

5. Inference with Monte Carlo methods

- Monte Carlo methods
 - Monte Carlo integration
 - Sampling from simple distributions
 - Rejection sampling
 - Importance sampling
- Markov chain Monte Carlo
 - Metropolis-Hastings algorithm
 - Gibbs sampling
 - Hamiltonian Monte Carlo

Outline

- Monte Carlo methods
 - Monte Carlo integration
 - Sampling from simple distributions
 - Rejection sampling
 - Importance sampling
- Markov chain Monte Carlo
 - Metropolis-Hastings algorithm
 - Gibbs sampling
 - Hamiltonian Monte Carlo

Monte Carlo integration

$$\mathbf{E} f(X) = \int f(x)p(x) dx$$

- $x \in \mathbf{R}^n$
- $f: \mathbf{R}^n \rightarrow \mathbf{R}^m$
- $p(x)$: target distribution of X
- target distribution as posterior $p(x | y)$
 - use the unnormalized distribution $\tilde{p}(x) = p(x, y)$
 - normalize the result with $Z = \int p(x, y) dx = p(y)$

Monte Carlo (MC) integration: draw n random samples $x \sim p(x)$

$$\mathbf{E} f(X) = \int f(x)p(x) dx \approx \frac{1}{n} \sum_{i=1}^n f(x_i)$$

Monte Carlo integration

let $\mu = \mathbf{E} f(x)$ be the exact mean, $\hat{\mu}$ be the MC approximation

- with independence samples:

$$(\hat{\mu} - \mu) \rightarrow \mathcal{N}\left(0, \frac{\hat{\sigma}^2}{n}\right)$$

$$- \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (f(x_i) - \hat{\mu})^2$$

- for large enough n :

$$\mathbf{P}\left(\hat{\mu} - 1.96\sqrt{\frac{\hat{\sigma}^2}{n}} \leq \mu \leq \hat{\mu} + 1.96\sqrt{\frac{\hat{\sigma}^2}{n}}\right) \approx 0.95$$

- $\sqrt{\hat{\sigma}^2/n}$: (numerical) standard error, denotes the uncertainty about our μ estimation
- standard error of μ estimation is independent of the integration dimensionality

Monte Carlo integration

example: estimating π by MC integration

- given a (Euclidean) ball in \mathbf{R}^2 : $B(r) = \{(x, y) \mid x^2 + y^2 \leq r^2\}$
- the area of the ball $S = \pi r^2 = \int_{-r}^r \int_{-r}^r I_B(x, y) \, dx dy$
 - I_B : indicator function of ball $B(r)$, equals to 1 for points inside the ball, and 0 outside
- let $p(x), p(y) \sim \mathcal{U}(-r, r)$

$$\begin{aligned}\pi &= \frac{1}{r^2} S = \frac{1}{r^2} (2r)(2r) \iint I_B(x, y) p(x) p(y) \, dx dy \\ &= \frac{1}{r^2} 4r^2 \iint I_B(x, y) p(x) p(y) \, dx dy \\ &\approx 4 \times \frac{1}{n} \sum_{i=1}^n I_B(x_i, y_i)\end{aligned}$$

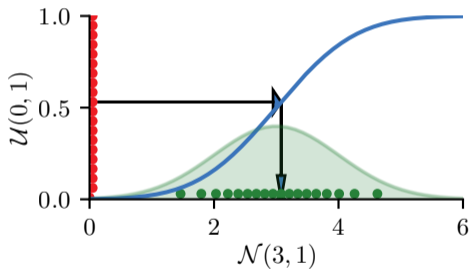
Sampling from simple distributions

inverse probability transform

$$U \sim \mathcal{U}(0, 1) \implies F^{-1}(U) \sim F$$

- F : **cumulative density function (CDF)** of target distribution
- F^{-1} : the inverse of CDF F
- proof:

$$\begin{aligned} \mathbf{P}(F^{-1}(U) \leq x) &= \mathbf{P}(U \leq F(x)) \\ &= F(x) \end{aligned}$$



