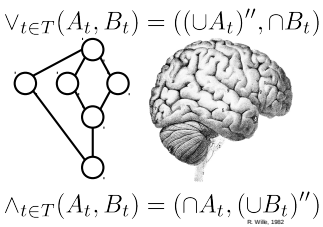
Sampling Methods

Dominik Endres
with slides by Prof.Dr. Cristobal Curio, FH Reutlingen

February 3, 2016

Philipps Universität Marburg



Outline

1. Why sampling?
2. Basic Sampling Algorithms
   - Rejection sampling
   - Importance sampling
3. Markov Chain Monte Carlo (MCMC)
   - Markov Chain methods
   - Gibbs Sampling

Sampling Methods Motivation

Why:

- Useful when deterministic approximations don't work (or we're too lazy)
- Stochastic approximations (Sampling methods)

Pro:

- Given infinite computational resources, they can generate exact results.
- Approximation arises from the use of a finite amount of processor time.
- Enable the use of Bayesian techniques across many domains.

Con: computationally demanding, often limited to small-scale problems.

Why sampling?
The summation/ integration problem

Almost everything we have learned about summation is valid for integration.

Normalization (Θ: e.g. parameters of a model, $\mathbf{x}$: evidence/data)
To obtain the posterior $p(\Theta|\mathbf{x})$ given the prior $p(\Theta)$ and the likelihood $p(\mathbf{x}|\Theta)$, the normalizing factor in Bayes theorem needs to be computed.

$$p(\Theta|\mathbf{x}) = \frac{p(\mathbf{x}|\Theta)\, p(\Theta)}{\int_{states} p(\mathbf{x}|\Theta)\, p(\Theta)\, d\Theta}$$

Example: making a prediction (Θ: est. model parameters, x: state, $\mathcal{D}$: evidence)

$$p(x|\mathcal{D}) = \int P(x|\Theta, \mathcal{D})\, P(\Theta|\mathcal{D})\, d\Theta$$

Why sampling?
The summation/ integration problem

Almost everything we have learned about summation is valid for integration.

Normalization (Θ: e.g. parameters of a model, $\mathbf{x}$: evidence/data)
To obtain the posterior $p(\Theta|\mathbf{x})$ given the prior $p(\Theta)$ and the likelihood $p(\mathbf{x}|\Theta)$, the normalizing factor in Bayes theorem needs to be computed.

$$p(\Theta|\mathbf{x}) = \frac{p(\mathbf{x}|\Theta)\, p(\Theta)}{\int_{states} p(\mathbf{x}|\Theta)\, p(\Theta)\, d\Theta}$$

Example: making a prediction (Θ: est. model parameters, x: state, $\mathcal{D}$: evidence)

$$p(x|\mathcal{D}) = \int P(x|\Theta, \mathcal{D})\, P(\Theta|\mathcal{D})\, d\Theta$$

Monte Carlo principle

Simple Monte Carlo

Idea: Sample from $p(x)$, average values of $f(x)$:

$$\textit{Estimator}: \int f(x)\, p(x)\, dx \approx \hat{f} \equiv \frac{1}{S} \sum_{s=1}^{S} f\left(x^{(s)}\right),\ x^{(s)} \sim p(x)$$

where $x^{(s)}$ are (independent) samples drawn from $p(x)$.

Attractions:

Estimator is unbiased:

$$\mathbb{E}_{P(\{x^{(s)}\})}\left[\hat{f}\right] = \frac{1}{S} \sum_{s=1}^{S} \mathbb{E}_{P(x^{(s)})}\left[f(x^{(s)})\right] = \mathbb{E}_{P(x)}\left[f(x)\right]$$

Variance shrinks $\propto 1/S$ independent of dimension!:

$$var_{P(\{x^{(s)}\})}\left[\hat{f}\right] = \frac{1}{S^2} \sum_{s=1}^{S} var_{P(x^{(s)})}\left[f(x^{(s)})\right] = var_{P(x)}\left[f(x)\right]/S$$

Monte Carlo principle

Simple Monte Carlo

Idea: Sample from $p(x)$, average values of $f(x)$:

$$\textit{Estimator}: \int f(x)\, p(x)\, dx \approx \hat{f} \equiv \frac{1}{S}\sum_{s=1}^{S} f\left(x^{(s)}\right),\ x^{(s)} \sim p(x)$$

where $x^{(s)}$ are (independent) samples drawn from $p(x)$.

Attractions:

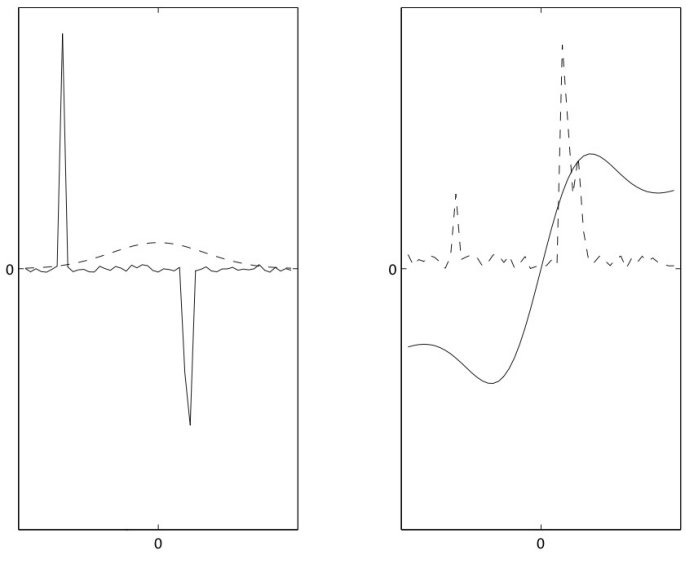
Estimator is unbiased:

$$\mathbb{E}_{P\left(\{x^{(s)}\}\right)}\left[\hat{f}\right] = \frac{1}{S}\sum_{s=1}^{S}\mathbb{E}_{P\left(x^{(s)}\right)}\left[f\left(x^{(s)}\right)\right] = \mathbb{E}_{P(x)}\left[f(x)\right]$$

Variance shrinks $\propto 1/S$ independent of dimension!:

$$var_{P\left(\{x^{(s)}\}\right)}\left[\hat{f}\right] = \frac{1}{S^2}\sum_{s=1}^{S} var_{P\left(x^{(s)}\right)}\left[f\left(x^{(s)}\right)\right] = var_{P(x)}\left[f(x)\right]/S$$

Why sampling?
The integration problem.



Z. Ghahramani

We often need to compute integrals of the form

$$\int f(x)\, p(x)\, dx$$

where $f(x)$ is some function of a random variable X which has probability density $p(x)$.

Three typical difficulties can arise:

left panel:

full line is some complicated function, dashed is density;

right panel:

full line is some function and dashed is complicated density;

not shown: integral (or sum) very high dimensions.

Monte Carlo sampling
Difficulties

Monte Carlo Estimator:

$$\int f(x) P(x) dx \approx \hat{f} \equiv \frac{1}{S} \sum_{s=1}^{S} f\left(x^{(s)}\right), x^{(s)} \sim P(x)$$

Problems with Monte Carlo sampling:

- it may be difficult (impossible) to obtain the samples directly from $P(x)$
- regions of high density $P(x)$ may not correspond to regions where $f(x)$ varies a lot

Monte Carlo sampling
Difficulties

Monte Carlo Estimator:

$$\int f(x) P(x) dx \approx \hat{f} \equiv \frac{1}{S} \sum_{s=1}^{S} f\left(x^{(s)}\right), x^{(s)} \sim P(x)$$

Problems with Monte Carlo sampling:

- it may be difficult (impossible) to obtain the samples directly from $P(x)$
- regions of high density $P(x)$ may not correspond to regions where $f(x)$ varies a lot

Monte Carlo sampling
Difficulties

Monte Carlo Estimator:

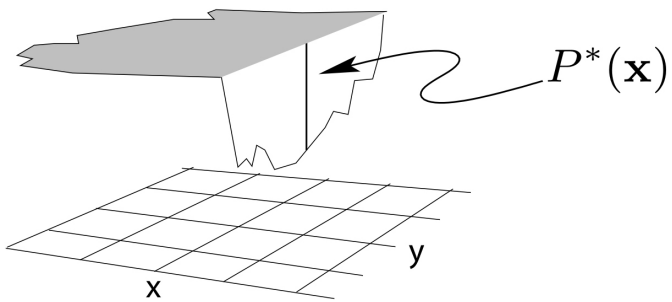
$$\int f(x) P(x) dx \approx \hat{f} \equiv \frac{1}{S} \sum_{s=1}^{S} f\left(x^{(s)}\right), x^{(s)} \sim P(x)$$

Problems with Monte Carlo sampling:

- it may be difficult (impossible) to obtain the samples directly from $P(x)$
- regions of high density $P(x)$ may not correspond to regions where $f(x)$ varies a lot

A real-world problem

Task: draw random water samples from a lake and find the average plankton concentration.



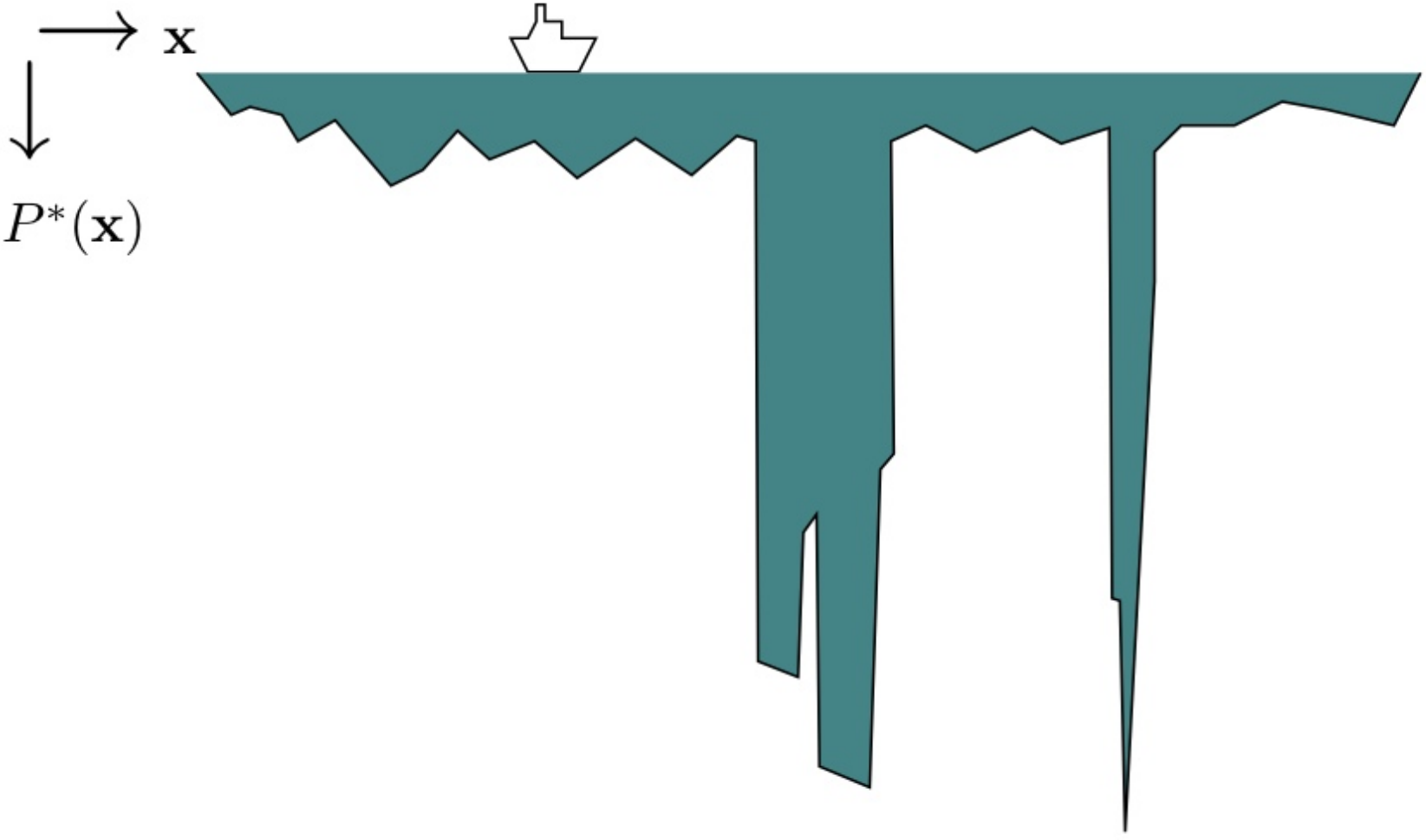
Assume we can measure

- The depth of the lake: $P^{\star}(\mathbf{x}), \mathbf{x} = (x, y)$
- Plankton concentration: $f(\mathbf{x})$

The required average concentration is an integral of the above form, namely

$$< f(\mathbf{x}) > \quad = \quad \frac{1}{Z} \int f(\mathbf{x}) P^{\star}(\mathbf{x}) \, d\mathbf{x} \quad Z = \int P^{\star}(\mathbf{x}) \, dxdy$$

Difficult samples



Sampling discrete distributions

Sampling from a univariate discrete distribution p with K states

Fundamental to understand how to sample probability distributions:
How can we "sample from a distribution"?
Assume for now:
univariate discrete (not conditional) distribution p, with K states, e.g.

$$p(x) = \begin{cases} 0.6 & x = 1 \\ 0.1 & x = 2 \\ 0.3 & x = 3 \end{cases}$$

Throw uniformly a dart to this partition of distribution and determine the corresponding $k \in 1, 2, 3$ as state of the sample

1	×	2	3

Sampling discrete distributions

Sampling from a univariate discrete distribution p with K states

Implementation

1: Label the K states as $i = 1, \ldots, K$, with associated probabilities p_i.
2: Calculate the *cumulant*

$$c_i = \sum_{j \leq i} p_j$$

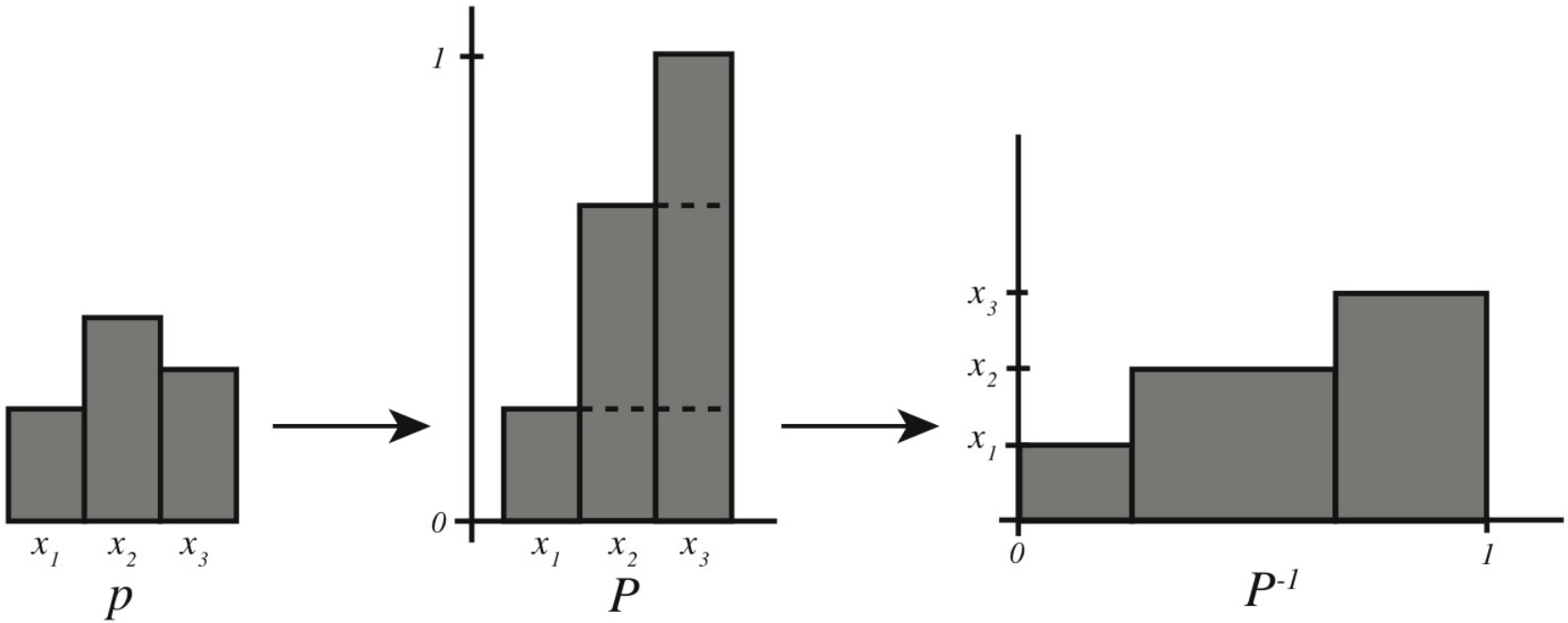
and set $c_0 = 0$.
3: Draw a value u uniformly at random from the unit interval $[0, 1]$.
4: Find that i for which $c_{i-1} \leq u \leq c_i$.
5: Return state i as a sample from p.

First, compute the discrete cumulative histogram and then draw a uniform random variable $u \sim U[0, 1]$, and select a discrete **state** i, c.f. denoting *the sample*.

Sampling discrete distributions

Sampling from a univariate discrete distribution p with K states

Another illustration of sampling from discrete distribution:

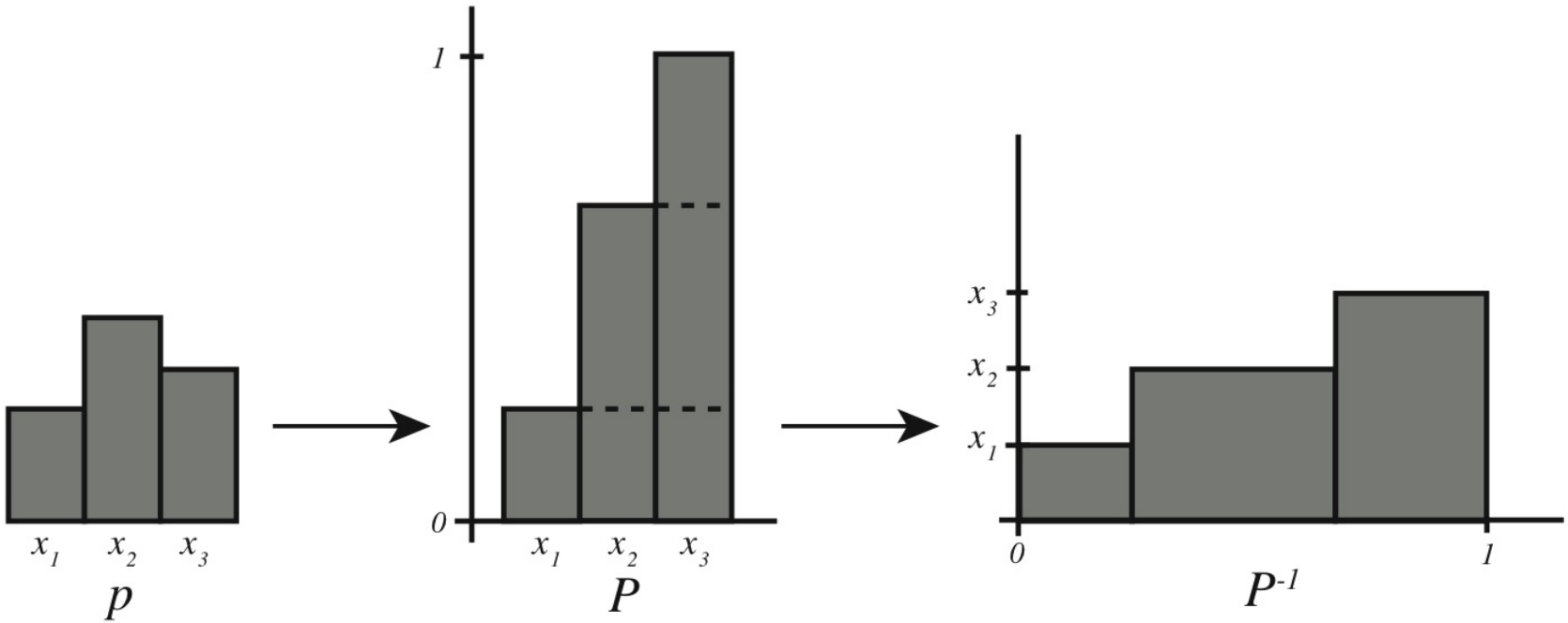


1. Distribution is given by p with discrete states $x = 1, 2, 3$ or $x_1, x_2,$ and x_3
2. $\rightarrow$ derive the cumulative distribution P
3. $\rightarrow$ Map uniform sample $U[0, 1]$ via inverse cumulative function P^{-1} to state 1, 2 or 3

