DATA130008 《人工智能导论》

复旦大学大数据学院
School of Data Science, Fudan University

Lab-4 Bayes'Net Inference

Enumeration VS Elimination

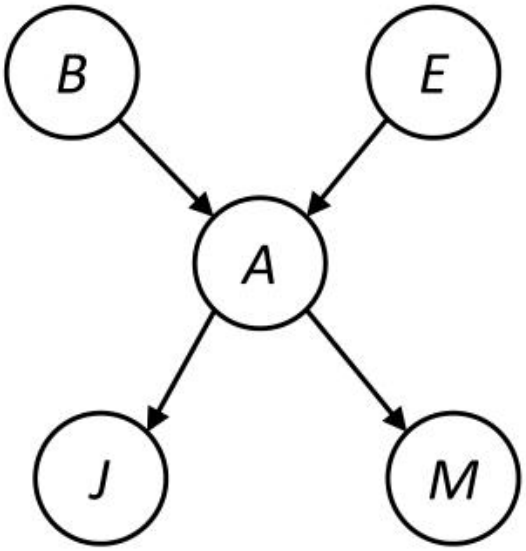
B: Burglary

E: Earthquake

A: Alarm

M: Mary calls

J: John calls



$$P(B|j,m) = \alpha P(B,j,m)$$

$$= \alpha \sum_e \sum_a P(B,j,m,e,a)$$

Enumeration

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

Elimination

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

Factors in Inference

- Factors in Alarm Network
  - $f_1(B)$: P(B)
  - $f_2(E)$: P(E)
  - $f_3(A,B,E)$: P(A|B, E)
  - $f_4(A)$: P(+j|A)
  - $f_5(A)$: P(+m|A)

- Join Factors:

$P(R)$

+r	0.1
-r	0.9

$\times$

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$\Rightarrow$

$P(R,T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

R,T

- Eliminate:

$P(R,T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

sum R $\Rightarrow$

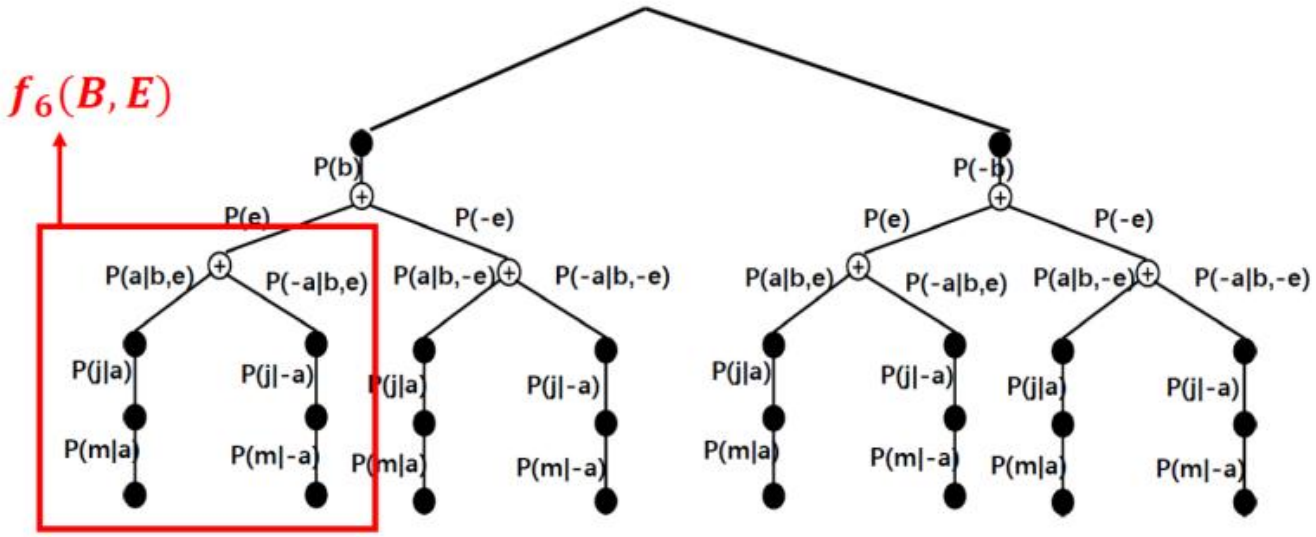
$P(T)$

+t	0.17
-t	0.83

Factors in Inference

- Enumeration:
  $\alpha \sum_e \sum_a f_1(B) \times f_2(E) \times f_3(A,B,E) \times f_4(A) \times f_5(A)$

