Bayesian Modeling

Bayesian computation

Yixin Wang

Preliminary Draft.
Please do not distribute.

6 Posterior approximation with the Gibbs sampler

6.1 Conjugate vs non-conjugate prior

In the previous section we considered a particular prior for the normal sampling model Normal(θ, σ^2). This is a conjugate prior for the parameters θ, σ^2 (or alternatively, $\theta, \tilde{\sigma}^2 = 1/\sigma^2$):

$$\begin{aligned} \tilde{\sigma}^2 &\sim \text{Gamma}(\nu_0/2, \nu_0\sigma_0^2) \\ \theta|\tilde{\sigma}^2 &\sim \text{Normal}(\mu_0, \kappa_0\tilde{\sigma}^2). \end{aligned}$$

We found that by applying the Bayes update, the posterior distribution $p(\theta, \tilde{\sigma}^2|y_1, \ldots, y_n)$ carries the same form:

$$\begin{aligned} \tilde{\sigma}^2|y_1, \ldots, y_n &\sim \text{Gamma}(\nu_n/2, \nu_n\sigma_n^2/2) \\ \theta|y_1, \ldots, y_n, \tilde{\sigma}^2 &\sim \text{Normal}(\mu_n, \tilde{\tau}_n^2). \end{aligned}$$

The posterior distributions' parameters are updated as

$$\begin{aligned} \tilde{\tau}_n^2 &= \kappa_0\tilde{\sigma}^2 + n\tilde{\sigma}^2 =: \kappa_n\tilde{\sigma}^2 \\ \mu_n &= \frac{\kappa_0\tilde{\sigma}^2\mu_0 + (n\tilde{\sigma}^2)\bar{y}}{\kappa_0\tilde{\sigma}^2 + n\tilde{\sigma}^2} = \frac{\kappa_0\mu_0 + n\bar{y}}{\kappa_n}, \end{aligned}$$

and

$$\begin{aligned} \nu_n &= \nu_0 + n \\ \sigma_n^2 &= \frac{1}{\nu_n}\left\{\nu_0\sigma_0^2 + \frac{\kappa_0 n(\mu_0 - \bar{y})^2}{\kappa_0 + n} + (n-1)s^2\right\}. \end{aligned}$$

The price we have to pay for the computational convenience is the coupling between the two parameters θ and $\tilde{\sigma}^2$ imposed in the prior specification. Such coupling results in a prior bias: the higher the precision $\tilde{\sigma}^2$ (the lower the variance value σ), the the more certain we are about parameter θ.

In general, when dealing with multiple parameters, it is difficult to come up with a conjugate prior jointly for all parameters. And even if we can, the discussion from the previous section suggests that it is important to explore non-conjugate priors, because in some situations they may be more appropriate for our understanding of the parameter space.

In the case of the normal model above, we may want to express our uncertainty about θ as *independent* of $\tilde{\sigma}^2$. Such a prior specification is clearly less stringent than the one given above. Intuitively, such a prior would be less subjective. In particular, consider the following independent prior:

$$\tilde{\sigma}^2 \sim \text{Gamma}(\nu_0/2, \nu_0\sigma_0^2/2) \tag{13a}$$

$$\theta \sim \text{Normal}(\mu_0, \tau_0^2). \tag{13b}$$

The particular choices of Gamma and Normal come from our computations in subsection 5.2 and the beginning of subsection 5.3. Although this prior distribution is not conjugate, in the sense that the joint posterior distribution $p(\theta, \tilde{\sigma}^2|y_1, \ldots, y_n)$ does not carry the same form as the prior distribution $p(\theta, \tilde{\sigma}^2)$, the *full conditional distributions* $p(\theta|\tilde{\sigma}^2, y_1, \ldots, y_n)$ and $p(\tilde{\sigma}^2|\theta, y_1, \ldots, y_n)$ can be easily computed and in fact carry the same form as the corresponding marginal prior distribution. The full conditional distributions are the distribution of a parameter given *everything else*, including the data and all remaining parameters. We call prior for which the full conditional distributions have the same form as the marginal prior "semiconjugate".

6.2 The Gibbs sampler

Gibbs sampler is a sampling technique for multivariate distributions that exploits the fact that the full conditional distributions can be easily computed or sampled from. This crucial fact allows one to generate a *dependent* sequence of parameter samples that converge in distribution to the joint posterior distribution of interest.

Continuing with our semiconjugate prior specification given in Eq. (13). From the previous section, we have obtained that (cf. Eq. (5))

$$\theta|\tilde{\sigma}^2, y_1, \ldots, y_n \quad \sim \quad \text{Normal}(\mu_n, \tau_n^2),$$

where

$$\tau_n^2 = \frac{1}{\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}}, \quad \mu_n = \frac{b}{a} = \frac{\frac{1}{\tau_0^2}\mu_0 + \frac{n}{\sigma^2}\bar{y}}{\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}}.$$

Note carefully that the updated parameters μ_n and τ_n are in fact dependent on the conditioned $\tilde{\sigma}^2 = 1/\sigma^2$. And from Eq. (8),

$$\tilde{\sigma}^2|\theta, y_1, \ldots, y_n \quad \sim \quad \text{Gamma}(\nu_n/2, \nu_n\sigma_n^2/2),$$

where

$$\nu_n = \nu_0 + n, \quad \sigma_n^2 = \frac{1}{\nu_n}(\nu_0\sigma_0^2 + ns_n^2),$$

and $s_n^2 = \sum(y_i - \theta)^2/n$, the unbiased estimate of σ^2 if θ were known. Note carefully also that the updated parameter σ_n^2 is dependent of the conditioned θ.

These full conditionals tell us that

- if we know $\tilde{\sigma}^2$, we can draw a sample for θ from $p(\theta|\tilde{\sigma}^2, y_1, \ldots, y_n)$
- if we know θ, we can draw a sample for $\tilde{\sigma}^2$ from $p(\tilde{\sigma}^2|\theta, y_1, \ldots, y_n)$.

These full conditionals do not give us a direct way of drawing a sample from the joint posterior $p(\theta, \tilde{\sigma}^2|y_1, \ldots, y_n)$, but they suggest an iterative procedure for drawing the joint samples $\phi := (\theta, \tilde{\sigma}^2)$. In each iteration, we take turn to draw a random sample for one parameter using the relevant full conditional distribution for that parameter given the latest values of all other parameters.

This procedure is called the Gibbs sampler.

More precisely for our present model, let $\phi^{(s)} := (\theta^{(s)}, \sigma^{2(s)})$, where s is the index for the iterations.

- Start with an arbitrary initial value $\phi^{(1)} = (\theta^{(1)}, \sigma^{2(1)})$.
- For $s = 1, 2, \ldots$
  - sample $\theta^{(s+1)} \sim p(\theta|\tilde{\sigma}^{2(s)}, y_1, \ldots, y_n)$;
  - sample $\tilde{\sigma}^{(2(s+1))} \sim p(\tilde{\sigma}^2|\theta^{(s+1)}, y_1, \ldots, y_n)$;
  - let $\phi^{(s+1)} = \{\theta^{(s+1)}, \tilde{\sigma}^{2(s+1)}\}$.

