Chapter 14: Markov Chain Monte Carlo Methods–Part 2[a]

In this chapter we focus on two MCMC algorithms, namely the slice sampling-within Gibbs algorithm and the Hamiltonian (or hybrid) Monte Carlo algorithm.

The main advantage of these two algorithms is that they depend on very few tuning parameters, which can be easily tuned in an automatic way.

For this reason, the two main existing software that can be used to compute an MCMC approximation of a probability distribution, namely JAGS [22] and Stan [3], are based on these two algorithms.

More precisely, JAGS is particularly well suited for approximating the posterior distribution arising in hierarchical models (see Chapter 15), and the default sampler used by this software is the slice sampling-within Gibbs algorithm.

By contrast, Stan is a general purpose software for approximating the posterior distribution of a Bayesian model, which uses Hamiltonian Monte Carlo (HMC) to perform this task.

[a]The main references for this chapter are [25, Chapter 8] and [18].

The fundamental theorem of simulation

The slice sampler algorithm builds on the following result, known as the fundamental theorem of simulation.

Theorem 14.1 *Let* $\mu \in \mathcal{P}(\mathcal{Z})$ *and* $(Z, U) \sim \mathcal{U}(\mathcal{S}_\mu)$ *with*

$$\mathcal{S}_\mu = \big\{(z, u) \in \mathcal{Z} \times \mathbb{R} : \, 0 < u < \mu(z)\big\}.$$

Then, $Z \sim \mu$.

Proof: Let $p(\cdot)$ denote the density of the $\mathcal{U}(\mathcal{S}_\mu)$ distribution. Then, the result follows from the fact that, for all $z \in \mathcal{Z}$, we have

$$\int_{\mathcal{S}_\mu} p(z, u) \mathrm{d}u = \int_0^{\mu(z)} \mathrm{d}u = \mu(z).$$

□

The conclusion of Theorem 14.1 is illustrated below in Figure 14.1, where μ is the Beta(2,5) distribution.

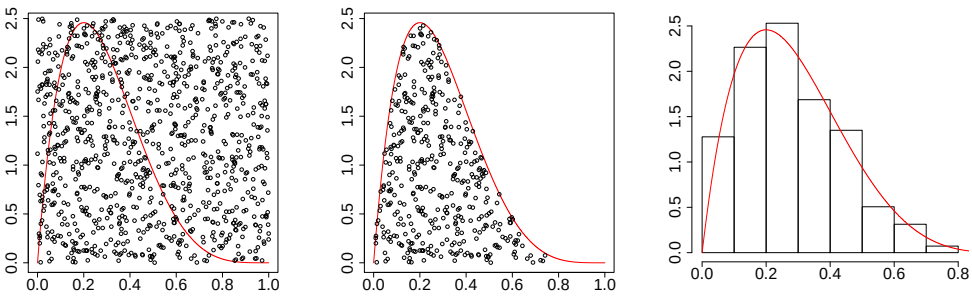

Figure 14.1: Illustration of Theorem 14.1. The dots are random draws from the $\mathcal{U}([0, 1] \times [0, 2.5])$ distribution and the red curves represent the density of the Beta(2,5) distribution. The right plot presents the histogram of the points shown in the middle plot.

The Slice sampler algorithm

From Theorem 14.1 and Chapter 13, a natural idea to approximate μ is to simulate a Markov chain $(Z_t)_{t\geq 0}$ on the set $\mathcal{S}_\mu$ having the $\mathcal{U}(\mathcal{S}_\mu)$ distribution as invariant distribution.

As we saw in Chapter 13, a possible way to simulate such a Markov Chain is to use the Gibbs sampler, in which case the resulting algorithm for approximating μ is known as the slice sampler.

Slice Sampler (Algorithm A6)

Input: $\mu \in \mathcal{P}(\mathcal{Z})$ and $z_0 \in \mathcal{Z}$

(i) Set $Z_0 = z_0$ and $\gamma(z') = \mu(z') \int_{\mathcal{Z}} \mu(\mathrm{d}z)$ for all $z' \in \mathcal{Z}$

for $t \geq 1$ **do**

(ii) Let $U_t \sim \mathcal{U}([0, \gamma(Z_{t-1})])$

(iii) Let $Z_t \sim \mathcal{U}(\mathcal{A}_t)$ with $\mathcal{A}_t = \{z \in \mathcal{Z} : U_t \leq \gamma(z)\}$

end for

Remark: This description of the slice sampler makes clear that, to implement it, we only need to know μ up to a normalizing constant.

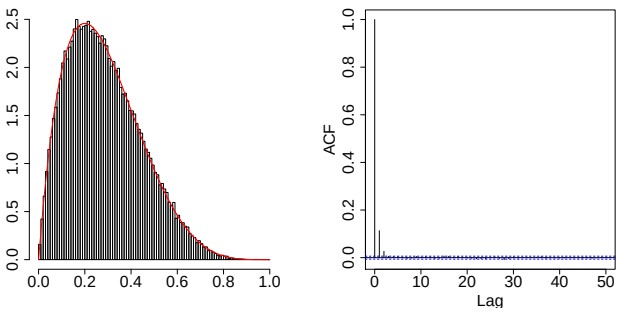

Figure 14.2: Approximation of the Beta(2,5) distribution obtained with the Slice sampler.

Slice sampler: key limitation

The key difficulty to implement Algorithm A5 is to simulate a uniform random variable on the set $\mathcal{A}_t$.

Assuming henceforth that $\mathcal{Z} \subseteq \mathbb{R}$, a natural approach to sample from the $\mathcal{U}(\mathcal{A}_t)$ distribution is to construct an interval $\mathcal{I}_t = [L_t, R_t]$ such that $\mathcal{A}_t \subseteq \mathcal{I}_t$, and then to sample $\tilde{Z} \sim \mathcal{U}(\mathcal{I}_t)$ until we obtain a realization $\tilde{z}$ of $\tilde{Z}$ such that $\tilde{z} \in \mathcal{A}_t$.

The main limitation of this approach is that, for obvious computational reasons, it requires to find an interval $\mathcal{I}_t \supseteq \mathcal{A}_t$ which is not too large compared to the set $\mathcal{A}_t$[a].

When the density μ is unimodal (in which case $\mathcal{A}_t$ is an interval), constructing such an interval $\mathcal{I}_t = [L_t, R_t]$ is easy. For instance, this can be achieved by first letting $L_t = Z_{t-1} - w$ and $R_t = Z_{t-1} + w$ for some small $w > 0$, and then, if needed, by progressively increasing R_t until $R_t \notin \mathcal{A}_t$ and progressively reducing L_t until $L_t \notin \mathcal{A}_t$.

However, beyond the case where μ is a unimodal density, sampling from $\mathcal{U}(\mathcal{A}_t)$ is usually a computationally intractable task.

Fortunately, there exist various variants of Algorithm A5, where its step (iii) is replaced by a tractable procedure. Below we focus on the stepping out and shrinkage procedure.

[a] Indeed, if $\mathcal{I}_t$ is large compared to $\mathcal{A}_t$ then, with high probability, we need to sample a large number of times from the $\mathcal{U}(\mathcal{I}_t)$ distribution before obtaining a realization $\tilde{z}$ of $\tilde{Z}$ such that $\tilde{z} \in \mathcal{A}_t$.

Slice sampler with stepping out and shrinkage procedure

Let $w > 0$. Then, still assuming that $\mathcal{Z} \subseteq \mathbb{R}$, the stepping out and shrinkage procedure is as follows:

1. Let $U \sim \mathcal{U}([0,1])$ and let $L_t = Z_{t-1} - Uw$ and $R_t = L_t + w$. Remark that $[L_t, R_t] \cap \mathcal{A}_t \neq \emptyset$ since $Z_{t-1} \in \mathcal{A}_t$.

2. As long as $L_t \in \mathcal{A}_t$, let $L_t \leftarrow L_t - w$ and, as long as $R_t \in \mathcal{A}_t$, let $R_t \leftarrow R_t + w$. (stepping out step)

3. Sample $\tilde{Z}_t \sim \mathcal{U}([L_t, R_t])$.

4. Then,
   - If $\tilde{Z}_t \in \mathcal{A}_t$ set $Z_t = \tilde{Z}_t$
   - If $\tilde{Z}_t \notin \mathcal{A}_t$ let $L_t = \tilde{Z}_t$ if $\tilde{Z}_t < Z_{t-1}$ and let $R_t = \tilde{Z}_t$ if $\tilde{Z}_t > R_{t-1}$ (shrinkage step), and go back to Step 2.

Remark: The goal of the shrinkage step is to reduce the size of $[L_t, R_t]$ in order to increase the probability that the next sampled value of $\tilde{Z}_t$ belongs to the set $\mathcal{A}_t$.

Remark: The proof that Algorithm A5 still has μ as invariant distribution when its step (iii) is replaced by the above procedure can be found [17].

Below we denote by P_η^{slice} the transition kernel of the Markov chain $(Z_t)_{t\geq 0}$ defined by the Slice sampler based on the stepping out and shrinkage procedure and targeting $\eta \in \mathcal{P}(\mathbb{R})$.

The Slice sampling-within-Gibbs algorithm

Since the kernel P_η^{slice} has $\eta \in \mathcal{P}(\mathbb{R})$ as invariant distribution, when $\mathcal{Z}$ is a multivariate state space it can be used within the Hybrid Gibbs sampler Algorithm A4.

More precisely, assume that $\mathcal{Z} = \times_{i=1}^{d} \mathcal{Z}_i$ for some integer $d > 1$ and where $Z_i \subset \mathbb{R}^{k_i}$ for all i. Assume that we can sample from the full conditional distribution $\mu^{(i)}(\cdot|z^{(-i)})$ for $i \leq d_1$ for some $d_1 < d$ and that $k_i = 1$ for all $i > d_1$. Then, the Slice sampling within-Gibbs algorithm is as follows[a].