Sampling discrete distributions

Sampling from a univariate discrete distribution p with K states

Another illustration of sampling from discrete distribution:

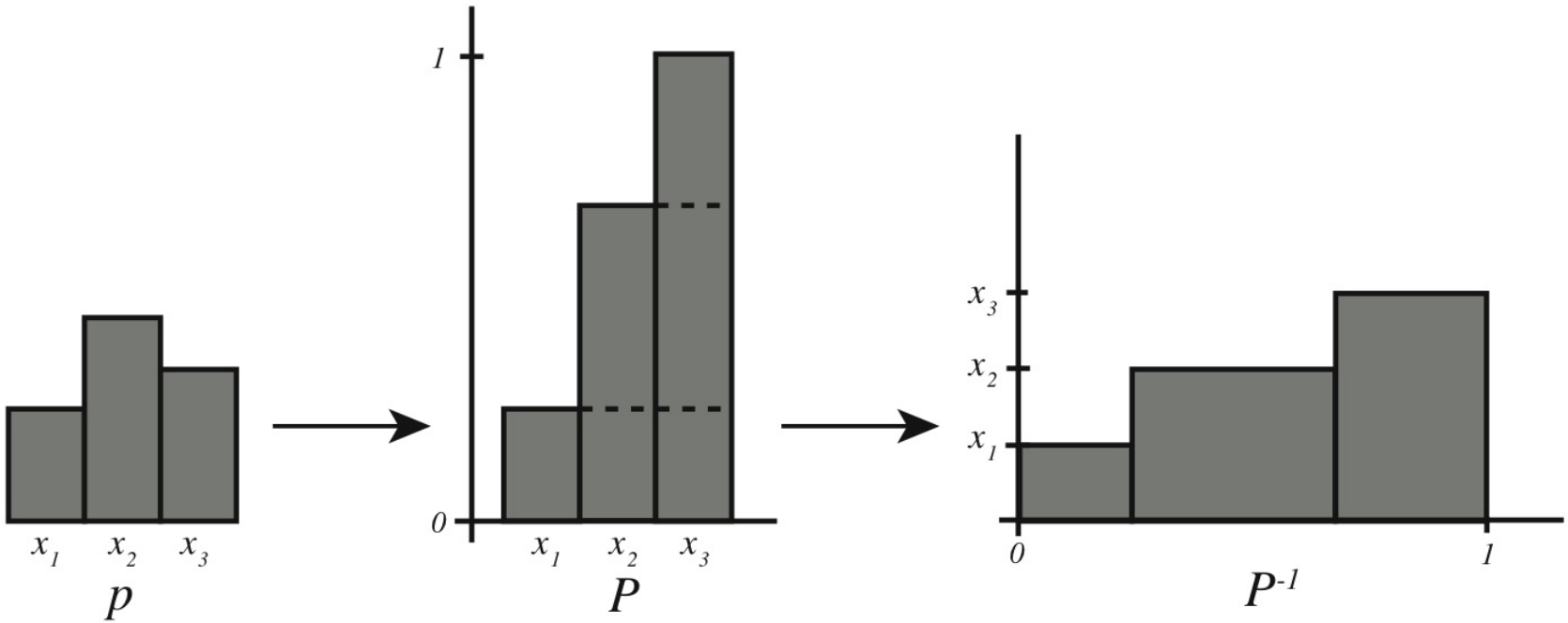


1. Distribution is given by p with discrete states $x = 1, 2, 3$ or $x_1, x_2,$ and x_3
2. $\rightarrow$ derive the cumulative distribution P
3. $\rightarrow$ Map uniform sample $U[0, 1]$ via inverse cumulative function P^{-1} to state 1, 2 or 3

Sampling discrete distributions

Sampling from a univariate discrete distribution p with K states

Another illustration of sampling from discrete distribution:

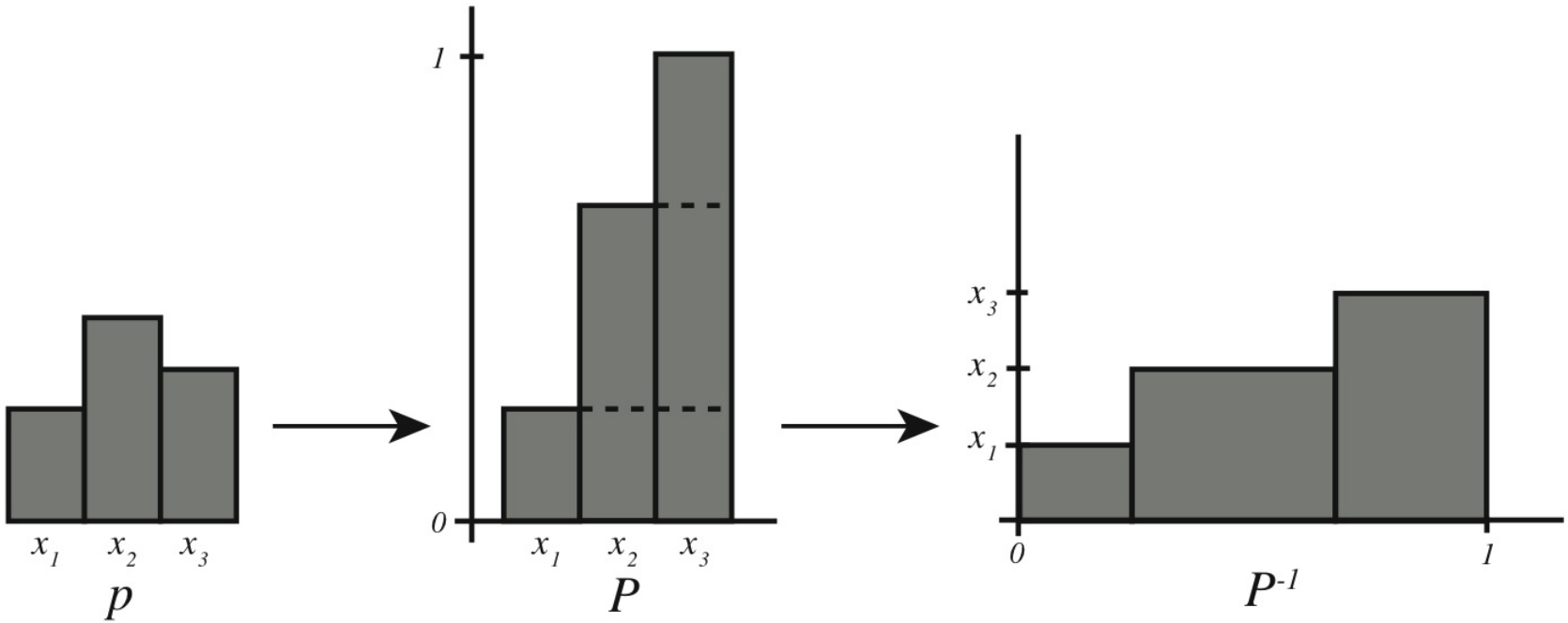


1. Distribution is given by p with discrete states $x = 1, 2, 3$ or $x_1, x_2,$ and x_3
2. $\rightarrow$ derive the cumulative distribution P
3. $\rightarrow$ Map uniform sample $U[0, 1]$ via inverse cumulative function P^{-1} to state $1, 2$ or 3

Sampling discrete distributions

Sampling from a univariate discrete distribution p with K states

Another illustration of sampling from discrete distribution:



1. Distribution is given by p with discrete states $x = 1, 2, 3$ or $x_1, x_2,$ and x_3
2. $\rightarrow$ derive the cumulative distribution P
3. $\rightarrow$ Map uniform sample $U[0, 1]$ via inverse cumulative function P^{-1} to state $1, 2$ or 3

Sampling discrete distributions

Multidimensional distributions

Problem: Sample from $p(x_1, \ldots, x_n)$ We can sample from the joint distribution $p(x_1, \ldots, x_n)$ by

1. sampling a state for x_1 from the one-dimensional $p(x_1)$
2. with x_1 clamped/ fixed to this state, sampling from the one-dimensional $p(x_2|x_1)$ a state for x_2

We can generalize this scheme by cascading this approach by decomposition into conditional probability distributions

$$p(x_1, \ldots, x_n) = p(x_n|x_{n-1}, \ldots, x_n)\, p(x_{n-1}|x_{n-2}, \ldots, x_1) \ldots p(x_2|x_1)\, p(x_1)$$

In **Belief Networks**, however, these conditionals are computationally tractable and specified which leads to ancestral sampling

Sampling discrete distributions

Multidimensional distributions

Problem: Sample from $p(x_1, \ldots, x_n)$ We can sample from the joint distribution $p(x_1, \ldots, x_n)$ by

1. sampling a state for x_1 from the one-dimensional $p(x_1)$
2. with x_1 clamped/ fixed to this state, sampling from the one-dimensional $p(x_2|x_1)$ a state for x_2

We can generalize this scheme by cascading this approach by decomposition into conditional probability distributions

$$p(x_1, \ldots, x_n) = p(x_n|x_{n-1}, \ldots, x_n)\, p(x_{n-1}|x_{n-2}, \ldots, x_1) \ldots p(x_2|x_1)\, p(x_1)$$

In **Belief Networks**, however, these conditionals are computationally tractable and specified which leads to ancestral sampling

Sampling discrete distributions

Multidimensional distributions

Problem: Sample from $p(x_1, \ldots, x_n)$ We can sample from the joint distribution $p(x_1, \ldots, x_n)$ by

1. sampling a state for x_1 from the one-dimensional $p(x_1)$
2. with x_1 clamped/ fixed to this state, sampling from the one-dimensional $p(x_2|x_1)$ a state for x_2

We can generalize this scheme by cascading this approach by decomposition into conditional probability distributions

$$p(x_1, \ldots, x_n) = p(x_n|x_{n-1}, \ldots, x_n)\, p(x_{n-1}|x_{n-2}, \ldots, x_1) \ldots p(x_2|x_1)\, p(x_1)$$

In **Belief Networks**, however, these conditionals are computationally tractable and specified which leads to ancestral sampling

Sampling discrete distributions

Multidimensional distributions

Problem: Sample from $p(x_1, \ldots, x_n)$ We can sample from the joint distribution $p(x_1, \ldots, x_n)$ by

1. sampling a state for x_1 from the one-dimensional $p(x_1)$
2. with x_1 clamped/ fixed to this state, sampling from the one-dimensional $p(x_2|x_1)$ a state for x_2

We can generalize this scheme by cascading this approach by decomposition into conditional probability distributions

$$p(x_1, \ldots, x_n) = p(x_n|x_{n-1}, \ldots, x_n)\, p(x_{n-1}|x_{n-2}, \ldots, x_1) \ldots p(x_2|x_1)\, p(x_1)$$

In **Belief Networks**, however, these conditionals are computationally tractable and specified which leads to ancestral sampling

Sampling discrete distributions

Multidimensional distributions

Problem: Sample from $p(x_1, \ldots, x_n)$ We can sample from the joint distribution $p(x_1, \ldots, x_n)$ by

1. sampling a state for x_1 from the one-dimensional $p(x_1)$
2. with x_1 clamped/ fixed to this state, sampling from the one-dimensional $p(x_2|x_1)$ a state for x_2

We can generalize this scheme by cascading this approach by decomposition into conditional probability distributions

$$p(x_1, \ldots, x_n) = p(x_n|x_{n-1}, \ldots, x_n)\, p(x_{n-1}|x_{n-2}, \ldots, x_1) \ldots p(x_2|x_1)\, p(x_1)$$

In **Belief Networks**, however, these conditionals are computationally tractable and specified which leads to ancestral sampling

Sampling discrete distributions

Multidimensional distributions

Problem: Sample from $p(x_1, \ldots, x_n)$ We can sample from the joint distribution $p(x_1, \ldots, x_n)$ by

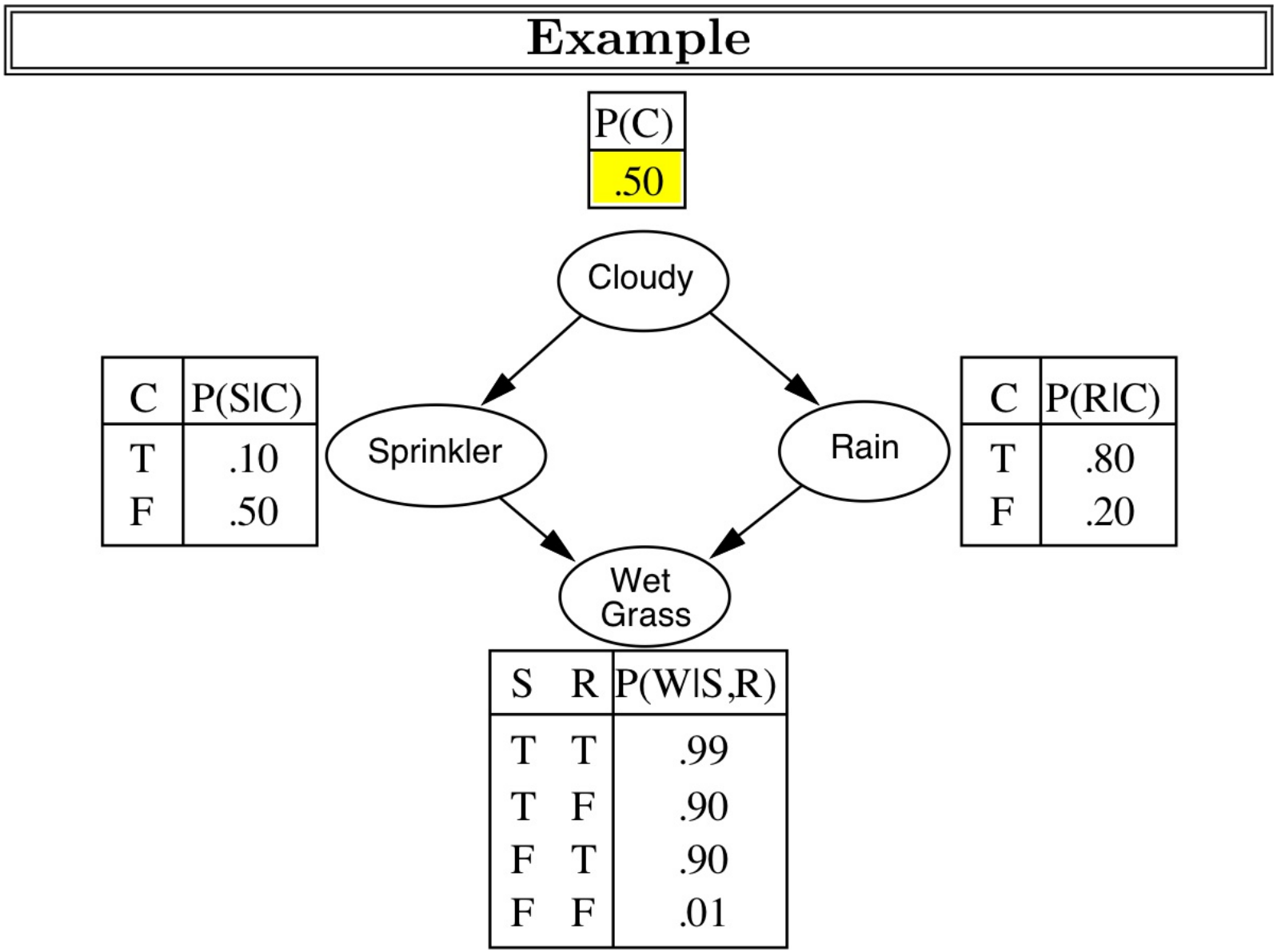
1. sampling a state for x_1 from the one-dimensional $p(x_1)$
2. with x_1 clamped/ fixed to this state, sampling from the one-dimensional $p(x_2|x_1)$ a state for x_2

We can generalize this scheme by cascading this approach by decomposition into conditional probability distributions

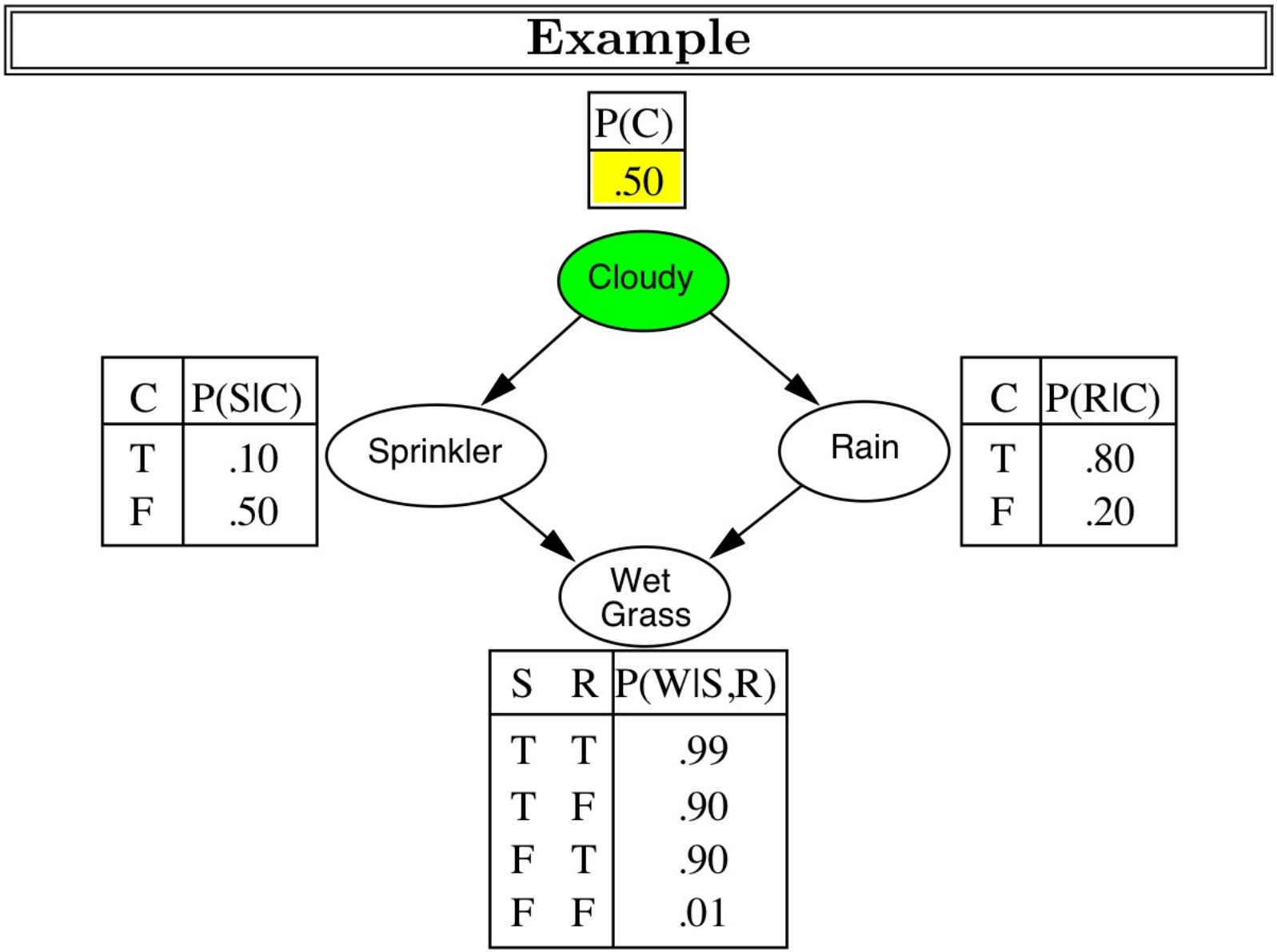
$$p(x_1, \ldots, x_n) = p(x_n|x_{n-1}, \ldots, x_n)\, p(x_{n-1}|x_{n-2}, \ldots, x_1) \ldots p(x_2|x_1)\, p(x_1)$$

In **Belief Networks**, however, these conditionals are computationally tractable and specified which leads to ancestral sampling

