Algorithm 9 2D slice sampler
At iteration t, simulate

1. $U^{(t+1)} \sim \mathcal{U}_{[0,f(x^{(t)})]}$;
2. $X^{(t+1)} \sim \mathcal{U}_{A^{(t+1)}}$, with

$$A^{(t+1)} = \{x : \ f(x) \geq u^{(t+1)}\}.$$

The appeal of this algorithm is that it formally applies to any density known up to a multiplicative constant with no restriction on its shape or dimension. Obviously, its implementation may be hindered by the uniform simulation over the set $A^{(t)}$.

Example 7.10. Consider the density $f(x) = \frac{1}{2}e^{-\sqrt{x}}$ defined for $x > 0$. While it can be directly simulated, it also yields easily to the slice sampler. Indeed, applying the formulas above, we have

$$U|x \sim \mathcal{U}\left(0, \frac{1}{2}e^{-\sqrt{x}}\right), \quad X|u \sim \mathcal{U}\left(0, [\log(2u)]^2\right).$$

We implement the sampler to generate 5000 variates and plot them along with the density in Figure 7.8, which shows that the agreement is very good. The right panel does show some strong autocorrelations, which is typical of the slice sampler. ◀

Exercise 7.11 Referring to Example 7.10 and the density $f_X(x) = (1/2)\exp(-\sqrt{x})$:

a. Verify that the conditional distributions are

$$U|x \sim \mathcal{U}\left(0, (1/2)\exp(-\sqrt{x})\right) \text{ and } X|u \sim \mathcal{U}\left(0, [\log(2u)]^2\right),$$

and implement a Gibbs sampler to generate random variables from $f_X(x)$.

b. Make the transformation $Y = \sqrt{X}$ and show that $Y \sim \mathcal{G}(3/2, 1)$. Use this fact to simulate directly X. Compare this algorithm with the slice sampler.

There is an obvious extension to the 2D slice sampler above, akin to the multistage extension to the two-stage Gibbs sampler. If the target density is written as a product of functions,

$$f(x) = \prod_{i=1}^{n} g_i(x),$$

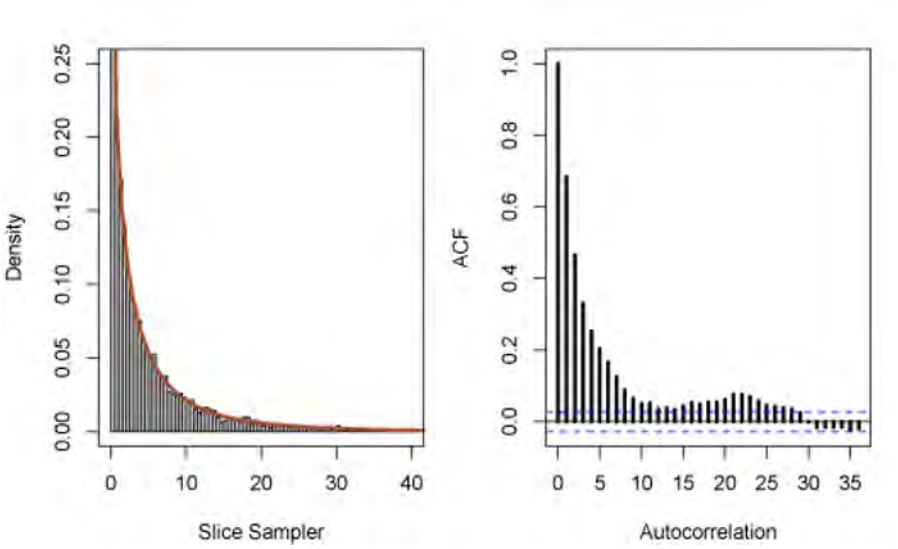

Fig. 7.8. A slice sampler histogram and target density for Example 7.10 using 5000 iterations. The left panel is the histogram with the true density overlaid, and the right panel shows the autocorrelation function.

as for instance in the case of a posterior distribution associated with a sample of n observations (where the g_i's are then the componentwise densities), an associated completion is

$$f(x, u_1, \ldots, u_n) = \prod_{i=1}^{n} \mathbb{I}\{0 < u_i < g_i(x)\},$$

which leads to a slice sampler with $(n+1)$ steps, $X^{(t)}$ then being uniformly generated over the set

$$A^{(t)} = \bigcap_{i=1}^{n} \left\{x \,:\, g_i(x) > u_i^{(t)}\right\}.$$

Example 7.11. Recall logistic regression, which we first saw in Example 4.11 and fit with a Metropolis–Hastings algorithm in Exercise 6.13. The model is

$$Y_i \sim \text{Bernoulli}(p(x_i)), \quad p(x) = \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta x)},$$

where $p(x)$ is the success probability and x is a unidimensional covariate. The likelihood associated with a sample $(\mathbf{y}, \mathbf{x}) = (y_1, x_1), \ldots, (y_n, x_n)$ is

$$L(\alpha, \beta|\mathbf{y}) \propto \prod_{i=1}^{n} \left(\frac{e^{\alpha+\beta x_i}}{1 + e^{\alpha+\beta x_i}}\right)^{y_i} \left(\frac{1}{1 + e^{\alpha+\beta x_i}}\right)^{1-y_i}.$$

