Advanced Data Analytics
Lecture week 6: Sampling

Ian T. Nabney

University of BRISTOL

Overview

- Understand motivation for sampling in Bayesian inference
- Overview of a few simple low-dimensional sampling methods
- Understand principles of Markov Chain Monte Carlo sampling

Further reading: Bishop chapter 11. Nabney chapter 8.

Purpose of sampling

- For most situations the posterior distribution is required primarily for the purpose of evaluating expectations, e.g. to make predictions. The fundamental problem that we therefore wish to address is to find the expectation of some function $f(\mathbf{z})$ with respect to a probability distribution $p(\mathbf{z})$.
- The general idea behind sampling methods is to obtain a set of samples $\mathbf{z}^{(l)}$ (where $l = 1, \ldots, L$) drawn independently from the distribution $p(\mathbf{z})$. This allows an expectation to be approximated by a finite sum

$$\widehat{f} = \frac{1}{L} \sum_{l=1}^{L} f(\mathbf{z}^{(l)}). \tag{1}$$

- The variance of the estimator is given by

$$\mathrm{var}[\widehat{f}] = \frac{1}{L} \mathbb{E}\left[(f - \mathbb{E}[f])^2\right] \tag{2}$$

  is the variance of the function $f(\mathbf{z})$ under the distribution $p(\mathbf{z})$.

- Two problems can arise:
  - The samples $\{\mathbf{z}^{(l)}\}$ might not be independent, and so the effective sample size might be much smaller than the apparent sample size.
  - If $f(\mathbf{z})$ is small in regions where $p(\mathbf{z})$ is large, and vice versa, then the expectation may be dominated by regions of small probability, implying that relatively large sample sizes will be required to achieve sufficient accuracy.

Transformation method

- These algorithms generate a sequence of pseudo-random numbers. That is, the numbers are generated deterministically, but pass a wide range of statistical tests for randomness (e. g. zero autocorrelation).
- Everything is based on an algorithm that generates pseudo-random numbers uniformly over $(0, 1)$.
- Transformation method. If x is uniform, then what is the density of $y(x)$?

$$p(y) = p(x) \left| \frac{dx}{dy} \right| .$$

- If x is uniform on $(0, 1)$ and $y(x) = -\log(x)$, then

$$p(y)\, dy = 1. \left| \frac{dx}{dy} \right| \, dy = e^{-y}\, dy$$

and y/λ has density $\lambda e^{-\lambda y}$, which is exponential with parameter λ.

