663 Final Project Outline

Jake Coleman and Sayan Patra

1 Abstract

For our project we will use Neal's 2011 paper, "MCMC using Hamiltonian dynamics." The paper discusses how to use Hamiltonian dynamics as a sampling scheme to explore target spaces better than traditional Metropolis-Hastings algorithms. The Hamiltonian is the sum of potential energy (based on position) and kinetic energy (based on momentum) - Hamilton's equations relate the two partial derivatives of the Hamiltonian to each other, and define a mapping from the state at time t to the state at time $t + s$. In Hamiltonian Monte Carlo (HMC), we draw auxiliary momentum variables from a Gaussian distribution, and use Hamiltonian dynamics simulations to update the position variable (which follows the distribution of interest). At the end of a user-defined number of steps of simulation, the new variables are accepted or rejected in a Metropolis-Hastings step.

In this report we will explore basic HMC with the "Leapfrog" discretization method, and follow some examples (such as highly-correlated multivariate Gaussian distributions) comparing HMC to random-walk Metropolis Hastings that show improvement for HMC. We will establish the superiority of the HMC method over regular random walk sampling schemes in the case of higher dimensions. We will also implement an extension of HMC proposed by Neal (1994) that uses "windows" of states to allow for a high probability of acceptance for all trajectories. Finally, we plan on converting the code to Cython or JIT to speed up implementation, and compare to existing HMC packages.

2 Introduction

The classic paper of Metropolis et al.(1953) introduced the idea of simulating the distribution of states for a system of idealized molecules through their likelihood function, known popularly as Markov Chain Monte Carlo (MCMC). Shortly after, Alder and Wainwright (1959) published their paper where they proposed a deterministic approach to molecular simulation following Newton's laws of motion, which have elegant formalization as *Hamiltonian dynamics*. The methods are asymptotically equivalent, as even in deterministic simulation each local region of the material experiences effectively random influences from distant regions, yet the methods have continued to co-exist. In 1987 Duane, Kennedy, Pendleton and Roweth united the MCMC and molecular dynamics approaches. The methods subsequently called "Hybrid Monte Carlo" or "Hamiltonian Monte Carlo" (HMC) was primarily applied to lattice field theory, but statistical applications began with Neals (1996) use of it neural network models.

In the subsequent sections we describe the hamiltonian dynamics (section 3) and how to construct a MCMC method out of it (section 4). One starts by defining a Hamiltonian function in terms of the probability distribution on the variables (the "position" variables) we want to sample from. Additionally auxiliary "momentum" variables are introduced, which typically have independent gaussian distributions. The HMC algorithm alternately updates the variables in simplistic fashion, with occasional metropolis updates to propose new states. This method has high probability of acceptance, therefore the exploration of the state space is quite fast. Like in other Markov Chain methods HMC has its issues of tuning, which are discussed briefly in section 4. In section 5 a variation of HMC is presented, where the acceptance rate of HMC is shown to be increased by looking at the "windows" of states at the beginning and the end of the trajectories.

3 Hamiltonian Dynamics

In MCMC the aim is to simulate a system parametrized by a vector q, called the "position" variable. In the dynamical simulation method, we introduce a momentum vector p which has the same dimension as q and a Hamiltonian function $H(q, p)$ which depicts the energy of the system. In statistical applications, the position will correspond to the variables of interest. The potential energy is thus the negative of the log of the probability density for those variables. Momentum variables are introduced artificially.

3.1 Hamilton's Equations

Hamiltonian dynamics results from a certain set of differential equations. The d-dimensional vector p and q describe the full state space of $2d$ dimensions. The partial derivatives of the Hamiltonian $H(q, p)$ determine how the vectors change over time,

as follows.

$$\begin{aligned}\frac{dq_i}{dt} &= \frac{\partial H}{\partial p_i} \\ \frac{dP_i}{dt} &= -\frac{\partial H}{\partial q_i}\end{aligned} \tag{3.1}$$

for $i = 1, ..., d$. For any interval of duration s, these equations define a mapping T_s from the state space at any time t to the state at time $t + s$.

