Approximate Inference in Bayesian Networks

Outline

I. Direct sampling methods

II. Rejection sampling

III. Importance sampling

* Figures are either from the textbook site.

Approximate Inference Methods

- These methods, also called Monte Carlo algorithms, are based on the principle of stochastic sampling.

Approximate Inference Methods

- These methods, also called Monte Carlo algorithms, are based on the principle of stochastic sampling.
  - Generate random events based on the probabilities in the Bayes net.
  - Count up different answers found in these events.
  - Monte Carlo algorithms are used to estimate quantities that are difficult to calculate exactly.

Approximate Inference Methods

- These methods, also called Monte Carlo algorithms, are based on the principle of stochastic sampling.
  - Generate random events based on the probabilities in the Bayes net.
  - Count up different answers found in these events.
  - Monte Carlo algorithms are used to estimate quantities that are difficult to calculate exactly.
- Accuracy depends on the size of the sample set.
  - Can get arbitrarily close to the true probability distribution as the size increases.

Approximate Inference Methods

- These methods, also called Monte Carlo algorithms, are based on the principle of stochastic sampling.
  - Generate random events based on the probabilities in the Bayes net.
  - Count up different answers found in these events.
  - Monte Carlo algorithms are used to estimate quantities that are difficult to calculate exactly.
- Accuracy depends on the size of the sample set.
  - Can get arbitrarily close to the true probability distribution as the size increases.
- Two families of algorithms: direct sampling and Markov chain sampling.

I. Direct Sampling

- Sample variables in topological order.

  The value of a node is *conditioned* on those already assigned to its parents (the assignments are *guaranteed* by the topological order).

I. Direct Sampling

- Sample variables in topological order.

The value of a node is *conditioned* on those already assigned to its parents (the assignments are *guaranteed* by the topological order).

function PRIOR-SAMPLE(bn) **returns** an event sampled from the prior specified by bn
inputs: bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

$\mathbf{x} \leftarrow$ an event with n elements
for each variable X_i **in** $X_1, \ldots, X_n$ **do**
$\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
return x

I. Direct Sampling

- Sample variables in topological order.

The value of a node is *conditioned* on those already assigned to its parents (the assignments are *guaranteed* by the topological order).

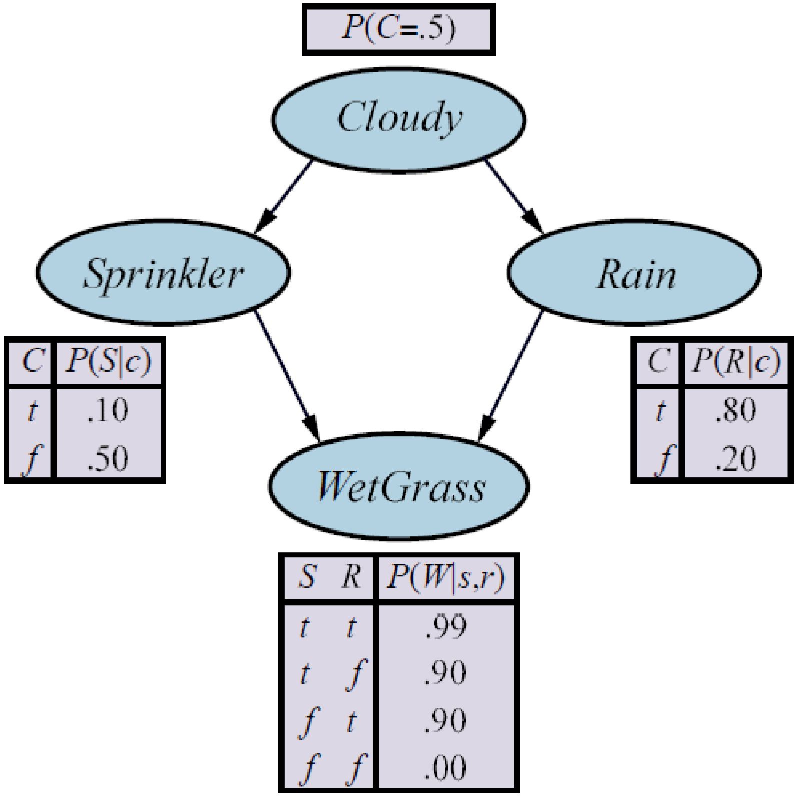
function PRIOR-SAMPLE(bn) **returns** an event sampled from the prior specified by bn
  inputs: bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $\mathbf{x} \leftarrow$ an event with n elements
  for each variable X_i **in** $X_1, \ldots, X_n$ **do**
    $\mathbf{x}[i] \leftarrow$ a $\boxed{\text{random sample}}$ from $\mathbf{P}(X_i \mid parents(X_i))$
  return x

from the domain of
X_i, e.g., true or false

The Sprinkler Network

- Every morning Mary checks the weather.
- If it’s cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.



$P(C{=}.5)$

C	$P(S \mid c)$
t	.10
f	.50

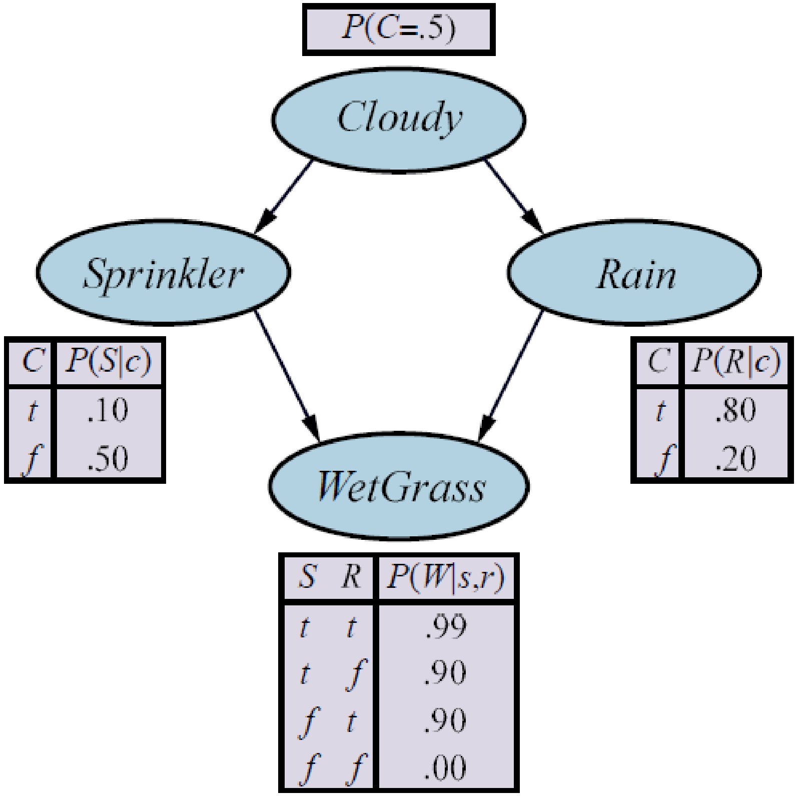
C	$P(R \mid c)$
t	.80
f	.20

S	R	$P(W \mid s,r)$
t	t	.99
t	f	.90
f	t	.90
f	f	.00

The Sprinkler Network

- Every morning Mary checks the weather.
- If it’s cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.

Order: *Cloudy*, *Sprinkler*, *Rain*, *WetGrass*

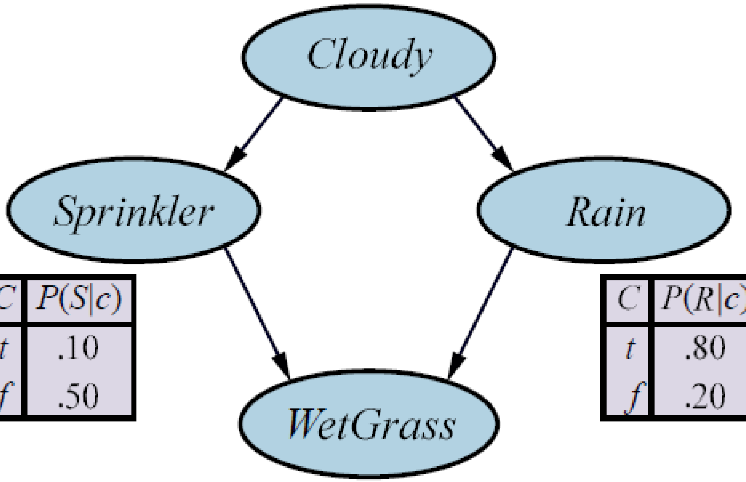


The Sprinkler Network

- Every morning Mary checks the weather.
- If it's cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.

Order: *Cloudy*, *Sprinkler*, *Rain*, *WetGrass*

1. Value *true* sampled from $\mathbf{P}(Cloudy) = \langle 0.5, 0.5 \rangle$.



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	.00

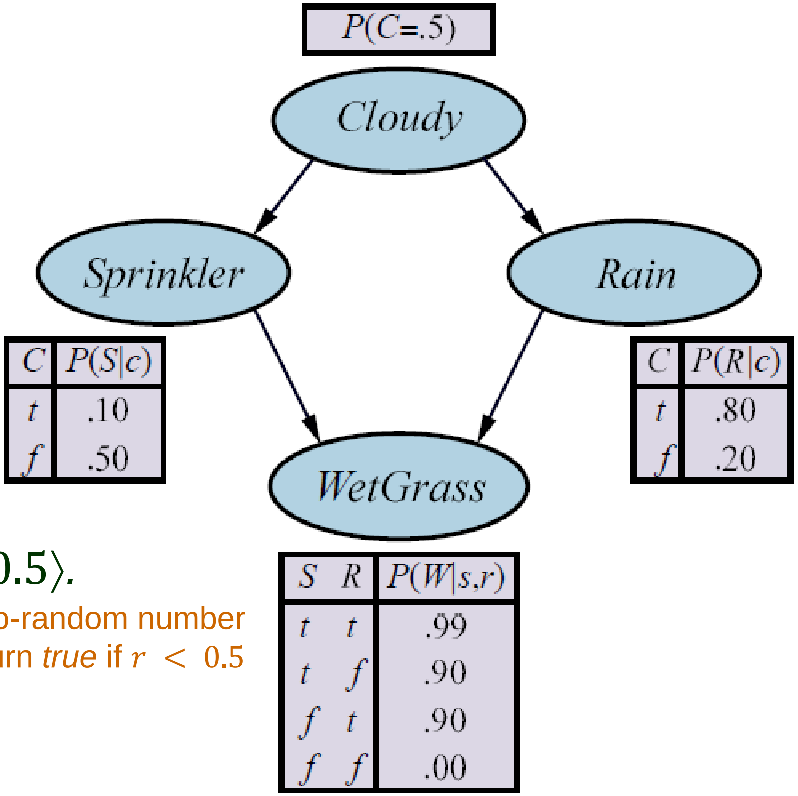
The Sprinkler Network

- Every morning Mary checks the weather.
- If it's cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.

Order: *Cloudy*, *Sprinkler*, *Rain*, *WetGrass*

1. Value *true* sampled from $\mathbf{P}(Cloudy) = \langle 0.5, 0.5 \rangle$.

How? Generate a pseudo-random number r in the range [0, 1]. Return *true* if $r < 0.5$ and *false* otherwise.



