DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF COPENHAGEN

Sampling and Markov Chain Monte Carlo (MCMC) Methods, Part I

Kim Steenstrup Pedersen

Plan for today and next lecture

- Basic sampling methods
  - Rejection sampling
  - Importance sampling
  - Sampling-Importance-Resampling (SIR)
- Sampling Bayesian networks
- Markov Chain Monte Carlo (MCMC) methods
  - Metropolis algorithm
  - Metropolis-Hastings algorithm
  - Gibbs sampler

Motivation

Why do we need to do sampling?

Remember the goal of machine learning: We are modelling a mapping of features x into labels/targets $y(x)$ and usually also a probabilistic model $p(x,y)$.

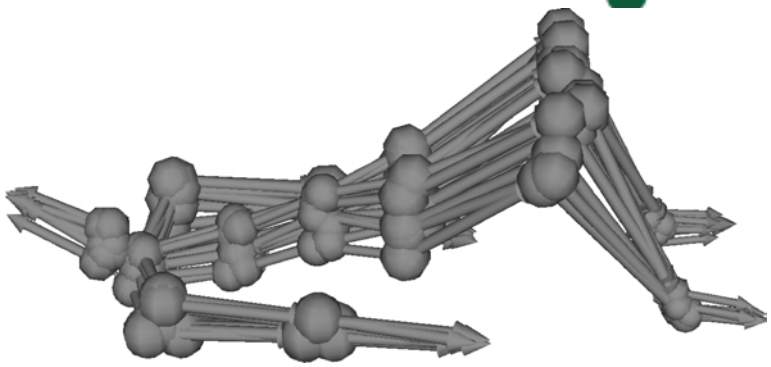
We can use sampling:

- To estimate expectation value, e.g. the mean, of the probability distribution $p(x,y)$.
- To be able to synthesize data for testing purposes.
- Sampling forms an integral part of some machine learning methods. Ex.:
  - Sequential Monte Carlo techniques such as particle filtering (an extension of sampling-importance-resampling).

Example: 3D Human motion tracking

Visual articulated tracking of 3D human motion:

- Learn a distribution $p(y|x)$ of poses y from video sequence x.
- Sample from this pose distribution $p(y|x)$ to get several hypotheses.
- For each pose hypothesis evaluate how well it fits with video data and compute average to get the current estimate of pose.



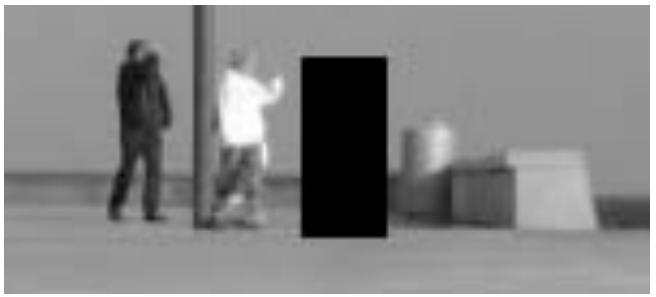
(Hauberg et al, 2010)

Example: Image inpainting: Fill holes of missing pixels

- Synthesize content to fill holes in images.
- Exemplar-based=find similar image patches and paste (puzzle).
- Our approach: Keep several hypotheses in play. E.g. allow for several solution and choose the one that is globally optimal.

Original

Hole



Exemplar approach

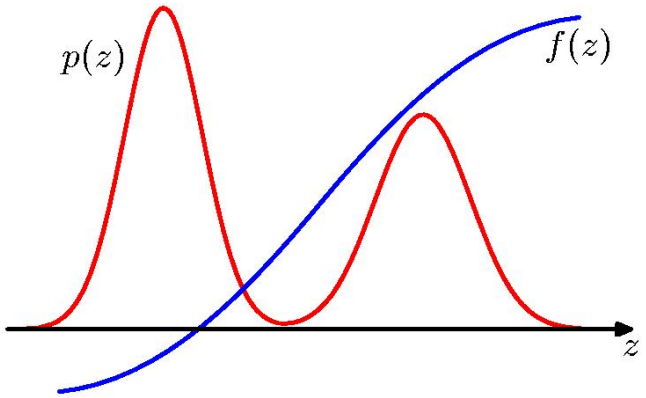
Our approach

Cuzol et al: Field of Particle Filters for Image Inpainting. In Journal of Mathematical Imaging and Vision, 31(2-3): 147-156, 2008.

Estimating expectations using sampling

- We wish to estimate expectations

$$E[f] = \int f(\mathbf{z}) p(\mathbf{z}) d\mathbf{z}$$



- Draw L samples *independently* from $p(\mathbf{z})$ and approximate the expectation with

$$\hat{f} = \frac{1}{L} \sum_{l=1}^{L} f(\mathbf{z}^{(l)}) \approx E[f]$$

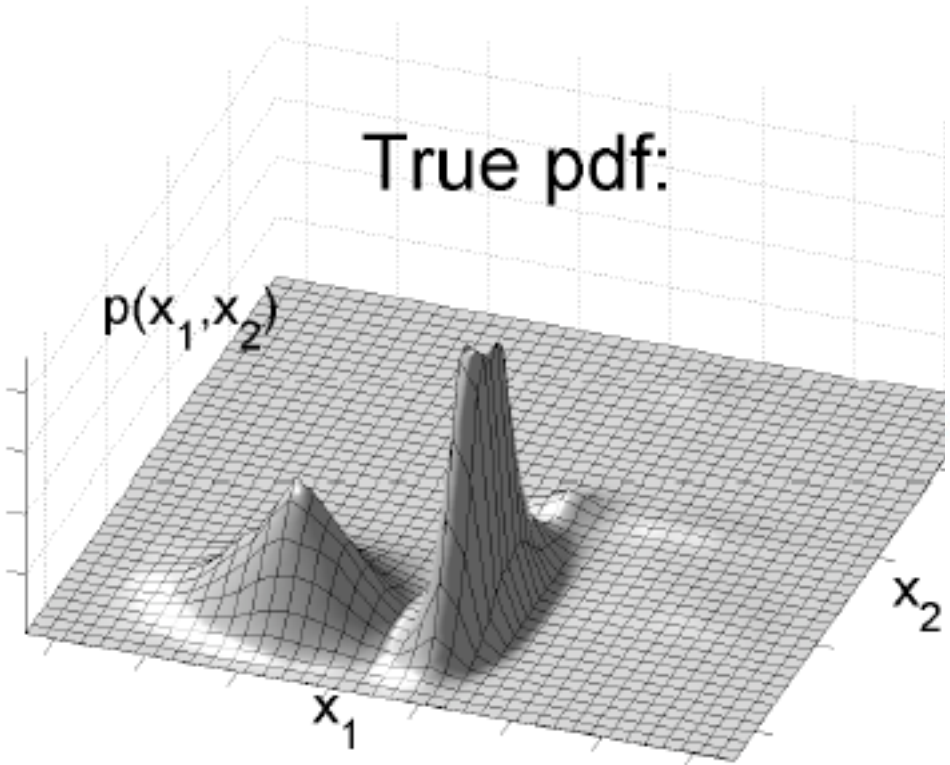
This is an instance of *Monte Carlo integration*.

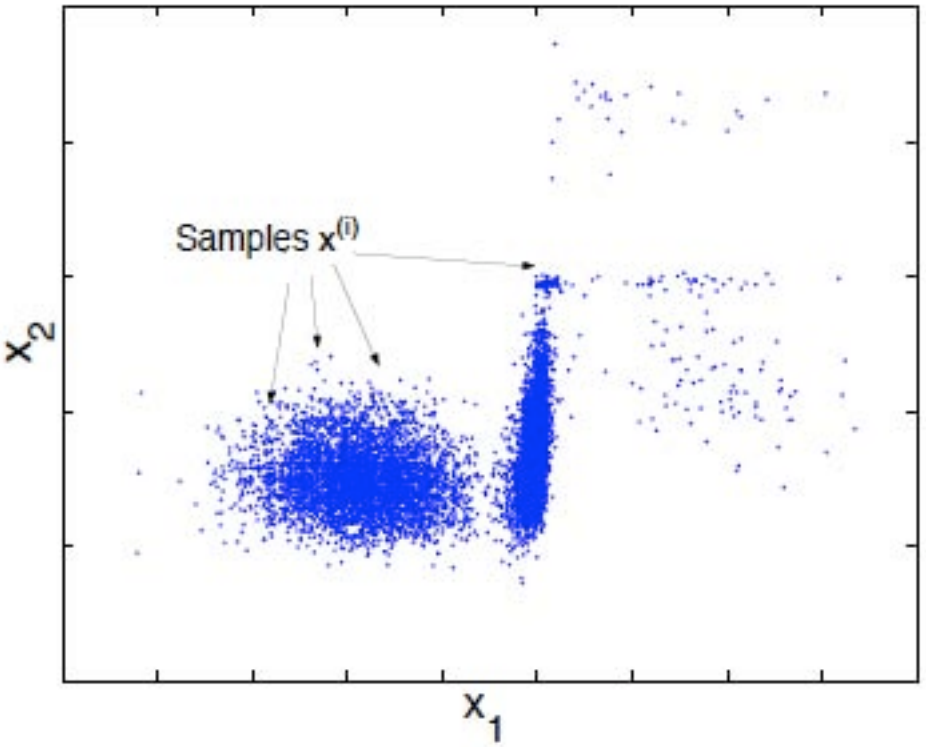
- The samples represent the distribution and in the limit

$$\lim_{L \to \infty} \hat{f} = E[f]$$

The estimator variance is $\mathrm{var}[\hat{f}] = \frac{1}{L} E[(f - E[f])^2]$

An example





Observation: Clearly for this complex distribution we need a lot of samples *L*

Basic Sampling Methods

Sampling from “simple” distributions

- If we know the analytical expression for $p(\mathbf{z})$ - use the transformation method (recall from the StatML course).

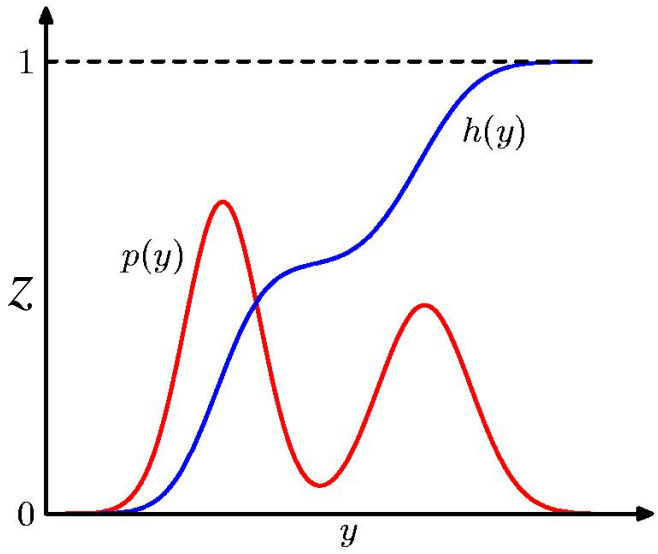
Recall: The transformation method

We want to sample from $p(y)$:

- Assume that z is uniformly distributed $U(z|0,1)$ and define the relationship

$$z = h(y) \equiv \int_{-\infty}^{y} p(\hat{y})d\hat{y}$$

Sample z uniformly and apply $y = h^{-1}(z)$ which is distributed as $p(y)$.



