

VARIABLE ELIMINATION

Inference

- ◇ Exact inference
 - Enumeration
 - Variable elimination
 - Junction trees and belief propagation
- ◇ Approximate inference
 - Loopy belief propagation
 - Sampling methods: likelihood weighting, Markov chain Monte Carlo
 - Variational approximation

Inference tasks

Simple queries: compute posterior marginal $\mathbf{P}(X_i|\mathbf{E} = \mathbf{e})$

e.g., $P(\text{NoGas}|\text{Gauge} = \text{empty}, \text{Lights} = \text{on}, \text{Starts} = \text{false})$

Conjunctive queries: $\mathbf{P}(X_i, X_j|\mathbf{E} = \mathbf{e}) = \mathbf{P}(X_i|\mathbf{E} = \mathbf{e})\mathbf{P}(X_j|X_i, \mathbf{E} = \mathbf{e})$

Optimal decisions: decision networks include utility information;
probabilistic inference required for $P(\text{outcome}|\text{action}, \text{evidence})$

Value of information: which evidence to seek next?

Sensitivity analysis: which probability values are most critical?

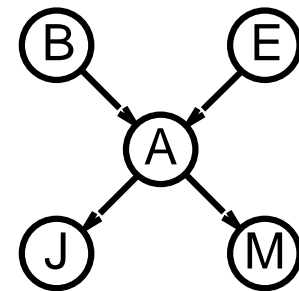
Explanation: why do I need a new starter motor?

Inference by enumeration

Slightly intelligent way to sum out variables from the joint without actually constructing its explicit representation

Simple query on the burglary network:

$$\begin{aligned} & \mathbf{P}(B|j, m) \\ &= \mathbf{P}(B, j, m) / P(j, m) \\ &= \alpha \mathbf{P}(B, j, m) \\ &= \alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) \end{aligned}$$



Rewrite full joint entries using product of CPT entries:

$$\begin{aligned} & \mathbf{P}(B|j, m) \\ &= \alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a) \end{aligned}$$

Recursive depth-first enumeration: $O(n)$ space, $O(d^n)$ time

Enumeration algorithm

function **ENUMERATION-ASK**(X, \mathbf{e}, bn) **returns** a distribution over X

inputs: X , the query variable

\mathbf{e} , observed values for variables \mathbf{E}

bn , a Bayesian network with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$

$Q(X) \leftarrow$ a distribution over X , initially empty

for each value x_i of X **do**

 extend \mathbf{e} with value x_i for X

$Q(x_i) \leftarrow$ **ENUMERATE-ALL**(**VARS**[bn], \mathbf{e})

return **NORMALIZE**($Q(X)$)

function **ENUMERATE-ALL**($vars, \mathbf{e}$) **returns** a real number

if **EMPTY?**($vars$) **then return** 1.0

$Y \leftarrow$ **FIRST**($vars$)

