STA5904: Bayesian Modeling and Inference **Topic 2**

Bayesian Computation

1 The Problem

We take a slight detour in material so that we can cover, early in the course, how one actually performs Bayesian inference in practice. Suppose we have a model $\{f_\theta : \theta \in \Theta\}$ for data Y and a prior distribution $\pi(\theta)$ for the parameter of interest θ. The posterior density of θ is given according to Bayes rule,

$$\pi(\theta \mid y) = \frac{f_\theta(y)\,\pi(\theta)}{\int f_\theta(y)\,\pi(\theta)\;d\theta}. \tag{1}$$

There are several things one might wish to do with this posterior distribution; for example we might want to compute

1. the *Bayes estimator* of θ, $E(\theta \mid Y)$;
2. a $100(1-\alpha)\%$ *credible interval* for θ, which satisfies $P(L \leq \theta \leq U \mid Y) = 1-\alpha$;
3. a measure of uncertainty for θ, such as $\mathrm{Var}(\theta \mid Y)$.

These quantities are difficult to compute in general, for a number of reasons. For example, the Bayes estimator is

$$E(\theta \mid Y = y) = \frac{\int \theta\, f_\theta(y)\,\pi(\theta)\;d\theta}{\int f_\theta(y)\,\pi(\theta)\;d\theta}.$$

Most of the time, we will be able to compute neither the numerator nor the denominator in closed form. For many years, this was one of the primary drawbacks to performing Bayesian inference in practice.

2 Conjugate Priors

Occasionally, it will be the case that the posterior distribution $\pi(\theta \mid y)$ is actually easy to work with. For example, this was the case when we analyzed Laplace's birth records data — starting from a Uniform$(0,1)$ prior, we ended up with a Beta$(y+1, n-y+1)$ posterior distribution. Because the Beta(α,β) distribution is easy to work with, we did not have any issues performing inference (ignoring the fact that we used a computer to get the tail probabilities). This problem is easy because we are used a *conjugate prior* for the parameter θ.

Definition 1. Consider a model $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$. A family of distribution Π is said to be *conjugate* to $\mathcal{F}$ if

$$\pi(\cdot) \in \Pi \qquad \text{implies} \qquad \pi(\cdot \mid y) \in \Pi,$$

for all possible realizations of y.

Example 2. Consider the binomial likelihood $f_\theta(y) = \binom{n}{y}\theta^y(1-\theta)^{n-y}$. The family of Beta$(\alpha, \beta)$ priors for θ is conjugate to this likelihood; note that

$$\pi(\theta \mid y) \propto \theta^y(1-\theta)^{n-y} \cdot \theta^{\alpha-1}(1-\theta)^{\beta-1} = \theta^{\alpha+y-1}(1-\theta)^{\beta+n-y-1}.$$

This is the kernel of a Beta$(\alpha+y, \beta+n-y)$ distributions. Hence, if $\pi(\theta)$ is a Beta distribution, so is $\pi(\theta \mid y)$. Noting that the Uniform$(0,1)$ distribution is also a Beta$(1,1)$ prior, this reproduces Laplace's solution to the birth records problem.

Conjugacy is somewhat rare; useful instances of conjugate priors are usually limited to exponential families.

Exercise 1. Show the following.

(a) If $[Y_1, \ldots, Y_n \mid \lambda] \overset{\text{iid}}{\sim} \text{Poisson}(\lambda)$, and $\lambda \sim \text{Gam}(\alpha, \beta)$, then $[\lambda \mid Y_1, \ldots, Y_n] \sim \text{Gam}(\alpha + \sum_i Y_i, \beta + n)$.

(b) If $[Y_1, \ldots, Y_n \mid \beta] \overset{\text{iid}}{\sim} \text{Gam}(\alpha, \beta)$, and $\beta \sim \text{Gam}(a, b)$, then $[\beta \mid Y_1, \ldots, Y_n] \sim \text{Gam}(N\alpha + a, \sum_i Y_i + b)$.

(c) If $[Y_1, \ldots, Y_n \mid \mu, \tau] \overset{\text{iid}}{\sim} \text{Normal}(\mu, \tau^{-1})$, $[\mu \mid \tau] \sim \text{Normal}(m, \tau^{-1}\kappa^{-1})$, and $\tau \sim \text{Gam}(\alpha, \beta)$, then

$$[\mu \mid Y_1, \ldots, Y_n, \tau] \sim \text{Normal}(\widehat{\mu}, \tau^{-1}\widehat{\kappa}^{-1}) \quad \text{and} \quad [\tau \mid Y_1, \ldots, Y_n] \sim \text{Gam}(\widehat{\alpha}, \widehat{\beta}),$$

where

$$\widehat{\mu} = \frac{\sum_{i=1}^n Y_i + \kappa m}{n + \kappa}, \qquad \widehat{\kappa} = \kappa + n,$$

$$\widehat{\alpha} = \alpha + \frac{n}{2}, \qquad \widehat{\beta} = \beta + \frac{1}{2}\sum_i (Y_i - \bar{Y})^2 + \frac{n\kappa(\bar{Y} - m)^2}{2(n+\kappa)}.$$

Example 3. We borrow an example from Gelman et al. regarding point spreads in football. In football, experts provide a *point spread*, which states how much the home team will win (or lose) by. Individuals can then gamble against the point spread. An accurate scheme for constructing point spreads, then, should be close to the actual point spread of a given game.

We have data on 672 professional NFL football games, as well as the point spread constructed by experts before the game. Let S_i denote the predicted point spread for each game, and X_i denote the actual point spread from the game, and let $Y_i = S_i - X_i$ denote the deviation of the predicted from the observed point spread.

Our goal will be to determine whether there is any bias in the predicted point spreads. One way of getting at this is to attempt to learn if Y_i has mean 0. So, we assume $Y_i \sim \text{Normal}(\mu, \sigma^2)$, and our goal is to determine the reasonable values for μ.

Now, I happen to know from prior gambling experience that the point spreads tend to be accurate, and moreover that the standard error of the point spread is roughly two touchdowns (14 points). So I set $\mu \sim \text{Normal}(0, \tau^{-1}/2)$ and $\tau \sim \text{Gam}(1/2, 14^2/2)$. This prior expresses the prior information that (i) I expect the point spread to be zero on average, (ii) I expect the standard deviation to be roughly 14, and (iii) I do not have that much faith in my guesses. Note that from part (c) of Exercise 1, κ and α are roughly on the scale of the number of observations; hence, κ can be viewed as the "number of observations my prior is worth"; the same is true of 2α. I set $\kappa = 2$ (making my prior guess of $\mu = 0$ worth about two observations) and $\alpha = 1/2$ and $\beta = 14^2/2$ making my prior guess on the variance worth one observation.

