CS 188: Artificial Intelligence
Fall 2007

Lecture 18: Bayes Nets III

10/30/2007

Dan Klein – UC Berkeley

Announcements

- Project shift:
  - Project 4 moved back a little
  - Instead, mega-mini-homework, worth 3x, graded
- Contest is live

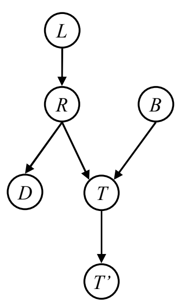
Inference

- Inference: calculating some statistic from a joint probability distribution
- Examples:
  - Posterior probability:

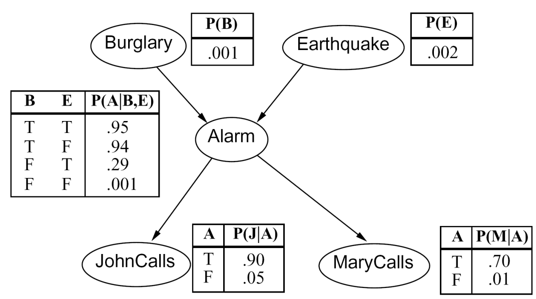
    $$P(Q|E_1 = e_1, \ldots E_k = e_k)$$

  - Most likely explanation:

    $$\text{argmax}_q \; P(Q = q|E_1 = e_1 \ldots)$$



Reminder: Alarm Network



P(B)
.001

P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

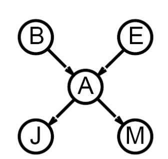
A	P(M|A)
T	.70
F	.01

Normalization Trick

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

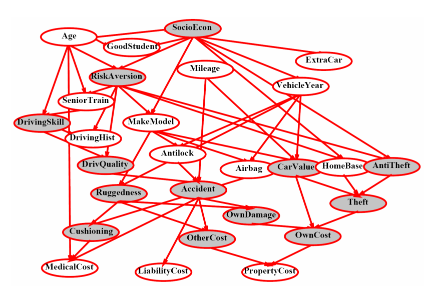
$$P(b,j,m) = \sum_{e,a} P(b,e,a,j,m)$$

$$P(\bar{b},j,m) = \sum_{e,a} P(\bar{b},e,a,j,m)$$

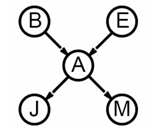


$$\begin{pmatrix} P(b,j,m) \\ P(\bar{b},j,m) \end{pmatrix} \xrightarrow{\text{Normalize}} \begin{pmatrix} P(b|j,m) \\ P(\bar{b}|j,m) \end{pmatrix}$$

Inference by Enumeration?



Nesting Sums

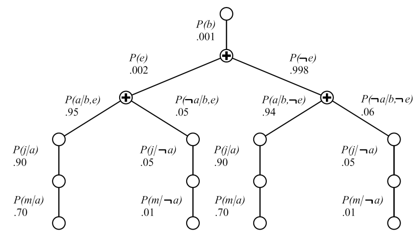


- Atomic inference is extremely slow!
- Slightly clever way to save work:
  - Move the sums as far right as possible
  - Example:

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

Evaluation Tree

- View the nested sums as a computation tree:

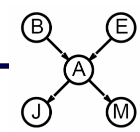


- Still repeated work: calculate P(m | a) P(j | a) twice, etc.

Variable Elimination: Idea

- Lots of redundant work in the computation tree
- We can save time if we cache all partial results
  - Join on one hidden variable at a time
  - Project out that variable immediately
- This is the basic idea behind variable elimination

Basic Objects



- Track objects called factors
- Initial factors are local CPTs

$$\underbrace{P(B)}_{f_B(B)} \quad \underbrace{P(J|A)}_{f_J(A,J)} \quad \underbrace{P(A|B,E)}_{f_A(A,B,E)}$$

- During elimination, create new factors
- Anatomy of a factor:

$$f_{A\bar{B}\bar{C}D}(D,E)$$

4 numbers, one for each value of D and E

Variables introduced

Variables summed out

Argument variables, always non-evidence variables

Basic Operations

- First basic operation: join factors
- Combining two factors:
  - Just like a database join
  - Build a factor over the union of the domains
- Example:

$$f_1(A,B) \times f_2(B,C) \Rightarrow f_3(A,B,C)$$

$$f_3(a,b,c) = f_1(a,b) \cdot f_2(b,c)$$

$$\text{“}P(a,b|c) = P(a|b) \cdot P(b|c)\text{”}$$

Basic Operations

- Second basic operation: marginalization
- Take a factor and sum out a variable
  - Shrinks a factor to a smaller one
  - A projection operation
- Example:

$$f_{\bar{A}B}(b) = \sum_a f_{AB}(a,b)$$

$$\text{“}P(b) = \sum_a P(a,b)\text{”}$$

Example

$P(b,j,m)$

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

$$= f_B(b) \sum_e f_E(e) \sum_a f_A(a,b,e) f_J(a) f_M(a)$$

$$= f_B(b) \sum_e f_E(e) \sum_a f_{AJM}(a,b,e)$$

$$= f_B(b) \sum_e f_E(e) f_{\bar{A}JM}(b,e)$$

Example

$P(b,j,m)$

$$= f_B(b) \sum_e f_E(e) f_{\bar{A}JM}(b,e)$$

$$= f_B(b) \sum_e f_{\bar{A}EJM}(b,e)$$

$$= f_B(b) f_{\bar{A}\bar{E}JM}(b)$$

$$= f_{\bar{A}B\bar{E}JM}(b)$$

General Variable Elimination

- Query: $P(Q|E_1 = e_1, \dots E_k = e_k)$
- Start with initial factors:
  - Local CPTs (but instantiated by evidence)
- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Project out H
- Join all remaining factors and normalize

