Approximate Inference in Probabilistic Graphical Models

Luigi Portinale

University of Piemonte Orientale, Italy

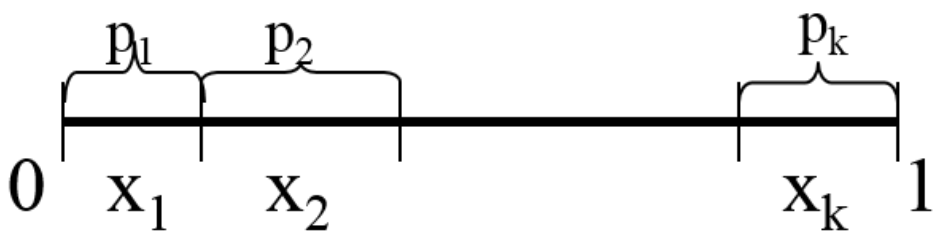
March 7-10, 2017

1. Inference by Sampling
   - Acceptance-Rejection Sampling
   - Importance Sampling
   - Markov Chain Monte Carlo
2. Sampling in Probabilistic Graphical Models
   - Logical Sampling
   - Likelihood Weighting
   - Gibbs Sampling

Inference by Sampling

Consider a discrete random variable with a finite number of possible states $X = \{x_1, \ldots x_k\}$ with $p_i = \mathcal{P}(X = x_i)$.

To sample the variable X from $\mathcal{P}$ means to assign X one of the possible values x_i according to the probability p_i (i.e., if we repeatedly sample the variable, the frequency of the value x_i must converge to the probability p_i)



Sampling procedure

- Generate a uniform random number between 0 and 1;
- set $X = x_i$ if the number is in the interval corresponding to x_i

- If variable X is continuous and $X \sim \mathbb{U}(0,1)$, then standard methods for pseudo-random generation exist. E.g., *Marsenne-Twister* (with period of $2^{19937}-1$)
- If the variable X is not $\mathbb{U}(0,1)$, then if the *cdf* is $F(x) = \mathcal{P}(X \leq x)$, then $X \equiv F^{-1}(\mathbb{U}(0,1)) \sim F$. Requires F to be invertible
  $F(x) = 1 - e^{-\lambda x} \rightarrow x = \frac{1}{\lambda}\log(1-z)$ with $Z \sim \mathbb{U}(0,1)$ (*exponential distribution*)
- Specific methods are also available for important distributions. E.g., *Box-Muller method* for $\mathcal{N}(0,1)$; let $U_1, U_2 \sim \mathbb{U}(0,1)$, then
  $Z = \sqrt{-2\log(U_1)}\cos(2\pi U_2) \sim \mathcal{N}(0,1)$
  Remember: if $Z \sim \mathcal{N}(0,1)$ then $\sigma Z + \mu \sim \mathcal{N}(\mu, \sigma^2)$

Monte Carlo Principle

Given a very large set X and a probability distribution $\mathcal{P}(X)$ over it, draw a set $X^{(1)}, \ldots X^{(N)}$ of i.i.d samples of X.

Approximate the distribution using such samples:

$$\mathcal{P}_N(X = x_i) = \frac{1}{N} \sum_{k=1}^{N} \mathbb{1}(X^{(k)} = x_i) \underset{N \to \infty}{\longrightarrow} \mathcal{P}(X = x_i)$$

Computing expectations:

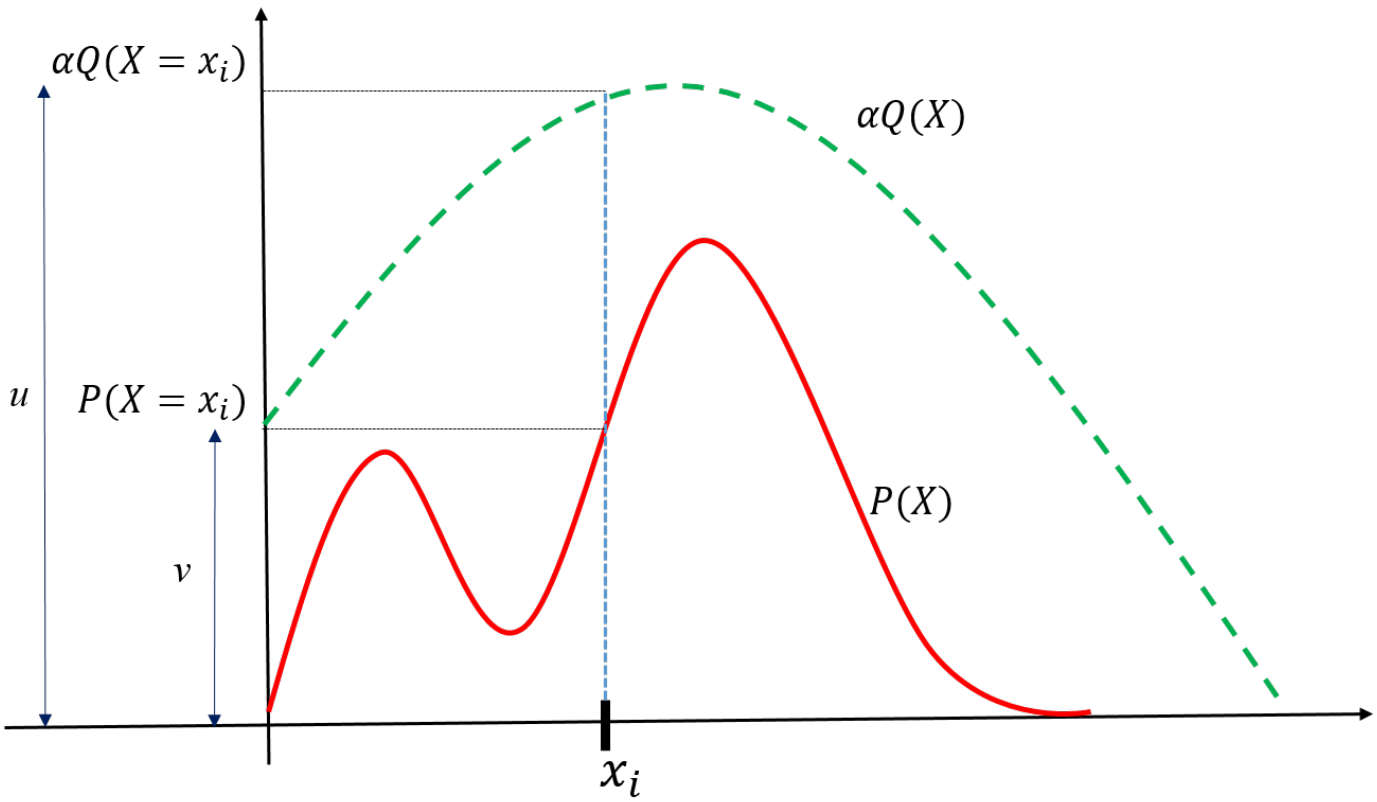
$$\mathbb{E}_N(f) = \frac{1}{N} \sum_{k=1}^{N} f(X^{(k)}) \underset{N \to \infty}{\longrightarrow} \mathbb{E}(f) = \sum_{X} f(X)\mathcal{P}(X)$$

$$(\int_X f(X)\mathcal{P}(X) \text{ in a continuous space})$$

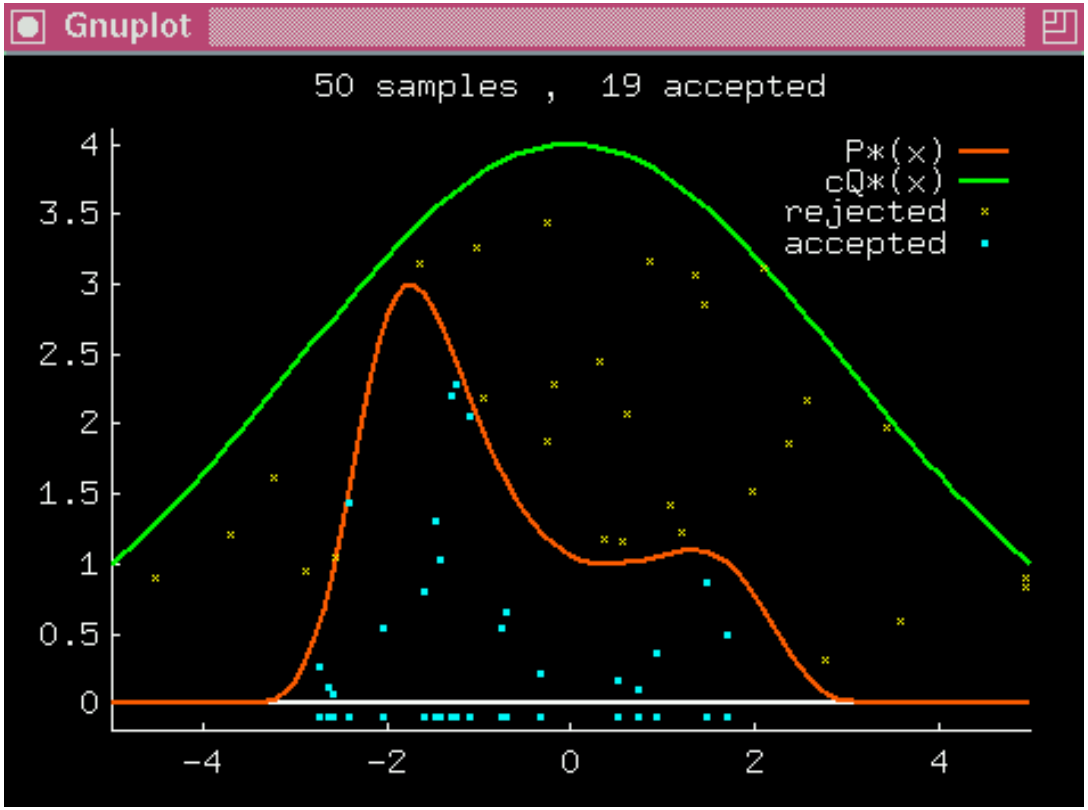
Acceptance-Rejection Sampling

Acceptance-Rejection Sampling

- When is too complicated to sample directly from distribution $\mathcal{P}$, we can sample from a simpler distribution $\mathcal{Q}$ called the proposal distribution
- The proposal distribution $\mathcal{Q}$ must satisfy the following: $\mathcal{P}(X) \leq \alpha \mathcal{Q}(X)$ for some $\alpha < \infty$
- We sample a candidate $X = x_i$ from $\mathcal{Q}$ and we *accept* the candidate with probability $\mathcal{A}(x_i) = \frac{\mathcal{P}(x_i)}{\alpha \mathcal{Q}(x_i)}$
- Result has a distribution $\sim \mathcal{Q}(x)\mathcal{A}(x) = \frac{\mathcal{P}(x)}{\alpha} \propto \mathcal{P}(x)$



$$\mathcal{P}_{accept}(x_i) \propto \frac{v}{u} = \frac{\mathcal{P}(X = x_i)}{\alpha \mathcal{Q}(X = x_i)}$$



Remarks

It works well when $\mathcal{P}$ and $\mathcal{Q}$ are similar.

If α is too large, then we rarely accept samples

In high dimensional space you have too much to sample from (many rejections)

We should avoid to sample in regions with low values of $\mathcal{P}$

Importance Sampling

- In case of rare events (function values belonging to regions with low probability) a lot of samples are unuseful
- Idea: to sample from a proposal where the event is not rare and always accept the sample; however, suitably weight and adjusts for the introduced bias.
- The resulting strategy is called Importance Sampling

Importance Sampling

Let $\mathcal{Q}(x)$ be a proposal distribution, $\mathcal{P}(x)$ the original distribution from which we want to sample and $f(x)$ the function whose expected value has to be computed (we require $f(x)\mathcal{P}(x) \neq 0$).

$$\mathbb{E}(f) = \sum_x f(x)\mathcal{P}(x) = \sum_x \frac{f(x)\mathcal{P}(x)}{\mathcal{Q}(x)}\mathcal{Q}(x)$$

$$\mathbb{E}_N(f) = \frac{1}{N}\sum_{k=1}^{N} \frac{f(X^{(k)})\mathcal{P}(X^{(k)})}{\mathcal{Q}(X^{(k)})} \quad X^{(k)} \sim \mathcal{Q}$$

$w^{(k)} = \frac{\mathcal{P}(X^{(k)})}{\mathcal{Q}(X^{(k)})}$ is the importance weight of the sample (the bias introduced by sampling from the proposal)

Finding a good importance sampler is an art and a science

Importance Sampling

Let $\mathcal{Q}(x)$ be a proposal distribution, $\mathcal{P}(x)$ the original distribution from which we want to sample and $f(x)$ the function whose expected value has to be computed (we require $f(x)\mathcal{P}(x) \neq 0$).

$$\mathbb{E}(f) = \sum_x f(x)\mathcal{P}(x) = \sum_x \frac{f(x)\mathcal{P}(x)}{\mathcal{Q}(x)}\mathcal{Q}(x)$$

$$\mathbb{E}_N(f) = \frac{1}{N}\sum_{k=1}^{N} \frac{f(X^{(k)})\mathcal{P}(X^{(k)})}{\mathcal{Q}(X^{(k)})} \quad X^{(k)} \sim \mathcal{Q}$$

$w^{(k)} = \frac{\mathcal{P}(X^{(k)})}{\mathcal{Q}(X^{(k)})}$ is the importance weight of the sample (the bias introduced by sampling from the proposal)

Finding a good importance sampler is an art and a science

Normalized Importance Sampling

Frequently, target distribution $\mathcal{P}$ is known up to a normalizing constant i.e., $\tilde{\mathcal{P}}(X) = Z\mathcal{P}(X)$ (e.g., we know $\mathcal{P}(X, \mathbf{e})$ but need $\mathcal{P}(X|\mathbf{e})$, or we have unormalized product of cliques in a MRF)

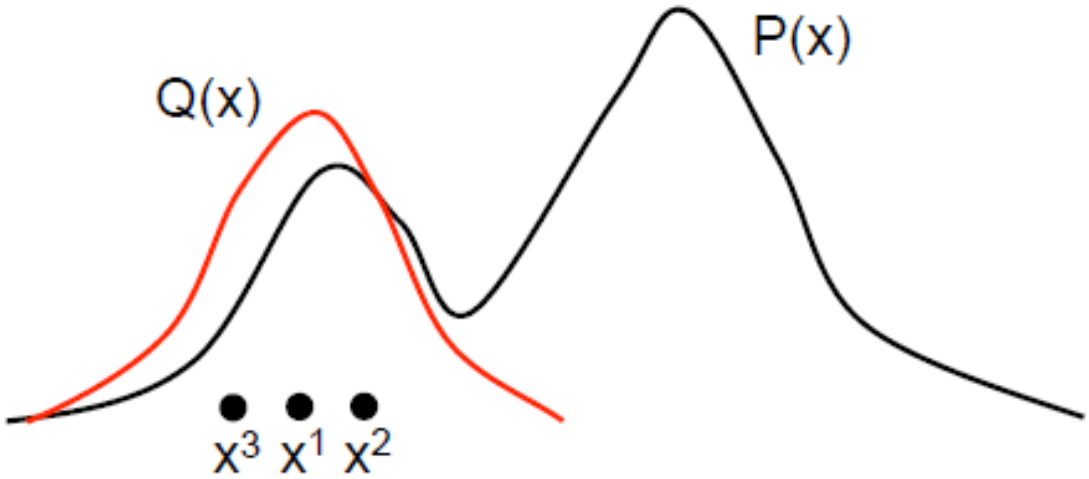
- Define $w(X) = \frac{\tilde{\mathcal{P}}(X)}{\mathcal{Q}(X)}$
- The expected value wrt $\mathcal{Q}$ of w is
  $\mathbb{E}_{\mathcal{Q}}(w(X)) = \sum_x \mathcal{Q}(x)\frac{\tilde{\mathcal{P}}(X)}{\mathcal{Q}(X)} = \sum_x \tilde{\mathcal{P}}(X) = Z$
- $\mathbb{E}_{\mathcal{P}}(f(X)) = \sum_x \mathcal{P}(x)f(x) = \sum_x \mathcal{Q}(x)f(x)\frac{\mathcal{P}(x)}{\mathcal{Q}(x)}$
  $= \frac{1}{Z}\sum_x \mathcal{Q}(x)f(x)\frac{\tilde{\mathcal{P}}(x)}{\mathcal{Q}(x)} = \frac{1}{Z}\mathbb{E}_{\mathcal{Q}}(f(X)w(X))$
  $= \frac{\mathbb{E}_{\mathcal{Q}}(f(X)w(X))}{\mathbb{E}_{\mathcal{Q}}(w(X))}$

