Bayesian networks: Chapter 14.4–5

November 16, 2021

Outline

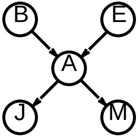
- Exact inference by enumeration
- Exact inference by variable elimination
- Approximate inference by stochastic simulation
- Approximate inference by Markov chain Monte Carlo

Inference tasks

- Simple queries: compute posterior marginal $P(X_i|\mathbf{E}=e)$
  e.g., $P(NoGas|Gauge=empty, Lights=on, Starts=false)$
- Conjunctive queries:
  $P(X_i, X_j|\mathbf{E}=e) = P(X_i|\mathbf{E}=e)P(X_j|X_i, \mathbf{E}=e)$
- 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 by enumeration

- Slightly intelligent way to sum out variables from the joint without actually constructing its explicit representation
- Simple query on the burglary network:



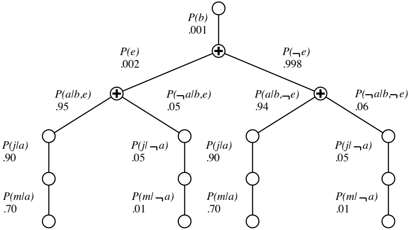
$$
\begin{aligned}
&P(B|j,m) = \\
&= P(B,j,m)/P(j,m) = \\
&= \alpha P(B,j,m) \\
&= \alpha \sum_e \sum_a P(B,e,a,j,m)
\end{aligned}
$$

- Rewrite full joint entries using product of CPT entries:

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

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

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 – 1

- Variable elimination Idea: carry out summations right-to-left, storing intermediate results (**factors**) to avoid re-computation
- To justify this consider the following expression: $uwy + uwz + uxy + uxz + vwy + vwz + vxy + vxz$
- To compute this we need 7 additions and $2 \times 8 = 16$ multiplications.
- There are 4 repeated sub-expressions, e.g., uw, ux, etc.
- Notice that we can rewrite this expression as

$$(u + v)(w + x)(y + z),$$

  in which case we will have 2 multiplications and 3 additions.
- This is the idea that we try to capture in variable elimination:

$$\begin{aligned}
&\textstyle\sum_{e,a} P(B)P(e)P(a|B,e)P(j|a)P(m|a)\\
&= P(B)P(e)P(a|B,e)P(j|a)P(m|a)\\
&+P(B)P(\neg e)P(a|B,\neg e)P(j|a)P(m|a)\\
&+P(B)P(e)P(\neg a|B,e)P(j|\neg a)P(m|\neg a)\\
&+P(B)P(\neg e)P(\neg a|B,\neg e)P(j|\neg a)P(m|\neg a)
\end{aligned}$$

Lots of repeated subexpressions!

Inference by variable elimination – 2

$$P(B|j,m) = \alpha \underbrace{P(B)}_{B} \textstyle\sum_e \underbrace{P(e)}_{E} \textstyle\sum_a \underbrace{P(a|B,e)}_{A} \underbrace{P(j|a)}_{J} \underbrace{P(m|a)}_{M}$$

$= \alpha P(B) \sum_e P(e) \sum_a P(a|B,e) P(j|a) f_M(a)$
$= \alpha P(B) \sum_e P(e) \sum_a P(a|B,e) f_J(a) f_M(a)$
$= \alpha P(B) \sum_e P(e) \sum_a f_A(a,b,e) f_J(a) f_M(a)$
$= \alpha P(B) \sum_e P(e) f_{\bar{A}JM}(b,e)$ (sum out A)
$= \alpha P(B) f_{\bar{E}\bar{A}JM}(b)$ (sum out E)
$= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)$
Note f_* is a notation for the summation operation.

