Problem Solving with AI Techniques
Bayesian Networks: Inference

Paul Weng

UM-SJTU Joint Institute

VE593, Fall 2018

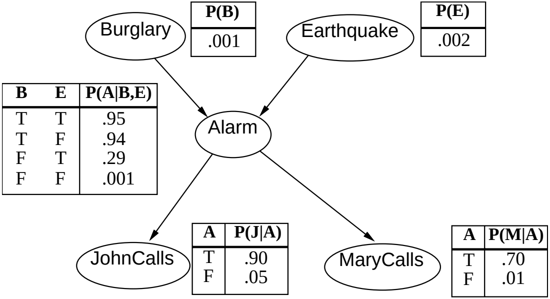
1. How to Do Inference?
   - Definition
   - Exact Inference by Enumeration
   - Exact Inference by Variable Elimination
   - Exact Inference by Belief Propagation
   - Approximate Inference by Sampling

Inference

- Inference: Given some pieces of information (e.g., prior, observed variables), what is the implication (e.g., posterior) on a non-observed variables?
- In a Bayes net, all random variables are divided in three groups:
  - Z observed variables
  - X and Y hidden random variables
  - We are interested in X, but not in Y
- Formally, we want to compute the posterior marginal $P(X \mid Z = z)$

$$P(X \mid Z = z) = \frac{P(X, Z = z)}{P(Z = z)} = \frac{1}{P(Z = z)} \sum_{Y} P(X, Y, Z = z)$$

Example of Inference

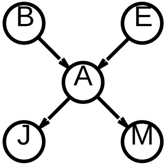


Is there a burglary if the two neighbors call? What is $\mathbb{P}(B \mid J = j, M = m)$?

1. How to Do Inference?
   - Definition
   - Exact Inference by Enumeration
   - Exact Inference by Variable Elimination
   - Exact Inference by Belief Propagation
   - Approximate Inference by Sampling

Inference by Enumeration

$$\begin{aligned}
\mathbb{P}(B \mid j, m) &= \frac{\mathbb{P}(B, j, m)}{\mathbb{P}(j, m)} \\
&\propto \mathbb{P}(B, j, m) \\
&\propto \sum_E \sum_A \mathbb{P}(B, E, A, j, m) \\
&\propto \sum_e \sum_a \mathbb{P}(B)\mathbb{P}(e)\mathbb{P}(a \mid B, e)\mathbb{P}(j \mid a)\mathbb{P}(m \mid a)
\end{aligned}$$

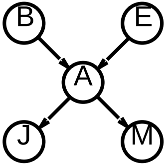


$\text{Time complexity} = O(n2^n)$

Inference by Enumeration

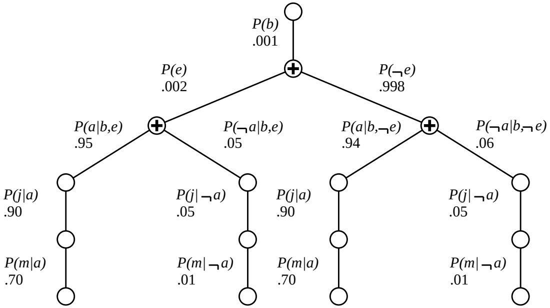
$$\begin{aligned}
\mathbb{P}(B \mid j, m) &= \frac{\mathbb{P}(B, j, m)}{\mathbb{P}(j, m)} \\
&\propto \mathbb{P}(B, j, m) \\
&\propto \sum_E \sum_A \mathbb{P}(B, E, A, j, m) \\
&\propto \sum_e \sum_a \mathbb{P}(B)\mathbb{P}(e)\mathbb{P}(a \mid B, e)\mathbb{P}(j \mid a)\mathbb{P}(m \mid a)
\end{aligned}$$

Time complexity $= O(n2^n)$



$$\propto \mathbb{P}(B) \sum_e \mathbb{P}(e) \sum_a \mathbb{P}(a \mid B, e)\mathbb{P}(j \mid a)\mathbb{P}(m \mid a)$$