Slice sampling-within-Gibbs algorithm

(Algorithm A6)

Input: $\mu \in \mathcal{P}(\mathcal{Z})$, $z_0 \in \mathcal{Z}$

(i) Set $Z_0 = z_0$

for $t \geq 1$ **do**

  for $i = 1, \ldots, d_1$ **do**

    (ii) $Z_t^{(i)} \sim \mu^{(i)}(z_t^{(i)}|Z_t^{(1:i-1)}, Z_{t-1}^{(i+1:d)})\mathrm{d}z_t$

  end for

  for $i = d_1 + 1, \ldots, d$ **do**

    (iii) $Z_t^{(i)} \sim P^{\text{slice}}_{\mu^{(i)}(\cdot|Z_t^{(1:i-1)}, Z_{t-1}^{(i+1:d)})}(Z_{t-1}^{(i)}, \mathrm{d}z_t^{(i)})$

  end for

end for

Remark that Algorithm A6 depends on a single tuning parameter, namely the step-size $w > 0$ used by the stepping out and shrinkage procedure, and is therefore easy to tune. This feature of Algorithm A6 explains why, as mentioned earlier, it is the default sampler used by the software JAGS to approximate a distribution μ.

[a] With obvious convention when $d_1 = 0$.

Hamiltonian Monte Carlo (HMC)

Let $\boldsymbol{M}$ be a symmetric positive-define matrix and μ_V be the density of the $\mathcal{N}_d(0, \boldsymbol{M}^{-1})$ distribution[a].

Then, the Hamiltonian (or hybrid) Monte Carlo is a M-H algorithm that as the particularity

1. To target the $2d$-dimensional distribution

$$\tilde{\mu}(\mathrm{d}(z, v)) := \mu(\mathrm{d}z)\mu_V(\mathrm{d}v).$$

2. To use the Hamiltonian dynamic to generate the proposed values $(\tilde{Z}_t, \tilde{V}_t)$.

Expanding on this latter point, in HMC the Hamiltonian dynamic is used to determine the time evolution of the pair (z, v), where z is interpreted as the position and v as the velocity.

A Hamiltonian system is fully characterized by a function $H(z, v)$, called the Hamiltonian, and that if $r_\tau := (z_\tau, v_\tau)$ denotes the position of the system at time $\tau \geq 0$ then r_τ evolves over time according to

$$\frac{\mathrm{d}z_\tau}{\mathrm{d}\tau} = \frac{\partial H(z_\tau, v)}{\partial v}\Big|_{v=v_\tau}, \quad \frac{\mathrm{d}v_\tau}{\mathrm{d}\tau} = -\frac{\partial H(z, v_\tau)}{\partial z}\Big|_{z=z_\tau}. \tag{14.1}$$

[a] We assume henceforth that $\mathcal{Z} \subseteq \mathbb{R}^d$.

Hamiltonian dynamic used in HMC, and its three key properties

The Hamiltonian dynamic on which HMC relies to generate the proposed values $(\tilde{Z}_t, \tilde{V}_t)$ uses the Hamiltonian defined by

$$H(z, v) = -\log \mu(z) - \log \mu_V(v), \quad (z, v) \in \mathcal{Z} \times \mathbb{R}^d$$

so that, using (14.1),

$$\frac{\mathrm{d}z_\tau}{\mathrm{d}\tau} = \boldsymbol{M}^{-1} v_\tau, \quad \frac{\mathrm{d}v_\tau}{\mathrm{d}\tau} = \frac{\partial \log \mu(z)}{\partial z}\Big|_{z=z_\tau}. \tag{14.2}$$

For every $s > 0$ we let $T_s : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ be the mapping from the state at time τ to the state at time $\tau + s$, i.e. T_s is such that

$$\big(z_{\tau+s}, v_{\tau+s}\big) = T_s\big(z_\tau, v_\tau\big), \quad \forall \tau \geq 0.$$

Then, the Hamiltonian dynamic (14.2) has the following key properties[a]:

P1: Conservation of the Hamiltonian. Since

$$\frac{\mathrm{d}H(z_\tau, v_\tau)}{\mathrm{d}\tau} = \sum_{j=1}^{d} \left(\frac{\partial H}{\partial v_j} \frac{\mathrm{d}v_j}{\mathrm{d}\tau} + \frac{\partial H}{\partial z_j} \frac{\mathrm{d}x_i}{\mathrm{d}\tau} \right) = \sum_{j=1}^{d} \left(\frac{\mathrm{d}z_j}{\mathrm{d}\tau} \frac{\mathrm{d}v_j}{\mathrm{d}\tau} - \frac{\mathrm{d}v_j}{\mathrm{d}\tau} \frac{\mathrm{d}z_j}{\mathrm{d}\tau} \right) = 0$$

it follows that $H(z_\tau, v_\tau) = H(z_0, v_0)$ for all $\tau \geq 0$.

P2: Reversibility. $T_s\big(z_{\tau+s}, -v_{\tau+s}\big) = (z_\tau, -v_\tau)$ for all $\tau \geq 0$.

P3: Volume preservation. The absolute value of the Jacobian determinant of T_s is one.

[a] See [34] for the last two properties.

The ideal HMC

Consider the following algorithm for approximating μ.

Idealized HMC algorithm (Algorithm A7)

Input: $\mu \in \mathcal{P}(\mathcal{Z})$, $z_0 \in \mathcal{Z}$, real number $s > 0$ and a symmetric positive definite matrix $\boldsymbol{M}$

(i) Set $Z_0 = z_0$

for $t \geq 1$ **do**

(ii) Sample $\xi_t \sim \mathcal{N}_d(0, \boldsymbol{M}^{-1})$

(iii) Compute $(Z_t, V_t) = T_s(Z_{t-1}, \xi_t)$

end for

We then have the following result for Algorithm A7.

Theorem 14.2 *Let P^{HMC} be the Markov kernel defined by Algorithm A7. Then, P^{HMC} has $\tilde{\mu}$ as invariant distribution.*

Remark: A direct consequence of Theorem 14.2 is that the mapping T_s preserves $\tilde{\mu}$, in the sense that if $(Z, V) \sim \tilde{\mu}$ then $T_s(Z, V) \sim \tilde{\mu}$.

Proof of Theorem 14.2

Let $(Z, V) \sim \tilde{\mu}$, $(Z', V') = T_s(Z, V)$ and remark that because Z and V are independent random variables and $V \sim \mathcal{N}_d(0, \boldsymbol{M}^{-1})$, to prove the theorem it suffices to show that $(Z', V') \sim \tilde{\mu}$.

To this aim, remark that due to the reversibility property of the Hamiltonian dynamic, the mapping T_s is invertible. In addition,

$$\det\big(\boldsymbol{J}_{T_s^{-1}}(z', v')\big) = \det\big(\boldsymbol{J}_{T_s}^{-1}(z', v')\big) = \Big(\det\big(\boldsymbol{J}_{T_s}(z', v')\big)\Big)^{-1} \in \{-1, 1\} \tag{14.3}$$

where the first equality uses the inverse function theorem and the last equality the volume preservation property of the Hamiltonian dynamic.

Therefore, using the change of variable formula, the distribution of (Z', V') has a density $p(\cdot)$ such that, for all $(z', v') \in \mathbb{R}^{d2}$ and with $(z, v) = T_s^{-1}(z', v')$,

$$\begin{aligned} p(z', v') &= \left|\det\big(\boldsymbol{J}_{T_s^{-1}}(z', v')\big)\right| \tilde{\mu} \circ T_s^{-1}(z', v') \\ &= \tilde{\mu} \circ T_s^{-1}(z', v') \\ &\propto e^{-H \circ T_s^{-1}(z', v')} \\ &= e^{-H(z, v)} \\ &= e^{-H(z', v')} \end{aligned} \tag{14.4}$$

where the second equality uses (14.3) and the fourth equality uses conservation property of the Hamiltonian dynamic.

Hence, by (14.4) we have

$$p(z', v') = \tilde{\mu}(z', v'), \quad \forall (z', v') \in \mathbb{R}^{2d}$$

and the proof is complete. □

Two comments on Algorithm A7

1. Let $\big((Z_\tau^c, V_\tau^c)\big)_{\tau\in[0,\infty)}$ be the continuous time version of the Markov chain $\big((Z_t, V_t)\big)_{t\geq 0}$ defined in Algorithm A7, that is, let

$$(Z_t^c, V_t^c) = (Z_t, V_t), \quad \forall t \in \mathbb{N}_0$$

and let

$$(Z_\tau^c, V_\tau^c) = T_{\tau-t}(Z_t, V_t), \quad \forall \tau \in (t, t+1), \quad \forall t \in \mathbb{N}_0.$$

Then, in Algorithm A7, the value of the Hamiltonian $H(Z_\tau^c, V_\tau^c)$ is "refreshed" at times $\tau \in \{t,\, t \in \mathbb{N}\}$ by sampling a new velocity v from π_V, and then remains constant on intervals of the form $(t, t+1)$.

Hence, for $\tau \in (t, t+1)$ the process (Z_τ^c, V_τ^c) moves along a given level curve of the extended target density $\tilde{\mu}(\cdot)$ and, at time $t+1$, it jumps to another level curve of $\tilde{\mu}(\cdot)$.

2. In practice, the mapping T_s is not computable and therefore needs to be approximated. Such an approximation is done using a discrete time approximation of the Hamiltonian dynamic (14.2), that we discuss below.

Remark: The fact that the mapping T_s is intractable explains why we refer to Algorithm A7 as the idealized HMC algorithm.

The Leapfrog method for approximating T_s[a]

For $\epsilon > 0$ let $F_\epsilon : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ be such that $F_\epsilon(z, v) = (z', v')$ with

$$z' = z + \epsilon \boldsymbol{M}^{-1}\big(v + \frac{\epsilon}{2}\nabla \log \mu(z)\big) \tag{14.5}$$

$$v' = v + \frac{\epsilon}{2}\nabla \log \mu(z) + \frac{\epsilon}{2}\nabla \log \mu(z'). \tag{14.6}$$

Then, for a discretization time step $\epsilon > 0$ and an integer $L \geq 1$, we let $T_{L_s,\epsilon} : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ be defined by

$$T_{L,\epsilon}(z, v) = \underbrace{F_\epsilon \circ \ldots F_\epsilon}_{L \text{ times}}(z, v), \quad (z, v) \in \mathbb{R}^{2d}.$$

For a given a trajectory length $s > 0$ let $L_s = \lfloor s/\epsilon \rfloor$, so that $T_{L_s,\epsilon}$ provides an approximation of the mapping T_s.

It can be shown that, as the mapping T_s, the mapping $T_{L_s,\epsilon}$ is reversible and the absolute value of its Jacobian determinant is one.

