3: Markov Chain Monte Carlo

Introduction

While the copulas can be useful for many applications, ultimately they only work exactly if the distribution we are looking to sample from happens to have a copula we know how to work with.

Let's return to the basic question: how do we sample from a d-sample dimensional density $f(x_1, ..., x_d)$?

Markov Chain Monte Carlo gives us a way of generating samples that closely approximate this distribution. Moreover, it turns out that we often don't even have to know the exact density; we just need to know it up to a constant of proportionality.

A note about Bayesian Statistics

This kind of problem frequently arises in Bayesian statistics. The Bayesian approach works as follows. Our data $X_1, ..., X_n$ is assumed to come from a parametric model $P(X_1,...,X_n|\Theta)$. We have some parameters Θ about which we have some prior information, which can be expressed in terms of a prior distribution $P(\Theta)$ - this could be the outcome of some previous experiments, some knowledge that the investigator has, or just the application of certain heuristics. Given the data and the prior, we can apply Bayes rule to get the posterior distribution for Θ - our estimates will be the expectation or mode or medians of this distribution:

$$P(\Theta|X_1, ..., X_n) = \frac{P(X_1,...,X_n|\Theta)P(\Theta)}{P(X_1,...,X_n)}$$

where $P(X_1, ..., X_n) = \int P(X_1, ..., X_n|\Theta)P(\Theta)d\Theta$, which is typically a high-dimensional integral, and difficult to approximate.

MCMC allows us to just work with the numerator in this expression: $g(\Theta|X_1, ..., X_n) = P(X_1, ..., X_n|\Theta)P(\Theta)$.

The basic idea

As you presumably know, a Markov chain is a sequence of random variables $X_1, X_2, ...$ with shared support S, a starting probability distribution p_1 for X_1 and a transition kernel $p(x|y) = p(x|X_{i-1} = y)$ i.e. the conditional density (or mass function) of $X_i|X_{i-1} = y$. Notably, the transition kernel depends only on the value of the previous element in the sequence; the history of the path before that does not matter.

Generally speaking, the unconditional distribution p_j of X_j will not be the same as p_i of X_i; however, given some technical conditions, there is a choice of p_1 - we'll call it π that causes the Markov chain to be stationary; that is, every random variable has the same distribution π.

Moreover, regardless of the choice of p_1, $\lim_{i \to \infty} p_i = \pi$.

So the idea for MCMC is that we are going to choose a transition kernel on our support in a clever manner, such that the limiting/stationary distribution is $f(x_1, ..., x_n)$. Then, regardless of how we X_1, X_j for j large will be have approximately distribution $f(x_1, ..., x_n)$, and our sequence will be a (correlated sample) from our target distribution.

Metropolis-Hasting Algorithm

So, under which technical conditions does a stationary distribution exist?

Essentially, if for all x, y in S, we have the following is true:

$$p(x|y)f(y) = p(y|x)f(x)$$

for some density $f(x)$, then $f(x)$ will be the stationary distribution. This is called the detailed balance condition. We will actually work backwards here; we know $f(x)$ (up to a constant of proportionality), so we can use to find a transition probability:

$$\frac{p(x|y)}{p(y|x)} = \frac{f(x)}{f(y)}$$

We will further split the transition into two sub-steps: (1) proposing a new location with density $h(x|y)$ and (2) accepting with $a(x|y)$. We re-write the above equation:

$$\frac{a(x|y)}{a(y|x)} = \frac{f(x)h(y|x)}{f(y)h(x|y)}$$

For an arbitrary choice of $h(\cdot|\cdot)$, we can see that if we choose:

$$a(x|y) = \min(1, \frac{f(x)h(y|x)}{f(y)h(x|y)})$$

the detailed balance condition will hold.

Moreover, suppose we only know the density $f(x)$ up to a constant of proportionality, i.e. we known $g(x) = Mf(x)$ but we do not know exactly what M is. However, we see that if we suppose g into the above equation, the M terms will cancel out.

