10-708: Probabilistic Graphical Models 10-708, Spring 2018

19 : Slice Sampling and HMC

Lecturer: Kayhan Batmanghelich *Scribes: Boxiang Lyu*

1 MCMC (Auxiliary Variables Methods)

In inference, we are often interested in some form of expectation of the function $f(x)$, and we wish to find the value $\int f(x)P(x)dx$. The idea behind auxiliary variables is to introduce an auxiliary variable v, completely fabricated around the variable x, such that the distributions of $P(x|v)$ and $P(v|x)$ is easy, and $P(x, v)$ lives in a space which is easy to navigate. We can then write the expectation as

$$\int f(x)P(x)dx = \int f(x)P(x,v)dxdv \approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}),\ x, v \sim P(x, v)$$

If drawing samples from $P(x|v)$ and $P(v|x)$ is tractable, we may then draw the samples x, v from the distribution easily. To achieve this, the auxiliary variable v must influence how we can draw a sample x and drawing v must be influenced by the value of x.

2 Slice Sampling

Slice sampling is similar to Gibbs sampling in that it samples one variable at a time. It is also similar to rejection sampling in some sense where we draw samples from region under the curve. However, it is slightly more efficient because slice sampling does not require multiplying the sample density with a large number M. It is similar to MCMC in that the proposal is adaptive.

The general idea is to introduce an auxiliary variable y and define the joint distribution of x, y to be

$$P(x,y) = \begin{cases} 1/Z, & 0 \le y \le p^*(x) \\ 0, & \text{otherwise} \end{cases}$$

where $p(x)$ is the distribution we are interested in, but we do not know the normalization constant Z and only have access to the unnormalized density $p^*(x)$ (or an estimate of the unnormalize density) where $p(x) = \frac{p^*(x)}{Z}$. In this case, we sample y from 0 to $p^*(x)$ using a uniform distribution.

Even though we don't have access to the normalization constant Z, we can fix a certain x and draw from the conditional distribution $P(y|x) \sim Uniform(0, p^*(x))$ instead.

We observe that marginalizing over y in this case gives us the following integral

$$\int p(x,y)dy = \int_0^{p^*(x)} \frac{1}{Z}dy = \frac{p^*(x)}{Z} = p(x)$$

Note that this auxiliary variable y also has the nice properties we want from an auxiliary variable, as:

$$p(y|x) = Uniform(0, p^*(x))$$
$$p(x|y) = \begin{cases} 1, & p^*(x) \geq y \\ 0, & otherwise \end{cases}$$

.

To visualize the two equations, the first equation is related to a "vertical slice" of the distribution $p(x)$ and the second equation is related to a "horizontal slice". If we can calculate the second term exactly, the slice sampling could transform this into a collection to uniform distributions and can be done very fast. However, the problem is that sampling from $p(x|y)$ may not always be feasible, and we introduce the following slice sampling algorithm to solve the problem.

2.1 Algorithm

Start at some x, plug into the unnormalized distribution $p^*(x)$. Sample from the "vertical slice", using the distribution $p(y|x)$ to obtain a value for y. We then sample from the "horizontal slice", sampling from an interval encompassing x with width w. We keep on expanding w until both end points are outside of the region under the curve, which can be determined fairly easily using the estimated density $P^*(x)$.

We may then find the updated value for x by sampling from this interval that we obtain b. In other words, we sample from the distribution $Uniform(x-w, x+w)$ for an updated value of x. The pseudo-code for the algorithm is shown below.

Goal: sample (x, u) given $(u^{(t)}, x^{(t)})$.
$u \sim \text{Uniform}(0, p(x^{(t)}))$
Part 1: Stepping Out
  Sample interval (x_l, x_r) enclosing $x^{(t)}$.
    $r \sim \text{Uniform}(u, w)$
    $(x_l, x_r) = (x^{(t)} - r, x^{(t)} + w - r)$
  Expand until endpoints are "outside" region under curve.
    while$(\tilde{p}(x_l) > u)\{x_l = x_l - w\}$
    while$(\tilde{p}(x_r) > u)\{x_r = x_r + w\}$
Part 2: Sample x (Shrinking)
while(true) {
  Draw x from within the interval (x_l, x_r), then accept or shrink.
    $x \sim \text{Uniform}(x_l, x_r)$
    if$(\tilde{p}(x) > u)$\{break\}
    else if$(x > x^{(t)})\{x_r = x\}$
    else$\{x_l = x\}$
}
$x^{(t+1)} = x,\ u^{(t+1)} = u$

Figure 1: Slice Sampling Algorithm

In the multivariate case, we re-sample the value for each dimension x_i of the variable x one at a time, similar to what we do in Gibbs sampling. We do not to consider the distribution of this distribution x_i, conditioned on other dimensions of the variable directly and can an unnormalized estimate of the conditional distribution instead.

3 Hamiltonian Monte Carlo

To understand the motivation behind Hamiltonian Monte Carlo, we first examine the reasons why the Metropolis-Hastings algorithm can be inefficient at times. Recall that samples from the Metropolis-Hastings algorithm can exhibit random walk behavior, leading to more rejections when the variance of the samples is too high, and slower movements in the distribution space when the variance is too slow.

