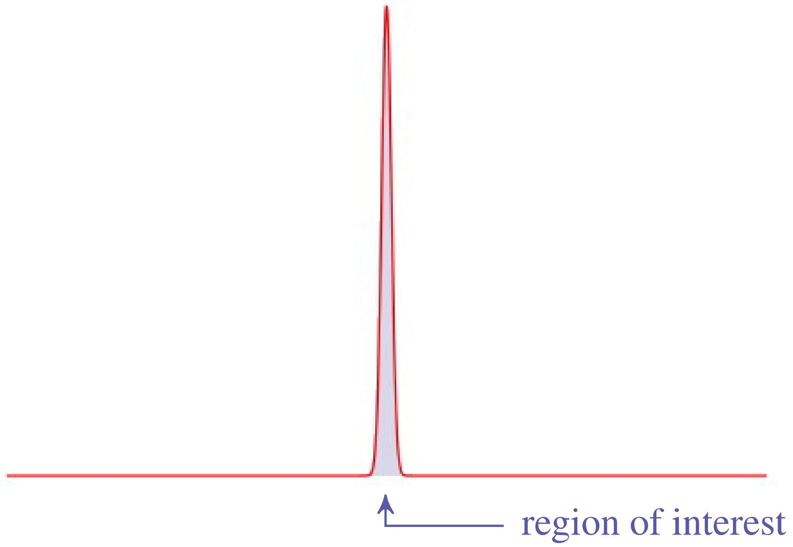
Markov Chain Monte Carlo

MOTIVATION

Suppose we rejection-sample a distribution like this:



Once we have drawn a sample in the narrow region of interest, we would like to continue drawing samples within the same region. That is only possible if each sample *depends on the location of the previous sample*.

Proposals in rejection sampling are i.i.d. Hence, once we have found the region where p concentrates, we forget about it for the next sample.

MCMC: Idea

Recall: Markov chain

- A sufficiently nice Markov chain (MC) has an invariant distribution P_{inv}.
- Once the MC has converged to P_{inv}, each sample x_i from the chain has marginal distribution P_{inv}.

Markov chain Monte Carlo

We want to sample from a distribution with density p. Suppose we can define a MC with invariant distribution $P_{\text{inv}} \equiv p$. If we sample $x_1, x_2, \dots$ from the chain, then once it has converged, we obtain samples

$$x_i \sim p \ .$$

This sampling technique is called **Markov chain Monte Carlo** (**MCMC**).

Note: For a Markov chain, x_{i+1} can depend on x_i, so at least in principle, it is possible for an MCMC sampler to "remember" the previous step and remain in a high-probability location.

Continuous Markov Chain

The Markov chains we discussed so far had a finite state space **X**. For MCMC, state space now has to be the domain of p, so we often need to work with continuous state spaces.

Continuous Markov chain

A continuous Markov chain is defined by an initial distribution P_{init} and conditional probability $t(y|x)$, the **transition probability** or **transition kernel**.

In the discrete case, $t(y = i|x = j)$ is the entry $\mathbf{p}_{ij}$ of the transition matrix $\mathbf{p}$.

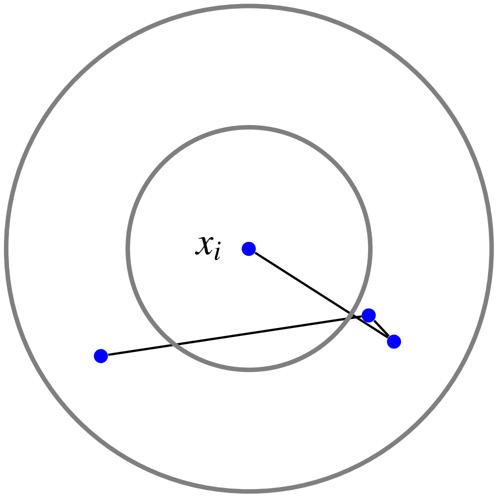
Example: A Markov chain on $\mathbb{R}^2$

We can define a very simple Markov chain by sampling

$$x_{i+1} \sim g(\,.\,|x_i, \sigma^2)$$

where $g(x|\mu, \sigma^2)$ is a spherical Gaussian with fixed variance. In other words, the transition distribution is

$$t(x_{i+1}|x_i) := g(x_{i+1}|x_i, \sigma^2)\ .$$



A Gaussian (gray contours) is placed around the current point x_i to sample x_{i+1}.

