Metropolis Hastings, Langevin, and Hamiltonian Monte Carlo

Anthony Bugatto

June 2022

1 Monte Carlo Algorithms

Monte Carlo algorithms are a framework of statistical algorithms that rely on sampling to produce numerical results.

1.1 Monte Carlo Markov Chain (MCMC)

Monte Carlo Markov Chains are built on the assumption that each state is only dependant on the previous state, or the Markov property. The stochastic probability of transitioning from state x_k to x_{k+1} creates a random walk, eventually converging to a stationary distribution corresponding to the target distribution. The downside is that the random walk sampling regime results in very high convergence times with high correlation between the samples.

1.2 Metropolis Hastings

The metropolis hastings algorithm solves the problems of MCMC's by using a sample-reject sampling scheme. The acceptance probability is defined as the proportion of target probabilities at sample x_k and x_{k+1} scaled with the proportion of the forward and reverse transition probabilities to account for asymmetric distributions. The acceptence probability is compared to a unit uniform random variable as such:

$$acceptance = min(1, \frac{\pi(x_k)Q(x_{k-1}|x_k)}{\pi(x_{k-1})Q(x_k|x_{k-1})})$$

and for random variable $u \in N(0, 1)$:

$$u < acceptance => sample_k = x_k$$

1.3 Langevin Dynamics

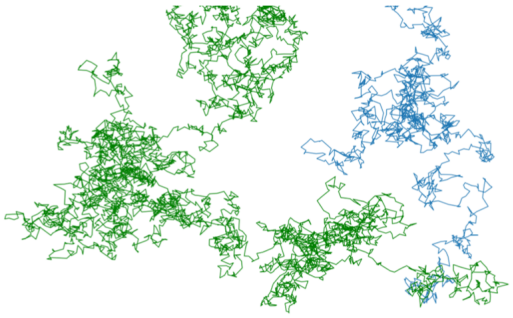

Figure 1: Langevin Dynamics

Langevin dynamics uses gradient information of the target distribution to converge more quickly and move around the distribution. Because of the discretation from the continuous langevin equation the dynamics no longer converge, and consequently will have the same look as brownian motion generally. The metropolis adjustment improves the efficiently significantly. The Euler-Murayama update rule is:

$$x_{k+1} = x_k + \nabla\pi(x_k) + \tau B_k$$

where B_k is brownian motion with distribution $N(0, 1)$. In the case of high correlation we can use every nth sample.

1.4 Hamiltonian Dynamics

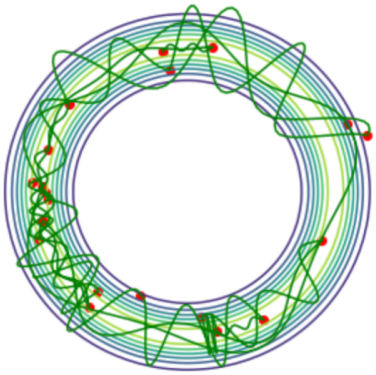

Figure 2: Hamiltonian Trajectories on Donut Distribution

Hamiltonian monte carlo samples trajectories along the target distribution manifold and uses the final value of each trajectory as the monte carlo sample. These samples are then used with the a modified metropolis acceptance probability. The hamilton equaitons are general physics equations that take the energy function of a function and output the equations of motion in the form of a system of first order differential equations with respect to generalized position and momentum. The hamilton energy function is:

$$H(q,p) = U(q) + K(p)$$

with $U(q)$ being the potential energy and $K(p)$ being the kinetic energy. The hamilton energy can be plugged into the hamilton equaitons to get the system of equations:

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}$$
$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}$$

Integrating for the new values (q,p) at each step in discrete time we can use leapfrog integration, a simplectic integrator that preserves energy between iterations. This operates by updating the generalized momentum with a half timestep once before and after a generalized position update:

$$p_i^{half} = p_{i-1} + \frac{\Delta t}{2}\nabla U(q_i)$$
$$q_{i+1} = q_i + \Delta t p_i^{half}$$
$$p^i = p_i^{half} + \frac{\Delta t}{2}\nabla U(q_{i+1})$$

In the case of monte carlo using an arbitrary target distribution we assume a momentum term taking the form of a zero centered normal distribution. The target function becomes:

$$H(q,p) = -log(\pi(q)) + \frac{1}{2}p^T p$$

where the potential is the negative log target distribution and the kinetic energy is the normal distribution squared. This along with leapfrog integration can be used to move along the target distribution manifold, sampling periodically to decrease the correlation while converging to the target more efficiently.

2 Implementation and Results

In my implementation it was important for my understanding of the material to be able visually inspect the results, which is why I programmed every MCMC algorithm in 2D. The downside of this is that diagnostics such as autocorrelation, monte carlo standard errors, and gelman rubin were difficult to implement. Consequently, the analysis of each algorithm will be by inspection on 2D distributions. Each algorithm was implemented from scratch in python using only numpy and matplotlib to aid the learning process.