Limitations of Transformation Method

- The transformation method works by finding $y(x)$ such that

$$\frac{dx}{dy} = p(y).$$

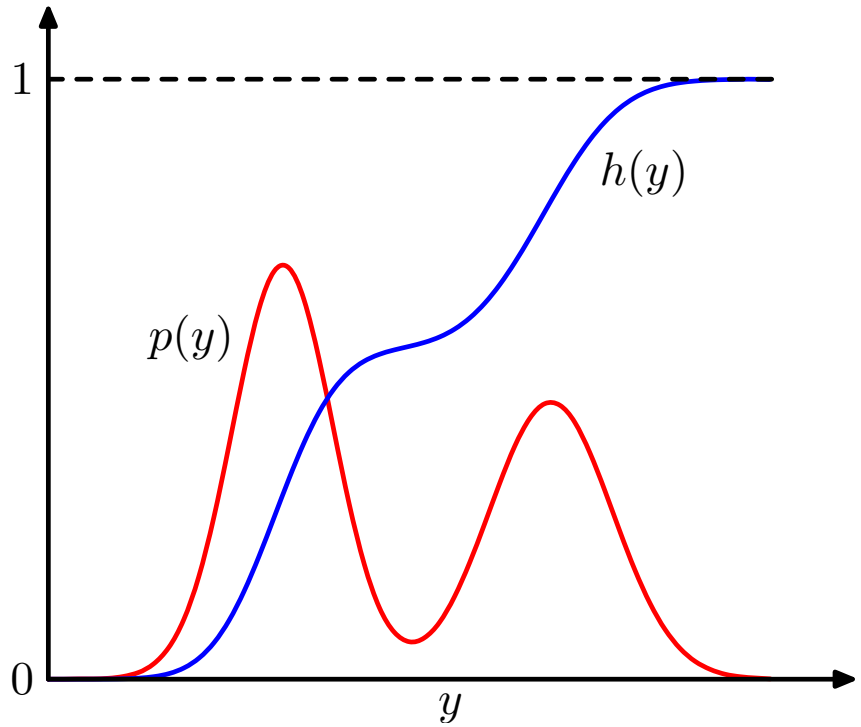
- Let

$$h(y) = \int_{\infty}^{y} p(\tau)\, d\tau.$$

  Then $x = h(y)$ is the solution, and $y(x) = h^{-1}(x)$.
- So this method can be applied only if h^{-1}, the inverse function of the indefinite integral of $p(y)$ (i.e. the inverse of the cumulative distribution function), can be computed.
- This is only possible for a very small number of distributions, and the list does not include the normal distribution!
- This also explains why sampling is such a difficult task in general.

Geometric interpretation

- $h(y)$ is the indefinite integral of the desired distribution $p(y)$.
- If a uniformly distributed random variable z is transformed using $y = h^{-1}(z)$, then y will be distributed according to $p(y)$.



Sampling a Gaussian

Let x_1 and x_2 be $U(0,1)$ and

$$y_1 = \sqrt{-2\log x_1}\cos 2\pi x_2$$
$$y_2 = \sqrt{-2\log x_1}\sin 2\pi x_2$$

or

$$x_1 = \exp\left[\frac{-1}{2}(y_1^2 + y_2^2)\right]$$
$$x_2 = \frac{1}{2\pi}\arctan\frac{y_2}{y_1}$$

Then

$$\frac{\partial(x_1,x_2)}{\partial(y_1,y_2)} = -\left[\frac{1}{\sqrt{2\pi}}e^{-y_1^2/2}\right]\left[\frac{1}{\sqrt{2\pi}}e^{-y_2^2/2}\right]$$

which implies that y_1 and y_2 are independent normal distributions.

Rejection Sampling

Suppose that we are trying to sample from a random variable with density function $p(\mathbf{x})$, and that we have a function $f(\mathbf{x})$ which has the following properties:

1. $f(\mathbf{x}) \geq p(\mathbf{x}) \quad \forall \mathbf{x}$.
2. $f(\mathbf{x}) = kq(\mathbf{x})$ for known k and density function q which can be sampled from.

Then to generate samples from p we use the following procedure:

1. Generate a value $\mathbf{y}$ from q.
2. Generate a value u from $U(0, 1)$.
3. If $\frac{p(\mathbf{y})}{f(\mathbf{y})} \geq u$ then accept the sample $\mathbf{y}$. Otherwise reject it.
4. Repeat from step 1 until sufficiently many samples have been generated.

