)] + H(x ; q) - H(x_B ; q) \right\}
\]

Same as fully summation

“truncating” the entropies of the max nodes; enforcing deterministic choice

Joint max \((A = \emptyset)\) \(\iff\) \(\max_{q \in \mathcal{M}} \{ \mathbb{E}_q[\theta(x)] \} \)

Joint sum \((B = \emptyset)\) \(\iff\) \(\max_{q \in \mathcal{M}} \{ \mathbb{E}_q[\theta(x)] + H(x ; q) \} \)
Variational Representations (UAI12)

- Sequential decision making:

\[
Z = \max_{D_1} \sum_{R_1} \cdots \max_{D_k} \sum_{R_k} \exp(\theta(\mathbf{x}))
\]

\[
\log Z = \max_{q \in \mathcal{M}} \{ \mathbb{E}_q[\theta(\mathbf{x})] + H(\mathbf{x} ; q) - \sum_i H(x_{D_i} | x_{\text{pa}(D_i)} ; q) \}
\]

Same as fully summation

“truncating” conditional entropies of decision nodes

- Directly extends to more general (multi-agent) decision making (details in UAI12)
<table>
<thead>
<tr>
<th>Primal Form</th>
<th>Dual Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>$\max_x \exp(\theta(x))$</td>
</tr>
<tr>
<td>Sum</td>
<td>$\sum_x \exp(\theta(x))$</td>
</tr>
<tr>
<td>Max-sum</td>
<td>$\max_{x_B} \sum_{x_A} \exp(\theta(x))$</td>
</tr>
<tr>
<td>Decision Making (sequential)</td>
<td>$\max_{x_{B_1}} \sum_{x_{A_1}} \cdots \max_{x_{B_k}} \sum_{x_{A_k}} \exp(\theta(x))$</td>
</tr>
<tr>
<td>Decision Making (multi-agent)</td>
<td>$\max_{x_{B_1}} \sum_{x_{A_1}} \cdots \max_{x_{B_k}} \sum_{x_{A_k}} \exp(\theta(x))$</td>
</tr>
</tbody>
</table>

$$\max_{q \in \mathcal{M}} \left\{ \mathbb{E}_q[\theta(x)] + H(x; q) - \sum_i H(x_{A_i} | x_{pa(A_i)} ; q) \right\}$$

Derive all kinds of algorithms:
- Dual decomposition
- ADMM
- Tree reweighted
- Mean field
- Kikuchi
- Entropy decomposition
- Expectation propagation
“Mixed-product” BP (for marginal MAP)

• “Mixed-product” BP: sends different messages between different types of nodes

\[ \text{Max } \Rightarrow \text{Max: } \begin{align*} m_{i \rightarrow j} & \propto \max_{x_i} \psi_{ij} \prod_{k \neq j} m_{k \rightarrow i} \end{align*} \]

\[ \text{(Max-Product)} \]

\[ \text{Sum } \Rightarrow \text{Sum } \cup \text{ Max: } \begin{align*} m_{i \rightarrow j} & \propto \sum_{x_i} \psi_{ij} \prod_{k \neq j} m_{k \rightarrow i} \end{align*} \]

\[ \text{(Sum-Product)} \]

\[ \text{Max } \Rightarrow \text{Sum: } \begin{align*} m_{i \rightarrow j} & \propto \sum_{x_i \in \mathcal{X}_i^*} \psi_{ij} \prod_{k \neq j} m_{k \rightarrow i} \end{align*} \]

\[ \text{(Argmax-Product)} \]

• Intuition

  – Optimize \( x_i \), pass the result to the sum part

• Optimality guarantees (details in UAI11, UAI12, JMRL13)
Experiments (marginal MAP)

- Hidden Markov Chain (of length 10):
  - max: \( x_B \)
  - sum: \( x_A \)

\[
\max_{x_B} \sum_{x_A} p([x_A, x_B])
\]

```
\begin{align*}
\text{Max}\: \text{product BP} \\
\text{Sum}\: \text{product BP} \\
\text{Jiang et al 11} \\
\text{Proximal BP} \\
\text{Local search (SamIam)}
\end{align*}
```
Experiments (marginal MAP)

