CSE 573: Artificial Intelligence
Autumn 2010

Lecture 13: Bayesian Networks: Independence and Inference

11/15/2010

Luke Zettlemoyer

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

Outline

- Probabilistic models and inference
  - Bayesian Networks (BNs)
  - Independence in BNs
  - Exact Inference: Variable Elimination
  - Approximate Inference: Sampling

Announcements

- PS3 grades out yesterday
- PS4 in, done with Pacman -- Congrats!
- Mini-project guidelines out
- Exam Thursday
  - In class, closed book, one page of notes (front and back)
- Look at Berkley exams for practice:
  - http://inst.eecs.berkeley.edu/~cs188/fa10/midterm.html

Exam Topics

- Search
  - BFS, DFS, UCS, A* (tree and graph)
  - Completeness and Optimality
  - Heuristics: admissibility and consistency
- Games
  - Minimax, Alpha-beta pruning, Expectimax, Evaluation Functions
- MDPs
  - Definition, rewards, values, q-values
  - Bellman equations
  - Value and policy iteration
- Reinforcement Learning
  - Exploration vs Exploitation
  - Model-based vs. model-free
  - TD learning and Q-learning
  - Linear value function approx.
- Hidden Markov Models
  - Markov chains
  - Forward algorithm
  - Particle Filter
- Bayesian Networks
  - Basic definition
  - Types of independence

Model for Ghostbusters

- Reminder: ghost is hidden, sensors are noisy
- T: Top sensor is red
  B: Bottom sensor is red
  G: Ghost is in the top
- Queries:
  P(+g) = ??
  P(+g | +t) = ??
  P(+g | +t, -b) = ??
- Problem: joint distribution too large / complex

Joint Distribution

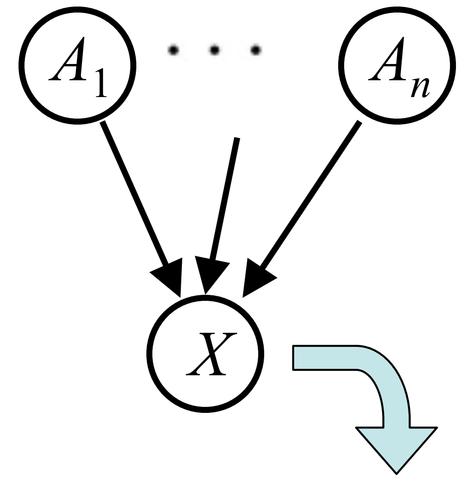
T	B	G	P
+t	+b	+g	0.16
+t	+b	¬g	0.16
+t	¬b	+g	0.24
+t	¬b	¬g	0.04
¬t	+b	+g	0.04
¬t	+b	¬g	0.24
¬t	¬b	+g	0.06
¬t	¬b	¬g	0.06

Recap: Bayes' Net Semantics

- Let's formalize the semantics of a Bayes' net
- A set of nodes, one per variable X
- A directed, acyclic graph
- A conditional distribution for each node
  - A collection of distributions over X, one for each combination of parents' values