What this algorithm does is that it generates a *dependent* sequence of parameter vector $\phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(s)}, \ldots$, where the $s+1$-parameter vector $\phi^{(s+1)}$ is generated by the conditional distribution given the previous value $\phi^{(s)}$, namely $p(\phi^{(s+1)}|\phi^{(s)})$. This sequence of random vector $\{\phi^{(s)}\}$ is called a *Markov chain*. Under very weak conditions this Markov chain of random variables converge to a stationary distribution. Moreover, by our construction of the Gibbs sampler, that stationary distribution is the posterior distribution $p(\phi|y_1, \ldots, y_n)$ — the joint posterior distribution of interest.

Note carefully that we do not say that we have obtained a valid sample from the joint posterior $p(\theta, \tilde{\sigma}|y_1, \ldots, y_n)$. What we said is that if we run the Markov chain (the Gibbs sampler) long enough, i.e., if s is large, then $\phi^{(s)}$ can be viewed as a good *approximation* of the posterior sample.

A nice feature of Gibbs samplers is that they tend to be very easy to implement. In R codes:

```
### data
mean.y<-mean(y) ; var.y<-var(y) ; n<-length(y)
###

### starting values
S<-1000
PHI<-matrix(nrow=S,ncol=2)
PHI[1,]<-phi<-c( mean.y, 1/var.y)
###

### Gibbs sampling
set.seed(1)
for(s in 2:S) {

# generate a new theta value from its full conditional
mun<-  ( mu0/t20 + n*mean.y*phi[2] ) / ( 1/t20 + n*phi[2] )
t2n<- 1/( 1/t20 + n*phi[2] )
phi[1]<-rnorm(1, mun, sqrt(t2n) )

# generate a new 1/sigma^2 value from its full conditional
nun<- nu0+n
s2n<- (nu0*s20 + (n-1)*var.y + n*(mean.y-phi[1])^2 ) /nun
phi[2]<- rgamma(1, nun/2, nun*s2n/2)

PHI[s,]<-phi          }
###
```

In this code, we have used the identity:

$$ns_n^2 = \sum_{i=1}^{n}(y_i - \theta)^2 = (n-1)\bar{s}^2 + n(\bar{y} - \theta)^2.$$

The RHS is fast to update with each iteration because $(n-1)\bar{s}^2 = \sum(y_i - \bar{y})^2$ does not change, only $(\bar{y} - \theta)^2$ gets updated.

Let us examine the performance of the Gibbs sampler using the midge data from the previous section and the independent semiconjugate prior (13). A Gibbs sanpler consisting of 1000 iterations were constructed. Fig. 6.1 plots the first 5, 15 and 100 simulated values of the sampler.

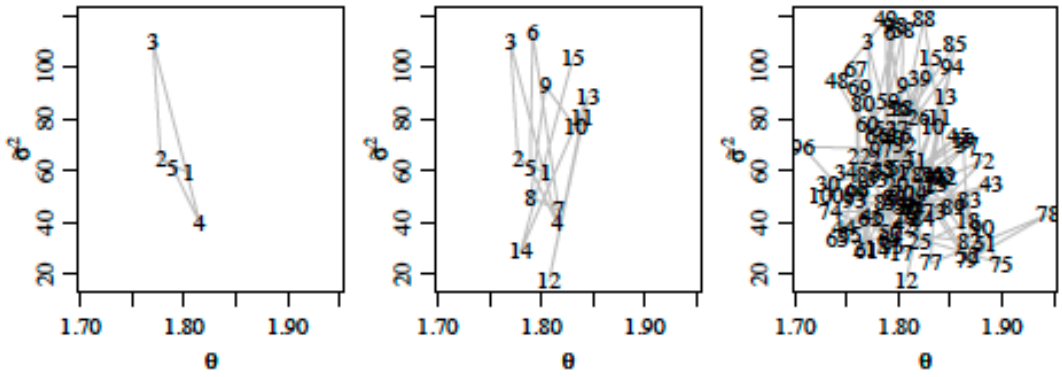

Figure 6.1: The first 5, 15 and 100 iterations of a Gibbs sampler.

Once the Gibbs samples are collected we can find some empirical quantiles, which can be verified to be very close to a discrete approximation of the joint posterior distribution. (Hoff's text book (Chapter 6, Sec. 6.2) gives further details of this discrete approximation technique.)

```
### CI for population mean
> quantile(PHI[,1],c(.025,.5,.975))
    2.5%      50%    97.5%
1.707282 1.804348 1.901129

### CI for population precision
> quantile(PHI[,2],c(.025,.5, .975))
     2.5%       50%     97.5%
 17.48020  53.62511 129.20020


### CI for population standard deviation
> quantile(1/sqrt(PHI[,2]),c(.025,.5, .975))
      2.5%        50%       97.5%
0.08797701 0.13655763 0.23918408
```

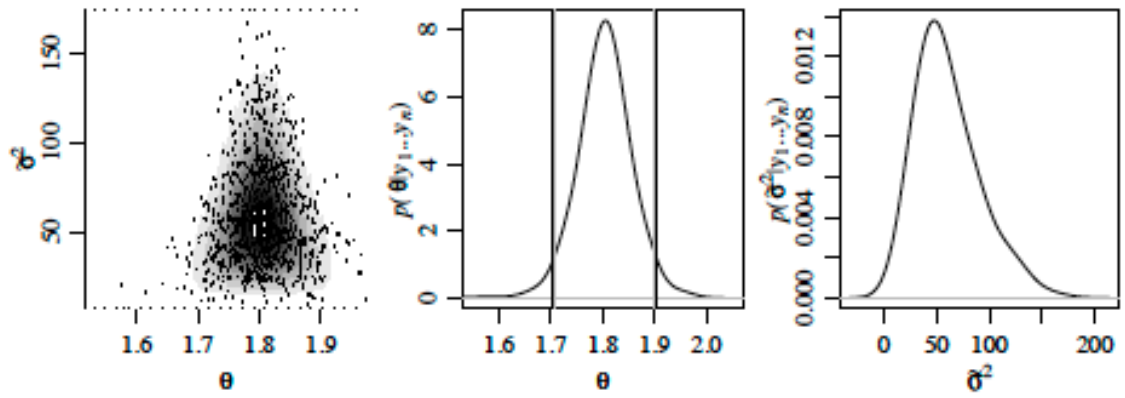

Figure 6.2: The first panel shows 1,000 samples from the Gibbs sampler, plotted over the contours of a discrete approximation. The second and third panels give kernel density estimates to the distributions of Gibbs samples of θ and $\tilde{\sigma}^2$. Vertical gray bars on the second plot indicate 2.5% and 97.5% quantiles of the Gibbs samplers of θ, while nearly identical black vertical bars indicate the 95% confidence interval based on the t-test.

6.3 Markov chain Monte Carlo algorithms

6.3.1 The prototypical Gibbs sampler

Suppose we have a vector of parameters $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)$, and our information about $\boldsymbol{\phi}$ is measured with the probability distribution $p(\boldsymbol{\phi}) = p(\phi_1, \ldots, \phi_q)$. In the example from the previous subsection, $\boldsymbol{\phi} = (\theta, \sigma^2)$ and the probability distribution of interest is $p(\theta, \sigma^2 | y_1, \ldots, y_n)$, a posterior distribution given the observed n-data sample.

