11 Sampling and Markov Chain Monte Carlo

For me, sampling is a high art. Most people don't see it that way, but it's a beautiful thing.
—JPEGMAFIA

Sampling, sometimes known as the *Monte Carlo method*, is a powerful tool in mathematics, in machine learning, and in statistics. Large language models (LLMs) and other forms of generative AI generate their outputs by sampling from some learned probability distribution. Sampling is also the primary way to evaluate integrals in high dimension, and many important quantities are essentially integrals. For example, expected values of random variables are integrals, and most of these expected values (and other integrals) cannot be evaluated symbolically in closed form. Quadrature methods like Clenshaw–Curtis and Gaussian quadrature work well in one dimension but suffer from the curse of dimensionality—the complexity of those methods grows exponentially with the dimension of the problem. But sampling works relatively well even in high dimensions.

Bayesian statistics is especially dependent on sampling. Working with a Bayesian posterior distribution requires computing integrals of the form $\int p(X \,|\, \boldsymbol{\theta}) f(\boldsymbol{\theta})\, d\boldsymbol{\theta}$, which show up in the denominator of Bayes' rule. Except in very special cases, this integral rarely has a closed-form solution, but we can often approximate many properties of the posterior using Monte Carlo methods and sampling techniques without ever evaluating this integral. And even when we really need the value of the integral, the best way to estimate its value is usually through sampling.

Remarkably, we often don't need the exact posterior distribution—just a constant multiple of it will do. In particular, several powerful sampling methods can sample from a distribution just knowing a constant multiple of the distribution. That is, without knowing $f(\boldsymbol{\theta} \,|\, \mathbf{D})$, but knowing a function $g(\boldsymbol{\theta})$ such that $f(\boldsymbol{\theta} \,|\, \mathbf{D}) \propto g(\boldsymbol{\theta})$, we can sample from a distribution with p.d.f. equal to $f(\boldsymbol{\theta} \,|\, \mathbf{D})$.

Here are several things we can do easily with a draw $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from a distribution with density $f_X(\mathbf{x})$:

(i) Estimate the expected value $\mathbb{E}[X]$, if it exists,[40] as $\mathbb{E}[X] \approx \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$. And more generally, for any function $k(\mathbf{x})$ such that $\mathbb{E}[k(X)]$ exists, we can estimate $\mathbb{E}[k(X)]$ as $\mathbb{E}[k(X)] \approx \frac{1}{n}\sum_{i=1}^{n} k(\mathbf{x}_i)$.

(ii) Estimate probabilities or densities. For any measurable set A, the fundamental bridge shows that $P(X \in A) = \mathbb{E}[\mathbb{1}_A(X)]$, and this can be estimated as

$$P(X \in A) = \mathbb{E}[\mathbb{1}_A(X)] \approx \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_A(\mathbf{x}_i).$$

This allows us to approximate many probabilities. For example, the c.d.f. $F_X(x)$ of a univariate random variable X can be estimated as the fraction of the samples that are not greater than x:

$$\begin{aligned} F_X(x) = P(X \le x) &= E[\mathbb{1}_{(-\infty,x]}(X)] \\ &\approx \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{(-\infty,x]}(x_i) \\ &= \frac{|\{x_i : x_i \le x\}|}{n}. \end{aligned}$$

These function approximations don't behave well under differentiation (the derivative is zero or undefined at every point), but the density $f_X(\mathbf{x})$ can be approximated using a KDE or histogram from $x_1, \ldots, x_n$.

(iii) Marginalize. If $X = (Y, Z)$ and $\mathbf{x}_i = (\mathbf{y}_i, \mathbf{z}_i)$, then we can just forget all of the $\mathbf{z}_i$. The remaining $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$ is a draw from the marginal distribution (with density $f_Y(\mathbf{y}) = \int f_{(Y,Z)}(\mathbf{y}, \mathbf{z})\, d\mathbf{z}$). Therefore, probabilities of the form $P(Y \in B)$ as well as the c.d.f. and the p.d.f. for Y can all be estimated from $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$, just ignoring the $\mathbf{z}_i$.

(iv) Estimate the quantiles or percentiles of a distribution, simply by computing the quantiles or percentiles of the sample. For example, the median of a univariate distribution can be estimated from samples $x_1, \ldots, x_n$ by sorting the samples $x_{i_1} \le x_{i_2} \le \cdots \le x_{i_n}$ and taking the middle-most sample $x_{i_{(n+1)/2}}$ if n is odd or the average $\frac{1}{2}(x_{i_{n/2}} + x_{i_{n/2+1}})$ if n is even.

In the first section of this chapter we review some classical sampling methods, and in the remainder of the chapter we focus on the powerful sampling technique called *Markov-chain Monte Carlo*.

11.1 Review of Classical Sampling

[40] It's important to note that the sample mean $\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ always exists, even if the expected value it is estimating does not. But if the expected value doesn't exist, then the central limit theorem doesn't apply and the value of the sample mean is meaningless.

Nota Bene 11.1.1. Much of the content in this section is covered in Volume 2; but students may have forgotten it, and it's important, so we review it here.

11.1.1 Monte Carlo Integration

We can use Monte Carlo ideas to estimate integrals of the form

$$\int_C k(\mathbf{x}) f_X(\mathbf{x})\, d\mathbf{x}$$

for any distribution X supported on $C \subset \mathbb{R}^d$ with p.d.f. $f_X(\mathbf{x})$, provided $k(\mathbf{x})$ is sufficiently well behaved, and provided we can sample from X. If $\mathbb{E}[k(\mathbf{x})]$ exists (this requires that $\int_C |k(\mathbf{x})| f_X(\mathbf{x})\, d\mathbf{x} < \infty$), then we have

$$\int_C k(\mathbf{x}) f_X(\mathbf{x})\, d\mathbf{x} = \mathbb{E}[k(X)] \approx \frac{1}{n} \sum_{i=1}^{n} k(\mathbf{x}_i),$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is a draw from the distribution X. If $\mathbb{E}[k(\mathbf{x})^2]$ also exists, then the variance exists, and the standard error is (we drop the subscript)

$$SE \approx \frac{\widehat{s}}{\sqrt{n}} = \frac{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n} (k(\mathbf{x}_i) - \bar{\mathbf{y}})^2}}{\sqrt{n}} = \sqrt{\frac{\sum_{i=1}^{n} (k(\mathbf{x}_i) - \bar{\mathbf{y}})^2}{n(n-1)}}, \tag{11.1}$$

where $\bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n} k(\mathbf{x}_i)$.

Example 11.1.2. Let $X \sim \mathcal{N}(0,1)$ have standard normal distribution. There are many efficient algorithms for sampling from the standard normal (the Box–Muller method of Section 4.7.3, for example), so we can estimate the c.d.f. $F_X(a) = P(X \le a) = \int_{-\infty}^{a} f_X(x)\, dx$ by sampling from the standard normal:

$$\int_{-\infty}^{a} f_X(x)\, dx = \int_{-\infty}^{\infty} \mathbb{1}_{(-\infty,a]}(x) f_X(x)\, dx = \mathbb{E}[\mathbb{1}_{(-\infty,a](X)}] \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{(-\infty,a]}(x_i).$$

Using this method to calculate $F_X(1)$, and taking 10^3, 10^4, and 10^5 samples, we gained the estimates 0.8350, 0.8451, and 0.8422, respectively, with standard error of 0.012, 0.004, and 0.001, respectively. Compare this to the true value $F_X(1) = 0.8413$.

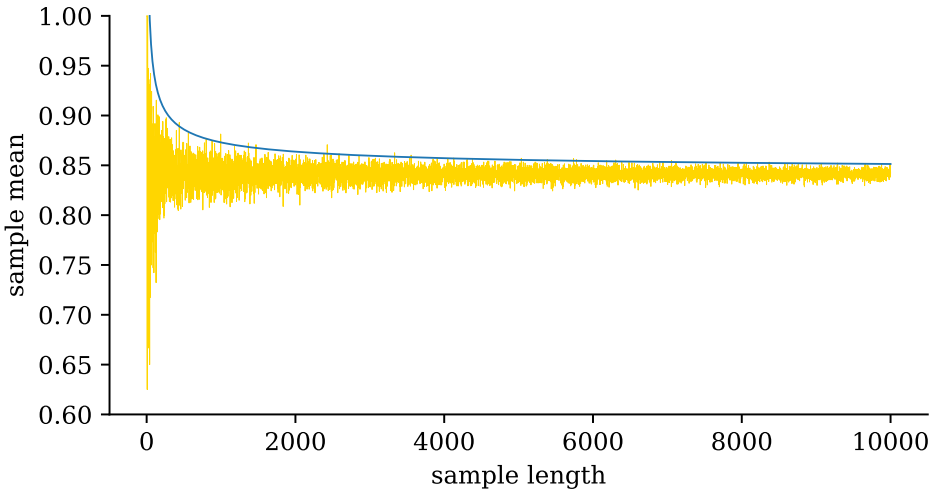


Figure 11.1: The value of the normal c.d.f. $F_X(1)$ at 1 can be estimated by drawing a sample $\mathbf{x} = (x_1, \ldots, x_n)$ from the standard normal distribution and then computing $\widehat{F}_{\mathbf{x}}(1) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{(-\infty,1]}(x_i)$. This figure shows in yellow the result of taking longer and longer samples (with n ranging from 1 to 10^4) from the standard normal distribution and computing $\widehat{F}_{\mathbf{x}}(1)$. The value of $\widehat{F}_{\mathbf{x}}(1)$ converges to the true value $F_X(1) = 0.8413$ as $n \to \infty$, and the standard error (the blue curve is SE shifted upward by $\mathbb{E}[X]$) shrinks proportionally to $\frac{1}{\sqrt{n}}$.

Unexample 11.1.3. If $\mathbb{E}[k(\mathbf{x})]$ does not exist, then of course no sample will approach the nonexistant expected value. But the sample mean $\frac{1}{n}\sum_{i=1}^{n} k(\mathbf{x}_i)$ will still give you some value—it will prefer to lie to you and make something up rather than admit it doesn't know the answer. Consider, for example the Cauchy distribution, which has no expected value. Figure 11.2 shows the result of taking longer and longer samples $x_1, \ldots, x_n$ (with n ranging from 1 to 10^4) from the Cauchy distribution and computing $\frac{1}{n}\sum_{i=1}^{n} x_i$. Although the sample means seem to be sort of centered near 0, they do not get close to or converge to any particular value as $n \to \infty$.

11.1.2 Importance Sampling

If $f_X(\mathbf{x})$ never vanishes, then for any $h(\mathbf{x})$ we can set $k(\mathbf{x}) = \frac{h(\mathbf{x})}{f_X(\mathbf{x})}$ to calculate

$$\int_C h(\mathbf{x})\,d\mathbf{x} = \int_C \frac{h(\mathbf{x})}{f_X(\mathbf{x})} f_X(\mathbf{x})\,d\mathbf{x} = \mathbb{E}\left[\frac{h(\mathbf{x})}{f_X(\mathbf{x})}\right] \approx \frac{1}{n}\sum_{i=1}^{n} \frac{h(\mathbf{x}_i)}{f_X(\mathbf{x}_i)}, \tag{11.2}$$

where the $\mathbf{x}_i$ are drawn from the distribution of f_X.

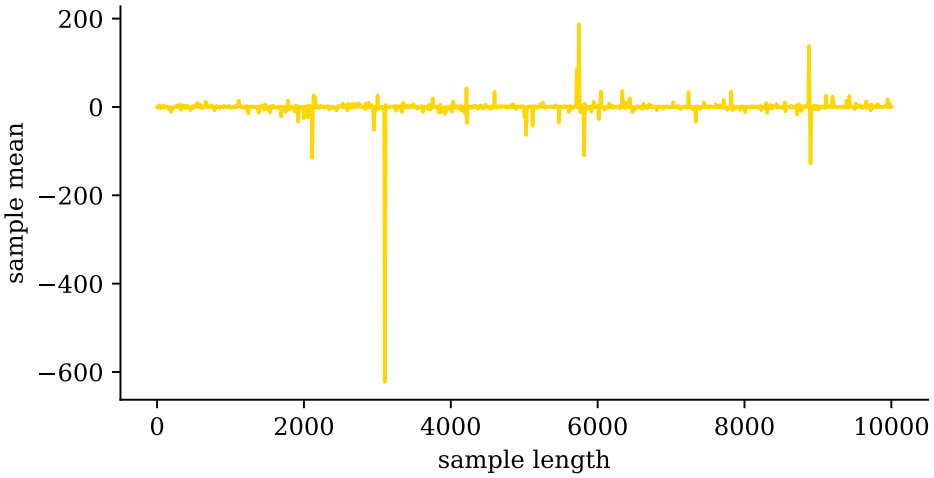


Figure 11.2: The Cauchy distribution has no expected value, but sampling from it and computing a sample mean still gives a value, which could easily be mistaken for an estimate of the nonexistant expected value of the distribution. This figure shows result of taking longer and longer samples $x_1, \ldots, x_n$ (with n ranging from 1 to 10^4) from the Cauchy distribution and computing the sample mean $\frac{1}{n}\sum_{i=1}^{n} x_i$. Although the sample means seem to be sort of centered near 0, they do not get close to or converge to any particular value as $n \to \infty$. .

It might seem most convenient to use a uniform distribution $f_X(x) \propto 1$ in this setting and compute

$$\int_C h(\mathbf{x})\, d\mathbf{x} \approx \frac{1}{n}\sum_{i=1}^{n} h(\mathbf{x}_i),$$

where the $\mathbf{x}_i$ are drawn uniformly on C. But the uniform distribution is only well defined on sets of finite volume, so if C is an infinite domain, we cannot use the uniform distribution. A natural choice when integrating over $\mathbb{R}^n$ is the normal distribution, in part because many good techniques have been developed for sampling from the standard normal distribution.

Another reason to use a nonuniform p.d.f. in this setting is to reduce the variance of the sample, and hence reduce the size of the standard error (11.1) of the Monte Carlo estimate. In the rare case that $f_X(\mathbf{x})$ is exactly proportional to $h(\mathbf{x})$, the ratio $\frac{h(\mathbf{x}_i)}{f_X(\mathbf{x}_i)}$ is constant for all $\mathbf{x}_i$, which means that $\widehat{s}^2 = 0$ and $SE = 0$. Choosing $f_X(\mathbf{x})$ close to $Mh(\mathbf{x})$ for some constant M makes the standard error of the Monte Carlo estimate small. This translates into computational efficiency gains, since fewer samples are required to achieve the desired accuracy.

Integrating by sampling from a distribution whose p.d.f. is nearly proportional to $h(\mathbf{x})$ is called *importance sampling*, because the sampling favors the more important regions that really contribute to the integral over the less important regions.

