

STA 601: Hamiltonian Monte Carlo

Merlise Clyde

Nov 19, 2021



Gibbs sampling

- Consider model

$$\begin{aligned}\mathbf{Y}_1, \dots, \mathbf{Y}_n &\sim N_2(\boldsymbol{\theta}, \Sigma); \\ \theta_j &\sim N(0, 1) \quad j = 1, 2.\end{aligned}$$



Gibbs sampling

- Consider model

$$\mathbf{Y}_1, \dots, \mathbf{Y}_n \sim N_2(\boldsymbol{\theta}, \Sigma);$$
$$\theta_j \sim N(0, 1) \quad j = 1, 2.$$

- Suppose that the covariance matrix Σ is known and has the form

$$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$



Gibbs sampling

- Consider model

$$\mathbf{Y}_1, \dots, \mathbf{Y}_n \sim N_2(\boldsymbol{\theta}, \Sigma);$$
$$\theta_j \sim N(0, 1) \quad j = 1, 2.$$

- Suppose that the covariance matrix Σ is known and has the form

$$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

- What happens when $\rho = 0.995$?



Gibbs sampling

- Consider model

$$\mathbf{Y}_1, \dots, \mathbf{Y}_n \sim N_2(\boldsymbol{\theta}, \Sigma);$$
$$\theta_j \sim N(0, 1) \quad j = 1, 2.$$

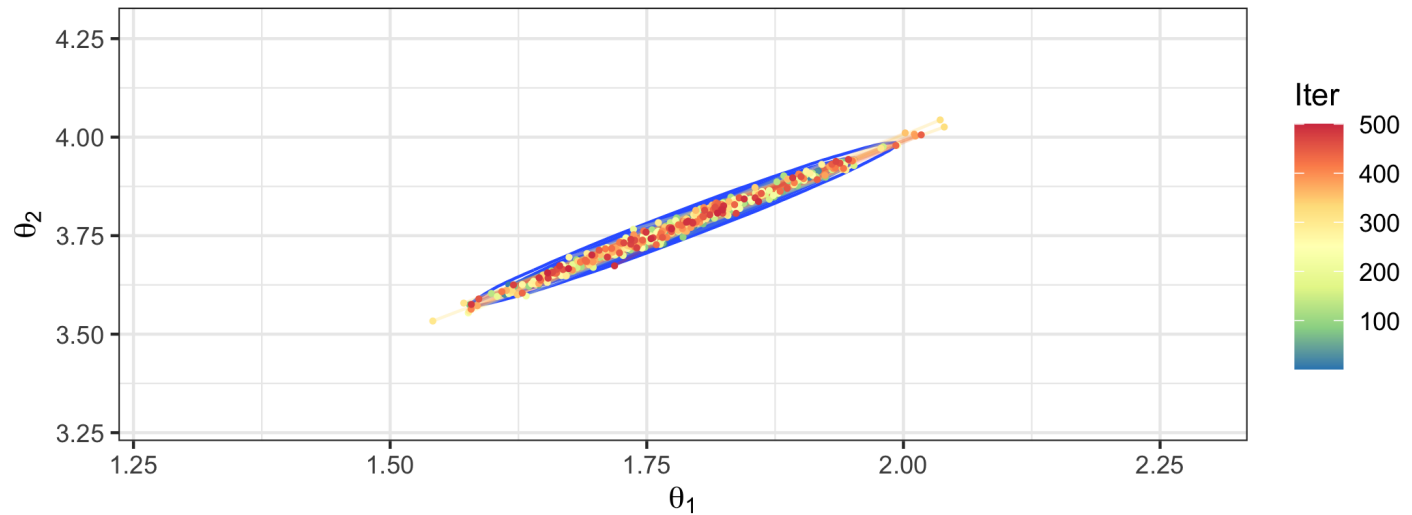
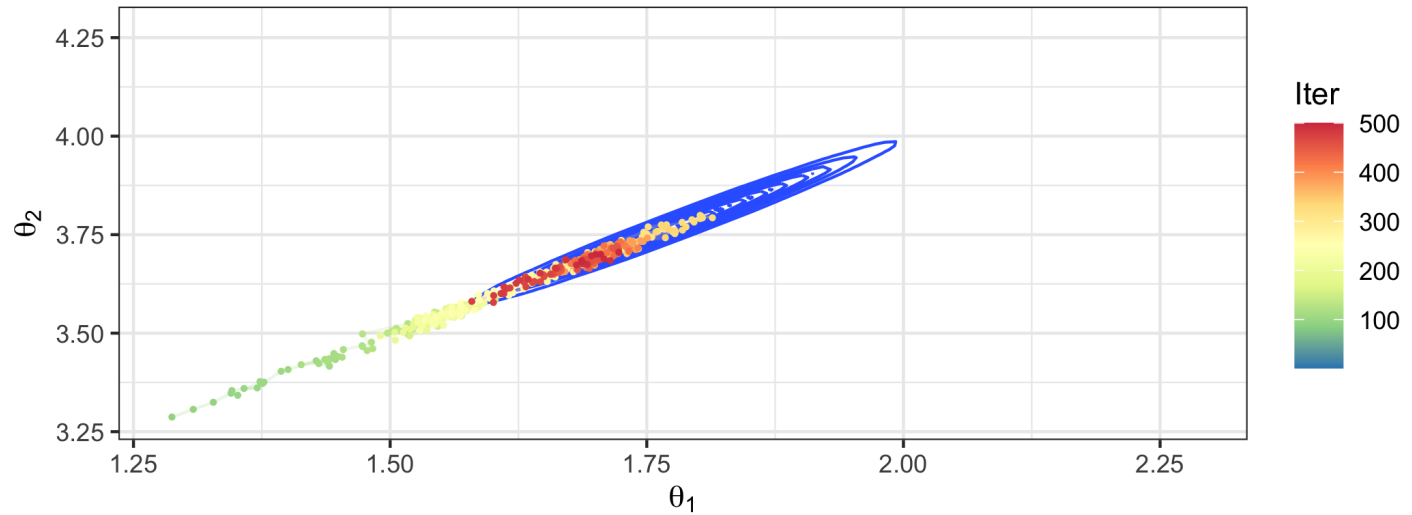
- Suppose that the covariance matrix Σ is known and has the form