Thus to estimate $\mathbb{E}_{\mathcal{P}}(f(X))$ we can compute

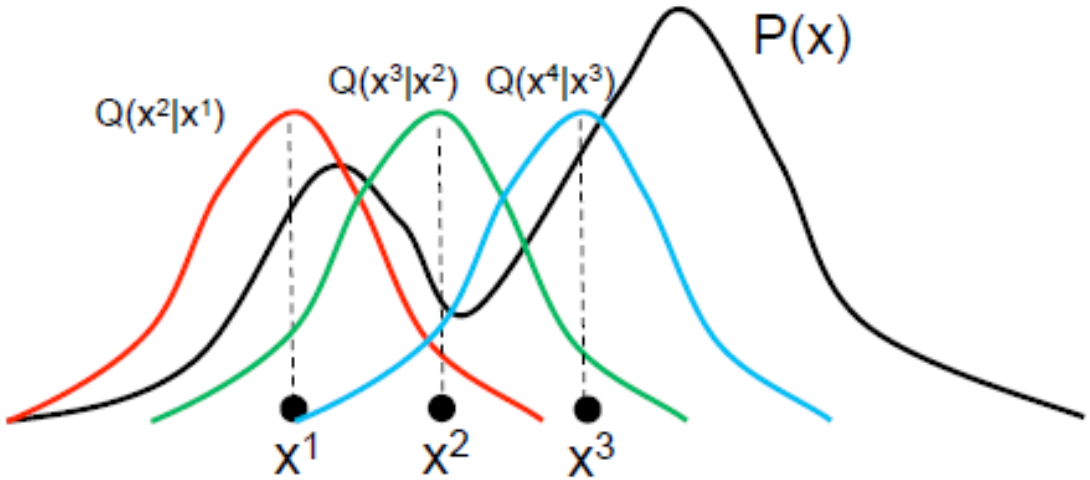
$$\frac{\sum_{k=1}^{N} f(X^{(k)})w(X^{(k)})}{\sum_{k=1}^{N} w(X^{(k)})} \quad X^{(k)} \sim \mathcal{Q}$$

Markov Chain Monte Carlo

- The choice of a proposal distribution is crucial both in rejection and importance sampling.
- Idea: instead of using a fixed proposal distribution, choose an adaptive one.
- Consider the sampling process as visiting a set of states: each state has an associated proposal distribution that depends on the previous state.
- Sampling process as a Markov Chain!



Importance sampling with a bad proposal



An adaptive proposal sampling

A digression

Markov Chain

A Markov Chain is a sequence od random variables indexed by a parameter t called the *time*: $X^{(1)}, X^{(2)}, \ldots X^{(t)} \ldots$ satisfying the *Markov property*

$$\mathcal{P}(X^{(t)} = x' | X^{(1)}, X^{(2)}, \ldots X^{(t-1)}) = \mathcal{P}(X^{(t)} = x' | X^{(t-1)})$$

If parameter t is discrete we have a DTMC (Discrete Time Markov Chain), otherwise it is called CTMC (Continuous Time Markov Chain).

We consider only DTMCs and $X^{(t)}$ will be the t-th sample (the entire set of all the variables in case of a PGM)

$\mathcal{P}(X^{(t)} = x' | X^{(t-1)})$ is called the *transition kernel*

- We consider *Homogeneouos Markov Chain* where transition kernel is independent from time

$$\mathcal{P}(X^{(t)} = x' | X^{(t-1)} = x) = \mathcal{T}(x'|x)$$

where x is the *previous state* and x' the *next state*

- If $\pi^t(x)$ is the probability distribution over the states of the MC at time t (the possible values of variable $X^{(t)}$), then

$$\pi^{t+1}(x') = \sum_x \pi^t(x)\mathcal{T}(x'|x)$$

- A distribution is stationary if it does not change under transitions

$$\pi(x') = \sum_x \pi(x)\mathcal{T}(x'|x) \text{ for all } x'$$

- Irreducible: a MC where any state x' can be reach from any state x in a finite number of steps

$$\mathcal{P}(X^{(t+n)} = x' | X^{(t)} = x) > 0 \text{ for a finite } t + n$$

- Aperiodic: a MC where any state can be reached at any time (no period)
- Ergodic: a MC that is irreducible and aperiodic.
- Regular: a MC with a *regular* transition matrix (i.e., a matrix P such that some power of the matrix P^n has only positive entries);if a MC is regular then it is ergodic (vice versa is not true)

Existence of stationary distribution

If a MC is *ergodic*, then it has a unique stationary distribution which is independent from the initial state

Reversibility

The stationary distribution satisfies the *detailed balance condition* or *reversibility*:

$$\pi(x')\mathcal{T}(x|x') = \pi(x)\mathcal{T}(x'|x)$$

MCMC

Given a desired distribution $\mathcal{P}(x)$, we can build a MC such that $\mathcal{P}(x)$ is the stationary distribution; by simulating the MC, once we have reached the stationary distribution, we can take sample from it.

Let us introduce a first MCMC algorithm:

Metropolis Algorithm

- Consider a symmetric proposal distribution $\mathcal{Q}(x'|x) = \mathcal{Q}(x|x')$ over the considered space X and an initial state $x = x^0$.
- Draw a sample x' from the proposal distribution and the current state x.
- Accept the sample (set x' to the current state) with probability $\mathcal{A}(x'|x) = \min(1, r)$, where $r = \frac{\mathcal{P}(x')}{\mathcal{P}(x)}$
- repeat N times, by discarding the first K runs (burn-in phase)

- samples are always accepted if more probable than the current one ($r > 1$)
- only need to compute $\frac{\mathcal{P}(x')}{\mathcal{P}(x)}$ rather than $\mathcal{P}(x)$ and $\mathcal{P}(x')$ separately.

Analysis of Metropolis Algorithm

- The transition kernel of Metropolis Algorithm is $\mathcal{T}(x'|x) = \mathcal{Q}(x'|x)\frac{\mathcal{P}(x')}{\mathcal{P}(x)}$ if $\mathcal{P}(x) > \mathcal{P}(x')$ and $\mathcal{T}(x'|x) = \mathcal{Q}(x'|x)$ otherwise.
- Let $\mathcal{P}(x') \geq \mathcal{P}(x)$
- $\mathcal{P}(x)\mathcal{T}(x'|x) = \mathcal{P}(x)\mathcal{Q}(x'|x)$
  $= \mathcal{P}(x)\mathcal{Q}(x|x')$ (symmetric proposal)
  $= \mathcal{P}(x')\mathcal{Q}(x|x')\frac{\mathcal{P}(x)}{\mathcal{P}(x')}$
  $= \mathcal{P}(x')\mathcal{T}(x|x')$
- The above is the detailed balance equation showing that $\mathcal{P}(x)$ is the stationary distribution of the generated MC

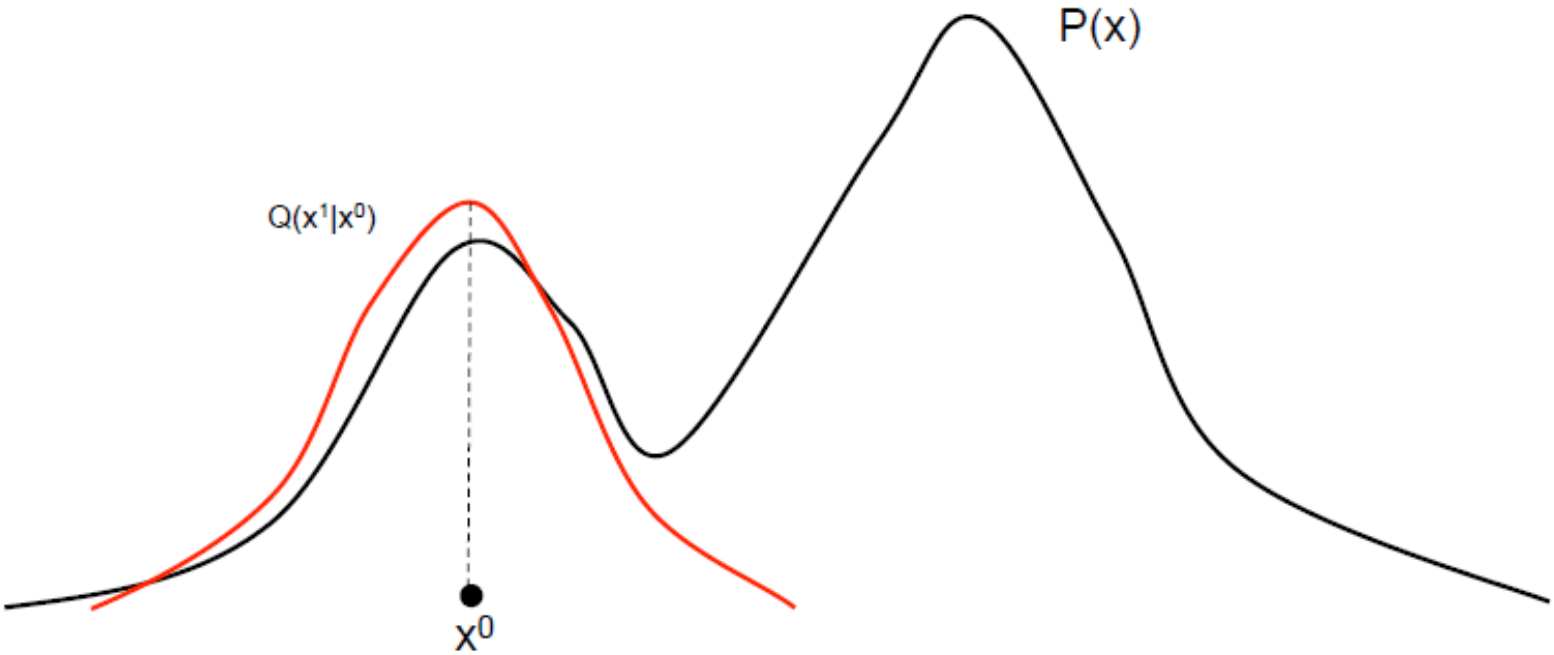
Another MCMC algorithm:

Metropolis-Hasting Algorithm

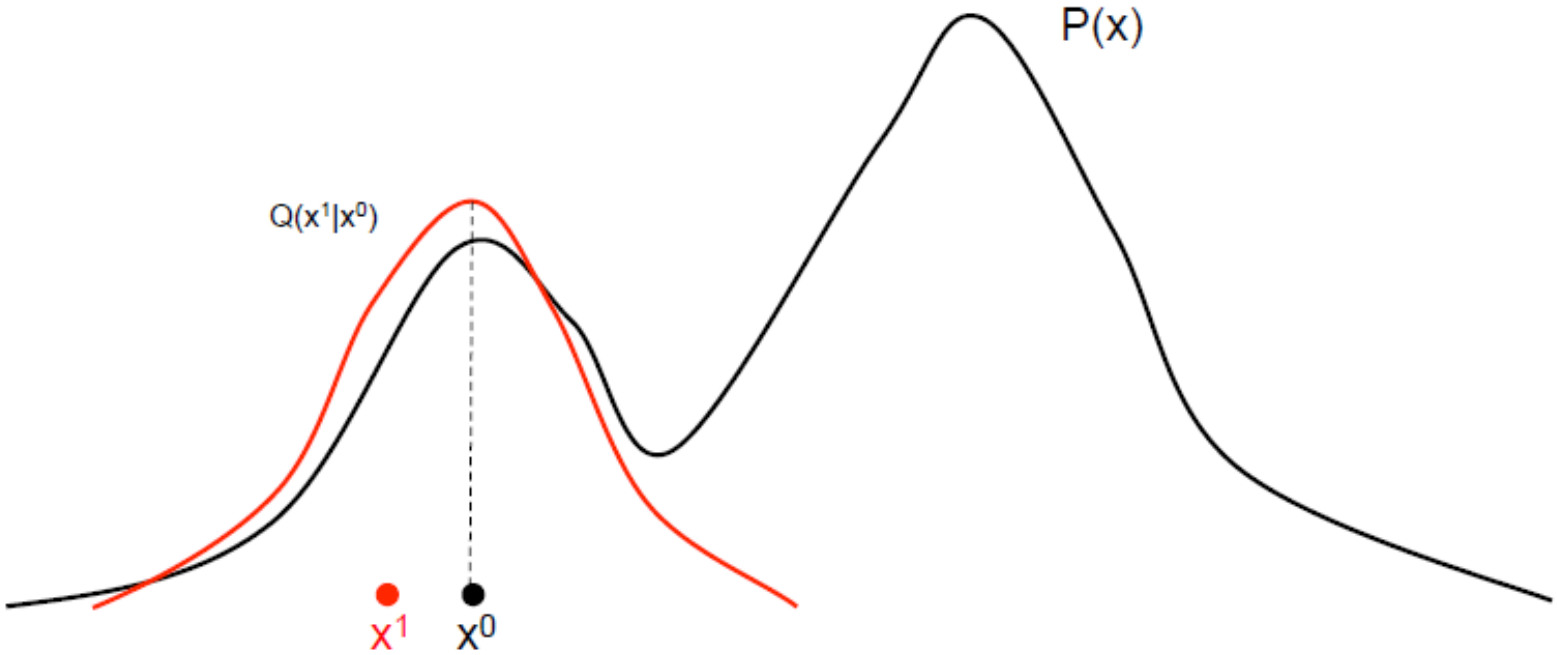
- Consider a proposal distribution $\mathcal{Q}(x'|x)$ over the considered space X and an initial state $x = x^0$.
- Draw a sample x' from the proposal distribution and the current state x.
- Accept the sample (set x' to the current state) with probability $\mathcal{A}(x'|x) = \min(1, r)$, where $r = \frac{\mathcal{P}(x')\mathcal{Q}(x|x')}{\mathcal{P}(x)\mathcal{Q}(x'|x)}$
- repeat N times, by discarding the first K runs (burn-in phase)

- **N.B.** $\frac{\mathcal{P}(x')}{\mathcal{Q}(x'|x)}$ is the importance weigth of x'.
- $\mathcal{A}(x'|x)$ is the ratio of the importance weights of x' and x.
- do not require a symmetric proposal ansd (again) only need to compute $\frac{\mathcal{P}(x')}{\mathcal{P}(x)}$.