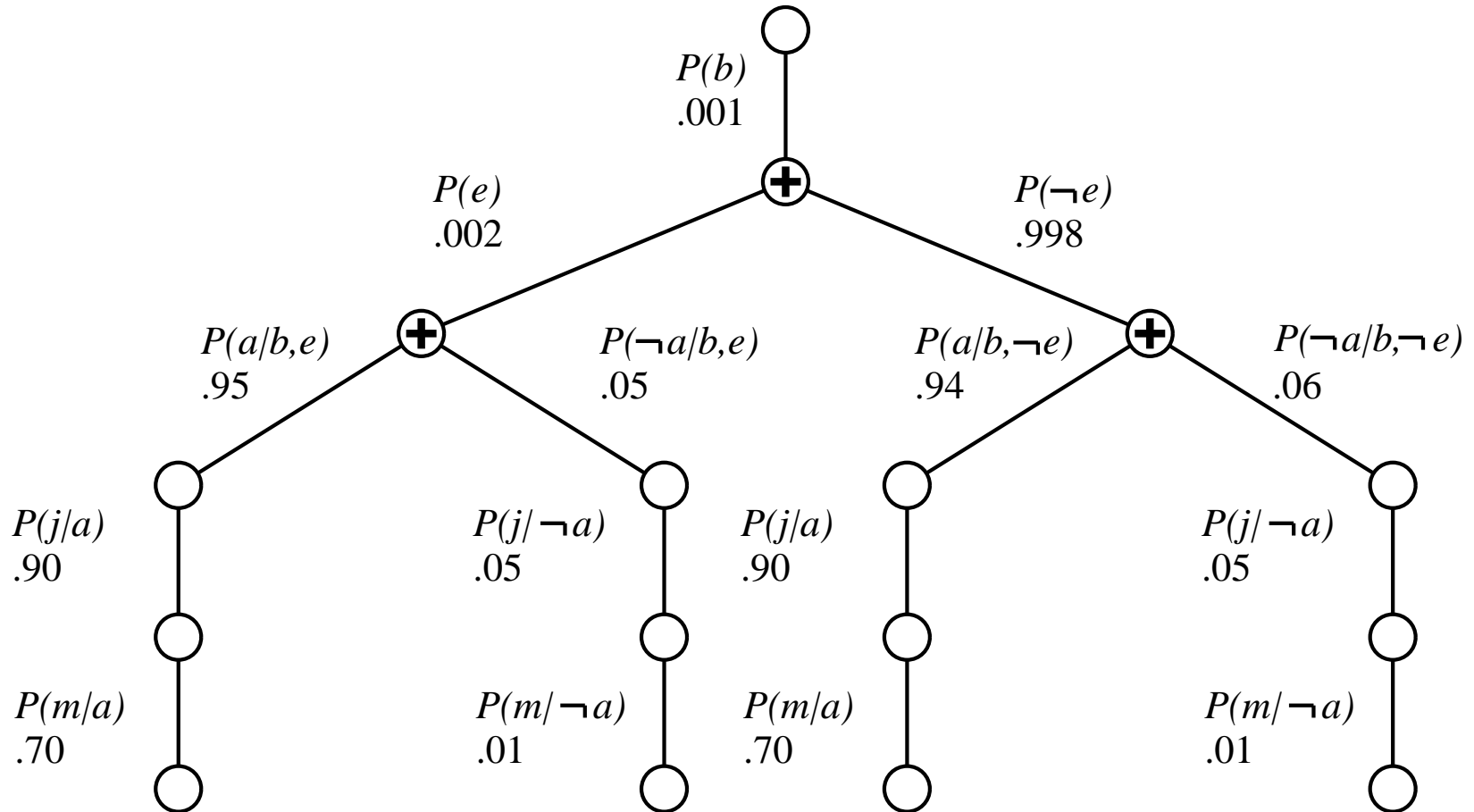
if Y has value y in \mathbf{e}

then return $P(y \mid Pa(Y)) \times$ **ENUMERATE-ALL**(**REST**($vars$), \mathbf{e})

else return $\sum_y P(y \mid Pa(Y)) \times$ **ENUMERATE-ALL**(**REST**($vars$), \mathbf{e}_y)

 where \mathbf{e}_y is \mathbf{e} extended with $Y = y$

Evaluation tree



Enumeration is inefficient: repeated computation
 e.g., computes $P(j|a)P(m|a)$ for each value of e

Inference by variable elimination

Variable elimination: carry out summations right-to-left, storing intermediate results (**factors**) to avoid recomputation

$$\begin{aligned} \mathbf{P}(B|j, m) &= \alpha \underbrace{\mathbf{P}(B)}_B \sum_e \underbrace{P(e)}_E \sum_a \underbrace{\mathbf{P}(a|B, e)}_A \underbrace{P(j|a)}_J \underbrace{P(m|a)}_M \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) P(j|a) f_M(a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) f_J(a) f_M(a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a f_A(a, b, e) f_J(a) f_M(a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) f_{\bar{A}JM}(b, e) \text{ (sum out } A) \\ &= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\ &= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b) \end{aligned}$$