Code in `Notes-01-Spread.R`

Exercise 2. Suppose $Y \sim \text{Binomial}(n, \theta)$. Consider the family of densities of the form

$$\pi(\theta) = \lambda \,\text{Beta}(\theta \mid \alpha_1, \beta_1) + (1 - \lambda)\,\text{Beta}(\theta \mid \alpha_2, \beta_2),$$

where $0 \leq \lambda \leq 1$ and $\alpha_1, \beta_1, \alpha_2, \beta_2 > 0$. Show that this family of densities is conjugate.

As mentioned above, useful conjugate relationships are limited to distributions in the exponential family. This is the result of the following facts: (i) useful conjugacy relationships require the prior $\{\pi(\theta) : \theta \in \Theta\}$ to be of fixed dimension, but (ii) the posterior will only be of fixed dimension when there is a fixed dimension sufficient statistic which (iii) in the regular case, only occurs when the sampling distribution $f_\theta(y)$ is in the exponential family. A formal statement of fact (iii) is given by the Pitman-Koopman lemma.

Theorem 4 (Pitman-Koopman Lemma, as given in Section 3.3. of Robert)**.** *If a family of distributions $f_\theta(y)$ is such that, for a sample size large enough, there exists a sufficient statistic of constant dimension, the family is exponential if the support of f_θ does not depend on θ.*

So, (fixed-dimension) conjugacy is usually limited to exponential families. Additionally, given an exponential family, it is easy to construct a conjugate prior.

Example 5 (Exponential Families)**.** Consider an exponential family, which is defined as admitting a density of the form

$$f_\theta(y) = \exp\left\{T(y)^\top \theta - \psi(\theta) + h(y)\right\},$$

with respect to some containing measure $h(y)$. Define

$$\pi(\theta) = K(\mu, \lambda) \exp\left\{\mu^\top \theta - \lambda\psi(\theta)\right\},$$

where $K(\mu, \lambda) = \int \exp\{\mu^\top \theta - \lambda\psi(\theta)\}\, d\theta$; this can be shown to be finite if-and-only-if $\lambda > 0$ and μ/λ lies in the interior of the parameter space Θ.

Under these conditions, the posterior given $Y_1, \ldots, Y_n$ iid from $f_\theta(y)$ is

$$\pi(\theta \mid Y_1, \ldots, Y_n) \propto \exp\left\{\left(\mu + \sum_{i=1}^N T(Y_i)\right)^\top \theta - (\lambda + n)\psi(\theta)\right\}.$$

Hence the posterior is in the same family as the prior, with μ replaced by $\mu + \sum_i T(Y_i)$ and λ replaced by $\lambda + n$.

When the support varies with θ, there are some other opportunities for conjugate priors. For example, the family of Uniform$(0, \theta)$ sampling distributions (which is not a an exponential family) admits a fixed-dimension family of conjugate priors.

Example 6 (Uniform distribution)**.** Suppose $Y_1, \ldots, Y_n \overset{\text{iid}}{\sim} \text{Uniform}(0, \theta)$. Then we have the likelihood

$$L(\theta) \propto \prod_{i=1}^n \frac{1}{\theta} I(0 \leq Y_i \leq \theta) = \theta^{-n} I(\theta \geq \max_i Y_i).$$

A conjugate prior for θ is given by $\pi(\theta) \propto \theta^{-\alpha} I(\theta \geq \beta)$, and the posterior is easily shown to be $\pi(\theta \mid Y_1, \ldots, Y_n) \propto \theta^{-(\alpha+n)} I(\theta \geq \beta \wedge \max_i Y_i)$.

3 The Laplace Approximation

When a (useful) family of conjugate priors exists and can be reasonably used, computation is not difficult. Unfortunately, there are many situations where there are no (useful) conjugate priors.

Example 7. Consider $Y_i \overset{\text{ind}}{\sim} \text{Bernoulli}(\pi_i)$ with $\text{logit}\,\pi_i = X_i^\top \beta$ for $i = 1, \ldots, n$. The likelihood of β is given by

$$L(\beta) = \prod_{i=1}^{n} \frac{\exp(Y_i X_i^\top \beta)}{1 + \exp(X_i^\top \beta)}.$$

Following the recipe for exponential families, we get the following conjugate prior.

$$\pi(\beta) \propto \exp\left(\beta^\top \mu\right) \prod_{i=1}^{n} (1 + \exp(\beta^\top X_i))^{-\lambda}.$$

This is called the *Diaconis-Ylvisaker prior* for logistic regression. Unfortunately, there is no known way to simulate efficiently from this distribution, no formulae for the moments of this distribution, and no formula for the normalizing constant.

Perhaps surprisingly, Laplace himself already developed a key tool for approximating the posterior distribution in large samples. The idea is to essentially appeal to likelihood theory! We consider the a Taylor expansion of the log of the posterior density as

$$\log \pi(\theta \mid Y) \approx \log \pi(\widehat{\theta} \mid Y) + (\theta - \widehat{\theta})^\top \underbrace{s(\widehat{\theta})}_{=0} - \frac{1}{2}(\theta - \widehat{\theta})^\top \mathcal{I}(\widehat{\theta})(\theta - \widehat{\theta}),$$

where $\widehat{\theta} = \text{argmax}\, \pi(\theta \mid Y)$, $s(\theta) = \frac{\partial}{\partial \theta} \log \pi(\theta \mid Y)$, and $\mathcal{I}(\theta) = -\frac{\partial^2}{\partial \theta \partial \theta^\top} \log \pi(\theta \mid Y)$. Ignoring constants, this is the kernel of a Gaussian distribution with mean $\widehat{\theta}$ and covariance matrix $\mathcal{I}(\widehat{\theta})^{-1}$. This gives the approximation

$$\pi(\theta \mid Y) \approx |\mathcal{I}(\widehat{\theta})|^{1/2} (2\pi)^{-P/2} \exp\left(-\frac{1}{2}(\theta - \widehat{\theta})^\top \mathcal{I}(\widehat{\theta})(\theta - \widehat{\theta})\right)$$

where P is the dimension of θ.

