Exact Inference

CSci 5512: Artificial Intelligence II

Instructor: Paul Schrater

Overview: Inference Tasks

- Simple Queries: Compute posterior marginals P(b|j,¬m)

Overview: Inference Tasks

- Simple Queries: Compute posterior marginals $P(b|j,\neg m)$
- Conjunctive Queries: Compute joint marginals $P(b,\neg e|j,\neg m)$

Overview: Inference Tasks

- Simple Queries: Compute posterior marginals P(b|j,¬m)
- Conjunctive Queries: Compute joint marginals P(b,¬e|j,¬m)
- Optimal Decisions: Compute P(outcome|action,evidence)

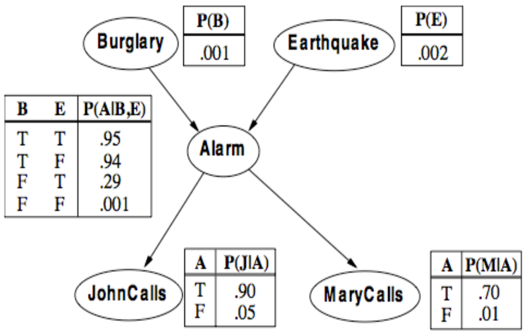
Overview: Inference Tasks

- Simple Queries: Compute posterior marginals $P(b|j,\neg m)$
- Conjunctive Queries: Compute joint marginals $P(b,\neg e|j,\neg m)$
- Optimal Decisions: Compute P(outcome|action,evidence)
- Value of Information: Which evidence to seek next?

Overview: Inference Tasks

- Simple Queries: Compute posterior marginals $P(b|j,\neg m)$
- Conjunctive Queries: Compute joint marginals $P(b,\neg e|j,\neg m)$
- Optimal Decisions: Compute P(outcome|action,evidence)
- Value of Information: Which evidence to seek next?
- Explanation: Why do I need a new starter motor?

Inference



How can we compute $P(b|j,m)$?

Inference by Enumeration

- Simple query can be answered using Bayes rule

Inference by Enumeration

- Simple query can be answered using Bayes rule
  - From Bayes Rule

$$P(b|j,m) = \frac{P(b,j,m)}{P(j,m)}$$

  - Each marginal can be obtained from the joint distribution

$$\begin{aligned} P(b,j,m) &= \sum_E \sum_A P(b,E,A,j,m) \\ P(j,m) &= \sum_B \sum_E \sum_A P(B,E,A,j,m) \end{aligned}$$

  - Each term can be written as product of conditionals

$$P(b,E,A,j,m) = P(b)P(E)P(A|b,E)P(j|A)P(m|A)$$

- The complexity of the simple approach is $O(n2^n)$

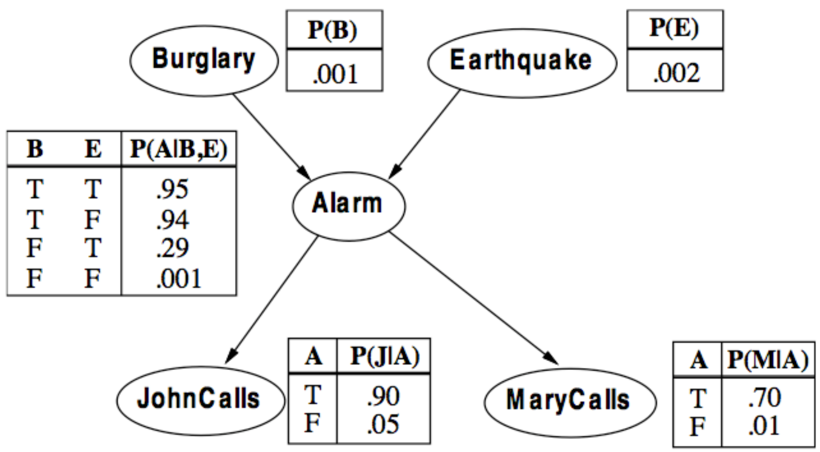
Inference by Enumeration (Contd)

