Examination of the likelihood function by simulation of a fictional Hamiltonian system

Joshua N. Pritikin

Department of Psychology
University of Virginia

16 October 2014

Likelihood

The likelihood of parameter vector θ given observed data x and some model is defined as

$$L(\theta) \equiv \Pr(x|\theta).$$

Recall Bayes' theorem,

$$\Pr(\theta|x) = \frac{\Pr(\theta)\Pr(x|\theta)}{\Pr(x)}.$$

Therefore, likelihood can be regarded as a posterior density with a uniform prior $\Pr(\theta)$,

$$\Pr(\theta|x) = \text{constant} \times L(\theta)$$

1. Likelihood is one of the main engines of statistical modeling.
2. We need to analyze the likelihood function regardless of whether we take a Bayesian or frequentist perspective.
3. Broadly, there are two strategies: Maximum likelihood point estimates (posterior mode) and the posterior distribution.
4. In this talk, I'm going to focus on the posterior density.

Likelihood

The likelihood of parameter vector θ given observed data x and some model is defined as

$$L(\theta) \equiv \Pr(x|\theta).$$

Recall Bayes' theorem,

$$\Pr(\theta|x) = \frac{\Pr(\theta)\Pr(x|\theta)}{\Pr(x)}.$$

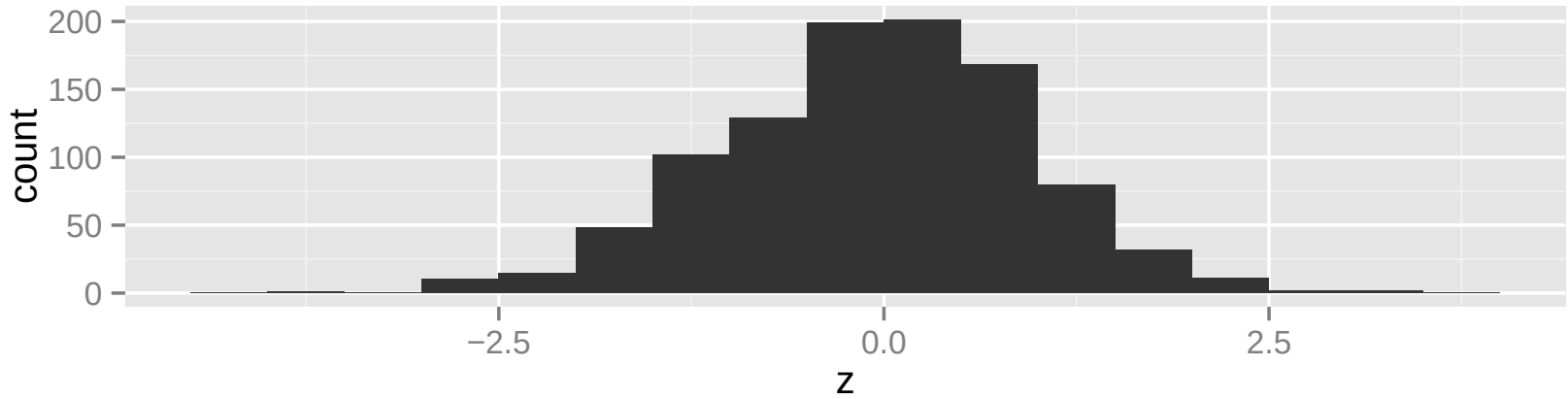
Therefore, likelihood can be regarded as a posterior density with a uniform prior $\Pr(\theta)$,

$$\Pr(\theta|x) = \text{constant} \times L(\theta)$$

Sampling, example

quantile	0.500	0.250	0.750	0.100	0.900	0.999
Z score	0.000	-0.674	0.674	-1.282	1.282	3.090

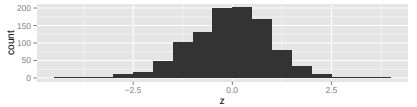
```
df <- data.frame(z=qnorm(runif(1000)))
ggplot(df, aes(z)) + geom_histogram(binwidth=.5)
```



Sampling, example

quantile	0.500	0.250	0.750	0.100	0.900	0.999
Z score	0.000	-0.674	0.674	-1.282	1.282	3.090

```
df <- data.frame(z=qnorm(runif(1000)))
ggplot(df, aes(z)) + geom_histogram(binwidth=.5)
```



1. If we could sample from the likelihood then we could discover summary statistics of the parameters, marginal distribution of each parameter, etc. It would be really useful.
2. Computers can generate uniformly distributed samples.
3. For instance, to sample from a univariate Gaussian, we need the inverse cumulative distribution function (a.k.a. the quantile function).
4. What if you don't have an inverse cumulative distribution function?

Sampling method

There are many strategies to sample from $L(\theta|x)$. The more popular algorithms include

- Metropolis-Hastings
- Gibbs
- Hamiltonian Monte Carlo

Sampling method

There are many strategies to sample from $L(\theta|x)$. The more popular algorithms include

- Metropolis-Hastings
- Gibbs
- Hamiltonian Monte Carlo

1. These methods are listed in the historical order of development: Metropolis 1953, Hastings 1970, Gibbs 1984, HMC (as a real simulation) 1987, HMC (for statistics) 1996.

Metropolis-Hastings

Choose an arbitrary point $y_{t=0}$ for the first sample and an arbitrary conditional distribution $Q(y_{t+1}|y_t)$.

Select the next candidate point $y' \sim Q(y_t)$ and compute the acceptance ratio,

$$\alpha = \frac{L(y')}{L(y_t)}.$$

If $\alpha \geq 1$ then set $y_{t+1} = y'$. Otherwise accept the candidate y' with probability α. If the candidate is rejected, set $y_{t+1} = y_t$.