Evaluation Tree



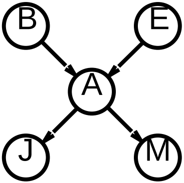
- Enumeration is inefficient: repeated computation e.g., $\mathbb{P}(j \mid a)\mathbb{P}(m \mid a)$ is computed for each possible value of E
- Time complexity: $O(2^n)$ for n Boolean variables
- Space complexity: $O(n)$

1. How to Do Inference?
   - Definition
   - Exact Inference by Enumeration
   - Exact Inference by Variable Elimination
   - Exact Inference by Belief Propagation
   - Approximate Inference by Sampling

Variable Elimination: Principle and Example

- Idea: Carry out summations from right-to-left, storing intermediate results (factors) to avoid recomputation

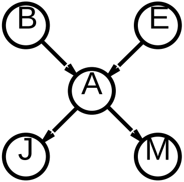
$$
\begin{aligned}
\mathbb{P}(B \mid j, m) &\propto \underbrace{\mathbb{P}(B)}_{B} \sum_{E} \underbrace{\mathbb{P}(E)}_{E} \sum_{A} \underbrace{\mathbb{P}(A \mid B, E)}_{A} \underbrace{\mathbb{P}(j \mid A)}_{J} \underbrace{\mathbb{P}(m \mid A)}_{M} \\
&\propto f_1(B) \sum_{E} f_2(E) \sum_{A} f_3(A, B, E) f_4(A) f_5(A) \\
&\propto f_1(B) \sum_{E} f_2(E) f_6(B, E) \\
&\propto f_1(B) f_7(B)
\end{aligned}
$$



Variable Elimination: Principle and Example

- Idea: Carry out summations from right-to-left, storing intermediate results (factors) to avoid recomputation

$$
\begin{aligned}
\mathbb{P}(B \mid j, m) &\propto \underbrace{\mathbb{P}(B)}_{B} \sum_{E} \underbrace{\mathbb{P}(E)}_{E} \sum_{A} \underbrace{\mathbb{P}(A \mid B, E)}_{A} \underbrace{\mathbb{P}(j \mid A)}_{J} \underbrace{\mathbb{P}(m \mid A)}_{M} \\
&\propto f_1(B) \sum_{E} f_2(E) \sum_{A} f_3(A, B, E) f_4(A) f_5(A) \\
&\propto f_1(B) \sum_{E} f_2(E) f_6(B, E) \\
&\propto f_1(B) f_7(B)
\end{aligned}
$$



- 2 operations needed: pointwise product and summing out a variable

Basic Operations: Pointwise Product

- Pointwise product of factors f_1 and f_2:

$$\begin{aligned} f(X_1, \ldots, X_j, Y_1, \ldots, Y_k) \times f'(Y_1, \ldots, Y_k, Z_1, \ldots, Z_l) \\ = f''(X_1, \ldots, X_j, Y_1, \ldots, Y_k, Z_1, \ldots, Z_l) \end{aligned}$$

e.g., $f(A, B) \times f'(B, C) = f''(A, B, C)$

Basic Operations: Pointwise Product

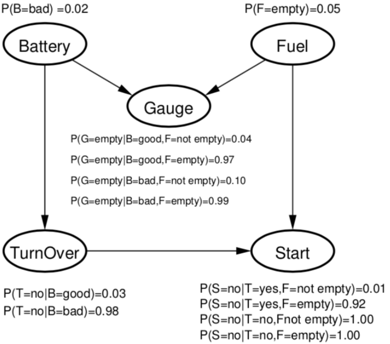
- Pointwise product of factors f_1 and f_2:

$$f(X_1,\dots,X_j,Y_1,\dots,Y_k)\times f'(Y_1,\dots,Y_k,Z_1,\dots,Z_l)$$
$$= f''(X_1,\dots,X_j,Y_1,\dots,Y_k,Z_1,\dots,Z_l)$$

  e.g., $f(A,B)\times f'(B,C) = f''(A,B,C)$

- Example: Compute $f_T(T,B)f_S(S,T,F)$

(Heckermann 1995)