INVARIANT DISTRIBUTION

Recall: Finite case

- The invariant distribution P_{inv} is a distribution on the finite state space $\mathbf{X}$ of the MC (i.e. a vector of length $|\mathbf{X}|$).
- "Invariant" means that, if x_i is distributed according to P_{inv}, and we execute a step $x_{i+1} \sim t(\,.\,|x_i)$ of the chain, then x_{i+1} again has distribution P_{inv}.
- In terms of the transition matrix $\mathbf{p}$:

$$\mathbf{p} \cdot P_{\text{inv}} = P_{\text{inv}}$$

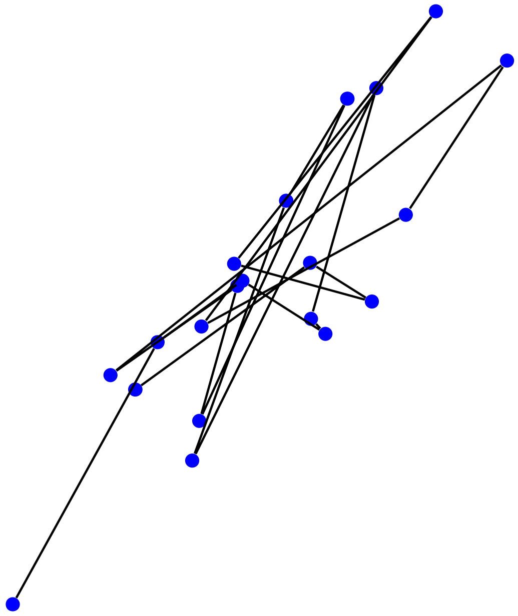
Continuous case

- $\mathbf{X}$ is now uncountable (e.g. $\mathbf{X} = \mathbb{R}^d$).
- The transition matrix $\mathbf{p}$ is substituted by the conditional probability t.
- A distribution P_{inv} with density p_{inv} is invariant if

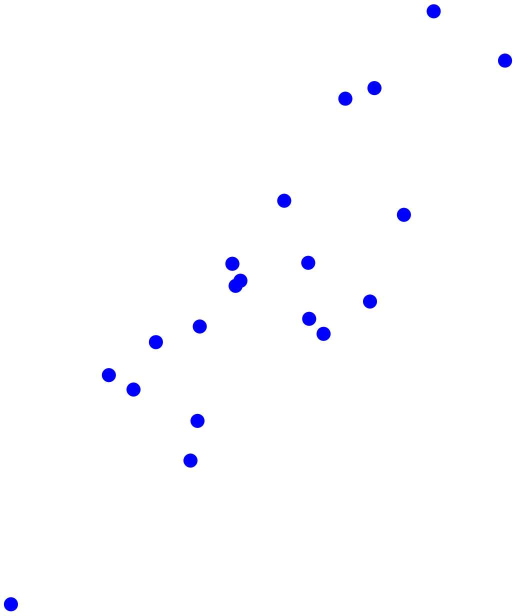
$$\int_{\mathbf{X}} t(y|x) p_{\text{inv}}(x) dx = p_{\text{inv}}(y)$$

This is simply the continuous analogue of the equation $\sum_i \mathbf{p}_{ij} (P_{\text{inv}})_i = (P_{\text{inv}})_j$.