Metropolis-Hastings

Choose an arbitrary point $y_{t=0}$ for the first sample and an arbitrary conditional distribution $Q(y_{t+1}|y_t)$.

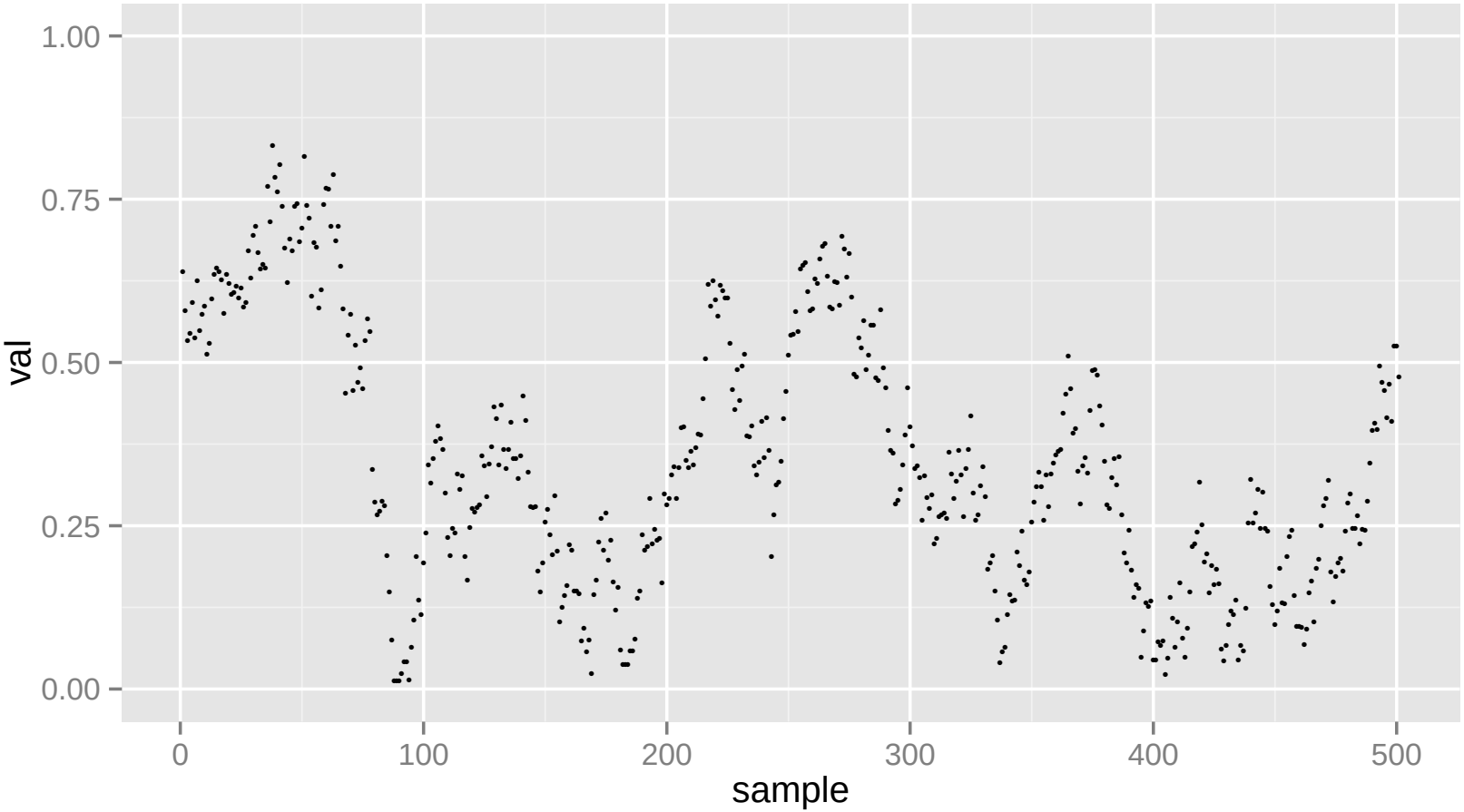
Select the next candidate point $y' \sim Q(y_t)$ and compute the acceptance ratio,

$$\alpha = \frac{L(y')}{L(y_t)}.$$

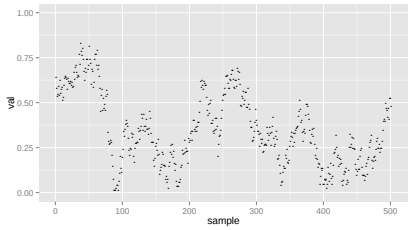
If $\alpha \geq 1$ then set $y_{t+1} = y'$. Otherwise accept the candidate y' with probability α. If the candidate is rejected, set $y_{t+1} = y_t$.

1. Typically Q is a Gaussian centered at the current point.
2. Q is often called the *proposal density* or *jumping distribution*.
3. Q must be symmetric. That is, we require $Q(y_{t+1}|y_t) = Q(y_t|y_{t+1})$.
4. Since L is a likelihood, the acceptance ratio is similar to the likelihood ratio test.
5. There is controversy over the credit for discovery of the algorithm. According to Rosenbluth, neither Metropolis nor Teller participated in any way (Gubernatis, 2005).
6. Awesome: Performance is not very sensitive to the dimensionality of the parameter space. In other words, M-H does not suffer from the curse of dimensionality.