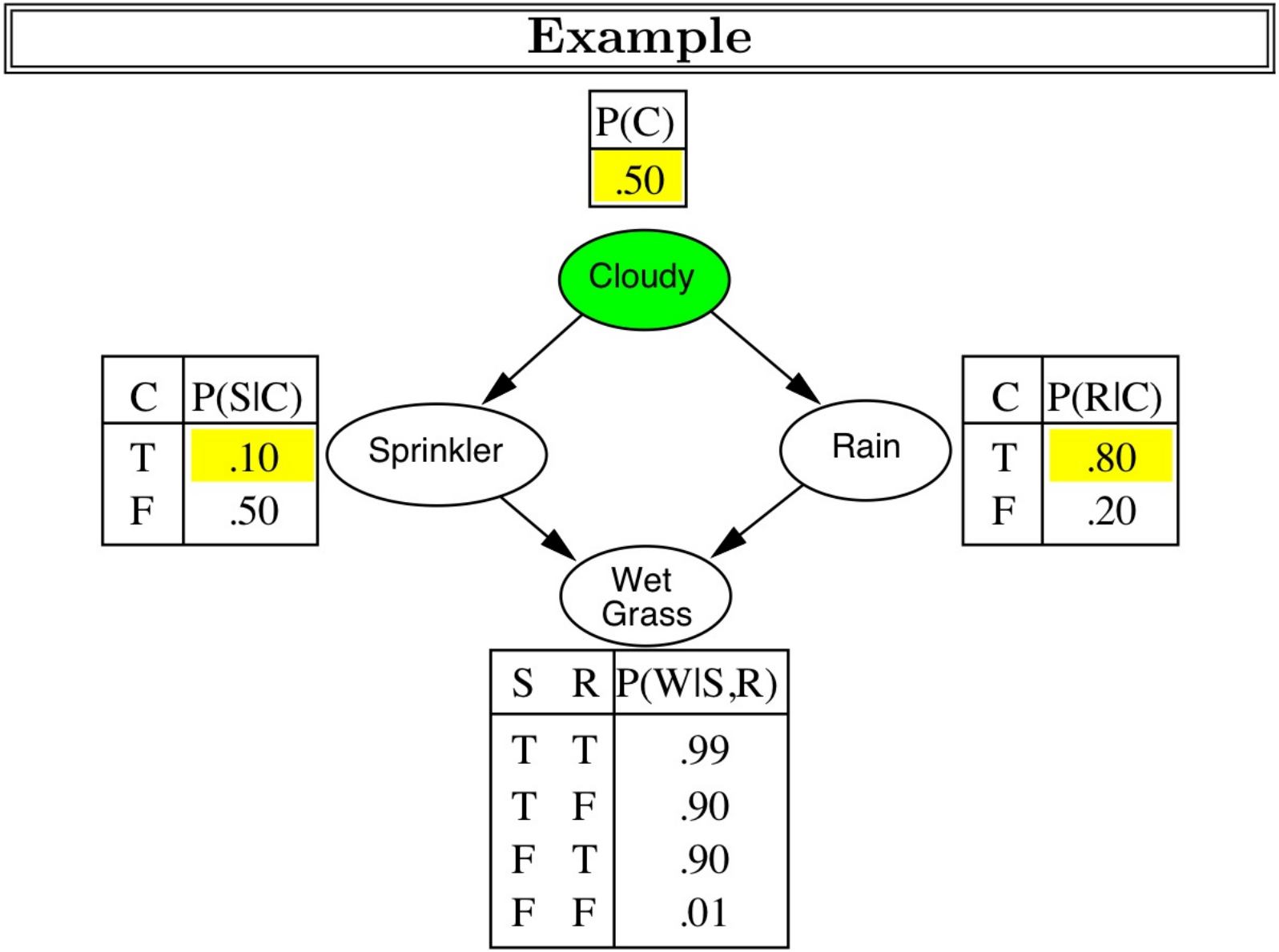
Sampling from an empty Network



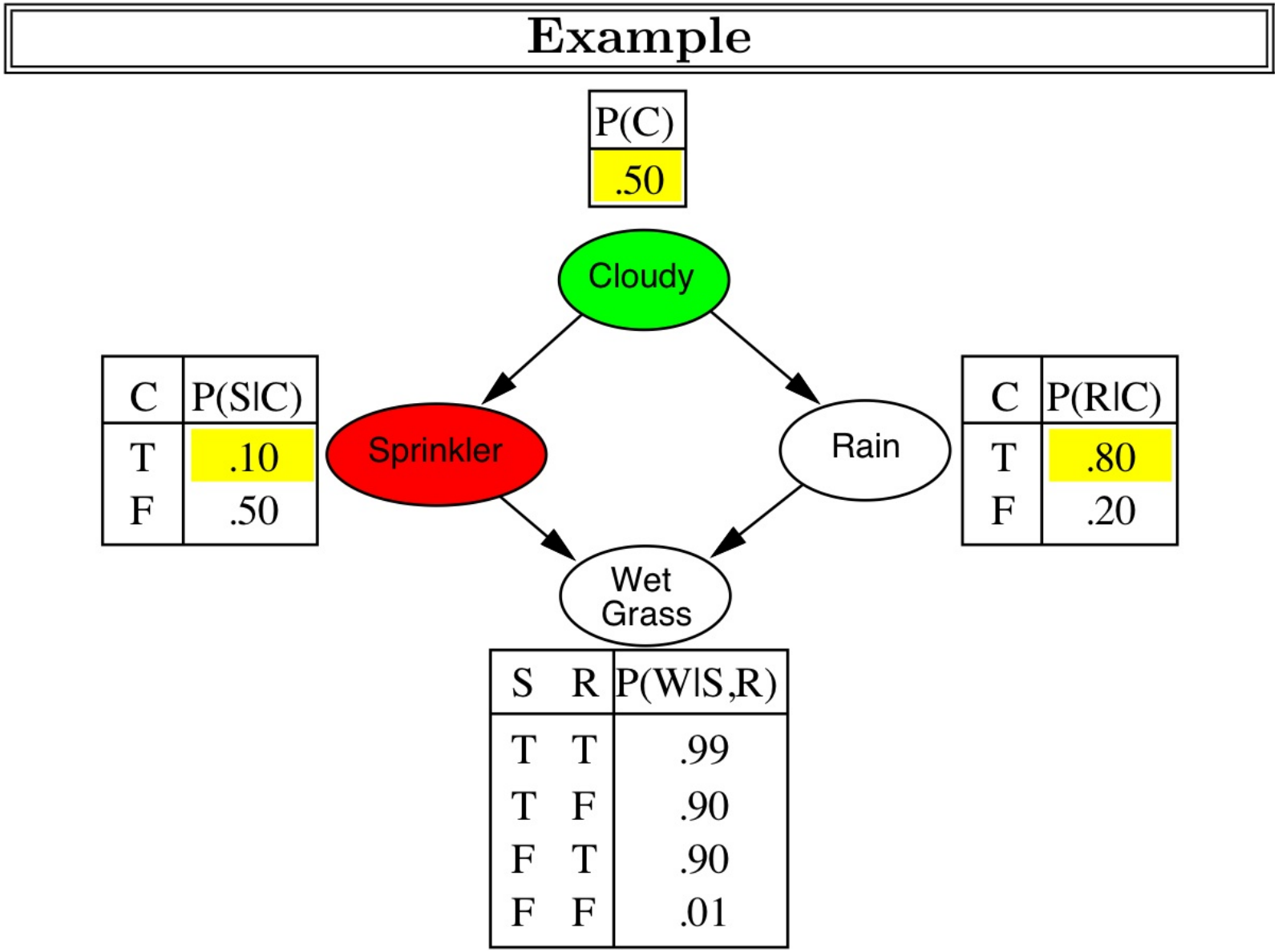
Sampling from an empty Network



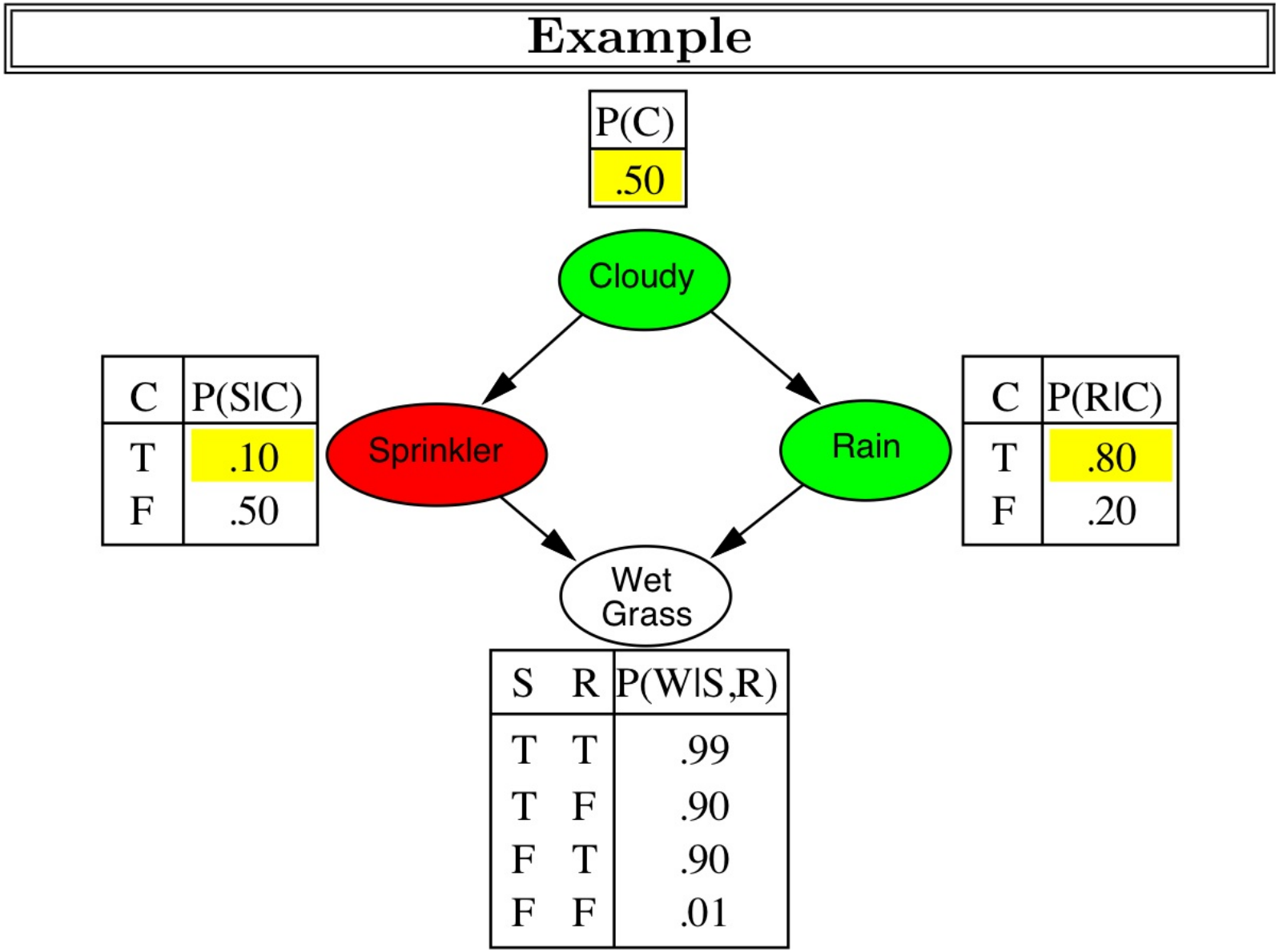
Sampling from an empty Network



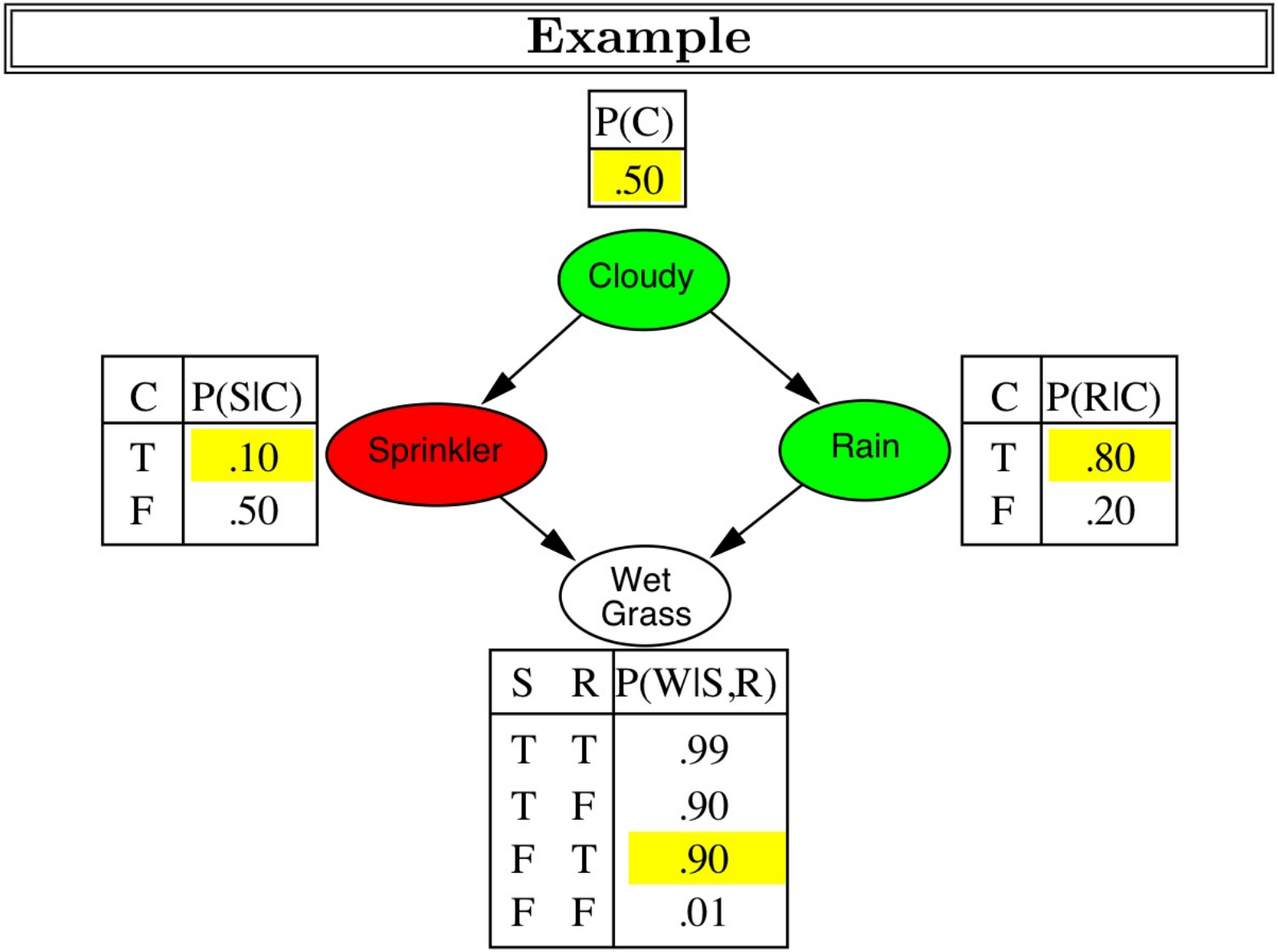
Sampling from an empty Network



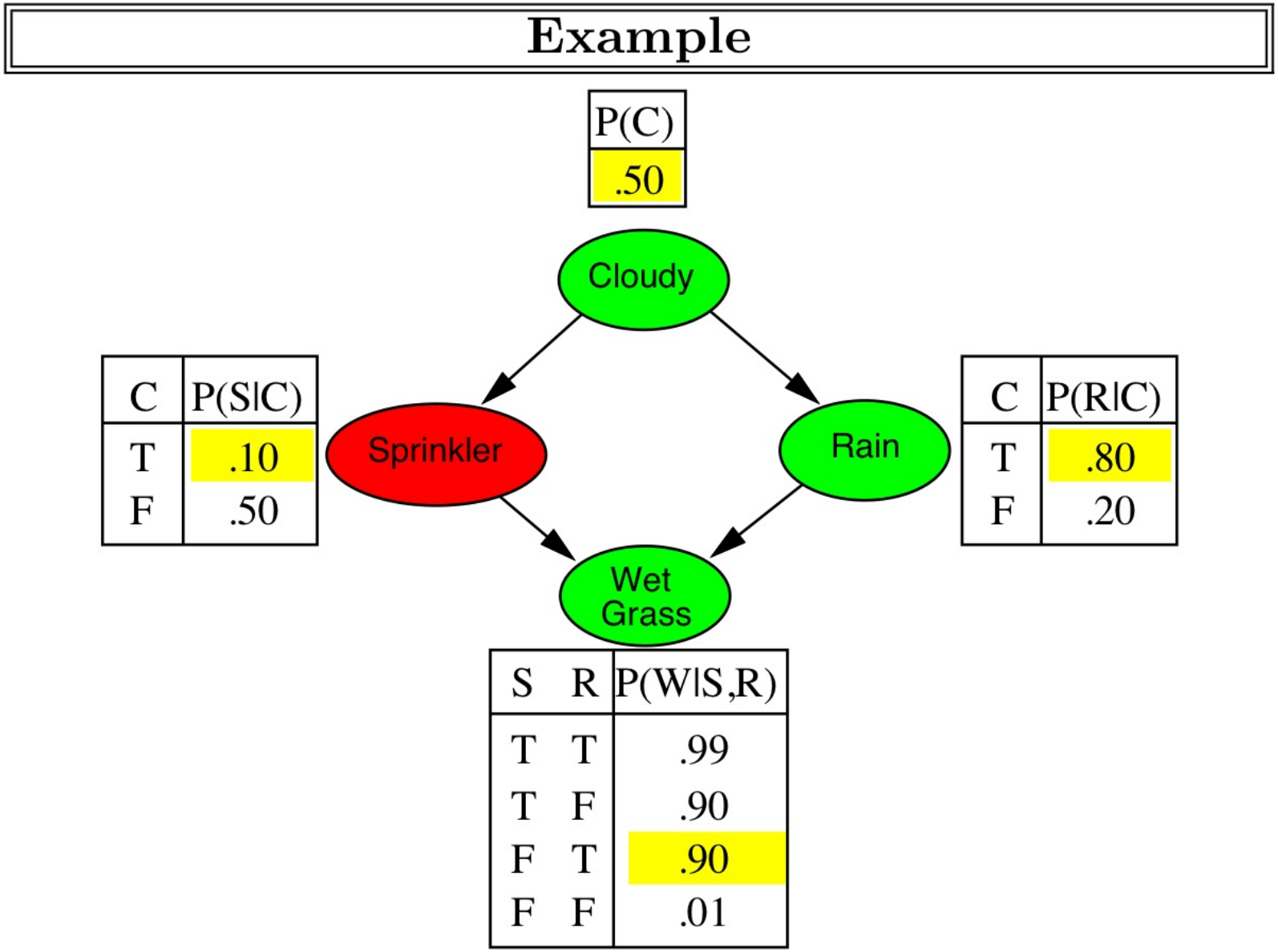
Sampling from an empty Network



Sampling from an empty Network



Sampling from an empty Network



Ancestral sampling (Prior Sampling) from empty network

Probability that this prior sampling (PS) scheme generates a particular event

$$S_{PS}(x_1, \ldots, x_n) = \prod_i^n P(x_i | parents(X_i)) = P(x_1, \ldots, x_n)$$

i.e., the true prior probability.

E.g., $S_{PS}(t, f, t, t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 = P(t, f, t, t)$

Let $N_{PS}(x_1, \ldots, x_n)$ be the number of samples generated for event $x_1, \ldots, x_n$

Then we have

$$\begin{aligned} \lim_{N \to \infty} \hat{P}(x_1, \ldots, x_n) &= \lim_{N \to \infty} N_{PS}(x_1, \ldots, x_n)/N \\ &= S_{PS}(x_1, \ldots, x_n) \\ &= P(x_1, \ldots, x_n) \end{aligned}$$

That is, estimates derived from Ancestral Sampling are **consistent**.

Shorthand: $\hat{P}(x_1, \ldots, x_n) \approx P(x_1, \ldots, x_n)$

Rejection sampling in network with observed nodes

- We have a Bayesian Network with RVs $X_{1:n}$, some of which are observed: $X_{obs} = y_{obs}, obs \subset \{1 : n\}$
- The goal is to compute marginal posteriors $P(X|X_{obs} = y_{obs})$ conditioned on the observations
- Our strategy is to generate a set of K (joint) samples of all variables

$$\mathcal{S} = \left\{(x_1^k, x_2^k, \ldots, x_n^k)\right\}_{k=1}^{K}$$

Note: Each sample $x_{1:n}^k$ is a list of instantiations of all RVs.

Rejection sampling in network with observed nodes

To generate a single sample $x^k_{1:n}$

1. Sort all RVs in topological order; start with $i = 1$
2. Sample a value $x^k_i \sim P\left(X_i | x_{\text{Parents(i)}}\right)$ for the ith RV conditional to the previous samples $x^k_{1:i-1}$
3. If $i \in obs$ compare the sampled value x^k_i with the observation y_i. Reject and repeat from 1) if the sample is not equal to the observation.
4. Repeat with $i \leftarrow i + 1$ from 2.

Rejection sampling in network with observed nodes

- Since computers are fast, we can sample for large K
- The sample set

$$\mathcal{S} = \left\{(x_1^k, x_2^k, \ldots, x_n^k)\right\}_{k=1}^{K}$$

implies the necessary information:

- We can compute the marginal probabilities from these statistics:

$$P\left(X_i = x | X_{obs} = y_{obs}\right) \approx \frac{count_{\mathcal{S}}\left(X_i^k = x\right)}{K}$$

- or pair-wise marginals:

$$P\left(X_i = x, X_i = x' | X_{obs} = y_{obs}\right) \approx \frac{count_{\mathcal{S}}\left(X_i^k = x \wedge X_i^k = x'\right)}{K}$$

- etc.