    $$P(X|a_1 \ldots a_n)$$

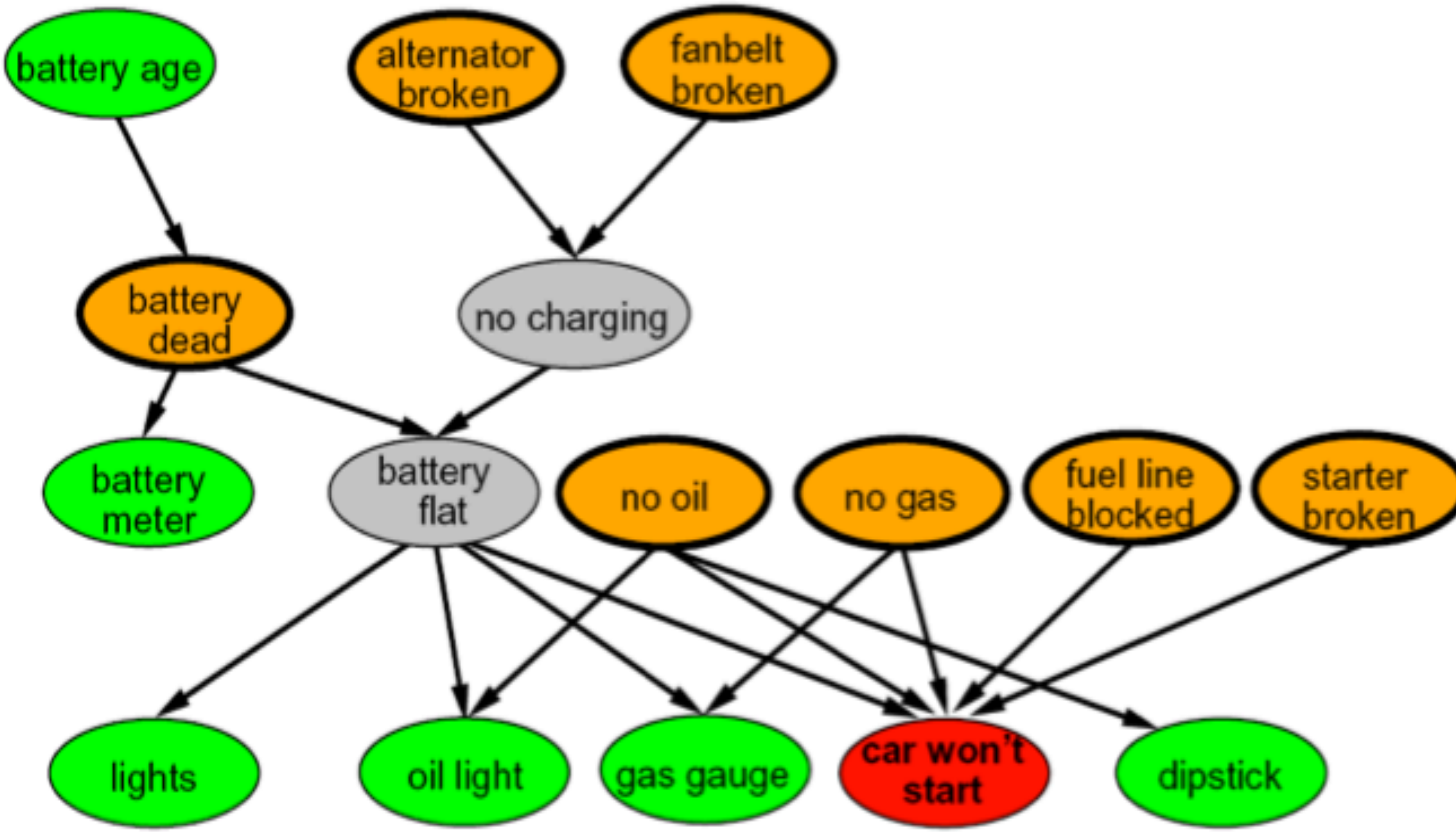
  - CPT: conditional probability table



$$P(X|A_1 \ldots A_n)$$

A Bayes net = Topology (graph) + Local Conditional Probabilities

Example Bayes' Net: Car



Recap: Probabilities in BNs

- Bayes' nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

- This lets us reconstruct any entry of the full joint
- Not every BN can represent every joint distribution
  - The topology enforces certain *independence* assumptions
  - Compare to the exact decomposition according to the chain rule!

Recap: Independence

- Two variables are *independent* if:

$$\forall x, y : P(x, y) = P(x)P(y)$$

  - This says that their joint distribution *factors* into a product two simpler distributions
  - Another form:

$$\forall x, y : P(x|y) = P(x)$$

  - We write: $X \perp\!\!\!\perp Y$

- Independence is a simplifying *modeling assumption*
  - *Empirical* joint distributions: at best "close" to independent
  - What could we assume for {Weather, Traffic, Cavity, Toothache}?

Recap: Conditional Independence

- Unconditional (absolute) independence very rare (why?)

- *Conditional independence* is our most basic and robust form of knowledge about uncertain environments:

$$\forall x, y, z : P(x, y|z) = P(x|z)P(y|z)$$

$$\forall x, y, z : P(x|z, y) = P(x|z)$$

$$X \perp\!\!\!\perp Y | Z$$

- What about this domain:
  - Traffic
  - Umbrella
  - Raining
- What about fire, smoke, alarm?

Ghostbusters Chain Rule

- Each sensor depends only on where the ghost is
- That means, the two sensors are conditionally independent, given the ghost position
- T: Top square is red
  B: Bottom square is red
  G: Ghost is in the top
- Can assume:
  $P(+g) = 0.5$
  $P(+t \mid +g) = 0.8$
  $P(+t \mid \neg g) = 0.4$
  $P(+b \mid +g) = 0.4$
  $P(+b \mid \neg g) = 0.8$

$$P(T,B,G) = P(G)\, P(T|G)\, P(B|G)$$

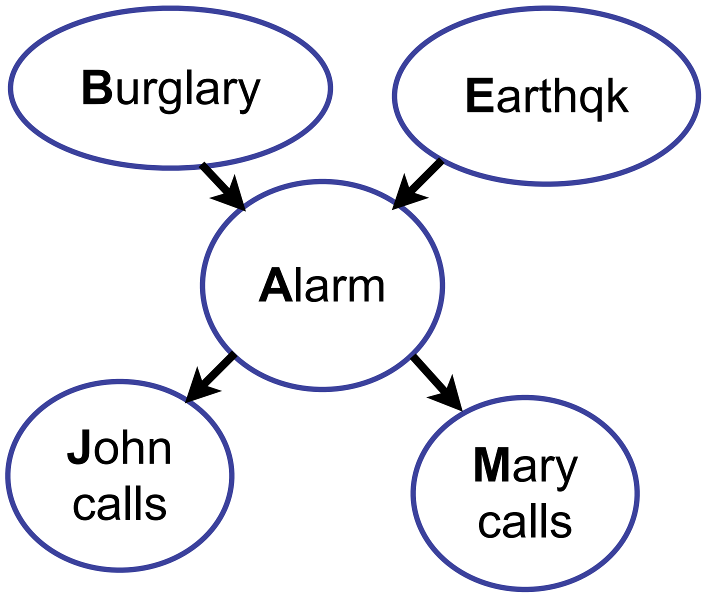
T	B	G	P
+t	+b	+g	0.16
+t	+b	¬g	0.16
+t	¬b	+g	0.24
+t	¬b	¬g	0.04
¬t	+b	+g	0.04
¬t	+b	¬g	0.24
¬t	¬b	+g	0.06
¬t	¬b	¬g	0.06

Example: Alarm Network

- Variables
  - B: Burglary
  - A: Alarm goes off
  - M: Mary calls
  - J: John calls
  - E: Earthquake!