However, $T_{L_s,\epsilon}$ does not preserve the Hamiltonian and therefore, simply replacing T_s by $T_{L_s,\epsilon}$ in Algorithm A7, does not define a valid MCMC algorithm.

The Hamiltonian Monte Carlo algorithm is obtained (i) by replacing T_s by $T_{L_s,\epsilon}$ in Algorithm A7 and (ii) by adding a Metropolis-type "correction" step which ensures the validity of the algorithm for approximating μ.

[a] This description of the Leapfrog method is from [37].

The Hamiltonian Monte Carlo algorithm

The HMC algorithm (Algorithm A8)

Input: $\mu \in \mathcal{P}(\mathcal{Z})$, $z_0 \in \mathcal{Z}$, real numbers $\epsilon > 0$ and $s > 0$, and a symmetric positive definite matrix $\boldsymbol{M}$

(i) Set $Z_0 = z_0$

for $t \geq 1$ **do**

(ii) $\xi_t \sim \mathcal{N}_d(0, \boldsymbol{M}^{-1})$.

(iii) Compute $(\tilde{Z}_t, V_t') = T_{L_s,\epsilon}(Z_{t-1}, \xi_t)$ and $\tilde{V}_t = -V_t'$

(iv) Set $(Z_t, V_t) = (\tilde{Z}_{t-1}, \tilde{V}_t)$ with probability

$$\alpha_t := \min\left\{1, \exp\left(H(Z_{t-1}, \xi_t) - H(\tilde{Z}_{t-1}, \tilde{V}_t)\right)\right\}$$

and $(Z_t, V_t) = (\tilde{Z}_{t-1}, \xi_t)$ with probability $1 - \alpha_t$

end for

A proof that the correction step (iv) ensures that the transition kernel of the Markov chain $\left((Z_t, V_t)\right)_{t\geq 1}$ defined Algorithm A8 has the extended target distribution $\tilde{\mu}(\mathrm{d}(z, v)) = \mu(\mathrm{d}z)\mu_V(\mathrm{d}v)$ as invariant distribution can be found e.g. in [18].

Remark: In practice, we often choose L instead of choosing s and ϵ.

Remark: In Algorithm A8, the proposed value for the velocity is $-V_t'$ and not V_t'. Informally speaking, the negation of the velocity makes the proposal distribution for $(\tilde{Z}_t, \tilde{V}_t)$ symmetrical, as needed for the acceptance probability α_t define in Algorithm A8 to be valid.

Impact of ϵ and s on the behaviour of HMC

It can be shown [13] that there exists a constant $C < \infty$ such that

$$|H(z,v) - H(\tilde{z},\tilde{v})| \leq C\epsilon^2, \quad (\tilde{z},-\tilde{v}) = T_{L_s,\epsilon}\big(z,v\big), \quad \forall (z,c) \in \mathbb{R}^{2d}.$$

Therefore, for a given choice of s and $\boldsymbol{M}$,

- If ϵ is large then $H(Z_{t-1},\xi_t) - H(\tilde{Z}_t,\tilde{V}_t)$ can be small (i.e. very negative), resulting in a low acceptance rate for Algorithm A8.

- If ϵ is small then $H(Z_{t-1},\xi_t) - H(\tilde{Z}_t,\tilde{V}_t) \approx 0$ and the acceptance rate of Algorithm A8 is close to one. However, the computational cost per iteration of the algorithm is large when ϵ is small, since evaluating $T_{L_s,\epsilon}$ requires $L_s = \lfloor s/\epsilon \rfloor$ evaluations of the mapping F_ϵ defined in (14.5).

Some theoretical results [1] suggest that the optimal acceptance rate for Algorithm A8 is around 0.65, and ϵ is often chosen so that the acceptance rate of HMC is close to this number.

To discuss the choice of s recall call that at each iteration of Algorithm A8 the mapping $T_{L_s,\epsilon}$ is used to approximatively trace out a trajectory of length s of the Hamiltonian dynamic.

Then, for a given choice of ϵ and $\boldsymbol{M}$,

- If s is small then $\tilde{Z}_t$ is typically close to Z_{t-1} and Algorithm A8 tends to behave like a random walk M-H algorithm, that is, it to evolve through local proposals.

- If s is large then the cost per iteration of Algorithm A8 is large.

Stan: HMC with no hand tuning parameters

The no-U-turn sampler (NUTS, [9]) is a version of HMC where s is not fixed across the iterations.

More precisely, at each iteration the length s_t of the trajectory of the Hamiltonian dynamic approximatively traced out by the algorithm is chosen so that to maximize a measure of distance between the proposed value $\tilde{Z}_t$ and the current location Z_{t-1}.

Remark: In NUTS, the definition of s_t ensures that the algorithm is still a valid MCMC algorithm for approximating μ.

The software Stan provides an efficient implementation of NUTS, where the first few iterations are used to automatically tune the remaining two parameters, namely ϵ and $\boldsymbol{M}$.

More precisely, during these "warmup iterations",

- the discretization time step ϵ is optimized to match a target value δ for the acceptance rate. By default, $\delta = 0.8$[a]
- a matrix $\boldsymbol{M}$ such that $\boldsymbol{M}^{-1}$ is approximatively equal to the variance of $Z \sim \mu$ is computed.

[a] Remark that this value for the acceptance rate is larger than optimal value of 0.65 mentioned above. However, this latter is for Algorithm A8, where s is constant over time, and not for NUTS.

Using Stan for Gaussian Process regression with a hierarchical prior: The fossil dataset

We let $\{(y_i^0, x_i^0)\}_{i=1}^n$ be the fossil dataset, that we already used in Chapters 8 and 12, and consider the Gaussian process regression model

$$Y_i^0 = f(x_i^0) + \epsilon_i, \quad \epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}_1(0, \lambda) \quad f \sim \text{GP}(0, ak_\gamma) \tag{14.7}$$

where we recall that k_γ denotes the Gaussian kernel with bandwidth parameter $\gamma > 0$.

As discussed in Chapter 11, the performance of Gaussian process regression depends crucially on the choice of the hyperparameters (λ, a, γ).

While, for the same dataset, in Chapter 12 we used an empirical Bayes approach to choose the value of these hyperparameters, we consider here a fully Bayesian approach where we take the following prior distribution for (λ, a, γ):

$$\lambda, a, \gamma \overset{\text{iid}}{\sim} \text{InvGamma}(5, 5).$$

We then use Stan to approximate the posterior distribution $\pi(\lambda, a, \gamma | y_{1:n}^0)$, that is to simulate a trajectory $\{(\lambda_t, a_t, \gamma_t)\}_{t=1}^T$ of a Markov chain having $\pi(\lambda, a, \gamma | y_{1:n}^0)$ as invariant distribution.

We finally estimate the posterior distribution of interest $\pi(f | y_{1:n}^0)$ by

$$\hat{\pi}_T(f | y_{1:n}^0) = \frac{1}{T} \sum_{t=1}^T \pi\big(f | y_{1:n}^0, \lambda_t, a_t, \gamma_t\big)$$

where $\pi\big(f | y_{1:n}^0, \lambda_t, a_t, \gamma_t\big)$ denotes the distribution of f given $y_{1:n}^0$ implied by (14.7) when $(\lambda, a, \gamma) = (\lambda_t, a_t, \gamma_t)$.

Convergence of NUTS: The fossil dataset

Figure 14.3 below shows the ACFs and trace plots of the trajectory $\{(\lambda_t, a_t, \gamma_t)\}_{t=1}^{T}$ simulated by Stan, with $T = 10\,000$.

The results presented in this figure suggest that the Markov chain mixes well and has ACFs that decrease quickly.

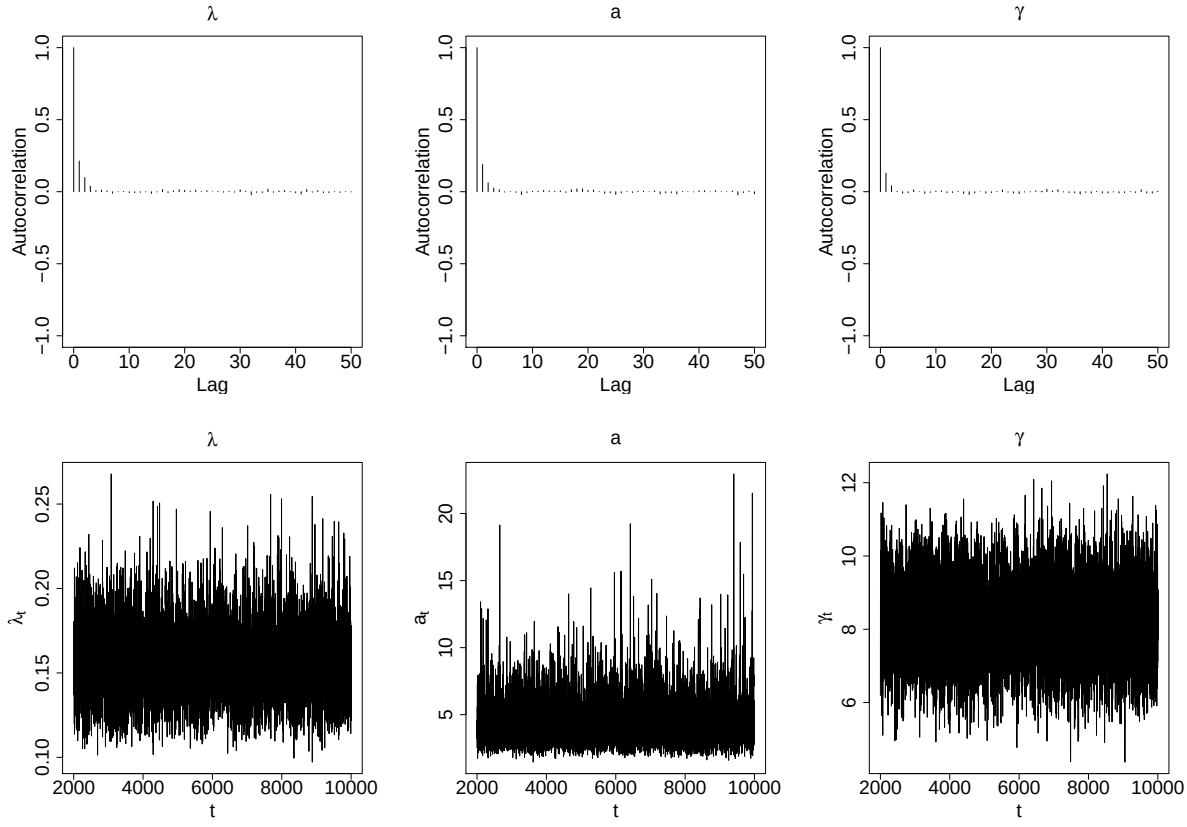

Figure 14.3: ACFs and trace plots for the estimation of the posterior distribution $\pi(\lambda, a, \gamma | y_{1:n}^{0})$ using Stan.

