

Approximate Inference in Bayes Nets

- ♠ Exact inference in large networks is intractable.
- ♦ Randomized sampling is used for computing conditional probabilities.

Direct sampling methods (generate samples from scratch):

I. Random sampling

II. Rejection sampling

III. Importance sampling

Approximate Inference Methods

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

Approximate Inference Methods

- ◆ Sampling 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

- ◆ Sampling 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

- ◆ Sampling 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. Random 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. Random 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 $\mathbf{P}(X_1, \dots, X_n)$

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

I. Random 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 $\mathbf{P}(X_1, \dots, X_n)$

$\mathbf{x} \leftarrow$ an event with n elements

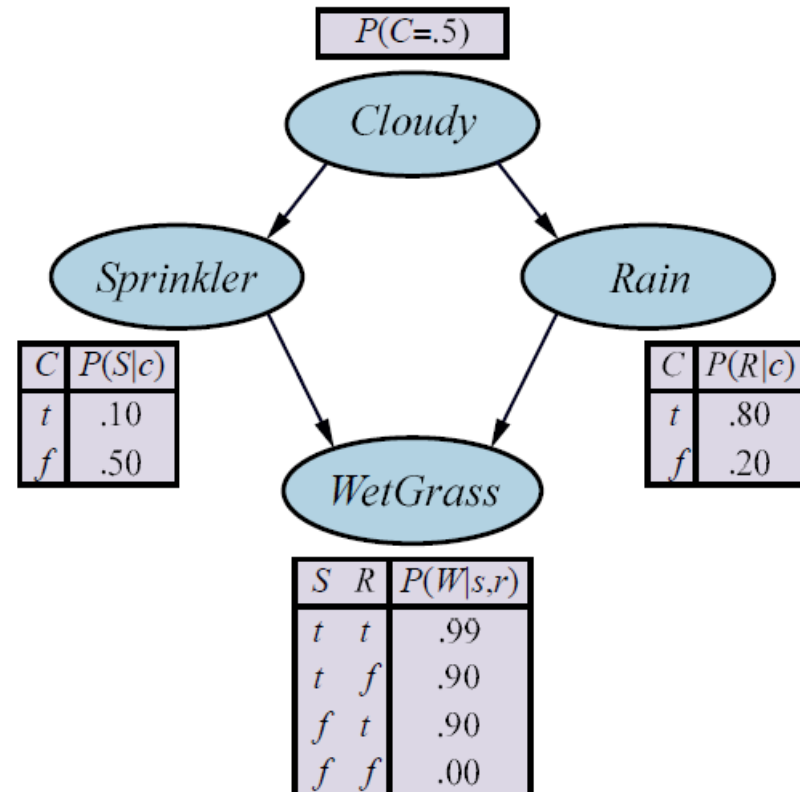
for each variable X_i **in** X_1, \dots, X_n **do**

$\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid \text{parents}(X_i))$

return \mathbf{x} from the domain of X_i , e.g., true or false

The Sprinkler Network

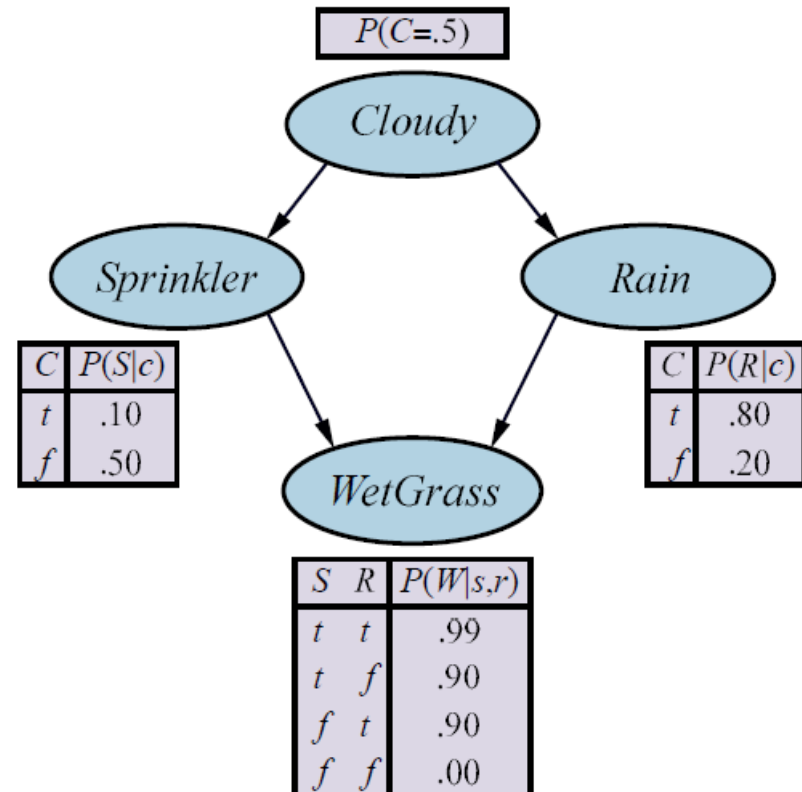
- Every morning Mary checks the weather.
- If it's cloudy, she usually does not turn on the sprinkler.
- The grass will be wet if the sprinkler is on, or if it rains during the day.



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*

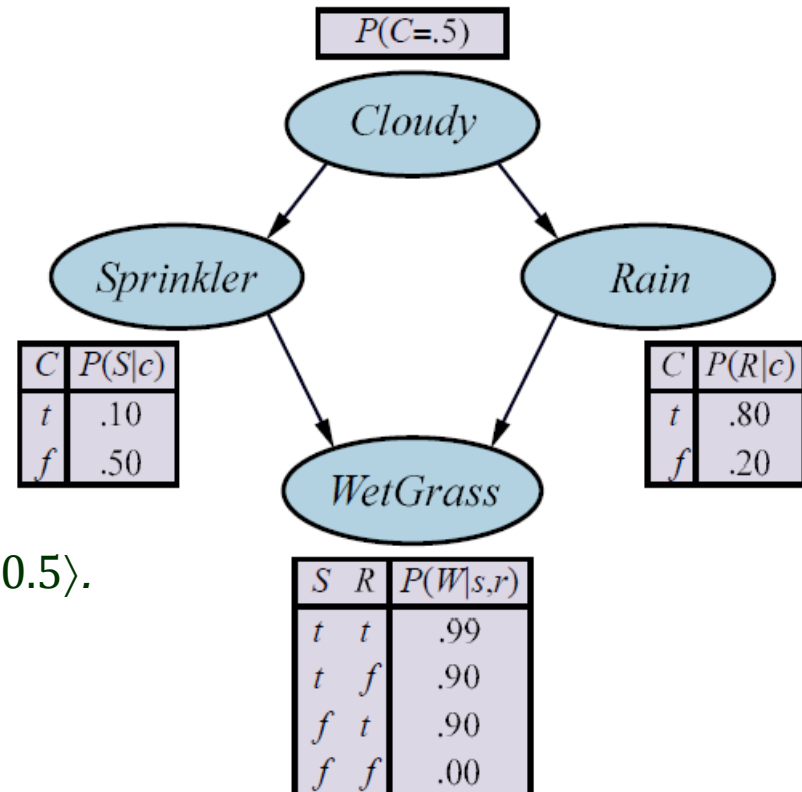


The Sprinkler Network

- 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*

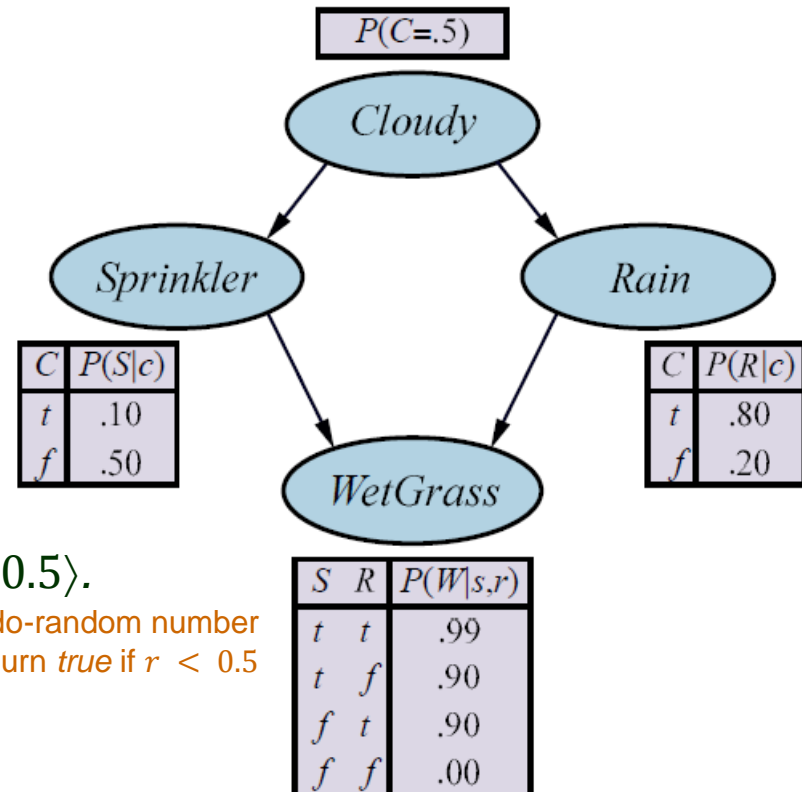
1. Value *true* sampled from $P(\text{Cloudy}) = \langle 0.5, 0.5 \rangle$.