Basic Operations: Summing Out

- Summing out a variable X from a factor:

$$\sum_X f(X, Y_1, \ldots, Y_n) = f'(Y_1, \ldots, Y_n)$$

Basic Operations: Summing Out

- Summing out a variable X from a factor:

$$\sum_X f(X, Y_1, \ldots, Y_n) = f'(Y_1, \ldots, Y_n)$$

- Example: Compute $\sum_T f_T(T, B) f_S(S, T, F)$

Variable Elimination Algorithm

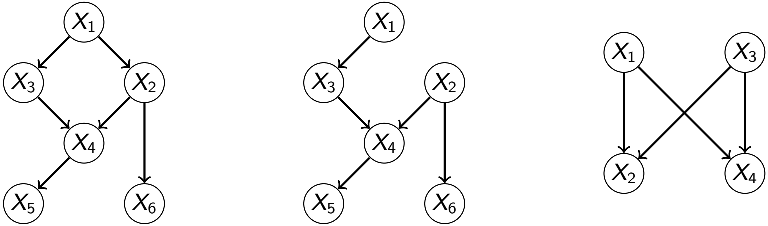
Algorithm for query $\mathbb{P}(X \mid Z = z)$

- Start with initial factors:
    - Local CPTs (but instantiated with z)
- While there are still hidden variables (not X nor Z)
    - Pick a hidden variable H
    - Join all factors depending on H
    - Sum out H
- Join all remaining factors and normalize

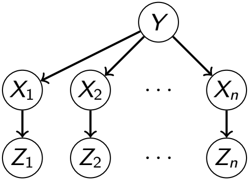
Time/space complexity of Variable Elimination

- Time/space complexity: exponential in treewidth
- Treewidth: size of largest factor -1
- Efficient if Bayesian network = polytree (i.e., singly-connected graph) with bounded treewidth

Examples: are they polytrees?

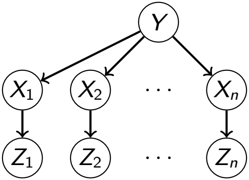


Remark: Importance of Variable Ordering



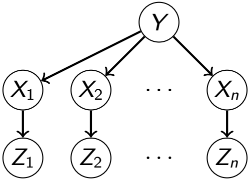
- Query: $\mathbb{P}(X_n \mid z_1, \ldots, z_n)$
- What is the size of the maximum factor generated with order $Y, X_1, \ldots, X_{n-1}$?

Remark: Importance of Variable Ordering



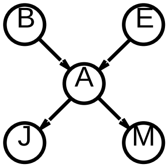
- Query: $\mathbb{P}(X_n \mid z_1, \ldots, z_n)$
- What is the size of the maximum factor generated with order $Y, X_1, \ldots, X_{n-1}$?
- What is the size of the maximum factor generated with order $X_1, \ldots, X_{n-1}, Y$?

Remark: Importance of Variable Ordering



- Query: $\mathbb{P}(X_n \mid z_1, \ldots, z_n)$
- What is the size of the maximum factor generated with order $Y, X_1, \ldots, X_{n-1}$?
- What is the size of the maximum factor generated with order $X_1, \ldots, X_{n-1}, Y$?
- Conclusion: var. ordering can greatly impact time/space complexity
  Unfortunately computing the optimal variable ordering is NP-hard!

Remark: Irrelevant Variables

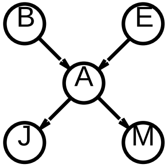


- Consider the query $\mathbb{P}(J \mid b)$

$$\mathbb{P}(J|b) \propto \mathbb{P}(b) \sum_e \mathbb{P}(e) \sum_a \mathbb{P}(a \mid b, e) \mathbb{P}(J \mid a) \sum_m P(m \mid a)$$

What is the value of $\sum_m \mathbb{P}(m \mid a)$?