3.1 How to implement the Laplace approximation?

The main payoff from using a Laplace approximation is that we have converted a very difficult problem (integration over a large space) to a less difficult problem (optimization of a function). All we need to do is compute $\widehat{\theta}$ to apply the approximation, and there are a large number of highly effective tools for optimizing smooth functions. Additionally, notice that

$$\log \pi(\theta \mid Y) = \log \pi(\theta, Y) - \log M$$

where $M = \int \pi(\theta, Y)\, d\theta$ is the *marginal likelihood* of Y. Because M does not depend on θ, it suffices to optimize $\log \pi(\theta, Y) = \log \pi(\theta) + \log f(Y \mid \theta)$ to compute $\widehat{\theta}$ (i.e., we do not need to be able to compute M).

3.2 Example

`Notes-02-Field-Goals.R`

3.3 When will the Laplace approximation work?

The Laplace approximation can be expected to work under essentially the same conditions that likelihood theory works: the problem should be regular, with a fixed number of parameters, and the number of observations should be quite large. Given the form of the approximation, we should also expect the same sorts of artifacts typically associated with "Wald" methods — we are using the plug-in estimate $\widehat{\theta}$ to approximate the covariance, which we know leads to all sorts of issues. The Laplace approximation is also not parameterization invariant, and we should choose a parameterization for which the normal approximation is good.

The Laplace approximation will not be accurate when P is large relative to N, or when N itself is small. When the Laplace approximation *is* accurate, we also expect that it will give roughly the same answers as traditional Frequentist methods.

4 Monte Carlo and MCMC

For many years, asymptotic approximations like the Laplace approximation were the only tools available for performing Bayesian inference. Eventually, physicists working on the *Manhattan Project* (which lead to the development of atomic weapons) invented the *Monte Carlo* method for computing integrals.

The idea behind Monte Carlo is very simple, and likely would have been studied earlier had computers been available earlier. The idea is that we can compute an integral $\int g(x)\, f(x)\, dx$ by sampling many values $X_1, \ldots, X_N \sim f$ and computing $\widehat{g} = N^{-1} \sum_i g(X_i)$. By the central limit theorem, we have

$$\widehat{g} \overset{\bullet}{\sim} \text{Normal}\left(\int g(x)\, f(x)\, dx, N^{-1} \text{Var}(g(X_i)) \right),$$

provided that $\text{Var}(g(X_i)) < \infty$.

Example 8 (Computing π)**.** We show how to use the Monte Carlo to compute the circle constant π. To do this, let $(X_i, Y_i) \overset{\text{iid}}{\sim} \text{Uniform}([-1,1]^2)$ for $i = 1, \ldots, N$, for some very large N. Then it can be shown that $\Pr(X_i^2 + Y_i^2 \leq 1) = \pi/4$ — this is because the area of the square $[-1, 1]$ is 4, while the area of the unit circle is π.

We define $Z_i = I(X_i^2 + Y_i^2 \leq 1)$; then $Z_i \overset{\text{iid}}{\sim} \text{Bernoulli}(\pi/4)$. We can create a confidence interval for π as $4(\bar{Z} \pm 1.96 \cdot \sqrt{\bar{Z}(1-\bar{Z})/N})$; we do this in the file `Notes-02-Pi.R` to get $(3.139811, 3.146245)$ using a sample of $N = 10^6$.

While this strategy works, evidently it lacks precision: using $N = 10^6$ we are only estimate two significant digits. This is true in general of Monte Carlo methods; the accuracy improves like $\sqrt{N}$, so that to get a 10-fold increase in accuracy requires a 100-fold increase in N. To get a third digit, for example, would require 10^8 samples.

While Monte Carlo may seem inefficient in this example, an important point is that the accuracy improves like $\sqrt{N}$ *regardless of the dimension of the problem.* This property is not shared by most numeric integration schemes, which instead have accuracy which improves like $N^{1/d}$, where d is the dimension of the parameter.

4.1 Naive Monte Carlo

To apply Monte Carlo to computing (say) $E(\theta \mid Y)$, we would need to use it to compute $\int \theta\, \pi(\theta \mid y)\, d\theta$. This converts the problem of computing an integral to the problem of *sampling from* $\pi(\theta \mid y)$. If we can sample

from $\pi(\theta \mid y)$, we will be able to do everything we want to do.

Unfortunately, sampling from $\pi(\theta \mid y)$ is itself a very difficult problem in most cases. But we can still make progress. First, recall that

$$E(\theta \mid y) = \frac{\int \theta\, \pi(\theta)\, f_\theta(y)\; d\theta}{\int \pi(\theta)\, f_\theta(y)\; d\theta}.$$

We have two integrals that we wish to compute. Both of these are expectations with respect to the prior distribution! That is, we need to compute

$$E(\theta \cdot f_\theta(y)) \qquad \text{and} \qquad E(f_\theta(y)),$$

with respect to the prior distribution $\pi(\theta)$. Both of these can be computed by sampling $\pi(\theta)$ from the prior, giving

$$M^{-1} \sum_{m=1}^{M} \theta_m f_{\theta_m}(y) \qquad \text{and} \qquad M^{-1} \sum_{m=1}^{M} f_{\theta_m}(y).$$

Example 9 (Logistic Regression)**.** `Notes-02-Field-Goals.R` In this file, we sample directly from the Cauchy prior to approximate the posterior distribution.

4.2 Importance Sampling

While Naive Monte Carlo does, in some sense, work, it is not generally a good strategy for a number of reasons. We saw this in the case of logistic regression: sampling from the prior did not produce very accurate estimates of the quantities of interest.

The problem in the above example is that the functions $\theta f_\theta(y)$ and $f_\theta(y)$ *are effectively zero* for most values that $\pi(\theta)$ assigns mass to, and spike in a very narrow area.

Essentially, we will be able to compute $\int g(x)\, f(x)\; dx$ by sampling from $f(x)$ efficiently if $f(x)$ assigns its mass to the same areas where $g(x)$ is large. The above example samples from $\pi(\theta)$, which assigns most of its mass to areas of where $f_\theta(y)$ is small. One way to fix this is to use *importance sampling*, which notes that (for any proposal density $q(\theta)$) we have

$$\int \pi(\theta) f_\theta(y)\; d\theta = \int \frac{\pi(\theta) f_\theta(y)}{q(\theta)}\; q(\theta) d\theta.$$

Hence, an alternative way to compute $E(\theta \mid Y)$ is to sample $\theta_1, \ldots, \theta_M$ from $q(\theta)$ and compute

$$M^{-1} \sum_{m=1}^{M} \frac{\pi(\theta_m) f_{\theta_m}(y)}{q(\theta_m)} \qquad \text{and} \qquad M^{-1} \sum_{m=1}^{M} \frac{\theta_m \pi(\theta_m) f_{\theta_m}(y)}{q(\theta_m)},$$

and form the approximation

$$E(\theta \mid Y) \approx \frac{\sum_{m=1}^{M} \dfrac{\theta_m \pi(\theta_m) f_{\theta_m}(y)}{q(\theta_m)}}{\sum_{m=1}^{M} \dfrac{\pi(\theta_m) f_{\theta_m}(y)}{q(\theta_m)}}$$

This will work well if the support of $q(\theta)$ overlaps with the support of $\pi(\theta) f_\theta(y) \propto \pi(\theta \mid y)$. That is, we would like to choose $q(\theta)$ so that it approximates the posterior distribution.

