CSE 473: Artificial Intelligence
Autumn 2011

Bayesian Networks: Inference

Luke Zettlemoyer

Many slides over the course adapted from either Dan Klein, Stuart Russell or Andrew Moore

Outline

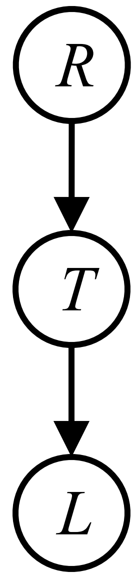
- Bayesian Networks Inference
  - Exact Inference: Variable Elimination
  - Approximate Inference: Sampling

Variable Elimination

- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables
  - You end up repeating a lot of work!
- Idea: interleave joining and marginalizing!
  - Called “Variable Elimination”
  - Still NP-hard, but usually much faster than inference by enumeration
- We’ll need some new notation to define VE

Example: Traffic Domain

- Random Variables
  - R: Raining
  - T: Traffic
  - L: Late for class!



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|R)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- First query: P(L)

$$P(l) = \sum_t \sum_r P(l|t)P(t|r)P(r)$$

Variable Elimination Outline

- Maintain a set of tables called factors
- Initial factors are local CPTs (one per node)

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- Any known values are selected
  - E.g. if we know $L = +\ell$, the initial factors are

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(+\ell|T)$

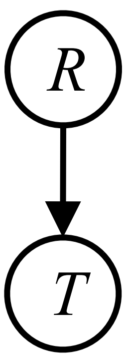
+t	+l	0.3
-t	+l	0.1

- VE: Alternately join factors and eliminate variables

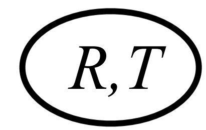
Operation 1: Join Factors

- First basic operation: joining factors
- Combining factors:
  - Just like a database join
  - Get all factors over the joining variable
  - Build a new factor over the union of the variables involved

- Example: Join on R



$P(R) \quad \times \quad P(T|R) \qquad \longrightarrow \qquad P(R,T)$



+r	0.1
-r	0.9

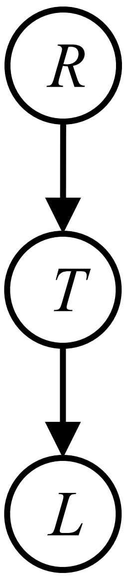
+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

- Computation for each entry: pointwise products

$$\forall r, t : \quad P(r,t) = P(r) \cdot P(t|r)$$

Example: Multiple Joins



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

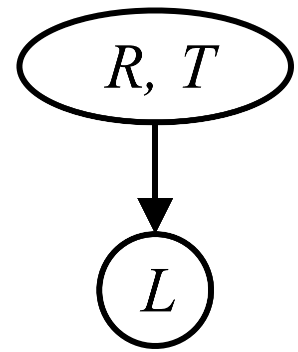
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R

$P(R,T)$

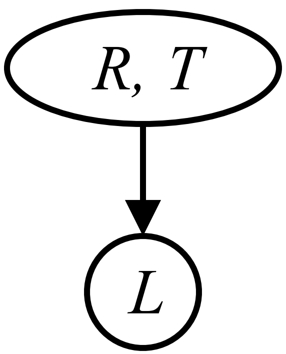
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Example: Multiple Joins



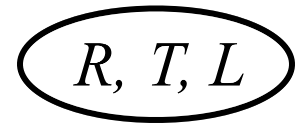
$P(R, T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join T



$P(R, T, L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729

Operation 2: Eliminate

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

$P(R, T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

$\xrightarrow{\text{sum } R}$

$P(T)$

+t	0.17
-t	0.83

Multiple Elimination

R, T, L — T, L — L

$P(R,T,L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729

Sum out R

$P(T,L)$

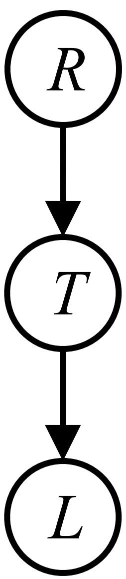
+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

Sum out T

$P(L)$

+l	0.134
-l	0.886

P(L) : Marginalizing Early!