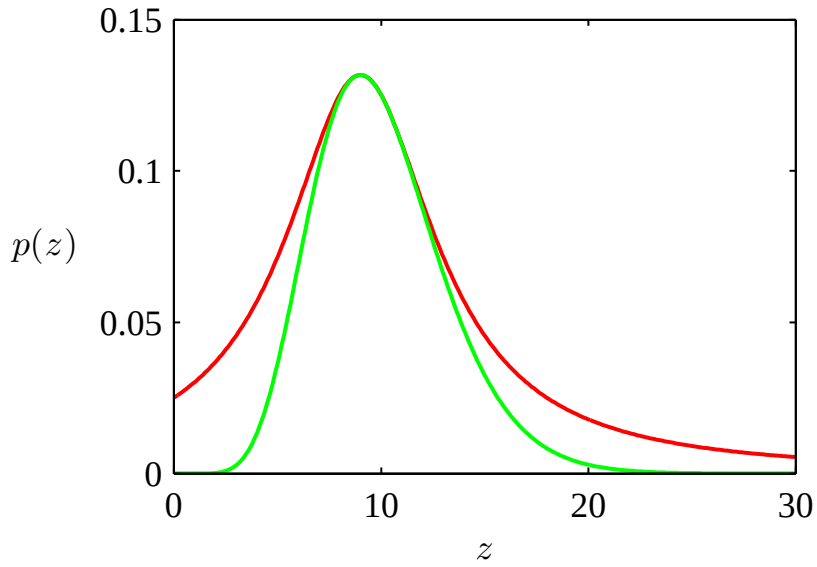
Rejection sampling example

- Plot showing the gamma distribution given by

$$\mathrm{Gam}(z|a,b) = \frac{b^a z^{a-1} \exp(-bz)}{\Gamma(a)} \tag{3}$$

as the green curve, with a scaled Cauchy proposal distribution shown by the red curve.

- Samples from the gamma distribution can be obtained by sampling from the Cauchy (using a transformation method) and then applying the rejection sampling criterion.



Integration with Dependent Samples

Write $Q(\theta)$ for the posterior distribution of model parameters.

- The expectation of $a(\theta)$ is

$$E[a] = \int a(\theta) Q(\theta)\, d\theta \approx \frac{1}{N} \sum_{t=1}^{N} a(\theta^{(t)}), \tag{4}$$

  where $\theta^{(1)}, \ldots, \theta^{(n)}$ are independent samples from Q.
- The formula can still give an unbiased estimate of $E[a]$ even when the $\theta^{(t)}$ are dependent.
- The variance of the estimate of $E[a]$ is given by $\text{var}[a]/N$ if the samples are independent.
- For large N and dependent samples, the variance is $\text{var}[a]/(N/\tau)$, where $\tau = 1 + 2\sum_{s=1}^{\infty} \rho(s)$, and $\rho(s)$ is the autocorrelation of $a(\theta^{(t)})$ at lag s.

Continuous Markov Chains

- This is all very interesting, but a discrete Markov chain is of relatively little use for sampling: if a distribution is defined on a finite space, it is very easy to sample from directly.
- We define a Markov chain on a continuous space by the initial distribution for the first state, $\mathbf{x}^{(1)}$, and a transition density $T(\mathbf{x}^{(n+1)}|\mathbf{x}^{(n)})$, which replaces the transition matrix.
- The definitions of recurrence, aperiodicity etc. can all be extended to the continuous case.
- A density function $\boldsymbol{\pi}$ is the stationary (or invariant) distribution of the chain if

$$\boldsymbol{\pi}(\mathbf{x}') = \int T(\mathbf{x}'|\mathbf{x})\boldsymbol{\pi}(\mathbf{x}) \, \mathrm{d}\mathbf{x}. \tag{5}$$

Stationary Distributions and Detailed Balance

- Consider the detailed balance condition (which is the analogue of time-reversibility):

$$T(\mathbf{x}'|\mathbf{x})\pi(\mathbf{x}) = T(\mathbf{x}|\mathbf{x}')\pi(\mathbf{x}'). \tag{6}$$

- Detailed balance implies that π is a stationary distribution for the Markov chain.

$$\int T(\mathbf{x}'|\mathbf{x})\pi(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int T(\mathbf{x}|\mathbf{x}')\pi(\mathbf{x}')\,\mathrm{d}\mathbf{x} \tag{7}$$

$$= \pi(\mathbf{x}') \int T(\mathbf{x}|\mathbf{x}')\,\mathrm{d}\mathbf{x} \tag{8}$$

$$= \pi(\mathbf{x}'), \tag{9}$$

- A continuous Markov chain that is ergodic has a unique stationary distribution to which it converges from any initial state. There are deep theorems which describe the conditions necessary for ergodicity.

What has this got to do with Sampling?

- Our aim is to define a Markov chain which has the distribution $Q(\theta)$ that we want to sample from as its stationary distribution, and is such that we can generate samples easily from the chain, which means that we must be able to sample easily from the transition distribution T.
- This seems impossible! Even if we could find such a Markov chain, it would seem likely that it would have to be very carefully tailored to the properties of the particular distribution $Q(\theta)$.
- However, it is a remarkable fact that there are several general purpose methods for constructing Markov chains with a given stationary distribution.