$$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

- What happens when $\rho = 0.995$?

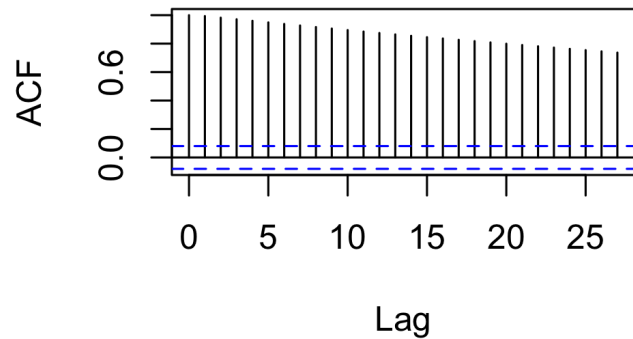


Gibbs vs Stan samples

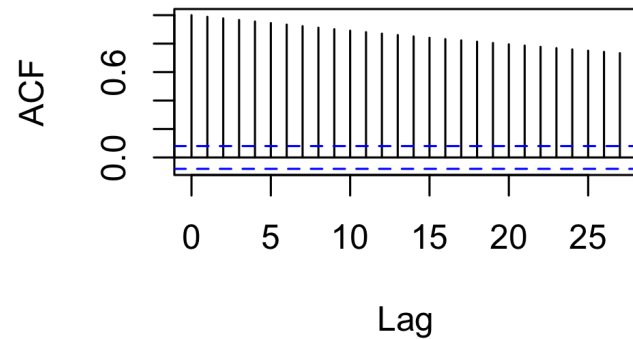


ACF

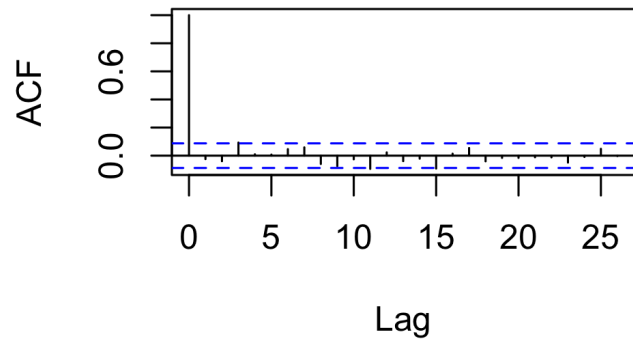
Series norm_gibbs_samps[, 1]



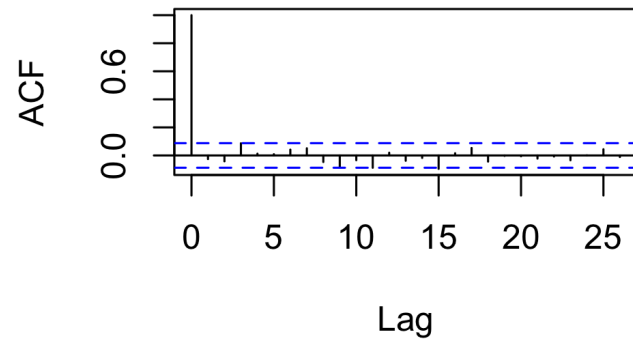
Series norm_gibbs_samps[, 2]



Series stan_res\$theta[, 1]



Series stan_res\$theta[, 2]



Hamiltonian Monte Carlo (HMC)

- HMC creates transitions that *efficiently explore the parameter space* by using concepts from Hamiltonian mechanics.



Hamiltonian Monte Carlo (HMC)

- HMC creates transitions that *efficiently explore the parameter space* by using concepts from Hamiltonian mechanics.
- In Hamiltonian mechanics, a physical system is specified by positions \mathbf{q} and momenta \mathbf{p} .



Hamiltonian Monte Carlo (HMC)

- HMC creates transitions that *efficiently explore the parameter space* by using concepts from Hamiltonian mechanics.
- In Hamiltonian mechanics, a physical system is specified by positions \mathbf{q} and momenta \mathbf{p} .
- A space defined by these coordinates is called a **phase space**



Hamiltonian Monte Carlo (HMC)

- HMC creates transitions that *efficiently explore the parameter space* by using concepts from Hamiltonian mechanics.
- In Hamiltonian mechanics, a physical system is specified by positions \mathbf{q} and momenta \mathbf{p} .
- A space defined by these coordinates is called a **phase space**
- If the parameters of interest in a typical MCMC method are denoted as q_1, \dots, q_K , then HMC introduces auxiliary **momentum** parameters p_1, \dots, p_K such that the algorithm produces draws from the joint density:

$$\pi(\mathbf{q}, \mathbf{p}) = \pi(\mathbf{p}|\mathbf{q})\pi(\mathbf{q})$$



Hamiltonian Monte Carlo (HMC)

- HMC creates transitions that *efficiently explore the parameter space* by using concepts from Hamiltonian mechanics.
- In Hamiltonian mechanics, a physical system is specified by positions \mathbf{q} and momenta \mathbf{p} .
- A space defined by these coordinates is called a **phase space**
- If the parameters of interest in a typical MCMC method are denoted as q_1, \dots, q_K , then HMC introduces auxiliary **momentum** parameters p_1, \dots, p_K such that the algorithm produces draws from the joint density:

$$\pi(\mathbf{q}, \mathbf{p}) = \pi(\mathbf{p}|\mathbf{q})\pi(\mathbf{q})$$

- marginalizing over the p_k 's, we recover the marginal distribution of the q_k 's Therefore, if we create a Markov Chain that converges to $\pi(\mathbf{q}, \mathbf{p})$, we have immediate access to samples from $\pi(\mathbf{q})$, which is our target distribution.