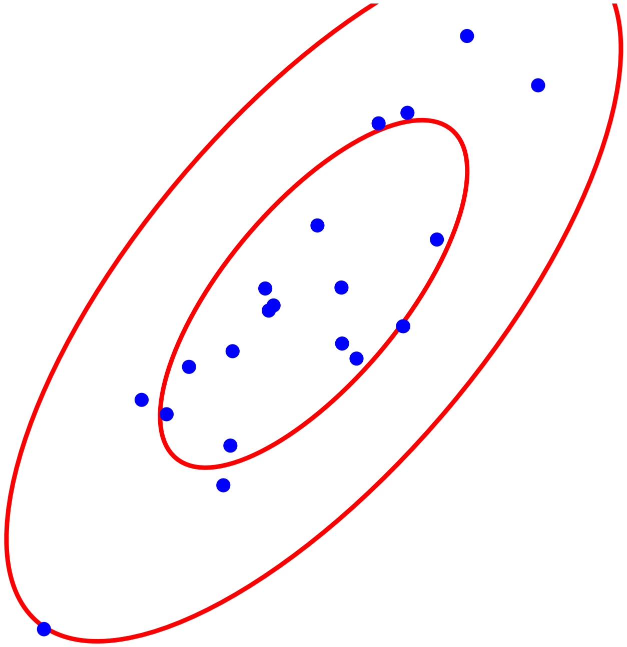
Markov Chain Sampling



We run the Markov chain n for steps. Each step moves from the current location x_i to a new x_{i+1}.



We "forget" the order and regard the locations $x_{1:n}$ as a random set of points.



If p (red contours) is both the invariant and initial distribution, each x_i is distributed as $x_i \sim p$.

Problems we need to solve

1. We have to construct a MC with invariant distribution p.
2. We cannot actually start sampling with $x_1 \sim p$; if we knew how to sample from p, all of this would be pointless.
3. Each point x_i is *marginally* distributed as $x_i \sim p$, but the points are *not* i.i.d.

Constructing the Markov Chain

Given is a continuous target distribution with density p.

Metropolis-Hastings (MH) kernel

1. We start by defining a conditional probability $q(y|x)$ on $\mathbf{X}$.

   q has nothing to do with p. We could e.g. choose $q(y|x) = g(y|x, \sigma^2)$, as in the previous example.

2. We define a **rejection kernel** A as

$$A(x_{n+1}|x_n) := \min\left\{1, \frac{q(x_i|x_{i+1})p(x_{i+1})}{q(x_{i+1}|x_i)p(x_i)}\right\}$$

   The normalization of p cancels in the quotient, so knowing $\tilde{p}$ is again enough.

3. We define the transition probability of the chain as

$$t(x_{i+1}|x_i) := q(x_{i+1}|x_i)A(x_{i+1}|x_i) + \delta_{x_i}(x_{i+1})c(x_i) \quad \text{where} \quad c(x_i) := \overbrace{\int q(y|x_i)(1 - A(y|x_i))dy}^{\text{total probability that a proposal is sampled and then rejected}}$$

Sampling from the MH chain

At each step $i + 1$, generate a proposal $x^* \sim q(\,.\,|x_i)$ and $U_i \sim \text{Uniform}[0, 1]$.

- If $U_i \leq A(x^*|x_i)$, accept proposal: Set $x_{i+1} := x^*$.
- If $U_i > A(x^*|x_i)$, reject proposal: Set $x_{i+1} := x_i$.

Problem 1: Initial distribution

Recall: Fundamental theorem on Markov chains

Suppose we sample $x_1 \sim P_{\text{init}}$ and $x_{i+1} \sim t(\,.\, | x_i)$. This defines a distribution P_i of x_i, which can change from step to step. If the MC is nice (recall: recurrent and aperiodic), then

$$P_i \rightarrow P_{\text{inv}} \qquad \text{for} \qquad i \rightarrow \infty \;.$$

Note: Making precise what aperiodic means in a continuous state space is a bit more technical than in the finite case, but the theorem still holds. We will not worry about the details here.

Implication

- If we can show that $P_{\text{inv}} \equiv p$, we do not have to know how to sample from p.
- Instead, we can start with *any* P_{init}, and will get arbitrarily close to p for sufficiently large i.

BURN-IN AND MIXING TIME

The number m of steps required until $P_m \approx P_{\text{inv}} \equiv p$ is called the **mixing time** of the Markov chain. (In probability theory, there is a range of definitions for what exactly $P_m \approx P_{\text{inv}}$ means.)

