Bayes Nets IV: Sampling

Hal Daumé III

Computer Science
University of Maryland

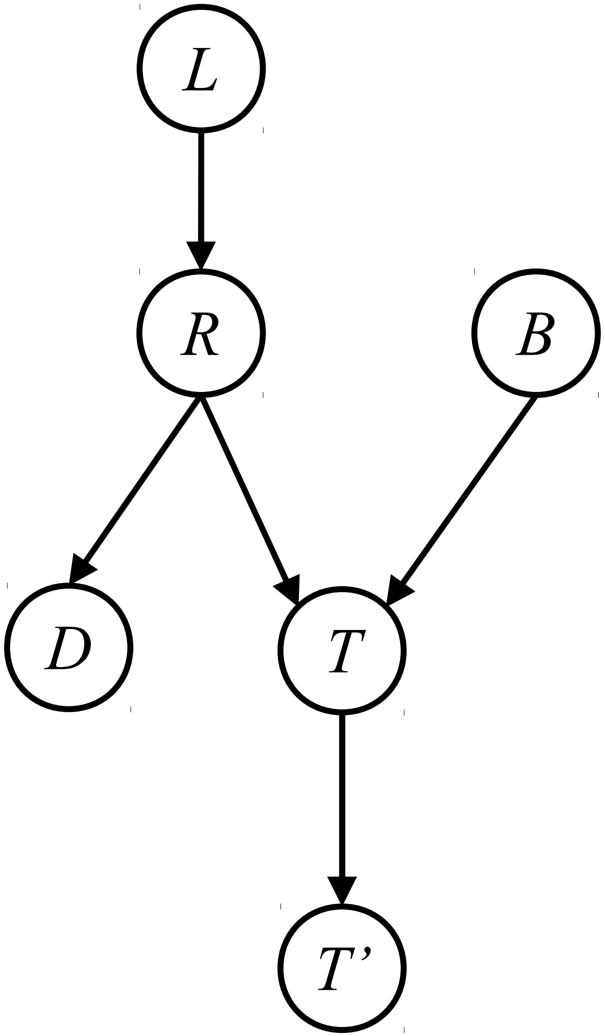
me@hal3.name

CS 421: Introduction to Artificial Intelligence

12 Apr 2012

Many slides courtesy of Dan Klein, Stuart Russell, or Andrew Moore

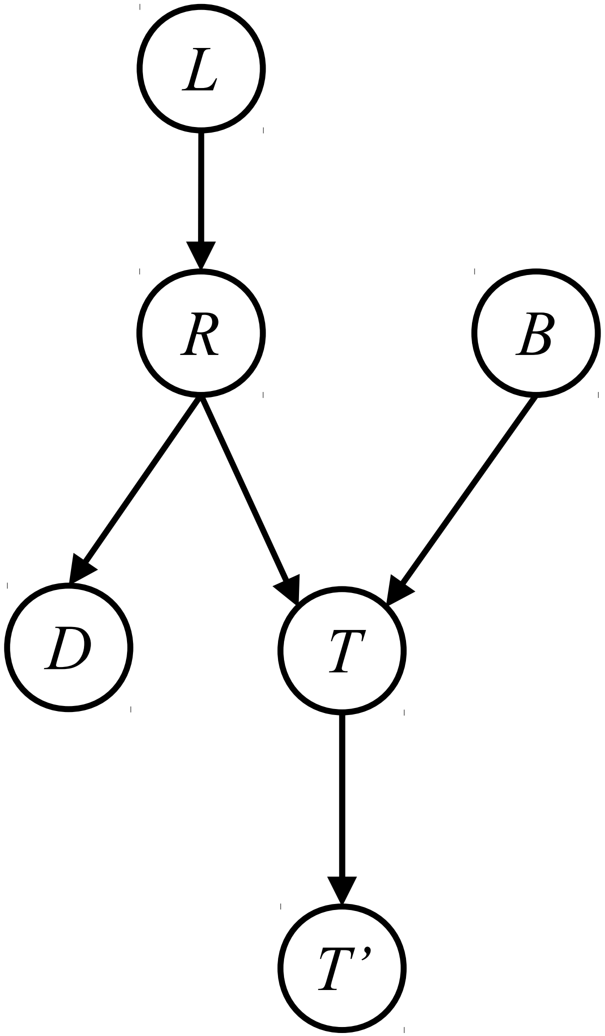
Exercise



- Consider:
  - Evidence = b, not t'
  - Query = r?