Remark: Irrelevant Variables

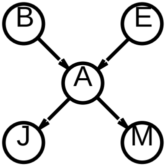


- Consider the query $\mathbb{P}(J \mid b)$

$$\mathbb{P}(J|b) \propto \mathbb{P}(b) \sum_e \mathbb{P}(e) \sum_a \mathbb{P}(a \mid b, e) \mathbb{P}(J \mid a) \sum_m P(m \mid a)$$

What is the value of $\sum_m \mathbb{P}(m \mid a)$?
It is equal to 1! M is irrelevant to the query

Remark: Irrelevant Variables



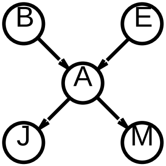
- Consider the query $\mathbb{P}(J \mid b)$

$$\mathbb{P}(J|b) \propto \mathbb{P}(b) \sum_e \mathbb{P}(e) \sum_a \mathbb{P}(a \mid b, e) \mathbb{P}(J \mid a) \sum_m P(m \mid a)$$

What is the value of $\sum_m \mathbb{P}(m \mid a)$?
It is equal to 1! M is irrelevant to the query

- Theorem: Y is irrelevant unless $Y \in \text{Ancestors}(X \cup Z)$

Remark: Irrelevant Variables



- Consider the query $\mathbb{P}(J \mid b)$

$$\mathbb{P}(J|b) \propto \mathbb{P}(b) \sum_e \mathbb{P}(e) \sum_a \mathbb{P}(a \mid b, e)\mathbb{P}(J \mid a) \sum_m P(m \mid a)$$

What is the value of $\sum_m \mathbb{P}(m \mid a)$?
It is equal to 1! M is irrelevant to the query

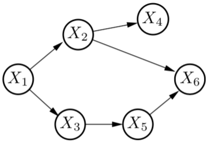
- Theorem: Y is irrelevant unless $Y \in \text{Ancestors}(X \cup Z)$

Here, $X = \{J\}$, $Z = \{B\}$ and $\text{Ancestors}(X \cup Z) = \{A, E\}$, therefore M is irrelevant

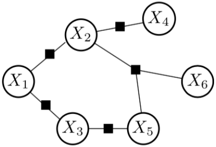
1. How to Do Inference?
   - Definition
   - Exact Inference by Enumeration
   - Exact Inference by Variable Elimination
   - Exact Inference by Belief Propagation
   - Approximate Inference by Sampling

Belief Propagation

- Issue: Variable Elimination is query sensitive: for any new query, the entire algorithm has to be rerun
- Belief propagation (message passing) algorithm computes all marginal probabilities by storing and reusing intermediate factors
- We present the algorithm in factor graphs

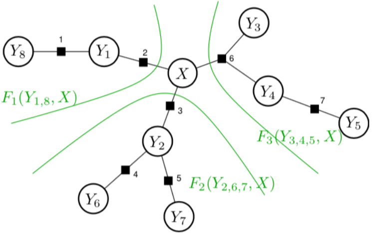


$P(x_{1:6}) = P(x_1)\,P(x_2|x_1)\,P(x_3|x_1)\,P(x_4|x_2)\,P(x_5|x_3)\,P(x_6|x_2,x_5)$



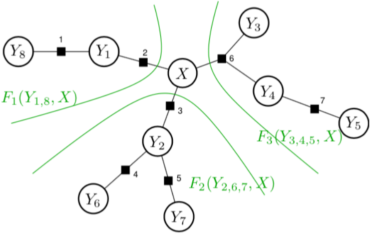
$P(x_{1:6}) = f_1(x_1,x_2)\,f_2(x_3,x_1)\,f_3(x_2,x_4)\,f_4(x_3,x_5)\,f_5(x_2,x_5,x_6)$