One way of understanding importance sampling, relative to usual Monte Carlo, is that both approximate the posterior as

$$\pi(\theta \mid Y) \approx \sum_{i=1}^{M} \omega_m \delta_{\theta_m}$$

where the θ_m's are iid from some distribution and the ω_m's are probabilities assigned to each θ_m. In the case of Monte Carlo, the weights ω_m are identically $1/M$ and the distribution we are sampling from is required to be $\pi(\theta \mid Y)$. On the other hand, importance sampling takes θ_m from $q(\theta)$ and uses the weights

$$\omega_k = \frac{\frac{\pi(\theta_k) f_{\theta_k}(y)}{q(\theta_k)}}{\sum_{i=1}^{M} \frac{\pi(\theta_m) f_{\theta_m}(y)}{q(\theta_m)}}$$

Notice also that, in our previous "naive" version of Monte Carlo, we were also implicitly using this strategy; in this case, the importance distribution was $\pi(\theta)$, which it turns out is not a very good choice. The best choice would be to use $\pi(\theta \mid Y)$ but, absent of that, it is desirable to use a distribution which *approximates* $\pi(\theta \mid Y)$.

Example 10 (Logistic regression again)**.** We revisit the field goals example in `Notes-02-Filed-Goals.R` using the Laplace approximation as an importance sampling distribution.

4.3 Markov chain Monte Carlo

In high dimensions, even constructing a reasonable *approximation* to $\pi(\theta \mid Y)$ becomes challenging; while the Laplace approximation perhaps works well for inferential purposes, it generally does not do very well as an importance sampling distribution.

This issue was, again, well-known to those working on the Manhattan project. To handle situations where neither Monte Carlo nor importance sampling could be effective, they invented *Markov chain Monte Carlo*. The idea behind MCMC is to construct a sequence of draws $\theta_1, \theta_2, \ldots$ by iteratively sampling

$$\theta_m \sim Q(\theta \mid \theta_{m-1})$$

where the distribution $Q(\cdot \mid \theta)$ is called a *transition kernel*. The sequence $\theta_1, \theta_2, \ldots$ forms a *Markov sequence* in the sense that θ_i is independent of $\theta_1, \ldots, \theta_{i-2}$, given θ_{i-1}. That is, we have a dependence structure like

$$\theta_1 \to \theta_2 \to \cdots \to \theta_{m-1} \to \theta_m \to \cdots$$

in which conditioning on the immediate past (θ_{m-1}) makes all previous values of θ irrelevant to the distribution of θ_m.

Now, suppose we choose $Q(\cdot \mid \theta)$ so that it has the following property: (i) if $\theta_m \sim \pi(\theta \mid Y)$ and (ii) $\theta_{m+1} \sim Q(\cdot \mid \theta_m)$, then (iii) $\theta_{m+1} \sim \pi(\theta \mid Y)$. This essentially says that the distribution $\pi(\theta \mid Y)$ is a *fixed point* of the transition operator: the posterior distribution is *invariant* under the transition operation. When such invariance holds, it can be shown that *under extremely weak conditions*, we have $\theta_m \to \pi(\theta \mid Y)$ in distribution as $m \to \infty$.

The idea behind Markov chain Monte Carlo is to sample such a $\theta_1, \theta_2, \ldots$ from an appropriate $Q(\cdot \mid \theta)$, and approximate the posterior as

$$\pi(\theta \mid Y) \approx M^{-1} \sum_{m=1}^{M} \delta_{\theta_m},$$

exactly as we would using Monte Carlo. The difference here is that the draws are no longer independent, and no longer identically distributed.

4.3.1 Technical Conditions: Irreducibility and Aperiodicity

Technically, there are three conditions we require from a Markov chain in order for us to sample from the posterior distribution.

Definition 11. Consider a Markov chain $\theta_1, \theta_2, \ldots$, operating on a space Θ defined by a transition kernel $Q(\cdot \mid \theta_m)$.

1. This chain is called *aperiodic* if there do not exist disjoint, non-empty, subsets $\Theta_1, \ldots, \Theta_r$ $(r \geq 2)$ such that $\Pr(\theta_{m+1} \in \Theta_{i+1} \mid \theta_m)$ whenever $\theta_m \in \Theta_i$ for $1 \leq i \leq r-1$ and $\Pr(\theta_{m+1} \in \Theta_1 \mid \theta_m)$ whenever $\theta_m \in \Theta_r$.
2. This chain is called *ϕ-irreducible* if there exists a non-zero measure ϕ on Θ such that for all $\theta \in \Theta$ and all $A \subseteq \Theta$ with $\phi(A) > 0$, there is positive probability that the chain will eventually hit A if started at a given point θ_0.
3. The chain is called *ergodic* with *stationary density* $\pi(\cdot \mid Y)$ if it is irreducible, aperiodic, and leaves $\pi(\cdot \mid Y)$ invariant.

The point of MCMC is to construct an ergodic Markov chain with stationary density $\pi(\cdot \mid Y)$, because this chain satisfies similar properties to an iid sequence from $\pi(\cdot \mid Y)$. In particular, we have the *ergodic theorem* which states

$$\frac{1}{M}\sum_{m=1}^{M} f(\theta_m) \to \int f(\theta)\,\pi(\theta \mid Y)\,d\theta, \tag{2}$$

almost surely (provided this expectation is exists). Additionally, for any (measurable) $B \subseteq \Theta$ and $\pi(\cdot \mid Y)$-almost-all $\theta' \in \Theta$, we have

$$\Pr(\theta_m \in B \mid \theta_1 = \theta') \to \int_B \pi(\theta \mid Y)\,d\theta. \tag{3}$$

The point of aperiodicity is that it ensures that θ_m can be "anywhere" if ran long enough; for example, it does not flip back and forth deterministically between (say) the odd and even numbers on successive iterations. The point of ϕ-irreducibility is that it ensures that the chain can reach any given state with positive probability. These conditions, with the additional condition of keeping the posterior invariant, are (surprisingly) sufficient to guarantee that we can obtain (3).