- Questions:
  - What are the initial factors?
  - If you ran inference by enumeration, how many things would you have to sum over?
  - Run variable elimination, choosing alphabetically when you have to select a variable

Exercise II



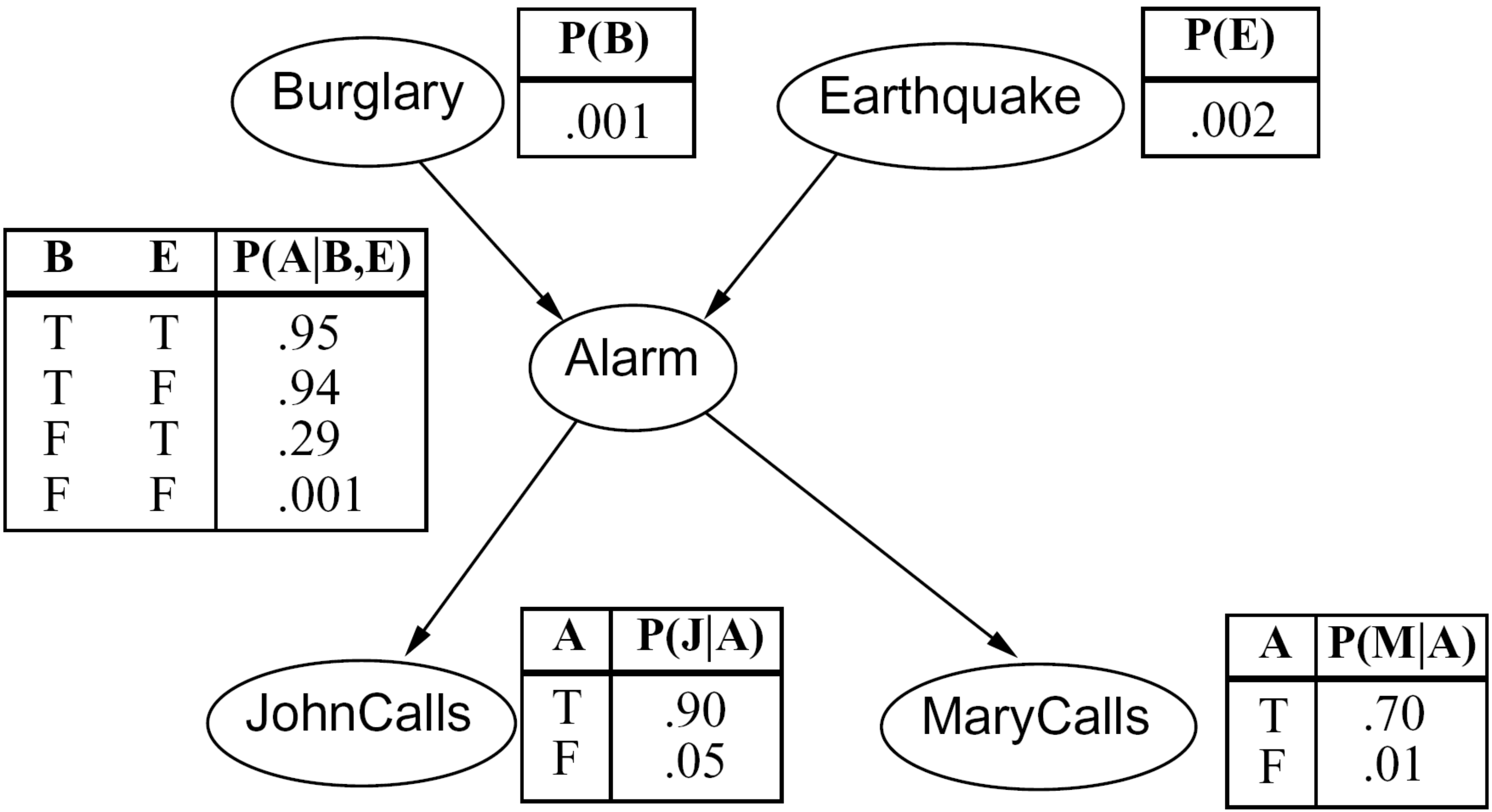
- Consider:
  - Evidence = b, not t'
  - Query = r?

L	p(r | L)
T	0.8
F	0.4

R	p(d | R)
T	0.6
F	0.5

R	B	p(t | R, B)
T	T	0.2
T	F	0.1
F	T	0.7
F	F	0.3

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

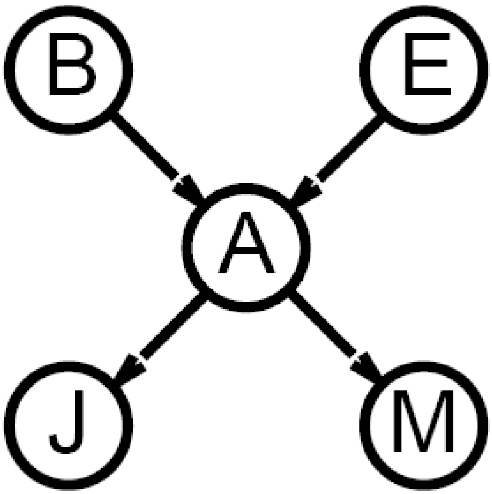
A	P(J|A)
T	.90
F	.05

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

Inference by Enumeration

- Given unlimited time, inference in BNs is easy
- Recipe:
  - State the marginal probabilities you need
  - Figure out ALL the atomic probabilities you need
  - Calculate and combine them
- Example:

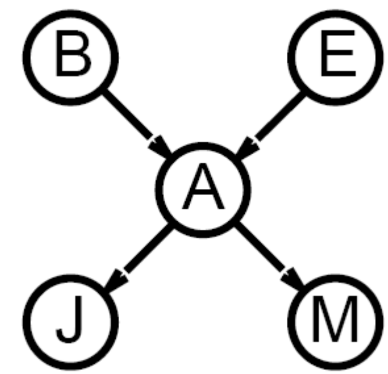
$$P(b|j,m) = \frac{P(b,j,m)}{P(j,m)}$$



General Variable Elimination

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



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

$P(B)$	$P(E)$	$P(A\|B,E)$	$P(j\|A)$	$P(m\|A)$

Choose A

$P(A|B,E)$

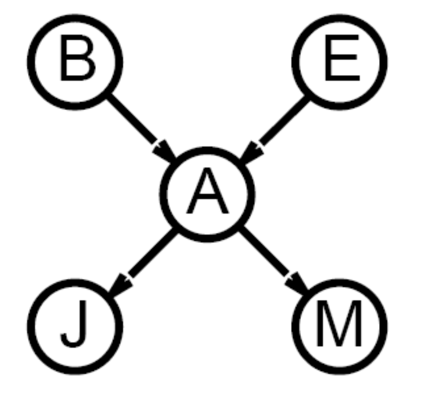
$P(j|A)$

$P(m|A)$

$$\xrightarrow{\times} P(j,m,A|B,E) \xrightarrow{\sum} P(j,m|B,E)$$

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

Example



$$P(B) \qquad P(E) \qquad P(j, m|B, E)$$

Choose E

$$P(E) \quad P(j, m|B, E) \quad \xrightarrow{\times} \quad P(j, m, E|B) \quad \xrightarrow{\sum} \quad P(j, m|B)$$

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

Finish with B

$$P(B) \quad P(j, m|B) \quad \xrightarrow{\times} \quad P(j, m, B) \quad \xrightarrow{\text{Normalize}} \quad P(B|j, m)$$

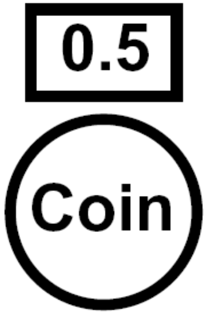
Variable Elimination

- What you need to know:
  - Should be able to run it on small examples, understand the factor creation / reduction flow
  - Better than enumeration: VE caches intermediate computations
  - Saves time by marginalizing variables as soon as possible rather than at the end
  - Polynomial time for tree-structured graphs – sound familiar?
- We will see special cases of VE later
  - You'll have to implement the special cases

- Approximations
  - Exact inference is slow, especially with a lot of hidden nodes
  - Approximate methods give you a (close, wrong?) answer, faster

Sampling

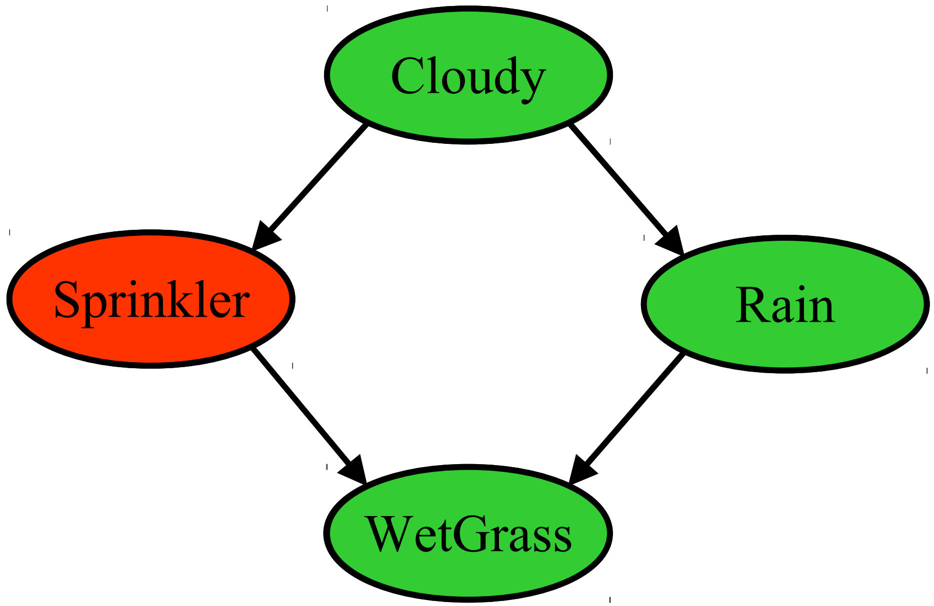
- Basic idea:
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability
  - Show this converges to the true probability P