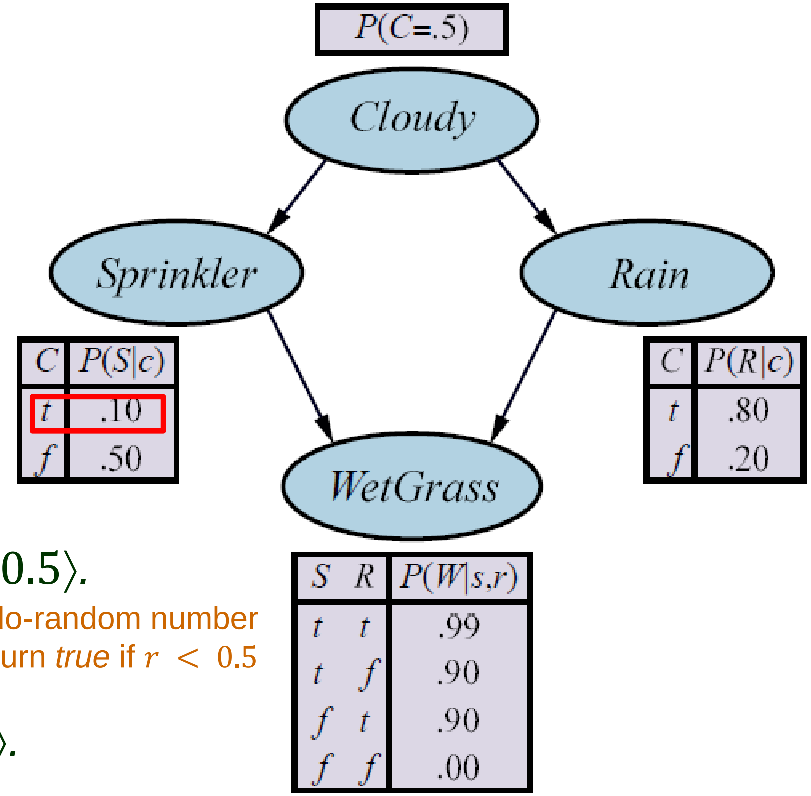
C	$P(S\|c)$
t	.10
f	.50

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	.00

The Sprinkler Network

- Every morning Mary checks the weather.
- If it's cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.



Order: *Cloudy*, *Sprinkler*, *Rain*, *WetGrass*

1. Value *true* sampled from $\boldsymbol{P}(\mathit{Cloudy}) = \langle 0.5, 0.5 \rangle$.

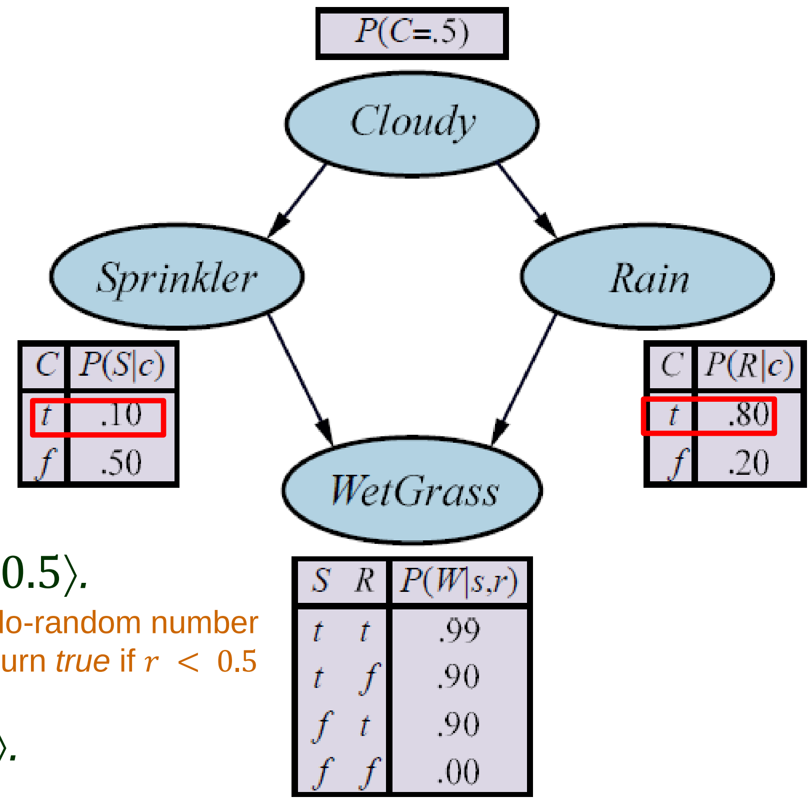
   How? Generate a pseudo-random number r in the range [0, 1]. Return *true* if $r < 0.5$ and *false* otherwise.

2. Value *false* sampled from

   $\boldsymbol{P}(\mathit{Sprinkler} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.1, 0.9 \rangle$.

The Sprinkler Network

- Every morning Mary checks the weather.
- If it's cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.



Order: *Cloudy*, *Sprinkler*, *Rain*, *WetGrass*

1. Value *true* sampled from $\boldsymbol{P}(\mathit{Cloudy}) = \langle 0.5, 0.5 \rangle$.

   How? Generate a pseudo-random number r in the range [0, 1]. Return *true* if $r < 0.5$ and *false* otherwise.

2. Value *false* sampled from

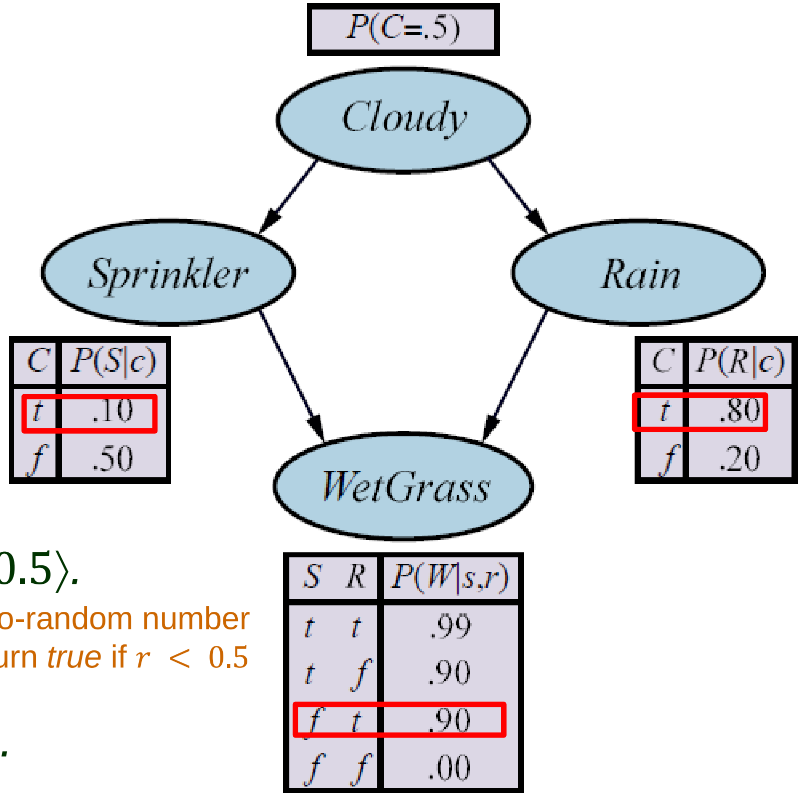
   $\boldsymbol{P}(\mathit{Sprinkler} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.1, 0.9 \rangle$.

3. Value *true* sampled from

   $\boldsymbol{P}(\mathit{Rain} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.8, 0.2 \rangle$.

The Sprinkler Network

- Every morning Mary checks the weather.
- If it's cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.



Order: *Cloudy*, *Sprinkler*, *Rain*, *WetGrass*

1. Value *true* sampled from $\mathbf{P}(Cloudy) = \langle 0.5, 0.5 \rangle$.

   How? Generate a pseudo-random number r in the range [0, 1]. Return *true* if $r < 0.5$ and *false* otherwise.

2. Value *false* sampled from

   $$\mathbf{P}(Sprinkler \mid Cloudy = true) = \langle 0.1, 0.9 \rangle.$$

3. Value *true* sampled from

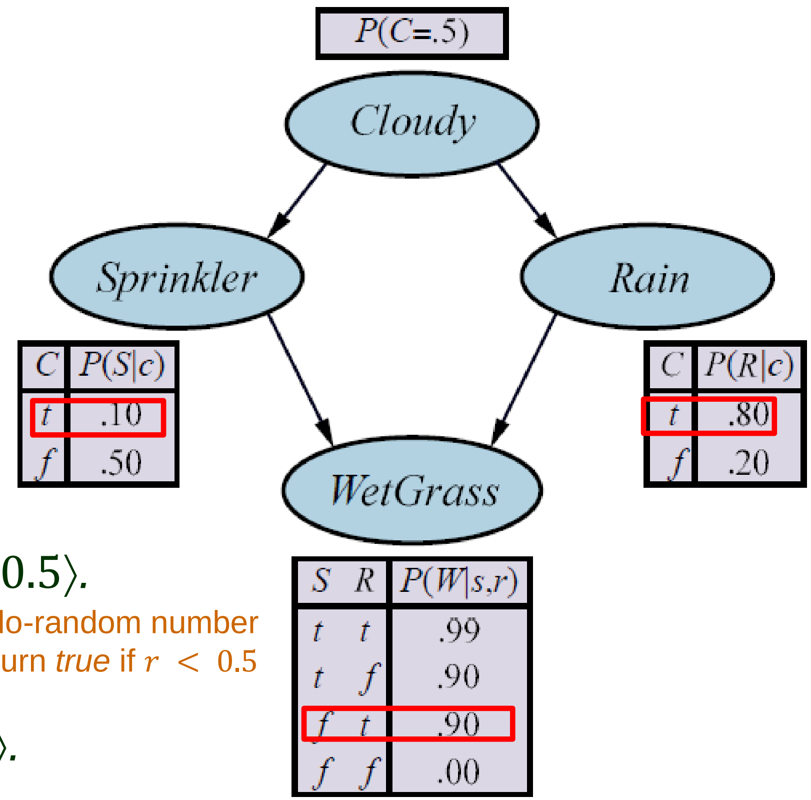
   $$\mathbf{P}(Rain \mid Cloudy = true) = \langle 0.8, 0.2 \rangle.$$

4. Value *true* sampled from

   $$\mathbf{P}(WetGrass \mid Sprinkler = false, Rain = true) = \langle 0.9, 0.1 \rangle.$$

The Sprinkler Network

- Every morning Mary checks the weather.
- If it's cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.



Order: *Cloudy*, *Sprinkler*, *Rain*, *WetGrass*

1. Value *true* sampled from $\boldsymbol{P}(Cloudy) = \langle 0.5, 0.5 \rangle$.

   How? Generate a pseudo-random number r in the range [0, 1]. Return *true* if $r < 0.5$ and *false* otherwise.

2. Value *false* sampled from

   $\boldsymbol{P}(Sprinkler \mid Cloudy = true) = \langle 0.1, 0.9 \rangle$.

