G: Alternative MCMC algorithms

Fish 559; Day 3: 13h30-14h30

Hamiltonian Monte Carlo-I

Hamiltonian Monte Carlo uses the gradient of the log-likelihood function as the basis of the jump function

Algorithm 1 Hamiltonian Monte Carlo

Given θ^0, ϵ, L, $\mathcal{L}, M$:
for $m = 1$ to M **do**
  Sample $r^0 \sim \mathcal{N}(0, I)$.
  Set $\theta^m \leftarrow \theta^{m-1}, \tilde{\theta} \leftarrow \theta^{m-1}, \tilde{r} \leftarrow r^0$.
  for $i = 1$ to L **do**
    Set $\tilde{\theta}, \tilde{r} \leftarrow \text{Leapfrog}(\tilde{\theta}, \tilde{r}, \epsilon)$.
  end for
  With probability $\alpha = \min\left\{1, \frac{\exp[\mathcal{L}(\tilde{\theta}) - \frac{1}{2}\tilde{r}\cdot\tilde{r}]}{\exp[\mathcal{L}(\theta^{m-1}) - \frac{1}{2}r^0\cdot r^0]}\right\}$, set $\theta^m \leftarrow \tilde{\theta}$, $r^m \leftarrow -\tilde{r}$.
end for

function Leapfrog(θ, r, ϵ)
Set $\tilde{r} \leftarrow r + (\epsilon/2)\nabla_\theta \mathcal{L}(\theta)$.
Set $\tilde{\theta} \leftarrow \theta + \epsilon\tilde{r}$.
Set $\tilde{r} \leftarrow \tilde{r} + (\epsilon/2)\nabla_\theta \mathcal{L}(\tilde{\theta})$.
return $\tilde{\theta}, \tilde{r}$.

Leapfrog – upgrades r (sample) and updates theta.

r changes in the direction of increasing probability using info about the gradient and then theta is updates (then repeat, hence leapfrog)

Hamiltonian Monte Carlo-II

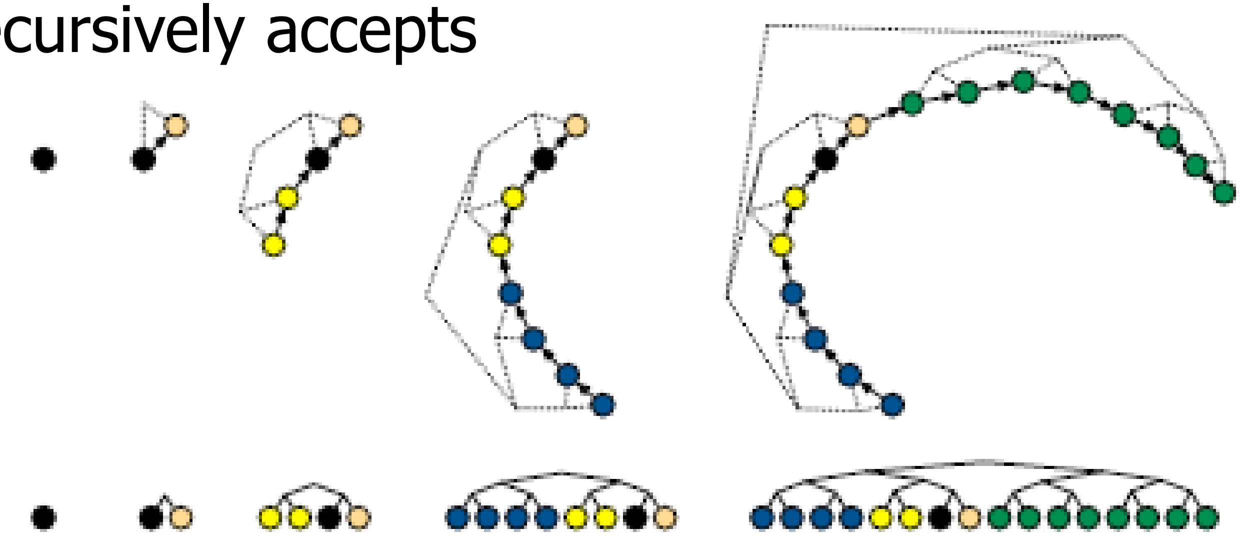
A challenge with this sampler is that it requires that *L (number of leapfrogs)* and ε *(how big your jump is)* be set by the user – unfortunately the performance of the algorithm can depend quite substantially on the values for these parameters.

