Lecture 11

Sampling

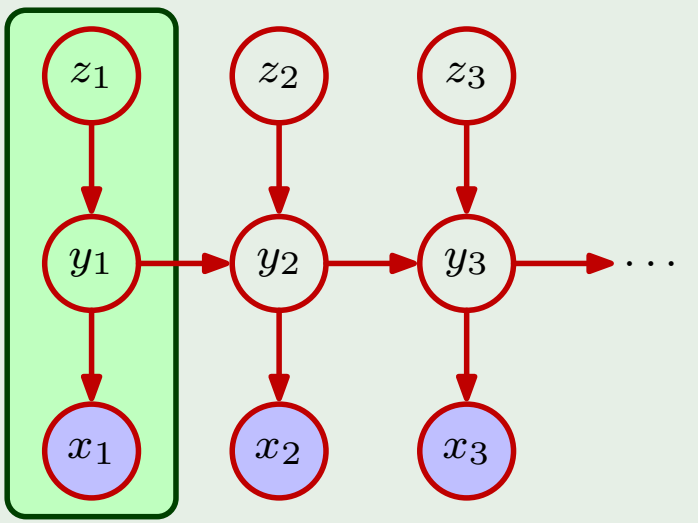
University of Amsterdam

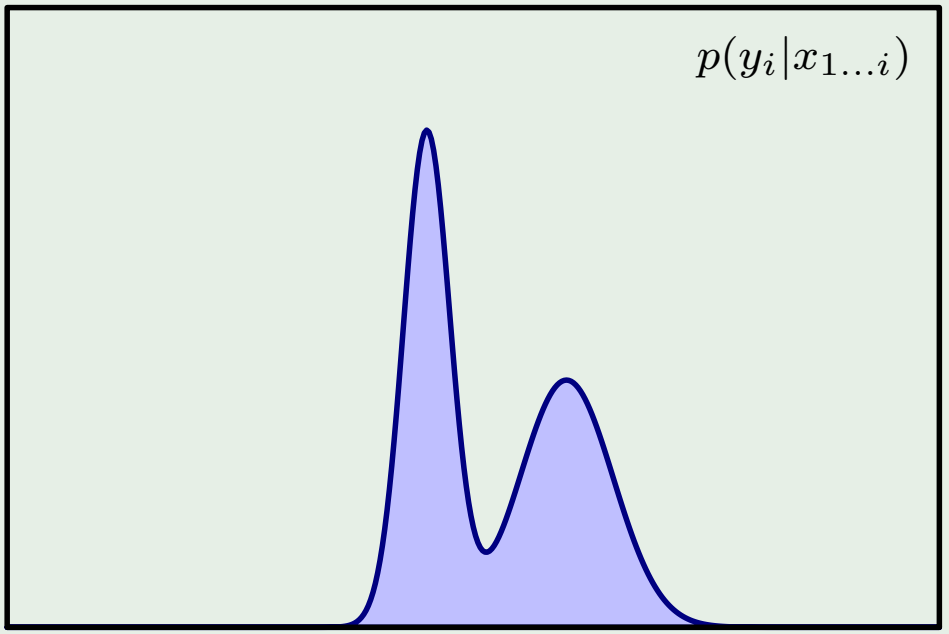
Introduction

Recall how graphical models specify the factorisation of a joint probability distribution

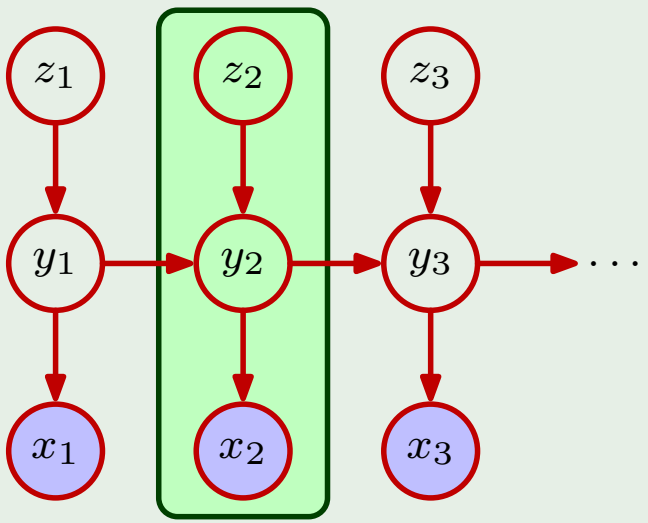
- It provides us with algorithms to do marginalisation efficiently
- We are still free to choose the specific form of the distributions
- However choosing the form is tricky: if we do not choose wisely, the distribution may not be computable analytically

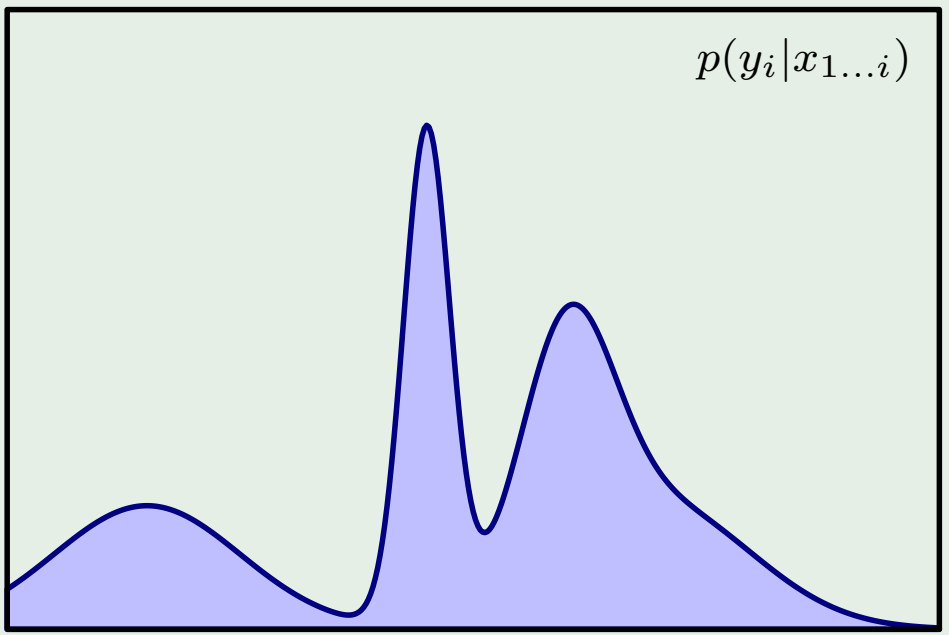
Example: Intractable distributions



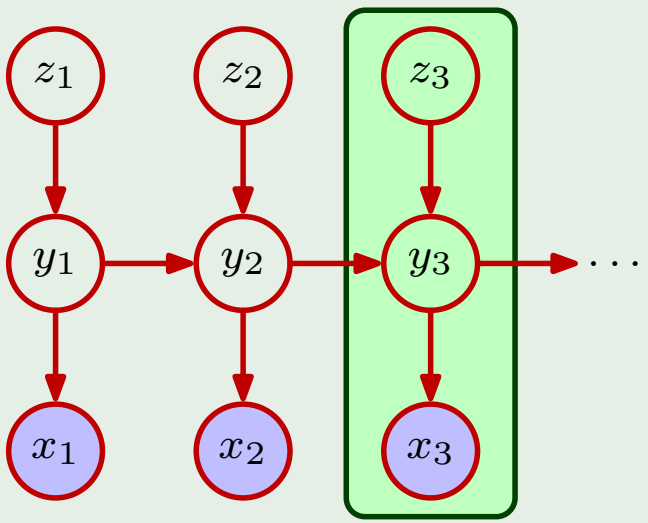


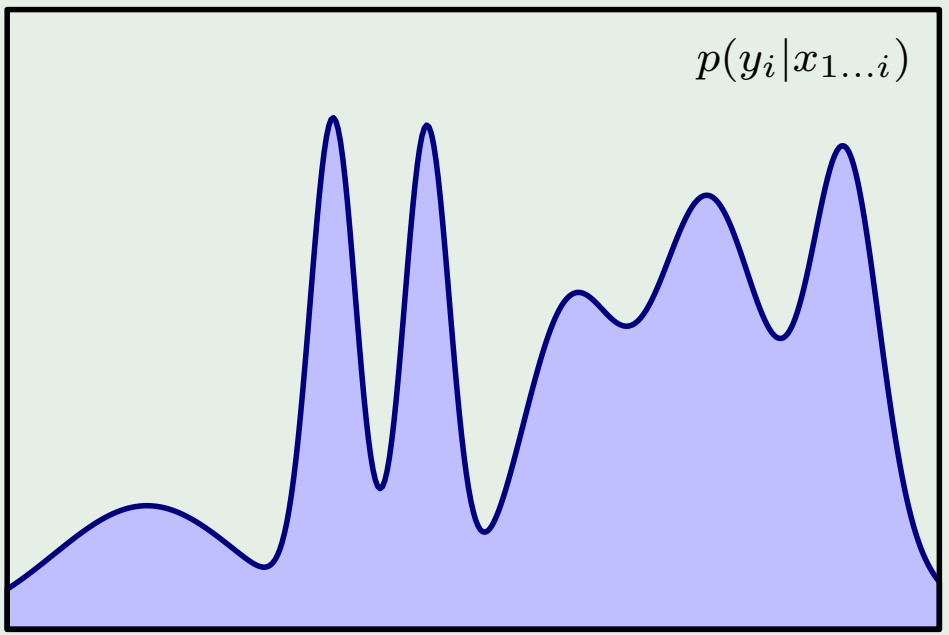
Example: Intractable distributions





Example: Intractable distributions





Sampling

We're generally not really interested in the posterior distribution for its own sake

- Often, we need the posterior distribution to compute the expectation of some function

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

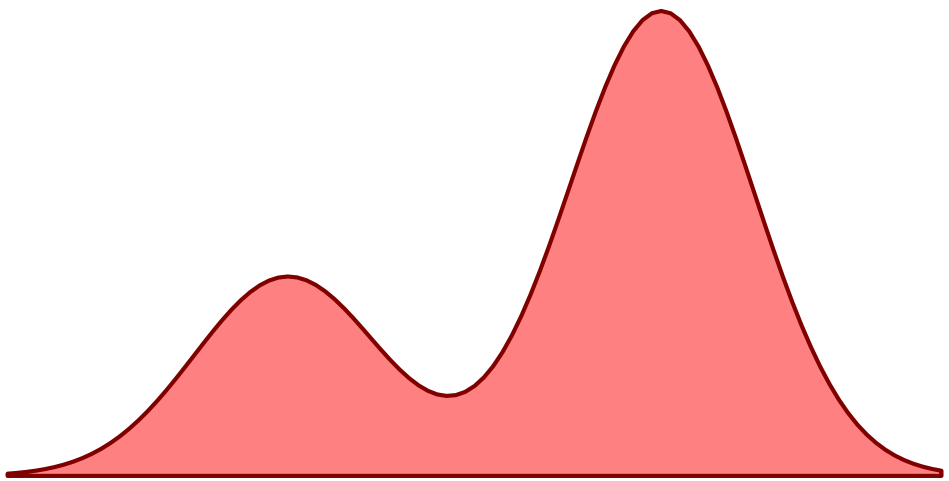
- Sometimes we want to marginalise out variables
- This is often impossible to do analytically