Proposal: Gaussian distribution centered on x

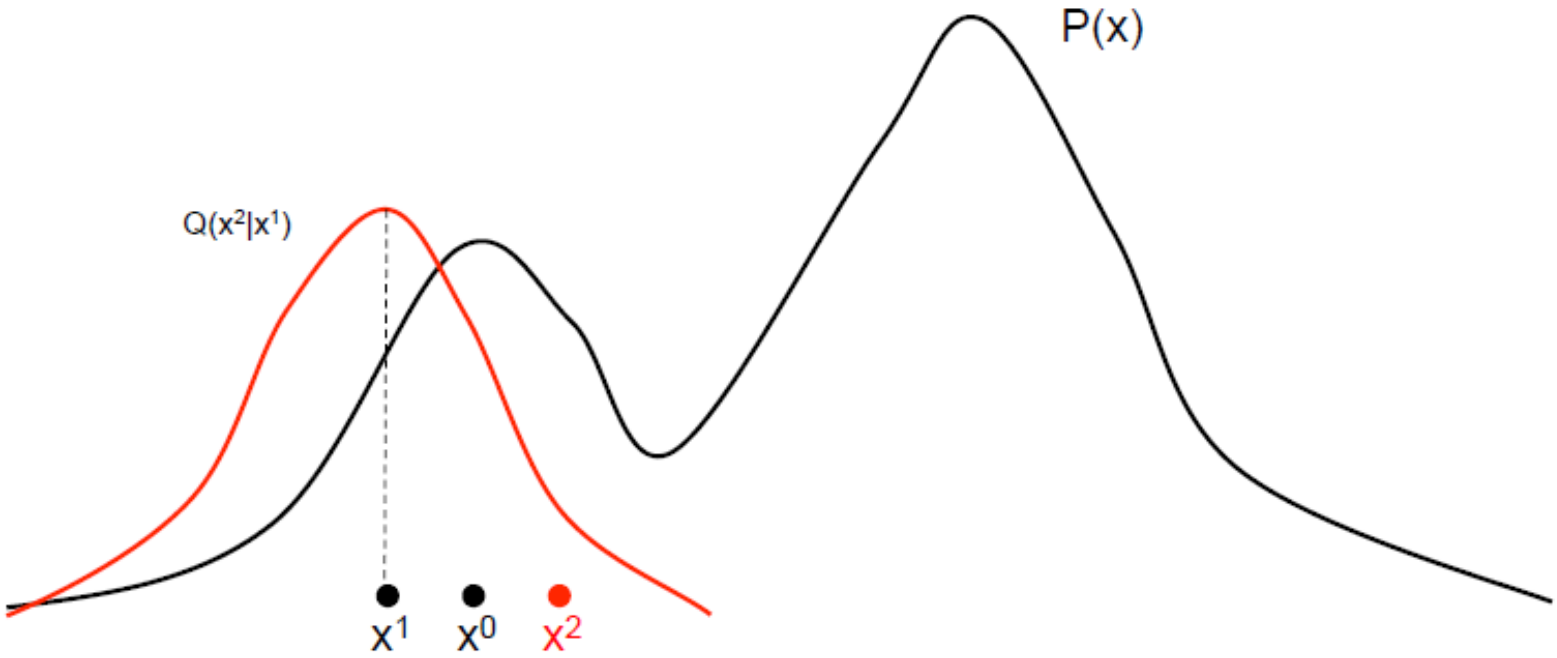


Initialize x^0



Initialize x^0

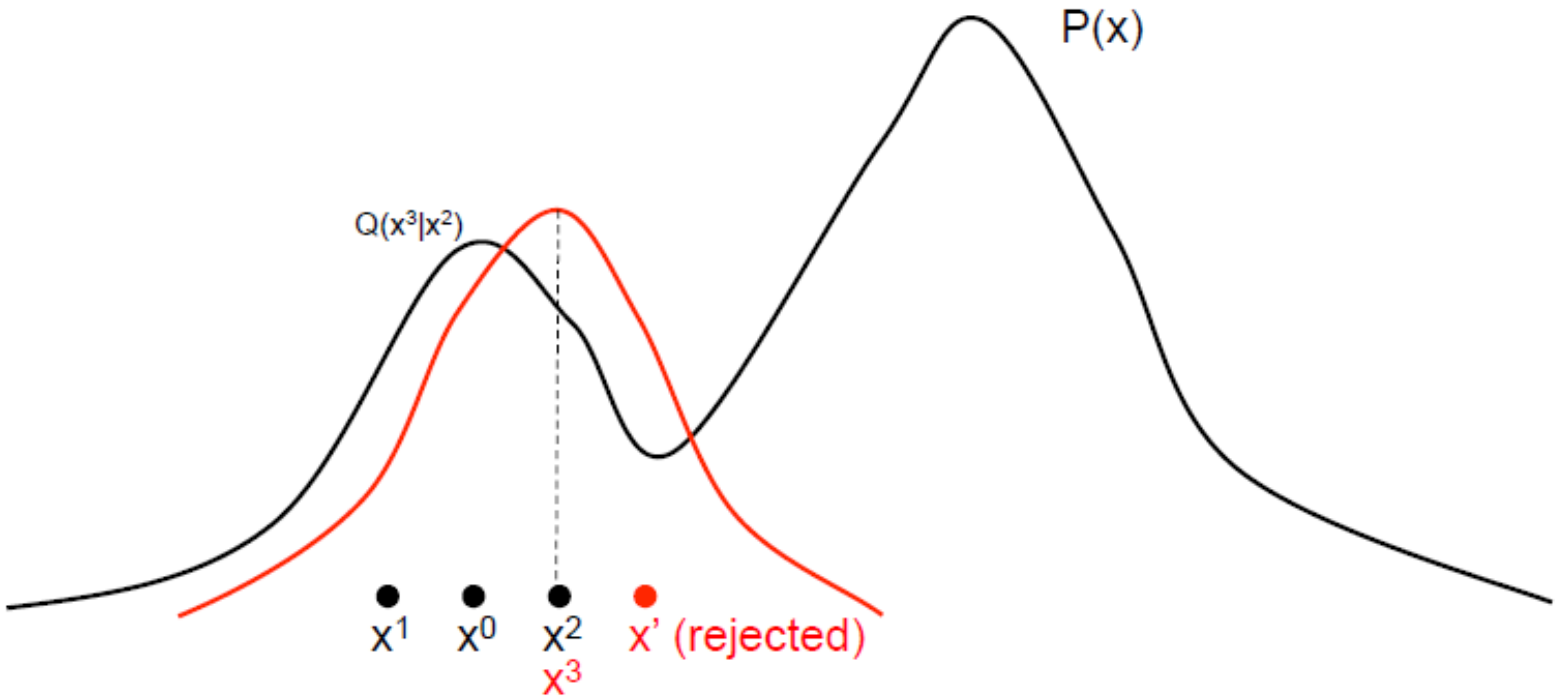
Draw and accept x^1



Initialize x^0

Draw and accept x^1

Draw and accept x^2



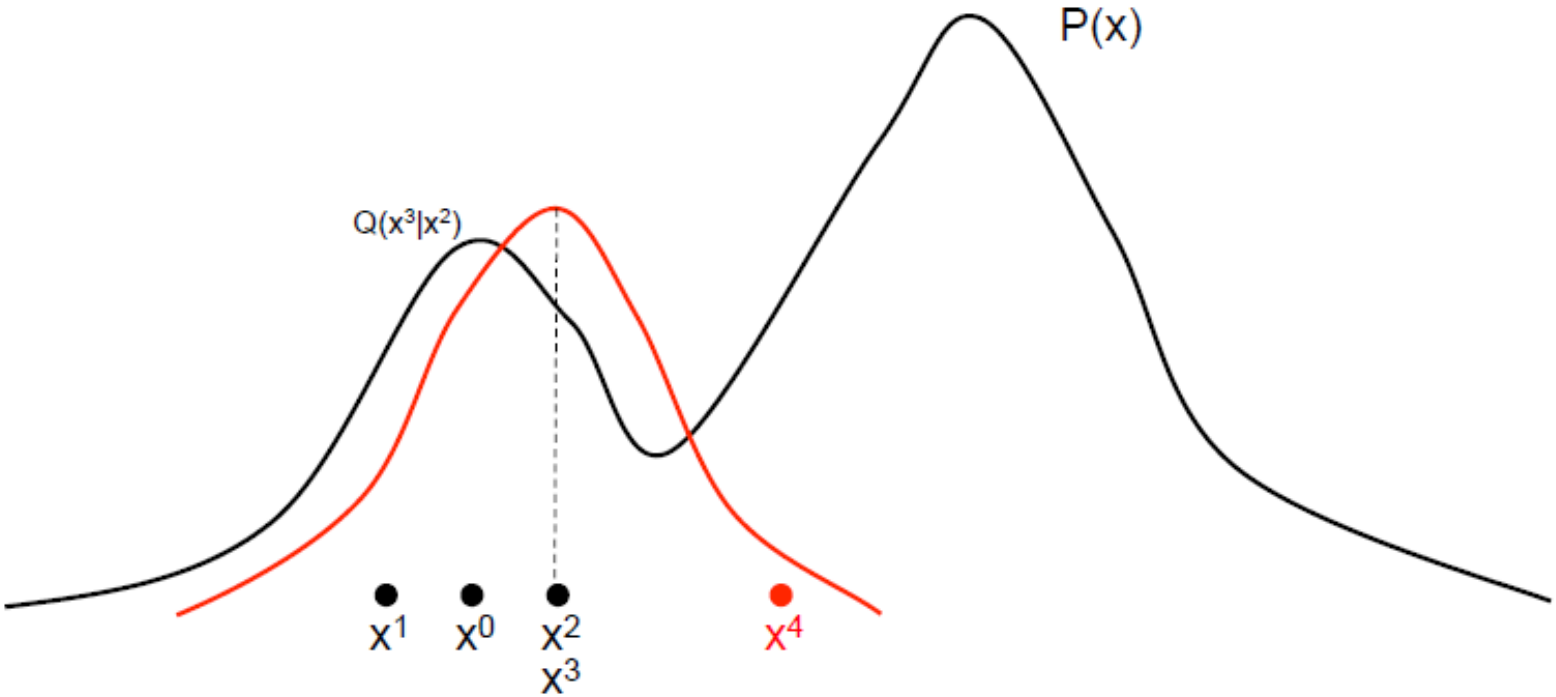
Initialize x^0

Draw and accept x^1

Draw and accept x^2

Draw x', reject and set $x^3 = x^2$

We reject because $\frac{\mathcal{P}(x')}{\mathcal{P}(x^2)}$ is very small so $\mathcal{A}(x'|x^2)$ is close to 0



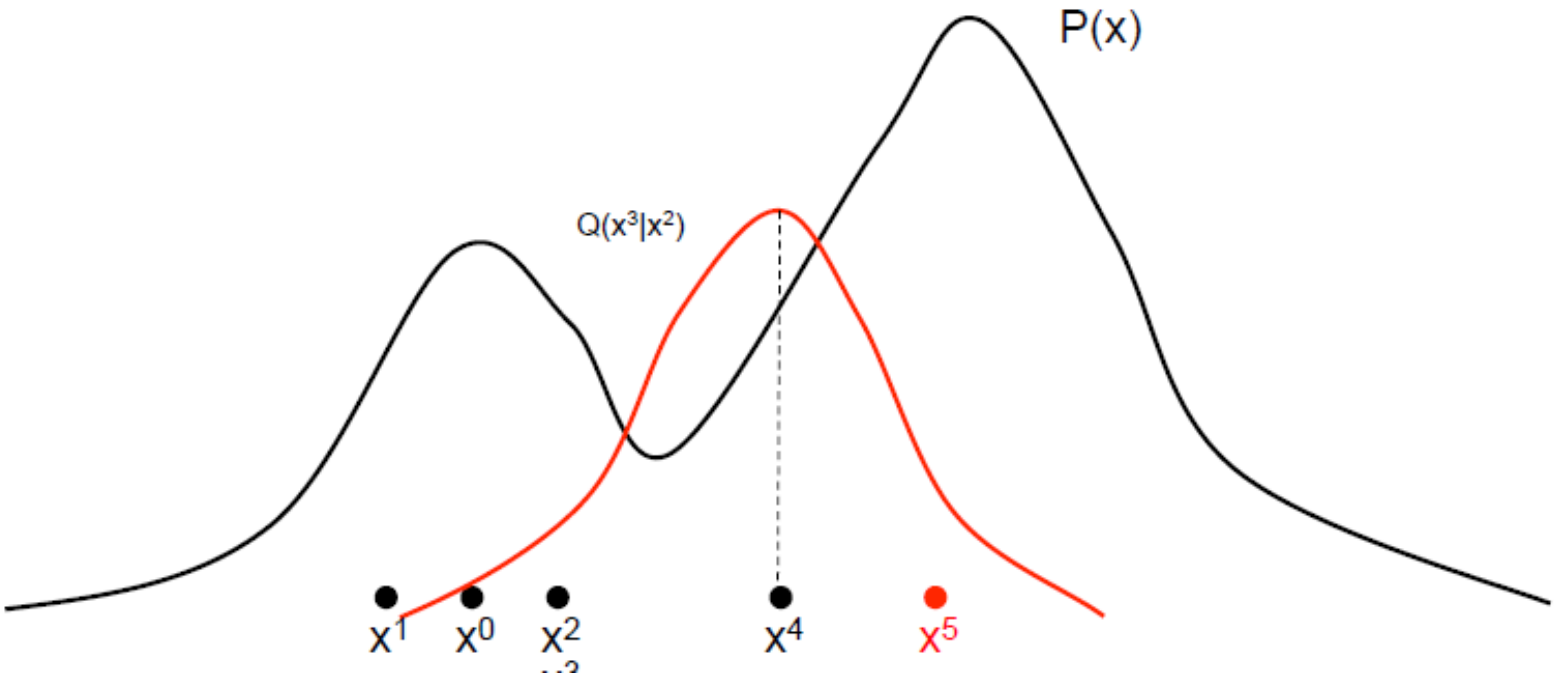
Initialize x^0

Draw and accept x^1

Draw and accept x^2

Draw x', reject and set $x^3 = x^2$

Draw and accept x^4



Initialize x^0

Draw and accept x^1

Draw and accept x^2

Draw x', reject and set $x^3 = x^2$

Draw and accept x^4

Draw and accept x^5

Analysis of Metropolis-Hasting

- The transition kernel of MH is $\mathcal{T}(x'|x) = \mathcal{Q}(x'|x)\mathcal{A}(x'|x)$
- If $\mathcal{A}(x'|x) = \min(1, \frac{\mathcal{P}(x')\mathcal{Q}(x|x')}{\mathcal{P}(x)\mathcal{Q}(x'|x)}) < 1$, then $\frac{\mathcal{P}(x)\mathcal{Q}(x'|x)}{\mathcal{P}(x')\mathcal{Q}(x|x')} > 1$ and $\mathcal{A}(x|x') = 1$
- If $\mathcal{A}(x'|x) < 1$, then $\mathcal{A}(x'|x) = \frac{\mathcal{P}(x')\mathcal{Q}(x|x')}{\mathcal{P}(x)\mathcal{Q}(x'|x)}$ and $\mathcal{A}(x|x') = 1$
- $\mathcal{P}(x)\mathcal{Q}(x'|x)\mathcal{A}(x'|x) = \mathcal{P}(x')\mathcal{Q}(x|x')$
- $\mathcal{P}(x)\mathcal{Q}(x'|x)\mathcal{A}(x'|x) = \mathcal{P}(x')\mathcal{Q}(x|x')\mathcal{A}(x|x')$
- $\mathcal{P}(x)\mathcal{T}(x'|x) = \mathcal{P}(x')\mathcal{T}(x|x')$
- The above is the detailed balance equation showing that $\mathcal{P}(x)$ is the stationary distribution of the generated MC

- The mixing time of the algorithm is the number of steps required to reach the stationary distribution.
- All the samples generated before the mixing time should be discarded, since they are not sampled from the required distribution.
- The mixing time may be very long if the proposal distribution is not good.
- The mixing time should be estimated and doing this is an art!
- The steps before mixing constitute the burn-in phase.
- The burn-in is not *mathematically necessary*: it is a computational statistics trick (without burn-in the bias introduced by the first samples, may require a lot of subsequent samples to vanish)

A special case

Gibbs Sampling

- Suppose we want to sample from a multivariate distribution $\mathcal{P}(x_1, \ldots x_n)$ and we have a proposal distribution which is $\mathcal{P}(x_i|\mathbf{x}_{-i})$ where $\mathbf{x}_{-i}$ is the set of all the variables but x_i.
- Initialize a random sample $X^{(0)} = (x_1^{(0)}, \ldots x_n^{(0)})$
- For step $t = 1 \ldots N$:
  - for each variable x_i:
  - sample $x_i^{(k)}$ from $\mathcal{P}(x_i|x_1^{(t)}, \ldots x_{i-1}^{(t)}, x_{i+1}^{(t-1)}, \ldots x_n^{(t-1)})$
- Optional (but important): discard first K samples (burn-in) and consider only one sample every h (period)

- first variable may be uninitialized (it is sampled from the others at the first step)
- period h mitigates the fact the subsequent samples are *correlated* (*thinning*)

- Gibbs sampling is a special case of Metropolis-Hasting
- Proposal distribution $\mathcal{Q}(x_i', \mathbf{x}_{-i} | x_i, \mathbf{x}_{-i}) = \mathcal{P}(x_i' | \mathbf{x}_{-i})$
- $\mathcal{A}(x_i', \mathbf{x}_{-i} | x_i, \mathbf{x}_{-i}) = \min(1, \frac{\mathcal{P}(x_i', \mathbf{x}_{-i}) \mathcal{Q}(x_i, \mathbf{x}_{-i} | x_i', \mathbf{x}_{-i})}{\mathcal{P}(x_i, \mathbf{x}_{-i}) \mathcal{Q}(x_i', \mathbf{x}_{-i} | x_i, \mathbf{x}_{-i})})$
  $= \min(1, \frac{\mathcal{P}(x_i', \mathbf{x}_{-i}) \mathcal{P}(x_i | \mathbf{x}_{-i})}{\mathcal{P}(x_i, \mathbf{x}_{-i}) \mathcal{P}(x_i' | \mathbf{x}_{-i})})$
  $= \min(1, \frac{\mathcal{P}(x_i', | \mathbf{x}_{-i}) \mathcal{P}(\mathbf{x}_{-i}) \mathcal{P}(x_i | \mathbf{x}_{-i})}{\mathcal{P}(x_i, | \mathbf{x}_{-i}) \mathcal{P}(\mathbf{x}_{-i}) \mathcal{P}(x_i' | \mathbf{x}_{-i})})$
  $= \min(1, 1) = 1$
- Gibbs sampling is a version of Metropolis-Hasting where the next sample is always accepted!