Sampling from simple distributions

example: sampling from $\text{Exp}(\lambda)$

$$p_\lambda(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases} \implies F_\lambda(x) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$\implies F_\lambda^{-1}(u) = -\frac{\log(1-u)}{\lambda}$$

- $\text{dom } F_\lambda^{-1} = [0, 1)$
- sample $U \sim \mathcal{U}(0, 1)$, then $F_\lambda^{-1}(U) \sim \text{Exp}(\lambda)$

Rejection sampling

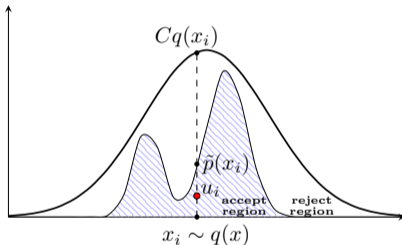
given:

- target distribution $p(x) = \tilde{p}(x)/Z_p$, where $Z_p = \int \tilde{p}(x) dx$
- proposal distribution $q(x)$, satisfying $Cq(x) \geq \tilde{p}(x)$ for some $C \in \mathbf{R}$

rejection sampling

1. Sample $x_i \sim q(x)$.
 2. Sample $u_i \sim \mathcal{U}(0, Cq(x_i))$.
 3. If $u_i > \tilde{p}(x_i)$, reject the sample, otherwise accept it.
-

- $q(x_i \mid \text{accept}) = p(x_i)$



Rejection sampling

proof

$$q(\text{accept} \mid x_i) = \int_0^{\tilde{p}(x_i)} \frac{1}{Cq(x_i)} du = \frac{\tilde{p}(x_i)}{Cq(x_i)}$$

$$\implies q(\text{propose and accept } x_i) = q(x_i)q(\text{accept} \mid x_i) = q(x_i) \frac{\tilde{p}(x_i)}{Cq(x_i)} = \frac{\tilde{p}(x_i)}{C}$$

$$\implies \int q(x_i)q(\text{accept} \mid x_i) dx_i = q(\text{accept}) = \frac{\int \tilde{p}(x_i) dx_i}{C} = \frac{Z_p}{C}$$

$$\implies q(x_i \mid \text{accept}) = \frac{q(x_i, \text{accept})}{q(\text{accept})} = \frac{\tilde{p}(x_i)}{C} \frac{C}{Z_p} = \frac{\tilde{p}(x_i)}{Z_p} = p(x_i)$$

Rejection sampling

$$q(\text{accept}) = \frac{Z_p}{C}$$

- $q(\text{accept}) = 1/C$ if \tilde{p} is a normalized target distribution
- example: $p(x) = \mathcal{N}(0, \sigma_p^2 I)$, $q(x) = \mathcal{N}(0, \sigma_q^2 I)$ ($\sigma_q^2 \geq \sigma_p^2$)
 - in n dimensions, optimum $C = (\sigma_q/\sigma_p)^n$
 - acceptance rate decreases exponentially with dimension

Importance sampling

direct importance sampling with normalized target distribution:

$$\mathbf{E} f(x) = \int f(x)p(x) dx = \int f(x)\frac{p(x)}{q(x)}q(x) dx$$

- $q(x)$: proposal distribution
 - $\text{supp } p \subseteq \text{supp } q$: the proposal is non-zero whenever the target is non-zero
- draw n samples $x \sim q(x)$

$$\mathbf{E} f(x) \approx \frac{1}{n} \sum_{i=1}^n \frac{p(x_i)}{q(x_i)} f(x_i) = \frac{1}{n} \sum_{i=1}^n w_i f(x_i)$$

- $w_i = \frac{p(x_i)}{q(x_i)}$: importance weight
- unbiased estimate of the true mean $\mathbf{E} f(x)$

Importance sampling

self-normalized importance sampling (SNIS) with unnormalized target distribution $\tilde{p}(x)$:

draw n samples $x \sim q(x)$

$$\begin{aligned}\mathbf{E} f(x) &= \int f(x)p(x) dx = \frac{\int f(x)\tilde{p}(x) dx}{\int \tilde{p}(x) dx} = \frac{\int \left(\frac{\tilde{p}(x)}{q(x)} f(x)\right) q(x) dx}{\int \left(\frac{\tilde{p}(x)}{q(x)}\right) q(x) dx} \\ &\approx \frac{\frac{1}{n} \sum_{i=1}^n \tilde{w}_i f(x_i)}{\frac{1}{n} \sum_{i=1}^n \tilde{w}_i} = \sum_{i=1}^n W_i f(x_i)\end{aligned}$$