Estimation of the marginal posterior distributions using NUTS: The fossil dataset

Figure 14.5 shows the resulting estimated marginal posterior distribution of the three parameters λ, a and γ.

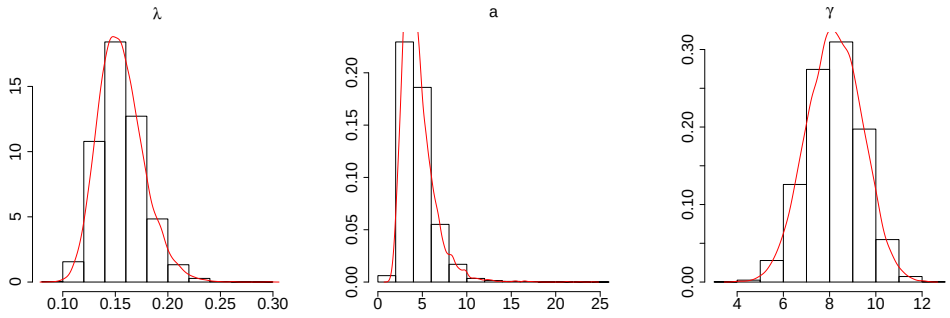

Figure 14.4: NUTS estimate of the marginal distributions for the three model parameters.

Finally, Figure 14.5 below shows, for all i, the NUTS estimated value of $\mathbb{E}[f(x_i^0)|y_{1:n}^0]$ and as well as a 95% credible set for $f(x_i^0)$.

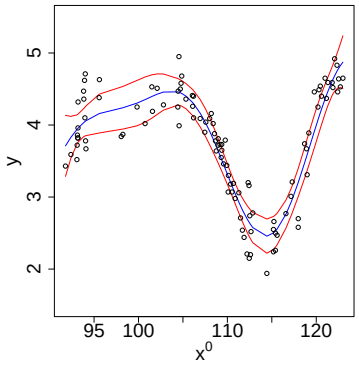

Figure 14.5: Estimated value of $\mathbb{E}[f(x_i^0)|y_{1:n}^0]$ (blue curve) and of a 95% credible set (red curve), obtained from the output of Stan.

Remark: The results in Figure 14.5 are similar to those obtained in Chapter 12 using the empirical Bayes approach (see Figure 12.1).

The Gaussian random walk Metropolis-Hastings versus HMC

In addition to its ease of tuning, HMC has two important advantages over the popular Gaussian random walk Metropolis-Hastings (G-MH) algorithm that we now discuss.

Letting $\mathcal{Z} \subseteq \mathbb{R}^d$, the G-MH algorithm is obtained by letting, in the M-H Algorithm A2,

$$Q(z, \mathrm{d}\tilde{z}) = \mathcal{N}_d(z, c\boldsymbol{\Sigma})$$

for some constant $c > 0$ and for some covariance matrix $\boldsymbol{\Sigma}$.

As mentioned in the previous chapter, the optimal acceptance rate for a M-H algorithm is 0.234. If c is chosen so that the acceptance rate for the G-MH algorithm is around 0.234 then we must have [26]

$$c = \mathcal{O}(1/d),$$

so that $\mathcal{O}(d)$ steps are required by the G-MH algorithm to make $\mathcal{O}_{\mathbb{P}}(1)$ moves in the state space.

More precisely, if $Z_s = \tilde{Z}_s$ for $s = t, \ldots, t + d - 1$ (i.e. if we accept d consecutive proposed values) then, for $c = \mathcal{O}(1/d)$, we have

$$Z_{t+d-1} = Z_{t-1} + c^{1/2} \sum_{s=t}^{t+d-1} \epsilon_s, \quad \epsilon_s \overset{\text{iid}}{\sim} \mathcal{N}_d(0, \Sigma)$$

where $\mathrm{Var}\left(c^{1/2} \sum_{s=t}^{t+d-1} \epsilon_s\right) = dc\boldsymbol{\Sigma} = \mathcal{O}(1)$. In words, in d steps the optimally tuned G-MH algorithm can only make a move of size $\mathcal{O}_{\mathbb{P}}(1)$.

In general, each iteration of the G-MH algorithm costs $\mathcal{O}(d)$ operations, an therefore the overall complexity of the G-MG algorithm is $\mathcal{O}(d^2)$.

HMC in high dimension

To achieve the optimal acceptance rate of 0.65, in the HMC Algorithm A8 we must have [1]

$$\epsilon = \mathcal{O}(d^{-1/4}).$$

Then, for a given value of s, we have

$$L_s = \lfloor s/\epsilon \rfloor = \mathcal{O}(d^{1/4})$$

and therefore evaluating the mapping $T_{L_s,\epsilon}$ requires to evaluate $\mathcal{O}(d^{1/4})$ times the mapping F_ϵ defined in (14.5).

Since each evaluation of the mapping F_ϵ requires $\mathcal{O}(d)$ operations, for a given value of s the overall complexity of HMC is of size $\mathcal{O}(d^{5/4})$.

$\Longrightarrow$ Important improvement of HMC over the $\mathcal{O}(d^2)$ complexity of the G-MH algorithm.

A second advantage of HMC over the G-MH algorithm

The covariance matrix $\mathbf{\Sigma}$ used by the G-MH algorithm should (to some extend) match the variance of $Z \sim \mu$ in order to sample the proposed values $\tilde{Z}_t$'s in the "right" direction.

To illustrate this point Figure 14.7 shows the output of the G-MH algorithm targeting $\mu = \mathcal{N}_2\big(0, \left(\begin{smallmatrix} 1 & 0.98 \\ 0.98 & 1 \end{smallmatrix}\right)\big)$ obtained for $\mathbf{\Sigma} = \left(\begin{smallmatrix} 1 & 0.98 \\ 0.98 & 1 \end{smallmatrix}\right)$ (left plot) and for $\mathbf{\Sigma} = \left(\begin{smallmatrix} 1 & -0.98 \\ -0.98 & 1 \end{smallmatrix}\right)$ (right plot).

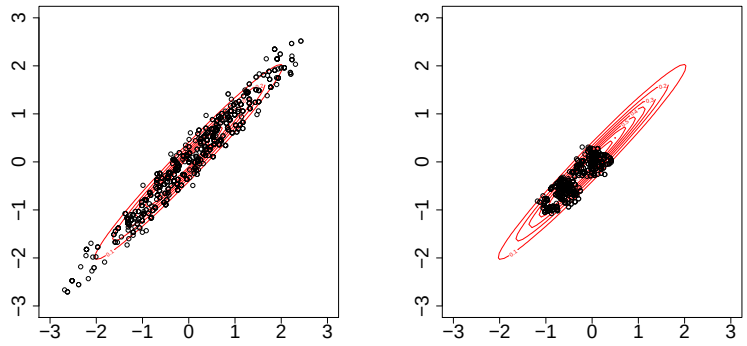

Figure 14.6: Impact of $\mathbf{\Sigma}$ on the performance of the G-MH algorithm. In the two plots c is chosen so that the acceptance rate of the algorithm is approximatively equal to 0.234.

A second advantage of HMC over the G-MH algorithm (end)

For $d \geq 2$ the "right" direction may vary across the state space, in which case the G-MH algorithm may performs poorly, regardless of the choice of $\boldsymbol{\Sigma}$.

This point illustrated in Figure 14.7 below, where μ is a two dimensional "banana shaped" distribution and $\boldsymbol{\Sigma} = c\boldsymbol{I}_2$. We observe in Figure 14.7 that, with $T = 1\,000$ iterations, G-MH explores well the middle of the distribution and visits its left tail. However, the algorithm fails to explores the right tails of μ.

For the same target distribution and number T of iterations, in the right plot of Figure 14.7 we show the sample $\{z_t\}_{t=1}^{T}$ obtained with HMC (Algorithm A8). We remark that, unlike G-MH, with only $T = 1\,000$ iterations HMC manages to explore well the two tails of μ, and to generate a sample that represents well the target distribution.

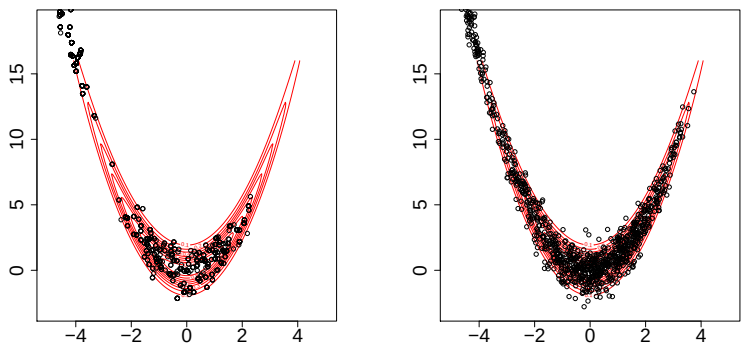

Figure 14.7: Output of G-MH algorithm (left plot) and of HMC (right plot) for a banana shape target distribution μ. The acceptance rate of G-MH is around 0.234 and that of HMC is close to one. The number of iteration is $T = 1\,000$ for the two algorithms.

Chapter 15: Hierarchical Modeling[a]

In this chapter we consider data points $\{(y_i^0, x_i^0)\}_{i=1}^n$ in $\mathcal{Y} \times \mathbb{R}^p$ and we are interested in building (and estimating) a Bayesian model for the distribution of Y_i^0 given X_i^0.