Constructing Markov Chains

We will discuss two simple methods:

Gibbs sampling: applicable when we want to sample from a multi-dimensional parameter vector and we can sample from the conditional distribution (under Q) of one component of θ given values for all the other components.

Metropolis–Hastings sampling can be used for any density: it is related to simulated annealing.

In general, we can combine several methods for MCMC sampling in a deterministic (e.g. alternate method (a) and method (b)) or stochastic (e.g. choose method (a) with probability p, otherwise, choose method (b)) way. This is because ergodicity and detailed balance are both preserved by these combination rules.

Applications of Gibbs Sampling

- This sampling algorithm is useful for Bayesian inference if it is reasonably easy to define and sample from the posterior distribution of one parameter conditional on the values of all the rest.
- For many statistical models this is the case. However, for neural networks, the posterior conditional distributions are extremely complex, and so Gibbs sampling cannot be used.
- However, it does form a component of the hybrid Monte Carlo algorithm, which we will meet later on. It is often an appropriate way to sample hyperparameters, particularly when we assume that their distribution factorises.

Metropolis–Hastings Algorithm

- Unlike Gibbs sampling, the Metropolis–Hastings algorithm makes no assumptions about the form of the desired distribution Q.
- In the Markov chain defined by the Metropolis–Hastings algorithm, a new state $\theta^{(t+1)}$ is generated from the old state $\theta^{(t)}$ by first (stochastically) generating a candidate state from a proposal distribution, and then deciding whether or not to accept the candidate state.
- If it is accepted, then $\theta^{(t+1)}$ is made equal to the candidate state, otherwise the new state is the same as the previous state.

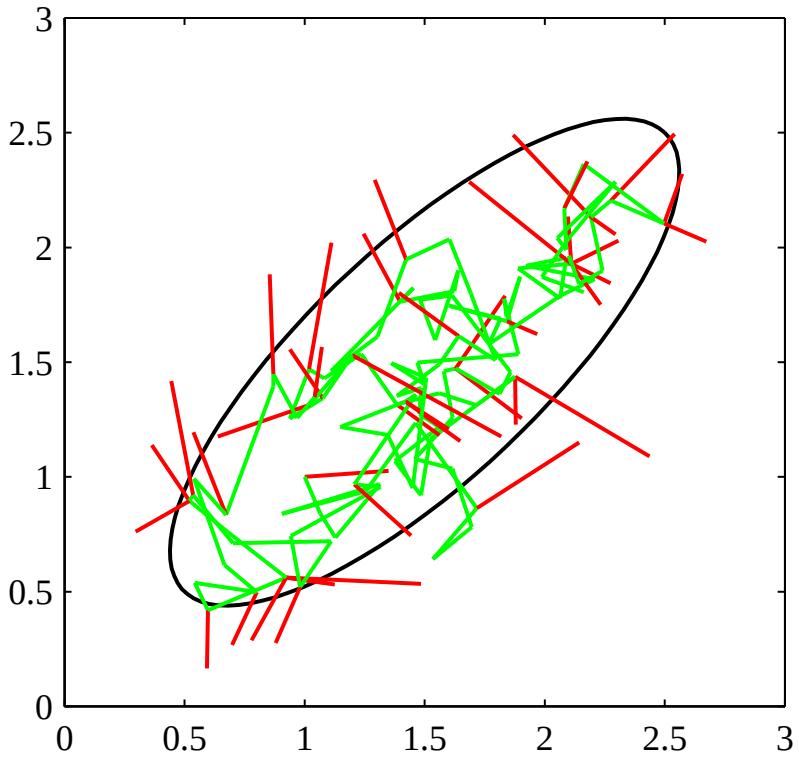
Metropolis–Hastings Implementation

1. Generate the candidate state θ^* with density given by the proposal distribution $S(\theta^*|\theta^{(t)})$. Note that the proposal distribution typically depends on the old state.
2. If $Q(\theta^*) \geq Q(\theta^{(t)})$, then accept the candidate state. If $Q(\theta^*) < Q(\theta^{(t)})$, then accept the candidate state with probability $Q(\theta^*)/Q(\theta^{(t)})$. (This stochastic acceptance step is similar to that used in the simulated annealing optimisation algorithm.)
3. If the candidate state is accepted, let $\theta^{(t+1)} = \theta^*$. If the candidate state is rejected, let $\theta^{(t+1)} = \theta^{(t)}$.

