Chapter 6

Approximations

6.1 The grid method

When one is confronted with a low-dimensional problem with a continuous parameter, then it is usually easy to approximate the posterior density at a dense grid of points that covers the relevant part of the parameter space. We discuss the method for a one-dimensional parameter θ.

We suppose that the posterior is available in the unnormalized form

$$f_{\Theta|Y}(\theta \mid y) = \frac{1}{c(y)}\, q(\theta \mid y),$$

where we know how to evaluate the unnormalized density $q(\theta \mid y)$, but do not necessarily know the value of the normalizing constant $c(y)$.

Instead of the original parameter space, we consider a finite interval $[a, b]$, which should cover most of the mass of the posterior distribution. We divide $[a, b]$ evenly into N subintervals

$$B_i = [a + (i-1)h, a + ih], \qquad i = 1, \dots, N.$$

The width h of one subinterval is

$$h = \frac{b-a}{N}.$$

Let θ_i be the midpoint of the i'th subinterval,

$$\theta_i = a + (i - \frac{1}{2})h, \qquad i = 1, \dots, N.$$

We use the midpoint rule for numerical integration. This means that we approximate the integral over the i'th subinterval of any function g by the rule

$$\int_{B_i} g(\theta)\, \mathrm{d}\theta \approx h g(\theta_i).$$

Using the midpoint rule on each of the subintervals, we get the following approximation for the normalizing constant

$$\begin{aligned} c(y) &= \int q(\theta \mid y)\,\mathrm{d}\theta \approx \int_a^b q(\theta \mid y)\,\mathrm{d}\theta = \sum_{i=1}^N \int_{B_i} q(\theta \mid y)\,\mathrm{d}\theta \\ &\approx h \sum_{i=1}^N q(\theta_i \mid y) \end{aligned}$$

Using this approximation, we can approximate the value of the posterior density at the point θ_i,

$$f_{\Theta|Y}(\theta_i \mid y) = \frac{1}{c(y)}\, q(\theta_i \mid y) \approx \frac{1}{h} \frac{q(\theta_i \mid y)}{\sum_{j=1}^N q(\theta_j \mid y)}. \tag{6.1}$$

We also obtain approximations for the posterior probabilities of the subintervals,

$$\begin{aligned} P(\Theta \in B_i \mid Y = y) &= \int_{B_i} f_{\Theta|Y}(\theta \mid y)\,\mathrm{d}\theta \approx h f_{\Theta|Y}(\theta_i \mid y) \\ &\approx \frac{q(\theta_i \mid y)}{\sum_{j=1}^N q(\theta_j \mid y)}. \end{aligned} \tag{6.2}$$

These approximations can be surprisingly accurate even for moderate values of N provided we are able to identify an interval $[a, b]$, which covers the essential part of posterior distribution. The previous formulas give means for plotting the posterior density and simulating from it. This is the grid method for approximating or simulating the posterior distribution.

- First evaluate the unnormalized posterior density $q(\theta \mid y)$ at a regular grid of points $\theta_1, \dots, \theta_N$ with spacing h. The grid should cover the main support of the posterior density.

- If you want to plot the posterior density, normalize these values by dividing by their sum and additionally by the bin width h as in eq. (6.1). This gives an approximation to the posterior ordinates $p(\theta_i \mid y)$ at the grid points θ_i.

- If you want a sample from the posterior, sample with replacement from the grid points θ_i with probabilities proportional to the numbers $q(\theta_i \mid y)$, cf. (6.2).

The midpoint rule is considered a rather crude method of numerical integration. In the numerical analysis literature, there are available much more sophisticated methods of numerical integration (or numerical quadrature) and they can be used in a similar manner. Besides dimension one, these kinds of approaches can be used in dimensions two or three. However, as the dimensionality of the parameter space grows, computing at every point in a dense multidimensional grid becomes more and more expensive.

6.2 Normal approximation to the posterior

We can try to approximate a multivariate posterior density by a multivariate normal density based on the behavior of the posterior density at its mode.

This approximation can be quite accurate, when the sample sizes is large, and when the posterior is unimodal. We will call the resulting approximation a normal approximation to the posterior, but the result is sometimes also called Laplace approximation or modal approximation. A normal approximation can be used directly as an approximate description of the posterior. However, such an approximation can be utilized also indirectly, e.g., to form a good proposal distribution for the Metropolis–Hastings method.

We first discuss normal approximation in the univariate situation. The statistical model has a single parameter θ, which has a continuous distribution. Let the unnormalized posterior density be given by $q(\theta \mid y)$. The normalizing constant of the posterior density can be unknown. We consider the case, where $\theta \mapsto q(\theta \mid y)$ is unimodal: i.e., it has only one local maximum. We suppose that we have located the mode $\hat{\theta}$ of $q(\theta \mid y)$. Actually, $\hat{\theta}$ depends on the data y, but we suppress this dependence in our notation. Usually we would have to run some numerical optimization algorithm in order to find the mode.

The basic idea of the method is to use the second degree Taylor polynomial of the logarithm of the posterior density centered on the mode $\hat{\theta}$,

$$\log f_{\Theta|Y}(\theta \mid y) \approx \log f_{\Theta|Y}(\hat{\theta} \mid y) + b(\theta - \hat{\theta}) - \frac{1}{2}A(\theta - \hat{\theta})^2, \tag{6.3}$$

where

$$b = \frac{\partial}{\partial\theta} \log f_{\Theta|Y}(\theta \mid y)\bigg|_{\theta=\hat{\theta}} = \frac{\partial}{\partial\theta} \log q(\theta \mid y)\bigg|_{\theta=\hat{\theta}} = 0,$$

and

$$A = -\frac{\partial^2}{\partial\theta^2} \log f_{\Theta|Y}(\theta \mid y)\bigg|_{\theta=\hat{\theta}} = -\frac{\partial^2}{\partial\theta^2} \log q(\theta \mid y)\bigg|_{\theta=\hat{\theta}}.$$

Notice the following points.

- The first and higher order (partial) derivatives with respect to θ of $\log q(\theta \mid y)$ and $\log f_{\Theta|Y}(\theta \mid y)$ agree, since these function differ only by an additive constant (which depends on y but not on θ).
- The first order term of the Taylor expansion disappears, since $\hat{\theta}$ is also the mode of $\log f_{\Theta|Y}(\theta \mid y)$.
- $A \geq 0$, since $\hat{\theta}$ is a maximum of $q(\theta \mid y)$. For the following, we need to assume that $A > 0$.

Taking the exponential of the second degree Taylor approximation (6.3), we see that we may approximate the posterior by the function

$$\pi_{\text{approx}}(\theta) \propto \exp\left(-\frac{A}{2}(\theta - \hat{\theta})^2\right),$$

at least in the vicinity of the mode $\hat{\theta}$. Luckily, we recognize that $\pi_{\text{approx}}(\theta)$ is an unnormalized form of the density of the normal distribution with mean $\hat{\theta}$ and variance $1/A$. The end result is that the posterior distribution can be approximated with the normal distribution

$$N\left(\hat{\theta}, \frac{1}{-L''(\hat{\theta})}\right), \tag{6.4}$$

where

$$L(\theta) = \log q(\theta \mid y)$$

and $L''(\hat{\theta})$ is the second derivative of $L(\theta)$ evaluated at the mode $\hat{\theta}$.

The multivariate analog of the result starts with the second degree expansion of the log-posterior centered on the mode $\hat{\theta}$,

$$\log f_{\Theta|Y}(\theta \mid y) \approx \log f_{\Theta|Y}(\hat{\theta} \mid y) + 0 - \frac{1}{2}(\theta - \hat{\theta})^T A(\theta - \hat{\theta}),$$

where A is minus the Hessian matrix of $L(\theta) = \log q(\theta \mid y)$ evaluated at the mode,

$$A_{ij} = -\left.\frac{\partial^2}{\partial\theta_i \partial\theta_j} \log f_{\Theta|Y}(\theta \mid y)\right|_{\theta=\hat{\theta}} = -\left.\frac{\partial^2}{\partial\theta_i \partial\theta_j} L(\theta)\right|_{\theta=\hat{\theta}} = -\left[\frac{\partial^2}{\partial\theta\, \partial\theta^T} L(\theta)\right]_{ij}$$

The first degree term of the expansion vanishes, since $\hat{\theta}$ is the mode of the log-posterior. Here A is at least positively semidefinite, since $\hat{\theta}$ is a maximum. If A is positively definite, we can proceed with the normal approximation.

Exponentiating, we find out that approximately (at least in the vicinity of the mode)

$$f_{\Theta|Y}(\theta \mid y) \propto \exp\left(-\frac{1}{2}(\theta - \hat{\theta})^T A(\theta - \hat{\theta})\right).$$

Therefore we can approximate the posterior with the corresponding multivariate normal distribution with mean $\hat{\theta}$ and covariance matrix given by A^{-1}, i.e., the approximating normal distribution is

$$N\left(\hat{\theta}, \left(-L''(\hat{\theta})\right)^{-1}\right), \tag{6.5}$$

where $L''(\hat{\theta})$ is the Hessian matrix of $L(\theta) = \log q(\theta \mid y)$ evaluated at the mode $\hat{\theta}$. The precision matrix of the approximating normal distribution is minus the Hessian of the log-posterior (evaluated at the mode), and hence the covariance matrix is minus the inverse of the Hessian.

If the (unnormalized) posterior has K modes $\hat{\theta}_1, \ldots, \hat{\theta}_K$, which are well separated, then Gelman *et al.* [1, Ch. 12.2] propose that the posterior could be approximated by the normal mixture

$$\frac{1}{C}\sum_{k=1}^{K} q(\hat{\theta}_k \mid y) \exp\left(-\frac{1}{2}(\theta - \hat{\theta}_k)^T[-L''(\hat{\theta}_k)](\theta - \hat{\theta}_k)\right). \tag{6.6}$$

This approximation is reasonable, if

$$j \neq k \quad \Rightarrow \quad \exp\left(-\frac{1}{2}(\hat{\theta}_j - \hat{\theta}_k)^T[-L''(\hat{\theta}_k)](\hat{\theta}_j - \hat{\theta}_k)\right) \approx 0$$

The normalizing constant in the normal mixture approximation (6.6) is

$$C = \sum_{k=1}^{K} q(\hat{\theta}_k \mid y)\; (2\pi)^{d/2} (\det(-L''(\hat{\theta}_k))^{-1/2},$$

where d is the dimensionality of θ.

Before using the normal approximation, it is often advisable to reparameterize the model so that the transformed parameters are defined on the whole real line and have roughly symmetric distributions. E.g., one can use logarithms of positive parameters and apply the logit function to parameters which take values on the interval $(0, 1)$. The normal approximation is then constructed for the transformed parameters, and the approximation can then be translated back to the original parameter space. One must, however, remember to multiply by the appropriate Jacobians.

Example 6.1. We consider the unnormalized posterior

$$q(\theta \mid y) = \theta^{y_4}\,(1-\theta)^{y_2+y_3}\,(2+\theta)^{y_1}, \qquad 0 < \theta < 1,$$

where $y = (y_1, y_2, y_3, y_4) = (13, 1, 2, 3)$. The mode and the second derivative of $L(\theta) = \log q(\theta \mid y)$ evaluated at the mode are given by

$$\hat{\theta} \approx 0.677, \qquad L''(\hat{\theta}) \approx -37.113.$$

(The mode $\hat{\theta}$ can be found by solving a quadratic equation.) The resulting normal approximation in the original parameter space is $N(0.677, 1/37.113)$.

We next reparametrize by defining ϕ as the logit of θ,

$$\phi = \operatorname{logit}(\theta) = \ln \frac{\theta}{1-\theta} \quad \Leftrightarrow \quad \theta = \frac{e^{\phi}}{1+e^{\phi}}.$$

The given unnormalized posterior for θ transforms to the following unnormalized posterior for ϕ,

$$\begin{aligned} \tilde{q}(\phi \mid y) &= q(\theta \mid y) \left| \frac{d\theta}{d\phi} \right| \\ &= \left(\frac{e^{\phi}}{1+e^{\phi}} \right)^{y_4} \left(\frac{1}{1+e^{\phi}} \right)^{y_2+y_3} \left(\frac{2+3e^{\phi}}{1+e^{\phi}} \right)^{y_1} \frac{e^{\phi}}{(1+e^{\phi})^2}. \end{aligned}$$

The mode and the second derivative of $\tilde{L}(\phi) = \log \tilde{q}(\phi \mid y)$ evaluated at the mode are given by

$$\hat{\phi} \approx 0.582, \qquad \tilde{L}''(\hat{\phi}) \approx -2.259.$$

(Also $\hat{\phi}$ can be found by solving a quadratic.) This results in the normal approximation $N(0.582, 1/2.259)$ for the logit of θ.

When we translate that approximation back to the original parameter space, we get the approximation

$$f_{\Theta|Y}(\theta \mid y) \approx N(\phi \mid 0.582, 1/2.259) \left| \frac{d\phi}{d\theta} \right|,$$

i.e.,

$$f_{\Theta|Y}(\theta \mid y) \approx N(\operatorname{logit}(\theta) \mid 0.582, 1/2.259)\,\frac{1}{\theta(1-\theta)}.$$

Both of these approximations are plotted in Figure 6.1 together with the true posterior density (whose normalizing constant can be found exactly using computer algebra). △

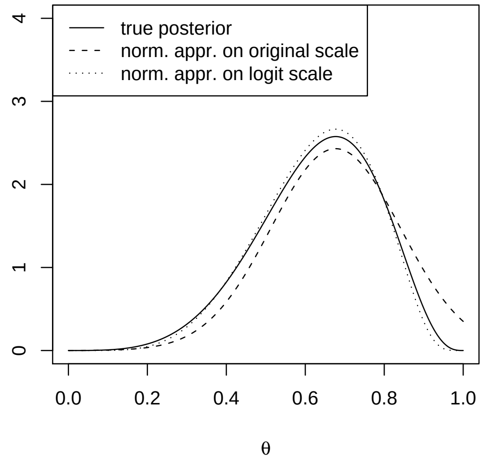


Figure 6.1: The exact posterior density (solid line) together with its normal approximation (dashed line) and the approximation based on the normal approximation for the logit of θ. The last approximation is markedly non-normal on the original scale, and it is able to capture the skewness of the true posterior density.

6.3 Posterior expectations using Laplace approximation

Laplace showed in the 1770's how one can form approximations to integrals of highly peaked positive functions by integrating analytically a suitable normal approximation. We will now apply this idea to build approximations to posterior expectations. We assume that the posterior density is highly peaked while the function h, whose posterior expectation we seek is relatively flat. To complicate matters, the posterior density is typically known only in the unnormalized form $q(\theta \mid y)$, and then

$$E[h(\Theta) \mid Y = y] = \frac{\int h(\theta)\, q(\theta \mid y)\, \mathrm{d}\theta}{\int q(\theta \mid y)\, \mathrm{d}\theta}. \tag{6.7}$$

Tierney and Kadane [5] approximated separately the numerator and the denominator of eq. (6.7) using Laplace's method, and analyzed the resulting error.

To introduce the idea of Laplace's approximation (or Laplace's method), consider a highly peaked function $L(\theta)$ of a scalar variable θ such that $L(\theta)$ has a unique mode (i.e., a maximum) at $\hat{\theta}$. Suppose that $g(\theta)$ is a function, which varies slowly. We seek an approximation to the integral

$$I = \int g(\theta)\, \mathrm{e}^{L(\theta)}\, \mathrm{d}\theta. \tag{6.8}$$

Heuristically, the integrand is negligible when we go far away from $\hat{\theta}$, and so we should be able to approximate the integral I by a simpler integral, where we take into account only the local behavior of $L(\theta)$ around its mode. To this end, we first approximate $L(\theta)$ by its second degree Taylor polynomial centered at the mode $\hat{\theta}$,

$$L(\theta) \approx L(\hat{\theta}) + 0 \cdot (\theta - \hat{\theta}) + \frac{1}{2} L''(\hat{\theta})(\theta - \hat{\theta})^2.$$

Since $g(\theta)$ is slowly varying, we may approximate the integrand as follows

$$g(\theta)\, \mathrm{e}^{L(\theta)} \approx g(\hat{\theta}) \exp\left(L(\hat{\theta}) - \frac{1}{2}\tau^2(\theta - \hat{\theta})^2) \right),$$

where

$$\tau^2 = -L''(\hat{\theta}).$$

For the following, we must assume that $L''(\hat{\theta}) < 0$. Integrating the approximation, we obtain

$$\begin{aligned} I &\approx \int g(\hat{\theta})\, \mathrm{e}^{L(\hat{\theta})} \exp(-\frac{1}{2}\tau^2(\theta - \hat{\theta})^2)\, \mathrm{d}\theta \\ &= \frac{\sqrt{2\pi}}{\tau}\, g(\hat{\theta})\, \mathrm{e}^{L(\hat{\theta})} \end{aligned}$$

This is Laplace's approximation. (Actually, it is only the leading term in a Laplace expansion, which is an asymptotic expansion for the integral.)