559

No-U-Turn Sampler-I

The idea behind the (naïve) no-U-turn sampler is keep moving on a path until the path starts to bend back on itself (has a U-turn).

Recursively accepts



No-U-Turn Sampler-II

Algorithm 2 Naive No-U-Turn Sampler

Given $\theta^0, \epsilon, \mathcal{L}, M$:
for $m = 1$ **to** M **do**
  Resample $r^0 \sim \mathcal{N}(0, I)$.
  Resample $u \sim \text{Uniform}([0, \exp\{\mathcal{L}(\theta^{m-1} - \frac{1}{2}r^0 \cdot r^0\}])$
  Initialize $\theta^- = \theta^{m-1}, \theta^+ = \theta^{m-1}, r^- = r^0, r^+ = r^0, j = 0, \mathcal{C} = \{(\theta^{m-1}, r^0)\}, s = 1$.
  while $s = 1$ **do**
    Choose a direction $v_j \sim \text{Uniform}(\{-1, 1\})$.
    if $v_j = -1$ **then**
      $\theta^-, r^-, -, -, \mathcal{C}', s' \leftarrow \text{BuildTree}(\theta^-, r^-, u, v_j, j, \epsilon)$.
    else
      $-, -, \theta^+, r^+, \mathcal{C}', s' \leftarrow \text{BuildTree}(\theta^+, r^+, u, v_j, j, \epsilon)$.
    end if
    if $s' = 1$ **then**
      $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$.
    end if
    $s \leftarrow s'\mathbb{I}[(\theta^+ - \theta^-) \cdot r^- \geq 0]\mathbb{I}[(\theta^+ - \theta^-) \cdot r^+ \geq 0]$.
    $j \leftarrow j + 1$.
  end while
  Sample θ^m, r uniformly at random from $\mathcal{C}$.
end for