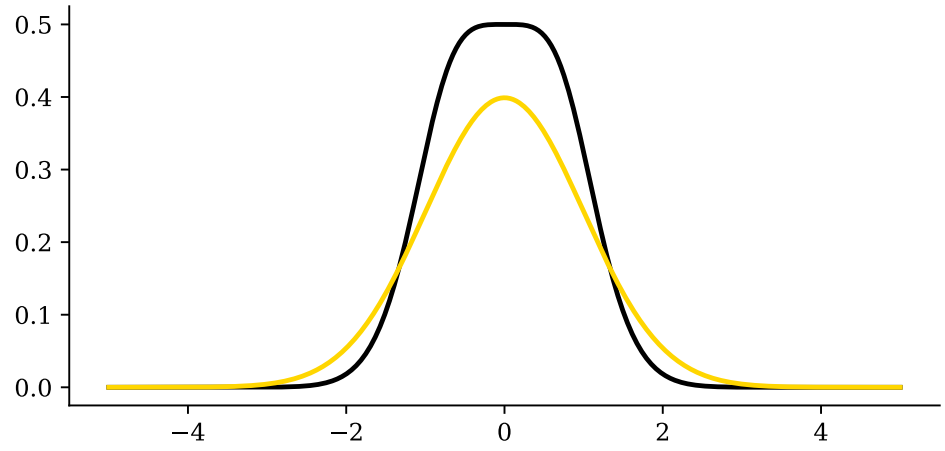


Figure 11.3: Plot of the function $h(x) = \frac{1}{2}\operatorname{sech}(x^2)$ (black) and the p.d.f. of the standard normal distribution (yellow). Because the p.d.f. of the standard normal has a shape similar to h, Monte Carlo estimation of the integral $\int_{-5}^{5} h(x)\,dx$ has improved accuracy when importance sampling is used with draws taken from the standard normal distribution instead of from the uniform distribution, as discussed in Example 11.1.4.

Example 11.1.4. Let $h(x) = \frac{1}{2}\operatorname{sech}(x^2)$, and consider the integral $\int_{-5}^{5} h(x)\,dx$; see Figure 11.3. Since this function has a maximum at the origin and then drops off rapidly as x moves away from the origin, it makes sense to try sampling from a normal distribution rather than from a uniform distribution. Let $X \sim \mathcal{N}(0,1)$ with p.d.f. $f_X(x)$. We can estimate $\int_{-5}^{5} h(x)\,dx$ as

$$\begin{aligned}\int_{-5}^{5} h(x)\,dx &= \int_{-\infty}^{\infty} \frac{\mathbb{1}_{[-5,5]}(x)h(x)}{f_X(x)} f_X(x)\,dx \\ &= \mathbb{E}\left[\frac{\mathbb{1}_{[-5,5]}(x)h(x)}{f_X(x)}\right] \approx \frac{1}{n}\sum_{i=1}^{n} \frac{\mathbb{1}_{[-5,5]}(x_i)h(x_i)}{f_X(x_i)},\end{aligned}$$

where $x_1, \ldots, x_n$ is a draw from X. When we ran this Monte Carlo estimate with $n = 10^6$, we found $\bar{y} = 1.18361$ and $SE \approx 0.00032$. This is better than the Monte Carlo estimate using the uniform distribution, which gave us $\bar{y} = 1.18558$ and $SE = 0.00183$.

Unfortunately, even if the overall shape of the sampled distribution $f_X(x)$ is similar to that of $h(x)$, if there are places where $f_X(x)$ is much smaller than $h(x)$, then for a draw $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ from those places, the ratio $\frac{h(x_i)}{f_X(x_i)}$ can be very large. This can cause the sample variance

$$\widehat{s}^2_{\mathbf{x}} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{h(x_i)}{f_X(x_i)} - \bar{y}\right)^2$$

to grow uncontrollably, especially if $\frac{h(x_i)}{f_X(x_i)}$ is large on an unbounded region, that is, in the tails of the distributions. Since the goal of importance sampling is to reduce the variance, rather than to let it grow, it is essential to choose a distribution $f_X(x)$ with tails that are large (fat), compared to the tails of the original function $h(x)$.

Unexample 11.1.5. Let $Y \sim \text{Gamma}(8,1)$. One straightforward Monte Carlo method for computing probabilities like $P(Y \geq c) = \int_c^\infty f_Y(x)\,dx$ is to use the fundamental bridge and sample from Y to get

$$P(Y \geq y) = \mathbb{E}[\mathbb{1}_{[c,\infty)}] \approx \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{[c,\infty)}(y_i),$$

where y_i is drawn from Y. But this can only work if we have a good way to sample from Y. If not, it seems natural to try to use a normal distribution $X \sim \mathcal{N}(8,8)$, since $\mathcal{N}(8,8)$ is a good approximation to Gamma$(8,1)$ (by the central limit theorem). Trying this gives

$$\int_c^\infty f_Y(t)\,dt = \int_{-\infty}^\infty \mathbb{1}_{[c,\infty)}(t)\frac{f_Y(t)}{f_X(t)}f_X(t)\,dt \approx \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{[c,\infty)}(x_i)\frac{f_Y(x_i)}{f_X(x_i)},$$

where $x_1, \ldots, x_n$ are drawn from $\mathcal{N}(8,8)$. Unfortunately, actually running the computation with $n = 10^4$, 10^5, and 10^6, with $c = 8$ we find completely nonsensical estimates 771.9, 1455.7, and 3226.4, respectively, with standard error 185.5, 206.6, and 810.8, respectively. So the estimates and the standard error are both failing to converge. This is because the tail of $\mathcal{N}(8,8)$ is substantially smaller than that of Gamma$(8,1)$. The ratio of the two is

$$\frac{\sqrt{2\pi}t^7e^{-t}}{e^{-\frac{(t-8)^2}{16}}},$$

which diverges to infinity as $t \to \infty$.

This can be remedied by sampling from a different distribution with a larger tail. For example, the distribution on $[0,\infty)$ with p.d.f. equal to $\frac{1}{(1+x)^2}$ will do, since

$$\lim_{x\to\infty}\frac{x^7e^{-x}}{7!/(1+x)^2} = \lim_{x\to\infty}\frac{x^7(1+x)^2}{7!e^x} = 0.$$

We show how to sample from $f_X(x) = \frac{1}{(1+x)^2}$ in Example 11.1.9.

11.1.3 Inversion Sampling

Sampling methods are all about generating random (or pseudorandom) samples from various distributions. Most modern computing systems have high-quality methods for generating uniformly distributed and normally distributed pseudorandom numbers, but in many cases (for example in importance sampling) one needs to sample from other distributions. A key tool for doing this is the following theorem.

Theorem 11.1.6 (Universality of the Uniform).

(*i*) *Let $F : (a, b) \to (0, 1)$ be bijective and increasing, with inverse F^{-1}. If $U \sim$ Uniform$(0, 1)$, then $X = F^{-1}(U)$ is a random variable with c.d.f. equal to F (with the obvious extension that $F(x) = 0$ for all $x \leq a$ and $F(x) = 1$ for all $x \geq b$).*

(*ii*) *If X is a random variable with a continuous c.d.f. F, then $Y = F(X)$ is a random variable with $Y \sim$ Uniform$((0, 1))$.*

Proof. (i) The function $F^{-1} : (0, 1) \to (a, b)$ exists and is both increasing and bijective because $F : (a, b) \to (0, 1)$ is increasing and bijective. We now show that F^{-1} is continuous by showing that for all $r, s \in (a, b)$ with $r < s$ the set $(F^{-1})^{-1}(r, s) = F(r, s)$ is equal to $(F(r), F(s))$, and hence is open in $(0, 1)$ (see Volume 1, Thm. 5.2.3). To see this, note that F is increasing, so for every $x \in (r, s)$ we have $F(r) < F(x) < F(s)$; and, hence, $F(x) \in (F(r), F(s))$ and $F(r, s) \subset (F(r), F(s))$. But bijectivity of F implies that for every $y \in (F(r), F(s))$, there exists $z \in (a, b)$ with $y = F(z)$, and the fact that F^{-1} is increasing implies that $z \in (r, s)$; hence, $F((r, s)) = (F(r), F(s))$, which implies that F^{-1} is continuous.

The rest of the proof is Exercise 11.2.

The proof of (ii) is also part of Exercise 11.2. □

This theorem is useful for sampling from a given distribution with c.d.f. equal to F, because whenever the inverse F^{-1} is known, we can generate a sample of the original distribution by taking a sample U of the uniform distribution and computing $F^{-1}(U)$. This is called *inversion sampling*.

Remark 11.1.7. Part (i) of the theorem also holds in the case that F maps to $[0, 1]$ instead of to $(0, 1)$. The proof is essentially identical to the one given here for $(0, 1)$.

Example 11.1.8. The distribution Beta$(a, 1)$ has p.d.f. equal to $f(x) = ax^{a-1}$, and thus its c.d.f. is $F(x) = x^a$, which is strictly increasing on the support $[0, 1]$ of the distribution, and hence bijective there. Its inverse $F^{-1}(u) = u^{1/a}$ is also continuous. Therefore, to sample from Beta$(a, 1)$ we may take a sample U from Uniform$([0, 1])$ and compute $U^{1/a}$.

Example 11.1.9. Let D be a distribution with p.d.f. equal to $f(x) = \frac{1}{(1+x)^2}$ defined on $[0, \infty)$. The c.d.f. is $F(x) = \int_0^x \frac{dt}{(1+t)^2} = \frac{x}{1+x}$ and its inverse is $F^{-1}(u) = \frac{u}{1-u}$. Therefore, given a sample $U \sim \text{Uniform}([0, 1))$, taking $\frac{U}{1-U}$ gives a sample from D.

11.1.4 Rejection Sampling

Unfortunately it is often impossible to compute a closed-form expression for the inverse of the c.d.f. of a distribution. Hence, inversion sampling is not always feasible. Another approach is *rejection sampling* which uses the following two main ideas:

(i) To sample from a continuous distribution P on $\mathbb{R}^d$, one can sample uniformly from the region in $\mathbb{R}^d \times \mathbb{R}$ between the hyperplane $\mathbb{R}^d \times \{0\}$ and the the graph of the p.d.f. $f_P(\mathbf{x})$ of P and then project each sample down to $\mathbb{R}^d$.

(ii) To sample uniformly from any region C, one can sample uniformly from a larger region containing C and throw away (reject) any samples that do not lie in C.

Idea (i) is illustrated in the left panel of Figure 11.4. The probability that sample $X \sim P$ lies in a region $C \subset \mathbb{R}^d$, is the volume $P(X \in C) = \int_C f_P(\mathbf{x})\, d\mathbf{x}$ under the p.d.f. This is the same as the probability that a uniformly chosen point with coordinates (s, t) will lie in the region below the graph of $f_P(\mathbf{x})$ and above the region $C \times \{0\}$.

Idea (ii) is the same idea used to approximate π in Volume 2 Example 7.1.1: to sample uniformly from a region C, sample uniformly from a larger region and discard (reject) any sample that does not lie in C. See the right panel of Figure 11.4 for an illustration.

The two ideas are combined in the following way. If we know how to sample from a distribution Q with known p.d.f. $f_Q(\mathbf{x})$ (call this the *proposal distribution*), and we want to sample instead from a distribution P (the *target distribution*) with known p.d.f. $f_P(\mathbf{x})$, we can do this if there exists an M such that $Mf_Q(\mathbf{x}) \geq f_P(\mathbf{x})$ for all $\mathbf{x}$. In this case, the region in the bounded above by the graph $y = Mf_Q(\mathbf{x})$ contains the region bounded above by the graph $y = f_P(\mathbf{x})$. In the right panel of Figure 11.4, the black curve is the graph of Mf_Q, the blue curve is the graph of f_P. Now draw $\mathbf{z}$ from the proposal distribution Q and u from $\text{Uniform}(0, Mf_Q(\mathbf{z}))$. The point $(\mathbf{z}, u) \in C \times [0, \infty)$ corresponds to a uniform draw from the region in the plane below the graph $y = Mf_Q(\mathbf{x})$. If $u \leq f_P(z)$, then $(\mathbf{z}, u)$ lies inside the region bounded above by $f_P(\mathbf{x})$ and hence the first coordinate $\mathbf{z}$ is a draw from X; otherwise reject $\mathbf{z}$ and repeat the process.

One minor adjustment is usually made to this process: instead of drawing u from $\text{Uniform}(0, Mf_Q(\mathbf{z}))$, it is traditional (and sometimes more efficient) to draw $\tilde{u}$ from $\text{Uniform}(0, 1)$ and then use the acceptance rule $\tilde{u} \leq \frac{f_P(\mathbf{z})}{Mf_Q(\mathbf{z})}$. Combining all these parts gives the rejection sampling algorithm:

(i) Choose M such that $Mf_Q(\mathbf{x}) \geq f_P(\mathbf{x})$ for all $\mathbf{x}$.

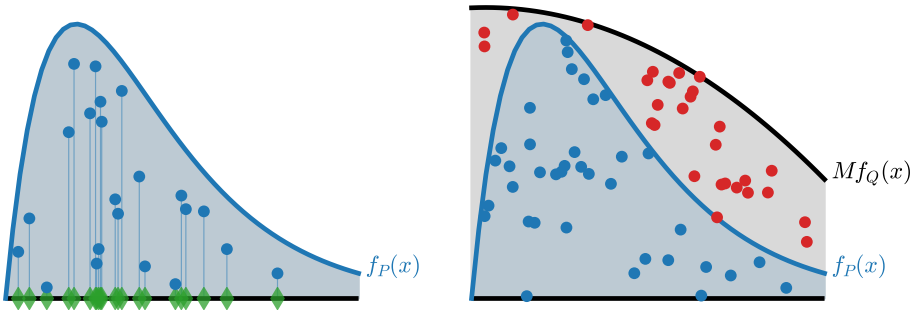


Figure 11.4: Illustration of the two main ideas behind rejection sampling. The first idea, illustrated in the left panel, is that sampling from a distribution is equivalent to sampling uniformly (blue dots) from the region between the graph of the p.d.f. and the x-axis and then projecting down to the x-axis (green diamonds). The second idea, illustrated in the right panel, is that sampling uniformly from one region (blue) can be accomplished by sampling uniformly from a larger region (gray and blue) and rejecting any samples (red dots) that do not lie inside the smaller region.

(ii) Draw $\mathbf{z}$ from Q and $\tilde{u}$ from Uniform$(0, 1)$.

(iii) If $\tilde{u} \leq \frac{f_P(\mathbf{z})}{Mf_Q(\mathbf{z})}$, then accept $\mathbf{z}$ as a draw from X; otherwise reject $\mathbf{z}$ and go back to (ii).

Remark 11.1.10. A given draw $\mathbf{z}$ has a probability $\frac{f_P(\mathbf{z})}{Mf_Q(\mathbf{z})}$ of being accepted, and one can show that the expected number of draws from Q needed to get one acceptable draw from P is proportional to M. Thus, it is generally best to choose a Q for which we can find a small M satisfying $f_P(\mathbf{x}) \leq Mf_Q(\mathbf{x})$ for all $\mathbf{x}$; and it is best to take the smallest M that satisfies the condition.