In Chapters 6-12 we assumed that Y_i^0 was conditionally independent of $(Y_{1:(i-1)}^0, X_{1:n}^0)$ given X_i^0, that is, we assumed that the joint conditional distribution of $Y_{1:n}^0$ given $X_{1:n}^0$ can be written as follows:

$$p(y_{1:n}^0 | x_{1:n}^0) = \prod_{i=1}^n p(y_i^0 | x_i^0). \tag{15.1}$$

In this chapter, we instead consider observations $\{(y_i^0, x_i^0)\}_{i=1}^n$ that are structured in groups belonging to a given global population, in which case the decomposition (15.1) of $p(y_{1:n}^0 | x_{1:n}^0)$ is not a reasonable assumption.

As we will see in this chapter, adopting a Bayesian approach allows to easily construct models for such observations, which can often be efficiently estimated using a Gibbs sampler (Algorithm A3) or a Hybrid Gibbs sampler (Algorithm A4).

We assume below that the observations belong $\{(y_i^0, x_i^0)\}_{i=1}^n$ to K (known) groups. For $k \in K$ we let $I_k \subset \{1, \ldots, n\}$ be such that $i \in I_k$ if and only if the observation (y_i^0, x_i^0) belongs to group k, and we let $y_{I_k}^0 = (y_i^0,\ i \in I_k)$ and $x_{I_k}^0 = (x_i^0,\ i \in I_k)$.

[a] The main reference for this chapter is [6, Part 2A].

A naive Bayesian approach for modelling data structured in groups

Consider the following strategy for modelling the observations:

- For all k choose a parametric model $\mathcal{M}_k := \{f_{k,\theta^{(k)}}(\cdot|\cdot),\ \theta^{(k)} \in \Theta_k\}$ for the distribution of $Y_i^0|X_i^0$ when $i \in I_k$.

- For all k choose a prior distribution $\pi_k(\theta^{(k)})$ for $\theta^{(k)}$.

- For all k compute the posterior distribution

$$\pi_k(\theta^{(k)}|y_{I_k}^0, x_{I_k}^0) \propto \pi_k(\theta^{(k)}) \prod_{i \in I_k} f_{k,\theta^{(k)}}(y_i^0|x_i^0).$$

While this approach has the advantage to be simple, it suffers from two important problems:

1. The number of observations $|I_k|$ in group k may be small, in which case the posterior distribution $\pi_k(\theta^{(k)}|y_{I_k}^0, x_{I_k}^0)$ will be essentially equal to the prior distribution $\pi_k(\theta^{(k)})$.

2. It does not provide a way to use the observations $\{(y_i^0, x_i^0)\}_{i=1}^n$ to predict an observation belonging to a group $k_* \notin \{1, \ldots, K\}$.

These two issues arise because the above "naive" approach for modelling the data does not take into account the existence of between group interactions, which are due to the fact that the different groups belong to the same population.

The hierarchical (or multilevel) approach

As above, let $\mathcal{M}_k := \{f_{k,\theta^{(k)}}(\cdot|\cdot),\ \theta^{(k)} \in \Theta_k\}$ be a parametric model for the distribution of $Y_i^0|X_i^0$ when $i \in I_k$.

Then, in the hierarchical approach, the dependence between the groups is incorporated in the statistical analysis through the prior distribution.

To illustrate this idea we consider the following simple hierarchical model:

$$\begin{aligned} &Y_i^0|(\theta^{(k)}, X_i^0) \sim f_{k,\theta^{(k)}}(y_i^0|X_i^0), \quad i \in I_k, \quad k = 1, \ldots, K \\ &\theta^{(k)}|\psi \sim \pi_{k,\psi}(\theta^{(k)}), \qquad\qquad k = 1, \ldots, K \\ &\psi \sim \pi(\psi). \end{aligned} \tag{15.2}$$

In model (15.2) the dependence between the groups is incorporated by assuming that the prior distribution of the $\theta^{(k)}$'s depend on a common hyperparameter ψ.

Remark: The $\theta^{(k)}$'s can be interpreted as the group specific parameter and ψ as the population parameter.

Remark: Model (15.2) can be easily generalized to situations where the data are structured in several levels (e.g. because each group contains several sub-groups).

Remark: The distribution $\pi_{k,\psi}$ may depend on some group level information z_k.

Completing the hierarchical model (15.2)

Completing the model (15.2) requires to choose $\pi(\psi)$ and

$$\mathcal{M}_k, \quad \mathcal{M}'_k := \{\pi_{k,\psi},\, \psi \in \Psi\}, \quad \forall k \in \{1, \ldots, K\}.$$

In practice, to simplify the modelling process we often assume that $\mathcal{M}_k = \mathcal{M}_1$ and that $\mathcal{M}'_k = \mathcal{M}'_1$ for all k.

In this case, independently of the number K of groups, to complete the model (15.2) we only need to specify three (family of) distributions, namely

$$\{f_{1,\theta},\, \theta \in \Theta_1\}, \quad \{\pi_{1,\psi},\, \psi \in \Psi\}, \quad \pi. \tag{15.3}$$

$\Longrightarrow$ This is the main reason why hierarchical models are (very) convenient to model data structured in groups.

The distributions in (15.3) are often chosen with the aim of

- Facilitating the MCMC approximation of the posterior distribution $\pi(\theta^{(1)}, \ldots, \theta^{(K)}, \psi | y^0_{1:n}, x^0_{1:n})$.

- Making the model predictions easy to compute from an MCMC approximation of $\pi(\theta^{(1)}, \ldots, \theta^{(K)}, \psi | y^0_{1:n}, x^0_{1:n})$

Below we start by discussing this latter point.

Prediction of a new observation in an existing group

We assume that we observe the predictor variable x_*^0 for a new observation (Y_*^0, X_*^0) belonging to group $k_* \in \{1, \ldots, K\}$, and consider the problem of predicting Y_*^0.

To this end we first extend the model (15.2) to include the new observation (Y_*^0, X_*^0):

$$\begin{aligned} &Y_i^0|(\theta^{(k)}, X_i^0) \sim f_{k,\theta^{(k)}}(y_i^0|X_i), \quad i \in I_k, \quad k = 1, \ldots, K \\ &Y_*^0|(\theta^{(k_*)}, X_*^0) \sim f_{k_*,\theta^{(k_*)}}(y_*^0|X_*^0), \\ &\theta^{(k)}|\psi \sim \pi_{k,\psi}(\theta^{(k)}), \qquad\qquad\qquad k = 1, \ldots, K \\ &\psi \sim \pi(\psi). \end{aligned} \tag{15.4}$$

Then, as usual in Bayesian statistics, the inference on Y_*^0 is based on $\pi(y_*^0|d_n, x_*^0)$, the posterior distribution of Y_*^0 given the available observations $d_n := \{(y_i^0, x_i^0)\}_{i=1}^n$ and x_*^0.

Remark: $\pi(y_*^0|d_n, x_*^0)$ is called the predictive distribution.

Under the model (15.4), we have

$$\begin{aligned} \pi(y_*^0|d_n, x_*^0) &= \int_{\Theta_{k_*}} \pi(y_*^0, \theta^{(k_*)}|d_n, x_*^0) \mathrm{d}\theta^{(k_*)} \\ &= \int_{\Theta_{k_*}} \pi(y_*^0|\theta^{(k_*)}, d_n, x_*^0) \pi(\theta^{(k_*)}|d_n, x_*^0) \mathrm{d}\theta^{(k_*)} \\ &= \int_{\Theta_{k_*}} f_{k_*,\theta^{(k_*)}}(y_*^0|x_*^0) \pi(\theta^{(k_*)}|d_n) \mathrm{d}\theta^{(k_*)}. \end{aligned} \tag{15.5}$$

Bayesian predictor for an observation in an existing group

Under a quadratic loss function, the Bayes estimator $f_{k_*}(x^*)$ of Y_*^0 is the posterior mean of Y_*^0, that is,

$$f_{k_*}(x_*^0) = \int_{\mathcal{Y}} y_*^0 \pi(y_*^0|d_n, x_*^0)\mathrm{d}y_*^0.$$

For $k \in \{1, \ldots, K\}$ let $m_k : \mathbb{R}^p \times \Theta \to \mathcal{Y}$ be defined by

$$m_k(x, \theta^{(k)}) = \int_{\mathcal{Y}} y\, f_{k,\theta^{(k)}}(y|x)\mathrm{d}y, \quad (x, \theta^{(k)}) \in \mathbb{R}^p \times \Theta_k. \tag{15.6}$$

Then, using (15.5),

$$\begin{aligned} f_{k_*}(x_*^0) &= \int_{\mathcal{Y}} y_*^0 \left(\int_{\Theta_{k_*}} f_{k_*,\theta^{(k_*)}}(y_*^0|x_*^0)\pi(\theta^{(k_*)}|d_n)\mathrm{d}\theta^{(k_*)} \right) \mathrm{d}y_*^0 \\ &= \int_{\Theta_{k_*}} \left(\int_{\mathcal{Y}} y_*^0 f_{k_*,\theta^{(k_*)}}(y_*^0|x_*^0)\mathrm{d}y_*^0 \right) \pi(\theta^{(k_*)}|d_n)\mathrm{d}\theta^{(k_*)} \\ &= \int_{\Theta} m_{k_*}(x_*^0, \theta^{(k_*)})\pi(\theta^{(k_*^0)}|d_n)\mathrm{d}\theta^{(k_*)}. \end{aligned}$$

Consequently, if $\{f_{k_*,\theta},\, \theta \in \Theta_{k_*}\}$ is chosen in such a way that the function m_{k_*} can be evaluated pointwise, given an MCMC sample $\{\theta_t^{(k_*)}\}_{t=1}^T$ that approximates $\pi(\theta^{(k_*)}|d_n)$ we can easily approximate the Bayesian prediction $f_{k_*}(x_*^0)$ of Y_*^0 using

$$f_{T,k_*}(x_*^0) := \frac{1}{T}\sum_{t=1}^{T} m_{k_*}\big(x_*^0, \theta_t^{(k_*)}\big).$$

Remark: The function $f_{T,k} : \mathbb{R}^p \to \mathcal{Y}$ is our estimated Bayesian predictor function for an observation in group $k \in \{1, \ldots, K\}$.

Some examples

Example 1 (continuous response variables)

Assume that $\mathcal{Y} = \mathbb{R}$ and let $\mathcal{M}_k = \mathcal{M}_1$ for all k, where $f_{1,\theta}(\cdot|x)$ is the density of the $\mathcal{N}_1(x^\top\beta, \sigma^2)$ distribution for all $x \in \mathbb{R}^p$ and all $\theta = (\beta, \sigma^2) \in \Theta_1 := \mathbb{R}^p \times (0, \infty)$.