Example: Alarm Network

B	P(B)
+b	0.001
¬b	0.999



E	P(E)
+e	0.002
¬e	0.998

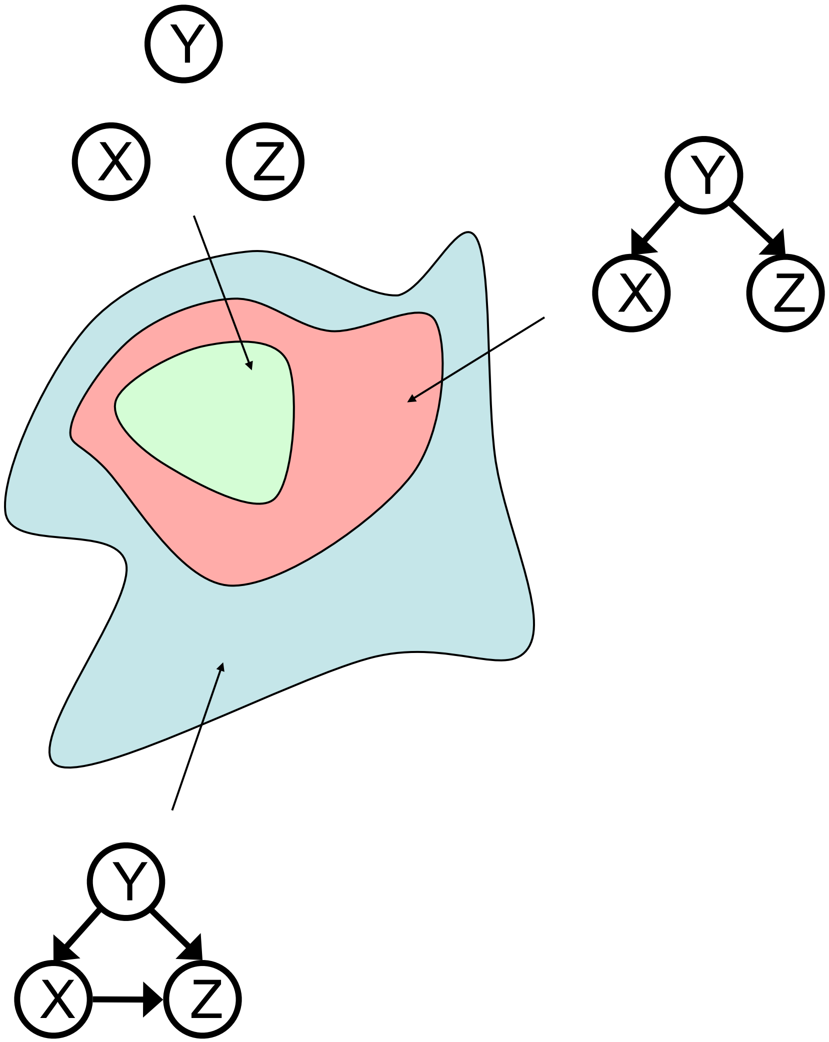
B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	¬a	0.05
+b	¬e	+a	0.94
+b	¬e	¬a	0.06
¬b	+e	+a	0.29
¬b	+e	¬a	0.71
¬b	¬e	+a	0.001
¬b	¬e	¬a	0.999

A	J	P(J|A)
+a	+j	0.9
+a	¬j	0.1
¬a	+j	0.05
¬a	¬j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	¬m	0.3
¬a	+m	0.01
¬a	¬m	0.99

Recap: Topology Limits Distributions

- Given some graph topology G, only certain joint distributions can be encoded
- The graph structure guarantees certain (conditional) independences
- (There might be more independence)
- Adding arcs increases the set of distributions, but has several costs
- Full conditioning can encode any distribution

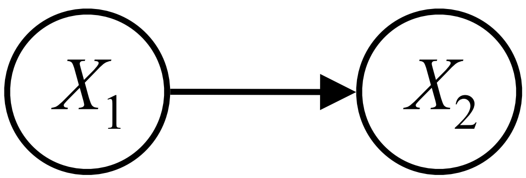


Changing Bayes' Net Structure

- The same joint distribution can be encoded in many different Bayes' nets
- Analysis question: given some edges, what other edges do you need to add?
  - One answer: fully connect the graph
  - Better answer: don't make any false conditional independence assumptions

Example: Coins

- Extra arcs don't prevent representing independence, just allow non-independence



$P(X_1)$

h	0.5
t	0.5

$P(X_2)$

h	0.5
t	0.5

$P(X_1)$

h	0.5
t	0.5

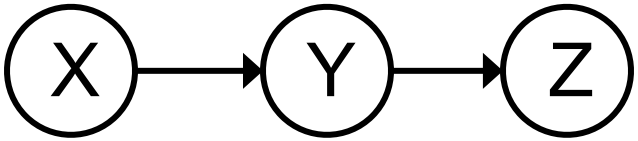
$P(X_2|X_1)$

h | h	0.5
t | h	0.5
h | t	0.5
t | t	0.5

- Adding unneeded arcs isn't wrong, it's just inefficient

Independence in a BN

- Important question about a BN:
  - Are two nodes independent given certain evidence?
  - If yes, can prove using algebra (tedious in general)
  - If no, can prove with a counter example
  - Example:



  - Question: are X and Z necessarily independent?
    - Answer: no. Example: low pressure causes rain, which causes traffic.
    - X can influence Z, Z can influence X (via Y)
    - Addendum: they *could* be independent: how?