Inference by variable elimination – 3

- Move summations inward as far as possible:

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

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

- Do the calculation from the inside out:
  - That is, sum over a first, then sum over e
  - **Problem:** $P(a|B,e)$ is not a single number, it's a bunch of different numbers depending on the values of B and e
  - **Solution:** Use *arrays of numbers* (of various dimensions) with appropriate operations on them; these are called **factors.**

Inference by variable elimination – 4 (Factor)

- Joint distribution $P(X, Y)$
  - Its entries are $P(x, y)$ for all values x of X, and y of Y
  - Thus is a $|X| \times |Y|$ matrix
  - Sums up to 1: $\sum_{x,y} P(x, y) = 1$

Table: $P(A, J)$

$A \setminus J$	true	false
true	0.09	0.01
false	0.045	0.855

- Projected joint: $P(x, Y)$
  - A "slice" of the joint distribution
  - Its entries are $P(x_0, y)$, i.e., for one x and all y
  - Thus it is an $|Y|$ - element vector
  - Sums up to $P(x_0)$: $\sum_y P(x_0, y) = P(x_0)$

Table: $P(a, J)$

$A \setminus J$	true	false
true	0.09	0.01

- 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

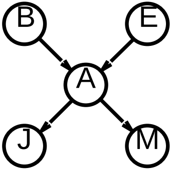
$$\sum_x f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \sum_x f_{i+1} \times \cdots \times f_k = f_1 \times \cdots \times f_i \times f_{\bar{X}}$$

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

- Pointwise product of factors f_1 and f_2:
  $f_1(x_1, \ldots, x_j, y_1, \ldots, y_k) \times f_2(y_1, \ldots, y_k, z_1, \ldots, z_l) =$
  $f(x_1, \ldots, x_j, y_1, \ldots, y_k, z_1, \ldots, z_l)$
  E.g., $f_1(a, b) \times f_2(b, c) = f(a, b, c)$

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 to $P(X|\ldots)$, unless $Y \in Ancestors(\{X\} \cup \mathbf{E})$

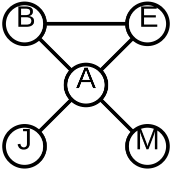
Here, $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and $Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$

so *MaryCalls* is irrelevant

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

Irrelevant variables contd.

- **Definition:** moral graph of Bayes net: marry all parents and drop arrows
- **Definition: A** is m-separated from **B** by **C** $\iff$ separated by **C** in the moral graph
- **Theorem 2:** Y is irrelevant if m-separated from X by



- For $P(JohnCalls|Alarm = true)$, both *Burglary* and *Earthquake* are irrelevant

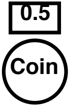
Complexity of exact inference –1

- 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 $\implies$ NP-hard
  - equivalent to **counting** 3SAT models $\implies$ #P-complete ("number-P hard"), i.e. strictly harder that NP-complete.

Complexity of exact inference – 2

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

- Basic idea:
  1. Draw N samples from a sampling distribution

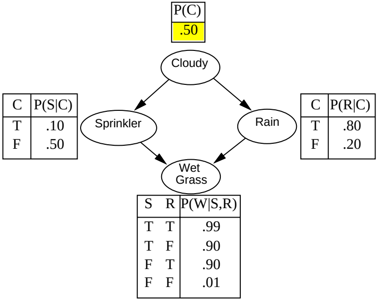


  2. Compute an approximate posterior probability $\widehat{P}$
  3. Show this converges to the true probability P
- Outline:
  1. Sampling from an empty network
  2. Rejection sampling: reject samples disagreeing with evidence
  3. Likelihood weighting: use evidence to weight samples

Sampling from an empty network

Prior-Sample(bn): an event sampled from bn
Input: **bn** - a belief network specifying joint distribution $P(X_1, \ldots, X_n)$
x is an event with n elements
For $i = 1$ to n do
    x_i: a random sample from $P(X_i | Parents(X_i))$
    given the values of $Parents(X_i)$ in x
Return

Example



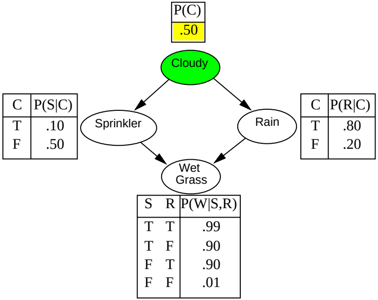
P(C)
.50

C	P(S|C)
T	.10
F	.50

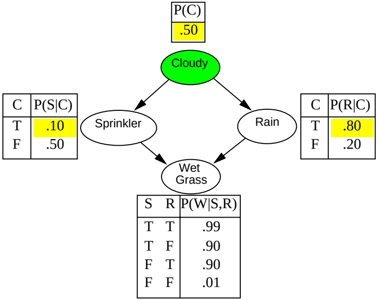
C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