Hamiltonian

- Hamilton's equations describe the time evolution of the system in terms of the **Hamiltonian**, \mathcal{H} , which corresponds to the total energy of the system:

$$\mathcal{H}(\mathbf{p}, \mathbf{q}) = K(\mathbf{q}, \mathbf{p}) + U(\mathbf{q})$$



Hamiltonian

- Hamilton's equations describe the time evolution of the system in terms of the **Hamiltonian**, \mathcal{H} , which corresponds to the total energy of the system:

$$\mathcal{H}(\mathbf{p}, \mathbf{q}) = K(\mathbf{q}, \mathbf{p}) + U(\mathbf{q})$$

- $K(\mathbf{q}, \mathbf{p})$ represents the **kinetic energy** of the system and is equal to the negative logarithm of the momentum distribution, e.g.

$$K(\mathbf{p}) = \frac{\mathbf{p}^T M^{-1} \mathbf{p}}{2} = \sum_i \frac{p_i^2}{2m_i}$$

- M is the Mass matrix



Hamiltonian

- Hamilton's equations describe the time evolution of the system in terms of the **Hamiltonian**, \mathcal{H} , which corresponds to the total energy of the system:

$$\mathcal{H}(\mathbf{p}, \mathbf{q}) = K(\mathbf{q}, \mathbf{p}) + U(\mathbf{q})$$

- $K(\mathbf{q}, \mathbf{p})$ represents the **kinetic energy** of the system and is equal to the negative logarithm of the momentum distribution, e.g.

$$K(\mathbf{p}) = \frac{\mathbf{p}^T M^{-1} \mathbf{p}}{2} = \sum_i \frac{p_i^2}{2m_i}$$

- M is the Mass matrix
- $U(\mathbf{q})$ the **potential energy** of the system; equal to the negative logarithm of the distribution of \mathbf{q} .

$$\pi(\mathbf{q}, \mathbf{p}) \propto e^{-\mathcal{H}(\mathbf{q}, \mathbf{p})} = e^{-K(\mathbf{p})} e^{-U(\mathbf{q})}$$



Evolution

- At each iteration of the sampling algorithm, HMC implementations make draws from some distribution $\pi(\mathbf{p}|\mathbf{q})$ and then *evolves the system* (\mathbf{q}, \mathbf{p}) to obtain the next sample of \mathbf{q} .



Evolution

- At each iteration of the sampling algorithm, HMC implementations make draws from some distribution $\pi(\mathbf{p}|\mathbf{q})$ and then *evolves the system* (\mathbf{q}, \mathbf{p}) to obtain the next sample of \mathbf{q} .
- To "evolve the system" is to move (\mathbf{q}, \mathbf{p}) forward in "time," i.e. to change the values of (\mathbf{q}, \mathbf{p}) according to Hamilton's differential equations:

$$\begin{aligned}\frac{d\mathbf{p}}{dt} &= -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} = -\frac{\partial K}{\partial \mathbf{q}} - \frac{\partial U}{\partial \mathbf{q}} \\ \frac{d\mathbf{q}}{dt} &= +\frac{\partial \mathcal{H}}{\partial \mathbf{p}} = +\frac{\partial K}{\partial \mathbf{p}}\end{aligned}$$



Evolution

- At each iteration of the sampling algorithm, HMC implementations make draws from some distribution $\pi(\mathbf{p}|\mathbf{q})$ and then *evolves the system* (\mathbf{q}, \mathbf{p}) to obtain the next sample of \mathbf{q} .
 - To "evolve the system" is to move (\mathbf{q}, \mathbf{p}) forward in "time," i.e. to change the values of (\mathbf{q}, \mathbf{p}) according to Hamilton's differential equations:

$$\begin{aligned}\frac{d\mathbf{p}}{dt} &= -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} = -\frac{\partial K}{\partial \mathbf{q}} - \frac{\partial U}{\partial \mathbf{q}} \\ \frac{d\mathbf{q}}{dt} &= +\frac{\partial \mathcal{H}}{\partial \mathbf{p}} = +\frac{\partial K}{\partial \mathbf{p}}\end{aligned}$$