Causal Chains

- This configuration is a "causal chain"

$$X \rightarrow Y \rightarrow Z$$

X: Low pressure

Y: Rain

Z: Traffic

$$P(x, y, z) = P(x)P(y|x)P(z|y)$$

- Is X independent of Z given Y?

$$P(z|x, y) = \frac{P(x, y, z)}{P(x, y)} = \frac{P(x)P(y|x)P(z|y)}{P(x)P(y|x)}$$

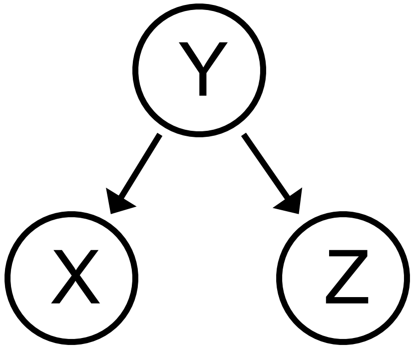
$$= P(z|y)$$

Yes!

- Evidence along the chain "blocks" the influence

Common Parent

- Another basic configuration: two children of the same parent
  - Are X and Z independent?
  - Are X and Z independent given Y?



$$P(z|x,y) = \frac{P(x,y,z)}{P(x,y)} = \frac{P(y)P(x|y)P(z|y)}{P(y)P(x|y)}$$

$$= P(z|y)$$

Yes!

Y: Project due

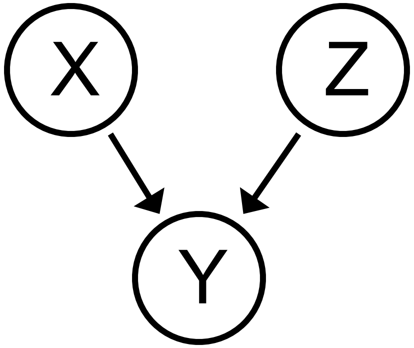
X: Newsgroup busy

Z: Lab full

- Observing the parent blocks influence between children.

Common Child

- Last configuration: two (or more) parents of one child (v-structures)
  - Are X and Z independent?
    - Yes: the ballgame and the rain cause traffic, but they are not correlated
    - Still need to prove they must be (try it!)
  - Are X and Z independent given Y?
    - No: seeing traffic puts the rain and the ballgame in competition as explanation?
  - This is backwards from the other cases
    - Observing an effect activates influence between possible parents.



X: Raining

Z: Ballgame

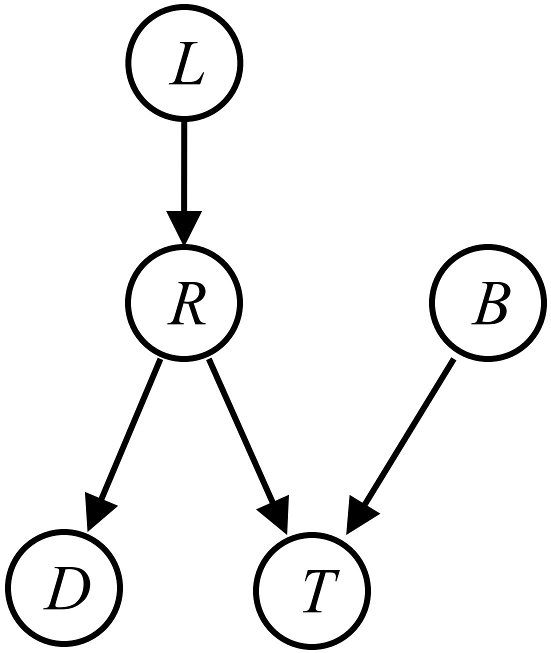
Y: Traffic

The General Case

- Any complex example can be analyzed using these three canonical cases
- General question: in a given BN, are two variables independent (given evidence)?
- Solution: analyze the graph

Reachability

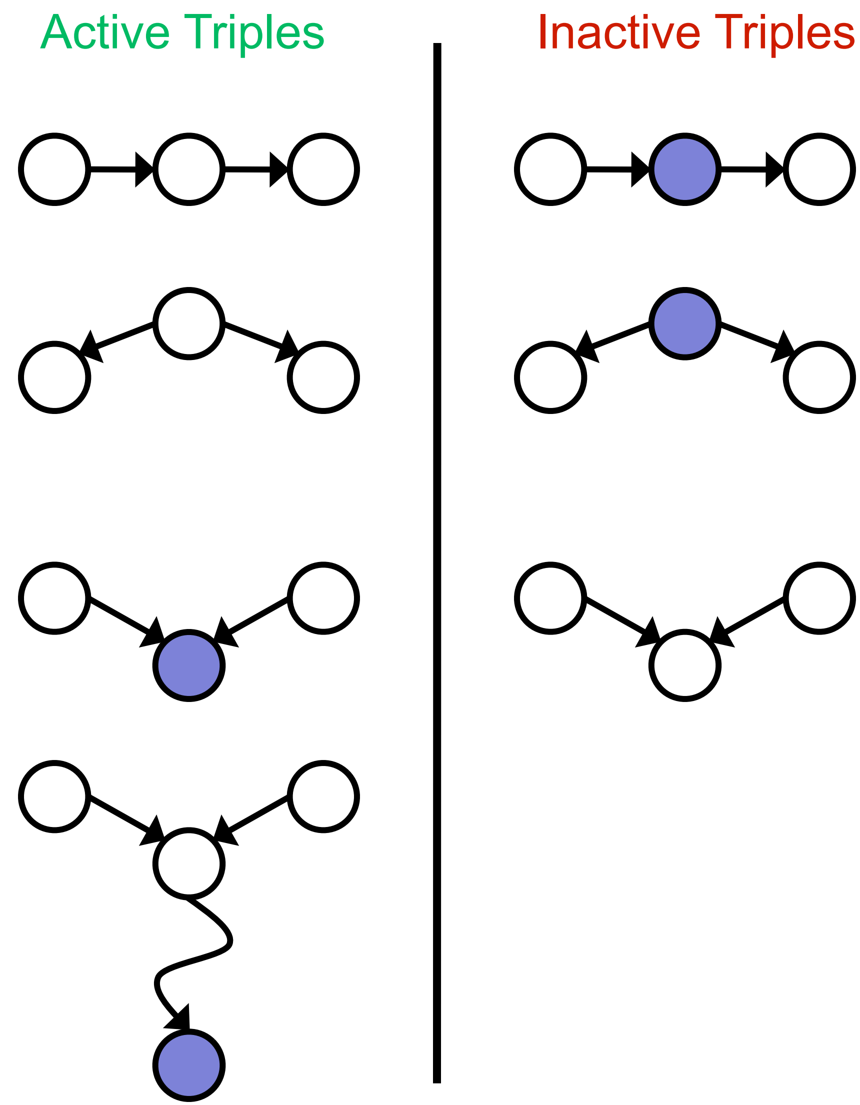
- Recipe: shade evidence nodes
- Attempt 1: if two nodes are connected by an undirected path not blocked by a shaded node, they are conditionally independent
- Almost works, but not quite
  - Where does it break?
  - Answer: the v-structure at T doesn't count as a link in a path unless "active"



