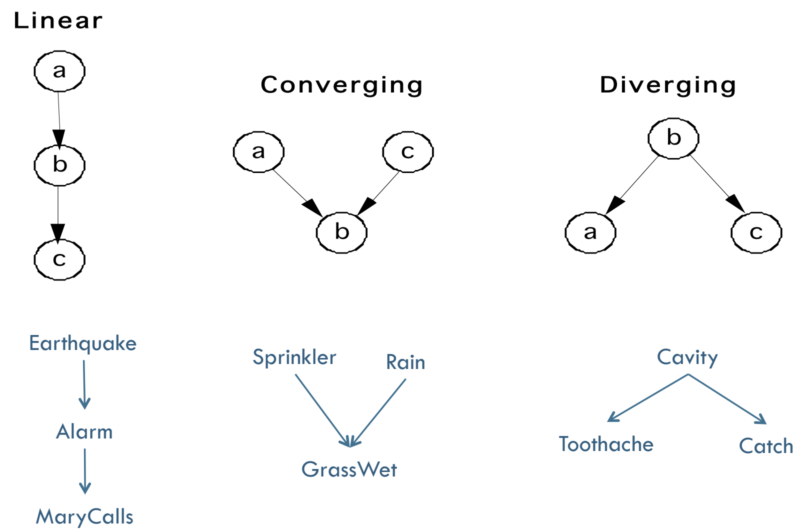
INFERENCE IN BAYESIAN NETWORKS

Today

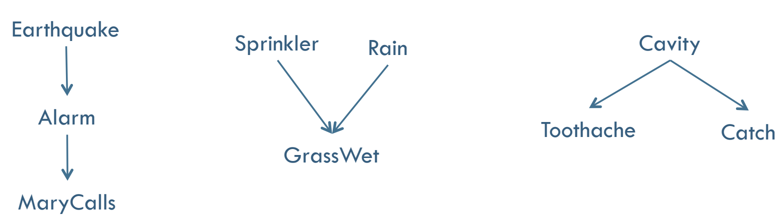
- Reading
  - AIMA 14.4 – 14.5
- Goals
  - Reading independencies
  - Exact inference
  - Approximate inference
  - Case Study: Latent Dirichlet Allocation

Three Types of Connections



Connection patterns and independence

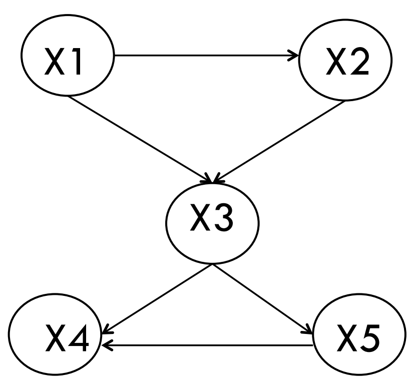
- **Linear connection:** The two end variables are dependent on each other. The middle variable renders them independent.
- **Converging connection:** The two end variables are independent of each other. The middle variable renders them dependent.
- **Divergent connection:** The two end variables are dependent on each other. The middle variable renders them independent.



D-Separation

- Algorithm to determine independencies in BN
- Query: Are two variables X_i and X_j independent?
- Check all paths between X_i and X_j
  - If all paths are blocked, then independent
  - If any path is not blocked then not independent

List the independencies in the following Bayesian Network



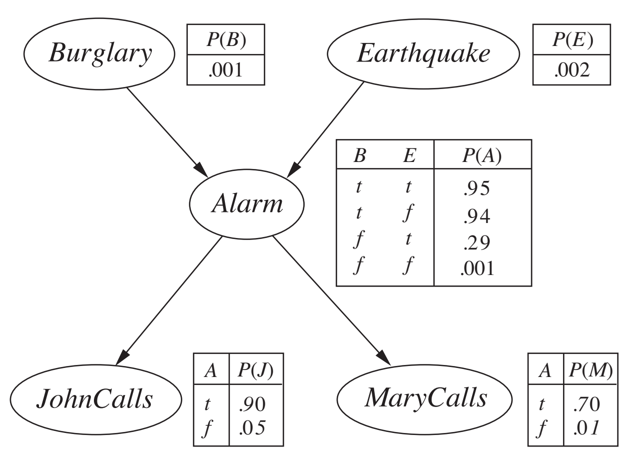
Inference in Bayesian Networks

- Probabilistic inference refers to the task of computing some desired probability given other known probabilities (evidence)