Generally, these conditions are easy to check - for example, if the transition density assigns positive density to every $\theta \in \Theta$, then we automatically have both aperiodicity and irreducibility. Moreover, any chain which is irreducible and allows for *self-loops* (i.e., it is possible for the chain not to move) is automatically aperiodic. As almost all chains which are used in practice allow self-loops, this makes aperiodicity very easy to obtain.

4.3.2 The Metropolis-Hastings Algorithm

While it may seem difficult to come up with such a $Q(\cdot \mid \theta)$ which holds the posterior distribution invariant, there actually exist very general methods for constructing such a $Q(\cdot \mid \theta)$. The most useful general-purpose

approach is probably the *Metropolis-Hastings* algorithm; this algorithm converts essentially *any* transition $\kappa(\cdot \mid \theta)$ into a transition $Q(\cdot \mid \theta)$ which has the posterior as a stationary distribution.

The Metropolis-Hastings algorithm proceeds as follows. Suppose we have a transition density $\kappa(\cdot \mid \theta)$, which is arbitrary. We iterate the following procedure.

1. Sample $\theta^\star \sim \kappa(\theta \mid \theta_m)$.
2. Compute an *acceptance ratio*
$$a(\theta^\star \mid \theta_m) = \min\left\{\frac{\pi(\theta^\star \mid Y)\,\kappa(\theta_m \mid \theta^\star)}{\pi(\theta_m \mid Y)\,\kappa(\theta^\star \mid \theta_m)}, 1\right\}.$$
3. Set $\theta_{m+1} = \theta^\star$ with probability $a(\theta^\star \mid \theta_m)$; otherwise, set $\theta_{m+1} = \theta_m$.

This can be shown to hold $\pi(\theta \mid Y)$ invariant; the proof, which is not too difficult, is omitted for time.

Again, one might be troubled by the fact that $a(\theta^\star \mid \theta_m)$ depends on $\pi(\theta \mid Y)$, which in principle we do not know. This is no problem, however, because

$$\frac{\pi(\theta \mid Y)}{\pi(\theta^\star \mid Y)} = \frac{\pi(\theta) f_\theta(Y)}{\pi(\theta^\star) f_{\theta^\star}(Y)},$$

with the normalizing constant $M = \int \pi(\theta)\, f_\theta(Y)\, d\theta$ canceling in the numerator and denominator. Therefore the acceptance ratio can be computed simply as

$$a(\theta^\star \mid \theta_m) = \min\left\{\frac{\pi(\theta^\star) f_{\theta^\star}(Y)\,\kappa(\theta_m \mid \theta^\star)}{\pi(\theta_m) f_{\theta_m}(Y)\,\kappa(\theta^\star \mid \theta_m)}, 1\right\}.$$

The only thing that the user must specify to apply Metropolis-Hastings is a κ; it should be no surprise, at this point, that choosing a good κ is essential for this strategy to be successful. We describe two common choices.

Random Walk Metropolis. The Random Walk Metropolis Hastings (RWMH) algorithm takes $\kappa(\theta \mid \theta_m)$ so that $\theta^\star = \theta_m + \epsilon$ for some ϵ which is independent of θ_m. A common choice would be $\epsilon \sim \text{Normal}(0, \Sigma)$ (the multivariate-t distribution is also sometimes used) for some covariance matrix Σ. The matrix Σ is a tuning parameter. In general, we can write $\kappa(\theta \mid \theta_m) = h(\theta - \theta_m)$ for some density $h(\cdot)$. Furthermore, if h is symmetric about 0 then $h(\theta - \theta_m) = h(\theta_m - \theta)$ which simplifies the acceptance ratio:

$$a(\theta^\star \mid \theta_m) = \min\left\{\frac{\pi(\theta^\star) f_{\theta^\star}(Y)}{\pi(\theta_m) f_{\theta_m}(Y)}, 1\right\}.$$

To implement RWMH, we must select the tuning parameter Σ. One guideline for choosing Σ is so that the acceptance probability is equal, on average, to some number (say 25% or 50%). If the acceptance probability is too small, we will never accept any moves, while if it is too large then this suggests that we are not making steps which are large enough to efficiently explore the target distribution.

It is difficult to give general guidelines for tuning RWMH; when possible, it is probably best to use a software package which automatically implements the tuning for you.

Independence Metropolis. Independence Metropolis algorithms choose $\kappa(\theta \mid \theta_m)$ so that the proposal *does not depend on* θ_m. That is, we simply propose $\theta^\star \sim q(\theta)$ for density $q(\theta)$. The issues here are very similar to the issues with importance sampling: it is quite difficult to choose a good $q(\theta)$. In this case, what we should aim for is to maximize the acceptance rate. In the ideal case where $q(\theta) = \pi(\theta \mid Y)$, we get an acceptance rate of 1.

Dimensionality Issues. In the case of both RWMH and independence Metropolis, we really are not doing much better than the other methods proposed. This is because both of these procedures become increasingly inefficient as the dimension of the problem increases. In the case of independence Metropolis, it becomes very difficult to find a suitable $q(\theta)$, while in the case of RWMH, higher dimensionality requires taking very small step sizes (resulting in efficient exploration of the posterior).

4.3.3 The Gibbs Sampler and Metropolis within Gibbs

A particularly important MCMC algorithm is known as the *Gibbs sampler*; for our purposes, the Gibbs sampler is the most important MCMC method to understand. The Gibbs sampler is important because it is one of only two (to my knowledge) MCMC algorithms which can be used to tackle high-dimensional problems.

Suppose we are interested in sampling from a joint density $f(x, y)$. We assume that

1. Sampling from $f(x \mid y)$ is easy; and
2. Sampling from $f(y \mid x)$ is easy.

Then, we can construct a sequence $(X_1, Y_1), (X_2, Y_2), \ldots$ which samples approximately from $f(x, y)$ by iterating the following steps.