Then, for all $k \in \{1, \ldots, K\}$ and $x \in \mathbb{R}^p$, we have

$$f_{T,k}(x) = x^\top \Big(\frac{1}{T} \sum_{t=1}^{T} \beta_t^{(k)} \Big).$$

Example 2 (discrete response variables)

Assume that $\mathcal{Y} = \mathbb{N}_0$ and let $\mathcal{M}_k = \mathcal{M}_1$ for all k, where $f_{1,\theta}(\cdot|x)$ is the density of the Poisson$(\exp(x^\top\theta))$ distribution for all $\theta \in \Theta_1 := \mathbb{R}^p$ and all $x \in \mathbb{R}^p$.

Then, for all $k \in \{1, \ldots, K\}$ and $x \in \mathbb{R}^p$, we have

$$f_{T,k}(x) = \frac{1}{T} \sum_{t=1}^{T} \exp\big(x^\top \theta_t^{(k)}\big). \tag{15.7}$$

Example 3 (binary response variables)

Assume that $\mathcal{Y} = \{0, 1\}$ and let $\mathcal{M}_k = \mathcal{M}_1$ for all k, where $f_{1,\theta}(\cdot|x)$ is the density of the Bernoulli$\big(\Phi(x^\top\theta)\big)$ distribution for all $x \in \mathbb{R}^p$ and all $\theta \in \Theta_1 := \mathbb{R}^p$. Then, for all $k \in \{1, \ldots, K\}$ and $x \in \mathbb{R}^p$, we have

$$f_{T,k}(x) = \frac{1}{T} \sum_{t=1}^{T} \Phi\big(x^\top \theta_t^{(k)}\big).$$

Prediction of a new observation belonging to a new group

We now assume that we observe the predictor variable x_*^0 for a new observation (Y_*^0, X_*^0) belonging to a new group $k_* \notin \{1, \ldots, K\}$, and consider the problem of predicting Y_*^0.

Remark: The sample $\{(y_i^0, x_i^0)\}_{i=1}^n$ contains no observation for the group k_*.

To this end we first extend the model (15.2) to include the new group k_* and the new observation (Y_*^0, X_*^0):

$$\begin{aligned}
&Y_i^0|(\theta^{(k)}, X_i^0) \sim f_{k,\theta^{(k)}}(y_k^0|x_i^0), \quad i \in I_k, \qquad k = 1, \ldots, K \\
&Y_*^0|(\theta^{(k_*)}, X_*^0) \sim f_{k_*,\theta^{(k_*)}}(y_*^0|X_*^0) \\
&\theta^{(k)}|\psi \sim \pi_{k,\psi}(\theta^{(k)}), \qquad\qquad\qquad k \in \{1, \ldots, K\} \cup \{k_*\} \\
&\psi \sim \pi(\psi).
\end{aligned} \tag{15.8}$$

For all $k \in \{1, \ldots, K\} \cup \{k_*\}$ let $\tilde{m}_k : \mathbb{R}^p \times \Psi \to \mathcal{Y}$ be the function defined by

$$\tilde{m}_k(x, \psi) = \int_{\Theta_k} m_k(x, \theta^{(k)}) \pi_{k,\psi}(\theta^{(k)}) \mathrm{d}\theta^{(k)}, \quad (x, \psi) \in \mathbb{R}^p \times \Psi$$

where m_k is as defined in (15.6).

Bayesian predictor for an observation in a new group

Then, under the model (15.8), the predictive distribution for Y_*^0 is

$$\begin{aligned}
&\pi(y_*^0|d_n, x_*^0) \\
&\quad = \int_{\Theta_{k^*}\times\Psi} \pi\big(y_*^0, \theta^{(k_*)}, \psi|d_n, x_*^0\big)\mathrm{d}(\theta^{(k_*)}, \psi) \\
&\quad = \int_{\Theta_{k^*}\times\Psi} \pi(y_*^0|\theta^{(k_*)},\, \psi,\, d_n, x_*^0)\pi(\theta^{(k_*)}|\psi,\, d_n, x_*^0)\pi(\psi|d_n, x_*^0)\mathrm{d}(\theta^{k_*}, \psi) \\
&\quad = \int_{\Theta_{k^*}\times\Psi} f_{k_*,\theta^{(k^*)}}(y_*^0|x_*^0)\pi_{k_*,\psi}(\theta^{(k_*)})\pi(\psi|d_n)\mathrm{d}(\theta^{(k_*)}, \psi).
\end{aligned}$$

Under a quadratic loss function, the Bayes estimator $\tilde{f}_{k_*}(x_*^0)$ of Y_*^0 is the posterior mean, that is,

$$\begin{aligned}
\tilde{f}_{k_*}(x_*^0) &= \int_{\mathcal{Y}} y_*^0\pi(y_*^0|d_n, x_*^0)\mathrm{d}y_*^0 \\
&= \int_{\Psi}\left[\int_{\Theta_{k^*}}\left(\int_{\mathcal{Y}} y_*^0 f_{k_*,\theta^{(k_*)}}(y_*^0|x_*^0)\mathrm{d}y_*^0\right)\pi_{k_*,\psi}(\theta^{(k_*)})\mathrm{d}\theta^{(k_*)}\right]\pi(\psi|d_n)\mathrm{d}\psi \\
&= \int_{\Psi}\tilde{m}_{k_*}(x_*^0, \psi)\pi(\psi|d_n)\mathrm{d}\psi.
\end{aligned}$$

Consequently, if $\{f_{k_*,\theta},\, \theta \in \Theta_{k_*}\}$ and $\{\pi_{k_*,\psi},\, \psi \in \Psi\}$ are chosen so that the function $\tilde{m}_{k_*}$ can be evaluated pointwise then, given an MCMC sample $\{\psi_t\}_{t=1}^T$ that approximates the posterior distribution $\pi(\psi|d_n)$, we can easily compute an approximation of $\tilde{f}_{k_*}(x_*^0)$ using

$$\tilde{f}_{T,k_*}(x_*^0) := \frac{1}{T}\sum_{t=1}^{T}\tilde{m}_{k_*}(x_*^0, \psi_t).$$

Remark: The function $\tilde{f}_{T,k_*} : \mathbb{R}^p \to \mathcal{Y}$ is our estimated Bayesian predictor function for an observation in group k_* for which no observation is available.

Some examples

Example 1 (continued)

For all k let $\pi_{k,\psi}(\beta, \sigma^2)$ be such that $\pi_{k,\psi}(\cdot|\sigma^2)$ is the density of the $\mathcal{N}_d(\boldsymbol{M} z_k, \sigma^2 \boldsymbol{\Sigma})$ distribution, where $z_k \in \mathbb{R}^m$ is a vector that contains some information about group k, where $\boldsymbol{M} \in \mathbb{R}^{p \times m}$, where $\boldsymbol{\Sigma}$ is $p \times p$ covariance matrix and where $\boldsymbol{M}$ and $\boldsymbol{\Sigma}$ are elements of ψ.

Then, for all $x \in \mathbb{R}^p$, we have

$$\tilde{f}_{T,k_*}(x) = x^\top \Big(\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{M}_t z_{k_*} \Big).$$

Example 2 (continued)

For all k let $\pi_{k,\psi}(\cdot)$ be the density of the $\mathcal{N}_d(\boldsymbol{M} z_k, \boldsymbol{\Sigma})$ distribution where z_k, $\boldsymbol{M}$ and $\boldsymbol{\Sigma}$ are as in Example 1, and where $\boldsymbol{M}$ and $\boldsymbol{\Sigma}$ are elements of ψ.

Then, for all $x \in \mathbb{R}^p$, we have

$$\tilde{f}_{T,k_*}(x) = \frac{1}{T} \sum_{t=1}^{T} \exp\Big(\frac{x^\top \boldsymbol{\Sigma}_s \, x + 2 x^\top \boldsymbol{M}_t z_{k_*}}{2} \Big). \tag{15.9}$$

Example 3 (continued)

Let $\pi_{k,\psi}(\cdot)$ be as in Example 2, where $\boldsymbol{M}$ and $\boldsymbol{\Sigma}$ are elements of ψ.

Then, for all $x \in \mathbb{R}^p$, we have

$$\tilde{f}_{T,k_*}(x) = \frac{1}{T} \sum_{t=1}^{T} \Phi\Big(\frac{x^\top \boldsymbol{M}_t z_{k_*}}{\sqrt{1 + x^\top \boldsymbol{\Sigma}_t \, x}} \Big).$$

MCMC estimation of the posterior distribution in the hierarchical model (15.2)

As illustrated with the above three examples, it is often possible to specify the distributions of a hierarchical model in such a way that, once the posterior distribution of its parameters has been estimated (e.g. using MCMC), the model can easily be used to make predictions.

As in the above examples, this is usually achieved by building the hierarchical model using exponential family of distributions (see Chapter 7, page 173, for a definition) such that that the family of distributions used in layer $L \geq 1$ of the hierarchy (i.e. used in the L-th line of (15.2)) is conjugate for the family used in layer $L+1$, in the following sense:

Definition 15.21 *Let $\mathcal{F}$ be a parametric family of distributions on Λ. Then, we say that $\mathcal{F}$ is conjugate for the parametric family of distributions $\{p(\cdot|\lambda),\ \lambda \in \Lambda\}$ of $\mathcal{Z}$ if*

$$q \in \mathcal{F} \Rightarrow \tilde{p}(\cdot|z) := \frac{p(z|\cdot)q(\cdot)}{\int_\Lambda p(z|\lambda)q(\lambda)} \in \mathcal{F}, \quad \forall z \in \mathcal{Z}.$$

It turns out that this strategy for specifying the distributions used in a hierarchical model often leads to a posterior distribution of the model parameter which has many full conditional distributions for which it is easy to sample from.

For this reason, hierarchical model are often estimated using a Gibbs sampler (Algorithm A3) or a hybrid Gibbs sampler (Algorithm A4).