To handle the multivariate result, we use the normalizing constant of the $N_d(\mu, Q^{-1})$ distribution to evaluate the integral

$$\int \exp\left(-\frac{1}{2}(x-\mu)^T Q (x - \mu) \right) \mathrm{d}x = \frac{(2\pi)^{d/2}}{\sqrt{\det Q}}. \tag{6.9}$$

This result is valid for any symmetric and positive definite $d \times d$ matrix Q. Integrating the multivariate second degree approximation of $g(\theta) \exp(L(\theta))$, we obtain

$$I = \int g(\theta)\, \mathrm{e}^{L(\theta)}\, \mathrm{d}\theta \approx \frac{(2\pi)^{d/2}}{\sqrt{\det(Q)}}\, g(\hat{\theta})\, \mathrm{e}^{L(\hat{\theta})}, \tag{6.10}$$

where d is the dimensionality of θ, and Q is minus the Hessian of L evaluated at the mode,

$$Q = -L''(\hat{\theta}),$$

and we must assume that the $d \times d$ matrix Q is positively definite.

Using these tools, we can approximate the posterior expectation (6.7) in several different ways. One idea is to approximate the numerator by choosing

$$g(\theta) = h(\theta), \qquad \mathrm{e}^{L(\theta)} = q(\theta \mid y)$$

in eq. (6.10), and then to approximate the denominator by choosing

$$g(\theta) \equiv 1, \qquad \mathrm{e}^{L(\theta)} = q(\theta \mid y).$$

These choices yield the approximation

$$E[h(\Theta) \mid Y = y] \approx \frac{\dfrac{(2\pi)^{d/2}}{\sqrt{\det(Q)}}\, h(\hat{\theta})\, \mathrm{e}^{L(\hat{\theta})}}{\dfrac{(2\pi)^{d/2}}{\sqrt{\det(Q)}}\, \mathrm{e}^{L(\hat{\theta})}} = h(\hat{\theta}), \tag{6.11}$$

where

$$\hat{\theta} = \arg\max L(\theta), \qquad Q = -L''(\hat{\theta}).$$

Here we need a single maximization, and do not need to evaluate the Hessian at all.

A less obvious approach is to choose

$$g(\theta) \equiv 1, \qquad \mathrm{e}^{L(\theta)} = h(\theta)\, q(\theta \mid y)$$

to approximate the numerator, and

$$g(\theta) \equiv 1, \qquad \mathrm{e}^{L(\theta)} = q(\theta \mid y)$$

to approximate the denominator. Here we need to assume that h is a positive function, i.e., $h > 0$. The resulting approximation is

$$E[h(\Theta) \mid Y = y] \approx \left(\frac{\det(Q)}{\det(Q^*)}\right)^{1/2} \frac{h(\hat{\theta}^*)\, q(\hat{\theta}^* \mid y)}{q(\hat{\theta} \mid y)}, \tag{6.12}$$

where

$$\hat{\theta}^* = \arg\max[h(\theta)\, q(\theta \mid y)], \qquad \hat{\theta} = \arg\max q(\theta \mid y).$$

and Q^* and Q are the minus Hessians

$$Q^* = -L^{*''}(\hat{\theta}^*), \qquad Q = -L''(\hat{\theta}),$$

where

$$L^*(\theta) = \log(h(\theta)\, q(\theta \mid y)), \qquad L(\theta) = \log q(\theta \mid y).$$

We need two separate maximizations and need to evaluate two Hessians for this approximation.

Tierney and Kadane analyzed the errors committed in these approximations in the situation, where we have n (conditionally) i.i.d. observations, and the sample size n grows. The first approximation (6.11) has relative error of order $O(n^{-1})$, while the second approximation (6.12) has relative error of order $O(n^{-2})$. That is,

$$E[h(\Theta) \mid Y = y] = h(\hat{\theta})\ \big(1 + O(n^{-1})\big)$$

and

$$E[h(\Theta) \mid Y = y] = \left(\frac{\det(Q)}{\det(Q^*)}\right)^{1/2} \frac{h(\hat{\theta}^*)\, q(\hat{\theta}^* \mid y)}{q(\hat{\theta} \mid y)}\ \big(1 + O(n^{-2})\big)\,.$$

Hence the second approximation is much more accurate (at least asymptotically).

6.4 Posterior marginals using Laplace approximation

Tierney and Kadane discuss also an approximation to the marginal posterior, when the parameter vector θ is composed of two vector components $\theta = (\phi, \psi)$.

The form of the approximation is easy to derive, and was earlier discussed by Leonard [2]. However, Tierney and Kadane [5, Sec. 4] were the first to analyze the error in this Laplace approximation. We first derive the form of the approximation, and then make some comments on the error terms based on the discussion of Tierney and Kadane.

Let $q(\phi, \psi \mid y)$ be an unnormalized form of the posterior density, based on which we try to approximate the normalized marginal posterior $p(\phi \mid y)$. Let the dimensions of ϕ and ψ be d_1 and d_2, respectively. We have

$$p(\phi \mid y) = \int p(\phi, \psi \mid y) \, \mathrm{d}\psi = \int \exp(\log p(\phi, \psi \mid y)) \, \mathrm{d}\psi,$$

where $p(\phi, \psi \mid y)$ is the normalized posterior. The main difference with approximating a posterior expectation is the fact, that now we are integrating only over the component(s) ψ of $\theta = (\phi, \psi)$.

Fix the value of ϕ for the moment. Let $\psi^*(\phi)$ be the maximizer of the function

$$\psi \mapsto \log p(\phi, \psi \mid y),$$

and let $Q(\phi)$ be minus the Hessian matrix of this function evaluated at $\psi = \psi^*(\phi)$. Notice that we can equally well calculate $\psi^*(\phi)$ and $Q(\phi)$ as the maximizer and minus the $d_2 \times d_2$ Hessian matrix of $\psi \mapsto \log q(\phi, \psi \mid y)$, respectively,

$$\begin{aligned} \psi^*(\phi) &= \arg\max_{\psi} \left(\log q(\phi, \psi \mid y)\right) = \arg\max_{\psi} q(\phi, \psi \mid y) & (6.13) \\ Q(\phi) &= -\left[\frac{\partial^2}{\partial \psi \, \partial \psi^T} \log q(\phi, \psi \mid y)\right]_{|\psi = \psi^*(\phi)}. & (6.14) \end{aligned}$$

For fixed ϕ, we have the second degree Taylor approximation in ψ,

$$\log p(\phi, \psi \mid y) \approx \log p(\phi, \psi^*(\phi) \mid y) - \frac{1}{2}(\psi - \psi^*(\phi))^T Q(\phi)(\psi - \psi^*(\phi)), \quad (6.15)$$

and we assume that matrix $Q(\phi)$ is positive definite.

Next we integrate the exponential function of the approximation (6.15) with respect to ψ, with the result

$$p(\phi \mid y) \approx p(\phi, \psi^*(\phi) \mid y) \, (2\pi)^{d_2/2} \, (\det Q(\phi))^{-1/2}.$$

To evaluate this approximation, we need the normalizing constant of the unnormalized posterior $q(\phi, \psi \mid y)$, which we obtain by another Laplace approximation, and the end result is

$$p(\phi \mid y) \approx (2\pi)^{-d_1/2} \, q(\phi, \psi^*(\phi) \mid y) \sqrt{\frac{\det Q}{\det Q(\phi)}}, \quad (6.16)$$

where Q is minus the $(d_1 + d_2) \times (d_1 + d_2)$ Hessian of the function

$$(\phi, \psi) \mapsto \log q(\phi, \psi \mid y)$$

evaluated at the MAP, the maximum point of the same function. However, it is often enough to approximate the functional form of the marginal posterior. When considered as a function of ϕ, we have, approximately,

$$p(\phi \mid y) \propto q(\phi, \psi^*(\phi) \mid y) \, (\det Q(\phi))^{-1/2}. \quad (6.17)$$

The unnormalized Laplace approximation (6.17) can be given another interpretation (see, e.g., [3, 4]). By the multiplication rule,

$$p(\phi \mid y) = \frac{p(\phi, \psi \mid y)}{p(\psi \mid \phi, y)} \propto \frac{q(\phi, \psi \mid y)}{p(\psi \mid \phi, y)}.$$

This result is valid for any choice of ψ. Let us now form a normal approximation for the denominator for a fixed value of ϕ, i.e.,

$$p(\psi \mid \phi, y) \approx N(\psi \mid \psi^*(\phi), Q(\phi)^{-1}).$$

However, this approximation is accurate only in the vicinity of the mode $\psi^*(\phi)$, so let us use it only at the mode. The end result is the following approximation,

$$\begin{aligned} p(\phi \mid y) &\propto \left[\frac{q(\phi, \psi \mid y)}{N(\psi \mid \psi^*(\phi), Q(\phi)^{-1})} \right]_{|\psi = \psi^*(\phi)} \\ &= (2\pi)^{d_2/2} \det(Q(\phi))^{-1/2} \, q(\phi, \psi^*(\phi) \mid y) \\ &\propto q(\phi, \psi^*(\phi) \mid y) \, (\det Q(\phi))^{-1/2}, \end{aligned}$$

which is the same as the unnormalized Laplace approximation (6.17) to the marginal posterior of ϕ.

Tierney and Kadane show that the relative error in the approximation (6.16) is of the order $O(n^{-1})$, when we have n (conditionally) i.i.d. observations, and that most of the error comes from approximating the normalizing constant. They argue that the approximation (6.17) captures the correct functional form of the marginal posterior with relative error $O(n^{-3/2})$ and recommend that one should therefore use the unnormalized approximation (6.17), which can then be normalized by numerical integration, if need be. For instance, if we want to simulate from the approximate marginal posterior, then we can use the unnormalized approximation (6.17) directly, together with accept–reject, SIR or the grid-based simulation method of Sec. 6.1. See the articles by H. Rue and coworkers [3, 4] for imaginative applications of these ideas.

Another possibility for approximating the marginal posterior would be to build a normal approximation to the joint posterior, and then marginalize. However, a normal approximation to the marginal posterior would only give the correct result with absolute error of order $O(n^{-1/2})$, so the accuracies of both of the Laplace approximations are much better. Since the Laplace approximations yield good relative instead of absolute error, the Laplace approximations maintain good accuracy also in the tails of the densities. In contrast, the normal approximation is accurate only in the vicinity of the mode.

Example 6.2. Consider normal observations

$$[Y_i \mid \mu, \tau] \overset{\text{i.i.d.}}{\sim} N(\mu, \frac{1}{\tau}), \qquad i = 1, \ldots, n,$$

together with the non-conjugated prior

$$p(\mu, \tau) = p(\mu) \, p(\tau) = N(\mu \mid \mu_0, \frac{1}{\psi_0}) \; \mathrm{Gam}(\tau \mid a_0, b_0).$$

The full conditional of μ is readily available,

$$p(\mu \mid \tau, y) = N(\mu \mid \mu_1, \frac{1}{\psi_1})$$

where

$$\psi_1 = \psi_0 + n\tau \qquad \psi_1\,\mu_1 = \psi_0\,\mu_0 + \tau \sum_{i=1}^{n} y_i$$

The mode of the full conditional $p(\mu \mid \tau, y)$ is

$$\mu^*(\tau) = \mu_1 = \frac{\psi_0\,\mu_0 + \tau \sum_{i=1}^{n} y_i}{\psi_0 + n\tau}.$$

We now use this knowledge to build a Laplace approximation to the marginal posterior of τ.

Since, as a function of μ,

$$p(\mu, \tau \mid y) \propto p(\mu \mid \tau, y),$$

$\mu^*(\tau)$ is also the mode of $p(\mu, \tau \mid y)$ for any τ. We also need the second derivative

$$\frac{\partial^2}{\partial \mu^2}\left(\log p(\mu, \tau \mid y)\right) = \frac{\partial^2}{\partial \mu^2}\left(\log p(\mu \mid \tau, y)\right) = -\psi_1,$$

for $\mu = \mu^*(\tau)$, but the derivative does not in this case depend on the value of μ at all. An unnormalized form of the Laplace approximation to the marginal posterior of τ is therefore

$$p(\tau \mid y) \propto \frac{q(\mu^*(\tau), \tau \mid y)}{\sqrt{\psi_1}}, \quad \text{where} \quad q(\mu, \tau \mid y) = p(y \mid \mu, \tau)\,p(\mu)\,p(\tau).$$

In this toy example, the Laplace approximation (6.17) for the functional form of the marginal posterior $p(\tau \mid \mu)$ is exact, since by the multiplication rule,

$$p(\tau \mid y) = \frac{p(\mu, \tau \mid y)}{p(\mu \mid \tau, y)}$$

for any choice of μ, in particular for $\mu = \mu^*(\tau)$. Here the numerator is known only in an unnormalized form.

Figure 6.2 (a) illustrates the result using data $y = (-1.4, -1.6, -2.4, 0.7, 0.6)$ and hyperparameters $\mu_0 = 0$, $\psi_0 = 0.5$, $a_0 = 1$, $b_0 = 0.1$. The unnormalized (approximate) marginal posterior has been drawn using the grid method of Sec. 6.1. Figure 6.2 (b) shows an i.i.d. sample drawn from the approximate posterior

$$\tilde{p}(\tau \mid y)\,p(\mu \mid \tau, y),$$

where $\tilde{p}(\tau \mid y)$ is a histogram approximation to the true marginal posterior $p(\tau \mid y)$, which has been sampled using the grid method.

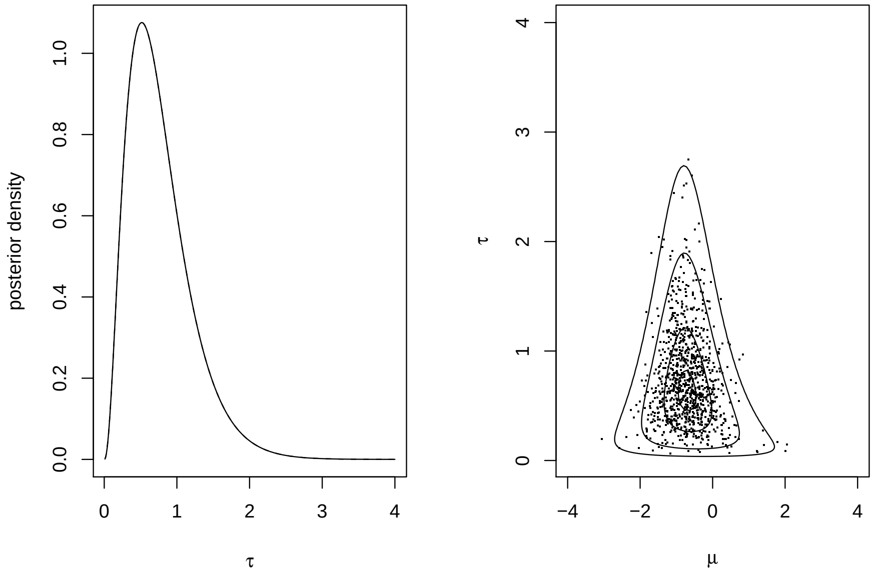
△

Bibliography

[1] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC Press, 2nd edition, 2004.

[2] Tom Leonard. A simple predictive density function: Comment. *Journal of the American Statistical Association*, 77:657–658, 1982.

Figure 6.2: (a) Marginal posterior density of τ and (b) a sample drawn from the approximate joint posterior together with contours of the true joint posterior density.



(a)

(b)

[3] H. Rue and S. Martino. Approximate Bayesian inference for hierarchical Gaussian Markov random fields models. *Journal of Statistical Planning and Inference*, 137(10):3177–3192, 2007.

[4] H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models using integrated nested Lapalce approximations. *Journal of the Royal Statistical Society: Series B*, 2009. to appear.

[5] Luke Tierney and Joseph B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81:82–86, 1986.

Chapter 7

MCMC algorithms

7.1 Introduction

In a complicated Bayesian statistical model it may be very difficult to analyze the mathematical form of the posterior and it may be very difficult to draw an i.i.d. sample from it. Fortunately, it is often easy to generate a correlated sample, which approximately comes from the posterior distribution. (In this context, the word *correlated* means *not independent*). However, we would very much prefer to have an i.i.d. sample from the posterior, instead. After one has available a sample, one can estimate posterior expectations and posterior quantiles using the same kind of techniques that are used with i.i.d. samples. This is the idea behind Markov chain Monte Carlo (MCMC) methods.

In this chapter we will introduce the basic MCMC sampling algorithms that are used in practical problems. The emphasis is on trying to understand what one needs to do in order to implement the algorithms. In Chapter 11 we will see why these algorithms work using certain concepts from the theory of Markov chains in a general state space.

There are available computer programs that can implement an MCMC simulation automatically. Perhaps the most famous such program is the BUGS system (Bayesian inference Using Gibbs Sampling), which has several concrete implementations, most notably WinBUGS and OpenBUGS. You can analyze most of the models of interest easily using BUGS. What the user of BUGS needs to do is to write the description of the model in a format that BUGS understands, read the data into the program, and then let the program do the simulation. Once the simulation has finished, one can let the program produce various summaries of the posterior. Using such a tool, it is simple to experiment with different priors and different likelihoods for the same data.

However, in this chapter the emphasis is on understanding how you can write your own MCMC programs. Why would this be of interest?

- If you have not used MCMC before, you get a better understanding of the methods if you try to implement (some of) them yourself.
- For some models, the automated tools fail. Sometimes you can, however, rather easily design and implement a MCMC sampler yourself, once you understand the basic principles. (In some cases, however, designing an efficient MCMC sampler can be an almost impossibly difficult task.)