The proposal distribution must satisfy a symmetry condition $S(\theta'|\theta) = S(\theta|\theta')$.

Metropolis-Hastings example

- Using Metropolis algorithm to sample from a Gaussian distribution whose one standard-deviation contour is shown by the ellipse.
- The proposal distribution is an isotropic Gaussian distribution whose standard deviation is 0.2.
- Steps that are accepted are shown as green lines, and rejected steps are shown in red. A total of 150 candidate samples are generated, of which 43 are rejected.



Choice of Proposal Distribution

- One simple choice is a Gaussian distribution centred on $\theta^{(t)}$, with standard deviation chosen so that the probability of the candidate state being accepted is reasonably high (which usually means that the standard deviation should be small).
- However, a small standard deviation also leads to a high degree of dependence between successive states, since many steps are needed to move to a distant point in the distribution. This problem is made worse by the fact that these movements take the form of a random walk. The proposal distribution does not depend on Q.
- If there are strong correlations in the distribution that we wish to sample from (and there commonly are for neural networks) then most random steps will try to climb the side of the valley walls. Thus, although the Metropolis–Hastings algorithm is feasible for small neural networks, it breaks down for networks of more reasonable size.

Principles of the Hybrid Monte Carlo Algorithm

- This algorithm combines the Metropolis algorithm with sampling techniques based on dynamical simulation.
- This allows us to incorporate gradient information from the distribution Q, which is used to bias the directions in which we move.
- This information can be calculated relatively easily for neural networks using back-propagation.
- The original algorithm is due to Duane, Kennedy, Pendleton and Roweth.

Problem Reformulation

- We sample from the canonical (or Boltzmann) distribution for the state of a hypothetical physical system defined in terms of an energy function.
- We use a position variable q, which has n real-valued components q_i, each corresponding to the value of a model parameter. The probability density for this variable under the canonical distribution is defined by

$$P(q) \propto \exp(-E(q)), \tag{10}$$

  where $E(q)$ is the potential energy function.
- Any probability density that is nowhere zero can be put in this form by defining

$$E(q) = -\log P(q) - \log Z,$$

  for any convenient value Z.

Hamiltonian Definition

- To introduce dynamics of the system, we create a momentum variable p, which also has n real-valued components p_i. The canonical distribution over the space of q and p together is defined by

$$P(q,p) \propto \exp(-H(q,p)), \tag{11}$$

where $H(q,p) = E(q) + K(p)$ is the Hamiltonian function giving the total energy.

- $K(p)$ is the 'kinetic energy', which can be computed from the momentum by

$$K(p) = \sum_{i=1}^{n} \frac{p_i^2}{2m_i}, \tag{12}$$

where m_i is the 'mass' associated with the ith component.

- Adjustment of these masses can improve efficiency, but we shall assume that they are one for the moment.

University of BRISTOL

Sampling Strategy

- In the distribution defined in (11), q and p are independent, and the marginal distribution of q is the same as that in (10). We can therefore proceed by defining a Markov chain that converges to the canonical distribution $P(q, p)$ and then simply ignore the values of p and generate samples using q.
- Sampling from $P(q, p)$ is performed in two stages:
  1. sampling uniformly from the values of q and p at a fixed total energy $H(q, p)$,
  2. sampling states with different values of H.

Dynamical Simulation

Sampling at a fixed total energy is done by simulating the Hamiltonian dynamics of the system, in which the state evolves in a fictitious time, τ, according to the dynamical equations

$$\frac{dq_i}{d\tau} = +\frac{\partial H}{\partial p_i} = \frac{p_i}{m_i} \tag{13}$$

$$\frac{dp_i}{d\tau} = -\frac{\partial H}{\partial q_i} = -\frac{\partial E}{\partial q_i}. \tag{14}$$

For this to work, we must be able to compute the partial derivatives of E with respect to the components q_i.