2.1 Test Distributions

In order to facilitate visual inspection of each algorithm, five multimodal and or disjoint distributions were chosen to demonstrate the convergence, covering, and correlation characteristics of each sampling regime:

- Gaussian Distribution: Easy to sample using monte carlo and even analytical methods. Functions as a good test of the algorithm works.
- Bimodal Gaussian Distribution: Also possible to sample with analytical methods but requires monte carlo to find multiple modes
- Rosenbrock Distribution: Difficult for monte carlo algorithms to evenly sample in reasonable time with low autocorrelation. This distribution wouldn't display properly in pyplot
- Donut Distribution: Difficult for single step gradient based monte carlo algorithms to evenly sample
- Disjoint Multimodal Gaussian Distribution: Difficult for monte carlo algorithms to find each mode without parameter tuning

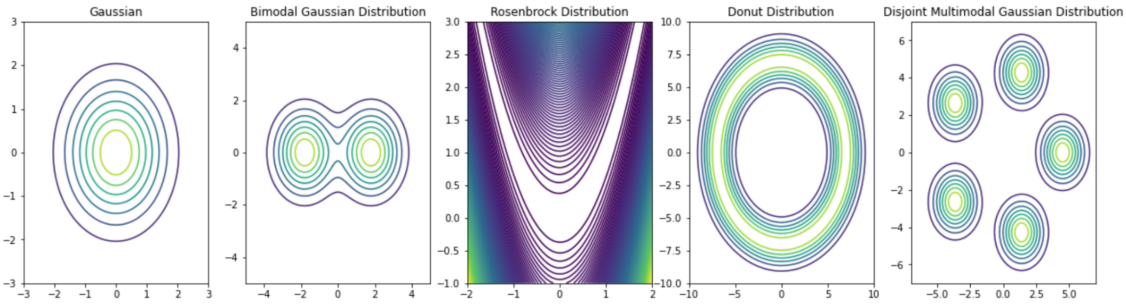

Figure 3: Test Distributions

2.2 Metropolis Hastings

The metropolis hastings performed very well on the gaussian, though this experiment doesn't show samples fully covering the space. We can see that the bimodal distribution is adequately sampled, showing the ability to sample multimodal distributions. However, the donut and disjoint multi-modal distributions were very poorly sampled after 5000 iterations. We can see that they likely would converge after many iterations but since they move so slowly around the distributions they would have very high correlation.

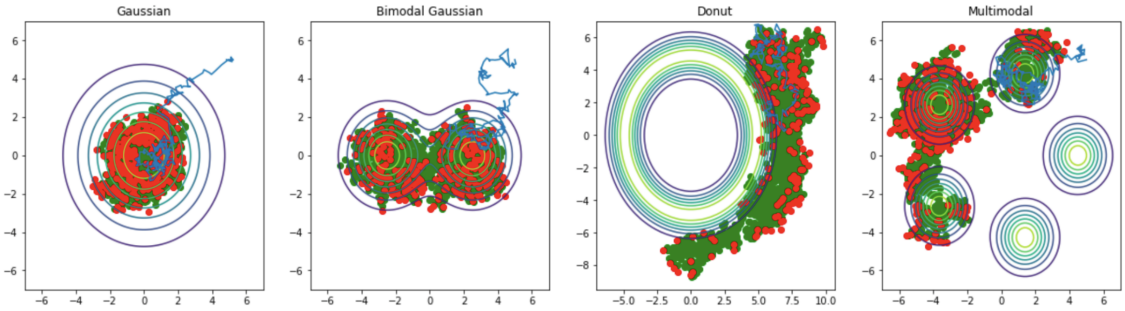

Figure 4: Metropolis Hastings

2.3 Langevin Monte Carlo

The langevin monte carlo is the only one tested without the metropolis hastings accept-reject regime. Consequently, it takes much longer to converge and in most cases it never does. We can see that the algorithm fails to converge in all but the donut distribution. It is probable that the donut converged because the initial point was within the distribution and it is a single mode distribution with a direction for the langevin process to travel. Though there were no empirical examples in the tests, it seems likely that different step or variance parameters would result in a better convergence.

