Hamiltonian Monte Carlo

As we have seen, a major problem with Gibbs and MH sampling can be that the resulting Markov chains exhibit undesirable *random walk* behaviour and take many iterations to *diffuse* slowly around the sample space. *Hamiltonian Monte Carlo* (HMC), sometimes known as *hybrid Monte Carlo*, is a Metropolis-like method that uses gradient information about the target distribution and the equations of Hamiltonian dynamics to propose transitions efficiently in the sample space[1] [2] [3].

HMC is a Metropolis-like method in that it requires only a function proportional to the target PDF, not the normalised distribution. Unlike previous algorithms we have studied, HMC will also required that we are able to evaluate the gradient of this function with respect to the parameters. It is convenient to work with the logarithm of the target PDF,

$$\log P(x) = -E(x) + \text{const}, \tag{1}$$

where the quantity $E(x)$ is called the *potential energy*. The negative logarithm of the PDF of the target distribution will define a gravitational *potential well* for our particle. The d-dimensional gradient vector will be denoted $\nabla E(x)$.

Points $x \in \mathcal{X}$ in the parameter space are called *positions*. HMC augments these with *momenta* $p \in T_x\mathcal{X}$ in the tangent space which has the same dimension as $\mathcal{X}$. HMC also introduces a new distribution $\mathcal{Q}$ on the tangent space $T_x\mathcal{X}$. This distribution should be easy to sample from; a common choice for $\mathcal{Q}$ is a multivariate Gaussian;

$$\log Q(p) = -K(p) + \text{const}, \quad \text{where} \quad K(p) = \frac{1}{2} p^{\mathrm{T}} M^{-1} p. \tag{2}$$

[1] Duane, Kennedy, Pendleton & Roweth (1987) "Hybrid Monte Carlo", Physics Letters B, 195 (2) 216–222 doi:10.1016/0370-2693(87)91197-X.

[2] Neal (1993) "Probabilistic inference using Markov chain Monte Carlo methods", Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, link.

[3] Neal (2011) "Handbook of Markov Chain Monte Carlo", chapter 5 "MCMC Using Hamiltonian Dynamics", CRC Press, link.

Here $K(p)$ is called the *kinetic energy* and M is a symmetric, positive definite *mass matrix*.

Position and momentum pairs (x, p) exist in a space of twice the dimensionality of the sample space $\mathcal{X}$ (technically, the tangent bundle). HMC defines an *augmented distribution* R (sometimes called the *canonical distribution*) on this tangent bundle. The *augmented distribution* constructed from P and $\mathcal{Q}$; it is defined as having log-PDF equal to the negative of the *Hamiltonian*,

$$\log R(x, p) = -\mathcal{H}(x, p) + \text{const}, \quad \text{where} \quad \mathcal{H}(x, p) = E(x) + K(p). \tag{3}$$

The counter-intuitive starting point of HMC is that we double dimensionality of the distribution we are working with. This might seem silly but go with it for the moment. Instead of try to sample $x \sim P$ directly in d dimensions, we will instead try to sample $(x, p) \sim R$ in $2d$ dimensions. We will then simply throw away the momentum p variable leaving $x \sim P$, as desired. This works because we have defined the new distribution R such that the target distribution is recovered when we marginalise over p;

$$P(x) = \int \mathrm{d}p \; R(x, p). \tag{4}$$

This is only useful if we can find an efficient way of sampling from R. In order to do this, HMC introduces a *fictitious time* parameter, t. The position $x(t)$ and momentum $p(t)$ are promoted to functions of t and evolved according to the equations of *Hamiltonian dynamics*,

$$\frac{\mathrm{d}x^k}{\mathrm{d}t} = \frac{\partial \mathcal{H}}{\partial p^k} \quad \text{and} \quad \frac{\mathrm{d}p^k}{\mathrm{d}t} = -\frac{\partial \mathcal{H}}{\partial x^k}. \tag{5}$$

For any duration s, these equations define a map from states at time t to states at time $t + s$; i.e. $T_s : (x(t), p(t)) \to (x(t + s), p(t + s))$. The Hamiltonian $\mathcal{H}$, and hence the map T_s, do not depend explicitly on t. Note, the continuous parameter t introduced as part of the Hamiltonian dynamics has nothing to do with the discrete index i which labels points in the chain undergoing Markovian dyanamics.

Hamiltonian dynamics has three well-known properties relevant for HMC. (Proofs can be found in any advanced classical mechanics textbook.)

Lemma 0.0.1. *Hamiltonian dynamics is reversible, meaning T_s has an inverse, T_{-s}.*

Lemma 0.0.2. *Hamiltonian dynamics is volume preserving. Consider all points (x, p) in some region A of phase space (with volume V_A). If B is the image of A under T_s (with volume V_B) then $V_A = V_B$. This result is known as Liouville's theorem.*

Lemma 0.0.3. *Hamiltonian dynamics conserves the Hamiltonian, meaning* $\mathrm{d}\mathcal{H}/\mathrm{d}t = 0$.