Rejection sampling in network with observed nodes

- Since computers are fast, we can sample for large K
- The sample set

$$\mathcal{S} = \left\{\left(x_1^k, x_2^k, \ldots, x_n^k\right)\right\}_{k=1}^{K}$$

implies the necessary information:

  - We can compute the marginal probabilities from these statistics:

$$P\left(X_i = x | X_{obs} = y_{obs}\right) \approx \frac{count_{\mathcal{S}}\left(X_i^k = x\right)}{K}$$

  - or pair-wise marginals:

$$P\left(X_i = x, X_i = x' | X_{obs} = y_{obs}\right) \approx \frac{count_{\mathcal{S}}\left(X_i^k = x \wedge X_i^k = x'\right)}{K}$$

  - etc.

Rejection sampling in network with observed nodes

- Since computers are fast, we can sample for large K
- The sample set

$$\mathcal{S} = \left\{\left(x_1^k, x_2^k, \ldots, x_n^k\right)\right\}_{k=1}^{K}$$

implies the necessary information:

  - We can compute the marginal probabilities from these statistics:

$$P\left(X_i = x | X_{obs} = y_{obs}\right) \approx \frac{count_{\mathcal{S}}\left(X_i^k = x\right)}{K}$$

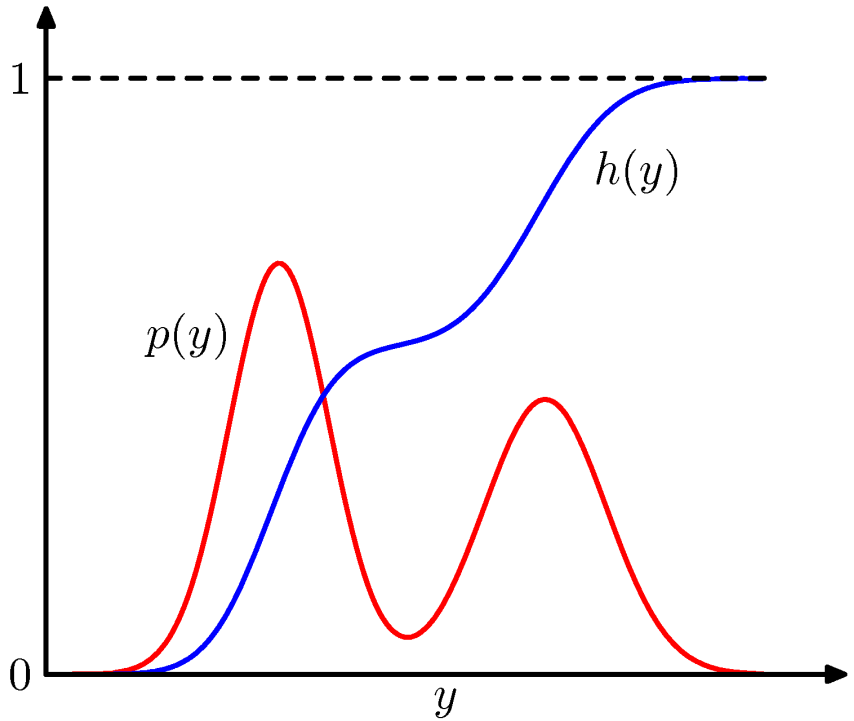
  - or pair-wise marginals:

$$P\left(X_i = x, X_i = x' | X_{obs} = y_{obs}\right) \approx \frac{count_{\mathcal{S}}\left(X_i^k = x \wedge X_i^k = x'\right)}{K}$$

  - etc.

Sampling continuous distributions

Again, as in the discrete case, assume a uniform sample generator given.
Now, how can we convert samples from a Uniform $[0, 1]$ random number generator?



$$F(y) = \int_{-\infty}^{y} p(y')\, dy' \qquad (1)$$

Draw mass to left of point:
$u \approx$ Uniform $[0, 1]$
Sample, $y(u) = F^{-1}(u)$

Sampling continuous distributions

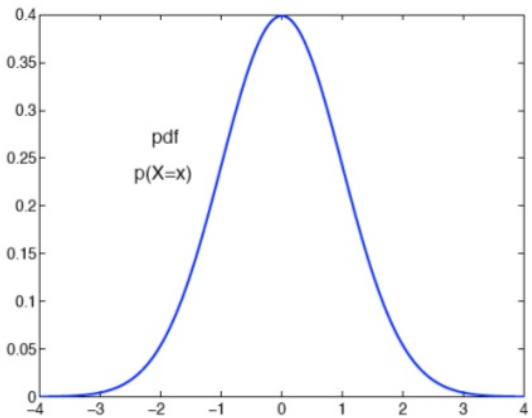
- **Given: 1-dim. Gaussian pdf (probability density function) $p(\mathbf{x}|\mu,\sigma^2)$ and the corresponding cumulative distribution:**

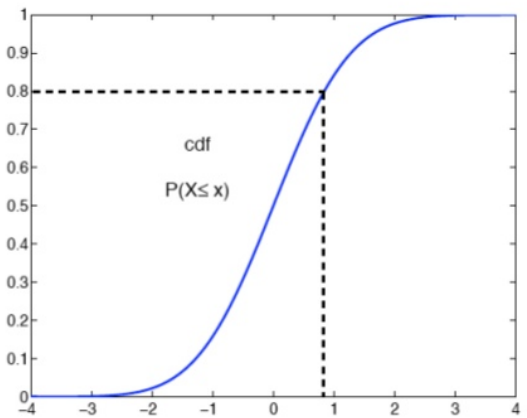
$$F_{\mu,\sigma^2}(x) = \int_{-\infty}^{x} p(x|\mu,\sigma^2)dx$$

- **To draw samples from a Gaussian, we can invert the cumulative distribution function:**

$$u \sim Uniform(0,1) \Rightarrow F^{-1}_{\mu,\sigma^2}(u) \sim p(x|\mu,\sigma^2)$$

$p(x|\mu,\sigma^2)$





$F_{\mu,\sigma^2}(x)$

Sampling continuous distributions

Parametric density model example

A simple multivariate (D-dimensional) Gaussian model

$$p(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}$$

- This is a "generative" model in the sense that we can generate samples $\mathbf{x}$ according to the distribution.
- With a $\mathbf{C}$ fulfulling $\mathbf{\Sigma} = \mathbf{C}\mathbf{C}^T$ and $T : \mathbf{y} = \mathbf{C}^{-1}(\mathbf{x} - \mu)$ we can transform $p(\mathbf{x}|\mu, \mathbf{\Sigma}) \rightarrow p(\mathbf{y}|\mathbf{0}, \mathbf{I})$

We can now draw indepedent Gaussian samples y_i of $\mathbf{y}$, since $p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{I}) = \prod_i \mathcal{N}(y_i|0, 1) = \prod_i p(y_i)$ and transform them back $T^{-1} : \mathbf{x} = \mathbf{C}\mathbf{y} + \mu$ to generate data

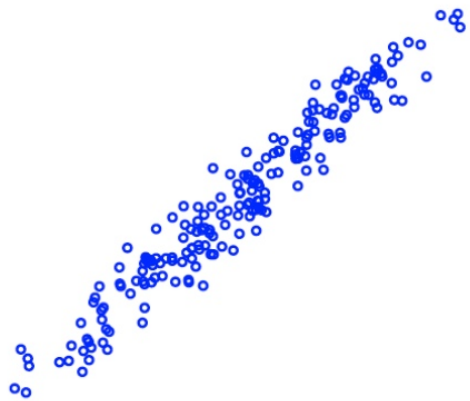
Sampling continuous distributions

Parametric density model example

A simple multivariate (D-dimensional) Gaussian model

$$p(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}$$

- This is a "generative" model in the sense that we can generate samples $\mathbf{x}$ according to the distribution.



- With a $\mathbf{C}$ fulfulling $\mathbf{\Sigma} = \mathbf{C}\mathbf{C}^T$ and $T : \mathbf{y} = \mathbf{C}^{-1}(\mathbf{x} - \mu)$ we can transform $p(\mathbf{x}|\mu, \mathbf{\Sigma}) \rightarrow p(\mathbf{y}|\mathbf{0}, \mathbf{I})$

We can now draw indepedent Gaussian samples y_i of $\mathbf{y}$, since $p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{I}) = \prod_i \mathcal{N}(y_i|0, 1) = \prod_i p(y_i)$ and transform them back $T^{-1} : \mathbf{x} = \mathbf{C}\mathbf{y} + \mu$ to generate data

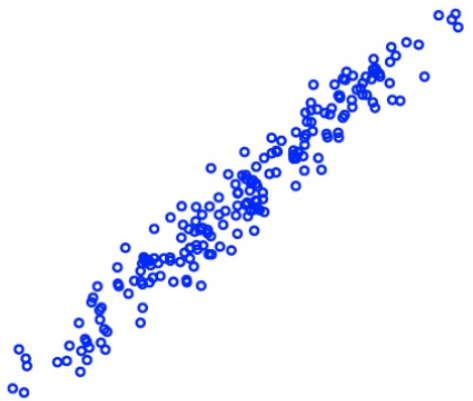
Sampling continuous distributions

Parametric density model example

A simple multivariate (D-dimensional) Gaussian model

$$p(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}$$

- This is a "generative" model in the sense that we can generate samples $\mathbf{x}$ according to the distribution.
- With a $\mathbf{C}$ fulfulling $\mathbf{\Sigma} = \mathbf{C}\mathbf{C}^T$ and $T : \mathbf{y} = \mathbf{C}^{-1}(\mathbf{x} - \mu)$ we can transform $p(\mathbf{x}|\mu, \mathbf{\Sigma}) \rightarrow p(\mathbf{y}|\mathbf{0}, \mathbf{I})$



We can now draw indepedent Gaussian samples y_i of $\mathbf{y}$, since $p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{I}) = \prod_i \mathcal{N}(y_i|0, 1) = \prod_i p(y_i)$ and transform them back $T^{-1} : \mathbf{x} = \mathbf{C}\mathbf{y} + \mu$ to generate data

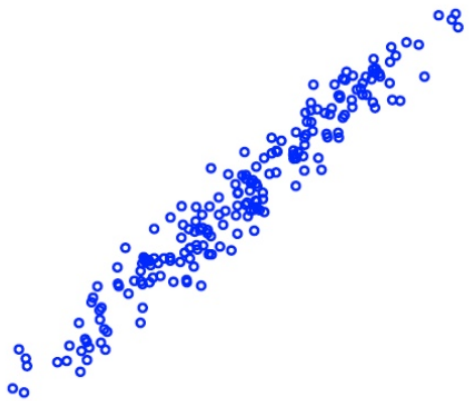
Sampling continuous distributions

Parametric density model example

A simple multivariate (D-dimensional) Gaussian model

$$p(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}$$

- This is a "generative" model in the sense that we can generate samples $\mathbf{x}$ according to the distribution.
- With a $\mathbf{C}$ fulfulling $\mathbf{\Sigma} = \mathbf{C}\mathbf{C}^T$ and $T : \mathbf{y} = \mathbf{C}^{-1}(\mathbf{x} - \mu)$ we can transform $p(\mathbf{x}|\mu, \mathbf{\Sigma}) \rightarrow p(\mathbf{y}|\mathbf{0}, \mathbf{I})$



We can now draw indepedent Gaussian samples y_i of $\mathbf{y}$, since $p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{I}) = \prod_i \mathcal{N}(y_i|0, 1) = \prod_i p(y_i)$ and transform them back $T^{-1} : \mathbf{x} = \mathbf{C}\mathbf{y} + \mu$ to generate data

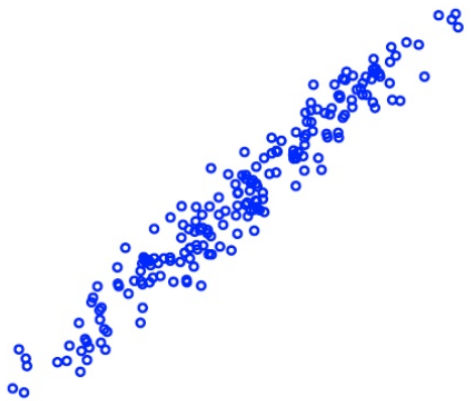
Sampling continuous distributions

Parametric density model example

A simple multivariate (D-dimensional) Gaussian model

$$p(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}$$

- This is a "generative" model in the sense that we can generate samples $\mathbf{x}$ according to the distribution.
- With a $\mathbf{C}$ fulfulling $\mathbf{\Sigma} = \mathbf{C}\mathbf{C}^T$ and $T : \mathbf{y} = \mathbf{C}^{-1}(\mathbf{x} - \mu)$ we can transform $p(\mathbf{x}|\mu, \mathbf{\Sigma}) \rightarrow p(\mathbf{y}|\mathbf{0}, \mathbf{I})$



We can now draw indepedent Gaussian samples y_i of $\mathbf{y}$, since $p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{I}) = \prod_i \mathcal{N}(y_i|0, 1) = \prod_i p(y_i)$ and transform them back $T^{-1} : \mathbf{x} = \mathbf{C}\mathbf{y} + \mu$ to generate data

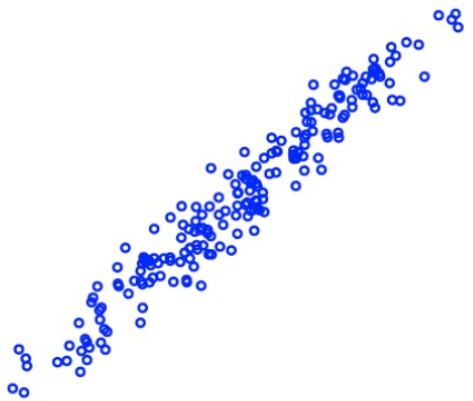
Sampling continuous distributions

Parametric density model example

A simple multivariate (D-dimensional) Gaussian model

$$p\left(\mathbf{x}|\mu,\mathbf{\Sigma}\right)=\frac{1}{\left(2\pi\right)^{D/2}|\mathbf{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^{T}\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}$$

- This is a "generative" model in the sense that we can generate samples $\mathbf{x}$ according to the distribution.
- With a $\mathbf{C}$ fulfulling $\mathbf{\Sigma}=\mathbf{C}\mathbf{C}^{T}$ and $T:\mathbf{y}=\mathbf{C}^{-1}\left(\mathbf{x}-\mu\right)$ we can transform $p\left(\mathbf{x}|\mu,\mathbf{\Sigma}\right)\rightarrow p\left(\mathbf{y}|\mathbf{0},\mathbf{I}\right)$



We can now draw indepedent Gaussian samples y_i of $\mathbf{y}$, since $p\left(\mathbf{y}\right)=\mathcal{N}\left(\mathbf{y}|\mathbf{0},\mathbf{I}\right)=\prod_i\mathcal{N}\left(y_i|0,1\right)=\prod_i p(y_i)$ and transform them back $T^{-1}:\mathbf{x}=\mathbf{C}\mathbf{y}+\mu$ to generate data

Discussion

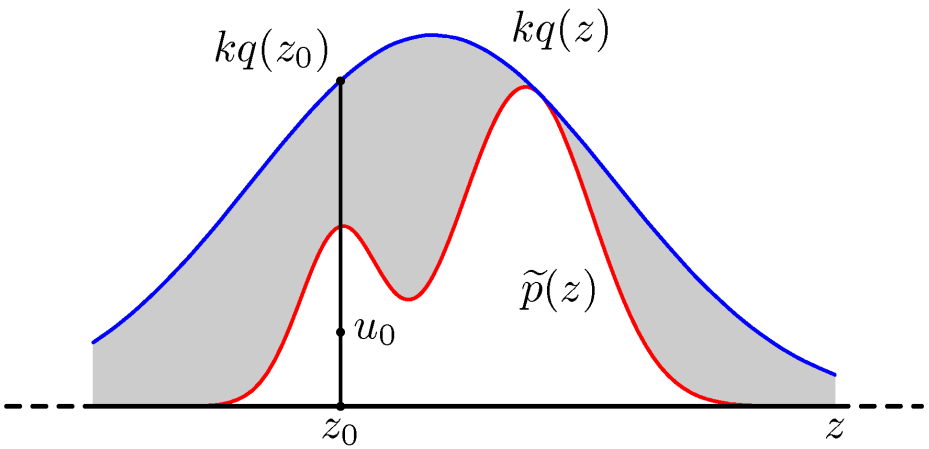
- Transformation method
  - Limited applicability, as we need to invert the indefinite integral of the required distribution $p(x)$
- More general
  - Rejection Sampling
  - Importance Sampling

Rejection Sampling

- Assumptions
  - Sampling directly from $p(\mathbf{z})$ is difficult
  - But we can easily evaluate $p(\mathbf{z})$ (up to some normalization factor Z_p):

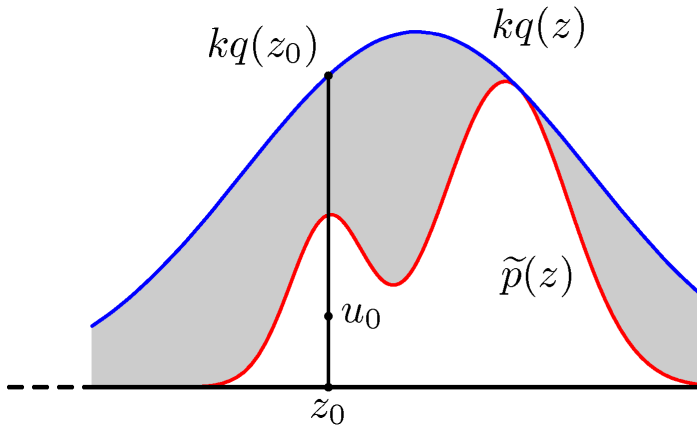
$$p(\mathbf{z}) = \frac{1}{Z_P}\tilde{p}(\mathbf{z})$$

- Idea
  - We need some simpler distribution $q(\mathbf{z})$ (called proposal distribution) from which we can draw samples.
  - Choose a constant k such that: $\forall z : kq(z) \geqslant \tilde{p}(z)$



Rejection Sampling

Sampling procedure



- Generate a number z_0 from $q(z)$
- Generate a number u_0 from the uniform distribution over $[0, kq(z_0)]$.
- If $u_0 > \tilde{p}(z_0)$ reject sample, otherwise accept.
  - Sample is rejected if it lies in the grey shaded area.
  - The remaining pairs (u_0, z_0) have uniform distribution under the curve $\tilde{p}(z)$.

Discussion

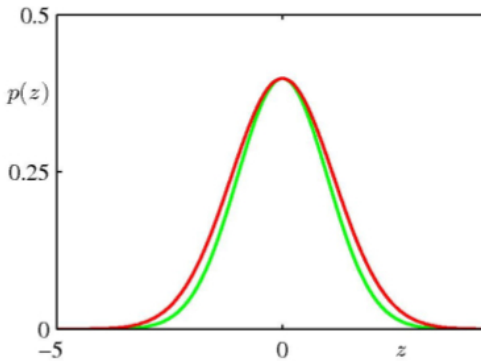
- Original values of z are generated from the distribution $q(z)$.
- Samples are accepted with probability $\tilde{p}(z)/kq(z)$

$$p(accept) = \int (\tilde{p}(z)/kq(z))\, q(z) dz = \frac{1}{k} \int \tilde{p}(z_0)\, dz$$

$\rightarrow$ k should be as small as possible!!!

Rejection Sampling - Discussion

- **Limitation: high-dimensional spaces**
  - For rejection sampling to be of practical value, we require that $kq(z)$ be close to the required distribution, so that the rate of rejection is minimal.
- **Artificial example**
  - Assume that $p(\mathbf{z})$ is Gaussian with covariance matrix $\sigma_p^2 I$
  - Assume that $q(\mathbf{z})$ is Gaussian with covariance matrix $\sigma_q^2 I$
  - Obviously: $\sigma_q^2 \geq \sigma_p^2$
  - In D dimensions: $k = (\sigma_q/\sigma_p)^D$.
    - Assume σ_q is just 1% larger than σ_p.
    - $D = 1000 \Rightarrow k = 1.01^{1000} \geq 20{,}000$
    - And $p(accept) \leq \frac{1}{20000}$



$\Rightarrow$ Often impractical to find good proposal distributions for high dimensions!