3.2 Potential and Kinetic energy

The energy of the dynamics at point q, p of the state space is denoted by $H(q, p)$ and is generally considered the sum of energy generated independently by position and the momentum vector i.e.

$$H(q, p) = U(q) + K(p) \tag{3.2}$$

Here, $U(q)$ and $K(p)$ are called the potential and kinetic energy respectively. For our purpose the potential energy will be negative of the log of the distribution we want to sample from. The kinetic energy for ease of sampling is usually defined as

$$K(p) = p^T M^{-1} p/2 \tag{3.3}$$

where M is a symmetric, positive definite matrix. M is typically diagonal, which results in p being sampled from independent gaussians with the corresponding variances.
With these forms for H and K, Hamilton's equation 3.1 reduces to

$$\begin{aligned}\frac{dq_i}{dt} &= [M^{-1}p]_i \\ \frac{dP_i}{dt} &= -\frac{\partial U}{\partial q_i}\end{aligned} \tag{3.4}$$

3.3 Properties of Hamiltonian dynamics

Though the Hamiltonian dynamics in very intuitive for a molecular simulation, it needs to meet several crucial properties before we can construct a Markov chain algorithm from it. Luckily, it does. We will briefly mention the properties and its usefulness in this section. A detailed description can be found at [1].

Reversibility. Hamiltonian dynamics in reversible, i.e. the mapping T_s from the state at time t, $(q(t), p(t))$ to the state at time $t+s$, $(q(t+s), p(t+s))$, is bijective, and hence has an inverse T_{-s}. This inverse can be obtained by simply negating the time derivatives in equation 3.1. The reversibility is important for showing that MCMC updates that use the dynamics leave the desired distribution invariant, since this is most easily shown by using the reversibility of the Markov chain transitions, which requires reversability of the dynamics used to propose a state.

Conservation of the Hamiltonian. The hamiltonian dynamics keeps it invariant.

$$\frac{dH}{dt} = \sum_{i=1}^{d} \left[\frac{dq_i}{dt}\frac{\partial H}{dq_i} + \frac{dp_i}{dt}\frac{\partial H}{dp_i} \right] = 0 \tag{3.5}$$

This is important as in the metropolis update step of Hamiltonian dynamics the acceptance probability is one if the system is invariant.

Volume preservation. Another important property of Hamiltonian dynamics is that it preserves volume in (q, p) space. The significance of this is that we do not need to compute determinant of the Jacobian matrix for the mapping to account for change in the volume that might have been introduce by the Metropolis updates.

3.4 Discretizing Hamiltons Equation

In practice, the dynamics is simulated with some small finite sample size ϵ and hence must be discretized. The discretizing methods is applicable for any form of $H(q, p)$, however we will assume the form in 3.2 as it simplifies the expressions. Also M is assumed to be diagonal with diagonal elements $m_1, ...m_d$, so that the kinetic energy has the following form.

$$K(p) = \sum_{i=1}^{d} \frac{p_i^2}{2m_i} \tag{3.6}$$

Euler's method is standard method to approximate the solution to a system of differential equation. However for HMC the Leapfrog method produces better results. Thus we will skip the Eulers method and go straight to Leapfrog method.

3.4.1 Leapfrog Method

The *leapfrog* method works as follows,

$$\begin{aligned} p_i(t+\epsilon/2) &= p_i(t) - (\epsilon/2)\frac{\partial U}{\partial q_i}(q(t)) \\ q_i(t+\epsilon) &= q_i(t) + \epsilon\frac{p_i(t+\epsilon/2)}{m_i} \\ p_i(t+\epsilon) &= p_i(t+\epsilon/2) - (\epsilon/2)\frac{\partial U}{\partial q_i}(q(t+\epsilon)) \end{aligned} \tag{3.7}$$

The method starts with a half step for the momentum variables, then do a full step for the position variables using the new values of the momentum variables followed by another half step of the momentum variables where we use the new values of position variables.
Because 3.7 is mere transformations, the leapfrog method preserves volume exactly. One achieves reversability by simply negating p due to its symmetry.

3.5 Local and Global error of discretization methods

It is important to understand how the error from the discretizing the dynamics behave in the limit as the stepsize ϵ goes to 0, a detailed discussion is provided in [2]. Any useful method has the error going to zeros as ϵ goes to zero, so that ny upper limit on the error will apply i.e. if the error for (q, p) is no more than order ϵ^2, the error for $H(q, p)$ will also be no more than order ϵ^2.