Remark 6.1. In Bayesian statistics, the application of Gibbs sampling is typically to posterior distributions, hence the conditioning on the observed data. However, it is important to note that Gibbs sampler is applicable to any joint probability distribution for a random vector $\boldsymbol{\phi}$ of interest; regardless of whether we are dealing with an additional conditioning (in the case of Bayesian inference), or not.

The general recipe should be clear. Given a starting point $\boldsymbol{\phi}^{(0)} = \{\phi_1^{(0)}, \ldots, \phi_q^{(0)}\}$, the Gibbs sampler generates $\boldsymbol{\phi}^{(s)}$ from $\boldsymbol{\phi}^{(s-1)}$ as follows

1. sample $\phi_1^{(s)} \sim p(\phi_1 | \phi_2 = \phi_2^{(s-1)}, \ldots, \phi_q = \phi_q^{(s-1)})$
2. sample $\phi_2^{(s)} \sim p(\phi_2 | \phi_1 = \phi_1^{(s)}, \phi_3 = \phi_3^{(s-1)}, \ldots, \phi_q = \phi_q^{(s-1)})$

...

q. sample $\phi_q^{(s)} \sim p(\phi_q | \phi_1^{(s)}, \phi_2^{(s)}, \ldots, \phi_{q-1}^{(s)})$.

After S iterations, this algorithm generates a dependent sequence of random vectors

$$\begin{aligned}\boldsymbol{\phi}^{(1)} &= \{\phi_1^{(1)}, \ldots, \phi_q^{(1)}\} \\ \boldsymbol{\phi}^{(2)} &= \{\phi_1^{(2)}, \ldots, \phi_q^{(2)}\} \\ &\ldots \\ \boldsymbol{\phi}^{(S)} &= \{\phi_1^{(S)}, \ldots, \phi_q^{(S)}\}.\end{aligned}$$

This sequence forms what we call *a Markov chain*, because the random vector $\boldsymbol{\phi}^s$ is conditionally independent of all the past instances $\boldsymbol{\phi}^{(1)}, \ldots, \boldsymbol{\phi}^{(s-2)}$, given $\boldsymbol{\phi}^{(s-1)}$. (Markov property: the future is conditionally independent of the past, given the presence). We will define Markov chains shortly in the sequel.

The main point to quickly get into is that under suitable conditions that are easily met, as $s \to \infty$, $\boldsymbol{\phi}^{(s)}$ converges in distribution to the Markov chain's *stationary* distribution $p(\phi)$. We also refer to $p(\phi)$ as the target distribution of the Markov chain (MC). In particular, for any measurable event A of interest, we may write

$$\boxed{\Pr(\boldsymbol{\phi}^{(s)} \in A) \to \Pr(\boldsymbol{\phi} \in A) \text{ as } s \to \infty.}$$

In other words, if we run the chain long enough then $\boldsymbol{\phi}^{(s)}$ can be used to approximate a sample for the joint distribution $p(\boldsymbol{\phi})$ of interest.

More importantly, take any function $g(\phi)$ for which we may be interested in the expectation under $p(\boldsymbol{\phi})$, then the following law of large numbers holds quite generally, as $S \to \infty$:

$$\boxed{\frac{1}{S}\sum_{s=1}^{S} g(\boldsymbol{\phi}^{(s)}) \to \mathbb{E}g(\boldsymbol{\phi}) = \int g(\boldsymbol{\phi})p(\boldsymbol{\phi})\mathrm{d}\boldsymbol{\phi}.} \tag{14}$$

In other words, we can apply Monte Carlo approximation technique to the Markov chain's generated samples to evaluate the expectation of interest. For this reason, we call all such approximations *Markov chain Monte Carlo* (MCMC) approximations, and the overall procedure an MCMC algorithm.

Remark 6.2.

- The good: While it is generally difficult to construct a sample for the joint distribution $p(\boldsymbol{\phi})$, it is *relatively* easier to construct a Markov chain that converges in the limit to the target $p(\boldsymbol{\phi})$.
- The advent of MCMC algorithms is the primary reason that helped to push Bayesian statistics into a central place of modern statistics, because they provide a generic mechanism for posterior computation for complex models. From a modeling standpoint, we can go beyond conjugate prior specification; from a scalability standpoint we can work with very large number of variables and parameters.
- MCMC approximation techniques are quite remarkable because they exploit the strong law of large numbers for non-i.i.d. random variables —- MC's generated samples are clearly dependent.
- Hence the bad: there are infinitely many Markov chains for the same target distribution, not all equal.
  - Some may take a long time to get close to the target stationary distribution, i.e., they have a slow *mixing* time. In such a case, to produce even approximately good sample for the target distribution, S needs to be very large (and we don't generally know how large).
  - Moreover, some Markov chain may produce strongly correlated samples, hence the Monte Carlo technique may carry very high variance. Hence, the empirical average requires a considerably larger number S of dependent samples than one would with independent Monte Carlo samples.

6.3.2 General Markov chain framework

Gibbs samplers are very easy to implement and can be applied to almost any complex statistical models. For this reason they are very popular. Its popularity is also its curse, as Gibbs sampling can be very inefficient for the reasons we've just mentioned.

Therefore, it is important to gain intuition of Gibbs sampling by placing it within a more general framework of Markov chain, so we can get a feel of what a Gibbs sampler tries to achieve, when does it "works" and when it may not. And when it does not work, what can we do. In fact, there are many variants of Gibbs sampler (we have introduced only one such variant). More importantly, there are many non-Gibbs Markov chain Monte Carlo techniques, including Metropolis-Hastings, Hamiltonian MCMC, and so on.

Bear with me a bit of formalism in the next couple of pages. The payoff is worth it. [4]

[4] I largely follow Charles Geyer (2005) for the rest of this subsection.

Definition 6.1. A Markov chain is a discrete time stochastic process $\boldsymbol{\phi}^{(1)}, \boldsymbol{\phi}^{(2)}, \ldots$ taking values in an arbitrary state space $\mathcal{S}$, having the property that the conditional distribution of $\boldsymbol{\phi}^{(s+1)}$ given the past $\boldsymbol{\phi}^{(1)}, \ldots, \boldsymbol{\phi}^{(s)}$ depends only on the present state $\boldsymbol{\phi}^{(s)}$.

$\boldsymbol{\phi}^{(s)}$ is called the state variable at time s. A Markov chain is defined by its *transition probabilities*.

- For discrete state space $\mathcal{S}$, these are specified by defining a matrix p:

$$p(x, y) := \Pr(\boldsymbol{\phi}^{(s+1)} = y | \boldsymbol{\phi}^{(s)} = x), \quad x, y \in \mathcal{S}$$

  that gives the probability of moving from any element $x \in \mathcal{S}$ at time s to any element $y \in \mathcal{S}$ at time $s + 1$. The transition probability matrix $p(x, y)$ does not depend on time s.

- For continuous state space S, the proper way to think of the transition probabilities is via a notion of *kernel* P, which can be represented by a regular conditional probability: for any measurable subset $A \subset \mathcal{S}$, the kernel P is given as

$$P(x, A) := \Pr(\boldsymbol{\phi}^{(s+1)} \in A | \boldsymbol{\phi}^{(s)} = x).$$

  Kernel $P(x, A)$ is defined by two arguments, x is an element in the state space $\mathcal{S}$ and A a subset of $\mathcal{S}$. It gives the probability of moving from an element $x \in \mathcal{S}$ into a subset A at time $s + 1$.