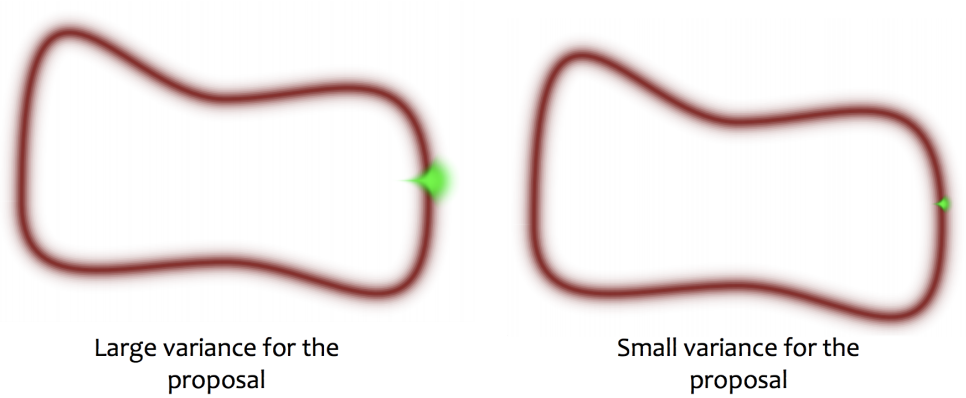

Figure 2: How Variance of Proposal Affects Metropolis-Hastings

Hamiltonian Monte Carlo attempts to ameliorate this issue by assuming and utilizing the smoothness of the density function $p(x)$. The direction in which the sample moves is "guided" by the gradient of the function $p(x)$, and will help us explore the distribution space faster.

In order to utilize the smoothness, we need a vector field for vectors to "guide" us. In other words, we wish to obtain a vector field where the directions of the vectors are aligned to the high probability regions. Here we use a different notation. Let p be the auxiliary variable (momentum) instead, and we introduce a distribution $\pi(p|x)$ to form the joint distribution

$$\pi(x, p) = \pi(p|x)\pi(x)$$

The expanded system defines a Hamiltonian system that decomposes into a *potential* energy and *kinematic* energy, where we let the Hamiltonian be

$$H(x, p) = -\log \pi(x, p) = -\log \pi(p|x) - \log \pi(x)$$

where $-\log \pi(p|x)$ is referred to as kinematic energy $K(p, x)$ and $-\log \pi(x)$ potential energy $E(x)$ We use the gradient information.

Further suppose that the distribution $\pi(x)$ is of the form $\pi(x) = \exp(-E(x))/Z$ for $x \in \mathbb{R}^n$. While the Metropolis-Hastings algorithm can still be used in this case, the gradient information $\nabla_x E(x)$ can be used to help speed up the exploration of the distribution space, as the gradient is capable of showing us the high-probability regions.

The vector field that we are interested in and guides our sampling then consists of the following derivatives

$$\frac{dx}{dt} = \frac{\partial K(x,p)}{\partial p}$$
$$\frac{dp}{dt} = -\frac{\partial K(x,p)}{\partial x} - \frac{\partial E(x)}{\partial x}$$

where the first equation acts like velocity, and the second equation can be thought of as a corrected version of the potential energy in the system.

A popular choice for the kinematic energy is $K(p) = p^T p/2$, and the resulting Hamiltonian can be written as

$$H(x,p) = E(x) + K(p)$$

3.1 Algorithm

With these quantities defined, the Hamiltonian Monte Carlo algorithm could be explained. We start at an arbitrary point in the space and use the Hamiltonian dynamics defined above to determine the next sample.

The Leapfrog method shown below is often used to simulate the dynamics. The numerical trajectories produced by the Leapfrog method will not deviate away from the exact trajectory as the integration continues, while most numerical integrators suffer from this drawback.

Parameters to tune:

1. Step size, ϵ
2. Number of iterations, L

Leapfrog Algorithm:

for τ in $1 \dots L$:

$$\boldsymbol{p} = \boldsymbol{p} - \frac{\epsilon}{2}\nabla_{\boldsymbol{x}} E(\boldsymbol{x})$$
$$\boldsymbol{x} = \boldsymbol{x} + \epsilon \boldsymbol{p}$$
$$\boldsymbol{p} = \boldsymbol{p} - \frac{\epsilon}{2}\nabla_{\boldsymbol{x}} E(\boldsymbol{x})$$

Figure 3: The Leapfrog Method

With the help from the Leapfrog method, we can define the Hamiltonian Monte Carlo algorithm. Let $\pi(x) = \exp(-E(x))/Z$ be the distribution of interest, $K(p) = p^T p/2, H(x,p) = E(x) + K(p)$. The joint distribution is then

$$\pi(x,p) = \exp(-H(x,p))/Z_H = \exp(-E(x))\exp(-K(p))/Z_H$$

Observe that the distribution above is separable, and we have

$$\sum_p \pi(x,p) = \exp(-E(x))/Z$$
$$\sum_x \pi(x,p) = \exp(-K(p))/Z_K$$

where the first term is exactly our target distribution, and the second term follows a Gaussian distribution, assuming p is a random variable. We then perform the following update steps for the HMC algorithm.

1. Sample momentum (p) from distribution implied by the kinetic $\pi(p|x)$.
2. Update (x,p) according to Hamiltonian Dynamics
$$x \leftarrow x + \epsilon \frac{\partial K}{\partial p}$$
$$p \leftarrow p - \epsilon \left(\frac{\partial K}{\partial x} + \frac{\partial E}{\partial x} \right)$$
3. Accept/Reject the new sample
$$\pi(\text{accept}) = \min\left(1, \frac{\pi(\Phi_\tau(x,p))}{\pi(x,p)}\right)$$

Figure 4: Hamiltonian Monte Carlo Algorithm