Sampling in Probabilistic Graphical Models

Logical Sampling

Approximate Inference on a Bayesian Network: $\mathcal{P}(Q|\mathbf{e})$

Logical or Ancestral Sampling

- Take a topological order $X_1, \ldots X_n$ of the variable nodes
- For $i = 1 \ldots N$,
  - draw a sample of each variable following the topological order, producing a network sample $(x_1^{(i)}, \ldots x_n^{(i)})$
  - if the network sample does not agree with evidence $\mathbf{e}$, reject the sample
- Let N' be the number of accepted sample, let $\mathbf{q}$ be an assignment to variables in Q and let $N_{\mathbf{q}}$ be the number of samples satisfying $\mathbf{q}$; estimate $\hat{\mathcal{P}}(\mathbf{q}|\mathbf{e}) = \frac{N_{\mathbf{q}}}{N'}$

- A network sample can be rejected as soon as a sampled variable does not agree with the evidence (i.e., the samples value is different than the observed one).

Approximate Inference on a Bayesian Network: $\mathcal{P}(Q|\mathbf{e})$

Logical or Ancestral Sampling

- Take a topological order $X_1, \ldots X_n$ of the variable nodes
- For $i = 1 \ldots N$,
  - draw a sample of each variable following the topological order, producing a network sample $(x_1^{(i)}, \ldots x_n^{(i)})$
  - if the network sample does not agree with evidence $\mathbf{e}$, reject the sample
- Let N' be the number of accepted sample, let $\mathbf{q}$ be an assignment to variables in Q and let $N_{\mathbf{q}}$ be the number of samples satisfying $\mathbf{q}$; estimate $\hat{\mathcal{P}}(\mathbf{q}|\mathbf{e}) = \frac{N_{\mathbf{q}}}{N'}$

- Variable samples start from root nodes and proceed forward (indeed, *Forward Sampling* is an alternative name)

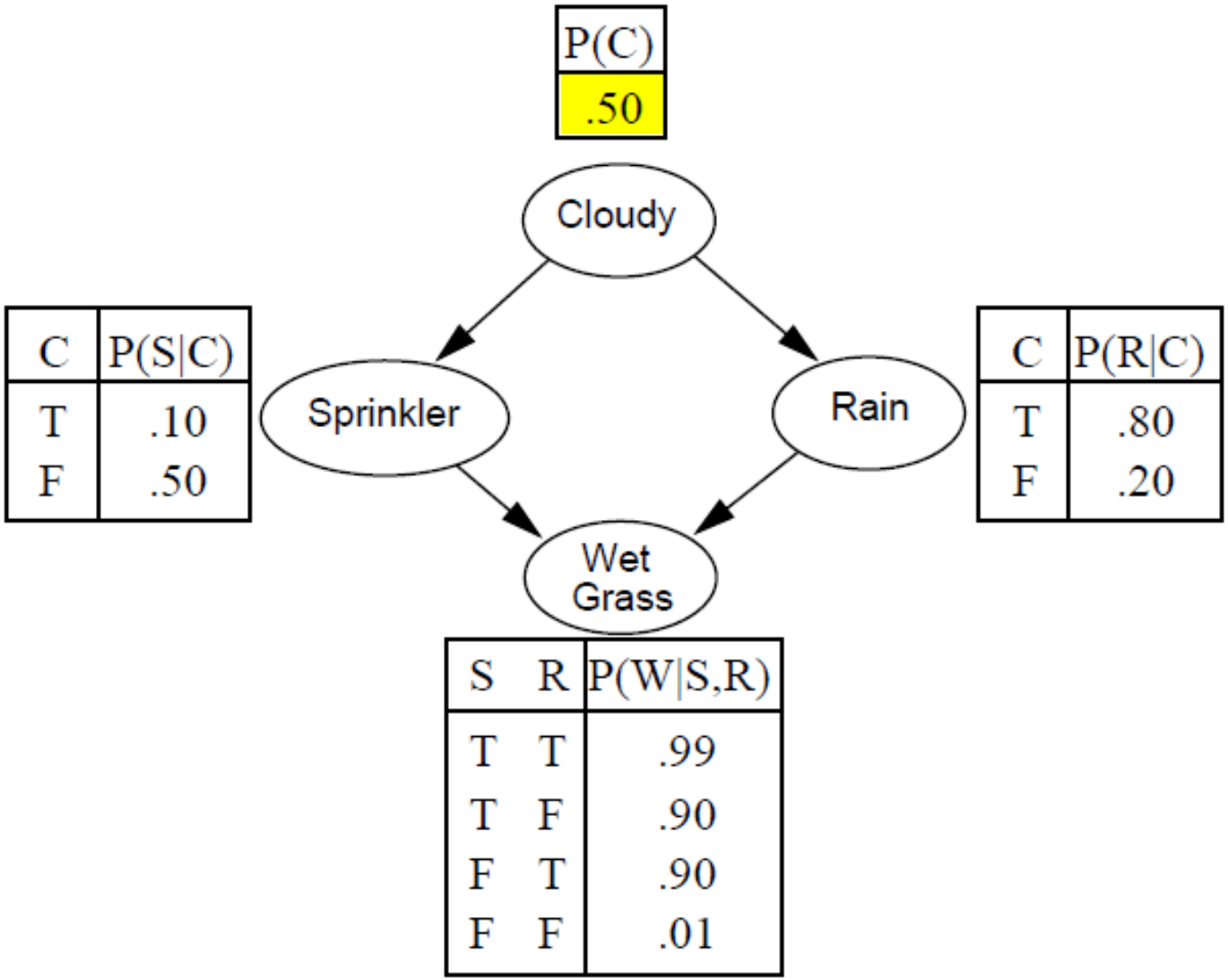
Approximate Inference on a Bayesian Network: $\mathcal{P}(Q|\mathbf{e})$

Logical or Ancestral Sampling

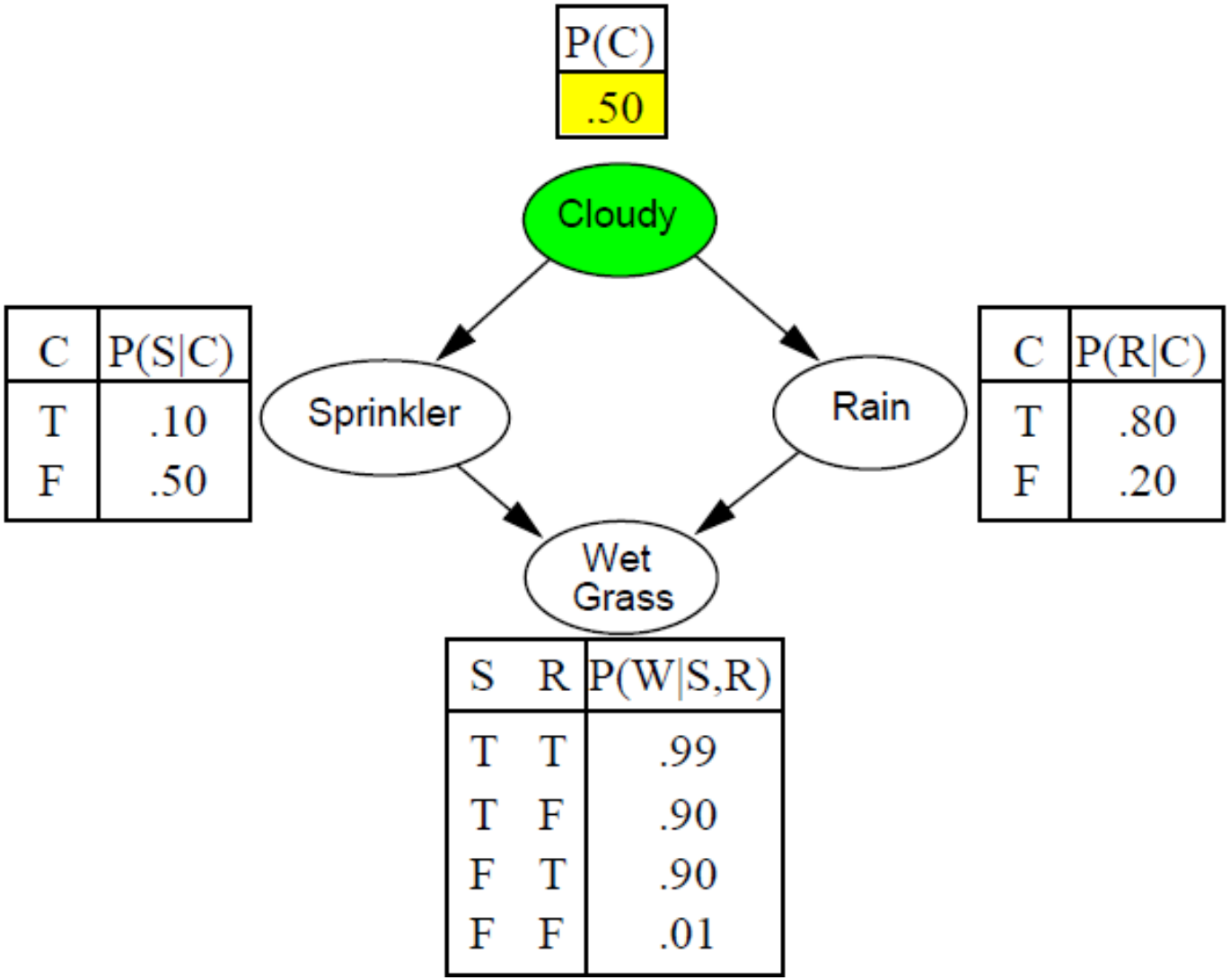
- Take a topological order $X_1, \ldots X_n$ of the variable nodes
- For $i = 1 \ldots N$,
  - draw a sample of each variable following the topological order, producing a network sample $(x_1^{(i)}, \ldots x_n^{(i)})$
  - if the network sample does not agree with evidence $\mathbf{e}$, reject the sample
- Let N' be the number of accepted sample, let $\mathbf{q}$ be an assignment to variables in Q and let $N_{\mathbf{q}}$ be the number of samples satisfying $\mathbf{q}$; estimate $\hat{\mathcal{P}}(\mathbf{q}|\mathbf{e}) = \frac{N_{\mathbf{q}}}{N'}$

- Following topological order, every variable can be sampled from its CPD (parents are already assigned when sampling the variable)