The transformation method applied to discrete distributions

Sampling from $p(X)$:

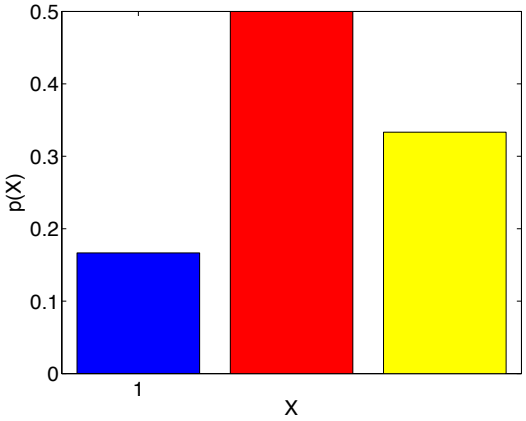
1. Compute the cumulative sum $h(X)$
2. Flatten $h(X)$ on to the unit interval and form lookup table H
3. Draw uniform sample on $z \in [0,1]$
4. Lookup z sample in H and find subinterval

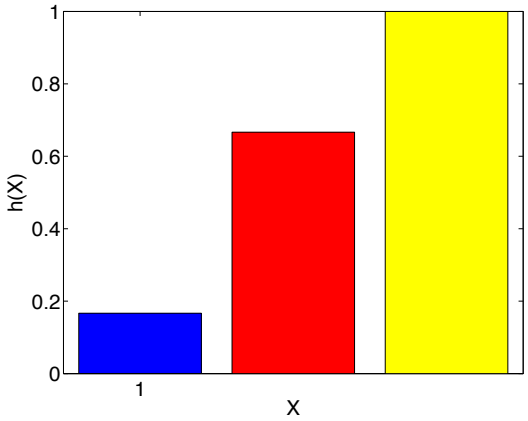
$z = 0.4 \Rightarrow$ Rød

$z = 0.8 \Rightarrow$ Gul

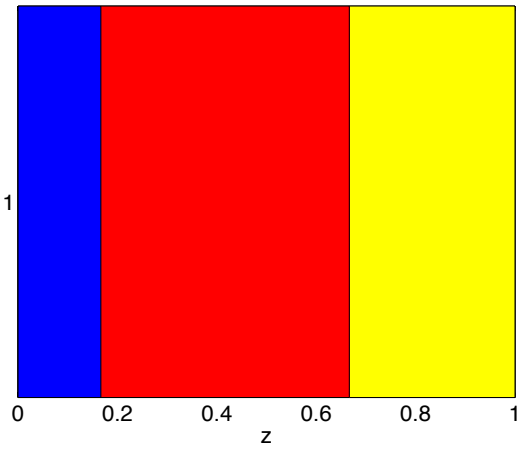
$z = 0.1 \Rightarrow$ Blå

$z = 0.6 \Rightarrow$ Rød





H



Sampling from “simple” distributions

- If we know the analytical expression for $p(\mathbf{z})$ - use the transformation method (recall from the StatML course).

- Specialized algorithms for some standard distributions exist. Ex.:
  - Uniform distribution
  - Gaussian distribution, e.g. the Box-Muller method

Random Number Generators

- True random number generators:
  - Roll a dice, flip a coin, roulette wheel, …
  - Measure random fluctuations at atomic level (e.g. radioactive decay)
  - Measure hard disk head activity, computer clock drift, …

- Pseudo random number generators:
  - Based on a deterministic algorithm that generates a sequence of seemingly random numbers.
  - E.g. Linear congruential generator: $X_{n+1} = \left(aX_n + c\right) \bmod m$
  - Problems: The sequence is finite and deterministic when you know the starting point X_0 , called the random seed.
  - Try out: $m = 5$, $a = 4$, $c = 2$, $X_0 = 0$

    $m = 2^{32}$, $a = 1664525$, $c = 1013904223$, $X_0 = 100$

Pseudo random number generators: Take care!

- Pseudo random number generators are available in most programming languages and as libraries.
- But quality may vary! We want long periods
- At least remember to choose seed “randomly”!
- E.g.: All Matlab random generator functions (`rand`, `randn`, …) uses `rng` as generator.
  - Default seed is always 0!
  - Consequence: You always get the same sequence of random numbers!
  - Unless you choose seed at random, e.g. `rng('shuffle')`
- A standard approach is to use the wall time as seed: E.g. in matlab: `myseed = prod(clock)`

Sampling from “simple” distributions

- When we know the analytical expression for $p(\mathbf{z})$ - use the transformation method (recall from the StatML course).

- Specialized algorithms for some standard distributions exist:
  - Uniform distribution
  - Gaussian distribution, e.g. the Box-Muller method

- But what if our distribution is not standard and we cannot apply the transformation method?

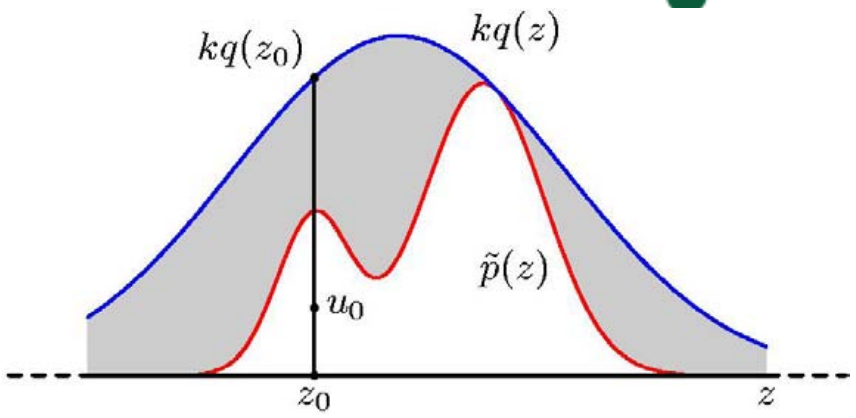
Introducing proposal distributions

- Consider the distribution: $p(\mathbf{z}) = \tilde{p}(\mathbf{z}) / Z_p$

- It may be difficult to sample from the distribution $p(\mathbf{z})$.
- Often $Z_p = \int \tilde{p}(\mathbf{z}) d\mathbf{z}$ is difficult to compute, but $\tilde{p}(\mathbf{z})$ may be evaluated for any **z**.

- Common strategy (used in rejection sampling, importance sampling, Metropolis-Hastings, etc.):
  - Use a much simpler proposal distribution $q(z)$ from which we can sample.
  - Generate a proposal sample and evaluate an acceptance criterion for the sample.

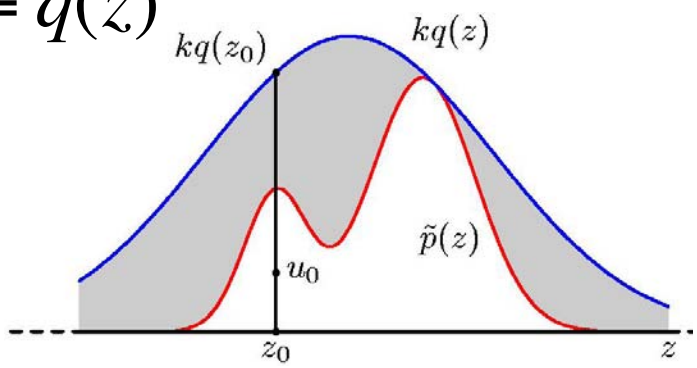
Rejection sampling



- Choose constant *k* and proposal distribution so $kq(z) \geq \tilde{p}(z)$ for all z.
- Sample z_0 from $q(z)$
- Sample u_0 from $\mathcal{U}(u \mid [0, kq(z_0)])$
- Reject z_0, if $u_0 > \tilde{p}(z_0)$ otherwise keep z_0

Assumption: The proposal distribution $q(z)$ must have a support larger than or equal to $p(z)$

Rejection sampling
Do we get the correct result?



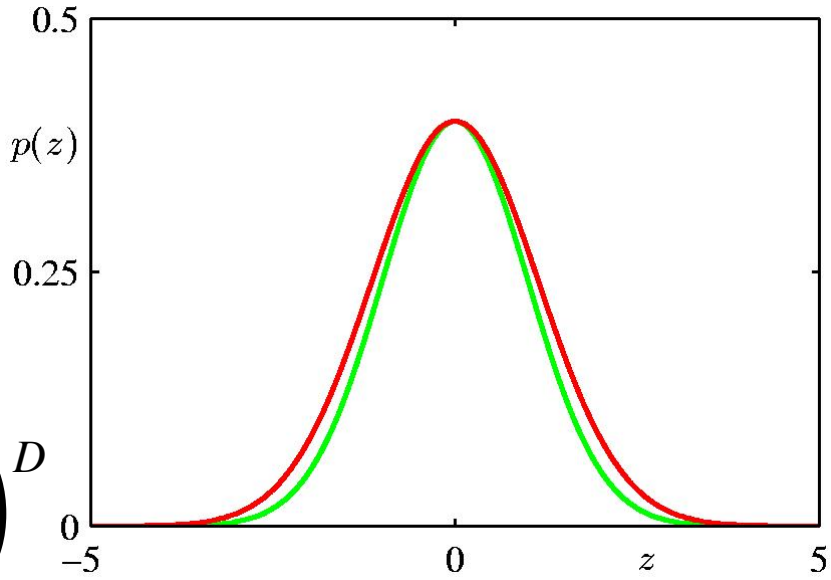
- The probability of a sample is $\Pr(z) = q(z)$
- For a given sample, the probability of acceptance is
  $\Pr(\text{accept} \mid z) = \tilde{p}(z)/kq(z)$
- The probability of acceptance is

$$\Pr(\text{accept}) = \int \Pr(\text{accept} \mid z)\Pr(z)dz = \int \frac{\tilde{p}(z)}{kq(z)} q(z)dz$$
$$= \frac{1}{k}\int \tilde{p}(z)dz = Z_p/k$$

- The distribution of accepted samples is $\Pr(z \mid \text{accept}) =$

$$\frac{\Pr(\text{accept} \mid z)\Pr(z)}{\Pr(\text{accept})} = \frac{\{\tilde{p}(z)/kq(z)\}q(z)}{Z_p/k} = \tilde{p}(z)/Z_p = p(z)$$

Scalability of rejection sampling
What happens when we consider a D-dim z?



- Example: $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}\,|\,0,\sigma_p^2\mathbf{I})$

  $q(\mathbf{z}) = \mathcal{N}(\mathbf{z}\,|\,0,\sigma_q^2\mathbf{I})$

- Where $\sigma_q^2 \geq \sigma_p^2$ so $kq(\mathbf{z}) \geq p(\mathbf{z})$

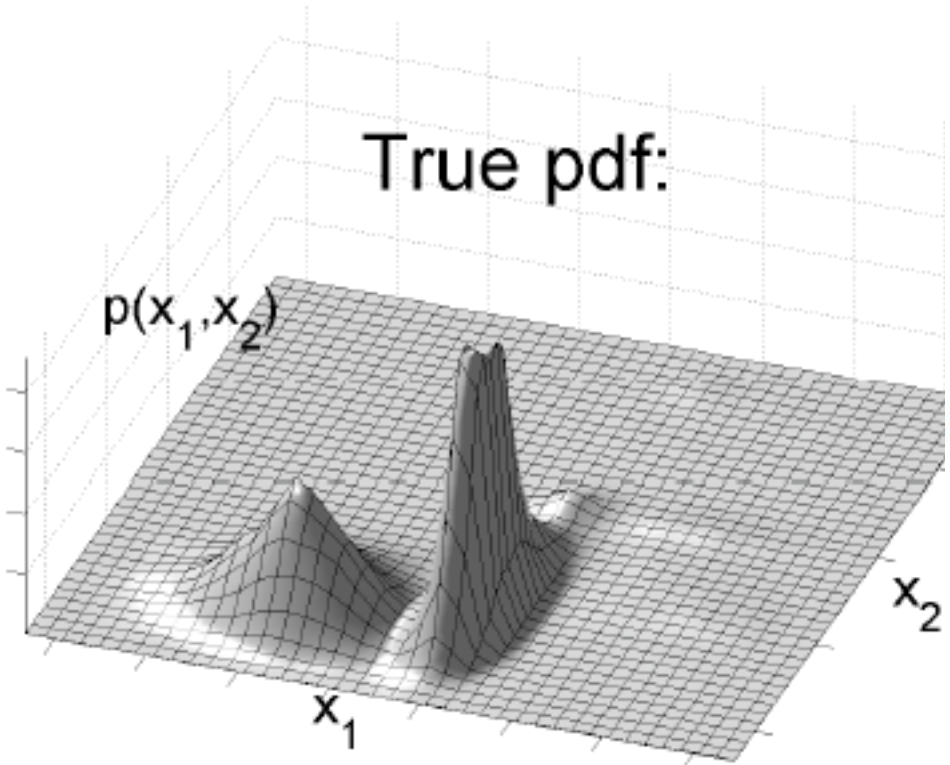
- The optimal choice: $k = \left(\sigma_q / \sigma_p\right)^D$