3. Value *true* sampled from

   $\boldsymbol{P}(Rain \mid Cloudy = true) = \langle 0.8, 0.2 \rangle$.

4. Value *true* sampled from

   $\boldsymbol{P}(WetGrass \mid Sprinkler = false, Rain = true) = \langle 0.9, 0.1 \rangle$.

PRIOR-SAMPLE() returns the event [*true*, *false*, *true*, *true*].

Sampling Process

$S_{PS}(x_1, \ldots, x_n)$: probability of a specific event generated by PRIOR-SAMPLE.

Sampling Process

$S_{PS}(x_1, \ldots, x_n)$: probability of a specific event generated by PRIOR-SAMPLE.

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

Sampling Process

$S_{PS}(x_1, \ldots, x_n)$: probability of a specific event generated by PRIOR-SAMPLE.

$$\boxed{S_{PS}(x_1, \ldots, x_n)} = \prod_{i=1}^{n} P(x_i \mid \mathit{Parents}(X_i))$$

$$\boxed{= P(x_1, \ldots, x_n)}$$ (probability of joint distribution)

Sampling Process

$S_{PS}(x_1, \ldots, x_n)$: probability of a specific event generated by PRIOR-SAMPLE.

$$\boxed{S_{PS}(x_1, \ldots, x_n)} = \prod_{i=1}^{n} P(x_i \mid \mathit{Parents}(X_i))$$

$$\boxed{= P(x_1, \ldots, x_n)}$$ (probability of joint distribution)

- N total samples produced by PRIOR-SAMPLE.

Sampling Process

$S_{PS}(x_1, \ldots, x_n)$: probability of a specific event generated by PRIOR-SAMPLE.

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

$$= P(x_1, \ldots, x_n)$$ (probability of joint distribution)

- N total samples produced by PRIOR-SAMPLE.
- $N_{ps}(x_1, \ldots, x_n)$ times that the specific event $X_1 = x_1 \wedge \cdots \wedge X_n = x_n$ occurs.

Sampling Process

$S_{PS}(x_1, \ldots, x_n)$: probability of a specific event generated by PRIOR-SAMPLE.

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

$$\boxed{= P(x_1, \ldots, x_n)}$$ (probability of joint distribution)

- N total samples produced by PRIOR-SAMPLE.
- $N_{ps}(x_1, \ldots, x_n)$ times that the specific event $X_1 = x_1 \wedge \cdots \wedge X_n = x_n$ occurs.

$$\lim_{N\to\infty} \frac{N_{ps}(x_1, \ldots, x_n)}{N} = S_{PS}(x_1, \ldots, x_n) = P(x_1, \ldots, x_n)$$

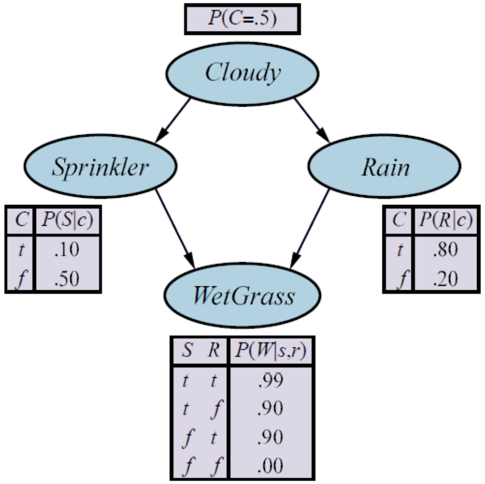
Sampling Process

$S_{PS}(x_1, \ldots, x_n)$: probability of a specific event generated by PRIOR-SAMPLE.

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

$$= P(x_1, \ldots, x_n)$$ (probability of joint distribution)

- N total samples produced by PRIOR-SAMPLE.
- $N_{ps}(x_1, \ldots, x_n)$ times that the specific event $X_1 = x_1 \wedge \cdots \wedge X_n = x_n$ occurs.



$$\lim_{N \to \infty} \frac{N_{ps}(x_1, \ldots, x_n)}{N} = S_{PS}(x_1, \ldots, x_n) = P(x_1, \ldots, x_n)$$

$$S_{PS}(true, false, true, true) = 0.5 \cdot 0.9 \cdot 0.8 \cdot 0.9 = 0.324$$

Partially Specified Event

Estimate the probability of the partial event $X_1 = x_1 \wedge \cdots \wedge X_m = x_m$, $m \le n$:

$$P(x_1, \dots, x_m) \approx \frac{N_{ps}(x_1, \dots, x_m)}{N} \equiv \hat{P}(x_1, \dots, x_m)$$

the fraction of all sampled complete events
that match the partially specified event

Partially Specified Event

Estimate the probability of the partial event $X_1 = x_1 \wedge \cdots \wedge X_m = x_m$, $m \leq n$:

$$P(x_1, \ldots, x_m) \approx \frac{N_{ps}(x_1, \ldots, x_m)}{N} \equiv \hat{P}(x_1, \ldots, x_m)$$

the fraction of all sampled complete events
that match the partially specified event

Example *Rain* = *true* holds for 511 of 1,000 samples generated from the sprinkler network.

$$\hat{P}(\mathit{Rain} = \mathit{true}) = 0.511$$

II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $\boldsymbol{P}(X \mid \boldsymbol{e})$ as follows:

II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $\boldsymbol{P}(X \mid \boldsymbol{e})$ as follows:

- Generate samples from the prior distribution specified by the BN.

II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $\boldsymbol{P}(X \mid \boldsymbol{e})$ as follows:

- Generate samples from the prior distribution specified by the BN.
- Rejects all those that do not match the evidence $\boldsymbol{e}$.

$N_{PS}(\boldsymbol{e})$: the number of samples that are not rejected.

II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $\boldsymbol{P}(X \mid \boldsymbol{e})$ as follows:

- Generate samples from the prior distribution specified by the BN.
- Rejects all those that do not match the evidence $\boldsymbol{e}$.

  $N_{PS}(\boldsymbol{e})$: the number of samples that are not rejected.

- Count how often $X = x$ occurs in the remaining $N_{PS}(\boldsymbol{e})$ samples for every value x of X.

II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $\boldsymbol{P}(X \mid \boldsymbol{e})$ as follows:

- Generate samples from the prior distribution specified by the BN.
- Rejects all those that do not match the evidence $\boldsymbol{e}$.

  $N_{PS}(\boldsymbol{e})$: the number of samples that are not rejected.

- Count how often $X = x$ occurs in the remaining $N_{PS}(\boldsymbol{e})$ samples for every value x of X.

  $\boldsymbol{N}_{PS}(X, \boldsymbol{e})$: the vector of counts, one for each value x of X, of samples that contain $X = x$ and also agree with the evidence $\boldsymbol{e}$.

II. Rejection Sampling (RS)

Generate samples from a hard-to-sample distribution given an easy-to-sample distribution.

Estimate $\boldsymbol{P}(X \mid \boldsymbol{e})$ as follows:

- Generate samples from the prior distribution specified by the BN.
- Rejects all those that do not match the evidence $\boldsymbol{e}$.

  $N_{PS}(\boldsymbol{e})$: the number of samples that are not rejected.

- Count how often $X = x$ occurs in the remaining $N_{PS}(\boldsymbol{e})$ samples for every value x of X.

  $\boldsymbol{N}_{PS}(X, \boldsymbol{e})$: the vector of counts, one for each value x of X, of samples that contain $X = x$ and also agree with the evidence $\boldsymbol{e}$.

$$\widehat{\boldsymbol{P}}(X \mid \boldsymbol{e}) = \alpha \boldsymbol{N}_{PS}(X, \boldsymbol{e}) = \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})}{N_{PS}(\boldsymbol{e})}$$

Estimation of True Probability by RS

$$\widehat{\boldsymbol{P}}(X \mid \boldsymbol{e}) = \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})}{N_{PS}(\boldsymbol{e})}$$

Estimation of True Probability by RS

$$\widehat{\boldsymbol{P}}(X \mid \boldsymbol{e}) = \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})}{N_{PS}(\boldsymbol{e})}$$

$$= \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})/N}{N_{PS}(\boldsymbol{e})/N}$$

Estimation of True Probability by RS

$$\widehat{\boldsymbol{P}}(X \mid \boldsymbol{e}) = \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})}{N_{PS}(\boldsymbol{e})}$$

$$= \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})/N}{N_{PS}(\boldsymbol{e})/N}$$

$$\approx \frac{\boldsymbol{P}(X, \boldsymbol{e})}{\boldsymbol{P}(\boldsymbol{e})}$$

since $P(x_1, \ldots, x_m) \approx \frac{N_{ps}(x_1, \ldots, x_m)}{N}$

Estimation of True Probability by RS

$$\widehat{\boldsymbol{P}}(X \mid \boldsymbol{e}) = \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})}{N_{PS}(\boldsymbol{e}\,)}$$

$$= \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})/N}{N_{PS}(\boldsymbol{e}\,)/N}$$

$$\approx \frac{\boldsymbol{P}(X, \boldsymbol{e})}{\boldsymbol{P}(\boldsymbol{e})}$$

since $P(x_1, \dots, x_m) \approx \frac{N_{ps}(x_1,\dots,x_m)}{N}$

$$\boxed{\widehat{\boldsymbol{P}}(X \mid \boldsymbol{e}) \approx \boldsymbol{P}(X \mid \boldsymbol{e})}$$

Estimation of True Probability by RS

$$\hat{\boldsymbol{P}}(X \mid \boldsymbol{e}) = \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})}{N_{PS}(\boldsymbol{e}\,)}$$

$$= \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})/N}{N_{PS}(\boldsymbol{e}\,)/N}$$

$$\approx \frac{\boldsymbol{P}(X, \boldsymbol{e})}{\boldsymbol{P}(\boldsymbol{e})}$$

since $P(x_1, \ldots, x_m) \approx \frac{N_{ps}(x_1, \ldots, x_m)}{N}$

Standard deviation $1/\sqrt{N}$
of error in each probability

$$\boxed{\hat{\boldsymbol{P}}(X \mid \boldsymbol{e}) \approx \boldsymbol{P}(X \mid \boldsymbol{e})}$$

Estimation of True Probability by RS

$$\hat{\mathbf{P}}(X \mid \mathbf{e}) = \frac{\mathbf{N}_{PS}(X, \mathbf{e})}{N_{PS}(\mathbf{e})}$$

$$= \frac{\mathbf{N}_{PS}(X, \mathbf{e})/N}{N_{PS}(\mathbf{e})/N}$$

Standard deviation $1/\sqrt{N}$
of error in each probability

$$\approx \frac{\mathbf{P}(X, \mathbf{e})}{\mathbf{P}(\mathbf{e})}$$

since $P(x_1, \ldots, x_m) \approx \frac{N_{ps}(x_1, \ldots, x_m)}{N}$

$$\boxed{\hat{\mathbf{P}}(X \mid \mathbf{e}) \approx \mathbf{P}(X \mid \mathbf{e})}$$

Example Estimate $\mathbf{P}(\mathit{Rain} \mid \mathit{Sprinkler} = \mathit{true})$ using 100 samples.