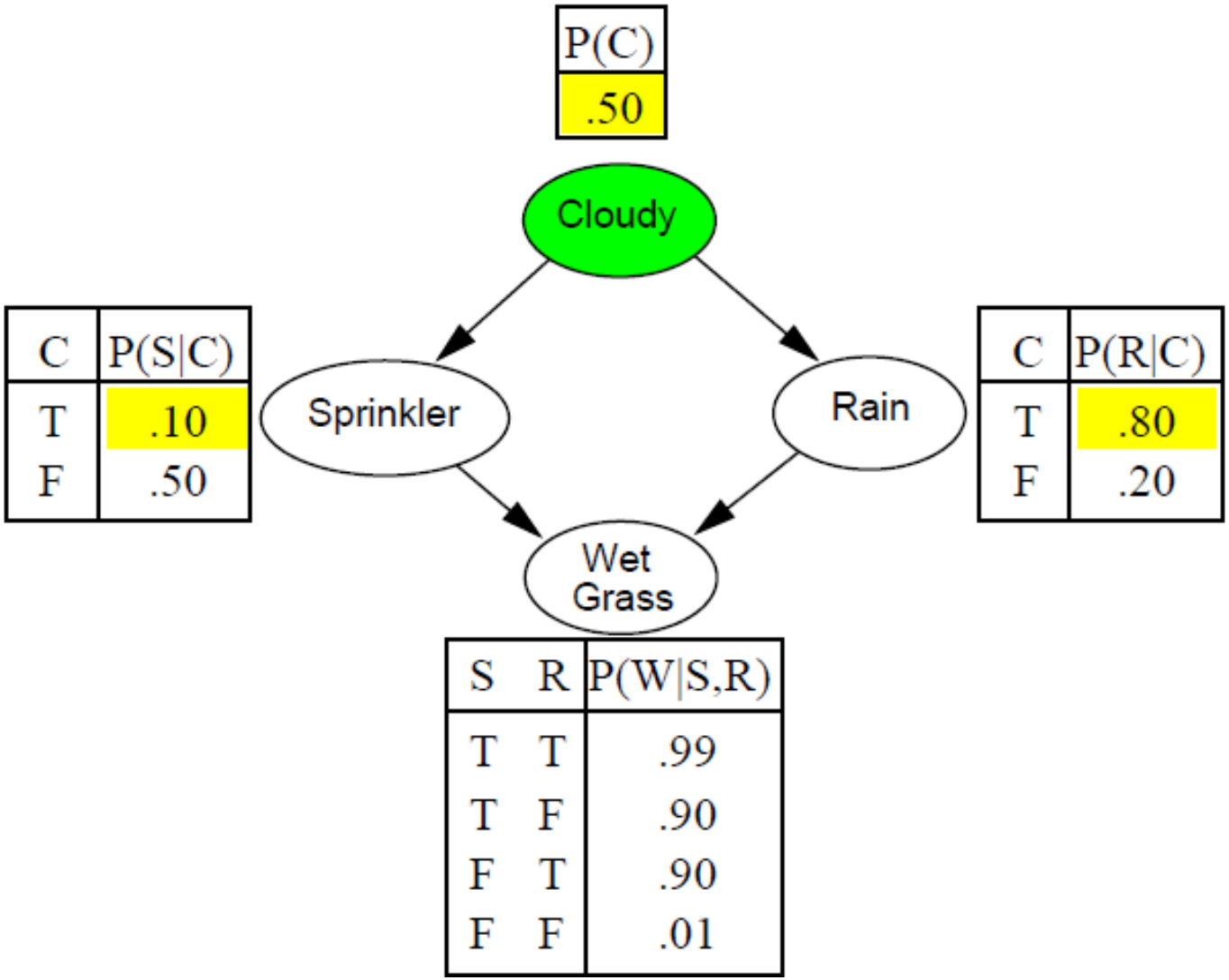
Example



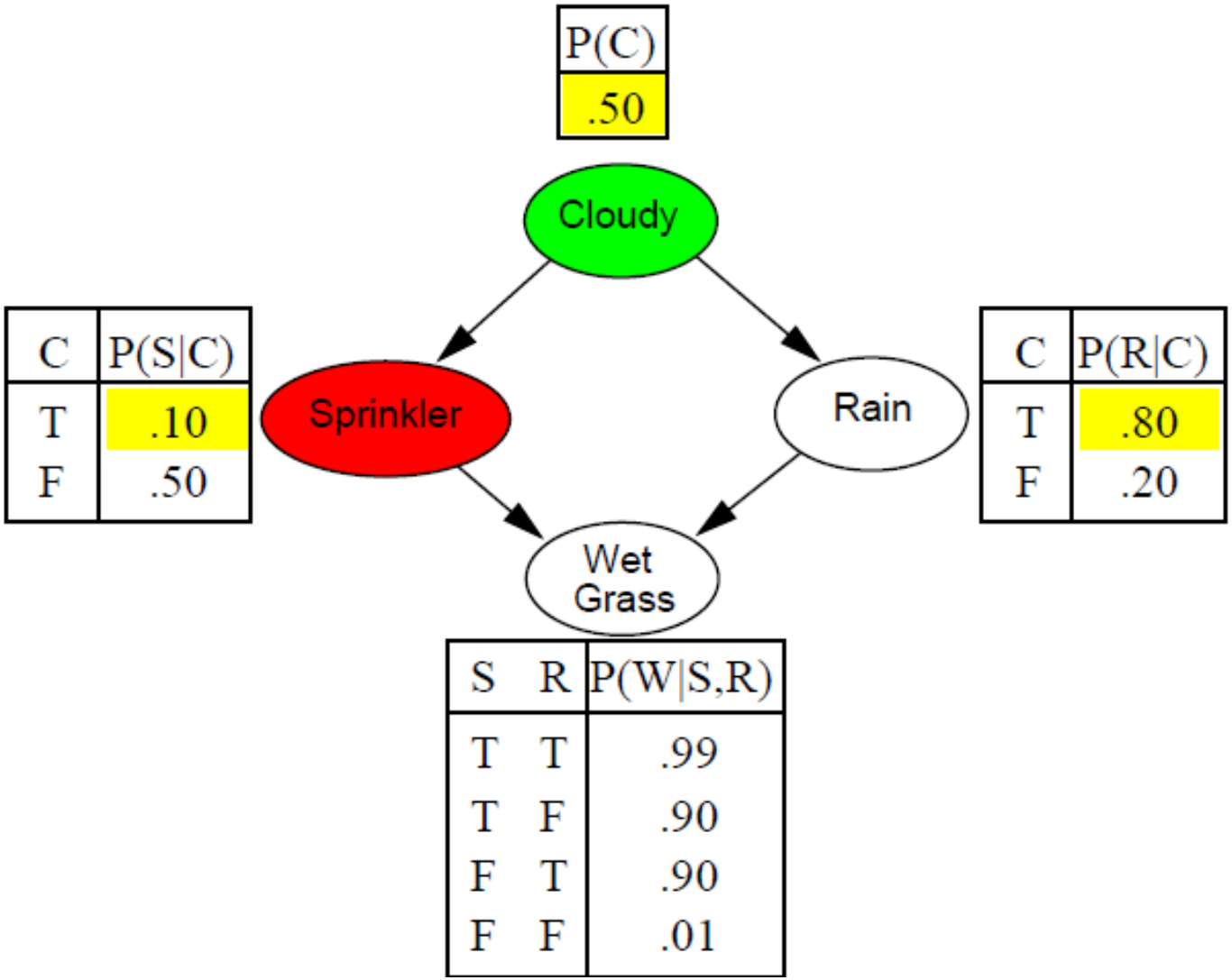
Example



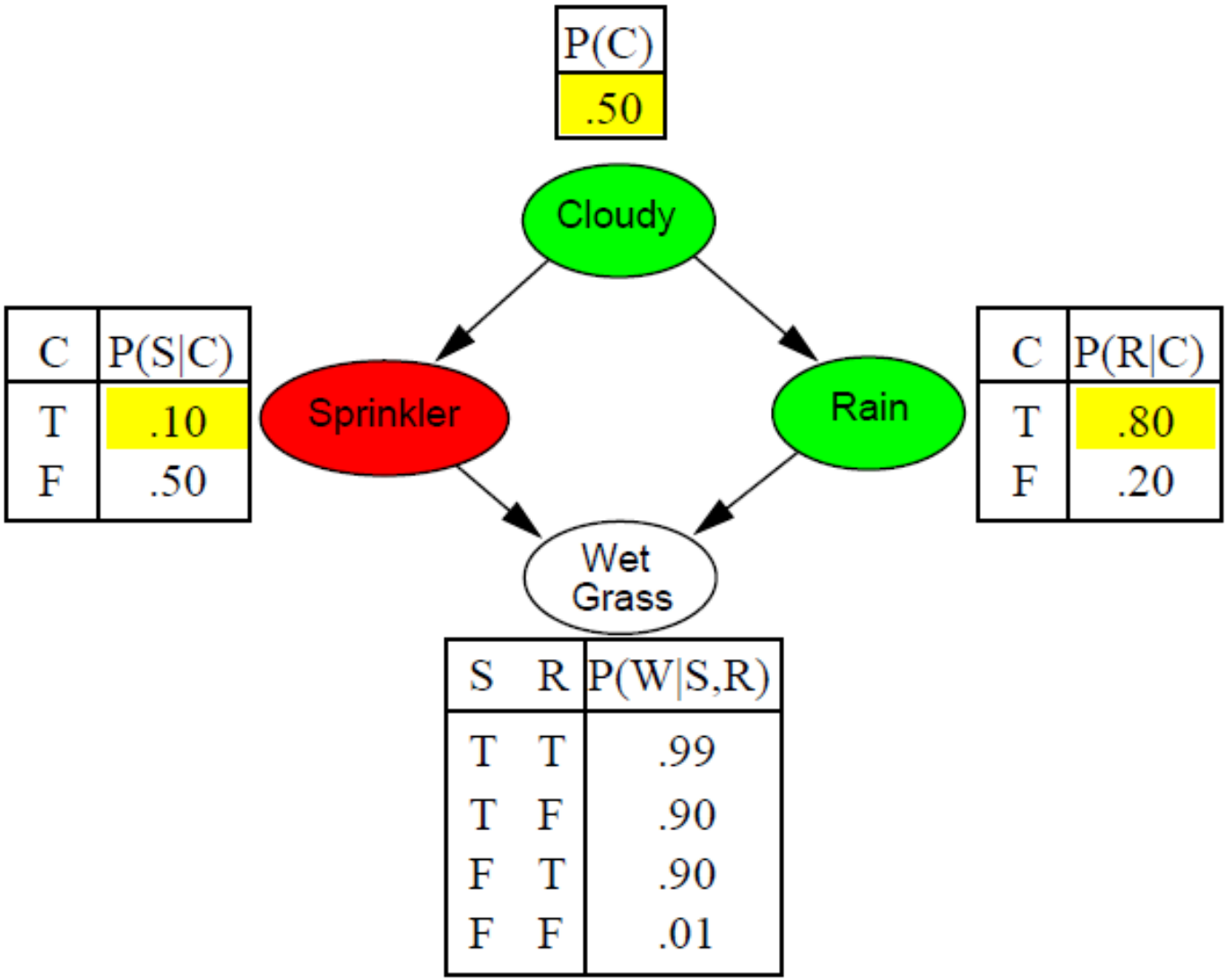
Example



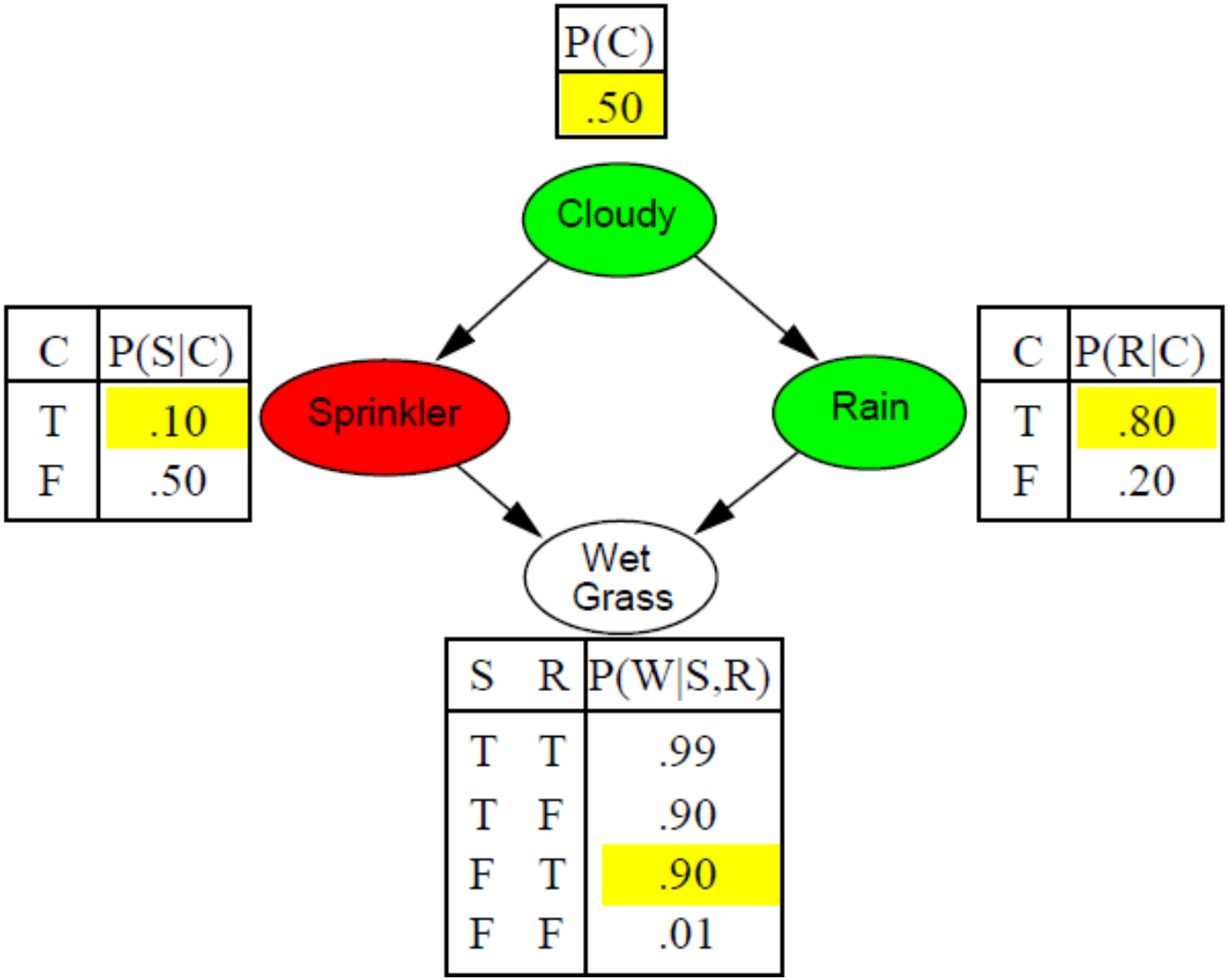
Example



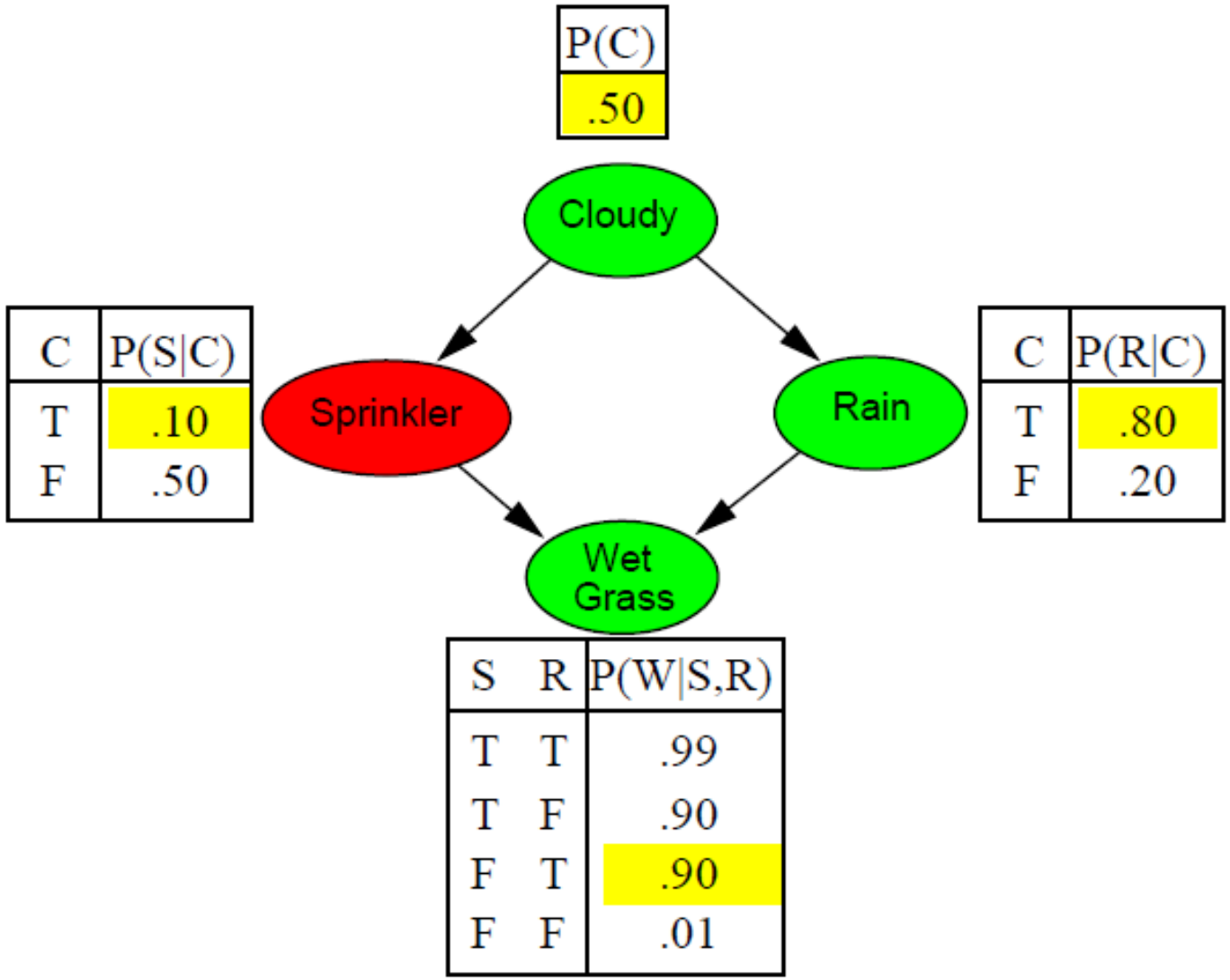
Example



Example



Example



- Logical sampling is a special case of Acceptance/Rejection sampling
- Target distribution is $\mathcal{P}(X) = \mathcal{P}(\mathbf{x}_{-\mathbf{e}}|\mathbf{e})$
- Proposal distribution is $\mathcal{Q}(X) = \mathcal{P}(\mathbf{x}_{-\mathbf{e}}, \mathbf{e})$
- Probability of acceptance is $\mathcal{A}(X) = \mathcal{P}(\mathbf{e})$ (a sample is accepted if it agrees with $\mathbf{e}$)
- Since $\mathcal{A}(X) = \frac{\mathcal{P}(\mathbf{x}_{-\mathbf{e}}|\mathbf{e})}{\alpha\mathcal{P}(\mathbf{x}_{-\mathbf{e}},\mathbf{e})}$, it follows that $\alpha = \mathcal{P}(\mathbf{e})^{-2}$
- The requirement $\mathcal{P}(X) \leq \alpha\mathcal{Q}(X)$ is satisfied.
  $\frac{\mathcal{P}(\mathbf{x}_{-\mathbf{e}},\mathbf{e})}{\mathcal{P}(\mathbf{e})} \leq \mathcal{P}(\mathbf{e})^{-2}\mathcal{P}(\mathbf{x}_{-\mathbf{e}}, \mathbf{e})$ since $\mathcal{P}(\mathbf{e})^{-1} \leq \mathcal{P}(\mathbf{e})^{-2}$

Pros and Cons

Pro: easy to implement

Cons: too many rejections if evidence is rare, applicable only to directed models

Likelihood Weighting

Likelihood-weighted particle generation

Procedure LW-Sample (
  $\mathcal{B}$, // Bayesian network over $\mathcal{X}$
  $Z = z$ // Event in the network
)

1 Let $X_1, \ldots, X_n$ be a topological ordering of $\mathcal{X}$
2 $w \leftarrow 1$
3 **for** $i = 1, \ldots, n$
4 $u_i \leftarrow x\langle \mathrm{Pa}_{X_i} \rangle$ // Assignment to Pa_{X_i} in $x_1, \ldots, x_{i-1}$
5 **if** $X_i \notin Z$ **then**
6 Sample x_i from $P(X_i \mid u_i)$
7 **else**
8 $x_i \leftarrow z\langle X_i \rangle$ // Assignment to X_i in z
9 $w \leftarrow w \cdot P(x_i \mid u_i)$ // Multiply weight by probability of desired value
10 **return** $(x_1, \ldots, x_n), w$

Likelihood-weighted particle generation

Procedure LW-Sample (
  $\mathcal{B}$, // Bayesian network over $\mathcal{X}$
  $Z = z$ // Event in the network
)

1 Let $X_1, \ldots, X_n$ be a topological ordering of $\mathcal{X}$
2 $w \leftarrow 1$
3 **for** $i = 1, \ldots, n$
4 $u_i \leftarrow x\langle \mathrm{Pa}_{X_i} \rangle$ // Assignment to Pa_{X_i} in $x_1, \ldots, x_{i-1}$
5 **if** $X_i \notin Z$ **then**
6 Sample x_i from $P(X_i \mid u_i)$
7 **else**
8 $x_i \leftarrow z\langle X_i \rangle$ // Assignment to X_i in z
9 $w \leftarrow w \cdot P(x_i \mid u_i)$ // Multiply weight by probability of desired value
10 **return** $(x_1, \ldots, x_n), w$