The error commited after one step, that moves from time t to $t + \epsilon$ is called *local error*. The *global error* is ther error after the algorithm has been used to sumulate for some fixed interval, s, which will require s/ϵ steps. Consequently if the local error is order ϵ^p, the global error is of the order ϵ^{p-1}. Leapfrog method has a local error of order ϵ^3 and a global error of order ϵ^2, which means the error converges to zero faster than Euler method as ϵ decreases to zero.

4 Hamiltonian Monte Carlo

There are two important steps to sample from a distribution using Hamiltonian dynamics. First one needs to translate the density of the required distribution to a potential energy function and second one has to introduce "momentum" variables which go with the original variable of interest.

4.1 Probability and the Hamiltonian

The distribution to be sampled from can be related to a potential energy function via the concept of *canonical distribution* from statistical mechanics, which is also known as the *Boltzman distribution*. Let x be a state from the state space, then the canonical distribution over states has the probability density function

$$P(x) = \frac{1}{Z}exp(-E(x)/T) \tag{4.1}$$

where $E(x)$ is the energy of the system at state x, T is the temperature of the system and Z is the normalizing constant.
As Hamiltonian is an energy function it induces a joint distribution over the position and the momentum as follows:

$$P(p, q) = \frac{1}{Z}exp(-H(p, q)/T) \tag{4.2}$$

Under the assumption of particular form of Hamiltonian energy as in 3.2 the joint density is

$$P(p, q) = \frac{1}{Z}exp(-U(q)/T)exp(-K(p)/T) \tag{4.3}$$

It is clear from 4.3 that q and p are independent and each have canonican distribution with their own energy function. Thus the sample is generated by simulating an ergodic Markov chain that has the canonical distribution for (q, p) as its stationary distribution. In bayesian statistics, the posterior distribution is generally the focus of interest and hence will assume the postion of q. We set the temperature equal to 1, unless we are using some tempering methods and define the potential energy to be

$$U(q) = -log\big[\pi(q)L(q|D)\big] \tag{4.4}$$

where $\pi(q)$ is the prior density and $L(q|D)$ is the likelihood function given data D.

4.2 The Algorithm

Each iteration of the HMC algorithm has two steps. Only the momentum is changed in the first step and both the position and the momentum is changed at the second step. This explores the target density $P(q)$ defined by $U(q)$ more efficiently than using a proposal probability distribution. Starting at an initial state (q_0, p_0), we simulate Hamiltonian dynamics for a short time using the leapfrog method. We then use the state of the position and momentum variables at the end of the simulation as our proposed states variables q^* and 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 $P(q^*, p^*)$ is greater than probability of the state prior to the Hamiltonian dynamics $P(q_0, p_0)$ 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 $t-1$. For a given set of initial conditions, Hamiltonian dynamics will follow contours of constant energy in phase space. Therefore we must randomly perturb the dynamics so as to explore all of $P(q)$. This is done by simply drawing a random momentum from the corresponding canonical distribution $P(p)$ before running the dynamics prior to each sampling iteration t. Combining these steps, sampling random momentum, followed by Hamiltonian dynamics and Metropolis acceptance criterion defines the HMC algorithm for drawing M samples from a target distribution:

1. set $t = 0$
2. generate an initial position state $q_0 \sim U(q)$
3. repeat until $t = M$
   (a) set $t = t + 1$
   (b) sample an initial momentum state $p_0 \sim K(p)$
   (c) set $x_0 = x^{(t-1)}$
   (d) run leapfrog starting at (p_0, q_0) for L steps and stepsize ϵ to obtain (q^*, p^*)
   (e) calculate acceptance probability $\alpha = min(1, exp(U(p_0) + K(q_0) - U(q^*) - K(p^*)))$
   (f) draw $u \sim U(0, 1)$
   (g) if $u \leq \alpha$, accept and set $x^{(t)} = x^*$, else set $x^{(t)} = x^{(t-1)}$

HMC satisfies the "detailed balance" criterion and hence leaves the canonical distribution invariant. The algorithm is also "ergodic" and therefore explores the entire sample space.

5 Illustration of HMC and its benefits

In this section we will use several examples to demonstrate the potential of the HMC algorithm described in previous section. We will compare these to random-walk Metropolis mostly.

