Graphical models

Aim to represent complex multivariate probabilistic models

- multivariate -> cover multiple random variables

\[
P(\mathbf{X}) = P(X_1, X_2, \ldots, X_d) \\
p(\mathbf{X}) = p(X_1, X_2, \ldots, X_d)
\]

- Parametric distribution models:
  - Bernoulli (outcome of coin flip)
  - Binomial (outcome of multiple coin flips)
  - Multinomial (outcome of die)
  - Poisson
  - Exponential
  - Gamma distribution
  - Gaussian (this one is multivariate)
Modeling complex multivariate distributions

How to model/parameterize complex multivariate distributions \( P(X) \) with a large number of variables?

**One solution:**
- Decompose the distribution. Reduce the number of parameters, using some form of independence.

**Two graphical models:**
- **Bayesian belief networks (BBNs)**
- **Markov Random Fields (MRFs)**

**Learning of these models** relies on the decomposition.

---

**Bayesian belief network.**

1. **Directed acyclic graph**
   - **Nodes** = random variables
   - **Links** = direct (causal) dependencies between variables
     - Missing links encode independences
Bayesian belief network.

2. Local conditional distributions
   - relate variables and their parents

Bayesian belief network.
Full joint distribution in BBNs

- Parameter complexity problem
  - In the BBN the full joint distribution is defined as:
  $$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | pa(X_i))$$
  - What did we save? # of parameters of the full joint:

Example:
Assume the following assignment of values to random variables:
$$B = T, E = T, A = T, J = T, M = F$$
Then its probability is:

- What did we save? # of parameters of the full joint:

Full joint distribution is defined in terms of local conditional distributions (obtained via the chain rule):
Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:
  \[ P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid pa(X_i)) \]
- What did we save?

**Alarm example:** binary (True, False) variables

# of parameters of the full joint:

\[ 2^5 = 32 \]

One parameter depends on the rest:

\[ 2^5 - 1 = 31 \]

# of parameters of the BBN:

\[ ? \]

---

**Bayesian belief network: parameters count**

<table>
<thead>
<tr>
<th>( P(B) )</th>
<th>2</th>
<th>( P(E) )</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>T</strong></td>
<td></td>
<td><strong>T</strong></td>
<td></td>
</tr>
<tr>
<td><strong>F</strong></td>
<td></td>
<td><strong>F</strong></td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td></td>
<td>0.002</td>
<td></td>
</tr>
<tr>
<td>0.999</td>
<td></td>
<td>0.998</td>
<td></td>
</tr>
</tbody>
</table>

**\( P(A \mid B, E) \)**

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>( T )</th>
<th>( F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>0.95</td>
<td>0.05</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>0.94</td>
<td>0.06</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>0.29</td>
<td>0.71</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>0.001</td>
<td>0.999</td>
</tr>
</tbody>
</table>

**\( P(A \mid B) \)**

<table>
<thead>
<tr>
<th>A</th>
<th>( T )</th>
<th>( F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>0.90</td>
<td>0.1</td>
</tr>
<tr>
<td>F</td>
<td>0.05</td>
<td>0.95</td>
</tr>
</tbody>
</table>

**\( P(M \mid A) \)**

<table>
<thead>
<tr>
<th>A</th>
<th>( T )</th>
<th>( F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>0.7</td>
<td>0.3</td>
</tr>
<tr>
<td>F</td>
<td>0.01</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Total: 20
Parameter complexity problem

- In the BBN the full joint distribution is defined as:
  \[ P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid pa(X_i)) \]
- What did we save?

