Computational statistics and data analysis

Exercises

Alessio Spurio Mancini & Björn Malte Schäfer

20/06/2016

Hamiltonian Monte Carlo

The main idea behind Hamiltonian/Hybrid Monte Carlo is to develop a Hamiltonian function $H(\mathbf{x}, \mathbf{p})$ such that the resulting Hamiltonian dynamics allow us to efficiently explore some target distribution $p(\mathbf{x})$.

We want to relate $H(\mathbf{x}, \mathbf{p})$ to $p(\mathbf{x})$ using canonical distribution. For any energy function $E(\theta)$ over a set of variables θ we can define the corresponding canonical distribution as: $p(\theta) = \frac{1}{Z} e^{-E(\theta)}$. The variable Z is a normalizing constant called the partition function that scales the canonical distribution such that is sums to one, creating a valid probability distribution.

The energy function for Hamiltonian dynamics is a combination of potential and kinetic energies: $E(\theta) = H(\mathbf{x}, \mathbf{p}) = U(\mathbf{x}) + K(\mathbf{p})$.

The canonical distribution for the Hamiltonian dynamics energy function is

$$p(\mathbf{x}, \mathbf{p}) \propto e^{-H(\mathbf{x},\mathbf{p})} = e^{-[U(\mathbf{x})-K(\mathbf{p})]} = e^{-U(\mathbf{x})} e^{-K(\mathbf{p})}$$
$$\propto p(\mathbf{x}) p(\mathbf{p})$$

For $\mathbf{p}$, a common choice is to use a zero-mean Normal distribution with unit variance: $p(\mathbf{p}) \propto \frac{\mathbf{p}^\mathbf{T}\mathbf{p}}{2}$. Note that this is equivalent to having a quadratic potential energy term in the Hamiltonian: $K(\mathbf{p}) = \frac{\mathbf{p}^\mathbf{T}\mathbf{p}}{2}$

Another way of analysing the problem is to define the potential energy function as $U(\mathbf{x}) = -\log p(\mathbf{x})$. If we can calculate $-\frac{\partial \log(p(\mathbf{x}))}{\partial x_i}$, then we can simulate Hamiltonian dynamics that can be used in an MCMC technique.

In HMC we use Hamiltonian dynamics as a proposal function for a Markov Chain in order to explore the target (canonical) density $p(\mathbf{x})$ defined by $U(\mathbf{x})$ more efficiently than using a proposal probability distribution.

Starting at an initial state $[\mathbf{x}_0, \mathbf{p}_0]$, we simulate Hamiltonian dynamics for a short time using the Leap Frog method. We then use the state of the position and momentum variables at the end of the simulation as our proposed states variables $\mathbf{x}^*$ and $\mathbf{p}^*$. The proposed state is accepted using an update rule analogous to the Metropolis acceptance criterion.

Specifically if the probability of the proposed state after Hamiltonian dynamics

$$p(\mathbf{x}^*, \mathbf{p}^*) \propto e^{-[U(\mathbf{x}^*)+K\mathbf{p}^*]}$$

is greater than the probability of the state prior to the Hamiltonian dynamics

$$p(\mathbf{x}_0, \mathbf{p}_0) \propto e^{-[U(\mathbf{x}^{(t-1)}),K(\mathbf{p}^{(t-1)})]}$$

then the proposed state is accepted, otherwise, the proposed state is accepted randomly. If the state is rejected, the next state of the Markov chain is set as the state at $(t-1)$.
We must randomly perturb the dynamics so as to explore all of $p(\mathbf{x})$. This is done by simply drawing a random momentum from the corresponding canonical distribution $p(\mathbf{p})$ before running the dynamics prior to each sampling iteration t.