- Exact Inference
  - Enumeration
  - Variable elimination

- Approximate Inference
  - Direct sampling
  - Rejection sampling
  - Likelihood weighting
  - MCMC

Recall: Burglary network



Inference by Enumeration

~~Step One: select the entries in the table consistent with the evidence (this becomes our world)~~

Step Two: sum over the H variables to get the joint distribution of the query and evidence variables

Step Three: Normalize

$$p(b|j,m) \propto \sum_e \sum_a p(b,j,m,e,a)$$

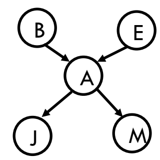
← Conditional and joint differ only by the normalizing constant

$$= \sum_e \sum_a p(b) \cdot p(e) \cdot p(j|a) \cdot p(m|a) \cdot p(a|b,e)$$

← Independencies read from BN

$$= p(b) \sum_e p(e) \sum_a p(j|a) \cdot p(m|a) \cdot p(a|b,e)$$

← Algebraic simplifications



- Compute p(b|j,m) and p(-b|j,m) and then normalize
- May compute the same expression more than once

Inference by Variable Elimination

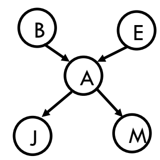
- Carry out sums from right to left storing intermediate results to avoid recomputation

$$p(B|j,m) = \alpha\, p(B) \sum_e p(e) \sum_a p(a|B,e)\; p(j|a)\; p(m|a)$$
$$= \alpha\, f_1(B) \sum_e f_2(e) \sum_a f_3(A,B,E)\; f_4(A)\; f_5(A)$$
$$= \alpha\, f_1(B) \sum_e f_2(e)\; f_6(B,E)$$
$$= \alpha\, f_1(B)\; f_7(B)$$

- Results are stored in factors (matrices)
- Two operations: pointwise multiplication and summation

Inference by Variable Elimination

- Every variable that is not an ancestor of a query variable or evidence variable is irrelevant
- Ordering of variables for summing out affects the time and space of VE
  - For polytrees (at most one path between any two nodes), VE is linear in the size of the network
  - In general, time and space are exponential



Approximate Inference

- Analogous to uninformed/informed search algorithms that use an incremental formulation
  - Direct sampling
  - Rejection sampling
  - Likelihood weighting
- Analogous to local search algorithms that use a complete-state formulation and make local modifications
  - Gibbs sampling (special case of MCMC methods)

Incremental formulation

- Basic Idea:
  - Draw S samples from a distribution of interest P
  - Compute the approximate probability
  - (Show this converges to the true probability P)

[T, T, F, T]
[F, F, F, F]
[F, T, F, T]
[F, F, T, T]
[T, F, F, F]
[T, T, F, T]
[F, T, F, T]
[T, F, F, F]
[F, T, T, F]
[T, T, F, F]

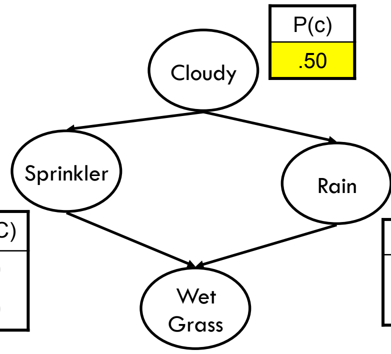
S samples generated using stochastic simulation