Stochastic Dynamics

- Transitions based on Hamiltonian dynamics will eventually explore the whole region of phase space with a given value of H.
- Such transitions are not sufficient to produce an ergodic Markov chain, since the value of H does not change.
- To correct this, we alternately perform deterministic dynamical transitions and stochastic Gibbs sampling updates of the momentum.
- Since q and p are independent, p may be updated by drawing a new value with probability density proportional to $\exp(-K(p))$. For the quadratic kinetic energy function of (12), this is easy since the p_i have independent Gaussian distributions. Updates of p can change the total energy H, allowing the whole phase space to be explored.

Hybrid Monte Carlo

- The HMC algorithm samples points in phase space with a Markov chain that alternates stochastic and dynamic transitions.
- The stochastic transitions replace the momentum values using Gibbs sampling. The dynamical transitions are similar to those for stochastic dynamics, but with two changes:
  1. A random decision is made for each transition whether to simulate the dynamics forward or backward in time.
  2. The point reached at the end of the dynamics is only a candidate for a new state, to be accepted or rejected based on the change in total energy H just as in the Metropolis algorithm.
- Because of the discretisation, H may change, and therefore some moves may be rejected: these rejections exactly eliminate the bias introduced by the non-zero step size.

Practical Issues

- How many iterations should we discard at the start of the chain?
- One run or several?
- Deciding on a starting point.

Burn-in

- In order to reduce the possibility of bias caused by the initial conditions, it is usual to throw away the samples in an initial transient phase (called the burn-in period).
- It is difficult to determine how long the burn-in should be, since rates of convergence of the chain to the desired stationary distribution depend dramatically on the details of the algorithm and the target distribution.
- Unfortunately, most of the theoretical results are either extremely specific to an algorithm or target distribution or give impractical bounds (or both), so in practice, empirical measures have to be used.

One Run or Several?

The arguments for a single run:

- the chain will be closer to the target distribution at the end of one long run than it would be at the end of a number of shorter ones;
- the burn-in samples must be discarded from the start of each shorter run.

Multiple runs have the advantage that:

- they can be initialised from several different starting positions, so are more likely to explore the entire sample space;
- it is possible to monitor the sample paths to see how well the chains are mixing (i.e. to what extent the output from the different chains are indistinguishable) and also to carry out other diagnostics.

Initialising the Chain

- If several runs are made, it is argued that the distribution of the starting points should be over-dispersed with respect to the target distribution, as this helps to detect convergence.
- To find starting points from different modes, non-linear optimisation from widely dispersed starting points, or simulated annealing can be used.

Convergence Diagnostics

- In the absence of theoretical results, it is necessary to perform a statistical analysis to assess convergence in a less rigorous way. We describe an algorithm of Gelman which is relatively simple to compute and has given good results in practice.
- The principle of this method is that when a chain has converged it has forgotten its starting point and so several sequences drawn from different starting points should be indistinguishable. A group of sequences can be overlaid on a single plot to see if they have similar properties, but a more quantitative approach is to determine if the variance between different sequences is of a similar size to the variance within each sequence.
- This method can be applied to any scalar function or summary statistic $\psi(\mathbf{x})$ of each sample, for example, a coordinate value x_i or the energy.

Gelman's Algorithm

1. Generate $m \geq 2$ sequences of length $2n$, each beginning at different starting points which are overdispersed. The burn-in period is taken to be the first n samples of each sequence. We write the set of retained scalar summary values as ψ_{ij} where $i = 1, \ldots, m$ indexes the sequences and $j = 1, \ldots, n$ indexes the samples.
2. Compute the standard analysis of variance (ANOVA) statistics B, the between-sequence variance, and W, the within-sequence variance:

$$B = \frac{n}{m-1} \sum_{i=1}^{m} (\overline{\psi}_{i.} - \overline{\psi}_{..})^2 \tag{15}$$

$$W = \frac{1}{m} \sum_{i=1}^{m} s_i^2, \quad \text{where} \quad s_i^2 = \frac{1}{n-1} \sum_{j=1}^{n} (\psi_{ij} - \overline{\psi}_{i.})^2. \tag{16}$$