In MC samplers, the first m samples are also called the **burn-in** phase. The first m samples of each run of the sampler are discarded:

$$\underbrace{x_1, \dots, x_{m-1}}_{\substack{\text{Burn-in;}\\ \text{discard.}}}, \underbrace{x_m, x_{m+1}, \dots}_{\substack{\text{Samples from}\\ \text{(approximately) } p\text{;}\\ \text{keep.}}}$$

Convergence diagnostics

In practice, we do not know how large j is. There are a number of methods for assessing whether the sampler has mixed. Such heuristics are often referred to as **convergence diagnostics**.

PROBLEM 2: SEQUENTIAL DEPENDENCE

Even after burn-in, the samples from a MC are not i.i.d.

Strategy

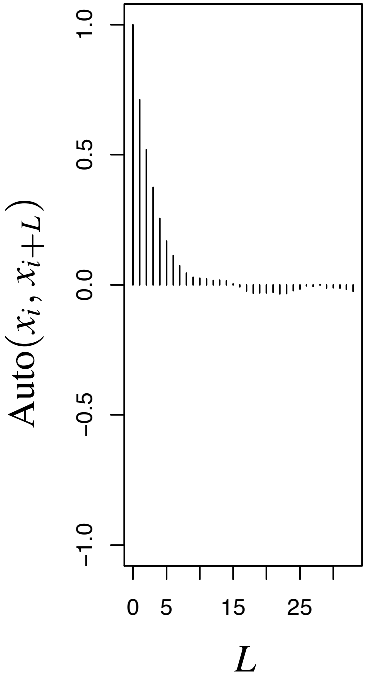
- Estimate empirically how many steps L are needed for x_i and x_{i+L} to be approximately independent. The number L is called the **lag**.
- After burn-in, keep only every Lth sample; discard samples in between.

Estimating the lag

The most commen method uses the **autocorrelation function**:

$$\text{Auto}(x_i, x_j) := \frac{\mathbb{E}[x_i - \mu_i] \cdot \mathbb{E}[x_j - \mu_j]}{\sigma_i \sigma_j}$$

We compute $\text{Auto}(x_i, x_{i+L})$ empirically from the sample for different values of L, and find the smallest L for which the autocorrelation is close to zero.

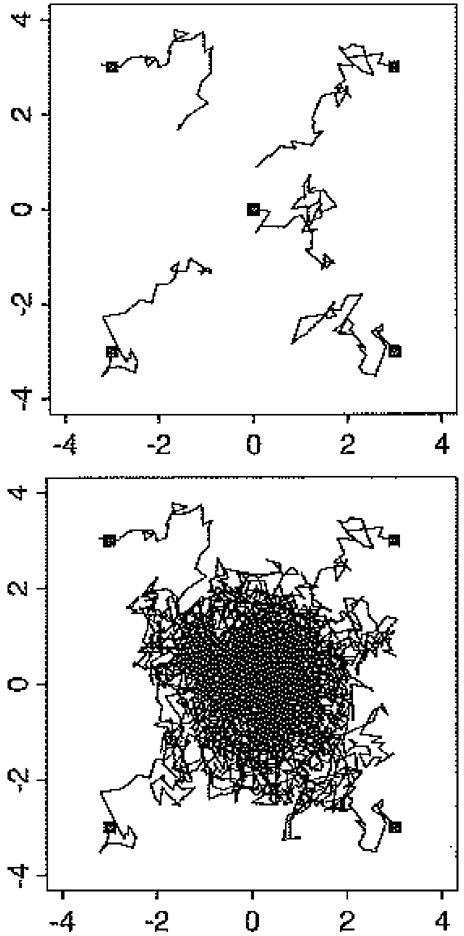


CONVERGENCE DIAGNOSTICS

There are about half a dozen popular convergence crieteria; the one below is an example.