Integrating numerically

We could compute the solution numerically

- Computing the integral by summing approximations
  - Becomes intractable very fast as the dimensionality goes up
  - How about indefinite integrals (we cannot reach ∞)
- Use sampling: if we can draw independent samples $\{\mathbf{z}^{(1)} \dots \mathbf{z}^{(L)}\}$ of $p(\mathbf{z})$, we can approximate

$$\mathbb{E}[f] = \int f(\mathbf{z})\, p(\mathbf{z})\, \mathrm{d}\mathbf{z} \simeq \frac{1}{L} \sum_{l=1}^{L} f(\mathbf{z}^{(l)})$$

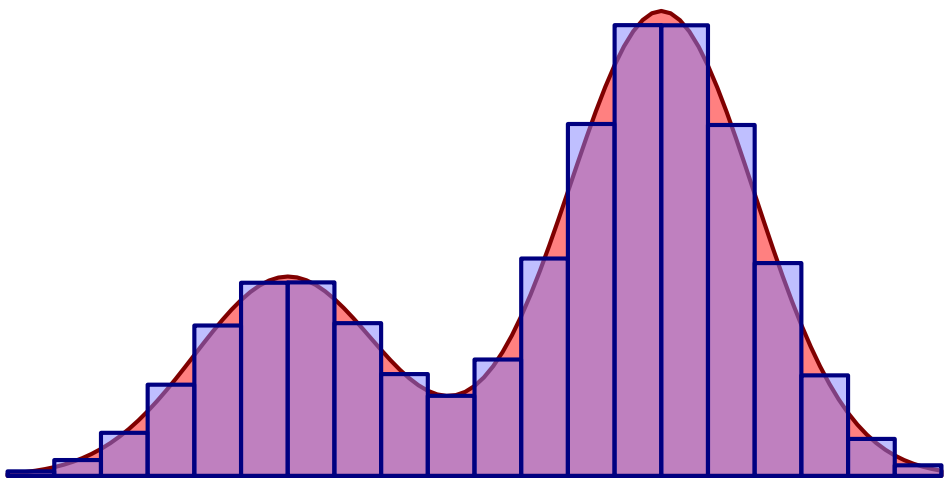


Integrating numerically

We could compute the solution numerically

- Computing the integral by summing approximations
  - Becomes intractable very fast as the dimensionality goes up
  - How about indefinite integrals (we cannot reach ∞)
- Use sampling: if we can draw independent samples $\{\mathbf{z}^{(1)} \dots \mathbf{z}^{(L)}\}$ of $p(\mathbf{z})$, we can approximate

$$\mathbb{E}[f] = \int f(\mathbf{z})\, p(\mathbf{z})\, \mathrm{d}\mathbf{z} \simeq \frac{1}{L} \sum_{l=1}^{L} f(\mathbf{z}^{(l)})$$

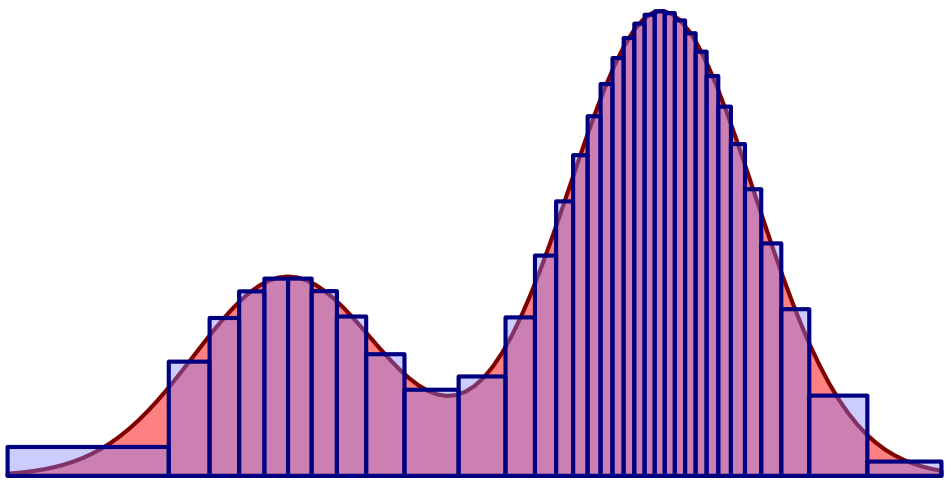


Integrating numerically

We could compute the solution numerically

- Computing the integral by summing approximations
  - Becomes intractable very fast as the dimensionality goes up
  - How about indefinite integrals (we cannot reach ∞)
- Use sampling: if we can draw independent samples $\{\mathbf{z}^{(1)} \dots \mathbf{z}^{(L)}\}$ of $p(\mathbf{z})$, we can approximate

$$\mathbb{E}[f] = \int f(\mathbf{z})\, p(\mathbf{z})\, \mathrm{d}\mathbf{z} \simeq \frac{1}{L} \sum_{l=1}^{L} f(\mathbf{z}^{(l)})$$



About Sampling

Sampling is therefore useful when:

- We cannot represent the joint distribution tractably but we can evaluate its building blocks
- We cannot compute the expectation of f analytically, but we can evaluate $f(\mathbf{x})$ for given $\mathbf{x}$.

Remember

Samples are easy to manipulate, even if their distribution is not

About Sampling

Things to notice:

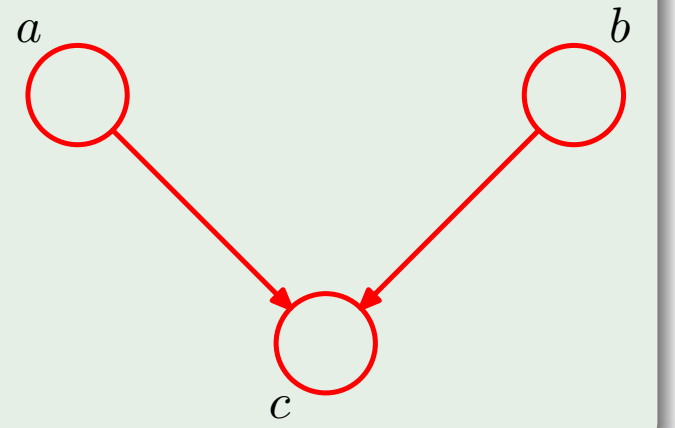
- The approximation is unbiased: $\mathbb{E}[\hat{f}] = \mathbb{E}[f]$, and $\text{var}[\hat{f}] = \text{var}_{p(\mathbf{z})}[f]$.
- The approximation is independent of the dimensionality of $\mathbf{z}$. In practice 10-20 independent samples generally suffice for a good approximation.
- If the function we want the expectation of is small where $p(\mathbf{x})$ is large and vice versa, a larger number of samples is needed for good accuracy.
- To get the marginal distribution $p(\mathbf{z})$: sample from the joint distribution $p(\mathbf{x}, \mathbf{z})$ and ignore the values of $\mathbf{x}$

Graphical Models

In the case of Bayesian Networks, we can use *ancestral sampling*:

- Start at root nodes, sample from those
- For each child node, sample from the conditional distribution

Example



However how do we deal with observed values?

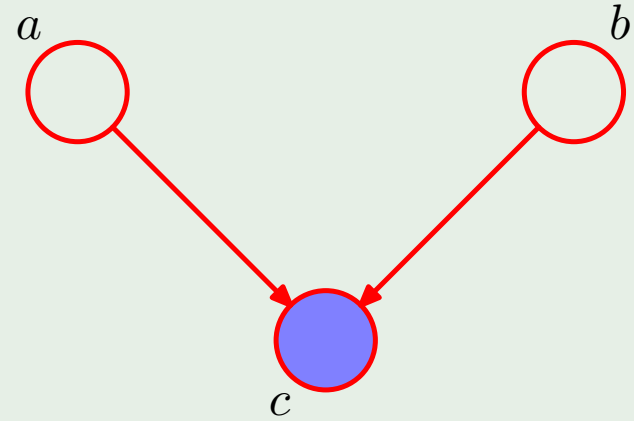
- Logic sampling: sample until you reach the observation
- If the sampled value agrees with the observation, keep it
- Otherwise: discard
- Very inefficient and rarely used.

Graphical Models

In the case of Bayesian Networks, we can use *ancestral sampling*:

- Start at root nodes, sample from those
- For each child node, sample from the conditional distribution

Example



However how do we deal with observed values?

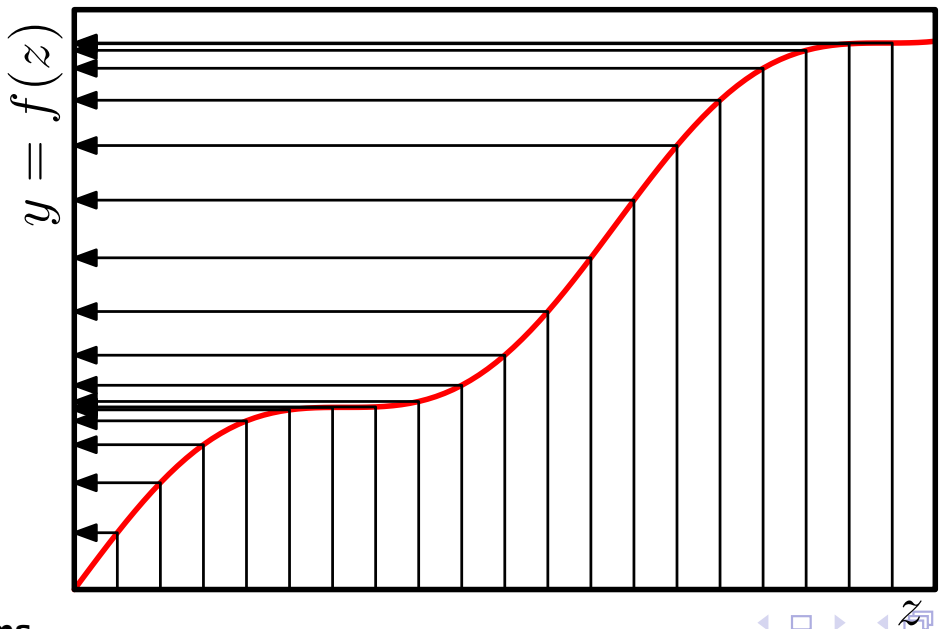
- Logic sampling: sample until you reach the observation
- If the sampled value agrees with the observation, keep it
- Otherwise: discard
- Very inefficient and rarely used.