The Metropolis-Hastings algorithm then is as follows:

1. Let n = 0, let X_n = y (this can be generated from an arbitrary distribution
2. Generate a proposal x from $h(x|y)$
3. Calculate the acceptance probability $A(x|y) = min(1, \frac{g(x)h(y|x)}{g(y)h(x|y)})$
4. Generate $U \sim Unif(0, 1)$, if $U \leq A(x|y)$, then $X_{n+1} = x$, else $X_{n+1} = y$ (i.e. we stay at the same place)
5. Let n = n+1, go back to step 2

What is cool about this is that we do not suffer from the curse of dimensionality - the acceptance probability depends ultimately on the $\frac{g(x)h(y|x)}{g(y)h(x|y)}$; if $h(y|x) = h(x|y)$ this will just be $\frac{g(x)}{g(y)}$

Gibbs Sampler

The Gibbs sampler is a special version of the Metropolis-Hastings algorithm.

Suppose we want to sample from the density $f(x_1, ..., x_d)$; we will use the transition density. We will construct a Markov chain that updates one coordinate at a time, randomly choosing a coordinate to update. Let's first lay out the algorithm, and then dig into how it works.

1. We start with a point X_0 in the sample space S. Then we let n=1.
2. Randomly choose j from $1, ..., d$ with equally likely probabilities (i.e. $\frac{1}{d}$). Then let $X_n(i) = X_{n-1}(i)$ for $i \neq j$.
3. Update $X_n(j)$ according to $P(X(j)|X(i) = X_n(i), \forall i \neq j)$
4. Let n=n+1 and goto step 2

Essentially, we are using the following transition kernel at every step here:

$$\begin{aligned} p(x|y) &= \frac{1}{d} p(x(j)|x(i) = y(i), \forall i \neq j) \\ &= \frac{p(x)}{dp(x(i) = y(i), \forall i \neq j)} \end{aligned}$$

If we plug this kernel into the acceptance formula for metropolis hastings:

$$\begin{aligned} A(x|y) &= \min(1, \frac{p(x)p(y)p(x(i) = y(i), \forall i \neq j)}{p(y)p(x)p(y(i) = x(i), \forall i \neq j)} \\ &= 1 \end{aligned}$$

So there is no need for an acceptance-rejection step; giving us the exact algorithm above as a special case of metropolis-hastings.

Random Walk Metropolis-Hastings

We will briefly introduce the idea of a d-dimensional multivariate normal distribution with parameters $\mu \in \mathbb{R}^d$ and $d \times d$ symmetric positive definite matrix Σ.

They have the following density:

$$p(x|\mu, \Sigma) = (2\pi)^{-d/2}|det(\Sigma)|^{-1/2} \exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$$

Such random vectors can be easily sampled regardless of dimension.

Random walk Metropolis uses the transition kernel $h(x|y) = p(x|y, \Sigma) = h(y|x)$

This gives us the Metropolis-Hastings acceptance kernel:

$$a(x|y) = \min\left(1, \frac{h(y|x)f(x)}{h(x|y)f(y)}\right) \tag{1}$$

$$= \min\left(1, \frac{f(x)}{f(y)}\right) \tag{2}$$

This works reasonably well for some problems, but for highly peaked densities, you will tend to get stuck around the peak and only move away slowly.

Hamiltonian Monte Carlo

Introduction

Hamiltonian Monte Carlo is the paradigm within which most modern MCMC schemes work. It is inspired by physics - we are applying random amounts of force to our parameters, and having them move around a surface given by the posterior.

Sampling

In order to generate proposals, we generate a random "momentum vector" ρ and count the total energy in the system as follows:

$$\begin{aligned} H(\theta, \rho) &= -\log(p(\theta, \rho) \\ &= -log((\rho|\theta)) - \log(p(\theta)) \\ &= T(\rho|\theta) + V(\theta) \end{aligned}$$

where $T(\rho|\theta)$ is the "kinetic energy" in the system and $V(\theta)$ is the "potential energy."