Gelman-Rubin criterion

- Start several chains at random. For each chain k, sample x_i^k has a marginal distribution P_i^k.
- The distributions of P_i^k will differ between chains in early stages.
- Once the chains have converged, all $P_i = P_{\text{inv}}$ are identical.
- Criterion: Use a hypothesis test to compare P_i^k for different k (e.g. compare P_i^2 against null hypothesis P_i^1). Once the test does not reject anymore, assume that the chains are past burn-in.



Reference: A. Gelman and D. B. Rubin: "Inference from Iterative Simulation Using Multiple Sequences", *Statistical Science*, Vol. 7 (1992) 457-511.

Stochastic Hill-Climbing

The Metropolis-Hastings rejection kernel was defined as:

$$A(x_{n+1}|x_n) = \min\left\{1, \frac{q(x_i|x_{i+1})p(x_{i+1})}{q(x_{i+1}|x_i)p(x_i)}\right\} .$$

Hence, we certainly accept if the second term is larger than 1, i.e. if

$$q(x_i|x_{i+1})p(x_{i+1}) > q(x_{i+1}|x_i)p(x_i) .$$

That means:

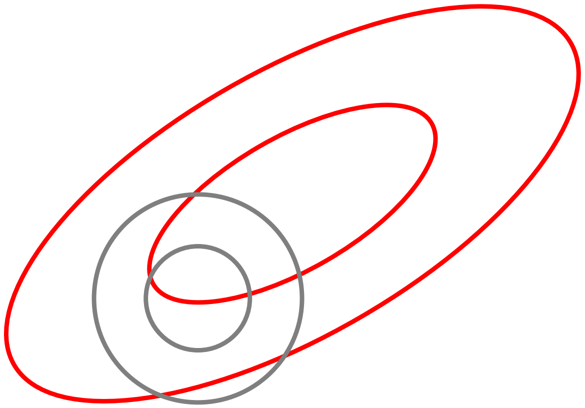
- We always accept the proposal x_{i+1} if it *increases* the probability under p.
- If it *decreases* the probability, we still accept with a probability which depends on the difference to the current probability.

Hill-climbing interpretation

- The MH sampler somewhat resembles a gradient ascent algorithm on p, which *tends* to move in the direction of increasing probability p.
- However:
  - The actual steps are chosen at random.
  - The sampler can move "downhill" with a certain probability.
  - When it reaches a local maximum, it does not get stuck there.

Selecting a Proposal Distribution

Everyone's favorite example: Two Gaussians



red = target distribution p
gray = proposal distribution q

- $\text{Var}[q]$ too large:
  Will overstep p; many rejections.
- $\text{Var}[q]$ too small:
  Many steps needed to achieve good coverage of domain.

If p is unimodal and can be roughly approximated by a Gaussian, $\text{Var}[q]$ should be chosen as smallest covariance component of p.

More generally

For complicated posteriors (recall: small regions of concentration, large low-probability regions in between) choosing q is much more difficult. To choose q with good performance, we already need to know something about the posterior.

There are many strategies, e.g. mixture proposals (with one component for large steps and one for small steps).

Summary: MH Sampler

- MCMC samplers construct a MC with invariant distribution p.
- The MH kernel is one generic way to construct such a chain from p and a proposal distribution q.
- Formally, q does not depend on p (but arbitrary choice of q usually means bad performance).
- We have to discard an initial number m of samples as burn-in to obtain samples (approximately) distributed according to p.
- After burn-in, we keep only every Lth sample (where L = lag) to make sure the x_i are (approximately) independent.

$$\underbrace{x_1,\dots,x_{m-1}}_{\text{Burn-in; discard.}}, \overset{\text{Keep.}}{\overset{\downarrow}{x_m}}, \underbrace{x_{m+1},\dots,x_{m+L-1}}_{\text{Samples correlated with } x_j\text{; discard.}}, \overset{\text{Keep.}}{\overset{\downarrow}{x_{m+L}}}, \underbrace{x_{m+L+1},\dots x_{m+2L-1}}_{\text{Samples correlated with } x_{j+L}\text{; discard.}}, \overset{\text{Keep.}}{\overset{\downarrow}{x_{m+2L}}}, \dots$$