5.1 Trajectories for a two-dimensional problem

In this example we are sampling from a two dimensional gaussian distribution with zero mean, standard deviation one and correlation 0.95. The momentum variables are generated from an independent standard two dimensional bivariate normal. Therefore the Hamitonian has the form

$$H(q,p) = q^T \Sigma^{-1} q/2 + p^T p/2, \quad \text{with} \quad \Sigma = \begin{bmatrix} 1 & 0.95 \\ 0.95 & 1 \end{bmatrix} \tag{5.1}$$

Figure(1) shows trajectories for the position coordinates and the momentum coordinates respectively along with the hamiltonian value for each of the $L = 25$ leapfrog steps with a stepsize of $\epsilon = 0.25$ that is used to generate the plots.

The trajectory for HMC is widely different from that of a random-walk. Instead, starting from lower-left corner the position variables systematically move upwards and to the right, until they reach the upper-right corner, at which point the direction of the motion is reversed. The projection of the position in the diagonal direction changes slowly since the gradient in that direction is small, thus the direction of the diagonal motion stays same for many leapfrog steps. Alongside the large-scale diagonal motion, smaller-scale oscillations occur, moving back and forth across the contour created by the gaussian distribution. A similar behaviour is shown by the momentum coordinates.

The stepsize is mostly controlled by the need to keep the smaller stepsize under control. If larger stepsize were used the oscillation would be larger in the hamiltonian values. At a critical stepsize ($\epsilon = 0.45$), the trajectory becomes unstable and the value of Hamiltonian grows without bound (figure 2). As long as the stepsize is less than that, the error in the hamiltonian stays bounded regardless of the number of leapfrog steps.