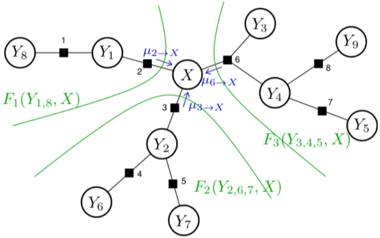
Belief Propagation: Example for Computing $\mathbb{P}(X)$



$$F_1(Y_{1,8}, X) = f_1(Y_8, Y_1)f_2(Y_1, X)$$
$$F_2(Y_{2,6,7}, X) = f_3(X, Y_2)f_4(Y_2, Y_6)f_5(Y_2, Y_7)$$
$$F_3(Y_{3,4,5}, X) = f_6(X, Y_3, Y_4)f_7(Y_4, Y_5)$$
$$\mathbb{P}(Y_{1:8}, X) = F_1(Y_{1,8}, X)F_2(Y_{2,6,7}, X)F_3(Y_{3,4,5}, X)$$

Belief Propagation: Example for Computing $\mathbb{P}(X)$



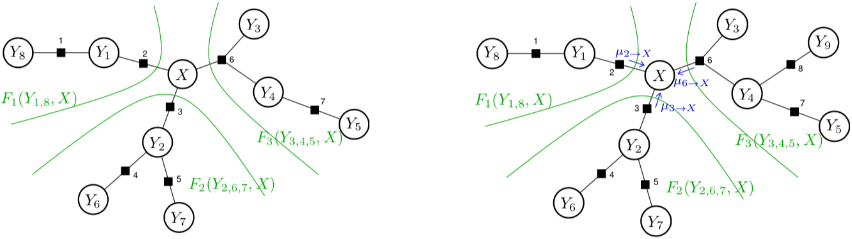


$$F_1(Y_{1,8}, X) = f_1(Y_8, Y_1) f_2(Y_1, X)$$
$$F_2(Y_{2,6,7}, X) = f_3(X, Y_2) f_4(Y_2, Y_6) f_5(Y_2, Y_7)$$
$$F_3(Y_{3,4,5}, X) = f_6(X, Y_3, Y_4) f_7(Y_4, Y_5)$$
$$\mathbb{P}(Y_{1:8}, X) = F_1(Y_{1,8}, X) F_2(Y_{2,6,7}, X) F_3(Y_{3,4,5}, X)$$

$$\mu_{1\to X}(X) = \sum_{Y_{1,8}} F_1(Y_{1,8}, X)$$
$$\mu_{2\to X}(X) = \sum_{Y_{2,6,7}} F_2(Y_{2,6,7}, X)$$
$$\mu_{3\to X}(X) = \sum_{Y_{3,4,5}} F_3(Y_{3,4,5}, X)$$
$$\mathbb{P}(X) = \mu_{1\to X}(X) \mu_{2\to X}(X) \mu_{3\to X}(X)$$

- Object oriented view of computation: nodes exchange messages

Belief Propagation: Example for Computing $\mathbb{P}(X)$

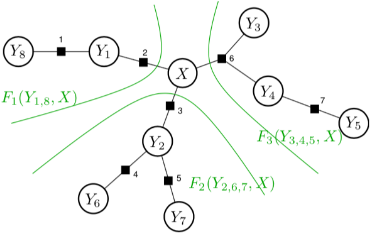


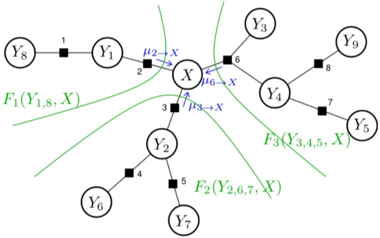
$$F_1(Y_{1,8}, X) = f_1(Y_8, Y_1) f_2(Y_1, X)$$
$$F_2(Y_{2,6,7}, X) = f_3(X, Y_2) f_4(Y_2, Y_6) f_5(Y_2, Y_7)$$
$$F_3(Y_{3,4,5}, X) = f_6(X, Y_3, Y_4) f_7(Y_4, Y_5)$$
$$\mathbb{P}(Y_{1:8}, X) = F_1(Y_{1,8}, X) F_2(Y_{2,6,7}, X) F_3(Y_{3,4,5}, X)$$

$$\mu_{1\to X}(X) = \sum_{Y_{1,8}} F_1(Y_{1,8}, X)$$
$$\mu_{2\to X}(X) = \sum_{Y_{2,6,7}} F_2(Y_{2,6,7}, X)$$
$$\mu_{3\to X}(X) = \sum_{Y_{3,4,5}} F_3(Y_{3,4,5}, X)$$
$$\mathbb{P}(X) = \mu_{1\to X}(X) \mu_{2\to X}(X) \mu_{3\to X}(X)$$