- Sometimes you want to have more control over the sampling algorithm than is provided by the automated tools. In some cases implementation details can make a big difference to the efficiency of the method.

The most famous MCMC methods are the Metropolis–Hastings sampler and the Gibbs sampler. Where do these names come from?

- Nicholas (Nick) Metropolis (1915–1999) was an American mathematician, physicist and pioneer of computing, who was born in Greece. He published the Metropolis sampler in 1953 jointly with two husband-and-wife teams, namely A.W. and M.N. Rosenbluth and A.H. and E. Teller. At that time the theory of general state space Markov chains was largely unexplored. In spite of this, the authors managed to give a heuristic proof for the validity of the method.

- W. Keith Hastings (1930–) is a Canadian statistician, who published the Metropolis–Hastings sampler in 1970. It is a generalization of the Metropolis sampler. Hastings presented his algorithm using a discrete state space formalism, since the theory of general state space Markov chains was then known only to some specialists in probability theory. Hastings' article did not have a real impact on statisticians until much later.

- The name Gibbs sampler was introduced by the brothers S. and D. Geman in an article published in 1984. Related ideas were published also by other people at roughly the same time. The method is named after the American mathematician and physicist J. Willard Gibbs (1893–1903), who studied thermodynamics and statistical physics, but did not have anything to do with MCMC.

In the late 1980's and early 1990's there was an explosion in the number of studies, where people used MCMC methods in Bayesian inference. Now there was available enough computing power to apply the methods, and besides, the theory of general state space Markov chains had matured so that readable expositions of the theory were available.

Nowadays, many statisticians routinely use the concept of a Markov chain which evolves in a general state space. Unfortunately, their mathematical theory is still explained only in a handful of text books.

7.2 Basic ideas of MCMC

MCMC algorithms are based on the idea of a Markov chain which evolves in discrete time. A Markov chain is a stochastic process

$$\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \dots$$

Here $\theta^{(i)}$ (the state of the process at time i) is a RV whose values lie in a state space, which usually is a subset of some Euclidean space $\mathbb{R}^d$. The state space is the same for all times i. We write the time index as a superscript so that we can index the components $\theta^{(i)}$ using a subscript.

Markov chains have the following **Markov property**: the distribution of the next state $\theta^{(i+1)}$ depends on the history $\theta^{(0)}, \theta^{(1)}, \dots, \theta^{(i)}$ only through the present state $\theta^{(i)}$. The Markov chains used in MCMC methods are **homogeneous**: the conditional distribution of $\theta^{(i+1)}$ given $\theta^{(i)}$ does not depend on the index i.

The following algorithm shows how one can simulate a Markov chain. Intuitively, a Markov chain is nothing else but the mathematical idealization of this simulation algorithm.

Algorithm 14: Simulating a Markov chain.

```
Generate $\theta^{(0)}$ from a given initial distribution;
for $i = 0, 1, 2, \ldots$ do
    Compute the next state $\theta^{(i+1)}$ using some rule, where you can use the
    present state $\theta^{(i)}$ (but no earlier states) and freshly generated random
    numbers.
end
```

If the rule for calculating the next state does not change depending on the value of the loop index i, then the generated Markov chain is homogeneous.

Some (but not all) Markov chains have an **invariant distribution** (or a stationary distribution or equilibrium distribution), which can be defined as follows. If the initial state of the chain $\theta^{(0)}$ follows the invariant distribution, then also all the subsequent states $\theta^{(i)}$ follow it.

If a Markov chain has an invariant distribution, then (under certain regularity conditions) the distribution of the state $\theta^{(i)}$ converges to that invariant distribution (in a certain sense). Under certain regularity conditions, such a chain is **ergodic**, which ensures that an arithmetic average (or an ergodic average) of the form

$$\frac{1}{N} \sum_{i=1}^{N} h(\theta^{(i)})$$

converges, almost surely, to the corresponding expectation calculated under the invariant distribution as $N \to \infty$. That is, the ergodic theorem for Markov chains then states that the strong law of large numbers holds, i.e.,

$$\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} h(\theta^{(i)}) \to E_f h(\Theta) = \int h(\theta) f(\theta) \, \mathrm{d}\theta, \tag{7.1}$$

where f is the density of the invariant distribution. This will then hold for all functions h for which the expectation $E_f h(\Theta)$ exists, so the convergence is as strong as in the strong law of large numbers for i.i.d. sequences. There are also more advanced forms of ergodicity (geometric ergodicity and uniform ergodicity), which a Markov chain may either have or not have.

Under still more conditions, Markov chains also satisfy a central limit theorem, which characterizes the speed of convergence in the ergodic theorem. The central limit theorem for Markov chains is of the form

$$\sqrt{N} \left(\frac{1}{N} \sum_{i=1}^{N} h(\theta^{(i)}) - E_f h(\Theta) \right) \xrightarrow{\mathrm{d}} N(0, \sigma_h^2).$$

The speed of convergence is of the same order of N as in the central limit theorem for i.i.d. sequences. However, estimating the variance σ_h^2 in the central limit theorem is lot trickier than with i.i.d. sequences.

After this preparation, it is possible to explain the basic idea of MCMC methods. The idea is to set up an ergodic Markov chain which has the posterior distribution as its invariant distribution. Doing this is often surprisingly easy. Then one simulates values

$$\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \dots$$

of the chain. When t is sufficiently large, then $\theta^{(t)}$ and all the subsequent states $\theta^{(t+i)}, i \geq 1$ follow approximately the posterior distribution. The time required for the chain to approximately achieve its invariant distribution is called the **burn-in**. After the initial burn-in period has been discarded, the subsequent values

$$\theta^{(t)}, \theta^{(t+1)}, \theta^{(t+2)}, \dots$$

can be treated as a dependent sample from the posterior distribution, and we can calculate posterior expectations, quantiles and other summaries of the posterior distribution based on this sample.

After the burn-in period we need to store the simulated values of the chain for later use. So, for a scalar parameter we need a vector to store the results, for a vector parameter we need a matrix to store the results and so on. To save space, one often decides to **thin** the sequences by keeping only every kth value of each sequence and by discarding the rest.

Setting up *some* MCMC algorithm for a given posterior is usually easy. However, the challenge is to find an MCMC algorithm which converges rapidly and then explores efficiently the whole support of the posterior distribution. Then one can get a reliable picture of the posterior distribution after stopping the simulation after a reasonable number of iterations.

In practice one may want to try several approaches for approximate posterior inference in order to become convinced that the posterior inferences obtained with MCMC are reliable. One can, e.g., study simplified forms of the statistical model (where analytical developments or maximum likelihood estimation or other asymptotic approximations to Bayesian estimation may be possible), simulate several chains which are initialized from different starting points and are possibly computed with different algorithms, and compute approximations to the posterior.

7.3 The Metropolis–Hastings algorithm

Now we consider a target distribution with density $\pi(\theta)$, which may be available only in an unnormalized form $\tilde{\pi}(\theta)$. Usually the target density is the posterior density of a Bayesian statistical model,

$$\pi(\theta) = p(\theta \mid y).$$

Actually we only need to know an unnormalized form of the posterior, which is given, e.g., in the form of prior times likelihood,

$$\tilde{\pi}(\theta) = p(\theta)\, p(y \mid \theta).$$

The density $\pi(\theta)$ may be a density in the generalized sense, so we may have a discrete distribution for some components of θ and a continuous distribution for others.

For the Metropolis–Hastings algorithm we need a proposal density $q(\theta' \mid \theta)$, from which we are able to simulate. (Some authors call the proposal density the jumping density or candidate generating density.) As a function of θ', the proposal density $q(\theta' \mid \theta)$ is a density on the parameter space for each value of θ. When the current state of the chain is $\theta = \theta^{(i)}$, we propose a value for the next state from the distribution with density

$$\theta' \mapsto q(\theta' \mid \theta)$$

The proposed value θ' is then accepted or rejected in the algorithm. If the proposal is accepted, then the next state $\theta^{(i+1)}$ is taken to be θ', but otherwise the chain stays in the same state, i.e., $\theta^{(i+1)}$ is assigned the current state $\theta^{(i)}$.

The acceptance condition has to be selected carefully so that we get the target distribution as the invariant distribution of the chain. The usual procedure works as follows. We calculate the value of the Metropolis–Hastings ratio (M–H ratio)

$$r = r(\theta', \theta) = \frac{\pi(\theta')\, q(\theta \mid \theta')}{\pi(\theta)\, q(\theta' \mid \theta),} \tag{7.2}$$

where $\theta = \theta^{(i)}$ is the current state and θ' is the proposed state. Then we generate a value u from the standard uniform Uni$(0, 1)$. If $u < r$, then we accept the proposal and otherwise reject it. For the analysis of the algorithm, it is essential to notice that the probability of accepting the proposed θ', when the current state is θ, is given by

$$\Pr(\text{proposed value is accepted} \mid \theta^{(i)} = \theta, \theta') = \min(1, r(\theta', \theta)). \tag{7.3}$$

We need here the minimum of one and the M–H ratio, since the M–H ratio may very well be greater than one.

Some explanations are in order.

- The denominator of the M–H ratio (7.2) is the joint density of the proposal θ' and the current state θ, when the current state already follows the posterior.

- The numerator is of the same form as the denominator, but θ and θ' have exchanged places.

- If $\pi(\theta^{(0)}) > 0$, then the denominator of the M–H ratio is always strictly positive during the algorithm. When $i = 0$ this follows from the observation that $q(\theta' \mid \theta^{(0)})$ has to be positive, since θ' is generated from that density. Also $\pi(\theta^{(1)})$ has to be positive, thanks to the form of the acceptance test. The rest follows by induction.

- We do not need to know the normalizing constant of the target distribution, since it cancels in the M–H ratio,

$$r = \frac{\pi(\theta')\, q(\theta \mid \theta')}{\pi(\theta)\, q(\theta' \mid \theta)} = \frac{\tilde{\pi}(\theta')\, q(\theta \mid \theta')}{\tilde{\pi}(\theta)\, q(\theta' \mid \theta)} \tag{7.4}$$

- If the target density is a posterior distribution, then the M–H ratio is given by

$$r = \frac{f_{Y|\Theta}(y \mid \theta')\, f_\Theta(\theta')\, q(\theta \mid \theta')}{f_{Y|\Theta}(y \mid \theta)\, f_\Theta(\theta)\, q(\theta' \mid \theta)}. \tag{7.5}$$

- Once you know what the notation is supposed to mean, you can use an abbreviated notation for the M–H ratio, such as

  $$r = \frac{p(\theta' \mid y)\, q(\theta \mid \theta')}{p(\theta \mid y)\, q(\theta' \mid \theta)}.$$

  Here, e.g., $p(\theta' \mid y)$ is the value of the posterior density evaluated at the proposal θ'.

An explanation of why the target distribution is the invariant distribution of the resulting Markov chain will be given in Chapter 11. Then it will become clear, that other formulas in place of eq. (7.2) would work, too. However, the formula (7.2) is known to be optimal (in a certain sense), and therefore it is the one that is used in practice.

In the Metropolis–Hastings algorithm the proposal density can be selected otherwise quite freely, but we must be sure that we can reach (with positive probability) any reasonably possible region in the parameter space starting from any initial state $\theta^{(0)}$ with a finite number of steps. This property is called **irreducibility** of the Markov chain.

Algorithm 15: The Metropolis–Hastings algorithm.

```
Input: An initial value $\theta^{(0)}$ such that $\tilde{\pi}(\theta^{(0)}) > 0$ and the number of
       iterations $N$.
Result: Values simulated from a Markov chain which has as its invariant
        distribution the distribution corresponding to the unnormalized
        density $\tilde{\pi}(\theta)$.
for $i = 0, 1, 2, \dots, N$ do
    $\theta \leftarrow \theta^{(i)}$;
    Generate $\theta'$ from $q(\cdot \mid \theta)$ and $u$ from $\mathrm{Uni}(0, 1)$;
    Calculate the M–H ratio
        $r = \frac{\tilde{\pi}(\theta')\, q(\theta \mid \theta')}{\tilde{\pi}(\theta)\, q(\theta' \mid \theta)}$
    Set
        $\theta^{(i+1)} \leftarrow \begin{cases} \theta', & \text{if } u < r \\ \theta, & \text{otherwise.} \end{cases}$
end
```

Algorithm 15 sums up the Metropolis–Hastings algorithm. When implementing the algorithm, one easily comes across problems, which arise because of underflow or overflow in the calculation of the M–H ratio r. Most of such problems can be cured by calculating with logarithms. E.g., when the target distribution is a posterior distribution, then one should first calculate $s = \log r$ by

$$\begin{aligned} s = \log(f_{Y \mid \Theta}(y \mid \theta')) - \log(f_{Y \mid \Theta}(y \mid \theta)) \\ + \log(f_\Theta(\theta')) - \log(f_\Theta(\theta)) + \log(q(\theta \mid \theta')) - \log(q(\theta' \mid \theta)) \end{aligned}$$

and only then calculate $r = \exp(s)$. Additionally, one might want cancel common factors from r before calculating its logarithm.

Implementing some Metropolis–Hastings algorithm for any given Bayesian statistical model is usually straightforward. However, finding a proposal distribution which allows the chain to converge quickly to the target distribution and allows it to explore the parameter space efficiently may be challenging.

7.4 Concrete Metropolis–Hastings algorithms

In the Metropolis–Hastings algorithm, the proposal θ' is in practice produced by running a piece of code, which can use the current state $\theta^{(i)}$, freshly generated random numbers from any distribution and arbitrary arithmetic operations. We must be able to calculate the density of the proposal θ', when the current state is equal to θ. This is then $q(\theta' \mid \theta)$, which we must be able to evaluate. Or at least we must be able to calculate the value of the ratio

$$q(\theta \mid \theta')/q(\theta' \mid \theta).$$

Different choices for the proposal density correspond to different choices for the needed piece of code. The resulting Metropolis–Hastings algorithms are named after the properties of the proposal distribution.

7.4.1 The independent Metropolis–Hastings algorithm

In the independent M–H algorithm (other common names: independence chain independence sampler), the proposal density is a fixed density, say $s(\theta')$, which does not depend on the value of the current state. In the corresponding piece of code, we only need to generate the value θ' from the proposal distribution.

If the proposal distribution happens to be the target distribution, then every proposal will be accepted, and as a result we will get an i.i.d. sample from the target distribution.

In order to to sample the target distribution properly with the independent M–H algorithm, the proposal density s must be positive everywhere, where the target density is positive. If there exist a majorizing constant M, such that

$$\pi(\theta) \le M s(\theta) \qquad \forall \theta,$$

then the resulting chain can be shown to have good ergodic properties, but if this condition fails, then the convergence properties of the chain can be bad. (In the independent M–H algorithm one does not need to know the value of M.) This implies that the proposal density should be such that the accept–reject method or importance sampling using that proposal distribution would be possible, too. In particular, the tails of the proposal density s should be at least as heavy as the tails of the target density. Finding such proposal densities may be difficult in high-dimensional problems.

7.4.2 Symmetric proposal distribution

If the proposal density is symmetric in that

$$q(\theta' \mid \theta) = q(\theta \mid \theta'), \qquad \forall \theta, \theta',$$

then the proposal density cancels from the M–H ratio,

$$r = \frac{\pi(\theta')\, q(\theta \mid \theta')}{\pi(\theta)\, q(\theta' \mid \theta)} = \frac{\pi(\theta')}{\pi(\theta)}.$$

This the sampling method that was originally proposed by Metropolis. Proposals leading to a higher value for the target density are automatically accepted, and other proposals may be accepted or rejected. Later Hastings generalized the method for non-symmetric proposal densities.

7.4.3 Random walk Metropolis–Hastings

Suppose that g is a density on the parameter space an that we calculate the proposal as follows,

$$\text{generate } w \text{ from density } g \text{ and set } \theta' \leftarrow \theta + w.$$

Then the proposal density is

$$q(\theta' \mid \theta) = g(\theta' - \theta).$$

This kind of a proposal is called a random walk proposal. If the density g is symmetric, i.e.,

$$g(-w) = g(w) \quad \forall w,$$

then the proposal density $q(\theta' \mid \theta)$ is also symmetric, and thus cancels from the M–H ratio. In the case of a symmetric random walk proposal, one often speaks of the random walk Metropolis (RWM) algorithm.

Actually, a random walk is a stochastic process of the form $X_{t+1} = X_t + w_t$, where the random variables w_t are i.i.d. Notice that the stochastic process produced by the random walk M–H algorithm is **not** a random walk, since the proposals can either be accepted or rejected.

The symmetric random walk Metropolis–Hastings algorithm (also known as the random walk Metropolis algorithm) is one of the most commonly used forms of the Metropolis–Hastings method. The most commonly used forms for g are the multivariate normal or multivariate Student's t density centered at the origin. This is, of course, appropriate only for continuous posterior distributions.

Often one selects the covariance matrix of the proposal distribution as

$$a\, C,$$

where C is an approximation to the covariance matrix of the target distribution (in Bayesian inference C is an approximation to the posterior covariance matrix) and the scalar a is a tuning constant which should be calibrated carefully. These kind of proposal distributions work well when the posterior distribution is approximately normal. One sometimes needs to reparametrize the model in order to make this approach work better.