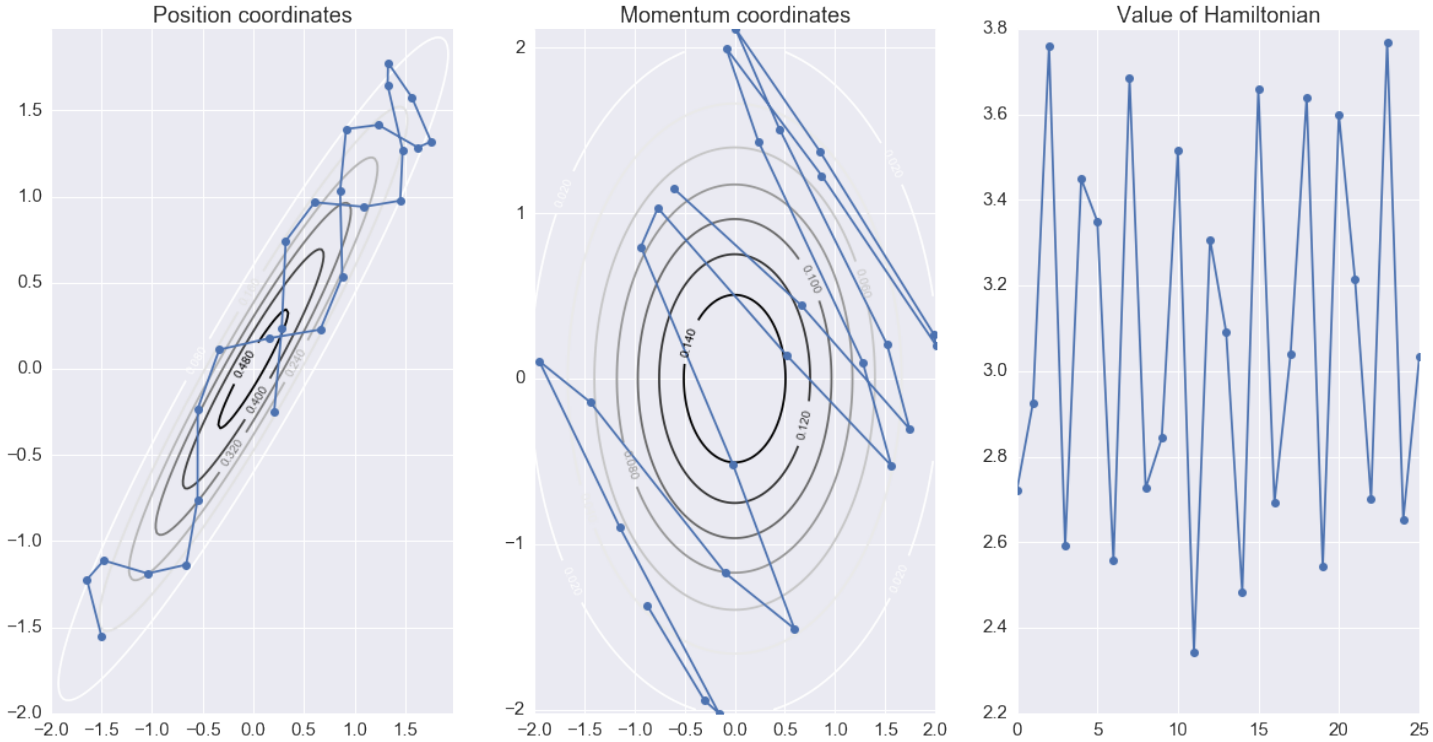

Figure 1: A trajectory for a 2D Gaussian distribution, simulated using 25 leapfrog steps with a stepsize of 0.25. The initial state for position variable q is $[-1.5, -1.55]^T$.

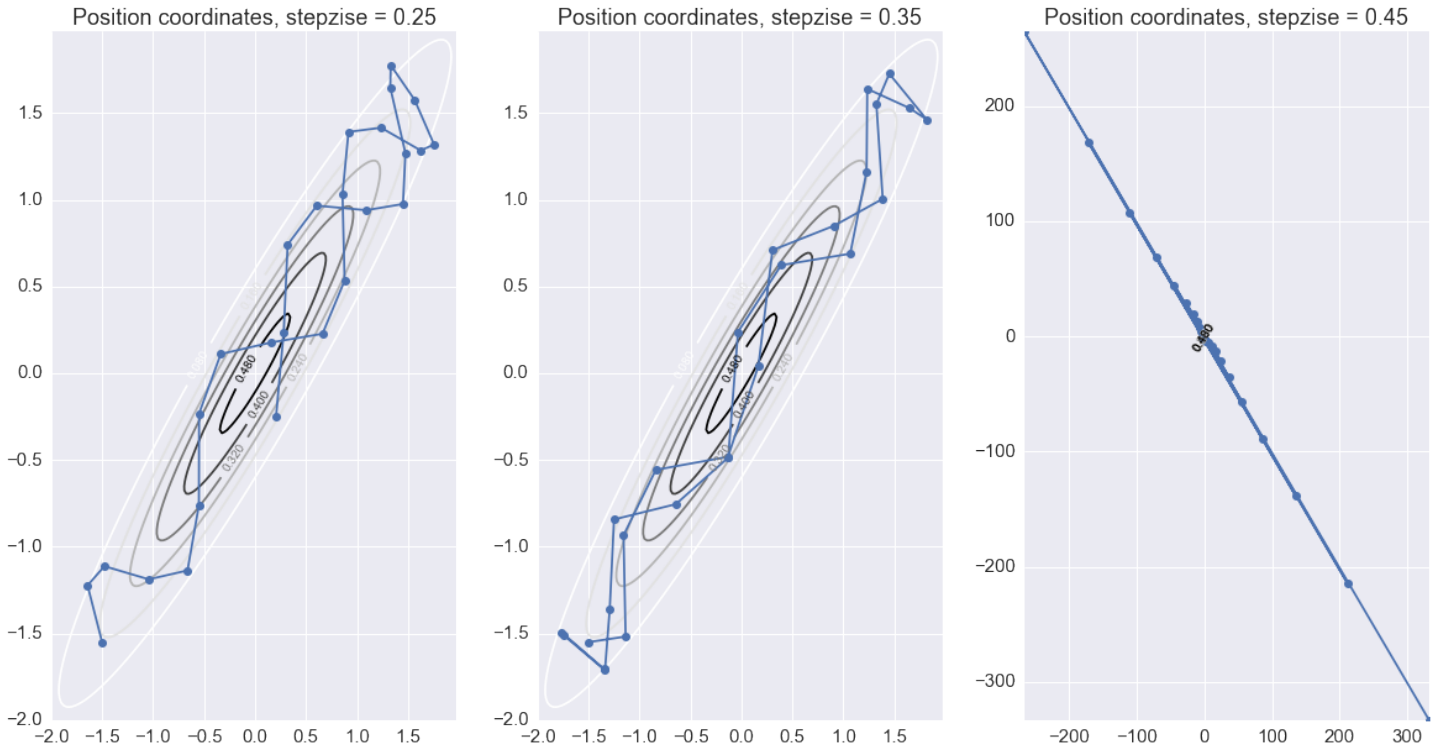

Figure 2: Illustration of Hamiltonian trajectories with increasing stepsize until it reaches critical state.