C is the set of points we are a

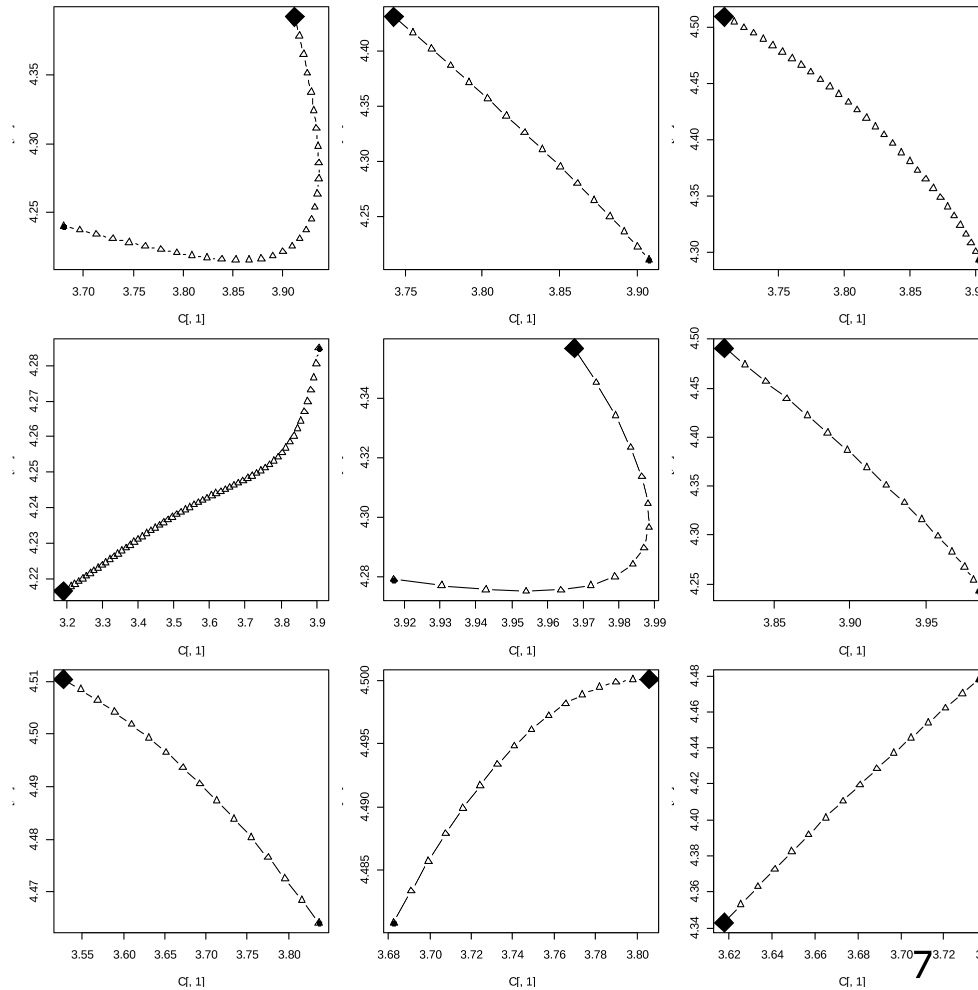
No-U-Turn Sampler-III

function BuildTree$(\theta, r, u, v, j, \epsilon)$
if $j = 0$ **then**
  Base case—take one leapfrog step in the direction v.
  $\theta', r' \leftarrow \text{Leapfrog}(\theta, r, v\epsilon)$.
  $$C' \leftarrow \begin{cases} \{(\theta', r')\} & \text{if } u \leq \exp\{\mathcal{L}(\theta') - \frac{1}{2} r' \cdot r'\} \\ \emptyset & \text{else} \end{cases}$$
  $s' \leftarrow \mathbb{I}[\mathcal{L}(\theta') - \frac{1}{2} r' \cdot r' > \log u - \Delta_{\max}]$.
  return $\theta', r', \theta', r', C', s'$.
else
  Recursion—build the left and right subtrees.
  $\theta^-, r^-, \theta^+, r^+, C', s' \leftarrow \text{BuildTree}(\theta, r, u, v, j-1, \epsilon)$.
  if $v = -1$ **then**
    $\theta^-, r^-, -, -, C'', s'' \leftarrow \text{BuildTree}(\theta^-, r^-, u, v, j-1, \epsilon)$.
  else
    $-, -, \theta^+, r^+, C'', s'' \leftarrow \text{BuildTree}(\theta^+, r^+, u, v, j-1, \epsilon)$.
  end if
  $s' \leftarrow s' s'' \mathbb{I}[(\theta^+ - \theta^-) \cdot r^- \geq 0] \mathbb{I}[(\theta^+ - \theta^-) \cdot r^+ \geq 0]$.
  $C' \leftarrow C' \cup C''$.
  return $\theta^-, r^-, \theta^+, r^+, C', s'$.
end if

Recursive fxn, buildtree calls itself

Application to LectF

This is a 3-parameter model, but when it is constrained to only 2 parameters the behaviour of NUTS is easier to see.



No-U-Turn Sampler-IV

The No-U-turn sampler does not require that *L* be specified. However, algorithms are available:

- to make the sampler more efficient; and
- automatically select ε.

