Inference in Graphical Models
Variable Elimination and Message Passing Algorithm

Le Song

Machine Learning II: Advanced Topics
CSE 8803ML, Spring 2012
Conditional Independence Assumptions

- **Local Markov Assumption**
  \[ X \perp \text{Nondescendant}_X | Pa_X \]

- **Global Markov Assumption**
  \[ A \perp B | C, \text{sep}_G(A, B; C) \]

**Diagrams:**
- **BN** (Bayesian Network)
- **MN** (Markov Network)
- **Undirected Tree**
- **Undirected Chordal Graph**

**Other Concepts:**
- **Moralize**
- **Triangulate**
Distribution Factorization

- **Bayesian Networks (Directed Graphical Models)**
  
  \[ I - \text{map: } I_l(G) \subseteq I(P) \]
  
  \[ \iff \]
  
  \[ P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid P a_{X_i}) \]

- **Markov Networks (Undirected Graphical Models)**
  
  *strictly positive* \( P \), \( I - \text{map: } I(G) \subseteq I(P) \)
  
  \[ \iff \]
  
  \[ P(X_1, \ldots, X_n) = \frac{1}{Z} \prod_{i=1}^{m} \Psi_i(D_i) \]

  \[ Z = \sum_{x_1, x_2, \ldots, x_n} \prod_{i=1}^{m} \Psi_i(D_i) \]

- **Conditional Probability Tables (CPTs)**
- **Clique Potentials**
- **Normalization (Partition Function)**
- **Maximal Clique**
- **Clique**
- **Strictly Positive**
Inference in Graphical Models

- Graphical models give compact representations of probabilistic distributions $P(X_1, ..., X_n)$ (n-way tables to much smaller tables)

- How do we answer queries about $P$?

- We use inference as a name for the process of computing answers to such queries
Query Type 1: Likelihood

Most queries involve evidence
- Evidence $e$ is an assignment of values to a set $E$ variables
- Evidence are observations on some variables
- Without loss of generality $E = \{X_{k+1}, \ldots, X_n\}$

Simplest query: compute probability of evidence
- $P(e) = \sum_{x_1} \ldots \sum_{x_k} P(x_1, \ldots, x_k, e)$
- This is often referred to as computing the likelihood of $e$
Query Type 2: Conditional Probability

- Often we are interested in the conditional probability distribution of a variable given the evidence

\[ P(X|e) = \frac{P(X, e)}{P(e)} = \frac{P(X, e)}{\sum_x P(X = x, e)} \]

- It is also called a posteriori belief in \( X \) given evidence \( e \)

- We usually query a subset \( Y \) of all variables \( \mathcal{X} = \{Y, Z, e\} \) and “don’t care” about the remaining \( Z \)

\[ P(Y|e) = \sum_z P(Y, Z = z|e) \]

- Take all possible configuration of \( Z \) into account
- The processes of summing out the unwanted variable \( Z \) is called marginalization
Query Type 2: Conditional Probability Example

Sum over this set of variables

Interested in the conditionals for these variables

Interested in the conditionals for these variables

Sum over this set of variables
**Application of a posteriori Belief**

- Prediction: what is the probability of an outcome given the starting condition
  - The query node is a descendent of the evidence

![Diagram of network with nodes A, B, C, and directed edges A → B → C]

- Diagnosis: what is the probability of disease/fault given symptoms
  - The query node is an ancestor of the evidence

![Diagram of network with nodes A, B, C, and directed edges A → B → C]

- Learning under partial observations (Fill in the unobserved)

- Information can flow in either direction
  - Inference can combine evidence from all parts of the networks
Query Type 3: Most Probable Assignment

- Want to find the most probably joint assignment for some variables of interests

- Such reasoning is usually performed under some given evidence \( e \), and ignoring (the values of other variables) \( Z \)
  - Also called maximum a posteriori (MAP) assignment for \( Y \)
    - \( MAP(Y|e) = arg\max_y P(Y|e) = arg\max_y \sum_z P(Y, Z = z|e) \)

Interested in the most probable values for these variables

Sum over this set of variables
Application of MAP assignment

- Classification
  - Find most likely label, given the evidence

- Explanation
  - What is the most likely scenario, given the evidence

Cautionary note:
- The MAP assignment of a variable dependence on its context – the set of variables being jointly queried

Example:
- MAP of $(X, Y)$?
  - $(0, 0)$
- MAP of $X$?
  - $1$

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>P(X,Y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.35</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>P(X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.4</td>
</tr>
<tr>
<td>1</td>
<td>0.6</td>
</tr>
</tbody>
</table>
Computing the a posteriori belief $P(X|e)$ in a GM is NP-hard in general.