M-H sampling of a 1-dimensional uniform

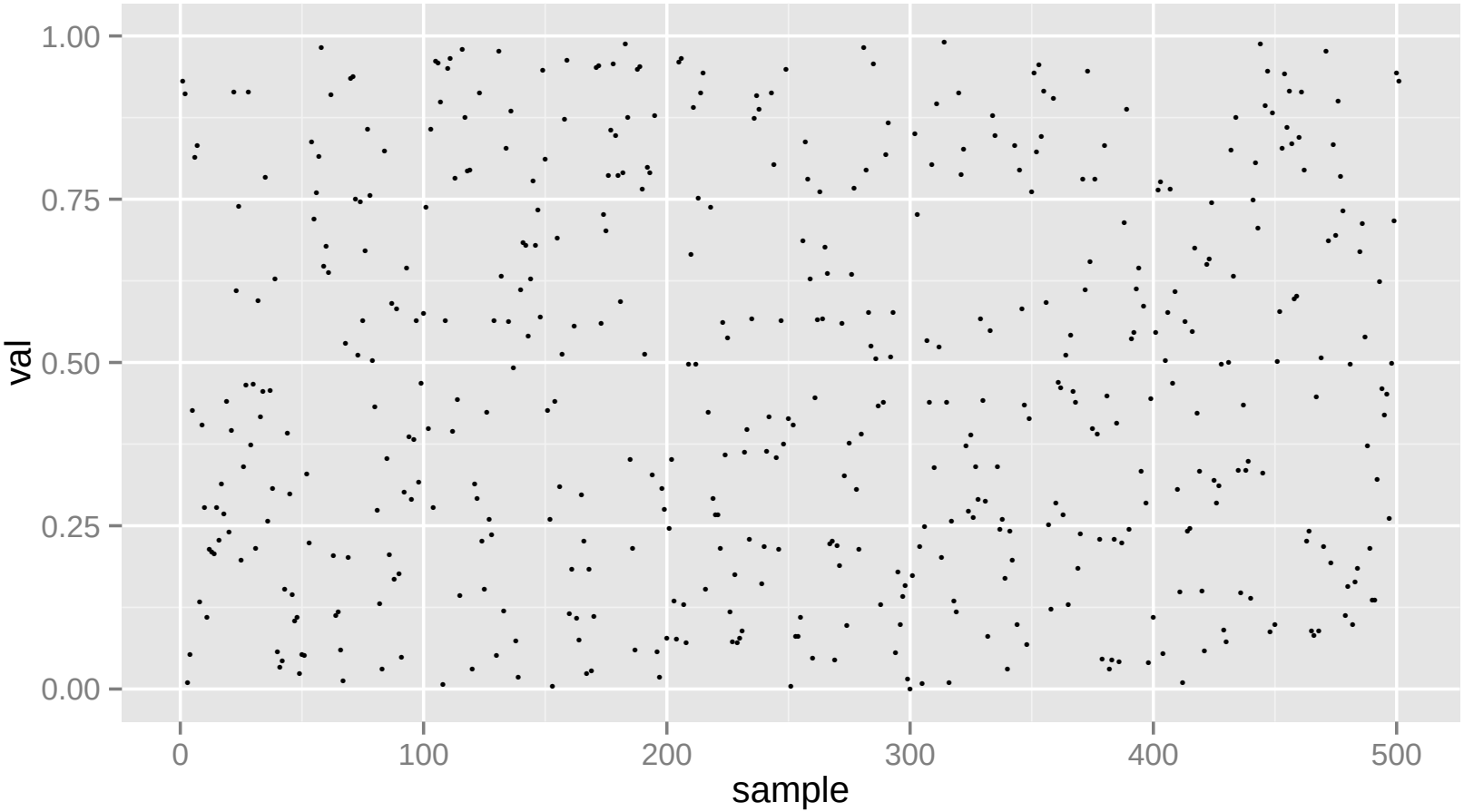


M-H sampling of a 1-dimensional uniform

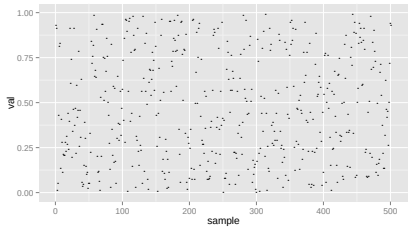


1. Suppose we use M-H to sample from the uniform distribution.
2. The acceptance ratio α is always 1. We always accept the next candidate (except at the boundaries).
3. Standard deviation of the Gaussian proposal density was set to 0.05.
4. Problem: Doesn't look very uniform. Too much autocorrelation. Potential fixes: thinning.

M-H sampling of a 1-dimensional uniform



M-H sampling of a 1-dimensional uniform



1. If we only take 1 sample out of 100, it looks a lot better.
2. Foreshadow next slide: Why does this work? What is the mathematical justification?

Markov Chain

A memoryless Markov Chain can be represented as a transition matrix,

	s_1	s_2	s_3
s_1	0.17	0.56	0.28
s_2	0.05	0.51	0.44
s_3	0.23	0.60	0.17

Markov Chain

A memoryless Markov Chain can be represented as a transition matrix,

	s_1	s_2	s_3
s_1	0.17	0.56	0.28
s_2	0.05	0.51	0.44
s_3	0.23	0.60	0.17

1. *Memoryless* means that the next state only depends on the previous state.
2. This is a *right* stochastic matrix so we read from rows to columns.
3. This is a discrete transition matrix. We're going to be working with transition "matrices" that are actually continuous joint densities.

Markov Chain

To construct a Markov Chain that samples from a unique stationary distribution π, two conditions are required:

1. Ergodicity
2. Detailed balance (a.k.a. reversibility) means that

$$\pi_i \Pr(i \to j) = \pi_j \Pr(j \to i)$$

$$\ldots \text{ or alternately that}$$

$$\frac{\pi_i}{\pi_j} = \frac{\Pr(j \to i)}{\Pr(i \to j)}$$

where π_i and π_j are equalibrium probabilities of being in states i and j, respectively.