The optimal value of a and the corresponding optimal acceptance rate has been derived theoretically, when the target density is a multivariate normal $N_d(\mu, C)$ and the random walk proposal is $N_d(0, aC)$, see [13]. The scaling constant a should be about $(2.38)^2/d$ when d is large. The corresponding acceptance rate (the number of accepted proposals divided by the total number of proposals) is from around 0.2 (for high-dimensional problems) to around 0.4 (in dimensions one or two). While these results have been derived using the very restrictive assumption that the target density is a multivariate normal, the results anyhow give rough guidelines for calibrating a in a practical problem.

How and why should one try to control the acceptance rate in the random walk M–H algorithm? If the acceptance rate is too low, then the chain is not able to move, and the proposed updating steps are likely to be too large. In this case one could try a smaller value for a. However, a high acceptance rate may also signal a problem, since then the updating steps may be too small. This may lead to the situation where the chain explores only a small portion of the parameter space. In this case one should try a larger value for a. From the convergence point of view, too high acceptance rate is a bigger problem. A low acceptance rate is a problem only from the computing time point of view.

In order to calibrate the random walk M–H algorithm, one needs an estimate of its acceptance rate. A simple-minded approach is just to keep track of the number of accepted proposals. A better approach is to calculate the average of the acceptance probabilities,

$$\frac{1}{N} \sum_{i=1}^{n} \min(1, r_i),$$

where r_i is the M–H ratio in the ith iteration.

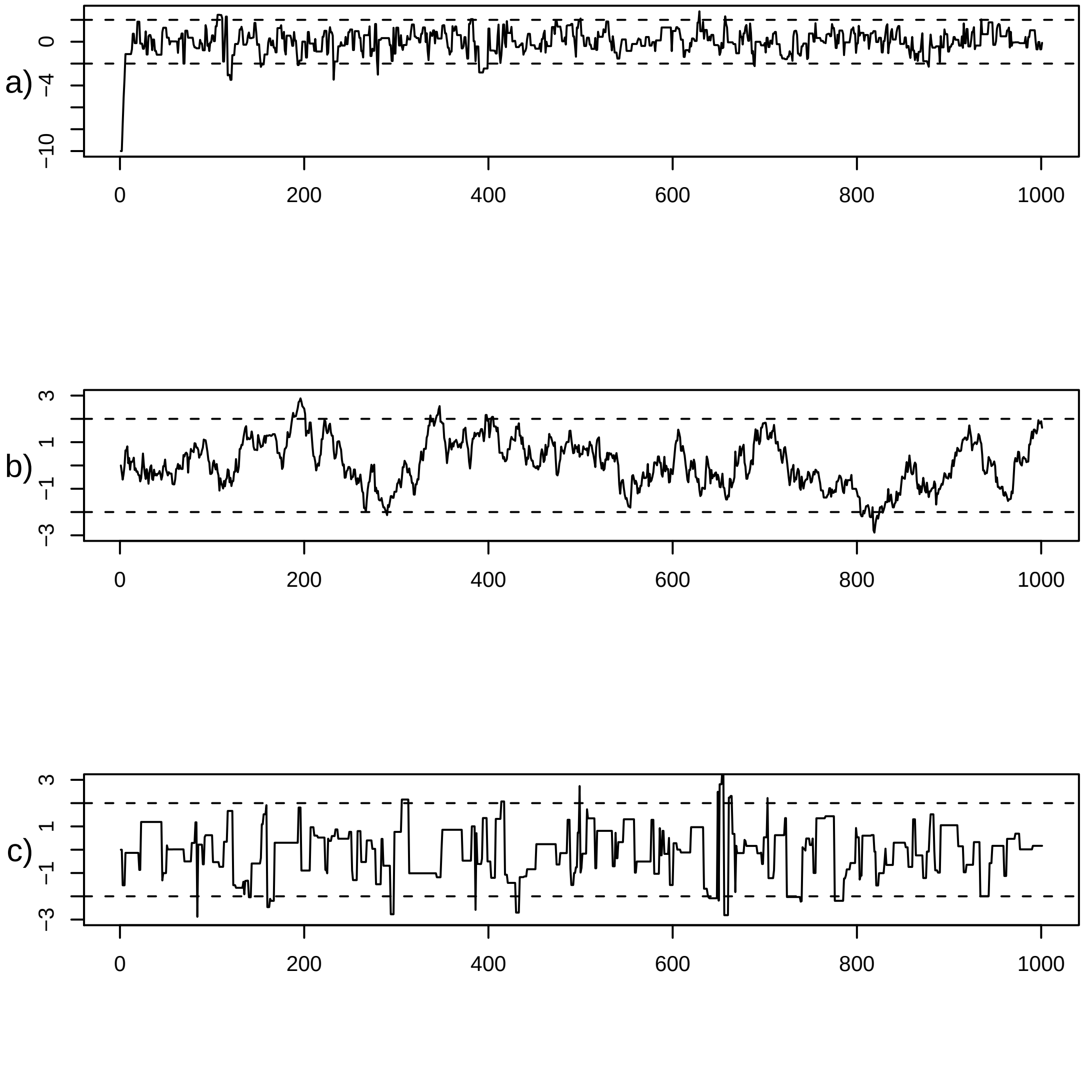
In practice, one can try to tune a iteratively, until the acceptance rate is acceptable. The tuning iterations are discarded, and the MCMC sample on which the inference is based is calculated using the fixed proposal distribution, whose scale a is the selected value. Fixing the proposal distribution is necessary, since the theory of the Metropolis–Hastings algorithm requires a homogeneous Markov chain, i.e., a proposal density $q(\theta' \mid \theta)$ which does not depend on the iteration index.

Recently, several researchers have developed adaptive MCMC algorithms, where the proposal distribution is allowed to change all the time during the iterations, see [1] for a review. Be warned that the design of valid adpative MCMC algorithms is subtle and that their analysis requires tools which are more difficult than the general state space Markov chain theory briefly touched upon in Chapter 11.

Example 7.1. Let us try the random walk chain for the target distribution $N(0, 1)$ by generating the increment from the normal distribution $N(0, \sigma^2)$ using the following values for the variance: a) $\sigma^2 = 4$ b) $\sigma^2 = 0.1$ c) $\sigma^2 = 40$. In situation a) the chain is initialized far away in the tails of the target distribution, but nevertheless it quickly finds its way to the main portion of the target distribution and then explores it efficiently. Such a chain is said to **mix** well. In situations b) and c) the chains are initialized at the center of the target distribution, but the chains mix less quickly. In situation b) the step length is too small, but almost all proposals get accepted. In situation c) the algorithm proposes too large steps, almost all of which get rejected. Figure 7.1 presents trace plots (or time series plots) of the chain in the three situations.

△

Figure 7.1: Trace plots of the random walk chain using the three different proposal distributions.



7.4.4 Langevin proposals

Unlike a random walk, the Langevin proposals introduce a drift which moves the chain towards the modes of the posterior distribution. When the current state is θ, the proposal θ' is generated with the rule

$$\theta' = \theta + \frac{\sigma^2}{2} \nabla(\log \pi(\theta)) + \sigma\, \epsilon, \qquad \epsilon \sim N_p(0, I).$$

Here

$$\nabla(\log \pi(\theta)) = \nabla(\log \tilde{\pi}(\theta))$$

is the gradient of the (unnormalized) posterior density. Here $\sigma > 0$ is a tuning parameter. The proposal distribution is motivated by stochastic differential equation, which has π as its stationary distribution.

This proposal is then accepted or rejected using the ordinary Metropolis–Hastings rule, where the proposal density is

$$q(\theta' \mid \theta) = N_p(\theta' \mid \theta + \frac{\sigma^2}{2} \nabla(\log \pi(\theta)),\ \sigma^2 I).$$

7.4.5 Reparametrization

Suppose that the posterior distribution of interest is a continuous distribution and that we have implemented functions for calculating the log-prior and the log-likelihood in terms of the parameter θ. Now we want to consider a diffeomorphic reparametrization

$$\phi = g(\theta) \quad \Leftrightarrow \quad \theta = h(\phi).$$

Typical reparametrizations one might consider are taking the logarithm of a positive parameter or calculating the logit function of a parameter constrained to the interval $(0, 1)$. What needs to be done in order to implement the Metropolis–Hastings algorithm for the new parameter vector ϕ?

First of all, we need a proposal density $q(\phi' \mid \phi)$ and the corresponding code. We also need to work out how to compute the Jacobian

$$J_h(\phi) = \frac{\partial \theta}{\partial \phi}$$

or the Jacobian

$$J_g(\theta) = \frac{\partial \phi}{\partial \theta}.$$

The M–H ratio, when we propose ϕ' and the current value is ϕ, is given by

$$\begin{aligned} r &= \frac{f_{\Phi|Y}(\phi' \mid y)\, q(\phi \mid \phi')}{f_{\Phi|Y}(\phi \mid y)\, q(\phi' \mid \phi)} \\ &= \frac{f_{\Theta|Y}(\theta' \mid y)\, |J_h(\phi')|\, q(\phi \mid \phi')}{f_{\Theta|Y}(\theta \mid y)\, |J_h(\phi)|\, q(\phi' \mid \phi)} \\ &= \frac{f_{Y|\Theta}(y \mid \theta')\, f_\Theta(\theta')\, q(\phi \mid \phi')}{f_{Y|\Theta}(y \mid \theta)\, f_\Theta(\theta)\, q(\phi' \mid \phi)}\ \frac{|J_h(\phi')|}{|J_h(\phi)|} \end{aligned}$$

here $\theta' = h(\phi')$ and $\theta = h(\phi)$. Here the Jacobian arises from transforming the proposal density from θ space to ϕ space. Sometimes it is more convenient to work with the Jacobian J_g, but this is easy, since

$$J_g(\theta) = \frac{1}{J_h(\phi)}.$$

Above we viewed the Jacobians as arising from expressing the proposal distribution in ϕ space instead of θ space. An alternative interpretation is that we have transformed the prior from θ parameterization to ϕ parametrization. Both viewpoints yield the same formulas.

In order to calculate the logarithm of the M–H ratio, we need to do the following.

- Calculate the θ and θ' values corresponding to the current ϕ and proposed ϕ' values.
- Calculate the log-likelihood and log-prior using the values θ and θ'.
- Calculate the logarithm s of the M–H ratio as

$$\begin{aligned} s = \log(f_{Y|\Theta}(y \mid \theta')) - \log(f_{Y|\Theta}(y \mid \theta)) \\ + \log(f_\Theta(\theta')) - \log(f_\Theta(\theta)) + \log(q(\phi \mid \phi')) - \log(q(\phi' \mid \phi)) \\ + \log(|J_h(\phi')|) - \log(|J_h(\phi)|). \end{aligned}$$

  Finally, calculate $r = \exp(s)$.

- The difference of the logarithms of the absolute Jacobians can be calculated either on the ϕ scale or on the θ scale by using the identity

$$\log(|J_h(\phi')|) - \log(|J_h(\phi)|) = \log(|J_g(\theta)|) - \log(|J_g(\theta')|).$$

7.4.6 State-dependent mixing of proposal distributions

Let θ be the current state of the chain. Suppose that the proposal θ' is drawn from a proposal density, which is selected randomly from a list of alternatives

$$q(\theta' \mid \theta, j), \qquad j = 1, \ldots K,$$

What is more, the selection probabilities may depend on the current state, as follows.

- Draw j from the pmf $\beta(\cdot \mid \theta), j = 1, \ldots, K$.
- Draw θ' from the density $q(\theta' \mid \theta, j)$ which corresponds to the selected j.
- Accept the proposed value θ' as the new state, if $U < r$, where $U \sim \text{Uni}(0,1)$, and

$$r = \frac{\pi(\theta')\,\beta(j \mid \theta')\,q(\theta \mid \theta', j)}{\pi(\theta)\,\beta(j \mid \theta)\,q(\theta' \mid \theta, j)}. \tag{7.6}$$

  Otherwise the chain stays at θ.

This formula (7.6) for the M–H ratio r is contained in Green's article [6], which introduced the reversible jump MCMC method. The algorithm could be called the Metropolis–Hastings–Green algorithm.

The lecturer does know any trick for deriving formula (7.6) from the M–H ratio of the ordinary M–H algorithm. The beauty of formula (7.6) lies in the fact that one only needs to evaluate $q(\theta' \mid \theta, j)$ and $q(\theta \mid \theta', j)$ for the proposal density which was selected. A straightforward application of the M–H algorithm would require one to evaluate these densities for all of the K possibilities.

If the selection probabilities $\beta(j \mid \theta)$ do not actually depend on θ, then they cancel from the M–H ratio. In this case (7.6) is easily derived from the ordinary M–H algorithm.

7.5 Gibbs sampler

One of the best known ways of setting up an MCMC algorithm is Gibbs sampling, which is now discussed supposing that the target distribution is a posterior distribution. However, the method can be applied to any target distribution, when the full conditional distributions of the target distribution are available.

Suppose that the parameter vector has been divided into components

$$\theta = (\theta_1, \theta_2, \dots, \theta_d),$$

where θ_j need not be a scalar. Suppose also that the posterior full conditional distributions of each of the components are available in the sense that we know how to simulate them. This is the case when the statistical model exhibits conditional conjugacy with respect to all of the components θ_j. Then the basic idea behind Gibbs sampling is that we simulate successively each component θ_j from its (posterior) full conditional distribution. It is convenient to use the abbreviation θ_{-j} for the vector, which contains all the other components of θ but θ_j, i.e.

$$\theta_{-j} = (\theta_1, \dots, \theta_{j-1}, \theta_{j+1}, \dots, \theta_d). \tag{7.7}$$

Then the posterior full conditional of θ_j is

$$p(\theta_j \mid \theta_{-j}, y) = f_{\Theta_j \mid \Theta_{-j}, Y}(\theta_j \mid \theta_{-j}, y). \tag{7.8}$$

A convenient shorthand notation for the posterior full conditional is

$$p(\theta_j \mid \cdot),$$

where the dot denotes all the other random variables except θ_j.

The most common form of the Gibbs sampler is the systematic scan Gibbs sampler, where the components are updated in a fixed cyclic order. It is also possible to select at random which component to update next. In that case one has the random scan Gibbs sampler.

Algorithm 16 presents the systematic scan Gibbs sampler, when we update the components using the order $1, 2, \dots, d$. In the algorithm i is the time index of the Markov chain. One needs d updates to get from $\theta^{(i)}$ to $\theta^{(i+1)}$. To generate the j'th component, $\theta_j^{(i+1)}$, one uses the most recent values for the other components, some of which have already been updated. I.e., when the value for

$\theta_j^{(i+1)}$ is generated, it is generated from the corresponding full conditional using the following values for the other components,

$$\theta_{-j}^{\mathrm{cur}} = (\theta_1^{(i+1)}, \ldots, \theta_{j-1}^{(i+1)}, \theta_{j+1}^{(i)}, \ldots, \theta_d^{(i)}).$$

Algorithm 16: Systematic scan Gibbs sampler.

```
Input: An initial value $\theta^{(0)}$ such that $f_{\Theta|Y}(\theta^{(0)} \mid y) > 0$ and the number
       of iterations $N$.
Result: Values simulated from a Markov chain which has the posterior
        distribution as its invariant distribution.
$\theta^{\mathrm{cur}} \leftarrow \theta^{(0)}$
for $i = 0, 1, \ldots, N$ do
    for $j = 1, \ldots, d$ do
        draw a new value for the $j$th component $\theta_j^{\mathrm{cur}}$ of $\theta^{\mathrm{cur}}$ from the
        posterior full conditional $f_{\Theta_j|\Theta_{-j},Y}(\theta_j \mid \theta_{-j}^{\mathrm{cur}}, y)$
    end
    $\theta^{(i+1)} \leftarrow \theta^{\mathrm{cur}}$
end
```

Usually the updating steps for the components of θ are so heterogeneous, that the inner loop is written out in full. E.g., in the case of three components, $\theta = (\phi, \psi, \tau)$, the actual implementation would probably look like the following algorithm 17. This algorithm also demonstrates, how one can write the algorithm using the abbreviated notation for conditional densities.

Algorithm 17: Systematic scan Gibbs sampler for three components $\theta = (\phi, \psi, \tau)$ given initial values for all the components except the one that gets updated the first.

```
$\psi^{\mathrm{cur}} \leftarrow \psi_0$;   $\tau^{\mathrm{cur}} \leftarrow \tau_0$;
for $i = 0, 1, \ldots, N$ do
    draw $\phi^{\mathrm{cur}}$ from $p(\phi \mid \psi = \psi^{\mathrm{cur}}, \tau = \tau^{\mathrm{cur}}, y)$;
    draw $\psi^{\mathrm{cur}}$ from $p(\psi \mid \phi = \phi^{\mathrm{cur}}, \tau = \tau^{\mathrm{cur}}, y)$;
    draw $\tau^{\mathrm{cur}}$ from $p(\tau \mid \phi = \phi^{\mathrm{cur}}, \psi = \psi^{\mathrm{cur}}, y)$;
    $\phi_{i+1} \leftarrow \phi^{\mathrm{cur}}$;   $\psi_{i+1} \leftarrow \psi^{\mathrm{cur}}$;   $\tau_{i+1} \leftarrow \tau^{\mathrm{cur}}$;
end
```

Algorithm 18 presents the random scan Gibbs sampler. Now one time step of the Markov chain requires only one update of a randomly selected component. In the random scan version, one can have different probabilities for updating the different components of θ, and this freedom can be useful for some statistical models.