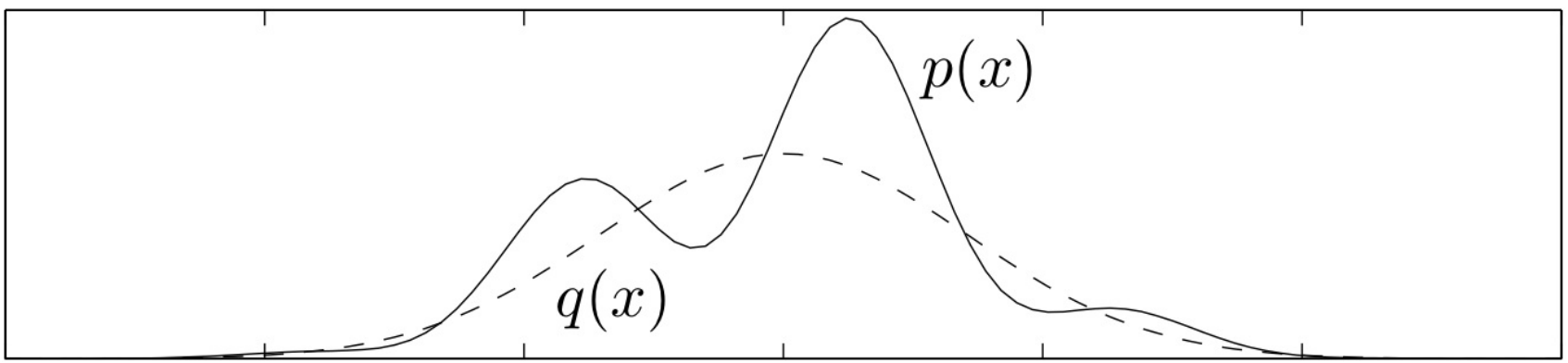
Idea importance sampling

Weighting scheme

Idea: Sample from a different distribution $q(x)$ and weight those samples by $p(x)/q(x)$
Sample $x^{(t)}$ from $q(x)$:

$$\int f(x)\, p(x)\, dx = \int f(x)\, \frac{p(x)}{q(x)} q(x) dx \simeq \frac{1}{T} \sum_{t=1}^{T} f\left(x^{(t)}\right) \frac{p\left(x^{(t)}\right)}{q\left(x^{(t)}\right)}$$

where $q(x)$ is non-zero wherever $p(x)$ is; $w^{(t)} \equiv p\left(x^{(t)}\right)/q\left(x^{(t)}\right)$.



Importance sampling (with likelihood weighting)

- Rejection whole sample (see Ancrestral/Rejection Sampling) may become very inefficient in large Bayes Nets!
- New strategy: We generate a **weighted** sample set

$$\mathcal{S} = \left\{(\omega^k, x_1^k, x_2^k, \ldots, x_n^k)\right\}_{k=1}^{K}$$

where each sample $x_{1:n}^k$ is associated with a weight ω^k.

- In our case, we will choose the weights proportional to the likelihood $P\left(X_{obs} = y_{obs} | X_{1:n} = x_{1:n}^k\right)$ of the observations conditional to the sample $x_{1:n}^k$

Importance sampling

To generate a single sample $(\omega^k, x^k_{1:n})$:

1. Sort all RVs in topological order; start with $i = 1$ and $\omega^k = 1$
2. a) If $i \notin obs$, sample a value $x^k_i \sim P\left(X_i | x^k_{\text{Parents}(i)}\right)$ for the ith RV conditional to the previous samples $x^k_{1:i-1}$
   b) If $i \in obs$, set the value $x^k_i = y_i$ and update the weight according to likelihood
   $$\omega^k \leftarrow \omega^k P\left(X_i = y_i | x^k_{1:i-1}\right)$$
3. Repeat with $i \leftarrow i + 1$ from 2.

Importance sampling (with likelihood weighting)

From a weighted sample set

$$\mathcal{S} = \left\{\left(\omega^k, x_1^k, x_2^k, \ldots, x_n^k\right)\right\}_{k=1}^{K}$$

- we can compute the marginal probabilities:

$$P\left(X_i = x | X_{obs} = y_{obs}\right) \approx \frac{\sum_{k=1}^{K} \omega^k [[x_i^k = x]]}{\sum_{k=1}^{K} \omega^k}$$

- and likewise pair-wise marginals, etc.

Notation: $[[expr]] = 1$ if *expr* is true and zero otherwise

Importance Sampling - Discussion

- Observations
  - Success of importance sampling depends crucially on how well the sampling distribution $q(x)$ matches the desired distribution $p(x)$.
  - Often, $p(x)f(x)$ is strongly varying and has a significant proportion of its mass concentrated over small regions of x-space.

  $\rightarrow$ Weights w may be dominated by a few weights having large values.
  - Practical issue: if none of the samples falls in the regions where $p(x)f(x)$ is large
    - The results may be arbitrary in error.
    - And there will be no diagnostic indication
- Key requirement for sampling distribution $q(x)$:
  - Should not be small or zero in regions where $p(x)$ is significant!

Summary so far

- Sums and integrals, often expectations, occur frequently in statistics
- Monte Carlo approximates expectations with a sample average
- Rejection sampling draws samples from complex distributions
- Importance sampling applies Monte Carlo to any sum/integral

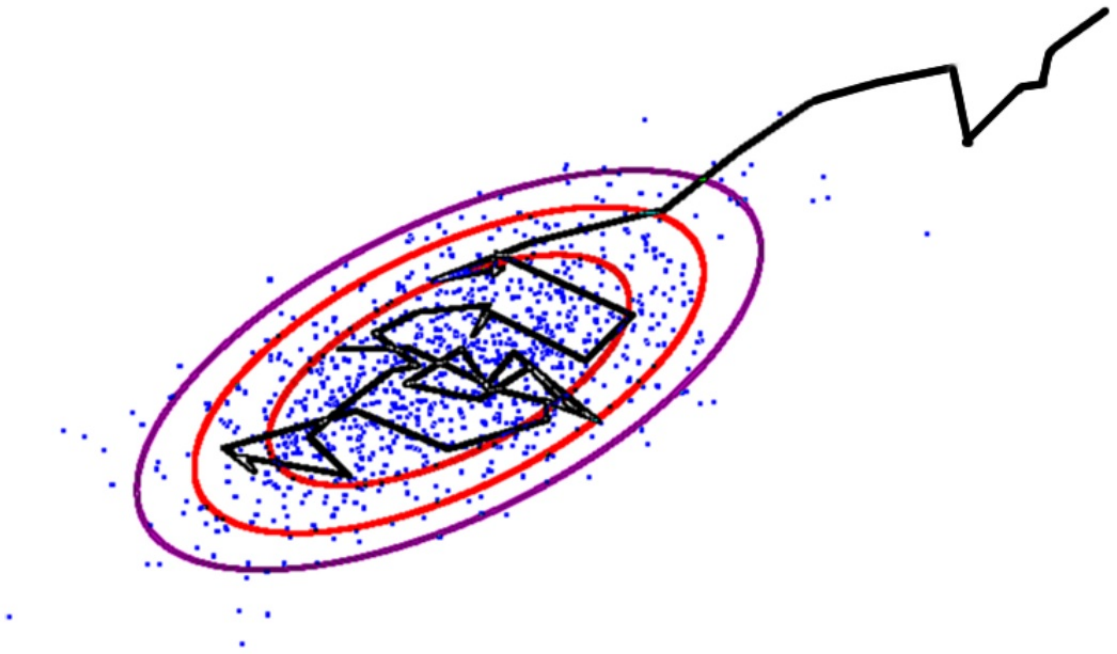
Independent Sampling vs. Markov Chains

- So far
  - We've considered two methods, Rejection Sampling and Importance Sampling, which were both based on independent samples from $q(x)$.
  - However, for many problems of practical interest, it is difficult or impososible to find $q(x)$ with the necessary properties.
- Different approach
  - We abandon the idea of independent sampling.
  - Instead, rely on Markov Chain to generate dependent samples from the target distribution.
  - Independence would be a nice thing, but it is not necessary for the Monte Carlo estimate to be valid.

Markov Chain Monte Carlo

Construct a biased random walk that explores target dist $P^{\star}(x)$

Markov steps, $x_t \sim T(x_{t-1} \to x_t)$



MCMC gives approximate, correlated samples of $P^{\star}$.

MCMC - Markov Chain Monte Carlo

- Overview
  - Allows to sample from a large class of distributions
  - Scales well with the dimensionality of the sample space.
- Idea
  - We maintain a record of the current state x_t
  - The proposal distribution depends on the current state : $q(x|x_t)$
  - The sequence of samples forms a Markov chain $x_1, x_2, \ldots$
- Setting
  - We can evaluate $p(z)$ (up to some normalization factor Z_P): $p(z) = \frac{\tilde{p}(z)}{Z_P}$
  - At each time step, we generate a candidate sample from the proposal distribution and accept the sample according to a criterion.

MCMC - Markov Chain Monte Carlo

- Overview
  - Allows to sample from a large class of distributions
  - Scales well with the dimensionality of the sample space.
- Idea
  - We maintain a record of the current state x_t
  - The proposal distribution depends on the current state : $q(x|x_t)$
  - The sequence of samples forms a Markov chain $x_1, x_2, \ldots$
- Setting
  - We can evaluate $p(z)$ (up to some normalization factor Z_P): $p(z) = \frac{\tilde{p}(z)}{Z_P}$
  - At each time step, we generate a candidate sample from the proposal distribution and accept the sample according to a criterion.

MCMC - Markov Chain Monte Carlo

- Overview
  - Allows to sample from a large class of distributions
  - Scales well with the dimensionality of the sample space.
- Idea
  - We maintain a record of the current state x_t
  - The proposal distribution depends on the current state : $q(x|x_t)$
  - The sequence of samples forms a Markov chain $x_1, x_2, \ldots$
- Setting
  - We can evaluate $p(z)$ (up to some normalization factor Z_P): $p(z) = \frac{\tilde{p}(z)}{Z_P}$
  - At each time step, we generate a candidate sample from the proposal distribution and accept the sample according to a criterion.

Markov Chain Monte Carlo

What are the ingredients of a sampler

Design issues:

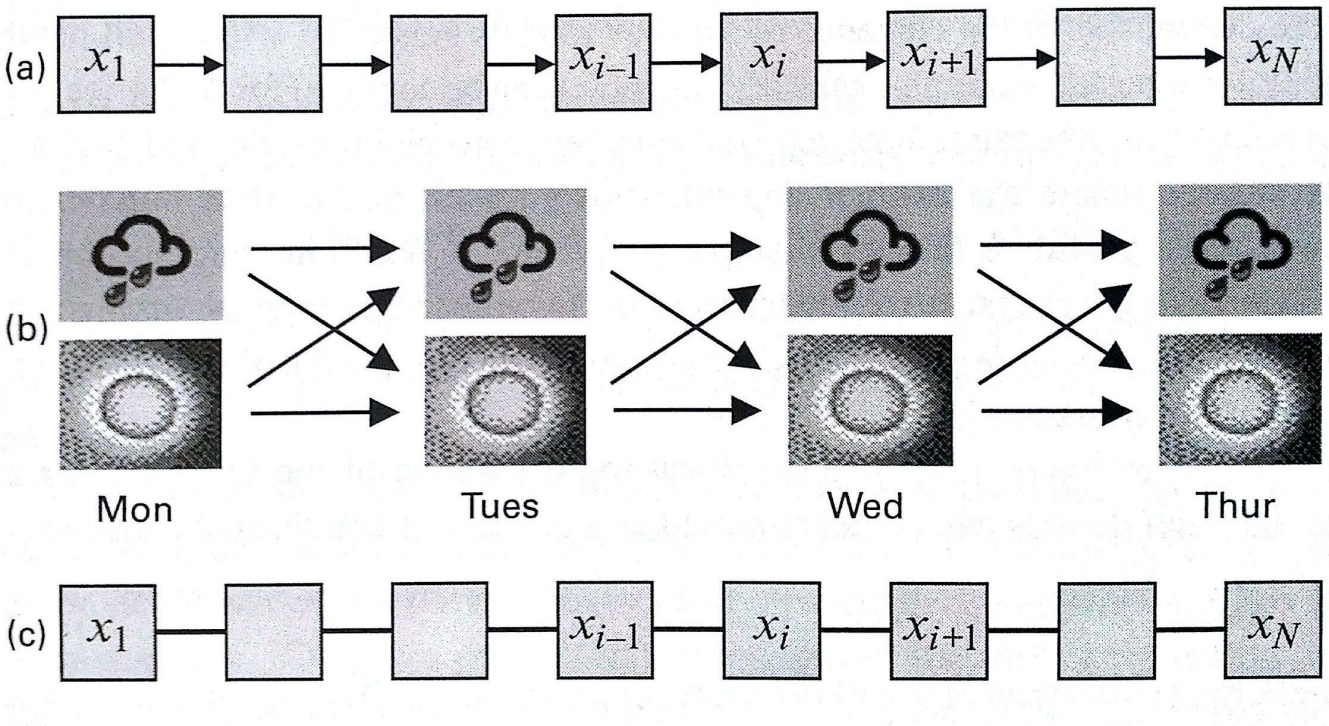
- to sample from a proposal density, conditioned on the previous sample, but which *does not need to match the target density closely*.
- and subsequently reject a certain amount of the samples according to the following rules.

Ingredient rules for MCMC samplers

- we need a proposal density $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ from which we sample
- rule for rejecting or accepting samples from that density $A(\mathbf{x}_n, \mathbf{x}_{n-1})$

Together, these rules must ensure eventually, the samples will be drawn from the taget distribution $P^\star(\mathbf{x})$

Markov Chain



		Yesterday (X_{i-1})	
		Rain	Sun
Today (X_i)	Rain	0.4	0.8
	Sun	0.6	0.2

Markov Chain
Basics

Simple directed graph

We define a Markov Chain:

$$x_1 \to x_2 \to x_3 \to x_4 \to x_5 \to \ldots \to x_N \to$$

where $x_1 \sim p_1(x)$, $x_2 \sim p_2(x)$, etc., with the property that:

$$p_t(x') = \sum_x p_{t-1}(x)\, T(x \to x')$$

where $T(x \to x') = p(X_t = x' | X_{t-1} = x)$ is the Markov chain transition probability from x to x'.

We say that $p^\star(x)$ is an invariant (or stationary) distribution of the Markov chain defined by T if:

$$p^\star(x') = \sum_x p^\star(x)\, T(x \to x')\, \forall x'$$

Markov Chain Monte Carlo (MCMC) methods

Detailed balance

We have a Markov chain $x_1 \to x_2 \to x_3 \to\to \ldots x_N \to$ where $x_1 \sim p_1(x)$, $x_2 \sim p_2(x)$, etc., with the property

$$p_t(x') = \sum_x p_{t-1}(x)\, T(x \to x')$$

where $T(x \to x')$ is the Markov chain transition probability from x to x'.

A useful condition that implies invariance of $p^\star(x')$ is detailed balance:

$$p^\star(x')\, T(x' \to x) = p^\star(x)\, T(x \to x')$$

Stationary distribution of Markov Chain

Discrete example

$$P^\star = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} \qquad T = \begin{pmatrix} 2/3 & 1/2 & 1/2 \\ 1/6 & 0 & 1/2 \\ 1/6 & 1/2 & 0 \end{pmatrix} \qquad T_{ij} = T(x_i \leftarrow x_j)$$

$P^\star$ is an **invariant distribution** of T because $TP^\star = P^\star$, i.e.

$$\sum_x T(x' \leftarrow x) P^\star(x) = P^\star(x')$$

Also $P^\star$ is *the* **equilibrium distribution** of T:

To machine precision: $T^{100} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} = P^\star$

Ergodicity requires: $T^K(x' \leftarrow x) > 0$ for all $x' : P^\star(x') > 0$, for some K

Outline

Markov Chain Monte Carlo Methods:

- Metropolis(-Hastings) Algorithm
- Gibbs Sampling
- not covered today Slice Sampling, Hybrid Monte Carlo

The Metropolis-Hastings algorithm

Transition operator

- Propose a move from the current state $Q(x';x)$, e.g. $\mathcal{N}(x, \sigma^2)$
- Accept with probability $\min\left(1,\ \frac{P(x')Q(x;x')}{P(x)Q(x';x)}\right)$
- Otherwise next state in chain is a copy of current state

Notes

- Can use $\tilde{P} \propto P(x)$; normalizer cancels in acceptance ratio
- Satisfies detailed balance (shown below)
- Q must be chosen to fulfill the other technical requirements

---

$$P(x) \cdot T(x' \leftarrow x) = P(x) \cdot Q(x';x) \min\left(1,\ \frac{P(x')Q(x;x')}{P(x)Q(x';x)}\right) = \min\Big(P(x)Q(x';x),\ P(x')Q(x;x')\Big)$$

$$= P(x') \cdot Q(x;x') \min\left(1,\ \frac{P(x)Q(x';x)}{P(x')Q(x;x')}\right) = P(x') \cdot T(x \leftarrow x')$$

The Metropolis-Hastings algorithm

Implementation

1: Choose a starting point x^1.
2: **for** $i = 2$ to L **do**
3: Draw a candidate sample x^{cand} from the proposal $\tilde{q}(x'|x^{l-1})$.
4: Let $a = \frac{\tilde{q}(x^{l-1}|x^{cand})p(x^{cand})}{\tilde{q}(x^{cand}|x^{l-1})p(x^{l-1})}$
5: **if** $a \geq 1$ **then** $x^l = x^{cand}$ ▷ Accept the candidate
6: **else**
7: draw a random value u uniformly from the unit interval $[0, 1]$.
8: **if** $u < a$ **then** $x^l = x^{cand}$ ▷ Accept the candidate
9: **else**
10: $x^l = x^{l-1}$ ▷ Reject the candidate
11: **end if**
12: **end if**
13: **end for**

MH-Algorithm example

2D-Gaussian target distribution $P^\star$

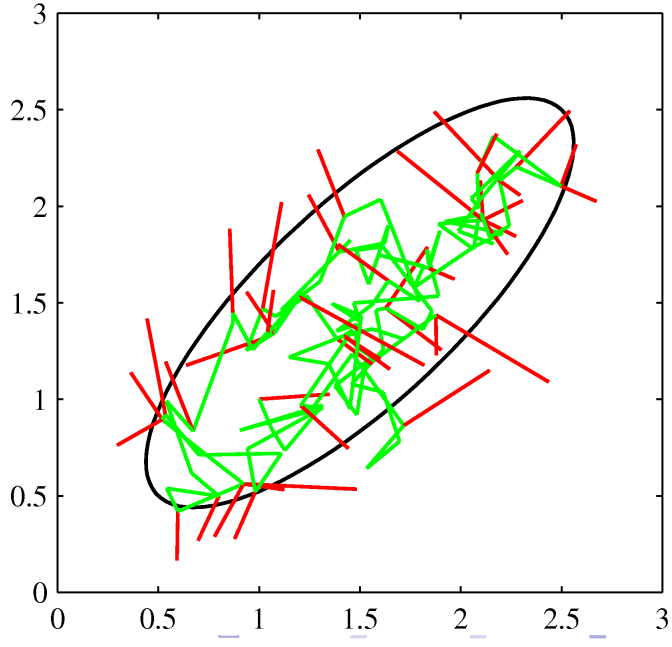
Property: when proposal $q(x, x') > 0$ for all x, the distribution of x_t tends to $p(x)$ as $t \to \infty$.

Note:

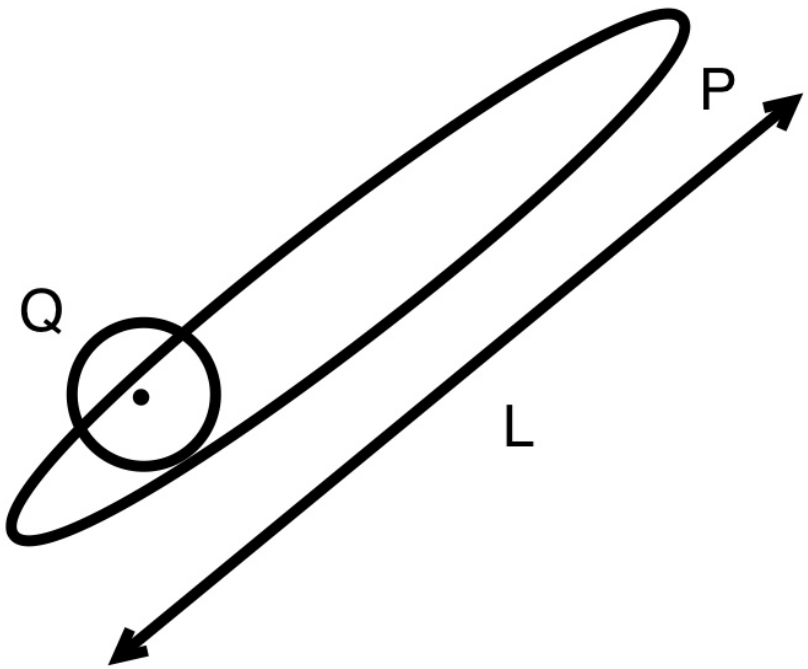
- Sequence $x_1, x_2, \ldots$ is not a set of independent samples from $p(x)$, as successive samples are highly correlated.
- We can obtain (largely) independent samples by just retaining every M^{th} sample (thinning).