- Object oriented view of computation: nodes exchange messages
- After all exchanges in all nodes, each node can compute its marginal

Belief Propagation: Example for Computing $\mathbb{P}(X)$





$$F_1(Y_{1,8}, X) = f_1(Y_8, Y_1) f_2(Y_1, X)$$
$$F_2(Y_{2,6,7}, X) = f_3(X, Y_2) f_4(Y_2, Y_6) f_5(Y_2, Y_7)$$
$$F_3(Y_{3,4,5}, X) = f_6(X, Y_3, Y_4) f_7(Y_4, Y_5)$$
$$\mathbb{P}(Y_{1:8}, X) = F_1(Y_{1,8}, X) F_2(Y_{2,6,7}, X) F_3(Y_{3,4,5}, X)$$

$$\mu_{1\to X}(X) = \sum_{Y_{1,8}} F_1(Y_{1,8}, X)$$
$$\mu_{2\to X}(X) = \sum_{Y_{2,6,7}} F_2(Y_{2,6,7}, X)$$
$$\mu_{3\to X}(X) = \sum_{Y_{3,4,5}} F_3(Y_{3,4,5}, X)$$
$$\mathbb{P}(X) = \mu_{1\to X}(X) \mu_{2\to X}(X) \mu_{3\to X}(X)$$

- Object oriented view of computation: nodes exchange messages
- After all exchanges in all nodes, each node can compute its marginal
- With same reasoning, joint distrib. in factor nodes can be computed

General Message Passing Equations

- Message $\mu_{k\to i}$ from factor k to node i:

$$\mu_{k\to i}(X_i) = \sum_{X_{\partial k\setminus i}} f_k(X_{\partial k}) \prod_{j\in\partial k\setminus i} \overline{\mu}_{j\to k}(X_j)$$

  where ∂k represents all the neighbor nodes (e.g., variables) of factor k, $\partial k\setminus i$ is ∂k with variable i excluded. If $\partial k\setminus i = \{\,\}$, $\mu_{k\to i}(X_i) = f_k(X_{\partial k})$

- Message $\overline{\mu}_{j\to k}$ from node j to factor k:

$$\overline{\mu}_{j\to k}(X_j) = \prod_{k'\in\partial j\setminus k} \mu_{k'\to j}(X_j)$$

  where ∂j represents all the neighbor nodes (e.g., factors) of variable i, $\partial j\setminus k$ is ∂j with factor k excluded. If $\partial j\setminus k = \{\,\}$, $\overline{\mu}_{j\to k}(X_j) = 1$.

Other Algorithms

- Issue: Belief propagation is only guaranteed to work for polytrees
- Loopy belief propagation: apply iteratively belief propagation on general graphs
- Junction Tree Algorithms: transform a network into a polytree by joining variables and apply belief propagation
  - Shafer-Shenoy Algorithm
  - Hugin Algorithm

Other Type of Query

- MAP Query: What is the most probable assignment?

$$X^{MAP} = \arg\max_X \mathbb{P}(X \mid Z = z) = \arg\max_X \mathbb{P}(X, Z = z)$$

- Assuming there is no hidden variables Y, previous algorithms can be adapted by replacing $\sum$ by max!