If the statistical model exhibits conditional conjugacy with respect to all the components of θ, then the Gibbs sampler is easy to implement and is the method of choice for many statisticians. One only needs random number generators for all the posterior full conditionals, and these are easily available for the standard distributions. An appealing feature of the method is the fact that one does not need to choose the proposal distribution as in the Metropolis–Hastings sampler;

Algorithm 18: Random scan Gibbs sampler.

```
Input: An initial value $\theta^{(0)}$ such that $f_{\Theta|Y}(\theta^{(0)} \mid y) > 0$, the number of
       iterations $N$ and a probability vector $\beta_1, \ldots, \beta_d$: each $\beta_j > 0$ and
       $\beta_1 + \cdots + \beta_d = 1$.
Result: Values simulated from a Markov chain which has the posterior
        distribution as its invariant distribution.
$\theta^{\mathrm{cur}} \leftarrow \theta^{(0)}$;
for $i = 0, 1, \ldots, N$ do
    select $j$ from $\{1, \ldots, d\}$ with probabilities $(\beta_1, \ldots, \beta_d)$;
    draw a new value for the component $\theta_j^{\mathrm{cur}}$ from the posterior full
    conditional $f_{\Theta_j|\Theta_{-j},Y}(\theta_j \mid \theta_{-j}^{\mathrm{cur}}, y)$;
    $\theta^{(i+1)} \leftarrow \theta^{\mathrm{cur}}$;
end
```

the proposals of the Gibbs sampler are somehow automatically tuned to the target posterior. However, if some of the components of θ are strongly correlated in the posterior, then the convergence of the Gibbs sampler suffers. So one might want to reparametrize the model so that the transformed parameters are independent in their posterior. Unfortunately, most reparametrizations destroy the conditional conjugacy properties on which the attractiveness of the Gibbs sampler depends.

The name Gibbs sampling is actually not quite appropriate. Gibbs studied distributions arising in statistical physics (often called Gibbs distributions or Boltzmann distributions), which have densities of the form

$$f(x_1, \ldots, x_d) \propto \exp\left(-\frac{1}{kT} E(x_1, \ldots, x_d)\right),$$

where $(x_1, \ldots, x_d)$ is the state of physical system, k is a constant, T is the temperature of the system, and $E(x_1, \ldots, x_d) > 0$ is the energy of the system. The Geman brothers used a computational method (simulated annealing), where a computational parameter corresponding to the the temperature of a Gibbs distribution was gradually lowered towards zero. At each temperature the distribution of the system was simulated using the Gibbs sampler. This way they could obtain the configurations of minimal energy in the limit. The name Gibbs sampling was selected in order to emphasize the relationship with the Gibbs distributions. However, when the Gibbs sampler is applied to posterior inference, the temperature parameter is not needed, and therefore the reason for the name Gibbs has disappeared. Many authors have pointed this out this deficiency and proposed alternative names for the sampling method, but none of them have stuck.

7.6 Componentwise updates in the Metropolis–Hastings algorithm

Already Metropolis *et al.* and Hastings pointed out that one can use componentwise updates in the Metropolis–Hastings algorithm. This sometimes called single-site updating.

When the parameter vector is divided into d components

$$\theta = (\theta_1, \theta_2, \dots, \theta_d),$$

one needs d proposal densities

$$\theta'_j \mapsto q_j(\theta'_j \mid \theta^{\text{cur}}), \qquad j = 1, \dots, d,$$

which may all be different.

When it is time to update the jth component, we do a single Metropolis–Hastings step. When the current value of the parameter vector is θ^{cur}, we propose the vector θ', where the jth component is drawn from the proposal density $q_j(\theta_j \mid \theta^{\text{cur}})$, and the rest of the components of θ' are equal to those of the current value θ^{cur}. Then the proposal is accepted or rejected using the M–H ratio

$$r = \frac{p(\theta' \mid y)\, q_j(\theta^{\text{cur}}_j \mid \theta')}{p(\theta^{\text{cur}} \mid y)\, q_j(\theta'_j \mid \theta^{\text{cur}})} \tag{7.9}$$

The vectors θ' and θ^{cur} differ only in the jth place, and therefore one can write the M–H ratio (for updating the jth component) also in the form

$$r = \frac{p(\theta'_j \mid \theta'_{-j}, y)\, q_j(\theta^{\text{cur}}_j \mid \theta')}{p(\theta^{\text{cur}}_j \mid \theta^{\text{cur}}_{-j}, y)\, q_j(\theta'_j \mid \theta^{\text{cur}})} \tag{7.10}$$

Although eqs. (7.9) and (7.10) are equivalent, notice that in eq. (7.9) we have the M–H ratio when we regard the joint posterior as the target distribution, but eq. (7.10) seems to be the M–H ratio, when the target is the posterior full conditional of component j. If one then selects as q_j the posterior full conditional of the component θ_j for each j, then the Gibbs sampler ensues.

One can use this procedure either a systematic or a random scan sampler, as is the case with the Gibbs sampler. The resulting algorithm is often called the Metropolis–within–Gibbs sampler. (The name is illogical: the Gibbs sampler is a special case of the Metropolis–Hastings algorithm with componentwise updates.) This is also a very popular MCMC algorithm, since then one does not have to design a single complicated multivariate proposal density but p simpler proposal densities, many of which may be full conditional densities of the posterior.

Small modifications in the implementation can sometimes make a big difference to the efficiency of the sampler. One important decision is how to divide the parameter vector into components. This is called **blocking** or **grouping**. As a general rule, the less dependent the components are in the posterior, the better the sampler. Therefore it may be a good idea to combine highly correlated components into a single block, with is then updated as a single entity.

It is sometimes useful to update the whole vector jointly using a single Metropolis–Hastings acceptance test, even if the proposed value is build up component by component taking advantage of conditional conjugacy properties. These and other ways of improving the performance of MCMC algorithms in the context of specific statistical models are topics of current research.

7.7 Analyzing MCMC output

After the MCMC algorithm has been programmed and tested, the user should investigate the properties of the algorithm for the particular problem he or she is trying to solve. There are available several tools, e.g., for

- diagnosing convergence
- estimating Monte Carlo standard errors.

We discuss some of the simpler tools.

A **trace plot** of a parameter ϕ is a plot of the iterates $\phi^{(t)}$ against the iteration number t. These are often examined for each of the components of the parameter vector, and sometimes also for selected scalar functions of the parameter vector. A trace plot is also called a *sample path*, a *history plot* or a *times series plot*. If the chain mixes well, then the trace plots move quickly away from their starting values and they wiggle vigorously in the region supported by the posterior. In that case one may select the length of the burn-in by examining trace plots. (This is not foolproof, since the chain may only have converged momentarily to some neighborhood of a local maximum of the posterior.) If the chain mixes poorly, then the traces will remain nearly constant for many iterations and the state may seem to wander systematically towards some direction. Then one may need a huge number of iterations before the traces show convergence.

An **autocorrelation plot** is a plot of the autocorrelation of the sequence $\phi^{(t)}$ at different iteration lags. These can be produced for all the interesting components of θ, but one should reject the burn-in before estimating the autocorrelation so that one analyzes only that part of the history where the chain is approximately stationary. The autocorrelation function (acf) of a stationary sequence of RVs (X_i) at lag k is defined by

$$R(k) = \frac{E[(X_i - \mu)(X_{i+k} - \mu)]}{\sigma^2}, k = 0, 1, 2, \dots,$$

where $\mu = EX_i$, $\sigma^2 = \operatorname{var} X_i$, and the assumption of stationarity entails that μ, σ^2 and $R(k)$ do not depend on index i. For an i.i.d. sequence the autocorrelation function is one at lag zero and zero otherwise. A chain that mixes slowly exhibits slow decay of the autocorrelation as the lag increases. When there are more than one parameter, one may also examine cross-correlations between the parameters.

There exist tools for **convergence diagnostics**, which try to help in deciding whether the chain has already approximately reached its stationary distribution and in selecting the length of the burn-in period. E.g., in the approach of Gelman and Rubin, the chain is run many times starting from separate starting values dispersed over the support of the posterior. After the burn-in has been discarded, one calculates statistics which try to check whether all the chains have converged to the same distribution. In some other approches one needs to simulate only a single chain and one compares the behaviour of the chain in the beginning and in the end of the simulation run. Such convergence diagnostic are available in the `coda` R package and in the `boa` R package. However, convergence diagnostic tools can not prove that the chain has converged. They only help you to detect obvious cases of non-convergence.

If the chain seems to have converged, then it is of interest to estimate standard errors for the scalar parameters. The naive estimate (which is correct for i.i.d. sampling) would be to calculate the sample standard deviation of the last L iterations divided by $\sqrt{L}$ (after the burn-in has been discarded). However, MCMC iterates are typically positively correlated, and therefore this would underestimate severely the standard error.

A simple method for estimating the standard errors for posterior expectations

$$E[h(\Theta) \mid Y = y]$$

is the method of **batch means** [8], where the L last iterates are divided into a non-overlapping batches of length b. Then one computes the mean $\bar{h}_j$ of the values $h(\theta^{(t)})$ inside each of the batches $j = 1, \ldots, a$ and estimates the standard error of the grand mean $\bar{h}$ as the square roof of

$$\frac{1}{a} \frac{1}{a-1} \sum_{j=1}^{a} (\bar{h}_j - \bar{h})^2,$$

where $\bar{h}$ is the grand mean calculated from all the the L last iterates $h(\theta^{(t)})$. The idea here is to treat the batch means as i.i.d. random variables whose expected value is the posterior expectation. One should perhaps select the batch length as a function of the simulation length, e.g., with the rule $b = \lfloor \sqrt{L} \rfloor$.

7.8 Example

Consider the two dimensional normal distribution $N(0, \Sigma)$ as the target distribution, where

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}, \qquad -1 < \rho < 1, \quad \sigma_1, \sigma_2 > 0,$$

and ρ is nearly one. Of course, it is possible to sample this two-variate normal distribution directly. However, we next apply MCMC algorithms to this highly correlated toy problem in order to demonstrate properties of the Gibbs sampler and a certain Metropolis–Hastings sampler.

The full conditionals of the target distribution are given by

$$\begin{aligned} [\Theta_1 \mid \Theta_2 = \theta_2] \quad &\sim N\left(\tfrac{\rho\sigma_1}{\sigma_2}\theta_2, (1-\rho^2)\sigma_1^2\right) \\ [\Theta_2 \mid \Theta_1 = \theta_1] \quad &\sim N\left(\tfrac{\rho\sigma_2}{\sigma_1}\theta_1, (1-\rho^2)\sigma_2^2\right), \end{aligned}$$

and these are easy to simulate. We now suppose that

$$\rho = 0.99, \quad \sigma_1 = \sigma_2 = 1.$$

Figure 7.2 shows the ten first steps of the Gibbs sampler, when all the component updates ("half-steps" of the sampler) are shown. Since ρ is almost one, the Gibbs sampler is forced to take small steps, and it takes a long time for it to explore the main support of the target distribution.

Another strategy would be to generate the proposal in two stages as follows. We first draw θ_1' from some convenient proposal distribution, e.g., by the random walk proposal

$$\theta_1' = \theta_1^{\text{cur}} + w,$$

Figure 7.2: The first ten iterations of the Gibbs sampler. The three contour lines enclose 50 %, 90 % and 99 % of the probability mass of the target distribution.

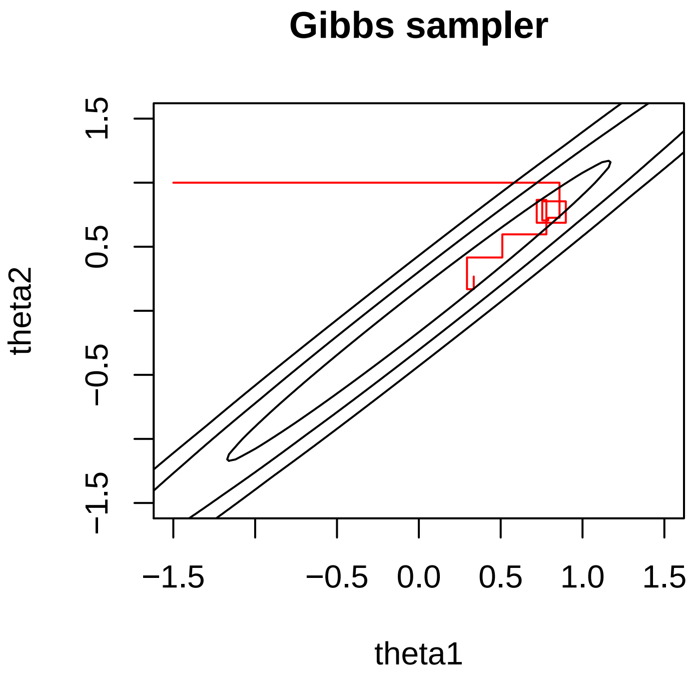
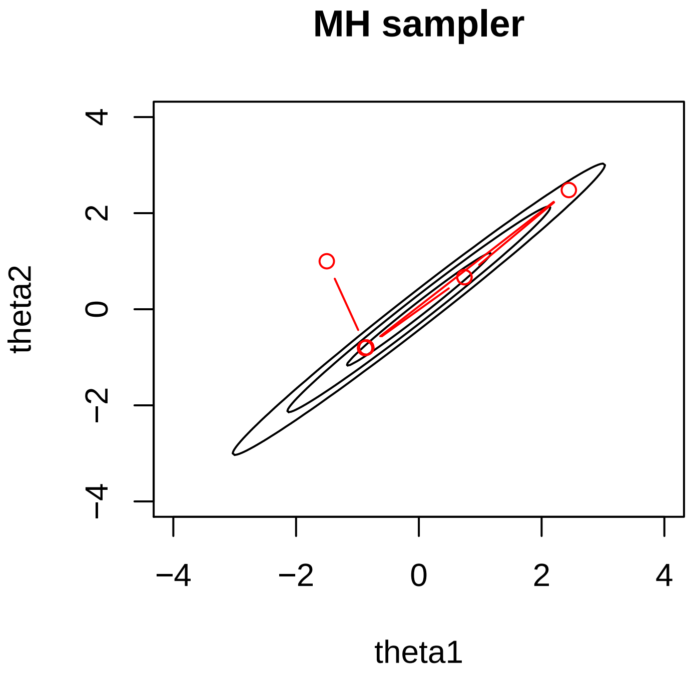


Figure 7.3: The first ten iterations of the Metropolis–Hastings sampler. Notice the sampler produced less than ten distinct θ values. The three contour lines enclose 50 %, 90 % and 99 % of the probability mass of the target distribution.



where w is generated from (say) $N(0, 4)$. Then we draw θ'_2 from the full conditional distribution of θ_2 conditioning on the proposed value θ'_1. Then the overall proposal density is given by

$$q((\theta'_1, \theta'_2) \mid (\theta_1^{\mathrm{cur}}, \theta_2^{\mathrm{cur}})) = N(\theta'_1 - \theta_1^{\mathrm{cur}} \mid 0, 4)\, N(\theta'_2 \mid \frac{\rho\sigma_2}{\sigma_1}\theta'_1, (1-\rho^2)\sigma_2^2)$$

We then either accept or reject the transition from θ^{cur} to θ' using the ordinary acceptance rule of the Metropolis–Hastings sampler. This algorithm explores the target distribution much more efficiently, as can be guessed from Figure 7.3, which shows the first ten iterations of the sampler. The random walk proposal gives the component θ_1 freedom to explore the parameter space, and then the proposal from the full conditional for θ_2 draws the proposed pair into the main support of the target density.

Figure 7.4 shows the traces of the components using the two algorithms. The Metropolis–Hastings sampler seems to mix better than the Gibbs sampler, since there seems to be less dependence between the consecutive simulated values. Figure 7.5 shows the autocorrelation plots for the two components using the two different samplers. The autocorrelation functions produced by the Gibbs sampler decay more slowly than those produced by the Metropolis–Hastings sampler, and this demonstrates that we obtain better mixing with the Metropolis–Hastings sampler.

7.9 Literature

The original references on the Metropolis sampler, the Metropolis–Hastings sampler and the Gibbs sampler are [9, 7, 4]. The article by Gelfand and Smith [3] finally convinced the statistical community about the usefulness of these methods in Bayesian inference. The book [5] contains lots of information on MCMC methods and their applications.

The books by Nummelin [11] or Meyn and Tweedie [10] can be consulted for the theory of Markov chains in a general state space. The main features of the general state space theory are explained in several sources, including [2, Ch. 14] or [12, Ch. 6].

Bibliography

[1] Christophe Andrieu and Johannes Thoms. A tutorial on adaptive MCMC. *Statistics and Computing*, 18:343–373, 2008.

[2] Krishna B. Athreya and Soumendra N. Lahiri. *Measure Theory and Probability Theory.* Springer Texts in Statistics. Springer, 2006.

[3] A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.

[4] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

Figure 7.4: Sampler traces for the two components θ_1 and θ_2 using the Gibbs sampler and the Metropolis–Hastings sampler.