Reachability (D-Separation)

- Question: Are X and Y conditionally independent given evidence vars {Z}?
  - Yes, if X and Y "separated" by Z
  - Look for active paths from X to Y
  - No active paths = independence!
- A path is active if each triple is active:
  - Causal chain $A \rightarrow B \rightarrow C$ where B is unobserved (either direction)
  - Common cause $A \leftarrow B \rightarrow C$ where B is unobserved
  - Common effect (aka v-structure)

    $A \rightarrow B \leftarrow C$ where B *or one of its descendents* is observed
- All it takes to block a path is a single inactive segment

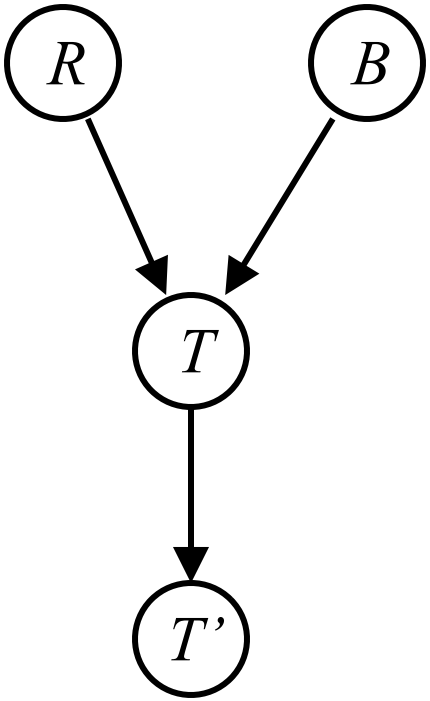


Example: Independent?

$R \perp\!\!\!\perp B$ Yes

$R \perp\!\!\!\perp B | T$

$R \perp\!\!\!\perp B | T'$



Example: Independent?