It is not generally possible to solve Hamilton's equations exactly. Instead, it is necessary to integrate Eqs. 5 numerically. This is usually done using the *leapfrog method* where a time step Δt from $(x(t), p(t))$ to $(x(t + \Delta t), p(t + \Delta t))$ is done in three stages: a half step in momentum to find $p_i(t + \frac{1}{2}\Delta t)$, a full step in position to find $x_i(t + \Delta t)$, and finally another half step in position to find $p_i(t + \Delta t)$. For a particular choice of Δt, applying the leapfrog method (Alg. 0.1) repeatedly produces an *approximate Hamiltonian dynamics*.

Algorithm 0.1 Leapfrog step

1: **procedure** LEAPFROG(x, p, Δt, M)
2: $\quad p \leftarrow p - \frac{1}{2}\Delta t\, \nabla|_x E(x)$ ▷ half step for momentum
3: $\quad x \leftarrow x + \Delta t\, M^{-1} \cdot p$ ▷ full step for position
4: $\quad p \leftarrow p - \frac{1}{2}\Delta t\, \nabla|_x E(x)$ ▷ half step for momentum
5: $\quad$ **return** x, p
6: **end procedure**

The reason for choosing the leapfrog integrator is because it is *sympletic*, meaning that it is exactly *time reversible* (lemma 0.0.4) and *volume preserving* (lemma 0.0.5). This is not true of other numerical integrators, such as Euler's method or Runge-Kutta integration, which only have these properties approximately in the limit of small Δt.

Lemma 0.0.4. *The approximate leapfrog Hamiltonian dynamics is exactly reversible.*

Proof. Consider the state $(x(t + \Delta t), p(t + \Delta t))$. Reversing the sign of p gives $(x(t + \Delta t), -p(t+\Delta t))$. It is easy to check that applying the leapfrog to this state gives $(x(t), -p(t))$. Reversing the sign of p again gives $(x(t), p(t))$, as required. □

Lemma 0.0.5. *The approximate leapfrog Hamiltonian dynamics is exactly volume preserving.*

Proof. This follows from the fact that all three steps of the leapfrog are "shear transformations" in which one set of variables (either x or p) change by an amount that depends only on the other. Shear transformations have unit Jacobian. E.g. consider the middle step of the leapfrog (line 3); this defines a transformation of the form

$$\begin{pmatrix} x \\ p \end{pmatrix} \to \begin{pmatrix} x' \\ p' \end{pmatrix} = \begin{pmatrix} x + \Delta t f(p) \\ p \end{pmatrix}, \quad \text{with Jacobian } J = \begin{vmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial p} \\ \frac{\partial p'}{\partial x} & \frac{\partial p'}{\partial p} \end{vmatrix} = \begin{vmatrix} 1 & \Delta t \frac{\partial f}{\partial p} \\ 0 & 1 \end{vmatrix} = 1. \tag{6}$$

The other steps of the leapfrog are of similar form and all have unit determinant. The Jacobian of the composition of transformations is the product of the individual Jacobians. It follows that a full leapfrog step has unit determinant and is volume preserving. □

The approximate leapfrog Hamiltonian dynamics does not perfectly conserve the Hamiltonian due to discretisation errors in the numerical integration. However, for small Δt the Hamiltonian is approximately conserved; it is easy to show that

$$\mathcal{H}\big(x(t+\Delta t), p(t+\Delta t)\big) = \mathcal{H}\big(x(t), p(t)\big) + \mathcal{O}\big(\Delta t^2\big). \tag{7}$$

With all this setup in place, we are now in a position to state the HMC algorithm.

Algorithm 0.2 HMC

$x_0 \sim \alpha$ ▷ Initialise
$i \leftarrow 0$
while $i \geq 0$ **do** ▷ Iterate $i = 0, 1, 2, \ldots$
  $p \sim \mathcal{Q}$ ▷ Draw random momentum
  $x \leftarrow x_i$ ▷ Integrate from current chain position
  $H_{\text{initial}} \leftarrow \mathcal{H}(x, p)$ ▷ Initial Energy
  for $\ell = 0, 1, \ldots, L-1$ **do**
    $x, p \leftarrow \textsc{Leapfrog}(x, p, \Delta t, M)$ ▷ Integrate
  end for
  $H_{\text{final}} \leftarrow \mathcal{H}(x, p)$ ▷ Final Energy
  $a \leftarrow \exp\left(H_{\text{initial}} - H_{\text{final}}\right)$ ▷ MH acceptance probability
  $u \sim \mathcal{U}(0, 1)$
  if $u < a$ **then**
    $x_{i+1} \leftarrow y$ ▷ Markov transition (accept)
  else
    $x_{i+1} \leftarrow x_i$ ▷ Markov transition (reject)
  end if
  $i \leftarrow i + 1$
end while

The output of the HMC algorithm (Alg. 0.2) is the Markov chain $x_0, x_1, x_2 \ldots$

Box 0.1: Motivating the HMC algorithm

HMC can be viewed as a combination of the blocked Gibbs and MH algorithm, with a clever choice for the proposal distribution.