$p(X_1 = T) \approx 5/10$

$p(X_2 = F \mid X_3 = F) \approx 3/10$

Approximations become exact as S approaches infinity

Direct Sampling: no evidence



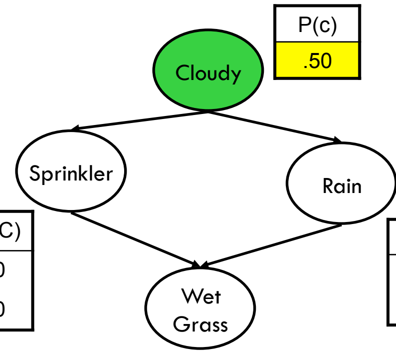
P(c)
.50

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

Direct Sampling: no evidence



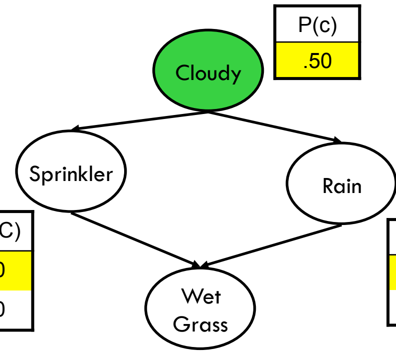
P(c)
.50

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

Direct Sampling: no evidence



P(c)
.50

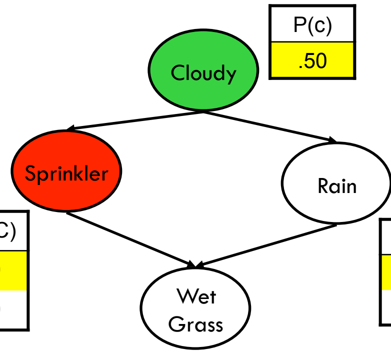
C	P(s|C)
T	.10
F	.50

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

Direct Sampling: no evidence

P(c)
.50



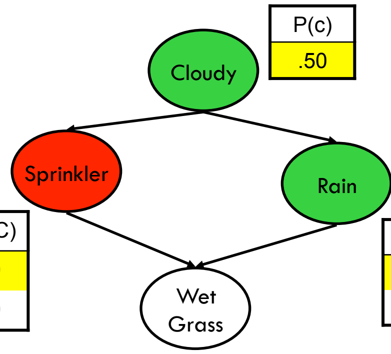
C	P(s|C)
T	.10
F	.50

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

Direct Sampling: no evidence

P(c)
.50

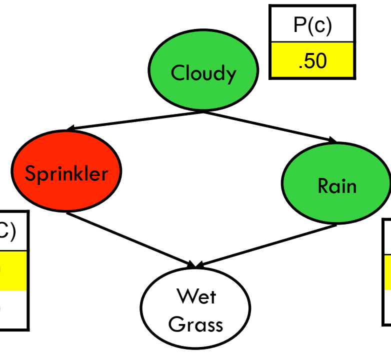


C	P(s|C)
T	.10
F	.50

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

Direct Sampling: no evidence



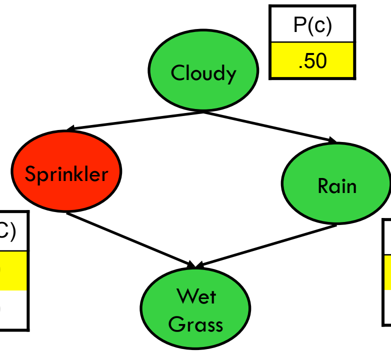
P(c)
.50

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

Direct Sampling: no evidence



P(c)
.50

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

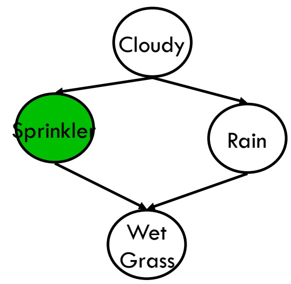
C	P(r|C)
T	.80
F	.20

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

Sample: [T, F, T, T]

Rejection Sampling: evidence

- Perform direct sampling
- "Reject", i.e. remove, any samples that are inconsistent with the evidence



[C, S, R, W]	
[T, T, F, T]	[T, T, F, T]
[F, F, F, F]	~~[F, F, F, F]~~
[F, T, F, T]	[F, T, F, T]
[F, F, T, T]	~~[F, F, T, T]~~
[T, F, F, F]	~~[T, F, F, F]~~
[T, T, F, T]	[T, T, F, T]
[F, T, F, T]	[F, T, F, T]
[T, F, F, F]	~~[T, F, F, F]~~
[F, T, T, F]	[F, T, T, F]
[T, T, F, F]	[T, T, F, F]

p(R | S = true)
p(R = true | S = true) ≈ 1/6
p(R = false | S = true) ≈ 5/6

Likelihood weighting

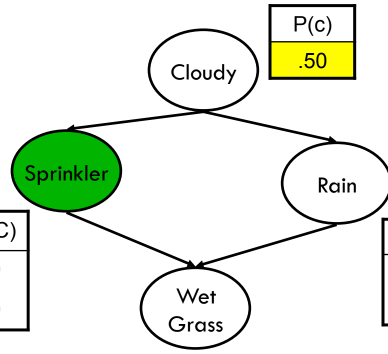
- Fixes the values for the evidence so there are no wasted samples
- Sample only the non-evidence variables
- Not every sample is created equal
  - Need to weight each sample by how likely the evidence is given the sampled values
  - Compute the product of the conditional distribution of the evidence given the sampled values of its parents

$$\text{weight} = p(e_1 \mid \text{Parents}(e_1)) * p(e_2 \mid \text{Parents}(e_2)) \ldots$$