1. Sample $X_i \sim f(x \mid Y_{i-1})$; and
2. Sample $Y_i \sim f(y \mid X_i)$.

We can explicitly write the transition density for this algorithm as $Q(x, y \mid x', y') = f(x \mid y')f(y \mid x)$. From this, we can verify that this transition holds the target distribution $f(x, y)$ invariant:

$$\begin{aligned}\int f(x', y')\, Q(x, y \mid x', y')\; dx'\; dy' &= \int f(x', y')\, f(x \mid y')\, f(y \mid x)\; dx'\; dy' = f(y \mid x) \int f(x \mid y') f(y')\; dy' \\ &= f(y \mid x) \int f(x, y')\; dy' = f(y \mid x)\, f(x) = f(x, y).\end{aligned}$$

Assuming that the sequence $(X_1, Y_1), \ldots$ can also be shown to be irreducible and aperiodic, then this procedure can be used to approximately sample from $f(x, y)$.

The Gibbs Sampling Algorithm. To apply Gibbs sampling to sampling from the posterior distribution, first suppose that we can write $\theta = (\theta_1, \ldots, \theta_K)$ where each θ_k is a Euclidean parameter of some dimension. Define the *full conditional* of θ_k as its conditional distribution given Y and $\theta_{-k} = (\theta_1, \ldots, \theta_{k-1}, \theta_{k+1}, \ldots, \theta_K)$ and write $\pi(\theta_k \mid Y, \theta_{-k})$ for this conditional density. A typical Gibbs sampling algorithm proceeds as follows:

1. Initialize $\theta_{01}, \ldots, \theta_{0K}$ in some fashion and set $m = 1$.
2. For $k = 1, \ldots K$, sample $\theta_{(m+1)k}$ from $\pi(\theta_k \mid Y, \theta_{(m+1)1}, \ldots, \theta_{(m+1)(k-1)}, \theta_{m(k+1)}, \ldots, \theta_{mK})$.
3. Set $m \leftarrow m + 1$ and return to step 2.

Example 12 (Bayesian Linear Regression)**.** Consider a linear model with $Y = X\beta + \epsilon$ where $\epsilon \sim \text{Normal}(0, \tau^{-1}\text{I})$. We consider the following prior:

$$\beta \sim \text{Normal}(0, \Omega^{-1}) \qquad \text{and} \qquad \tau \sim \text{Gam}(a, b).$$

Despite the friendly-looking nature of this prior, it is not actually conjugate and the posterior cannot be derived in closed form. On the other hand, it is simple to derive the full conditional distributions as

$$[\beta \mid Y, \tau] \sim \text{Normal}\left(\tau(\Omega + \tau X^\top X)^{-1} X^\top Y, (\Omega + \tau X^\top X)^{-1}\right)$$
$$[\tau \mid Y, \beta] \sim \text{Gam}\left(a + N/2, b + \|Y - X\beta\|^2/2\right).$$

Hence we can perform the Gibbs sampler by iteratively sampling β and τ from these two distributions.

Blocking. The Gibbs sampler requires that we break θ into components $(\theta_1, \ldots, \theta_K)$. Each θ_k is referred to as a *block* of parameters. Generally, Gibbs sampling works best when *the number of blocks is small.* Like Metropolis-Hastings, Gibbs sampling suffers when the dimensionality is high; using a small number of blocks (say, one or two) can go a long way to minimizing the impact of dimensionality. For example, it is quite common to apply the Gibbs sampler in ultra-high dimensions (say, $P = 10^6$ or more), but this is usually only successful when the number of blocks is small.

Example 13 (Blocking example)**.** Consider the case of *probit regression* in which $Y_i \sim \text{Bernoulli}(\pi_i)$ with

$$\pi_i = \Phi(X_i^\top \beta),$$

and $\Phi(\cdot)$ denotes the cdf of a standard normal distribution. This model has a latent variable representation as

$$Z_i = X_i^\top \beta + \epsilon_i, \qquad \epsilon_i \overset{\text{iid}}{\sim} \text{Normal}(0, 1), \qquad Y_i = I(Z_i > 0).$$

We aim to sample from the posterior $\pi(\beta, \boldsymbol{Z} \mid \boldsymbol{Y})$ where $\boldsymbol{Z} = (Z_1, \ldots, Z_n)$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$. Suppose we impose a flat prior $\pi(\beta) \propto 1$. Then we have

$$\pi(\beta, \boldsymbol{Z} \mid \boldsymbol{Y}) \propto \prod_{i=1}^n \text{Normal}(Z_i \mid X_i^\top \beta, 1)$$

subject to the restriction that $Z_i > 0$ and $Z_i < 0$ according as $Y_i = 1$ or $Y_i = 0$. Conditional on $\boldsymbol{Z}$, the full conditional of β is then

$$\pi(\beta \mid \boldsymbol{Z}, \boldsymbol{Y}) \propto \prod_{i=1}^n \text{Normal}(Z_i \mid X_i^\top \beta)$$

which, as we have seen in before, is proportional to $\text{Normal}(\beta \mid \widehat{\beta}, (X^\top X)^{-1})$ where $X^\top = (X_1, \ldots, X_n)$. The full conditional of $\boldsymbol{Z}$ is the same

$$\pi(\boldsymbol{Z} \mid \beta, \boldsymbol{Y}) \propto \prod_{i=1}^n \text{Normal}(Z_i \mid X_i^\top \beta, 1) I(Z_i \in A(Y_i))$$

where $A(0) = (-\infty, 0)$ and $A(1) = (0, \infty)$. This is proportional to a *truncated normal* density function, giving

$$Z_i \overset{\text{ind}}{\sim} \text{TruncNormal}(X_i^\top \beta, 1, A(Y_i)) \qquad \text{given } \beta \text{ and } \boldsymbol{Y}.$$

This leads to the following two-block Gibbs sampler.

1. Sample $\beta \sim \text{Normal}(\widehat{\beta}, (X^\top X)^{-1})$ where $\widehat{\beta} = (X^\top X)^{-1} X^\top \boldsymbol{Z}$.
2. For $i = 1, \ldots, n$, sample $Z_i \sim \text{TruncNormal}(X_i^\top \beta, 1, A(Y_i))$.

3. Record $(\beta, \boldsymbol{Z})$ and return to step 1.

Metropolis within Gibbs. Gibbs sampling works best when the full conditionals $\pi(\theta_k \mid Y, \theta_{-k})$ can be easily sampled from. Interestingly, we can still apply the Gibbs sampler, even when this is not the case! An inspection of the proof that Gibbs sampling leaves the target distribution invariant shows that *we can also use any transition density which leaves the full conditional invariant.*

For example, suppose that $Q_x(x \mid x', y')$ leaves $f(x \mid y')$ invariant, while $Q_y(y \mid x, y')$ leaves $f(y \mid x)$ invariant. Then we have

$$\begin{aligned}
\int f(x', y')\, Q(x, y \mid x', y')\; dx'\; dy' &= \int f(x', y')\, Q_x(x \mid x', y')\, Q_y(y \mid x, y')\; dx'\; dy' \\
&= \int f(y') f(x \mid y')\, Q_x(x \mid x', y')\, Q_y(y \mid x, y')\; dx'\; dy' \\
&= \int f(y') f(x \mid y')\, Q_y(y \mid x, y')\; dy' \\
&= \int f(x)\, f(y' \mid x)\, Q_y(y \mid x, y')\; dy' \\
&= f(x)\, f(y \mid x) = f(x, y).
\end{aligned}$$

Hence, if have a complicated full conditional $\pi(\theta_k \mid Y, \theta_{-k})$, we can always revert back to Metropolis-Hastings to sample this.