Example 1 (end)

For the choice of distributions already made the model (15.2) becomes:

$$
\begin{aligned}
&Y_i^0|(\beta^{(k)}, \sigma^{2\,(k)}, X_i^0) \sim \mathcal{N}_1(X_i^{0\top}\beta^{(k)}, \sigma^{2\,(k)}), && i \in I_k, \quad k = 1, \ldots, K \\
&\beta^{(k)}|(\sigma^{2\,(k)}, \psi) \sim \mathcal{N}_d(\boldsymbol{M} z_k, \boldsymbol{\Sigma}), && k = 1, \ldots, K \\
&\sigma^{2\,(k)}|\psi \sim \pi_{k,\psi}^{(\sigma^2)}(\sigma^{2\,(k)}), && k = 1, \ldots, K \\
&\psi \sim \pi(\psi)
\end{aligned}
$$

for some distribution $\{\pi_{k,\psi}^{(\sigma^2)}\}_{k=1}^K$ on $(0, \infty)$ and some distribution π on Ψ that need to be specified.

We assume that $\pi_{k,\psi}^{(\sigma^2)} = \pi_{1,\psi}^{(\sigma^2)}$ for all k, where $\pi_{1,\psi}^{(\sigma^2)} = \delta_{\{s^2\}}$ with $s \in (0, \infty)$ and $\psi = (\text{vec}(\boldsymbol{M}), \boldsymbol{\Sigma}, s^2)$. Finally, to complete the model we let

$$\pi(\psi) = \pi_{\boldsymbol{M},\boldsymbol{\Sigma}}\big(\text{vec}(\boldsymbol{M}), \boldsymbol{\Sigma}\big)\pi^{(s)}(s^2)$$

where $\pi_{\boldsymbol{M},\boldsymbol{\Sigma}}(\text{vec}(\boldsymbol{M}), \boldsymbol{\Sigma})$ is the density of a normal-inverse-Wishart distribution and $\pi^{(s)}(s^2)$ the density of an inverse-Gamma distribution.

Then, the resulting hierarchical model is such that

- The full conditional distribution of $\beta^{(k)}$ is a normal distribution, for all k.
- The full conditional distribution of $(\text{vec}(\boldsymbol{M}), \boldsymbol{\Sigma})$ is a normal-inverse-Wishart distribution.
- The full conditional distribution of s^2 is an inverse-Gamma distribution.

Therefore, a Gibbs sampler algorithm can be used to approximate the posterior distribution of the model parameters.

Example 2 (end)

For the choice of distributions that we have already made the model (15.2) becomes:

$$
\begin{aligned}
&Y_i^0|(\theta^{(k)}, X_i^0) \sim \text{Poisson}(\exp({X_i^0}^\top \theta^{(k)}), \quad i \in I_k, \quad k = 1, \ldots, K \\
&\theta^{(k)}|\psi \sim \mathcal{N}_d(\boldsymbol{M} z_k, \boldsymbol{\Sigma}), \qquad\qquad\qquad\quad k = 1, \ldots, K \\
&\psi \sim \pi(\psi)
\end{aligned}
$$

where $\psi = (\text{vec}(\boldsymbol{M}), \boldsymbol{\Sigma})$.

To complete the model we let $\pi(\psi) = \pi(\text{vec}(\boldsymbol{M}), \boldsymbol{\Sigma})$ with $\pi(\text{vec}(\boldsymbol{M}), \boldsymbol{\Sigma})$ as in Example 1.

Then, the resulting hierarchical model is such that

- The full conditional distribution of $(\text{vec}(\boldsymbol{M}), \boldsymbol{\Sigma})$ is a normal-inverse-Wishart distribution.
- The full conditional distribution of $\theta^{(k)}$ does not belong to a well-known family of distributions (and thus cannot be easly sampled from), for all k.

This hierarchical model can for instance be estimated using a Hybrid Gibbs sampler such as, for instance, the Slice sampling-within Gibbs algorithm (Algorithm A6).

Remark: When $\{f_{k,\theta^{(k)}}(\cdot|x),\ \theta^{(k)} \in \Theta_k\}$ is a generalized linear model (GLM), as in this example, a version of the Hybrid Gibbs sampler more efficient than the Slice sampling-within Gibbs algorithm is usually available [21, Section 10.2], as illustrated at the end of this chapter.

Example 3 (continued)

For the choice of distributions that we have already made the model (15.2) becomes:

$$\begin{aligned} &Y_i^0|(\theta^{(k)}, X_i^0) \sim \text{Bern}(\Phi(X_i^{0\top}\theta^{(k)})), \quad i \in I_k,\ k = 1, \ldots, K \\ &\theta^{(k)}|\psi \sim \mathcal{N}_d(\boldsymbol{M}z_k, \boldsymbol{\Sigma}), \qquad\qquad k = 1, \ldots, K \\ &\psi \sim \pi(\psi) \end{aligned} \tag{15.10}$$

where $\psi = (\text{vec}(\boldsymbol{M}), \boldsymbol{\Sigma})$.

Implementing a Gibbs sampler for the model (15.10) is not possible. However, a Gibbs sampler can be implemented for the following equivalent model

$$\begin{aligned} &Y_i^0 = \mathbf{1}_{(0,\infty)}(\tilde{Y}_i^0), \qquad\qquad i = 1, \ldots, n \\ &\tilde{Y}_i^0|(\theta^{(k)}, X_i^0) \sim \mathcal{N}_1\big(X_i^{0\top}\theta^{(k)}, 1\big), \quad i \in I_k, \quad k = 1, \ldots, K \\ &\theta^{(k)}|\psi \sim \mathcal{N}_d(\boldsymbol{M}z_k, \boldsymbol{\Sigma}), \qquad\qquad k = 1, \ldots, K \\ &\psi \sim \pi(\psi). \end{aligned} \tag{15.11}$$

Remark: The model (15.11) treats the latent variables $\{\tilde{Y}_i^0\}_{i=1}^n$ as "parameters".

To complete the model (15.11) we let π be as in Example 2, and the resulting hierarchical model is such that

- The full conditional distribution of $\theta^{(k)}$ is a normal distribution, for all k.
- The full conditional distribution of $\tilde{Y}_i$ is a truncated normal distribution, for all i.
- The full conditional distribution of $(\text{vec}(\boldsymbol{M}), \boldsymbol{\Sigma})$ is a normal-inverse-Wishart distribution.

Therefore, a Gibbs sampler algorithm can be used to approximate the posterior distribution of the model parameters.

Application: The frisks dataset[a]

Background: In the 90's there were complaints in New York City that the police harass members of ethnic minority groups. In 1999 a study of the New York City "stop and frisk" policy (the practice to temporarily detain, question and search civilians on streets) was ordered, resulting in a dataset containing information for all stops over a period of 15 months between 1998 and 1999. This example is based on a noisy version of the dataset (where the noise has been added to protect confidentiality).

The dataset contains the number of persons stopped and frisked by the police, for different ethnic groups and in different precincts.

More precisely, three ethnic groups are considered, namely Blacks, Hispanics and Whites, and data are available for 74 precincts, resulting in a sample of $n = 222$ observations.

In addition, the dataset contains, for each of the 74 precincts and for each of the three ethnic groups, the number of persons arrested the previous year.

Our objective is to build a hierarchical model that can be used to

- Analyse whether or not the police did harass members of ethnic minority groups.
- Predict, for the three ethnic groups, the number of persons stopped and frisked by the police in a precinct s_* for which no observation is available, given that the number of Blacks, Hispanics and Whites arrested the previous year is, respectively, 16, 44 and 312.

[a]This dataset is available at `http://www.stat.columbia.edu/~gelman/arm/`

Statistical model for the data

Let $Y^0_{e,s}$ be the number of persons in ethnic group $e \in \{1,2,3\}$ stopped and frisked in police station $s \in \{1,\dots,75\}$, where $e=1$ for Blacks, $e=2$ for Hispanics and $e=3$ for Whites, and let $X^0_{e,s}$ be the number of persons in ethnic group e that were arrested in police station s the previous year.

Note that in this dataset there are two kinds of groups, since there are $K_{\text{eth}}=3$ ethnic groups and $K_{\text{pol}}=74$ police stations, and that the global population is New York City.

Following [6, Chapter 15] we consider the following model for the data

$$
\begin{aligned}
&Y^0_{e,s}|(\mu,\gamma,\theta^{(e)},\vartheta^{(s)},X^0_{ep}) \sim \text{Poisson}\Big(\frac{15}{12}e^{\mu+\gamma\log X^0_{e,s}+\theta^{(e)}+\vartheta^{(s)}}\Big) && \forall e,s\\
&\theta^{(e)} \sim \mathcal{N}_1(0,\sigma^2_{\text{eth}}), && e=1,\dots,3\\
&\vartheta^{(s)} \sim \mathcal{N}_1(0,\sigma^2_{\text{pol}}), && s=1,\dots,74\\
&\mu,\gamma \overset{\text{iid}}{\sim} \mathcal{N}_1(\mu_0,\sigma^2_0)\\
&\sigma^2_{\text{eth}},\ \sigma^2_{\text{pol}} \overset{\text{iid}}{\sim} \text{InvGamma}(a_0,b_0)
\end{aligned}
$$

where $\mu_0 \in \mathbb{R}$, $\sigma_0>0$, $a_0>0$ and $b_0>0$ are hyperparameters to be chosen.

We let $\mu_0=0$ and, since no prior information is available, $\sigma^2_0=100$ so that we have a vague prior for μ and γ. Lastly, we take $a_0=b_0=0.01$ so that the prior distribution of σ^2_{eth} and of σ^2_{pol} has mean 1 and variance equal to 100 (vague prior).

Remark: The factor 15/12 is used to scale to a period of 15 months.

Remark: The model has 81 unknown parameters.

Estimation of the posterior distribution using JAGS

The model we consider is such that we can only sample from the full conditional distributions of σ^2_{eth} and of σ^2_{pol} (which are both inverse gamma distributions), and below we use a Hybrid Gibbs sampler to approximate the posterior distribution of the parameters.

We first use the Slice sampling-within-Gibbs algorithm (as implemented in JAGS) to approximate the posterior distribution. As illustrated in Figure 15.1, the slice sampling kernel performs poorly in this example, as the ACF of the parameters "updated' by means of the slice sampling kernel decreases very slowly. On the other hand the ACF for the parameters "updated" using a Gibbs step decreases quickly, as illustrated in Figure 15.1.