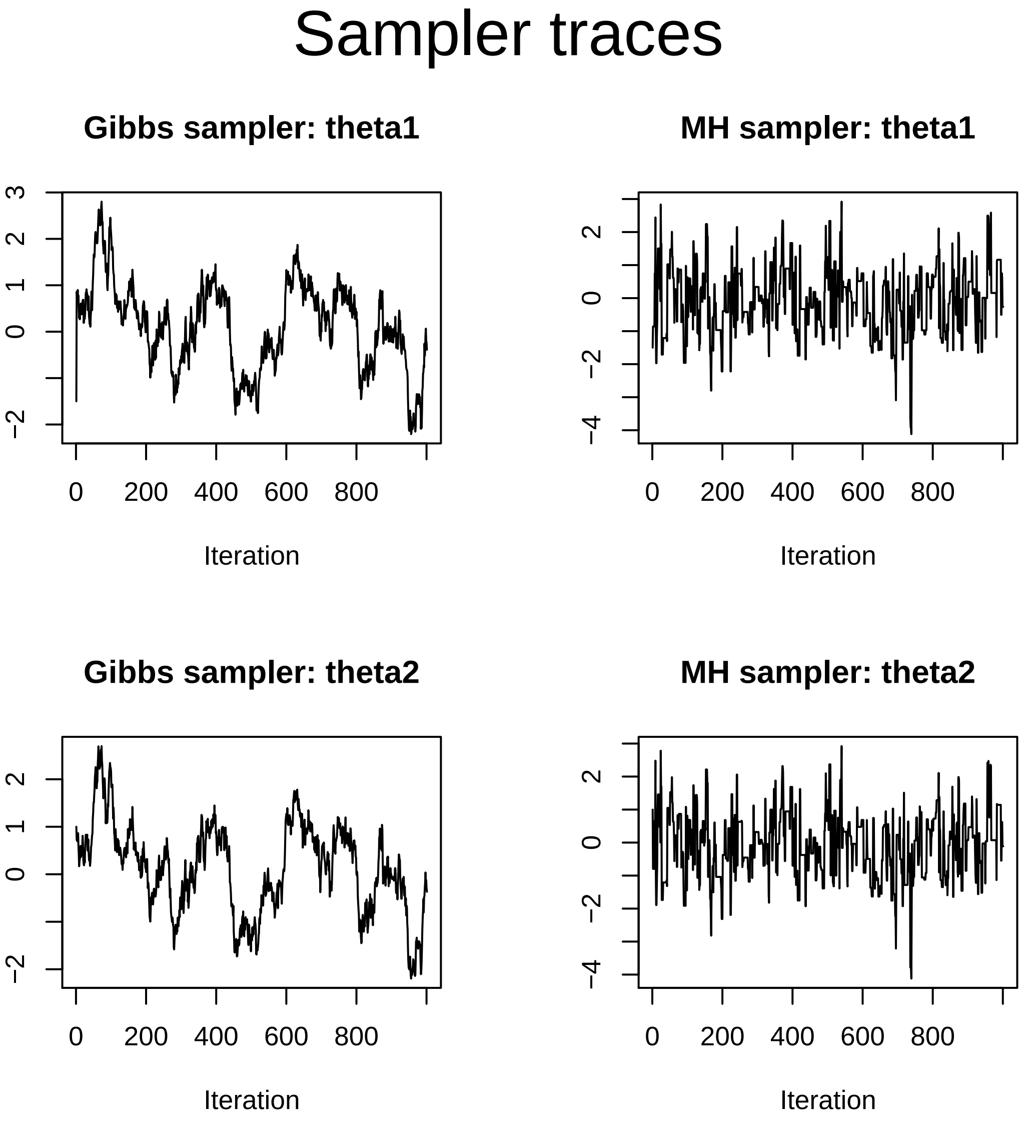
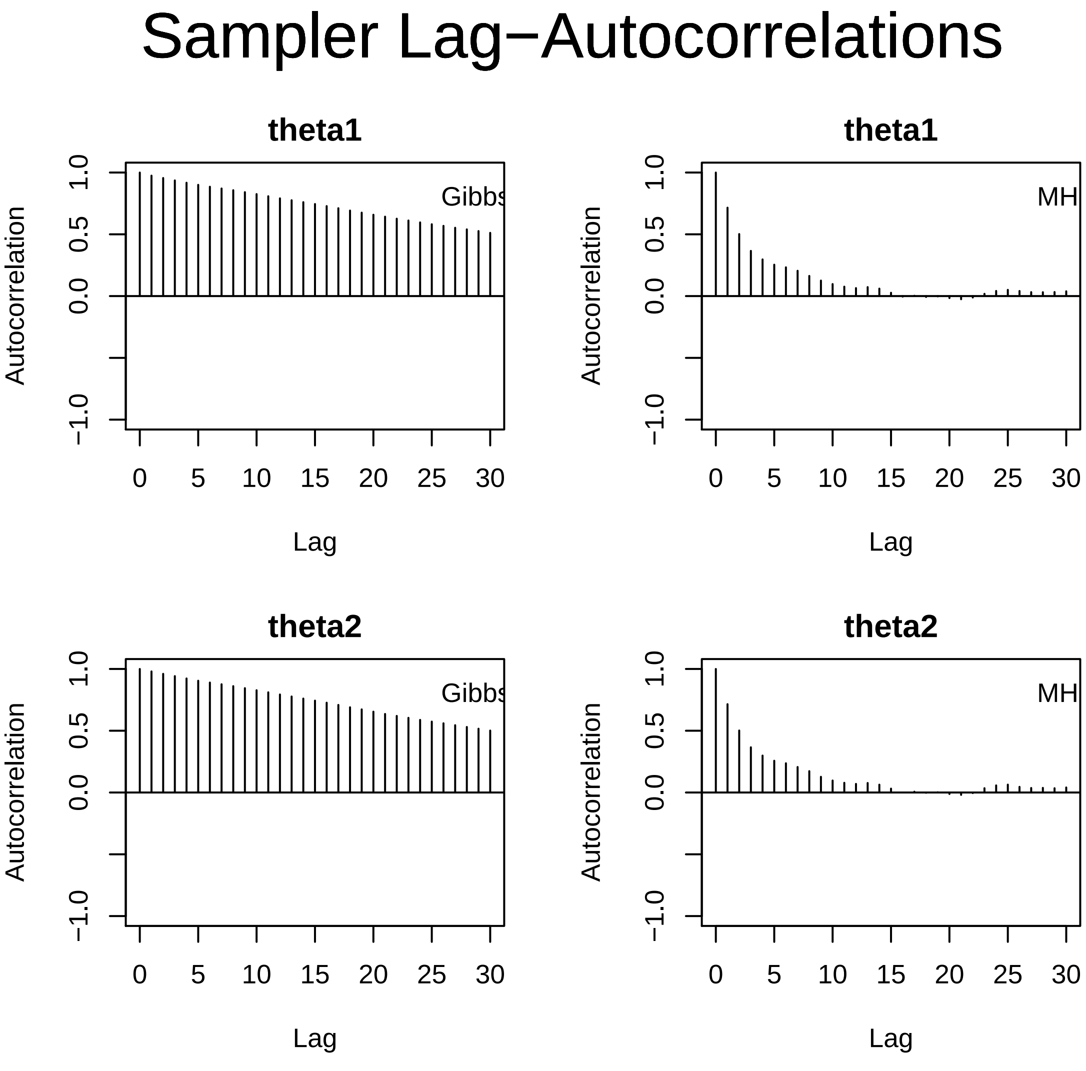


Figure 7.5: Sampler autocorrelation functions for the two components θ_1 and θ_2 using the Gibbs sampler and the Metropolis–Hastings sampler.



[5] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.

[6] Peter J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

[7] W. Hastings. Monte Carlo sampling methods using Markov chains and their application. *Biometrika*, 57:97–109, 1970.

[8] Averll M. Law and W. David Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, Inc., 2nd edition, 1991.

[9] N. Metropolis, A. Rosenbluth, , M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[10] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer, 1993.

[11] Esa Nummelin. *General Irreducible Markov Chains and Nonnegative Operators*. Cambridge University Press, 1984.

[12] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, second edition, 2004.

[13] Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling for various Metropolis–Hastings algorithms. *Statistical Science*, 16(4):351–367, 2001.

Chapter 8

Auxiliary Variable Models

8.1 Introduction

We are interested in an actual statistical model, with joint distribution

$$\mathrm{p}_{\mathrm{act}}(y, \theta) = \mathrm{p}_{\mathrm{act}}(y \mid \theta)\ \mathrm{p}_{\mathrm{act}}(\theta),$$

but where the posterior $\mathrm{p}_{\mathrm{act}}(\theta \mid y)$ is awkward to sample from. Suppose we are able to reformulate the original model by introducing a new random variable Z such that the marginal distribution of (y, θ) in the new model is the same as the joint distribution of (y, θ) in the original model, i.e., we assume that

$$\int \mathrm{p}_{\mathrm{aug}}(y, \theta, z)\, \mathrm{d}z = \mathrm{p}_{\mathrm{act}}(y, \theta). \tag{8.1}$$

When this is the case, we can forget the distinction between the actual model $\mathrm{p}_{\mathrm{act}}(\cdot)$ and the augmented model $\mathrm{p}_{\mathrm{aug}}(\cdot)$ and use the generic symbol $p(\cdot)$ to denote the densities calculated under either of the models. Here the *augmentation parameter*, the *auxiliary variable*, the *latent variable* or the *latent data* Z can be anything. However, it requires ingenuity and insight to come up with useful auxiliary variables.

Sometimes it is possible to sample much more efficiently from $p(\theta, z \mid y)$ than from $p(\theta \mid y)$. In such a case we can sample from the posterior $p(\theta, z \mid y)$, and we get a sample from the marginal posterior of θ by ignoring the z components of the (θ, z) sample. If both the full conditionals $p(\theta \mid z, y)$ and $p(z \mid \theta, y)$ are available in the sense that we know how to sample from these distribution, then implementing the Gibbs sampler is straightforward.

8.2 Slice sampler

Suppose we want to simulate from a distribution having the unnormalized density $q(\theta)$. By the fundamental theorem of simulation, this is equivalent to simulating (θ, z) from the uniform distribution under the graph of q, i.e., from $\mathrm{Uni}(A)$, the uniform distribution on the set

$$A = \{(\theta, z) : 0 < z < q(\theta)\}.$$

This distribution has the unnormalized density

$$p(\theta, z) \propto 1_A(\theta, z) = 1_{(0,q(\theta))}(z) = 1(0 < z < q(\theta))$$

The full conditional of Z is proportional to the joint density, considered as a function of z, i.e.,

$$p(z \mid \theta) \propto p(\theta, z) \propto 1(0 < z < q(\theta)),$$

and this an unnormalized density of the uniform distribution on the interval $(0, q(\theta))$.

Similarly, the full conditional of θ is the uniform distribution on the set (depending on z), where

$$1(0 < z < q(\theta)) = 1,$$

since the joint density is constant on this set. That is, the full conditional of θ is the uniform distribution on the set

$$B(z) = \{\theta : q(\theta) > z\}.$$

The resulting Gibbs sampler is called the slice sampler (for the distribution determined by q). The slice sampler is attractive, if the uniform distribution on the set $B(z)$ is easy to simulate.

Example 8.1. Let us consider the truncated standard normal distribution corresponding to the unnormalized density

$$q(\theta) = \exp\left(-\frac{1}{2}\theta^2\right) 1_{(\alpha,\infty)}(\theta),$$

where the truncation point $\alpha > 0$.

We can get a correlated sample $\theta_1, \theta_2, \ldots$ from this distribution as follows.

1. Pick an initial value $\theta_0 > \alpha$.
2. For $i = 1, 2 \ldots$
   - Draw z_i from $\mathrm{Uni}(0, q(\theta_{i-1}))$.
   - Draw θ_i from $\mathrm{Uni}(\alpha, \sqrt{-2 \ln z_i})$.

△

Simulating the uniform in the set $B(z)$ may turn out to be unwieldy. Usually, the target density can be decomposed into a product of functions,

$$p(\theta \mid y) \propto \prod_{i=1}^{n} q_i(\theta).$$

Then one may try the associated augmentation, where one introduces n auxiliary variables Z_i such that, conditionally on θ, the Z_i have independently the uniform distribution on $(0, q_i(\theta))$. In the augmented model, the full conditional of θ is the uniform distribution on the set

$$C(z) = \cap_{i=1}^{n} \{\theta : q_i(\theta) > z_i\},$$

and this may be easier to simulate. Typically, the more auxiliary variables one introduces, the slower is the mixing of the resulting chain.

8.3 Missing data problems

In many experiments the posterior distribution is easy to summarize if all the planned data are available. However, if some of the observations are missing, then the posterior is more complex. Let Z be the missing data and let y be the observed data. The full conditional

$$p(\theta \mid z, y)$$

is the posterior from the complete data, and it is of a simple form (by assumption). Often also the full conditional of the missing data

$$p(z \mid \theta, y)$$

is easy to sample from. Then it is straightforward to use the Gibbs sampler.

Here the joint distribution in the reformulated model is

$$\mathrm{p}_{\mathrm{aug}}(y, \theta, z) = \mathrm{p}_{\mathrm{act}}(\theta)\ \mathrm{p}_{\mathrm{aug}}(y, z \mid \theta).$$

In order to check the equivalence of the original and of the reformulated model, see (8.1), it is sufficient to check that

$$\int \mathrm{p}_{\mathrm{aug}}(y, z \mid \theta)\,\mathrm{d}z = \mathrm{p}_{\mathrm{act}}(y \mid \theta).$$

Example 8.2. Let us consider the famous genetic linkage example, where we have the multinomial likelihood

$$p(y \mid \theta) = \mathrm{Mult}\left((y_1, y_2, y_3, y_4) \mid n, \left(\frac{1}{2} + \frac{\theta}{4}, \frac{1}{4}(1-\theta), \frac{1}{4}(1-\theta), \frac{\theta}{4}\right)\right).$$

Here $0 < \theta < 1$, and $y = (y_1, y_2, y_3, y_4)$, where the y_j:s are the observed frequencies of the four categories. We take the uniform prior $\mathrm{Uni}(0, 1)$ for θ. The posterior is not of a standard form.

However, suppose that the first category with frequency y_1 is an amalgamation of two subclasses with probabilities $\theta/4$ and $1/2$, but the distinction between the subclasses has not been observed. Let Z be the frequency of the first subclass (with class probability $\theta/4$). Then the frequency of the second subclass (with class probability $1/2$) is $y_1 - Z$. Our reformulated model states that

$$p(z, y \mid \theta) = p(z, y_1, y_2, y_3, y_4 \mid \theta) = \\ \mathrm{Mult}\left((z, y_1 - z, y_2, y_3, y_4) \mid n, \left(\frac{1}{4}\theta, \frac{1}{2}, \frac{1}{4}(1-\theta), \frac{1}{4}(1-\theta), \frac{1}{4}\theta\right)\right)$$

Let us check that the reformulated model and the original model are equivalent. If we combine the frequencies X_{11} and X_{12} in the the multinomial distribution

$$(X_{11}, X_{12}, X_2, X_3, X_4) \sim \mathrm{Mult}(n, (p_{11}, p_{12}, p_2, p_3, p_4)),$$

then we obtain the multinomial distribution

$$(X_{11} + X_{12}, X_2, X_3, X_4) \sim \mathrm{Mult}(n, (p_{11} + p_{12}, p_2, p_3, p_4)),$$

and this is obvious when one thinks of the repeated sampling definition of the multinomial distribution. This shows that our original model and the reformulated model are equivalent.

The posterior of θ given the complete data consisting of y and z is given by

$$\begin{aligned} p(\theta \mid y, z) &\propto p(y, z \mid \theta)\, p(\theta) \\ &\propto \left(\frac{1}{4}\theta\right)^{z} \left(\frac{1}{2}\right)^{y_1 - z} \left(\frac{1}{4}(1-\theta)\right)^{y_2} \left(\frac{1}{4}(1-\theta)\right)^{y_3} \left(\frac{1}{4}\theta\right)^{y_4} \\ &\propto \theta^{z+y_4}\, (1-\theta)^{y_2+y_3}. \end{aligned}$$

This is an unnormalized density of the beta distribution $\mathrm{Be}(z+y_4+1, y_2+y_3+1)$, which can be sampled directly.

The full conditional of Z is trickier to recognize. Notice that Z is an integer such that $0 \le Z \le y_1$. It is critical to notice that the normalizing constant of the multinomial pmf $p(z, y \mid \theta)$ depends on z. While you can omit from the likelihood any terms which depend only on the *observed* data, you must keep those terms which depend on the unknowns: parameters or *missing* data.