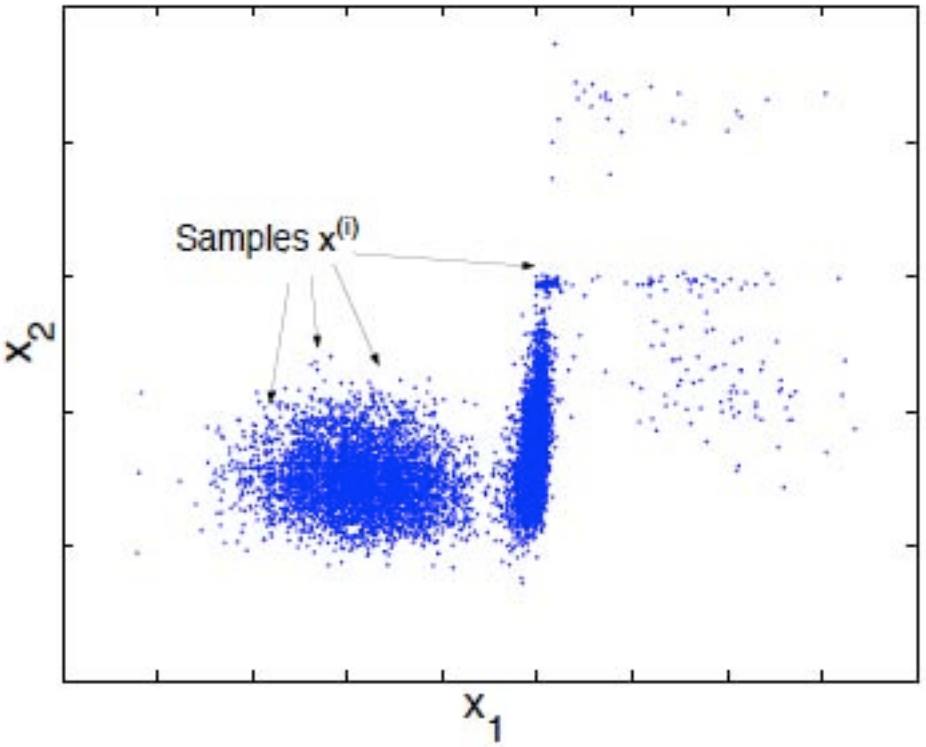
- Hence the acceptance diminishes exponentially with dimensionality

$$\Pr(\text{accept}) = \frac{1}{k}\int p(z)dz = 1/k = \left(\sigma_q/\sigma_p\right)^{-D}$$

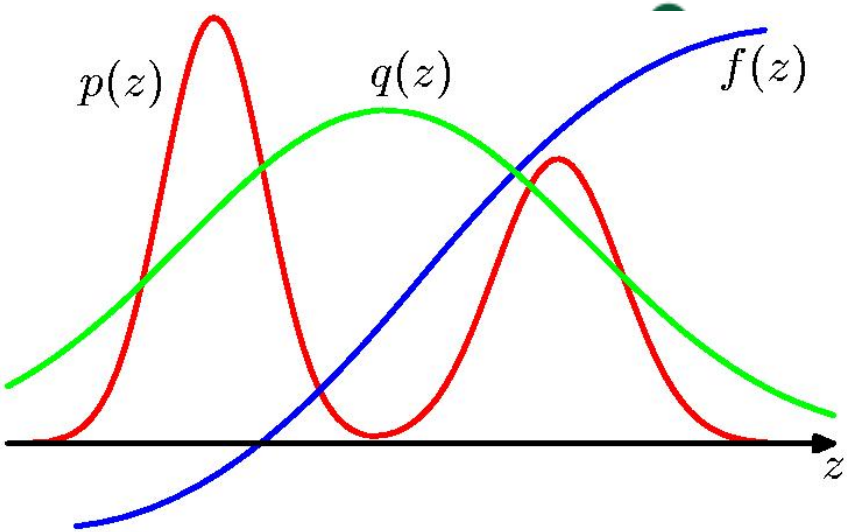
- Conclusion: As D grows more samples will be rejected, so rejection sampling will take more time to get X samples.

An example where rejection sampling is a poor fit





Importance Sampling
Approximate expectation $E[f]$



- Sample i.i.d. from $q(\mathbf{z})$ $(\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(L)})$
- Use samples to approximate $E[f]$ by

$$E[f] = \int f(\mathbf{z})p(\mathbf{z})d\mathbf{z} = \int f(\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})d\mathbf{z}$$
$$\approx \frac{1}{L}\sum_{l=1}^{L}\frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})}f(\mathbf{z}^{(l)})$$

Importance weights $r_l = \frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})}$ compensate for bias.

Importance Sampling
What if we only know $\tilde{p}(\mathbf{z})$ and $\tilde{q}(\mathbf{z})$?

- Assume $p(\mathbf{z}) = 1/Z_p \, \tilde{p}(\mathbf{z})$ and $q(\mathbf{z}) = 1/Z_q \, \tilde{q}(\mathbf{z})$ then

$$E[f] \approx \sum_{l=1}^{L} \omega_l f(\mathbf{z}^{(l)})$$

- Renormalized importance weights

$$\omega_l = \frac{\tilde{r}_l}{\sum_m \tilde{r}_m} = \frac{\tilde{p}(\mathbf{z}^{(l)})/\tilde{q}(\mathbf{z}^{(l)})}{\sum_m \tilde{p}(\mathbf{z}^{(m)})/\tilde{q}(\mathbf{z}^{(m)})}$$

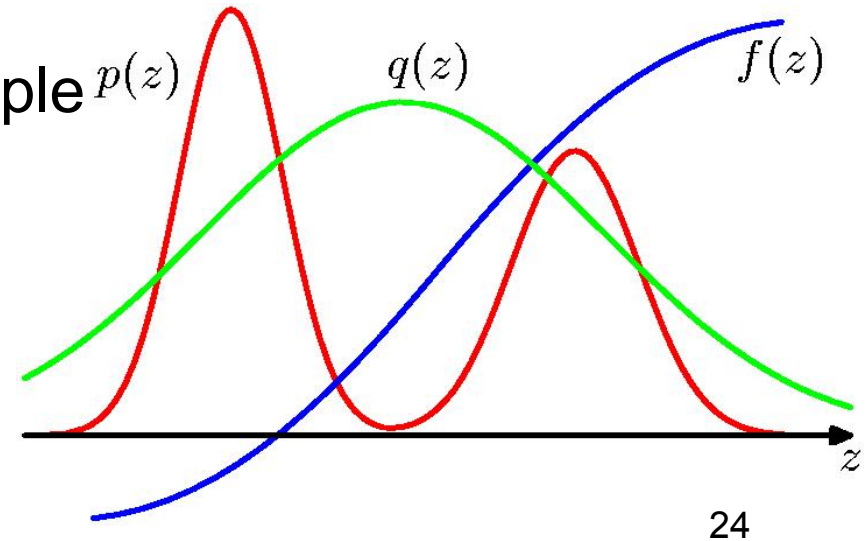
where $\tilde{r}_l = \tilde{p}(\mathbf{z}^{(l)})/\tilde{q}(\mathbf{z}^{(l)})$

Observations on Importance Sampling

- Not really a sampling method, but allow for approximating expectations with a strategy for sampling.

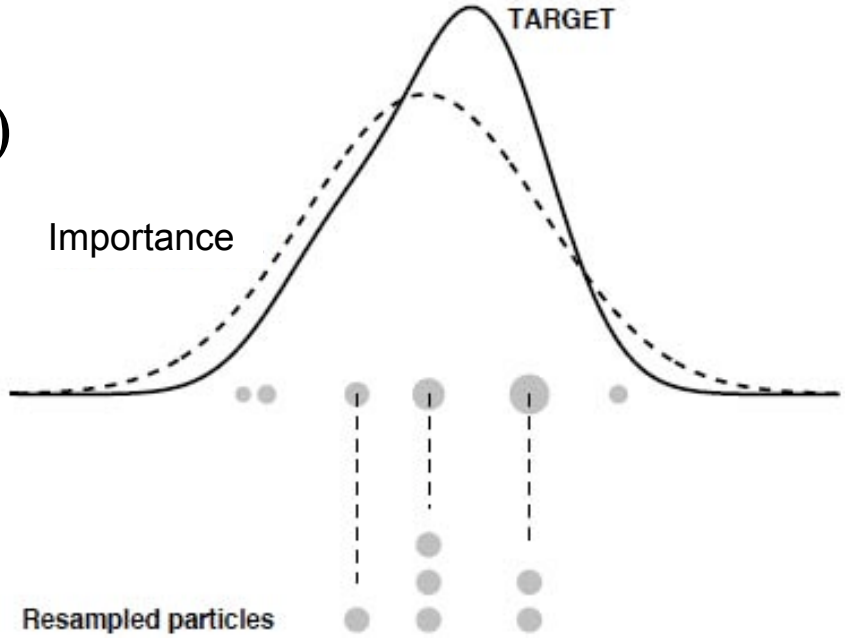
- Potential problems:
  For strongly varying $p(\mathbf{z})f(\mathbf{z})$ there is a risk of a few samples with significant weights.

  This reduces the effective sample size (more samples needed).



Sampling-Importance-Resampling (SIR)
A two stage approach

- Sampling: Sample i.i.d. samples $\mathcal{M} = (\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(M)})$ from $q(\mathbf{z})$
- Importance: Compute importance weights
- Resampling: Sample with replacement from $\mathcal{M}$ based on weights $\omega^{(l)}$ $(\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(N)})$
  Usually $M \geq N$



Relevant for sequential Monte Carlo methods
(more on this in an upcoming lecture)

