Chapter 8
Independent and identically distributed random draws

Sampling methods discussed in this note are based on two fundamental iid random generators that are available as built-in functions in Matlab. The first one is `rand.m` function which can draw iid random numbers (vectors) from the uniform distribution in $[0,1]$, denoted as $U[0,1]$, and the second one is `randn.m` function that generates iid numbers (vectors) from standard normal distribution $\mathscr{N}(0,I)$, where I is the identity matrix of appropriate size.

The most trivial task is how to draw iid samples $\{m_1, m_2, \dots, m_N\}$ from a multivariate Gaussian $\mathscr{N}(\overline{m}, \Gamma)$. This can be done through a so-called *whitening* process. The first step is to carry out the following decomposition

$$\Gamma = RR^T,$$

which can be done, for example, using Cholesky factorization. The second step is to define a new random variable as

$$Z = R^{-1}(m - \overline{m}),$$

then Z is a standard multivariate Gaussian, i.e. its density is $\mathscr{N}(0,I)$, for which `randn.m` can be used to generate iid samples

Show that Z is a standard multivariate Gaussian.

$$\{Z_1, Z_2, \dots, Z_N\} = \texttt{randn(n,N)}.$$

We now generate iid samples m_i via

$$m_i = \overline{m} + RZ_i.$$

Exercise 8.1. Look at `BayesianPriorElicitation.m` to see how we apply the above whitening process to generate multivariate Gaussian prior random realizations. •

You may ask what if the distribution under consideration is not Gaussian, which is true for most practical applications. Well, if the target density $\pi(m)$ is one dimensional or multivariate with independent components (in this case, we can draw samples from individual components separately), then we still can draw iid samples from $\pi(m)$, but this time via the standard uniform distribution $U[0,1]$. $U[0,1]$ has 1 as its density function, i.e.,

$$\mu_U(A) = \int_A ds, \quad \forall A \subset [0,1]. \tag{8.1}$$

Now suppose that we would like to draw iid samples from a one dimensional ($S = \mathbb{R}$) distribution with density $\pi(m) > 0$. We still allow $\pi(m)$ to be zero, but only at isolated points on $\mathbb{R}$, and the reason will be clear in a moment. Define the cumulative distribution function (CDF) as

$$\Phi(w) \stackrel{\text{def}}{=} \mathbb{P}[m < w] = \int_{-\infty}^{w} \pi(m)\, dm, \tag{8.2}$$

Why?

then it is clearly that $\Phi(w)$ is non-decreasing and $0 \le \Phi(w) \le 1$. Let us define a new random variable Z as

$$Z = \Phi(m). \tag{8.3}$$

Our next step is to show that Z is actually a standard uniform random variable, i.e. $Z \sim U[0,1]$, and then show how to draw m via Z. We begin by the following observation

$$\mathbb{P}[Z < a] = \mathbb{P}[\Phi(m) < a] = \mathbb{P}\left[m < \Phi^{-1}(a)\right] = \int_{-\infty}^{\Phi^{-1}(a)} \pi(m)\, dm, \tag{8.4}$$

where we have used (8.3) in the first equality, the monotonicity of $\Phi(m)$ in the second equality, and the definition of CDF (8.2) in the last equality. Now, we can view (8.3) as the change of variable formula $z = \Phi(m)$, then combining this fact with (8.2) to have

This already shows that the density of z is 1!

$$dz = d\Phi(m) = \Phi'(m)\, dm = \pi(m)\, dm, \text{ and } z = a \text{ when } x = \Phi^{-1}(a).$$

Do you see the second equality?

Consequently, (8.4) becomes

$$\mathbb{P}[Z < a] = \int_{-\infty}^{a} dz = \mu_Z(Z < a),$$

which says that the density of Z is 1, and hence Z must be a standard uniform random variable. In terms of our language at the end of Section 9.1, we can define $m = g(Z) = \Phi^{-1}(Z)$, then drawing iid samples for m is simple by first drawing iid samples from Z, then mapping them through g. Let us summarize the idea in Algorithm 1.

The above method works perfectly if one can compute the analytical inverse of the CDF easily and efficiently; it is particularly efficient for discrete random variables, as we shall show. Of course we can always compute the inverse CDF numerically. However, note that the CDF is an integral operation, and hence its inverse is

Algorithm 1 CDF-based sampling algorithm

1. Draw $z \sim U[0,1]$,
2. Compute the inverse of the CDF to draw m, i.e. $m = \Phi^{-1}(z)$. Go back to Step 1.

some kind of differentiation. As shown before, numerical differentiation could be an ill-posed problem and we do not want to add extra ill-posedness on top of the original ill-posed inverse problem that we started with. Instead, let us introduce a simpler but more robust algorithm that works for multivariate distribution without requiring the independence of individual components. We shall first discuss the algorithm and then analyze it to show why it works.