- $q(x)$: proposal distribution, $\text{supp } \tilde{p} \subseteq \text{supp } q$
- $\tilde{w}_i = \frac{\tilde{p}(x_i)}{q(x_i)}$: unnormalized weight
- $W_i = \frac{\tilde{w}_i}{\sum_{i'=1}^n \tilde{w}_{i'}}$: normalized weight
- $p(x) \approx \hat{p}(x) = \sum_{i=1}^n W_i \delta(x - x_i)$, and $Z_p \approx \hat{Z}_p = \frac{1}{n} \sum_{i=1}^n \tilde{w}_i$

Outline

- Monte Carlo methods
 - Monte Carlo integration
 - Sampling from simple distributions
 - Rejection sampling
 - Importance sampling
- Markov chain Monte Carlo
 - Metropolis-Hastings algorithm
 - Gibbs sampling
 - Hamiltonian Monte Carlo

Markov chain Monte Carlo (MCMC)

idea

- construct a Markov chain on state space X , whose stationary distribution is the target density $p^*(x)$
- by drawing (correlated) samples x_0, x_1, \dots from the Markov chain, we can perform Monte Carlo integration w.r.t. p^*
- the initial samples from the chain do not come from the stationary distribution, and should be discarded (**burn in**)

Metropolis-Hastings (MH) algorithm

idea

at each step:

- propose to move from x to x' with proposal distribution $q(x' | x)$
- decide whether to accept this proposal, or to reject it, according to some acceptance probability A
- if the proposal is accepted, the new state is x' , otherwise the new state is the same as the current state x

Metropolis-Hastings (MH) algorithm

acceptance probability A

- symmetric proposal $q(x' | x) = q(x | x')$:

$$A = \min \left\{ 1, \frac{p^*(x')}{p^*(x)} \right\}$$

- if x' is more probable than x , we definitely move there (since $\frac{p^*(x')}{p^*(x)} > 1$)
- if x' is less probable than x , we move there depending on the probability ratio $\frac{p^*(x')}{p^*(x)}$

- asymmetric proposal $q(x' | x) \neq q(x | x')$: Hastings correction

$$A = \min \{1, \alpha\}, \quad \alpha = \frac{p^*(x')q(x | x')}{p^*(x)q(x' | x)}$$

Metropolis-Hastings (MH) algorithm

- we should be able to **evaluate** target distribution $p^*(x)$
- valid proposal q : $\text{supp } p^* \subseteq \cup_x \text{supp } q(\cdot | x)$
- sampling from unnormalized target distribution $p^*(x) = \frac{1}{Z_p} \tilde{p}(x)$:

$$\alpha = \frac{(\tilde{p}(x')/Z_p)q(x | x')}{(\tilde{p}(x)/Z_p)q(x' | x)} = \frac{\tilde{p}(x')q(x | x')}{\tilde{p}(x)q(x' | x)}$$

Metropolis-Hastings (MH) algorithm

given proposal distribution q .

initialize x .

repeat

1. Sample $x' \sim q(x' | x)$.
2. Compute $\alpha := \frac{p^*(x')q(x|x')}{p^*(x)q(x'|x)}$.
3. Compute acceptance probability $A := \min\{1, \alpha\}$.
4. Sample $u \sim \mathcal{U}(0, 1)$.
5. Set new sample to $x := \begin{cases} x' & u \leq A \text{ (accept)} \\ x & u > A \text{ (reject)}. \end{cases}$

until number of iterations reached.