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

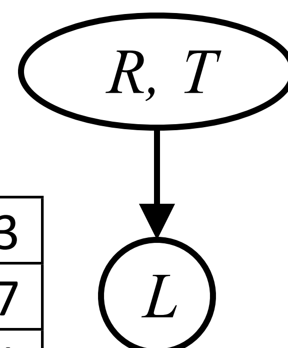
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R

$P(R,T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



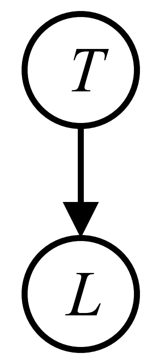
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Sum out R

$P(T)$

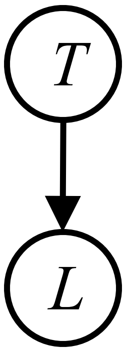
+t	0.17
-t	0.83



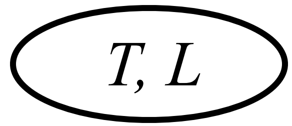
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

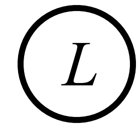
Marginalizing Early (aka VE*)



Join T



Sum out T



$P(T)$

+t	0.17
-t	0.83

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

$P(T, L)$

+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

$P(L)$

+l	0.134
-l	0.886

* VE is variable elimination

Evidence

- If evidence, start with factors that select that evidence
  - No evidence uses these initial factors:

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

  - Computing $P(L|+r)$, the initial factors become:

$P(+r)$

+r	0.1

$P(T|+r)$

+r	+t	0.8
+r	-t	0.2

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- We eliminate all vars other than query + evidence

Evidence II

- Result will be a selected joint of query and evidence
  - E.g. for P(L | +r), we'd end up with:

$P(+r, L)$

+r	+l	0.026
+r	-l	0.074

Normalize

$P(L| + r)$

+l	0.26
-l	0.74

- To get our answer, just normalize this!

- That's it!

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
  - Eliminate (sum out) H

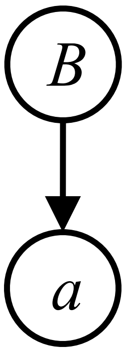
- Join all remaining factors and normalize

Variable Elimination Bayes Rule

Start / Select

$P(B)$

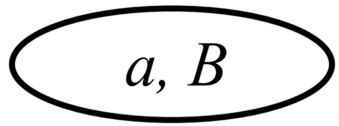
B	P
+b	0.1
¬b	0.9



$P(A|B) \rightarrow P(a|B)$

B	A	P
+b	+a	0.8
~~+b~~	~~¬a~~	~~0.2~~
¬b	+a	0.1
~~¬b~~	~~¬a~~	~~0.9~~

Join on B



$P(a, B)$

A	B	P
+a	+b	0.08
+a	¬b	0.09

Normalize

$P(B|a)$

A	B	P
+a	+b	8/17
+a	¬b	9/17

Example

Query: $P(B|j,m)$

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

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

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

Choose E

$P(E)$

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

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

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

Finish with B

$P(B)$

$P(j, m|B)$

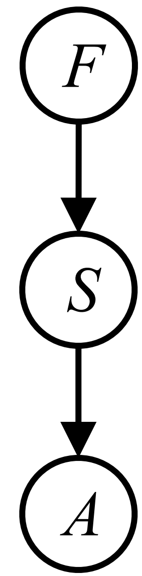
$\xrightarrow{\times}$ $P(j, m, B)$ $\xrightarrow{\text{Normalize}}$ $P(B|j, m)$

Exact Inference: Variable Elimination

- Remaining Issues:
  - Complexity: exponential in tree width (size of the largest factor created)
  - Best elimination ordering? NP-hard problem
- What you need to know:
  - Should be able to run it on small examples, understand the factor creation / reduction flow
  - Better than enumeration: saves time by marginalizing variables as soon as possible rather than at the end
- We have seen a special case of VE already
  - HMM Forward Inference