Alarm example: 5 binary (True, False) variables

- \# of parameters of the full joint:
  \[ 2^5 = 32 \]
  \[ \text{One parameter depends on the rest:} \]
  \[ 2^5 - 1 = 31 \]
- \# of parameters of the BBN:
  \[ 2^3 + 2(2^2) + 2(2) = 20 \]
  \[ \text{One parameter in every conditional depends on the rest:} \]

Bayesian belief network: free parameters

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>P(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>0.001</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>0.999</td>
</tr>
</tbody>
</table>

| A     | P(A|B,E) |
|-------|----------|
| T | E | T | F |
| T | 0.95 | 0.05 |
| T | 0.94 | 0.06 |
| F | 0.29 | 0.71 |
| F | 0.001 | 0.999 |

| P(M|A) |
|-------|
| T | F |
| T | 0.7 | 0.3 |
| F | 0.01 | 0.99 |

Total free params: 10
Parameter complexity problem

• In the BBN the **full joint distribution** is defined as:
  \[ P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid pa(X_i)) \]

• What did we save?

Alarm example: 5 binary (True, False) variables

# of parameters of the full joint:
\[ 2^5 = 32 \]

One parameter depends on the rest:
\[ 2^5 - 1 = 31 \]

# of parameters of the BBN:
\[ 2^3 + 2(2^2) + 2(2) = 20 \]

One parameter in every conditional depends on the rest:
\[ 2^2 + 2(2) + 2(1) = 10 \]

---

Inference in Bayesian network

• **Bad news:**
  – Exact inference problem in BBNs is NP-hard (Cooper)
  – Approximate inference is NP-hard (Dagum, Luby)

• **But** very often we can achieve significant improvements

• Assume our Alarm network

• Assume we want to compute: \( P(J = T) \)
Inference in Bayesian networks

- Full joint uses the decomposition
- **Calculation of marginals:**
  - Requires summation over variables we want to take out

\[
P(J = T) = \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m} P(B = b, E = e, A = a, J = T, M = m)
\]

- How to compute sums and products more efficiently?

\[
\sum_x a f(x) = a \sum_x f(x)
\]

---

Variable elimination

- **Variable elimination:**
  - E.g. Query \( P(J = T) \) requires to eliminate A,B,E,M and this can be done in different order

\[
P(J = T) = \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)
\]
Variable elimination

**Assume order:** M, E, B, A to calculate $P(J = T)$

$$= \sum_{b \in T} \sum_{e \in T} \sum_{a \in T} \sum_{m \in T} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)$$

---

Variable elimination

**Assume order:** M, E, B, A to calculate $P(J = T)$

$$= \sum_{b \in T} \sum_{e \in T} \sum_{a \in T} \sum_{m \in T} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \sum_{m \in T} P(M = m | A = a)$$
Variable elimination

Assume order: M, E, B, A to calculate $P(J = T)$

$$= \sum_{bcT,f} \sum_{ceT,f} \sum_{aeT,f} \sum_{mcT, T} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)$$

$$= \sum_{bcT,f} \sum_{ceT,f} \sum_{aeT,f} \sum_{mcT, T} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left( \sum_{mcT, T} P(M = m | A = a) \right)$$

$$= \sum_{bcT,f} \sum_{ceT,f} \sum_{aeT,f} \sum_{mcT, T} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)$$

$$= \sum_{bcT,f} \sum_{ceT,f} \sum_{aeT,f} \sum_{mcT, T} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)$$
**Variable elimination**

*Assume order: M, E, B, A* to calculate \( P(J = T) \)

\[
\mathcal{C} = \sum_{b \in B} \sum_{e \in E} \sum_{a \in A} \sum_{m \in M} \sum_{P(J = T \mid A = a)P(M = m \mid A = a)P(A = a \mid B = b, E = e)P(B = b)P(E = e)} \left( \sum_{n \in N} \sum_{P(M = m \mid A = a)} \right)
\]

\[
\mathcal{C} = \sum_{b \in B} \sum_{e \in E} \sum_{a \in A} \sum_{m \in M} \sum_{P(J = T \mid A = a)P(A = a \mid B = b, E = e)P(B = b)P(E = e)} \left( \sum_{n \in N} \sum_{P(M = m \mid A = a)} \right)
\]