Suppose that we want to draw iid samples from a target density $\pi(m)$, but we only know it up to a constant $C > 0$, i.e., $C\pi(m)$. (This is perfect for our Bayesian inversion framework since we typically know the posterior up to a constant as in (5.5).) Assume that we have a *proposal distribution* $q(m)$ at hand, for which we know how to sample easily and efficiently. This is not a limitation since we can always take either the standard normal distribution or uniform distribution as the proposal distribution. We further assume that there exists $D > 0$ such that

$$C\pi(m) \leq Dq(m), \tag{8.5}$$

then we can draw a sample from $\pi(m)$ by the rejection-acceptance sampling Algorithm 2.

Algorithm 2 Rejection-Acceptance sampling algorithm

1. Draw m from the proposal $q(m)$,
2. Compute the *acceptance probability*

$$\alpha = \frac{C\pi(m)}{Dq(m)},$$

3. Accept m with probability α or reject it with probability $1-\alpha$. Go back to Step 1.

In practice, we carry out Step 3 of Algorithm 2 by flipping an "α-coin". In particular, we draw u from $U[0,1]$, then accept m if $\alpha > u$. It may seem to be magic why Algorithm 2 provides random samples from $\pi(m)$. Let us confirm this using the Bayes formula (4.6).

Proposition 8.1. *Accepted m in Algorithm 2 is distributed by the target density π.*

Make sure you understand this proof since we will reuse most of it for the Metropolis-Hastings algorithm!

Proof. Denote B as the event of accepting a draw q (or the acceptance event). Algorithm 2 tells us that the probability of B given m, which is precisely the acceptance probability, is

$$\mathbb{P}[B|m] = \alpha = \frac{C\pi(m)}{Dq(m)}. \tag{8.6}$$

On the other hand, the prior probability of m in the incremental event $dA = \left[m', m' + dm\right]$ in Step 1 is $q(m)\,dm$. Applying the Bayes formula for conditional probability (3.4) yields the distribution of a draw m provided that it has been already accepted

$$\mathbb{P}[m \in dA|B] = \frac{\mathbb{P}[B|m]\,q(m)\,dm}{\mathbb{P}[B]} = \pi(m)\,dm,$$

where we have used (8.6) and $\mathbb{P}[B]$, the probability of accepting a draw from q, is the following marginal probability

$$\mathbb{P}[B] = \int_S \mathbb{P}[B|m]\,q(m)\,dm = \frac{C}{D}\int_S \pi(m)\,dm = \frac{C}{D}.$$

Note that

$$\mathbb{P}[B, m \in dm] = \pi(B, m)\,dm = \mathbb{P}[B|m]\,\pi_{\text{prior}}(m) = \mathbb{P}[B|m]\,q(m)\,dm,$$

an application of (3.3), is the probability of the joint event of drawing an m from $q(m)$ and accept it. The probability of B, the acceptance event, is the total of accepting probability, which is exactly the marginal probability. As a result, we have

$$\mathbb{P}[m \in A|B] = \int_A \pi(m)\,dm,$$

which, by definition (4.2), says that the accepted m in Step 3 of Algorithm 2 is distributed by $\pi(m)$, and this is the desired result.

Algorithm 2 is typically slow in practice in the sense that a large portion of samples is rejected, particularly for high dimensional problem, though it provides iid samples from the true underlying density. Another problem with this algorithm is the computation of D. Clearly, we can take very large D and the condition (8.5) would be satisfied. However, the larger D is the smaller the acceptance probability α, making Algorithm 2 inefficient since most of draws from $q(m)$ will be rejected. As a result, we need to minimize D and this could be nontrivial depending the complexity of the target density.

Exercise 8.2. You are given the following target density

$$\pi(m) = \frac{g(m)}{C}\exp\left(-\frac{m^2}{2}\right),$$

where C is some constant independent of m, and

$$g(m) = \begin{cases} 1 \text{ if } x > a \\ 0 \text{ otherwise} \end{cases}, \quad a \in \mathbb{R}.$$

Take the proposal density as $q(m) = \mathscr{N}(0, 1)$.

1. Find the smallest D that satisfies condition (8.5).

2. Implement the rejection-acceptance sampling Algorithm 2 in Matlab and draw 10000 samples, by taking $a = 1$. Use Matlab `hist.m` to plot the histogram. Does its shape resemble the exact density shape?
3. Increase a as much as you can, is there any problem with Algorithm 2? Can you explain why?

•

Exercise 8.3. You are given the following target density

$$\pi(m) \propto \exp\left(-\frac{1}{2\sigma^2}\left(\sqrt{m_1^2+m_2^2}-1\right)^2 - \frac{1}{2\delta^2}(m_2-1)^2\right),$$

where $\sigma = 0.1$ and $\delta = 1$. Take the proposal density as $q(m) = \mathcal{N}(0, I_2)$, where I_2 is the 2×2 identity matrix.

1. Find a reasonable D, using any means you like, that satisfies condition (8.5).
2. Implement the rejection-acceptance sampling Algorithm 2 in Matlab and draw 10000 samples. Plot a contour plot for the target density, and you should see the horse-shoe shape, then plot all the samples as dots on top of the contour. Do most of the samples sit on the horse-shoe?

•