Conservative Systems

In general, we will have ρ be an independently sampled multivariate normal with mean $\mathbf{0}$ and covariance matrix Σ, so we can just use $T(\rho)$.

If $\theta(t)$ and $rho(t)$ are the paths of $theta$ and rho across time, then the energy in the system evolves like so over time:

$$\frac{\partial}{\partial t} H(\theta, \rho) = \frac{\partial H}{\partial \theta} \frac{\partial \theta}{\partial t} + \frac{\partial H}{\partial \rho} \frac{\partial \rho}{\partial t}$$

This system will be conservative (i.e. no energy will be lost) if:

$$\frac{\partial \theta}{\partial t} = \frac{\partial H}{\partial \rho} = \frac{\partial T}{\partial \rho}$$

and

$$\frac{\partial \rho}{\partial t} = -\frac{\partial H}{\partial \theta} = -\frac{\partial V}{\partial \theta}$$

We additionally note that $\frac{\partial T}{\partial \rho} = \Sigma^{-1}\rho$ (we get this by differentiating the log of multivariate normal density.

Solving this set of integral equations will allow us to traverse a contour of the surface that has a constant value of $H(\theta, \rho)$, i.e. a level set.

Leapfrog Integrator

In an ideal world, we could solve this system of integral equations directly but in practice that will not be the case. Instead, we will numerically integrate them. The leapfrog integrator is guaranteed to be numerically stable for solving these kinds of problems; it consists of evolving ρ in half steps with evolutions of θ in between. We update steps of size ϵ as follows (starting at ρ_0 and θ_0:

$$\begin{aligned}\rho_{1/2} &= \rho_0 - \epsilon/2\frac{\partial V}{\partial \theta_0} \\ \theta_1 &= \theta_0 + \epsilon\Sigma^{-1}\rho_{1/2} \\ \rho_1 &= \rho_{1/2} - \epsilon/2\frac{\partial V}{\partial \theta_1}\end{aligned}$$

Let us denote $\theta_K(\theta, \rho)$ $\rho_K(\theta, \rho)$ be the values we get by running the leepfrog integrator for K iterations starting at θ and ρ. It should be clear that this is a deterministic path given the start point, and moreover, if we start at $\theta_K(\theta, \rho)$ and apply initial momentum $-\rho_K(\theta, \rho)$, we will end up at θ and $-\rho$ after K steps. Since we are approximating a conservative system, the Hamiltonian should hopefully change very minimally from the start to the end of the path.

Putting it all together

We can apply this to Metropolis-Hastings as follows (given a step size ϵ, number of steps K, and Covariance Matrix Σ - these are essentially the hyperparameters of the simulation):

(1) Starting at $X_t = \theta$, generate $\rho \sim MVN(0, \Sigma)$

(2) Evolve the system K steps to end up at $\theta_K(\theta, \rho)$.

Note that our transition kernel is $h(\theta_K(\theta, \rho)|\theta) = p_\rho(\rho) = \exp(-T(\rho))$ and the reverse direction $h(\theta|\theta_K(\theta, \rho)) = \exp(-T(-\rho_K(\theta, \rho)) = \exp(-T(\rho_K(\theta, \rho))$ since the multivariate normal density is symmetric around zero.

Note then that $h(\theta_K(\theta, \rho)|\theta)p(\theta) = \exp(-T(\rho))\exp(-V(\theta)) = \exp(-H(\rho, \theta))$, so:

(3) Our acceptance kernel is then:

$$a(\theta_K|\theta) = \min(1, \exp(H(\rho, \theta) - H(\rho_K(\theta, \rho), \theta_K(\theta, \rho))))$$

which, should, by construction, be close to one...