\[
\mathcal{C} = \sum_{b \in B} \sum_{e \in E} \sum_{a \in A} \sum_{m \in M} \sum_{P(J = T \mid A = a)P(A = a \mid B = b, E = e)P(B = b)P(E = e)} \left( \sum_{n \in N} \sum_{P(M = m \mid A = a)} \right)
\]

\[
\mathcal{C} = \sum_{b \in B} \sum_{e \in E} \sum_{a \in A} \sum_{m \in M} \sum_{P(J = T \mid A = a)P(A = a \mid B = b, E = e)P(B = b)P(E = e)} \left( \sum_{n \in N} \sum_{P(M = m \mid A = a)} \right)
\]

\[
\mathcal{C} = \sum_{b \in B} \sum_{e \in E} \sum_{a \in A} \sum_{m \in M} \sum_{P(J = T \mid A = a)P(A = a \mid B = b, E = e)P(B = b)P(E = e)} \left( \sum_{n \in N} \sum_{P(M = m \mid A = a)} \right)
\]

\[
\mathcal{C} = \sum_{b \in B} \sum_{e \in E} \sum_{a \in A} \sum_{m \in M} \sum_{P(J = T \mid A = a)P(A = a \mid B = b, E = e)P(B = b)P(E = e)} \left( \sum_{n \in N} \sum_{P(M = m \mid A = a)} \right)
\]
Variable elimination

Assume order: M, E, B, A to calculate \( P(J = T) \)

\[
= \sum_{bcT,T \in T} \sum_{acT,F} \sum_{mct,F} \sum_{T} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)
\]

\[
= \sum_{bcT,F \in T} \sum_{acT,F} \sum_{mct,F} \sum_{T} P(J = T \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e) \left[ \sum_{mct,F} P(M = m \mid A = a) \right]
\]

\[
= \sum_{bcT,F \in T} \sum_{acT,F} \sum_{mct,F} \sum_{T} P(J = T \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e) \quad 1
\]

\[
= \sum_{acT,F} \sum_{bcT,F} \sum_{T} P(J = T \mid A = a) P(B = b) \left[ \sum_{T} P(A = a \mid B = b, E = e) P(E = e) \right]
\]

\[
= \sum_{acT,F} \sum_{bcT,F} \sum_{T} P(J = T \mid A = a) P(B = b) \tau_i(A = a, B = b)
\]

\[
= \sum_{acT,F} \sum_{bcT,F} \sum_{T} P(J = T \mid A = a) \left[ \sum_{T} P(B = b) \tau_i(A = a, B = b) \right]
\]

---

Variable elimination

Assume order: M, E, B, A to calculate \( P(J = T) \)

\[
= \sum_{bcT,F \in T} \sum_{acT,F} \sum_{mct,F} \sum_{T} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)
\]

\[
= \sum_{bcT,F \in T} \sum_{acT,F} \sum_{mct,F} \sum_{T} P(J = T \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e) \left[ \sum_{mct,F} P(M = m \mid A = a) \right]
\]

\[
= \sum_{bcT,F \in T} \sum_{acT,F} \sum_{mct,F} \sum_{T} P(J = T \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e) \quad 1
\]

\[
= \sum_{acT,F} \sum_{bcT,F} \sum_{T} P(J = T \mid A = a) P(B = b) \left[ \sum_{T} P(A = a \mid B = b, E = e) P(E = e) \right]
\]

\[
= \sum_{acT,F} \sum_{bcT,F} \sum_{T} P(J = T \mid A = a) P(B = b) \tau_i(A = a, B = b)
\]

\[
= \sum_{acT,F} \sum_{bcT,F} \sum_{T} P(J = T \mid A = a) \left[ \sum_{T} P(B = b) \tau_i(A = a, B = b) \right]
\]

\[
\tau_i(A = a)
\]
Variable elimination

Assume order: M, E, B, A to calculate \( P(J = T) \)

\[
\sum_{aT} \sum_{bcT} \sum_{acT} \sum_{mT} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) = \sum_{acT} \sum_{bcT} \sum_{mT} \sum_{eT} P(\tau_1(A = a, B = b) | J = T, A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)
\]