Estimation of True Probability by RS

$$\hat{\boldsymbol{P}}(X \mid \boldsymbol{e}) = \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})}{N_{PS}(\boldsymbol{e}\ )}$$

$$= \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})/N}{N_{PS}(\boldsymbol{e}\ )/N}$$

Standard deviation $1/\sqrt{N}$
of error in each probability

$$\approx \frac{\boldsymbol{P}(X, \boldsymbol{e})}{\boldsymbol{P}(\boldsymbol{e})}$$

since $P(x_1, \ldots, x_m) \approx \frac{N_{ps}(x_1, \ldots, x_m)}{N}$

$$\boxed{\hat{\boldsymbol{P}}(X \mid \boldsymbol{e}) \approx \boldsymbol{P}(X \mid \boldsymbol{e})}$$

Example Estimate $\boldsymbol{P}(Rain \mid Sprinkler = true)$ using 100 samples.

- 73 samples have *Sprinkler* = *false* and are rejected.
- Of the 27 samples with *Sprinkler* = *true*, only 8 have *Rain* = *true*.

Estimation of True Probability by RS

$$\hat{\boldsymbol{P}}(X \mid \boldsymbol{e}) = \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})}{N_{PS}(\boldsymbol{e}\,)}$$

$$= \frac{\boldsymbol{N}_{PS}(X, \boldsymbol{e})/N}{N_{PS}(\boldsymbol{e}\,)/N}$$

Standard deviation $1/\sqrt{N}$
of error in each probability

$$\approx \frac{\boldsymbol{P}(X, \boldsymbol{e})}{\boldsymbol{P}(\boldsymbol{e})}$$

since $P(x_1, \ldots, x_m) \approx \frac{N_{ps}(x_1,\ldots,x_m)}{N}$

$$\hat{\boldsymbol{P}}(X \mid \boldsymbol{e}) \approx \boldsymbol{P}(X \mid \boldsymbol{e})$$

Example Estimate $\boldsymbol{P}(Rain \mid Sprinkler = true)$ using 100 samples.

- 73 samples have $Sprinkler = false$ and are rejected.
- Of the 27 samples with $Sprinkler = true$, only 8 have $Rain = true$.

$$\boldsymbol{P}(Rain \mid Sprinkler = true) \approx \alpha\langle 8{,}19\rangle = (0.296{,}0.704)$$

How Fast Does RS Converge?

- How many samples are needed before the resulting estimates are close to the correct answers with high probability?

- The complexity of rejection sampling depends primarily on the $\underbrace{\text{fraction of samples}}_{=\ \text{prior probability of the evidence}\ P(\boldsymbol{e})}$ that are accepted.

How Fast Does RS Converge?

- How many samples are needed before the resulting estimates are close to the correct answers with high probability?

- The complexity of rejection sampling depends primarily on the $\underbrace{\text{fraction of samples}}_{=\ \text{prior probability of the evidence } P(\boldsymbol{e})}$ that are accepted.

- $P(\boldsymbol{e})$ is vanishingly small for complex networks with many evidence variables.
  - The fraction of samples consistent with $\boldsymbol{e}$ drops exponentially as the number of evidence variables grows.
  - Rejection sampling is unusable for complex problems.

III. Importance Sampling (IS)

- Emulate the effect of sampling from one distribution P using samples from another distribution Q.
  - It is too hard to sample from the true posterior distribution on all the evidence.
  - So we sample from an easy distribution.
- To ensure correctness in the limit, we use a correction factor $P(\boldsymbol{x})/Q(\boldsymbol{x})$, to each sample $\boldsymbol{x}$ when it is counted.

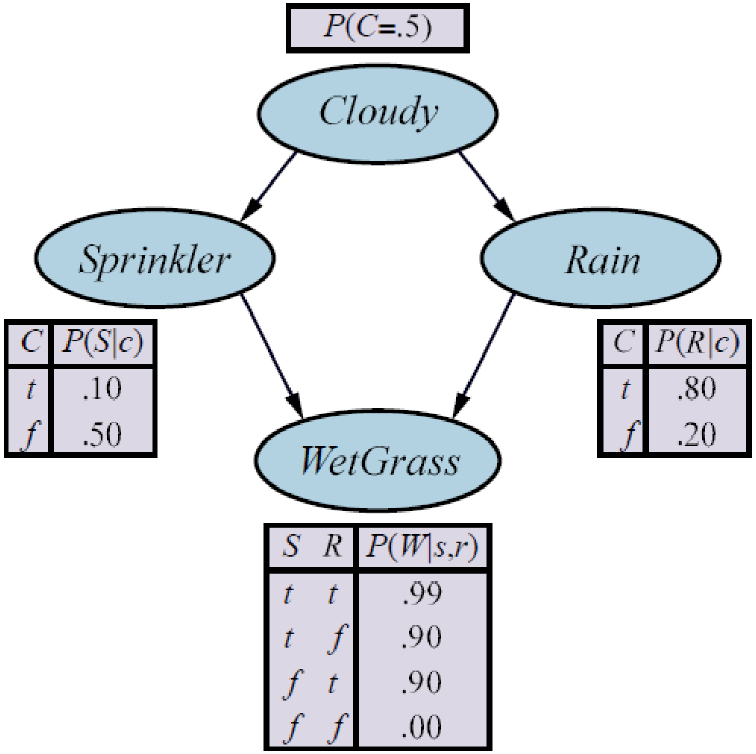
III. Importance Sampling (IS)

- Emulate the effect of sampling from one distribution P using samples from another distribution Q.
  - It is too hard to sample from the true posterior distribution on all the evidence.
  - So we sample from an easy distribution.

- To ensure correctness in the limit, we use a correction factor $P(\boldsymbol{x})/Q(\boldsymbol{x})$, to each sample $\boldsymbol{x}$ when it is counted.

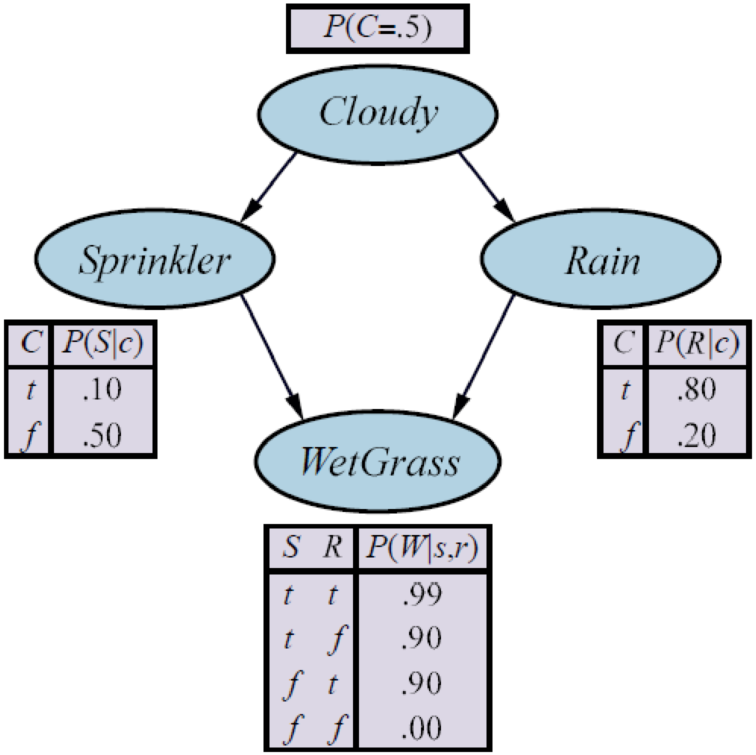
How does this work?

Weight of a Sample



Query $\mathbf{P}(Rain \mid Cloudy = true, WetGrass = true)$

Weight of a Sample

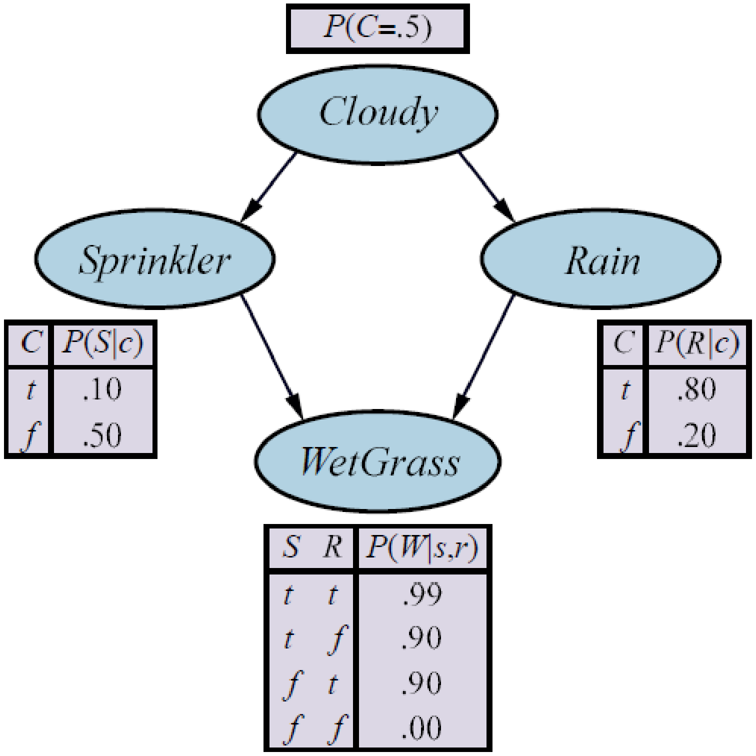


Query $\mathbf{P}(Rain \mid Cloudy = true, WetGrass = true)$

- Pick a topological order:

Cloudy, *Sprinkler*, *Rain*, *WetGrass*

Weight of a Sample



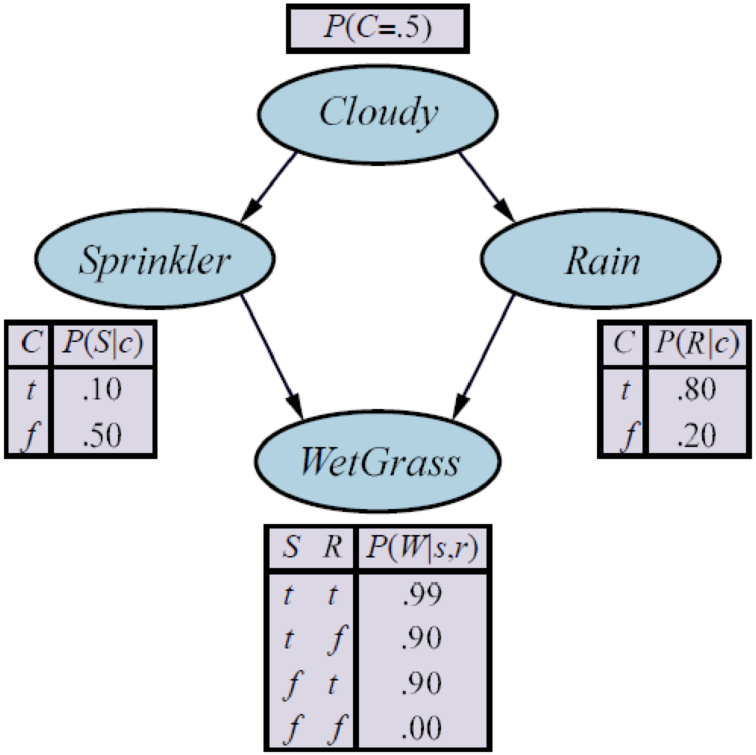
Query $\boldsymbol{P}(Rain \mid Cloudy = true, WetGrass = true)$