Note that the transition probabilities do not by themselves define the probability distribution of the Markov chain. To do so, we need to additionally specify the initial distribution of the chain, namely, the marginal distribution for $\boldsymbol{\phi}^{(1)}$.

A key concept of a Markov chain is

Definition 6.2. A probability distribution π is a *stationary* distribution or an *invariant* distribution for the Markov chain if it is "preserved" by the transition probability. That is if the initial distribution is π, then the marginal of $\boldsymbol{\phi}^{(2)}$ is also π. Hence, so is the marginal distribution of $\boldsymbol{\phi}^{(3)}$ and all the rest of the chain.

- For discrete state space $\mathcal{S}$, π is specified by a vector $\pi(x)$, and the stationary property is

$$\pi(y) = \sum_{x \in \mathcal{S}} \pi(x) p(x, y). \tag{15}$$

  If we think of transition probabilities as a matrix P with entries $p(x, y)$, Eq. (15) can be written as $\pi = \pi P$, where the RHS is the multiplication of the matrix P on the left by the row vector π.

- For continuous state space $\mathcal{S}$, the stationary property is

$$\pi(A) = \int_{\mathcal{S}} \pi(\mathrm{d}x) P(x, A). \tag{16}$$

  Eqs.(15) and (16) are the same except that a sum over a discrete state space has been replaced by an integral over a continuous state space.

In MCMC we often construct a Markov chain with a specified stationary distribution π in mind, so there is never a question whether a stationary distribution exists —- it does so by construction. Moreover it is unique under easily met conditions and more importantly it admits the law of large numbers, described earlier in Eq. (14).

6.3.3 Variants of Gibbs samplers

With the general Markov chain framework in mind, we can see that the Gibbs sampler is a very simple construction of a Markov chain of state space variable represented by vector $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_q)$ taking value in $\mathcal{S}$ for some $q \geq 2$.

The Gibbs sampler is composed of elementary update steps, which we call *Gibbs update*: an elementary Gibbs update changes only one component of the state vector, say ϕ_i for some $i = 1, \ldots, q$. This component is given a new value which is a sample from its "full conditional" — its conditional distribution given the rest $\pi(\phi_i|\phi_{-i})$, where $\phi_{-i} := (\phi_1, \ldots, \phi_{i-1}, \phi_{i+1}, \ldots, \phi_q)$.

It is easy to verify that the elementary Gibbs update preserves the stationary distribution: if the current state ϕ is a realization from π, then ϕ_{-i} is distributed according to its marginal $\pi(\phi_{-i})$ derived from π, and the state after the update will have the distribution

$$\pi(\phi_i|\phi_{-i})\pi(\phi_{-i})$$

which is $\pi(\boldsymbol{\phi})$ by definition of conditional probability: joint equals conditional times marginal.

We can represent an elementary Gibbs update for component i by a kernel denoted by P_i, for $i = 1, \ldots, q$.

Furthermore, a composition of an elementary Gibbs update, say P_1 followed by an elementary Gibbs update, say P_2 can be represented by the composite kernel $P_1 P_2$. It has a concrete meaning:

- For a discrete state space $\mathcal{S}$, $P_1 P_2$ represents the multiplication of two transition probability matrices. The result is a matrix with entries
$$\sum_{y \in \mathcal{S}} p_1(x, y) p_2(y, z).$$
- For a continuous state space $\mathcal{S}$, we need to replace the sum by the integral:
$$(P_1 P_2)(x, A) = \int P_1(x, dy) Q(y, A).$$

Composition of kernels Now we can write the first Gibbs sampling introduced in subsection 6.3.1 as the construction of a Markov chain using the kernel

$$P = P_1 P_2 \dots P_q$$

In words: this Markov chain is constructed by first updating ϕ_1 via its full conditional, and then ϕ_2, ..., until ϕ_q. The compositions of the q elementary Gibbs update result in the kernel P. And application of P allows us to generate a Markov chain sample $\boldsymbol{\phi}^{s+1}$ if we are to start from $\boldsymbol{\phi}^s$.

It is easy to verify that the composition of kernels this way preserves the stationary distribution: $\pi(P_1 P_2 P_3) = ((\pi P_1)P_2)P_3 = (\pi P_2)P_3 = \pi P_3 = \pi$, and so on.

Mixing kernels But we can also create new Markov chains from the elementary Gibbs update by mixing:

$$P = \frac{1}{q} \sum_{i=1}^{q} P_i.$$

In words: pick a coordinate i to update with equal probabilities $1/q$. Then update ϕ_i according to kernel P_i.

There is no reason to stay with equality probabilities: take any weights $(\alpha_1, \dots, \alpha_q) \in \Delta^{q-1}$. Pick coordinate i to update with probability α_i. If i is chosen, then update ϕ_i according to kernel P_i.

Combining composition and mixing We can combine the composition and mixing tricks. The best known example of this is the so-called *random sequence scan* that combines q elementary update mechanisms by choosing a random permutation $(i_1, i_2, \dots, i_q)$ of the integers $1, 2, \dots, q$ and then appplying the updates $P_{i_j}, j = 1, \dots, q$ in that order. If $\mathscr{P}$ denotes the set of all $q!$ permutations, the kernel of this scan is

$$P = \frac{1}{q!} \sum_{(i_1,\dots,i_q)\in\mathscr{P}} P_{i_1} \dots P_{i_q}.$$

6.4 MCMC diagnostics

Now with so many Gibbs variants (and in the future non-Gibbs Markov chains) available to consider, how can we tell which one works, and works better? Remember that all Gibbs samplers and MCMC algorithms in general work in theory, if we were allowed to run the Markov chain until infinity. But we can never do that in practice. We may come up with one or several Markov chain constructions, run them for a while and evaluate. This requires techniques for assessing the effectiveness of MCMC algorithms. This section provides a brief introduction into MCMC diagnostics.

The goal of Monte Carlo or Markov chain Monte Carlo approximation is to obtain a sequence of parameter values $\{\phi^{(1)}, \ldots, \phi^{(S)}\}$ such that, for some function g of interest and a target distribution $p(\phi)$,

$$\frac{1}{S}\sum_{s=1}^{S} g(\phi^{(s)}) \approx \int g(\phi)p(\phi)\mathrm{d}\phi.$$

In order to obtain a good approximation, there are primarily two main issues that we need to worry about

(i) the empirical distribution of the simulated sequence $\{\phi^{(1)}, \ldots \phi^{(S)}\}$ need to approximate well the target distribution $p(\phi)$.

(ii) the members of the simulated sequence need to be as weakly correlated as possible (zero correlation is the best).