Example: Sampling from a Gaussian

- Proposal: Gaussian with $\sigma = 0.2$
- Green: accepted samples
- Red: rejected samples



Example Metropolis-Hastings algorithm

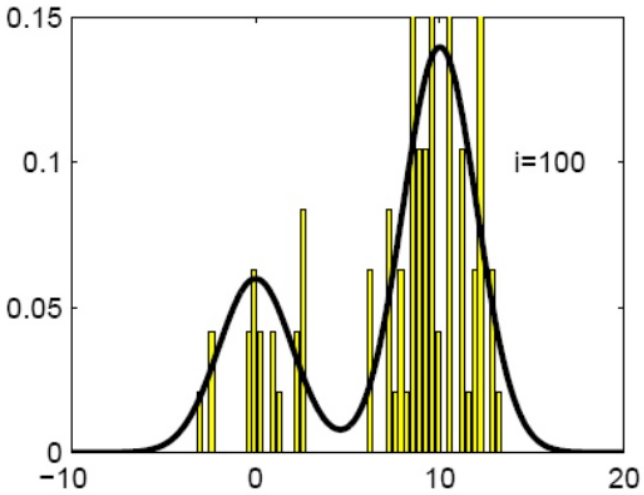


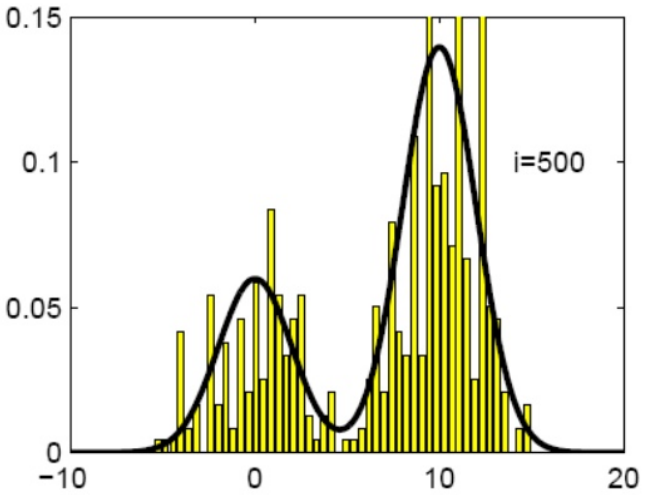
Generic proposals use
$Q(x';x) = \mathcal{N}(x, \sigma^2)$

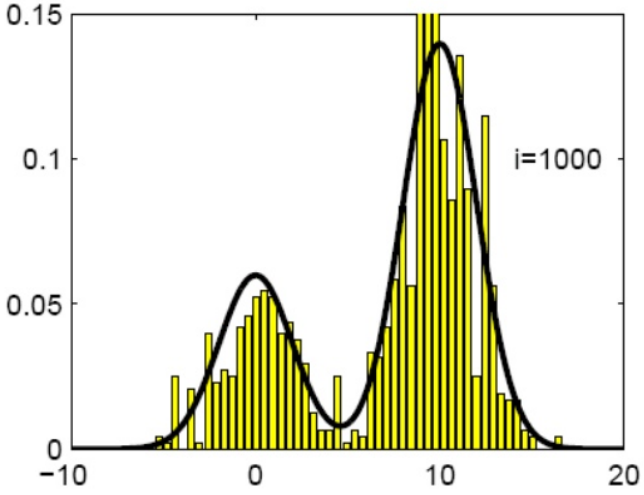
σ large $\rightarrow$ many rejections

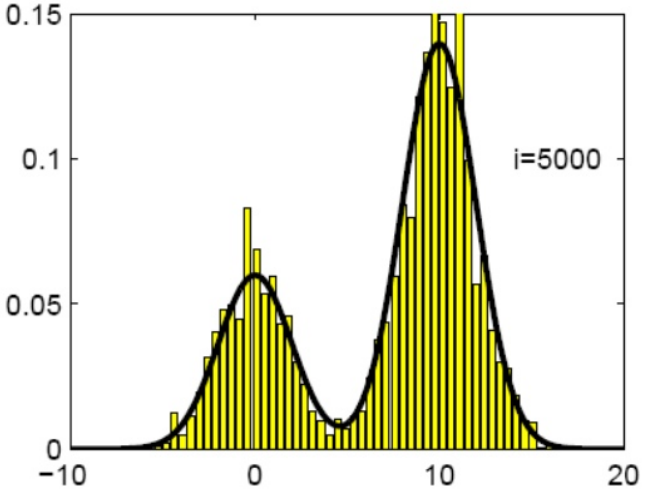
σ small $\rightarrow$ slow diffusion:
$\sim (L/\sigma)^2$ iterations required

Example Metropolis-Hastings algorithm

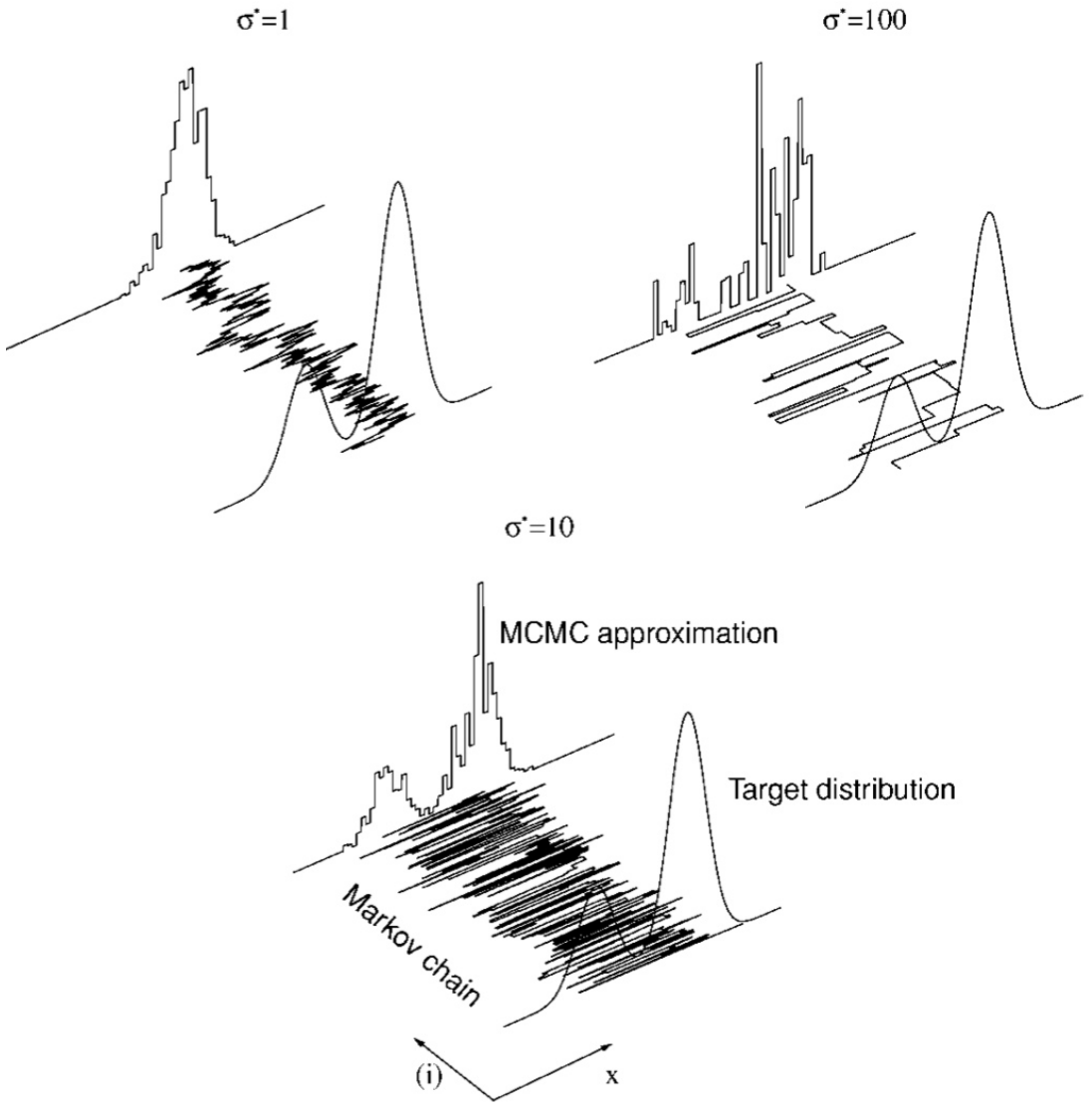








Example Metropolis-Hastings algorithm contd.



Gibbs Sampling

Method for sampling from a multivariate distribution $p(\mathbf{x})$

Idea: sample from a conditional of each variable given the settings of the other variables.
Repeatedly:

1. pick i (either at random or in turn)
2. replace x_i by a sample from the conditional distribution

$$p(x_i | x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_n)$$

This creates a Markov Chain

$$\mathbf{x}^1 \rightarrow \mathbf{x}^2 \rightarrow \mathbf{x}^3 \rightarrow \mathbf{x}^t$$

Under some (mild) conditions, the **equilibrium distribution**, i.e. $p(\mathbf{x}^\infty)$, of this Markov chain is $p(\mathbf{x})$

Gibbs Sampling contd.

Method for sampling from a multivariate distribution $p(\mathbf{x})$ of dimension D

- $x_1^{(t+1)} \sim P\left(x_1 | x_2^{(t)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$
- $x_2^{(t+1)} \sim P\left(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$

  $\ldots$
- $x_D^{(t+1)} \sim P\left(x_D | x_1^{(t+1)}, x_2^{(t+1)}, \ldots x_{D-1}^{(t+1)}\right)$

Gibbs Sampling contd.

Method for sampling from a multivariate distribution $p(\mathbf{x})$ of dimension D

- $x_1^{(t+1)} \sim P\left(x_1 | x_2^{(t)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$
- $x_2^{(t+1)} \sim P\left(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$

  $\ldots$
- $x_D^{(t+1)} \sim P\left(x_D | x_1^{(t+1)}, x_2^{(t+1)}, \ldots x_{D-1}^{(t+1)}\right)$

Gibbs Sampling contd.

Method for sampling from a multivariate distribution $p(\mathbf{x})$ of dimension D

- $x_1^{(t+1)} \sim P\left(x_1 | x_2^{(t)}, x_3^{(t)}, \dots x_D^{(t)}\right)$
- $x_2^{(t+1)} \sim P\left(x_2 | x_1^{(t+1)}, x_3^{(t)}, \dots x_D^{(t)}\right)$

  $\dots$
- $x_D^{(t+1)} \sim P\left(x_D | x_1^{(t+1)}, x_2^{(t+1)}, \dots x_{D-1}^{(t+1)}\right)$

Gibbs Sampling contd.

Method for sampling from a multivariate distribution $p(\mathbf{x})$ of dimension D

- $x_1^{(t+1)} \sim P\left(x_1 | x_2^{(t)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$
- $x_2^{(t+1)} \sim P\left(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$

  $\ldots$
- $x_D^{(t+1)} \sim P\left(x_D | x_1^{(t+1)}, x_2^{(t+1)}, \ldots x_{D-1}^{(t+1)}\right)$

Gibbs Sampling contd.

Method for sampling from a multivariate distribution $p(\mathbf{x})$ of dimension D

- $x_1^{(t+1)} \sim P\left(x_1 | x_2^{(t)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$
- $x_2^{(t+1)} \sim P\left(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$

  $\ldots$
- $x_D^{(t+1)} \sim P\left(x_D | x_1^{(t+1)}, x_2^{(t+1)}, \ldots x_{D-1}^{(t+1)}\right)$

Gibbs Sampling contd.

Method for sampling from a multivariate distribution $p(\mathbf{x})$ of dimension D

- $x_1^{(t+1)} \sim P\left(x_1 | x_2^{(t)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$
- $x_2^{(t+1)} \sim P\left(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$

  $\ldots$
- $x_D^{(t+1)} \sim P\left(x_D | x_1^{(t+1)}, x_2^{(t+1)}, \ldots x_{D-1}^{(t+1)}\right)$

Gibbs Sampling contd.

Method for sampling from a multivariate distribution $p(\mathbf{x})$ of dimension D

- $x_1^{(t+1)} \sim P\left(x_1 | x_2^{(t)}, x_3^{(t)}, \dots x_D^{(t)}\right)$
- $x_2^{(t+1)} \sim P\left(x_2 | x_1^{(t+1)}, x_3^{(t)}, \dots x_D^{(t)}\right)$

  $\dots$
- $x_D^{(t+1)} \sim P\left(x_D | x_1^{(t+1)}, x_2^{(t+1)}, \dots x_{D-1}^{(t+1)}\right)$

Gibbs Sampling contd.

Method for sampling from a multivariate distribution $p(\mathbf{x})$ of dimension D

- $x_1^{(t+1)} \sim P\left(x_1 | x_2^{(t)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$
- $x_2^{(t+1)} \sim P\left(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$

  $\ldots$
- $x_D^{(t+1)} \sim P\left(x_D | x_1^{(t+1)}, x_2^{(t+1)}, \ldots x_{D-1}^{(t+1)}\right)$

Gibbs Sampling contd.

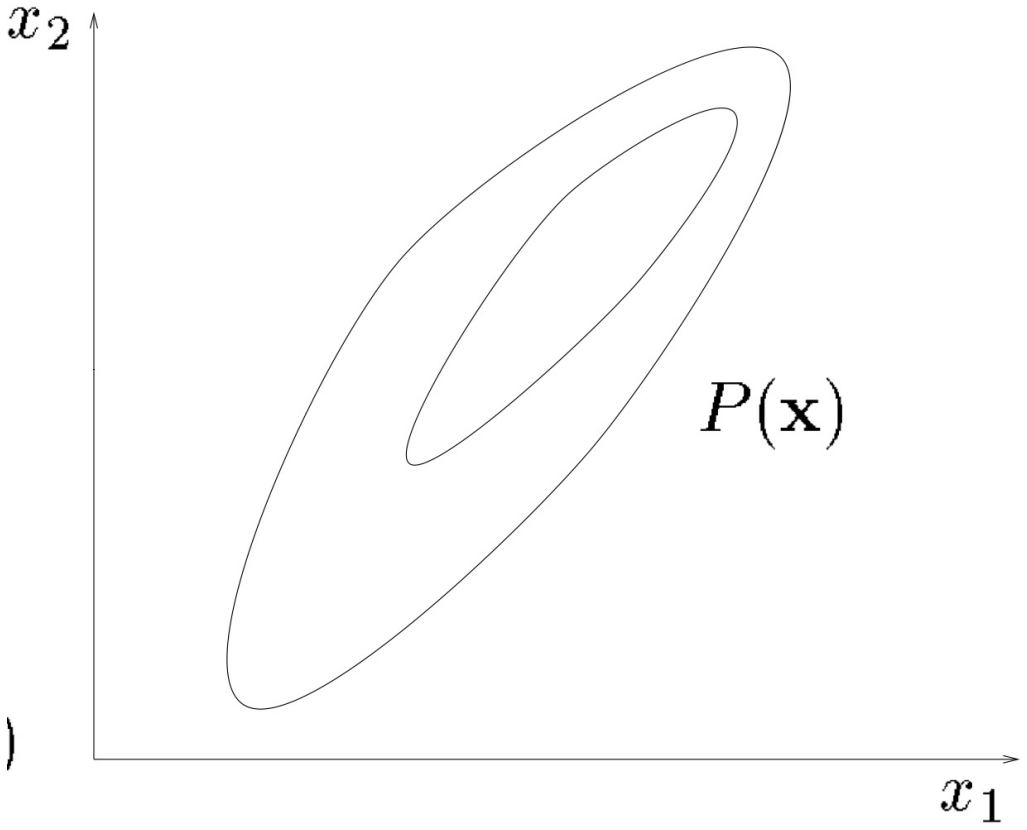
Method for sampling from a multivariate distribution $p(\mathbf{x})$ of dimension D

- $x_1^{(t+1)} \sim P\left(x_1 | x_2^{(t)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$
- $x_2^{(t+1)} \sim P\left(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots x_D^{(t)}\right)$

  $\ldots$
- $x_D^{(t+1)} \sim P\left(x_D | x_1^{(t+1)}, x_2^{(t+1)}, \ldots x_{D-1}^{(t+1)}\right)$

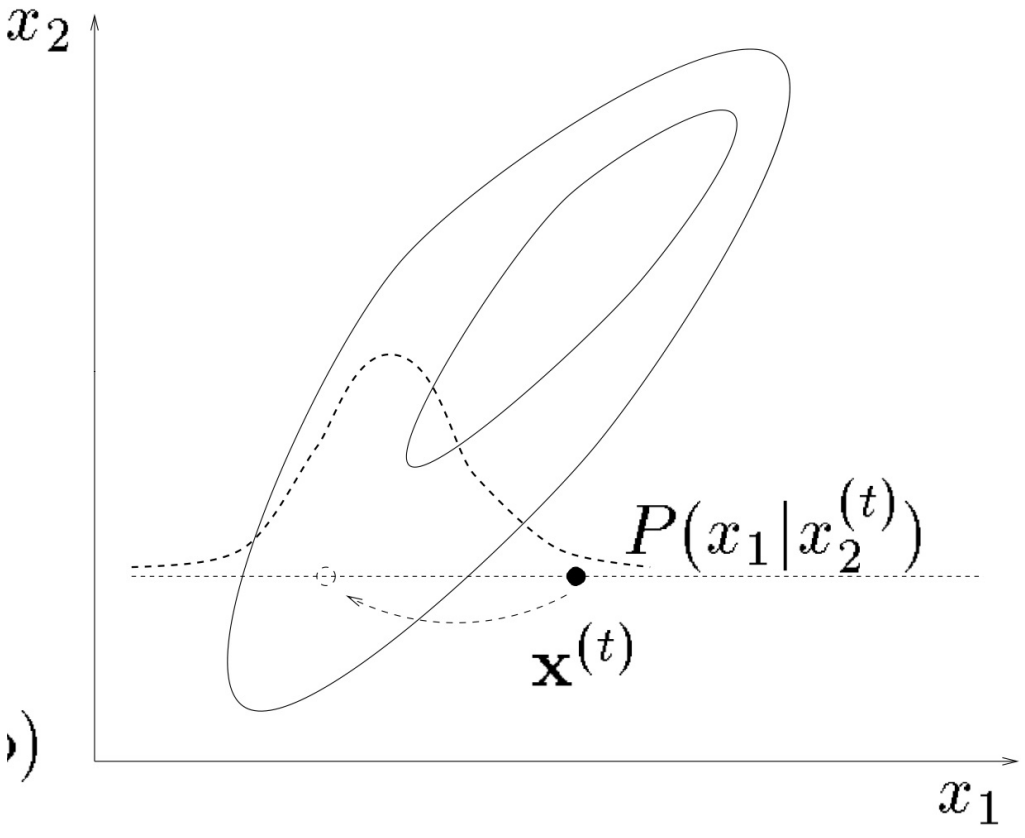
Gibbs Sampling contd.

Method for sampling from a multivariate distribution $p(\mathbf{x})$



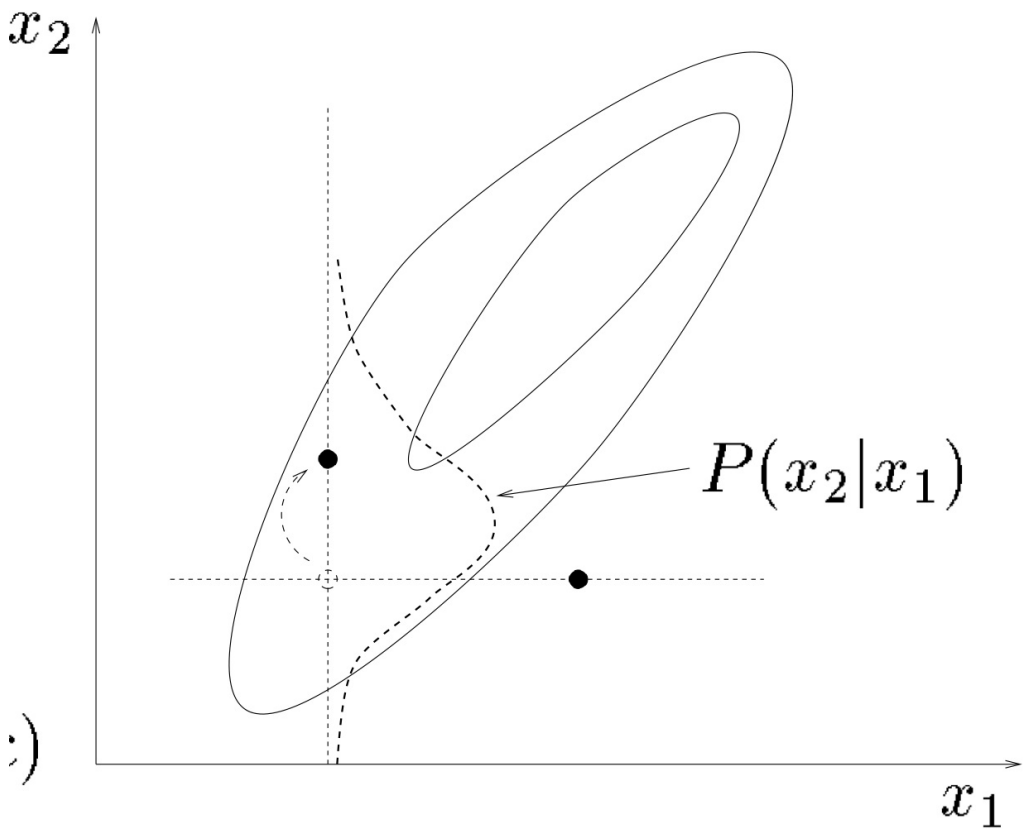
Gibbs Sampling contd.