Approximate Inference

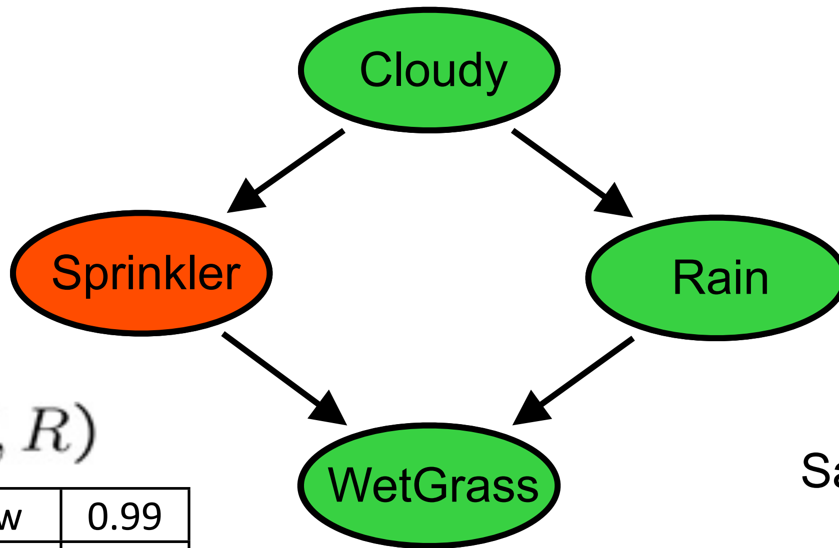
- Simulation has a name: sampling
- Sampling is a hot topic in machine learning, and it's really simple
- Basic idea:
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability
  - Show this converges to the true probability P
- Why sample?
  - Learning: get samples from a distribution you don't know
  - Inference: getting a sample is faster than computing the right answer (e.g. with variable elimination)



Prior Sampling

$P(C)$

+c	0.5
-c	0.5



$P(S|C)$

	+s	0.1
+c	-s	0.9
	+s	0.5
-c	-s	0.5

$P(R|C)$

	+r	0.8
+c	-r	0.2
	+r	0.2
-c	-r	0.8

$P(W|S,R)$

		+w	0.99
	+r	-w	0.01
		+w	0.90
+s	-r	-w	0.10
		+w	0.90
	+r	-w	0.10
		+w	0.01
-s	-r	-w	0.99

Samples:

+c, -s, +r, +w

-c, +s, -r, +w

…

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:

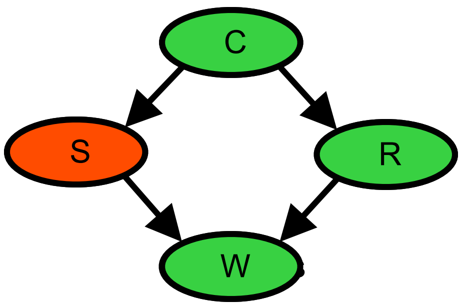
  +c, -s, +r, +w

  +c, +s, +r, +w

  -c, +s, +r, -w

  +c, -s, +r, +w

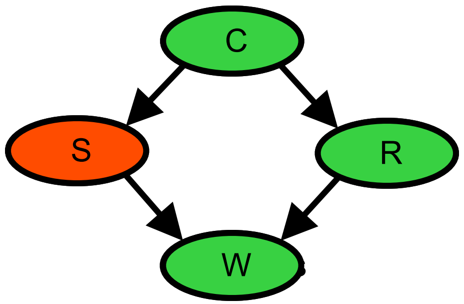
  -c, -s, -r, +w



- 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| +w)? P(C| +r, +w)? P(C| -r, -w)?
  - Fast: can use fewer samples if less time (what's the drawback?)

Rejection Sampling

- Let's say we want P(C)
  - No point keeping all samples around
  - Just tally counts of C as we go

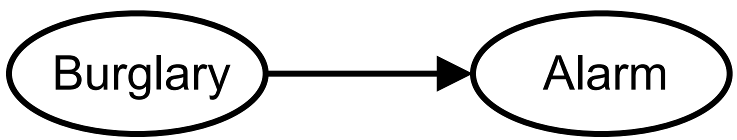


- Let's say we want P(C| +s)
  - Same thing: tally C outcomes, but ignore (reject) samples which don't have S=+s
  - This is called rejection sampling
  - It is also consistent for conditional probabilities (i.e., correct in the limit)

+c, -s, +r, +w
+c, +s, +r, +w
-c, +s, +r, -w
+c, -s, +r, +w
-c, -s, -r, +w

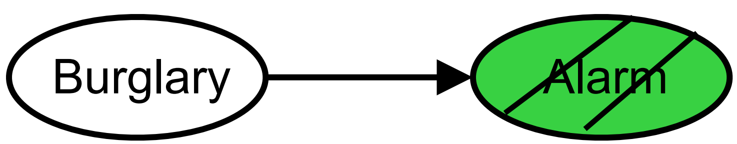
Likelihood Weighting

- 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)