\[
\sum_{aT} \sum_{bcT} \sum_{acT} \sum_{mT} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) = \sum_{acT} \sum_{bcT} \sum_{mT} \sum_{eT} P(\tau_2(A = a, B = b) | J = T, A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)
\]

\[
\sum_{aT} \sum_{bcT} \sum_{acT} \sum_{mT} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) = \sum_{acT} \sum_{bcT} \sum_{mT} \sum_{eT} P(\tau_3(A = a) | J = T, A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)
\]

\[
\sum_{aT} \sum_{bcT} \sum_{acT} \sum_{mT} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) = \sum_{acT} \sum_{bcT} \sum_{mT} \sum_{eT} P(\tau_4(A = a) | J = T, A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)
\]
Markov random fields

- Probabilistic models with symmetric dependences.
  - Typically models spatially varying quantities
    \[ P(x) \propto \prod_{c\in\mathcal{F}(x)} \phi_c(x_c) \]
    \[ \phi_c(x_c) \text{ - A potential function (defined over factors)} \]
    - If \( \phi_c(x_c) \) is strictly positive we can rewrite the definition in terms of a log-linear model:
      \[ P(x) = \frac{1}{Z} \exp \left( - \sum_{c\in\mathcal{F}(x)} E_c(x_c) \right) \]
      Energy function
    - Gibbs (Boltzmann) distribution
      \[ Z = \sum_{x\in\mathcal{X}} \exp \left( - \sum_{c\in\mathcal{F}(x)} E_c(x_c) \right) \]
      - A partition function

Graphical representation of MRFs

- An undirected network (also called independence graph)
  - \( G = (S, E) \)
    - \( S=1, 2,.. N \) correspond to random variables
    - \( (i, j) \in E \iff \exists c : \{i, j\} \subseteq c \)
      or \( x_i \) and \( x_j \) appear within the same factor \( c \)
  - Example:
    - variables A,B ..H
    - Assume the full joint of MRF
      \[ P(A, B, \ldots H) \sim \]
      \[ \phi_1(A, B, C)\phi_2(B, D, E)\phi_3(A, G) \]
      \[ \phi_4(C, F)\phi_5(G, H)\phi_6(F, H) \]
Markov random fields

• regular lattice (Ising model)

• Arbitrary graph

Markov random fields

• regular lattice (Ising model)

• Arbitrary graph
Markov random fields: independence relations

- **Pairwise Markov property**
  - Two nodes in the network that are not directly connected can be made independent given all other nodes

- **Local Markov property**
  - A set of nodes (variables) can be made independent from the rest of nodes variables given its immediate neighbors

- **Global Markov property**
  - A vertex set A is independent of the vertex set B (A and B are disjoint) given set C if all chains in between elements in A and B intersect C

Types of Markov random fields

- **MRFs with discrete random variables**
  - Clique potentials can be defined by mapping all clique-variable instances to R
  - Example: Assume two binary variables A, B with values \{a_1, a_2, a_3\} and \{b_1, b_2\} are in the same clique c. Then:

\[
\phi_c(A, B) \equiv \begin{array}{c|c|c}
  a_1 & b_1 & 0.5 \\
  a_1 & b_2 & 0.2 \\
  a_2 & b_1 & 0.1 \\
  a_2 & b_2 & 0.3 \\
  a_3 & b_1 & 0.2 \\
  a_3 & b_2 & 0.4 \\
\end{array}
\]
Types of Markov random fields

• **Gaussian Markov Random Field**
  \[ x \sim N(\mu, \Sigma) \]
  \[
p(x | \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left[ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right]
  \]

• **Precision matrix** \( \Sigma^{-1} \)

• Variables in \( x \) are connected in the network only if they have a **nonzero entry** in the precision matrix
  - All zero entries are not directly connected