- Sampling order as in logical sampling but:

Likelihood-weighted particle generation

Procedure LW-Sample (
  $\mathcal{B}$, // Bayesian network over $\mathcal{X}$
  $Z = z$ // Event in the network
)

1 Let $X_1, \ldots, X_n$ be a topological ordering of $\mathcal{X}$
2 $w \leftarrow 1$
3 **for** $i = 1, \ldots, n$
4 $u_i \leftarrow x\langle \mathrm{Pa}_{X_i} \rangle$ // Assignment to Pa_{X_i} in $x_1, \ldots, x_{i-1}$
5 **if** $X_i \notin Z$ **then**
6 Sample x_i from $P(X_i \mid u_i)$
7 **else**
8 $x_i \leftarrow z\langle X_i \rangle$ // Assignment to X_i in z
9 $w \leftarrow w \cdot P(x_i \mid u_i)$ // Multiply weight by probability of desired value
10 **return** $(x_1, \ldots, x_n), w$

- evidence variables are not sampled

Likelihood-weighted particle generation

Procedure LW-Sample (
  $\mathcal{B}$, // Bayesian network over $\mathcal{X}$
  $Z = z$ // Event in the network
)
1 Let $X_1, \ldots, X_n$ be a topological ordering of $\mathcal{X}$
2 $w \leftarrow 1$
3 **for** $i = 1, \ldots, n$
4 $u_i \leftarrow x\langle \mathrm{Pa}_{X_i} \rangle$ // Assignment to Pa_{X_i} in $x_1, \ldots, x_{i-1}$
5 **if** $X_i \notin Z$ **then**
6 Sample x_i from $P(X_i \mid u_i)$
7 **else**
8 $x_i \leftarrow z\langle X_i \rangle$ // Assignment to X_i in z
9 $w \leftarrow w \cdot P(x_i \mid u_i)$ // Multiply weight by probability of desired value
10 **return** $(x_1, \ldots, x_n), w$

- each sample is weighted with an importance weight which is the likelihood accorded to the evidence in such a sample, as measured by the product of the CPDs of each evidence variable

Likelihood-weighted particle generation

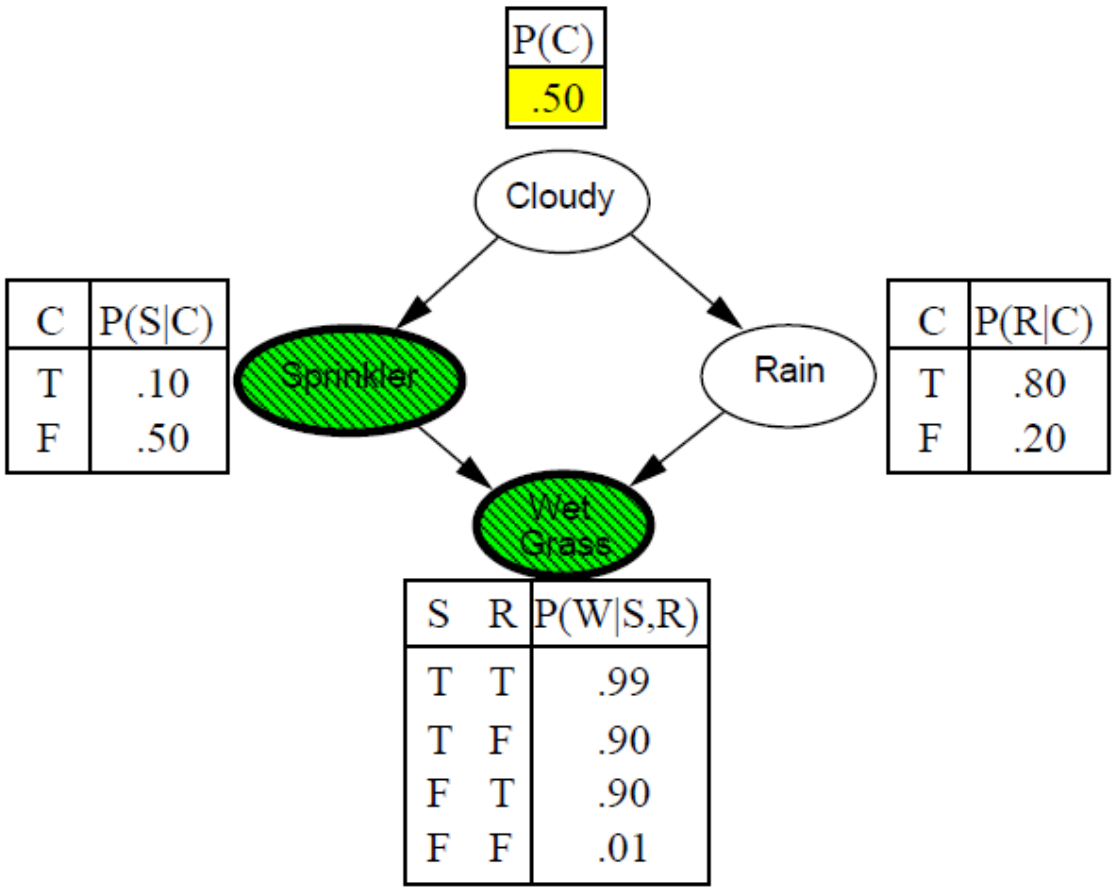
Procedure LW-Sample (
  $\mathcal{B}$, // Bayesian network over $\mathcal{X}$
  $Z = z$ // Event in the network
)

1 Let $X_1, \ldots, X_n$ be a topological ordering of $\mathcal{X}$
2 $w \leftarrow 1$
3 **for** $i = 1, \ldots, n$
4 $u_i \leftarrow x\langle \mathrm{Pa}_{X_i} \rangle$ // Assignment to Pa_{X_i} in $x_1, \ldots, x_{i-1}$
5 **if** $X_i \notin Z$ **then**
6 Sample x_i from $P(X_i \mid u_i)$
7 **else**
8 $x_i \leftarrow z\langle X_i \rangle$ // Assignment to X_i in z
9 $w \leftarrow w \cdot P(x_i \mid u_i)$ // Multiply weight by probability of desired value
10 **return** $(x_1, \ldots, x_n), w$

- if N is the total number of particles (runs), estimate

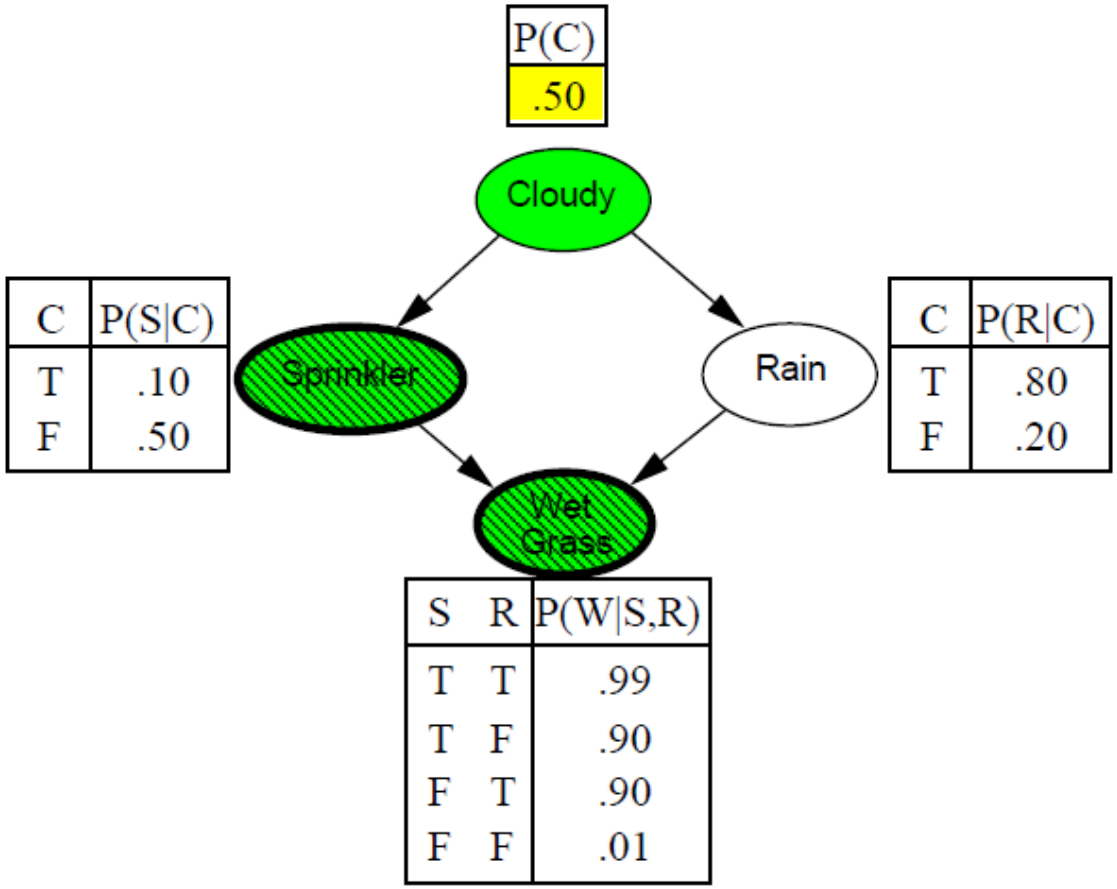
$$\hat{\mathcal{P}}(\mathbf{q}|\mathbf{e}) = \frac{\sum_{i=1}^{N} w^{(i)} \mathbb{1}(X^{(i)}\langle Q \rangle = \mathbf{q})}{\sum_{i=1}^{N} w^{(i)}}$$