- Defines a mapping T_s from the state at any time t to the state at $t + s$



Evolution

- At each iteration of the sampling algorithm, HMC implementations make draws from some distribution $\pi(\mathbf{p}|\mathbf{q})$ and then *evolves the system* (\mathbf{q}, \mathbf{p}) to obtain the next sample of \mathbf{q} .
 - To "evolve the system" is to move (\mathbf{q}, \mathbf{p}) forward in "time," i.e. to change the values of (\mathbf{q}, \mathbf{p}) according to Hamilton's differential equations:

$$\begin{aligned}\frac{d\mathbf{p}}{dt} &= -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} = -\frac{\partial K}{\partial \mathbf{q}} - \frac{\partial U}{\partial \mathbf{q}} \\ \frac{d\mathbf{q}}{dt} &= +\frac{\partial \mathcal{H}}{\partial \mathbf{p}} = +\frac{\partial K}{\partial \mathbf{p}}\end{aligned}$$

- Defines a mapping T_s from the state at any time t to the state at $t + s$

"The differential change in momentum parameters \mathbf{p} over time is governed in part by the differential information of the density over the target parameters."



Key Properties

1) **Reversibility** The mapping of the state at time t ($\mathbf{p}(t), \mathbf{q}(t)$) to the state at $t + s$ ($\mathbf{p}(t + s), \mathbf{q}(t + s)$) is one-to-one and we have an inverse T_{-s} - obtained by negating the derivatives; $K(\mathbf{p}) = K(-\mathbf{p})$
MCMC updates using the dynamics don't modify invariant distribution!



Key Properties

1) **Reversibility** The mapping of the state at time t ($\mathbf{p}(t), \mathbf{q}(t)$) to the state at $t + s$ ($\mathbf{p}(t + s), \mathbf{q}(t + s)$) is one-to-one and we have an inverse T_{-s} - obtained by negating the derivatives; $K(\mathbf{p}) = K(-\mathbf{p})$
MCMC updates using the dynamics don't modify invariant distribution!

2) **Invariance/Conservation** the dynamics keep the Hamiltonian invariant - if we use the dynamics to generate proposals, the acceptance probability of MH is equal to one if \mathcal{H} is kept invariant!



Key Properties

1) **Reversibility** The mapping of the state at time t ($\mathbf{p}(t), \mathbf{q}(t)$) to the state at $t + s$ ($\mathbf{p}(t + s), \mathbf{q}(t + s)$) is one-to-one and we have an inverse T_{-s} - obtained by negating the derivatives; $K(\mathbf{p}) = K(-\mathbf{p})$
MCMC updates using the dynamics don't modify invariant distribution!

2) **Invariance/Conservation** the dynamics keep the Hamiltonian invariant - if we use the dynamics to generate proposals, the acceptance probability of MH is equal to one if \mathcal{H} is kept invariant!

3) **Volume Preservation/Symplectiness** the mapping T_s of a region R to $T_s(R)$ preserves volume -- means that we do not need to compute Jacobians



Key Properties

1) **Reversibility** The mapping of the state at time t ($\mathbf{p}(t), \mathbf{q}(t)$) to the state at $t + s$ ($\mathbf{p}(t + s), \mathbf{q}(t + s)$) is one-to-one and we have an inverse T_{-s} - obtained by negating the derivatives; $K(\mathbf{p}) = K(-\mathbf{p})$
MCMC updates using the dynamics don't modify invariant distribution!

2) **Invariance/Conservation** the dynamics keep the Hamiltonian invariant - if we use the dynamics to generate proposals, the acceptance probability of MH is equal to one if \mathcal{H} is kept invariant!

3) **Volume Preservation/Symplectiness** the mapping T_s of a region R to $T_s(R)$ preserves volume -- means that we do not need to compute Jacobians

in practice we need to use approximations to solve the PDE's so won't have exact invariance etc so acceptance probability is not 1!