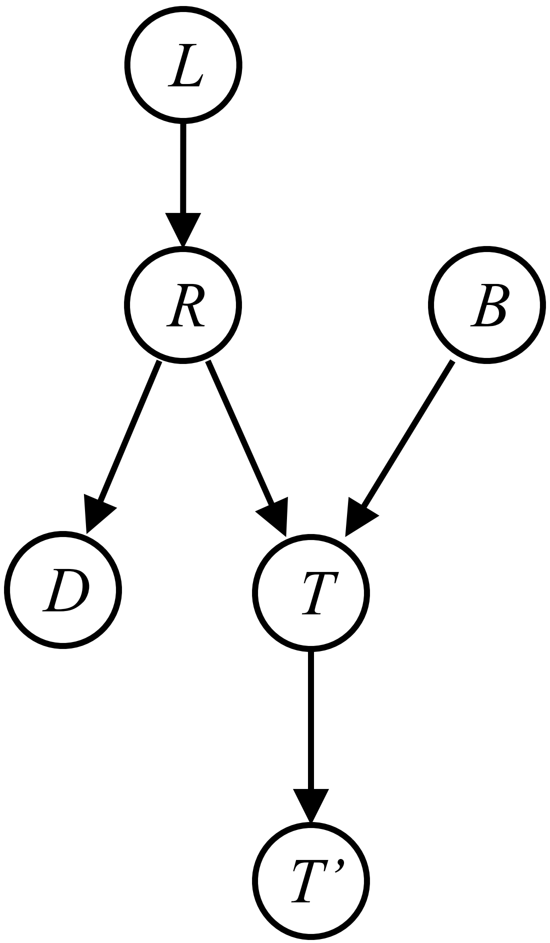
$L \perp\!\!\!\perp T' | T$ Yes

$L \perp\!\!\!\perp B$ Yes

$L \perp\!\!\!\perp B | T$

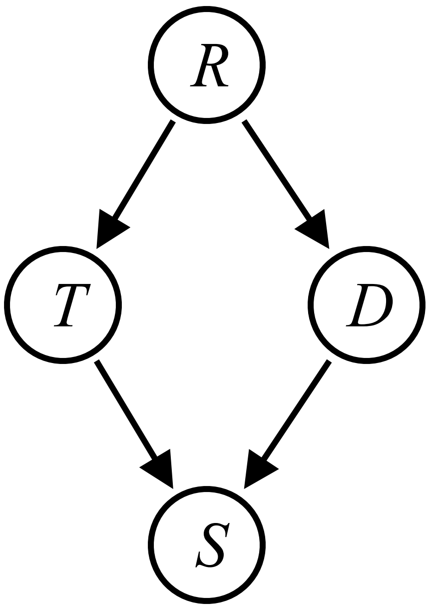
$L \perp\!\!\!\perp B | T'$

$L \perp\!\!\!\perp B | T, R$ Yes



Example

- Variables:
  - R: Raining
  - T: Traffic
  - D: Roof drips
  - S: I'm sad
- Questions:



$$T \perp\!\!\!\perp D$$

$$T \perp\!\!\!\perp D | R$$ Yes

$$T \perp\!\!\!\perp D | R, S$$

Summary

- Bayes nets compactly encode joint distributions
- Guaranteed independencies of distributions can be deduced from BN graph structure
- D-separation gives precise conditional independence guarantees from graph alone
- A Bayes’ net’s joint distribution may have further (conditional) independence that is not detectable until you inspect its specific distribution

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

Review: Factor Zoo I

- Joint distribution: P(X,Y)
  - Entries P(x,y) for all x, y
  - Sums to 1

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

- Selected joint: P(x,Y)
  - A slice of the joint distribution
  - Entries P(x,y) for fixed x, all y
  - Sums to P(x)

$P(cold, W)$

T	W	P
cold	sun	0.2
cold	rain	0.3

Review: Factor Zoo II

- Family of conditionals: P(X |Y)
  - Multiple conditionals
  - Entries P(x | y) for all x, y
  - Sums to |Y|

$P(W|T)$

T	W	P
hot	sun	0.8
hot	rain	0.2
cold	sun	0.4
cold	rain	0.6

$P(W|hot)$ (rows 1–2); $P(W|cold)$ (rows 3–4)

- Single conditional: P(Y | x)
  - Entries P(y | x) for fixed x, all y
  - Sums to 1

$P(W|cold)$

T	W	P
cold	sun	0.4
cold	rain	0.6

Review: Factor Zoo III

- Specified family: P(y | X)
  - Entries P(y | x) for fixed y, but for all x
  - Sums to … who knows!

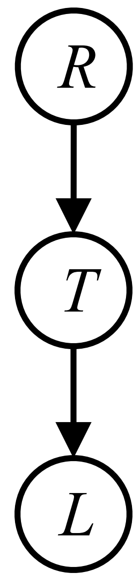
$P(rain|T)$

T	W	P	
hot	rain	0.2	$P(rain|hot)$
cold	rain	0.6	$P(rain|cold)$

- In general, when we write $P(Y_1 \ldots Y_N \mid X_1 \ldots X_M)$
  - It is a "factor," a multi-dimensional array
  - Its values are all $P(y_1 \ldots y_N \mid x_1 \ldots x_M)$
  - Any assigned X or Y is a dimension missing (selected) from the array

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

$P(+\ell|T)$

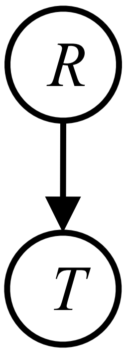
+t	+l	0.3
−t	+l	0.1

- 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



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

R,T

+r	0.1
−r	0.9

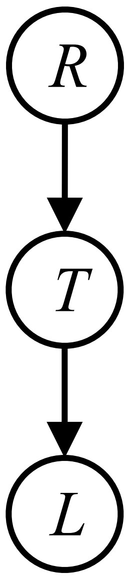
+r	+t	0.8
+r	−t	0.2
−r	+t	0.1
−r	−t	0.9

+r	+t	0.08
+r	−t	0.02
−r	+t	0.09
−r	−t	0.81

- Computation for each entry: pointwise products

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

Example: Multiple Joins



$P(R)$

+r	0.1
−r	0.9

$P(T|R)$

+r	+t	0.
+r	−t	0.
−r	+t	0.
−r	−t	0.

$P(L|T)$

+t	+l	0.
+t	−l	0.
−t	+l	0.
−t	−l	0.

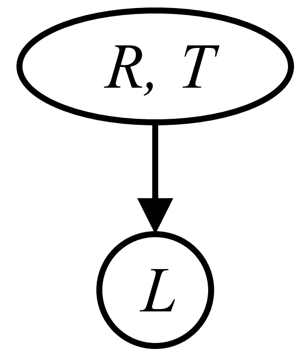
Join R

$P(R, T)$

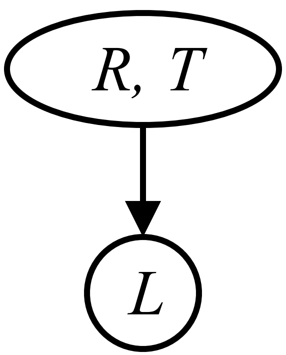
+r	+t	0.08
+r	−t	0.02
−r	+t	0.09
−r	−t	0.81

$P(L|T)$

+t	+l	0.
+t	−l	0.
−t	+l	0.
−t	−l	0.