Gelman's Diagnostic

- We then construct two estimates of the variance of ψ in the target distribution, var(ψ).
  1. W: this should underestimate the var(ψ) since the danger is that each sequence has not ranged over the full range of the target distribution.
  2. $\hat{V} = \frac{n-1}{n}W + \frac{1}{n}B$, is an estimate of var($\psi$) that is unbiased under stationarity (that is, the starting points were drawn from the target distribution) but is an overestimate if the starting points were overdispersed.
- The convergence diagnostic statistic, known as the estimated potential scale reduction is given by $\sqrt{\hat{R}}$, where $\hat{R} = \frac{\hat{V}}{W}$.
- Rather than developing a significance test, Gelman suggests monitoring the value of this statistic and accepting a group of sequences if it falls below 1.1 or 1.05 for all measures of interest.

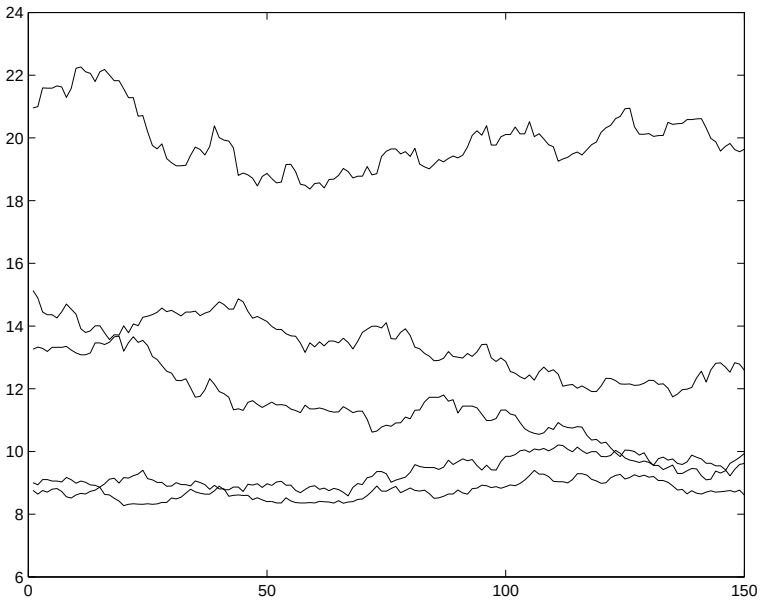
Diagnostic in Action

- Gaussian $N(0, 3)$ target distribution in $\mathbb{R}^5$, and the summary statistics are the five coordinates $x_1, \ldots, x_5$ and the energy $-\ln p(\mathbf{x})$.
- A Metropolis–Hastings MCMC sampler was run for 300 iterations in five runs with the starting point drawn from $N(0, 3)$. In the first set of runs the proposal distribution had a variance of 0.01 and in the second set the variance was 0.8.

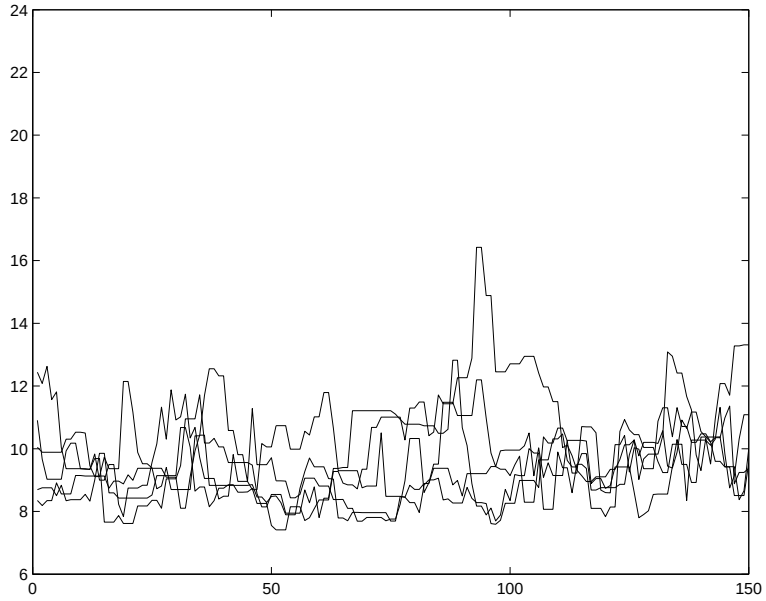
	First Run	**Second Run**
x_1	1.699	1.044
x_2	9.080	1.000
x_3	5.718	1.173
x_4	2.069	1.014
x_5	4.375	0.998
Energy	5.425	1.010