Markov Chain

To construct a Markov Chain that samples from a unique stationary distribution π, two conditions are required:

1. Ergodicity
2. Detailed balance (a.k.a. reversibility) means that

$$\pi_i \Pr(i \to j) = \pi_j \Pr(j \to i)$$

... or alternately that

$$\frac{\pi_i}{\pi_j} = \frac{\Pr(j \to i)}{\Pr(i \to j)}$$

where π_i and π_j are equalibrium probabilities of being in states i and j, respectively.

1. Ergodicity has 2 conditions: Aperiodic (the system does not return to the same state at fixed intervals) and positive recurrent (the expected number of steps for returning to the same state is finite).
2. i, j are coordinates in the parameter space.
3. Detailed balance is a stronger condition than is necessary to ensure that the chain converges to a stationary distribution. In any case, detailed balance is not a difficult condition to meet.
4. For Metropolis-Hastings, ergodicity is satisfied because any state is reachable from any other state and detailed balance is satisfied by the proposal acceptance criteria.

Gibbs sampling

Without loss of generality, suppose we have a model with 3 parameters,

$$L(\beta_1, \beta_2, \beta_3 | x).$$

We would like to draw samples consisting of $Y \equiv (\beta_1, \beta_2, \beta_3)$,

$$Y \sim L(x).$$

If simple conditional distributions are available, we can draw samples parameter-wise,

$$\begin{aligned}
\beta_1 &\sim L(\beta_2, \beta_3, x) \\
\beta_2 &\sim L(\beta_3, \beta_1, x) \\
\beta_3 &\sim L(\beta_1, \beta_2, x)
\end{aligned}$$

Potentially more computationally efficient that Metropolis-Hastings.

Gibbs sampling

Without loss of generality, suppose we have a model with 3 parameters,

$$L(\beta_1, \beta_2, \beta_3 | x).$$

We would like to draw samples consisting of $Y \equiv (\beta_1, \beta_2, \beta_3)$,

$$Y \sim L(x).$$

If simple conditional distributions are available, we can draw samples parameter-wise,

$$\beta_1 \sim L(\beta_2, \beta_3, x)$$
$$\beta_2 \sim L(\beta_3, \beta_1, x)$$
$$\beta_3 \sim L(\beta_1, \beta_2, x)$$

Potentially more computationally efficient that Metropolis-Hastings.

1. Marginal distributions should have a relatively simple function form.
2. For full Bayesian models with informative priors, *conjugacy* is a special relationship between the functional forms of the prior and posterior that simplifies the conditional marginal distribution. A great deal of complex math is involved in deriving simple marginal distributions.
3. This works because the Gibbs sampler is ergodic and forms a reversible Markov Chain.
4. The Gibbs requirement of simple marginal distributions is constraining.

Hamiltonian Monte Carlo (HMC)

Let q and p be d-dimensional position and momentum vectors, respectively. The system $H(q,p)$ is known as a *Hamiltonian*. The laws of motion are,

$$\frac{\mathrm{d}q_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i}$$
$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial q_i}$$

for $i \in \{1 \dots d\}$.

$$U(q) = \text{is the potential energy}$$
$$K(p) = \frac{p^T M^{-1} p}{2}$$

Hamiltonian Monte Carlo (HMC)

Let q and p be d-dimensional position and momentum vectors, respectively. The system $H(q,p)$ is known as a *Hamiltonian*. The laws of motion are,

$$\frac{\mathrm{d}q_i}{\mathrm{d}t} = \frac{\partial H}{\partial p_i}$$
$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H}{\partial q_i}$$

for $i \in \{1 \dots d\}$.

$U(q) =$ is the potential energy

$K(p) = \frac{p^T M^{-1} p}{2}$

1. A Hamiltonian system is a reformulation of Newton's classical mechanics.
2. In two dimensions, we can visualize the dynamics as that of a frictionless puck that slides over a surface of varying height. The state of this system consists of the position of the puck (q) and the momentum of the puck (its mass times velocity; p). The mass of the puck is represented by M. The potential energy $U(q)$ is proportional to the height of the surface at the puck's current position and is given by the minus log likelihood of the target distribution. Kinetic energy is given by $K(p)$. This form of potential energy is proportional to minus the log probability of a zero-mean Gaussian distribution with covariance matrix M. The position represents a sample from the target distribution. The momentum and kinetic energy are completely fictional additions that will facilitate the construction of a more flexible Markov Chain. (Note: Mostly paraphrased from Neil, 2010).

HMC, nice properties

- Hamiltonian dynamics is reversible
- Energy is conserved

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \sum_{i=1}^{d} \left[\frac{\mathrm{d}q_i}{\mathrm{d}t} \frac{\partial H}{\partial q_i} + \frac{\mathrm{d}p_i}{\mathrm{d}t} \frac{\partial H}{\partial p_i} \right] = 0$$

- Volume is preserved (due to sympelecticness)

HMC, nice properties

- Hamiltonian dynamics is reversible
- Energy is conserved

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

- Volume is preserved (due to sympelecticness)

1. Reversibility is important for proving detailed balance. Reversibility can be accomplished by flipping the sign of momentum or time.
2. For our parameterization of the Hamiltonian, the value of the Hamiltonian is half the squared distance from the origin and solutions stay at a constant distance from the origin. This is important because, for Metropolis updates using a proposal found by Hamiltonian dynamics, the acceptance probability is one if H is kept invariant. In practice, we can only make H approximately invariant because of errors that creep into the discrete simulation.
3. If we simulate the system on points in some region R of (q, p) space with volume V, the image of R will also have volume V. (Two different proofs are given in Neil, 2010.) For the Markov Chain, this means that we don't have to account for change in volume of the acceptance probability for Metropolis updates. (Otherwise we would have to compute the determinant of the Jacobian matrix for the mapping, messy.)