- Pick a topological order:

Cloudy, *Sprinkler*, *Rain*, *WetGrass*

evidence
variables
$\boldsymbol{E} = \boldsymbol{e}$

Weight of a Sample



Query $\boldsymbol{P}(Rain \mid Cloudy = true, WetGrass = true)$

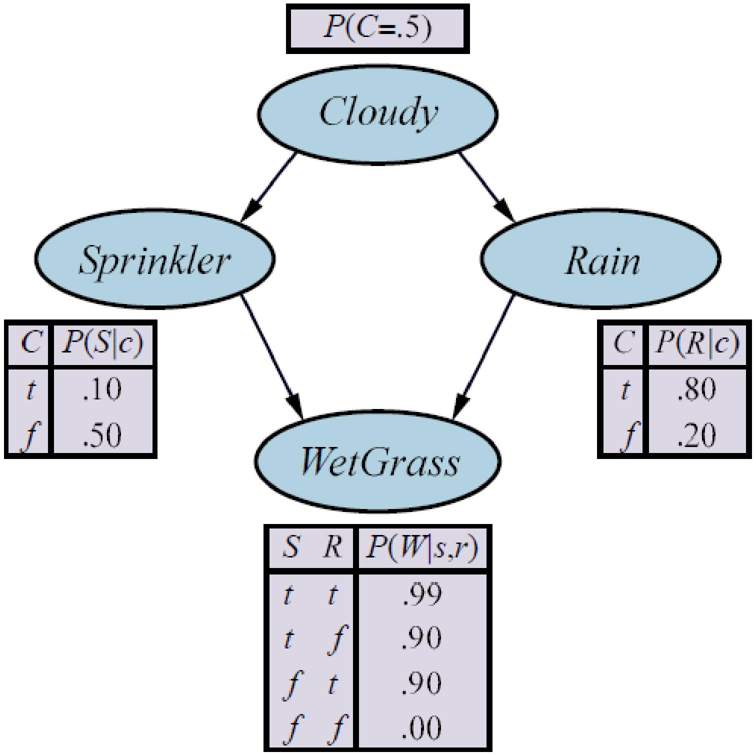
- Pick a topological order:

Cloudy, *Sprinkler*, *Rain*, *WetGrass*

evidence variables $\boldsymbol{E} = \boldsymbol{e}$

hidden variable $\boldsymbol{Z}$

Weight of a Sample



Query $\boldsymbol{P}(Rain \mid Cloudy = true, WetGrass = true)$

- Pick a topological order:

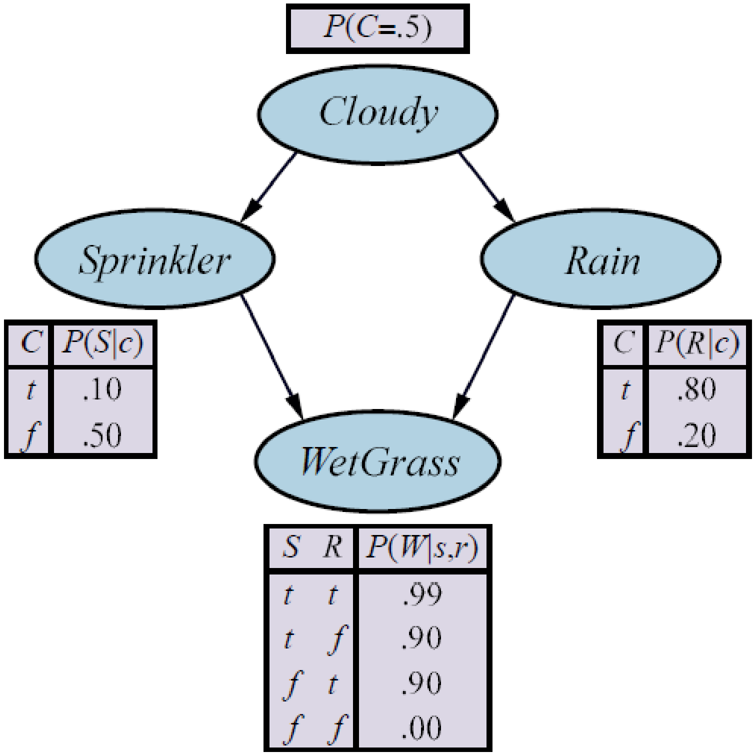
Cloudy, *Sprinkler*, *Rain*, *WetGrass*

evidence variables $\boldsymbol{E} = \boldsymbol{e}$

hidden variable $\boldsymbol{Z}$

query variable X

Weight of a Sample



Query $\boldsymbol{P}(Rain \mid Cloudy = true, WetGrass = true)$

- Pick a topological order:

Cloudy, *Sprinkler*, *Rain*, *WetGrass*

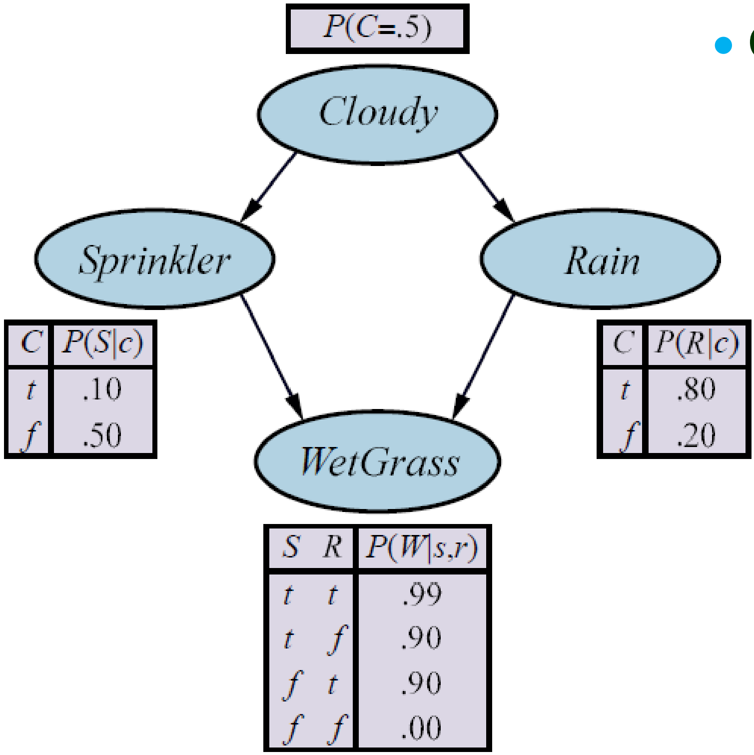
evidence variables $\boldsymbol{E} = \boldsymbol{e}$

hidden variable $\boldsymbol{Z}$

query variable X

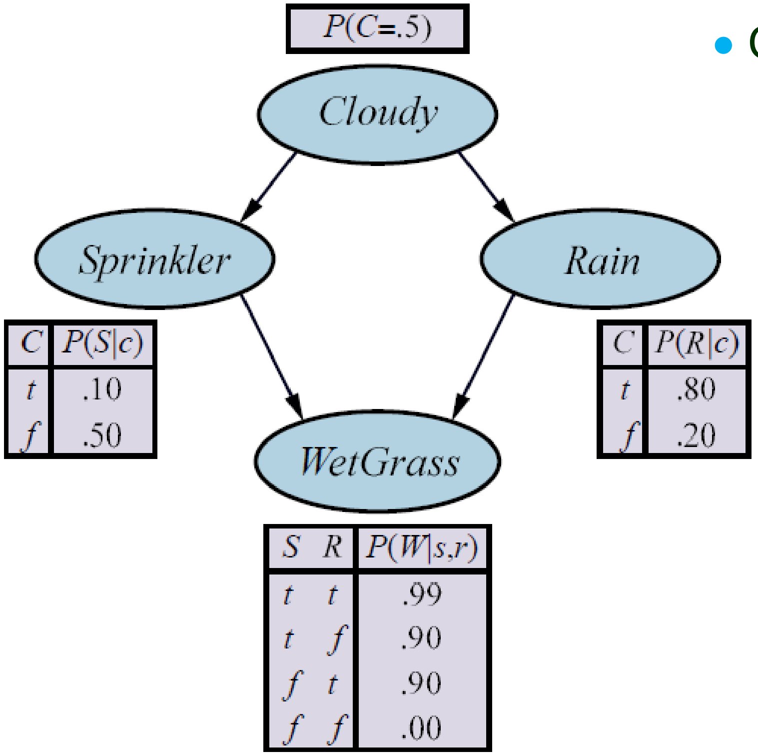
- Set the weight $w = 1$.

(cont'd)



- Generate an event in the chosen topological order.

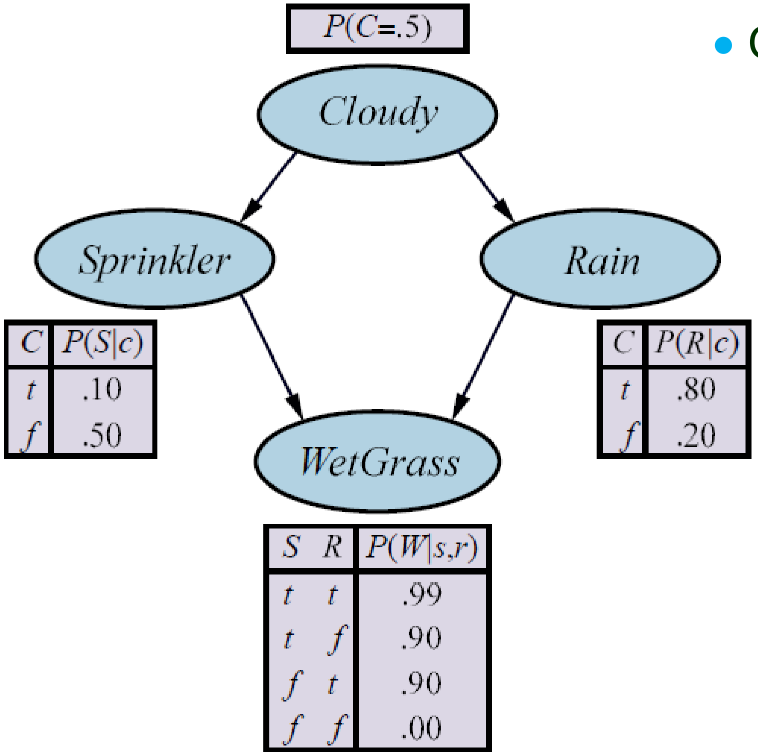
(cont'd)



- Generate an event in the chosen topological order.
    1. *Cloudy* is an evidence variable with value *true*.

$$w \leftarrow w \times P(\mathit{Cloudy} = \mathit{true}) = 0.5$$

(cont'd)



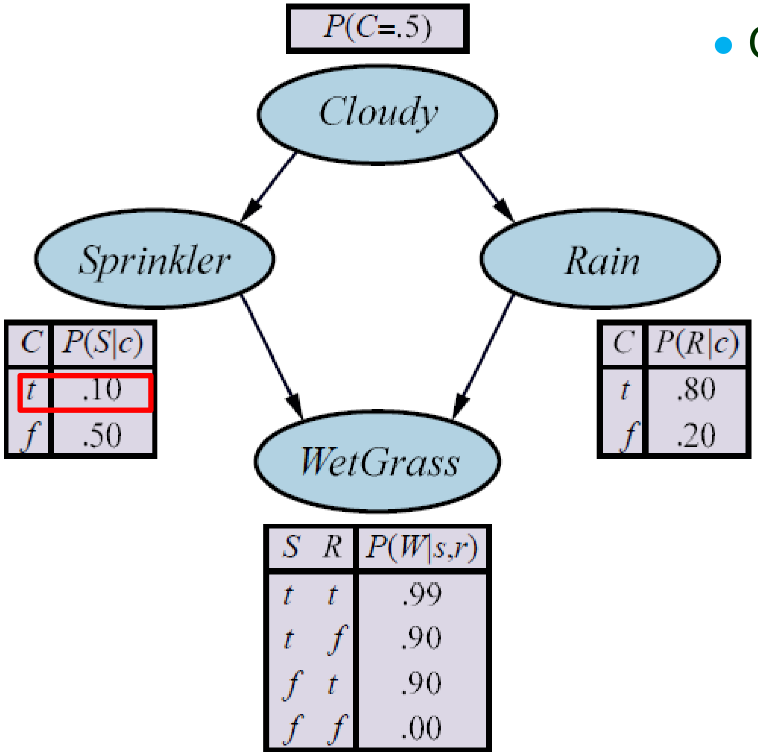
- Generate an event in the chosen topological order.