Metropolis-Hastings (MH) algorithm

convergence analysis

- MH defines a Markov chain with transition matrix

$$p(x' | x) = \begin{cases} q(x' | x)A(x' | x) & x' \neq x \\ q(x | x) + \sum_{x' \neq x} q(x' | x)(1 - A(x' | x)) & \text{otherwise} \end{cases}$$

- if the target distribution p^* is the stationary distribution of this Markov chain, then it satisfies the **detailed balance** criterion

$$p(x' | x)p^*(x) = p(x | x')p^*(x')$$

Metropolis-Hastings (MH) algorithm

proof

- assume that the Markov chain is ergodic and irreducible
- given two states x and x' , we have

$$p^*(x)q(x' | x) < p^*(x')q(x | x') \quad \text{or} \quad p^*(x)q(x' | x) \geq p^*(x')q(x | x')$$

- without losing generality, we can assume

$$\begin{aligned} p^*(x)q(x' | x) &> p^*(x')q(x | x') \\ \implies \alpha(x' | x) &= \frac{p^*(x')q(x | x')}{p^*(x)q(x' | x)} < 1 \\ \implies A(x' | x) &= \alpha(x' | x) \quad \text{and} \quad A(x | x') = 1 \end{aligned}$$

Metropolis-Hastings (MH) algorithm

$$\implies \begin{cases} p(x' | x) = q(x' | x)A(x' | x) = q(x' | x) \frac{p^*(x')q(x | x')}{p^*(x)q(x' | x)} = \frac{p^*(x')}{p^*(x)} q(x | x') \\ p(x | x') = q(x | x')A(x | x') = q(x | x') \end{cases}$$

$$\implies p^*(x)p(x' | x) = p^*(x')q(x | x')$$

$$\implies p^*(x)p(x' | x) = p^*(x')p(x | x')$$

\implies detailed balance is satisfied by p^*

Metropolis-Hastings (MH) algorithm

proposal distributions q

- independence sampler: $q(x' | x) = q(x')$, e.g., $x' \sim \mathcal{N}(\mu, \Sigma)$
- random walk Metropolis (RWM):

$$x' \sim \mathcal{N}(x, \sigma^2 I) \iff (x' - x) \sim \mathcal{N}(0, \sigma^2 I)$$

- composing proposals:

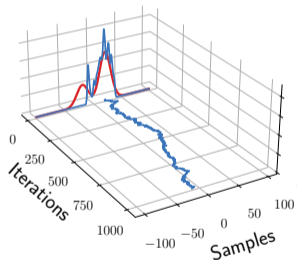
$$q(x' | x) = \sum_{i=1}^m w_i q_i(x' | x)$$

- w : mixing weights, and $\mathbf{1}^T w = 1$
- if each q_i is valid individually and $w \succeq 0$, the overall mixture proposal q is also valid

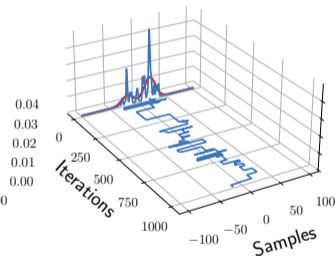
Metropolis-Hastings (MH) algorithm

example: sampling from a mixture of Gaussians with RWM

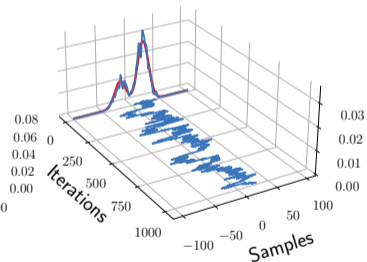
MH with $\mathcal{N}(0, 1^2)$ proposal



MH with $\mathcal{N}(0, 500^2)$ proposal



MH with $\mathcal{N}(0, 8^2)$ proposal



Gibbs sampling

- a special case of MH algorithm
- exploiting conditional independence properties of a graphical model on X to automatically create a good proposal q , with 100% acceptance probability

idea: sample each variable in turn, conditioned on the values of all the other variables in the distribution

example: sample some variable $X \in \mathbf{R}^3$ according to

$$x'_1 \sim p(x'_1 \mid x_2, x_3)$$

$$x'_2 \sim p(x'_2 \mid x'_1, x_3)$$

$$x'_3 \sim p(x'_3 \mid x'_1, x'_2)$$

Gibbs sampling

- $p(x'_i | x_{-i})$: **full conditional** for variable X_i
- if X_i is a known variable, we do not sample it, but it may be used as input to the another conditional distributions
- if we represent $p(x)$ as a graphical model,

$$x'_i \sim p(x'_i | x_{-i}) = p(x'_i | \mathbf{mb}(x_i))$$

- we should have access to the analytical expression of target distribution p to derive the full conditionals for each X_i