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*

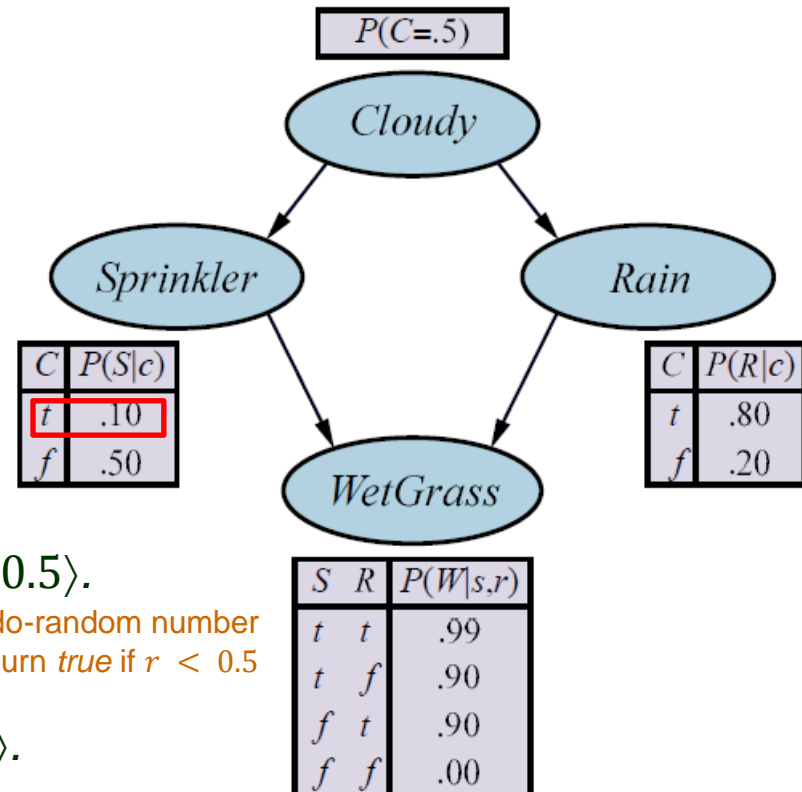


1. Value *true* sampled from $P(\text{Cloudy}) = \langle 0.5, 0.5 \rangle$.

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

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 $P(\text{Cloudy}) = \langle 0.5, 0.5 \rangle$.

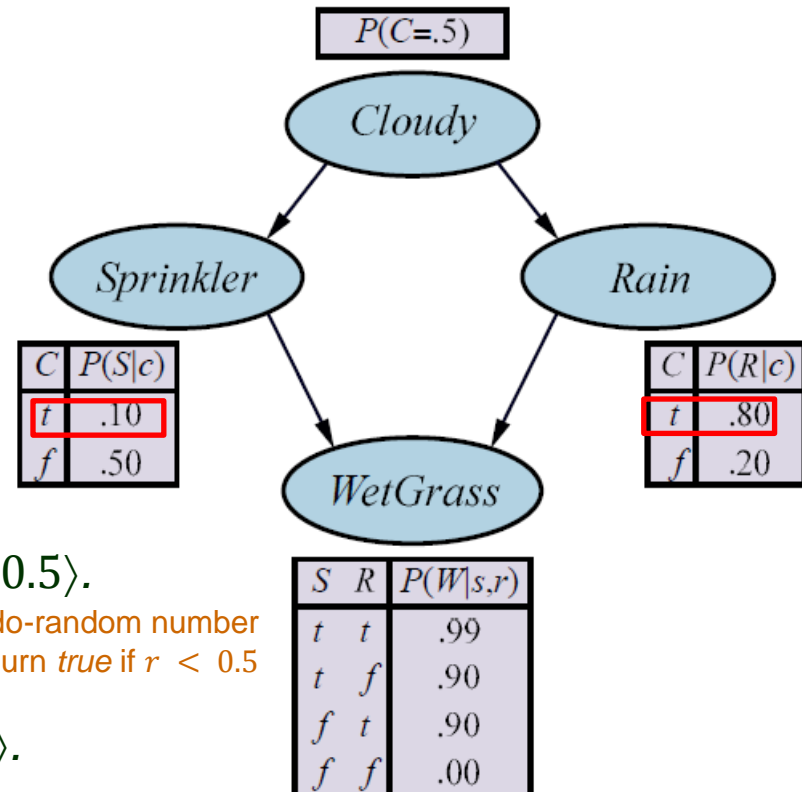
How? Generate a pseudo-random number r in the range $[0, 1]$. Return *true* if $r < 0.5$ and *false* otherwise.

2. Value *false* sampled from

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

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 $P(\text{Cloudy}) = \langle 0.5, 0.5 \rangle$.

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

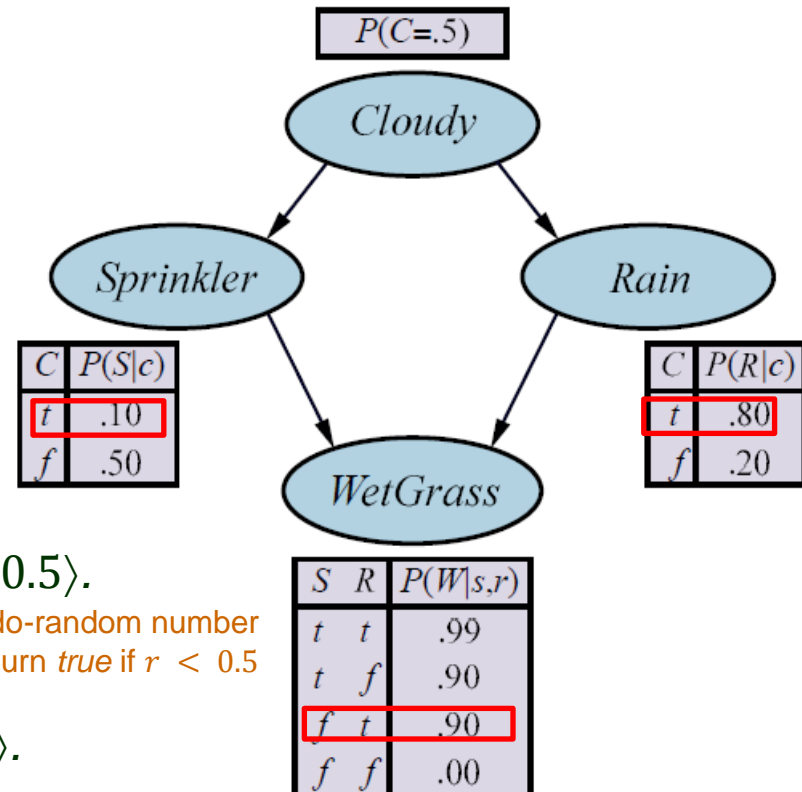
$$P(\text{Sprinkler} \mid \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle.$$

3. Value *true* sampled from

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

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 $P(\text{Cloudy}) = \langle 0.5, 0.5 \rangle$.

2. Value *false* sampled from r in the range $[0, 1]$. Return *true* if $r < 0.5$ and *false* otherwise.

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

3. Value *true* sampled from

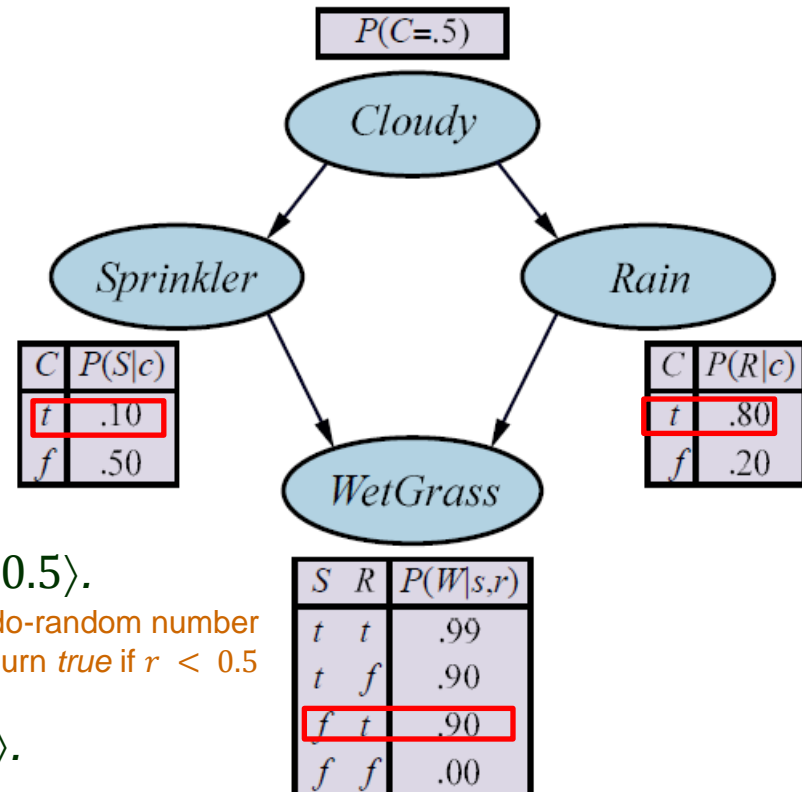
$$P(\text{Rain} \mid \text{Cloudy} = \text{true}) = \langle 0.8, 0.2 \rangle.$$

4. Value *true* sampled from

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

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 $P(\text{Cloudy}) = \langle 0.5, 0.5 \rangle$.

2. Value *false* sampled from r in the range $[0, 1]$. Return *true* if $r < 0.5$ and *false* otherwise.

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

3. Value *true* sampled from

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

4. Value *true* sampled from

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

PRIOR-SAMPLE() returns the event $[\text{true}, \text{false}, \text{true}, \text{true}]$.

Sampling Process

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

Sampling Process

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

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

Sampling Process

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

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

$$= P(x_1, \dots, x_n) \quad (\text{probability of joint distribution})$$

Sampling Process

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

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

$$= P(x_1, \dots, x_n) \quad (\text{probability of joint distribution})$$

- N total samples produced by PRIOR-SAMPLE.