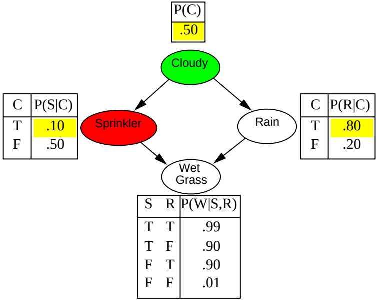
Example


P(C)
.50
Cloudy
C P(S|C)
T .10
F .50
Sprinkler
Rain
C P(R|C)
T .80
F .20
Wet Grass
S R P(W|S,R)
T T .99
T F .90
F T .90
F F .01

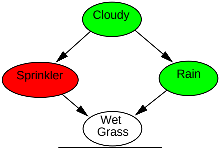
Example


P(C)
.50
Cloudy
C
P(S|C)
T
.10
F
.50
Sprinkler
Rain
C
P(R|C)
T
.80
F
.20
Wet Grass
S
R
P(W|S,R)
T T .99
T F .90
F T .90
F F .01

Example


P(C)
.50
Cloudy
C P(S|C)
T .10
F .50
Sprinkler
Rain
C P(R|C)
T .80
F .20
Wet Grass
S R P(W|S,R)
T T .99
T F .90
F T .90
F F .01

Example



P(C)
.50

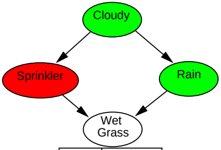
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example

P(C)
.50

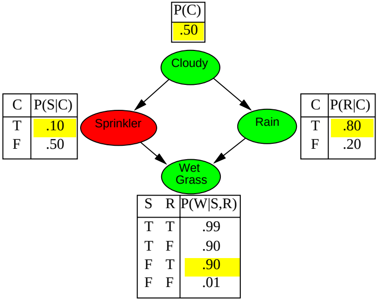


C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example



P(C)
.50
Cloudy
C P(S|C)
T .10
F .50
Sprinkler
Rain
C P(R|C)
T .80
F .20
Wet Grass
S R P(W|S,R)
T T .99
T F .90
F T .90
F F .01

Sampling from an empty network contd.

Probability that *PriorSample* generates a particular event :

$$S_{PS}(x_1 \ldots x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i)) = P(x_1 \ldots x_n)$$

i.e., the true prior probability
For example,
$S_{PS}(t, f, t, t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 = P(t, f, t, t)$
Let $N_{PS}(x_1 \ldots x_n)$ be the number of samples generated for event $x_1, \ldots, x_n$
Then we have

$$\begin{aligned} \lim_{N \to \infty} \widehat{P}(x_1, \ldots, x_n) &= \lim_{N \to \infty} N_{PS}(x_1, \ldots, x_n)/N \\ &= S_{PS}(x_1, \ldots, x_n) \\ &= P(x_1 \ldots x_n) \end{aligned}$$

That is, estimates derived from PriorSample are *consistent*
Shorthand: $\widehat{P}(x_1, \ldots, x_n) \approx P(x_1 \ldots x_n)$

Rejection sampling

$\widehat{P}(X|e)$ estimated from samples agreeing with e
E.g., estimate $P(Rain|Sprinkler = true)$ using 100 samples 27 samples have $Sprinkler = true$.

Of these, 8 have $Rain = true$ and 19 have $Rain = false$.

$\widehat{P}(Rain|Sprinkler = true) = Normalize(\langle 8, 19 \rangle) = \langle 0.296, 0.704 \rangle$
Similar to a basic real-world empirical estimation procedure

Likelihood weighting

Idea: fix evidence variables, sample only non-evidence variables, and weight each sample by the likelihood it accords the evidence

```
function LIKELIHOOD-WEIGHTING(X, e, bn, N) returns an
estimate of P(X|e)
   local variables: W, a vector of weighted counts over X,
   initially zero

   for j = 1 to N do
       x, w ← WEIGHTED-SAMPLE(bn)
       W[x] ← W[x] + w where x is the value of X in x
   return NORMALIZE(W[X])
```

```
function WEIGHTED-SAMPLE(bn, e) returns an event and a weight

   x ← an event with n elements; w ← 1
   for i = 1 to n do
       if X_i has a value x_i in e
           then w ← w × P(X_i = x_i | parents(X_i))
           else x_i ← a random sample from P(X_i | parents(X_i))
   return x, w
```