Gibbs sampling

connections to MH

- Gibbs sampling is a special case of MH using a sequence of proposals

$$q_i(x' | x) = p(x' | x_{-i})I_{x_{-i}}(x'_{-i}), \quad i = 1, \dots, n$$

– $X \in \mathbf{R}^n$

– $I_{x_{-i}}$: indicator function

- proof of 100% acceptance probability: for sampling each X_i , we have $x'_{-i} = x_{-i}$, thus

$$\begin{aligned} \alpha &= \frac{p(x')q_i(x | x')}{p(x)q_i(x' | x)} = \frac{p(x'_i | x'_{-i})p(x'_{-i})p(x_i | x'_{-i})}{p(x_i | x_{-i})p(x_{-i})p(x'_i | x_{-i})} \\ &= \frac{p(x'_i | x_{-i})p(x_{-i})p(x_i | x_{-i})}{p(x_i | x_{-i})p(x_{-i})p(x'_i | x_{-i})} = 1 \end{aligned}$$

Gibbs sampling

example: Gibbs sampling for Ising models

for a 2-dimensional lattice $G = (X, E)$:

- lattice model:

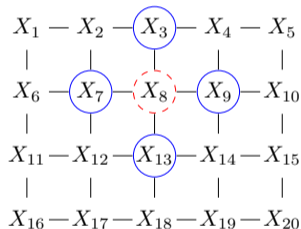
$$p(x) = \frac{1}{Z_p} \prod_{(X_i, X_j) \in E} \psi_{ij}(x_i, x_j)$$

– $\psi_{ij}(x_i, x_j)$: potential function of clique $C = \{X_i, X_j\}$

- Ising model:

- X_i are binary for all $i = 1, \dots, n$
- potential function expressed as

$$\psi_{ij}(x_i, x_j) = \begin{cases} e^J & x_i = x_j \\ e^{-J} & x_i \neq x_j \end{cases} = \exp(Jx_i x_j)$$



Gibbs sampling

to sample for an n -dimensional random vector X following an Ising model,

$$p(x_i | x_{-i}) \propto \prod_{X_j \in \mathbf{adj}(X_i)} \psi_{ij}(x_i, x_j), \quad i = 1, \dots, n$$

$$\begin{aligned} \implies p(x_i = +1 | x_{-i}) &= \frac{\prod_{X_j \in \mathbf{adj}(X_i)} \psi_{ij}(x_i = +1, x_j)}{\prod_{X_j \in \mathbf{adj}(X_i)} \psi_{ij}(x_i = +1, x_j) + \prod_{X_j \in \mathbf{adj}(X_i)} \psi_{ij}(x_i = -1, x_j)} \\ &= \frac{\exp(J \sum_{X_j \in \mathbf{adj}(X_i)} x_j)}{\exp(J \sum_{X_j \in \mathbf{adj}(X_i)} x_j) + \exp(-J \sum_{X_j \in \mathbf{adj}(X_i)} x_j)} \\ &= \frac{\exp(J\eta_i)}{\exp(J\eta_i) + \exp(-J\eta_i)} \end{aligned}$$

- $\eta_i = \sum_{X_j \in \mathbf{adj}(X_i)} x_j$

Gibbs sampling

Metropolis within Gibbs

- use MH algorithm to sample from the full conditionals

to sample $x'_i \sim p(x'_i \mid x'_{1:i-1}, x_{i+1:n})$:

-
1. Propose $x''_i \sim q(x''_i \mid x_i)$.
 2. Compute the acceptance probability $A_i = \min\{1, \alpha_i\}$, where

$$\alpha_i = \frac{p(x''_i \mid x'_{1:i-1}, x_{i+1:n})q(x_i \mid x''_i)}{p(x_i \mid x'_{1:i-1}, x_{i+1:n})q(x''_i \mid x_i)}.$$