Sampling Process

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

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

$$= P(x_1, \dots, x_n) \quad (\text{probability of joint distribution})$$

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

Sampling Process

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

$$S_{PS}(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i \mid \text{parents}(X_i))$$
$$= P(x_1, \dots, x_n) \quad (\text{probability of joint distribution})$$

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

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

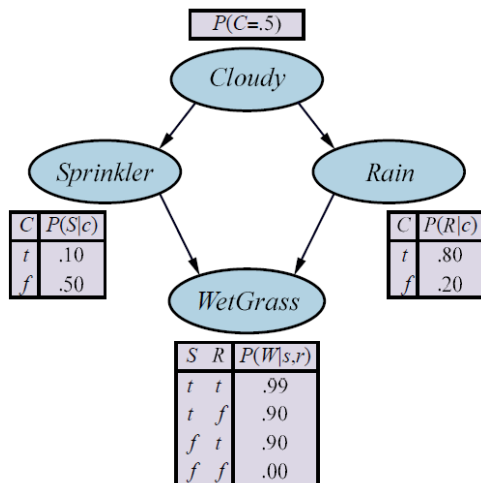
Sampling Process

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

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

$$= P(x_1, \dots, x_n) \quad (\text{probability of joint distribution})$$

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



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

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

Cloudy, Sprinkler, Rain, WetGrass

Partially Specified Event

Estimate the probability of the partial event $X_{i_1} = x_{i_1} \wedge \cdots \wedge X_{i_m} = x_{i_m}$,
 $1 \leq i_1 < \cdots < i_m \leq n$:

$$P(x_{i_1}, \dots, x_{i_m}) \approx \frac{N_{ps}(x_{i_1}, \dots, x_{i_m})}{N} \equiv \hat{P}(x_{i_1}, \dots, x_{i_m})$$

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

Partially Specified Event

Estimate the probability of the partial event $X_{i_1} = x_{i_1} \wedge \dots \wedge X_{i_m} = x_{i_m}$, $1 \leq i_1 < \dots < i_m \leq n$:

$$P(x_{i_1}, \dots, x_{i_m}) \approx \frac{N_{ps}(x_{i_1}, \dots, x_{i_m})}{N} \equiv \hat{P}(x_{i_1}, \dots, x_{i_m})$$

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.

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

II. Rejection Sampling (RS)

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

Estimate $P(X | e)$ as follows:

II. Rejection Sampling (RS)

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

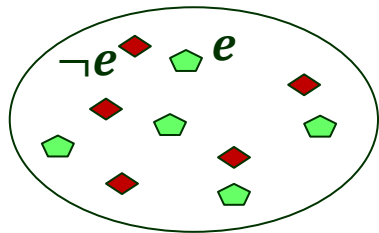
Estimate $P(X | e)$ as follows:

- 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 $P(X | e)$ as follows:

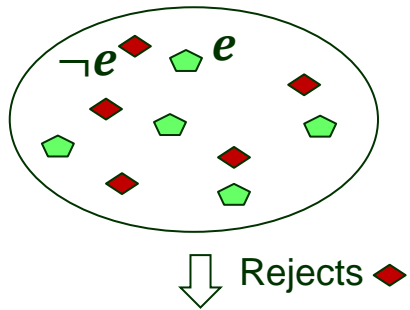


- Generate samples from the prior distribution specified by the BN.

II. Rejection Sampling (RS)

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

Estimate $P(X | e)$ as follows:

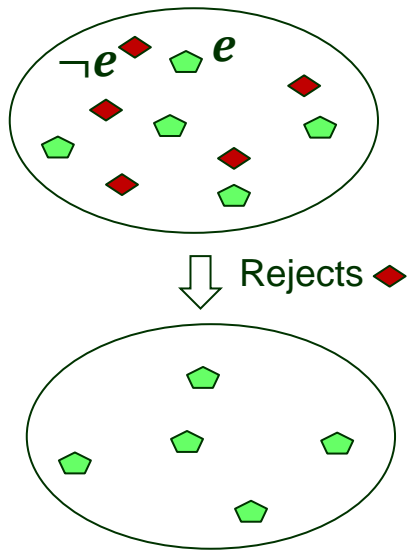


- Generate samples from the prior distribution specified by the BN.
- Rejects all those that do not match the evidence e .

II. Rejection Sampling (RS)

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

Estimate $P(X | e)$ as follows:

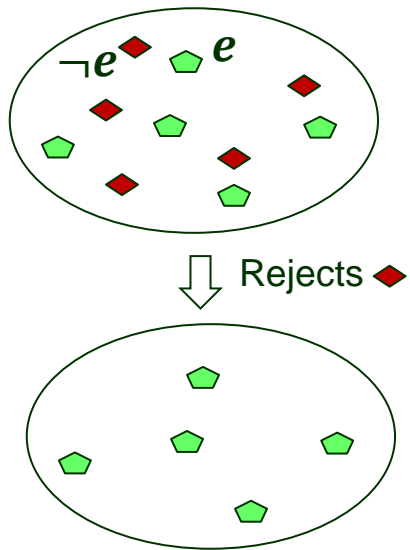


- Generate samples from the prior distribution specified by the BN.
- Rejects all those that do not match the evidence e .

II. Rejection Sampling (RS)

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

Estimate $P(X | e)$ as follows:



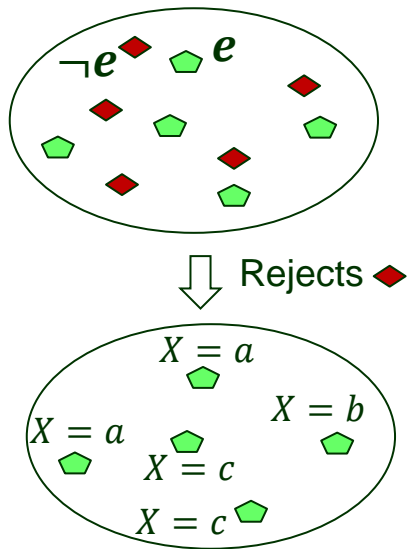
- Generate samples from the prior distribution specified by the BN.
- Rejects all those that do not match the evidence e .

$N_{PS}(e)$: 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 $P(X | e)$ as follows:



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

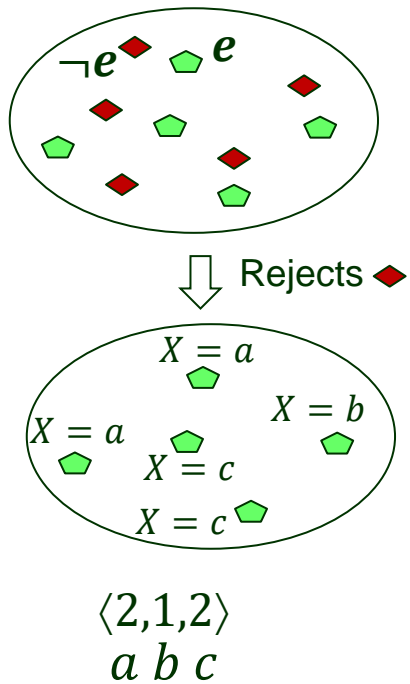
$N_{PS}(e)$: the number of samples that are not rejected.

- Count how often $X = x$ occurs in the remaining $N_{PS}(e)$ 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 $P(X | e)$ as follows:



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

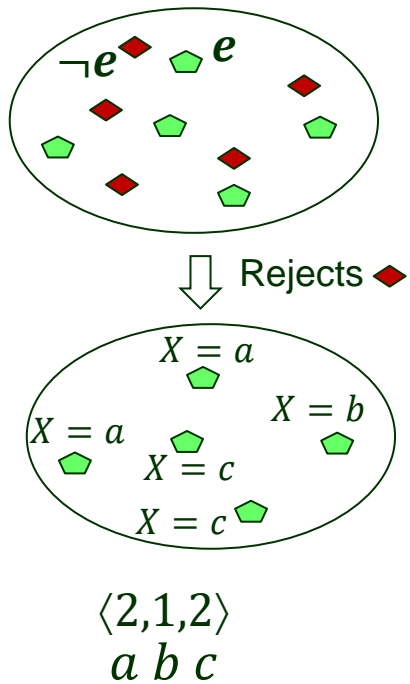
$N_{PS}(e)$: the number of samples that are not rejected.

- Count how often $X = x$ occurs in the remaining $N_{PS}(e)$ 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 $P(X | e)$ as follows:



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

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

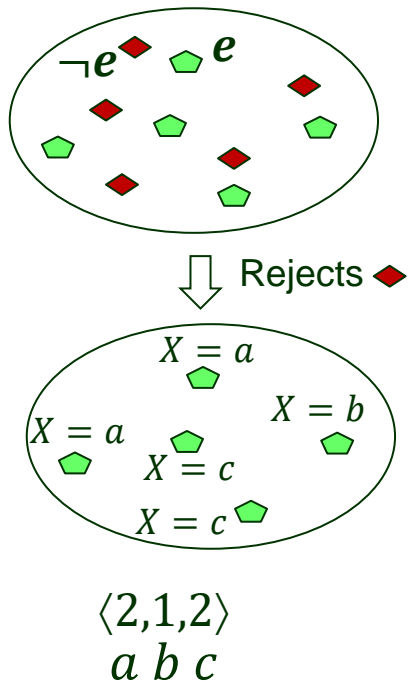
- Count how often $X = x$ occurs in the remaining $N_{PS}(e)$ samples for every value x of X .