---

MRF variable elimination inference

**Example:**

\[
P(B) = \sum_{A,C,D,...,H} P(A,B,...,H)
\]

\[
= \frac{1}{Z} \sum_{A,C,D,...,H} \phi_1(A,B,C) \phi_2(B,D,E) \phi_3(A,G) \phi_4(C,F) \phi_5(G,H) \phi_6(F,H)
\]

**Eliminate E**

\[
= \frac{1}{Z} \sum_{A,C,D,F,G,H} \phi_1(A,B,C) \left[ \sum_E \phi_2(B,D,E) \right] \phi_3(A,G) \phi_4(C,F) \phi_5(G,H) \phi_6(F,H)
\]

\[
\tau_1(B,D)
\]
Factors

- **Factor**: is a function that maps value assignments for a subset of random variables to $\mathbb{R}$ (reals)
- **The scope of the factor**: a set of variables defining the factor
- **Example**: Assume discrete random variables $x$ (with values $a_1, a_2, a_3$) and $y$ (with values $b_1$ and $b_2$)
  - Factor:
    \[
    \phi(x, y)
    \]
  - Scope of the factor:
    \[
    \{x, y\}
    \]

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$b_1$</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$b_2$</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_1$</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_2$</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_1$</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_2$</td>
<td>0.4</td>
<td></td>
</tr>
</tbody>
</table>

Factor Product

**Variables**: $A, B, C$

\[
\phi(A, B, C) = \phi(B, C) \circ \phi(A, B)
\]

\[
\phi(B, C)
\]

\[
\phi(A, B)
\]

\[
\phi(A, B, C)
\]

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$b_1$</th>
<th>$c_1$</th>
<th>0.5%1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_2$</td>
<td>0.5%0.6</td>
<td></td>
</tr>
<tr>
<td>$a_1$</td>
<td>$b_2$</td>
<td>$c_1$</td>
<td>0.2%0.3</td>
<td></td>
</tr>
<tr>
<td>$a_1$</td>
<td>$b_2$</td>
<td>$c_2$</td>
<td>0.2%0.4</td>
<td></td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>0.1%0.1</td>
<td></td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_1$</td>
<td>$c_2$</td>
<td>0.1%0.6</td>
<td></td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_2$</td>
<td>$c_1$</td>
<td>0.3%0.3</td>
<td></td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_2$</td>
<td>$c_2$</td>
<td>0.3%0.4</td>
<td></td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>0.2%0.1</td>
<td></td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_1$</td>
<td>$c_2$</td>
<td>0.2%0.6</td>
<td></td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_2$</td>
<td>$c_1$</td>
<td>0.4%0.3</td>
<td></td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_2$</td>
<td>$c_2$</td>
<td>0.4%0.4</td>
<td></td>
</tr>
</tbody>
</table>
Factor Marginalization

Variables: A, B, C

\[ \phi(A, C) = \sum_B \phi(A, B, C) \]

<table>
<thead>
<tr>
<th>a1</th>
<th>b1</th>
<th>c1</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>b1</td>
<td>c2</td>
<td>0.35</td>
</tr>
<tr>
<td>a1</td>
<td>b2</td>
<td>c1</td>
<td>0.4</td>
</tr>
<tr>
<td>a1</td>
<td>b2</td>
<td>c2</td>
<td>0.15</td>
</tr>
<tr>
<td>a2</td>
<td>b1</td>
<td>c1</td>
<td>0.5</td>
</tr>
<tr>
<td>a2</td>
<td>b1</td>
<td>c2</td>
<td>0.1</td>
</tr>
<tr>
<td>a2</td>
<td>b2</td>
<td>c1</td>
<td>0.3</td>
</tr>
<tr>
<td>a2</td>
<td>b2</td>
<td>c2</td>
<td>0.2</td>
</tr>
<tr>
<td>a3</td>
<td>b1</td>
<td>c1</td>
<td>0.25</td>
</tr>
<tr>
<td>a3</td>
<td>b1</td>
<td>c2</td>
<td>0.45</td>
</tr>
<tr>
<td>a3</td>
<td>b2</td>
<td>c1</td>
<td>0.15</td>
</tr>
<tr>
<td>a3</td>
<td>b2</td>
<td>c2</td>
<td>0.25</td>
</tr>
</tbody>
</table>