- Complexity can be improved by a simple observation

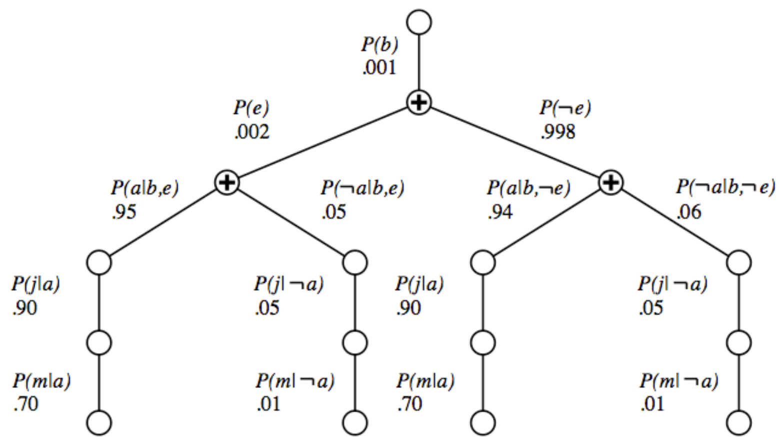
$$
\begin{aligned}
P(b|j,m) &= \frac{1}{P(j,m)} \sum_E \sum_A P(b,E,A,j,m) \\
&= \frac{1}{P(j,m)} \sum_E \sum_A P(b)P(E)P(A|E,b)P(j|A)P(m|A) \\
&= \frac{1}{P(j,m)} P(b) \left(\sum_E P(E) \left(\sum_A P(A|E,b)P(j|A)P(m|A) \right) \right)
\end{aligned}
$$

- Complexity is $O(2^n)$
  - Some computations are repeated

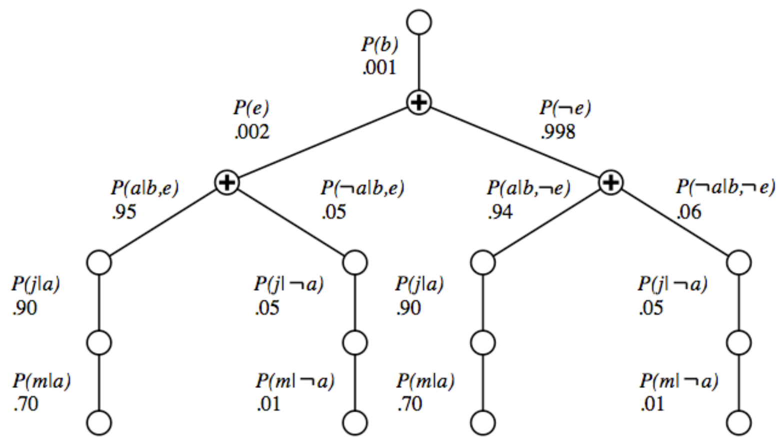
Burglary Network



Enumeration Tree



Enumeration Tree



- Enumeration Tree for P(b|j,m)
  - P(j|a)P(m|a) and P(j|¬a)P(m|¬a) are computed twice

Variable Elimination

- Queries can be written as sum-of-products
- Main idea
  - Sum over products to eliminate variables
  - Storage required to prevent repeat computations
- Burglary network

$$P(b|j,m)$$

$$= \frac{1}{P(j,m)} \underbrace{P(b)}_{B} \left(\sum_{E} \underbrace{P(E)}_{E} \left(\sum_{A} \underbrace{P(A|b,E)}_{A} \underbrace{P(j|A)}_{J} \underbrace{P(m|A)}_{M} \right) \right)))$$

- Have a *factor* for every variable

Factors for Variable Elimination

- Factor for M: $\mathbf{f}_m(A) = [P(m|a) \quad P(m|\neg a)]$
- Factor for J: $\mathbf{f}_j(A) = [P(j|a) \quad P(j|\neg a)]$
- Similarly, factors $\mathbf{f}_A(B, E, A), \mathbf{f}_E(E), \mathbf{f}_B(B)$
- Summing out to eliminate variables

$$
\begin{aligned}
\mathbf{f}_{\bar{A}jm}(B, E) &= \sum_A \mathbf{f}_A(B, E, A)\mathbf{f}_j(A)\mathbf{f}_m(A) \\
\mathbf{f}_{\bar{E}\bar{A}jm}(B) &= \sum_E \mathbf{f}_E(E)\mathbf{f}_{\bar{A}jm}(B, E) \\
P(B|j, m) &= \frac{1}{P(j, m)}\mathbf{f}_B(B)\mathbf{f}_{\bar{E}\bar{A}jm}(B)
\end{aligned}
$$