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

II. Rejection Sampling (RS)

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

Estimate $P(X | e)$ as follows:



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

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

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

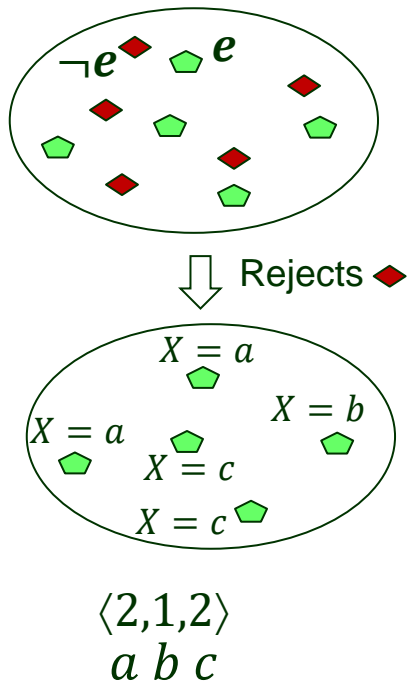
$N_{PS}(X, e)$: the vector of counts, one for each value x of X , of samples that contain $X = x$ and also agree with the evidence e .

$$\hat{P}(X | e) = \alpha N_{PS}(X, e) = \frac{N_{PS}(X, e)}{N_{PS}(e)}$$

II. Rejection Sampling (RS)

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

Estimate $P(X | e)$ as follows:



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

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

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

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

$$\hat{P}(X | e) = \alpha N_{PS}(X, e) = \frac{N_{PS}(X, e)}{N_{PS}(e)}$$

$$\hat{P}(X | e) = \langle 0.4, 0.2, 0.4 \rangle$$

Estimation of True Probability by RS

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

Estimation of True Probability by RS

$$\begin{aligned}\hat{\mathbf{P}}(X | \mathbf{e}) &= \frac{N_{PS}(X, \mathbf{e})}{N_{PS}(\mathbf{e})} \\ &= \frac{N_{PS}(X, \mathbf{e})/N}{N_{PS}(\mathbf{e})/N}\end{aligned}$$

Estimation of True Probability by RS

$$\begin{aligned}\hat{\mathbf{P}}(X | \mathbf{e}) &= \frac{N_{PS}(X, \mathbf{e})}{N_{PS}(\mathbf{e})} \\ &= \frac{N_{PS}(X, \mathbf{e})/N}{N_{PS}(\mathbf{e})/N} \\ &\approx \frac{\mathbf{P}(X, \mathbf{e})}{P(\mathbf{e})} \quad \text{since } P(x_1, \dots, x_m) \approx \frac{N_{ps}(x_1, \dots, x_m)}{N}\end{aligned}$$

Estimation of True Probability by RS

$$\begin{aligned}\hat{P}(X | \mathbf{e}) &= \frac{N_{PS}(X, \mathbf{e})}{N_{PS}(\mathbf{e})} \\ &= \frac{N_{PS}(X, \mathbf{e})/N}{N_{PS}(\mathbf{e})/N} \\ &\approx \frac{P(X, \mathbf{e})}{P(\mathbf{e})}\end{aligned}$$

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

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

Estimation of True Probability by RS

$$\begin{aligned}\hat{P}(X | \mathbf{e}) &= \frac{N_{PS}(X, \mathbf{e})}{N_{PS}(\mathbf{e})} \\ &= \frac{N_{PS}(X, \mathbf{e})/N}{N_{PS}(\mathbf{e})/N}\end{aligned}$$

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

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

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

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

Estimation of True Probability by RS

$$\begin{aligned}\hat{P}(X | e) &= \frac{N_{PS}(X, e)}{N_{PS}(e)} \\ &= \frac{N_{PS}(X, e)/N}{N_{PS}(e)/N}\end{aligned}$$

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

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

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



$$\hat{P}(X | e) \approx P(X | e)$$

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

Estimation of True Probability by RS

$$\begin{aligned}\hat{P}(X | e) &= \frac{N_{PS}(X, e)}{N_{PS}(e)} \\ &= \frac{N_{PS}(X, e)/N}{N_{PS}(e)/N}\end{aligned}$$

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

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

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

$$\hat{P}(X | e) \approx P(X | e)$$

Example Estimate $P(\text{Rain} | \text{Sprinkler} = \text{true})$ 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

$$\begin{aligned}\hat{P}(X | e) &= \frac{N_{PS}(X, e)}{N_{PS}(e)} \\ &= \frac{N_{PS}(X, e)/N}{N_{PS}(e)/N}\end{aligned}$$

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

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

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

$$\hat{P}(X | e) \approx P(X | e)$$

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

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

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

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 fraction of samples that are accepted.



= prior probability of the evidence $P(e)$

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 fraction of samples that are accepted.



= prior probability of the evidence $P(e)$

♠ $P(e)$ is vanishingly small for complex networks with many evidence variables.

- The fraction of samples consistent with e drops exponentially as the number of evidence variables grows.

- Rejection sampling is unusable for complex problems.

III. Importance Sampling (IS)

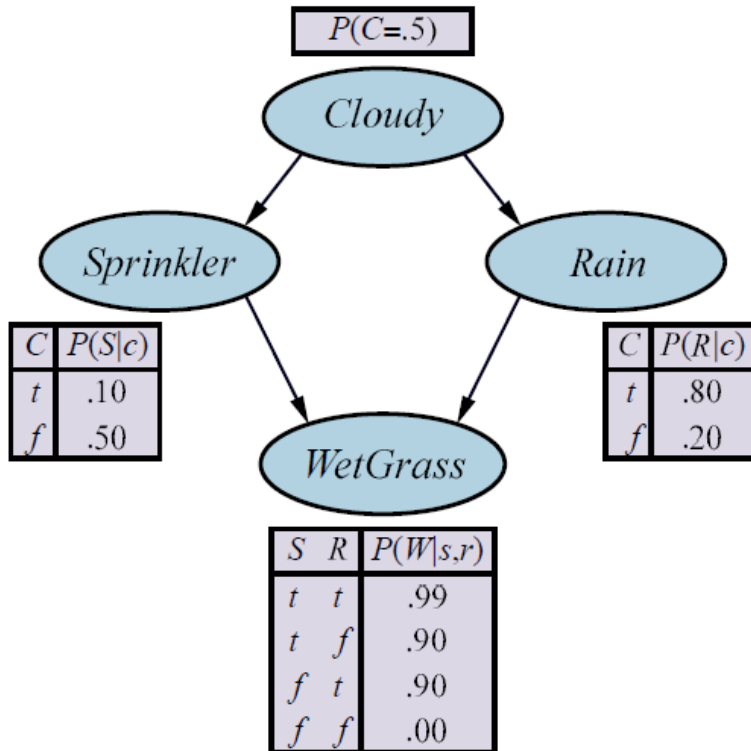
- ◆ Emulate the effect of sampling from one distribution P using samples from another distribution Q .
 - ♣ It is too hard to sample from the true posterior distribution on all the evidence.
 - ♣ So, we sample from an easy distribution.
- ◆ To ensure correctness in the limit, we use a correction factor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

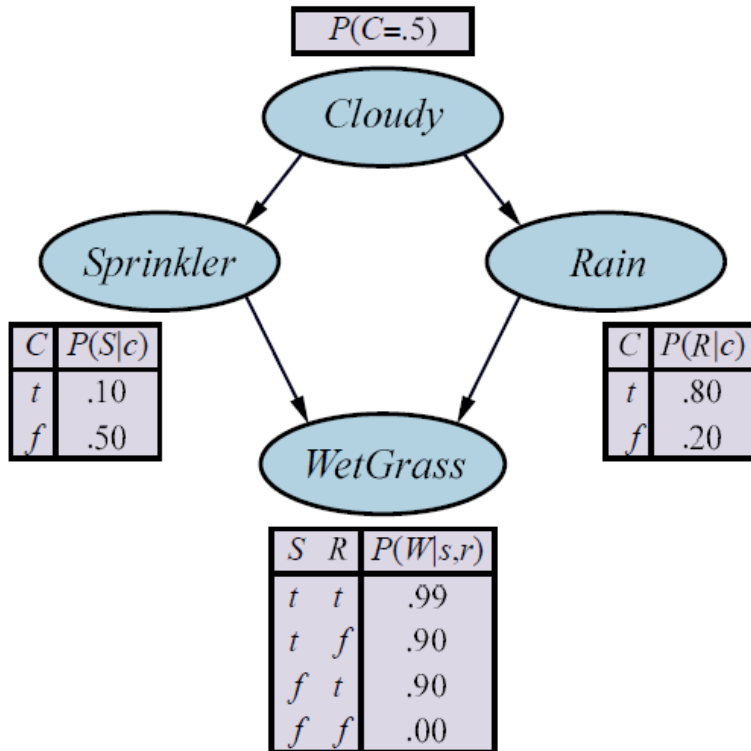
How does this work?

Weight of a Sample



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

Weight of a Sample

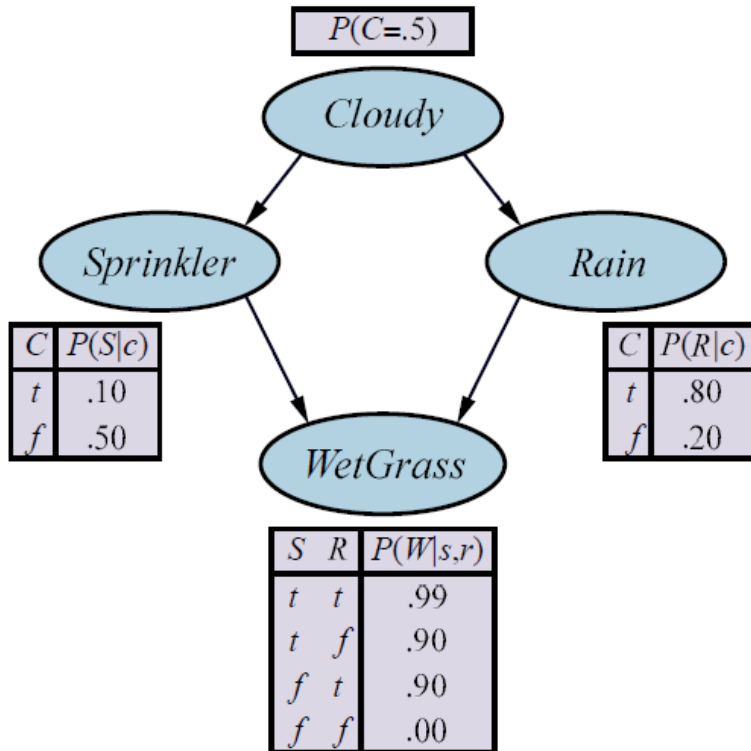


Query $P(\text{Rain} \mid \text{Cloudy} = \text{true}, \text{WetGrass} = \text{true})$