Monte Carlo Method

- Goal: estimate $P(X_I \,|\, X_O = \boldsymbol{z}_O)$ with $I \cup J \cup O = \{1, 2, \dots, n\}$, i.e., X_J are hidden variables we are not interested in
- Idea: Use a Monte Carlo method to estimate a probability using the fact that a probability is an expectation

- How to generate samples from a Bayes net?
  - Sort all random variables in topological order: $X_1, X_2, \dots, X_n$
  - For $i = 1, \dots, n$, sample $X_i \sim \mathbb{P}(X_i \,|\, Parents(X_i))$

Rejection Sampling

- Principle: Generate N samples from Bayes net and reject any sample that does not match $Z = z$:

$$\mathbb{P}(X_I = \boldsymbol{x}_I \mid X_O = \boldsymbol{z}_O) \approx \frac{1}{N}\sum_{k=1}^{N}[\boldsymbol{x}_I^k = \boldsymbol{x}_I]$$

  where $\boldsymbol{x}^1, \dots \boldsymbol{x}^N$ samples from Bayes net and $\forall k, \boldsymbol{x}_O^k = \boldsymbol{z}_O$
- Algorithm:
  Until we have N samples, loop from $i = 1, \dots, n$,
  - $x_i^k \leftarrow$ sample $X_i \sim \mathbb{P}(X_i \mid Parents(X_i))$
  - If $i \in O$ and $x_i^k \neq z_i$, restart from $i = 1$
- Issue: Not efficient, many samples may be rejected, especially in large networks

Importance Sampling using Likelihood Weighting

- Principle: Generate weighted sample set $(w^k, \boldsymbol{x}^k)$ for $k = 1, \ldots, N$ where $w^k = \mathbb{P}(X_O = \boldsymbol{z}_O \mid \boldsymbol{X} = \boldsymbol{x}^k)$:

$$\mathbb{P}(X_I = \boldsymbol{x}_I \mid X_O = \boldsymbol{z}_O) \approx \frac{1}{\sum_{k=1}^N w^k} \sum_{k=1}^N w^k [\boldsymbol{x}_I^k = \boldsymbol{x}_I]$$

- Algorithm: Assume w^k initialized to 1
For $k = 1, \ldots, N$, for $i = 1, \ldots, n$, do
    - If $i \notin O$, $x_i^k \leftarrow$ sample $X_i \sim \mathbb{P}(X_i \mid Parents(X_i))$
    - If $i \in O$, $x_i^k = z_i$ and $w^k \leftarrow w^k \mathbb{P}(X_i = z_i \mid Parents(X_i))$

Why does it Work?

- Sample $\boldsymbol{x}^k$ has probability $\prod_{i \in I \cup J} \mathbb{P}(X_i \mid Parents(X_i))$ to be sampled
- Weight $w^k = \prod_{j \in O} \mathbb{P}(X_j = \boldsymbol{x}_j^k \mid Parents(X_j)$ given by $\boldsymbol{x}^k)$
- Therefore

$$\hat{\mathbb{P}}(X_I = \boldsymbol{x}_I \mid X_O = \boldsymbol{z}_O) = \alpha \sum_{k=1}^{N} [\boldsymbol{x}_I^k = \boldsymbol{x}_I] w^k$$

$$= \alpha \sum_{\boldsymbol{x}_J} \sum_{k=1}^{N} [\boldsymbol{x}_I^k = \boldsymbol{x}_I, \boldsymbol{x}_J^k = \boldsymbol{x}_J] w^k$$

$$\approx \alpha' \sum_{X_J} \prod_{i \in I \cup J} \mathbb{P}(X_i \mid Parents(X_i)) \prod_{j \in O} \mathbb{P}(X_j \mid Parents(X_j))$$

where $X_i = x_i$ for $i \in I$

$$= \alpha' \sum_{X_J} \mathbb{P}(X_I = \boldsymbol{x}_I, X_J, X_O = \boldsymbol{z}_O)$$

$$= \alpha' \mathbb{P}(X_I = \boldsymbol{x}_I, X_O = \boldsymbol{z}_O) = \mathbb{P}(X_I = \boldsymbol{x}_I \mid X_O = \boldsymbol{z}_O)$$

Other Inference Methods

- Sampling:
  - Gibbs sampling
  - More generally, Markov-chain Monte Carlo (MCMC) methods
- Other approximations/variational methods:
  - Expectation propagation
  - Specialized variational methods depending on the model
- Reductions:
  - Mathematical programming (e.g., LP relaxations of MAP)
  - Compilation into arithmetic circuits (Darwiche et al.)