4.3.4 The Metropolis-Adjusted Langevin (MALA) algorithm and Hamiltonian Monte Carlo

The "gold standard" for general-purpose MCMC is Hamiltonian Monte Carlo (HMC), which resolves many of the problems with Metropolis-Hastings and has genuine aspirations of sampling from high-dimensional targets.

To motivate HMC, we first describe a related method known the Metropolis-Adjusted Langevin algorithm (MALA). MALA uses a proposal of the form

$$\theta^\star = \theta_m + \epsilon \nabla \log \pi(\theta_m \mid Y) + \xi_m$$

where $\xi_m \sim \text{Normal}(0, 2\epsilon)$ and ϵ is a tuning parameter. This proposal is then used as the κ for the Metropolis-Hastings algorithm.

The reason MALA gives a good choice for the proposal distribution is that the chain is pushed in the direction of the gradient; consequently, the proposal $\theta^\star$ will (tend to) have a higher likelihood than θ_m, leading to a higher probability of acceptance.

The key here is that the gradient provides a "hint" as to where it would be best for the chain to explore; additionally, this strategy does not require any in-depth knowledge of $\pi(\theta \mid Y)$ to implement. Thus MALA guides to chain to regions of high posterior probability.

HMC takes advantage of information in the gradient in a more profound way. The idea behind HMC is to construct a transition kernel which is based on simulating the *Hamiltonian dynamics* of a certain physical system. The dynamics of the system are determined by current *position* of a particle (θ) and certain *momentum variables* (p); the position and momentum of the system are then simulated according the Hamiltonian dynamics

$$\frac{d\theta_i}{dt} = \frac{\partial H}{dp_i}, \qquad \frac{dp_i}{dt} = -\frac{\partial H}{\partial \theta_i},$$

where $\theta, q \in \mathbb{R}^d$. The function $H(\theta, q)$ is the *Hamiltonian*

$$H(\theta, p) = U(\theta) + K(p)$$

where $U(\theta)$ represents the *potential energy* of the system and $K(p)$ represents the *kinetic energy* of the system.

An "ideal" variant of Hamiltonian Monte Carlo proceeds by first simulating $p \sim \exp\{-K(p)\}$ and second evolving θ according to the dynamics described above for some time T. Amazingly, this can be shown to leave $\exp\{-U(\theta)\}$ invariant; hence, if we take $U(\theta) = -\log \pi(\theta \mid Y)$ we can leave the posterior distribution invariant. This transition has the following two highly desirable properties:

1. It can make proposals which are far away from the current point, hence giving some prospect of obtaining nearly-independent samples of θ in a single trajectory.
2. The gradient is taken into account automatically, and the values of θ will tend towards the so-called "typical set" which contains most of the mass of the posterior.

Unfortunately, it is impossible to implement this idealized version of HMC in practice, because — except in very rare instances — we will be unable to simulate the Hamiltonian dynamics exactly. To bypass this, we evolve the system using a discretization of time. We can approximately evolve (θ, p) according to the Hamiltonian dynamics by considering a discrete-time version with a timescale ϵ, which is run for L steps, for a total simulation time of $T = \epsilon L$. This does not leave the posterior invariant, but we add a Metropolis Hastings step to correct this issue. A popular implementation of this idea is the "leapfrog algorithm" given below.

Algorithm 1 HMC Proposal using Leapfrog Integration with Gaussian Momentum

Given: initial $\theta_0 \in \mathbb{R}^P$, potential energy $U(\theta) = -\log \pi(\theta \cdot \mid Y)$, step size ϵ, and number of leapfrog steps L.

1. Sample $p \sim \text{Normal}(0, M)$.
2. Save $\theta_0 \leftarrow \theta$ and $p_0 \leftarrow p$.
3. Set $p \leftarrow p - \epsilon \dfrac{\partial}{\partial \theta} \dfrac{U(\theta)}{2}$.
4. For $j, = 1, \ldots, L$:
   (a) Set $\theta \leftarrow \theta + \epsilon \dfrac{\partial}{\partial p} K(p) = \theta + \epsilon M^{-1} p$.
   (b) if $j \neq L$, set $p \leftarrow p - \epsilon \dfrac{\partial}{\partial \theta} U(\theta)$.
5. Set $p \leftarrow p - \epsilon \dfrac{\partial}{\partial \theta} \dfrac{U(\theta)}{2}$.
6. Set $p \leftarrow -p$
7. Compute the acceptance probability

$$a = \min \left[1, \exp \left\{ U(\theta_0) + \frac{1}{2} p_0^\top M^{-1} p_0 - U(\theta) - \frac{1}{2} p^\top M^{-1} p \right\} \right].$$

8. With probability a, return θ, otherwise return θ_0.

The above algorithm chooses the Normal$(0, M)$ distribution for the momentum variables, which is a typical choice. In many cases, one takes $M = I_P$ or selects M to be diagonal; rarely, one takes M to be an unstructured covariance. The more complex the form of M, the more effort must be made in tuning the algorithm, and the more expensive the algorithm is. These modifications can improve the mixing of HMC in essential ways, but we will not discuss these details. See Neal (2011, Handbook of Markov Chain Monte Carlo) for an in-depth discussion of HMC and selection of tuning parameters.

The above algorithm has two tuning parameters.

- We must decide how long to run the dynamics before refreshing the momentum variables; this is called the integration time T.
- We must decide on the discretization of time ϵ.

Generally speaking, ϵ determines the acceptance rate of the Metropolis proposal, and can be tuned independently of T to get some targeted acceptance rate (say, 90%). The integration time T is quite difficult to choose in practice. A single iteration of HMC requires $L = T/\epsilon$ computations of the gradient $\nabla \log \pi(\theta \mid Y)$, hence we would like T to be small, but a value too small will lead to poor mixing. Interestingly, due to periodicities in Hamiltonian dynamics, values of T which are *too large* will also result in poor mixing.

