Hybrid (Hamiltonian) Monte Carlo

• We would like to increase the acceptance rate to improve efficiency, and explore the target distribution efficiently (‘good mixing’)

• We have a hard problem in many dimensions. Solution:

  • Make things harder: add in $M$ auxiliary variables, one for each parameter in the model.

  • Imagine each of the parameters in the problem as a coordinate.

  • Target distribution $\rightarrow$ effective potential

  • For each coordinate HMC generates a generalised momentum.

  • It then samples from the extended target distribution in $2M$ dimensions.

HMC

• HMC explores this $2M$-dimensional space by treating the problem as a dynamical system, and evolving the phase space coordinates by solving the dynamical equations.

• Finally, it ignores the momenta (marginalising, as in MCMC), and this gives a sample of the original target distribution.

• May help with decorrelating the points in the chain.

• Invented by particle physicists (Duan et al 1987)
Theory

• Potential $U(\theta) = -\ln p(\theta)$
• For each $\theta_\alpha$, generate a momentum $u_\alpha$.
• K.E. $K = u^T u / 2$
• Define a Hamiltonian
  $$H(\theta, u) \equiv U(\theta) + K(u)$$
• and define an extended target density
  $$p(\theta, u) = \exp [-H(\theta, u)]$$

Magic of HMC

• Evolve as a dynamical system
  $$\dot{\theta}_\alpha = u_\alpha$$
  $$\dot{u}_\alpha = -\frac{\partial H}{\partial \theta_\alpha}$$

• $H$ remains constant, so extended target density is uniform – all points get accepted!
• Also, you can make big jumps – good mixing, if you generate a new $u$ each time
Complications

- Evolving the system takes time. Take big steps.
- We don’t know the complete $U = -\ln p$ (it’s what we are looking for)
- If we can quickly evaluate the derivative of $U$, fine.
- We might approximate $U$ (from a short MCMC)
  $$U = \frac{1}{2} (\theta - \theta_0) \alpha C_{\alpha \beta}^{-1} (\theta - \theta_0) \beta$$
- $H$ is therefore not constant
- Use Metropolis-Hastings. Accept new point with probability
  $$\min \{1, \exp [-H(\theta^*, u^*) + H(\theta, u)]\}$$

Algorithm

Hamiltonian Monte Carlo

1: initialize $x_{(0)}$
2: for $i = 1$ to $N_{samples}$
3: \hspace{1em} $u \sim \mathcal{N}(0,1)$
4: \hspace{1em} $(x^*_{(i)}, u^*_{(i)}) = (x_{(i-1)}, u)$
5: \hspace{1em} for $j = 1$ to $N$
6: \hspace{2em} make a leapfrog move: $(x^*_{(j)}, u^*_{(j)}) \rightarrow$
7: \hspace{2em} $(x^*_{(j)}, u^*_{(j)})$
8: \hspace{1em} $(x^*, u^*) = (x_{(N)}, u_{(N)})$
9: \hspace{1em} draw $\alpha \sim (0,1)$
10: \hspace{1em} if $\alpha < \min\{1, e^{-H(x^*, u^*) - H(x, u)}\}$
11: \hspace{2em} $x_{(i)} = x^*$
12: \hspace{1em} else
13: \hspace{2em} $x_{(i)} = x_{(i-1)}$
14: end for

From Hajian 2006
**HMC vs MCMC**

Typical speed-ups: factor 4.

---

** Supernova distances**

- Luminosity Distance depends on $\Omega_m$ and $H_0$ (for flat Universe)

\[
D_L = \frac{c(1 + z)}{H_0} \int_0^z \frac{dz'}{\sqrt{\Omega_m(1 + z')^3 + 1 - \Omega_m}}
\]

\[
f = \frac{L}{4\pi D_L^2}
\]
MCMC run

\[ C = \begin{pmatrix} 7.1 \times 10^{-5} & -1.9 \times 10^{-4} \\ -1.9 \times 10^{-4} & 1.0 \times 10^{-3} \end{pmatrix} \]

Acceptance rate 0.15

HMC

Acceptance rate 0.4

Still work to do...