Hardness implies we cannot find a general procedure that works efficiently for arbitrary GMs.

- For particular families of GMs, we can have provably efficient procedures.
  - eg. trees

- For some families of GMs, we need to design efficient approximate inference algorithms.
  - eg. grids
Approaches to inference

- **Exact inference algorithms**
  - Variable elimination algorithm
  - Message-passing algorithm (sum-product, belief propagation algorithm)
  - The junction tree algorithm

- **Approximate inference algorithms**
  - Sampling methods/Stochastic simulation
  - Variational algorithms
Marginalization and Elimination

A metabolic pathway:
- What is the likelihood protein $E$ is produced

Query: $P(E)$
- $P(E) = \sum_d \sum_c \sum_b \sum_a P(a, b, c, d, E)$

Using graphical models, we get
- $P(E) = \sum_d \sum_c \sum_b \sum_a P(a)P(b|a)P(c|b)P(d|c)P(E|d)$

Naïve summation needs to enumerate over an exponential number of terms
Rearranging terms and the summations

\[ P(E) \]
\[ = \sum_{d} \sum_{c} \sum_{b} \sum_{a} P(a)P(b|a)P(c|b)P(d|c)P(E|d) \]
\[ = \sum_{d} \sum_{c} \sum_{b} P(c|b)P(d|c)P(E|d) \left( \sum_{a} P(a)P(b|a) \right) \]
Now we can perform innermost summation efficiently.

\[ P(E) \]
\[
= \sum_d \sum_c \sum_b P(c|b)P(d|c)P(E|d) \left( \sum_a P(a)P(b|a) \right) 
\]
\[
= \sum_d \sum_c \sum_b P(c|b)P(d|c)P(E|d)P(b) 
\]

The innermost summation eliminates one variable from our summation argument at a local cost.
Elimination in Chains (cont.)

Rearranging and then summing again, we get

\[ P(E) = \sum_{d} \sum_{c} \sum_{b} P(c|b)P(d|c)P(e|d)P(b) \]

\[ = \sum_{d} \sum_{c} P(d|c)P(E|d) \left( \sum_{b} P(c|b)P(b) \right) \]

\[ = \sum_{d} \sum_{c} P(d|c)P(E|d)P(c) \]

Equivalent to matrix-vector multiplication, |Val(B)| \* |Val(C)|
Elimination in Chains (cont.)

- Eliminate nodes one by one all the way to the end
  \[
P(E) = \sum_d P(E|d)P(d)
  \]

- Computational Complexity for a chain of length \(k\)
  - Each step costs \(O(|\text{Val}(X_i)| \times |\text{Val}(X_{i+1})|)\) operations: \(O(kn^2)\)
  - \(\Psi(X_i) = \sum_{x_{i-1}} P(X_i|X_{i-1})P(X_{i-1})\)
  - Compare to naïve summation: \(O(n^k)\)
    - \(\sum_{x_1} \cdots \sum_{x_{k-1}} P(x_1, \ldots, X_k)\)
Undirected Chains

Rearrange terms, perform local summation ...

\[ P(E) \]
\[ = \sum_d \sum_c \sum_b \sum_a \frac{1}{Z} \Psi(b, a)\Psi(c, b)\Psi(d, c)\Psi(E, d) \]
\[ = \frac{1}{Z} \sum_d \sum_c \sum_b \Psi(c, b)\Psi(d, c)\Psi(E, d) \left( \sum_a \Psi(b, a) \right) \]
\[ = \frac{1}{Z} \sum_d \sum_c \sum_b \Psi(c, b)\Psi(d, c)\Psi(E, d)\Psi(b) \]
The Sum-Product Operation

