Lecture notes for *Principles of Artificial Intelligence* (COMS 4720/5720)
Yan-Bin Jia, Iowa State University

Exact Inference in Bayesian Networks

Outline

I. Probabilistic query using a BN

II. Variable Elimination

III. Variable ordering and relevance

* Figures are from the textbook site.

II. Probabilistic Query

X: query variable

$\boldsymbol{E} = \{E_1, \ldots . E_m\}$: evidence variables

$\boldsymbol{e} = \{e_1, \ldots . e_m\}$: an observed event

$\boldsymbol{Y} = \{Y_1, \ldots . Y_l\}$: hidden variables

II. Probabilistic Query

X: query variable

$\boldsymbol{E} = \{E_1, \dots . E_m\}$: evidence variables

$\boldsymbol{e} = \{e_1, \dots . e_m\}$: an observed event

$\boldsymbol{Y} = \{Y_1, \dots . Y_l\}$: hidden variables

Complete variable set: $\{X\} \cup \boldsymbol{E} \cup \boldsymbol{Y}$

II. Probabilistic Query

X: query variable

$\boldsymbol{E} = \{E_1, \dots . E_m\}$: evidence variables

$\boldsymbol{e} = \{e_1, \dots . e_m\}$: an observed event

$\boldsymbol{Y} = \{Y_1, \dots . Y_l\}$: hidden variables

Complete variable set: $\{X\} \cup \boldsymbol{E} \cup \boldsymbol{Y}$

Query $\boldsymbol{P}(X \mid \boldsymbol{e})$?

II. Probabilistic Query

X: query variable

$\boldsymbol{E} = \{E_1, \ldots. E_m\}$: evidence variables

$\boldsymbol{e} = \{e_1, \ldots. e_m\}$: an observed event

$\boldsymbol{Y} = \{Y_1, \ldots. Y_l\}$: hidden variables

Complete variable set: $\{X\} \cup \boldsymbol{E} \cup \boldsymbol{Y}$

Query $\boldsymbol{P}(X \mid \boldsymbol{e})$?

// $\boldsymbol{P}(Burglary \mid j, m)$

$\boldsymbol{P}(Burglary \mid JohnCalls = true, MaryCalls = true) = \langle 0.284, 0.716 \rangle$

II. Probabilistic Query

X: query variable

$\boldsymbol{E} = \{E_1, \ldots. E_m\}$: evidence variables

$\boldsymbol{e} = \{e_1, \ldots. e_m\}$: an observed event

$\boldsymbol{Y} = \{Y_1, \ldots. Y_l\}$: hidden variables

Complete variable set: $\{X\} \cup \boldsymbol{E} \cup \boldsymbol{Y}$

Query $\boldsymbol{P}(X \mid \boldsymbol{e})$?

// $\boldsymbol{P}(Burglary \mid j, m)$

$\boldsymbol{P}(Burglary \mid JohnCalls = true, MaryCalls = true) = \langle 0.284, 0.716 \rangle$

We will discuss exact algorithms for posterior probability computation.

Inference by Enumeration

$$\boldsymbol{P}(X, \boldsymbol{e}) = \boldsymbol{P}(X \mid \boldsymbol{e}) P(\boldsymbol{e})$$

Vector

Scalar

Vector of evidences

Inference by Enumeration

$$\boldsymbol{P}(X, \boldsymbol{e}) = \boldsymbol{P}(X \mid \boldsymbol{e})P(\boldsymbol{e})$$

Vector

Scalar

Vector of evidences

$$\boldsymbol{P}(X \mid \boldsymbol{e}) = \alpha \boldsymbol{P}(X, \boldsymbol{e}) = \alpha \sum_{\boldsymbol{y}} \boldsymbol{P}(X, \boldsymbol{e}, \boldsymbol{y})$$

Inference by Enumeration

$$\mathbf{P}(X,\mathbf{e}) = \mathbf{P}(X \mid \mathbf{e})P(\mathbf{e})$$

Vector

Scalar

Vector of evidences

$$\mathbf{P}(X \mid \mathbf{e}) = \alpha\mathbf{P}(X,\mathbf{e}) = \alpha\sum_{\mathbf{y}}\mathbf{P}(X,\mathbf{e},\mathbf{y})$$

Inference by Enumeration

$$\mathbf{P}(X, \mathbf{e}) = \mathbf{P}(X \mid \mathbf{e}) P(\mathbf{e})$$

Vector

Scalar

Vector of evidences

$$\mathbf{P}(X \mid \mathbf{e}) = \alpha \mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$$

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

Inference by Enumeration

$$\mathbf{P}(X,\mathbf{e}) = \mathbf{P}(X \mid \mathbf{e})P(\mathbf{e})$$

Vector

Scalar

Vector of evidences

$$\mathbf{P}(X \mid \mathbf{e}) = \alpha\mathbf{P}(X,\mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X,\mathbf{e},\mathbf{y})$$

$$P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

computable as products of conditional probabilities from the BN.

Inference by Enumeration

$$\mathbf{P}(X, \mathbf{e}) = \mathbf{P}(X \mid \mathbf{e}) P(\mathbf{e})$$

Vector

Scalar

Vector of evidences

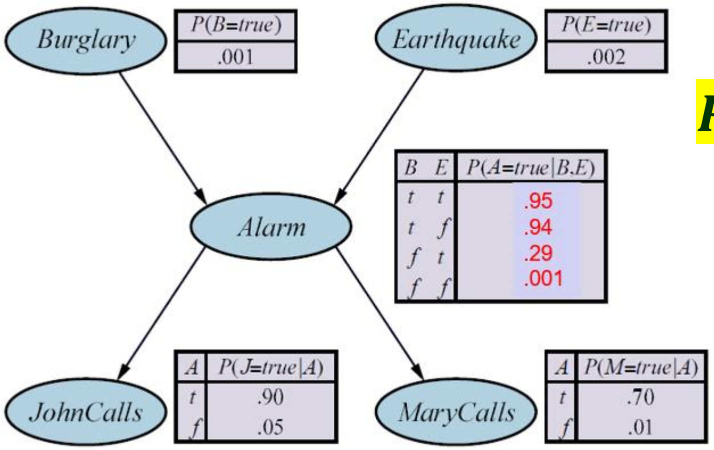
$$\mathbf{P}(X \mid \mathbf{e}) = \alpha \mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$$

$$P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

computable as products of conditional probabilities from the BN.

- Answer the query $\mathbf{P}(X \mid \mathbf{e})$ using a BN.

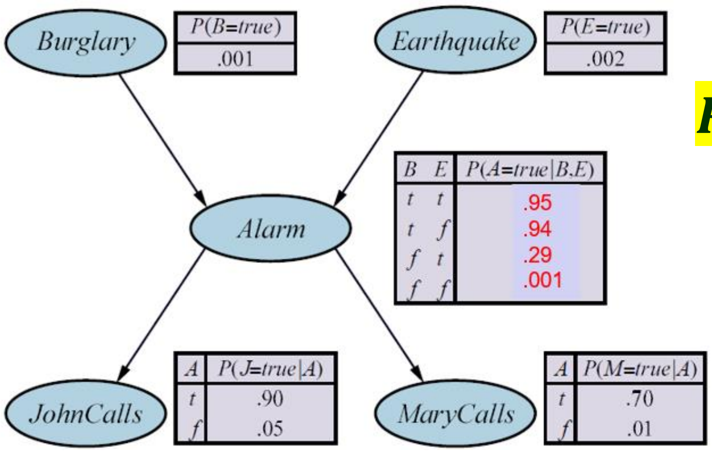
Burglary Example (revisited)



$\mathbf{P}(Burglary \mid JohnCalls = true, MaryCalls = true)$?

Hidden variables: $Y = \{Earthquake, Alarm\}$

Burglary Example (revisited)

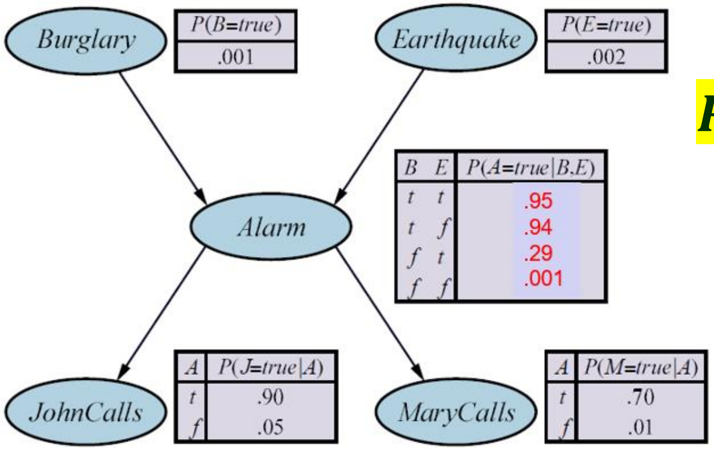


$\mathbf{P}(Burglary \mid JohnCalls = true, MaryCalls = true)?$

Hidden variables: $Y = \{Earthquake, Alarm\}$

$$\mathbf{P}(B \mid j, m) = \alpha \mathbf{P}(B, j, m) = \alpha \sum_{e' \in \{e, \neg e\}} \sum_{a' \in \{a, \neg a\}} \mathbf{P}(B, j, m, e', a')$$

Burglary Example (revisited)



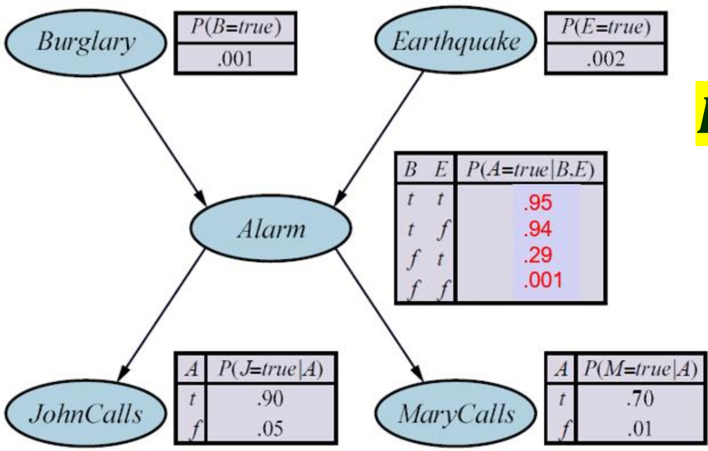
$\boldsymbol{P}(Burglary \mid JohnCalls = true, MaryCalls = true)$?

Hidden variables: $Y = \{Earthquake, Alarm\}$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{P}(B, j, m) = \alpha \sum_{e' \in \{e, \neg e\}} \sum_{a' \in \{a, \neg a\}} \boldsymbol{P}(B, j, m, e', a')$$

// the textbook uses e, a in the summation. this is incorrect
// because e is a constant value standing for
// $Earthquake = true$. e cannot be overloaded. the index
// variable e' can take on both e and $\neg e$ needed for the
// summation. meanwhile, we cannot use E here because
// then we would be computing the joint distribution of B and E.

Burglary Example (revisited)



$\mathbf{P}(Burglary \mid JohnCalls = true, MaryCalls = true)$?

Hidden variables: $Y = \{Earthquake, Alarm\}$

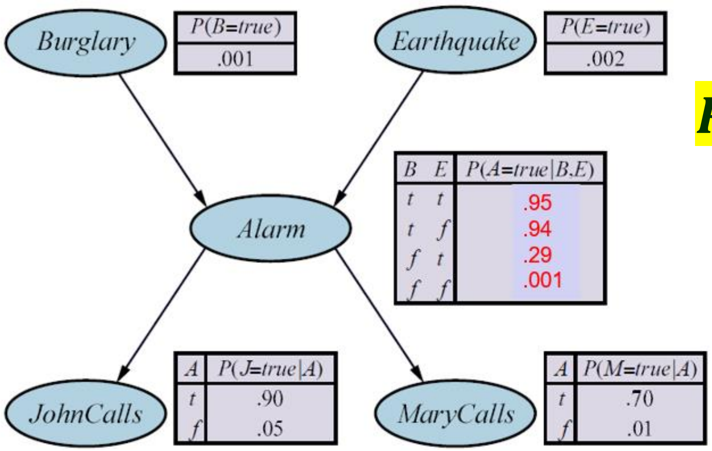
$$\mathbf{P}(B \mid j, m) = \alpha \mathbf{P}(B, j, m) = \alpha \sum_{e' \in \{e, \neg e\}} \sum_{a' \in \{a, \neg a\}} \mathbf{P}(B, j, m, e', a')$$

// the textbook uses e, a in the summation. this is incorrect
// because e is a constant value standing for
// *Earthquake = true*. e cannot be overloaded. the index
// variable e' can take on both e and $\neg e$ needed for the
// summation. meanwhile, we cannot use E here because
// then we would be computing the joint distribution of B and E.

evaluated for b (*Burglary = true*) based on

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

Burglary Example (revisited)



$\boldsymbol{P}(Burglary \mid JohnCalls = true, MaryCalls = true)$?