In addition to target distribution $P(x)$, we introduce another d-dimensional distribution $\mathcal{Q}(p)$. For simplicity, $\mathcal{Q}(p)$ is chosen to be a multivariate Gaussian.

We also introduce the augmented distribution $R(x,p) = P(x)\mathcal{Q}(p)$ with dimension $2d$. Note, that R is separable, so P and $\mathcal{Q}$ are the marginal distributions of R; i.e. $P(x) = \int \mathrm{d}p\ R(x,p)$ and $\mathcal{Q}(x) = \int \mathrm{d}x\ R(x,p)$. Note that P and $\mathcal{Q}$ are also the conditional distributions of R; i.e. $P(x) = R(x|p)$ and $\mathcal{Q}(p) = R(p|x)$.

The idea is to sample from the augmented distribution, $(x_i, p_i) \overset{\text{iid}}{\sim} R$, and simply discard the momentum variables to leave samples $x_i \overset{\text{iid}}{\sim} P$, as required.

Sampling from R is done with a blocked Gibbs algorithm where the position and momentum variables are the two parameter *blocks*. We iterate according to

$$p_{i+1} \sim R(p|x_i) \equiv \mathcal{Q}(p), \tag{i}$$
$$x_{i+1} \sim R(x|p_{i+1}) \equiv P(x). \tag{ii}$$

The momentum update is easy because $\mathcal{Q}$ was chosen to be easy to sample from. The position updates are done via a MH step. The clever part is in the choice of proposal. The proposal is made using the *leapfrog* to numerically integrate *Hamilton's equations*, for L steps; $(x_i(L\Delta t), p_i(L\Delta t))$. The leapfrog integration is *symmetric* and *volume preserving* which means is defines a symmetric Metropolis proposal. We accept the proposed point with probability $\min(a, 1)$, where $\log a$ is the difference between the Hamiltonian at the start and end of the integration.

The reason this proposal is so clever is that if the numerical integration is accurate and the Hamiltonian is approximately conserved then the acceptance probability is very high, $a \approx 1$ And it doesn't really matter if the numerical integration is a bit inaccurate because even if $a < 1$ this is still a valid MH step.

If L is sufficiently large and Δt is sufficiently small, this algorithm can take large steps in the target space with very high acceptance probabilities.

The HMC algorithm clearly defines a time-homogeneous Markov chain. Typically, the Markov chain will also be irreducible (although see discussion below). The HMC is designed such that the resulting Markov chain satisfies detailed balance with $\pi = P$.

The counter-intuitive starting point of HMC was the doubling the dimensionality of the space to be sampled. You would have been forgiven for thinking that this makes the problem harder. However, recall that Monte Carlo methods converge as $N_{\text{samples}}^{-1/2}$, regardless of dimensionality; the hard part is obtaining the N_{samples} independent samples! HMC increases the dimensionality in such a way that makes it possible to efficiently propose points, thereby hopefully reducing the IAT compared to, say, the standard MH algorithm. The acceptance probability of the HMC algorithm is $a = \min(1, \exp(\mathcal{H}_{\text{initial}} - \mathcal{H}_{\text{final}}))$ which is extremely high because the (discretised) Hamiltonian dynamics (approximately) conserves the Hamiltonian. This can allow us to produce more independent samples for the same computational cost.

Compared to the MH algorithm, HMC has the major benefit that the user doesn't have to provide a proposal distribution.

A potential drawback of HMC is that it requires the derivatives of $-\log P(x)$ w.r.t. x. Also, because the HMC algorithm uses derivatives, it can only be applied to smooth target distributions. (This was not a requirement of the Gibbs or MH algorithm.)

Another drawback is that the user has to tune the values of several input parameters: the time step Δt, the number of integration steps L, and the mass matrix M (this is often chosen to simply be $M = m\mathbb{I}_d$). Tuning Δt, L and m is important for the performance of the algorithm. If $L\Delta t$ is too small, then the algorithm takes small steps and requires many iterations to explore the space. If L is too large, then time is wasted on unnecessary integration at each iteration as the particle oscillates around the potential well. If Δt is too large, then the numerical integration becomes inaccurate, the Hamiltonian is not conserved, and the acceptance probability decreases[4].

As with Gibbs and MH, there are many variants of the HMC. One particularly important variant is NUTS which helps to automate the choice for L.

0.0.0.1 The No U-Turn Sampler (NUTS)

One variant of HMC that is worth mentioning because it is very widely used is the *No U-turn*, or NUTS, sampler[5]. The main appeal of NUTS is that it provides an automated

[4] Potentially even worse, a particularly unlucky choice of Δt and L can lead to the chain moving through a periodic sequence of points in the sample space. In this case the resulting Markov chain will fail to be irreducible. However, this is unlikely to happen in practice.

[5] Hoffman & Gelman (2014) "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo", Journal of Machine Learning Research, **15** 1593-1623, link

way to optimally choose the parameter L in HMC. The essential idea is to evolve the Hamiltonian dynamics both forwards and backwards in time until a U-Turn occurs; this allows us to go as far as possible in parameter space without wasteful oscillations. A point is then chosen randomly from evolution trajectory for the next MCMC sample.