MRF variable elimination inference

Example (cont):

\[ P(B) = \sum_{A,C,D,...H} P(A, B, ..., H) \]

\[ = \sum_{A,C,D,F,G,H} \phi_1(A, B, C) \tau_1(B, D) \phi_3(A, G) \phi_4(C, F) \phi_5(G, H) \phi_6(F, H) \]

Eliminate D

\[ = \sum_{A,C,F,G,H} \phi_1(A, B, C) \left[ \sum_D \tau_1(B, D) \right] \phi_3(A, G) \phi_4(C, F) \phi_5(G, H) \phi_6(F, H) \]

\[ \tau_2(B) \]

CS 3750 Advanced Machine Learning
Example (cont):

\[ P(B) = \sum_{A,C,D,...,H} P(A,B,...H) \]

\[ = \sum_{A,C,F,G,H} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\phi_5(C,F)\phi_6(G,H)\phi_8(F,H) \]

Eliminate H

\[ = \sum_{A,C,F,G} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\phi_4(C,F) \left[ \sum_H \phi_6(G,H)\phi_8(F,H) \right] \]

\[ \tau_3(F,G,H) \]

\[ \tau_4(F,G) \]

MRF variable elimination inference

---

Example (cont):

\[ P(B) = \sum_{A,C,D,...,H} P(A,B,...H) \]

\[ = \sum_{A,C,F,G} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\phi_4(C,F) \left[ \sum_F \phi_4(C,F)\tau_4(F,G) \right] \]

Eliminate F

\[ = \sum_{A,C,G} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G) \left[ \sum_F \phi_4(C,F)\tau_4(F,G) \right] \]

\[ \tau_5(C,F,G) \]

\[ \tau_6(G,C) \]
MRF variable elimination inference

Example (cont):

\[
P(B) = \sum_{A,C,D,H} P(A,B,...H) \\
= \sum_{A,C,G} \phi_1(A,B,C) \tau_2(B) \phi_3(A,G) \tau_6(C,G)
\]

Eliminate G

\[
= \sum_{A,C} \phi(A,B,C) \tau_2(B) \left[ \sum_F \phi_1(A,G) \tau_6(C,G) \right]
\]

\[
\tau_7(A,C,G) \quad \tau_8(A,C)
\]

MRF variable elimination inference

Example (cont):

\[
P(B) = \sum_{A,C,D,H} P(A,B,...H) \\
= \sum_{A,C} \phi_1(A,B,C) \tau_2(B) \tau_8(A,C)
\]

Eliminate C

\[
= \sum_A \tau_2(B) \left[ \sum_C \phi(A,B,C) \tau_8(A,C) \right]
\]

\[
\tau_9(A,B,C) \quad \tau_{10}(A,B)
\]
MRF variable elimination inference

Example (cont):

\[ P(B) = \sum_{A,C,D,...,H} P(A, B, ..., H) \]

\[ = \sum_A \tau_2(B) \tau_{10}(A, B) \]

\[ = \tau_2(B) \sum_A \tau_{10}(A, B) \]

Eliminate A

\[ = \tau_2(B) \sum_A \tau_{10}(A, B) \]

\[ = \tau_2(B) \tau_{11}(B) \]

Induced graph

- A graph induced by a specific variable elimination order:

- a graph G extended by links that represent intermediate factors

CS 3750 Advanced Machine Learning
Tree decomposition of the graph

- A tree decomposition of a graph $G$:
  - A tree $T$ with a vertex set associated to every node.
  - For all edges $\{v, w\} \in G$: there is a set containing both $v$ and $w$ in $T$.
  - For every $v \in G$: the nodes in $T$ that contain $v$ form a connected subtree.
Tree decomposition of the graph

- A tree decomposition of a graph $G$:
  - A tree $T$ with a vertex set associated to every node.
  - For all edges $\{v,w\} \in G$: there is a set containing both $v$ and $w$ in $T$.
  - For every $v \in G$: the nodes in $T$ that contain $v$ form a connected subtree.