Hidden variables: $Y = \{Earthquake, Alarm\}$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{P}(B, j, m) = \alpha \sum_{e' \in \{e, \neg e\}} \sum_{a' \in \{a, \neg a\}} \boldsymbol{P}(B, j, m, e', a')$$

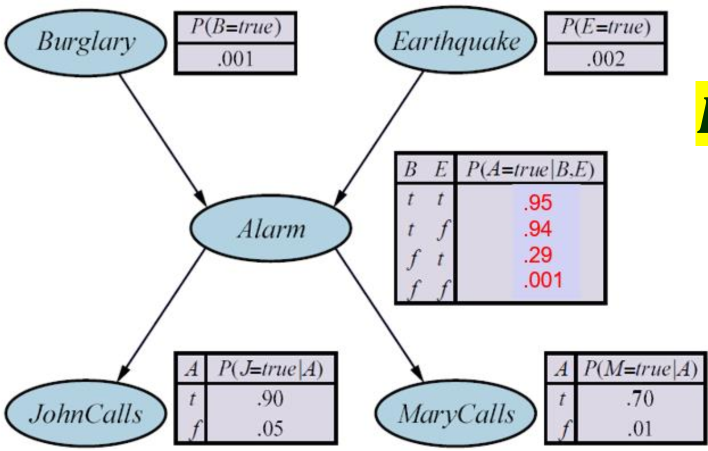
// the textbook uses e, a in the summation. this is incorrect
// because e is a constant value standing for
// $Earthquake = true$. e cannot be overloaded. the index
// variable e' can take on both e and $\neg e$ needed for the
// summation. meanwhile, we cannot use E here because
// then we would be computing the joint distribution of B and E.

evaluated for b ($Burglary = true$) based on

$$P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

$$P(b \mid j, m) = \alpha \sum_{e'} \sum_{a'} P(b) P(e') P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

Burglary Example (revisited)



$\boldsymbol{P}(Burglary \mid JohnCalls = true, MaryCalls = true)$?

Hidden variables: $Y = \{Earthquake, Alarm\}$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{P}(B, j, m) = \alpha \sum_{e' \in \{e, \neg e\}} \sum_{a' \in \{a, \neg a\}} \boldsymbol{P}(B, j, m, e', a')$$

// the textbook uses e, a in the summation. this is incorrect
// because e is a constant value standing for
// *Earthquake = true*. e cannot be overloaded. the index
// variable e' can take on both e and $\neg e$ needed for the
// summation. meanwhile, we cannot use E here because
// then we would be computing the joint distribution of B and E.

evaluated for b (*Burglary = true*) based on

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

$$P(b \mid j, m) = \alpha \sum_{e'} \sum_{a'} P(b) P(e') P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

In the general case with n hidden variables, there are 2^n summands, each as a product requires $O(n)$ computation time.

Expression Tree

Take advantage of the nested structure to move summations inwards as far as possible.

$$P(b \mid j, m) = \alpha \sum_{e'} \sum_{a'} P(b)P(e')P(a' \mid b, e')P(j \mid a')P(m \mid a')$$

Expression Tree

Take advantage of the nested structure to move summations inwards as far as possible.

$$P(b \mid j, m) = \alpha \sum_{e'} \sum_{a'} P(b) P(e') P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

Expression Tree

Take advantage of the nested structure to move summations inwards as far as possible.

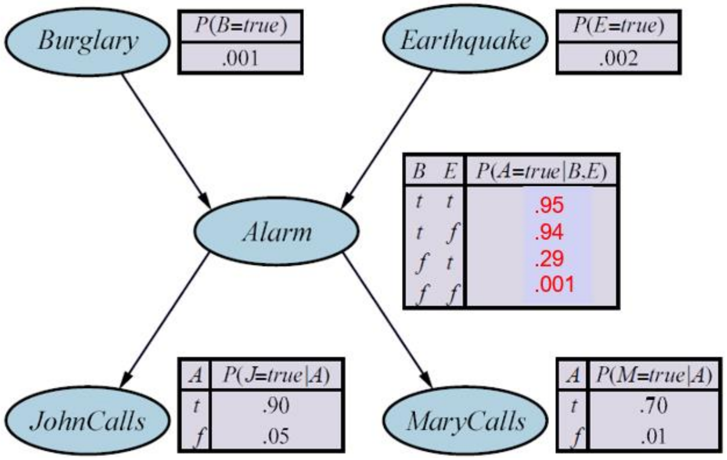
$$P(b \mid j, m) = \alpha \sum_{e'} \sum_{a'} P(b) P(e') P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

Expression Tree

$$P(b \mid j, m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

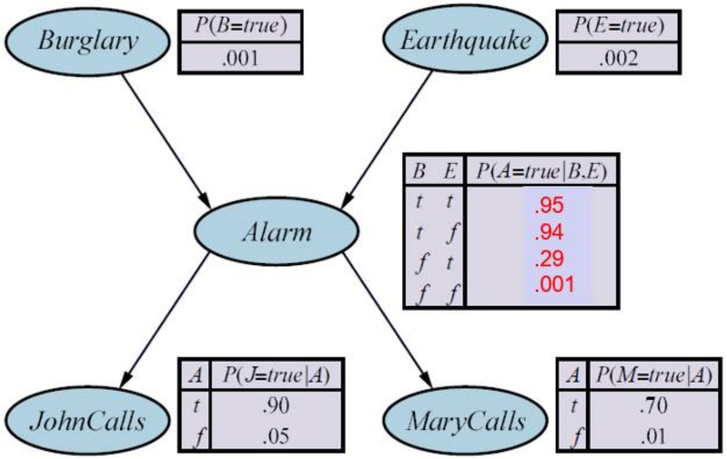
Expression Tree

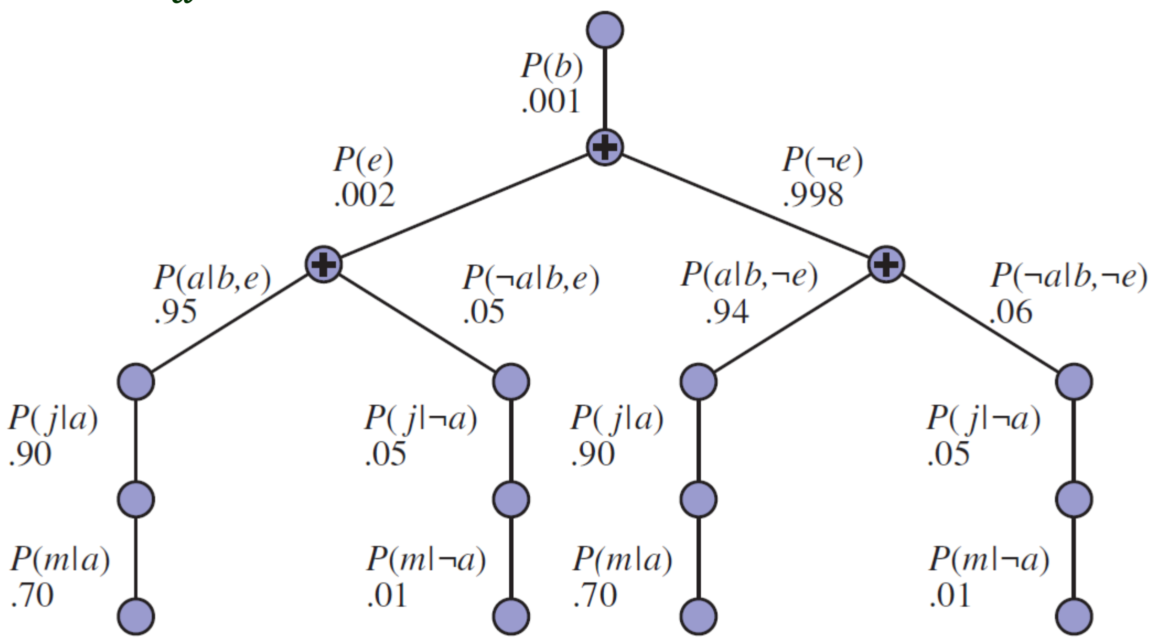
$$P(b \mid j,m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b,e') P(j \mid a') P(m \mid a')$$



Expression Tree

$$P(b \mid j, m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

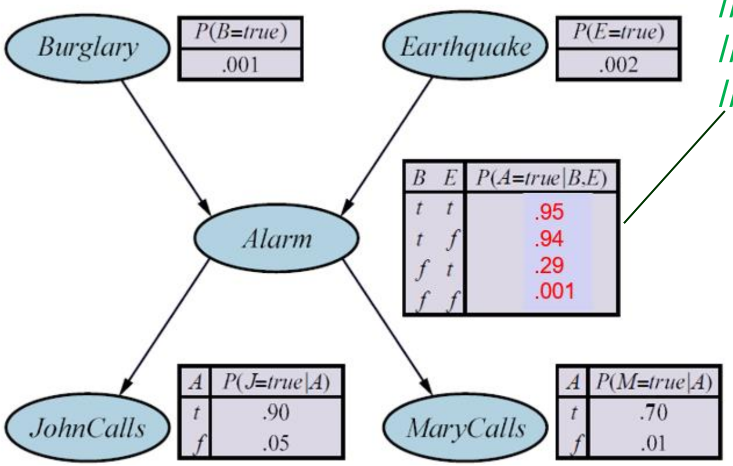


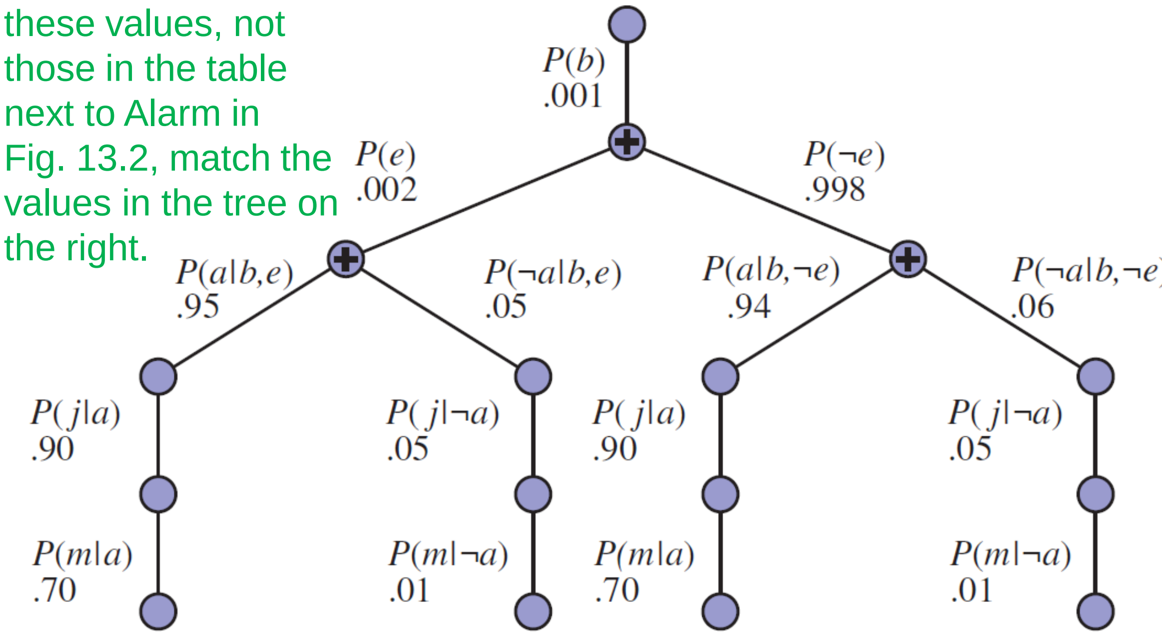


Expression Tree

$$P(b \mid j,m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b,e') P(j \mid a') P(m \mid a')$$

// these values, not
// those in the table
// next to Alarm in
// Fig. 13.2, match the
// values in the tree on
// the right.