Example: Multiple Joins



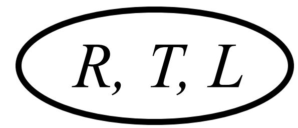
$P(R,T)$

+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

sum R $\longrightarrow$

$P(T)$

+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

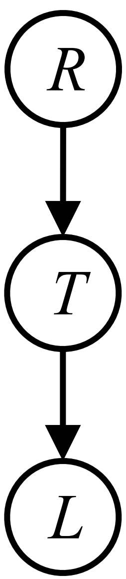
$P(L)$

+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

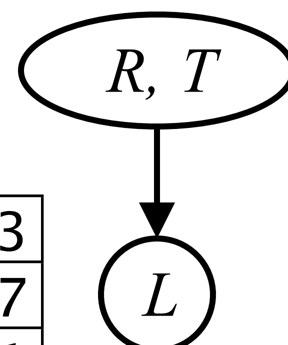
$P(L|T)$

+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



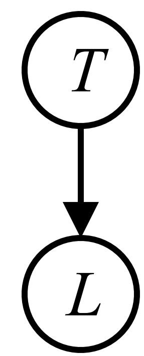
$P(L|T)$

+t	+l	0.3
+t	−l	0.7
−t	+l	0.1
−t	−l	0.9

Sum out R

$P(T)$

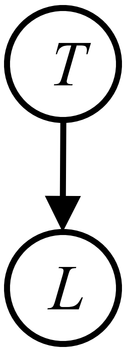
+t	0.17
−t	0.83



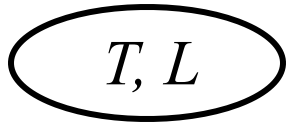
$P(L|T)$

+t	+l	0.3
+t	−l	0.7
−t	+l	0.1
−t	−l	0.9

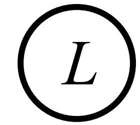
Marginalizing Early (aka VE*)



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 $\Rightarrow$

$P(L| + r)$

+l	0.26
-l	0.74

- To get our answer, just normalize this!

- That's it!

General Variable Elimination

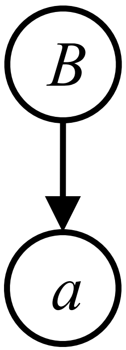
- Query: $P(Q|E_1 = e_1, \ldots E_k = e_k)$
- Start with initial factors:
  - Local CPTs (but instantiated by evidence)
- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H
- Join all remaining factors and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

Variable Elimination Bayes Rule

Start / Select

$P(B)$

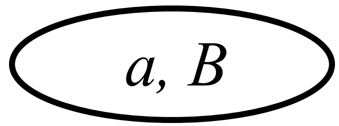
B	P
+b	0.1
¬b	0.9



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

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

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

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

Example

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

Choose E

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

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

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

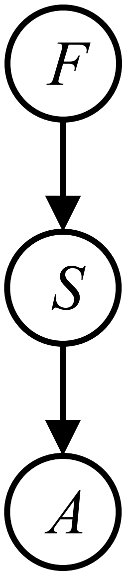
- Remaining Issues:
  - Complexity: exponential in tree width (size of the largest factor created)
  - Best elimination ordering? NP-hard problem

- 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

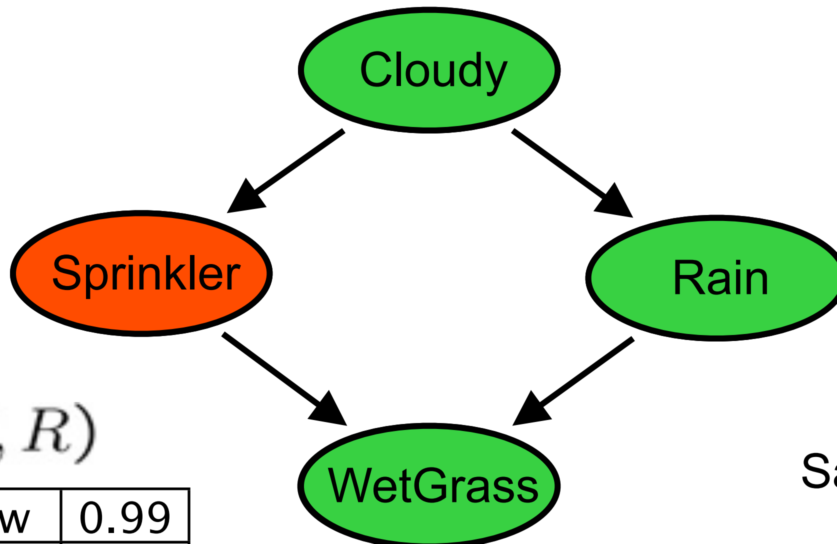
- Simulation has a name: sampling
- Sampling is a hot topic in machine learning, and it's really simple
- Basic idea:
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability
  - Show this converges to the true probability P
- Why sample?
  - Learning: get samples 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