Standard Monte Carlo samples represent the "gold standard", *if* they could be obtained: by assumption, the MC samples are identically and independently distributed according to the target $p(\phi)$. Thus, both criteria (i) and (ii) are perfectly achieved. Let $\bar{\phi}$ denote the empirical average of the Monte Carlo samples of ϕ, assuming for the moment to be scalar, then the variance of this Monte Carlo approximate is

$$\mathrm{Var}_{\mathrm{MC}}[\bar{\phi}] = \frac{1}{S}\mathrm{Var}[\phi]. \tag{17}$$

For samples simulated by a Markov chain, the aforementioned issues are generally non-trivial to address. The Markov chain may take a long time to get close to the target stationary distribution, requiring S to be large for (i) to be achieved. Moreover, there may be strong correlations among simulated samples $\{\phi^{(s)}\}_{s=1}^{S}$, resulting in difficulty in achieving (ii).

Example 6.1.

Consider the target distribution of the form

$$p(\theta) = \sum_{k=1}^{3} p_i \times \text{Normal}(\theta|\mu_i, \sigma_i^2),$$

where

$$\boldsymbol{p} = (p_1, p_2, p_3) = (.45, .10, .45); \quad (\mu_1, \mu_2, \mu_3) = (-3, 0, 3); \quad (\sigma_1^2, \sigma_2^2, \sigma_3^2) = (1/3, 1/3, 1/3).$$

This is a mixture of three normal densities.

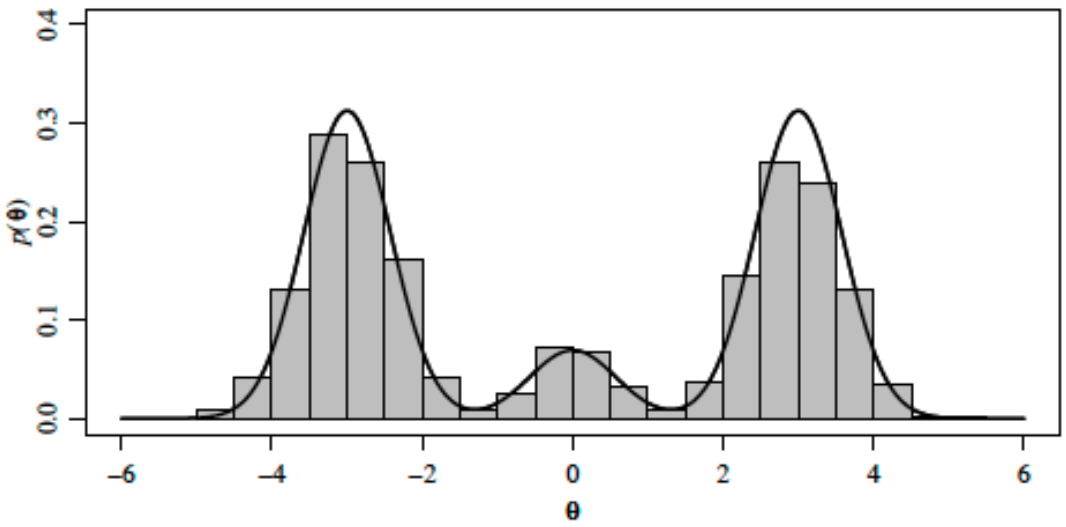

Figure 6.3: A mixture of normal densities and a Monte Carlo approximation.

A useful technique is not to draw samples for θ directly, but to add an auxiliary random variable Z such that the joint distribution for (Z, θ) induces marginal distribution for θ which is equal to the target distribution $p(\theta)$. We will then draw sample for the joint sample (Z, θ).

The joint distribution for (Z, θ) is given as follows

$$\begin{aligned} Z &\sim \text{Categorical}(\boldsymbol{p}) \\ \theta | Z = k &\sim \text{Normal}(\mu_k, \sigma_k^2). \end{aligned} \tag{18}$$

For a Gibbs sampler of (Z, θ), the full conditional for θ is already given by Eq. (18). The full conditional for Z is given by, via Bayes' rule:

$$\Pr(Z = k | \theta) = \frac{p_k \text{Normal}(\theta | \mu_k, \sigma_k)}{\sum_{j=1}^{3} p_j \text{Normal}(\theta | \mu_j, \sigma_j)}. \tag{19}$$

Fig. 6.4 illustartes the histogram and traceplot of the first 1,000 Gibbs samples.

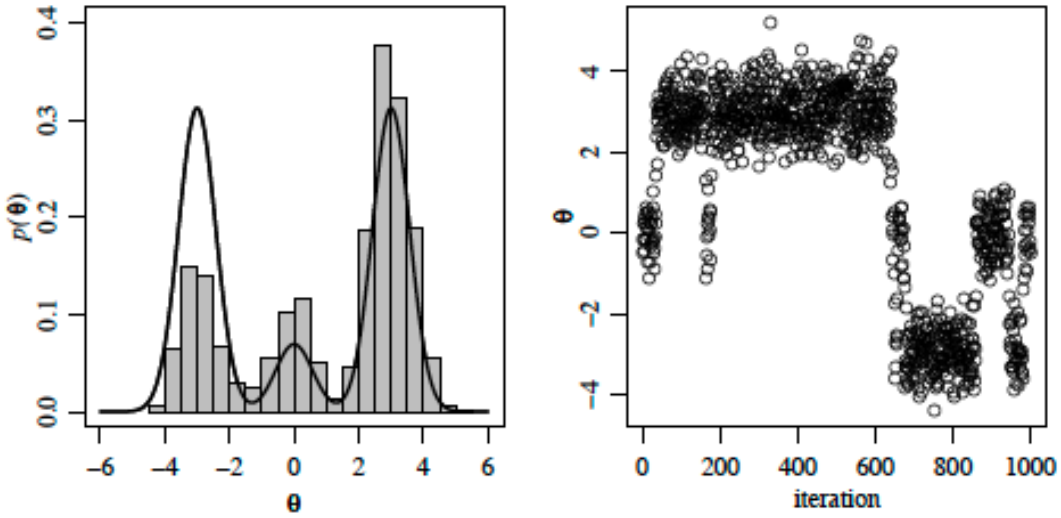

Figure 6.4: Histogram and traceplot of 1,000 Gibbs samples.

What do we see:

- For the Gibbs sampler for θ-values starts in the region corresponding to the second mode (from the left) of the distribution, then ventures to the region corresponding to the first mode, and get "stuck" there for a quite long time. It manages to get out of the second mode, passing through it, and transition to the region corresponding to the third mode. Nonetheless, it doesn't seem to spend "enough" time there before transitioning back to the second mode again.
- As shown by the first panel of Fig. 6.4, the Markov chain is not close to the stationary target distribution $p(\theta)$. It has not mixed after 1,000 iterations. If we run considerably longer, for 10,000 iterations, the mixing is considerably improved. See Fig. 6.5.
- The "stickiness" of the Markov chain at regions corresponding to the three modes, especially the first and third mode suggests strong correlation among the simulated samples.