Variable Elimination: Basic Operations

- Summing out a variable from a product of factors
  - Pointwise products of (a pair of) factors
  - Sum out a variable from a product of factors
- Pointwise product of factors

$g_1(a,b) \times g_2(b,c) = g(a,b,c)$

In general

$$g_1(x_1,\ldots,x_j,y_1,\ldots,y_k) \times g_2(y_1,\ldots,y_k,z_1,\ldots,z_l) = g(x_1,\ldots,x_j,y_1,\ldots,y_k,z_1,\ldots,z_l)$$

Assuming $f_1,\ldots,f_i$ do not depend on X

$$\begin{aligned}\sum_x f_1 \times \cdots \times f_k &= f_1 \times \cdots \times f_i \times \left(\sum_X f_{i+1} \times \cdots \times f_k\right) \\ &= f_1 \times \cdots \times f_i \times f_{\bar{X}}\end{aligned}$$

Example: Pointwise Product of Factors

A	B	$f_1(A,B)$	B	C	$f_2(B,C)$	A	B	C	$f_3(A,B,C)$
T	T	0.3	T	T	0.2	T	T	T	0.3×0.2
T	F	0.7	T	F	0.8	T	T	F	0.3×0.8
F	T	0.9	F	T	0.6	T	F	T	0.7×0.6
F	F	0.1	F	F	0.4	T	F	F	0.7×0.4
							T	T	0.9×0.2
							T	F	0.9×0.8
							F	T	0.1×0.6
							F	F	0.1×0.4

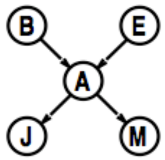
Algorithm

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 $\mathbf{P}(X_1, \ldots, X_n)$

  $factors \leftarrow [\,]$; $vars \leftarrow$ REVERSE(VARS[bn])
  for each var **in** $vars$ **do**
    $factors \leftarrow$ [MAKE-FACTOR(var, $\mathbf{e}$)|$factors$]
    if var is a hidden variable **then** $factors \leftarrow$ SUM-OUT(var, $factors$)
  return NORMALIZE(POINTWISE-PRODUCT($factors$))

Irrelevant variables

Consider the query $P(JohnCalls|Burglary = true)$



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

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

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

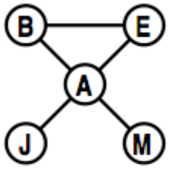
Here, $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and
$Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$
so $MaryCalls$ is irrelevant

Irrelevant variables

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

Defn: $\mathbf{A}$ is m-separated from $\mathbf{B}$ by $\mathbf{C}$ iff separated by $\mathbf{C}$ in the moral graph

Thm 2: Y is irrelevant if m-separated from X by $\mathbf{E}$



For $P(JohnCalls|Alarm=true)$, 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

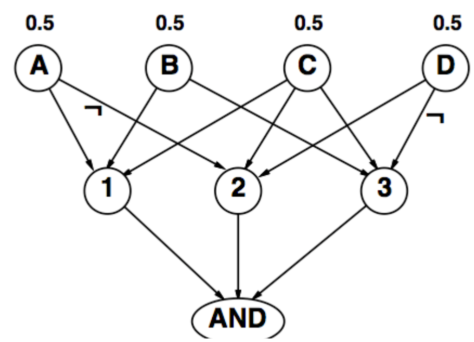
– time and space cost of variable elimination are $O(d^k n)$

Multiply connected networks:

– can reduce 3SAT to exact inference $\Rightarrow$ NP-hard

– equivalent to **counting** 3SAT models $\Rightarrow$ #P-complete

1. A v B v C
2. C v D v ¬A
3. B v C v ¬D



Inference Problems: Big Picture

Broadly the following types of problems

Compute ***likelihood*** of observations
Compute ***marginals*** $P(x_A)$ on subset A of nodes
Compute ***conditionals*** $P(x_A|x_B)$ on disjoint subsets A,B
Compute ***mode*** $\text{argmax}_x P(x)$

- First 3 problems are similar
  - Involves marginalization over sum-of-products
- Last problem is fundamentally different
  - Entails maximization rather than marginalization
- There is an important connection between the problems