- Elimination:
  $\alpha \; f_1(B) \times \sum_e f_2(E) \times \sum_a f_3(A,B,E) \times f_4(A) \times f_5(A)$



Summary

- Step 1:
  Select the entries consistent with the evidence

- Step 2:
  - Enumeration:
    Join and sum out hidden vars to get joint of Query and evidence

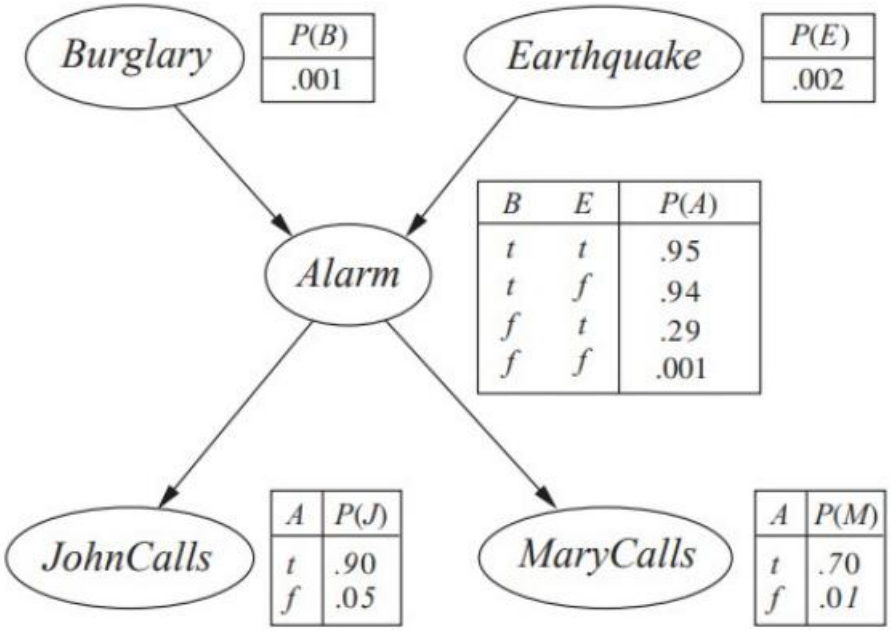
  - Elimination:
    Loop:
    - Pick a hidden variable H
    - Join all factors mentioning H
    - Eliminate (sum out) H

    Until all hidden variables are eliminated
    -> Join all remaining factors

- Step 3:
  Normalize

Problem Description

- Still, the burglary problem
- Input: Query and CPTs
- Output: **Query Probability**, and **Evidence Probability**
- Time limit: 1000ms
- Memory limit: 263MB
- Implement both methods: Enumeration and Elimination



Functions to be implemented

- Enumeration algorithm:
  - Def enumeration_ask(X, e, bn):
  - Def enumeration_all(X, e, bn):
- Elimination algorithm:
  - Def elimination_ask(X, e, bn):
- You need to get evidence probability in elimination_ask(X, e, bn) as well !
- Functions that will be directly used in your implementation:
  - def extend(s, var, val):
  - def make_factor(var, e, bn):
  - def sum_out(var, factors):
  - def pointwise_product(factors):

Classes:

```
class BayesNet:                                   used in building the BayesNet
    def __init__(self, node_specs=[]):
    def add(self, node_spec):
    def variable_node(self, var):
    def variable_values(self, vars):
```

```
class BayesNode:                                  used in building the BayesNode
    """A conditional probability distribution for a boolean variable, P(X | parents).
Part of a BayesNet."""
    def __init__(self, x, parents, cpt):
    def p(self, value, event):
```