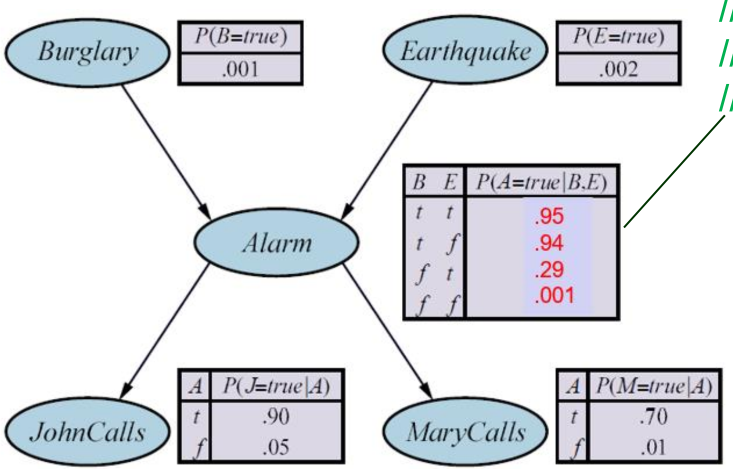


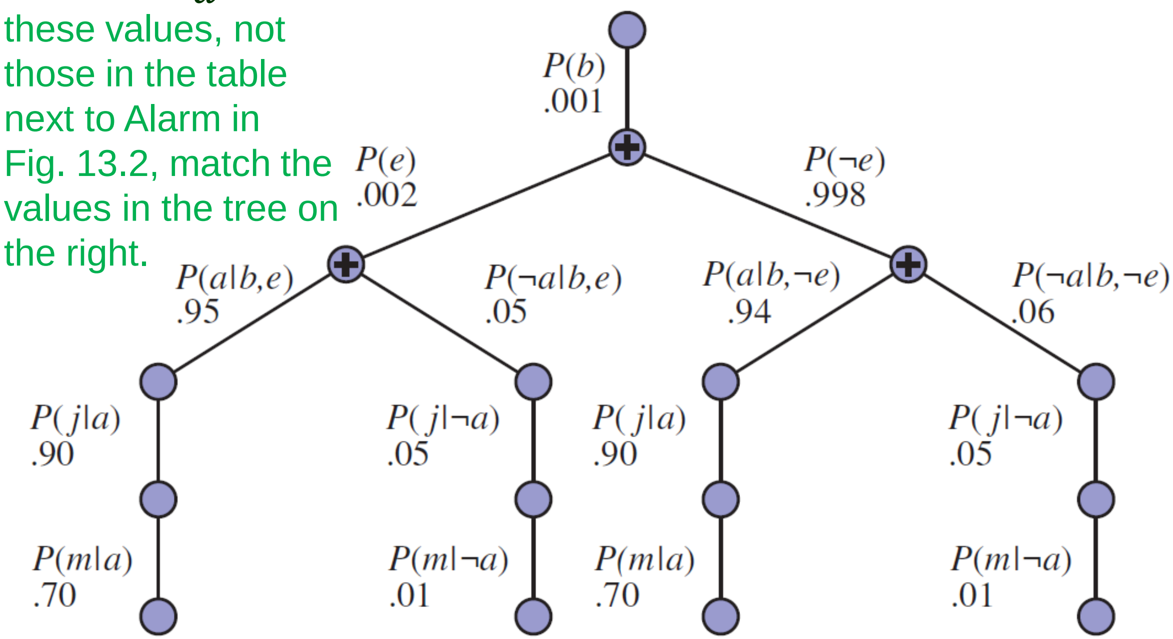


Expression Tree

$$P(b \mid j,m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b,e') P(j \mid a') P(m \mid a')$$

// these values, not
// those in the table
// next to Alarm in
// Fig. 13.2, match the
// values in the tree on
// the right.



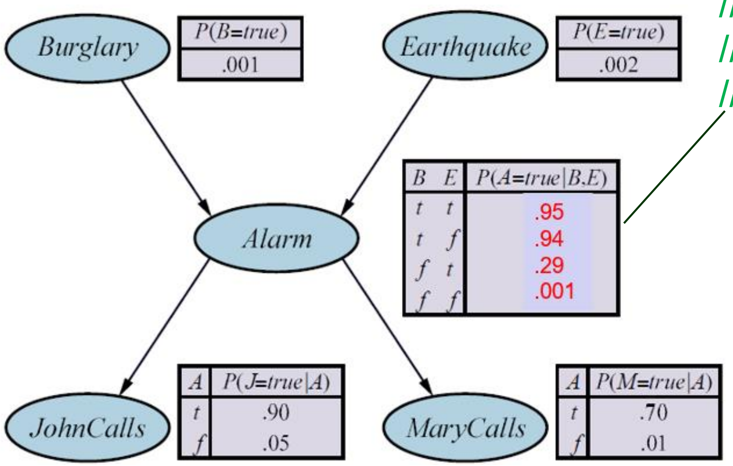


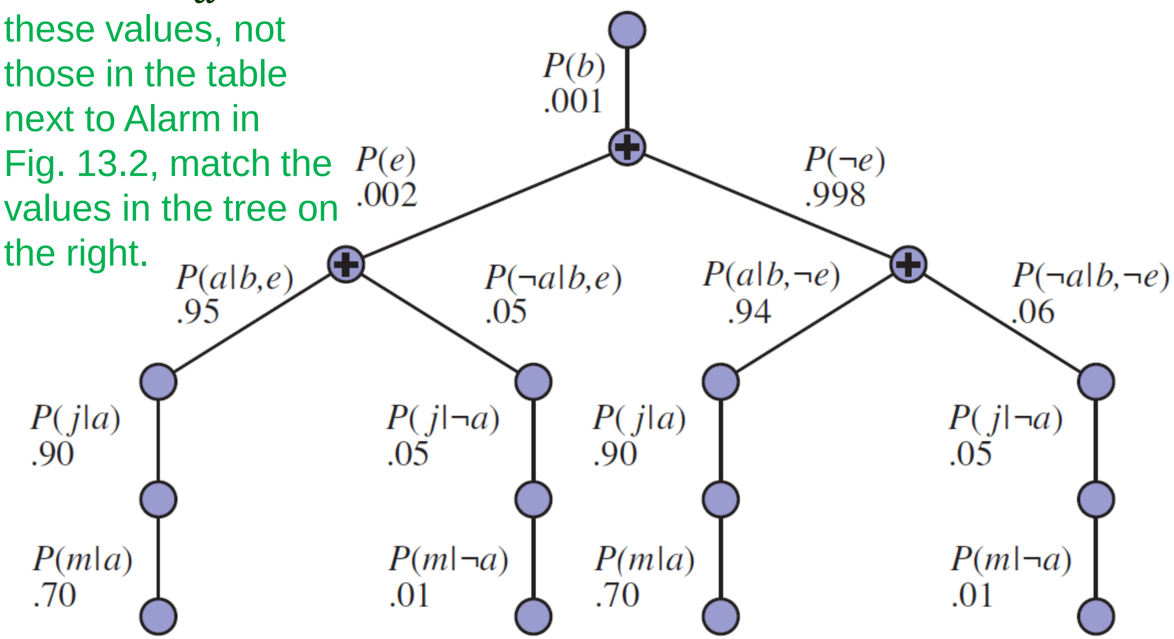
- Multiply values along each path.
- Sum at the "+" nodes.

Expression Tree

$$P(b \mid j,m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b,e') P(j \mid a') P(m \mid a')$$

// these values, not
// those in the table
// next to Alarm in
// Fig. 13.2, match the
// values in the tree on
// the right.





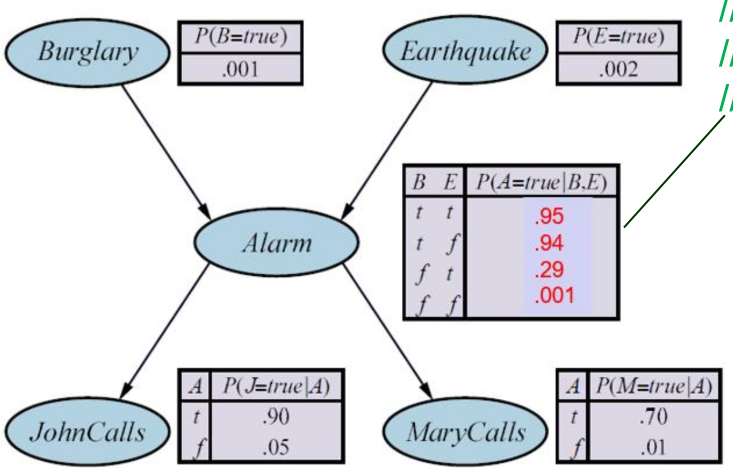
- Multiply values along each path.
- Sum at the “+” nodes.

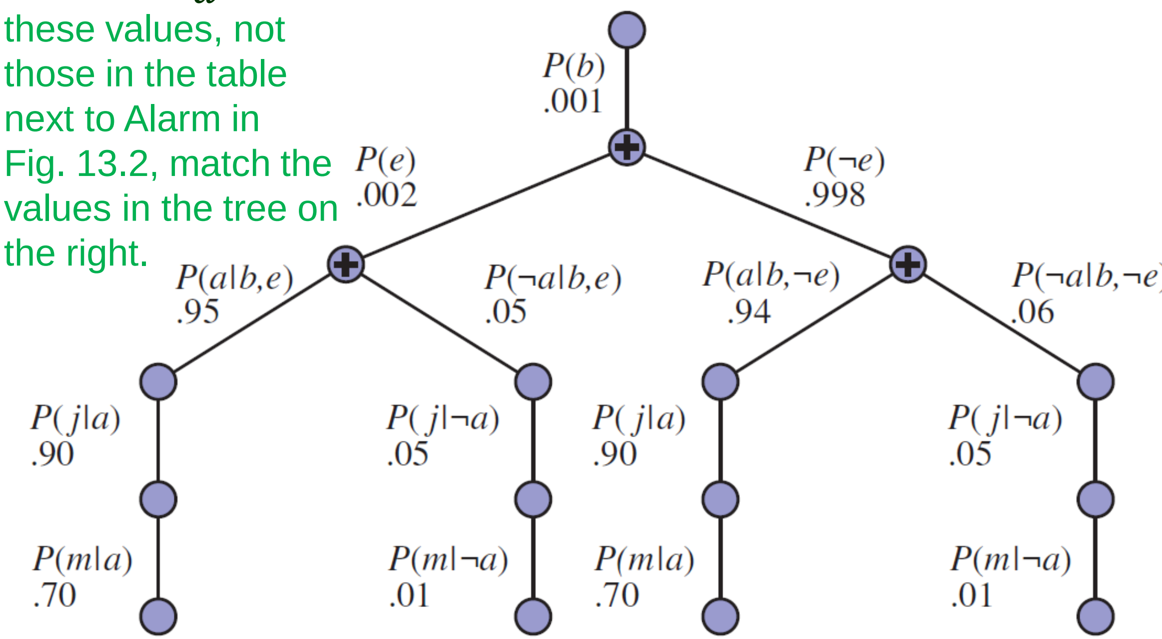
$$P(b \mid j,m) = \alpha \times 0.00059224$$

Expression Tree

$$P(b \mid j,m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b,e') P(j \mid a') P(m \mid a')$$

// these values, not
// those in the table
// next to Alarm in
// Fig. 13.2, match the
// values in the tree on
// the right.





- Multiply values along each path.
- Sum at the “+” nodes.

$$P(b \mid j,m) = \alpha \times 0.00059224$$

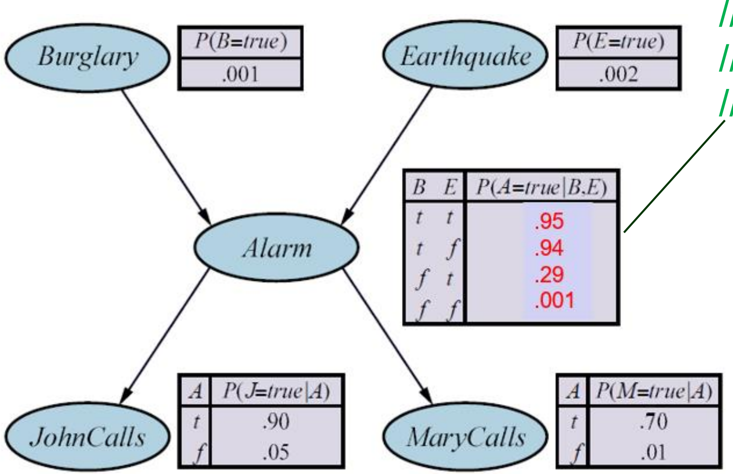
$$P(\neg b \mid j,m) = \alpha \times 0.0014919$$

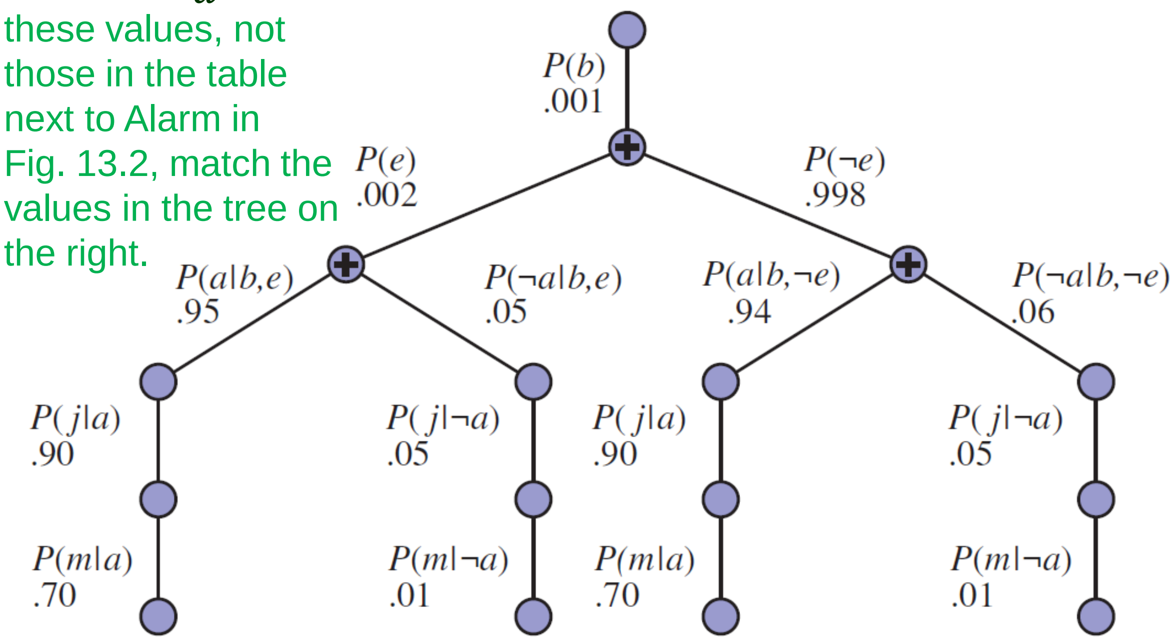
(computed using a tree of the same structure but with b replaced by $\neg b$ and probabilities changed)

Expression Tree

$$P(b \mid j,m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b,e') P(j \mid a') P(m \mid a')$$

// these values, not
// those in the table
// next to Alarm in
// Fig. 13.2, match the
// values in the tree on
// the right.