HMC, discrete simulation

For simplicity, assume M is diagonal with elements $m_1, \ldots, m_d$.

- Euler's method

$$p_i(t+\epsilon) = p_i(t) - \epsilon \frac{\partial U}{\partial q_i}(q(t))$$
$$q_i(t+\epsilon) = q_i(t) + \epsilon \frac{p_i(t)}{m_i}$$

- Euler's method (improved)

$$p_i(t+\epsilon) = p_i(t) - \epsilon \frac{\partial U}{\partial q_i}(q(t))$$
$$q_i(t+\epsilon) = q_i(t) + \epsilon \frac{p_i(t+\epsilon)}{m_i}$$

HMC, discrete simulation

For simplicity, assume M is diagonal with elements $m_1, \ldots, m_d$.

- Euler's method

$$p_i(t+\epsilon) = p_i(t) - \epsilon \frac{\partial U}{\partial q_i}(q(t))$$
$$q_i(t+\epsilon) = q_i(t) + \epsilon \frac{p_i(t)}{m_i}$$

- Euler's method (improved)

$$p_i(t+\epsilon) = p_i(t) - \epsilon \frac{\partial U}{\partial q_i}(q(t))$$
$$q_i(t+\epsilon) = q_i(t) + \epsilon \frac{p_i(t+\epsilon)}{m_i}$$

1. Hamilton's equations must be approximated by discretizing time using some small step size ϵ. Starting with the state at time zero, we iteratively approximate the state at times ϵ, 2ϵ, 3ϵ, etc. (Paraphrased from Neil, 2010, p. 7.)
2. Remind that M is the mass matrix.
3. We change the momentum p by the gradient evaluated at position t and change the position q by the velocity at time t.
4. (Improved) We change the momentum p by the gradient evaluated at position t and change the position q by the velocity at time $t + \epsilon$.

HMC, discrete simulation

- The leapfrog method

$$p_i(t+\epsilon/2) = p_i(t) - (\epsilon/2)\frac{\partial U}{\partial q_i}(q(t))$$

$$q_i(t+\epsilon) = q_i(t) + \epsilon\frac{p_i(t+\epsilon/2)}{m_i}$$

$$p_i(t+\epsilon) = p_i(t+\epsilon/2) - (\epsilon/2)\frac{\partial U}{\partial q_i}(q(t+\epsilon))$$

HMC, discrete simulation

- The leapfrog method

$$p_i(t+\epsilon/2) = p_i(t) - (\epsilon/2)\frac{\partial U}{\partial q_i}(q(t))$$
$$q_i(t+\epsilon) = q_i(t) + \epsilon\frac{p_i(t+\epsilon/2)}{m_i}$$
$$p_i(t+\epsilon) = p_i(t+\epsilon/2) - (\epsilon/2)\frac{\partial U}{\partial q_i}(q(t+\epsilon))$$

1. Here we make a half step update of the momentum p, update the position q by a full step, and then update the momentum p by another half step. The leapfrog method preserves volume exactly since each step is a shear transformation. Due to symmetry, it is also easily reversible by simply negating p, applying the same procedure, and then negating p again. (Paraphrased from Neil, 2010, p. 10.)