Example 11.1.11. Let P be a *truncated exponential* distribution on $[0, 20]$ with p.d.f. $f_P(x) = \frac{1}{Z}e^{-x}$, where $Z = \int_0^{20} e^{-x}\,dx$. Since $e^{-x} \leq 1$ for all $x \geq 0$, one possible choice of proposal distribution is the uniform distribution Q on $[0, 20]$ with $M = \frac{20}{Z}$, so that $f_P(x) \leq Mf_Q(x) = \frac{1}{Z}$ for all $x \in [0, 20]$. To use the method with this proposal distribution, draw z from Q and $\tilde{u}$ from Uniform($[0, 1]$) and reject any z whose corresponding u is greater than $\frac{f_P(z)}{Mf_Q(z)} = e^{-z}$. Implementing this and drawing one million times, we find that roughly 950,000 proposals are rejected and only 50,000 are accepted.

We can improve the efficiency of this rejection sampler by choosing a proposal distribution with a shape that is closer to that of the target. For example, it is easy to check that $e^{-x} \leq (1 + x)^{-1}$ for all $x \in [0, \infty)$, and $(1 + x)^{-1}$ has a shape that is similar to e^{-x}. Define a new proposal distribution R with

$$f_R(x) = \frac{1}{W}\frac{1}{1+x},$$

where $W = \int_0^{20} \frac{dx}{1+x} = \log(21)$. Setting $M = \frac{W}{Z}$ we have

$$f_P(x) \leq M f_R(x)$$

for all $x \in [0, 20]$. It is easy to sample from R using inversion sampling. We have

$$F_R(x) = \frac{1}{W} \int_0^x \frac{dt}{1+t} = \frac{1}{W} \log(1+x),$$

and

$$F_R^{-1}(v) = e^{Wv} - 1.$$

So the rejection sampling algorithm in this case consists of drawing both u and v from Uniform($[0, 1]$), letting $z = e^{Wv} - 1$, and rejecting z if $u > \frac{f_P(z)}{M f_R(z)} = \frac{1+z}{e^z}$. Implementing this and drawing one million times, we find that roughly $\frac{2}{3}$ of the proposals are rejected and $\frac{1}{3}$ are accepted—a better success rate than with the uniform proposal.

11.2 MCMC I: Gibbs Sampling

Sampling and Monte Carlo methods have revolutionized many areas of science and have been especially important in making Bayesian methods tractable. The classical methods described in the previous section have played a role in that, but one of the most important developments in sampling is the idea of *Markov Chain Monte Carlo (MCMC)* sampling methods. Although these methods were invented by physicists in the mid 1950s to predict the behavior of nuclear reactions, they were not properly appreciated nor widely adopted by statisticians until the end of the 20th century.

The idea of MCMC is that starting at any state $X_0 = x_0$ in an aperiodic irreducible Markov chain and moving in the chain gives a sequence $\mathbf{D} = x_0, x_1, \ldots, x_n$ in the state space, and irreducibility means that if n is large enough, then the distribution of $\mathbf{D}$ is close the stationary distribution of the Markov chain. Of course, the x_t are not i.i.d., because X_{t+1} depends on X_t for each t. Nevertheless, the full draw is fairly representative of the stationary distribution.

You might think that the main challenge in using this idea to draw from a given distribution $\boldsymbol{\pi}$ would be constructing an irreducible Markov chain whose stationary distribution is $\boldsymbol{\pi}$, but this is not actually very hard to do. There are several ways to do this. We'll start with the simplest, which is called *Gibbs sampling*, and then move to a more general construction we call *Rosenbluth–Teller–Metropolis–Hastings (RTMH)*.[41]

11.2.1 Gibbs Sampling

Gibbs sampling is an important and easy way to draw from a joint distribution $P(X, Y)$ if you know how to draw from each of the conditional distributions $P(X \,|\, Y)$ and $P(Y \,|\, X)$.

[41] The RTMH construction is due to Arianna Rosenbluth, Augusta Teller, Marshall Rosentbluth, Edward Teller, and Nicholas Metropolis and was later generalized by W. K. Hastings. It is often misleadingly just called the *Metropolis–Hastings* construction, omitting the names of four of the six original contributors.

Definition 11.2.1 (Gibbs Sampler). *Let X, Y be random variables, with joint distribution $P(X, Y)$ and joint support $\mathscr{S}$. Assume that for any $(x, y) \in \mathscr{S}$ we can draw from the conditional distributions $P(X \,|\, Y = y)$ and $P(Y \,|\, X = x)$. Define a Markov chain with state space $\mathscr{S}$ such that at any stage $\mathbf{s}_t = (x, y)$ the next stage $\mathbf{s}_{t+1}$ is given by the following procedure:*

(*i*) *Randomly (uniformly) choose one coordinate (X or Y) to change.*

(*ii*) *If X was chosen, then draw x' from the conditional distribution $P(X \,|\, Y = y)$ and let $\mathbf{s}_{t+1} = (x', y)$.*

(*iii*) *Otherwise (if Y was chosen) draw y' from the conditional distribution $P(Y \,|\, X = x)$ and let $\mathbf{s}_{t+1} = (x, y')$.*

The Gibbs sampler is a temporally homogeneous Markov chain because the transition probabilities are independent of time t and each stage depends only on the previous stage. Starting at any point $\mathbf{s}_0 = (x_0, y_0)$ and moving around this Markov chain gives a sequence $\mathbf{D} = (x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ of points in $\mathscr{S}$. We show below that the Gibbs sampler Markov chain is aperiodic and its stationary distribution is exactly the original joint distribution $P(X, Y)$; moreover, and under mild conditions, it is also irreducible. That means if n is large enough, then $\mathbf{D}$ is distributed approximately like a draw from the joint distribution $P(X, Y)$.

Example 11.2.2. Consider a bivariate normal distribution $(X, Y) \sim \mathscr{N}(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{\mu} = (\mu_x, \mu_y)$ and $\Sigma = \left[\begin{smallmatrix} \sigma_{00} & \sigma_{01} \\ \sigma_{01} & \sigma_{11} \end{smallmatrix}\right]$. Assuming that it is easy to draw from a univariate normal distribution, we can use the Gibbs sampler to draw from the joint, bivariate normal distribution. A little work completing the square in the bivariate p.d.f. shows that $X \,|\, (Y = y_t)$ is distributed like

$$\mathscr{N}\left(\mu_x + \frac{\sigma_{01}}{\sigma_{11}}(y_t - \mu_y), \sigma_{00} - \frac{\sigma_{01}^2}{\sigma_{11}}\right), \tag{11.3}$$

and $Y \,|\, (X = x_t)$ is distributed like

$$\mathscr{N}\left(\mu_y + \frac{\sigma_{01}}{\sigma_{00}}(x_t - \mu_x), \sigma_{11} - \frac{\sigma_{01}^2}{\sigma_{00}}\right), \tag{11.4}$$

To make things concrete, set $\boldsymbol{\mu} = (0, 0)$ and $\Sigma = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$. We'll start the Gibbs sampler at the (arbitrarily chosen) point $\mathbf{s}_0 = (x_0, y_0) = (1, 0)$ and take several steps to demonstrate the algorithm. Starting at $\mathbf{s}_0$, first choose one of the two coordinates randomly (uniformly). In this case the choice was y. Now draw from $Y \,|\, X = x_0$, which is the distribution $\mathscr{N}\left(\frac{x_0}{2}, \frac{3}{2}\right) = \mathscr{N}(0.0, 1.5)$; the p.d.f. of the conditional distribution is plotted in blue in the left panel of Figure 11.5. The result of this draw is $y = -1.51500746$, so the next stage of the Gibbs sampler is $\mathbf{s}_1 = (0.0, -1.51500746)$. This completes one step.

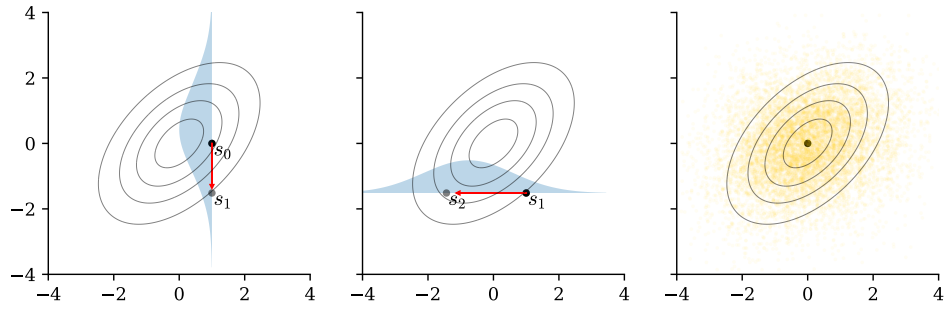


Figure 11.5: Steps of the Gibbs sampler for the bivariate normal of Example 11.2.2, starting at the point $\mathbf{s}_0 = (1, 0)$. The black ovals in each panel are contour lines for the joint p.d.f. The left panel shows the first step, where the (randomly chosen) x-coordinate is frozen, and y is drawn from the conditional distribution $Y \,\big|\, (X = x_0)$ (p.d.f. shown in blue), which is a univariate normal. This yields the point $\mathbf{s}_1 = (x_1, y_1) = (x_0, y_1)$. The center panel shows the next step, where the (randomly chosen) y-coordinate is frozen and x is drawn from the conditional distribution $X \,\big|\, (Y = y_1)$ (p.d.f. shown in blue), which is again univariate normal. The right panel shows the results (in yellow) of repeating this process 10,000 more times.

Repeating the process, again choose one coordinate randomly, this time it happens to be x, so draw from $X \,\big|\, Y = y_1$, which is the distribution $\mathcal{N}(\frac{y_1}{2}, \frac{3}{2}) = \mathcal{N}(-0.75750373, 1.5)$; the p.d.f. of this conditional distribution is plotted in blue in the center panel of Figure 11.5. The result of this draw is $x = -1.43594537$ so the next stage of the sampler is $\mathbf{s}_2 = (-1.43594537, -1.51500746)$.

The result of repeating this process for 10,000 more steps is illustrated in the right panel of Figure 11.5. Code for this example is given in Algorithm 11.1.

```
def gibbs_bvn(mu, Sig, s0, n):
    """Use a Gibbs sampler to sample from a bivariate normal
    distribution with mean mu and covariance Sig, starting at
    point s0 and running for n steps.
    """

    s = np.zeros((n, 2))  # initialize sample array
    s[0] = s0             # set the initial value
    for t in range(1, n):
        # choose a coordinate to update (0 or 1)
        coord = bernoulli.rvs(0.5)

        if coord == 0:
            # update x using formula for conditional
            s[t, 0] = normal(
                mu[0] + Sig[0,1]/Sig[1,1] * (s[t-1,1]-mu[1]),
                sqrt(Sig[0, 0] - Sig[0,1]**2 / Sig[1,1]))
            # keep y coordinate the same
            s[t,1] = s[t-1,1]
        else:
            # update y using formula for conditional
            s[t, 1] = normal(
                mu[1] + Sig[0,1]/Sig[0,0] * (s[t-1,0]-mu[0]),
                sqrt(Sig[1,1] - Sig[0,1]**2 / Sig[0,0]))
            # keep x coordinate the same
            s[t, 0] = s[t - 1, 0]
    return s
```

Algorithm 11.1: Python implementation of the Gibbs sampler draw from the bivariate normal distribution $\mathcal{N}(\texttt{mu}, \texttt{Sig})$, as described in Example 11.2.2. The updates are done with the explictly-computed formulae (11.3) and (11.4) for the conditional distributions $Y \,\big|\, (X = x_t)$ and $X \,\big|\, (Y = y_t)$. A plot of the output for the special case of Example 11.2.2 is plotted in the right panel of Figure 11.5.

Theorem 11.2.3. *The Gibbs sampler Markov chain of Definition 11.2.1, is aperiodic, and its stationary distribution is the original joint distribution $P(X, Y)$. Moreover, if $P(x, y) > 0$ whenever either $P(x) > 0$ or $P(y) > 0$, then the Gibbs sampler is irreducible.*

Proof. We first compute the transition probabilities in the Gibbs sampler Markov chain. Given $\mathbf{s} = (x, y), \mathbf{s}' = (x', y') \in \mathscr{S}$, to move from state $\mathbf{s}$ to $\mathbf{s}'$ in one step requires that either $x' = x$ or that $y' = y$, and also requires that the other coordinate is chosen (probability $\frac{1}{2}$), and then the correct value of other coordinate must be drawn. Said more precisely, the transition probability $Q_{\mathbf{s}'\mathbf{s}} = P(S_{t+1} = \mathbf{s}' \,|\, S_t = \mathbf{s})$ satisfies

$$Q_{\mathbf{s}'\mathbf{s}} = \begin{cases} \frac{1}{2}P(X = x' \,|\, Y = y) & \text{if } y' = y \text{ and } x' \neq x \\ \frac{1}{2}P(Y = y' \,|\, X = x) & \text{if } x' = x \text{ and } y' \neq y \\ \frac{1}{2}P(Y = y \,|\, X = x) + \frac{1}{2}P(X = x \,|\, Y = y) & \text{if } \mathbf{s}' = \mathbf{s} \\ 0 & \text{otherwise.} \end{cases}$$

To check the detailed balance condition (10.10) for the joint distribution, first let $\pi_{\mathbf{s}} = P((X, Y) = \mathbf{s}) = P((X, Y) = (x, y))$. Now consider the four possible cases:

(i) If $x' \neq x$ and $y' \neq y$, then $Q_{\mathbf{s}'\mathbf{s}}\pi_{\mathbf{s}} = 0 = Q\mathbf{s}\mathbf{s}'\pi_{\mathbf{s}'}$.

(ii) If $x' = x$ and $y' = y$, then $\mathbf{s}' = \mathbf{s}$ and $Q_{\mathbf{s}'\mathbf{s}}\pi_{\mathbf{s}} = Q_{\mathbf{s}\mathbf{s}}\pi_{\mathbf{s}} = Q_{\mathbf{s}\mathbf{s}'}\pi_{\mathbf{s}'}$.

(iii) If $x' \neq x$ and $y' = y$, then

$$\begin{aligned} Q_{\mathbf{s}'\mathbf{s}}\pi_{\mathbf{s}} &= \frac{1}{2}P(X = x' \,|\, Y = y)P((X, Y) = (x, y)) \\ &= \frac{1}{2}\frac{P((X, Y) = (x', y))}{P(Y = y)}P((X, Y) = (x, y)) \\ &= \frac{1}{2}P(X = x \,|\, Y = y)P((X, Y) = (x', y)) \\ &= Q_{\mathbf{s}\mathbf{s}'}\pi_{\mathbf{s}'}. \end{aligned}$$

(iv) The case where $x' = x$ and $y' \neq y$ is essentially identical to (iii).

Thus detailed balance holds for $\boldsymbol{\pi}$; and therefore $P(X, Y)$ is stationary for the Gibbs sampler.