- During inference, we try to compute an expression
  
  \[ \sum_z \prod_{\psi \in \mathcal{F}} \psi \]

  - \( \mathcal{X} = \{X_1, \ldots, X_n\} \) the set of variables
  - \( \mathcal{F} \) a set of factors such that for each \( \psi \in \mathcal{F}, \text{Scope}[\psi] \in \mathcal{X} \)
  - \( \mathcal{Y} \subset \mathcal{X} \) a set of query variables
  - \( \mathcal{Z} = \mathcal{X} - \mathcal{Y} \) the variables to eliminate

- The result of eliminating the variables in \( \mathcal{Z} \) is a factor
  
  \[ \tau(\mathcal{Y}) = \sum_z \prod_{\psi \in \mathcal{F}} \psi \]

  - This factor does not necessarily correspond to any probability or conditional probability in the network.

- \( P(\mathcal{Y}) = \frac{\tau(\mathcal{Y})}{\sum \tau(\mathcal{Y})} \)
Inference via Variable Elimination

General Idea

- Write query in the form
  \[
P(X_1, e) = \sum_{x_n} \cdots \sum_{x_3} \sum_{x_2} \prod_{i} P(x_i | P_{a_{x_i}})
  \]
  - The sum is ordered to suggest an elimination order

- Then iteratively
  - Move all irrelevant terms outside of innermost sum
  - Perform innermost sum, getting a new term
  - Insert the new term into the product

- Finally renormalize
  \[
P(X_1 | e) = \frac{\tau(X_1, e)}{\sum_{x_1} \tau(X_1, e)}
  \]
A more complex network

- A food web

- What is the probability \( P(A|H) \) that hawks are leaving given that the grass condition is poor?
Example: Variable Elimination

- Query: $P(A|h)$, need to eliminate $B, C, D, E, F, G, H$

- Initial factors
  - $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$

- Choose an elimination order: $H, G, F, E, D, C, B (<)$

- Step 1: Eliminate $G$
  - Conditioning (fix the evidence node on its observed value)
  - $m_h(e,f) = P(H = h|e,f)$
Example: Variable Elimination

- Query: $P(A|h)$, need to eliminate $B, C, D, E, F, G$

- Initial factors
  - $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$

- Step 2: Eliminate $G$
  - Compute $m_g(e) = \sum_g P(g|e) = 1$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_g(e)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$
Example: Variable Elimination

- Query: $P(A|h)$, need to eliminate $B, C, D, E, F$

- Initial factors
  
  $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
  
  $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$
  
  $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$

- Step 3: Eliminate $F$
  
  Compute $m_f(e,a) = \sum_f P(f|a)m_h(e,f)$
  
  $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(e,a)$
Example: Variable Elimination

- Query: $P(A|h)$, need to eliminate $B, C, D, E$

- Initial factors
  - $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(a,e)$

- Step 3: Eliminate $E$
  - Compute $m_e(a,c,d) = \sum_e P(e|c,d)m_f(a,e)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)m_e(a,c,d)$
Example: Variable Elimination

- Query: $P(A|h)$, need to eliminate $B, C, D$

- Initial factors
  
  \[
  P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f) \\
  \Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f) \\
  \Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f) \\
  \Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(a,e) \\
  \Rightarrow P(a)P(b)P(c|b)P(d|a)m_e(a,c,d)
  \]

- Step 3: Eliminate $D$
  
  Compute $m_d(a, c) = \Sigma_d P(d|a)m_e(a, c, d)$
  
  $\Rightarrow P(a)P(b)P(c|b)m_d(a, c)$
Example: Variable Elimination

- Query: $P(A|h)$, need to eliminate $B, C$

- Initial factors
  - $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)m_f(a,e)$
  - $\Rightarrow P(a)P(b)P(c|b)m_e(a,c,d)$
  - $\Rightarrow P(a)P(b)P(c|b)m_d(a,c)$

- Step 3: Eliminate $C$
  - Compute $m_c(a,b) = \sum_c P(c|b)m_d(a,c)$
  - $\Rightarrow P(a)P(b)m_c(a,b)$
Example: Variable Elimination