Transforming samples

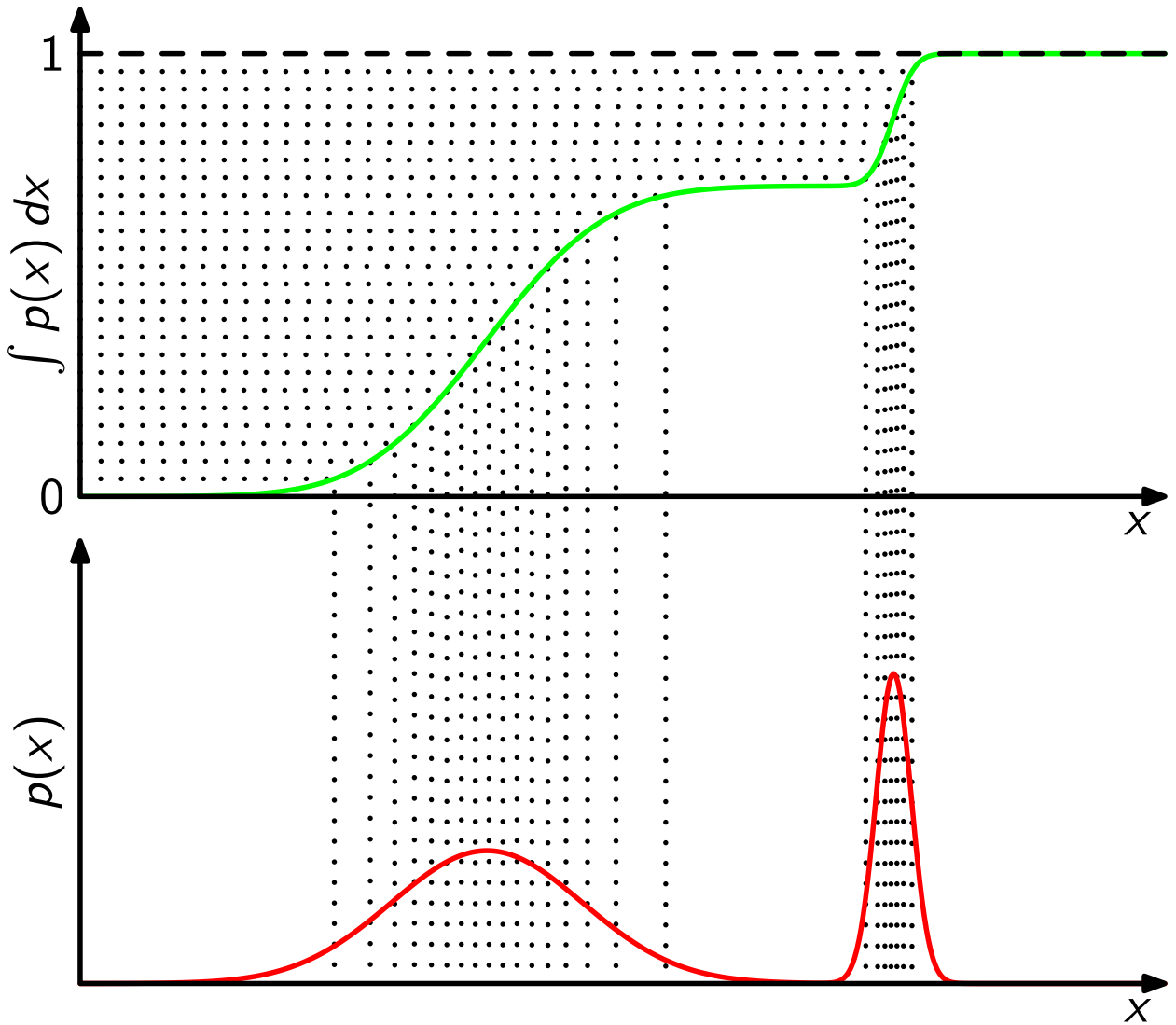
We assume we have a way of generating uniformly distributed pseudo-random numbers z in the range $(0, 1)$ ($p(z) = 1$).

- How can we obtain samples with a different, desired distribution?
- If $y = f(z)$, the distribution of y is given by

$$p(y) = p(z) \left| \frac{dz}{dy} \right|$$



Transforming samples



Transforming samples

The good:

- Fast and exact
- Can easily be extended to multivariate distributions
- Works for the (multivariate) Gaussian distribution
- Building block for more advanced sampling schemes

The bad:

- Only possible for limited, simple distributions

Rejection sampling

Suppose we know $\tilde{p}(\mathbf{z})$ and want to sample from

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

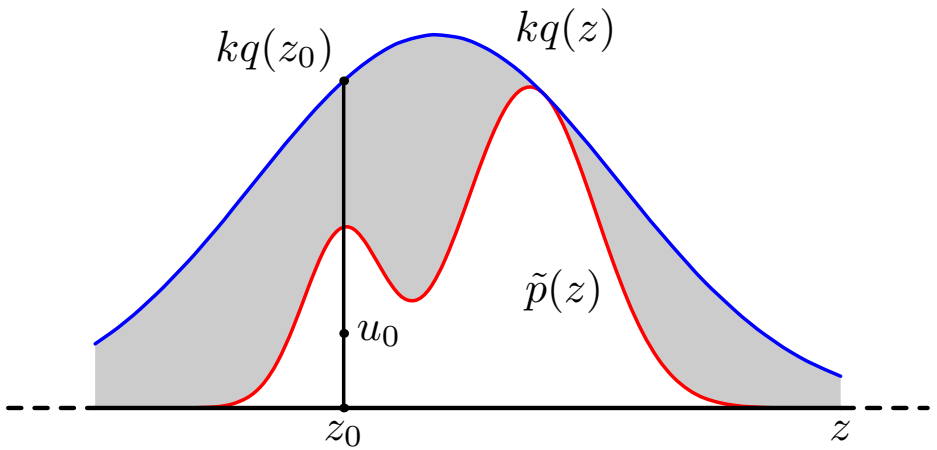
where $\tilde{p}(\mathbf{z})$ is easy to compute, but Z is unknown.

Example

In a Markov Random Fields, the probability densities are given by the potential functions associated with a clique

$$p(\mathbf{x}) = \frac{1}{Z}\prod_{C}\psi_C(\mathbf{x}_c),$$

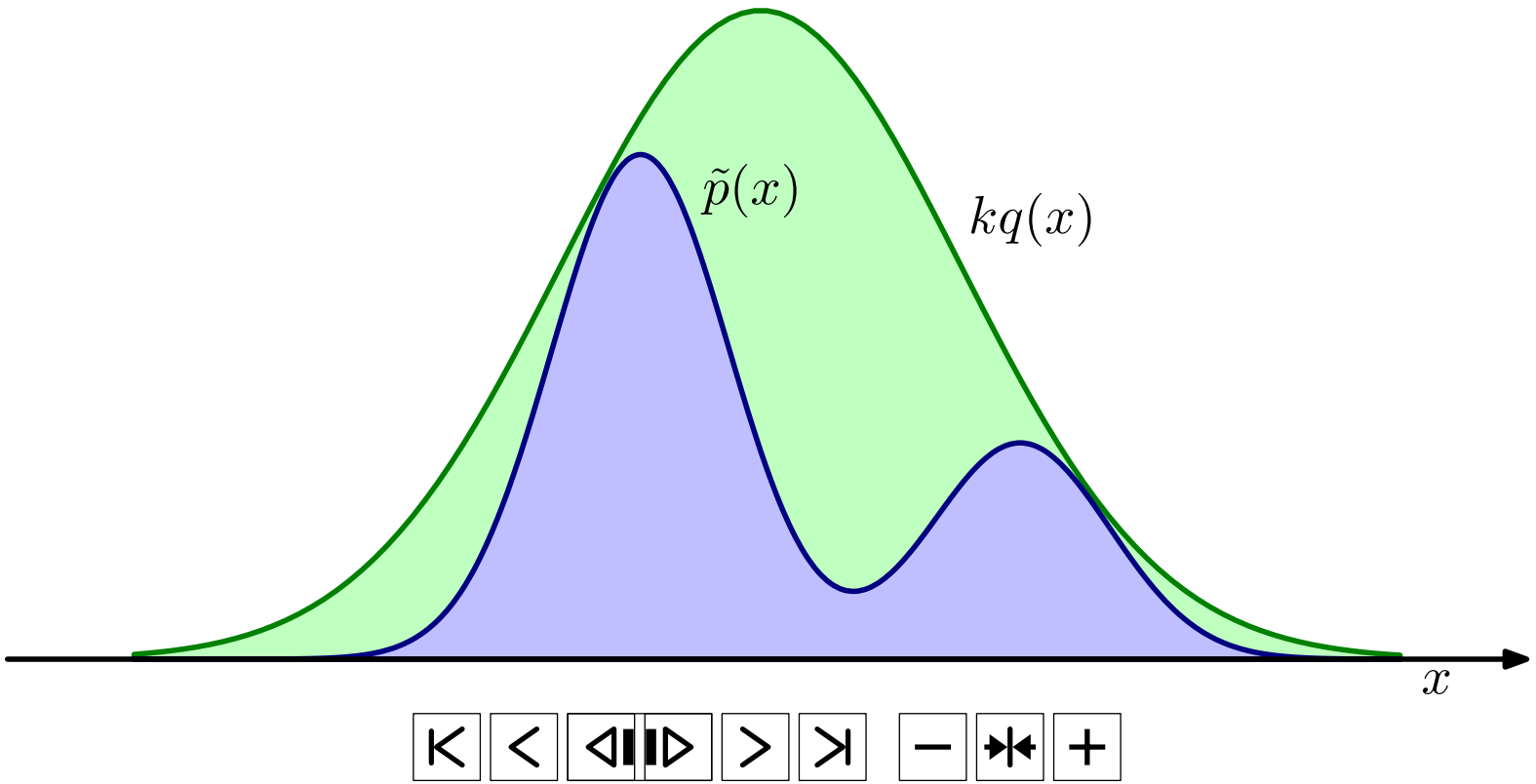
where Z is hard to compute.



Choose a proposal distribution $q(z)$ and find a constant k such that $kq(z) \geqslant \tilde{p}(z)$ everywhere

- Sample a value z_0 from $q(z$
- Sample a value u uniformly from $[0, kq(z_0)]$
- Reject the pair if $u > \tilde{p}(z_0)$
- The resulting values are uniformly distributed in the area under $\tilde{p}(z)$, so that the corresponding values z are distributed according to $p(z)$.

Rejection sampling



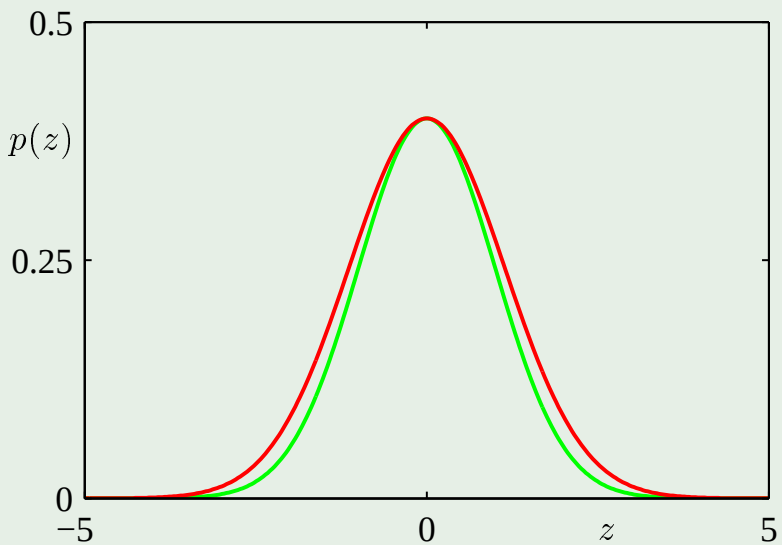
Importance of the proposal distribution

Samples from $q(x)$ are accepted with probability $\tilde{p}(z)/kq(z)$, so that the probability of acceptance is

$$p(\text{accept}) = \int \frac{\tilde{p}(z)}{kq(z)} q(z)\,\mathrm{d}z = \frac{1}{k}\int \tilde{p}(z)\,\mathrm{d}z$$

It is therefore important to keep k as small as possible.