As mentioned above, for GLM type hierarchical models more efficient kernels than the slice sampling kernel are available. This point is illustrated with the bottom plots of Figure 15.1 where, within the Hybrid Gibbs sampler, JAGS uses a "default GLM" kernel[a].

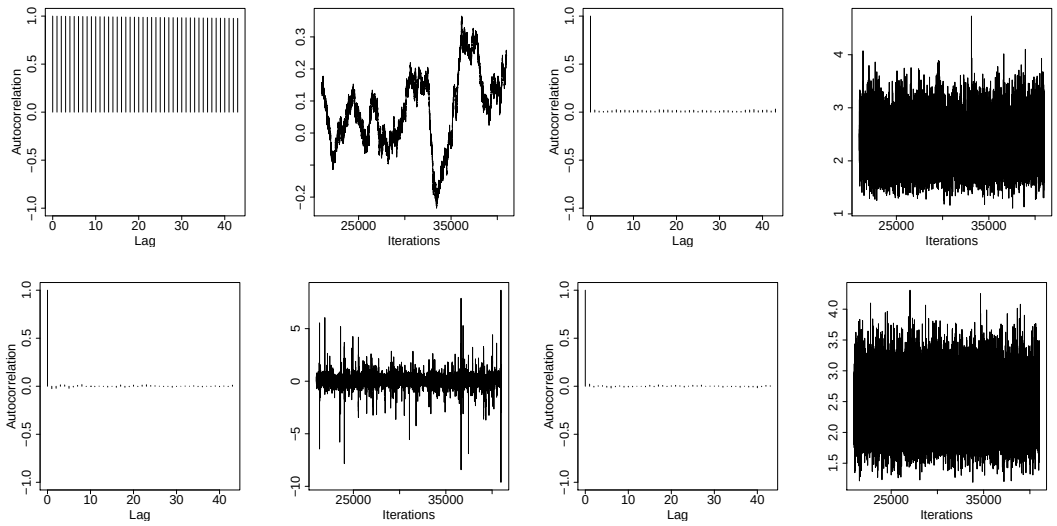

Figure 15.1: Trace plot and ACF for the parameters $\theta^{(1)}$ (left plots) and σ^2_{pol} (right plots) obtained with JAGS. The top plots are for the Hybrid Gibbs sampler with a slice sampling kernel and the bottom plots for a Hybrid Gibbs sampler with a "default GLM" kernel.

[a]See [21, Section 10.2] for more details about this "default GLM" kernel.

In-sample and out-of sample predictions

Let $\hat{y}_{e,s}$ be the predicted value of $Y^0_{e,s}$ under a quadratic loss function, that is, $\hat{y}_{e,s}$ is the posterior mean of $Y^0_{e,s}$ and is computed using (15.7). Then, as shown in Figure 15.2, the estimated model fits well the data.

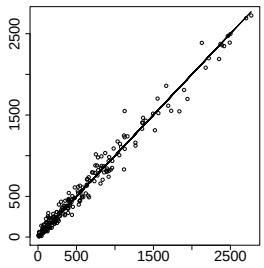

Figure 15.2: In-sample predictions versus observed data for the frisks dataset. Each dot in represents a pair $(\hat{y}_{e,s}, y^0_{e,s})$ and a diagonal line passing though (1,1) has been added.

We now use the estimated model to predict Y^0_{e,s_*} for $e = 1, \ldots, 3$, recalling that the data for the police station s_* has not been used to fit the model and that $x^0_{1,s_*} = 16$, $x^0_{2,s_*} = 44$ and $x^0_{3,s_*} = 312$.

Letting $\hat{y}_{e,s_*}$ be the predicted value of Y^0_{e,s_*} under a quadratic loss function, that is, $\hat{y}_{e,s_*}$ is the posterior mean of Y^0_{e,s_*} and is computed using (15.9), we obtain

$$(\hat{y}_{e,s_*}, y^0_{e,s_*}) = \begin{cases} (8.6, 7), & e = 1 \\ (26.9, 20), & e = 2 \\ (153.7, 295), & e = 3 \end{cases}$$

We observe that the predictions are in line with the observed values. In addition, for $e \in \{1, 2\}$ the predicted value of Y^0_{e,s_*} very close to the observed value. For $e = 3$, the prediction $\hat{y}_{e,s_*}$ is much smaller than the observed value, which may suggest that our model is too simple (note also that not much information is available).

Did the police harass members of ethnic minority groups?

Under our statistical model, information about whether the police did harass members of ethnic minority groups is contained in the posterior distribution of $(\theta^{(1)}, \theta^{(2)}, \theta^{(3)})$. In particular, the higher $\theta^{(e)}$ is the larger is the expected number of persons belonging to ethnic group e that were stopped and frisked by the police.

Figure 15.3 shows the (estimated) marginal posterior distribution of $\theta^{(e)}$ for $e \in \{1, 2, 3\}$. This figure clearly suggests that the Blacks (green histogram) and Hispanics (red histogram) tend to be stopped and frisked more often than the Whites.

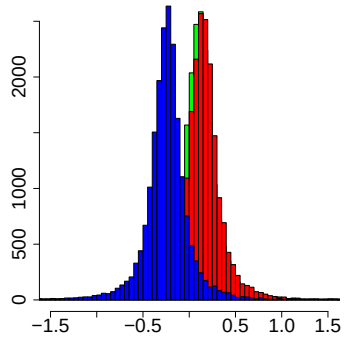

Figure 15.3: Marginal posterior distribution of $\theta^{(e)}$ for $e = 1$ (green), $e = 2$ (red) and $e = 3$ (blue).

To obtain a statistical answer to the above question we perform the test of hypothesis $H_0 : \theta \in \Theta_0$ against $H_1 : \theta \notin \Theta_0$ with

$$\Theta_0 := \{\theta \in \Theta : \theta^{(1)} > 0, \theta^{(2)} > 0, \theta^{(3)} < 0\}.$$

Taking the $a_0 - a_1$ loss function with $a_1 = a_0$, we accept the null hypothesis if $\pi(\Theta_0|d_n) > 0.5$, where $\pi(\Theta_0|d_n)$ denotes the posterior probability of Θ_0.

The estimated value of $\pi(\Theta_0|d_n)$ is around 0.63 and therefore we conclude that the white persons were indeed less likely to be arrested by the police than Black and Hispanic persons.

References

[1] Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., Stuart, A., et al. (2013). Optimal tuning of the hybrid monte carlo algorithm. *Bernoulli*, 19(5A):1501–1534.

[2] Bishop, C. M. (2006). Pattern recognition. *Machine learning*, 128(9).

[3] Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, 76(1).

[4] Comon, P. (1994). Independent component analysis, a new concept? *Signal processing*, 36(3):287–314.

[5] Friedman, J., Hastie, T., Tibshirani, R., et al. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

[6] Gelman, A. and Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.

[7] Hastie, T., Tibshirani, R., and Wainwright, M. (2019). *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC.

[8] Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications.

[9] Hoffman, M. D. and Gelman, A. (2014). The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623.

[10] Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430.

[11] Inaba, M., Katoh, N., and Imai, H. (1994). Applications of weighted voronoi diagrams and randomization to variance-based k-clustering. In *Proceedings of the tenth annual symposium on Computational geometry*, pages 332–339.

[12] Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. K. (2018). Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*.

[13] Leimkuhler, B. and Reich, S. (2004). *Simulating hamiltonian dynamics*, volume 14. Cambridge university press.

[14] Mairal, J. and Yu, B. (2012). Complexity analysis of the lasso regularization path. *arXiv preprint arXiv:1205.0079*.

[15] Mardia, K., Kent, J., and Bibby, J. (1979). *Multivariate analysis*. Probability and mathematical statistics. Academic Press Inc.

[16] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

[17] Neal, R. M. (2003). Slice sampling. *The annals of statistics*, 31(3):705–767.

[18] Neal, R. M. et al. (2011). Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2.

[19] Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14.

[20] Paulsen, V. I. and Raghupathi, M. (2016). *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Cambridge Studies in Advanced Mathematics. Cambridge University Press.

[21] Plummer, M. (2017). Jags version 4.3. 0 user manual.

[22] Plummer, M. et al. (2003). Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing*, volume 124, pages 1–10. Vienna, Austria.

[23] Quinonero-Candela, J. and Rasmussen, C. E. (2005). Analysis of some methods for reduced rank gaussian process regression. In *Switching and learning in feedback systems*, pages 98–127. Springer.

[24] Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer.

[25] Robert, C. P., Casella, G., and Casella, G. (1999). *Monte Carlo statistical methods*, volume 2. Springer.

[26] Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various metropolis-hastings algorithms. *Statistical science*, 16(4):351–367.

[27] Sande, E., Manni, C., and Speleers, H. (2020). Explicit error estimates for spline approximation of arbitrary smoothness in isogeometric analysis. *Numerische Mathematik*, 144(4):889–929.

[28] Särkkä, S. and Solin, A. (2019). *Applied stochastic differential equations*, volume 10. Cambridge University Press.

[29] Sniekers, S., van der Vaart, A., et al. (2015). Adaptive bayesian credible sets in regression with a gaussian process prior. *Electronic Journal of Statistics*, 9(2):2475–2527.

[30] Szabó, B., Van Der Vaart, A. W., van Zanten, J., et al. (2015). Frequentist coverage of adaptive nonparametric bayesian credible sets. *The Annals of Statistics*, 43(4):1391–1428.

[31] Vaart, A. v. d. and Zanten, H. v. (2011). Information rates of nonparametric gaussian process methods. *Journal of Machine Learning Research*, 12(Jun):2095–2119.

[32] van der Vaart, A. W., van Zanten, J. H., et al. (2009). Adaptive bayesian estimation using a gaussian random field with inverse gamma bandwidth. *The Annals of Statistics*, 37(5B):2655–2675.

[33] van Wieringen, W. N. (2015). Lecture notes on ridge regression. *arXiv preprint arXiv:1509.09169*.

[34] Vishnoi, N. K. (2021). An introduction to hamiltonian monte carlo method for sampling. *arXiv preprint arXiv:2108.12107*.

[35] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

[36] Wood, S. N. (2017). *Generalized additive models: an introduction with R*. CRC press.

[37] Wu, C., Stoehr, J., and Robert, C. P. (2018). Faster hamiltonian monte carlo by learning leapfrog scale. *arXiv preprint arXiv:1810.04449*.