Tree decomposition of the graph

- Another tree decomposition of a graph $G$:
  - A tree $T$ with a vertex set associated to every node.
  - For all edges $\{v,w\} \in G$: there is a set containing both $v$ and $w$ in $T$.
  - For every $v \in G$: the nodes in $T$ that contain $v$ form a connected subtree.
**Tree decomposition of the graph**

- **Another tree decomposition of a graph** $G$:
  - A tree $T$ with a vertex set associated to every node.
  - For all edges $\{v, w\} \in G$: there is a set containing both $v$ and $w$ in $T$.
  - For every $v \in G$: the nodes in $T$ that contain $v$ form a connected subtree.
Treewidth of the graph

- **Treewidth** of a graph $G$: $\text{tw}(G) =$ minimum width over all tree decompositions of $G$
- Why is it important?
  - The calculations can take advantage of the structure and be performed more efficiently
  - treewidth gives the best case complexity

Trees

Why do we like trees?
- Inference in trees structures can be done in time linear in the number of nodes in the tree
Converting BBNs to MRFs

**Moral-graph H[G]**: of a Bayesian network over X is an undirected graph over X that contains an edge between x and y if:
- There exists a directed edge between them in G.
- They are both parents of the same node in G.

\[
\begin{align*}
\phi_1(C) & \phi_2(D, C) \phi_3(G, I, D) \phi_4(S, I) \phi_5(L, G) \phi_6(J, L, S) \phi_7(H, G, J) \\
P(G | I, D) & = \frac{P(C)P(D | C)P(G | I, D)P(S | I)P(L | G)P(J | L, S)P(H | G, J)}{\phi_1(C) \phi_2(D, C) \phi_3(G, I, D) \phi_4(S, I) \phi_5(L, G) \phi_6(J, L, S) \phi_7(H, G, J)}
\end{align*}
\]

Moral Graphs

Why moralization?

\[
\]

\[
= \phi_1(C) \phi_2(D, C) \phi_3(G, I, D) \phi_4(S, I) \phi_5(L, G) \phi_6(J, L, S) \phi_7(H, G, J)
\]
**Chordal graphs**

**Chordal Graph:** an undirected graph $G$
- all cycles of four or more vertices have a chord (another edge breaking the cycle)
- minimum cycle for every vertex in a cycle is 3 (contains 3 vertices)

![Chordal Graph Examples](image1)

---

**Chordal Graphs**

**Properties:**
- There exists an elimination ordering that adds no edges.
- The minimal induced treewidth of the graph is equal to the size of the largest clique - 1.

![Chordal Graph Examples](image2)
**Triangulation**

The process of converting a graph $G$ into a chordal graph is called Triangulation.

A new graph obtained via triangulation is:
1) Guaranteed to be chordal.
2) Not guaranteed to be (treewidth) optimal.

- There exist exact algorithms for finding the **minimal chordal graphs**, and heuristic methods with a guaranteed upper bound.

---

**Chordal Graphs**

- Given a minimum triangulation for a graph $G$, we can carry out the variable-elimination algorithm in the minimum possible time.

- **Complexity** of the optimal triangulation:
  - Finding the minimal triangulation is **NP-Hard**.

- **The inference limit**:
  - Inference time is exponential in terms of the largest clique (factor) in $G$. 
Conclusions: MRFs

- We cannot escape exponential costs in the treewidth

- But in many graphs the tree-width is much smaller than the total number of variables

- Still a problem: Finding the optimal decomposition is hard
  - But, paying the cost up front may be worth it.
  - Triangulate once, query many times.
  - Real cost savings if not a bounded one