□

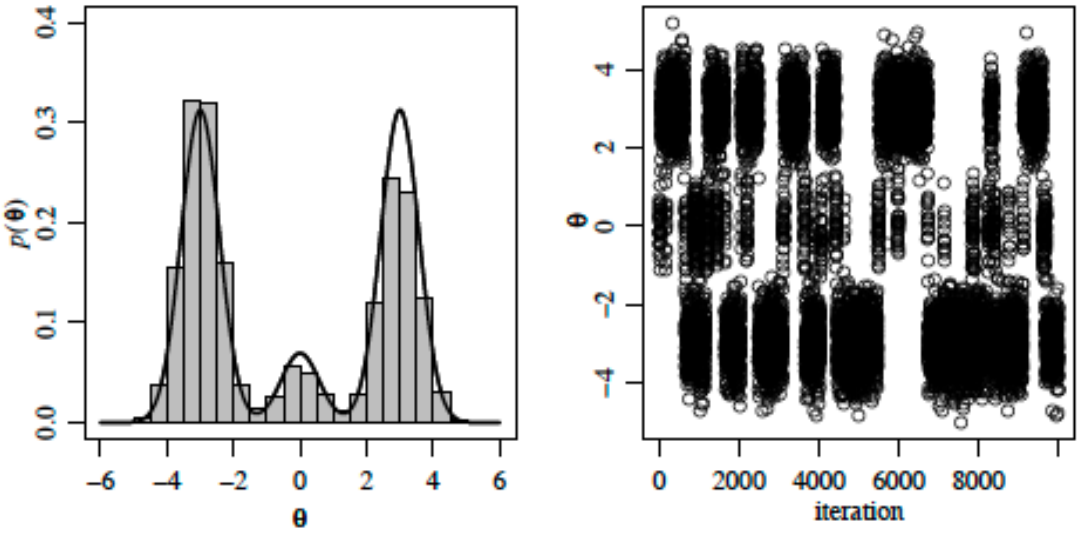

Figure 6.5: Histogram and traceplot of 10,000 Gibbs samples.

How do we verify both issues of mixing and strong correlation of Markov chain samples?

To verify mixing is difficult in theory. This is an active area of research, where researchers work on upper and lower bounds of the mixing time. Unfortunately, for complex models, tight bounds for the mixing time are rarely available. In practice, a standard method is to run multiple Markov chains (starting at different positions), and compare the distributions for the variables of interest. This works well when the number of variables of interest is not too large. For high-dimensional state spaces, having a robust way to verify the mixing of Markov chain remains a big challenge.

The reason we want to check the correlation of Markov chain samples — the technical term is *autocorrelation* — is that this quantity affects to the variance of the Monte Carlo estimate in a crucial way.

Assume that that stationarity of the Markov chain has been achieved. Let ϕ_0 be the expectation of a scalar ϕ under the stationary target distribution. The variance of the Monte Carlo estimate $\bar{\phi} := \frac{1}{S}\sum_s \phi^{(s)}$ can be computed as follows

$$\begin{aligned}
\mathrm{Var}_{\mathrm{MCMC}}(\bar{\phi}) &:= \mathbb{E}(\bar{\phi} - \phi_0)^2 \\
&= \mathbb{E}\left[\left(\frac{1}{S}\sum_{s=1}^{S}\phi^{(s)} - \phi_0\right)^2\right] \\
&= \frac{1}{S^2}\mathbb{E}\left[\sum_s(\phi^{(s)} - \phi_0)^2 + \sum_{s\neq t}(\phi^{(s)} - \phi_0)(\phi^{(t)} - \phi_0)\right] \\
&= \mathrm{Var}_{\mathrm{MC}}(\bar{\phi}) + \frac{1}{S^2}\mathbb{E}\left[\sum_{s\neq t}(\phi^{(s)} - \phi_0)(\phi^{(t)} - \phi_0)\right].
\end{aligned}$$

Thus, the MCMC variance is equal to the MC variance plus a term that depends on the correlation of samples within the Markov chain. This term is usually positive, so the MCMC variance is usually higher than the MC variance.

To assess how much correlation there is in the chain, we compute the *sample autocorrelation function*: for a generic sequence of numbers $\{\phi_1, \ldots, \phi_S\}$, the lag-$t$ autocorrelation function estimates the correlation between elements of the sequence that are t steps apart:

$$\text{acf}_t(\boldsymbol{\phi}) = \frac{\frac{1}{S-t}\sum_{s=1}^{S-t}(\phi_s - \bar{\phi})(\phi_{s+t} - \bar{\phi})}{\frac{1}{S-1}\sum_{s=1}^{S}(\phi_s - \bar{\phi})^2}. \tag{20}$$

In `R`, this quantity is computed by `R`-function `acf`. If we are close to stationarity, this quantity is almost always between [-1,1]. Being close to 1 means strong positive correlation. Being close to zero means small correlation. For the example in Fig. 6.5, for the sequence of 10K Gibbs samples for θ-values, the lag-10 autocorrelation is 0.93, and lag-50 autocorrelation is 0.812. This means that the Markov chain has very high correlation. Such a Markov chain explores the parameter space slowly, taking a long time to mix, and the empirical average also has a high variance.

A practically useful way is to consider the *effective sample size* of a Markov chain. Motivated by the Monte Carlo variance formula (see Eq. (17)), the MCMC effective sample size S_{eff} is the value such that

$$\text{Var}_{\text{MCMC}}(\bar{\phi}) = \frac{\text{Var}\,\phi}{S_{\text{eff}}}. \tag{21}$$

In `R`, this quantity is estimated by the `R`-command `effectiveSize`.

In the example of normal mixture discussed above, the effective sample size of the 10,000 Gibbs samples of θ is 18.42, indicating that the precision of the MCMC approximation to $\mathbb{E}[\theta]$ is as good as the precision that would have been obtained by utilizing only about 18 i.i.d. samples of θ. This may suggest two possible courses of action: either run the Gibbs sampler considerably longer, or design a better Markov chain.

10 Metropolis-Hasting algorithms

The Gibbs sampler constructs a Markov chain, whose transition probability kernel is defined as a composition of multiple Gibbs updates. A Gibbs update changes one variable at a time. This can be inefficient. Moreover, the Gibbs update often requires some sort of (semi) conjugacy in the model, so that the full conditional distributions can be computed in close form.

In this section we shall study a more general MC based sampling method known as Metropolis-Hastings algorithm. Most MCMC based algorithms in practice, including Gibbs sampling, are special cases of Metropolis-Hastings algorithm, which is versatile and powerful. M-H is especially useful in the non-conjugacy situation, and when there is a need and possibility of updating multiple variables simultaneously.

10.1 Metropolis-Hastings update

Let π be the (stationary) distribution of interest. Suppose that π is known only up to an unknown constant. That is, π is specified by an *unnormalized* density function $h(x)$ with respect to a counting measure on a discrete space S or Lebesgue measure $\mu(dx)$ with respect to an Euclidean space S. Write

$$\pi(x) = h(x)/c$$

where the normalizing constant $c = \int h(x)\mu(dx) < \infty$ is unknown. In Bayesian computation, $h(x)$ is often the product of the prior density and the likelihood function.

Proposal distribution The M-H update uses an auxiliary transition probability specified by a conditional density function $q(x, y)$. It's called "proposal distribution", or "candidate generating distribution". For every point $x \in \mathcal{S}$, $q(x, \cdot)$ is the probability density (wrt μ) having two properties

- for each x we can sample a random variable y having the density $q(x, \cdot)$
- we can evaluate $q(x, y)$ for each $x, y \in \mathcal{S}$

Roughly speaking, $q(x, y)$ represents the conditional probability "proposing" an update value y, given that we are presently at x. We can choose any density we know to propose. For instance, if $\mathcal{S} = \mathbb{R}^d$, a random walk proposal corresponds to $q(x, y) = \mathrm{N}_d(y|x, \sigma^2 I)$, a density function evaluated at $y \in \mathbb{R}^d$ of a d-variate normal density with mean $x \in \mathbb{R}^d$ and variance $\sigma^2 I$.