For time, we will not delve too far into the details of HMC. Interested students are referred to Neal (2011) and Betancourt (2018). The most popular method for tuning T is to use the No-U-Turn (NUTS) algorithm, which is implemented in the package `STAN`.

Why HMC? The reason HMC is the "gold-standard" is that it is the only MCMC algorithm that (i) jointly updates all components of θ and (ii) has a prayer of approximately sampling $\pi(\theta \mid Y)$ in one step when θ is high-dimensional. All other Metropolis-Hastings algorithms we have discussed fail miserably in high dimensions. The only other scheme which "works" in high-dimensions is Gibbs sampling, and a necessary condition for Gibbs sampling to work in difficult problems is that the number of blocks is small. HMC updates all the parameters in a single block, allowing it to work when Gibbs might not.

Combining HMC and Gibbs. HMC and Gibbs sampling can be used together by using HMC as a Metropolis-within-Gibbs step. The makers of `STAN` claim that this is not necessary, because HMC works better if we only use one block regardless. My experience is that it is occasionally useful to use them together, but we won't discuss what scenarios are required for this to be the case.

4.3.5 MCMC Software

For routine Bayesian analysis, there are fortunately many packages available which make performing inference (in many cases) routine. Even better, these packages allow for the specification of more-or-less arbitrary models! If you can write down a model; arguably, these packages have done more for the proliferation of Bayesian methods than anything else. The packages we will discuss also take care of tuning the algorithms, making application very easy.

JAGS. There are two packages that we will focus on for general purpose MCMC. The first is the `JAGS` package, which can be accessed in `R` through the package `rjags`. `JAGS` stands for "Just Another Gibbs Sampler" and, as its name suggests, it performs inference using Gibbs sampling. The package is intelligent enough to identify blocks of parameters whenever it knows of an efficient way to update the parameters within a block. When it is not able to identify an efficient blocking scheme, it will use Metropolis-within-Gibbs to update the individual parameters.

The success of `JAGS` depends, more-or-less, on whether it can construct efficient updates. This depends, partially, on the ability of the analyst to code the model in a way that `JAGS` likes. When parameters cannot be updated efficiently in a small number of blocks, `JAGS` may fail dramatically.

STAN. The package `STAN` is more recent than `JAGS` and implements Hamiltonian Monte Carlo to perform inference. The advantage of `STAN` is that it updates all the parameters in a single block, and uses a more sophisticated algorithm for exploring the posterior. This allows `STAN` to sample from the posterior distribution very efficiently. Like `JAGS`, however, `STAN` can be very sensitive to how precisely the model is coded.

Examples. We fit the field goals data using both `JAGS` and `STAN`.

4.3.6 Implementation Details

MCMC differs substantially from Monte Carlo and importance sampling in that

(i) the samples form a Markov chain; and

(ii) the samples are not drawn directly from the target distribution.

The combination of these factors increases the number of ways in which MCMC can fail.

To partially address (ii), all MCMC software packages will implement a "warmup" phase in which some initial segment $\theta_1, \theta_2, \ldots, \theta_B$ are *discarded*, i.e. they are not used to compute any quantities of interest. The idea is that θ_1 will be quite far away from the stationary distribution, but by running the chain for a sufficiently long time we will get to a point where θ_B is approximately a draw from the stationary distribution.

Most MCMC algorithms depend on properly selecting tuning parameters to obtain (say) a target acceptance ratio. In the case of HMC we must tune ϵ and T, for RWMH we must tune Σ, and for independence Metropolis we must select $q(\theta)$. These tuning parameters usually must be selected before the chain is run; if the parameters are changed as the chain runs, then we will no longer have a Markov chain. Consequently, the tuning parameters are often tuned during the warmup phase, and fixed when the actual samples are collected. We will not give any details on how this is done.

We must also deal with the fact that the draws from the chain are not independent. One way to assess the degree of dependence is through an *autocorrelation plot*. An example of a poorly mixing MCMC scheme is given in the field goals `R` code. Many packages give the opportunity to *thin* the chain, and collect (say) only $\theta_T, \theta_{2T}, \ldots, \theta_{MT}$. If the *thinning interval* T is sufficiently large, then the samples will be close to independent. Thinning can be convenient for some purposes — it will reduce the memory used to save samples. Thinning does not improve the properties of the chain per se, but it is sometimes convenient to have approximately independent samples from the posterior.

4.3.7 MCMC Diagnostics

After collecting samples via MCMC, it is essential to check the *mixing* of the chain. This usually amounts to looking at some diagnostic plots to check if there is anything obviously wrong with the samples. The first thing one usually looks at is the *trace plots* of the variables; so, for θ_j, we simply plot the sequence $\theta_{1j}, \ldots, \theta_{Mj}$.

Review the diagnostic plots from `Notes-02-Field-Goals.R` for `JAGS` and `STAN`. The output of `STAN` shows reasonable traceplots; roughly speaking, what we want is for the plot to look like a "fat hairy cattepillar." The `JAGS` traceplots look less nice; the dependence in values which are close together is very apparent.

Another commonly used diagnostic is the *autocorrelation plot*. This plot measures how correlated θ_{mj} is with $\theta_{(m+\ell)j}$, for all values of ℓ and j. What we would like to see is that θ_{mj} and $\theta_{(m+\ell)j}$ are close to uncorrelated for small ℓ; certainly, for large ℓ, we do not want to see any lingering correlation. For `STAN`, we see autocorrelation plots with reasonable behavior, while `JAGS` does not have a good autocorrelation plot.

In high dimensional models, it may not be feasible to check the traceplot and autocorrelation figures for *every* parameter. In this case, it is often standard to check the traceplot of "high-level" quantities, like the hyperparameters. The logic here is that the hyperparameters tend to be among the more slowly mixing quantities, so if the mixing of the chain is good for the hyperparameters then it is likely to be acceptable for the lower-level parameters as well. Additionally, it is common to check the mixing of the quantity $\log \pi(\theta, Y)$.

There are a number of other diagnostics one can look at. The default output of `STAN` also gives the $\widehat{R}$ diagnostic (called the Gelman-Rubin statistic) as well as the effective sample size. The Gelman-Rubin statistic, which requires running multiple independent chains, is a measure of whether the chain has reached the stationary distribution or not, and is ideally close to 1. The effective sample size is a measure of how many "independent" samples our collection of dependent samples is worth. Both are expounded on in more detail, for example, in the `STAN` user manual.