Using a flat prior on (a, b), the posterior distribution can be associated with a slice sampler based on uniform

$$U_i \sim \mathcal{U}\left(0, \frac{e^{y_i(\alpha+\beta x_i)}}{1+e^{\alpha+\beta x_i}}\right)$$

variables. Generating a uniform distribution over the set

$$\left\{(a,b)\ :\ y_i(a+bx_i) > \log\frac{u_i}{1-u_i}\right\}$$

being rather unwieldy, we can further decompose the uniform simulation by consecutively simulating

$$a^{(t)} \sim \mathcal{U}\left(\max_{i;y_i=1} \log\frac{u_i^{(t)}}{1-u_i^{(t)}} - b^{(t-1)}x_i, \min_{i;y_i=0} \log\frac{1-u_i^{(t)}}{u_i^{(t)}} - b^{(t-1)}x_i\right)$$

and

$$b^{(t)} \sim \mathcal{U}\left(\max_{i;y_i=1}\left[\log\frac{u_i^{(t)}}{1-u_i^{(t)}} - a^{(t)}\right]/x_i, \min_{i;y_i=0}\left[\log\frac{1-u_i^{(t)}}{u_i^{(t)}} - a^{(t)}\right]/x_i\right),$$

if we assume without loss of generality that all x_i's are positive. However, running the corresponding slice sampler on the `challenger` dataset described in Exercise 6.13 exhibits a random walk behavior on the chain $(a^{(t)}, b^{(t)})_t$, as shown in Figure 7.9. We therefore introduce instead normal $\mathcal{N}(0, \sigma^2)$ priors on both a and b. The modification on the slice sampler is minimal in that both uniform distributions above are replaced with truncated normals $\mathcal{N}(0, \sigma^2)$, the truncation intervals being those used above. The core of the `R` code is then

```
> for (t in 2:Nsim){
+   uni=runif(n)*exp(y*(a[t-1]+b[t-1]*x))/
+               (1+exp(a[t-1]+b[t-1]*x))
+   mina=max(log(uni[y==1]/(1-uni[y==1]))-b[t-1]*x[y==1])
+   maxa=min(-log(uni[y==0]/(1-uni[y==0]))-b[t-1]*x[y==0])
+   a[t]=rtrun(0,sigmaa,mina,maxa)
+   minb=max((log(uni[y==1]/(1-uni[y==1]))-a[t])/x[y==1])
+   maxb=min((-log(uni[y==0]/(1-uni[y==0]))-a[t])/x[y==0])
+   b[t]=rtrun(0,sigmab,minb,maxb)
+   }
```

with `sigmaa` equal to 5 and `sigmab` equal to 5 divided by the standard deviation of the x_i's. ◀

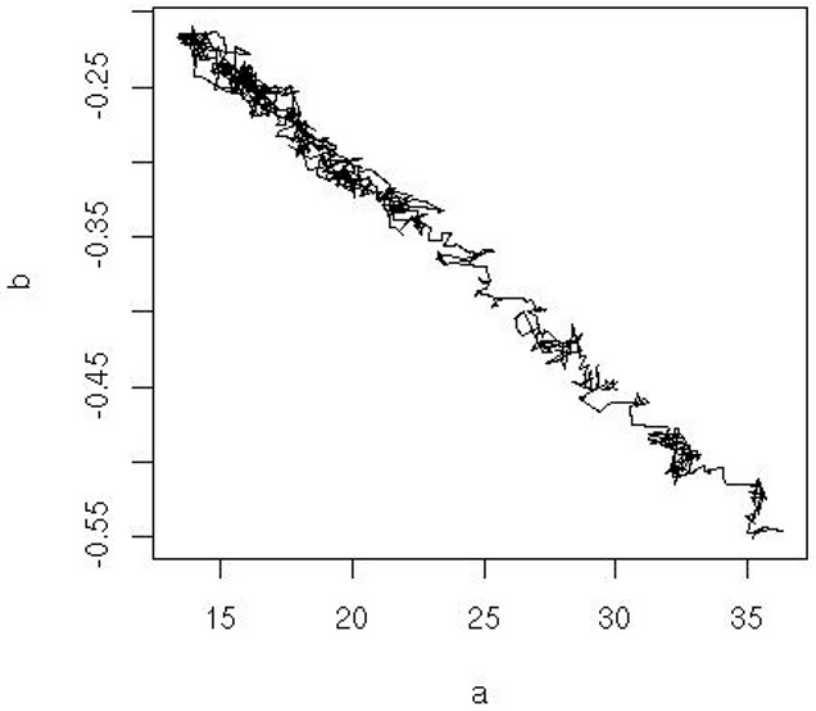

Fig. 7.9. Evolution of the chain $(a^{(t)}, b^{(t)})_t$ along 10^3 final iterations of a slice sampler for the `challenger` dataset under a flat prior.