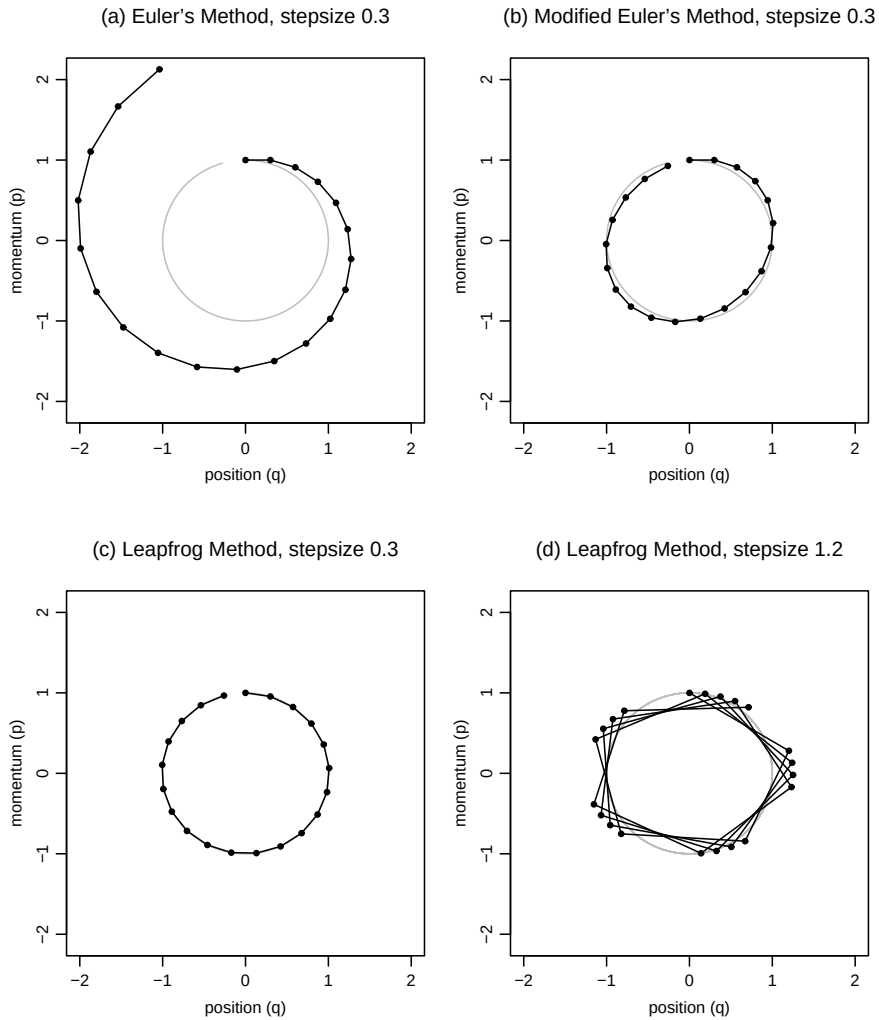


Figure 1: Results using three methods for approximating Hamiltonian dynamics, when $H(q,p) = q^2/2 + p^2/2$. The initial state was $q = 0$, $p = 1$. The stepsize was $\varepsilon = 0.3$ for (a), (b), and (c), and $\varepsilon = 1.2$ for (d). Twenty steps of the simulated trajectory are shown for each method, along with the true trajectory (in gray).

Construction of HMC

$$H(q,p) = \{U(q), K(P)\}$$

$$U(q) = -\log\left[\Pr(\theta)L(\theta)\right]$$

$$K(p) = \frac{p^T M^{-1} p}{2}$$

1. Draw new a momentum
2. Simulate Hamiltonian dynamics for L steps with step size ϵ. Uniformly sample a candidate from the newly simulated states. Accept the candidate with probability,

$$\min\left[1, \exp\left(-H(q',p') + H(q,p)\right)\right]$$

Construction of HMC

$$H(q,p) = \{U(q), K(P)\}$$

$$U(q) = -\log\left[\Pr(\theta)L(\theta)\right]$$

$$K(p) = \frac{p^T M^{-1} p}{2}$$

1. Draw new a momentum
2. Simulate Hamiltonian dynamics for L steps with step size ϵ. Uniformly sample a candidate from the newly simulated states. Accept the candidate with probability,

$$\min\left[1, \exp\left(-H(q',p') + H(q,p)\right)\right]$$

1. The potential function is the minus log likelihood, optionally including a prior density. Current practice is to use a quadratic kinetic energy which obtains a zero-mean multivariate Gaussian distribution. The distribution of the kinetic energy is the mass matrix M.
2. Step 1: The target distribution is invariant because q isn't changed and p is drawn from its correct conditional distribution (p is independent from q). (Compare with Gibbs sampling.)
3. I will discuss parameters L and ϵ shortly.
4. If the candidate state has better simulation quality than the prior state then we accept, otherwise we accept with the probability ratio.

Does HMC satisfy M-H conditions?

$$H(q,p) = \{U(q), K(P)\}$$

$$U(q) = -\log\left[\Pr(\theta)L(\theta)\right]$$

$$K(p) = \frac{p^T M^{-1} p}{2}$$

1. Detailed balance (a.k.a. reversibility)
2. Ergodicity

Does HMC satisfy M-H conditions?

$$H(q,p) = \{U(q), K(P)\}$$
$$U(q) = -\log\left[\Pr(\theta)L(\theta)\right]$$
$$K(p) = \frac{p^T M^{-1} p}{2}$$

1. Detailed balance (a.k.a. reversibility)
2. Ergodicity

1. If we regard the HMC as sampling from the joint distribution of q and p, the simulation leaves the probability density almost unchanged. Movement to a different probability density is mainly accomplished by drawing a new momentum p from the mass matrix. Simulation can move to a location with very different potential energy $U(q)$, but the total energy of the system $H(q,p)$ will remain nearly constant. A proof of detailed balance is given in Neil (2010, p. 13). The gist is that since both the sampling of momentum variables and the Metropolis update with a proposal found by Hamiltonian simulation leave the target distribution invariant, the HMC algorithm as a whole does as well.
2. In an HMC iteration, any value can be sampled for the momentum variables which will typically affect the position variables in arbitrary ways. HMC is unlikely to get stuck in some subset of the state space. However, with a suitable poor choice of leapfrog steps L and step size ϵ, it is conceivable that trajectories could return to the starting position.

Sounds cool, but does it work?

- 100-dimensional multivariate Gaussian distribution
- all dimensions independent
- standard deviations of 0.01, 0.02,. . . , 0.99, 1.00
- HMC L parameter set to 150
- For each HMC iteration, ϵ was drawn uniformly from $0.013 \pm 20\%$

Reject rate was 0.13 for HMC and 0.75 for random-walk Metropolis.