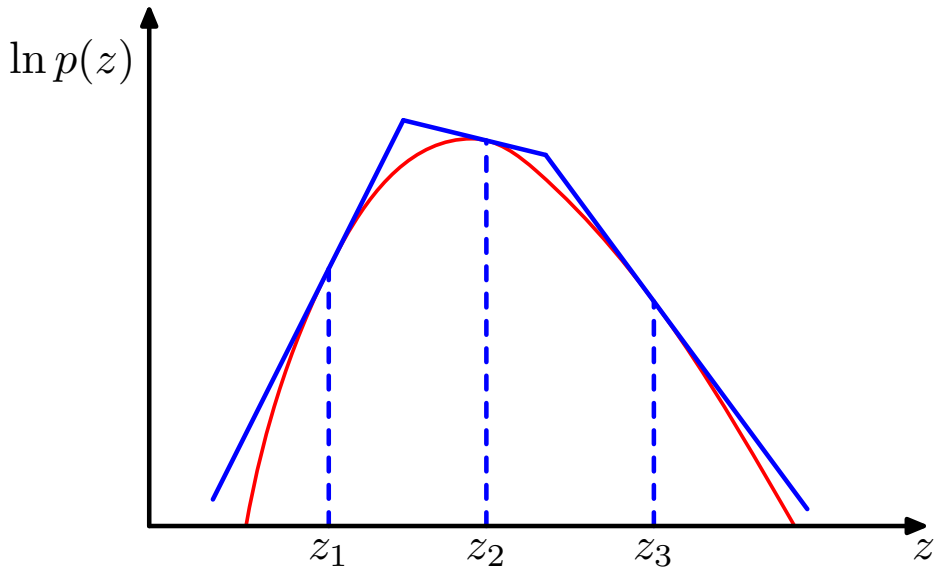
Example: $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \sigma_p^2\mathbf{I})$, $q(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \sigma_q^2\mathbf{I})$



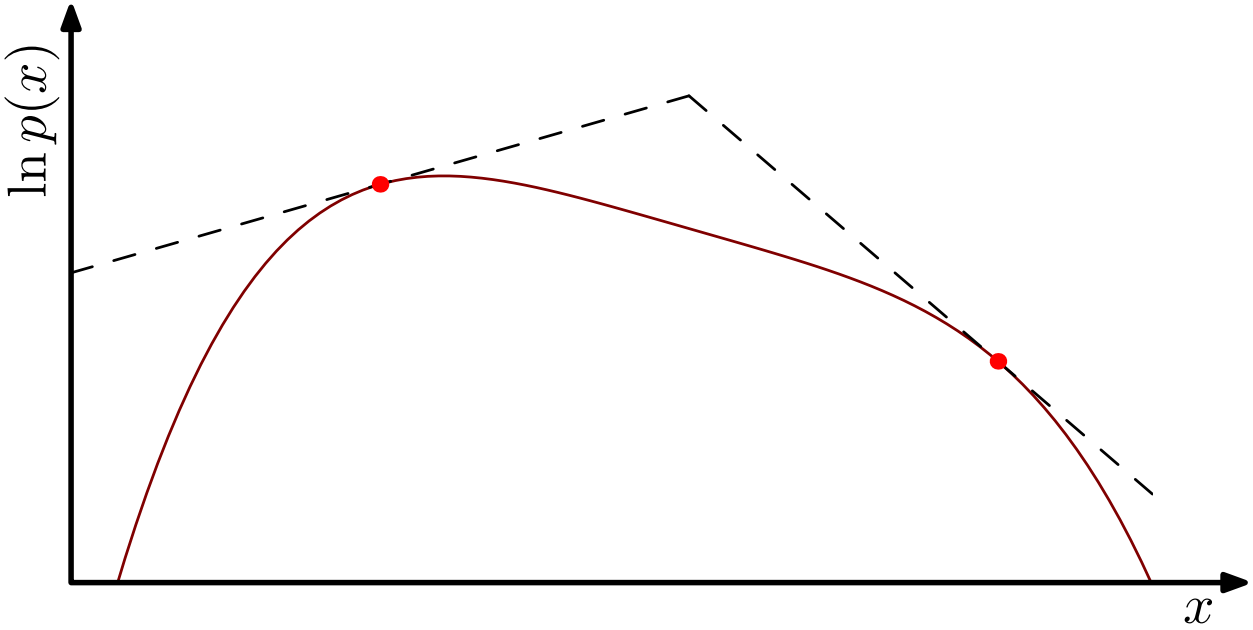
- Rejection rate: $1/k$
- The optimal $k = (\sigma_q/\sigma_p)^D$ in D dimensions
- If $D = 1000$ and $\sigma_q = 1.01\sigma_p$, 1 sample in 20000 is accepted.

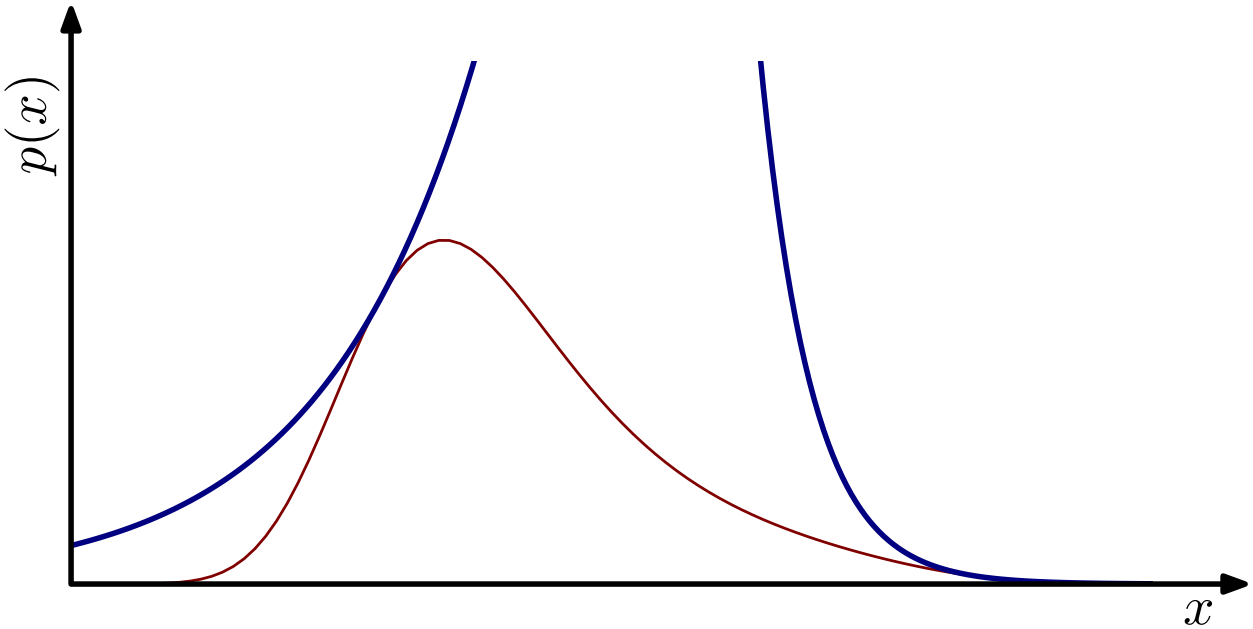
Adaptive Rejection Sampling

In practice it is often hard to find a good proposal distribution. However when $\tilde{p}(z)$ is log-concave, we can adapt it on the fly:



- Find the tangent to $\tilde{p}(z)$ at an initial grid of points.
- Sample from the piecewise-linear distribution (this is easy, by transforming uniform samples)
- If the sample is rejected, add the current point to the grid, update $q(z)$.


ln p(x)
x


p(x)
x

Importance sampling

In order to obtain a good estimate of an expectation, it is not necessary to be able to sample from the distribution

- For example: if we can evaluate $p(\mathbf{z})$ easily, we could discretise the space uniformly and approximate the integrand as

$$\mathbb{E}[f] \simeq \sum_{l=1}^{L} p(\mathbf{z}^{(l)}) f(\mathbf{z}^{(l)})$$

- Problem: Very inaccurate/inefficient, esp. in high dimensional spaces
- Better: Sample from proposal distribution $q(\mathbf{z})$ and reweigh:

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

Importance Sampling

Often we can only evaluate $p(\mathbf{z})$ up to a normalisation constant. Similarly, it is interesting to be able to sample from $q(\mathbf{z}) = \frac{1}{Z}\tilde{q}(\mathbf{z})$. Then we can define

$$\tilde{r}_l = \frac{\tilde{p}(\mathbf{z}^{(l)})}{\tilde{q}(\mathbf{z}^{(l)})} \text{ and } w_l = \frac{\tilde{r}_l}{\sum_{m=1}^{L} \tilde{r}_m}$$

and compute the expectation as follows

$$\mathbb{E}[f] \simeq \sum_{l=1}^{L} \mathbf{w}_l f(\mathbf{z}^{(l)})$$

No samples are thrown away.

Importance Sampling

pros:

- No need for a scaling constant
- Super-efficient sampling: if $q(\mathbf{z})$ is similar to $p(\mathbf{z})f(z)$, we require fewer samples for a given accuracy than if $q(\mathbf{z})$ is similar to $p(\mathbf{z})$

cons:

- if $p(\mathbf{z})f(\mathbf{z})$ is strongly varying, few samples will carry most of the weight
- if $q(\mathbf{z})$ is small where $p(\mathbf{z})f(\mathbf{z})$ is large, it is possible for no samples to lie in those regions: the samples may then have low variance while the estimates are severely wrong.

EM Algorithm

Sampling can of course be used for the E-step of the EM algorithm. Recall that we optimise the expectation of the complete log-likelihood:

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \int p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{Z}, \mathbf{X}|\boldsymbol{\theta}) \, \mathrm{d}\mathbf{Z}$$

If we draw from $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$, we get Monte Carlo EM

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) \simeq \frac{1}{L} \sum_{l=1}^{L} \ln p(\mathbf{Z}, \mathbf{X}|\boldsymbol{\theta}) \, \mathrm{d}\mathbf{Z}$$

which we can optimise with the usual M-step.