- Multiply values along each path.
- Sum at the “+” nodes.

$$P(b \mid j,m) = \alpha \times 0.00059224$$
$$P(\neg b \mid j,m) = \alpha \times 0.0014919$$

$\Longrightarrow$

$$\mathbf{P}(B \mid j,m) = \alpha\langle 0.00059224, 0.0014919\rangle \approx \langle 0.284, 0.716\rangle$$

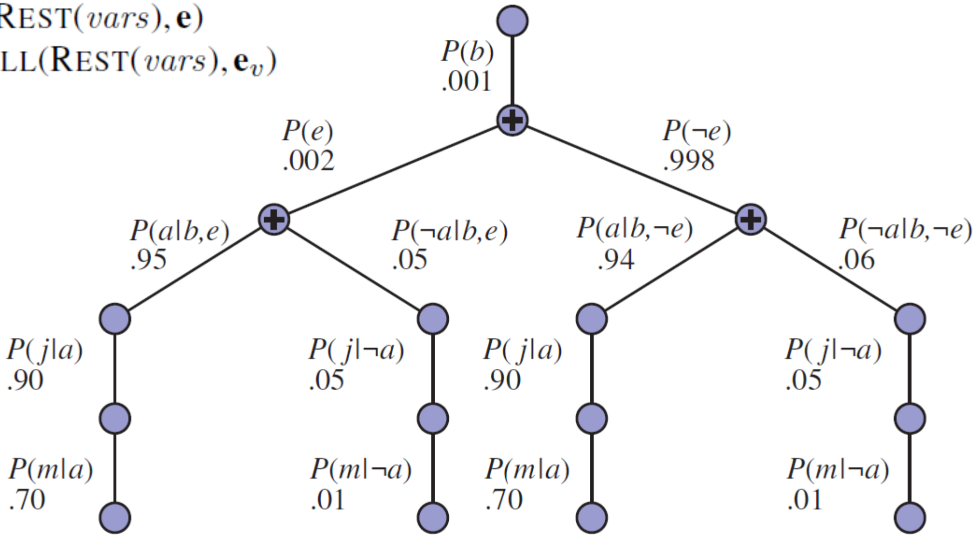
(computed using a tree of the same structure but with b replaced by $\neg b$ and probabilities changed)

Enumeration Algorithm

function ENUMERATION-ASK(X, **e**, bn) **returns** a distribution over X
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayes net with variables $vars$

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL($vars$, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is **e** extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

function ENUMERATE-ALL($vars$, **e**) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $V \leftarrow$ FIRST($vars$)
  if V is an evidence variable with value v in **e**
    then return $P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), **e**)
    else return $\sum_v P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_v$)
      where $\mathbf{e}_v$ is **e** extended with $V = v$



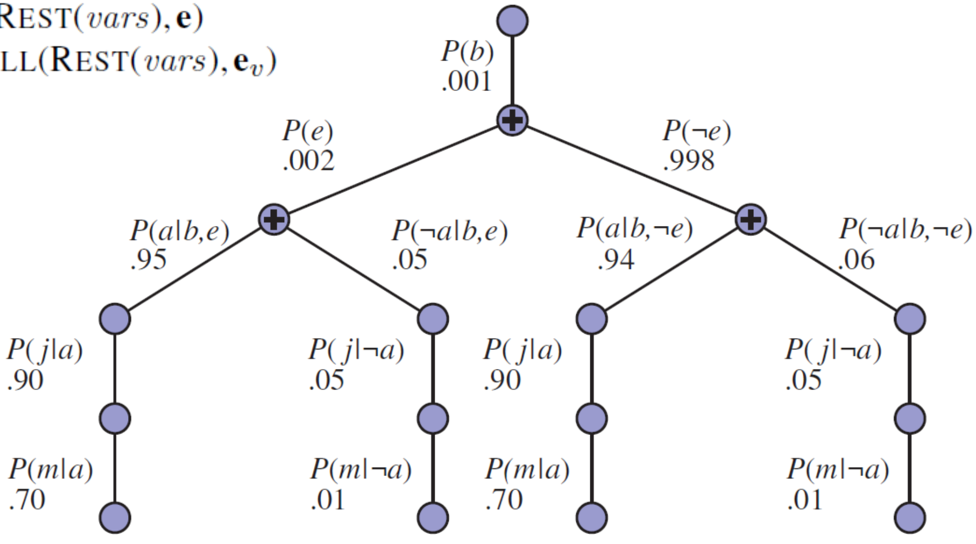
Enumeration Algorithm

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 Bayes net with variables $vars$

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL($vars$, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is $\mathbf{e}$ extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

function ENUMERATE-ALL($vars$, $\mathbf{e}$) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $V \leftarrow$ FIRST($vars$)
  if V is an evidence variable with value v in $\mathbf{e}$
    then return $P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}$)
    else return $\sum_v P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_v$)
      where $\mathbf{e}_v$ is $\mathbf{e}$ extended with $V = v$

- Depth-first recursion



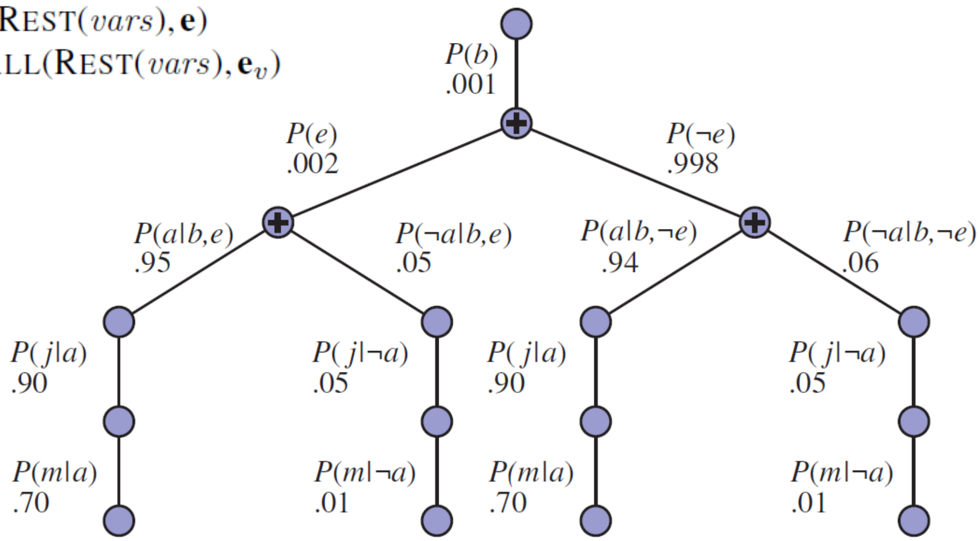
Enumeration Algorithm

function ENUMERATION-ASK(X, **e**, bn) **returns** a distribution over X
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayes net with variables $vars$

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL($vars$, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is **e** extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

function ENUMERATE-ALL($vars$, **e**) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $V \leftarrow$ FIRST($vars$)
  if V is an evidence variable with value v in **e**
    then return $P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), **e**)
    else return $\sum_v P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_v$)
      where $\mathbf{e}_v$ is **e** extended with $V = v$

- Depth-first recursion
- $O(2^n)$ time for n variables

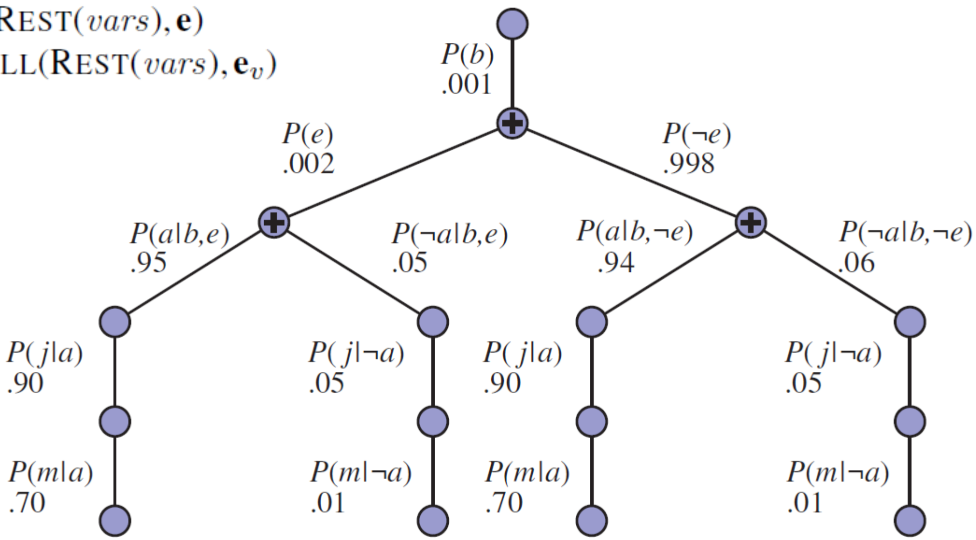


Enumeration Algorithm

```
function ENUMERATION-ASK(X, e, bn) returns a distribution over X
  inputs: X, the query variable
          e, observed values for variables E
          bn, a Bayes net with variables vars

  Q(X) ← a distribution over X, initially empty
  for each value x_i of X do
      Q(x_i) ← ENUMERATE-ALL(vars, e_{x_i})
         where e_{x_i} is e extended with X = x_i
  return NORMALIZE(Q(X))

function ENUMERATE-ALL(vars, e) returns a real number
  if EMPTY?(vars) then return 1.0
  V ← FIRST(vars)
  if V is an evidence variable with value v in e
      then return P(v | parents(V)) × ENUMERATE-ALL(REST(vars), e)
      else return Σ_v P(v | parents(V)) × ENUMERATE-ALL(REST(vars), e_v)
         where e_v is e extended with V = v
```



- Depth-first recursion
- $O(2^n)$ time for n variables
- Repeated evaluations of the same subexpressions

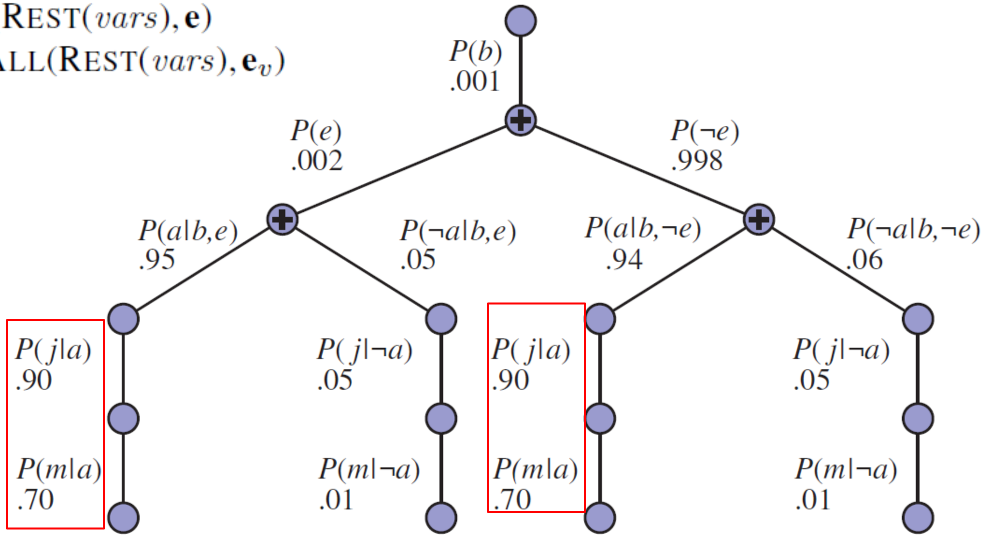
Enumeration Algorithm

function ENUMERATION-ASK(X, **e**, bn) **returns** a distribution over X
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayes net with variables $vars$

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL($vars$, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is **e** extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

function ENUMERATE-ALL($vars$, **e**) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $V \leftarrow$ FIRST($vars$)
  if V is an evidence variable with value v in **e**
    then return $P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), **e**)
    else return $\sum_v P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_v$)
      where $\mathbf{e}_v$ is **e** extended with $V = v$

- Depth-first recursion
- $O(2^n)$ time for n variables
- Repeated evaluations of the same subexpressions