Approximate Solutions to Differential Equations

- Discretize time into steps ϵ



Approximate Solutions to Differential Equations

- Discretize time into steps ϵ
- Euler's Method for i th coordinate

$$p_i(t + \epsilon) = p_i(t) + \epsilon \frac{dp_i}{dt}(t) = p_i(t) - \epsilon \frac{\partial U(q_i(t))}{\partial q_i}$$

$$q_i(t + \epsilon) = q_i(t) + \epsilon \frac{dq_i}{dt}(t) = q_i(t) + \epsilon \frac{\partial K(p_i(t))}{\partial p_i} = q_i(t) + \epsilon \frac{p_i(t)}{m_i}$$



Approximate Solutions to Differential Equations

- Discretize time into steps ϵ
- Euler's Method for i th coordinate

$$p_i(t + \epsilon) = p_i(t) + \epsilon \frac{dp_i}{dt}(t) = p_i(t) - \epsilon \frac{\partial U(q_i(t))}{\partial q_i}$$

$$q_i(t + \epsilon) = q_i(t) + \epsilon \frac{dq_i}{dt}(t) = q_i(t) + \epsilon \frac{\partial K(p_i(t))}{\partial p_i} = q_i(t) + \epsilon \frac{p_i(t)}{m_i}$$

- Modified Euler method

$$p_i(t + \epsilon) = p_i(t) - \epsilon \frac{\partial U(q_i(t))}{\partial q_i}$$

$$q_i(t + \epsilon) = q_i(t) + \epsilon \frac{p_i(t + \epsilon)}{m_i}$$



Leapfrog

- Divide into half steps



Leapfrog

- Divide into half steps
- apply Modified Euler

$$\begin{aligned}p_i(t + \epsilon/2) &= p_i(t) - \frac{\epsilon}{2} \frac{\partial U(q_i(t))}{\partial q_i} \\q_i(t + \epsilon) &= q_i(t) + \epsilon \frac{p_i(t + \epsilon/2)}{m_i} \\p_i(t + \epsilon) &= p_i(t) - \frac{\epsilon}{2} \frac{\partial U(q_i(t + \epsilon))}{\partial q_i}\end{aligned}$$



Leapfrog

- Divide into half steps
- apply Modified Euler

$$\begin{aligned}p_i(t + \epsilon/2) &= p_i(t) - \frac{\epsilon}{2} \frac{\partial U(q_i(t))}{\partial q_i} \\q_i(t + \epsilon) &= q_i(t) + \epsilon \frac{p_i(t + \epsilon/2)}{m_i} \\p_i(t + \epsilon) &= p_i(t) - \frac{\epsilon}{2} \frac{\partial U(q_i(t + \epsilon))}{\partial q_i}\end{aligned}$$

- Preserves volume exactly



Leapfrog

- Divide into half steps
- apply Modified Euler

$$p_i(t + \epsilon/2) = p_i(t) - \frac{\epsilon}{2} \frac{\partial U(q_i(t))}{\partial q_i}$$

$$q_i(t + \epsilon) = q_i(t) + \epsilon \frac{p_i(t + \epsilon/2)}{m_i}$$

$$p_i(t + \epsilon) = p_i(t) - \frac{\epsilon}{2} \frac{\partial U(q_i(t + \epsilon))}{\partial q_i}$$

- Preserves volume exactly
- Reversible



Leapfrog

- Divide into half steps
- apply Modified Euler

$$p_i(t + \epsilon/2) = p_i(t) - \frac{\epsilon}{2} \frac{\partial U(q_i(t))}{\partial q_i}$$

$$q_i(t + \epsilon) = q_i(t) + \epsilon \frac{p_i(t + \epsilon/2)}{m_i}$$

$$p_i(t + \epsilon) = p_i(t) - \frac{\epsilon}{2} \frac{\partial U(q_i(t + \epsilon))}{\partial q_i}$$

- Preserves volume exactly
- Reversible
- We don't get exact invariance (so probability of acceptance is not 1)