Variable elimination: Basic operations

Summing out a variable from a product of factors:

move any constant factors outside the summation

add up submatrices in pointwise product of remaining factors

$$\sum_x f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \sum_x f_{i+1} \times \cdots \times f_k = f_1 \times \cdots \times f_i \times f_{\bar{X}}$$

assuming f_1, \dots, f_i do not depend on X

Pointwise product of factors f_1 and f_2 :

$$\begin{aligned} f_1(x_1, \dots, x_j, y_1, \dots, y_k) \times f_2(y_1, \dots, y_k, z_1, \dots, z_l) \\ = f(x_1, \dots, x_j, y_1, \dots, y_k, z_1, \dots, z_l) \end{aligned}$$

E.g., $f_1(a, b) \times f_2(b, c) = f(a, b, c)$

Variable elimination algorithm

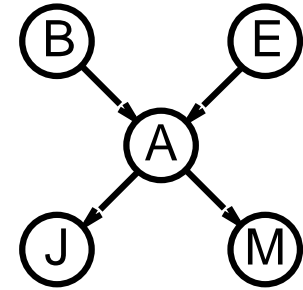
```
function ELIMINATION-ASK( $X, \mathbf{e}, bn$ ) returns a distribution over  $X$   
inputs:  $X$ , the query variable  
          $\mathbf{e}$ , evidence specified as an event  
          $bn$ , a belief network specifying joint distribution  $\mathbf{P}(X_1, \dots, X_n)$   
 $factors \leftarrow []$ ;  $vars \leftarrow \text{REVERSE}(\text{VARS}[bn])$   
for each  $var$  in  $vars$  do  
     $factors \leftarrow [\text{MAKE-FACTOR}(var, \mathbf{e}) | factors]$   
    if  $var$  is a hidden variable then  $factors \leftarrow \text{SUM-OUT}(var, factors)$   
return  $\text{NORMALIZE}(\text{POINTWISE-PRODUCT}(factors))$ 
```

Irrelevant variables

Consider the query $P(\text{JohnCalls} | \text{Burglary} = \text{true})$

$$P(J|b) = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(J|a) \sum_m P(m|a)$$

Sum over m is identically 1; M is **irrelevant** to the query



Thm 1: Y is irrelevant unless $Y \in \text{Ancestors}(\{X\} \cup \mathbf{E})$

Here, $X = \text{JohnCalls}$, $\mathbf{E} = \{\text{Burglary}\}$, and
 $\text{Ancestors}(\{X\} \cup \mathbf{E}) = \{\text{Alarm}, \text{Earthquake}\}$
so MaryCalls is irrelevant

(Compare this to backward chaining from the query in Horn clause KBs)

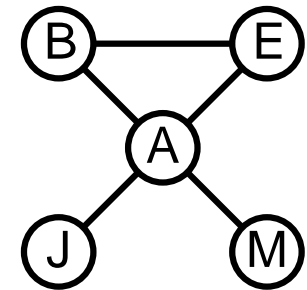
Irrelevant variables contd.

Defn: moral graph of Bayes net: marry all parents and drop arrows

Defn: **A** is m-separated from **B** by **C** iff separated by **C** in the moral graph

Thm 2: **Y** is irrelevant if m-separated from **X** by **E**

For $P(\text{JohnCalls} | \text{Alarm} = \text{true})$, both *Burglary* and *Earthquake* are irrelevant



Complexity of exact inference

Singly connected networks (or **polytrees**):

- any two nodes are connected by at most one (undirected) path
- time and space cost of variable elimination are $O(d^k n)$

Multiply connected networks:

- can reduce 3SAT to exact inference \Rightarrow NP-hard
- equivalent to **counting** 3SAT models \Rightarrow #P-complete

1. $A \vee B \vee C$
2. $C \vee D \vee \neg A$
3. $B \vee C \vee \neg D$