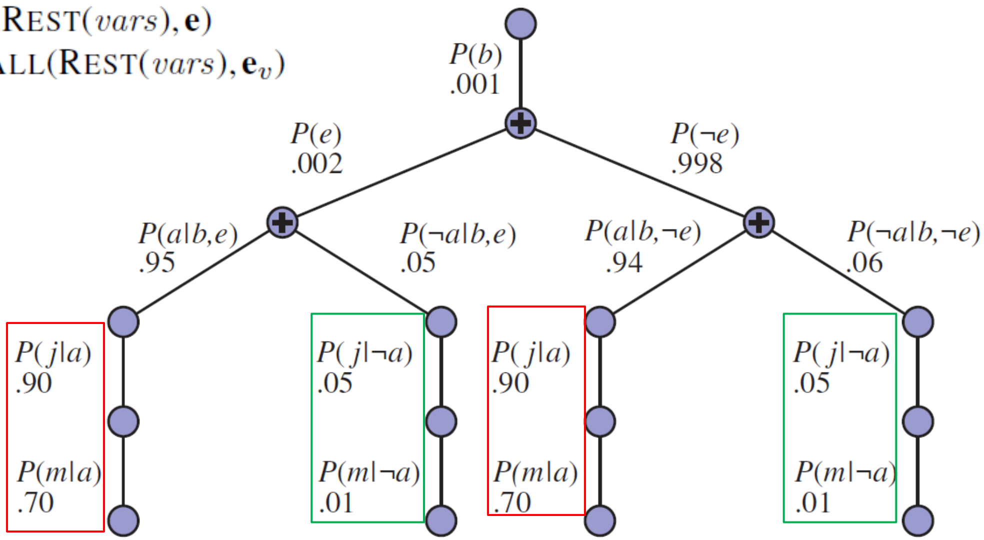
Enumeration Algorithm

function ENUMERATION-ASK(X, **e**, bn) **returns** a distribution over X
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayes net with variables $vars$

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL($vars$, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is **e** extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

function ENUMERATE-ALL($vars$, **e**) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $V \leftarrow$ FIRST($vars$)
  if V is an evidence variable with value v in **e**
    then return $P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), **e**)
    else return $\sum_v P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_v$)
      where $\mathbf{e}_v$ is **e** extended with $V = v$

- Depth-first recursion
- $O(2^n)$ time for n variables
- Repeated evaluations of the same subexpressions



III. Variable Elimination

Idea Do the calculation once and save the results for later use (like dynamic programming).

III. Variable Elimination

Idea Do the calculation once and save the results for later use (like dynamic programming).

- Evaluate an expression from right to left (i.e., bottom up in the expression tree).

$$P(b \mid j, m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

III. Variable Elimination

Idea Do the calculation once and save the results for later use (like dynamic programming).

- Evaluate an expression from right to left (i.e., bottom up in the expression tree).

$$P(b \mid j, m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

III. Variable Elimination

Idea Do the calculation once and save the results for later use (like dynamic programming).

- Evaluate an expression from right to left (i.e., bottom up in the expression tree).

$$P(b \mid j,m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b,e') P(j \mid a') P(m \mid a')$$

$$\boldsymbol{P}(B \mid j,m) = \alpha \boldsymbol{P}(B) \sum_{e'} P(e') \sum_{a'} \boldsymbol{P}(a' \mid B,e') P(j \mid a') P(m \mid a')$$

III. Variable Elimination

Idea Do the calculation once and save the results for later use (like dynamic programming).

- Evaluate an expression from right to left (i.e., bottom up in the expression tree).

$$P(b \mid j, m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

$$\mathbf{P}(B \mid j, m) = \alpha \underbrace{\mathbf{P}(B)}_{} \sum_{e'} \underbrace{P(e')}_{} \sum_{a'} \underbrace{\mathbf{P}(a' \mid B, e')}_{} \underbrace{P(j \mid a')}_{} \underbrace{P(m \mid a')}_{}$$

factors: $\boldsymbol{f}_1(B)$ $\boldsymbol{f}_2(E)$ $\boldsymbol{f}_3(A, B, E)$ $\boldsymbol{f}_4(A)$ $\boldsymbol{f}_5(A)$

III. Variable Elimination

Idea Do the calculation once and save the results for later use (like dynamic programming).

- Evaluate an expression from right to left (i.e., bottom up in the expression tree).

$$P(b \mid j, m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \underbrace{\boldsymbol{P}(B)} \sum_{e'} \underbrace{P(e')} \sum_{a'} \underbrace{\boldsymbol{P}(a' \mid B, e')} \underbrace{P(j \mid a')} \underbrace{P(m \mid a')}$$

factors: $\boldsymbol{f}_1(B)$ $\boldsymbol{f}_2(E)$ $\boldsymbol{f}_3(A, B, E)$ $\boldsymbol{f}_4(A)$ $\boldsymbol{f}_5(A)$

dimensions: 2×1

III. Variable Elimination

Idea Do the calculation once and save the results for later use (like dynamic programming).

- Evaluate an expression from right to left (i.e., bottom up in the expression tree).

$$P(b \mid j, m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \underbrace{\boldsymbol{P}(B)}_{} \sum_{e'} \underbrace{P(e')}_{} \sum_{a'} \underbrace{\boldsymbol{P}(a' \mid B, e')}_{} \underbrace{P(j \mid a')}_{} \underbrace{P(m \mid a')}_{}$$

factors: $\boldsymbol{f}_1(B)$ $\boldsymbol{f}_2(E)$ $\boldsymbol{f}_3(A, B, E)$ $\boldsymbol{f}_4(A)$ $\boldsymbol{f}_5(A)$

dimensions: 2×1 2×1 2×1 $2 \times 2 \times 2$ 2×1 2×1

// distribution now denoted as
// a column vector or, for
// compactness, a tensor (i.e.,
// a multi-dimensional array).

III. Variable Elimination

Idea Do the calculation once and save the results for later use (like dynamic programming).

- Evaluate an expression from right to left (i.e., bottom up in the expression tree).

$$P(b \mid j, m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \underbrace{\boldsymbol{P}(B)}_{\boldsymbol{f}_1(B)} \sum_{e'} \underbrace{P(e')}_{\boldsymbol{f}_2(E)} \sum_{a'} \underbrace{\boldsymbol{P}(a' \mid B, e')}_{\boldsymbol{f}_3(A,B,E)} \underbrace{P(j \mid a')}_{\boldsymbol{f}_4(A)} \underbrace{P(m \mid a')}_{\boldsymbol{f}_5(A)}$$

factors: $\boldsymbol{f}_1(B)$ $\boldsymbol{f}_2(E)$ $\boldsymbol{f}_3(A,B,E)$ $\boldsymbol{f}_4(A)$ $\boldsymbol{f}_5(A)$

dimensions: 2×1 2×1 2×1 $2 \times 2 \times 2$ 2×1 2×1

// distribution now denoted as
// a column vector or, for
// compactness, a tensor (i.e.,
// a multi-dimensional array).

$$\boldsymbol{f}_5(A) = \begin{pmatrix} P(m \mid a) \\ P(m \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.70 \\ 0.01 \end{pmatrix}$$

III. Variable Elimination

Idea Do the calculation once and save the results for later use (like dynamic programming).

- Evaluate an expression from right to left (i.e., bottom up in the expression tree).

$$P(b \mid j, m) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e') P(j \mid a') P(m \mid a')$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \underbrace{\boldsymbol{P}(B)} \sum_{e'} \underbrace{P(e')} \sum_{a'} \underbrace{\boldsymbol{P}(a' \mid B, e')} \underbrace{P(j \mid a')} \underbrace{P(m \mid a')}$$

factors: $\boldsymbol{f}_1(B)$ $\boldsymbol{f}_2(E)$ $\boldsymbol{f}_3(A, B, E)$ $\boldsymbol{f}_4(A)$ $\boldsymbol{f}_5(A)$

dimensions: 2×1 2×1 2×1 $2 \times 2 \times 2$ 2×1 2×1

// distribution now denoted as
// a column vector or, for
// compactness, a tensor (i.e.,
// a multi-dimensional array).

$$\boldsymbol{f}_4(A) = \begin{pmatrix} P(j \mid a) \\ P(j \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix} \quad \boldsymbol{f}_5(A) = \begin{pmatrix} P(m \mid a) \\ P(m \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.70 \\ 0.01 \end{pmatrix}$$

Evaluation Example

Pointwise product operation

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{e'} \boldsymbol{f_2}(e') \times \underbrace{\sum_{a'} \boldsymbol{f_3}(a', B, e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a')}_{\boldsymbol{f}_6(B, e')}$$

Evaluation Example

Pointwise product operation

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{e'} \boldsymbol{f_2}(e') \times \underbrace{\sum_{a'} \boldsymbol{f_3}(a', B, e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a')}_{\boldsymbol{f}_6(B, e')}$$

- Sum out a' from the product of $\boldsymbol{f}_3, \boldsymbol{f}_4, \boldsymbol{f}_5$.

$$\boldsymbol{f}_6(B, e') = \sum_{a' \in \{a, \neg a\}} \boldsymbol{f_3}(a', B, e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a')$$

2×2

Evaluation Example

Pointwise product operation

$$\boldsymbol{P}(B \mid j,m) = \alpha \boldsymbol{f_1}(B) \times \sum_{e'} \boldsymbol{f_2}(e') \times \underbrace{\sum_{a'} \boldsymbol{f_3}(a',B,e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a')}_{\boldsymbol{f}_6(B,e')}$$

- Sum out a' from the product of $\boldsymbol{f}_3, \boldsymbol{f}_4, \boldsymbol{f}_5$.

$$\underset{2\times 2}{\boldsymbol{f}_6(B,e')} = \sum_{a' \in \{a,\neg a\}} \boldsymbol{f_3}(a',B,e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a')$$

$$= \left(\underset{2\times 2}{\boldsymbol{f}_3(a,B,e')} \times P(j \mid a) \times P(m \mid a)\right) + \left(\underset{2\times 2}{\boldsymbol{f}_3(\neg a,B,e')} \times P(j \mid \neg a) \times P(m \mid \neg a)\right)$$

Evaluation Example

Pointwise product operation

$$\mathbf{P}(B \mid j,m) = \alpha \mathbf{f_1}(B) \times \sum_{e'} \mathbf{f_2}(e') \times \underbrace{\sum_{a'} \mathbf{f_3}(a',B,e') \times \mathbf{f_4}(a') \times \mathbf{f_5}(a')}_{\mathbf{f}_6(B,e')}$$

- Sum out a' from the product of $\mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_5$.

$$\underset{2\times 2}{\mathbf{f}_6(B,e')} = \sum_{a' \in \{a,\neg a\}} \mathbf{f_3}(a',B,e') \times \mathbf{f_4}(a') \times \mathbf{f_5}(a')$$

$$= \left(\underset{2\times 2}{\mathbf{f}_3(a,B,e')} \times P(j \mid a) \times P(m \mid a)\right) + \left(\underset{2\times 2}{\mathbf{f}_3(\neg a,B,e')} \times P(j \mid \neg a) \times P(m \mid \neg a)\right)$$

$$\mathbf{P}(B \mid j,m) = \alpha \mathbf{f_1}(B) \times \sum_{e'} \mathbf{f}_2(e') \times \mathbf{f}_6(B,e')$$

Evaluation Example

Pointwise product operation

$$\mathbf{P}(B \mid j,m) = \alpha \boldsymbol{f}_{\mathbf{1}}(B) \times \sum_{e'} \boldsymbol{f}_{\mathbf{2}}(e') \times \underbrace{\sum_{a'} \boldsymbol{f}_{\mathbf{3}}(a',B,e') \times \boldsymbol{f}_{\mathbf{4}}(a') \times \boldsymbol{f}_{\mathbf{5}}(a')}_{\boldsymbol{f}_6(B,e')}$$

- Sum out a' from the product of $\boldsymbol{f}_3, \boldsymbol{f}_4, \boldsymbol{f}_5$.

$$\underset{2\times 2}{\boldsymbol{f}_6(B,e')} = \sum_{a' \in \{a,\neg a\}} \boldsymbol{f}_{\mathbf{3}}(a',B,e') \times \boldsymbol{f}_{\mathbf{4}}(a') \times \boldsymbol{f}_{\mathbf{5}}(a')$$

$$= \left(\underset{2\times 2}{\boldsymbol{f}_3(a,B,e')} \times P(j \mid a) \times P(m \mid a)\right) + \left(\underset{2\times 2}{\boldsymbol{f}_3(\neg a,B,e')} \times P(j \mid \neg a) \times P(m \mid \neg a)\right)$$

$$\mathbf{P}(B \mid j,m) = \alpha \boldsymbol{f}_{\mathbf{1}}(B) \times \underbrace{\sum_{e'} \boldsymbol{f}_2(e') \times \boldsymbol{f}_6(B,e')}_{\boldsymbol{f}_7(B,e')}$$

Evaluation Example

Pointwise product operation

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{e'} \boldsymbol{f_2}(e') \times \underbrace{\sum_{a'} \boldsymbol{f_3}(a', B, e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a')}_{\boldsymbol{f}_6(B, e')}$$

- Sum out a' from the product of $\boldsymbol{f}_3, \boldsymbol{f}_4, \boldsymbol{f}_5$.