- Pick a topological order:

Cloudy, Sprinkler, Rain, WetGrass

Weight of a Sample



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

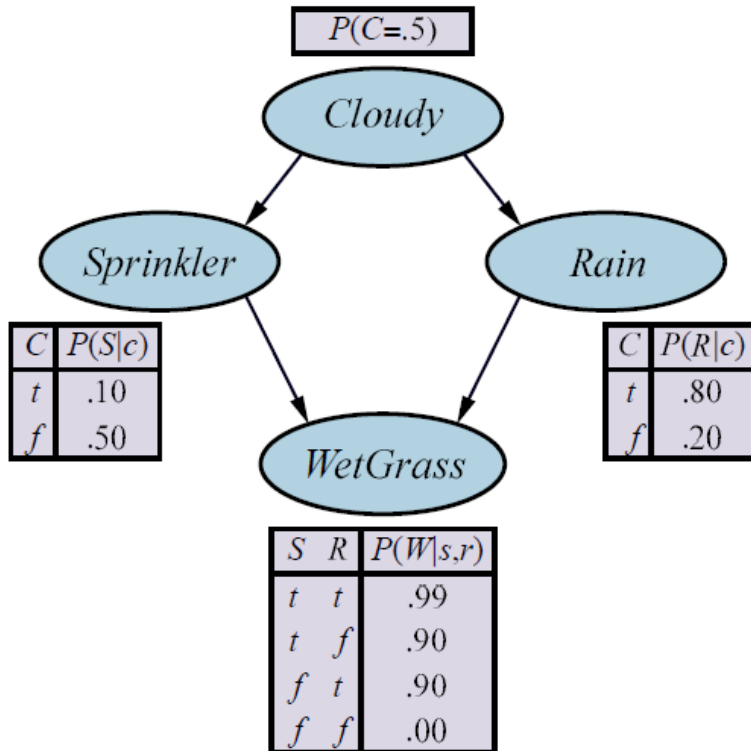
- Pick a topological order:

Cloudy, Sprinkler, Rain, WetGrass

evidence
variables

$$E = e$$

Weight of a Sample



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

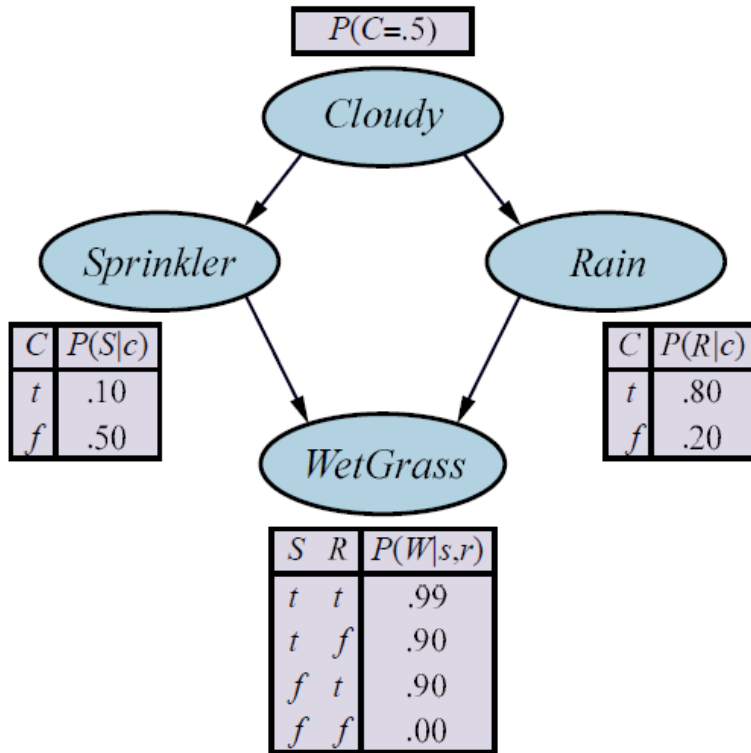
- Pick a topological order:

Cloudy, Sprinkler, Rain, WetGrass

evidence
variables
 $E = e$

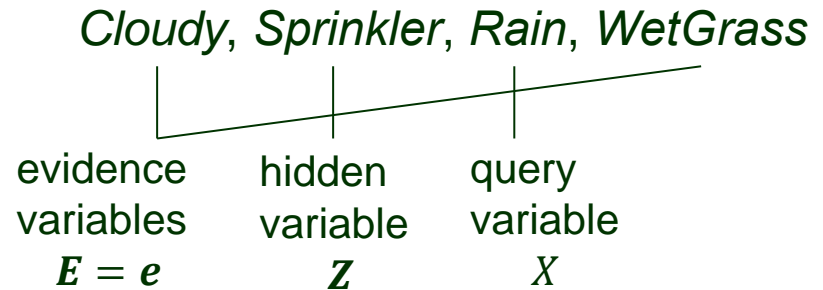
hidden
variable
 Z

Weight of a Sample

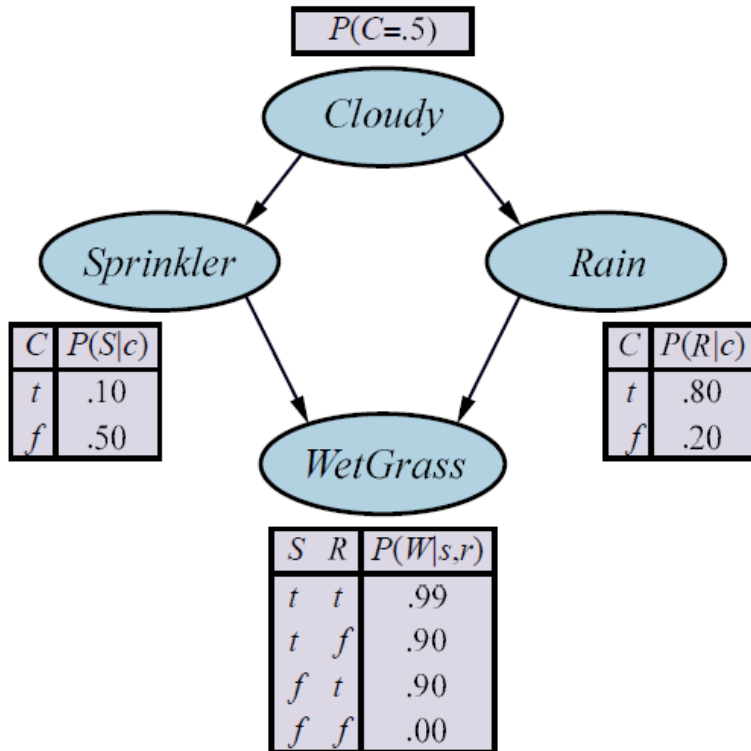


Query $P(\text{Rain} \mid \text{Cloudy} = \text{true}, \text{WetGrass} = \text{true})$

- Pick a topological order:



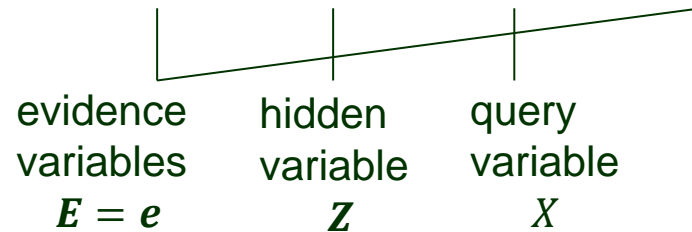
Weight of a Sample



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

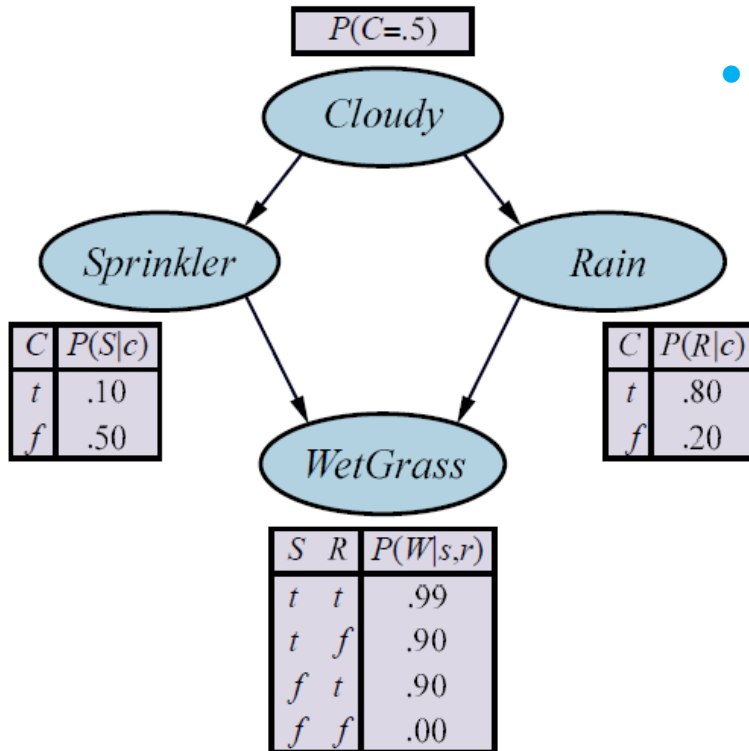
- Pick a topological order:

Cloudy, Sprinkler, Rain, WetGrass



- Set the weight $w = 1$.