Likelihood weighting

weight = 1

P(c)
.50



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

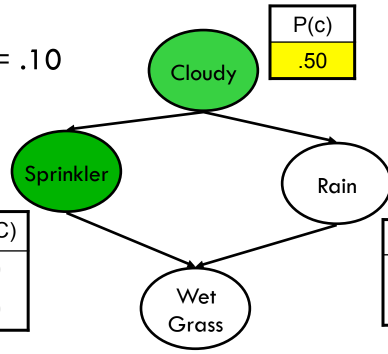
C	P(r|C)
T	.80
F	.20

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

Likelihood weighting

weight = p(s | c) = .10

P(c)
.50



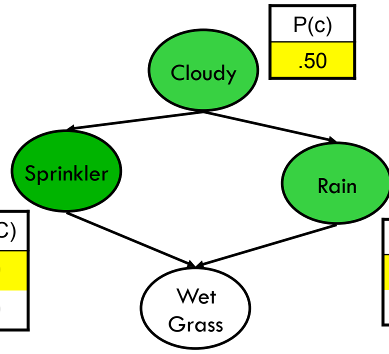
C	P(s|C)
T	.10
F	.50

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

Likelihood weighting

weight = .10



P(c)
.50

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

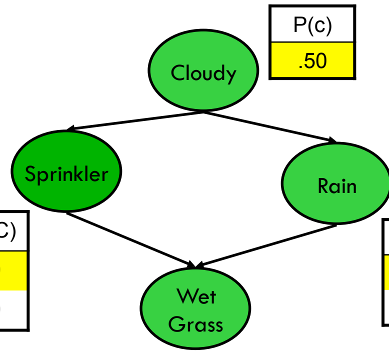
C	P(r|C)
T	.80
F	.20

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

Likelihood weighting

weight = .10

Sample: [T, T, T, T]



P(c)
.50

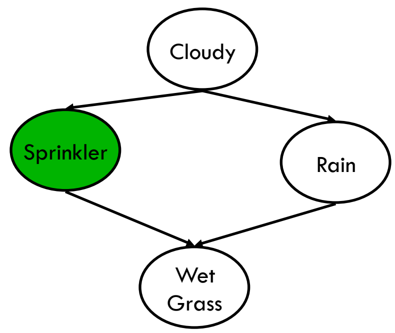
C	P(s|C)
T	.10
F	.50

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

Likelihood weighting

Sample [C,S,R,W]	Weight
[T, T, F, T]	p(s|c) = .10
[F, T, F, T]	p(s|-c) = .50
[T, T, F, T]	p(s|c) = .10
[F, T, T, F]	p(s|-c) = .50
[T, T, T, T]	p(s|c) = .10
[F, T, F, T]	p(s|-c) = .50



- Estimate probability of query using a weighted average

Gibbs Sampling

- Analogous to a local search algorithm where we make local modifications to our current state
  - Initial state = random assignment of non-evidence variables
  - States = complete assignment of values to variables
  - Transition = sample a new value for each variable in turn

Draw state space for WetGrass example on board

Gibbs Sampling

- Analogous to a local search algorithm where we make local modifications to our current state
  - Initial state = random assignment of non-evidence variables
  - States = complete assignment of values to variables
  - Transition = sample a new value for each variable in turn
- Each step to a new state is recorded as a sample
- In the limit, the probability of being in a state is proportional to that state's posterior probability

Gibbs Sampling

- Gibbs sampling is an instance of a more general class of algorithms known as Markov Chain Monte Carlo (MCMC) algorithms
  - Note the use of the phrase "Markov chain" which we saw an example of earlier
- Other methods you might hear mentioned
  - Metropolis-Hastings (a generalization of Gibbs sampling)
  - Variational method
  - Belief propagation