For any $\mathbf{s} = (x, y) \in \mathscr{S}$, we know that $P(x, y) > 0$, and thus $P(x \,|\, y) > 0$. This implies that $Q_{\mathbf{s}\mathbf{s}} > 0$, so the Markov chain is aperiodic.

Finally, any state $\mathbf{s}' = (x', y') \in \mathscr{S}$ is reachable from any $\mathbf{s} = (x, y) \in \mathscr{S}$ in two steps:

$$\mathbf{s} = (x, y) \to \mathbf{s}^* = (x', y) \to \mathbf{s}' = (x', y').$$

The first step has nonzero probability because $P(x, y) > 0$ implies that $P(y) > 0$, which, by hypothesis, implies that $P(x', y) > 0$; hence

$$Q_{\mathbf{s}^*\mathbf{s}} = \begin{cases} \frac{1}{2}P(X = x' \,|\, Y = y) > 0 & \text{if } x' \neq x \\ \frac{1}{2}P(Y = y \,|\, X = x) + \frac{1}{2}P(X = x \,|\, Y = y) > 0 & \text{if } x' = x. \end{cases}$$

The second step has nonzero probability because

$$Q_{\mathbf{s}'\mathbf{s}^*} = \begin{cases} \frac{1}{2}P(Y = y' \,|\, X = x') > 0 & \text{if } y' \neq y \\ \frac{1}{2}P(Y = y \,|\, X = x') + \frac{1}{2}P(X = x' \,|\, Y = y) > 0 & \text{if } y' = y. \end{cases}$$

Hence any state is accessible from any other state, and thus the Markov chain is irreducible. ∎

Nota Bene 11.2.4. Although we say that samples from the Gibbs sampler *approach* or *converge to* the target distribution $\boldsymbol{\pi}$, the samples themselves do not converge to anything—they move throughout the state space $\mathscr{S}$ and never settle down near one location. Instead the Markov chain converges *in distribution* to $\boldsymbol{\pi}$, so the distribution of the samples $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n$ (that is, the proportion of samples equal to each state) approaches the distribution $\boldsymbol{\pi}$ as $n \to \infty$.

Remark 11.2.5. The condition that $P(x, y) > 0$ whenever either $P(x) > 0$ or $P(y) > 0$ is stronger than we need to prove irreducibility. But a weaker condition would be more complicated to state and use, and this stronger condition is good enough for now.

Remark 11.2.6. Instead of selecting the coordinate i randomly, one can also alternate between the coordinates deterministically, that is, proceed in order from x to y and repeat.

11.2.2 Continuous Gibbs Sampling

As you might have guessed from Example 11.2.2, although the proofs given so far are only for discrete random variables, the Gibbs sampler construction is not limited to discrete distributions. It can also be used for continuous distributions if probabilities are replaced with densities in the usual way. Much of what we have proved about Markov chains and the Gibbs construction still holds in this continuous case, but the details of the proofs would take us beyond the scope of this text.

Example 11.2.7. Assume that X is supported on $\mathbb{R}^{>0}$ and $P(x \mid a, b) = abe^{-abx}$. Suppose that the parameters $a > 0$ and $b > 0$ have prior $P(a, b) \propto e^{-a-b}$. Given data $\mathbf{D} = \{x_1, \ldots, x_n\}$ we want to sample from the joint posterior for $(a, b \,|\, \mathbf{D})$. The conditional distribution $P(a \,|\, \mathbf{D}, b)$ is

$$\begin{aligned} P(a \,|\, \mathbf{D}, b) &= \frac{P(a, b, \mathbf{D})}{P(\mathbf{D}, b)} \\ &= \frac{P(\mathbf{D} \,|\, a, b) P(a, b)}{P(\mathbf{D}, b)} \\ &\propto P(\mathbf{D} \,|\, a, b) P(a, b), \end{aligned}$$

where the final $\propto$ follows because $P(\mathbf{D}, b)$ is independent of a. This gives

$$\begin{aligned} P(a \,|\, \mathbf{D}, b) &\propto \left(\prod_{i=1}^{n} abe^{-abx_i} \right) e^{-a-b} \\ &\propto a^n e^{-a(1+b\sum_{i=1}^{n} x_i)}. \end{aligned}$$

Note that both $P(x \,|\, a, b)$ and $P(a, b)$ are symmetric in a and b, thus swapping a and b in the previous computation gives the corresponding result for $P(b \,|\, \mathbf{D}, a)$. These are both proportional to gamma distributions, specifically

$$P(a \,|\, \mathbf{D}, b) \sim \text{Gamma}(1 + n, 1 + b \sum_{i=1}^{n} x_i)$$
$$P(b \,|\, \mathbf{D}, a) \sim \text{Gamma}(1 + n, 1 + a \sum_{i=1}^{n} x_i).$$

Both of these are easy to sample from, so we can use Gibbs sampling to sample from the joint posterior distribution $P(a, b \,|\, \mathbf{D})$. We could randomly choose a coordinate to sample from, but for simplicity we just alternate between a and b. The procedure begins with a random choice of $\mathbf{s}_0 = (a_0, b_0)$ and then for each $t \geq 0$ proceeds as

(i) Draw a_{t+1} from Gamma$(1 + n, 1 + b_t \sum_{i=1}^{n} x_i)$

(ii) Draw b_{t+1} from Gamma$(1 + n, 1 + a_{t+1} \sum_{i=1}^{n} x_i)$

(iii) Set $\mathbf{s}_{t+1} = (a_{t+1}, b_{t+1})$.

(iv) Increment t (`t+=1`) and repeat.

The resulting sequence $\mathbf{s}_0, \mathbf{s}_1, \ldots$ converges in distribution to the joint posterior $P(a, b \,|\, \mathbf{D})$, meaning that as $t \to \infty$ the distribution of the states $\mathbf{s}_0, \mathbf{s}_1, \ldots$ approaches the stationary distribution $P(a, b \,|\, \mathbf{D})$.

Example 11.2.8. Consider data $\mathbf{D} = \{x_1, \ldots, x_n\}$ drawn from a normal distribution $X \sim \mathcal{N}(\mu, \sigma^2)$. For convenience (to make all the algebra much nicer) define the *precision* of the distribution to be $\tau = \frac{1}{\sigma^2}$. For fixed precision τ, it can be shown that the prior $\mu \,|\, \tau \sim \mathcal{N}(\nu, \frac{1}{\lambda})$ (where λ is the precision of the prior) is conjugate to the likelihood with a corresponding posterior of the form

$$\mu \,|\, \mathbf{D}, \tau \sim \mathcal{N}\left(\frac{\nu\lambda + \tau \sum_{i=1}^{n} x_i}{\lambda + n\tau}, \frac{1}{\lambda + n\tau}\right).$$

Here $\lambda + n\tau$ is the precision (not the variance) of the posterior. Similarly, it can be shown that for fixed mean μ and unknown precision τ, the gamma prior $\tau \,|\, \mu \sim \text{Gamma}(a, b)$ is conjugate with posterior

$$\tau \,|\, \mathbf{D}, \mu \sim \text{Gamma}\left(a + \frac{n}{2}, b + \frac{1}{2} \sum_{i=1}^{n} (x_i - \mu)^2\right).$$

Taking the product $\mathcal{N}(\nu, \frac{1}{\lambda})\,\text{Gamma}(a, b)$ as the joint prior for (μ, τ), we can use Gibbs sampling to sample from the joint posterior distribution $P\left(\mu, \tau \,|\, \mathbf{D}\right)$ as follows. Choose an arbitary $s_0 = (\mu_0, \tau_0)$. For each $t \geq 0$

(i) Draw μ_{t+1} from $P(\mu \,|\, \mathbf{D}, \tau_t)$

(ii) Draw τ_{t+1} from $P\left(\tau \,|\, \mathbf{D}, \mu_{t+1}\right)$

(iii) Set $s_{t+1} = (\mu_{t+1}, \tau_{t+1})$.

(iv) Increment t (`t+=1`) and repeat.

The samples from the joint posterior are $s_0, s_1, \ldots$.

11.2.3 Using Gibbs to Draw from $P(X)$ Instead of $P(X, Y)$.

In some circumstances we would like to draw from a distribution X that is hard to draw from directly, but there is another related random variable Y, and we know how to draw from X given Y and vice versa. We can use Gibbs sampling to draw from the joint distribution of X and Y and then discarding the Y values gives a draw from X alone.

The general strategy is this: Assume you want to draw from one distribution X and the problem would be much easier if you knew some other random variable Y (you can draw from $P(X \,|\, Y)$). If $P(Y \,|\, X)$ is also not hard to sample from, then you can use Gibbs sampling to draw from $P(X, Y)$ and then ignore the Y part of each sample to get a draw from $P(X)$.

Example 11.2.9. Assume we have a free-range hen and rooster. The number N of eggs laid by the hen in a given period is distributed as Poisson(λ) for a known value of λ. Each egg hatches with probability Θ, where $\Theta \sim \text{Beta}(a, b)$, so the number X of hatchlings is distributed as $X \sim \text{Poisson}(\Theta\lambda)$. We cannot observe N or Θ directly, because the hen is good at hiding her eggs—we only see the number X of hatchlings.

We would like to find $\mathbb{E}[\Theta \,|\, X = x]$. To do this, note that Bayes' rule gives

$$f(\theta \,|\, x) \propto P(x \,|\, \theta) P(\theta) \propto \mathbf{e}^{-\lambda\theta} (\lambda\theta)^x \theta^{a-1} (1-\theta)^{b-1}.$$

This distribution is not a familiar one and may not be easy to work with analytically, but we can approximate $\mathbb{E}[\theta \,|\, x]$ by sampling.

The problem would be much easier if we knew the value of N, because if $N = n$, then

$$P(x \,|\, n, \theta) \sim \text{Binomial}(n, \theta) = \binom{n}{x} \theta^x (1-\theta)^{n-x}.$$

and

$$\begin{aligned} P(\theta \,|\, x, n) &\propto P(x \,|\, n, \theta) P(\theta \,|\, n) \\ &= \text{Binomial}(n, \theta)\, \text{Beta}(a, b) \\ &\propto \text{Beta}(a + x, b + n - x), \end{aligned}$$

since the beta distribution is a conjugate prior for the binomial likelihood.

We don't know N, but we can describe N in terms of X and the number U of unhatched eggs as $N = X + U$, where $U \sim \text{Poisson}(\lambda(1-\theta))$. Thus if θ is given, then we can sample from $P(N \,|\, x, \theta)$ by taking $n \sim x + \text{Poisson}(\lambda(1-\theta))$.

Gibbs sampling now gives a sample of $P(\theta, n \mid x)$ by starting with an arbitrary $\mathbf{s}_0 = (\theta_0, n_0)$ and for each time t

(i) Draw θ_{t+1} from $\text{Beta}(a + x, b + n_t - x)$;

(ii) Draw u from $\text{Poisson}(\lambda(1 - \theta_{t+1}))$ and set $n_{t+1} = x + u$;

(iii) Set $\mathbf{s}_{t+1} = (\theta_{t+1}, n_{t+1})$;

(iv) Increment t and repeat.

Once the sample is collected, ignoring all of the the values of n_t gives a draw $\theta_0, \ldots, \theta_T$ from $P(\theta \,|\, x)$. From this we compute

$$\mathbb{E}[\Theta] \approx \frac{1}{T} \sum_{i=1}^{\mathsf{T}} \theta_t.$$

11.2.4 Gibbs Sampling for More Dimensions

Gibbs sampling need not be limited to only two random variables. It extends to the general case of a joint distribution $(X_1, \ldots, X_d)$ when each of the conditional distributions $(X_1 \,|\, X_2, \ldots, X_d)$, $(X_2 \,|\, X_1, X_3, \ldots, X_d)$, $\ldots (X_d \,|\, X_1, X_2, \ldots, X_{d-1})$ is easy to sample from.

To describe the general Gibbs sampler we first need some notation. For a random variable $X = (X_1, \ldots, X_d)$ supported on $\mathbb{R}^d$, and for any $i \in \{1, \ldots, d\}$ write $P(x_i \mid x_1, \ldots, \widehat{x_i}, \ldots, x_d)$ to denote the conditional probability $P(X_i = x_i \mid X_1 = x_1, \ldots, \widehat{X_i}, \ldots, X_d = x_d)$, where $\widehat{X_i}$ denotes that the ith term is omitted from the list. Similarly, write $f_{(X_i \mid x_1, \ldots, \widehat{x_i}, \ldots, x_d)}$ to denote the distribution of the X_i coordinate given the remaining coordinates $X_1 = x_1, \ldots, \widehat{X_i}, \ldots, X_d = x_d$.

The general Gibbs sampler is as follows. From state $\mathbf{s} = (x_1, \ldots, x_d)$ choose $i \in \{1, \ldots, d\}$ at random (uniformly) and draw x_i' from $f_{(X_i \mid x_1, \ldots, \widehat{x_i}, \ldots, x_d)}$. Replace x_i by x_i' in the old state s to get the new state $\mathbf{s}' = (x_1, \ldots, x_i', \ldots, x_d)$. Thus

$$Q_{\mathbf{s}'\mathbf{s}} = P(\mathbf{s}' \,|\, \mathbf{s}) = \frac{1}{d} P(x_i' \,|\, x_1, \ldots, \widehat{x_i'}, \ldots, x_d). \tag{11.5}$$

The proof that this is an irreducible aperiodic Markov chain is essentially similar to the two-variable case; see Exercise 11.7.

Remark 11.2.10. As in the two-dimensional case, one can one can sweep through the coordinates deterministically, that is, proceed in order from 1 to d and repeat, instead of selecting the coordinate i randomly.

11.3 MCMC II: Rosenbluth–Teller–Metropolis–Hastings

Gibbs sampling constructs a Markov chain whose stationary distribution is the joint distribution of several random variables, and thus allows sampling from the joint distribution. But Gibbs does not apply when a distribution is not naturally a joint distribution. In this section we discuss the *Rosenbluth–Teller–Metropolis–Hastings (RTMH)* construction, which works for more a general distribution $\boldsymbol{\pi}$; that is, RTMH is a method of constructing an irreducible and aperiodic Markov chain whose stationary distribution is $\boldsymbol{\pi}$.