Query: $P(A|h)$, need to eliminate $B$

Initial factors
- $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
- $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$
- $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$
- $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(a,e)$
- $\Rightarrow P(a)P(b)P(c|b)P(d|a)m_e(a,c,d)$
- $\Rightarrow P(a)P(b)P(c|b)m_d(a,c)$
- $\Rightarrow P(a)P(b)m_c(a,b)$

Step 3: Eliminate $C$
- Compute $m_b(a) = \sum_b P(b)m_c(a,b)$
- $\Rightarrow P(a)m_b(a)$
Example: Variable Elimination

- Query: $P(A|h)$, need to renormalize over $A$

Initial factors

- $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
- $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$
- $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$
- $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(a,e)$
- $\Rightarrow P(a)P(b)P(c|b)P(d|a)m_e(a,c,d)$
- $\Rightarrow P(a)P(b)P(c|b)m_d(a,c)$
- $\Rightarrow P(a)P(b)m_c(a,b)$
- $\Rightarrow P(a)m_b(a)$

Step 3: renormalize

- $P(a,h) = P(a)m_b(a)$, compute $P(h) = \sum_a P(a)m_b(a)$
- $\Rightarrow P(a|h) = \frac{P(a)m_b(a)}{\sum_a P(a)m_b(a)}$
Complexity of variable elimination

- Suppose in one elimination step we compute
  \[ m_x(y_1, ..., y_k) = \sum_x m'_x(x, y_1, ..., y_k) \]
  \[ m'_x(x, y_1, ..., y_k) = \prod_{i=1}^{k} m_i(x, y_{c_i}) \]

- This requires
  \[ k \times |Val(X)| \times \prod_i |Val(Y_{c_i})| \] multiplications
    - For each value of \( x, y_1, ..., y_k \), we do \( k \) multiplications
  \[ |Val(X)| \times \prod_i |Val(Y_{c_i})| \] additions
    - For each value of \( y_1, ..., y_k \), we do \( |Val(X)| \) additions

- Complexity is exponential in the number of variables in the intermediate factor
Recall that induced dependency during marginalization is captured in elimination cliques
- Summation $\Leftrightarrow$ Elimination
- Intermediate term $\Leftrightarrow$ Elimination cliques

Can this lead to a generic inference algorithm?
Tree Graphical Models

**Undirected** tree: a unique path between any pair of nodes

**Directed** tree: all nodes except the root have exactly one parent
Equivalence of directed and undirected trees

Any undirected tree can be converted to a directed tree by choosing a root node and directing all edges away from it.

A directed tree and the corresponding undirected tree make the conditional independence assertions.

Parameterization are essentially the same

**Undirected tree:** \( P(X) = \frac{1}{Z} \prod_{i \in V} \Psi(X_i) \prod_{(i,j) \in E} \Psi(X_i, X_j) \)

**Directed tree:** \( P(X) = P(X_r) \prod_{(i,j) \in E} P(X_j | X_i) \)

**Equivalence:** \( \Psi(X_i) = P(X_r), \Psi(X_i, X_j) = P(X_j | X_i), Z = 1, \Psi(X_i) = 1 \)
Recall Variable Elimination Algorithm
- Choose an ordering in which the query node $f$ is the final node
- Eliminate node $i$ by removing all potentials containing $i$, take sum/product over $x_i$
- Place the resultant factor back

For a Tree graphical model:
- Choose query node $f$ as the root of the tree
- View tree as a directed tree with edges pointing towards $f$
- Elimination of each node can be considered as message-passing directly along tree branches, rather than on some transformed graphs
- Thus, we can use the tree itself as a data-structure to inference
Let $m_{ij}(X_i)$ denote the factor resulting from eliminating variables from below up to $i$, which is a function $X_i$

- $m_{ji}(X_i) = \sum_{x_j}(\Psi(x_j)\Psi(X_i, x_j) \prod_{k \in N(j) \setminus i} m_{kj}(x_j))$

- This is like a message sent from $j$ to $i$

$$P(x_f) \propto \Psi(x_f) \prod_{e \in N(f)} m_{ef}(x_f)$$

$m_{ef}(x_f)$ represents a belief on $x_f$ from $x_e$