The Metropolis-Hastings algorithm then works by constructing the Markov chain $\{X_t\}_{t\geq 1}$ as follows. Start $X_0 = x$ where x is in the support of h, i.e., $h(x) > 0$. Given the current position $X_t = x \in \mathcal{S}$, the update changes x to its value at the next iteration.

1. Draw a sample $y \sim q(x, \cdot)$.

2. Calculate the Hastings ratio:
$$R = \frac{h(y)q(y,x)}{h(x)q(x,y)}. \tag{59}$$

3. Accept the proposal by setting $X_{t+1} = y$ with probability $\min(1, R)$. Otherwise, keep the position unchanged by setting $X_{t+1} = x$.

Example 10.1. (Metropolis update) If we use a proposal density $q(x, y)$ that is symmetric: $q(x, y) = q(y, x)$. For instance, the "normal random walk" $q(x, y) = \mathrm{N}_d(y|x, \sigma^2 I)$. Then, Hastings ratio takes the form

$$R = h(y)/h(x).$$

There is no need to evaluate $q(x, y)$.

Metropolis algorithm is very popular, because it is easy to implement. It is also very intuitive: as long as one takes a symmetric proposal, then we always accept the proposed move from x to y if this represents an increase in the density of the stationary distribution, i.e., $\pi(y) \geq \pi(x)$. If the move represents a decrease, then the larger the decrease the less likely one will accept the move. □

Let us write down the transition probability kernel $P(x, A)$ for the general Metropolis-Hastings update, for any $x \in \mathcal{S}, A \subset \mathcal{S}$. The kernel has two terms related to accepted proposals and rejected one. For accepted proposals, we propose y and then accept it, which happens with probability density

$$p(x, y) = q(x, y)a(x, y),$$

where $a(x, y) = \min(R, 1)$. Thus

$$\int_A p(x, y)\mu(\mathrm{d}y)$$

represents the part of $P(x, A)$ that results from the accepted proposals. Moreover, $\int_{\mathcal{S}} p(x, y)\mu(\mathrm{d}y)$ gives the total probability that some proposed move is accepted (including the possibility that $y = x$) while

$$r(x) := 1 - \int_{\mathcal{S}} p(x, y)\mu(\mathrm{d}y)$$

is the probability a proposed move is rejected. If the proposed move is rejected, we stay put at x.

Thus, the probability of moving from x to a measurable subset $A \subset \mathcal{S}$ is

$$P(x, A) = \int_A p(x, y)\mu(\mathrm{d}y) + \left(1 - \int_{\mathcal{S}} p(x, y)\mu(\mathrm{d}y)\right) I(x, A). \tag{60}$$

In the above, $I(x, A)$ denotes *identity* kernel that represents "stay put": $I(x, A) = 1$ if $x \in A$ and 0 otherwise.

10.1.1 Detailed balance and reversibility

Definition 10.1. A Markov chain $\{X_t\}_{t\geq 0}$ with a stationary distribution π is said to be reversible if when X_t has the distribution π, then X_t and X_{t+1} are exchangeable random variables.

Recall that if π is called a stationary distribution of the Markov chain if the following holds: when X_t has distribution π, then so is X_{t+1}. Thus, exchangeability is a stronger condition, as we have learned earlier in Section 8.3: it requires that the ordered pair (X_t, X_{t+1}) has the same joint distribution as the ordered pair (X_{t+1}, X_t). (Exercise: verify that a basic Gibbs update is reversible).

Although reversibility is not a requirement, many MC constructions have this property. While reversibility has some theoretical benefits for the analysis of a MC; for us it is enough to note that reversibility is a useful property in that one automatically have the guarantee that a Markov chain construction admits π as stationary distribution by checking that it satisfies the stronger condition of reversibility, which tends to be easy to do in practice.

Recall $p(x,y) = q(x,y)a(x,y)$. The key to verify reversibility is to check that the Markov chain satisfies the *detailed balance*. That is:

$$h(x)p(x,y) = h(y)p(y,x), \quad \text{for all} \quad x, y \in \mathcal{S}. \tag{61}$$

Note that this is also equivalent to $\pi(x)p(x,y) = \pi(y)p(y,x)$.

Suppose that the detailed balance holds. Then for any $A, B \subset \mathcal{S}$, we have

$$\begin{aligned}
& \Pr(X_t \in A, X_{t+1} \in B) \\
= \; & \int\int 1_A(x) 1_B(y) \pi(x) P(x, \mathrm{d}y) \mu(\mathrm{d}x) \\
= \; & \int\int 1_A(x) 1_B(y) \pi(x) \Big(p(x,y) + r(x) 1(y = x) \Big) \mu(\mathrm{d}y) \mu(\mathrm{d}x) \\
= \; & \int\int 1_A(x) 1_B(y) \pi(x) p(x,y) \mu(\mathrm{d}y) \mu(\mathrm{d}x) + \int\int 1_A(x) 1_B(y) 1(y = x) r(x) \pi(x) \mu(\mathrm{d}y) \mu(\mathrm{d}x) \\
\overset{(61)}{=} \; & \int\int 1_A(x) 1_B(y) \pi(y) p(y,x) \mu(\mathrm{d}y) \mu(\mathrm{d}x) + \int\int 1_A(x) 1_B(y) 1(x = y) r(y) \pi(y) \mu(\mathrm{d}y) \mu(\mathrm{d}x) \\
= \; & \int\int 1_A(x) 1_B(y) \pi(y) \Big(p(y,x) + r(y) 1(x = y) \Big) \mu(\mathrm{d}y) \mu(\mathrm{d}x) \\
= \; & \Pr(X_t \in B, X_{t+1} \in A),
\end{aligned}$$

which confirms reversibility.

Reversibility of Metropolis-Hastings update Now we can verify that the M-H update is reversible by checking the detailed balance condition. But this is immediate

$$\begin{aligned} h(x)p(x,y) &= h(x)q(x,y)a(x,y) \\ &= h(x)q(x,y)\min\left\{1, \frac{h(y)q(y,x)}{h(x)q(x,y)}\right\} \\ &= \min\left\{h(x)q(x,y), h(y)q(y,x)\right\}. \end{aligned}$$

The last expression in the above display is symmetric with respect to x and y, so it is also equal to $h(y)p(y,x)$. We are done with the verification.

Metropolis-Hastings update for a subset of variables Although the above description is for the full set of variables x (e.g., $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, Metropolis-Hastings can and quite typically be applied to a subset of variables (just like the Gibbs sampler, which can also be applied to subset of variables). Suppose that a subset of variables $x_1, \ldots, x_j$ are to be updated for some $j < d$, then the proposal density $q((x_1, \ldots, x_j), (y_1, \ldots, y_j))$ should be taken as the density with respect to the base measure on the subspace $\mathbb{R}^j$ spanned by the j variables being updated. The procedure is then applied as described.