(cont'd)

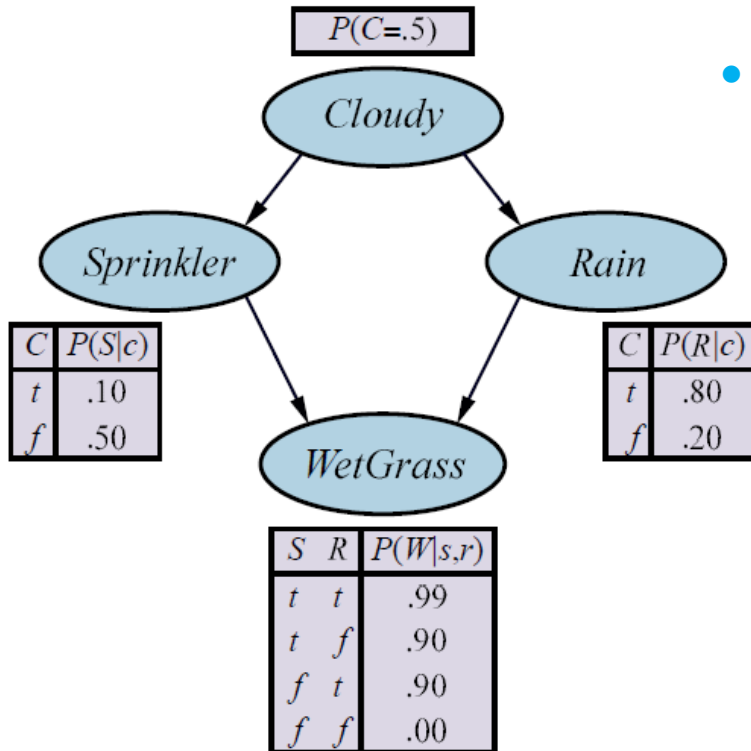


Query $P(\text{Rain} \mid \text{Cloudy} = \text{true}, \text{WetGrass} = \text{true})$

- Generate an event in the chosen topological order.

Order: *Cloudy, Sprinkler, Rain, WetGrass*

(cont'd)



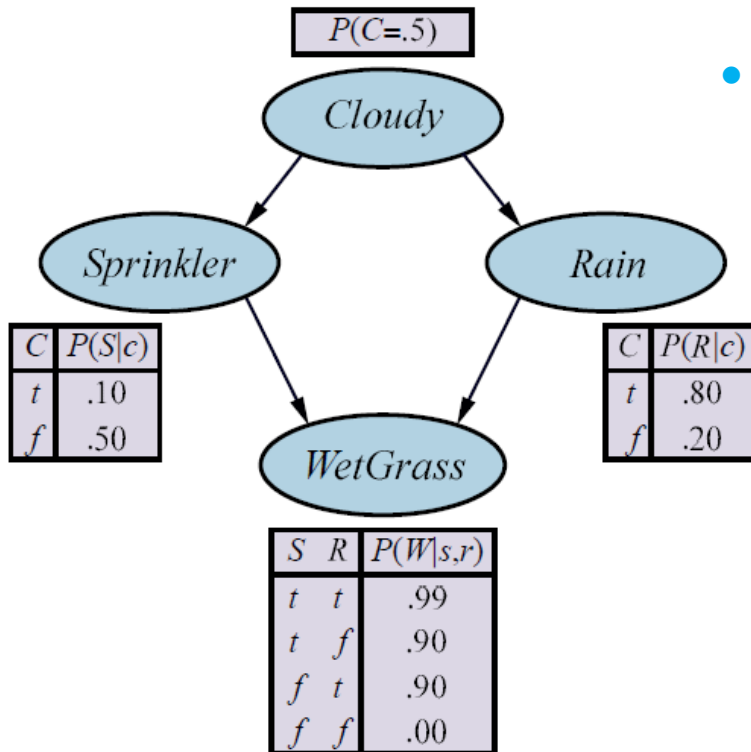
Query $P(\text{Rain} \mid \text{Cloudy} = \text{true}, \text{WetGrass} = \text{true})$

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

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

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

(cont'd)



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

- Generate an event in the chosen topological order.

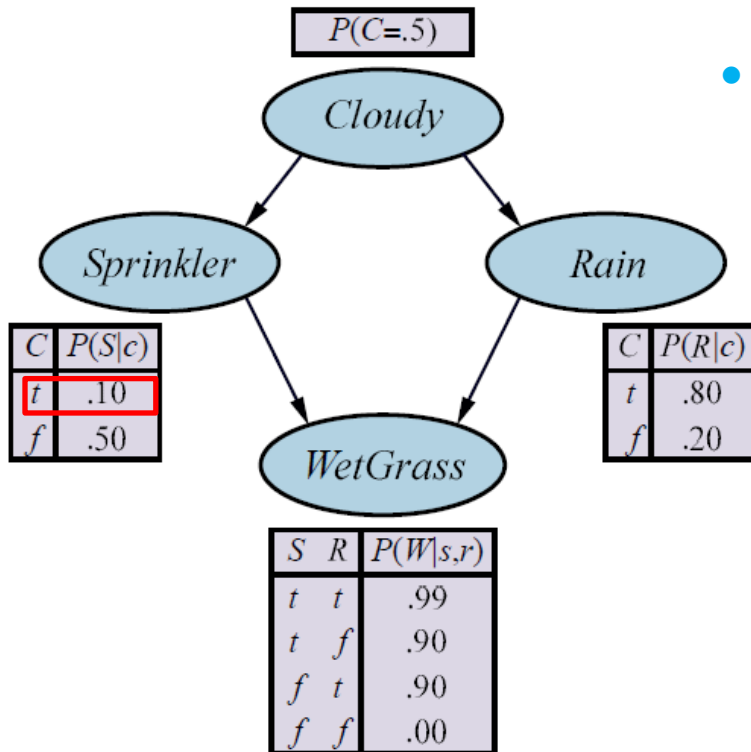
1. *Cloudy* is an evidence variable with value *true*.

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

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

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

(cont'd)



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

- Generate an event in the chosen topological order.

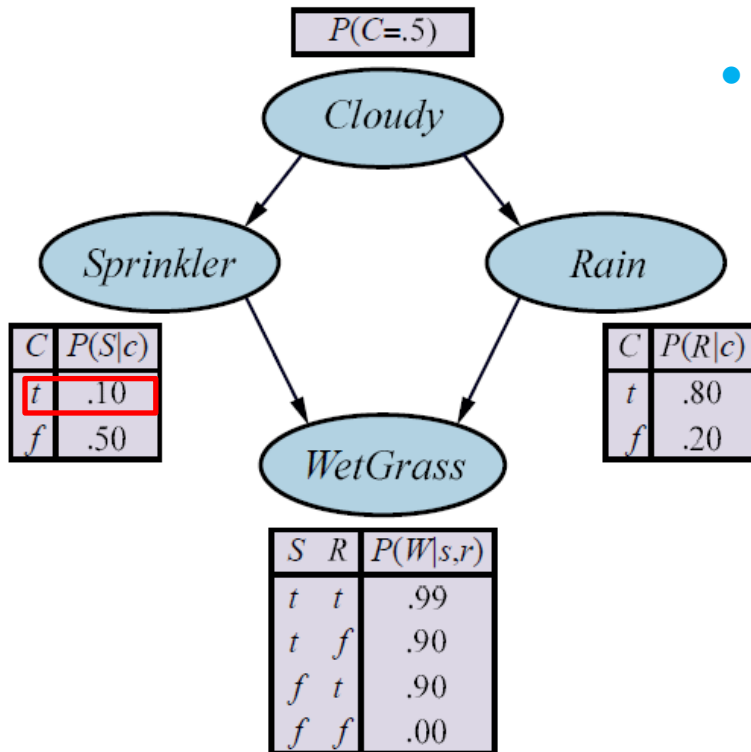
1. *Cloudy* is an evidence variable with value *true*.

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

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

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

(cont'd)



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

- Generate an event in the chosen topological order.

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

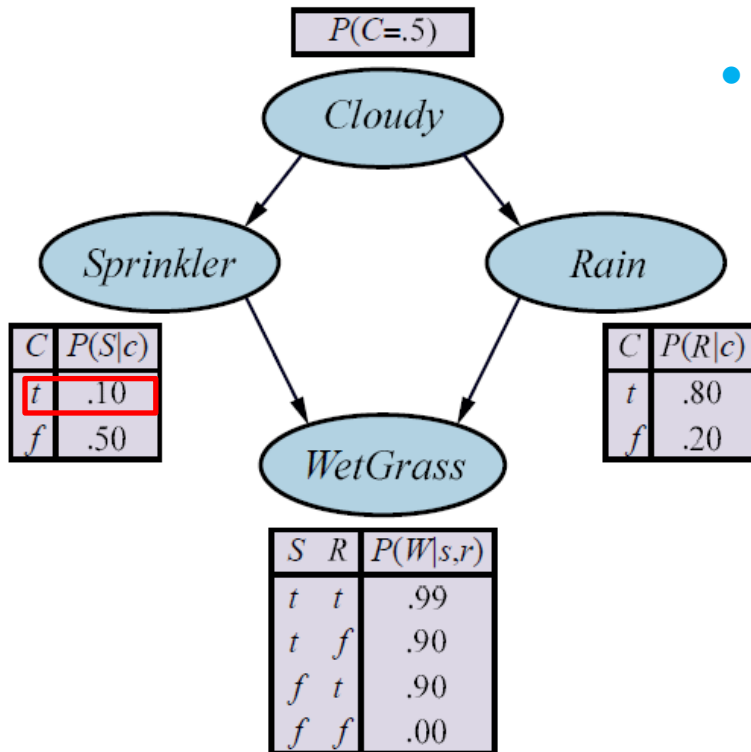
$$w \leftarrow w \times P(\text{Cloudy} = \text{true}) = 0.5$$

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

Sprinkler = *false* (returned value)

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

(cont'd)



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

- Generate an event in the chosen topological order.