3. Sample $u \sim \mathcal{U}(0, 1)$.
 4. Set $x'_i = x''_i$ if $u < A_i$, and $x'_i = x_i$ otherwise.
-

Hamiltonian Monte Carlo (HMC)

idea: create proposal q based on gradient information

Hamiltonian mechanics

the total energy of a particle rolling around an energy landscape is

$$\mathcal{H}(\theta, v) = \mathcal{E}(\theta) + \mathcal{K}(v)$$

- $\theta \in \mathbf{R}^n$: position
- $v \in \mathbf{R}^n$: momentum
- (θ, v) : phase space
- $\mathcal{E}(\theta)$: potential energy
- $\mathcal{K}(v) = \frac{1}{2}v^T \Sigma^{-1}v$: kinetic energy, where $\Sigma \in \mathbf{S}_{++}^n$ is the mass matrix
- $\mathcal{H}(\theta, v)$: total energy (Hamiltonian)

Hamiltonian Monte Carlo (HMC)

- trajectory of the particle can be obtained by solving **Hamilton's equations**:

$$\begin{cases} \frac{d\theta}{dt} = \frac{\partial \mathcal{H}}{\partial v} = \frac{\partial \mathcal{K}}{\partial v} \\ \frac{dv}{dt} = -\frac{\partial \mathcal{H}}{\partial \theta} = -\frac{\partial \mathcal{E}}{\partial \theta} \end{cases}$$

- energy is conserved under Hamiltonian mechanics, since

$$\frac{d\mathcal{H}}{dt} = \sum_{i=1}^n \left(\frac{\partial \mathcal{H}}{\partial \theta_i} \frac{d\theta_i}{dt} + \frac{\partial \mathcal{H}}{\partial v_i} \frac{dv_i}{dt} \right) = \sum_{i=1}^n \left(\frac{\partial \mathcal{H}}{\partial \theta_i} \frac{\partial \mathcal{H}}{\partial v_i} - \frac{\partial \mathcal{H}}{\partial \theta_i} \frac{\partial \mathcal{H}}{\partial v_i} \right) = 0$$

Hamiltonian Monte Carlo (HMC)

to solve Hamilton's equations in discrete time:

- **Euler's method:**

$$\begin{cases} v_{t+1} = v_t + \eta \frac{dv}{dt} \Big|_{\theta=\theta_t, v=v_t} & = v_t - \eta \frac{\partial \mathcal{E}}{\partial \theta} \Big|_{\theta=\theta_t} \\ \theta_{t+1} = \theta_t + \eta \frac{d\theta}{dt} \Big|_{\theta=\theta_t, v=v_t} & = \theta_t + \eta \frac{\partial \mathcal{K}}{\partial v} \Big|_{v=v_t} \end{cases} .$$

- η : step size
- if $\mathcal{K}(v) = \frac{1}{2}v^T \Sigma^{-1}v$, the second term reduces to

$$\theta_{t+1} = \theta_t + \eta \Sigma^{-1}v_t$$

Hamiltonian Monte Carlo (HMC)

- **modified Euler's method:**

$$\begin{cases} v_{t+1} = v_t + \eta \frac{dv}{dt} \Big|_{\theta=\theta_t, v=v_t} & = v_t - \eta \frac{\partial \mathcal{E}}{\partial \theta} \Big|_{\theta=\theta_t} \\ \theta_{t+1} = \theta_t + \eta \frac{d\theta}{dt} \Big|_{\theta=\theta_t, v=v_{t+1}} & = \theta_t + \eta \frac{\partial \mathcal{K}}{\partial v} \Big|_{v=v_{t+1}} \end{cases} .$$

- slightly more accurate than Euler's method
- asymmetry of this method can cause some theoretical problems

Hamiltonian Monte Carlo (HMC)