$$\underset{2 \times 2}{\boldsymbol{f}_6(B, e')} = \sum_{a' \in \{a, \neg a\}} \boldsymbol{f_3}(a', B, e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a')$$

$$= \left(\underset{2 \times 2}{\boldsymbol{f}_3(a, B, e')} \times P(j \mid a) \times P(m \mid a)\right) + \left(\underset{2 \times 2}{\boldsymbol{f}_3(\neg a, B, e')} \times P(j \mid \neg a) \times P(m \mid \neg a)\right)$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \underbrace{\sum_{e'} \boldsymbol{f}_2(e') \times \boldsymbol{f}_6(B, e')}_{\boldsymbol{f}_7(B, e')}$$

- Sum out e' from the product of $\boldsymbol{f}_2$ and $\boldsymbol{f}_6$.

$$\underset{2 \times 1}{\boldsymbol{f}_7(B)} = \sum_{e' \in \{e, \neg e\}} \boldsymbol{f_2}(e') \times \boldsymbol{f_6}(B, e') = P(e) \times \underset{2 \times 1}{\boldsymbol{f}_6(B, e)} + P(\neg e) \times \underset{2 \times 1}{\boldsymbol{f}_6(B, \neg e)}$$

Evaluation Example

Pointwise product operation

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f}_1(B) \times \sum_{e'} \boldsymbol{f}_2(e') \times \underbrace{\sum_{a'} \boldsymbol{f}_3(a', B, e') \times \boldsymbol{f}_4(a') \times \boldsymbol{f}_5(a')}_{\boldsymbol{f}_6(B, e')}$$

- Sum out a' from the product of $\boldsymbol{f}_3, \boldsymbol{f}_4, \boldsymbol{f}_5$.

$$\underset{2\times 2}{\boldsymbol{f}_6(B, e')} = \sum_{a' \in \{a, \neg a\}} \boldsymbol{f}_3(a', B, e') \times \boldsymbol{f}_4(a') \times \boldsymbol{f}_5(a')$$

$$= \left(\underset{2\times 2}{\boldsymbol{f}_3(a, B, e')} \times P(j \mid a) \times P(m \mid a)\right) + \left(\underset{2\times 2}{\boldsymbol{f}_3(\neg a, B, e')} \times P(j \mid \neg a) \times P(m \mid \neg a)\right)$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f}_1(B) \times \sum_{e'} \underbrace{\boldsymbol{f}_2(e') \times \boldsymbol{f}_6(B, e')}_{\boldsymbol{f}_7(B, e')}$$

- Sum out e' from the product of $\boldsymbol{f}_2$ and $\boldsymbol{f}_6$.

$$\underset{2\times 1}{\boldsymbol{f}_7(B)} = \sum_{e' \in \{e, \neg e\}} \boldsymbol{f}_2(e') \times \boldsymbol{f}_6(B, e') = P(e) \times \underset{2\times 1}{\boldsymbol{f}_6(B, e)} + P(\neg e) \times \underset{2\times 1}{\boldsymbol{f}_6(B, \neg e)}$$

- Finally, carry out the following pointwise product:

$$\underset{2\times 1}{\boldsymbol{P}(B \mid j, m)} = \alpha \underset{2\times 1}{\boldsymbol{f}_1(B)} \times \underset{2\times 1}{\boldsymbol{f}_7(B)}$$

Pointwise Product of Two Factors

$$\boldsymbol{f}(X_1, \ldots, X_j, Y_1, \ldots, Y_k) \times \boldsymbol{g}(Y_1, \ldots, Y_k, Z_1, \ldots, Z_l) = \boldsymbol{h}(X_1, \ldots, X_j, Y_1, \ldots, Y_k, Z_1, \ldots, Z_l)$$

Entries under $\boldsymbol{f}(X, Y)$, $\boldsymbol{g}(Y, Z)$, and $\boldsymbol{h}(X, Y, Z)$ do not represent probabilities.

X	Y	$\mathbf{f}(X, Y)$	Y	Z	$\mathbf{g}(Y, Z)$	X	Y	Z	$\mathbf{h}(X, Y, Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

Pointwise Product of Two Factors

$$\boldsymbol{f}(X_1,\ldots,X_j,Y_1,\ldots,Y_k)\times \boldsymbol{g}(Y_1,\ldots,Y_k,Z_1,\ldots,Z_l)=\boldsymbol{h}(X_1,\ldots,X_j,Y_1,\ldots,Y_k,Z_1,\ldots,Z_l)$$

common variables

Entries under $\boldsymbol{f}(X,Y)$, $\boldsymbol{g}(Y,Z)$, and $\boldsymbol{h}(X,Y,Z)$ do not represent probabilities.

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3\times.2=.06$
t	f	.7	t	f	.8	t	t	f	$.3\times.8=.24$
f	t	.9	f	t	.6	t	f	t	$.7\times.6=.42$
f	f	.1	f	f	.4	t	f	f	$.7\times.4=.28$
						f	t	t	$.9\times.2=.18$
						f	t	f	$.9\times.8=.72$
						f	f	t	$.1\times.6=.06$
						f	f	f	$.1\times.4=.04$

Pointwise Product of Two Factors

$$\boldsymbol{f}(X_1, \ldots, X_j, Y_1, \ldots, Y_k) \times \boldsymbol{g}(Y_1, \ldots, Y_k, Z_1, \ldots, Z_l) = \boldsymbol{h}(X_1, \ldots, X_j, Y_1, \ldots, Y_k, Z_1, \ldots, Z_l)$$

common variables

Entries under $\boldsymbol{f}(X, Y)$, $\boldsymbol{g}(Y, Z)$, and $\boldsymbol{h}(X, Y, Z)$ do not represent probabilities.

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

Pointwise Product of Two Factors

$$\boldsymbol{f}(X_1, \ldots, X_j, Y_1, \ldots, Y_k) \times \boldsymbol{g}(Y_1, \ldots, Y_k, Z_1, \ldots, Z_l) = \boldsymbol{h}(X_1, \ldots, X_j, Y_1, \ldots, Y_k, Z_1, \ldots, Z_l)$$

common variables

Entries under $\boldsymbol{f}(X,Y)$, $\boldsymbol{g}(Y,Z)$, and $\boldsymbol{h}(X,Y,Z)$ do not represent probabilities.

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

Pointwise Product of Two Factors

$$\boldsymbol{f}(X_1, \dots, X_j, Y_1, \dots, Y_k) \times \boldsymbol{g}(Y_1, \dots, Y_k, Z_1, \dots, Z_l) = \boldsymbol{h}(X_1, \dots, X_j, Y_1, \dots, Y_k, Z_1, \dots, Z_l)$$

common variables

Entries under $\boldsymbol{f}(X, Y)$, $\boldsymbol{g}(Y, Z)$, and $\boldsymbol{h}(X, Y, Z)$ do not represent probabilities.

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

Summing Out a Variable

This operation over a product of factors is carried out by adding up the submatrices, each for one value of the same variable.

$$\boldsymbol{l}(Y,Z) = \sum_{x' \in \{x, \neg x\}} \boldsymbol{h}(x', Y, Z) = \boldsymbol{h}(x, Y, Z) + \boldsymbol{h}(\neg x, Y, Z)$$

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

// Entries are just for illustration;
// they are not probabilities.

Summing Out a Variable

This operation over a product of factors is carried out by adding up the submatrices, each for one value of the same variable.

$$\boldsymbol{l}(Y,Z) = \sum_{x' \in \{x, \neg x\}} \boldsymbol{h}(x',Y,Z) = \boldsymbol{h}(x,Y,Z) + \boldsymbol{h}(\neg x,Y,Z)$$

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

// Entries are just for illustration;
// they are not probabilities.

Summing Out a Variable

This operation over a product of factors is carried out by adding up the submatrices, each for one value of the same variable.

$$\boldsymbol{l}(Y,Z) = \sum_{x' \in \{x, \neg x\}} \boldsymbol{h}(x',Y,Z) = \boldsymbol{h}(x,Y,Z) + \boldsymbol{h}(\neg x,Y,Z)$$

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	$.3$	t	t	$.2$	t	t	t	$.3 \times .2 = .06$
t	f	$.7$	t	f	$.8$	t	t	f	$.3 \times .8 = .24$
f	t	$.9$	f	t	$.6$	t	f	t	$.7 \times .6 = .42$
f	f	$.1$	f	f	$.4$	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

// Entries are just for illustration;
// they are not probabilities.

Summing Out a Variable

This operation over a product of factors is carried out by adding up the submatrices, each for one value of the same variable.

$$\boldsymbol{l}(Y,Z) = \sum_{x' \in \{x, \neg x\}} \boldsymbol{h}(x', Y, Z) = \boldsymbol{h}(x, Y, Z) + \boldsymbol{h}(\neg x, Y, Z)$$

$$= \begin{pmatrix} .06 & .24 \\ .42 & .28 \end{pmatrix}$$

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

// Entries are just for illustration;
// they are not probabilities.

Summing Out a Variable

This operation over a product of factors is carried out by adding up the submatrices, each for one value of the same variable.

$$\boldsymbol{l}(Y,Z) = \sum_{x' \in \{x, \neg x\}} \boldsymbol{h}(x', Y, Z) = \boldsymbol{h}(x, Y, Z) + \boldsymbol{h}(\neg x, Y, Z)$$

$$= \begin{pmatrix} .06 & .24 \\ .42 & .28 \end{pmatrix}$$

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

// Entries are just for illustration;
// they are not probabilities.

Summing Out a Variable

This operation over a product of factors is carried out by adding up the submatrices, each for one value of the same variable.

$$\boldsymbol{l}(Y,Z) = \sum_{x' \in \{x, \neg x\}} \boldsymbol{h}(x', Y, Z) = \boldsymbol{h}(x, Y, Z) + \boldsymbol{h}(\neg x, Y, Z)$$

$$= \begin{pmatrix} .06 & .24 \\ .42 & .28 \end{pmatrix}$$

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

// Entries are just for illustration;
// they are not probabilities.

Summing Out a Variable

This operation over a product of factors is carried out by adding up the submatrices, each for one value of the same variable.

$$\boldsymbol{l}(Y,Z) = \sum_{x' \in \{x, \neg x\}} \boldsymbol{h}(x', Y, Z) = \boldsymbol{h}(x, Y, Z) + \boldsymbol{h}(\neg x, Y, Z)$$

$$= \begin{pmatrix} .06 & .24 \\ .42 & .28 \end{pmatrix} + \begin{pmatrix} .18 & .72 \\ .06 & .04 \end{pmatrix}$$

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

// Entries are just for illustration;
// they are not probabilities.

Summing Out a Variable

This operation over a product of factors is carried out by adding up the submatrices, each for one value of the same variable.

$$\boldsymbol{l}(Y,Z) = \sum_{x' \in \{x, \neg x\}} \boldsymbol{h}(x', Y, Z) = \boldsymbol{h}(x, Y, Z) + \boldsymbol{h}(\neg x, Y, Z)$$

$$= \begin{pmatrix} .06 & .24 \\ .42 & .28 \end{pmatrix} + \begin{pmatrix} .18 & .72 \\ .06 & .04 \end{pmatrix} = \begin{pmatrix} .24 & .96 \\ .48 & .32 \end{pmatrix}$$

X	Y	$\mathbf{f}(X,Y)$	Y	Z	$\mathbf{g}(Y,Z)$	X	Y	Z	$\mathbf{h}(X,Y,Z)$
t	t	.3	t	t	.2	t	t	t	$.3 \times .2 = .06$
t	f	.7	t	f	.8	t	t	f	$.3 \times .8 = .24$
f	t	.9	f	t	.6	t	f	t	$.7 \times .6 = .42$
f	f	.1	f	f	.4	t	f	f	$.7 \times .4 = .28$
						f	t	t	$.9 \times .2 = .18$
						f	t	f	$.9 \times .8 = .72$
						f	f	t	$.1 \times .6 = .06$
						f	f	f	$.1 \times .4 = .04$

// Entries are just for illustration;
// they are not probabilities.

Variable Elimination Algorithm

Move outside the summation any factor independent of the variable to be summed out.

$$\sum_{x'} \boldsymbol{f}(x', Y) \times \boldsymbol{g}(Y, Z) = \boldsymbol{g}(Y, Z) \times \sum_{x'} \boldsymbol{f}(x', Y)$$

IV. Ordering Variables to Eliminate

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

IV. Ordering Variables to Eliminate

- Every choice of ordering yields a valid algorithm. ⇐ $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$

IV. Ordering Variables to Eliminate

- Every choice of ordering yields a valid algorithm. ⇐ $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{e'} \boldsymbol{f_2}(e') \times \sum_{a'} \boldsymbol{f_3}(a', B, e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a') \quad \text{(Order: } A, E\text{)}$$

IV. Ordering Variables to Eliminate

- Every choice of ordering yields a valid algorithm. ⇐ $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{e'} \boldsymbol{f_2}(e') \times \sum_{a'} \boldsymbol{f_3}(a', B, e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a') \quad \text{(Order: } A, E\text{)}$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{a'} \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a') \times \sum_{e'} \boldsymbol{f_2}(e') \times \boldsymbol{f_3}(a', B, e') \quad \text{(Order: } E, A\text{)}$$

IV. Ordering Variables to Eliminate

- Every choice of ordering yields a valid algorithm. ⇐ $P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{e'} \boldsymbol{f_2}(e') \times \sum_{a'} \boldsymbol{f_3}(a', B, e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a') \quad \text{(Order: } A, E\text{)}$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{a'} \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a') \times \sum_{e'} \boldsymbol{f_2}(e') \times \boldsymbol{f_3}(a', B, e') \quad \text{(Order: } E, A\text{)}$$

- Different orderings generates different intermediate factors.