Examples of sampling for Bayesian networks

Sampling a Bayesian network

- Ancestral sampling with no evidence variables

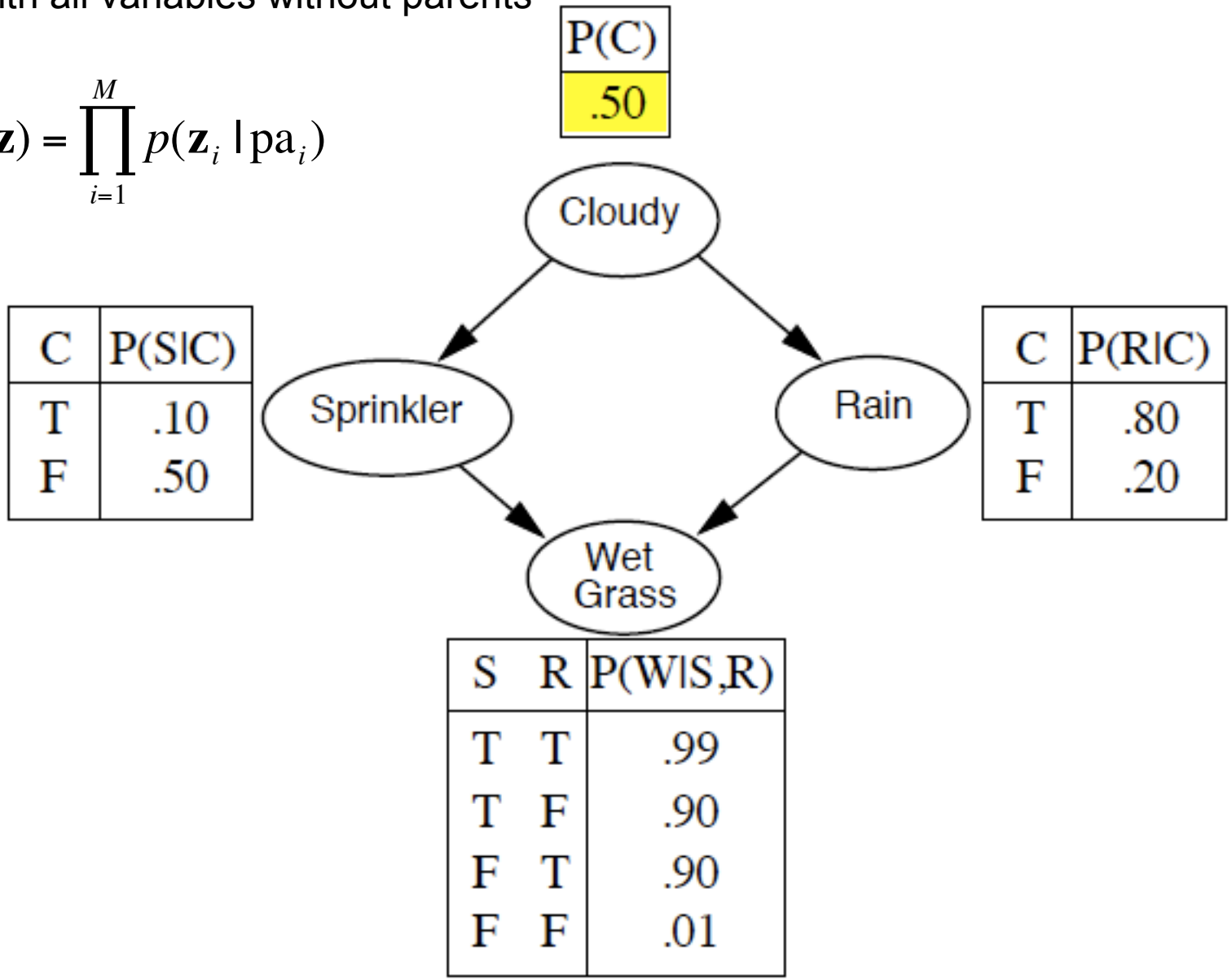
$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$

- Sampling with evidence variables

Ancestral sampling

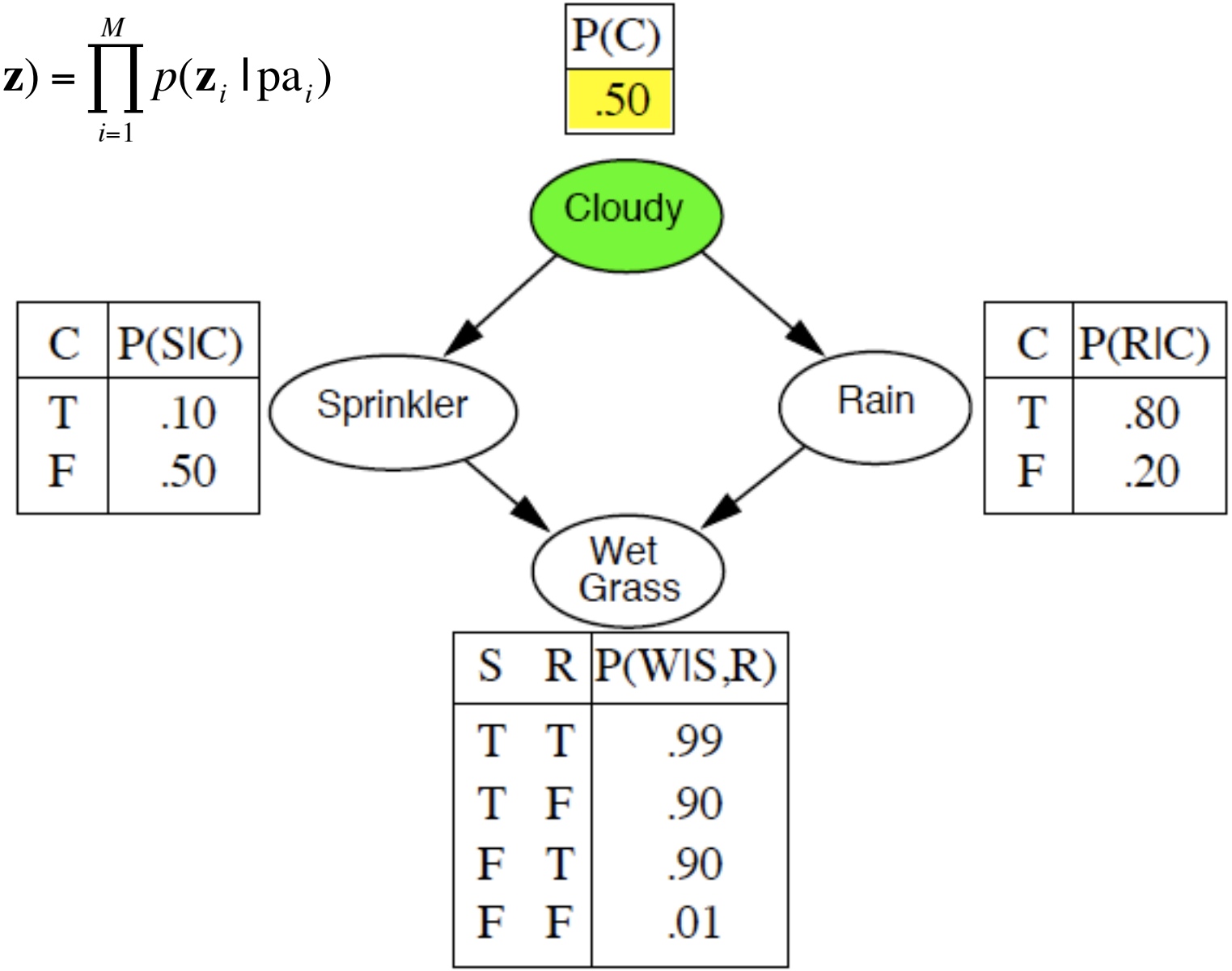
Start with all variables without parents and do