Leapfrog

- Divide into half steps
- apply Modified Euler

$$\begin{aligned}p_i(t + \epsilon/2) &= p_i(t) - \frac{\epsilon}{2} \frac{\partial U(q_i(t))}{\partial q_i} \\q_i(t + \epsilon) &= q_i(t) + \epsilon \frac{p_i(t + \epsilon/2)}{m_i} \\p_i(t + \epsilon) &= p_i(t) - \frac{\epsilon}{2} \frac{\partial U(q_i(t + \epsilon))}{\partial q_i}\end{aligned}$$

- Preserves volume exactly
- Reversible
- We don't get exact invariance (so probability of acceptance is not 1)
- Step size and number of steps is still important!



MCMC with HMC

Steps: replace q with θ



MCMC with HMC

Steps: replace \mathbf{q} with $\boldsymbol{\theta}$

1) sample a new value for the momentum $\mathbf{p}^{(t)} \sim N(0, M)$



MCMC with HMC

Steps: replace \mathbf{q} with $\boldsymbol{\theta}$

- 1) sample a new value for the momentum $\mathbf{p}^{(t)} \sim N(0, M)$
- 2) Metropolis: from current state $(\mathbf{q}^{(t-1)}, \mathbf{p}^{(t)})$ simulate proposal $(\mathbf{q}^*, \mathbf{p}^*)$ using Hamiltonian dynamics by applying Leapfrog with step size ϵ for L steps (tuning parameters)



MCMC with HMC

Steps: replace \mathbf{q} with $\boldsymbol{\theta}$

- 1) sample a new value for the momentum $\mathbf{p}^{(t)} \sim N(0, M)$
- 2) Metropolis: from current state $(\mathbf{q}^{(t-1)}, \mathbf{p}^{(t)})$ simulate proposal $(\mathbf{q}^*, \mathbf{p}^*)$ using Hamiltonian dynamics by applying Leapfrog with step size ϵ for L steps (tuning parameters)
- 3) Accept or reject acceptance probability is

$$\min\{1, \exp(-\mathcal{H}(\mathbf{q}^*, \mathbf{p}^*) + \mathcal{H}(\mathbf{q}^{(t-1)}, \mathbf{p}^{(t)}))\}$$



Tuning

- in addition to tuning ϵ and L , we can tune M



Tuning

- in addition to tuning ϵ and L , we can tune M
- $\text{Cov}(\mathbf{q}) = V$ can be highly variable



Tuning

- in addition to tuning ϵ and L , we can tune M
- $\text{Cov}(\mathbf{q}) = V$ can be highly variable
- Consider reparameterization $A\mathbf{q} = \mathbf{q}'$ so that $\text{Cov}(A\mathbf{q}) = AVA^T = I_d$; $A = V^{-1/2}$



Tuning

- in addition to tuning ϵ and L , we can tune M
- $\text{Cov}(\mathbf{q}) = V$ can be highly variable
- Consider reparameterization $A\mathbf{q} = \mathbf{q}'$ so that $\text{Cov}(A\mathbf{q}) = AVA^T = I_d; A = V^{-1/2}$
- eliminates posterior correlation!



Tuning

- in addition to tuning ϵ and L , we can tune M
- $\text{Cov}(\mathbf{q}) = V$ can be highly variable
- Consider reparameterization $A\mathbf{q} = \mathbf{q}'$ so that $\text{Cov}(A\mathbf{q}) = AVA^T = I_d; A = V^{-1/2}$
- eliminates posterior correlation!
- generally trick of reparameterizing to reduce posterior correlation is often called **pre-conditioning** - improves efficiency!