5.2 Sampling from a two-dimensional distribution

In this subsection we compare the trajectories between HMC and simple random-walk Metropolis method. The aim is to simulate from a bivariate Gaussian similar to the previous one but with stronger correlation of 0.98. The HMC used the same kinetic energy as before. The results of 20 HMC iteration with $L = 20$ leapfrog steps with stepsize $\epsilon = 0.18$ are shown in the right of figure(3). These values were chosen so that the trajectory length, ϵL, is sufficient to move to a distant point in the distribution without being so large that the trajectory will often back on itself. The rejection rate for this trajectories was 0.1.

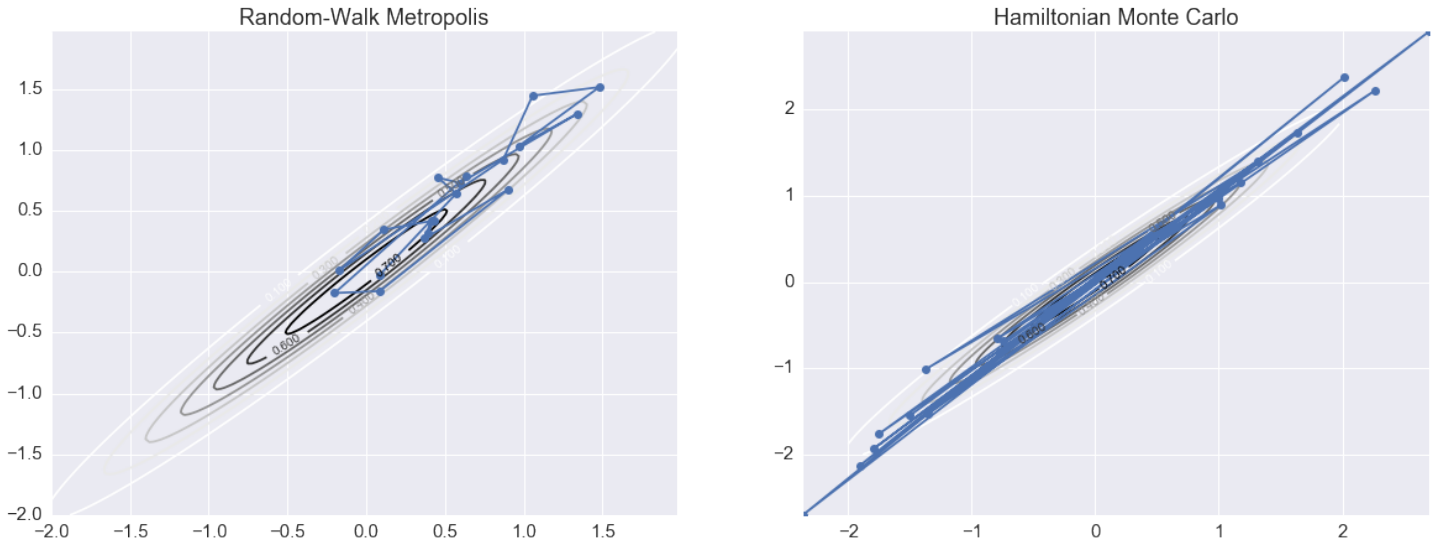

Figure 3: Twenty iteration of the random-walk Metropolis method (with 20 updates per iteration) and of the Hamiltonian Monte Carlo method (with 20 leapfrog steps per trajectory) for the two position coordinates.

Left image of the figure(3) shows every 20th state from multiple iteration of random-walk Metropolis, with a bivaraite Gaussian proposal distribution with the current state as mean, zero correlation and the same standard deviation for the two coordinates. The values used is 0.18, the same as the ϵ for HMC. so that the change in the state space is comparable. The rejection rate for the random-walk proposals is 0.38. Figure(4) illustrates the result for 200 HMC iterations and 20×200 random-walk Metropolis iterations. It is evident the autocorrelation between samples is way less in HMC and hence the state space is traversed more uniformly.