- 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

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

…i.e. the BN's joint probability

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

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

- I.e., the sampling procedure is consistent

Example

- We'll get a bunch of samples from the BN:

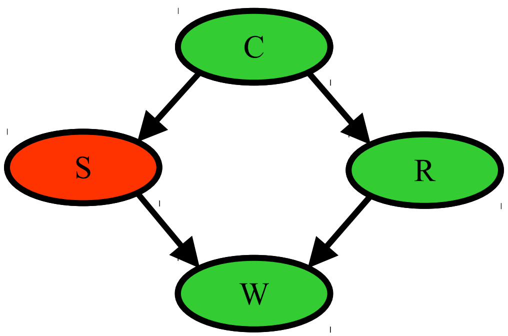
  c, ¬s, r, w

  c, s, r, w

  ¬c, s, r, ¬w

  c, ¬s, r, w

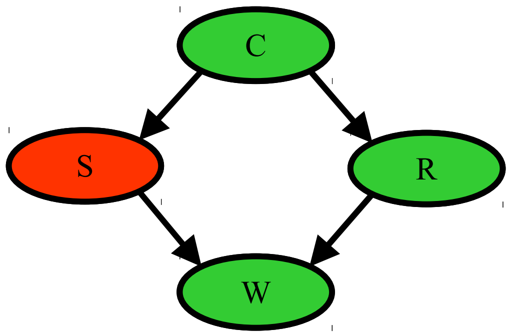
  ¬c, s, ¬r, w



- 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)

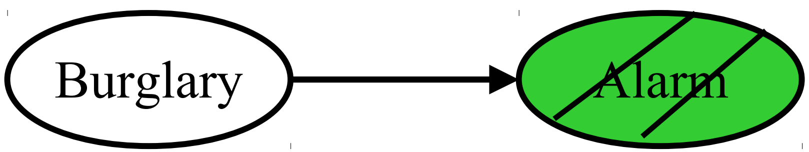


c, ¬s, r, w
c, s, r, w
¬c, s, r, ¬w
c, ¬s, r, w
¬c, s, ¬r, w

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)

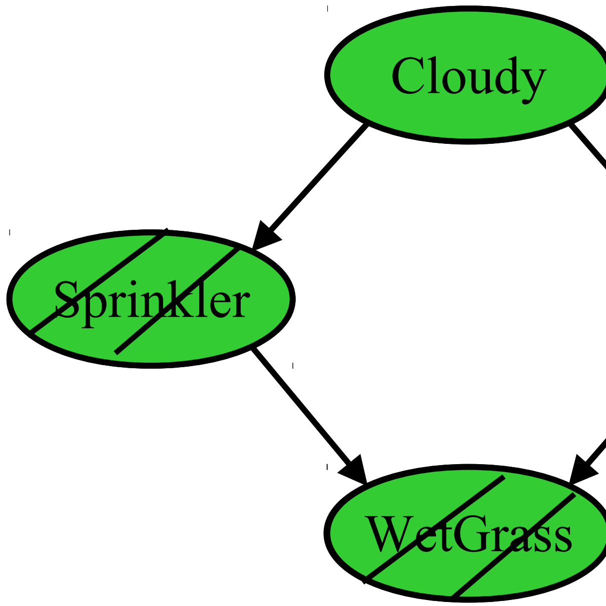
- Idea: fix evidence variables and sample the rest



- 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

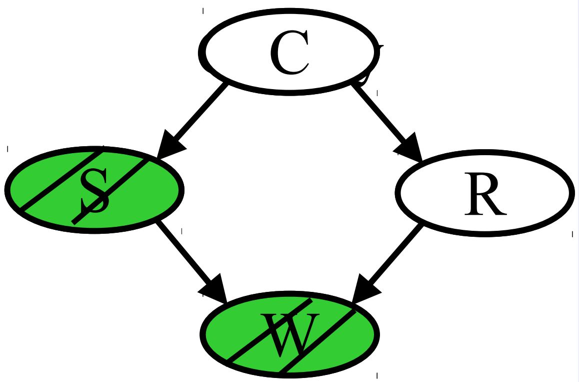
S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

w = 1.0 * 0.1 * 0.99

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

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

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