Leap Frog Method

1. Take a half step in time to update the momentum variable:
   $p_i(t + \delta/2) = p_i(t) - (\delta/2)\frac{\partial U}{\partial x_i(t)}$
2. Take a full step in time to update the position variable
   $x_i(t + \delta) = x_i(t) + \delta\frac{\partial K}{\partial p_i(t+\delta/2)}$
3. Take the remaining half step in time to finish updating the momentum variable
   $p_i(t + \delta) = p_i(t + \delta/2) - (\delta/2)\frac{\partial U}{\partial x_i(t+\delta)}$

- set $t = 0$
- generate an initial position state $\mathbf{x}^{(0)}, \mathbf{p}^{(0)}$
- repeat until $t = M$:
  set $t = t + 1$
  - sample a new initial momentum variable from the momentum canonical distribution $\mathbf{p}_0 \sim p(\mathbf{p})$
  - set $\mathbf{x}_0 = \mathbf{x}^{(t-1)}$
  - run Leap Frog algorithm starting at $[\mathbf{x}_0, \mathbf{p}_0]$ for L steps and stepsize δ to obtain proposed states $\mathbf{x}^*$ and $\mathbf{p}^*$
  - calculate the Metropolis acceptance probability:

$$\alpha = \min(1, \exp(-U(\mathbf{x}^*) + U(\mathbf{x}_0) - K(\mathbf{p}^*) + K(\mathbf{p}_0)))$$

  - draw a random number u from Unif$(0, 1)$
    if $u \leq \alpha$ accept the proposed state position $\mathbf{x}^*$ and set the next state in the Markov chain $\mathbf{x}^{(t)} = \mathbf{x}^*$
    else set $\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)}$.

Salpeter (1955) found that for stars of masses $\geq 1 M_{\odot}$ the mass distribution is given by a power law:

$$\frac{dN}{dM} \propto (M)^{-\alpha} \quad or \quad \frac{dN}{d\log M} = \frac{dN}{dM}\frac{dM}{d\log M} = \frac{dN}{dM} M \propto (M)^{1-\alpha}$$

He found $\alpha \approx 2.35$.

Our goal is to learn the probability distribution of α, given some data (the posterior probability). In fact, we are happy to only infer the likelihood function here.

We assume that we are given N i.i.d. samples of stellar masses (with negligible errors on the measurements). In that case, the likelihood of the problem is

$$\mathcal{L}(\{M_1, M_2, \ldots, M_N\}; \alpha) = \prod_{n=1}^{N} p(M_n|\alpha) = \prod_{n=1}^{N} c\,(M_n)^{-\alpha}$$

We can evaluate the normalisation constant by integrating over the observation interval $[M_{min}, M_{max}]$:

$$\int_{M_{min}}^{M_{max}} dM\, c\, M^{-\alpha} = c\frac{M_{max}^{1-\alpha} - M_{min}^{1-\alpha}}{1-\alpha} = 1$$

Instead of $\mathcal{L}$, once usually considers $\log \mathcal{L}$, which is numerically more stable and usually also simplifies the math,

$$\log \mathcal{L}(\{M_1, M_2, \ldots, M_N\}; \alpha) = N \log c - \alpha \sum_{n=1}^{N} \log (M_n)$$

$$= N \log \left(\frac{1-\alpha}{M_{max}^{1-\alpha} - M_{min}^{1-\alpha}} \right) - \alpha \sum_{n=1}^{N} \log (M_n)$$

Let us work out $\log \mathcal{L}$ using a Hamiltonian Monte-Carlo algorithm. First, we need to compute the gradient of our objective function, i.e., of the log-likelihood:

$$\frac{\partial \log \mathcal{L}}{\partial \alpha} = \tag{1}$$

$$- D - \frac{N}{1-\alpha}\left[1 + \frac{1-\alpha}{M_{max}^{1-\alpha} - M_{min}^{1-\alpha}}\left(M_{min}^{1-\alpha}\log M_{min} - M_{max}^{1-\alpha}\log M_{max}\right)\right] \tag{2}$$

where

$$D = \sum_{n=1}^{N} \log\left(M_n\right). \tag{3}$$