-b, -a
-b, -a
-b, -a
-b, -a
+b, +a

- Idea: fix evidence variables and sample the rest



-b +a
-b, +a
-b, +a
-b, +a
+b, +a

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

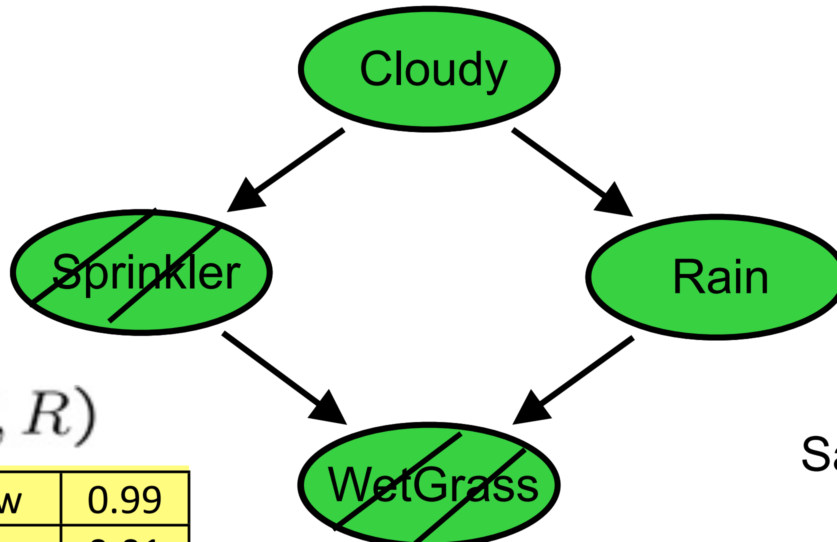
Likelihood Weighting

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

+c, +s, +r, +w

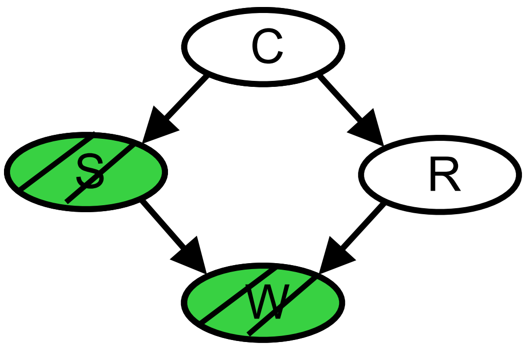
…

$$w = 1.0 \times 0.1 \times 0.99$$

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

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

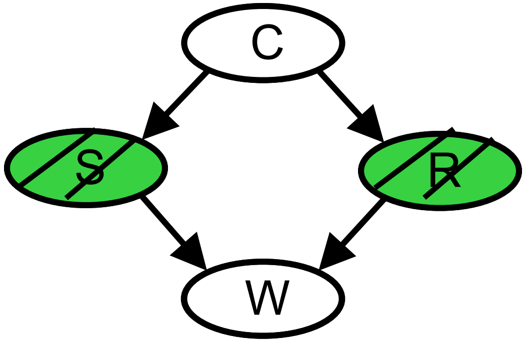
- Together, weighted sampling distribution is consistent

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

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

Likelihood Weighting

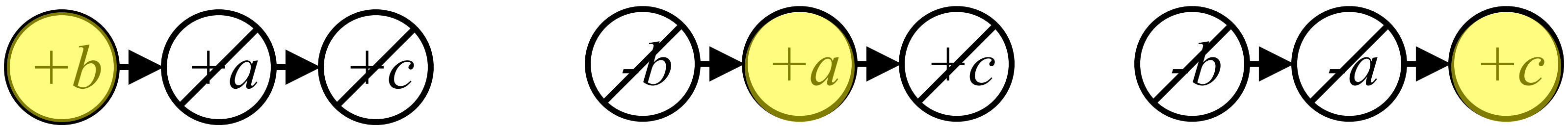
- Likelihood weighting is good
  - We have taken evidence into account as we generate the sample
  - E.g. here, W's value will get picked based on the evidence values of S, R
  - More of our samples will reflect the state of the world suggested by the evidence
- Likelihood weighting doesn't solve all our problems
  - Evidence influences the choice of downstream variables, but not upstream ones (C isn't more likely to get a value matching the evidence)
- We would like to consider evidence when we sample every variable



Markov Chain Monte Carlo*

- *Idea*: instead of sampling from scratch, create samples that are each like the last one.

- *Gibbs Sampling*: resample one variable at a time, conditioned on the rest, but keep evidence fixed.



- *Properties*: Now samples are not independent (in fact they're nearly identical), but sample averages are still consistent estimators!

- *What's the point*: both upstream and downstream variables condition on evidence.