As a function of z,

$$\begin{aligned} p(z \mid \theta, y) &\propto p(z, y \mid \theta)\, p(\theta) = p(z, y \mid \theta) \\ &= \frac{n!}{z!\,(y_1 - z)!\, y_2!\, y_3!\, y_4!} \left(\frac{1}{4}\theta\right)^{z} \left(\frac{1}{2}\right)^{y_1 - z} \left(\frac{1}{4}(1-\theta)\right)^{y_2} \left(\frac{1}{4}(1-\theta)\right)^{y_3} \left(\frac{1}{4}\theta\right)^{y_4} \\ &\propto \frac{y_1!}{z!\,(y_1 - z)!} \left(\frac{\theta}{4}\right)^{z} \left(\frac{1}{2}\right)^{y_1 - z} \\ &= \binom{y_1}{z} \left(\frac{\frac{\theta}{4}}{\frac{\theta}{4} + \frac{1}{2}}\right)^{z} \left(\frac{\frac{1}{2}}{\frac{\theta}{4} + \frac{1}{2}}\right)^{y_1 - z} \left(\frac{\theta}{4} + \frac{1}{2}\right)^{z + y_1 - z} \\ &\propto \binom{y_1}{z} \left(\frac{\theta}{2+\theta}\right)^{z} \left(1 - \frac{\theta}{2+\theta}\right)^{y_1 - z}, \qquad z = 0, 1, \ldots, y_1. \end{aligned}$$

From this we see that the full conditional of Z is the binomial $\mathrm{Bin}(y_1, \theta/(2+\theta))$, which we also are able to simulate directly. Gibbs sampling in the reformulated model is straightforward. △

8.4 Probit regression

We now consider a regression model, where each of the responses is binary: zero of one. In other words, each of the responses has the Bernoulli distribution (the binomial distribution with sample size one). Conditionally on the parameter vector θ, the responses Y_i are assumed to be independent, and Y_i is assumed to have success probability

$$q_i(\theta) = P(Y_i = 1 \mid \theta),$$

which is a function of the parameter vector θ. That is, the model assumes that

$$[Y_i \mid \theta] \overset{\text{ind}}{\sim} B(q_i(\theta)), \qquad i = 1, \ldots, n,$$

where $B(p)$ is the Bernoulli distribution with success probability $0 \le p \le 1$.

We assume that the success probability of the i'th response depends on θ and on the value of the covariate vector x_i for the i'th case. The covariate vector consists of observed characteristics which might influence the probability of success. We would like to model the success probability in terms of a linear predictor, which is the inner product $x_i^T \theta$ of the covariate vector and the parameter vector. For instance, if we have observed a single explanatory scalar variable t_i connected with ther response y_i, then the linear predictor could be

$$x_i^T \theta = \alpha + \beta \, t_i, \qquad x_i = (1, t_i), \quad \theta = (\alpha, \beta).$$

Notice that we typically include the constant "1" in the covariate vector.

The linear predictor is not constrained to the range $[0, 1]$ of the probability parameter, and therefore we need to map the values of the linear predictor into that range. The standard solution is to posit that

$$q_i(\theta) = F(x_i^T \theta), \qquad i = 1, \dots, n.$$

where F is the cumulative distribution function of some continuous distribution. Here F can be called a *link function*. Since $0 \le F \le 1$, here $q_i(\theta)$ is a valid probability parameter for the Bernoulli distribution for any value of θ.

In *probit regression* we take $F = \Phi$, where Φ is the cdf of the standard normal $N(0, 1)$, i.e., we assume that

$$q_i(\theta) = P(Y_i = 1 \mid \theta) = \Phi(x_i^T \theta), \qquad i = 1, \dots, n. \tag{8.2}$$

We can complete the Bayesian model by taking as our prior, e.g., the normal distribution with mean μ_0 and precision matrix Q_0,

$$p(\theta) = N(\theta \mid \mu_0, Q_o^{-1}).$$

An even more popular choice for the link function in binary regression is the logit link, which corresponds to the choice

$$F(u) = \frac{\mathrm{e}^u}{1 + \mathrm{e}^u} = \operatorname{logit}^{-1}(u).$$

The probit and logit regression models belong to the class of generalized linear models (GLMs). The logit link has a special status in binary regression, since the logit link happens to be what is known as the canonical link function. The maximum likelihood estimate (MLE) for probit or logit regression can be calculated with standard software, e.g., using the function `glm` of R.

We can write the likelihood for probit or logit regression immediately, i.e.,

$$p(y \mid \theta) = \prod_{i=1}^n p(y_i \mid \theta),$$

where

$$p(y_i \mid \theta) = F(x_i^T \theta)^{y_i} \left(1 - F(x_i^T \theta)\right)^{1 - y_i}, \qquad i = 1, \dots, n.$$

Posterior inference can be based directly on this expression. Gibbs sampling seems impossible, but a suitable MCMC algorithm could be, e.g., the independence sampler with a multivariate Student's t distribution, whose center and covariance matrix are selected based on the MLE and its approximate covariance matrix, which can be calculated with standard software.

From now on, we will discuss the probit regression model, and its well-known auxiliary variable reformulation, due to Albert and Chib [1]. Let us introduce n latent variables (i.e., unobserved random variables)

$$[Z_i \mid \theta] \overset{\text{ind}}{\sim} N(x_i^T\theta, 1), \qquad i = 1, \ldots, n.$$

This notation signifies that the Z_i's are independent, conditionally on θ. We may represent the latent variables Z_i using n i.i.d. random variables $\epsilon_i \sim N(0,1)$ (which are independent of everything else),

$$Z_i = x_i^T\theta + \epsilon_i, \qquad i = 1, \ldots, n.$$

Consider n RVs Y_i which are defined by

$$Y_i = 1(Z_i > 0) = \begin{cases} 1, & \text{when } Z_i > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Conditionally on θ, the random variables Y_i are independent, Y_i takes on the value zero or one, and

$$P(Y_i = 1 \mid \theta) = P(Z_i > 0 \mid \theta) = P(x_i^T\theta + \epsilon_i > 0) = P(-\epsilon_i < x_i^T\theta) = \Phi(x_i^T\theta).$$

Here we used the fact that $-\epsilon_i \sim N(0,1)$ which follows from the symmetry of the standard normal. Therefore the marginal distribution of $Y = (Y_1, \ldots, Y_n)$ given θ is the same as in the original probit regression model (8.2). Our reformulated model has the structure

$$\mathrm{p}_{\text{aug}}(y, \theta, z) = \mathrm{p}_{\text{act}}(\theta)\ \mathrm{p}_{\text{aug}}(y, z \mid \theta),$$

and we have just argued that

$$\int \mathrm{p}_{\text{aug}}(y, z \mid \theta)\,\mathrm{d}z = \mathrm{p}_{\text{act}}(y \mid \theta).$$

This shows that our reformulated model is equivalent with the original probit regression model.

The reformulated probit regression model has the following hierarchical structure,

$$\begin{aligned} \Theta &\sim N(\mu_0, Q_0^{-1}) && (8.3) \\ [Z \mid \Theta = \theta] &\sim N(X\theta, I) && (8.4) \\ Y &= 1_+(Z), && (8.5) \end{aligned}$$

where X is the known design matrix with ith row equal to x_i^T, Z is the column vector of latent variables, and $1_+(Z)$ means the vector

$$1_+(Z) = \begin{bmatrix} 1(Z_1 > 0) \\ \vdots \\ 1(Z_n > 0) \end{bmatrix},$$

where we write $1(Z_i > 0)$ for the indicator $1_{(0,\infty)}(Z_i)$. Therefore we can regard the original probit regression model as a missing data problem where we have a normal regression model on the latent data $Z = (Z_1, \ldots, Z_n)$ and the observed responses Y_i are incomplete in that we only observe whether $Z_i > 0$ or $Z_i \le 0$.

The joint distribution of the reformulated model can be expressed as

$$p(\theta, y, z) = p(\theta)\, p(z \mid \theta)\, p(y \mid z),$$

where

$$p(y \mid z) = \prod_{i=1}^{n} p(y_i \mid z_i),$$

and further

$$p(y_i \mid z_i) = 1(y_i = 1(z_i > 0)) = 1(z_i > 0)\, 1(y_i = 1) + 1(z_i \le 0)\, 1(y_i = 0).$$

(Y_i is a deterministic function of Z_i. The preceding representation is possible, since Y_i has a discrete distribution.)

The full conditional of θ is easy, since

$$p(\theta \mid z, y) \propto p(\theta, y, z) \propto p(\theta)\, p(z \mid \theta),$$

but this is the same as the posterior for a linear regression model, which is given by a certain multivariate normal distribution $N(\mu_1, Q_1^{-1})$, whose parameters μ_1 and Q_1 depend on the conditioning variables z and y. It is easy to derive expressions for μ_1 and Q_1.

The other full conditional distribution is also easy to derive. As a function of z, we have

$$p(z \mid \theta, y) \propto p(z \mid \theta)\, p(y \mid z) = \prod_{i=1}^{n} N(z_i \mid x_i^T \theta, 1)\, p(y_i \mid z_i)$$

This is a distribution, where the components Z_i are independent, and follow truncated normal distributions, i.e.,

$$\begin{aligned} [Z_i \mid \theta, y] &\sim N(x_i^T \theta, 1)\, 1(Z_i > 0), && \text{if } y_i = 1, \\ [Z_i \mid \theta, y] &\sim N(x_i^T \theta, 1)\, 1(Z_i \le 0), && \text{if } y_i = 0. \end{aligned}$$

Notice that the side of the truncation for Z_i depends on the value of the binary response y_i. Simulating the full conditional distribution $p(z \mid \theta, y)$ is also straightforward, since we only have to draw independently n values from truncated normal distributions with known parameters and known semi-infinite truncation intervals. Since all the needed full conditional distributions are easily simulated, implementing the Gibbs sampler is straightforward in the latent variable reformulation.

What is the practical benefit of the latent variable reformulation of the probit regression model? In the original formulation of the probit regression model, the components of θ are dependent in their posterior. MCMC sampling will be inefficient unless we manage to find a proposal distribution which is adapted to the form of the posterior distribution. After the reformulation, Gibbs sampling becomes straightforward. In the latent variable reformulation, most of the dependencies in the posterior are transferred to the multivariate normal distribution $p(\theta \mid z, y)$, where they are easy to handle. The components of Z are independent in the other needed full conditional distribution $p(z \mid \theta, y)$.

8.5 Scale mixtures of normals

Student's t distribution with $\nu > 0$ degrees of freedom can be expressed as a scale mixture of normal distributions as follows. If

$$\Lambda \sim \mathrm{Gam}(\nu/2, \nu/2), \quad \text{and} \quad [W \mid \Lambda = \lambda] \sim N(0, \frac{1}{\lambda}),$$

then the marginal distribution of W is t_ν. We can use this property to eliminate Student's t distribution from any statistical model.

Albert and Chib considered approximating the logit link with the t_ν link in binary regression. The logit link is already well approximated by the probit link in the sense that

$$\mathrm{logit}^{-1}(u) \approx \Phi\left(\sqrt{\frac{\pi}{8}}u\right),$$

when u is near zero. Here the scaling factor $\sqrt{\pi/8}$ has been selected so that the derivatives of the two curves are equal for $u = 0$. The approximation is not perfect away from zero. However, if one uses the distribution function F_ν of the t_ν distribution (e.g., with $\nu = 8$ degrees of freedom), then one can choose the value of the scaling factor s so that we have a much better approximation

$$\mathrm{logit}^{-1}(u) \approx F_\nu(su)$$

for all real u. Making use of the scaling factor s, we can switch between a logit regression model and its t_ν regression approximation.

We now consider, how we can reformulate the binary regression model which has the t_ν link, i.e.,

$$[Y_i \mid \theta] \overset{\text{ind}}{\sim} B(F_\nu(x_i^T \theta)), \qquad i = 1, \ldots, n. \tag{8.6}$$

Here the degrees of freedom parameter ν is fixed. Also this reformulation is due to Albert and Chib [1].

The first step is to notice that we can represent the responses as

$$Y_i = 1(Z_i > 0), \quad \text{where} \quad Z_i = x_i^T\theta + W_i, \qquad i = 1, \ldots, n,$$

where $W_i \sim t_\nu$ are i.i.d. and independent of everything else. This holds since

$$P(Z_i > 0 \mid \theta) = P(x_i^T\theta + W_i > 0) = P(-W_i < x_i^T\theta) = F_\nu(x_i^T\theta).$$

Here we used the fact that $-W_i \sim t_\nu$ which follows from symmetry of the t distribution. Besides, the Z_i's are independent, conditionally on θ. Next we eliminate the t_ν distribution by introducing n i.i.d. latent variables Λ_i, each having the $\mathrm{Gam}(\nu/2, \nu/2)$ distribution. If we choose $N(\mu_0, Q_0^{-1})$ as the prior for Θ, then we end up with the following hierarchical model

$$\begin{aligned}
\Theta &\sim N(\mu_0, Q_0^{-1}), && (8.7)\\
\Lambda_i &\overset{\text{i.i.d.}}{\sim} \mathrm{Gam}(\nu/2, \nu/2), \qquad i = 1, \ldots, n && (8.8)\\
[Z \mid \Theta = \theta, \Lambda = \lambda] &\sim N\left(X\theta, [\mathrm{diag}(\lambda_1, \ldots, \lambda_n)]^{-1}\right), && (8.9)\\
Y &= 1_+(Z). && (8.10)
\end{aligned}$$

This reformulation is equivalent with the original model (8.6).

The full conditionals in the reformulated model are easy to derive. The full conditional of θ is a multivariate normal. The full conditional of $\Lambda = (\Lambda_1, \dots, \Lambda_n)$ is the distribution of n independent gamma distributed variables with certain parameters. The full conditional of Z is, once again, a distribution, where the components are independent and have truncated normal distributions.

Another well-known distribution, which can be expressed as a scale mixture of normal distributions is the Laplace distribution (the double exponential distribution), which has the density

$$\frac{1}{2} \mathrm{e}^{-|y|}, \qquad y \in \mathbb{R}\,.$$

If Y has the Laplace distribution, then it can be expressed as follows

$$V \sim \mathrm{Exp}(1/2) \quad \text{and} \quad [Y \mid V = v] \sim N(0, v).$$

This relationship can be used to eliminate the Laplace distribution from any statistical model.

Even the logistic distribution with distribution function $\mathrm{logit}^{-1}(z)$ can be expressed as a scale mixture of normals, but then one needs the Kolmogorov-Smirnov distribution, whose density and distribution function are, however, available only as series expansions. Using this device, one can reformulate the logistic regression model exactly using the Kolmogorov-Smirnov distribution, multivariate normal distribution and truncation.

8.6 Literature

The slice sampler was proposed by Neal [4]. The data augmentation in the genetic linkage example is from the article by Tanner and Wong [5], who borrowed the idea from earlier work on the EM algorithm. The auxiliary variable formulation of probit regression was proposed by Albert and Chib [1]. Also the reformulation of the t link is from this article. Scale mixtures of normals were characterized by Andrews and Mallows [2]. Holmes and Held have managed to use the exact reformulation of the logit link as a scale mixture of normals [3].

Bibliography

[1] James H. Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88:669–679, 1993.

[2] D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society, Series B*, 36:99–102, 1974.

[3] Chris C. Holmes and Leonhard Held. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1:145–168, 2006.

[4] Radford M. Neal. Slice sampling. *Annals of Statistics*, 23:705–767, 2003.

[5] Martin A. Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82:528–550, 1987.

Chapter 9

The EM Algorithm

The EM (Expectation–Maximization) algorithm is an iterative method for finding the mode of a marginal posterior density. It can also be used for finding the mode of a marginal likelihood function. The idea is to replace the original maximization problem by a sequence of simpler optimization problems. In many examples the maximizers of the simple problems can be obtained in closed form.

Often the EM algorithm is applied in an auxiliary variable (latent variable) formulation $p(y, \theta, z)$ of the original model $p(y, \theta)$, where θ is the parameter of interest, and Z is the auxiliary variable (or latent variable or missing data). Then the marginal posterior of θ, namely

$$p(\theta \mid y) = \int p(\theta, z \mid y) \, \mathrm{d}z,$$

is the posterior in the original model, and the marginal likelihood of θ, namely

$$p(y \mid \theta) = \int p(y, z \mid \theta) \, \mathrm{d}z$$

is the likelihood in the original model. In such a case the EM algorithm can be used to find the posterior mode or the MLE (maximum likelihood estimate) of the original model.

9.1 Formulation of the EM algorithm

Let Z be the auxiliary variable and θ the parameter of interest. Often the auxiliary variable can be interpreted as missing data. The EM algorithm can be formulated either for the mode of the marginal posterior of θ or for the mode of the marginal likelihood of θ. In both cases one defines a function, usually called Q, which depends on two variables, θ and θ_0, where θ_0 stands for the current guess of the parameter vector θ_0. The function $Q(\theta \mid \theta_0)$ is defined as a certain expected value.

The EM algorithm alternates between two steps: first one calculates the Q function given the current guess θ_0 for the parameter vector (E-step), and then one maximizes $Q(\theta \mid \theta_0)$ with respect to θ in order to define the new guess for θ (M-step). This procedure is repeated until a fixed point of Q is obtained (or some other termination criterion is satisfied). This idea is formalized in algorithm 19. There arg max denotes the maximizing argument (maximum point) of the function it operates on. If the maximizer is not unique, we may select any global maximizer.