Method for sampling from a multivariate distribution $p(\mathbf{x})$



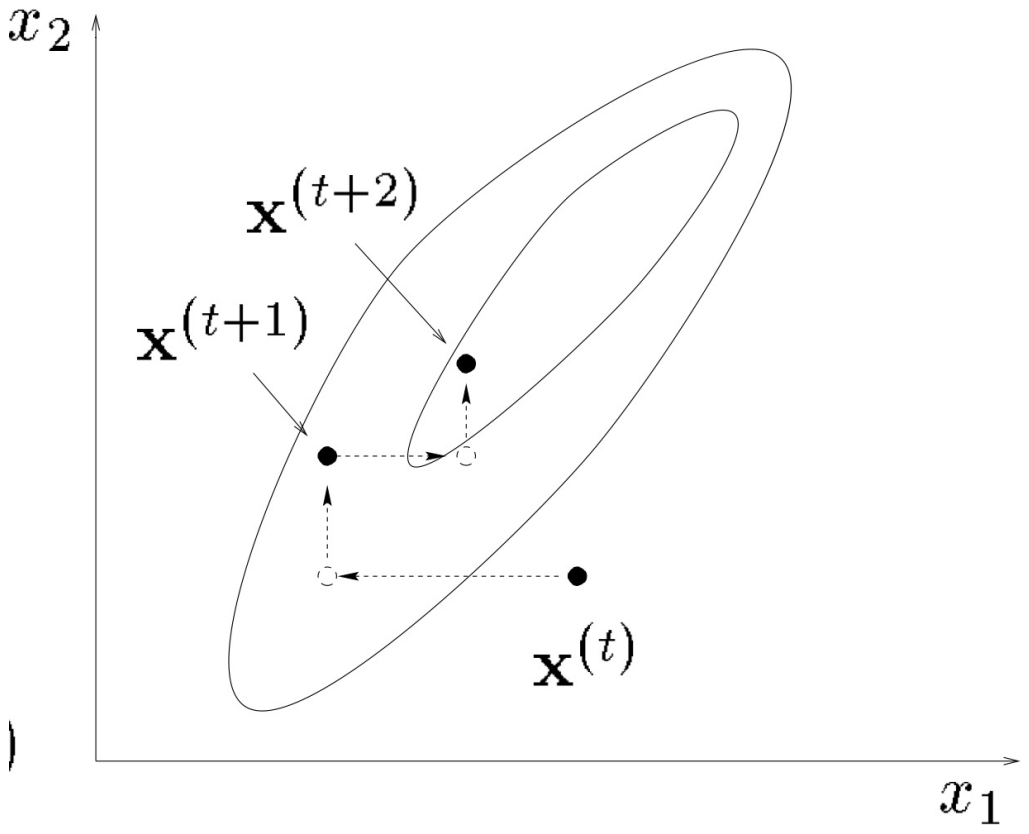
Gibbs Sampling contd.

Method for sampling from a multivariate distribution $p(\mathbf{x})$



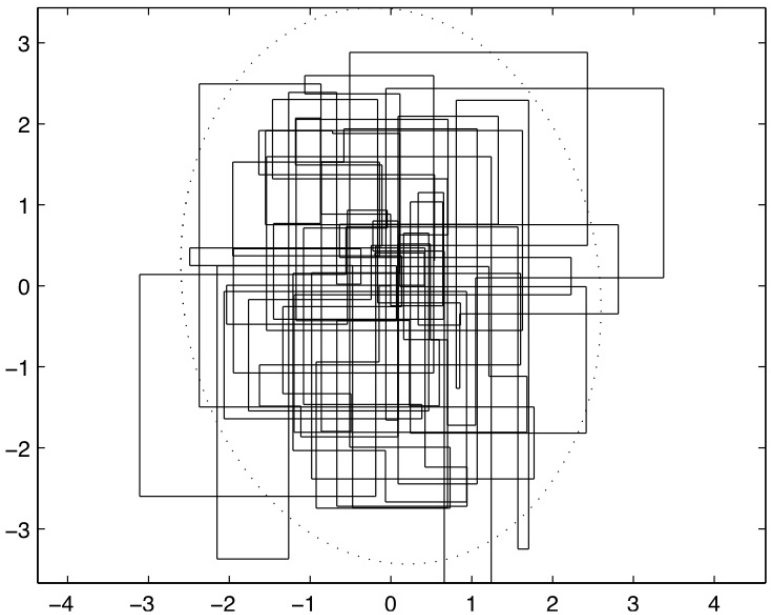
Gibbs Sampling contd.

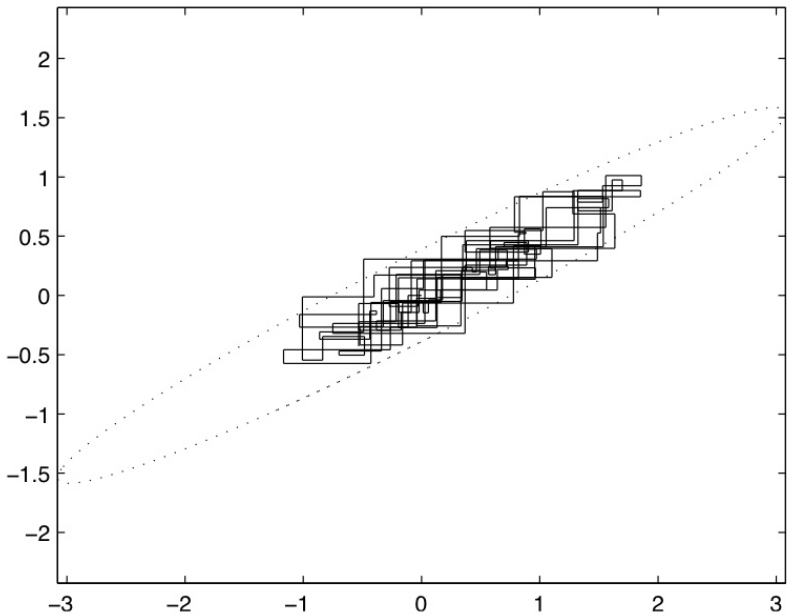
Method for sampling from a multivariate distribution $p(\mathbf{x})$



Slow convergence for correlated distributions

Sample from at least 2-Dimensional distributions

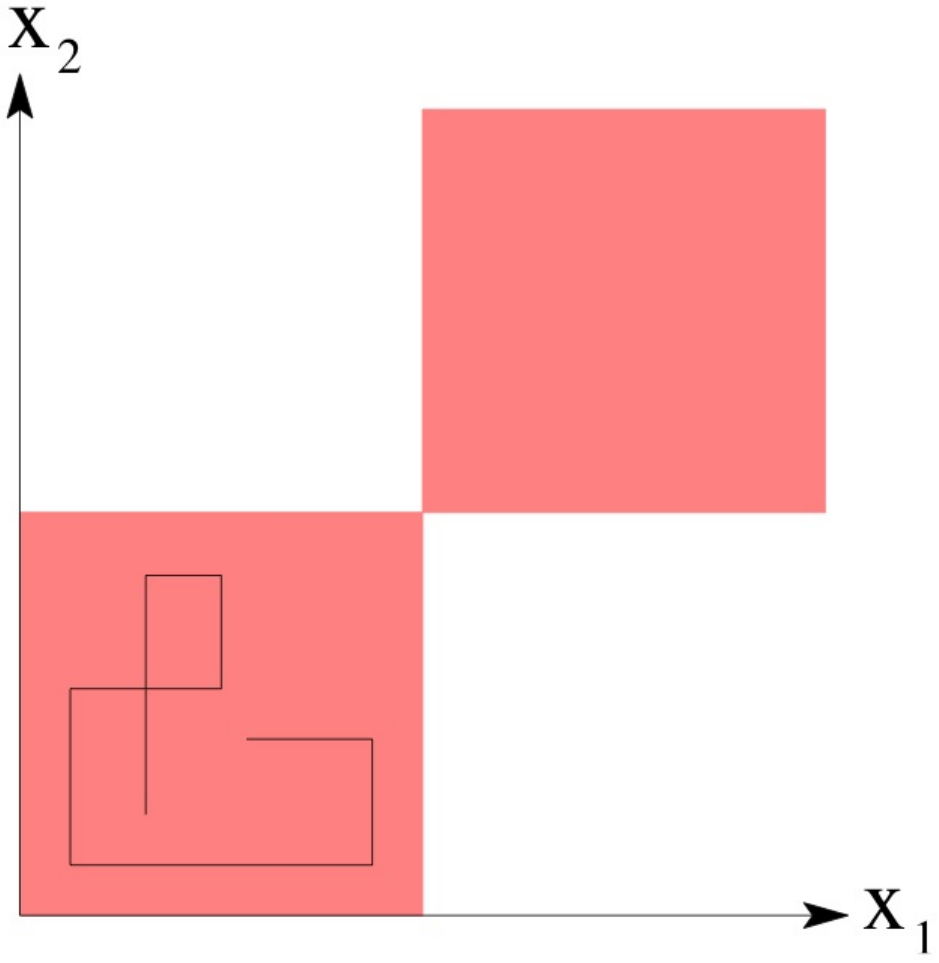




Left: Gibbs-Sampling from more uncorrelated distribution

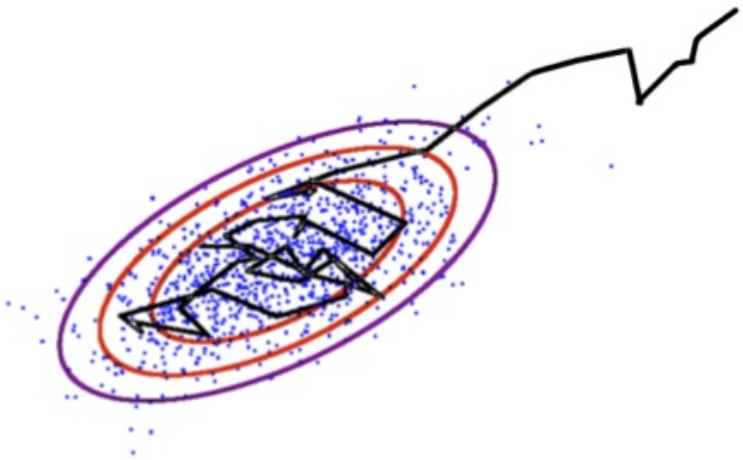
Right: Gibbs-Sampling from more correlated distribution

Ergodic/convergence behavior not guaranteed



Short review

Construct a biased random walk that explores a target dist.



Markov steps, $x^{(s)} \sim T\left(x^{(s)} \leftarrow x^{(s-1)}\right)$

MCMC gives approximate, correlated samples

$$\mathbb{E}_P[f] \approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)})$$

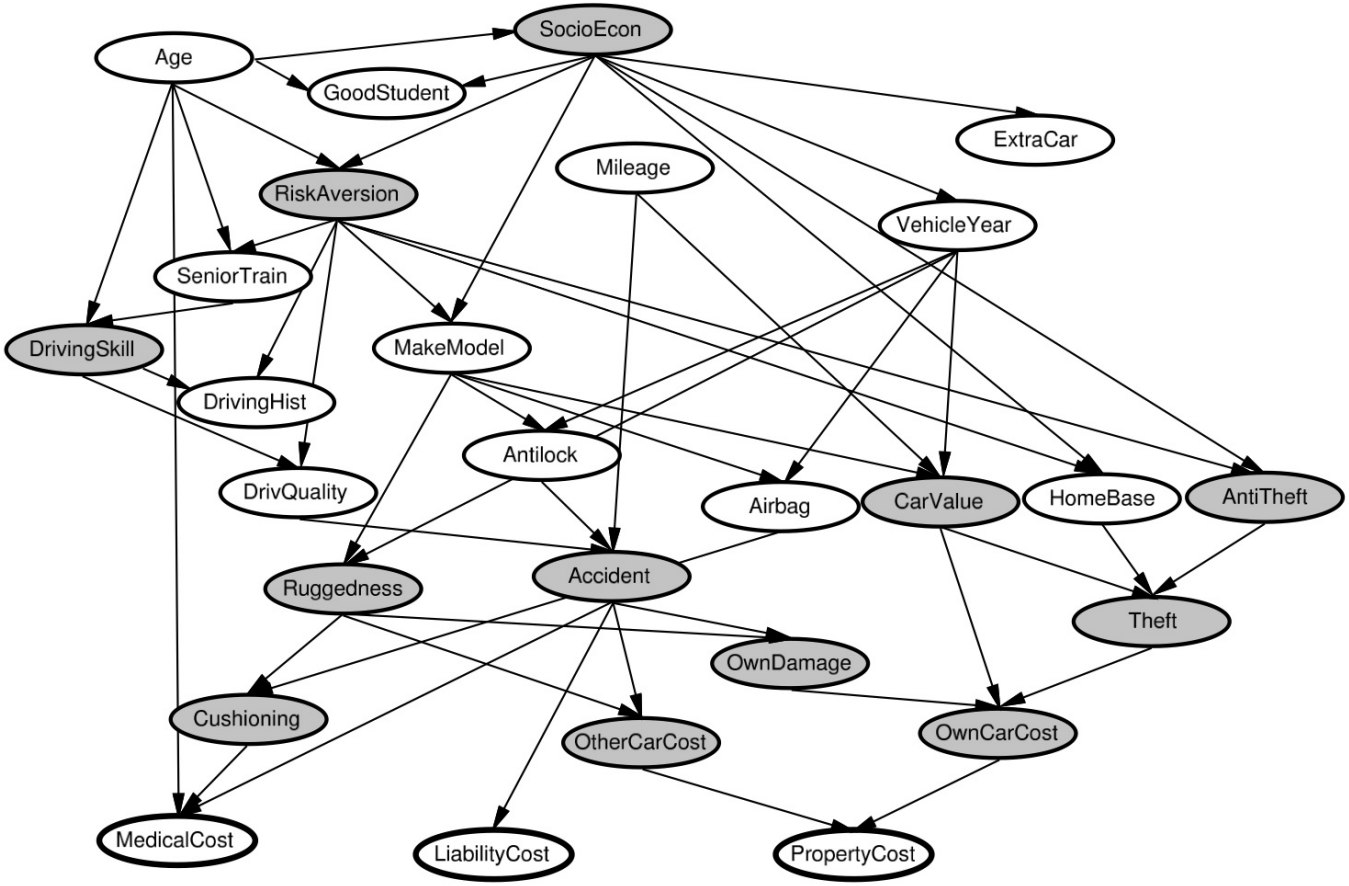
Example transitions:

Metropolis–Hastings: $T(x' \leftarrow x) = Q(x'; x)\, \min\left(1,\, \frac{P(x')\, Q(x; x')}{P(x)\, Q(x'; x)}\right)$

Gibbs sampling: $T_i(\mathbf{x}' \leftarrow \mathbf{x}) = P(x'_i \,|\, \mathbf{x}_{j\neq i})\, \delta(\mathbf{x}'_{j\neq i} - \mathbf{x}_{j\neq i})$

Gibbs Sampling

Gibbs Sampling for Graphical models



- Initialize all variables to some settings.
- Sample each variable conditioned on other variables.

Gibbs Sampling

Gibbs Sampling for Graphical models

A large-dimensional joint distribution is factored into a directed graph that encodes the conditional independencies in the model. In particular, if $x_{pa(j)}$ denotes the parent nodes of node x_j, we have

$$p(x) = \prod_j p\left(x_j | x_{pa(j)}\right) \tag{2}$$

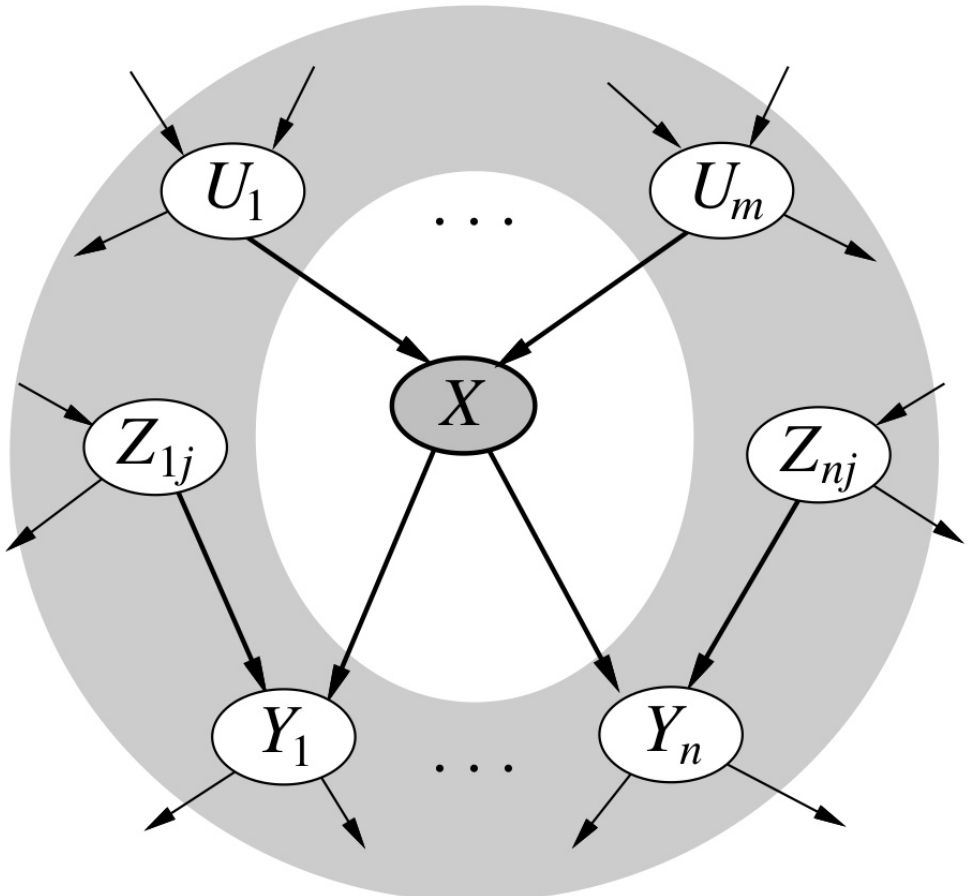
It follows that the full conditionals simplify as follows

$$p\left(x_j = |x_{-j}\right) = p\left(x_j | x_{pa(j)}\right) \prod_{k \in ch(j)} p\left(x_k | x_{pa(k)}\right) \tag{3}$$

where $ch(j)$ denotes the children nodes of x_j.

Gibbs Sampling in DAGs

- Recap: The Markov blanket of a node is the set that renders it independent of the rest of the graph
- This is the parents, children and co-parents.