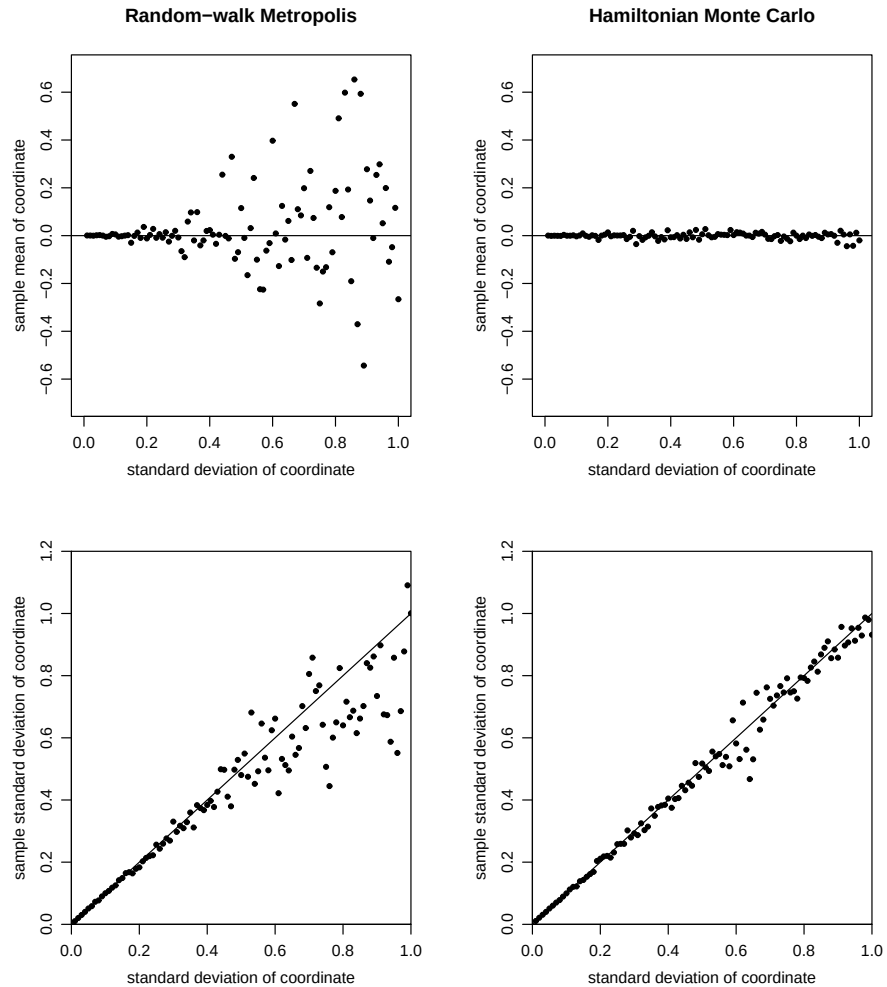


Figure 7: Estimates of means (top) and standard deviations (bottom) for the 100-dimensional example, using random-walk Metropolis (left) and HMC (right). The 100 variables are labelled on the horizontal axes by the true standard deviaton of that variable. Estimates are on the vertical axes.

No-U-Turn sampler

Problem: For HMC, we need to select the number of leapfrog steps L and step size ϵ.

Solution: NUTS (Hoffman & Gelman, 2011)

No-U-Turn sampler

Problem: For HMC, we need to select the number of leapfrog steps L and step size ϵ.

Solution: NUTS (Hoffman & Gelman, 2011)

1. Recall that the whole point of HMC was to minimize autocorrelation between successive samples. Trajectories that double back toward the original starting point are a waste of computation because we generate candidates that are less dispersed than they could be. NUTS traces out a path both forward and backward in fictitious time. It stops when latest point starts to double back on the earliest point. Most of the complexity in the algorithm is involved in ensuring detailed balance. This takes care of the number of leapfrog steps parameter L.
2. The ϵ parameter is automatically tuned during the burn-in phase. There is some evidence that the optimal Metropolis acceptance probability is 0.65. Since ϵ controls the simulation accuracy, we can tune ϵ for a target acceptance probability. The details are rather technical.

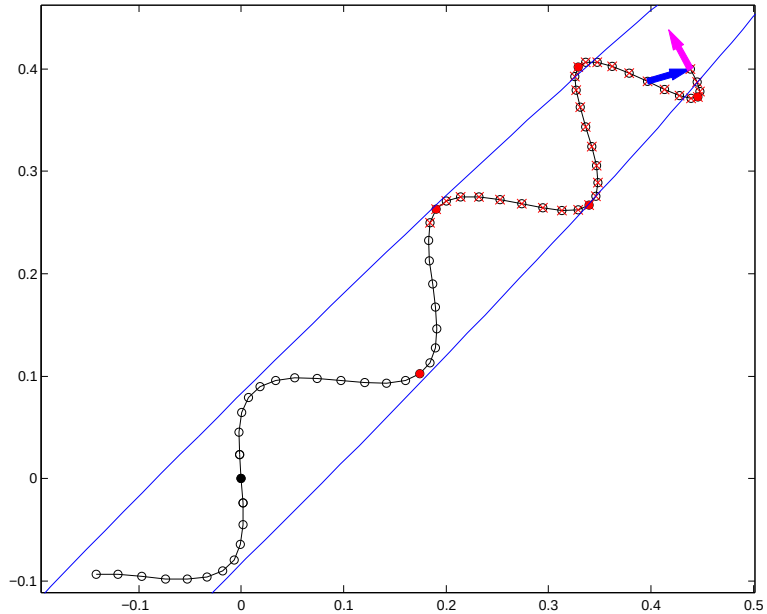


Figure 2: Example of a trajectory generated during one iteration of NUTS. The blue ellipse is a contour of the target distribution, the black open circles are the positions θ traced out by the leapfrog integrator and associated with elements of the set of visited states $\mathcal{B}$, the black solid circle is the starting position, the red solid circles are positions associated with states that must be excluded from the set $\mathcal{C}$ of possible next samples because their joint probability is below the slice variable u, and the positions with a red "x" through them correspond to states that must be excluded from $\mathcal{C}$ to satisfy detailed balance. The blue arrow is the vector from the positions associated with the leftmost to the rightmost leaf nodes in the rightmost height-3 subtree, and the magenta arrow is the (normalized) momentum vector at the final state in the trajectory. The doubling process stops here, since the blue and magenta arrows make an angle of more than 90 degrees. The crossed-out nodes with a red "x" are in the right half-tree, and must be ignored when choosing the next sample.

being more complicated, the analogous algorithm that eliminates the slice variable seems empirically to be slightly less efficient than the algorithm presented in this paper.