11.3.1 RTMH: Make a New Markov Chain From an Old

Definition 11.3.1 (RTMH construction). *Assume that $\boldsymbol{\pi}$ is a distribution (the* target distribution*) on a set S. Write π_s for the probability of s (or the density of s, if $\boldsymbol{\pi}$ is continuous) under this distribution. Assume that $\pi_s > 0$ for every $s \in S$ (if not, remove all s from S that have $\pi_s = 0$). Let X be a Markov chain with state space S, having transition probabilities $q^X_{s's} = P(X_{t+1} = s' \,|\, X_t = s)$. We call X the* proposal Markov Chain*. Define a new Markov chain Y on S as follows.*

(*i*) *For $Y_t = s$ choose a* proposal *$s' \in S$ using the Markov chain X; that is, set $X_t = s$ and draw from $X_{t+1} \,|\, X_t = s$ to get a proposal s' (with probability $q^X_{s's}$).*

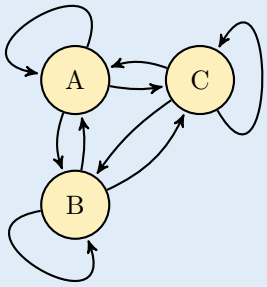
(*ii*) *If s' is proposed, then randomly, with probability*

$$a_{s's} = \min\left(\frac{\pi_{s'} q^X_{ss'}}{\pi_s q^X_{s's}}, 1\right), \tag{11.6}$$

set $Y_{t+1} = s'$ (accept the proposal), and otherwise set $Y_{t+1} = s$ (reject the proposal). In other words, choose $B \sim$ Bernoulli$(a_{s's})$ and accept the proposal whenever $B = 1$.

We call the resulting Markov chain Y the Rosenbluth–Teller–Metropolis–Hastings (RTMH) *construction.*

Example 11.3.2. Consider the set $S = \{A, B, C\}$ with target distribution $\boldsymbol{\pi} = (\pi_A, \pi_B, \pi_C) = (\frac{1}{2}, \frac{1}{3}, \frac{1}{6})$. We'll apply the Metropolis–Hastings construction with the proposals coming from the Markov chain X that has the following transition diagram



where every outgoing edge has equal probability, so

$$q^X_{AA} = q^X_{BA} = q^X_{CA} = \frac{1}{3}$$
$$q^X_{AB} = q^X_{BB} = q^X_{CB} = \frac{1}{3}$$
$$q^X_{AC} = q^X_{BC} = q^X_{CC} = \frac{1}{3}.$$

If $y_0 = A$, then to choose y_1, first take a proposal from X, starting in state A, that is, draw $x_1 \,|\, x_0 = A$. Assume for our example that $x_1 = B$ (this happens with probability $\frac{1}{3}$). To decide whether to accept the proposal or not, compute

$$a_{BA} = \min\left(\frac{\pi_B q^X_{AB}}{\pi_A q^X_{BA}}, 1\right) = \frac{\frac{1}{3}\frac{1}{3}}{\frac{1}{2}\frac{1}{3}} = \frac{2}{3}.$$

Accept the proposal with probability $a_{BA} = \frac{2}{3}$; that is, draw $B \sim \text{Bernoulli}(\frac{2}{3})$, and if $B = 1$, then set $y_1 = B$, and otherwise set $y_1 = y_0 = A$.

Proposition 11.3.3. *The RTMH construction Y is a Markov chain that satisfies the detailed balance equations for $\boldsymbol{\pi}$ (with $\pi_s > 0$ for all $s \in S$).*

Proof. That Y is a Markov chain follows immediately from the definition. The proof of the detailed balance equations (10.10) are a direct calculation with two cases: when $\pi_s q^X_{s's} \geq \pi_{s'} q^X_{ss'}$ and when $\pi_s q^X_{s's} < \pi_{s'} q^X_{ss'}$. The details of this calculation are Exercise 11.8. □

Theorem 11.3.4. *Let Y be the Markov chain resulting from the RTMH construction with proposal distribution X on a finite state space S. Assume that X is irreducible and aperiodic. If every pair $s, s' \in S$ has the property that the X-transition probability $q^X_{s's} > 0$ if and only if $q^X_{ss'} > 0$, then Y is irreducible and aperiodic.*

Before we prove the theorem, we need a lemma.

Lemma 11.3.5. *Under the conditions of the theorem, if a pair $s, s' \in S$ satisfies $q^X_{s's} > 0$, then the transition probability $p^Y_{s's} = P(Y_{t+1} = s' \,|\, Y_t = s)$ is not zero.*

Proof. The transition probability $p^Y s's$ satisfies

$$p^Y s's = P(X_{t+1} = s' \,|\, X_t = s) a_{s's} = q^X_{s's} a_{s's} = \begin{cases} \frac{\pi_{s'} q^X_{ss'}}{\pi_s} & \text{if } \pi_s q^X_{s's} \geq \pi_{s'} q^X_{ss'} \\ q^X_{s's} & \text{otherwise.} \end{cases}$$

But since $q^X_{s's}$, $q^X_{ss'}$, π_s, and $\pi_{s'}$ are all nonzero, we must also have $p^Y s's > 0$. ∎

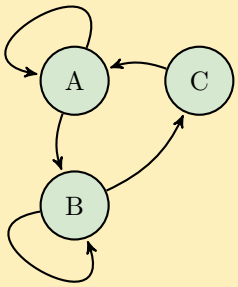
Proof. (of Theorem 11.3.4) To see that Y is irreducible whenever the given conditions on X hold, consider any pair of states $s, s' \in S$. Since X is irreducible, there must be a walk $s = s_0 \to s_1 \to \cdots \to s_{k-1} \to s_k = s'$ with $q^X_{s_i s_{i-1}} > 0$ for all $i \in \{1, \ldots, k\}$. We have

$$\begin{aligned} P(X_{t+k} = s' \,|\, X_t = s) &\geq \prod_{i=1}^{k} P(X_{t+i} = s_i \,|\, X_{t+i-1} = s_{i-1}) \\ &= \prod_{i=1}^{k} p^Y s_i s_{i-1} > 0, \end{aligned}$$

where the last inequality follows from the lemma. This shows that Y is irreducible and that for every walk in X there is a corresponding walk in Y with the same sequence of nodes and edges (but with transition probabilities $p^Y s_i s_{i-1}$ instead of $q^X_{s_i s_{i-1}}$).

Since X is aperiodic, then for every state $s \in S$ there exists a collection of closed walks in X from s to itself whose lengths are relatively prime. But the corresponding walks in Y have the same lengths, and thus Y is also aperiodic. ∎

Unexample 11.3.6. Consider the set $S = \{A, B, C\}$ with target distribution $\boldsymbol{\pi} = (\pi_A, \pi_B, \pi_C) = (\frac{1}{2}, \frac{1}{3}, \frac{1}{6})$. We'll try to apply the RTMH construction with the proposals coming from the Markov chain with the following transition diagram



where every outgoing edge has equal probability, so

$$\begin{aligned} q^X_{AA} &= q^X_{BA} = \frac{1}{2} \\ q^X_{BB} &= q^X_{CB} = \frac{1}{2} \\ q^X_{AC} &= 1 \end{aligned}$$

and all other q^X_{ij} are zero. It is straightforward to verify that X is aperiodic and irreducible. But for any two states $s' \neq s$, the transition probability $p^Y s's = P(Y_{t+1} = s' \,\big|\, Y_t = s)$ in the RTMH construction vanishes because either $q^X_{s's} = 0$, which means that s' is never proposed, or $q^X_{ss'} = 0$, which implies that

$$p^Y s's = q^X_{s's} a_{s's} = q^X_{s's} \min\left(\frac{\pi'_s q^X_{ss'}}{\pi_s q^X_{s's}}, 1\right) = 0.$$

This shows that the chain Y is not irreducible—it is impossible to move from any state to any other state.

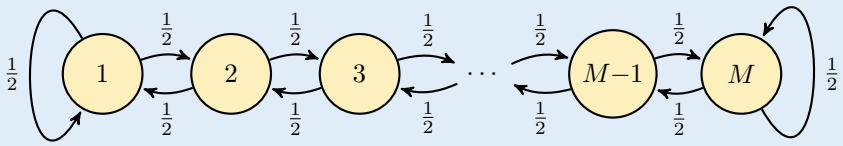
Nota Bene 11.3.7. Although the RTMH construction works, in theory, for any proposal X satisfying the condition of the theorem, the choice of the proposal has a large impact on how fast the RTMH construction converges to the stationary distribution. If the $a_{s's}$ are small, most of the proposals will be rejected, there will be little movement in Y, and it will take a long time for the samples to become distributed according to the stationary distribution $\boldsymbol{\pi}$. The $a_{s's}$ are larger when $\boldsymbol{\pi}$ is closer to being stationary for the proposal X. Thus the RTMH algorithm approaches the target distribution $\boldsymbol{\pi}$ more rapidly when the stationary distribution for the proposal distribution X is closer to the desired distribution $\boldsymbol{\pi}$.

Nota Bene 11.3.8. Although we say that samples from the RTMH construction *approach* or *converge to* the target distribution $\boldsymbol{\pi}$, the samples themselves do not converge to anything—they move throughout the state space S and never settle down near one location. Instead the Markov chain Y converges *in distribution* to $\boldsymbol{\pi}$, so the distribution of the samples from Y approaches the distribution $\boldsymbol{\pi}$.

Example 11.3.9. Given the set $S = \{1, \dots, M\}$ and a choice of $\alpha > 0$ the *Zipf distribution* on S satisfies

$$P(X = x) = \frac{x^{-\alpha}}{Z},$$

where $Z = \sum_{k=1}^{M} k^{-\alpha}$. This distribution is often used to describe word frequencies in linguistics. To use RTMH to sample from this distribution, we must first choose a proposal Markov chain. One choice is the symmetric random walk on S, which has the following transition diagram



The RTMH construction produces a chain $Y_0, Y_1, \dots$ by starting in any state $Y_0 = X_0 = x_0 \in S$ and at each step $Y_t = s$ takes a proposal $X_{t+1} = s' \,\big|\, X_t = s$. For any s' that can be generated this way, we must have $P(X_{t+1} = s' \,\big|\, X_t = s) = \frac{1}{2}$ (otherwise the transition probability is 0). The acceptance probability is

$$\begin{aligned} a_{s',s} &= \min\left(\frac{\pi_{s'} q^X_{s,s'}}{\pi_s q^X_{s',s}}, 1\right) \\ &= \min\left(\frac{\frac{(s')^{-\alpha}}{Z} \cancel{\frac{1}{2}}}{\frac{s^{-\alpha}}{Z} \cancel{\frac{1}{2}}}, 1\right) \\ &= \min\left(\frac{s^\alpha}{(s')^\alpha}, 1\right). \end{aligned}$$

This is implemented in Python in Algorithm 11.2. The results of using this algorithm for $M = 30$ and $M = 100$ whith $\alpha = 1.1$ and $n = 10^4$ are plotted in Figure 11.6.

```
import numpy as np
from numpy.random import randint, rand

def Zipf(M, n=10**4,alpha=2):
    """Draw a sample of length `n` from the Zipf distribution
    with exponent `alpha` on the set {1,2,...,M} using
    Metropolis-Hastings and the symmetric random-walk proposal.
    """

    def Bernoulli(theta):       # Draw from Bernoulli(theta)
        return rand() <= theta  # rand() is uniform on [0,1]

    t = 0                       # Initialize t
    y = np.empty(n)             # Initialize y

    y[0]= randint(1,M+1)        # Start at a random location

    while t < n-1:
        proposal = min(max(1, y[t] + (-1)**Bernoulli(0.5)), M)
        accept_prob = y[t]**alpha / proposal**alpha

        if Bernoulli(accept_prob):  # Proposal accepted
            y[t+1] = proposal
        else:                       # Proposal rejected
            y[t+1] = y[t]

        t += 1
    return y
```

Algorithm 11.2: Python implementation of the RTMH algorithm to sample from the Zipf distribution using the symmetric random walk proposal, as descibed in Example 11.3.9.

Nota Bene 11.3.10. The normalizing constant Z cancels in the computations in Example 11.3.9, so the constant need never be computed. This always happens with the RTMH construction. We **never** need the normalizing constant of the target distribution—it is always enough to know a function that is proportional to the p.m.f..

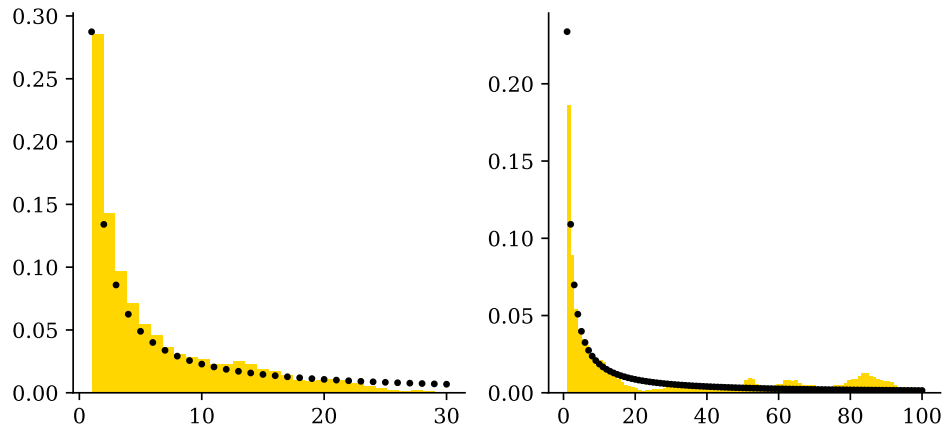


Figure 11.6: Histogram (yellow) of the results of using RTMH for $n = 10^4$ steps, as given in Algorithm 11.2, from the Zipf distribution (p.m.f. plotted in black) with $\alpha = 1.1$ on the set $S = \{1, \ldots, M\}$ with $M = 30$ (left panel) and $M = 100$ (right panel).

11.3.2 Continuous RTMH

As with Gibbs, the RTMH construction is not limited to discrete distributions. It can also be used for continuous distributions if probabilities are replaced with densities in the usual way. Thus the transition probability $q^X_{s',s}$ is replaced with a transition p.d.f. $f_{X_{t+1}|X_t=s}(s')$ in the equation (11.6) for the acceptance probability. Much of what we have proved about Markov chains and the RTMH constructions still hold in this continuous case, but the details of the proofs would take us beyond the scope of this text.

Example 11.3.11. The beta distribution Beta(a, b) has p.d.f.

$$f_B(x) \propto x^{a-1}(1-x)^{b-1}.$$

To use the RTMH construction to sample from this distribution, begin with the state space $S = (0, 1)$. Assuming we know how to sample from the uniform distribution, we can let the proposal Markov chain be $X_{t+1} \sim \text{Uniform}(0, 1)$ regardless of the value of X_t, that is $f_{X_{t+1}|X_t=x}(x') = \mathbb{1}_{(0,1)}(x')$. Given $Y_t = x$ the RTMH algorithm in this case is

(i) Choose a proposal $X_{t+1} = x'$ from Uniform$(0, 1)$.

(ii) Compute the acceptance probability

$$a_{x',x} = \min\left(\frac{f_{X_{t+1}|x'}(x) f_B(x')}{f_{X_{t+1}|x}(x') f_B(x)}, 1\right) = \min\left(\frac{x^{a-1}(1-x)^{b-1}}{(x')^{a-1}(1-x')^{b-1}}, 1\right).$$

(iii) Accept the proposal with probability $a_{x',x}$ and set $Y_{t+1} = x'$, otherwise reject and set $Y_{t+1} = x$.