7.5 Hierarchical structures

We have seen the multistage Gibbs sampler applied to a number of examples, most arising from missing-data structures. However, it is equally well-suited to sample in a straightforward way from any hierarchical model.

A *hierarchical model* is defined by a sequence of conditional distributions as, for instance, in the two-level generic hierarchy

$$\begin{aligned} X_i &\sim f_i(x|\theta), \quad i = 1, \ldots, n\,, \quad \theta = (\theta_1, \ldots, \theta_p)\,, \\ \theta_j &\sim \pi_j(\theta|\gamma), \quad j = 1, \ldots, p\,, \quad \gamma = (\gamma_1, \ldots, \gamma_s)\,, \\ \gamma_k &\sim g(\gamma), \quad k = 1, \ldots, s. \end{aligned}$$

The joint distribution from this hierarchy is

$$\prod_{i=1}^{n} f_i(x_i|\theta) \prod_{j=1}^{p} \pi_j(\theta_j|\gamma) \prod_{k=1}^{s} g(\gamma_k)\,.$$

Assuming that the x_i's are observations, the corresponding posterior distribution on (θ, γ) is associated with the full posterior conditionals

$$\theta_j \propto \pi_j(\theta_j|\gamma) \prod_{i=1}^{n} f_i(x_i|\theta), \quad j = 1, \ldots, p\,,$$
$$\gamma_k \propto g(\gamma_k) \prod_{j=1}^{p} \pi_j(\theta_j|\gamma), \quad k = 1, \ldots, s\,.$$

In standard hierarchies, these densities are straightforward to simulate from and are therefore naturally associated with a Gibbs sampler. In more complex hierarchies, we might need to use more sophisticated methods, such as a Metropolis–Hastings step or another slice sampler, to sample from the conditionals (as explained in Section 7.6.3). However, our main message here is that the full conditionals are quite easy to write down given the hierarchical specification, while they considerably reduce the dimension of the random variables to simulate at each step.

ϟ When a full conditional in a Gibbs sampler cannot be simulated directly, it is sufficient to run instead a single step of any MCMC algorithm associated with this full conditional. The theoretical validation is the same as with any MCMC sampler. In the event a slice sampler is used for this purpose, the auxiliary variable is simply added to the vector of parameters.

Example 7.12. A benchmark hierarchical example in the Gibbs sampling literature describes multiple failures of ten pumps in a nuclear plant, with the data given in Table 7.2. The modeling is based on the assumption that the number of

Table 7.2. Number of failures and times of observation of ten pumps in a nuclear plant (*source*: Gaver and O'Muircheartaigh, 1987).

Pump	1	2	3	4	5	6	7	8	9	10
Failures	5	1	5	14	3	19	1	1	4	22
Time	94.32	15.72	62.88	125.76	5.24	31.44	1.05	1.05	2.10	10.48

failures of the ith pump follows a Poisson process with parameter λ_i $(1 \leq i \leq 10)$. For an observation time t_i, the number of failures X_i is thus a Poisson $\mathcal{P}(\lambda_i t_i)$ random variable. The standard prior distributions are gamma distributions, which lead to the hierarchical model