Markov Chain Monte Carlo (MCMC)

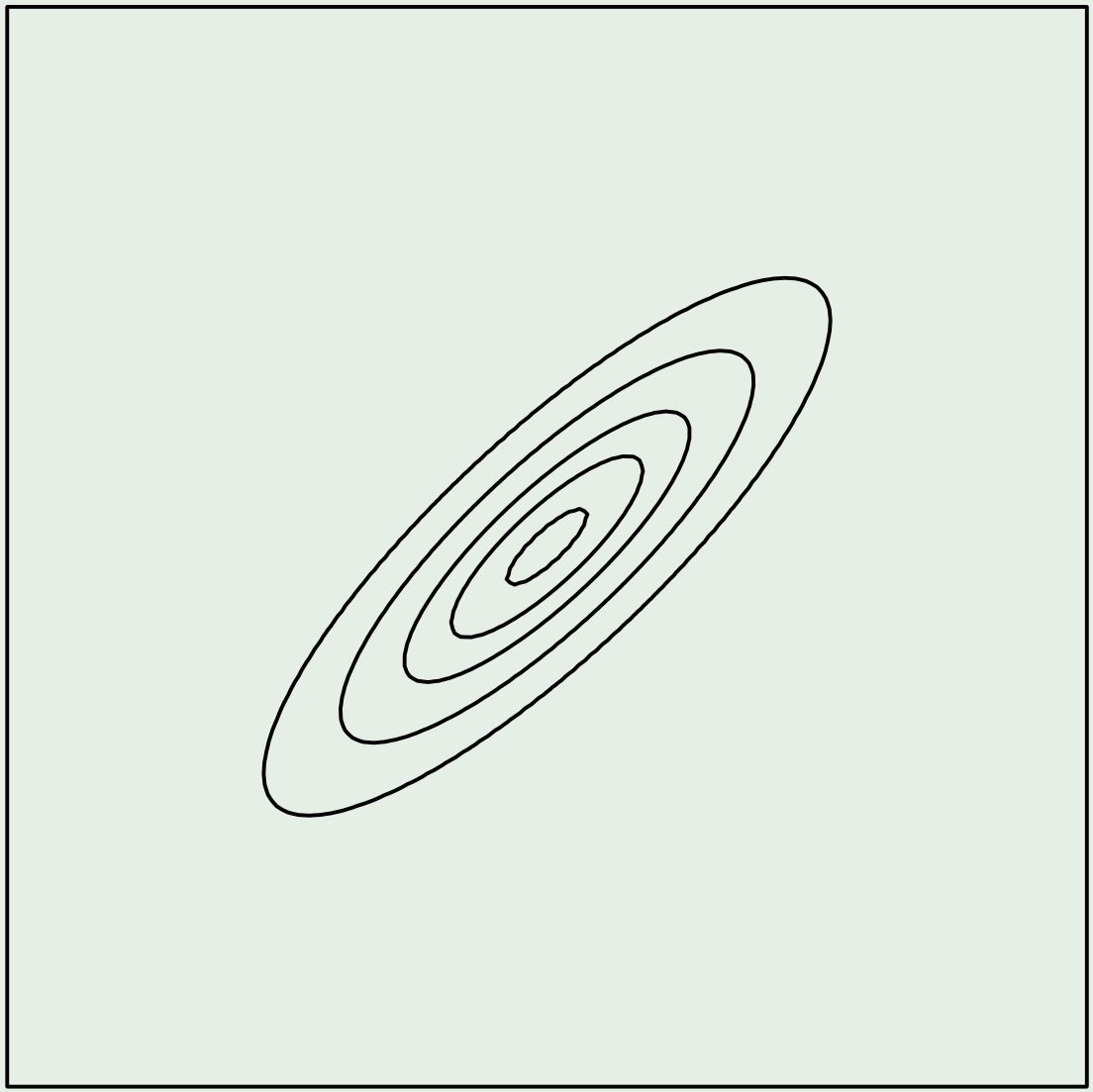
The techniques we saw until now suffer severe limitations in high-dimensional spaces. The Metropolis algorithm circumvents these by using a conditional proposal distribution:

- Sample candidate sample $\mathbf{z}^*$ from a symmetric proposal distribution $q(\mathbf{z}^*|\mathbf{z}^{(\tau)})$
- Accept the candidate sample with probability

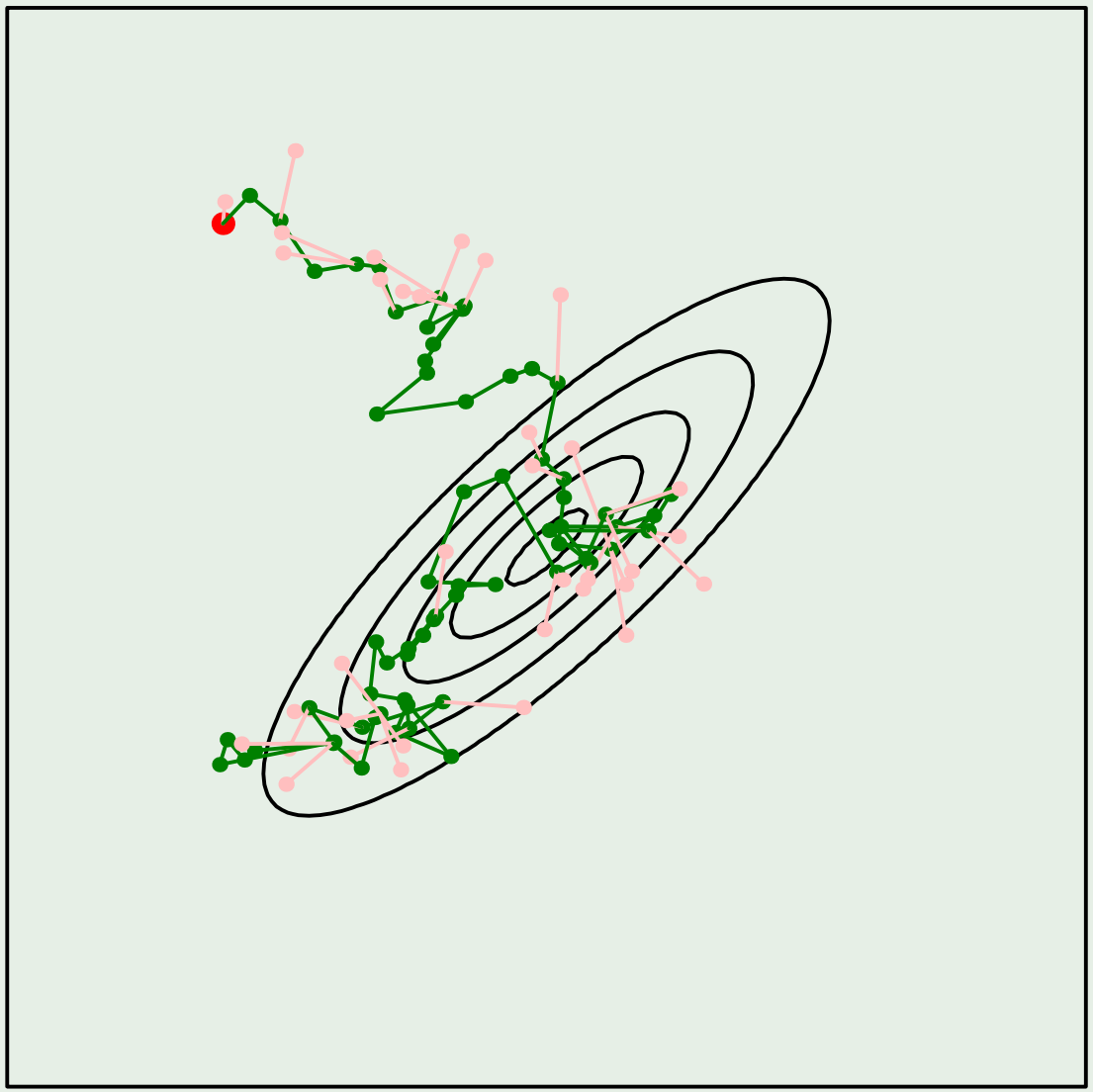
$$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})}\right)$$

- If $\mathbf{z}^*$ is rejected, $\mathbf{z}^{(\tau+1)} = \mathbf{z}^{(\tau)}$.
- This tends to $p(\mathbf{z})$ as $\tau \to \infty$, if $q(\mathbf{z}_A|\mathbf{z}_B) > 0$ for all $\mathbf{z}_A$, $\mathbf{z}_B$.
- The samples are not independent. Keep every M^{th} sample for "independent" samples

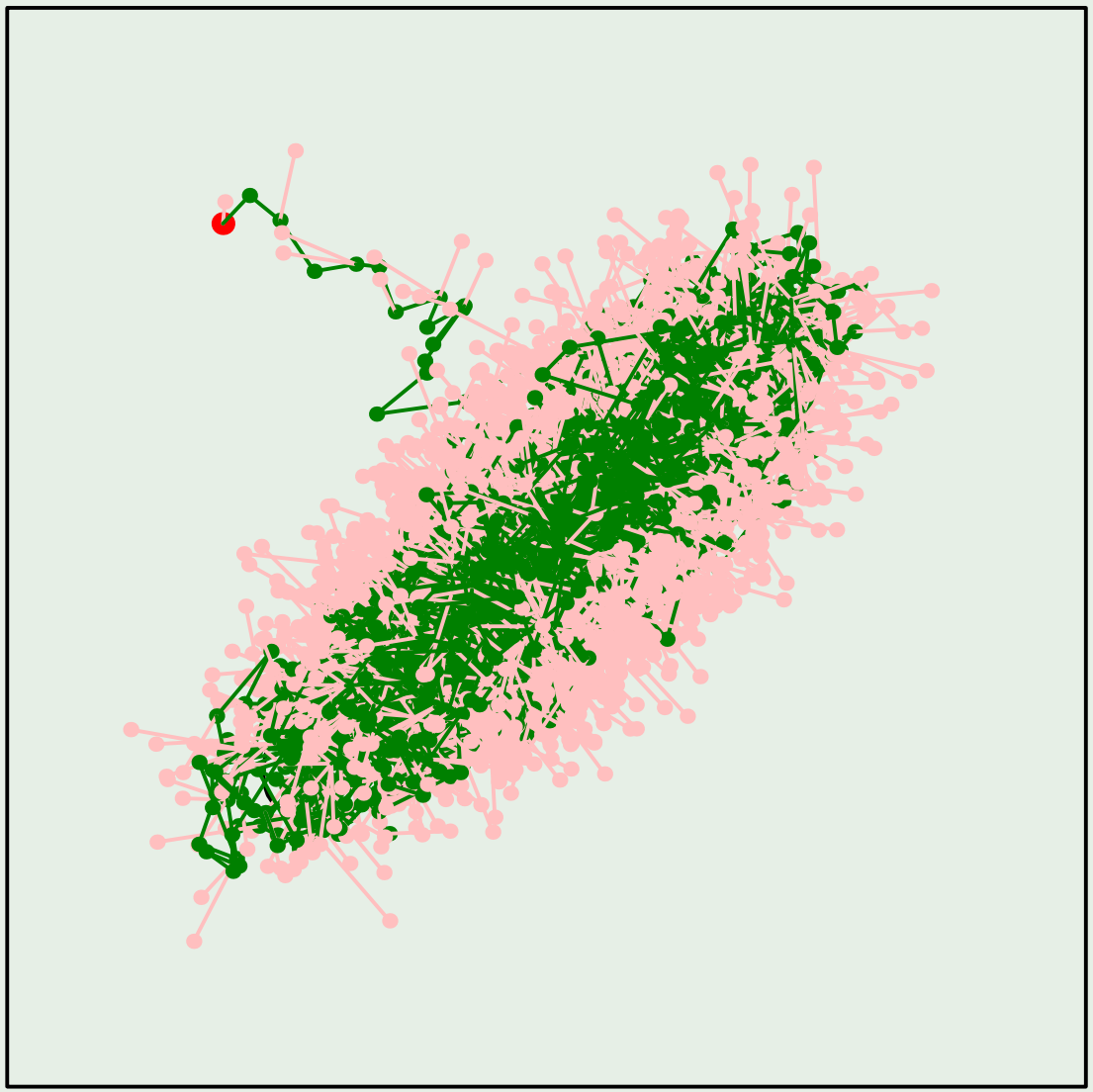
Example: Metropolis sampling of Gaussian