  1. *Cloudy* is an evidence variable with value *true*.

     $$w \leftarrow w \times P(\mathit{Cloudy} = \mathit{true}) = 0.5$$

  2. *Sprinkler* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Sprinkler} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.1, 0.9 \rangle$.

(cont'd)



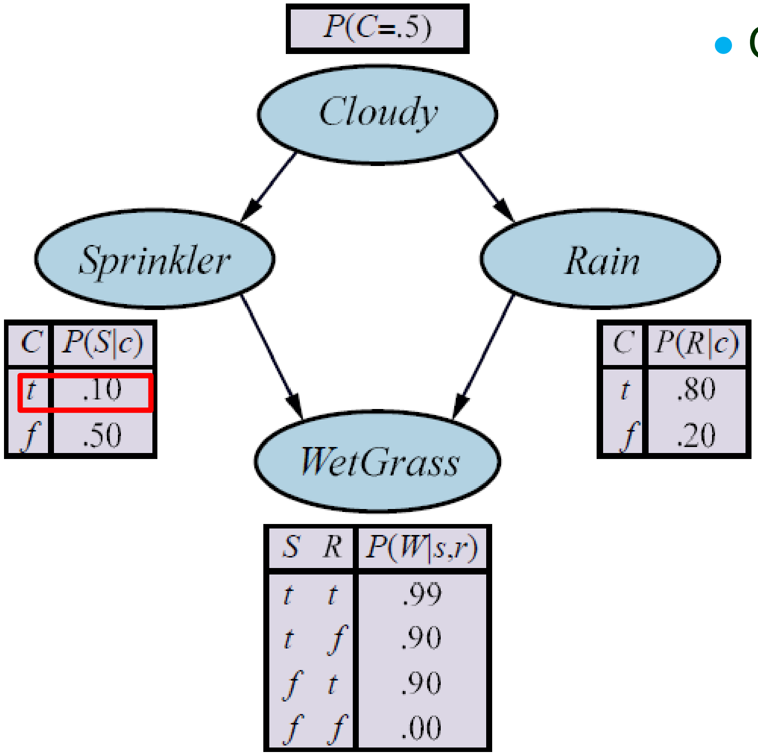
- Generate an event in the chosen topological order.

  1. *Cloudy* is an evidence variable with value *true*.

     $$w \leftarrow w \times P(\mathit{Cloudy} = \mathit{true}) = 0.5$$

  2. *Sprinkler* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Sprinkler} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.1, 0.9 \rangle$.

(cont'd)



- Generate an event in the chosen topological order.

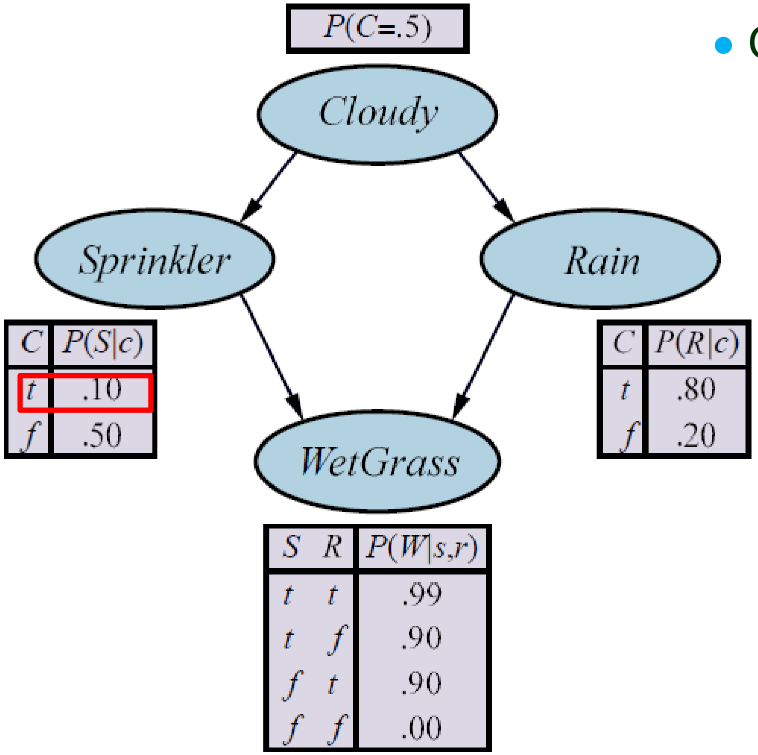
  1. *Cloudy* is an evidence variable with value *true*.

     $$w \leftarrow w \times P(\mathit{Cloudy} = \mathit{true}) = 0.5$$

  2. *Sprinkler* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Sprinkler} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.1, 0.9 \rangle$.

     Suppose this returns *false*.

(cont'd)



- Generate an event in the chosen topological order.

  1. *Cloudy* is an evidence variable with value *true*.

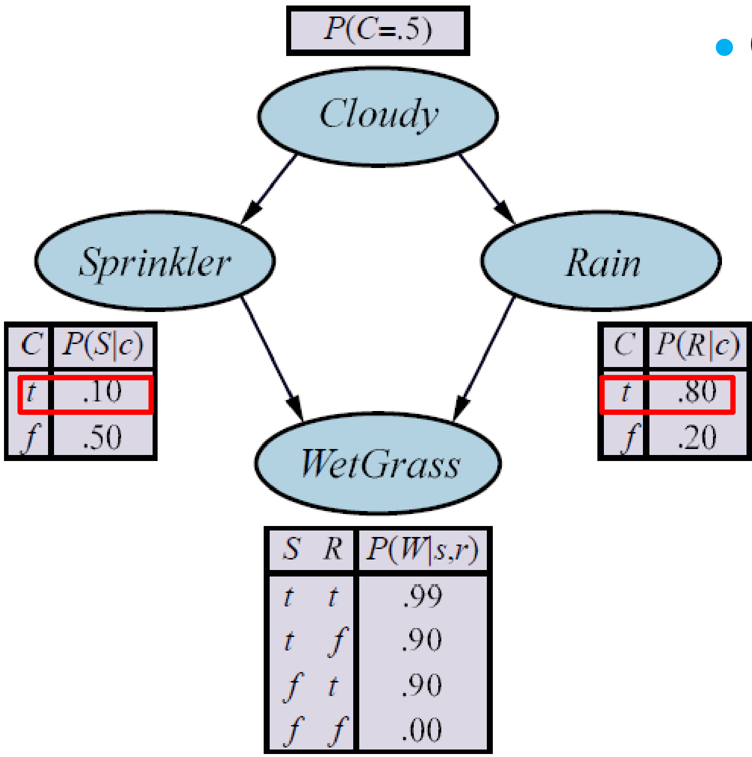
     $$w \leftarrow w \times P(\mathit{Cloudy} = \mathit{true}) = 0.5$$

  2. *Sprinkler* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Sprinkler} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.1, 0.9 \rangle$.

     Suppose this returns *false*.

  3. *Rain* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Rain} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.8, 0.2 \rangle$.

(cont'd)



- Generate an event in the chosen topological order.

  1. *Cloudy* is an evidence variable with value *true*.

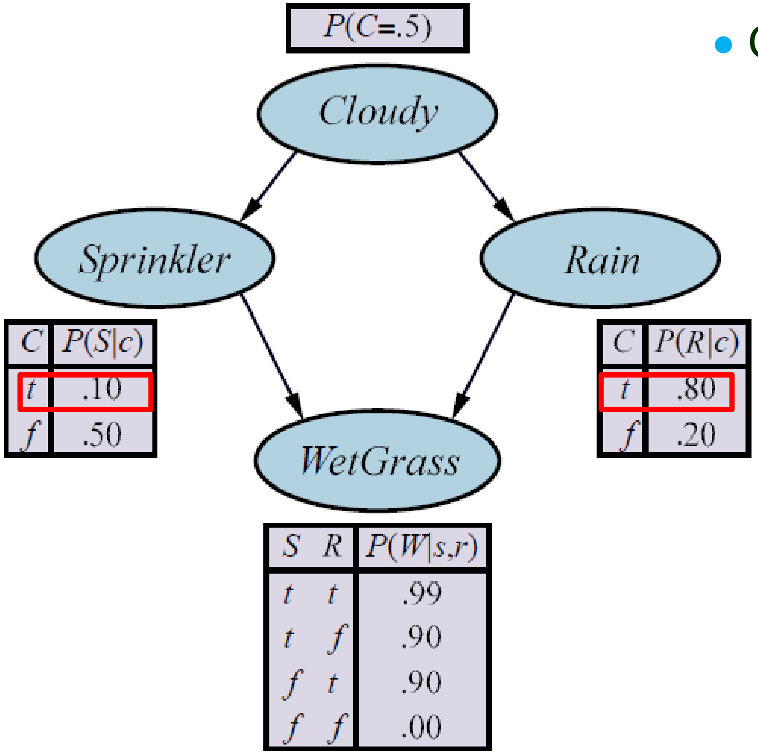
     $$w \leftarrow w \times P(\textit{Cloudy} = \textit{true}) = 0.5$$

  2. *Sprinkler* is not an evidence variable. Sample from $\mathbf{P}(\textit{Sprinkler} \mid \textit{Cloudy} = \textit{true}) = \langle 0.1, 0.9 \rangle$.

     Suppose this returns *false*.

  3. *Rain* is not an evidence variable. Sample from $\mathbf{P}(\textit{Rain} \mid \textit{Cloudy} = \textit{true}) = \langle 0.8, 0.2 \rangle$.