$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$



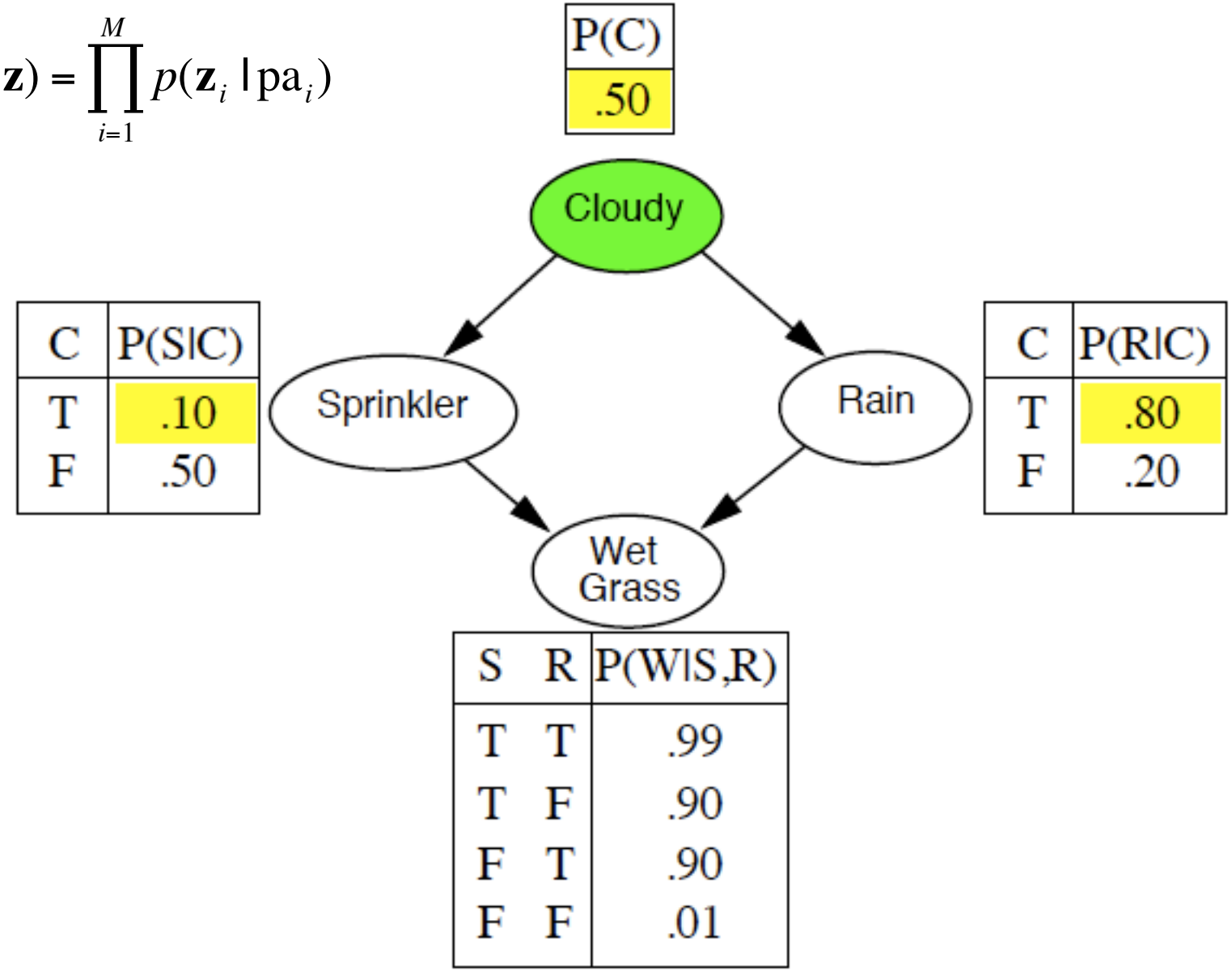
Ancestral sampling

$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$



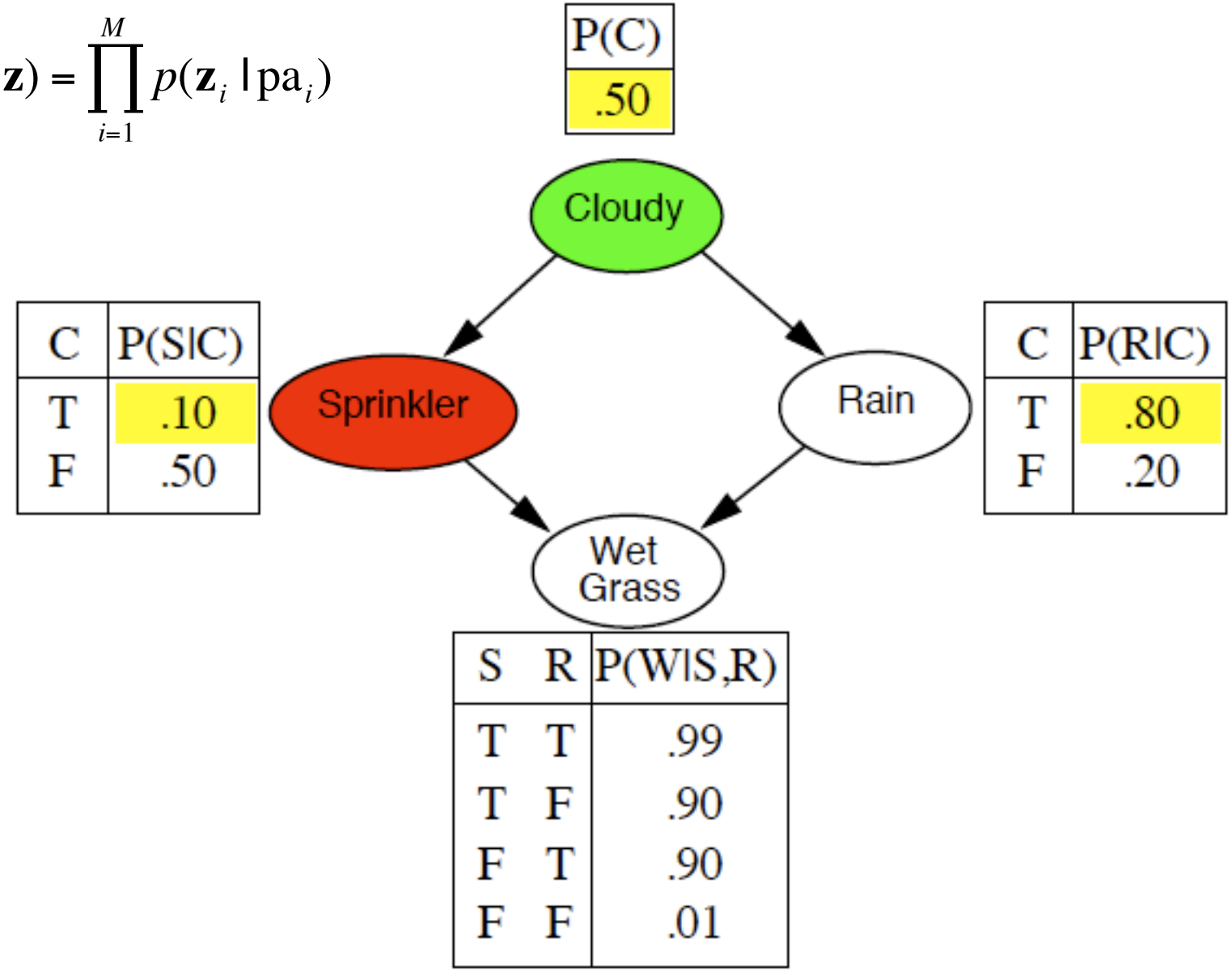
Ancestral sampling

$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$



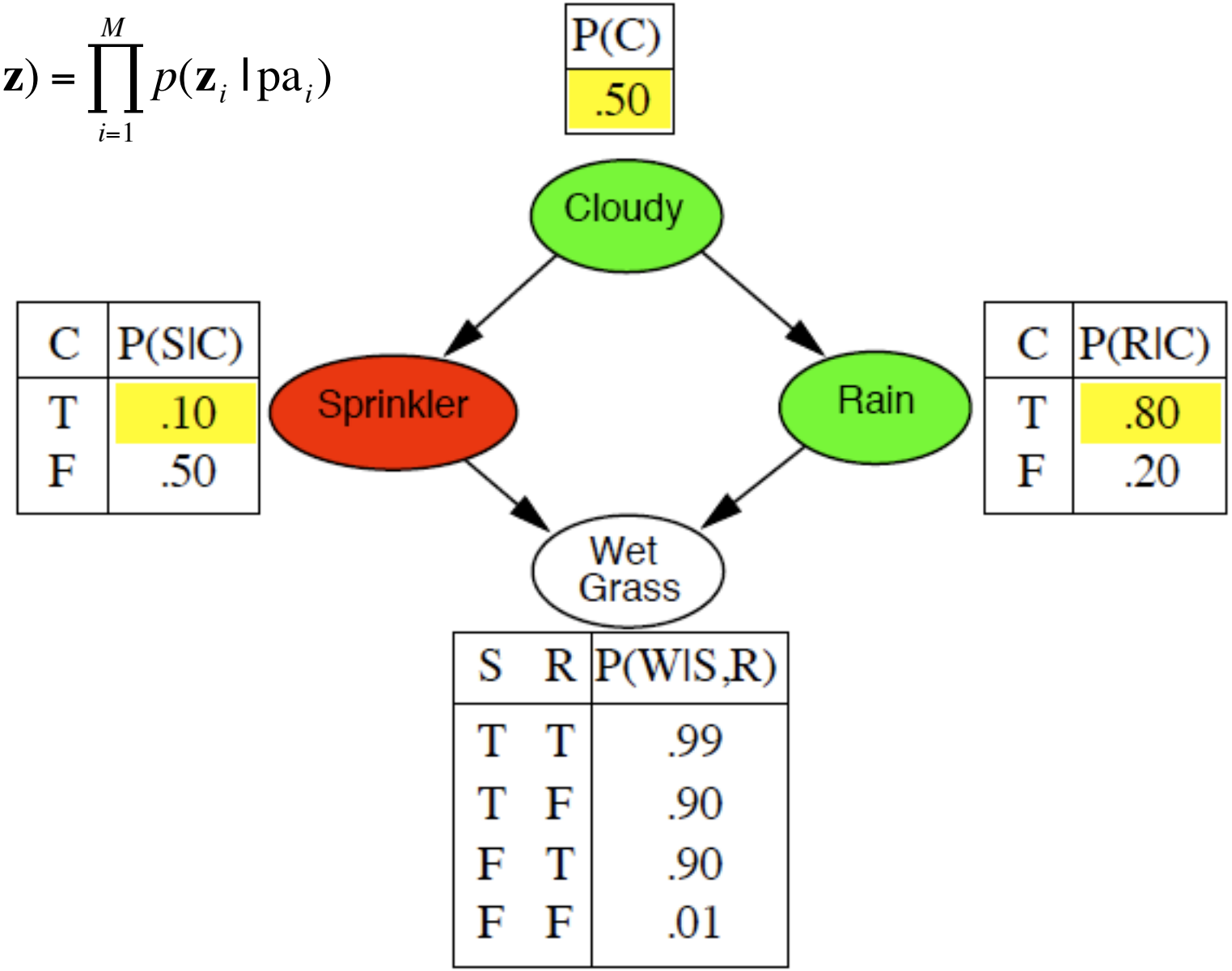
Ancestral sampling

$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$



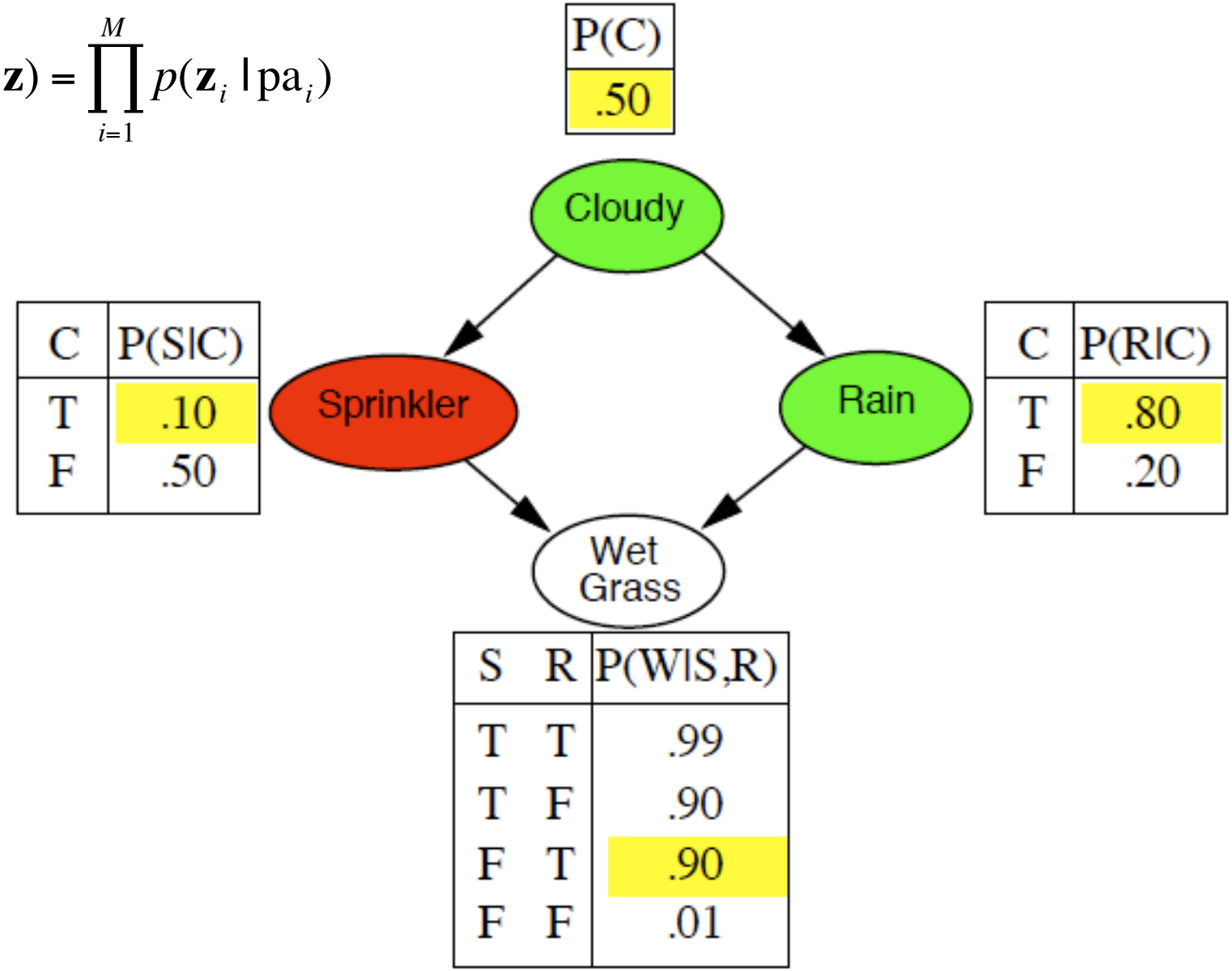
Ancestral sampling

$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$



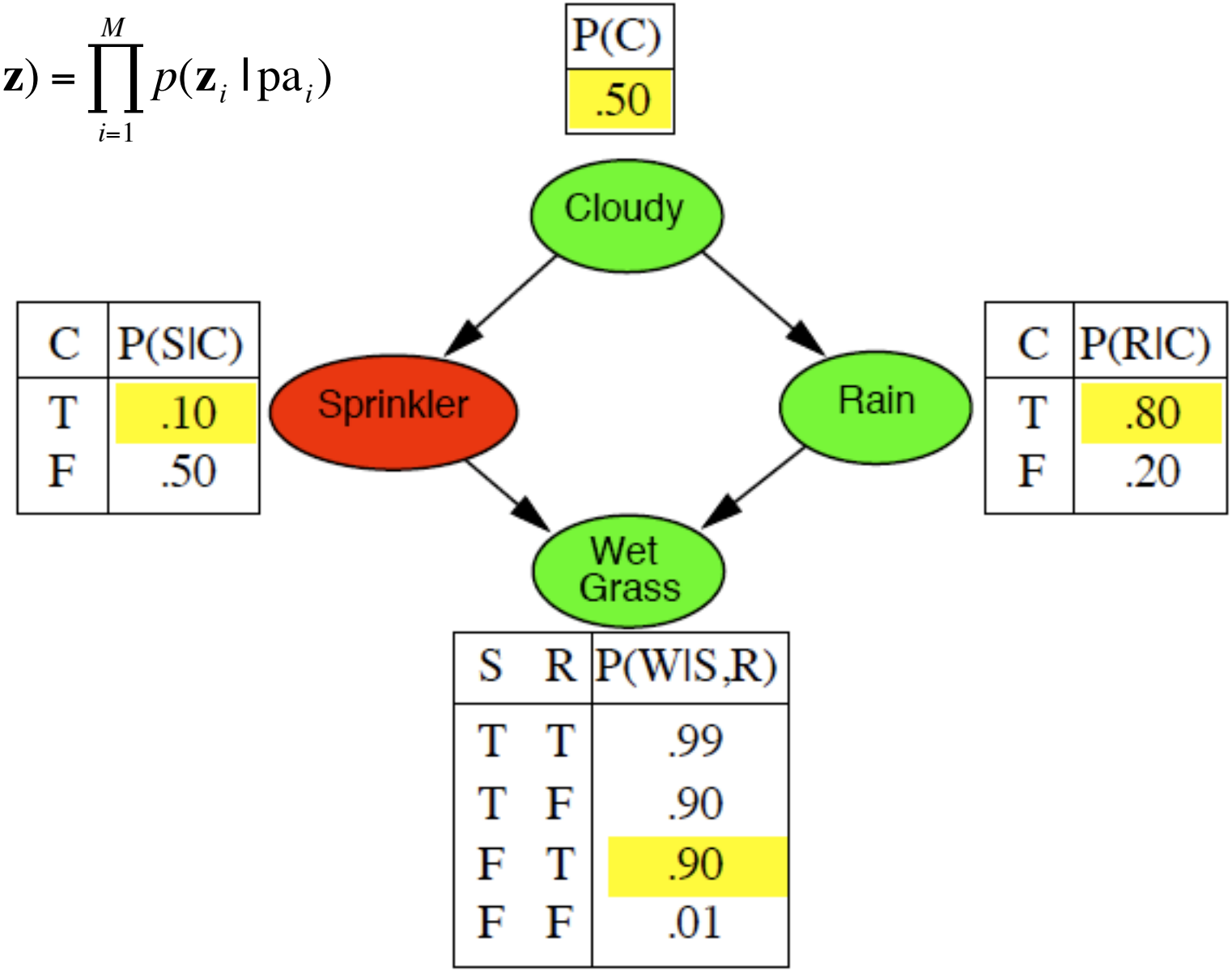
Ancestral sampling

$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$



Ancestral sampling

$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$



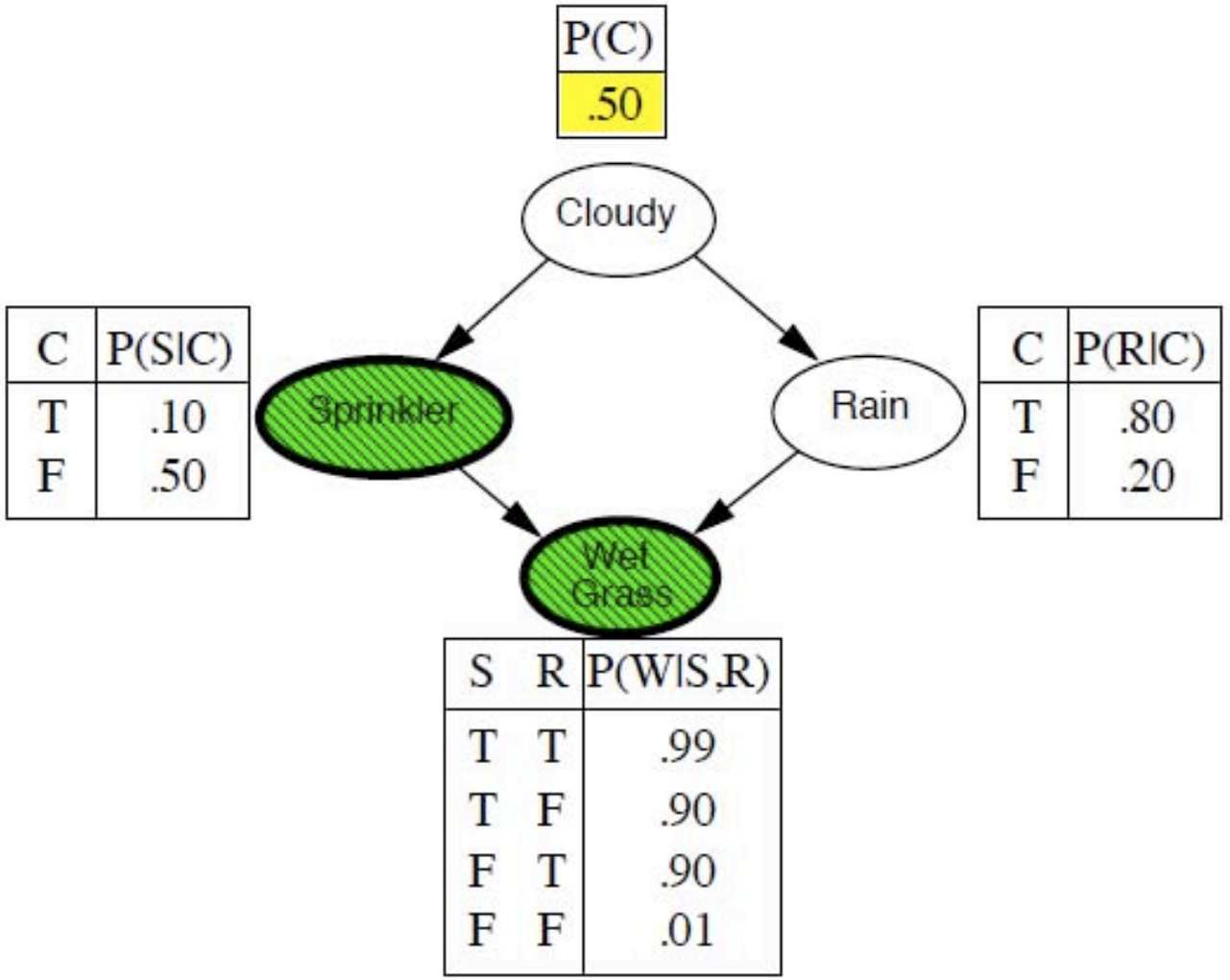
Sampling a Bayesian network

- Ancestral sampling with no evidence / observations variables

$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$

- Sampling with evidence / observations variables:
  - Rejection sampling (reject samples that do not fit evidence)