11.3.3 Tuning MCMC parameters

The plots in Figure 11.6 show more samples in the tail (far right) than expected. This can happen whenever one has not taken enough steps in the chain, especially when the starting point (here `y[0]`) is not near typical points of the distribution. In these two plots in Figure 11.6 the starting points were 23 and 70, respectively. The RTMH construction doesn't necessarily move very quickly away from a starting point, so if it starts in a location that is improbable (as measured by the target, stationary distribution) it produces many unlikely samples before arriving in more probable locations. To make the initial, improbable samples have the correct proportion, according to the target, stationary distribution, we may have to run the chain for a long time.

Burn-In

One method for getting a more representative sample without running the chain for a lot longer is to start the sampling in a state near the expected value of the target distribution. In the case of the Zipf distribution, we can calculate the mean explicitly as $\mu = \frac{1}{Z}\sum_{k=1}^{M} k^{\alpha-1}$. But in many settings the mean of the target distribution is not known. Indeed we usually sample in order to approximate the mean. In these situations it is common to let the chain run for a while, say b steps, and then start the chain at the resulting state; that is, discard steps 0 through $b-1$ and only keep samples from state b and onward. The idea is that if b is large enough, then the bth state should be more representative of the stationary distribution than the original starting place. The initial b steps are often called the *burn-in period*.

Approximating Independence

The samples from any MCMC method are not independent, but keeping only every kth term $Y_k, Y_{2k}, \ldots$, can approximate independence if k is large enough. Calculating the correlation coefficient for pairs (Y_{t+k}, Y_{t+2k}) gives one measure of how far these subsamples are from being independent.

Number of Samples

There is no clear rule that will always tell you how many samples to take, but here are some things to think about.

In a finite distribution if you want to see a given state s occurring approximately m times in your sample, and if the probability of s in the stationary distribution is π_s, and if the samples are approximately independent, then you would expect to need at least $\frac{m}{\pi_s}$ total samples. But the samples in MCMC are not independent.

A bad choice of proposal means many unaccepted proposals, and that means that a given state is kept for multiple steps. Let $\ell > 0$ be the average number of times a state is kept (repeated) because of bad proposals. To get samples that behave more like they are independent, we should keep only one in every $k \geq \ell$ steps of the Markov chain, as described earlier. Including a burn-in period of b steps, and taking every kth term in the chain, you need to run your chain $b + \frac{mk}{\pi_s}$ steps to expect to see state s about m times in the final sample.

11.3.4 *Optimization by Sampling

One naïve way to to use sampling to minimize a function φ is to sample uniformly from the space S of all possible inputs and just keep the input that gives the minimal result. We can be more sophisticated by giving the space S a probability distribution that gives greater probability to inputs that have smaller values of $\varphi(s)$. A standard choice is to fix some $\beta > 0$ and use the probability distribution $P(s) \propto e^{-\beta\varphi(s)}$. Of course, it is generally not feasible to compute the normalizing constant $\sum_{s\in S} e^{-\beta\varphi(s)}$, since that requires computing $\varphi(s)$ for every s. If we could do that, we'd already be done, since computing every $\varphi(s)$ is exactly what's needed for brute-force optimization.

But, as described in Nota Bene 11.3.10, the RTMH construction can sample from S with the distribution proportional to $e^{-\beta\varphi(s)}$ without ever computing the normalizing constant. This is usually much more effective than sampling uniformly from S.

Example 11.3.12. Consider the $(0,1)$-knapsack problem. You have m possible objects to include in your knapsack, the jth object has value v_j and has weight w_j. The total weight of all the objects may not exceed W. Your goal is to find the most valuable combination of objects to include in the knapsack without exceeding the maximum weight. This problem is known to be NP-complete.

If we let $z_j = 1$ when the jth object is included in the knapsack and 0 otherwise, then we can describe the state space as the set

$$S = \{\mathbf{y} = (z_1, \ldots, z_m) \in \{0,1\}^m \,|\, \sum_{j=1}^{m} z_j w_j \leq W\}.$$

The value of a tuple $\mathbf{y}$ is

$$V(\mathbf{y}) = \sum_{j=1}^{m} z_j v_j.$$

To maximize V (minimize $-V$), use the RTMH algorithm to sample from a distribution on S proportional to $e^{\beta V(\mathbf{y})}$ for some choice of $\beta > 0$. To begin, we need a proposal Markov chain X. Here is one candidate: given $X_t = \mathbf{y}$, construct $X_{t+1} = \mathbf{y}' \in S$ by choosing j uniformly from $\{1, \ldots, m\}$ and replacing z_j in $\mathbf{y}$ by $1 - z_j$. If the resulting $\mathbf{y}'$ is not in S, then return to $\mathbf{y}$ and repeat the process until finding a $\mathbf{y}' \in S$ and set $X_{t+1} = \mathbf{y}'$. This is the proposal.

If m' is the number of items in $\{1, \ldots, m\}$ that can be changed from $\mathbf{y}$ without leaving S, then $q_{\mathbf{y}',\mathbf{y}} = \frac{1}{m'} = q_{\mathbf{y},\mathbf{y}'}$. Thus the proposal Markov Chain has a symmetric transition matrix Q.

Given $Y_t = \mathbf{y}$ the RTMH constuction (with target distribution proportional to $e^{\beta V(\mathbf{y})}$) proceeds as follows.

(i) Choose j and proposal $\mathbf{y}'$ as described above.

(ii) Compute the acceptance probability

$$a_{\mathbf{y}',\mathbf{y}} = \min\left(\frac{\frac{1}{Z}e^{\beta V(\mathbf{y}')}q_{\mathbf{y}',\mathbf{y}}}{\frac{1}{Z}e^{\beta V(\mathbf{y})}q_{\mathbf{y},\mathbf{y}'}}, 1\right) = \min\left(e^{\beta(V(\mathbf{y}')-V(\mathbf{y}))}, 1\right)$$

As before, the normalizing constant Z cancels, and so do the terms $q_{\mathbf{y}',\mathbf{y}}$ and $q_{\mathbf{y},\mathbf{y}'}$ (because of the symmetry of Q).

(iii) Accept the proposal $\mathbf{y}'$ with probability $a_{\mathbf{y}',\mathbf{y}}$ and set $Y_{t+1} = \mathbf{y}'$. Otherwise reject the proposal (set $Y_{t+1} = \mathbf{y}$).

The resulting Markov chain Y has the desired stationary distribution. At each step, if the proposal $\mathbf{y}'$ has greater value than the previous choice $\mathbf{y}$, that is, if $V(\mathbf{y}') \geq V(\mathbf{y})$, then $a_{\mathbf{y}',\mathbf{y}} = 1$ and we are guaranteed to move to $\mathbf{y}'$. And if $V(\mathbf{y}') < V(\mathbf{y})$ there is still a nonzero probability of moving to the less-valuable proposal $\mathbf{y}'$. The larger we make β, the smaller the probability of moving to a less-valuable proposal. To try to maximize V, we run this MCMC sampler and remember the value of $\mathbf{y}$ that maximizes V.

This method of optimization is closely related to simulated annealing. Indeed, simulated annealing corresponds to following this RTMH algorithm, but increasing β over time.

11.4 Sampling from the Posterior

Most likelihoods don't have nice conjugate priors, and even when they do, the conjugate prior might not be the best choice to use in our model. But when the prior and the likelihood aren't conjugate, then explicitly computing the posterior is usually painful. But, thanks to MCMC, we can still (usually) sample from the posterior, and this is almost always enough. There are several efficient implementations of MCMC methods in Python (*PyMC*, *Pyro*, and *PyStan*) and in R (*RStan*). We use *PyMC* here.

11.4.1 A Gaussian Model for Height

Heights of human adult males are approximately normally distributed $h \sim \mathcal{N}(\mu, \sigma^2)$. Given a data set $\mathbf{D} = \{h_1, \ldots, h_N\}$ consisting of height measurements (in centimeters), we'd like to use the data to estimate the parameters μ and σ^2 of the distribution. In the past we used maximum likelihood estimation to get

$$\widehat{\mu}_{MLE} = \frac{1}{N}\sum_{i=1}^{N} h_i$$
$$\widehat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(h_i - \widehat{\mu}_{MLE})^2.$$

But those are just point estimates of the parameters μ and σ^2. We'd prefer to have a distribution for each of those parameters, which we can get using Bayesian methods.

As in Example 11.2.8, the algebra is much cleaner if we switch from variance σ^2 to precision $\tau = \frac{1}{\sigma^2}$. We must choose a prior on μ and τ. To simplify, we assume a prior with μ and τ independent (but they will not be independent in the posterior). As described in Example 11.2.8, a normal prior of the form $\mathcal{N}\left(\nu, \lambda^2\right)$ for μ with fixed τ is conjugate.

We'll use some domain knowledge to choose the priors: I'm 182 cm and most men I know are slightly shorter than I am, so I'll guess a mean for μ of $\nu = 180$. Also, almost all men I know are between 160 cm and 200 cm, and choosing a standard deviation of 10 would put 95% of the distribution $\mathcal{N}(180, 100)$ in that range, so $\lambda^2 = 10^2$ seems like a reasonable choice.

The quantity τ must always be positive, and the previous discussion makes a case for $\tau \approx \frac{1}{10^2}$, so we want a prior with positive support and for which values near 0.01 are not unlikely, but larger values are also allowed. For fixed μ, the gamma distribution $\tau \sim \text{Gamma}(a, b)$ for some choice of a and b is a conjugate prior. Let's take $a = 2$ and $b = 200$, which gives a mean of $\frac{a}{b} = 0.01$ for the prior on τ. Therefore, the model is

$$h \sim \mathcal{N}(\mu, \frac{1}{\tau})$$
$$\mu \sim \mathcal{N}(180, 100)$$
$$\tau \sim \text{Gamma}(2, 200)$$

The posterior distribution is

$$P(\mu, \tau \,|\, \mathbf{D}) = \frac{\prod_{i=1}^{N} \mathcal{N}\left(h_i \,|\, \mu, \frac{1}{\tau}\right) \mathcal{N}(\mu \,|\, 180, 100)\, \text{Gamma}(\tau \,|\, 2, 200)}{\iint \prod_{i=1}^{N} \mathcal{N}\left(h_i \,|\, \mu', \frac{1}{\tau'}\right) \mathcal{N}(\mu' \,|\, 180, 100)\, \text{Gamma}(\tau' \,|\, 2, 200)\, d\mu'\, d\tau'}$$

We can use MCMC to draw a sample of pairs (μ, τ) from this distribution, and the constant denominator is irrelevant for MCMC, so we only need

$$P(\mu, \tau) \propto \prod_{i=1}^{N} \mathcal{N}\left(h_i \,|\, \mu, \frac{1}{\tau}\right) \mathcal{N}(\mu \,|\, 180, 100)\, \text{Gamma}\left(\tau \,|\, 2, 200\right)$$

Because of the conjugate prior relationships in each coordinate (μ and τ), we can draw from this distribution using Gibbs sampling as described in Example 11.2.8 to get $(\mu_1, \tau_1), \ldots, (\mu_K, \tau_K)$. From this draw we can estimate the expected values $\mathbb{E}[(\mu, \tau)] \approx \frac{1}{K}\sum_{i=1}^{k}(\mu_i, \tau_i)$ and many other properties of the (both joint and marginal) posterior distribution.

```
import pymc as pm
import arviz as az        # visualization package

height_model = pm.Model()

with height_model:

    # Priors for unknown model parameters
    mu = pm.Normal('mu',mu=180,sigma=10)
    tau = pm.Gamma('tau',alpha=2,beta=200)

    # Likelihood
    obs = pm.Normal("obs", mu=mu, sigma=1/tau**0.5, observed=↩
        heights)

    # Sample 1000 times from the posterior
    trace = pm.sample(1000)

    # Show the traceplot
    az.plot_trace(trace)
```

Algorithm 11.3: Using PyMC to sample from the posterior of the model for height as described in Section 11.4.1. The result is a sample `trace` of length 1000 from posterior, and the trace plot shown in Figure 11.7. The sampler uses No-U-Turn (NUTS) Hamiltonian MCMC, and for this example it ran four Markov chains each for 1000 tuning steps before running each for 1000 more steps to draw, for a total of 8000 steps. The computation is much slower than computing the MLE—it took 13 seconds on a 2019 MacBook Pro. But it gives much more information, and there are many advantages to using a Bayesian model over just computing the point estimate of the MLE (or the MAP).

Alternatively, there are some powerful implementations of more general MCMC samplers that we can easily call, even when we don't have a nice conjugate prior for the coordinates. One popular implementation is PyMC. An example of how the Gaussian height model above can be sampled using PyMC is given in Algorithm 11.3.

One of the diagnostic tools for deciding if the sample was well mixed, meaning that it looks like a draw from the stationary distribution, is the *trace plot*. The trace plots for the sample run in Algorithm 11.3 is shown in the right panels of Figure 11.7. The trace plot just plots the value of each step in the Markov chain in the order it was drawn. Ideally we'd want samples to be independent, but in a Markov chain that won't happen. Nevertheless, if the trace plot looks like a fuzzy caterpillar, then that suggests consecutive steps in the Markov chain are not strongly correlated.

Another diagnostic tool is to run the Markov chain multiple times, with different starting points, plotting the density plot (KDE) for each chain. If the density plots are similar for the multiple chains, that suggests the chains are mixing well (are approaching the stationary distribution). If they are very different, that suggests the chains have not mixed well and need to be run for longer before they can give a good approximation of the stationary distribution. PyMC defaults to running four chains from different starting points.

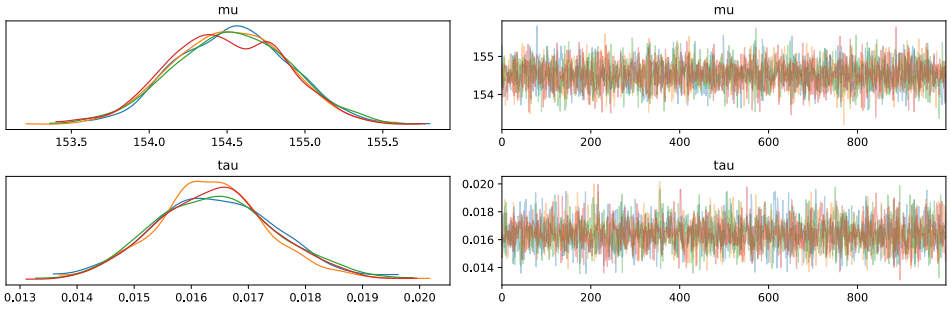


Figure 11.7: Density plots and trace plots of an MCMC sample of the posterior for the Gaussian height model of Section 11.4.1. In the left panels are the density plots (KDEs) of the posterior p.d.f. for each of the four chains run for μ (top left) and τ (bottom left). The fact that the plots for τ look fairly similar to each other suggests that the sample distribution has converged to the stationary distribution (the posterior). For μ the KDEs look less similar, but probably good enough for our purposes here. The right panels show the trace plots, consisting of steps in the Markov chain for μ (top right) and τ (bottom right). In both the top and bottom trace plots, consecutive steps do not show much, if any, correlation to one another, which suggests they are approximating independent draws.

