

HMC Talk Notes

Evan Miyakawa

4/7/2020

- When you have more complexity than information (like in big data with more variables than observations), having a prior allows us to do inference that we couldn't do otherwise
- Doing Bayes is hard because taking integrals is hard. We have to take integrals in Bayes equation
- Need to do numerical approximation
- We want to go where the integrand is big and focus our energy/computation there
- Problem is that, the higher in dimension we go, the most highest volume of our parameter space gets smaller and smaller relative to the rest of the space
- The product of density and volume (which is probability mass) $\pi(q)dq$ is what contributes to an integral
- A *Typical Set* is the center part of the probability mass around the mode
- One solution is Monte Carlo, which is drawing samples from the mass
- Approximate integrating over prob mass by integrating over the typical set, and approx integrating over the typical set by integrating over the monte carlo samples of the typical set. Average over the MC samples
- Monte Carlo estimators average a given function over these samples to approximate the expectation
 - $\frac{1}{N} \sum_{n=1}^N f(q_n) \sim N\left(E[f], \frac{\text{Var}[f]}{N}\right)$
- MC standard error is independent of the dimension of the data because we can quantify our MC error with the expression above
 - Problem is that drawing samples is hard. We haven't found the typical set yet
- Markov Chain allows us to get correlated samples
 - We can get transition density which preserves target distribution
 - * $\pi(q) = \int dq' \pi(q') T(q|q')$, where q' is the sample you just took and T is transition?
 - Our transition density are going to concentrate towards the areas of high probability mass
- If run long enough, our MCMC chains consistently estimate MC samples. Problem is that this is only true asymptotically. If you do it infinitely long, you will get the true expectation.
- How do you know how many initial samples to throw out? And how do you know if it's run long enough to get close enough to the true expectation?
- If we have geometries that are complex and small, our MCMC estimators will never explore that region, which leads to biased estimators
- Random Walk Metropolis explores posterior with "guided" diffusion
 - Add some random noise and have a proposed random step. Then accept or reject that point based on probability of density
 - $T(q|q') = N(q|q', \sigma^2) \min\left(1, \frac{\pi(q)}{\pi(q')}\right)$
 - In high dimensional spaces, almost all of the potential locations you could jump to will end up being rejected because they fall outside the typical set. Very slow algorithm
 - This is an inefficient guess-and-check procedure

HMC:

- If we can create a vector field that will tell us where to go to explore the entire set, then we can just follow the arrows
- Each parameter q has an associated momentum variable p .
- Construct prob distribution for momentum variables as well.

- $\pi(q, p) = \pi(q|p)\pi(p)$. This structure allows us at any time to throw out momentum and recover just $\pi(p)$, which is what we are interested in.
- The definition of the Hamiltonian:

$$\begin{aligned} H(q, p) &= -\log \pi(p|q)\pi(q) \\ &= -\log \pi(p|q) - \log \pi(q) \\ &= -K - V, \end{aligned}$$

where K is potential energy and V is kinetic energy.

- We now have a vector field for kinetic and potential. (meaning we have derivatives for each that point in a certain direction)
- (Can include some images of HMC visually from lecture / paper)
- HMC is very robust, hard to break
 - When it does break, it's super obvious
- “STAN is a platform for Bayesian inference that tries to abstract as much computation away as possible, so users can focus on building complex models”