Both the number of steps s and the step size h need to be "tuned" for good performance. Based on some idealized theoretical situations, we want to tune these such that the acceptance rate is around 65%.

Now at the nth iteration we are in state $\left(\vec{\theta}^{(n)}, \vec{p}^{(n)}\right)$. We run the algorithm to propose a new state $\left(\vec{\theta}^{*}, \vec{p^{*}}\right)$.

Now multiply $\vec{p^{*}}$ by -1 to make the proposal symmetric.

Set $\vec{\theta}^{(n+1)} = \vec{\theta}^{*}$ with probability α where

$$\alpha = \min\left(1, \frac{\exp(-H(\vec{\theta}^{*}, \vec{p^{*}})}{\exp(-H(\vec{\theta}^{(n)}, \vec{p}^{(n)})}\right),$$

otherwise set $\vec{\theta}^{(n+1)} = \vec{\theta}^{(n)}$. At this point, we don't need $\vec{p}$ any more so we don't need to update it.

4.6 Riemannian Manifold HMC (RMHMC)

In the kinetic energy term $K(\vec{p}) = \vec{p}^{\mathrm{T}} M^{-1} \vec{p}$ the matrix M is a mass matrix and can be any sensible matrix as long as it is SPD. So perhaps we can find a good matrix M? Turns out that in many cases we can.

Definition 45 (Manifold) *A* manifold *is a topological space that (locally) resembles Euclidean space.*

Definition 46 (Riemann Manifold) *A* Riemann Manifold *is a real smooth (differentiable) manifold M equipped with an inner product g_p on the tangent space T_pM at each point $p \in M$ that varies smoothly in the sense that if X and Y are vector fields on M then*

$$p \to g_p\left(X(p), Y(p)\right)$$

is a smooth function.

Definition 47 (Riemann Metric (Tensor)) *The family of inner products g_p, one for each p, is called a* Riemann metrix (tensor)*.*

4.6.1 Fisher-Rao Metric Tensor

Take two parametrized densities $\pi(y \mid \boldsymbol{\theta})$ and $\pi(y \mid \boldsymbol{\theta} + \delta\boldsymbol{\theta})$. Rao (1945) formalized the notion of distance between these two densities. The distance takes the quadratic form:

$$\delta\boldsymbol{\theta}^{\mathrm{T}} G(\boldsymbol{\theta})\boldsymbol{\theta}. \tag{36}$$

where

$$G(\boldsymbol{\theta}) = -\mathbb{E}_{\pi[y|\boldsymbol{\theta}]}\left[\frac{\partial^2 \ln \pi(y \mid \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] = \mathrm{Cov}\left[\frac{\partial \ln \pi(y \mid \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right].$$

That is, $G(\boldsymbol{\theta})$ is the expectd Fisher information. $G(\boldsymbol{\theta})$ is SPD and Rao noted that it is a position specific matrix of a Riemann manifold.

In our Bayesian framework we take

$$G(\boldsymbol{\theta}) = -\mathbb{E}_{\pi[y|\boldsymbol{\theta}]}\left[\frac{\partial^2 \ln \pi(y, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] = -\mathbb{E}_{\pi[y|\boldsymbol{\theta}]}\left[\frac{\partial^2 \ln \pi(y \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right]$$

which is the expected Fisher information plus the negative Hessian of the log prior. $G(\boldsymbol{\theta})$ is related to the curvature of the manifold. Thus, in (36), the larger $G(\boldsymbol{\theta})$ the farther away the two densities.

Now, recall that

$$\frac{d\vec{q}}{dt} = M^{-1}\vec{p} \quad \Rightarrow \quad \vec{p} = M\frac{d\vec{q}}{dt}.$$

Under the matrix M we have

$$\left|\left|\frac{d\vec{q}}{dt}\right|\right|_M^2 = \frac{d\vec{q}}{dt}^{\mathrm{T}} M \frac{d\vec{q}}{dt} = \vec{p}^{\mathrm{T}} M^{-1}\vec{p},$$

and defined on the Riemannian manifold

$$\left|\left|\frac{d\vec{q}}{dt}\right|\right|_{G(\vec{q})}^2 = \frac{d\vec{q}}{dt}^{\mathrm{T}} G(\vec{q}) \frac{d\vec{q}}{dt} = \vec{p}^{\mathrm{T}} G^{-1}(\vec{q})\vec{p},$$

Therefore we can take

$$K(\vec{q}, \vec{p}) = \vec{p}^{\mathrm{T}} G^{-1}(\vec{q})\vec{p}$$

and

$$H(\vec{q}, \vec{p}) = U(\vec{q}) + K(\vec{q}, \vec{p}) = -\ln \pi(\vec{q} \mid y) + \frac{1}{2}\ln\left((2\pi)^d \left|G(\vec{q})\right|\right) + \frac{1}{2}\vec{p}^{\mathrm{T}} G^{-1}(\vec{q})\vec{p}.$$

Unlike in the HMC algorithm, the Hamiltonian does not factorize into $U(\vec{q})$ and $K(\vec{p})$.

The Hamiltonian equations in this case are now

$$\begin{aligned}\frac{dq_i}{dt} &= \frac{\partial H}{\partial p_i} = \left(G^{-1}(\vec{q})\vec{p}\right)_i \\ \frac{dp_i}{dt} &= -\frac{\partial H}{\partial q_i} = \frac{\partial \ln \pi(\vec{q} \mid y)}{\partial q_i} - \frac{1}{2}\mathrm{tr}\left(G^{-1}(\vec{q})\frac{\partial G(\vec{q})}{\partial q_i}\right) + \frac{1}{2}\vec{p}^{\mathrm{T}} G^{-1}(\vec{q})\frac{\partial G(\vec{q})}{\partial q_i} G^{-1}(\vec{q})\vec{p}.\end{aligned}$$

This leads us to the generalized leap frog algorithm

$$\vec{p}\left(t+\frac{h}{2}\right) = \vec{p}(t) - \frac{h}{2}\nabla_{\vec{q}} H\left(\vec{q}(t), \vec{p}\left(t+\frac{h}{2}\right)\right) \tag{37}$$

$$\vec{q}(t+h) = \vec{q}(t) + \frac{h}{2}\left[\nabla_{\vec{p}} H\left(\vec{q}(t), \vec{p}\left(t+\frac{h}{2}\right)\right) + \nabla_{\vec{p}} H\left(\vec{q}(t+h), \vec{p}\left(t+\frac{h}{2}\right)\right)\right] \tag{38}$$

$$\vec{p}\,(t+h) = \vec{p}\left(t+\frac{h}{2}\right) - \frac{h}{2}\nabla_{\vec{q}} H\left(\vec{q}(t+h), \vec{p}\left(t+\frac{h}{2}\right)\right). \tag{39}$$

When H is separable, then the generalized leap frop reduces to the leap frog algorithm. If H is not separable, then (37) and (38) are implicitly defined. You can see that (37) and (38) are fixed point problems and must be solved iteratively within each generalized leap frog step.

A fixed point problem takes the general form:

$$x = f(x).$$

So we must find x such that $x = f(x)$. The solution can be found numerically. Start at $x = x_0$ and iterate

$$x_{n+1} = f(x_n), \quad n = 0, 1, \ldots$$

until

$$|f(x_n) - x_{n+1}| < \varepsilon.$$