Inference by Enumeration

Frederik Mallmann-Trenn
6CCS3AIN

Bayesian Networks

- Okay, so what can we do with Bayesian Networks?

Bayesian Networks

- Okay, so what can we do with Bayesian Networks?
- They are useful for inference (a conclusion reached on the basis of evidence and reasoning)

Inference tasks

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

  $\mathbf{P}(Listening = true|Status = excited)$
- Conjunctive queries

  $\mathbf{P}(X_i, X_j|\mathbf{E}=\mathbf{e}) = \mathbf{P}(X_i|\mathbf{E}=e)\mathbf{P}(X_j|X_i, \mathbf{E}=e)$
- Optimal decisions: decision networks include utility information; probabilistic inference required for $P(outcome|action, evidence)$
- 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 tasks

- We will focus on simple queries:
- Compute posterior marginal

  $\mathbf{P}(X_i|\mathbf{E} = e)$

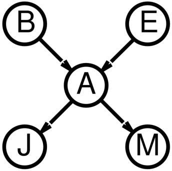
  $\mathbf{P}(Listening = true|Status = excited)$
- We will look a several ways of doing this.
  1. Enumeration
  2. Rejection sampling (using prior sampling)
  3. Likelihood weighting
  4. Gibbs sampling

Inference by enumeration

- Simplest approach to evaluating the network is to do just as we did for the dentist example
- Difference is that we use the structure of the network to tell us which sets of joint probabilities to use.
  - Thanks Professor Markov
- Gives us a slightly intelligent way to sum out variables from the joint without actually constructing its explicit representation

Inference by enumeration

- Simple query on the burglary network.



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

- Rewrite full joint entries taking network into account:

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

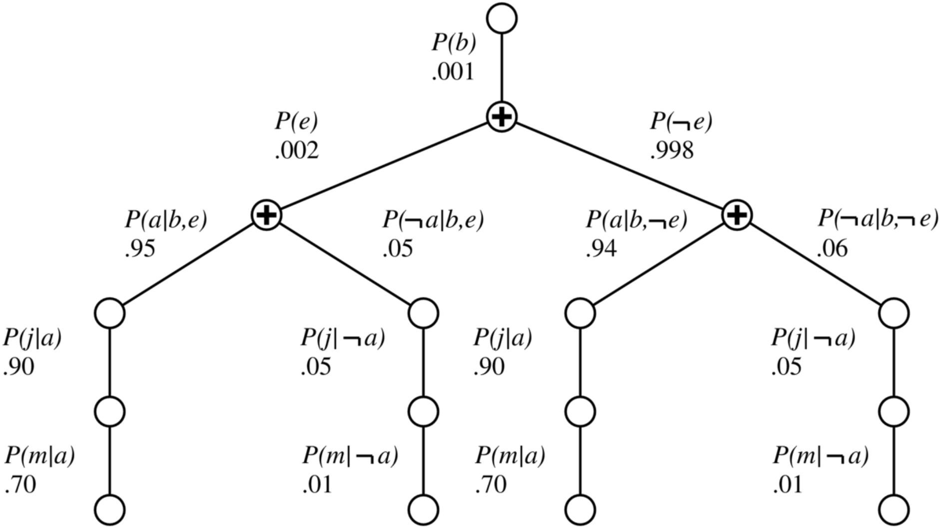
Inference by enumeration

- We evaluate this expression

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

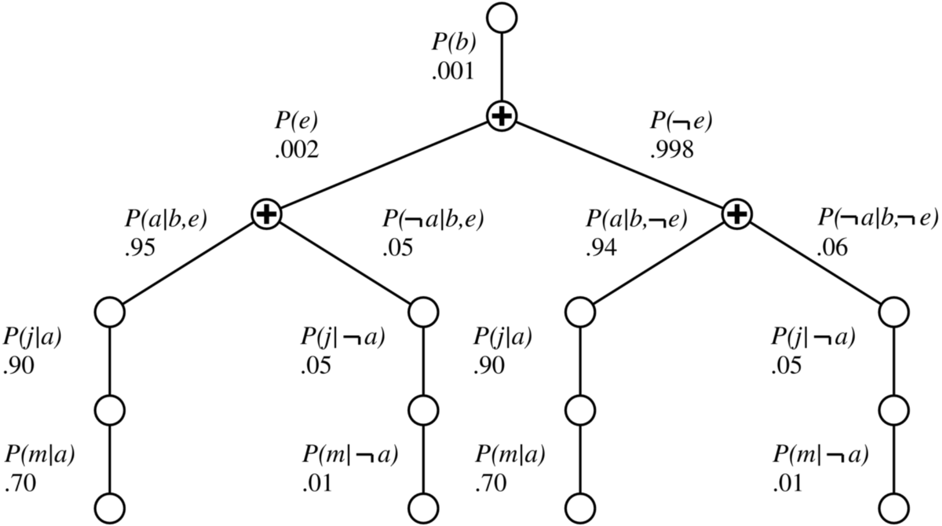
  by going through the variables in order, multiplying CPT entries along the way.
- At each point, we need to loop through the possible values of the variable.
- Involves a lot of repeated calculations.

Evaluation tree



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

Evaluation tree



Inefficient: computes $P(j|a)P(m|a)$ for each value of e

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 $\mathbb{E}$

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

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

  for each value x_i of X **do**

    extend $\mathbf{e}$ with value x_i for X

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

  return NORMALIZE($\mathbf{Q}(X)$)

Enumeration algorithm

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

Other exact approaches

- We can improve on enumeration.
- Variable elimination evaluates the enumeration tree bottom up, remembering intermediate values.
  - Simple and efficient for single queries
- Clustering algorithms can be more efficient for multiple queries
  - Group variables together strategically.
- However, *all* exact inference can be computationally intractable.

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:

$$O(d^k n)$$

for k parents, d values.

Complexity of exact inference

- Multiply connected networks.
- Exponential time and space complexity, even when number of parents of a node is bounded.
- Inference is NP-hard.
- In fact, #P-complete.

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