Tuning

- in addition to tuning ϵ and L , we can tune M
- $\text{Cov}(\mathbf{q}) = V$ can be highly variable
- Consider reparameterization $A\mathbf{q} = \mathbf{q}'$ so that $\text{Cov}(A\mathbf{q}) = AVA^T = I_d$; $A = V^{-1/2}$
- eliminates posterior correlation!
- generally trick of reparameterizing to reduce posterior correlation is often called **pre-conditioning** - improves efficiency!
- use $M = I_d$



Tuning

- in addition to tuning ϵ and L , we can tune M
- $\text{Cov}(\mathbf{q}) = V$ can be highly variable
- Consider reparameterization $A\mathbf{q} = \mathbf{q}'$ so that $\text{Cov}(A\mathbf{q}) = AVA^T = I_d$; $A = V^{-1/2}$
- eliminates posterior correlation!
- generally trick of reparameterizing to reduce posterior correlation is often called **pre-conditioning** - improves efficiency!
- use $M = I_d$
- Automatic tuning is achieved by the No-U-Turn-Sampler (NUTS)
- bit complicated, but used by STAN



Tuning

- in addition to tuning ϵ and L , we can tune M
- $\text{Cov}(\mathbf{q}) = V$ can be highly variable
- Consider reparameterization $A\mathbf{q} = \mathbf{q}'$ so that $\text{Cov}(A\mathbf{q}) = AVA^T = I_d$; $A = V^{-1/2}$
- eliminates posterior correlation!
- generally trick of reparameterizing to reduce posterior correlation is often called **pre-conditioning** - improves efficiency!
- use $M = I_d$
- Automatic tuning is achieved by the No-U-Turn-Sampler (NUTS)
- bit complicated, but used by STAN
- other variations Metropolis-Adjusted Langevin Algorithm (MALA)



Hybrid Approaches

- Recall mixed effects model

$$Y_{ij} = x_{ij}^T B + z_{ij}^T \beta_i + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$



Hybrid Approaches

- Recall mixed effects model

$$Y_{ij} = x_{ij}^T B + z_{ij}^T \beta_i + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

- random effects $b_i \sim N_d(0, \Psi)$ (diagonal Ψ)



Hybrid Approaches

- Recall mixed effects model

$$Y_{ij} = x_{ij}^T B + z_{ij}^T \beta_i + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

- random effects $b_i \sim N_d(0, \Psi)$ (diagonal Ψ)
- marginalize over the random effects



Hybrid Approaches

- Recall mixed effects model

$$Y_{ij} = x_{ij}^T B + z_{ij}^T \beta_i + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

- random effects $b_i \sim N_d(0, \Psi)$ (diagonal Ψ)
- marginalize over the random effects

$$Y_i = N(X_i B, Z_i \Psi Z_i^T + \sigma^2 I)$$



Hybrid Approaches

- Recall mixed effects model

$$Y_{ij} = x_{ij}^T B + z_{ij}^T \beta_i + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

- random effects $b_i \sim N_d(0, \Psi)$ (diagonal Ψ)
- marginalize over the random effects

$$Y_i = N(X_i B, Z_i \Psi Z_i^T + \sigma^2 I)$$

- we could use Gibbs on the conditional model, but we may get slow mixing (i.e. due to updating variance components)



Hybrid Approaches

- Recall mixed effects model

$$Y_{ij} = x_{ij}^T B + z_{ij}^T \beta_i + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

- random effects $b_i \sim N_d(0, \Psi)$ (diagonal Ψ)
- marginalize over the random effects

$$Y_i = N(X_i B, Z_i \Psi Z_i^T + \sigma^2 I)$$

- we could use Gibbs on the conditional model, but we may get slow mixing (i.e. due to updating variance components)
- run HMC within Gibbs to update the variance components Ψ and σ^2 using the marginal model given B



Hybrid Approaches

- Recall mixed effects model

$$Y_{ij} = x_{ij}^T B + z_{ij}^T \beta_i + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

- random effects $b_i \sim N_d(0, \Psi)$ (diagonal Ψ)
- marginalize over the random effects

$$Y_i = N(X_i B, Z_i \Psi Z_i^T + \sigma^2 I)$$

- we could use Gibbs on the conditional model, but we may get slow mixing (i.e. due to updating variance components)
- run HMC within Gibbs to update the variance components Ψ and σ^2 using the marginal model given B
- HMC in its basic form doesn't like constraints so reparameterize to use log transformations