Example

$P(B|j,m) \propto P(B,j,m)$

$P(B)$	$P(E)$	$P(A|B,E)$	$P(j|A)$	$P(m|A)$
$f_B(B)$	$f_E(E)$	$f_A(A,B,E)$	$f_J(A)$	$f_M(A)$

Choose A

$f_A(A,B,E)$, $f_J(A)$, $f_M(A)$ $\xrightarrow{\times}$ $f_{AJM}(A,B,E)$ $\xrightarrow{\Sigma}$ $f_{\bar{A}JM}(B,E)$

$f_B(B) \quad f_E(E) \quad f_{\bar{A}JM}(B,E)$

Example

$f_B(B) \quad f_E(E) \quad f_{\bar{A}JM}(B,E)$

Choose E

$f_E(E)$, $f_{\bar{A}JM}(B,E)$ $\xrightarrow{\times}$ $f_{\bar{A}EJM}(B,E)$ $\xrightarrow{\Sigma}$ $f_{\bar{A}\bar{E}JM}(B)$

$f_B(B) \quad f_{\bar{A}\bar{E}JM}(B)$

Finish

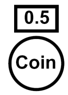
$f_B(B)$, $f_{\bar{A}\bar{E}JM}(B)$ $\xrightarrow{\times}$ $f_{\bar{A}B\bar{E}JM}(B)$ $\xrightarrow{\text{Normalize}}$ $P(B|j,m)$

Variable Elimination

- What you need to know:
  - VE caches intermediate computations
  - Polynomial time for tree-structured graphs!
  - Saves time by marginalizing variables ask soon as possible rather than at the end
- We will see special cases of VE later
  - You'll have to implement the special cases
- Approximations
  - Exact inference is slow, especially when you have a lot of hidden nodes
  - Approximate methods give you a (close) answer, faster

Sampling

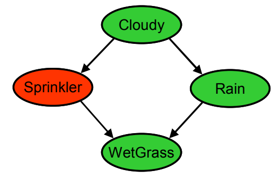
- Basic idea:
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability
  - Show this converges to the true probability P



- Outline:
  - Sampling from an empty network
  - Rejection sampling: reject samples disagreeing with evidence
  - Likelihood weighting: use evidence to weight samples

Prior Sampling

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

Prior Sampling

- This process generates samples with probability

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

…i.e. the BN's joint probability

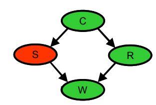
- Let the number of samples of an event be $N_{PS}(x_1 \dots x_n)$

- Then
$$\lim_{N\to\infty} \hat{P}(x_1, \dots, x_n) = \lim_{N\to\infty} N_{PS}(x_1, \dots, x_n)/N = S_{PS}(x_1, \dots, x_n) = P(x_1 \dots x_n)$$

- I.e., the sampling procedure is consistent

Example

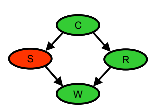
- We'll get a bunch of samples from the BN:
  c, ¬s, r, w
  c, s, r, w
  ¬c, s, r, ¬w
  c, ¬s, r, w
  ¬c, s, ¬r, w



- If we want to know P(W)
  - We have counts <w:4, ¬w:1>
  - Normalize to get P(W) = <w:0.8, ¬w:0.2>
  - This will get closer to the true distribution with more samples
  - Can estimate anything else, too
  - What about P(C| ¬r)?   P(C| ¬r, ¬w)?

Rejection Sampling

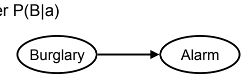
- Let's say we want P(C)
  - No point keeping all samples around
  - Just tally counts of C outcomes
- Let's say we want P(C| s)
  - Same thing: tally C outcomes, but ignore (reject) samples which don't have S=s
  - This is rejection sampling
  - It is also consistent (correct in the limit)



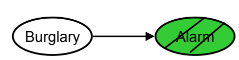
c, ¬s, r, w
c, s, r, w
¬c, s, r, ¬w
c, ¬s, r, w
¬c, s, ¬r, w

Likelihood Weighting

- Problem with rejection sampling:
  - If evidence is unlikely, you reject a lot of samples
  - You don't exploit your evidence as you sample
  - Consider P(B|a)



- Idea: fix evidence variables and sample the rest

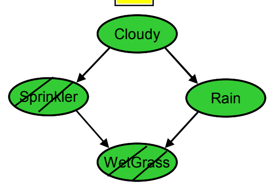


- Problem: sample distribution not consistent!
- Solution: weight by probability of evidence given parents

Likelihood Sampling

P(C)
.50

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



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

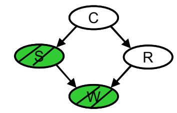
$w = 1.0 \times 0.1 \times 0.99$

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

Likelihood Weighting

- Sampling distribution if z sampled and e fixed evidence

$$S_{WS}(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{l} P(z_i | \text{Parents}(Z_i))$$



- Now, samples have weights

$$w(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{m} P(e_i | \text{Parents}(E_i))$$

- Together, weighted sampling distribution is consistent

$$S_{WS}(\mathbf{z}, \mathbf{e}) w(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{l} P(z_i | \text{Parents}(Z_i)) \prod_{i=1}^{m} P(e_i | \text{Parents}(E_i))$$

$$= P(\mathbf{z}, \mathbf{e})$$

Likelihood Weighting

- Note that likelihood weighting doesn't solve all our problems
- Rare evidence is taken into account for downstream variables, but not upstream ones
- A better solution is Markov-chain Monte Carlo (MCMC), more advanced
- We'll return to sampling for robot localization and tracking in dynamic BNs