Classes:

```
class ProbDist:                    used in the computation for probability distribution
    def __init__(self, varname='?', freqs=None):
    def normalize(self):
```

```
class Factor:                                     used in elimination algorithm
    def __init__(self, variables, cpt):
    def pointwise_product(self, other):
    def sum_out(self, var):
    def p(self, e):
    def normalize(self):
```

Pseudo Code

- Enumeration

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 $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$ /* **Y** = *hidden variables* */

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

---

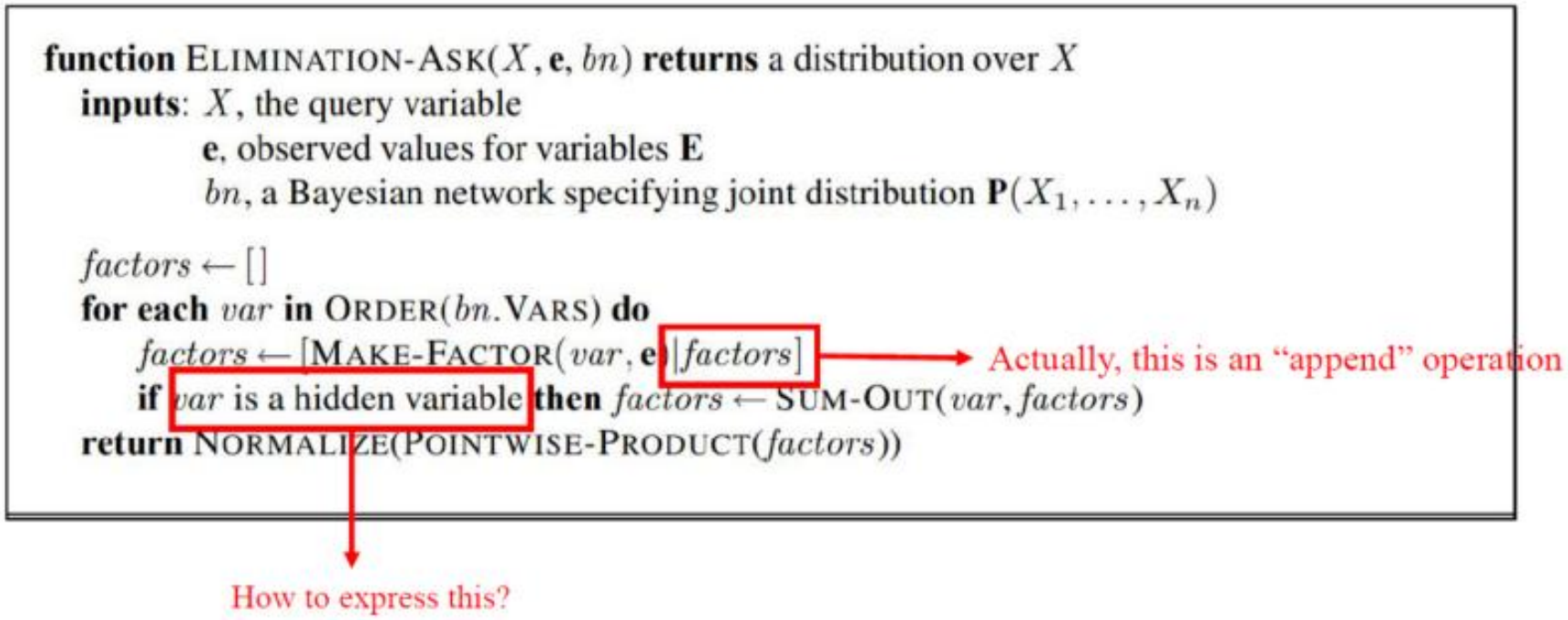
function ENUMERATE-ALL($vars$, **e**) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $Y \leftarrow$ FIRST($vars$)
  if Y has value y in **e**
    then return $P(y \mid parents(Y)) \times$ ENUMERATE-ALL(REST($vars$), **e**)
    else return $\sum_y P(y \mid parents(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_y$)
      where $\mathbf{e}_y$ is **e** extended with $Y = y$

You can use Ynode.p()

Figure 14.9 The enumeration algorithm for answering queries on Bayesian networks.

Pseudo Code

- Elimination



- How to calculate evidence probability ?