Algorithm 19: The EM algorithm.

```
Input: An initial value $\theta^{(0)}$.
$k \leftarrow 0$;
repeat
    (E-step) Calculate the function $Q(\theta \mid \theta^{(k)})$;
    (M-step) Maximize $Q(\theta \mid \theta^{(k)})$ with respect to $\theta$:
        $\theta^{(k+1)} \leftarrow \arg\max_\theta Q(\theta \mid \theta^{(k)})$
    Set $k \leftarrow k + 1$
until the termination criterion is satisfied ;
Return the last calculated value $\theta^{(k)}$;
```

Next we define the function Q for the two different objectives. When we want to calculate the mode of the (marginal) posterior density, we define $Q(\theta \mid \theta_0)$ as the expected value of the log joint posterior density, conditioned on the data and on the current value θ_0,

$$\begin{aligned} Q(\theta \mid \theta_0) &= E\left[\log p(\theta, Z \mid y) \mid \theta_0, y\right] \\ &= E\left[\log f_{\Theta,Z|Y}(\theta, Z \mid y) \mid \Theta = \theta_0, Y = y\right] \\ &= \int \log f_{\Theta,Z|Y}(\theta, z \mid y)\, f_{Z|\Theta,Y}(z \mid \theta_0, y)\,\mathrm{d}z. \end{aligned} \tag{9.1}$$

The only random object in the above expected value is Z, and we use its distribution conditioned on the current value θ_0 and the data y.

When we want to calculate the mode of the (marginal) likelihood of θ, we define $Q(\theta \mid \theta_0)$ as the expected complete-data log-likelihood, conditioning on the data and on the current value θ_0,

$$\begin{aligned} Q(\theta \mid \theta_0) &= E\left[\log p(y, Z \mid \theta) \mid \theta_0, y\right] \\ &= E\left[\log f_{Y,Z|\Theta}(y, Z \mid \theta) \mid \Theta = \theta_0, Y = y\right] \\ &= \int \log f_{Y,Z|\Theta}(y, z \mid \theta)\, f_{Z|\Theta,Y}(z \mid \theta_0, y)\,\mathrm{d}z. \end{aligned} \tag{9.2}$$

The Q function is defined as an expectation of a sum of a number terms. Luckily, we can treat all of the terms which do not depend on θ as constants. Namely, in the M-step we select a maximum point of the function $\theta \mapsto Q(\theta \mid \theta_0)$, and the ignored constants only shift the object function but do not change the location of the maximum point. That is, the functions

$$Q(\theta \mid \theta_0) \quad \text{and} \quad Q(\theta \mid \theta_0) + c(\theta_0, y)$$

achieve their maxima at the same points, when the "constant" $c(\theta_0, y)$ does not depend on the variable θ. In particular, we can ignore any factors which depend solely on the observed data y.

The maximization problem (M-step) can be solved in closed form in many cases where the joint posterior (or complete data likelihood) belongs to the exponential family. Then the E- and M-steps boil down to the following steps: finding the expectations (given the current θ_0) of the sufficient statistics (which now depend on the missing data Z), and maximizing the resulting function with respect to the parameters θ.

If the maximizer cannot be solved analytically, then instead of the maximum point one can (in the M-step) select any value $\theta^{(k+1)}$ such that

$$Q(\theta^{(k+1)} \mid \theta^{(k)}) > Q(\theta^{(k)} \mid \theta^{(k)}).$$

The resulting algorithm is then called the generalized EM algorithm (GEM).

We will show later that the logarithm of the marginal posterior

$$\log f_{\Theta|Y}(\theta^{(k)} \mid y)$$

increases monotonically during the iterations of the EM or the GEM algorithms, if one defines Q by (9.1). On the other hand, if one defines Q by (9.2), then the log marginal likelihood

$$\log f_{Y|\Theta}(y \mid \theta^{(k)})$$

increases monotonically during the iterations. If these functions can be calculated easily, then a good check of the correctness of the implementation is to check that they indeed increase at each iteration.

Because of this monotonicity property, the EM algorithm converges to some local mode of the object function (except in some artificially constructed cases). If the object function has multiple modes, then one can try to find all of them by starting the EM iterations at many points scattered throughout the parameter space.

9.2 EM algorithm for probit regression

We return to the latent variable reformulation of the probit regression problem, i.e.,

$$\begin{aligned} \Theta &\sim N(\mu_0, R_0^{-1}) \\ [Z \mid \Theta = \theta] &\sim N(X\theta, I) \\ Y &= 1_+(Z), \end{aligned}$$

where X is the known design matrix, Z is the column vector of latent variables, and $1_+(Z)$ is the vector of indicators $1(Y_i > 0)$. We use the symbols ϕ and Φ for the density and df of the standard normal $N(0, 1)$, and use R_0 to denote the precision matrix of the prior.

We have already obtained the distribution of the latent variables given θ and the data, $p(z \mid \theta, y)$. In it, the latent variables Z_i are independent and have the following truncated normal distributions

$$\begin{aligned} [Z_i \mid \theta, y] &\sim N(x_i^T\theta, 1)\, 1(Z_i > 0), &\qquad \text{if } y_i = 1, \\ [Z_i \mid \theta, y] &\sim N(x_i^T\theta, 1)\, 1(Z_i \le 0), &\qquad \text{if } y_i = 0. \end{aligned}$$

Now the joint posterior is

$$p(\theta, z \mid y) \propto p(y, \theta, z) = p(y \mid z)\, p(z \mid \theta)\, p(\theta).$$

Here $p(y \mid z)$ is simply the indicator of the constraints $y = 1_+(z)$. For any y and z values which satisfy the constraints $y = 1_+(z)$, the log joint posterior is given by

$$\begin{aligned}\log p(\theta, z \mid y) &= \log p(z \mid \theta) + \log p(\theta) + c_1 \\ &= -\frac{1}{2}(\theta - \mu_0)^T R_0 (\theta - \mu_0) - \frac{1}{2}(z - X\theta)^T (z - X\theta) + c_2 \\ &= -\frac{1}{2}(\theta - \mu_0)^T R_0 (\theta - \mu_0) - \frac{1}{2} z^T z + \theta^T X^T z - \frac{1}{2}\theta^T X^T X \theta + c_2\end{aligned}$$

where the constants c_i depends on the data y and the known hyperparameters, but not on z, θ or θ_0.

Since now

$$Q(\theta \mid \theta_0) = E[\log p(\theta, Z \mid y) \mid \Theta = \theta_0, Y = y],$$

at first sight it may appear that we need to calculate both the expectations

$$v(\theta_0) = E[Z^T Z \mid \Theta = \theta_0, Y = y], \quad \text{and} \quad m(\theta_0) = E[Z \mid \Theta = \theta_0, Y = y],$$

but on further thought we notice that we actually need only the expectation $m(\theta_0)$. This is so, since the term containing $z^T z$ in $\log p(\theta, z \mid y)$ does not depend on θ. In the maximization of $Q(\theta \mid \theta_0)$ its expectation therefore only shifts the object function but does not affect the location of the maximizer.

Let us next solve the maximizer of $\theta \mapsto Q(\theta \mid \theta_0)$ and then check which quantities need to be calculated. In the following, c_i is any quantity, which does not depend on the variable θ (but may depend on y, θ_0 or the known hyperparameters).

$$\begin{aligned}Q(\theta \mid \theta_0) &= E[\log p(\theta, Z \mid y) \mid \Theta = \theta_0, Y = y] \\ &= -\frac{1}{2}(\theta - \mu_0)^T R_0 (\theta - \mu_0) - \frac{1}{2}\theta^T X^T X \theta + \theta^T X^T m(\theta_0) + c_3 \qquad (9.3) \\ &= -\frac{1}{2}\theta^T (R_0 + X^T X)\theta + \theta^T \left[R_0 \mu_0 + X^T m(\theta_0)\right] + c_4\end{aligned}$$

We now make the following observations.

1. The matrix $R_0 + X^T X$ is symmetric and positive definite. Symmetry is obvious, and for any $v \neq 0$,

   $$v^T (R_0 + X^T X) v = v^T R_0 v + v^T X^T X v > 0,$$

   since $v^T R_0 v > 0$ and $v^T X^T X v = (Xv)^T (Xv) \geq 0$.

2. If the matrix K is symmetric and positive definite, then the maximizer of the quadratic form

   $$-\frac{1}{2}(\theta - a)^T K (\theta - a)$$

   is a, since the quadratic form vanishes if and only if $\theta = a$.

3. The preceding quadratic form can developed as

$$-\frac{1}{2}(\theta - a)^T K(\theta - a) = -\frac{1}{2}\theta^T K\theta + \theta^T Ka + \text{constant}.$$

Therefore, the maximum point of

$$-\frac{1}{2}\theta^T K\theta + \theta^T b + c,$$

where K is assumed to be symmetric and positive definite, is

$$\theta = K^{-1}b.$$

(An alternative way to derive the formula for the maximum point is to equate the gradient $-K\theta + b$ of the quadratic function to the zero vector, and to observe that the Hessian $-K$ is negative definite.)

Based on the preceding observations, the maximizer of $\theta \mapsto Q(\theta \mid \theta_0)$ given in eq. (9.3) is given by

$$\theta_1 = (R_0 + X^T X)^{-1}(R_0\mu_0 + X^T m(\theta_0)). \tag{9.4}$$

However, we still need to calculate a concrete formula for the vector

$$m(\theta_0) = E[Z \mid \Theta = \theta_0, Y = y].$$

We need a formula for the expected value of the truncated normal distribution $N(\mu, \sigma^2)1_{(\alpha,\beta)}$ corresponding to the unnormalized density

$$f(v) \propto N(v \mid \mu, \sigma^2)1_{(\alpha,\beta)}(v) \tag{9.5}$$

where we can have $\alpha = -\infty$ or $\beta = \infty$. The moment generating function of this distribution is easy to calculate. Then we obtain its expected value (and higher moments, if need be) by differentiating the result.

Let Φ be the distribution function and ϕ the density function of the standard normal $N(0,1)$. If V has the truncated normal distribution (9.5), then a simple calculation shows that

$$\begin{aligned} M(t) &= E(\exp(tV)) \\ &= \exp(\mu t + \frac{1}{2}\sigma^2 t^2)\frac{\Phi\left(\frac{\beta-\mu}{\sigma} - \sigma t\right) - \Phi\left(\frac{\alpha-\mu}{\sigma} - \sigma t\right)}{\Phi\left(\frac{\beta-\mu}{\sigma}\right) - \Phi\left(\frac{\alpha-\mu}{\sigma}\right)} \end{aligned} \tag{9.6}$$

The expected value of a distribution equals the first derivative of its moment generating function at $t = 0$, and hence

$$E[V] = M'(0) = \mu - \sigma\frac{\phi\left(\frac{\beta-\mu}{\sigma}\right) - \phi\left(\frac{\alpha-\mu}{\sigma}\right)}{\Phi\left(\frac{\beta-\mu}{\sigma}\right) - \Phi\left(\frac{\alpha-\mu}{\sigma}\right)} \tag{9.7}$$

Using the preceding results, we see that the components $m(\theta_0)_i$ of the vector $m(\theta_0)$ are given by

$$m(\theta_0)_i = \begin{cases} x_i^T\theta_0 + \dfrac{\phi(-x_i^T\theta_0)}{1-\Phi(-x_i^T\theta_0)}, & \text{if } y_i = 1 \\ x_i^T\theta_0 - \dfrac{\phi(-x_i^T\theta_0)}{\Phi(-x_i^T\theta_0)}, & \text{if } y_i = 0. \end{cases} \tag{9.8}$$

Formulas (9.4) and (9.8) define one step of the EM algorithm for calculating the posterior mode in probit regression. The EM algorithm for the MLE of probit regression is obtained from formulas (9.4) and (9.8) by setting R_0 as the zero matrix. (Then we need to assume that X^TX is positive definite.)

The truncated normal distribution features in many other statistical models besides the latent variable formulation of probit regression. One famous example is the tobit regression model. This is a linear regression model, where the observations are censored. Since the truncated normal distribution pops up in many different contexts, it is useful to know that there is a simple formula (9.6) for its moment generating function.

9.3 Why the EM algorithm works

The proof of the monotonicity of the EM and GEM algorithms is based on the non-negativity of the Kullback-Leibler divergence. If f and g are two densities, then the K-L divergence (or relative entropy) of g from f is defined by

$$D(f \parallel g) = \int f \, \ln \frac{f}{g}, \tag{9.9}$$

where the integral is calculated over the whole space. If the supports of f and g are not the whole space, then we use the conventions

$$f(x) \, \ln \frac{f(x)}{g(x)} = \begin{cases} 0, & \text{if } f(x) = 0, \\ \infty, & \text{if } f(x) > 0 \text{ and } g(x) = 0. \end{cases}$$

We will show that the K-L divergence is always non-negative. Therefore we can use it to measure the distance of g from f. However, the K-L divergence is not a metric (on the space of densities), since it is even not symmetric.

The proof of the non-negativity can be based on the elementary inequality

$$\ln x \le x - 1 \qquad \forall x > 0, \tag{9.10}$$

where equality holds if and only if $x = 1$. This inequality follows from the concavity of the logarithm function. The graph of a concave function lies below each of its tangents, and right hand side of (9.10) is the tangent at $x_0 = 1$.

Theorem 4. *Let f and g be densities defined on the same space. Then*

$$D(f \parallel g) \ge 0,$$

and equality holds if and only if $f = g$ (almost everywhere).

Proof. We give the proof only in the case, when f and g have the same support, i.e., when the sets $\{x : f(x) > 0\}$ and $\{x : g(x) > 0\}$ are the same (except perhaps modulo a set of measure zero). Extending the proof to handle the general case is straightforward. In the following calculation, the integral extends only over the common support of f and g.

$$\begin{aligned}(-1)D(f \parallel g) &= \int -f \ln \frac{f}{g} = \int f \ln \frac{g}{f} \\ &\le \int f(\frac{g}{f} - 1) \qquad \text{by (9.10)} \\ &= \int (g - f) = 1 - 1 = 0.\end{aligned}$$

We have equality if and only if

$$\ln \frac{g}{f} = \frac{g}{f} - 1,$$

almost everywhere, and this happens if and only if $f = g$ almost everywhere. □

The following theorem establishes the monotonicity of EM or GEM iterations.

Theorem 5. *Define the function Q by either the equation* (9.1) *or by* (9.2)*. Let θ_0 and θ_1 be any values such that*

$$Q(\theta_1 \mid \theta_0) \ge Q(\theta_0 \mid \theta_0). \tag{9.11}$$

Then, with the definition (9.1) *we have*

$$f_{\Theta|Y}(\theta_1 \mid y) \ge f_{\Theta|Y}(\theta_0 \mid y),$$

and with the definition (9.2) *we have*

$$f_{Y|\Theta}(y \mid \theta_1) \ge f_{Y|\Theta}(y \mid \theta_0).$$

In either case, if we have strict inequality in the assumption (9.11)*, then we have strict inequality also in the conclusion.*

Proof. We consider first the proof for the definition (9.1). We will use the abbreviated notations, and make use of the identity

$$p(\theta \mid y) = \frac{p(\theta, z \mid y)}{p(z \mid \theta, y)}.$$

For any θ, we have

$$\begin{aligned}\ln p(\theta \mid y) &= \int p(z \mid \theta_0, y) \, \ln p(\theta \mid y) \, \mathrm{d}z \\ &= \int p(z \mid \theta_0, y) \, \ln \frac{p(\theta, z \mid y)}{p(z \mid \theta, y)} \, \mathrm{d}z \\ &= Q(\theta \mid \theta_0) - \int p(z \mid \theta_0, y) \, \ln p(z \mid \theta, y) \, \mathrm{d}z\end{aligned}$$

Using this identity at the points θ_1 and θ_0, we obtain

$$\begin{aligned}
&\ln p(\theta_1 \mid y) - \ln p(\theta_0 \mid y) \\
&\qquad = Q(\theta_1 \mid \theta_0) - Q(\theta_0 \mid \theta_0) + \int p(z \mid \theta_0, y) \, \ln \frac{p(z \mid \theta_0, y)}{p(z \mid \theta_1, y)} \, \mathrm{d}z \\
&\qquad \geq Q(\theta_1 \mid \theta_0) - Q(\theta_0 \mid \theta_0),
\end{aligned}$$

since the K-L divergence is non-negative. This proves the claim for (9.1).

The proof for the definition (9.2) starts from the identity

$$\begin{aligned}
\ln p(y \mid \theta) &= \int p(z \mid \theta_0, y) \, \ln p(y \mid \theta) \, \mathrm{d}z \\
&= \int p(z \mid \theta_0, y) \, \ln \frac{p(y, z \mid \theta)}{p(z \mid \theta, y)} \, \mathrm{d}z \\
&= Q(\theta \mid \theta_0) - \int p(z \mid \theta_0, y) \, \ln p(z \mid \theta, y) \, \mathrm{d}z.
\end{aligned}$$

Rest of the proof is the same as before. □

9.4 Literature

The name EM algorithm was introduced by Dempster, Laird and Rubin in [1]. Many special cases of the method had appeared in the literature already in the 1950's, but this article gave a unified structure to the previous methods. The book [3] is dedicated to the EM algorithm and its variations. Many authors have extended the EM algorithm so that one obtains also the covariance matrix of the (marginal) posterior, or the approximate covariance matrix of the (marginal) MLE, see, e.g., [3] or [2].

Bibliography

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society (Series B)*, 45:1–38, 1977.

[2] Geof H. Givens and Jennifer A. Hoeting. *Computational Statistics*. Wiley-Interscience, 2005.

[3] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley series in probability and statistics. John Wiley & Sons, Inc., 1997.