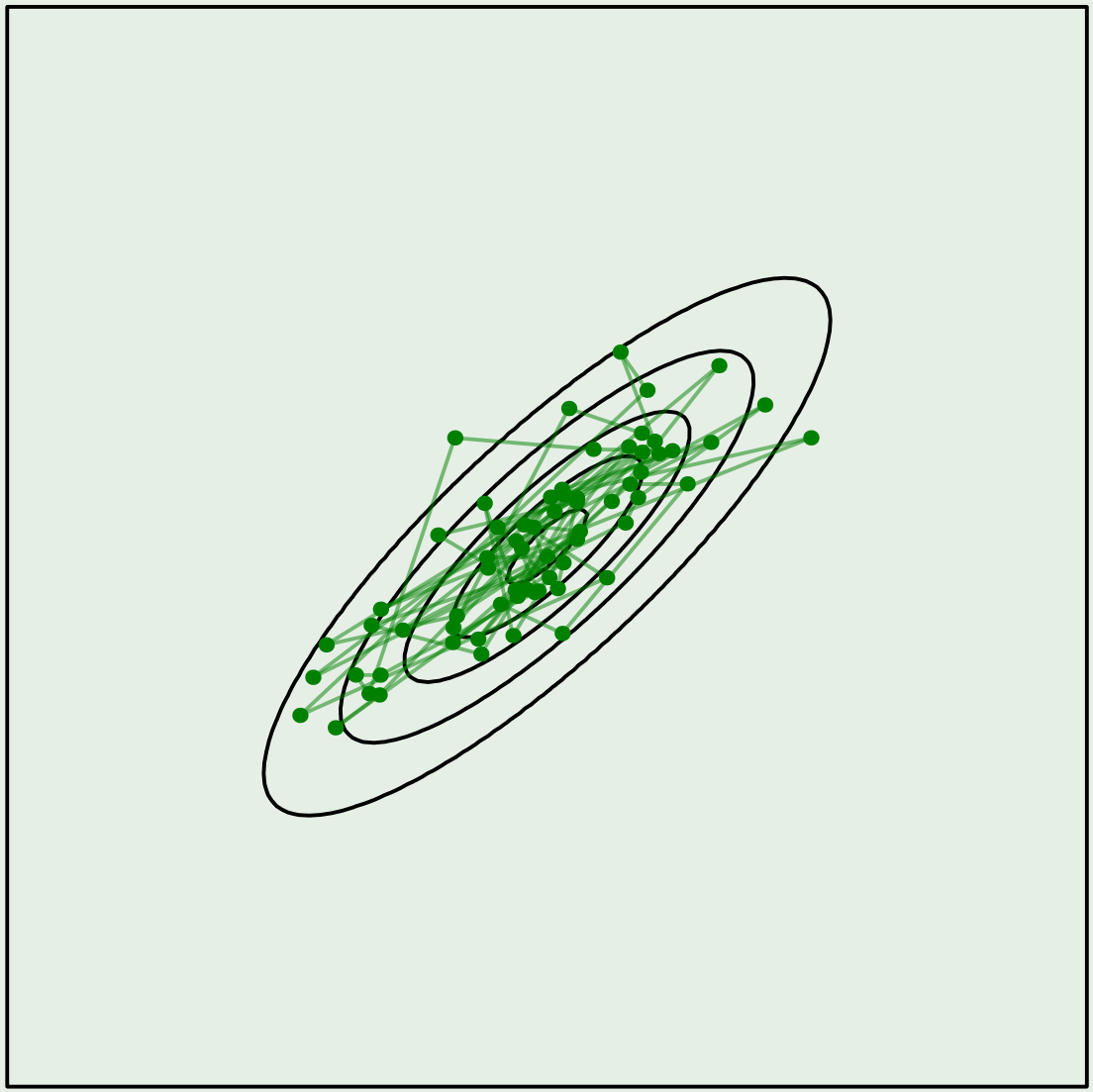
Example: Metropolis sampling of Gaussian



Example: Metropolis sampling of Gaussian



Example: Metropolis sampling of Gaussian



Convergence of the Markov Chain

We want the chain to converge to a stationary distribution $p^*(\mathbf{z})$. The distribution $p^*(\mathbf{z})$ is stationary with respect to the chain if

$$p^*(\mathbf{z}) = \sum_{\mathbf{z}'} p^*(\mathbf{z}')\, T(\mathbf{z}', \mathbf{z})$$

A sufficient condition for this is that the chain exhibits *detailed balance*:

$$p^*(\mathbf{z})\, T(\mathbf{z}, \mathbf{z}') = p^*(\mathbf{z}')\, T(\mathbf{z}', \mathbf{z})$$

in which case the chain is said to be *reversible*

Metropolis-Hastings

If the proposal distribution is not symmetric, we can apply the Metropolis-Hastings algorithm. Accept a candidate sample $\mathbf{z}^*$ with probability

$$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*)q(\mathbf{z}^{(\tau)}, \mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})q(\mathbf{z}^*, \mathbf{z}^{(\tau)})}\right),$$

which exhibits detailed balance with respect to $p(\mathbf{z})$.

Proof: Detailed Balance

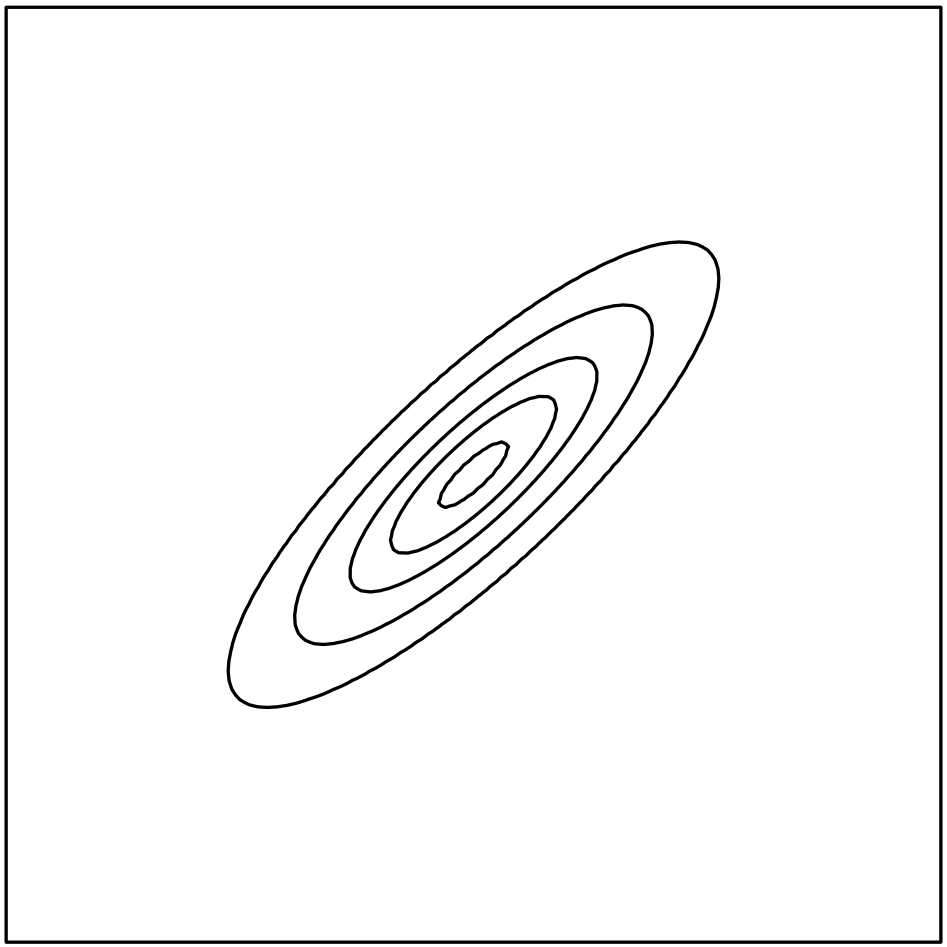
$$
\begin{aligned}
p(\mathbf{z})q(\mathbf{z}|\mathbf{z}')A(\mathbf{z}',\mathbf{z}) &= p(\mathbf{z})q(\mathbf{z}|\mathbf{z}')\min(1, \frac{p(\mathbf{z}')q(\mathbf{z}'|\mathbf{z})}{p(\mathbf{z})q(\mathbf{z}|\mathbf{z}')}) \\
&= \min(p(\mathbf{z})q(\mathbf{z}|\mathbf{z}'), p(\mathbf{z}')q(\mathbf{z}'|\mathbf{z})) \\
&= \min(p(\mathbf{z}')q(\mathbf{z}'|\mathbf{z}), p(\mathbf{z})q(\mathbf{z}|\mathbf{z}')) \\
&= p(\mathbf{z}')q(\mathbf{z}'|\mathbf{z})A(\mathbf{z},\mathbf{z}')
\end{aligned}
$$

The proposal distribution therefore only affects the efficiency of the sampling, not their end distribution.

Length scale

Markov chain methods trade off *exploration* vs. *rejection rate*.

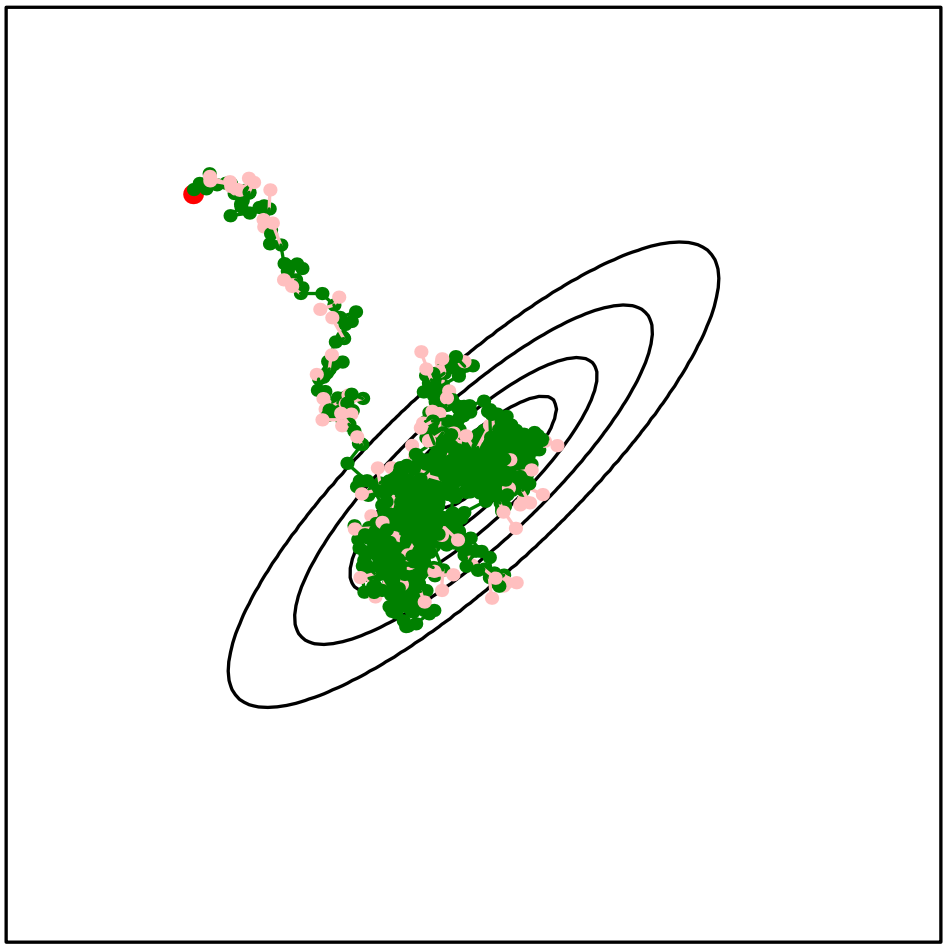
- For low rejection rate, proposal distribution scale $\rho \approx \sigma_{\text{min}}$
- Other directions explored by random walk
- Approx. independent samples: $(\sigma_{\text{max}}/\sigma_{\text{min}})^2$ steps
- Very different lengths scales $\Rightarrow$ slow to converge



Length scale

Markov chain methods trade off *exploration* vs. *rejection rate*.