Likelihood weighting example



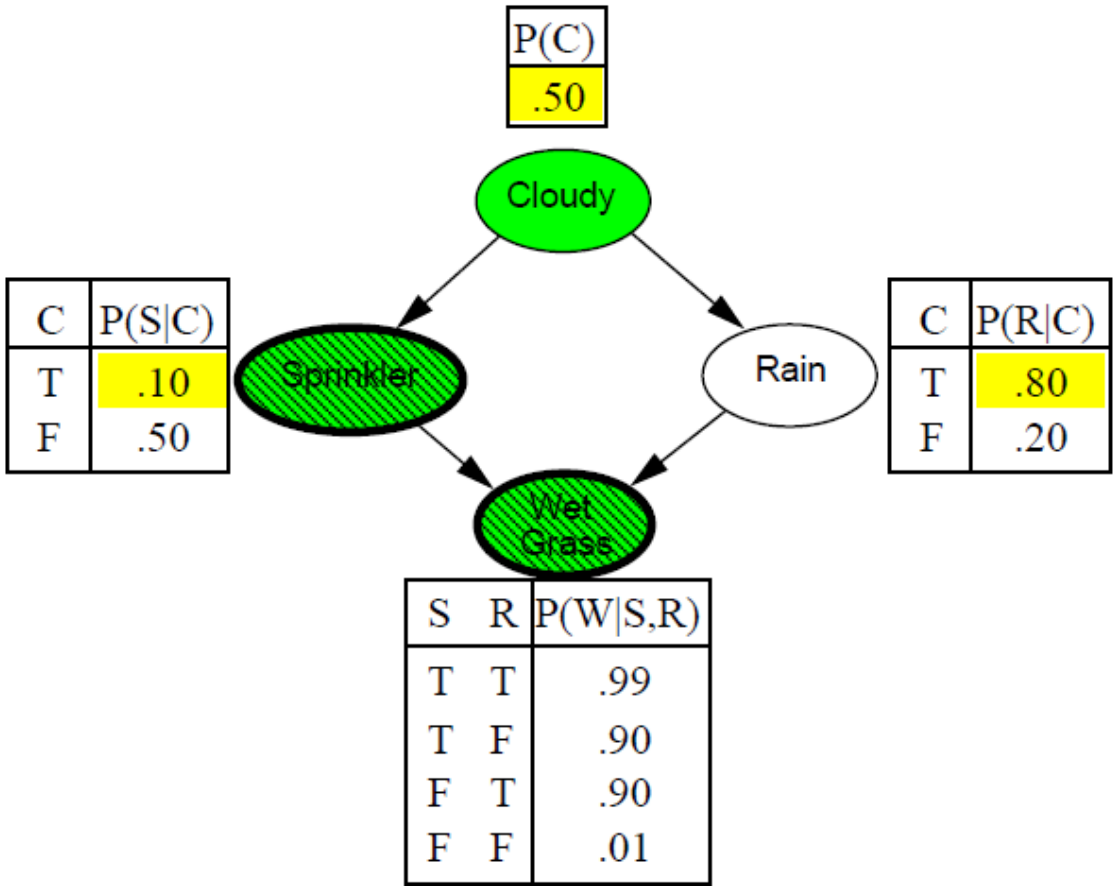
$w = 1.0$

Likelihood weighting example



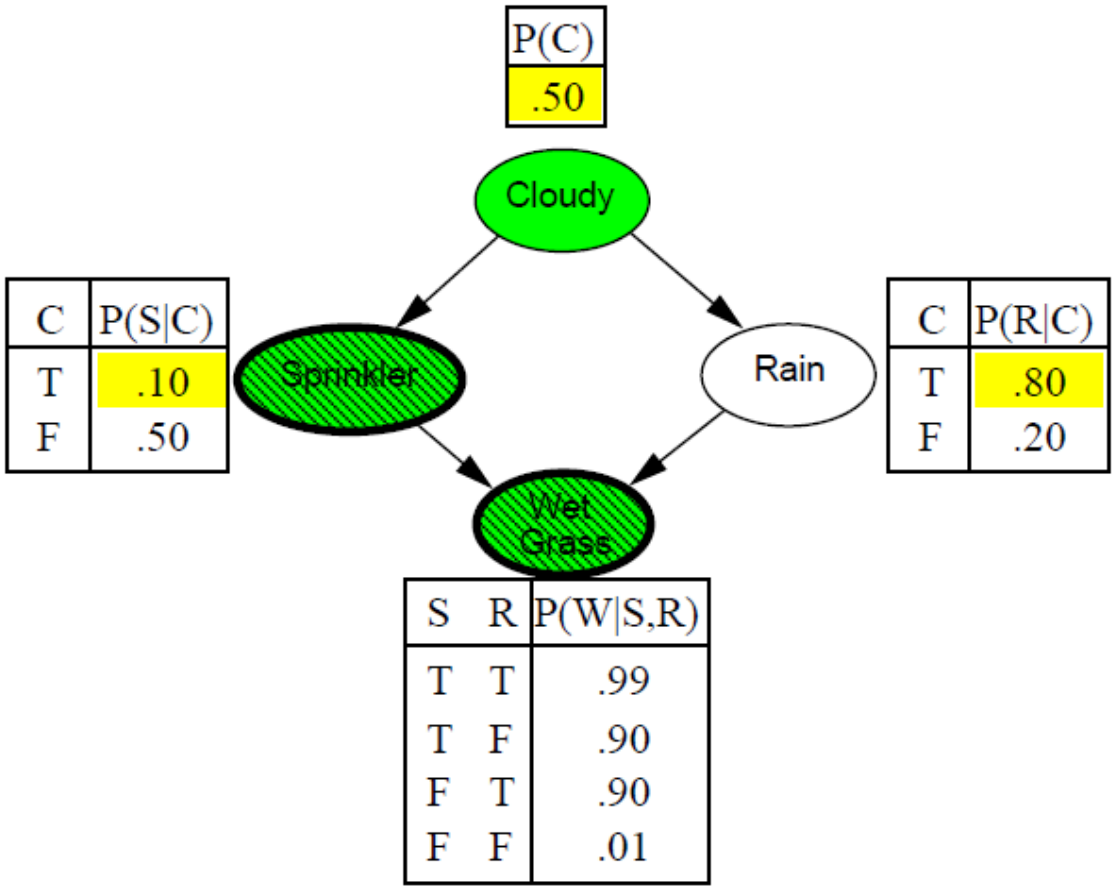
$w = 1.0$

Likelihood weighting example



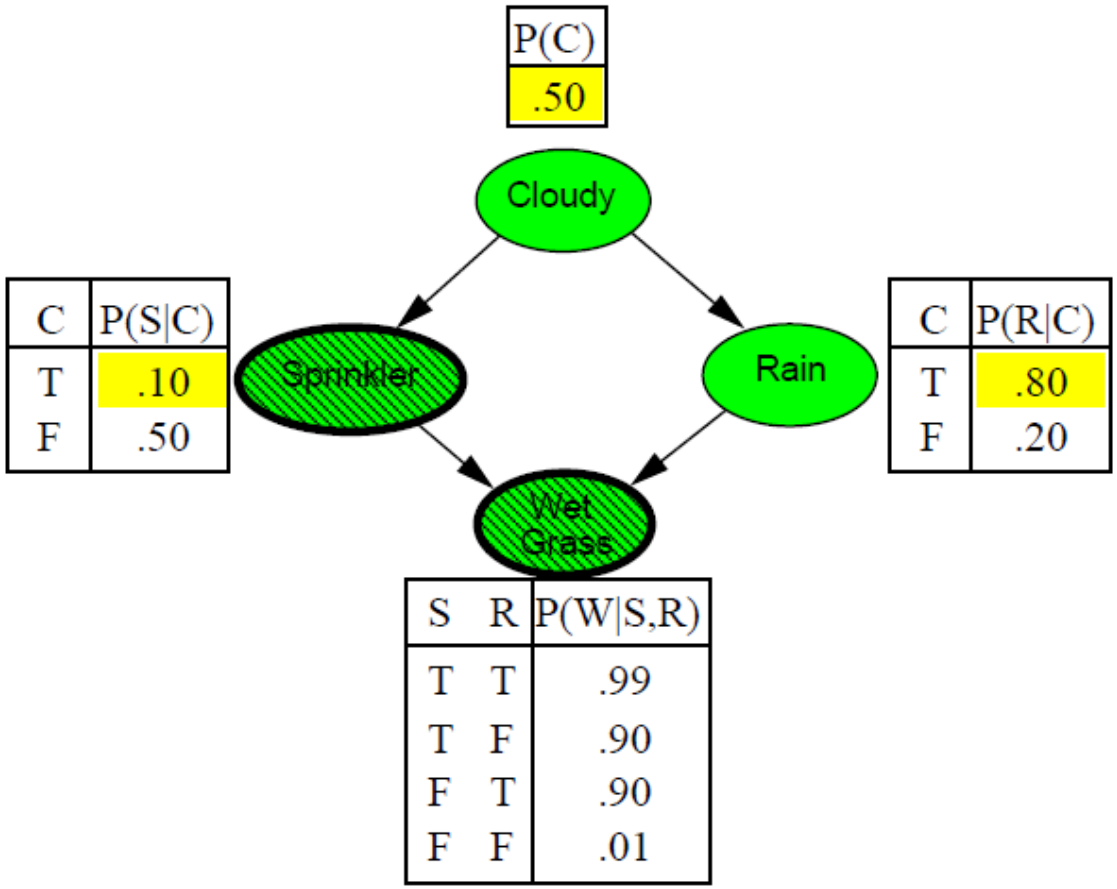
$w = 1.0$

Likelihood weighting example



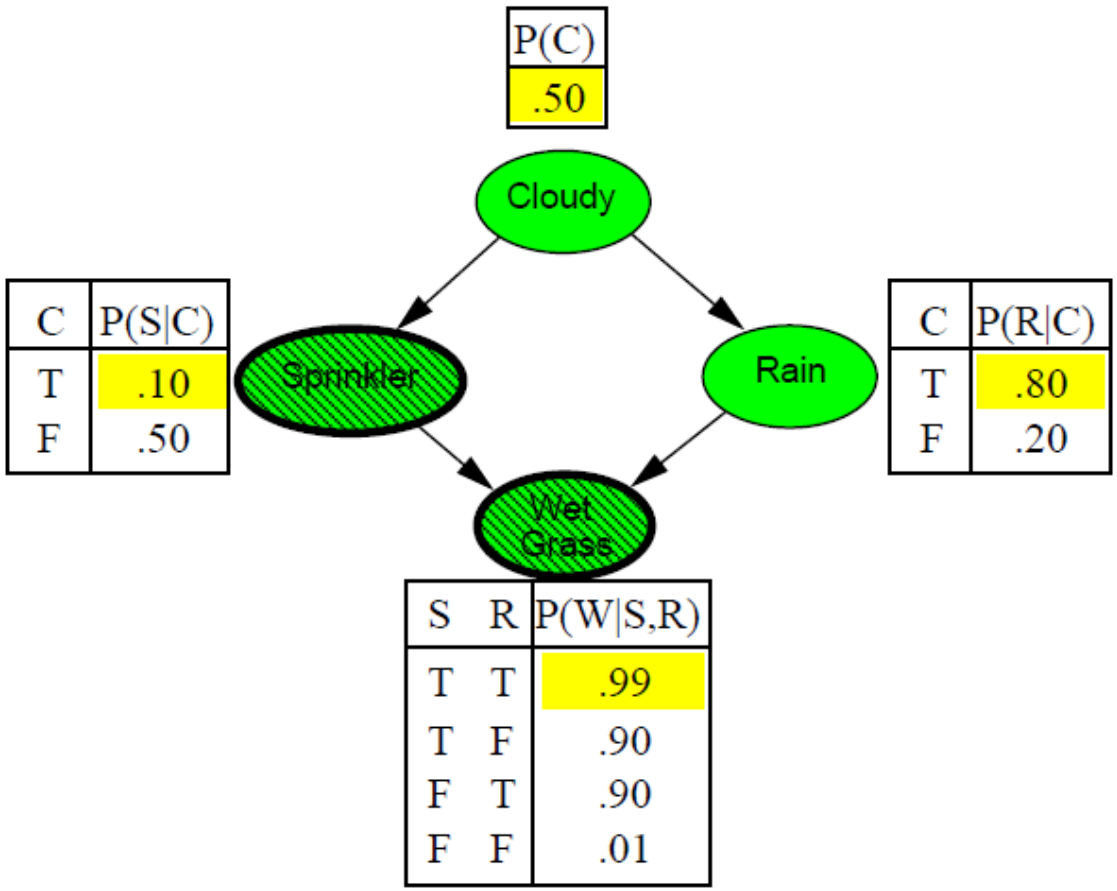
$w = 1.0 \times 0.1$

Likelihood weighting example



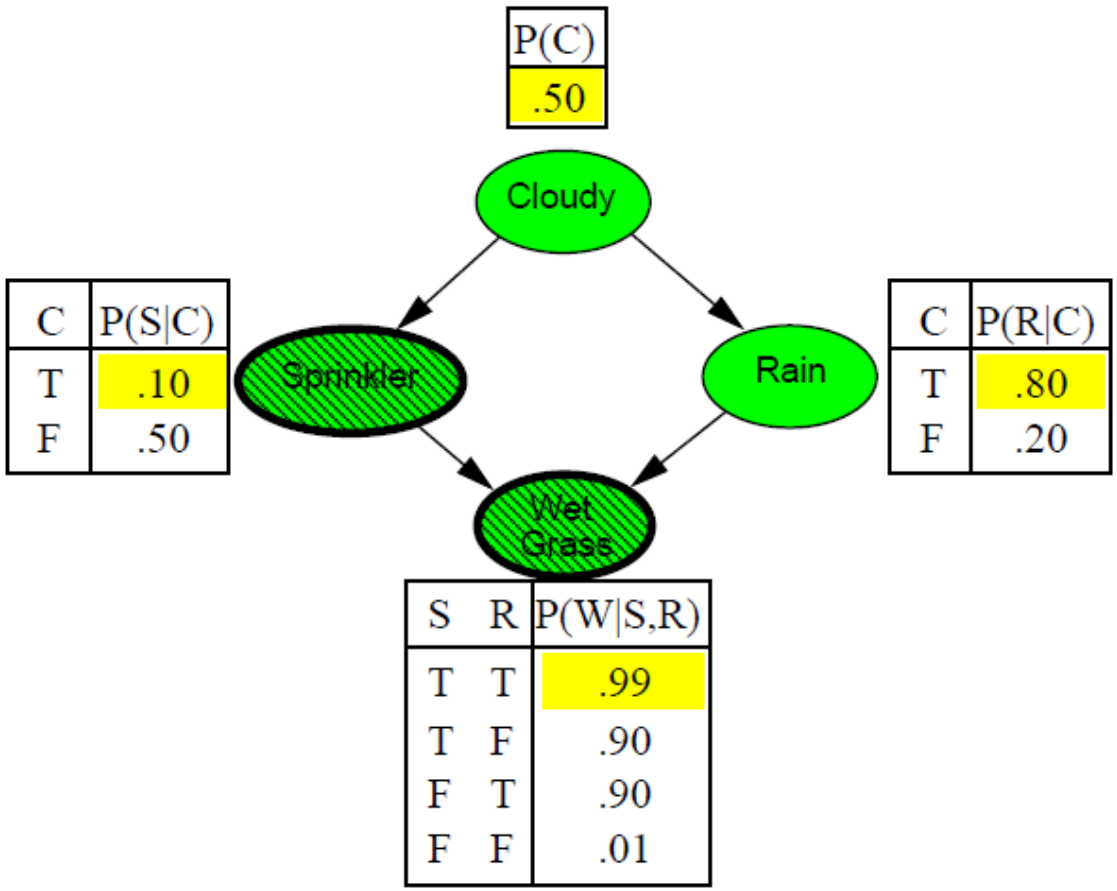
$w = 1.0 \times 0.1$

Likelihood weighting example



$w = 1.0 \times 0.1$

Likelihood weighting example



$$w = 1.0 \times 0.1 \times 0.99 = 0.099$$

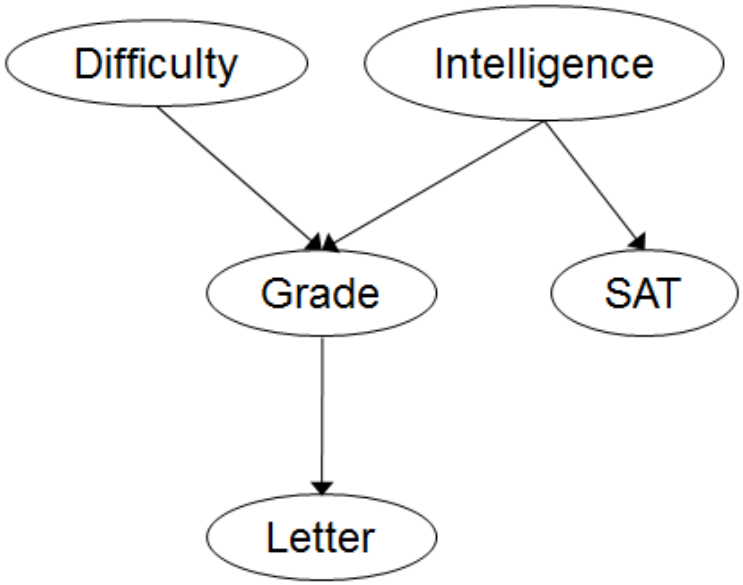
- Likelihood weighting is a special case of Normalized Importance Sampling
- Given a BN B and some evidence $\mathbf{e}$, we define the mutilated network $B_{\mathbf{e}}$ as follows:
  - every evidence node has no parent
  - the CPD of an evidence node $E_i = e_i$ is set to 1 for state e_i and set to 0 for every other state.
  - all other CPDs are kept unchanged

Student Example

D	P(D)
low	0.6
high	0.4

I	P(I)
low	0.7
high	0.3

D	I	G	P(G|D,I)
low	low	C	0.3
low	low	B	0.4
low	low	A	0.3
low	high	C	0.02
low	high	B	0.08
low	high	A	0.9
high	low	C	0.7
high	low	B	0.25
high	low	A	0.05
high	high	C	0.2
high	high	B	0.3
high	high	A	0.5



I	S	P(S|I)
low	low	0.95
low	high	0.05
high	low	0.2
high	high	0.8

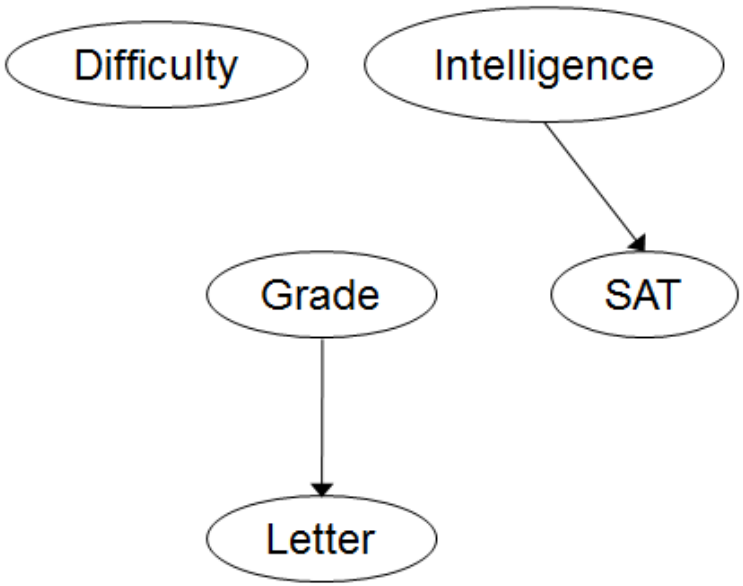
G	L	P(L|G)
C	weak	0.99
C	strong	0.01
B	weak	0.4
B	strong	0.6
A	weak	0.1
A	strong	0.9

Mutilated Network:

I	P(I)
low	0
high	1

D	P(D)
low	0.6
high	0.4

G	P(G|D,I)
C	0
B	1
A	0



I	S	P(S|I)
low	low	0.95
low	high	0.05
high	low	0.2
high	high	0.8

G	L	P(L|G)
C	weak	0.99
C	strong	0.01
B	weak	0.4
B	strong	0.6
A	weak	0.1
A	strong	0.9

Evidence: `Grade=B, Intelligence=high`

- Let B be the original network defining distribution $\mathcal{P}_B$, $\mathbf{e}$ the evidence and $B_{\mathbf{e}}$ the mutilated network defining distribution $\mathcal{P}_{B_{\mathbf{e}}}$.
- The proposal distribution of *LW* is the one defined by the mutilated network $\mathcal{P}_{B_{\mathbf{e}}}$.
- the weight of a sample s is $w(s) = \frac{\mathcal{P}_B(s)}{\mathcal{P}_{B_{\mathbf{e}}}(s)}$
- the estimation provided by the algorithm corresponds to the one of Normalized IS where:
  - proposal distribution is $\mathcal{Q}(X) = \mathcal{P}_{B_{\mathbf{e}}}(X)$
  - target distribution is $\mathcal{P}(X) = \mathcal{P}_B(X_{-\mathbf{e}}|\mathbf{e})$
  - unormalized distribution is $\tilde{\mathcal{P}}(X) = \mathcal{P}_B(X_{-\mathbf{e}}, \mathbf{e})$
  - the function whose expected value is estimated is the indicator function of the query

$$\tilde{\mathcal{P}}(X) = \prod_{x} \mathcal{P}(x|\pi(x)) = \prod_{e \in E} \mathcal{P}(e|\pi(e)) \prod_{x \in \mathbf{x_{-e}}} \mathcal{P}(x|\pi(x))$$

$$\mathcal{Q}(X) = \prod_{x \in \mathbf{x_{-e}}} \mathcal{P}(x|\pi(x))$$

$$w(X) = \frac{\tilde{\mathcal{P}}(X)}{\mathcal{Q}(X)} = \frac{\prod_{e \in E} \mathcal{P}(e|\pi(e)) \prod_{x \in \mathbf{x_{-e}}} \mathcal{P}(x|\pi(x))}{\prod_{x \in \mathbf{x_{-e}}} \mathcal{P}(x|\pi(x))} =$$

$$\prod_{e \in E} \mathcal{P}(e|\pi(e))$$

LW: pros and cons

- Pros: no sampling on evidence variables; no need for rejection.
- Cons: with many evidence variables (or in general with rare evidence) estimate dominated by a small fraction of samples, since the large part will have a small weight (slow convergence); downstream evidence does not influence samples (non evidence variables are sampled without taking into account evidence "from below")

Gibbs Sampling

- Forward sampling algorithms can be applied only to DAG or tree-based UGM.
- Gibbs sampling requires to sample a single variable conditioned on the other variables.
- PGM (either directed or undirected) have the notion of *Markov Blanket* that works as isolation of each variable wrt the rest of the model. A variable is independent from the rest of the network given its MB.
- If we devise an efficient way to sample a variable given its MB, then we could apply Gibbs sampling using a local view of each variable.