Gibbs as a special case of Metropolis-Hastings The Gibbs sampler updates a variable x_i from its full conditional distribution of x_i given all remaining variables x_{-i}. We will show that a Gibbs update for variable x_i is nothing but a Metropolis-Hastings with the proposal distribution $\pi(x_i|x_{-i})$.

Indeed, for variable x_i, take the proposal density to be

$$q(x, y) \propto h(x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_d)/c$$

where $y_j = x_j$ for $j \neq i$, and h is the unnormalized density function for the target stationary distribution π. Note that $q(x, y)$ so defined is exactly the full conditional distribution $\pi(x_i = y_i|x_{-i})$. Then, the Hastings ratio is

$$\begin{aligned} R = \frac{h(y)q(y,x)}{h(x)q(x,y)} &= \frac{h(y)h(y_1, \ldots, y_{i-1}, x_i, y_{i+1}, \ldots, y_d)}{h(x)h(x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_d)} \\ &= \frac{h(y)h(x)}{h(x)h(y)} = 1. \end{aligned}$$

It follows that the acceptance probability is $\min\{R, 1\} = 1$. Thus, by adopting the full conditional distribution as the proposal distribution, the Metropolis-Hasting proposal is always accepted. This is exactly the Gibbs update!

Remark 10.1. • The Metropolis-Hastings framework is so general and powerful that its introduction dramatically opened up the landscape of possibilities for MCMC based inference, because one can in principle adopt any reasonable distribution as a proposal, and still get a valid Markov chain for a target stationary distribution of interest. Ideally, we would like a proposal that allows one to explore efficiently the distribution, by spending proportionally more time in all high density regions.

- Metropolis and Gibbs samplers can be viewed as two extremes in this landscape of proposals. Metropolis is realized by applying an arbitrary symmetric proposal distribution — this allows the Markov chain to explore virtually any location in the state space as one likes. The price to pay is that the acceptance rate may be very small, if the proposal is too "reckless", as it may have nothing to do with the actually concentration of mass of the target distribution. When this is the case, one ends up rejecting the proposals most of the time, which amounts to a frustrating hit-and-miss sampling experience.

- Gibbs sampling, on the other hand, is too cautious in its proposal, which is automatically determined by the induced full conditional distributions. Although all its moves are accepted, the movements through the space of support can be hopelessly slow: due to the requirement of conjugacy needed for the computation of the full conditionals, one may update only one variable or a small subset of variables at a time and get stuck in local modes as a result.

- Finding a good proposal for a given posterior distribution is an active area of research. It requires a deeper understanding of the geometry of such a posterior distribution. Hamiltonian Monte Carlo Markov chains represent such one such promising approach, but the progress remains rudimentary at this point.

- In practice, one may mix and match between different proposal strategies. For instance, one may mixing up Gibbs updates for some subsets of variables with Metropolis-Hasting updates for other subsets.

10.2 Example

Poisson regression model Given a population of song sparrows, we are interested in learning about the relationship about the number of offsprings versus age. [13]

An approach is to consider a regression model: the response y represents the number of offspring of a song sparrow, while the regressors may be constructed of age variable x.

For instance, we assume $\log \mathbb{E}[Y|x] = \beta_1 + \beta_2 x + \beta_3 x^2$. This means $\mathbb{E}[Y|x] = \exp(\beta_1 + \beta_2 x + \beta_3 x^2)$.

Since Y is positive integer-valued, we may consider Poisson distribution as the conditional distribution for Y given x. The resulting model is called a Poisson regression model:

$$Y|\boldsymbol{x} \sim \text{Poisson}(\exp(\boldsymbol{\beta}^\top \boldsymbol{x})).$$

To complete the prior specification: we may endow $\boldsymbol{\beta}$ with a normal prior. Note immediately that this is not conjugate to the Poisson-type likelihood. In general, Poisson regression is a specific instance of a broad class of models known as *generalized linear model*, for which conjugate priors generally don't exist. Thus, Gibbs sampling is difficult to implement.

[13] Details of this example and more examples can be found in chapter 10 of Hoff [2009].

Let's consider Metropolis sampling. Provided a normal prior for $\boldsymbol{\beta}$: $\boldsymbol{\beta} \sim \mathrm{N}(\boldsymbol{\beta}_0, \Sigma_0)$. Given n-sample $(y_i, \boldsymbol{x}_i)_{i=1}^n$. The Hastings acceptance ratio is easy to compute: given the current $\boldsymbol{\beta}^{(s)}$ and a proposed $\boldsymbol{\beta}^*$,

$$\begin{aligned} R &= \frac{p(\boldsymbol{\beta}^*|\boldsymbol{X}, \boldsymbol{y})}{p(\boldsymbol{\beta}^{(s)}|\boldsymbol{X}, \boldsymbol{y})} \\ &= \frac{\text{normal}(\boldsymbol{\beta}^*|\boldsymbol{\beta}_0, \Sigma_0)}{\text{normal}(\boldsymbol{\beta}^{(s)}|\boldsymbol{\beta}_0, \Sigma_0)} \times \frac{\prod_{i=1}^n \text{poisson}(y_i, \boldsymbol{x}_i^\top \boldsymbol{\beta}^*)}{\prod_{i=1}^n \text{poisson}(y_i, \boldsymbol{x}_i^\top \boldsymbol{\beta}^{(s)})}. \end{aligned}$$

This ratio is easy to compute. In practice, when n is large, the ratio may be either too large or too small. To avoid numerical issues, it is advised to compute the logarithm of the ratio R instead of computing R directly. Then, the acceptance probability is

$$a(\boldsymbol{\beta}^{(s)}, \boldsymbol{\beta}^*) = e^{\min\{0, \log R\}}.$$

It remains to specify the proposal distribution for $\boldsymbol{\beta}^*$. A natural choice is to take a normal random walk, i.e., via a normal distribution centered at $\boldsymbol{\beta}^{(s)}$:

$$q(\boldsymbol{\beta}^{(s)}, \boldsymbol{\beta}^*) = \text{normal}(\boldsymbol{\beta}^* | \boldsymbol{\beta}^{(s)}, \Sigma).$$

How do we choose Σ? In a normal regression problem, the posterior variance of $\boldsymbol{\beta}$ will be close to $\sigma^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1}$, where σ^2 is the variance of Y. This gives us a hint for our Poisson regression problem: since $\log Y$ is taken to have expectation $\boldsymbol{\beta}^\top \boldsymbol{x}$, we can take the proposal variance to be

$$\Sigma := \hat{\sigma}^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1}$$

where $\hat{\sigma}$ is the sample variance of $\{\log(y_1 + 1/2), \ldots, \log(y_n + 1/2)\}$. (The addition of 1/2 is so the log can be applied to a positive valued number).

We can also consider other choice for Σ, or $q(\boldsymbol{\beta}^{(s)}, \boldsymbol{\beta}^*)$. For the chosen form of Σ above, we may also choose different $\hat{\sigma}$. All such choices result in a valid Markov chain, but they can have different mixing qualities and autocorrelation of the MC samples.

The general rule of thumb is to specify a proposal so that the acceptance rate is neither too large nor too small (say, between 20 and 50%).

The materials in this course are adapted from materials created by David Blei, Yang Chen, Andrew Gelman, Scott Linderman, Long Nguyen, and the 3blue1brown channel.