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

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

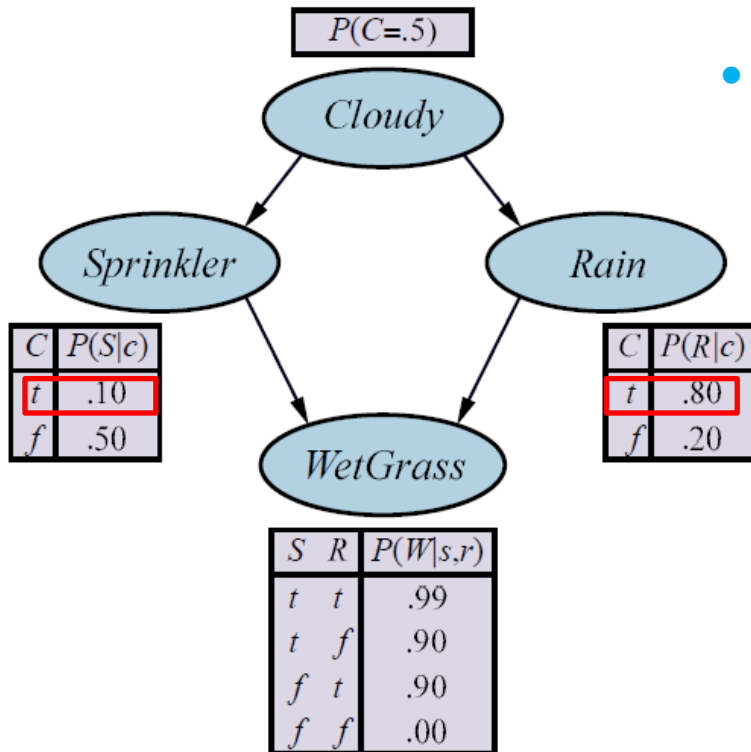
2. *Sprinkler* is not an evidence variable. Sample from $P(\text{Sprinkler} \mid \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle$.

Sprinkler = *false* (returned value)

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

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

(cont'd)



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

- Generate an event in the chosen topological order.

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

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

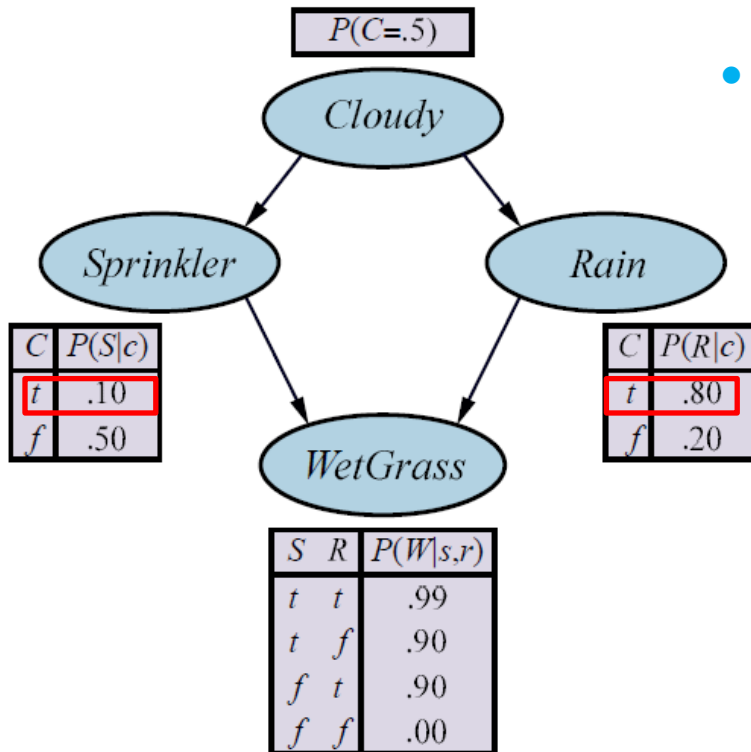
2. *Sprinkler* is not an evidence variable. Sample from $P(\text{Sprinkler} \mid \text{Cloudy} = \text{true}) = \langle 0.1, 0.9 \rangle$.

Sprinkler = false (returned value)

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

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

(cont'd)



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

- Generate an event in the chosen topological order.

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

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

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

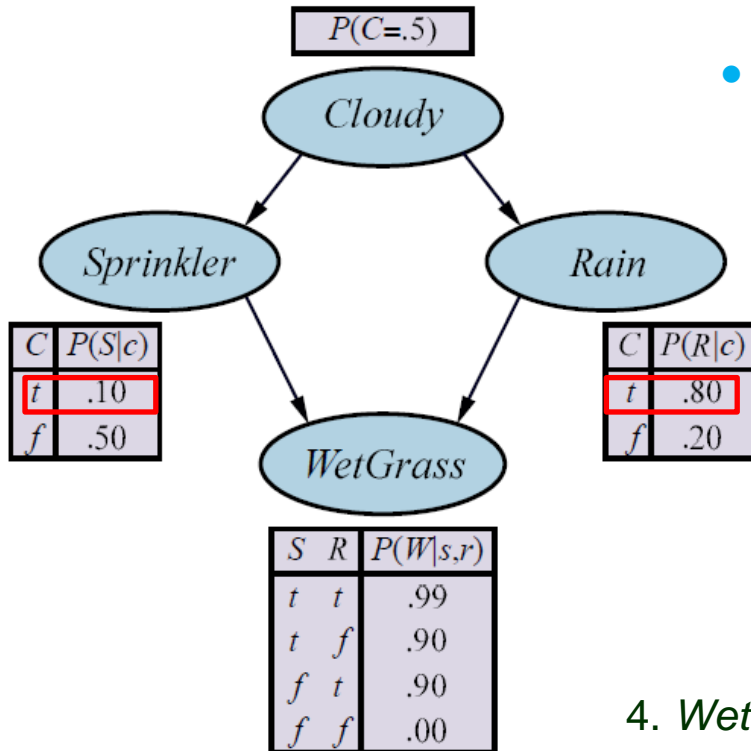
Sprinkler = false (returned value)

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

Rain = true

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

(cont'd)



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

- Generate an event in the chosen topological order.

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

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

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

Sprinkler = *false* (returned value)

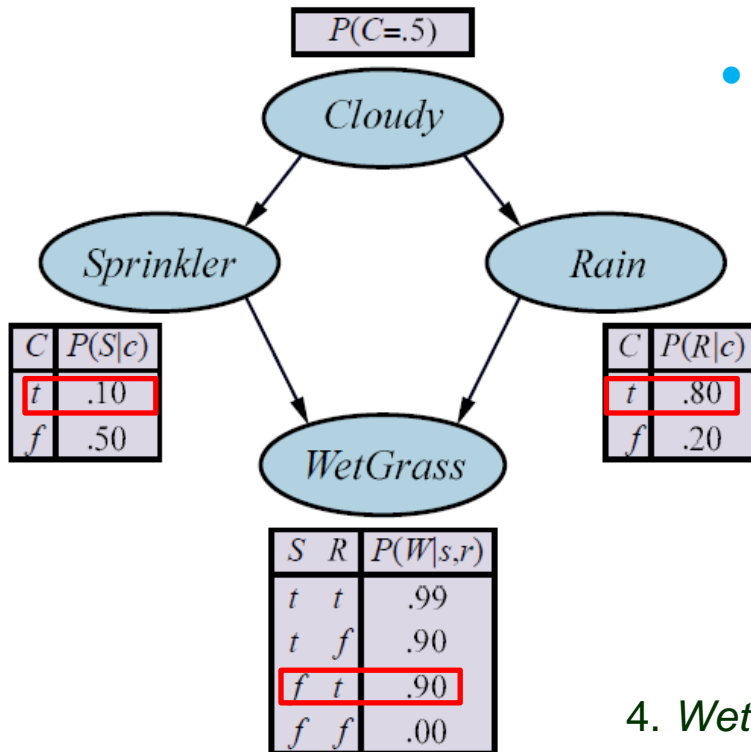
3. *Rain* is not an evidence variable. Sample from $P(\text{Rain} \mid \text{Cloudy} = \text{true}) = \langle 0.8, 0.2 \rangle$.

Rain = *true*

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

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

(cont'd)



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

- Generate an event in the chosen topological order.

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

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

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

Sprinkler = false (returned value)

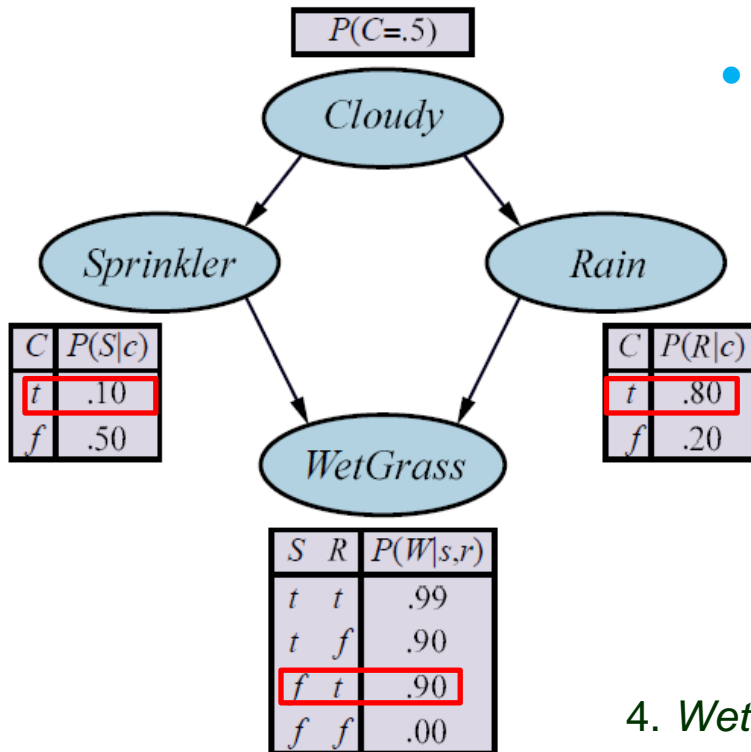
3. *Rain* is not an evidence variable. Sample from $P(\text{Rain} \mid \text{Cloudy} = \text{true}) = \langle 0.8, 0.2 \rangle$.