IV. Ordering Variables to Eliminate

- Every choice of ordering yields a valid algorithm. ⇐ $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{e'} \boldsymbol{f_2}(e') \times \sum_{a'} \boldsymbol{f_3}(a', B, e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a') \quad \text{(Order: } A, E\text{)}$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{a'} \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a') \times \sum_{e'} \boldsymbol{f_2}(e') \times \boldsymbol{f_3}(a', B, e') \quad \text{(Order: } E, A\text{)}$$

- Different orderings generates different intermediate factors.
- Time and space are dominated by the size of the largest factor constructed, which is subject to two factors:

IV. Ordering Variables to Eliminate

- Every choice of ordering yields a valid algorithm. ⇐ $P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{e'} \boldsymbol{f_2}(e') \times \sum_{a'} \boldsymbol{f_3}(a', B, e') \times \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a') \quad \text{(Order: } A, E\text{)}$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f_1}(B) \times \sum_{a'} \boldsymbol{f_4}(a') \times \boldsymbol{f_5}(a') \times \sum_{e'} \boldsymbol{f_2}(e') \times \boldsymbol{f_3}(a', B, e') \quad \text{(Order: } E, A\text{)}$$

- Different orderings generates different intermediate factors.
- Time and space are dominated by the size of the largest factor constructed, which is subject to two factors:
  - ordering of variables
  - structure of the network

IV. Ordering Variables to Eliminate

- Every choice of ordering yields a valid algorithm. ⇐ $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f}_1(B) \times \sum_{e'} \boldsymbol{f}_2(e') \times \sum_{a'} \boldsymbol{f}_3(a', B, e') \times \boldsymbol{f}_4(a') \times \boldsymbol{f}_5(a') \quad \text{(Order: } A, E\text{)}$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f}_1(B) \times \sum_{a'} \boldsymbol{f}_4(a') \times \boldsymbol{f}_5(a') \times \sum_{e'} \boldsymbol{f}_2(e') \times \boldsymbol{f}_3(a', B, e') \quad \text{(Order: } E, A\text{)}$$

- Different orderings generates different intermediate factors.
- Time and space are dominated by the size of the largest factor constructed, which is subject to two factors:
  - ordering of variables
  - structure of the network
- It is intractable to determine the optimal order.

IV. Ordering Variables to Eliminate

- Every choice of ordering yields a valid algorithm. ⇦ $P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f}_1(B) \times \sum_{e'} \boldsymbol{f}_2(e') \times \sum_{a'} \boldsymbol{f}_3(a', B, e') \times \boldsymbol{f}_4(a') \times \boldsymbol{f}_5(a') \quad \text{(Order: } A, E\text{)}$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{f}_1(B) \times \sum_{a'} \boldsymbol{f}_4(a') \times \boldsymbol{f}_5(a') \times \sum_{e'} \boldsymbol{f}_2(e') \times \boldsymbol{f}_3(a', B, e') \quad \text{(Order: } E, A\text{)}$$

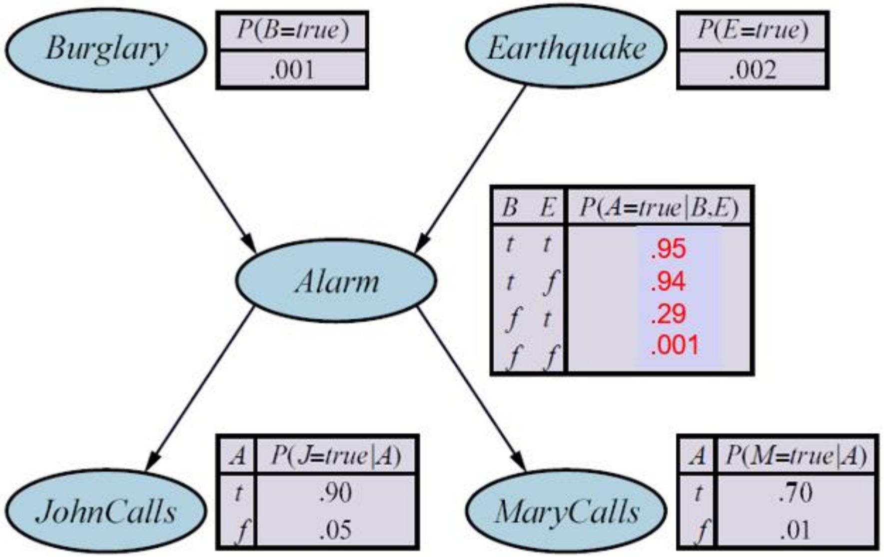
- Different orderings generates different intermediate factors.
- Time and space are dominated by the size of the largest factor constructed, which is subject to two factors:
  - ordering of variables
  - structure of the network
- It is intractable to determine the optimal order.
- Use a greedy heuristic: eliminate whichever variable minimizes the size of the next factor to be constructed.

Irrelevant Variable

$\boldsymbol{P}(JohnCalls \mid Burglary = true)?$

Irrelevant Variable

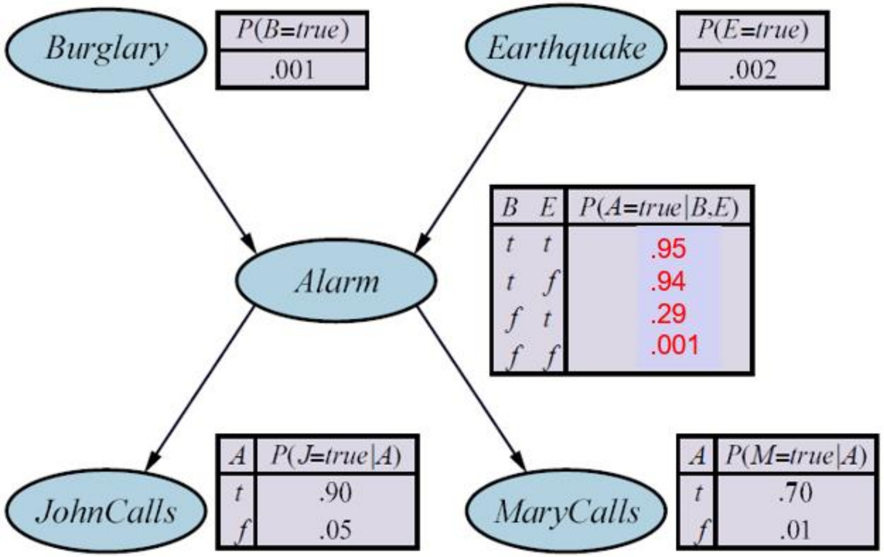
$\mathbf{P}(JohnCalls \mid Burglary = true)$?



Irrelevant Variable

$\boldsymbol{P}(JohnCalls \mid Burglary = true)$?

$$\boldsymbol{P}(J \mid b) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e')\, \boldsymbol{P}(J \mid a') \sum_{m'} P(m' \mid a')$$

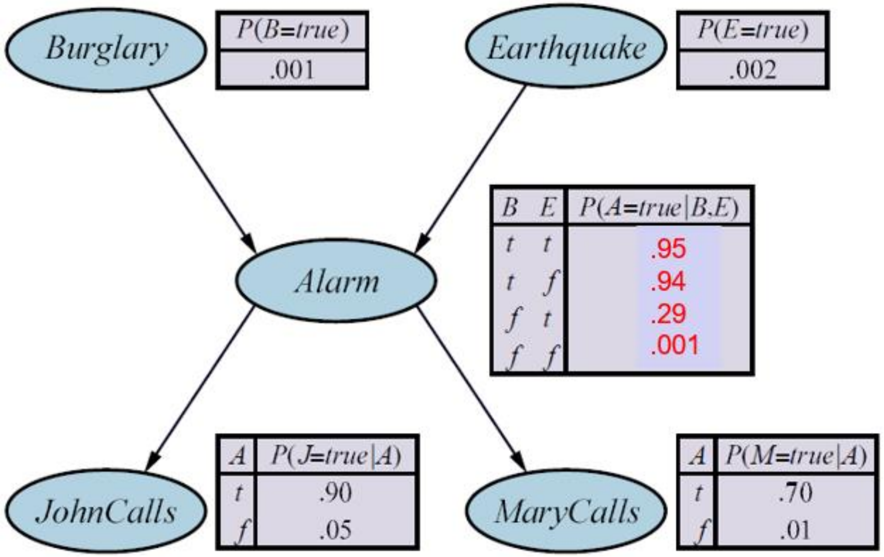


Irrelevant Variable

$\boldsymbol{P}(JohnCalls \mid Burglary = true)$?

$$\boldsymbol{P}(J \mid b) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e')\, \boldsymbol{P}(J \mid a') \sum_{m'} P(m' \mid a')$$

$$\sum_{m' \in \{m, \neg m\}} P(m' \mid a') = 1$$



Irrelevant Variable

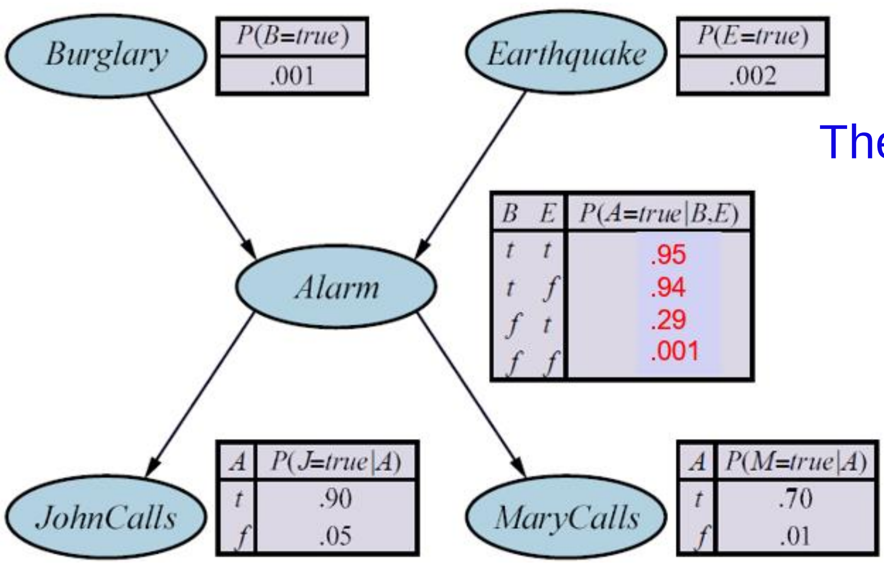
$\boldsymbol{P}(JohnCalls \mid Burglary = true)$?

$$\boldsymbol{P}(J \mid b) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e')\, \boldsymbol{P}(J \mid a') \sum_{m'} P(m' \mid a')$$

$$\sum_{m' \in \{m, \neg m\}} P(m' \mid a') = 1$$

$$\Downarrow$$

The eliminated variable M is irrelevant to the query.



Irrelevant Variable

$\boldsymbol{P}(JohnCalls \mid Burglary = true)$?

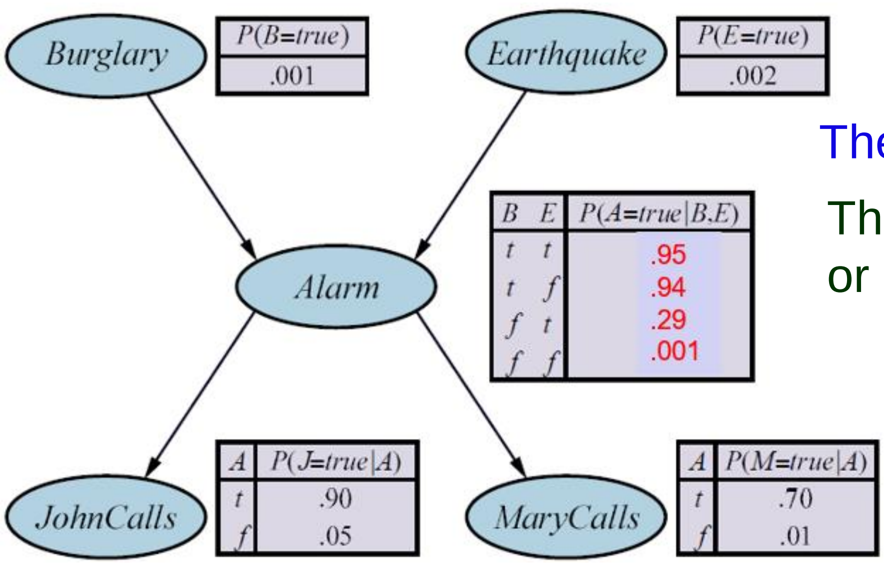
$$\boldsymbol{P}(J \mid b) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e')\, \boldsymbol{P}(J \mid a') \sum_{m'} P(m' \mid a')$$

$$\sum_{m' \in \{m, \neg m\}} P(m' \mid a') = 1$$

$\Downarrow$

The eliminated variable M is irrelevant to the query.