Rejection Sampling Example:



Sample all variables using ancestral sampling and reject those samples that do not fit with observations

This is a very inefficient approach!

Sampling a Bayesian network

- Ancestral sampling with no evidence / observations variables

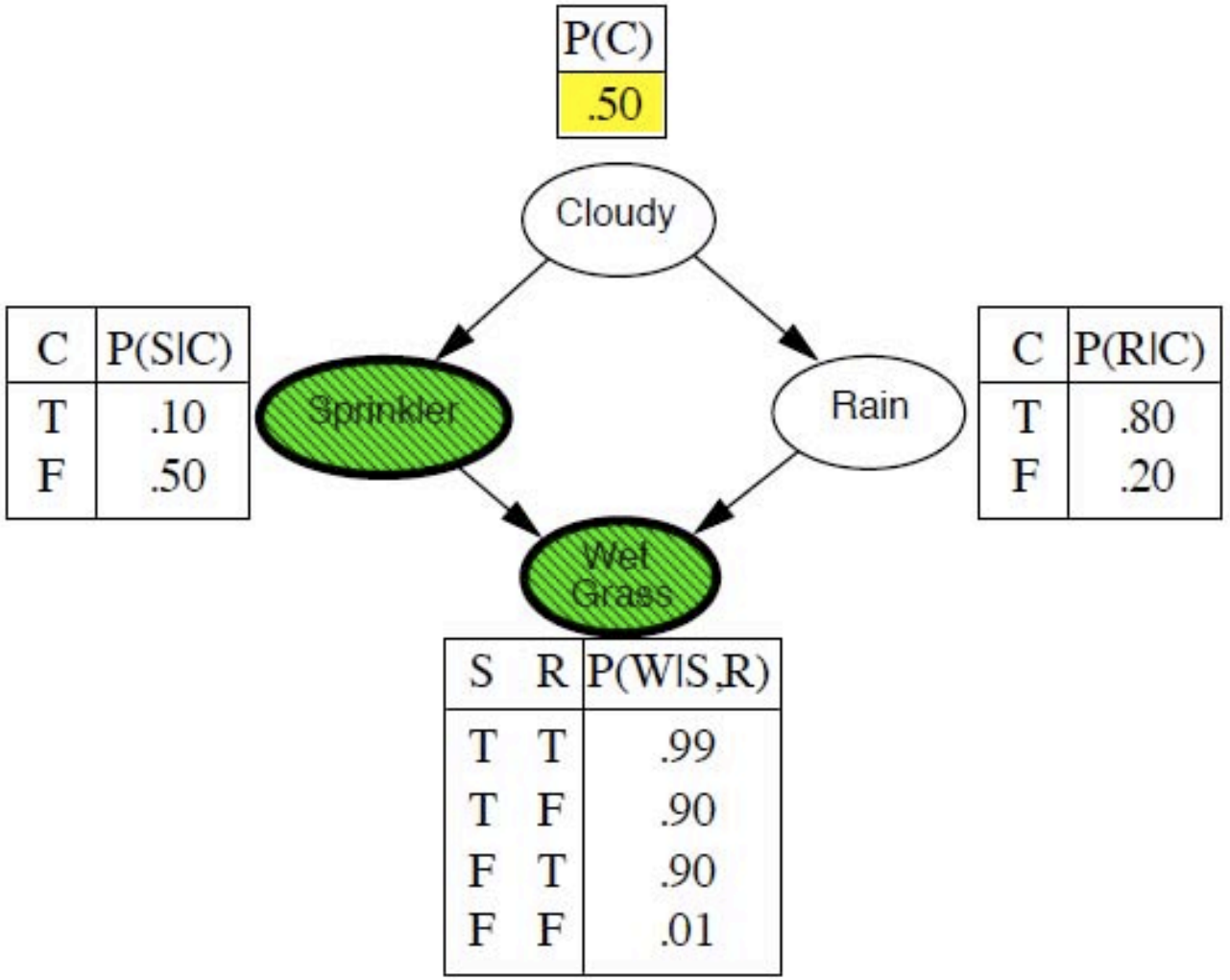
$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$

- Sampling with evidence / observations variables:
  - Rejection sampling (reject samples that do not fit evidence)
  - Likelihood weighted sampling (importance sampling for BN):
    - Proposal distribution $q(\mathbf{z}) = \prod_{\mathbf{z}_i \notin e} p(\mathbf{z}_i \mid \mathrm{pa}_i)$
    - Importance weights

$$r_l = \frac{p(\mathbf{z})}{q(\mathbf{z})} = \frac{\prod_{\mathbf{z}_i \notin e} p(\mathbf{z}_i \mid \mathrm{pa}_i) \prod_{\mathbf{z}_i \in e} p(\mathbf{z}_i \mid \mathrm{pa}_i)}{\prod_{\mathbf{z}_i \notin e} p(\mathbf{z}_i \mid \mathrm{pa}_i)} = \prod_{\mathbf{z}_i \in e} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$