(cont'd)



- Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.

$$w \leftarrow w \times P(\mathit{Cloudy} = \mathit{true}) = 0.5$$

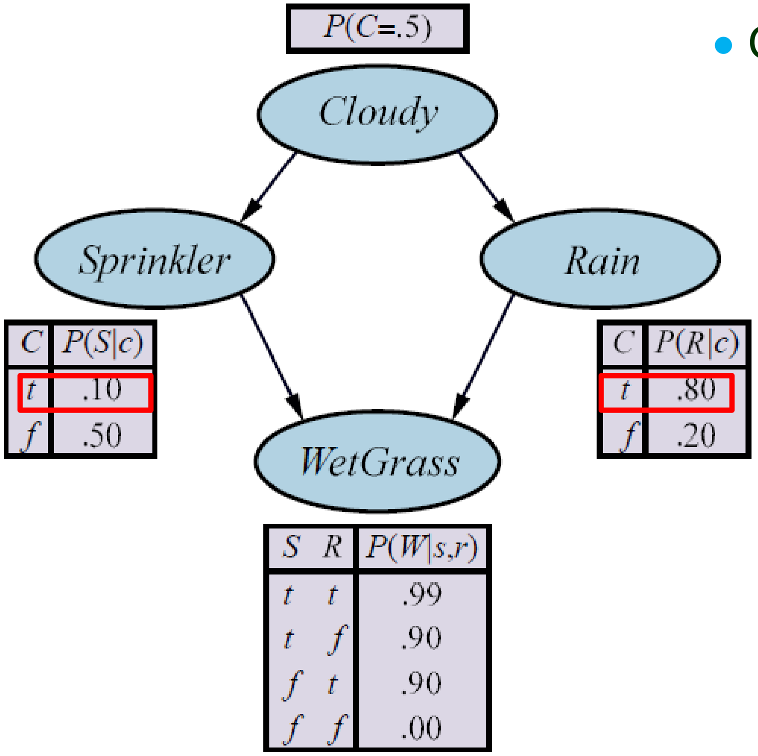
2. *Sprinkler* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Sprinkler} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.1, 0.9 \rangle$.

   Suppose this returns *false*.

3. *Rain* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Rain} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.8, 0.2 \rangle$.

   Suppose this returns *true*.

(cont'd)



- Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.

$$w \leftarrow w \times P(\mathit{Cloudy} = \mathit{true}) = 0.5$$

2. *Sprinkler* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Sprinkler} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.1, 0.9 \rangle$.

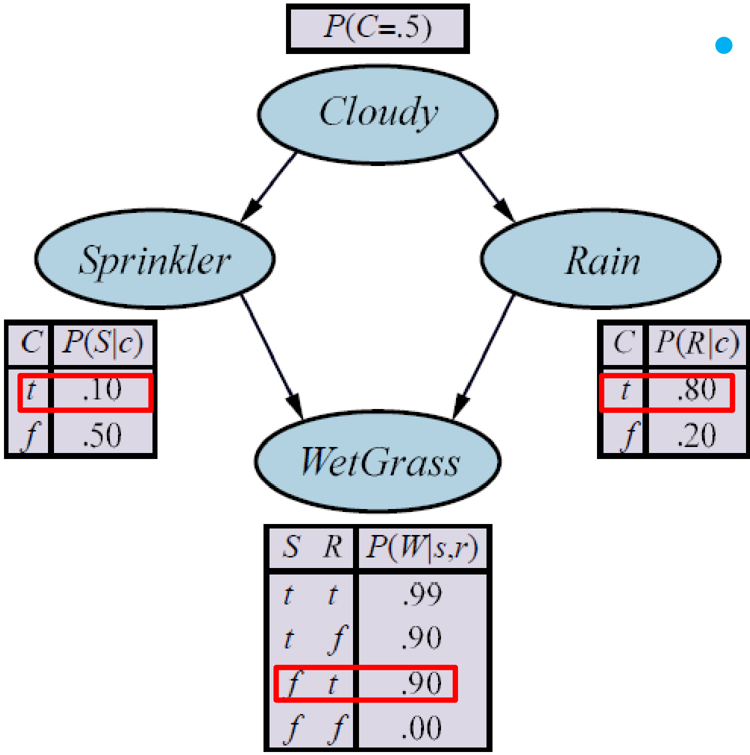
   Suppose this returns *false*.

3. *Rain* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Rain} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.8, 0.2 \rangle$.

   Suppose this returns *true*.

4. *WetGrass* is an evidence variable with value *true*.

(cont'd)



- Generate an event in the chosen topological order.

  1. *Cloudy* is an evidence variable with value *true*.

     $$w \leftarrow w \times P(\mathit{Cloudy} = \mathit{true}) = 0.5$$

  2. *Sprinkler* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Sprinkler} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.1, 0.9 \rangle$.

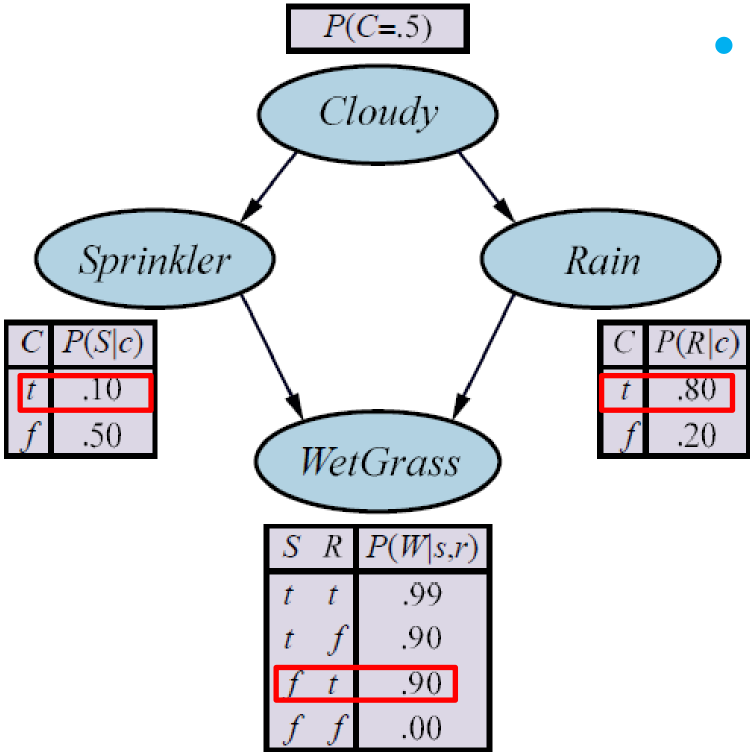
     Suppose this returns *false*.

  3. *Rain* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Rain} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.8, 0.2 \rangle$.

     Suppose this returns *true*.

  4. *WetGrass* is an evidence variable with value *true*.

(cont'd)



- Generate an event in the chosen topological order.

  1. *Cloudy* is an evidence variable with value *true*.

     $$w \leftarrow w \times P(\mathit{Cloudy} = \mathit{true}) = 0.5$$

  2. *Sprinkler* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Sprinkler} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.1, 0.9 \rangle$.

     Suppose this returns *false*.

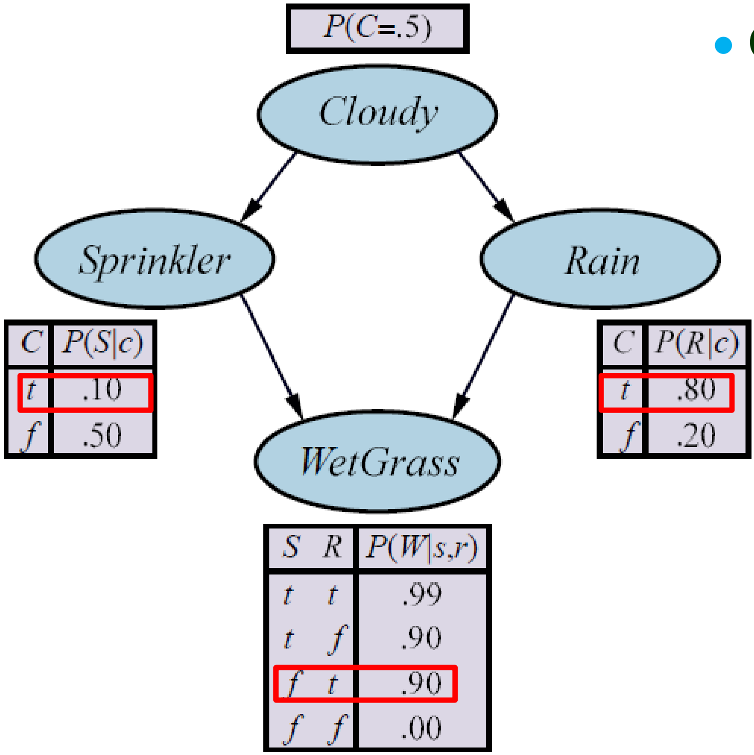
  3. *Rain* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Rain} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.8, 0.2 \rangle$.

     Suppose this returns *true*.

  4. *WetGrass* is an evidence variable with value *true*.

$$w \leftarrow w \times P(\mathit{WetGrass} = \mathit{true} \mid \mathit{Sprinkler} = \mathit{true}, \mathit{Rain} = \mathit{true}) \\ = 0.5 \times 0.9 = 0.45$$

(cont'd)



- Generate an event in the chosen topological order.

1. *Cloudy* is an evidence variable with value *true*.

$$w \leftarrow w \times P(\mathit{Cloudy} = \mathit{true}) = 0.5$$

2. *Sprinkler* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Sprinkler} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.1, 0.9 \rangle$.

   Suppose this returns *false*.

3. *Rain* is not an evidence variable. Sample from $\mathbf{P}(\mathit{Rain} \mid \mathit{Cloudy} = \mathit{true}) = \langle 0.8, 0.2 \rangle$.

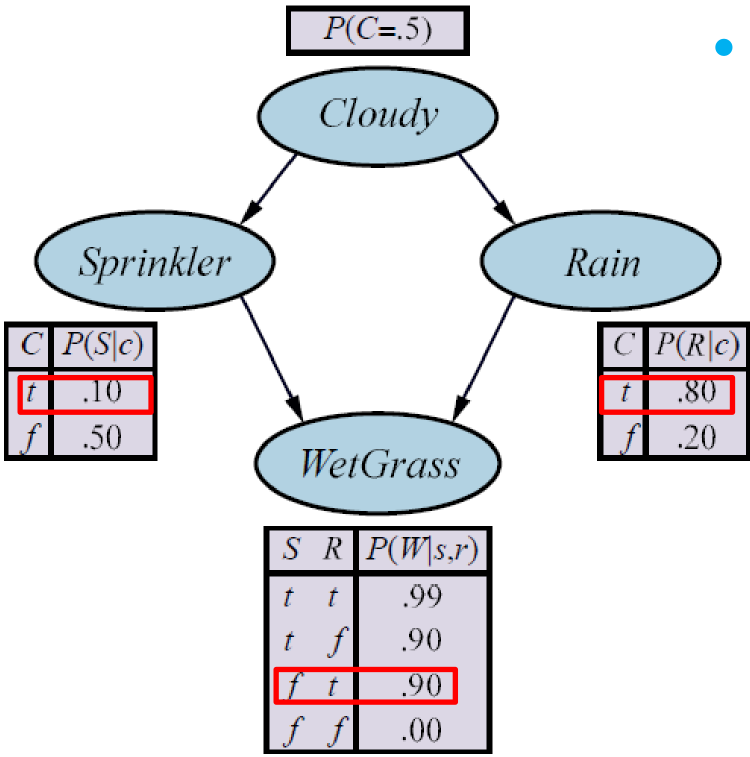
   Suppose this returns *true*.