$P(S|C)$

+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

-c, +s, -r, +w

…

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

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

- I.e., the sampling procedure is consistent

Example

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

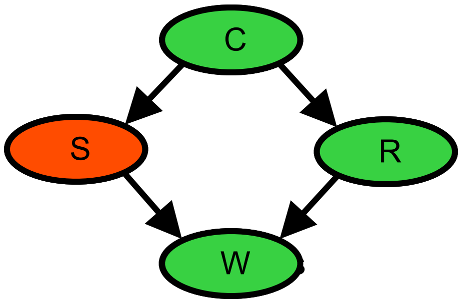
  +c, -s, +r, +w

  +c, +s, +r, +w

  -c, +s, +r, -w

  +c, -s, +r, +w

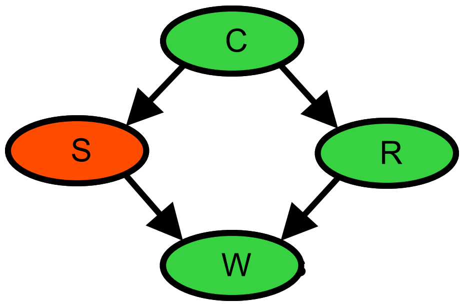
  -c, -s, -r, +w



- 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

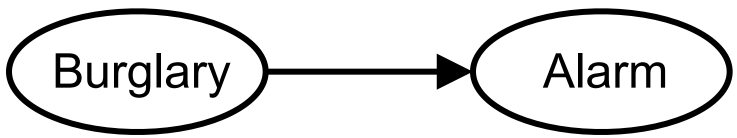


- Let's say we want P(C| +s)
  - Same thing: tally C outcomes, but ignore (reject) samples which don't have S=+s
  - This is called rejection sampling
  - It is also consistent for conditional probabilities (i.e., correct in the limit)

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

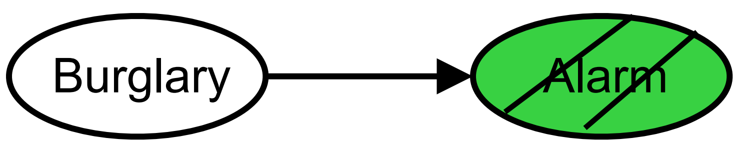
Likelihood Weighting

- 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

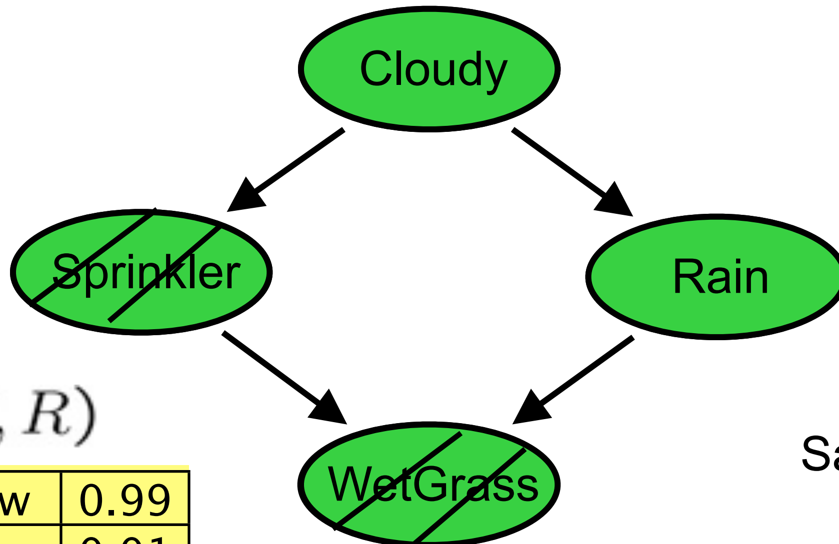
Likelihood Weighting

$P(C)$

+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

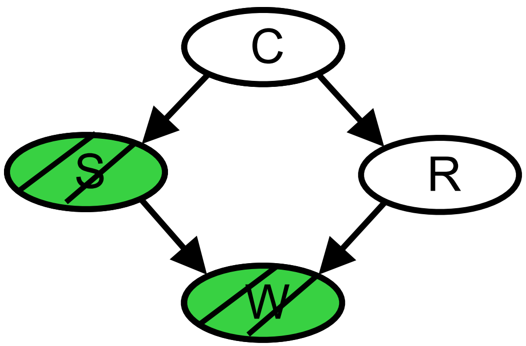
…

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

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

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

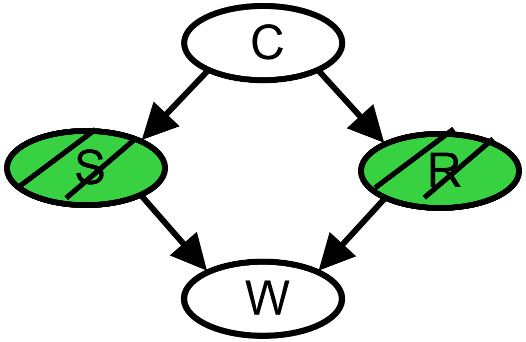
- Together, weighted sampling distribution is consistent

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

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

Likelihood Weighting

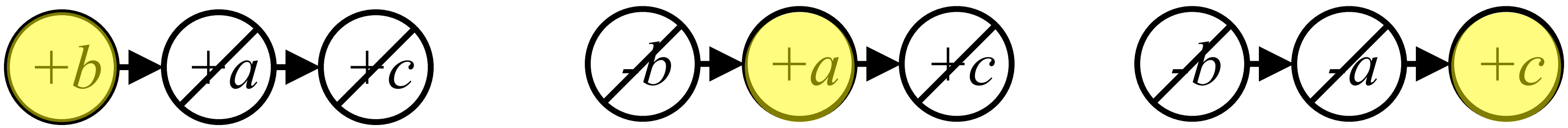
- Likelihood weighting is good
  - We have taken evidence into account as we generate the sample
  - E.g. here, W's value will get picked based on the evidence values of S, R
  - More of our samples will reflect the state of the world suggested by the evidence
- Likelihood weighting doesn't solve all 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.