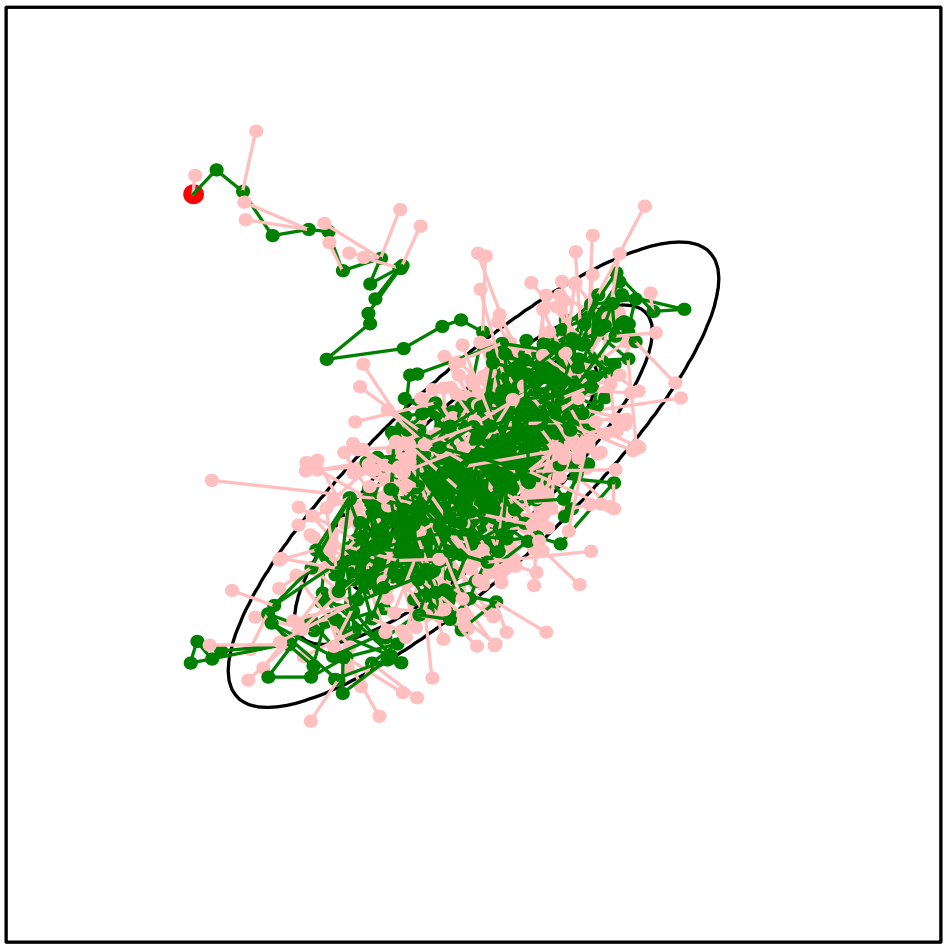
- For low rejection rate, proposal distribution scale $\rho \approx \sigma_{\text{min}}$
- Other directions explored by random walk
- Approx. independent samples: $(\sigma_{\text{max}}/\sigma_{\text{min}})^2$ steps
- Very different lengths scales $\Rightarrow$ slow to converge



Length scale

Markov chain methods trade off *exploration* vs. *rejection rate*.

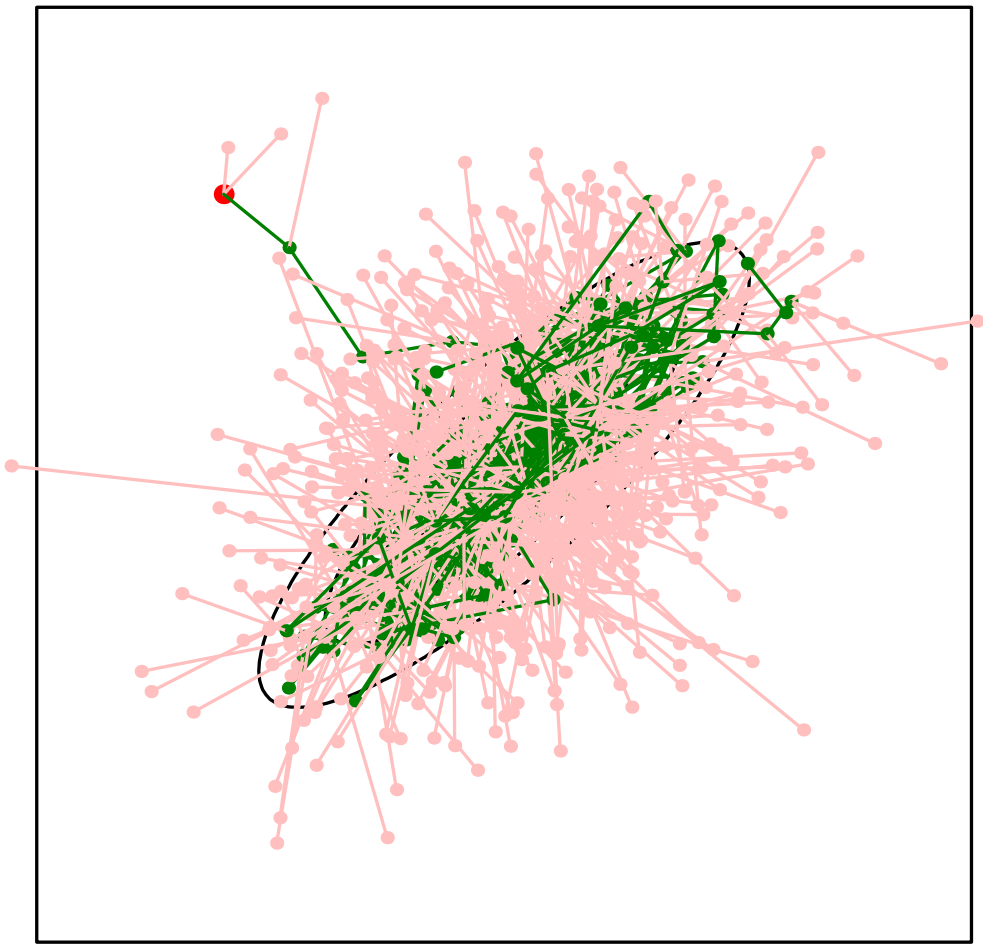
- For low rejection rate, proposal distribution scale $\rho \approx \sigma_{\text{min}}$
- Other directions explored by random walk
- Approx. independent samples: $(\sigma_{\text{max}}/\sigma_{\text{min}})^2$ steps
- Very different lengths scales $\Rightarrow$ slow to converge



Length scale

Markov chain methods trade off *exploration* vs. *rejection rate*.

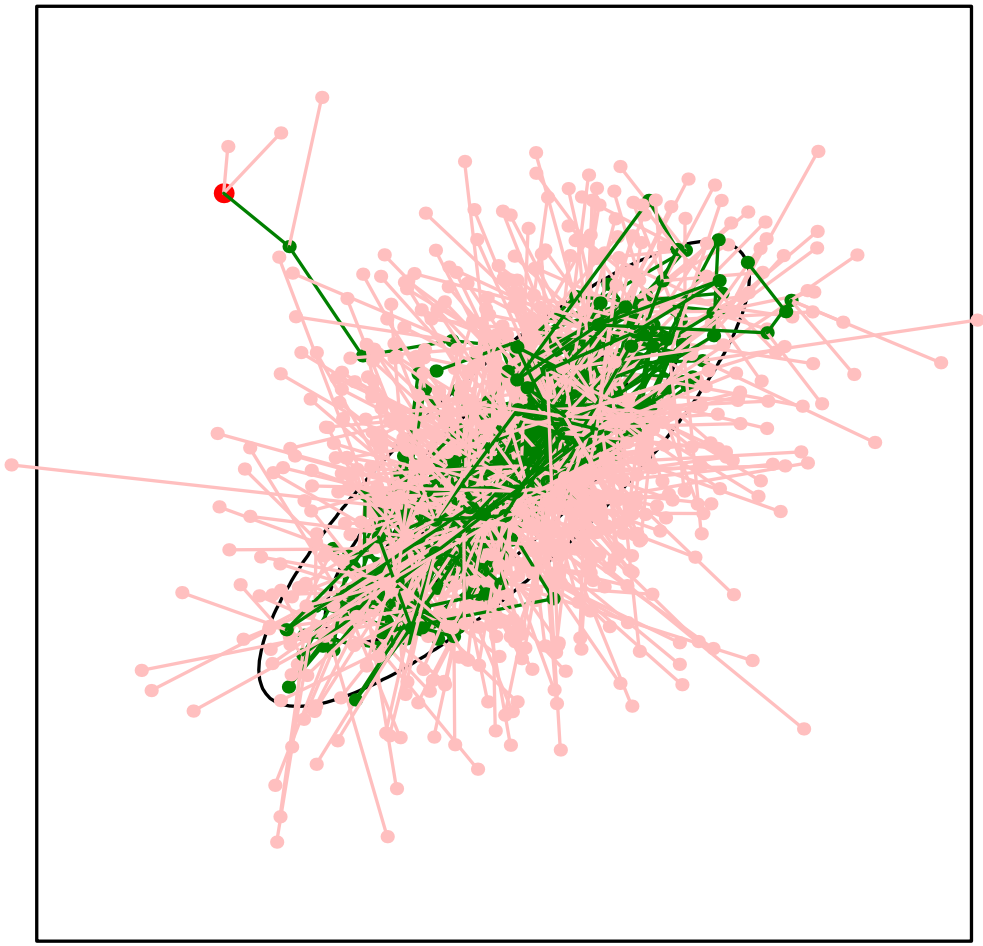
- For low rejection rate, proposal distribution scale $\rho \approx \sigma_{\text{min}}$
- Other directions explored by random walk
- Approx. independent samples: $(\sigma_{\text{max}}/\sigma_{\text{min}})^2$ steps
- Very different lengths scales $\Rightarrow$ slow to converge



Length scale

Markov chain methods trade off *exploration* vs. *rejection rate*.