4. *WetGrass* is an evidence variable with value *true*.

$$w \leftarrow w \times P(\mathit{WetGrass} = \mathit{true} \mid \mathit{Sprinkler} = \mathit{true}, \mathit{Rain} = \mathit{true}) = 0.5 \times 0.9 = 0.45$$

- This round of sampling returns the event [*true*, *false*, *true*, *true*] with weight 0.45.

(cont'd)



- Generate an event in the chosen topological order.

  1. *Cloudy* is an evidence variable with value *true*.

  $$w \leftarrow w \times P(\textit{Cloudy} = \textit{true}) = 0.5$$

  2. *Sprinkler* is not an evidence variable. Sample from $\mathbf{P}(\textit{Sprinkler} \mid \textit{Cloudy} = \textit{true}) = \langle 0.1, 0.9 \rangle$.

     Suppose this returns *false*.

  3. *Rain* is not an evidence variable. Sample from $\mathbf{P}(\textit{Rain} \mid \textit{Cloudy} = \textit{true}) = \langle 0.8, 0.2 \rangle$.

     Suppose this returns *true*.

  4. *WetGrass* is an evidence variable with value *true*.

$$w \leftarrow w \times P(\textit{WetGrass} = \textit{true} \mid \textit{Sprinkler} = \textit{true}, \textit{Rain} = \textit{true}) = 0.5 \times 0.9 = 0.45$$

- This round of sampling returns the event [*true*, *false*, *true*, *true*] with weight 0.45.
- This event is tallied under *Rain* = *true* in generating the distribution estimate

$$\hat{\mathbf{P}}(\textit{Rain} \mid \textit{Cloudy} = \textit{true}, \textit{WetGrass} = \textit{true})$$

Likelihood Weighting

Fix the values for the evidence variables $\boldsymbol{E}$ and sample all the nonevidence variables $\{X\} \cup \boldsymbol{Z}$ in topological order, each conditioned on its parents.

Likelihood Weighting

Fix the values for the evidence variables $\boldsymbol{E}$ and sample all the nonevidence variables $\{X\} \cup \boldsymbol{Z}$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $\boldsymbol{Z} = \{Z_1, \ldots, Z_l\}$ (for evidence $\boldsymbol{E} = \boldsymbol{e}$):

$$Q_{WS}(\boldsymbol{z}) = \prod_{i=1}^{l} P(z_i) \mid \mathit{parents}(Z_i))$$

Likelihood Weighting

Fix the values for the evidence variables $\boldsymbol{E}$ and sample all the nonevidence variables $\{X\} \cup \boldsymbol{Z}$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $\boldsymbol{Z} = \{Z_1, \ldots, Z_l\}$ (for evidence $\boldsymbol{E} = \boldsymbol{e}$):

$$Q_{WS}(\boldsymbol{z}) = \prod_{i=1}^{l} P(z_i) \mid parents(Z_i))$$

The weight $w(\boldsymbol{z})$ must satisfy

$$P(\boldsymbol{z}, \boldsymbol{e}) = P(\boldsymbol{e})P(\boldsymbol{z} \mid \boldsymbol{e}) = P(\boldsymbol{e})w(\boldsymbol{z})Q_{WS}(\boldsymbol{z})$$

Likelihood Weighting

Fix the values for the evidence variables $\boldsymbol{E}$ and sample all the nonevidence variables $\{X\} \cup \boldsymbol{Z}$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $\boldsymbol{Z} = \{Z_1, \dots, Z_l\}$ (for evidence $\boldsymbol{E} = \boldsymbol{e}$):

$$Q_{WS}(\boldsymbol{z}) = \prod_{i=1}^{l} P(z_i) \mid \mathit{parents}(Z_i))$$

The weight $w(\boldsymbol{z})$ must satisfy

$$P(\boldsymbol{z}, \boldsymbol{e}) = P(\boldsymbol{e})P(\boldsymbol{z} \mid \boldsymbol{e}) = P(\boldsymbol{e}\,)w(\boldsymbol{z})Q_{WS}(\boldsymbol{z})$$

$$\Downarrow$$

$$w(\boldsymbol{z}) = \frac{1}{P(\boldsymbol{e}\,)} \cdot \frac{P(\boldsymbol{z}, \boldsymbol{e})}{Q_{WS}(\boldsymbol{z})} = \alpha \frac{P(\boldsymbol{z}, \boldsymbol{e})}{Q_{WS}(\boldsymbol{z})}$$

Likelihood Weighting

Fix the values for the evidence variables $\boldsymbol{E}$ and sample all the nonevidence variables $\{X\} \cup \boldsymbol{Z}$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $\boldsymbol{Z} = \{Z_1, \dots, Z_l\}$ (for evidence $\boldsymbol{E} = \boldsymbol{e}$):

$$Q_{WS}(\boldsymbol{z}) = \prod_{i=1}^{l} P(z_i) \mid parents(Z_i))$$

The weight $w(\boldsymbol{z})$ must satisfy

$$P(\boldsymbol{z}, \boldsymbol{e}) = P(\boldsymbol{e}) P(\boldsymbol{z} \mid \boldsymbol{e}) = P(\boldsymbol{e}) w(\boldsymbol{z}) Q_{WS}(\boldsymbol{z})$$

$$\Downarrow$$

$$w(\boldsymbol{z}) = \frac{1}{P(\boldsymbol{e})} \cdot \frac{P(\boldsymbol{z}, \boldsymbol{e})}{Q_{WS}(\boldsymbol{z})} = \alpha \frac{P(\boldsymbol{z}, \boldsymbol{e})}{Q_{WS}(\boldsymbol{z})}$$

(normalization factor $\alpha = 1/P(\boldsymbol{e})$)

Likelihood Weighting

Fix the values for the evidence variables $\boldsymbol{E}$ and sample all the nonevidence variables $\{X\} \cup \boldsymbol{Z}$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $\boldsymbol{Z} = \{Z_1, \ldots, Z_l\}$ (for evidence $\boldsymbol{E} = \boldsymbol{e}$):

$$Q_{WS}(\boldsymbol{z}) = \prod_{i=1}^{l} P(z_i) \mid parents(Z_i))$$

The weight $w(\boldsymbol{z})$ must satisfy

$$P(\boldsymbol{z}, \boldsymbol{e}) = P(\boldsymbol{e})P(\boldsymbol{z} \mid \boldsymbol{e}) = P(\boldsymbol{e}\,)w(\boldsymbol{z})Q_{WS}(\boldsymbol{z})$$

$$\Downarrow$$

$$w(\boldsymbol{z}) = \frac{1}{P(\boldsymbol{e}\,)} \cdot \frac{P(\boldsymbol{z}, \boldsymbol{e})}{Q_{WS}(\boldsymbol{z})} = \alpha \frac{P(\boldsymbol{z}, \boldsymbol{e})}{Q_{WS}(\boldsymbol{z})}$$

(normalization factor $\alpha = 1/P(\boldsymbol{e})$)

$$= \alpha \frac{\prod_{i=1}^{l} P(z_i) \mid parents(Z_i)) \cdot \prod_{i=1}^{m} P(e_i) \mid parents(E_i))}{\prod_{i=1}^{l} P(z_i) \mid parents(Z_i))}$$

Likelihood Weighting

Fix the values for the evidence variables $\boldsymbol{E}$ and sample all the nonevidence variables $\{X\} \cup \boldsymbol{Z}$ in topological order, each conditioned on its parents.

Sampling distribution of the hidden variables $\boldsymbol{Z} = \{Z_1, \dots, Z_l\}$ (for evidence $\boldsymbol{E} = \boldsymbol{e}$):

$$Q_{WS}(\boldsymbol{z}) = \prod_{i=1}^{l} P(z_i) \mid parents(Z_i))$$

The weight $w(\boldsymbol{z})$ must satisfy

$$P(\boldsymbol{z}, \boldsymbol{e}) = P(\boldsymbol{e}) P(\boldsymbol{z} \mid \boldsymbol{e}) = P(\boldsymbol{e}) w(\boldsymbol{z}) Q_{WS}(\boldsymbol{z})$$

$$\Downarrow$$

$$w(\boldsymbol{z}) = \frac{1}{P(\boldsymbol{e})} \cdot \frac{P(\boldsymbol{z}, \boldsymbol{e})}{Q_{WS}(\boldsymbol{z})} = \alpha \frac{P(\boldsymbol{z}, \boldsymbol{e})}{Q_{WS}(\boldsymbol{z})}$$

(normalization factor $\alpha = 1/P(\boldsymbol{e})$)

$$= \alpha \frac{\prod_{i=1}^{l} P(z_i) \mid parents(Z_i)) \cdot \prod_{i=1}^{m} P(e_i) \mid parents(E_i))}{\prod_{i=1}^{l} P(z_i) \mid parents(Z_i))}$$

$$= \alpha \prod_{i=1}^{m} P(e_i) \mid parents(E_i))$$

Weighted Sampling

$$w(\mathbf{z}) = \alpha \prod_{i=1}^{m} P(e_i) \mid parents(E_i))$$

The weight is the product of the conditional probabilities for the evidence variables given their parents.

function WEIGHTED-SAMPLE(bn, **e**) **returns** an event and a weight

  $w \leftarrow 1$; $\mathbf{x} \leftarrow$ an event with n elements, with values fixed from **e**
  for i = 1 **to** n **do**
    if X_i is an evidence variable with value x_{ij} in **e**
      then $w \leftarrow w \times\ P(X_i = x_{ij} \mid parents(X_i))$
      else $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return $\mathbf{x}$, w

The Likelihood Weighting Algorithm

function LIKELIHOOD-WEIGHTING(X, $\mathbf{e}$, bn, N) **returns** an estimate of $\mathbf{P}(X \mid \mathbf{e})$
  inputs: X, the query variable
    $\mathbf{e}$, observed values for variables $\mathbf{E}$
    bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$
    N, the total number of samples to be generated
  local variables: $\mathbf{W}$, a vector of weighted counts for each value of X, initially zero

  for j = 1 **to** N **do**
    $\mathbf{x}$, $w \leftarrow$ WEIGHTED-SAMPLE(bn, $\mathbf{e}$)
    $\mathbf{W}[j] \leftarrow \mathbf{W}[j] + w$ where x_j is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{W}$)

function WEIGHTED-SAMPLE(bn, $\mathbf{e}$) **returns** an event and a weight

  $w \leftarrow 1$; $\mathbf{x} \leftarrow$ an event with n elements, with values fixed from $\mathbf{e}$
  for i = 1 **to** n **do**
    if X_i is an evidence variable with value x_{ij} in $\mathbf{e}$
      then $w \leftarrow w \times P(X_i = x_{ij} \mid parents(X_i))$
      else $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return $\mathbf{x}$, w