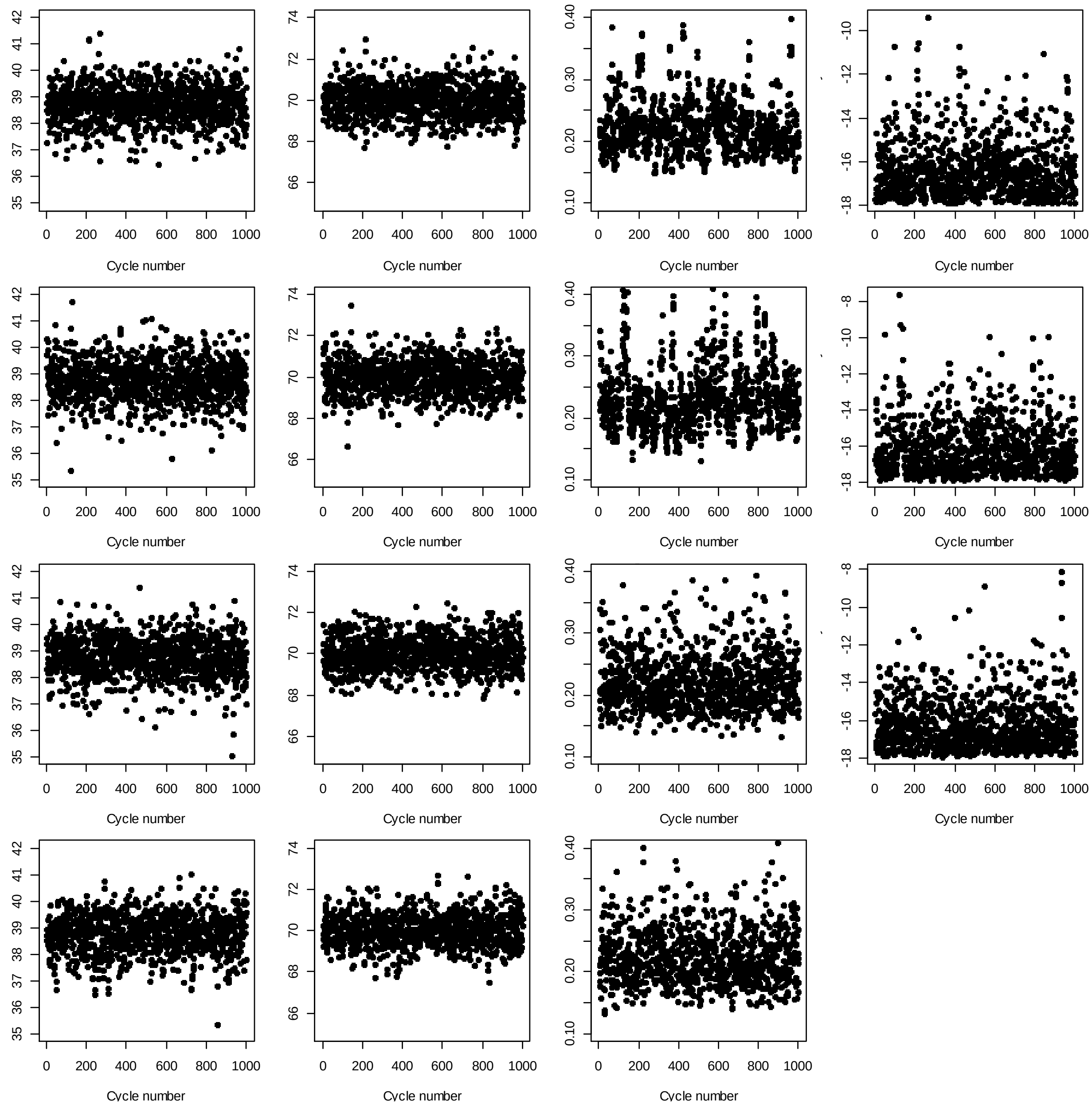
These extensions are available in *adnuts*.

Naïve NUTS

Hamiltonian

MCMC with a normal jump function

NUTS in adnuts



559

Reference

Hoffman, M.D. and A. Gelman. 2014. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning*. 15: 1361-1381.