The MAP estimates of the parameters of this model ($\mu_{MAP} = 154.5227$ and $\sigma_{MAP} = 7.8170$) are not significantly different from the MLE ($\mu_{MAP} = 154.4897$ and $\sigma_{MAP} = 7.7874$), which is generally to be expected, if there are enough data points to swamp the prior. But the MAP does not capture the full range of possibilities for the parameters. We can look at various descriptive statistics for the sample to approximate the corresponding quantity in the posterior distribution. Specifically, we can find the mean, median, first quartile, and standard deviation of μ by computing the corresponding statistics of the the sample, as in Algorithm 11.4. To find the mean of σ, first apply the transformation `sigma = 1/trace.posterior['tau']**0.5` to get the corresponding sample of σ and then proceed as with μ.

```
with(height_model):
    mu = trace.posterior['mu']
    print(mu.mean())
    print(np.median(mu))
    print(np.quantile(mu,0.25))
    print(mu.std())
```

Algorithm 11.4: A sample from a distribution can give good estimates of many properties of a distribution, including the mean, median, quartiles, and standard deviation. Here we compute some of these for the draw of `mu` in the `height_model`.

```
with height_model:
    mu = trace.posterior["mu"]  # mu part of sample from the ↩
        posterior
    tau = trace.posterior["tau"]  # tau part of sample from the ↩
        posterior

    # Draw h_j from N(mu_j, 1/tau_j) by rescaling and
    #                    translating standard normal
    sample_h = (np.random.randn(len(tau)) / tau**0.5) + mu
```

Algorithm 11.5: To draw a sample from the Bayesian weighted average, one need only sample from the posterior for μ and σ (just use the sample `trace` computed above) and then for each sample pair (μ_j, σ_j) draw once from the likelihood $\mathcal{N}(\mu_j, \sigma_j^2)$.

One can also do Bayesian model averaging, which consists of summing all the possible models (one for each choice of μ and σ) weighted by the posterior. To sample from this weighted average of models, first sample from the posterior for μ and σ and then for each sample pair (μ_j, σ_j) draw once from the likelihood $\mathcal{N}(\mu_j, \sigma_j^2)$ by drawing from the standard normal with `np.random.randn`, multiplying by σ, and adding μ. The code for this is given in Algorithm 11.5. A histogram of the sampled Bayesian weighted average is plotted in Figure 11.8 superimposed over a histogram of the original data.

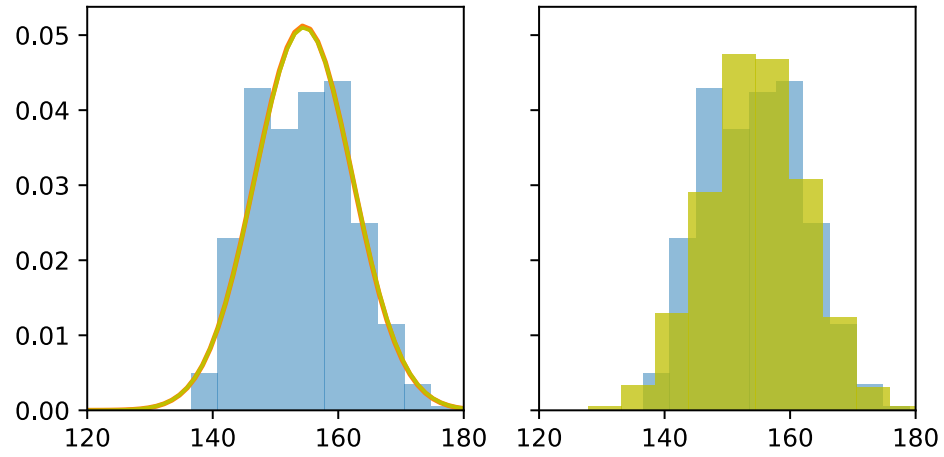


Figure 11.8: The height data $\mathbf{D}$ and the results of an MCMC sample of the posterior for the Gaussian height model of Section 11.4.1. In the left panel the p.d.f. (yellow) of the model using the MAP parameters, superimposed on a histogram of the original height data $\mathbf{D}$. On the right a histogram (yellow) from a draw of length 1000 from the weighted average Bayesian model, as described in Algorithm 11.5, also superimposed over the same histogram (blue) of the original data.

11.4.2 Bayesian Linear Regression

Consider fitting a linear model of weight and height for adult human males. It seems reasonable to assume that weight is approximately a linear function of height:

$$w = \alpha + \beta h + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

for yet-to-to-be-estimated values of α, β, and σ^2. For the Bayesian model it is nicer to rewrite this as

$$\begin{aligned} w &\sim N(\mu, \sigma^2) \\ \mu &= \alpha + \beta h. \end{aligned}$$

We must choose priors for α, β and σ. I'll take α and β to be normally distributed. I know a male who is 183cm tall and weighs 90kg, and I know another male who is 152cm tall and weights 55kg. If I draw a line between those two points it has slope $b = 1.13$ and intercept $a = -116.6$, so I'll take those as the means of my priors for β and α, respectively.

The variance σ^2 is always positive, so normal is not a good choice. But the logarithm of σ could be normal, so I'll try that. Most people's weight seems not to vary by much more than about 20kg, so we could guess that as the standard deviation, and the log of 20 is about 3. To be generous (that is, to avoid constraining the result very much with my priors), I'll let the variance on each of my normal priors be relatively large:

$$\begin{aligned} a &= -116.6 \\ b &= 1.13 \\ \alpha &\sim \mathcal{N}(a, 10000) \\ \beta &\sim \mathcal{N}(b, 100) \\ \log(\sigma^2) &\sim \mathcal{N}(3, 25) \end{aligned}$$

Given a data set of heights and weights (in this case for 50 human males), the posterior distribution satisfies

$$P(\alpha, \beta, \sigma^2 \,|\, \mathbf{D}) \propto \prod_{i=1}^{50} \mathcal{N}(w_i \,|\, \alpha+\beta h_i, \sigma^2) \mathcal{N}(\alpha \,|\, a, 10000) \mathcal{N}(\beta \,|\, b, 100) \mathcal{N}(\log(\sigma) \,|\, 3, 25)$$

We can use MCMC to draw from this distribution. Each draw of α, β, σ^2 from the posterior gives a line of the form $\alpha+\beta h$ and a variance σ^2 that together can be used for the model $w \sim \mathcal{N}(\alpha + \beta h, \sigma^2)$; see Figure 11.10. An example of how this can be sampled using PyMC is given in Algorithm 11.6. The trace plot and posterior histograms are given in Figure 11.9.

Each draw α, β, from the posterior corresponds to a possible linear model, and $\log(\sigma)$ determines a distribution around that linear model. It is sometimes useful to plot some of the resulting lines and a confidence interval around each line. An example of this is shown in Figure 11.10.

Finally, it is often useful to find the mean of the posterior and the highest-density interval (HDI). If we must choose a single value from the posterior, the mean is usually a better choice than the MAP (the mode of the posterior), since the mode, if it exists at all, can be atypical for the distribution, while the mean is more typical (it is the 'expected' value). The HDI is the smallest credible interval containing a given percentage (in PyMC, the default is 94%) of the posterior probability. Every point in the HDI has a higher density than any point outside the HDI, which makes the HDI a good choice for a credible interval representing "most" of the range that the parameter could lie in. The command `az.plot_posterior(trace, var_names=[ 'alpha','beta','logsigma']` yields the posterior plots in Figure 11.11, including the mean and the HDI. These density plots are essentially the same as those in Figure 11.9, but it's helpful to have them to give context to the HDI.

```
model = pm.Model()

with model:

    # Initial values
    a = -116.6
    b = 1.13

    # Priors for unknown model parameters
    alpha = pm.Normal('alpha', mu=a, sigma=100)
    beta = pm.Normal('beta', mu=b, sigma=10)
    logsigma = pm.Normal('logsigma', mu=3,sigma=5)

    # Expected value of outcome
    mu = alpha + beta * heights

    # Likelihood (sampling distribution) of observations
    Y_obs = pm.Normal('Y_obs', mu=mu, sigma=np.exp(logsigma), ↩
        observed=weights)

    # Draw 1000 times from the posterior
    trace = pm.sample(1000)

    # Show the traceplot
    az.plot_trace(trace, compact=False)
```

Algorithm 11.6: Using PyMC to sample from the posterior of the model for weight as a linear function of height. Note that PyMC uses the convention of parametrizing the normal distribution by the standard deviation σ instead of by the variance σ^2 (lines 10–12). Here we have a draw `trace` of length 1000 from posterior, with density and trace plots shown in Figure 11.9.

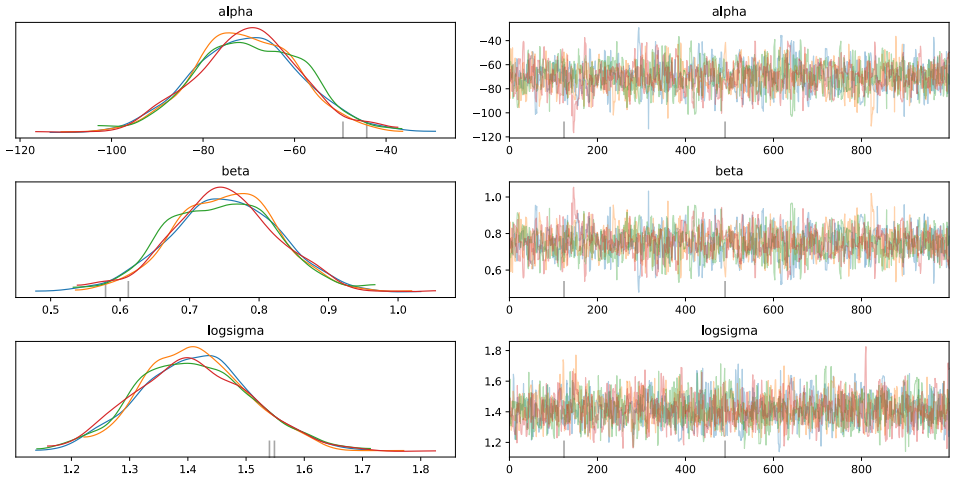


Figure 11.9: The density plots and trace plots for an MCMC sample of the posterior for the model of weight as a linear function of height. In the left panels are density plots of the posterior p.d.f. for each of four chains run for α (top left), β (middle left), and $\log(\sigma)$ (bottom left). The four different chains give fairly similar-looking KDEs, which suggests suggests that the sample distribution may have approximately converged to the stationary distribution (the posterior). The right panels show trace plots for α (top right) and β (middle right) and $\log(\sigma)$ (bottom right). All of them seem to show little or no correlation between consecutive points, which is an encouraging sign.

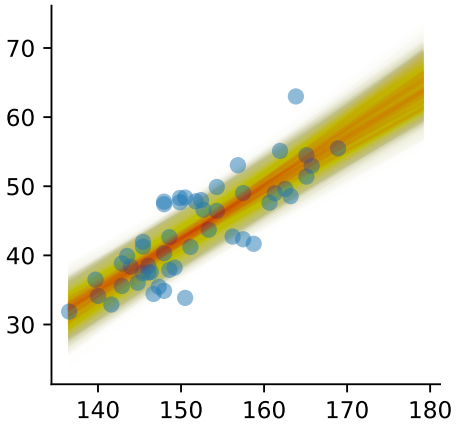


Figure 11.10: A scatterplot (blue) of height and weight for 50 individuals, with a sample of regression lines (red). For several draws of $\alpha, \beta, \log(\sigma)$ from the posterior distribution, we have plotted the line $\alpha + \beta h$ (red) with a neighborhood (yellow) of width σ.

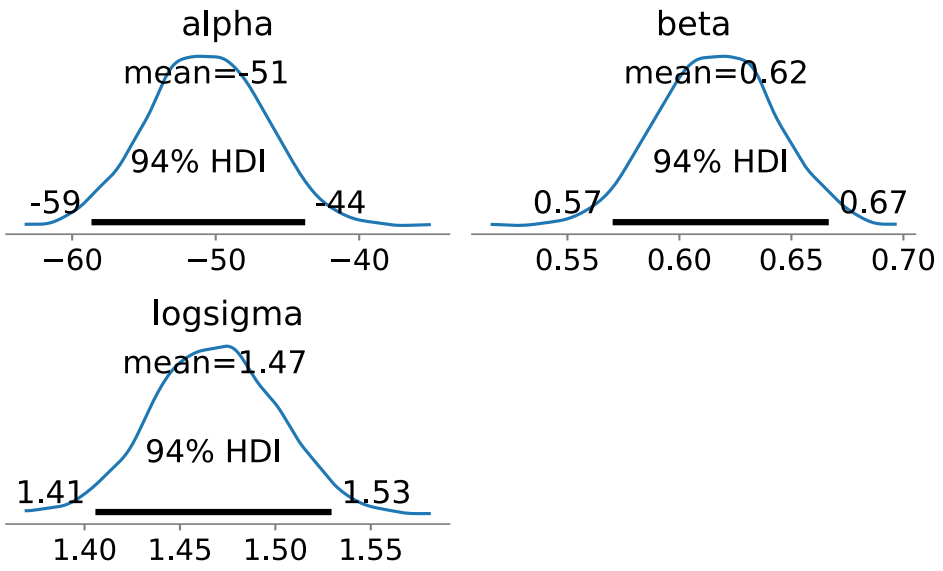


Figure 11.11: Density plots for an MCMC sample of the posterior for the model of weight as a linear function of height. These plots also give the mean of the posterior and the 94% highest density credible interval (HDI), which is the smallest interval containing 94% of the posterior probability. Every point in the HDI has a higher density than any point outside the HDI, which makes the HDI a good choice for a credible interval to summarize where most of the density of the posterior is located. The density plots here are essentially the same as those in Figure 11.9, but it's helpful to have them to give context to the HDI.

Exercises

Note to the student: Each section of this chapter has several corresponding exercises, all collected here at the end of the chapter. The exercises between the first and second line are for Section 1, the exercises between the second and third lines are for Section 2, and so forth.

You should **work every exercise** (your instructor may choose to let you skip some of the advanced exercises marked with *). We have carefully selected them, and each is important for your ability to understand subsequent material. Many of the examples and results proved in the exercises are used again later in the text. Exercises marked with ⚠ are especially important and are likely to be used later in this book and beyond. Those marked with † are harder than average, but should still be done.

Although they are gathered together at the end of the chapter, we strongly recommend you do the exercises for each section as soon as you have completed the section, rather than saving them until you have finished the entire chapter.

11.1. Use importance sampling, drawing from Beta(a, b) for various values of a and b, to estimate the integral $\int_0^\pi \frac{dx}{x^3+2x+3}$. Find values of a and b and sample size n that will give a standard error less than 10^{-3}.
Hint: The domain of Beta(a, b) is $[0, 1]$, but the integral is to be evaluated over $[0, \pi]$; so, you'll need to do a change of variables to be able to calculate this with samples from Beta(a, b).