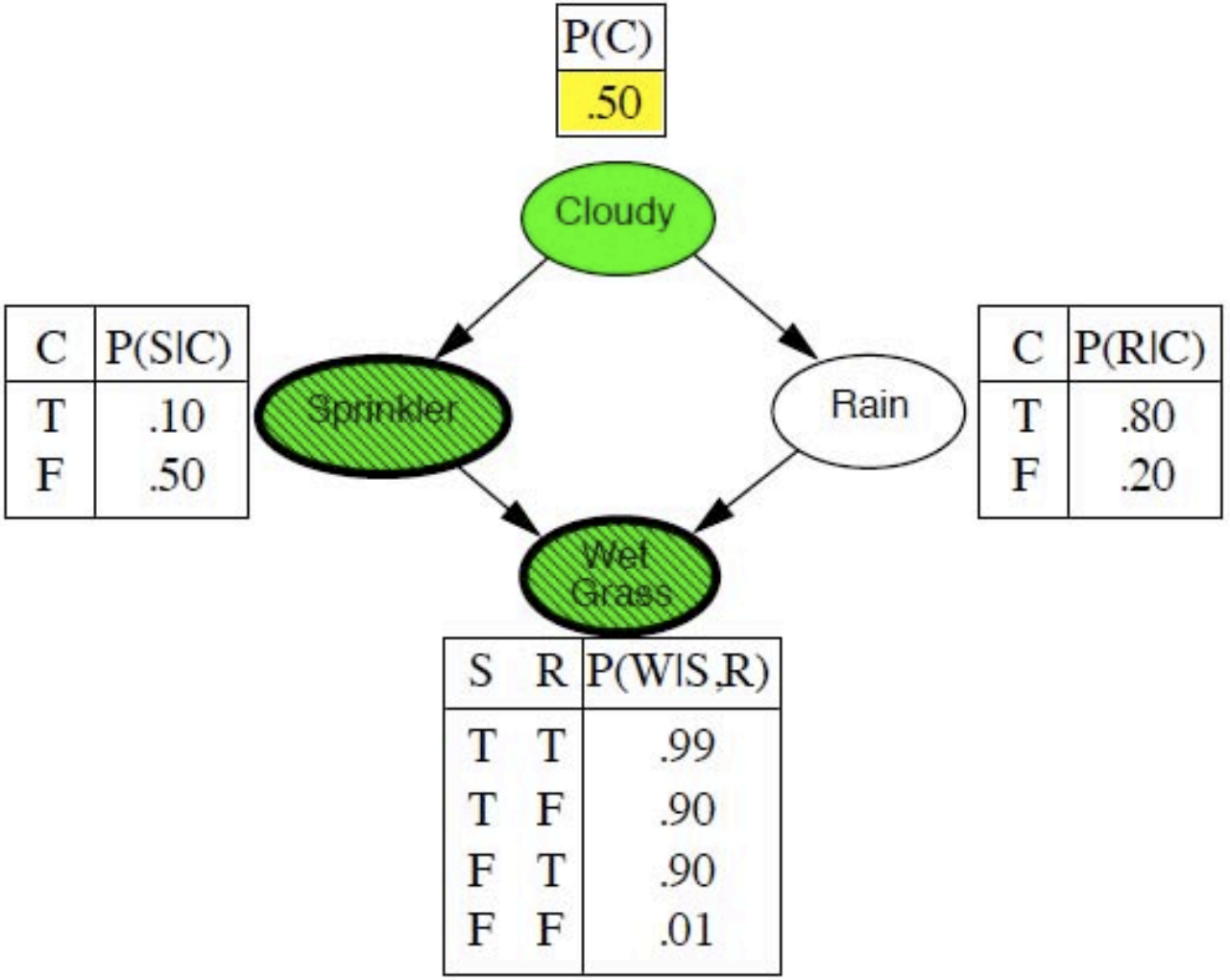
    - Approximate expectation $E[f] \approx \sum_{l=1}^{L} r_l f(\mathbf{z}_l)$