- Two diagnostic Bayesian networks from UAI08 challenge:
  - ~200-300 nodes, ~300-600 edges
  - Randomly select a set of max nodes

![Diagram of two diagnostic Bayesian networks](image)

- **(b). Diagnostic BN-1**
- **(c). Diagnostic BN-2**

**Figures:***

- Proximal BP
- Mixed-product BP
- Local search (SamIam)

**Acknowledgments**
We thank Arthur Choi for providing help on SamIam. This work was supported in part by NSF grant IIS-1065618 and a Microsoft Research Ph.D Fellowship.

**References**

Structured Prediction with Missing labels

(ICML14, experiments done by Wei Ping)

Accuracy (%):

<table>
<thead>
<tr>
<th>MSRC Data</th>
<th>MSSVM</th>
<th>LSSVM</th>
<th>HCRFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building</td>
<td>72.4</td>
<td>70.7</td>
<td>71.7</td>
</tr>
<tr>
<td>Grass</td>
<td>89.7</td>
<td>88.9</td>
<td>88.3</td>
</tr>
<tr>
<td>Sky</td>
<td>88.3</td>
<td>85.6</td>
<td>88.2</td>
</tr>
<tr>
<td>Tree</td>
<td>71.9</td>
<td>71.0</td>
<td>70.1</td>
</tr>
<tr>
<td>Car</td>
<td>70.8</td>
<td>69.4</td>
<td>70.2</td>
</tr>
</tbody>
</table>

Marginal MAP

\[ x = \arg \max_x \sum_y f(x, y) \]

Joint MAP

\[ [x, y] = \arg \max_{x, y} f(x, y) \]
Collaborative Decision Making (UAI12)

- Also derive message passing algorithms for decision
- Distributed detection:
  - Sensors make local predictions
  - Send signals to help other sensors
  - Reward = accuracy - signal cost

![Graph showing expected reward vs. signal unit cost]

- Proximal BP
- Mixed-product BP
- Single Policy Update

$h_i$: Hidden variables
$v_i$: Local measurements
$d_i$: Prediction decisions
$s_i$: Signal decisions
$u_i$: Reward utilities
$c_i$: Signal cost utilities
## Conclusion

<table>
<thead>
<tr>
<th></th>
<th>Primal Form</th>
<th>Dual Form</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max</strong></td>
<td>[ \max_x \exp(\theta(x)) ]</td>
<td>[ \max_{q \in \mathcal{M}} \left{ \mathbb{E}<em>q[\theta(x)] + H(x ; q) \right} - \sum_i H(x</em>{A_i}</td>
</tr>
<tr>
<td><strong>Sum</strong></td>
<td>[ \sum_x \exp(\theta(x)) ]</td>
<td></td>
</tr>
<tr>
<td><strong>Max-sum</strong></td>
<td>[ \max_{x_B} \sum_{x_A} \exp(\theta(x)) ]</td>
<td></td>
</tr>
<tr>
<td><strong>Decision Making (sequential)</strong></td>
<td>[ \max_{x_{B_1}} \sum_{x_{A_1}} \cdots \max_{x_{B_k}} \sum_{x_{A_k}} \exp(\theta(x)) ]</td>
<td>Derive all kinds of algorithms:</td>
</tr>
<tr>
<td><strong>Decision Making (multi-agent)</strong></td>
<td><img src="https://via.placeholder.com/150" alt="Image" /></td>
<td><img src="https://via.placeholder.com/150" alt="Diagram" /></td>
</tr>
</tbody>
</table>

**Derive all kinds of algorithms:**

- Dual decomposition
- ADMM
- Tree reweighted
- Convex relaxation
- Entropy decomposition
- Mean field
- Expectation propagation

![Image](https://via.placeholder.com/150)
Thank you!