Rain = true

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

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

(cont'd)



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

- Generate an event in the chosen topological order.

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

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

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

Sprinkler = false (returned value)

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

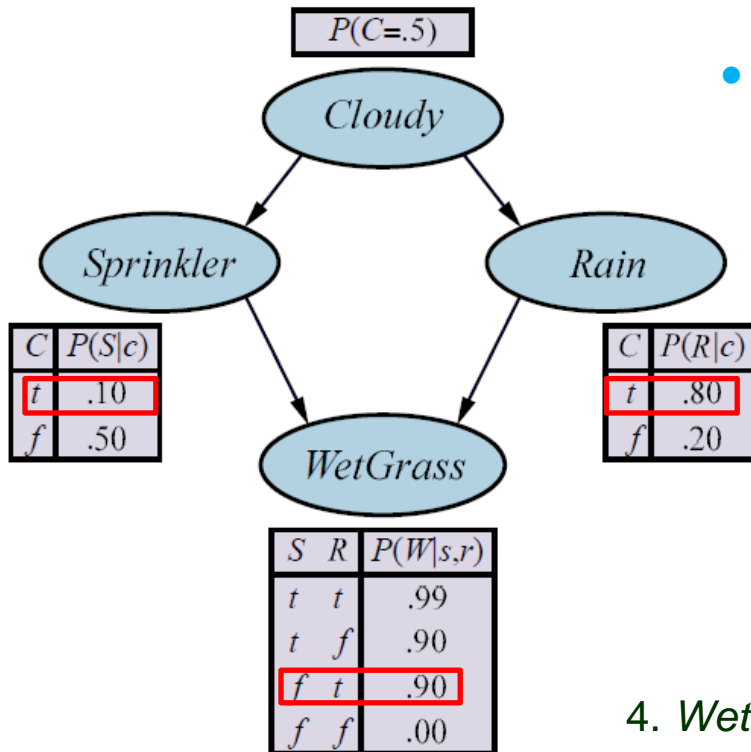
Rain = true

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

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

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

(cont'd)



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

- Generate an event in the chosen topological order.

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

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

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

Sprinkler = false (returned value)

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

Rain = true

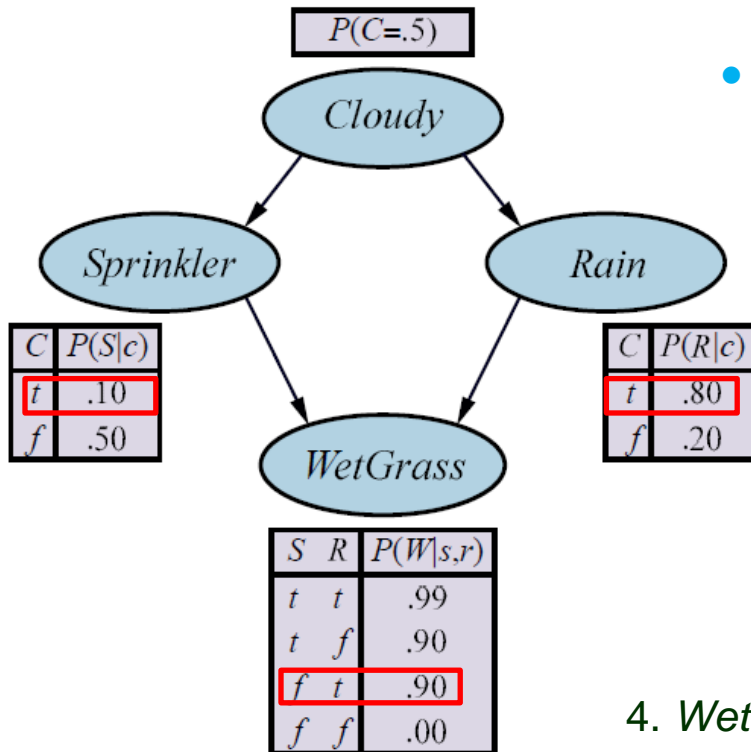
4. *WetGrass* is an evidence variable with value *true*.

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

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

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

(cont'd)



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

- Generate an event in the chosen topological order.

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

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

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

Sprinkler = *false* (returned value)

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

Rain = *true*

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

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

- 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

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

Likelihood Weighting

Fix the values for the evidence variables E and sample all the *nonevidence* variables Z (including X) in topological order, each conditioned on its parents.

Likelihood Weighting

Fix the values for the evidence variables E and sample all the *nonevidence* variables Z (including X) in topological order, each conditioned on its parents.

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

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

Likelihood Weighting

Fix the values for the evidence variables E and sample all the *nonevidence* variables Z (including X) in topological order, each conditioned on its parents.

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

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

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

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

Likelihood Weighting

Fix the values for the evidence variables E and sample all the *nonevidence* variables Z (including X) in topological order, each conditioned on its parents.

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

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

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

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

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

$$// \alpha = 1/P(\mathbf{e})$$

Likelihood Weighting

Fix the values for the evidence variables E and sample all the *nonevidence* variables Z (including X) in topological order, each conditioned on its parents.

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

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

The weight $w(z)$ must satisfy

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

$$\begin{aligned} \Downarrow \\ w(z) &= \frac{1}{P(e)} \cdot \frac{P(z, e)}{Q_{WS}(z)} = \alpha \frac{P(z, e)}{Q_{WS}(z)} && // \alpha = 1/P(e) \\ &= \alpha \frac{\prod_{i=1}^l P(z_i \mid \text{parents}(Z_i)) \cdot \prod_{i=1}^m P(e_i \mid \text{parents}(E_i))}{\prod_{i=1}^l P(z_i \mid \text{parents}(Z_i))} \end{aligned}$$

Likelihood Weighting

Fix the values for the evidence variables E and sample all the *nonevidence* variables Z (including X) in topological order, each conditioned on its parents.

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

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

The weight $w(z)$ must satisfy

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

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

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

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

Weighted Sampling

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

- The normalizing factor $\alpha = 1/P(\mathbf{e})$ is the same for all samples.
- Omitting α , the weight is set as the product of the conditional probabilities for the evidence variables given their parents.

Weighted Sampling

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

- The normalizing factor $\alpha = 1/P(\mathbf{e})$ is the same for all samples.
- Omitting α , the weight is set as the product of the conditional probabilities for the evidence variables given their parents.

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 $w \leftarrow w \times P(X_i = x_{ij} \mid \text{parents}(X_i))$

else $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid \text{parents}(X_i))$

return \mathbf{x}, w

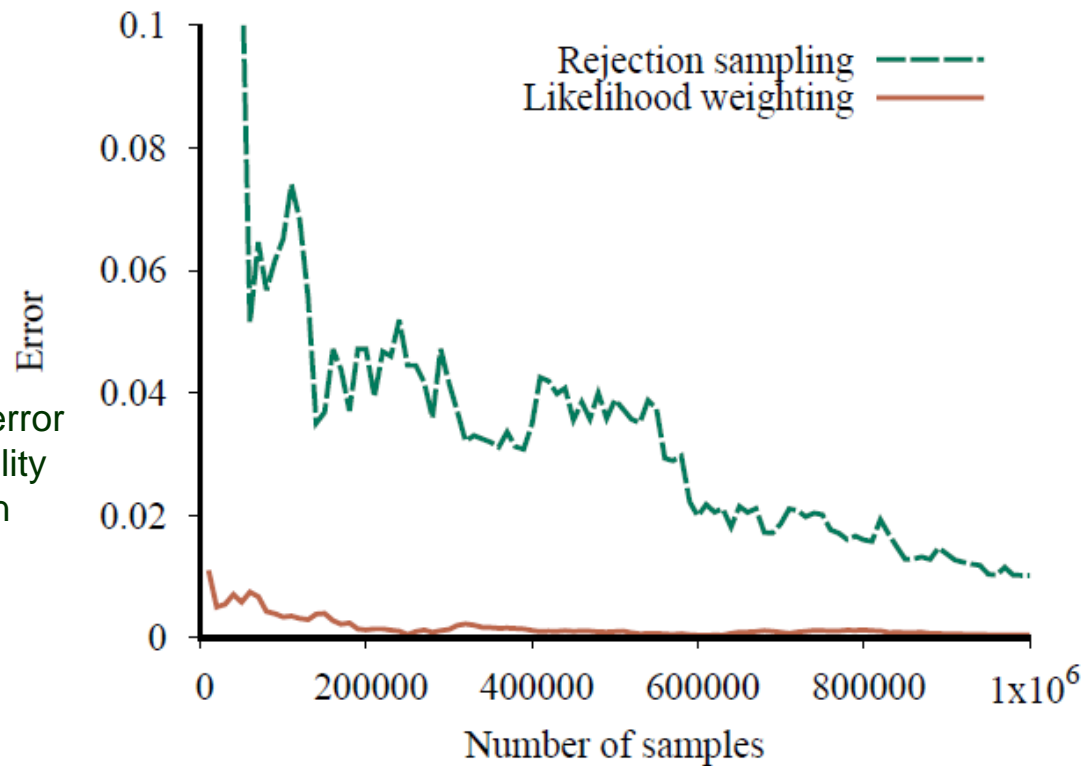
The Likelihood Weighting Algorithm

function LIKELIHOOD-WEIGHTING(X, \mathbf{e}, bn, N) **returns** an estimate of $\mathbf{P}(X | \mathbf{e})$
inputs: X , the query variable
 \mathbf{e} , observed values for variables \mathbf{E}
 bn , a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \dots, X_n)$
 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}) // sample \mathbf{x} with returned weight w
 $\mathbf{W}[j] \leftarrow \mathbf{W}[j] + w$ 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 $w \leftarrow w \times P(X_i = x_{ij} | \text{parents}(X_i))$
 else $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i | \text{parents}(X_i))$
 return \mathbf{x}, w

Performance Comparison

On the car insurance network



Maximum absolute error in any of the probability values for a query on *PropertyCost*

Likelihood weighting is considerably more efficient than rejection sampling.