11.2. Finish the proof of Theorem 11.1.6

11.3. The c.d.f. of a discrete distribution F_X is usually not invertible, so $F^{-1}(U)$ cannot be used for inversion sampling. Nevertheless, if $A = \text{supp}(X) \subset \mathbb{R}$ is a closed discrete set, then we can define $G(u) = \min\{a \in A | u \leq F_X(a)\}$.

(i) Prove that if $U \sim \text{Uniform}(0, 1)$, then the variable $Y = G(U)$ is a random variable with c.d.f. equal to F_X. This gives an algorithm for sampling from discrete distributions that we call *discrete inversion sampling*.
Hint: Rewrite $F_Y(y)$ as $P(\exists a \in A \text{ s.t. } U \leq F_X(a) \leq F_X(y))$ and then show that for every $y > 0$ there exists $a \in A$ such that $F_X(a) = F_X(y)$.

(ii) Implement discrete inversion sampling to sample from a Bernoulli(p) distribution for arbitrary p. Use your method to draw 1000 times with $p = 0.3$.

(iii) Use discrete inversion sampling to sample from a Binomial(n, p) distribution for arbitrary n and p. Another way to sample from Binomial(n, p) is to sample from a Bernoulli(p) distribution n times and sum the results. Implement both methods and draw 1000 times with $n = 10$ and $p = 0.3$ for each implementation. Compare the speed of each and plot a (normed) histogram for each.

(iv) Use discrete inversion sampling to sample from a NegBin(k, p) distribution for arbitrary k and p. Draw 1000 times with $k = 5$ and $p = 0.4$, and plot a (normed) histogram of the results along with the p.m.f. of the true distribution.

11.4. Use rejection sampling to sample 1000 times from a distribution with p.d.f. $f(x) \propto x(1-x)e^x$ on $[0,1]$. Plot a (normed) histogram of the results, along with a plot of the p.d.f.

11.5. Consider the distribution with p.d.f. $f_{(X,Y)}(x,y) \propto e^{-\frac{1}{2}(x^2y^2+x^2+y^2-8x-8y)}$.

(i) Show that $f_{(X,Y)}(x,y) \propto g(y)e^{-\frac{1}{2}(1+y^2)\left(x-\frac{4}{1+y^2}\right)^2}$, and hence the conditional distribution $f_{(X|Y)}(x \,|\, y)$ is normal with mean $\frac{4}{1+y^2}$ and variance $\frac{1}{1+y^2}$. Since $f_{(X,Y)}(x,y)$ is symmetric in x and y, a similar result holds for $f_{(Y|X)}(y \,|\, x)$.

(ii) Implement a Gibbs sampler to sample from the joint distribution $f_{(X,Y)}(x,y)$ and, starting at $(x_0,y_0)=(0,0)$, make a draw of length 10^5.

(iii) Plot a hexbin (a two-dimensional histogram, where height is represented by color) of the results.

(iv) Use your draw to estimate the mean and covariance matrix of this distribution.

11.6. For the Gibbs sampler in Example 11.2.9 for the joint (posterior) distribution $P(\theta, n \,|\, x)$ given the data that x hatchlings are observed:

(i) Implement the algorithm in code.

(ii) Given that $x = 17$ hatchlings are observed, identify a good starting value $\mathbf{s}_0 = (\theta_0, n_0)$. Does your choice depend on x? Explain why your choice of $\mathbf{s}_0$ is reasonable.

(iii) Let your Gibbs sampler run for 10^5 steps and plot histograms (or KDEs) estimating the p.d.f.s of Θ and N.

(iv) Use your sample from the previous step to estimate $\mathbb{E}[\Theta \mid X = 17]$, $\mathrm{Var}(\Theta \,|\, X = 17)$, $\mathbb{E}[N \,|\, X = 17]$, and $\mathrm{Var}(N \,|\, X = 17)$.

11.7. Prove the analogue of Theorem 11.2.3 for the Gibbs sampler with more than one random variable, as described in Section 11.2.4. That is,

(i) Prove it is aperiodic.

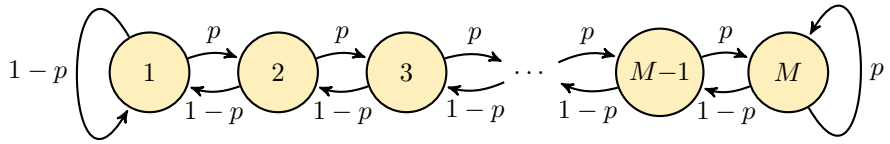
(ii) Prove its stationary distribution is the original joint distribution $P(X,Y)$, and.

(iii) Assuming that $P(x_1,\ldots,x_n) > 0$ whenever $P(x_2,x_3,\ldots,x_n) > 0$ or $P(x_1,\widehat{x_2},\ldots,x_n) > 0$ or ... or $P(x_1,\ldots,\widehat{x_j},\ldots,x_n) > 0$ or ... or $P(x_1,\ldots,x_{n-1}) > 0$, prove that the Gibbs sampler is irreducible.

Hint: You may use (11.5) for the transition probabilities.

11.8. Finish the proof of Proposition 11.3.3 by proving that the RTMH construction is reversible. Take care to also treat the edge cases where one of the terms $q_{s,s'}$ or $q_{s',s}$ is zero.

11.9. Adapt Example 11.3.9 to the situation where the proposal chain X is given by this diagram



for some value of $p \in (0, 1)$.

(i) What is the acceptance probability for each pair s, s'?

(ii) Implement a RTMH sampler for the Zipf distribution using this proposal Markov chain. Your code should accept a positive integer M as the number of states, a float $\alpha > 0$ defining the target distribution $P(k) \propto \frac{1}{k^\alpha}$, a probability $p \in (0, 1)$, a positive integer N as the total number of samples, and an initial state x_0. Your code should return an array $[y_0, y_1, \ldots, y_N]$ of samples from the Zipf distribution.

(iii) For $M = 20$, $\alpha = 2$, and each value of $p = \frac{k}{5}$ for $k = \{1, 2, 3, 4\}$, run your code for $n = 10^5$ steps, and plot a normed histogram (with $M = 20$ bins) of the results of each, along with a plot of the true distribution (this last step will require you to compute $Z = \sum_{k=1}^{20} \frac{1}{k^\alpha}$). Hint: SciPy has some overhead compared to NumPy because of the extra wrappers it puts around the NumPy methods, so if speed is a problem, you may want to call NumPy methods directly, if available.

11.10. Implement the RTMH sampling algorithm for continuous target distributions on $\mathbb{R}$, using the normal distribution $\mathcal{N}(\mu = y_t, \sigma^2)$ for the proposal (σ^2 fixed). Your code should do the following.

(i) Accept a callable function $f(s)$ that is proportional to the p.d.f. of the target distribution, a choice of σ^2, an initial starting point y_0, and a number of samples N to take.

(ii) At each stage $t \in \{0, \ldots, N-1\}$ draw a proposal x from $\mathcal{N}(\mu = y_t, \sigma^2)$.

(iii) Decide whether to accept x (set $y_{t+1} = x$) or reject x (set $y_{t+1} = y_t$) based on the acceptance probability a_{xy_t}.

(iv) Increment t by one (`t+=1`) and repeat from (ii).

Given a target of $f(s) = 4e^{-\frac{(s+3)^2}{4}} + 6e^{-\frac{(s-5)^2}{32}}$, with $\sigma^2 = 1$ and an initial value of $y_0 = 5$, run your sampler for 10^5 steps. Note that the target f is not a pdf because it does not integrate to 1. Compute or approximate $Z = \int_{-\infty}^{\infty} f(x)\,dx$ (Hint: it can be computed as a sum of Gaussians, or you can use a numerical quadrature method). Plot a normed histogram (find an appropriate number of bins) of your samples and on the same graph plot $\frac{f}{Z}$.

11.11. Adjust your code from Exercise 11.10 to implement the RTMH sampling algorithm for continuous target distributions on $\mathbb{R}$, using the distribution Uniform$(y_t - L, y_t + L)$ for the proposal. Your code should do the following.

(i) Accept a target $f(s)$, a choice of $L > 0$, an initial starting point y_0, and a number of samples N to take.

(ii) At each stage $t \in \{0, \ldots, N-1\}$ draw a proposal x from Uniform($[y_t - L, y_t + L]$).

(iii) Decide whether to accept x (set $y_{t+1} = x$) or reject x (set $y_{t+1} = y_t$) based on the acceptance probability a_{xy_t}.

(iv) Increment t by one (`t+=1`) and repeat from (ii).

Given the target $f(s) = 4e^{-\frac{(s+3)^2}{4}} + 6e^{-\frac{(s-5)^2}{32}}$, with $L = 1$ and an initial value of $y_0 = 5$:

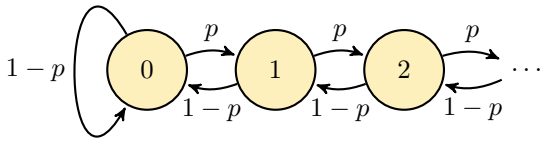
(i) Run your sampler for $N = 10^5$ steps.

(ii) Compute or estimate $Z = \int_{-\infty}^{\infty} f(x)\,dx$.

(iii) Plot a histogram of your samples, and plot $\frac{f}{Z}$ on the same graph.

(iv) Use your sample to estimate the mean, variance, median, and first and third quartiles of the distribution.

11.12.* Consider the following proposal Markov chain on $\mathbb{N}$



for some value of $p \in (0, 1)$.

(i) Implement a RTMH method to sample from Poisson(λ) using this proposal. Your code should accept floats λ and p, an initial starting point $x_0 \in \mathbb{N}$ and a number N of samples to take.

(ii) Let $p = 0.3$ and $x_0 = 3$. Run your code to produce 10^5 samples from Poisson(4).

(iii) Plot a normed histogram of your results together with the p.m.f. of the distribution.

11.13.* Using your code from Exercise 11.10, with a target proportional to $f(s) = 4e^{-\frac{(s+3)^2}{4}} + 6e^{-\frac{(s-5)^2}{32}}$, with normal proposal and $\sigma^2 = 1$:

(i) Using a starting point of $y_0 = -30$, run your sampler for 10^5 steps. For each $k \in \{1, 2, \ldots, 5\}$ plot a normed histogram of the first 10^k samples and also plot $\frac{f}{Z}$. Hint: Drawing lots of samples takes time. Spending a little effort to optimize your code can make a big difference. And even then, you should prepare for the fact that you may need to let your code run for a while.

(ii) Change your starting point to $y_0 = 50$ and run your sampler for 10^5 steps. For each $k \in \{1, 2, \ldots, 5\}$ plot a histogram of the first 10^k samples and also plot $\frac{f}{Z}$.

(iii) Use all these results and plots to determine a good value B for the burn in, independent of an initial value in $[-30, 50]$. Justify your choice.

11.14.* Adjust your code from Exercise 11.10 (with a normal proposal) to accept a number B of burn-in steps to run before collecting N samples (so it will compute a total of $N+B$ samples and discard the first B of them). Make the same adjustment for your code from Exercise 11.11 (with a uniform proposal).

(i) Apply the code for the normal proposal to a target of $f(s) = 4e^{-\frac{(s+3)^2}{0.25}} + 6e^{-(s-5)^2}$ with $\sigma^2 = 1$ with $B = 10^5$ and $N = 10^5$ for each starting point of $x_0 \in \{-10, -8, -6, \ldots, 6, 8, 10\}$.

(ii) Plot a histogram of the results of each run, and on the same graph plot $\frac{f}{Z}$, where $Z = \int_{-\infty}^{\infty} f(x)\,dx$ is the normalizing constant.

(iii) Repeat the previous two steps for the uniform proposal.

11.15. Install `PyMC` and `arviz` on your computer (pip install usually works for this) and do the following:

(i) Take a draw `data` of length 20 from a Bernoulli(0.3) distribution. This is the dataset. We'll now act as if θ were unknown.

(ii) Assume that the data are distributed as Bernoulli(θ) for some unknown value of θ having a prior of Beta(1, 1). Set up a PyMC model for this situation and sample from the posterior 1000 times:

```
with pm.Model() as bern_model:
    theta = pm.Beta('theta', alpha=1., beta=1.)
    y = pm.Bernoulli('y', p=theta, observed=data)
    trace = pm.sample(1000)
```

(iii) Print a trace plot and decide if the model has converged well.

```
with bern_model:
    az.plot_trace(trace, compact=False)
    plt.tight_layout() # Helps the labels not overlap
```

(iv) Use the samples to estimate the mean, median, and variance for the posterior. Compare the true value ($\theta = 0.3$) with the posterior distribution for θ.

(v) Analytically compute the posterior in this model, using the fact that the beta distribution is conjugate to the Bernoulli likelihood. Plot the analytically computed posterior and a KDE for the sampled posterior (you can use `az.plot_posterior(trace)` to plot the KDE).

11.16. Do the following:

(i) Take a sample `x` of length 100 from the standard normal and another sample `epsilon` of length 100 from $\mathcal{N}(0, 0.25)$. Let `y = 2.5 + 0.9 * x + epsilon`. The pair `x,y` will be your dataset. We'll now act as though the parameters used here were unknown.

(ii) Assume the data can be modeled as $y \sim \mathcal{N}(\mu, \sigma^2)$ for $\mu = a + bx$ for unknown a and b. Take priors of $a \sim \mathcal{N}(0, 10^2)$, $b \sim \mathcal{N}(0, 1)$, and $\sigma = |5t|$ with $t \sim$ Cauchy (see Example 4.4.9). We write this as $\sigma \sim$ HalfCauchy(5). Set up PyMC to take samples from the posterior as follows:

```
with pm.Model() as model:
    a = pm.Normal('a', mu=0, sigma=10)
    b = pm.Normal('b', mu=0, sigma=1)
    sd = pm.HalfCauchy('sd', 5)

    mu = a + b * x
    y_pred = pm.Normal('y_pred', mu=mu, sigma=sd, ↩
        observed=y)
```

(iii) Take 2000 samples from the posterior distribution, print the trace plot, and evaluate whether the chain converged well.

```
with model:
    trace = pm.sample(2000, tune=1000)
    az.plot_trace(trace, compact=False)
    plt.tight_layout() # Helps the labels not overlap
```

(iv) Compare the true values of $a = 2.5$, $b = 0.9$, and $\sigma = 0.5$ to the posterior distributions for each of these parameters.

(v) Plot a line with slope equal to the mean of `trace['b']` and intercept equal to the mean of `trace['a']`. Subsample 200 times from `trace` to get 200 pairs `(a,b)`, and on the same graph plot a corresponding semitransparent line (`alpha=0.2`) for each of these pairs. On that same graph plot (scatterplot) the data points `x` and `y`.

Notes

Sections 11.2 and **??** are heavily based on [BH15]. The RTMH algorithm originally appeared in [MRR+53]. For more on MCMC see [Gey11].