Gibbs Sampling in DAGs

- Recap: The Markov blanket of a node is the set that renders it independent of the rest of the graph
- This is the parents, children and co-parents.

$$
\begin{aligned}
p\left(X_i|X_{-i}\right) &= \frac{p\left(X_i, X_{-i}\right)}{\sum_x p\left(X_i, X_{-i}\right)} \\
&= \frac{p\left(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R\right)}{\sum_x p\left(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R\right)} \\
&= \frac{p\left(X_i|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i, Z_j\right)\right] p\left(U_{1:n}, Z_{1:m}, R\right)}{\sum_x p\left(X_i = x|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i = x, Z_j\right)\right] p\left(U_{1:n}, Z_{1:m}, R\right)} \\
&= \frac{p\left(X_i|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i, Z_j\right)\right]}{\sum_x p\left(X_i = x|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i = x, Z_j\right)\right]} \\
p\left(X_i|X_{-i}\right) &\propto p\left(X_i|Pa\left(X_i\right)\right) \prod_{Y_i \in ch(X_i)} p\left(Y_i|Pa\left(Y_i\right)\right)
\end{aligned}
$$

Gibbs Sampling in DAGs

- Recap: The Markov blanket of a node is the set that renders it independent of the rest of the graph
- This is the parents, children and co-parents.

$$
\begin{aligned}
p(X_i|X_{-i}) &= \frac{p(X_i, X_{-i})}{\sum_x p(X_i, X_{-i})} \\
&= \frac{p(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R)}{\sum_x p(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R)} \\
&= \frac{p(X_i|U_{1:n}) \left[\prod_j p(Y_j|X_i, Z_j)\right] p(U_{1:n}, Z_{1:m}, R)}{\sum_x p(X_i = x|U_{1:n}) \left[\prod_j p(Y_j|X_i = x, Z_j)\right] p(U_{1:n}, Z_{1:m}, R)} \\
&= \frac{p(X_i|U_{1:n}) \left[\prod_j p(Y_j|X_i, Z_j)\right]}{\sum_x p(X_i = x|U_{1:n}) \left[\prod_j p(Y_j|X_i = x, Z_j)\right]} \\
p(X_i|X_{-i}) &\propto p(X_i|Pa(X_i)) \prod_{Y_i \in ch(X_i)} p(Y_i|Pa(Y_i))
\end{aligned}
$$

Gibbs Sampling in DAGs

- Recap: The Markov blanket of a node is the set that renders it independent of the rest of the graph
- This is the parents, children and co-parents.

$$p(X_i|X_{-i}) = \frac{p(X_i, X_{-i})}{\sum_x p(X_i, X_{-i})}$$

$$= \frac{p(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R)}{\sum_x p(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R)}$$

$$= \frac{p(X_i|U_{1:n}) \left[\prod_j p(Y_j|X_i, Z_j)\right] p(U_{1:n}, Z_{1:m}, R)}{\sum_x p(X_i = x|U_{1:n}) \left[\prod_j p(Y_j|X_i = x, Z_j)\right] p(U_{1:n}, Z_{1:m}, R)}$$

$$= \frac{p(X_i|U_{1:n}) \left[\prod_j p(Y_j|X_i, Z_j)\right]}{\sum_x p(X_i = x|U_{1:n}) \left[\prod_j p(Y_j|X_i = x, Z_j)\right]}$$

$$p(X_i|X_{-i}) \propto p(X_i|Pa(X_i)) \prod_{Y_i \in ch(X_i)} p(Y_i|Pa(Y_i))$$

Gibbs Sampling in DAGs

- Recap: The Markov blanket of a node is the set that renders it independent of the rest of the graph
- This is the parents, children and co-parents.

$$p(X_i|X_{-i}) = \frac{p(X_i, X_{-i})}{\sum_x p(X_i, X_{-i})}$$

$$= \frac{p(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R)}{\sum_x p(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R)}$$

$$= \frac{p(X_i|U_{1:n}) \left[\prod_j p(Y_j|X_i, Z_j)\right] p(U_{1:n}, Z_{1:m}, R)}{\sum_x p(X_i = x|U_{1:n}) \left[\prod_j p(Y_j|X_i = x, Z_j)\right] p(U_{1:n}, Z_{1:m}, R)}$$

$$= \frac{p(X_i|U_{1:n}) \left[\prod_j p(Y_j|X_i, Z_j)\right]}{\sum_x p(X_i = x|U_{1:n}) \left[\prod_j p(Y_j|X_i = x, Z_j)\right]}$$

$$p(X_i|X_{-i}) \propto p(X_i|Pa(X_i)) \prod_{Y_i \in ch(X_i)} p(Y_i|Pa(Y_i))$$

Gibbs Sampling in DAGs

- Recap: The Markov blanket of a node is the set that renders it independent of the rest of the graph
- This is the parents, children and co-parents.

$$
\begin{aligned}
p(X_i|X_{-i}) &= \frac{p(X_i, X_{-i})}{\sum_x p(X_i, X_{-i})} \\
&= \frac{p(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R)}{\sum_x p(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R)} \\
&= \frac{p(X_i|U_{1:n}) \left[\prod_j p(Y_j|X_i, Z_j)\right] p(U_{1:n}, Z_{1:m}, R)}{\sum_x p(X_i = x|U_{1:n}) \left[\prod_j p(Y_j|X_i = x, Z_j)\right] p(U_{1:n}, Z_{1:m}, R)} \\
&= \frac{p(X_i|U_{1:n}) \left[\prod_j p(Y_j|X_i, Z_j)\right]}{\sum_x p(X_i = x|U_{1:n}) \left[\prod_j p(Y_j|X_i = x, Z_j)\right]} \\
p(X_i|X_{-i}) &\propto p(X_i|Pa(X_i)) \prod_{Y_i \in ch(X_i)} p(Y_i|Pa(Y_i))
\end{aligned}
$$

Gibbs Sampling in DAGs

- Recap: The Markov blanket of a node is the set that renders it independent of the rest of the graph
- This is the parents, children and co-parents.

$$
\begin{aligned}
p(X_i|X_{-i}) &= \frac{p(X_i, X_{-i})}{\sum_x p(X_i, X_{-i})} \\
&= \frac{p(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R)}{\sum_x p(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R)} \\
&= \frac{p(X_i|U_{1:n}) \left[\prod_j p(Y_j|X_i, Z_j)\right] p(U_{1:n}, Z_{1:m}, R)}{\sum_x p(X_i = x|U_{1:n}) \left[\prod_j p(Y_j|X_i = x, Z_j)\right] p(U_{1:n}, Z_{1:m}, R)} \\
&= \frac{p(X_i|U_{1:n}) \left[\prod_j p(Y_j|X_i, Z_j)\right]}{\sum_x p(X_i = x|U_{1:n}) \left[\prod_j p(Y_j|X_i = x, Z_j)\right]}
\end{aligned}
$$

$$
p(X_i|X_{-i}) \propto p(X_i|Pa(X_i)) \prod_{Y_i \in ch(X_i)} p(Y_i|Pa(Y_i))
$$

Gibbs Sampling in DAGs

- Recap: The Markov blanket of a node is the set that renders it independent of the rest of the graph
- This is the parents, children and co-parents.

$$
\begin{aligned}
p\left(X_i|X_{-i}\right) &= \frac{p\left(X_i, X_{-i}\right)}{\sum_x p\left(X_i, X_{-i}\right)} \\
&= \frac{p\left(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R\right)}{\sum_x p\left(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R\right)} \\
&= \frac{p\left(X_i|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i, Z_j\right)\right] p\left(U_{1:n}, Z_{1:m}, R\right)}{\sum_x p\left(X_i = x|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i = x, Z_j\right)\right] p\left(U_{1:n}, Z_{1:m}, R\right)} \\
&= \frac{p\left(X_i|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i, Z_j\right)\right]}{\sum_x p\left(X_i = x|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i = x, Z_j\right)\right]}
\end{aligned}
$$

$$
p\left(X_i|X_{-i}\right) \propto p\left(X_i|Pa\left(X_i\right)\right) \prod_{Y_i \in ch(X_i)} p\left(Y_i|Pa\left(Y_i\right)\right)
$$

Gibbs Sampling in DAGs

- Recap: The Markov blanket of a node is the set that renders it independent of the rest of the graph
- This is the parents, children and co-parents.

$$
\begin{aligned}
p\left(X_i|X_{-i}\right) &= \frac{p\left(X_i, X_{-i}\right)}{\sum_x p\left(X_i, X_{-i}\right)} \\
&= \frac{p\left(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R\right)}{\sum_x p\left(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, R\right)} \\
&= \frac{p\left(X_i|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i, Z_j\right)\right] p\left(U_{1:n}, Z_{1:m}, R\right)}{\sum_x p\left(X_i = x|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i = x, Z_j\right)\right] p\left(U_{1:n}, Z_{1:m}, R\right)} \\
&= \frac{p\left(X_i|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i, Z_j\right)\right]}{\sum_x p\left(X_i = x|U_{1:n}\right)\left[\prod_j p\left(Y_j|X_i = x, Z_j\right)\right]}
\end{aligned}
$$

$$
p\left(X_i|X_{-i}\right) \propto p\left(X_i|Pa\left(X_i\right)\right) \prod_{Y_i \in ch(X_i)} p\left(Y_i|Pa\left(Y_i\right)\right)
$$

Gibbs Sampling scheme

- In Gibbs sampling we also generate a sample set S – but in this case the samples are not independent from each other anymore. The next sample "modifies" the previous one:
- First, all observed RVs are *clamped* to their fixed value $x_i^k = y_i$ for any k.
- To generate the $(k+1)$th sample, iterate through the latent variables $i \notin obs$, updating:

$$
\begin{aligned}
x_i^{k+1} &\sim P(X_i \mid x_{1:n\setminus i}^k) \\
&\sim P(X_i \mid x_1^k, x_2^k, .., x_{i\text{-}1}^k, x_{i+1}^k, .., x_n^k) \\
&\sim P(X_i \mid x_{\text{Parents}(i)}^k) \prod_{j:i\in\text{Parents}(j)} P(X_j = x_j^k \mid X_i, x_{\text{Parents}(j)\setminus i}^k)
\end{aligned}
$$

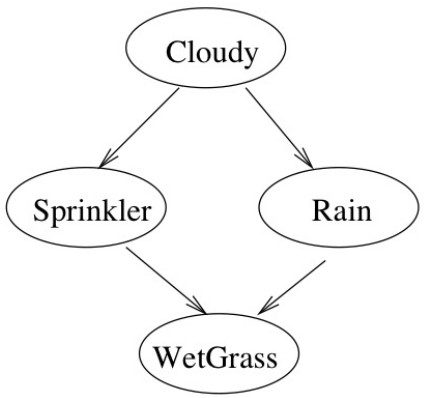
That is, each x_i^{k+1} is *resampled* conditional to the other (neighboring) current sample values.

Gibbs Sampling

Famous sprinkler DAG example

Binary nodes: (T & F)

P(C=F)	P(C=T)
0.5	0.5



C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

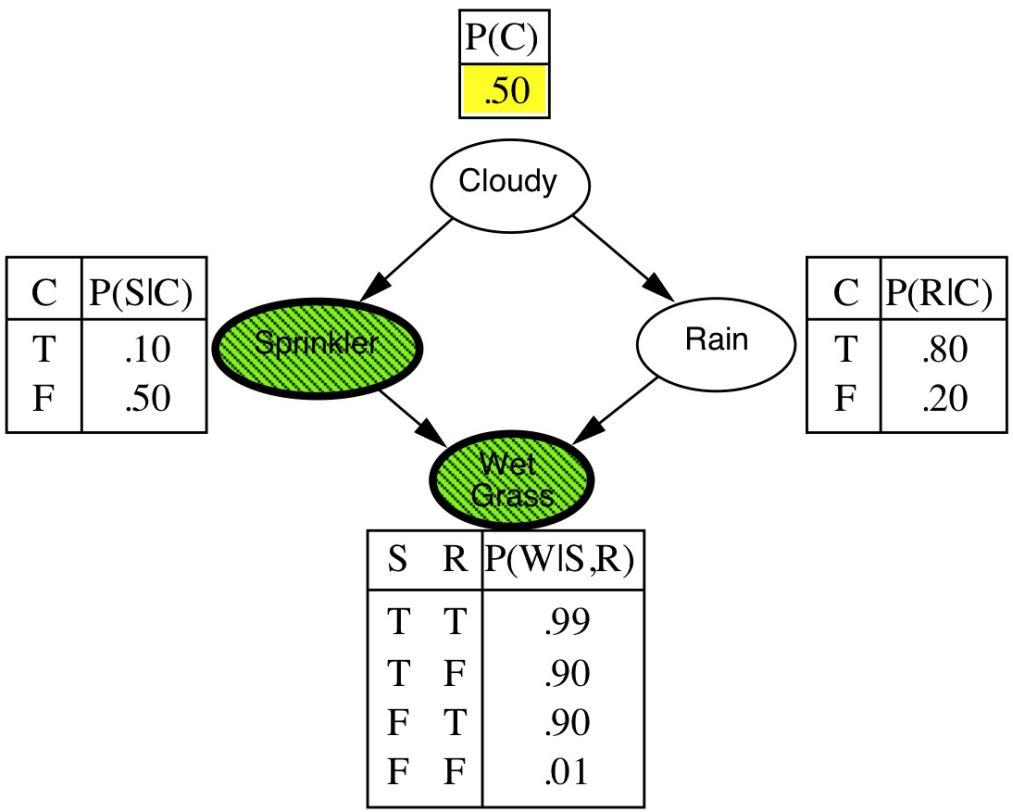
S	R	P(W=F)	P(W=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

Gibbs Sampling for sprinkler DAG with evidence

1. Initialize:
   (a) Instantiate X_i to one of its possible values $x_i, 1 \leq i \leq n$.
   (b) Let $\mathbf{x}^{(0)} = (x_1, \ldots, x_n)$
2. For $t = 1, 2, \ldots$
   (a) Pick an index $i, 1 \leq i \leq n$ uniformly at random.[1]
   (b) Sample x_i from $P(X_i | \mathbf{x}_{(-i)}^{(t-1)}, \mathbf{e})$.
   (c) Let $\mathbf{x}^{(t)} = (\mathbf{x}_{(-i)}, x_i)$

Gibbs Sampling for sprinkler DAG

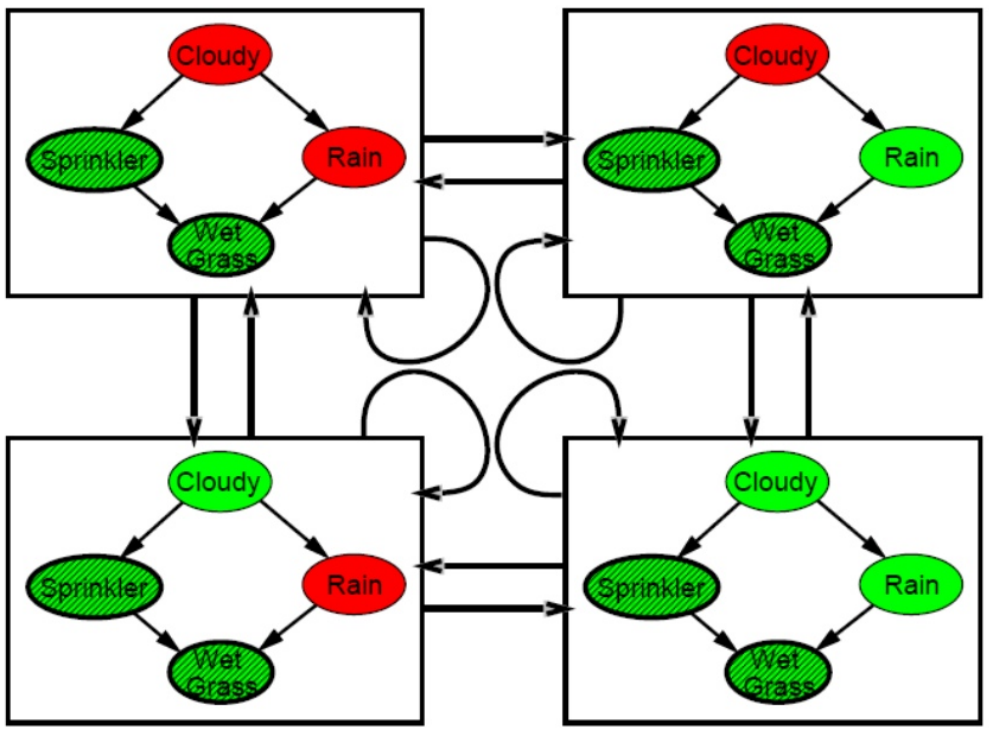
Question: $P(Rain|Sprinkler = true, WetGrass = true)$.
Since the observation/evidence *Sprinkler* and *WetGrass* is set to *true*, the Gibbs sampler draws samples from
$P(Rain, Cloudy|Sprinkler = true, WetGrass = true)$.



Markov Chain Gibbs Sampling for Sprinkler Problem

Question: $P(Rain|Sprinkler = true, WetGrass = true)$.
Since the observation/evidence *Sprinkler* and *WetGrass* is set to *true*, the Gibbs sampler draws samples from
$P(Rain, Cloudy|Sprinkler = true, WetGrass = true)$.

With $Sprinkler = true, WetGrass = true$, there are four states:

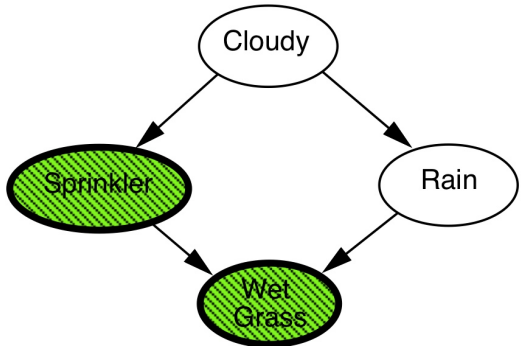


Wander about for a while, average what you see

Gibbs Sampling for sprinkler DAG

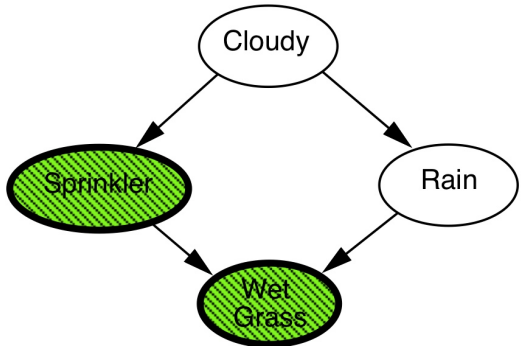
The Gibbs sampler for the rain network works as follows:

1. **Initialize**: (not.: $T = true$, $F = false$)
   1. Instantiate $Rain = T$, $Cloudy = T$
   2. Let $\mathbf{x}^{(0)} = (Rain = T, Cloudy = T)$
2. **For** $t = 1, 2, \ldots$:
   1. Pick variable to update from $\{Rain, Cloudy\}$ **uniformly** at random
   2. If *Rain* was picked
      1. Sample *Rain* from $P\left(Rain | Cloudy = [value]_{t-1}, Sprinkler = T, WetGrass = T\right)$
      2. Let $\mathbf{x}^{(t)} = \left(Rain = [value]_t, Cloudy = [value]_{t-1}\right)$
   3. Else
      1. Sample *Cloudy* from $P\left(Cloudy | Rain = [value]_{t-1}, Sprinkler = T, WetGrass = T\right)$
      2. Let $\mathbf{x}^{(t)} = \left(Rain = [value]_{t-1}, Cloudy = [value]_t\right)$



Gibbs Sampling for sprinkler DAG

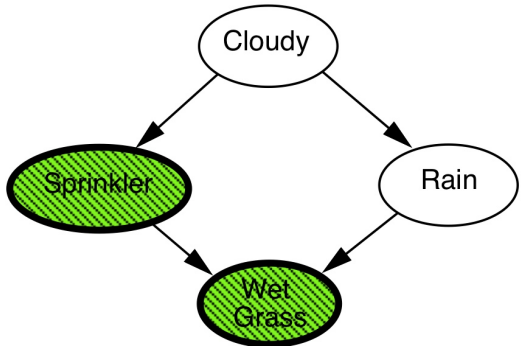
The Gibbs sampler for the rain network works as follows:



1. **Initialize**: (not.: $T = true$, $F = false$)
   1. Instantiate $Rain = T$, $Cloudy = T$
   2. Let $\mathbf{x}^{(0)} = (Rain = T, Cloudy = T)$
2. **For** $t = 1, 2, \ldots$:
   1. Pick variable to update from $\{Rain, Cloudy\}$ **uniformly** at random
   2. If $Rain$ was picked
      1. Sample $Rain$ from $P\left(Rain | Cloudy = [value]_{t-1}, Sprinkler = T, WetGrass = T\right)$
      2. Let $\mathbf{x}^{(t)} = \left(Rain = [value]_t, Cloudy = [value]_{t-1}\right)$
   3. Else
      1. Sample $Cloudy$ from $P\left(Cloudy | Rain = [value]_{t-1}, Sprinkler = T, WetGrass = T\right)$
      2. Let $\mathbf{x}^{(t)} = \left(Rain = [value]_{t-1}, Cloudy = [value]_t\right)$

Gibbs Sampling for sprinkler DAG

The Gibbs sampler for the rain network works as follows:



1. **Initialize**: (not.: $T = true$, $F = false$)
   1. Instantiate $Rain = T$, $Cloudy = T$
   2. Let $\mathbf{x}^{(0)} = (Rain = T, Cloudy = T)$
2. **For** $t = 1, 2, \ldots$:
   1. Pick variable to update from $\{Rain, Cloudy\}$ **uniformly** at random
   2. If $Rain$ was picked
      1. Sample $Rain$ from $P\left(Rain | Cloudy = [value]_{t-1}, Sprinkler = T, WetGrass = T\right)$
      2. Let $\mathbf{x}^{(t)} = \left(Rain = [value]_t, Cloudy = [value]_{t-1}\right)$
   3. Else
      1. Sample $Cloudy$ from $P\left(Cloudy | Rain = [value]_{t-1}, Sprinkler = T, WetGrass = T\right)$
      2. Let $\mathbf{x}^{(t)} = \left(Rain = [value]_{t-1}, Cloudy = [value]_t\right)$

Approximate inference by sampling

- Exact Bayesian Inference often intractable
- Rejection and Importance Sampling
  - Generate independent samples.
  - Impractical in high-dimensional state spaces.
- Markov Chain Monte Carlo (MCMC)
  - Simple & effective (even though typically computationally expensive).
  - Scales well with the dimensionality of the state space.
  - Issues of convergence have to be considered carefully.
- Gibbs Sampling
  - Used extensively in practice.
  - Parameter free.
  - Requires sampling conditional distributions.