Markov Blanket: Bayesian Network

- Given a variable node X, the Markov Blanket of X is given by its parents $\pi(X)$, its children $\gamma(X)$, and its mates (other parents of the children)

$$\mathcal{P}(X|MB(X)) \propto \mathcal{P}(X|\pi(X)) \prod_{Y \in \gamma(x)} \mathcal{P}(Y|\pi(Y))$$

Markov Blanket: MRF

- Given a variable node X, the Markov Blanket of X is given by its neighbors.
- Given a clique C, if S_C is the scope (set of variables) of C

$$\mathcal{P}(X|MB(X)) \propto \prod_{C:X \in S_C} \Phi_C(S_C)$$

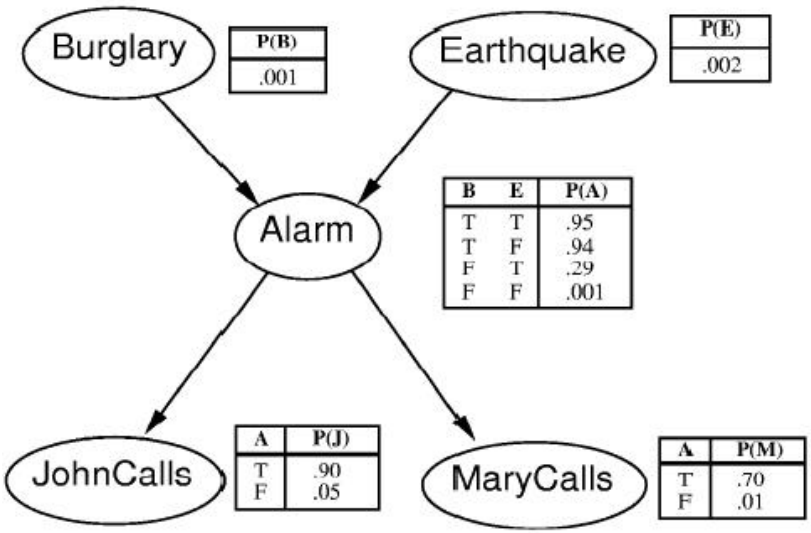
Gibbs Sampling for PGMs

Let $V = \{X_1 \dots X_n\}$ be the variables of a PGM with distribution $\mathcal{P}$, let $Q \subseteq V$ be a set of queried variables and $\mathbf{e}$ the evidence (i.e. an assignment to variables in $E \subseteq V \setminus Q$)

- set the variables in V to a random initial state (consistent with $\mathbf{e}$)
- For $k = 1 \dots N$ (number of runs)
  - For each $X_i \in V$:
  - if $X_i \in E$ then set X_i to the value assigned in $\mathbf{e}$
  - else sample X_i from $\mathcal{P}(X_i|MB(X_i))$

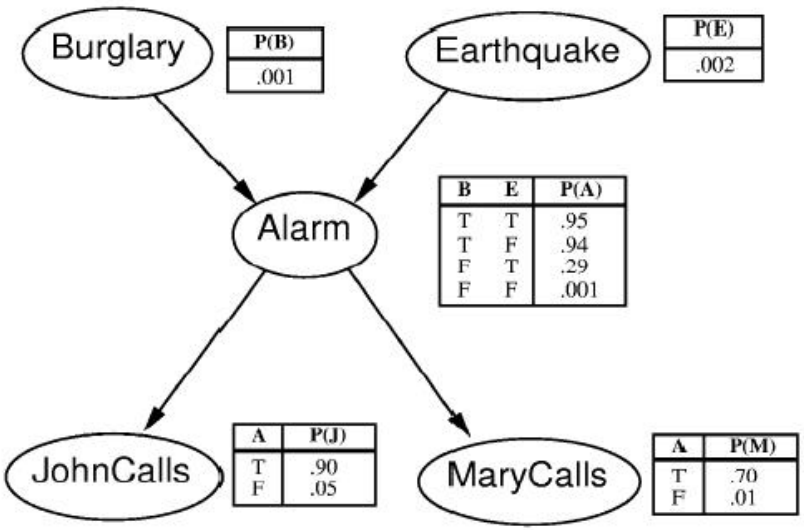
$$\hat{\mathcal{P}}(Q = \mathbf{q}|\mathbf{e}) = \frac{\sum_{j=b}^{N} \mathbb{1}(Q = \mathbf{q})}{N - b}$$

where b is the number of burn-in steps



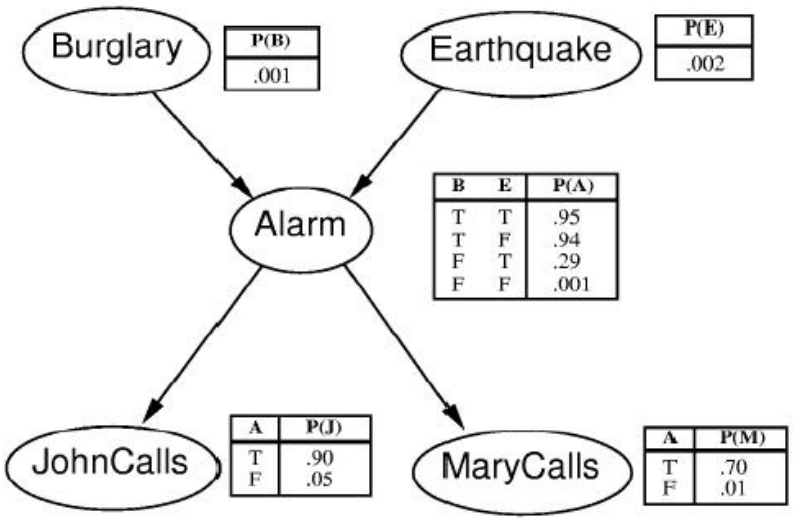
t	B	E	A	J	M
0	F	F	F	F	F
1					
2					
3					
4					

- Assume sampling order is B, E, A, J, M
- Initialize all variables to F



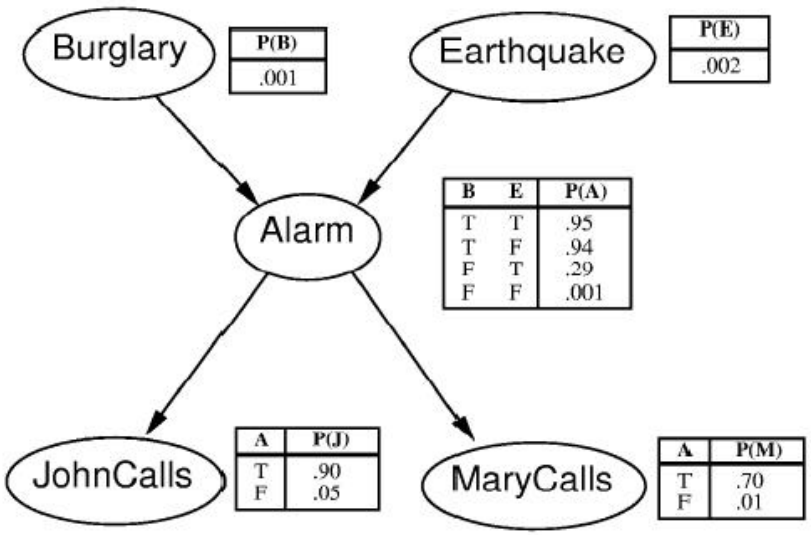
t	B	E	A	J	M
0	F	F	F	F	F
1	F				
2					
3					
4					

- Sampling B from $\mathcal{P}(B|A,E) \propto \mathcal{P}(A|B,E)\mathcal{P}(B)$
- $\mathcal{P}(B = \mathtt{T}|\mathtt{A} = \mathtt{F}, \mathtt{E} = \mathtt{F}) \propto (0.06)(0.01) = 0.0006$
  $\mathcal{P}(B = \mathtt{F}|\mathtt{A} = \mathtt{F}, \mathtt{E} = \mathtt{F}) \propto (0.999)(0.999) = 0.9980$



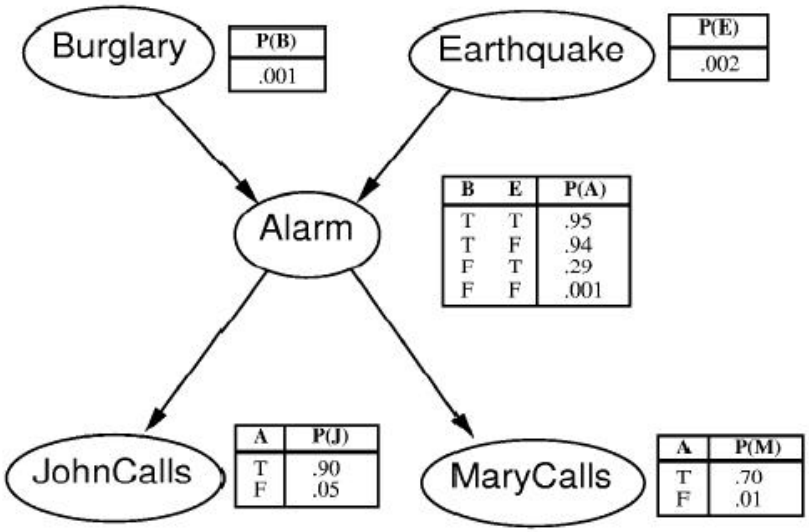
t	B	E	A	J	M
0	F	F	F	F	F
1	F	T			
2					
3					
4					

- Sampling E from $\mathcal{P}(E|A,B) \propto \mathcal{P}(A|B,E)\mathcal{P}(E)$
- $\mathcal{P}(E = \mathtt{T}|\mathtt{A} = \mathtt{F}, \mathtt{B} = \mathtt{F}) \propto (0.71)(0.02) = 0.142$
  $\mathcal{P}(E = \mathtt{F}|\mathtt{A} = \mathtt{F}, \mathtt{B} = \mathtt{F}) \propto (0.999)(0.998) = 0.9970$



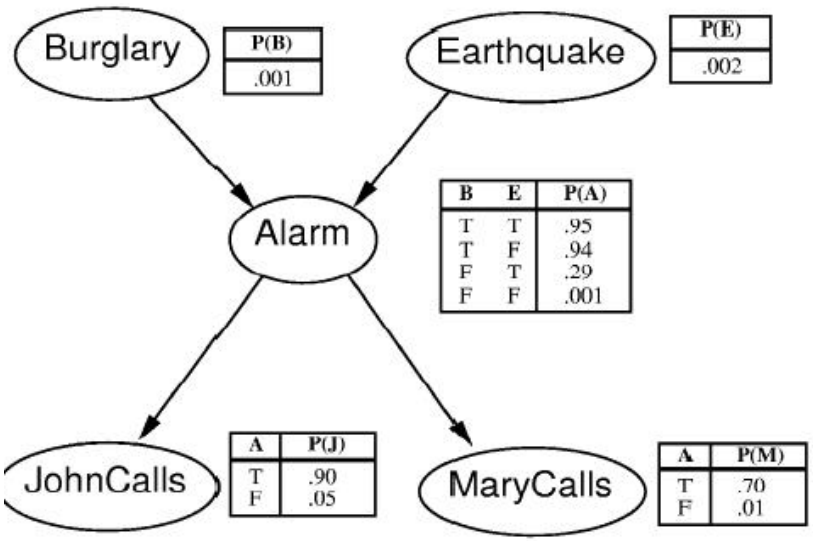
t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F		
2					
3					
4					

- Sampling A from
  $\mathcal{P}(A|B,E,J,M) \propto \mathcal{P}(J|A)\mathcal{P}(M|A)\mathcal{P}(A|B,E)$
- $\mathcal{P}(A = \mathtt{T}|\mathtt{B} = \mathtt{F}, \mathtt{E} = \mathtt{T}, \mathtt{J} = \mathtt{F}, \mathtt{M} = \mathtt{F}) \propto (0.1)(0.3)(0.29) = 0.0087$
  $\mathcal{P}(A = \mathtt{F}|\mathtt{B} = \mathtt{F}, \mathtt{E} = \mathtt{T}, \mathtt{J} = \mathtt{F}, \mathtt{M} = \mathtt{F}) \propto$
  $(0.95)(0.99)(0.71) = 0.6678$



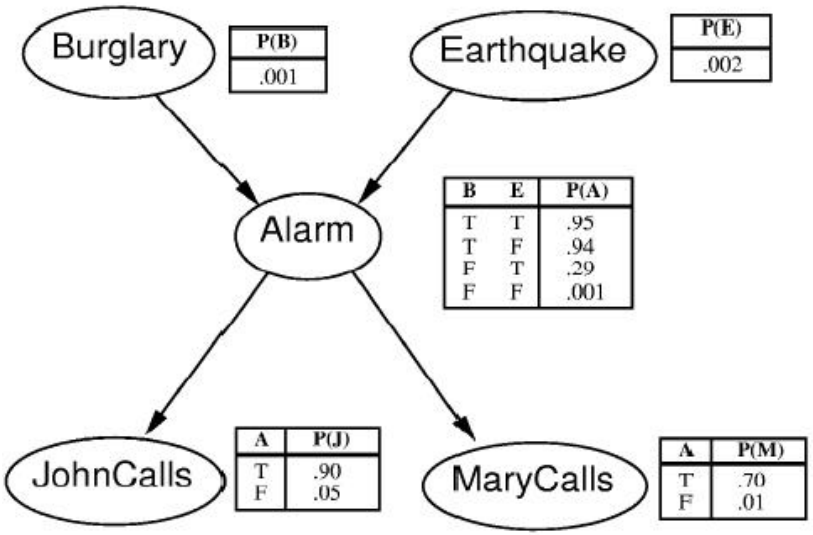
t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	
2					
3					
4					

- Sampling J from $\mathcal{P}(J|A)$
- $\mathcal{P}(J = \mathtt{T}|\mathtt{A} = \mathtt{F}) \propto 0.05$
  $\mathcal{P}(J = \mathtt{F}|\mathtt{A} = \mathtt{F}) \propto 0.95$



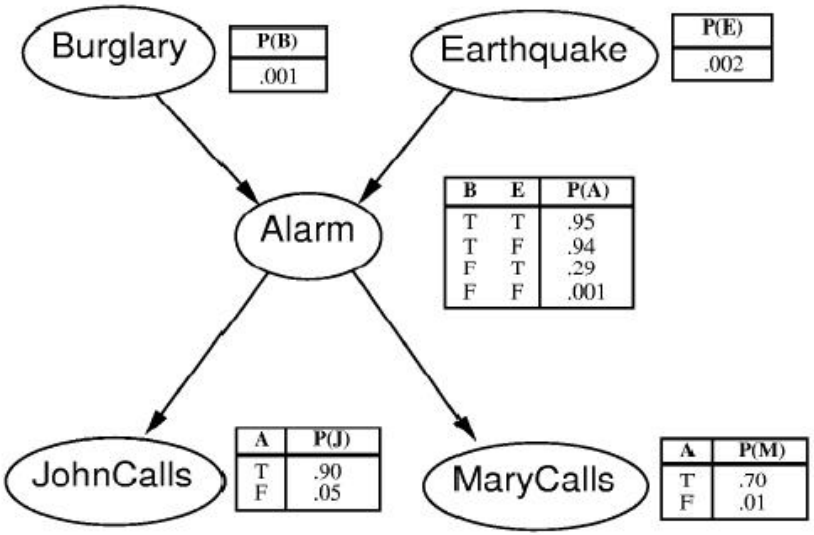
t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2					
3					
4					

- Sampling M from $\mathcal{P}(M|A)$
- $\mathcal{P}(M = \mathtt{T}|\mathtt{A} = \mathtt{F}) \propto 0.01$
  $\mathcal{P}(M = \mathtt{F}|\mathtt{A} = \mathtt{F}) \propto 0.99$



t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2	F	T	T	T	T
3					
4					

- New run $t = 2$
- repeat sampling B, E, A, J, M



t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2	F	T	T	T	T
3	T	F	T	F	T
4	T	F	T	F	F

- Similarly for $t = 3, 4, \ldots$
- Based on the 4 runs (no burn-in) we get e.g.,
  $\hat{\mathcal{P}}(A = \mathtt{T}) = \frac{3}{4}$
  $\hat{\mathcal{P}}(A = \mathtt{T}, \mathtt{B} = \mathtt{F}) = \frac{1}{4}$ etc...

GS: pros and cons

- Pros: usable in both BN and MRF; can take into account all the evidence in every sample.
- Cons: hard to determine when convergence has been achieved; wasteful if MB is large.