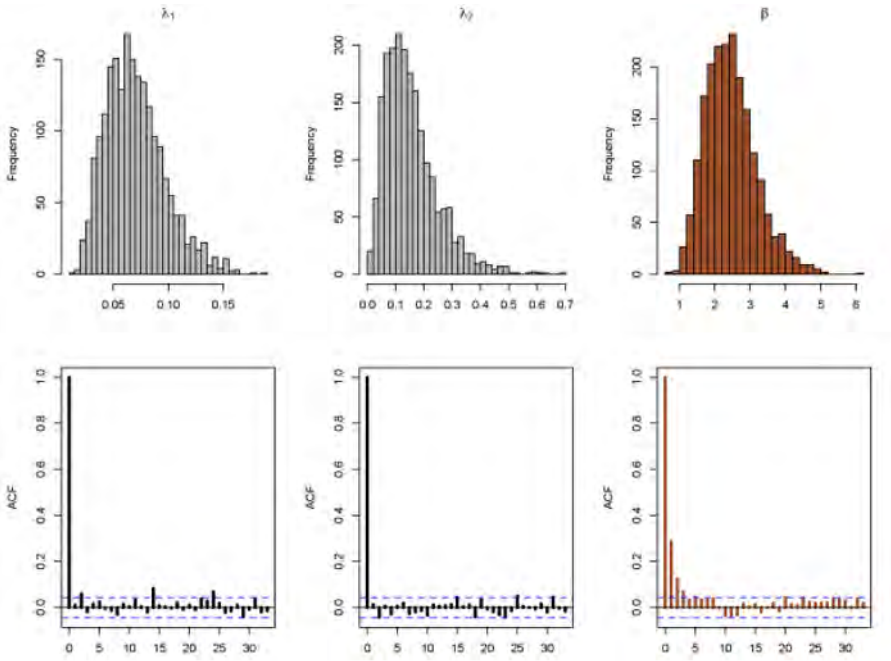

Fig. 7.10. Histograms of marginal distributions of λ_1, λ_2, and β from the pump failure data of Example 7.12. The corresponding bottom panels are autocorrelation plots. The hyperparameter values are $\alpha = 1.8$, $\gamma = 0.01$, and $\delta = 1$.

$$\begin{aligned} X_i &\sim \mathcal{P}(\lambda_i t_i), \quad i = 1, \ldots 10, \\ \lambda_i &\sim \mathcal{G}(\alpha, \beta), \quad i = 1, \ldots 10, \\ \beta &\sim \mathcal{G}(\gamma, \delta). \end{aligned}$$

The joint distribution is thus

$$\begin{aligned} &\pi(\lambda_1, \ldots, \lambda_{10}, \beta | t_1, \ldots, t_{10}, p_1, \ldots, p_{10}) \\ &\propto \prod_{i=1}^{10} \left\{ (\lambda_i t_i)^{x_i} \; e^{-\lambda_i t_i} \; \lambda_i^{\alpha-1} e^{-\beta \lambda_i} \right\} \beta^{10\alpha} \beta^{\gamma-1} e^{-\delta\beta} \\ &\propto \prod_{i=1}^{10} \left\{ \lambda_i^{x_i+\alpha-1} \; e^{-(t_i+\beta)\lambda_i} \right\} \beta^{10\alpha+\gamma-1} e^{-\delta\beta}, \end{aligned}$$

leading to the full conditional distributions

$$\lambda_i | \beta, t_i, x_i \sim \mathcal{G}(x_i + \alpha, t_i + \beta), \quad i = 1, \ldots 10,$$

$$\beta | \lambda_1, \ldots, \lambda_{10} \sim \mathcal{G}\left(\gamma + 10\alpha, \delta + \sum_{i=1}^{10} \lambda_i \right).$$

The associated Gibbs sampler is quite straightforward, with core R code

```
> for(i in 2:Nsim){
+  for(j in 1:nx)
+   lambda[i,j]=rgamma(1,sh=xdata[j]+alpha,ra=Time[j]+beta[i-1])
+  beta[i]=rgamma(1,sh=gamma+nx*alpha,ra=delta+sum(lambda[i,]))}
```

The result of a run over 5000 iterations is shown in Figure 7.10. ◀

Exercise 7.12 One reason for collecting the pump failure data is to identify which pumps are more reliable.

a. Run the Gibbs sampler for the pump failure data and get 95% posterior credible intervals for the parameters λ_i.
b. Based on the analysis, can you identify any pumps that are more or less reliable than the others?
c. How does your answer in b. change as the hyperparameter values are varied?

7.6 Other considerations

In this last section, we look at a few issues that could arise in the implementation of a Gibbs sampler.

7.6.1 Reparameterization

Many factors contribute to the convergence properties of a Gibbs sampler. For example, convergence performance may be greatly affected by the choice of the coordinates (or, in other words, the parameterization). If the covariance matrix Σ of the target has a wide range of eigenvalues, the Gibbs sampler may be very slow to explore the entire range of the support of the target.

Example 7.13. Recall Example 7.1, where we saw a first Gibbs sampler for the bivariate normal in (7.1). For that bivariate normal distribution, Figure 7.11 shows the autocorrelation for $\rho = .3, .6, .9$. The higher correlation results in a sampler that will have more trouble exploring the entire space and thus require more iterations. It is also interesting to note that no matter what is the value of ρ, $X+Y$ and $X-Y$ are independent, and thus changing coordinates from (x, y) to $(x+y, x-y)$ would lead to an immediately converging Gibbs algorithm. ◀

Exercise 7.13 For the bivariate normal distribution (7.1):

a. prove that $X+Y$ and $X-Y$ are independent.