300 samples are not nearly enough for the chain to converge with a proposal variance of 0.01, but that there is a reasonable level of confidence that the chains with proposal variance of 0.8 have converged.

Sampling Results: Energies

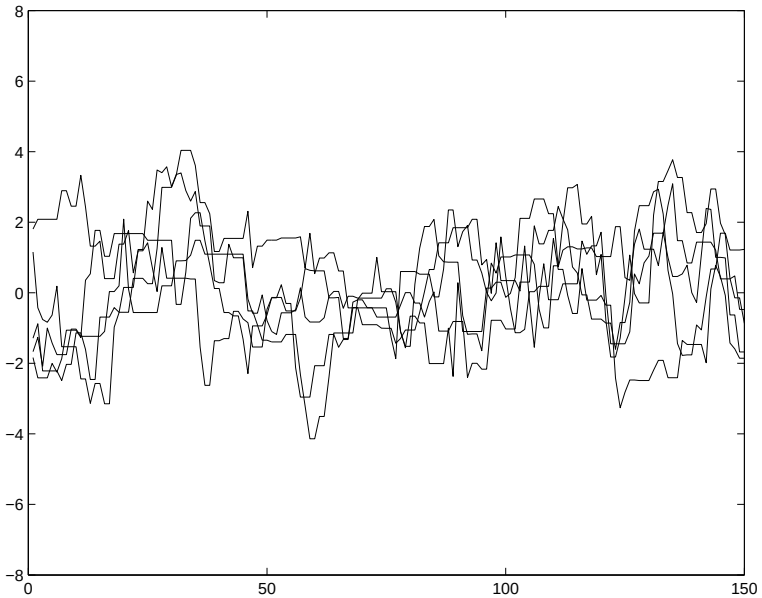


Proposal Variance 0.01

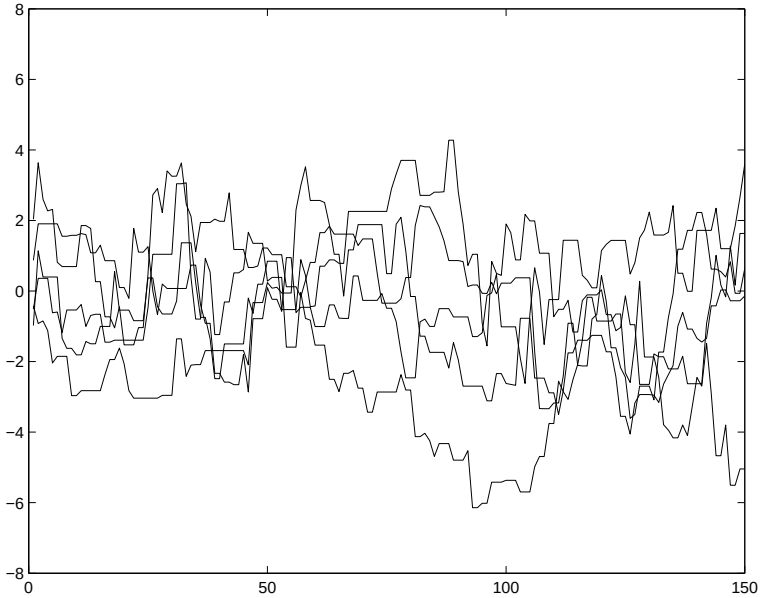


Proposal Variance 0.8

Sampling Results: Coordinates from Run 2



x_2



x_3