- For low rejection rate, proposal distribution scale $\rho \approx \sigma_{\text{min}}$
- Other directions explored by random walk
- Approx. independent samples: $(\sigma_{\text{max}}/\sigma_{\text{min}})^2$ steps
- Very different lengths scales $\Rightarrow$ slow to converge



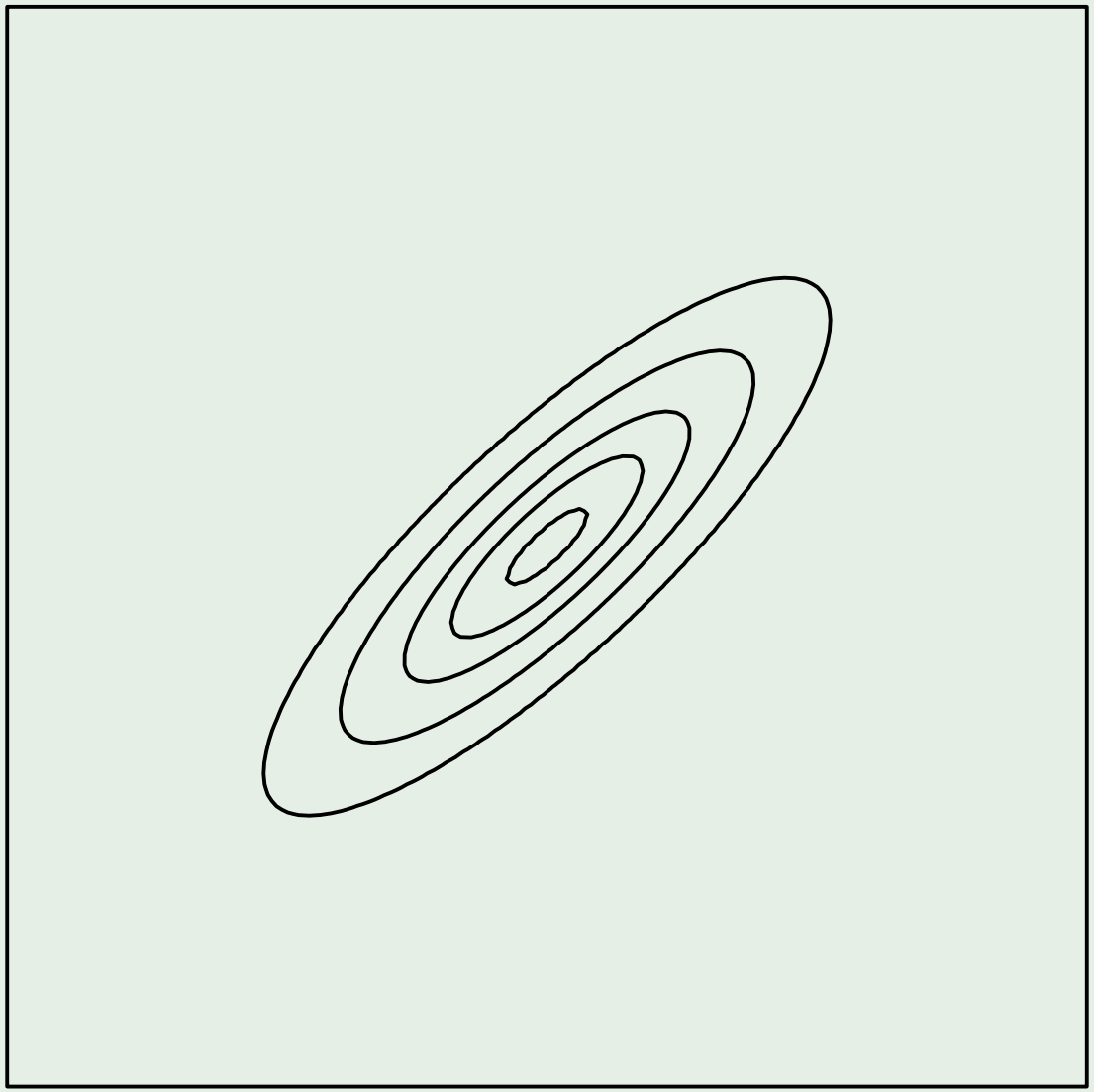
Gibbs Sampling

Gibbs sampling is a special case of Metropolis-Hastings sampling, where we sample $p(\mathbf{z}) = p(z_1 \dots z_D)$ by sampling each z_i in turn.

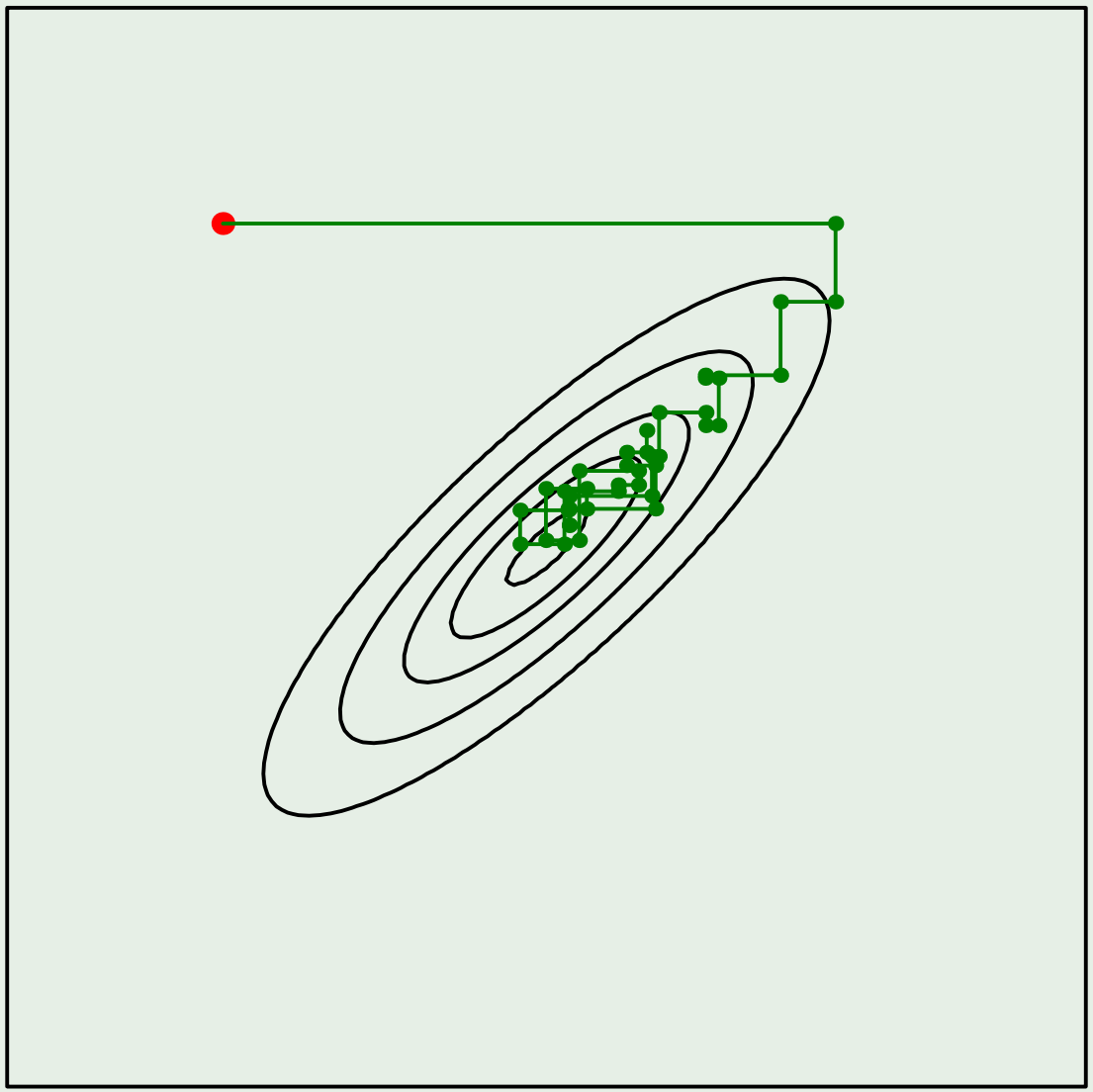
- The proposal distribution is $p(z_i|\mathbf{z}_{\setminus i})$
- All samples are accepted: $q(\mathbf{z}^*|\mathbf{z}) = p(z_i|\mathbf{z}_{\setminus i})$, $\mathbf{z}^*_{\setminus i} = \mathbf{z}_{\setminus i}$ and $p(\mathbf{z}) = p(z_i|\mathbf{z}_{\setminus i})p(\mathbf{z}_{\setminus i})$, so that

$$A(\mathbf{z}^*, \mathbf{z}) = \frac{p(\mathbf{z}^*)q(\mathbf{z}|\mathbf{z}^*)}{p(\mathbf{z})q(\mathbf{z}^*|\mathbf{z})} = \frac{p(z_i^*|\mathbf{z}^*_{\setminus i})p(\mathbf{z}^*_{\setminus i})p(z_i|\mathbf{z}^*_{\setminus i})}{p(z_i|\mathbf{z}_{\setminus i})p(\mathbf{z}_{\setminus i})p(z_i^*|\mathbf{z}_{\setminus i})} = 1$$

Example: Gibbs sampling of a Gaussian

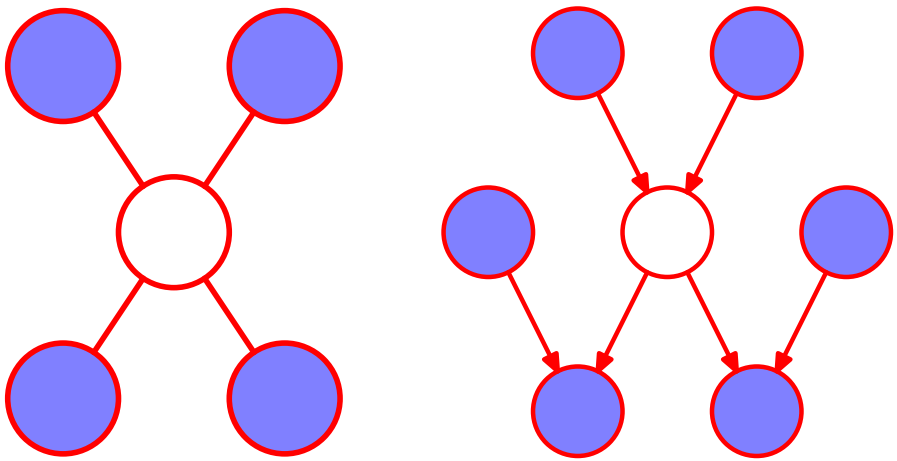


Example: Gibbs sampling of a Gaussian



Practical usability of Gibbs sampling

The practical applicability depends on the ease with which we can sample from $p(z_i|\mathbf{z}_{\setminus i})$. For graphical models, the conditional distribution of a node depends on its Markov Blanket:



For directed graphs, the conditional distributions for Gibbs sampling is often complex but log-concave. Adaptive rejection sampling is therefore widely applicable in this setting.

Particle Filter

One example where MCMC methods perform poorly is for sequential data, such as extensions of the linear dynamical system

- The states are by definition strongly correlated: sampling algorithms converge slowly
- Instead of sampling from the joint distribution, the particle filter will do a forward pass through the data
- Use the process noise as proposal distribution (which is exact)
- Update the weights of the particles according to the probability of the observation given the sample
- Resample, to keep more particles in the high-density areas

Summary

Today we've seen sampling:

- Why do we do this (Bishop, p. 523-525)
- Basic sampling algorithms (Bishop, p. 528-534)
- Markov Chain Monte Carlo (Bishop, p. 537-546)