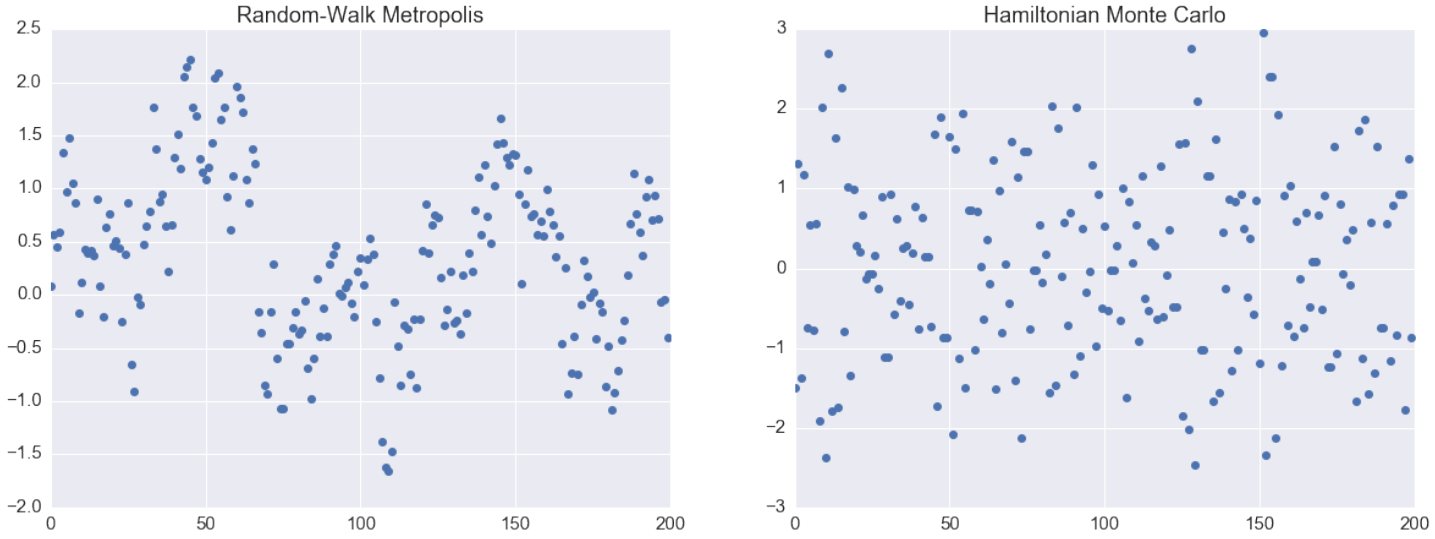

Figure 4: Two hundred iteration of the random-walk Metropolis method (with 20 updates per iteration) and of the Hamiltonian Monte Carlo method (with 20 leapfrog steps per trajectory) with only the first position coordinate plotted.

5.3 Efficiency in High dimension

In this section we show that HMC scales better than MH as a sampling algorithm. The target distribution to sample from is a 100 dimensional Gaussian distribution, where the variables are independent, with means of zero, and standard deviations of $0.01, 0.02, ..., 0.99, 1.00$. We will use the same Kinetic energy function for the HMC, however it is 100 dimensional instead of 2.

For this problem, the position coordinates, q_i and corresponding momentum coordinates p_i are all independent, so the leapfrogs steps operate independently. However the acceptance of a trajectory depends on the total error in the Hamiltonian due to the leapfrog discretization, which is a sum of the errors due to each (q_i, p_i) pair. To keep the error small one needs to limit the leapfrog size roughly equal to the smallest of the standard deviations (0.01), therefore many leapfrog steps are needed to move a distance comparable to the largest of the standard deviations (1.00). Keeping this points in mind the HMC is applied with $\epsilon = 0.13$ and $L = 150$. These are close to the optimal settings, the rejection rate was 0.15 for HMC and 0.75 for MH.

References

[1] Neal, R., M.: MCMC using Hamiltonian dynamics arXiv:1206.1901v1 5(2) (Jun 2012)

[2] Leimkuhler, B., Reikh, S.: Simulating Hamiltonian dynamics Cambridge University Press (2004)