- leapfrog integrator:

$$\begin{cases} v_{t+1/2} = v_t - \frac{\eta}{2} \frac{\partial \mathcal{E}}{\partial \theta} \Big|_{\theta=\theta_t} \\ \theta_{t+1} = \theta_t + \eta \frac{\partial \mathcal{K}}{\partial v} \Big|_{v=v_{t+1/2}} \\ v_{t+1} = v_{t+1/2} - \frac{\eta}{2} \frac{\partial \mathcal{E}}{\partial \theta} \Big|_{\theta=\theta_{t+1}} \end{cases} .$$

- symmetrized version of the modified Euler's method
- can be extended to multiple leapfrog steps, i.e., performing a half step update of v at the beginning and end of the trajectory, and alternating between full step updates of θ and v in between

Hamiltonian Monte Carlo (HMC)

the HMC algorithm

- establish a new target by introducing an auxiliary variable v to the initial target distribution $p(\theta)$:

$$p(\theta, v) = \frac{1}{Z} \exp(-\mathcal{H}(\theta, v)) = \frac{1}{Z} \exp\left(-\mathcal{E}(\theta) - \frac{1}{2}v^T \Sigma^{-1}v\right)$$

- after sampling w.r.t. $p(\theta, v)$, we just 'throw away' the v 's so that

$$p(\theta) = \int p(\theta, v) dv = \frac{1}{Z_\theta} e^{-\mathcal{E}(\theta)} \int \frac{1}{Z_v} e^{-\frac{1}{2}v^T \Sigma^{-1}v} dv = \frac{1}{Z_\theta} e^{-\mathcal{E}(\theta)}$$

Hamiltonian Monte Carlo (HMC)

suppose the previous state of the Markov chain is (θ_{t-1}, v_{t-1}) , to sample the next state,

- set the initial position to $\theta'_0 = \theta_{t-1}$, and sample a new random momentum $v'_0 \sim \mathcal{N}(0, \Sigma)$
- starting from (θ'_0, v'_0) , perform L leapfrogs to get the proposed state $(\theta^*, v^*) = (\theta'_L, v'_L)$
- check divergence of the simulated trajectory, if $\mathcal{H}_0 \neq \mathcal{H}_L$, reject the sample
- if the trajectory is not diverged, compute the MH acceptance probability as

$$A = \min \left\{ 1, \frac{p(\theta^*, v^*)}{p(\theta_{t-1}, v_{t-1})} \right\} = \min \{1, \exp(-\mathcal{H}(\theta^*, v^*) + \mathcal{H}(\theta_{t-1}, v_{t-1}))\}$$

– the transition probabilities cancel since the proposal is reversible

- accept the proposal with probability A by setting $(\theta_t, v_t) = (\theta^*, v^*)$, otherwise reject it $((\theta_t, v_t) = (\theta_{t-1}, v_{t-1}))$

Hamiltonian Monte Carlo (HMC)

given the number of leapfrog steps L , the step size η , and the covariance matrix Σ .

repeat

Generate random momentum $v_{t-1} \sim \mathcal{N}(0, \Sigma)$.

Set $(\theta'_0, v'_0) := (\theta_{t-1}, v_{t-1})$.

Half step for momentum: $v'_{1/2} := v'_0 - \frac{\eta}{2} \nabla \mathcal{E}(\theta'_0)$.

for $l = 1, \dots, L - 1$,

$\theta'_l := \theta'_{l-1} + \eta \Sigma^{-1} v'_{l-1/2}$.

$v'_{l+1/2} := v'_{l-1/2} - \eta \nabla \mathcal{E}(\theta'_l)$.

Full step for location: $\theta'_L := \theta'_{L-1} + \eta \Sigma^{-1} v'_{L-1/2}$.

Half step for momentum: $v'_L := v'_{L-1/2} - \frac{\eta}{2} \nabla \mathcal{E}(\theta'_L)$.

Obtain proposal $(\theta^*, v^*) := (\theta'_L, v'_L)$.

Compute acceptance probability $A := \min \{1, \exp(-\mathcal{H}(\theta^*, v^*) + \mathcal{H}(\theta_{t-1}, v_{t-1}))\}$.

Set $\theta_t := \theta^*$ with probability A , other wise $\theta_t := \theta_{t-1}$.

until number of iterations reached.

- we must pick a random momentum at the start of each iteration to ensure the sampler explores the full space