This cannot be inferred using the Markov blanket or d-separation.



Irrelevant Variable

$\boldsymbol{P}(JohnCalls \mid Burglary = true)$?

$$\boldsymbol{P}(J \mid b) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e') \, \boldsymbol{P}(J \mid a') \sum_{m'} P(m' \mid a')$$

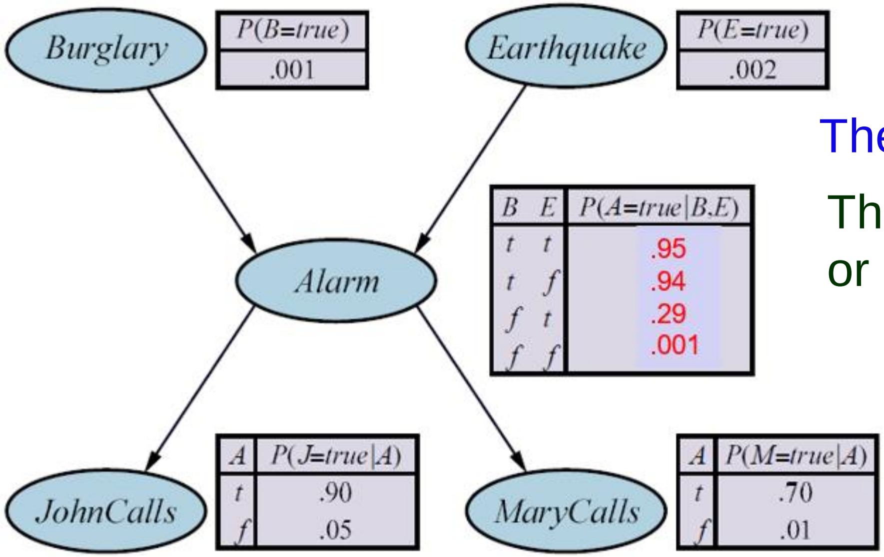
$$\sum_{m' \in \{m, \neg m\}} P(m' \mid a') = 1$$

⇓

The eliminated variable M is irrelevant to the query.

This cannot be inferred using the Markov blanket or d-separation.

- *M* is not in the Markov blanket $\{A\}$ of *J*. But neither is *B*. (Given A, M & B would be irrelevant.)



Irrelevant Variable

$\boldsymbol{P}(JohnCalls \mid Burglary = true)$?

$$\boldsymbol{P}(J \mid b) = \alpha P(b) \sum_{e'} P(e') \sum_{a'} P(a' \mid b, e')\, \boldsymbol{P}(J \mid a') \sum_{m'} P(m' \mid a')$$

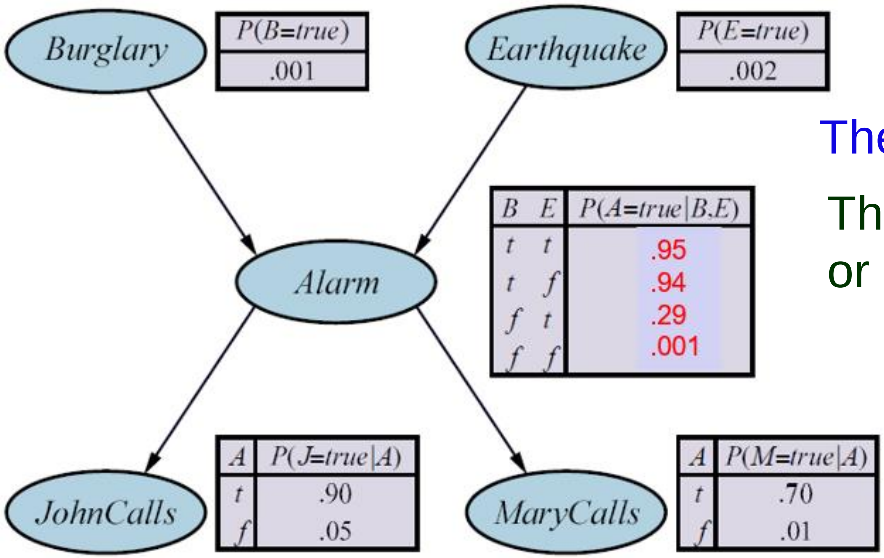
$$\sum_{m' \in \{m, \neg m\}} P(m' \mid a') = 1$$

$\Downarrow$

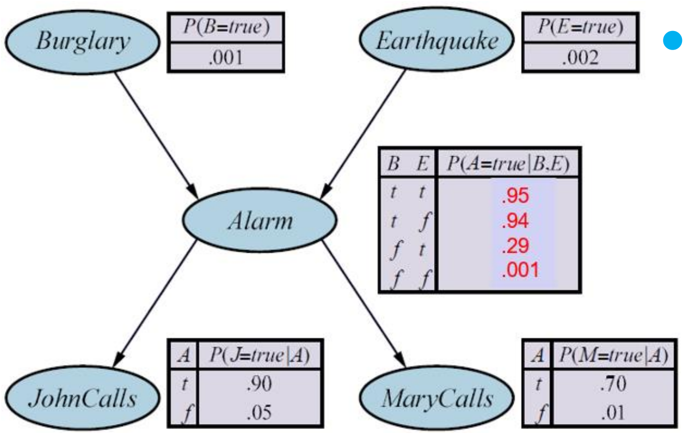
The eliminated variable M is irrelevant to the query.

This cannot be inferred using the Markov blanket or d-separation.

- *M* is not in the Markov blanket $\{A\}$ of *J*. But neither is *B*. (Given A, M & B would be irrelevant.)
- *J* and *M* are not d-separated by *B*. (The moral graph would contain the network plus an edge connecting *B* and *E*.)



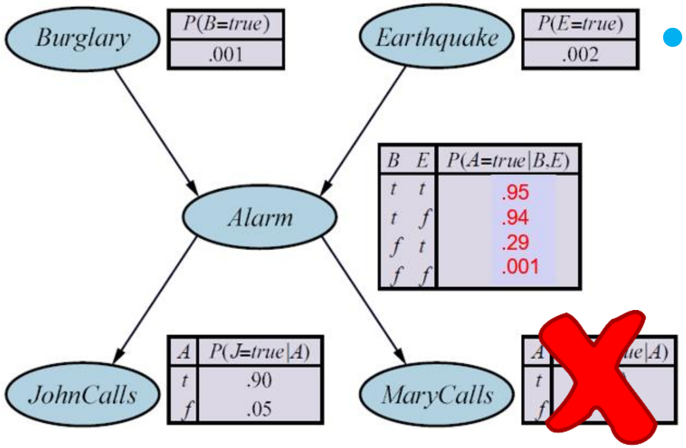
Inference with Elimination



- We can remove a leaf node (e.g., M) that is neither a query variable nor an evidence variable.

$$\boldsymbol{P}(JohnCalls \mid Burglary = true)?$$

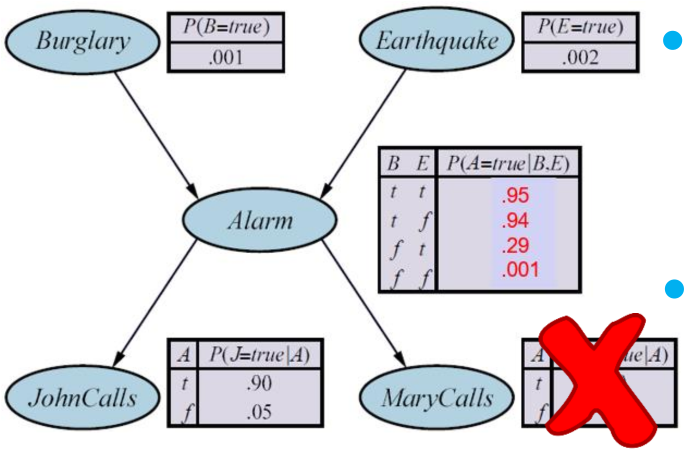
Inference with Elimination



- We can remove a leaf node (e.g., M) that is neither a query variable nor an evidence variable.

$$\boldsymbol{P}(JohnCalls \mid Burglary = true)?$$

Inference with Elimination

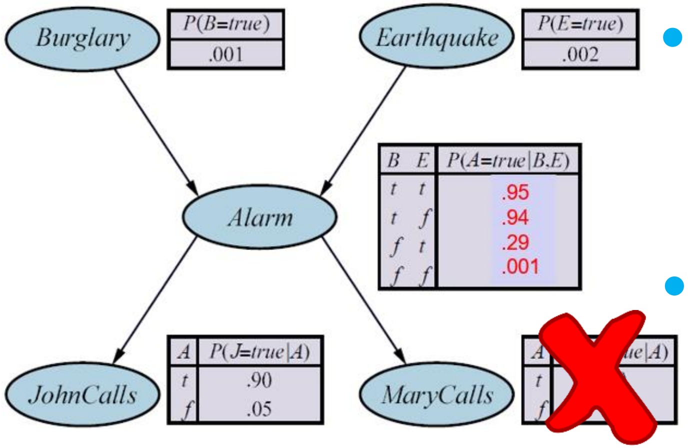


- We can remove a leaf node (e.g., M) that is neither a query variable nor an evidence variable.

  $$\boldsymbol{P}(JohnCalls \mid Burglary = true)?$$

- After the removal, there may be more leaf nodes that are irrelevant. Remove them as well, and so on.

Inference with Elimination



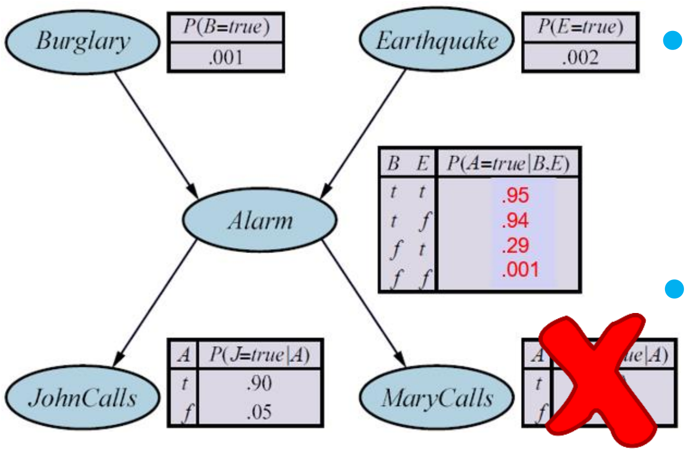
- We can remove a leaf node (e.g., M) that is neither a query variable nor an evidence variable.

  $$\boldsymbol{P}(JohnCalls \mid Burglary = true)?$$

- After the removal, there may be more leaf nodes that are irrelevant. Remove them as well, and so on.

Every variable that is not an ancestor of a query variable or evidence variable is irrelevant to the query.

Inference with Elimination



- We can remove a leaf node (e.g., M) that is neither a query variable nor an evidence variable.

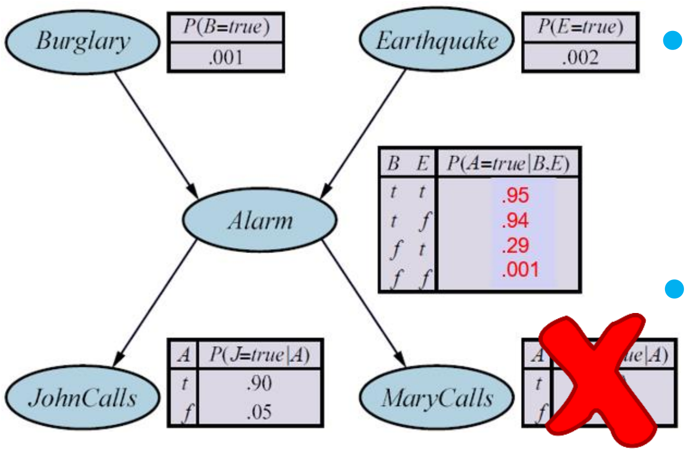
  $\mathbf{P}(JohnCalls \mid Burglary = true)$?

- After the removal, there may be more leaf nodes that are irrelevant. Remove them as well, and so on.

Every variable that is not an ancestor of a query variable or evidence variable is irrelevant to the query.

♦ Using *reverse topological order* for variables, exact inference with elimination can be 1,000 times faster than the enumeration algorithm.

Inference with Elimination



- We can remove a leaf node (e.g., M) that is neither a query variable nor an evidence variable.

$$\mathbf{P}(JohnCalls \mid Burglary = true)?$$

- After the removal, there may be more leaf nodes that are irrelevant. Remove them as well, and so on.

Every variable that is not an ancestor of a query variable or evidence variable is irrelevant to the query.

♦ Using *reverse topological order* for variables, exact inference with elimination can be 1,000 times faster than the enumeration algorithm.

♦ If we want to compute posterior probabilities for all the variables rather than answer individual queries, we can use clustering algorithms (i.e., join tree algorithms).