Likelihood weighting example



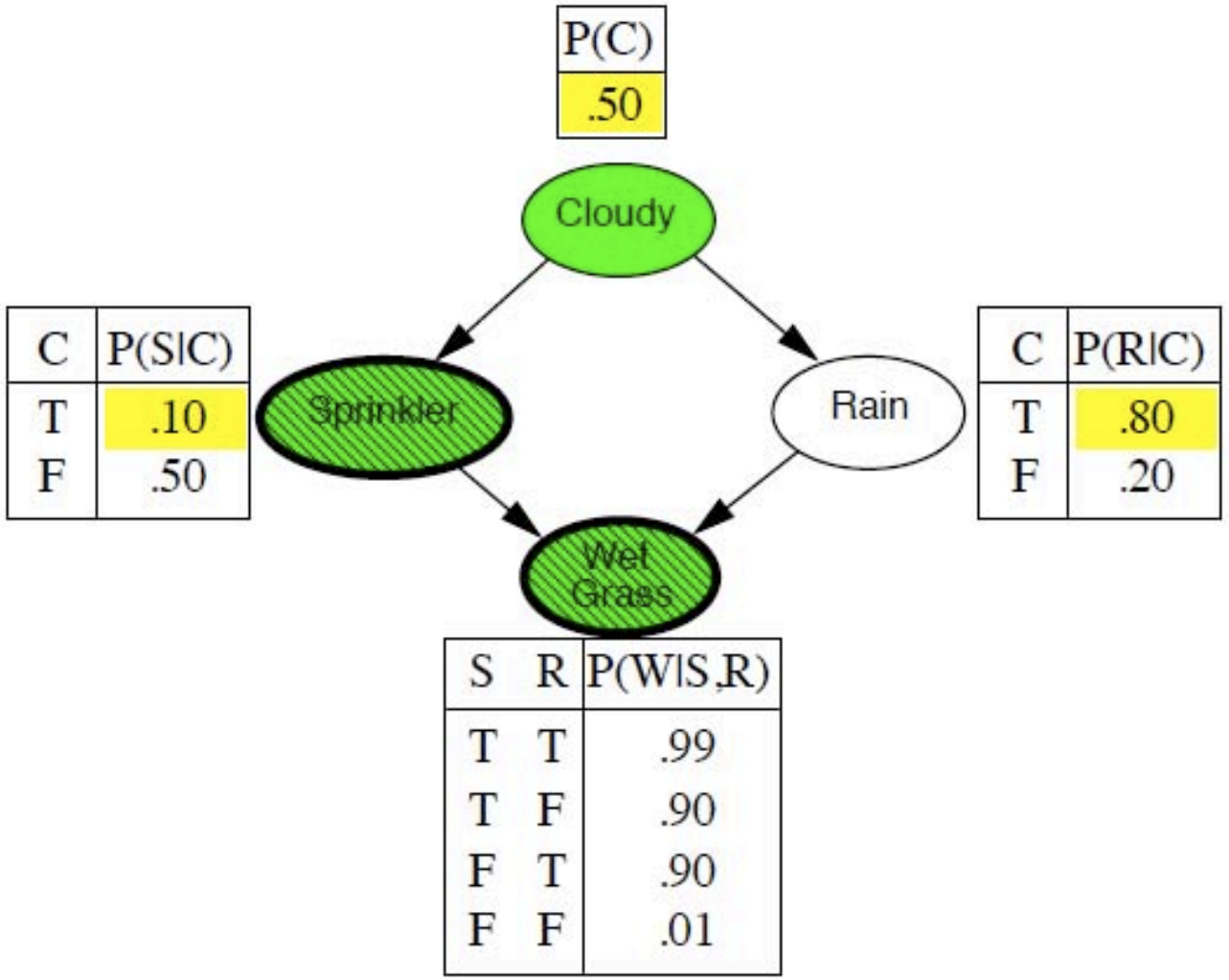
$r_l = 1.0$

Likelihood weighting example



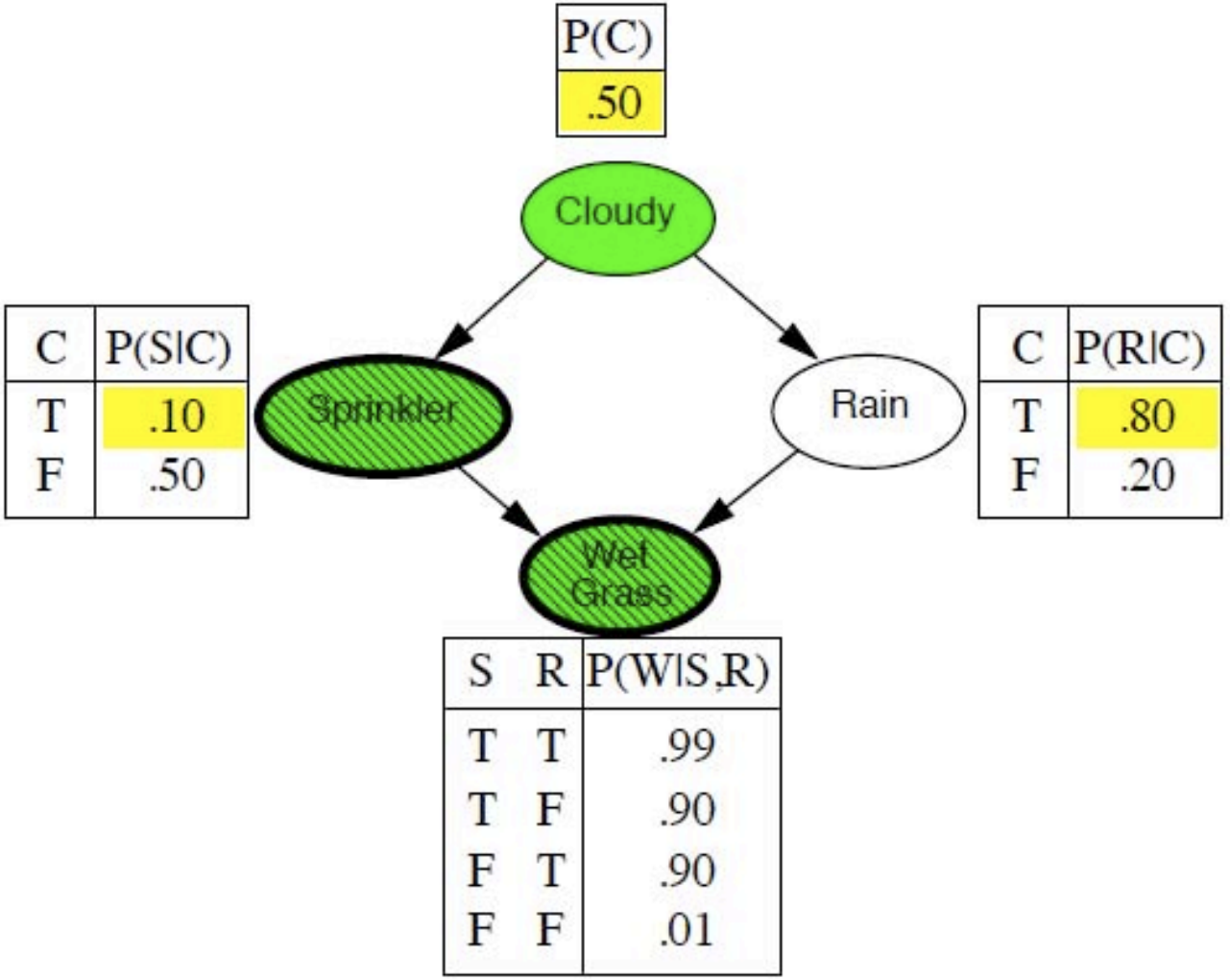
$r_l = 1.0$

Likelihood weighting example



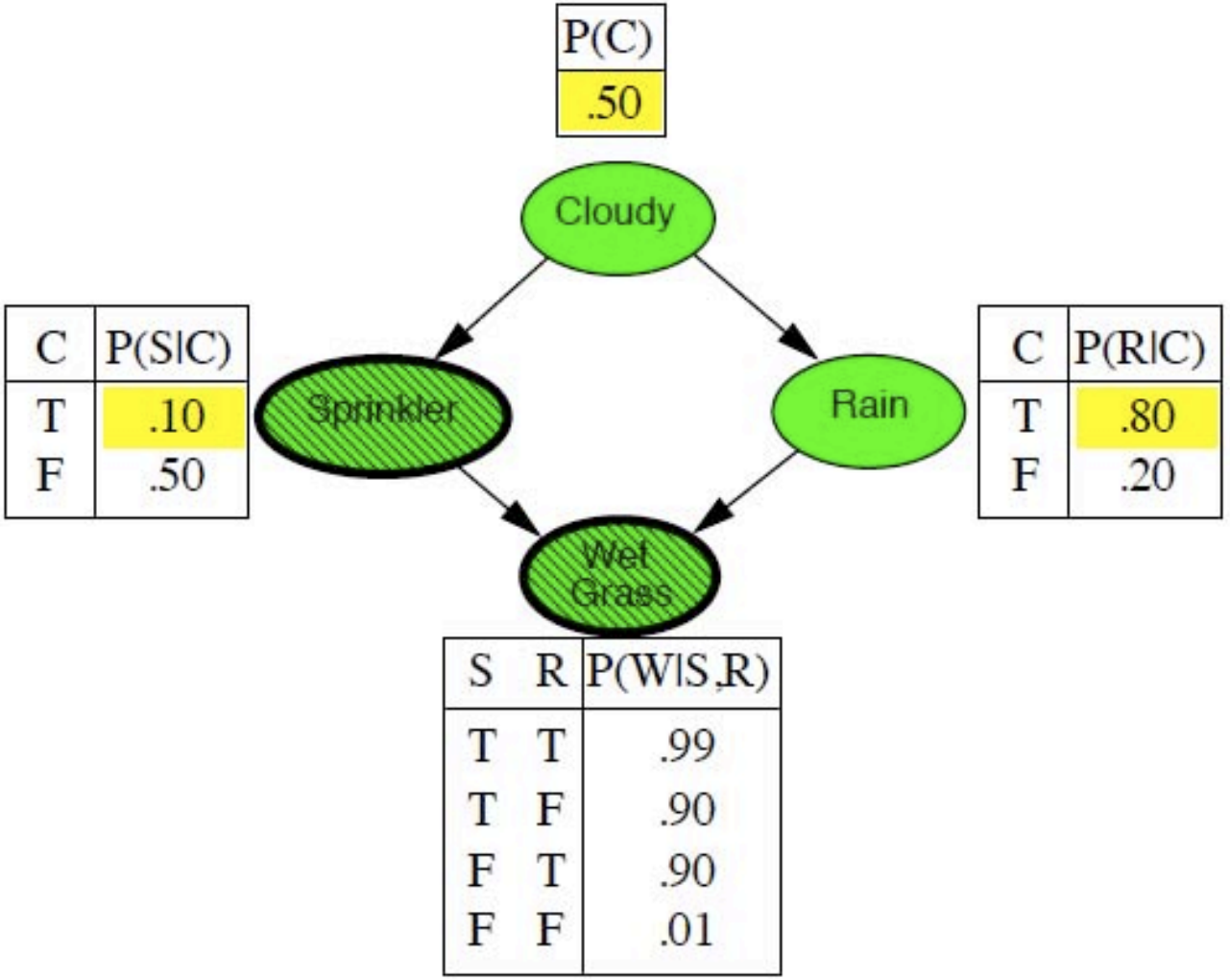
$r_l = 1.0$

Likelihood weighting example



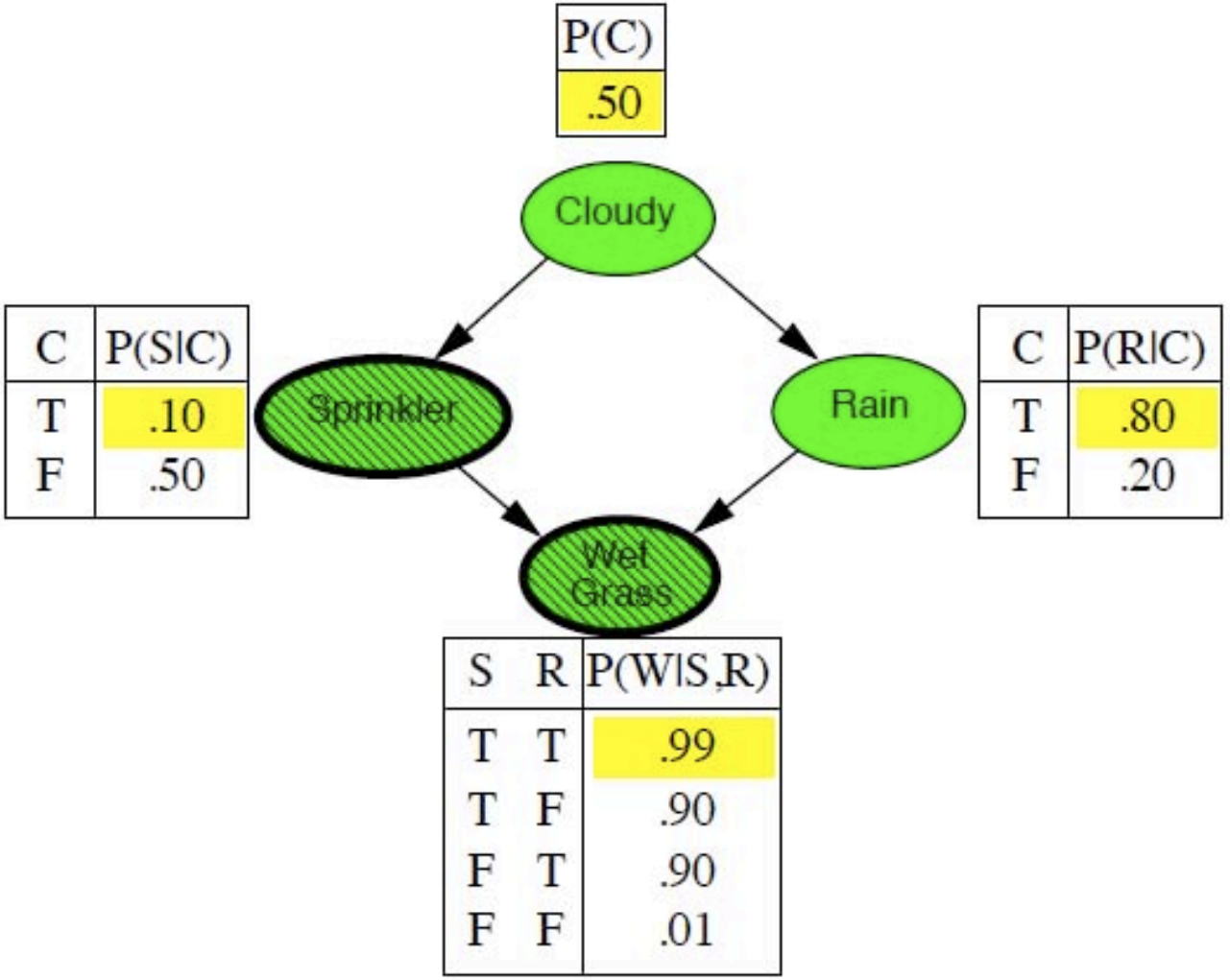
$r_l = 1.0 \times 0.1$

Likelihood weighting example



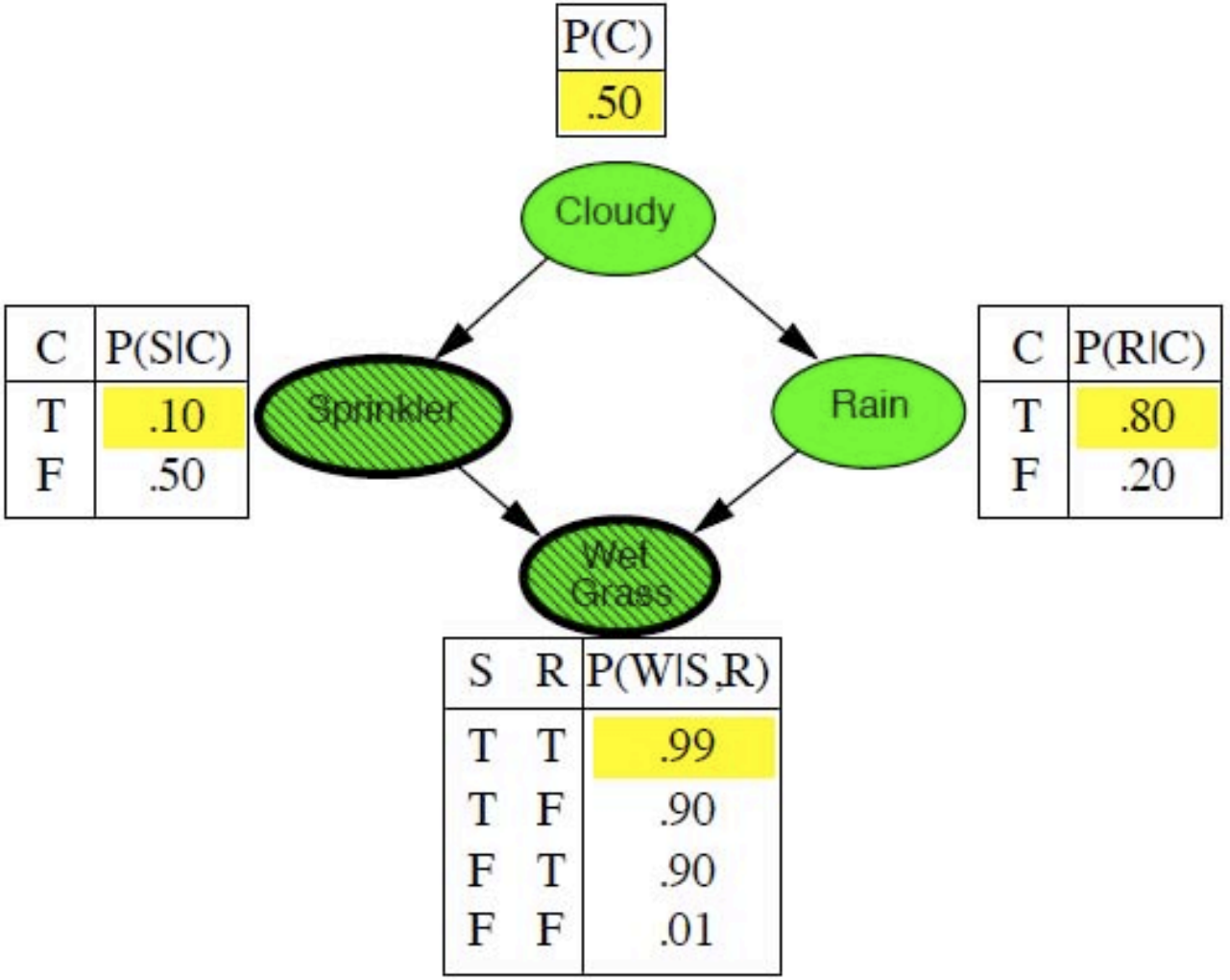
$r_l = 1.0 \times 0.1$

Likelihood weighting example



$r_l = 1.0 \times 0.1$

Likelihood weighting example



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

Sampling a Bayesian network

- Ancestral sampling with no evidence / observations variables

$$p(\mathbf{z}) = \prod_{i=1}^{M} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$

- Sampling with evidence / observations variables:
  - Rejection sampling (reject samples that do not fit evidence)
  - Likelihood weighted sampling (importance sampling for BN):
    - Proposal distribution $q(\mathbf{z}) = \prod_{\mathbf{z}_i \notin e} p(\mathbf{z}_i \mid \mathrm{pa}_i)$
    - Importance weights

$$r_l = \frac{p(\mathbf{z})}{q(\mathbf{z})} = \frac{\prod_{\mathbf{z}_i \notin e} p(\mathbf{z}_i \mid \mathrm{pa}_i) \prod_{\mathbf{z}_i \in e} p(\mathbf{z}_i \mid \mathrm{pa}_i)}{\prod_{\mathbf{z}_i \notin e} p(\mathbf{z}_i \mid \mathrm{pa}_i)} = \prod_{\mathbf{z}_i \in e} p(\mathbf{z}_i \mid \mathrm{pa}_i)$$

    - Approximate expectation $E[f] \approx \sum_{l=1}^{L} r_l f(\mathbf{z}_l)$

Summary

- Basic sampling methods
  - Rejection sampling
  - Importance sampling
  - Sampling-Importance-Resampling (SIR)
- Sampling Bayesian networks
- Up next: Markov Chain Monte Carlo (MCMC) methods

Literature

- Basic sampling: CB Sec. 11. – 11.1.5
- Ancestral sampling: CB Sec. 8.1.2
- Likelihood weighted sampling: CB Sec. 11.1.4
- MCMC methods: CB Sec. 11.2 – 11.3

- Suggestions for further reading on MCMC:
  - Pierre Brémaud. Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues. Springer, 1999.
  - Gerhard Winkler. Image Analysis, Random Fields and Markov Chain Monte Carlo Methods – A Mathematical Introduction. Springer, 2nd edition, 2003.