Dealing with curvature

NUTS works great for well behaved densities.

Consider the 2-dimensional *banana* density on manifold R,

$$V(R) = \frac{1}{2}\left[\frac{q_1^2(R)}{\sigma_1^2} + \frac{(q_2(R) + \beta q_1^2(R) - 100\beta)^2}{\sigma_2^2}\right]$$

with $\beta = 0.03$, $\sigma_1 = 0.01$, and $\sigma_2 = 1$ (Betancourt, 2013).

Dealing with curvature

NUTS works great for well behaved densities.

Consider the 2-dimensional *banana* density on manifold R,

$$V(R) = \frac{1}{2}\left[\frac{q_1^2(R)}{\sigma_1^2} + \frac{(q_2(R) + \beta q_1^2(R) - 100\beta)^2}{\sigma_2^2}\right]$$

with $\beta = 0.03$, $\sigma_1 = 0.01$, and $\sigma_2 = 1$ (Betancourt, 2013).

1. This density is not convex. NUTS terminates too early.
2. Betancourt's idea is to use an approximation of the information matrix as the mass matrix, and curve space according to the mass matrix. It's called a Riemannian manifold.

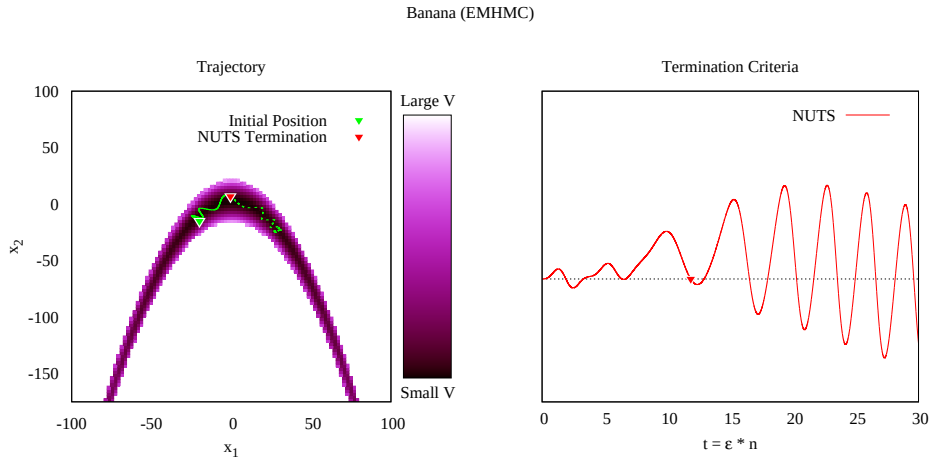


FIG. 4. On more complicated potentials, such as that arising from the banana distribution (4), the NUTS criterion terminates prematurely even when ignoring the transient behavior.

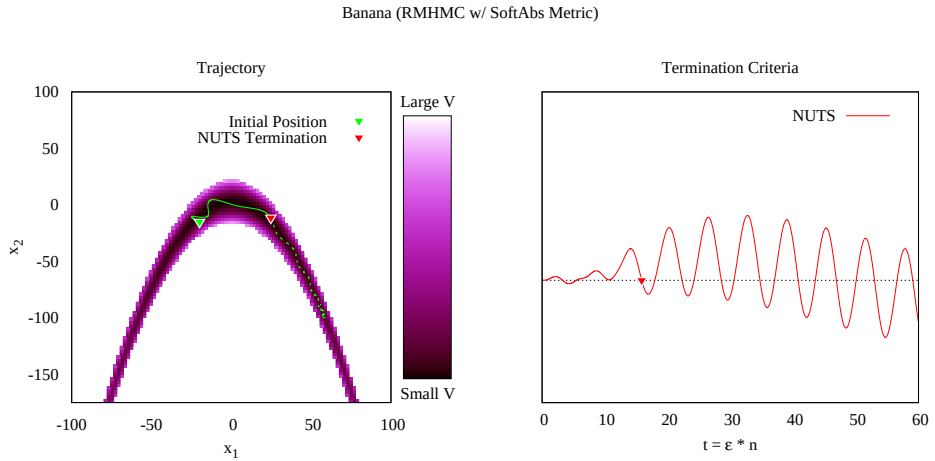


FIG. 5. RMHMC trajectories, here using the SoftAbs metric [8] with $\alpha = 1$, yield much smoother trajectories in the banana potential (4), but the ill-defined NUTS criterion still terminates prematurely.

Implementation status

Vanilla NUTS is implemented in stan, `http://mc-stan.org/`

Riemannian manifold Hamiltonian Monte Carlo is in development.

Acknowledgment

- Timo, Cynthia (Xin)
- UVa grad students
- colleagues who I forgot to mention

Questions?

Betancourt, M. J. (2013). Generalizing the No-U-Turn sampler to Riemannian manifolds. *arXiv preprint arXiv:1304.1920*.

Gubernatis, J. E. (2005). Marshall Rosenbluth and the Metropolis algorithm. *Physics of Plasmas (1994-present)*, *12*(5). doi: http://dx.doi.org/10.1063/1.1887186

Hoffman, M. D., & Gelman, A. (2011). The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *arXiv preprint arXiv:1111.4246*.

Neil, R. M. (2010). MCMC using Hamiltonian dynamics. In Steve Brooks, Andrew Gelman, Galin Jones, & Xiao-Li Meng (Eds.), *Handbook of Markov Chain Monte Carlo* (pp. 113–162). Chapman & Hall / CRC Press.