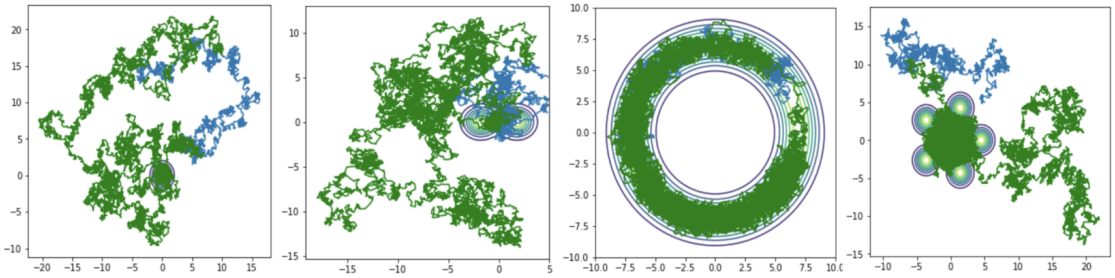

Figure 5: Langevin Monte Carlo

2.4 Metropolis Adjusted Langevin Algorithm

The metropolis adjusted langevin algorithm performs significantly better than the unmodified langevin algorithm, achieving much faster convergence, high sample convergence, and the ability to sample all of the given distributions fully after roughly 20000 iterations. The downside as with the other algorithms is that the langevin dynamics moves slowly around the distribution, likely causing high correlation. This may be solved by better step or variance parameters.

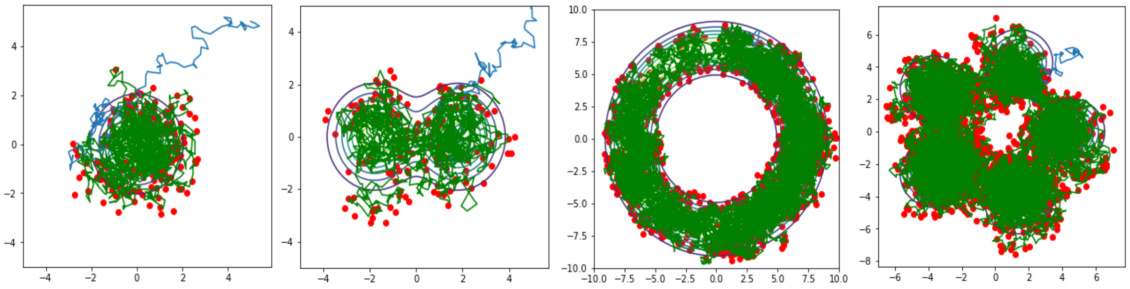

Figure 6: Metropolis Adjusted Langevin Algorithm

2.5 Hamiltonian Monte Carlo

Hamiltonian monte carlo successfully samples from the manifold. In the donut distribution it can be seen that the trajectories take the form of waves along the donut. We can also see that there are very few rejections and that the samples seem very far apart and evenly distributed.

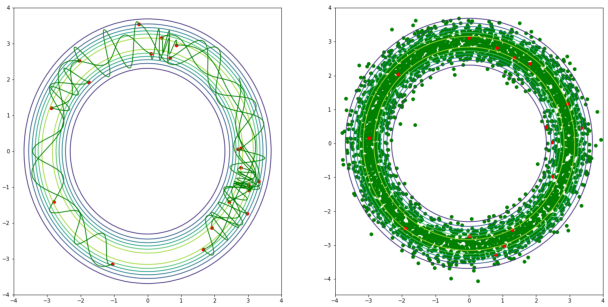

Figure 7: Hamiltonian Monte Carlo on the Donut Distribution

2.6 Convergence and Correlation

Though I would have liked to use some convergence diagnostics it didn't seem like they made sense with my target distributions since they were multi-modal and disjoint, lacking a mean and variance like a gaussian. I implemented monte carlo standard errors but I am unclear on how it would tell us anything if the target distribution is multi-modal or disjoint. With more time I may have implemented gelman rubin if it made sense on my test distributions.
One potential diagnostic could have been a dot product autocorrellation. This could work since it's a 1D trajectory and the dot product seems like a suitable generalization to the multiplication in an autocorrelation on 1D, single parameter data.

3 Conclusions and Future Work

It is clear that monte carlo markov chain variations can be extremely useful for sampling, estimation, and analysis of distributions, both unknown and known. For the metropolis hastings, langevin monte carlo, and hamiltonian monte carlo algorithms the results gave exactly what was predicted. The data showed that the accept-reject sampling regime is a game changer, significantly decreasing the number of samples it takes to converge to the stationary distribution. Using the langevin dynamics with appropriate noise and step size can make it much easier to estimate multi-modal or non-ordinary distributions. However, all of the experiments may still fail to have low correlation or convergence. For more formal results convergence diagnostics would be an important next step to verify the effectiveness of the algorithms.

4 Sources

- https://arxiv.org/pdf/1701.02434.pdf
- https://colindcarroll.com/2019/04/11/hamiltonian-monte-carlo-from-scratch/
- https://arxiv.org/pdf/1812.07978.pdf
- https://www.icts.res.in/sites/default/files/paap-2019-08-08-Eric
- https://link.springer.com/content/pdf/10.1007/s11222-020-09986-y.pdf
- https://chi-feng.github.io/mcmc-demo/app.html?algorithm=MALAtarget=banana