Markov Chain Monte Carlo

Markov Chain Monte Carlo methods sample from a probability distribution p(x) by creating a Markov chain whose stationary probability distribution is p(x). The sequence of states of the Markov chain after a sufficient number of steps to reach a random state given the initial state, provide a good sample of the distribution.

Given a multi-dimensional probability distribution $p(x_1,\cdots,x_n)$ where each variable ranges over a finite set of values, one might wish to calculate the marginal distribution

$$p(x_1)=\sum_{x_2,\cdots,x_n} p(x_1,\cdots,x_n)$$

or the expectation of some function $f(x_1,x_2,\cdots,x_n)$

$$E(f)=\sum_{x_1,\cdots,x_n} f(x_1,\cdots,x_n)\,p(x_1,\cdots,x_n).$$

The difficulty is that both computations require a summation over an exponential number of values. One could compute an approximation to these problems if one could generate a set of values for $x_1,\cdots,x_n$ according to the distribution $p(x_1,\cdots,x_n)$. If c is the total number of values generated and if $c(x_1,\cdots,x_n)$ is the total number of times that the value $x_1,\cdots,x_n$ is generated, then

$$p(x_1,\cdots,x_n)\cong\frac{c(x_1,\cdots,x_n)}{c}$$

Thus, the marginal distribution is calculated by

$$p(x_1)=\sum_{x_2,\cdots,x_n} p(x_1,\cdots,x_n)\cong\frac{\sum_{x_2,\cdots,x_n} c(x_1,\cdots,x_n)}{c}$$

and the expectation is calculated by

$$E(f)=\sum_{x_1,\cdots,x_n} f(x_1,\cdots,x_n)\,p(x_1,\cdots,x_n)\cong\frac{\sum_{x_1,\cdots,x_n} f(x_1,\cdots,x_n)c(x_1,\cdots,x_n)}{c}.$$

The question then is how to generate random samples according to the distribution $p(x_1,\cdots,x_n)$.

Consider an undirected graph whose vertices corresponded to the possible values of $(x_1,x_2,\cdots,x_n)$ called states. Suppose at a vertex we could compute the probabilities on the incident edges so that the stationary probability of a random walk on the vertices of the graph would be the distribution $p(x_1,\cdots,x_n)$. Then if we did a random walk on the graph we could get a set of values for $x_1,\cdots,x_n$ that represent the probability distribution over the states. Thus doing a random walk of a few million steps would give us a good approximation.

Two issues that we need to consider are the fact that when we start the random walk we need to walk for a little while until we get to a random position. An second when we get a sample, the next sample is not independent of the previous sample. We might need to walk a while to get independence although this should not take as long as at the

beginning of the walk since the probability of the point we are at is the stationary probability where as the first point was one we picked somehow.

Metropolis-Hasting Algorithm

Metropolis-Hasting Algorithm designs a Markov chain whose stationary distribution is a given target distribution $p(x_1,\cdots,x_n)$. The Markov chain has states that correspond to the set of possible values of $x_1,\cdots,x_n$. The stationary probability of a random walk on a connected, aperiodic, undirected graph has the property that the frequency of the walk from state *x* to state *y* is the same as the frequency of the walk from state *y* to state *x*. Conversely, if we can assign a probability to each edge at each vertex so that

$$p(x)\,p(x\to y)=p(y)\,p(y\to x), \tag{1}$$

then the stationary distribution of the random walk will be p. This follows since

$$p(x)=\sum_y p(x)\,p(x\to y)=\sum_y p(y)\,p(y\to x)$$

which implies that p is a stationary distribution. Alternatively we have specified the probability transition matrix of the underlying graph. Assuming that the graph is connected and aperiodic the stationary distribution is unique and thus the stationary distribution of the random walk is the desired target distribution $p(x)$. The question remains how do we construct $p(x\to y)$ to satisfy (1) not knowing $p(y\to x)$.

Design the transition probabilities $p(x\to y)$ as follows. Select an arbitrary transition probability $q(x\to y)$ that is irreducible and an acceptance function $\rho(x,y)$ and let the Markov chain have transition probabilities

$$r(x,y)=q(x,y)\,\rho(x,y)$$

If we are at state x, the next state is y with probability $q(x,y)$ and acceptance $\rho(x,y)$. The state remains x with probability $1-\rho(x,y)$. We set ρ so that the balance condition is satisfied. To do this, we want

$$p(x)\,\rho(x,y)\,q(x\to y)=p(y)\,\rho(y,x)\,q(y\to x)$$

or

$$\rho(x,y)=\tfrac{p(y)q(y\to x)}{p(x)q(x\to y)}\,\rho(y,x)$$

Since $\rho(x,y)$ and $\rho(y,x)$ are probabilities, they must be less than or equal to 1. If $\frac{p(y)q(y\to x)}{p(x)q(x\to y)}<1$ set $\rho(y,x)=1$ and $\rho(x,y)=\frac{p(y)q(y\to x)}{p(x)q(x\to y)}\,\rho(y,x)$. If $\frac{p(y)q(y\to x)}{p(x)q(x\to y)}\ge 1$ set $\rho(x,y)=1$ and $\rho(y,x)=\frac{p(x)q(x\to y)}{p(y)q(y\to x)}\,\rho(x,y)$. In summary, by defining

$$\rho(x,y)=\min\left(\tfrac{p(y)q(y\to x)}{p(x)q(x\to y)},1\right)$$

we generate a Markov chain whose stationary distribution is the target distribution $p(x)$ and we could use samples from this Markov chain for statistical inference.

Note that $p(x,y)$'s dependency on $p(x)$ is only through the ratio $\frac{p(x)q(x\to y)}{p(y)q(y\to x)}$. Thus, we only need to know $p(x)$ up to a constant in Metropolis-Hastings algorithm. This is important since in many cases p(x) is known only up to a normalizing factor. For example, $p(x_1,\cdots,x_n)$ might be

$$\tfrac{1}{z}(x_1+x_2x_3)(x_1+x_4+x_5)(x_2+x_5)$$

where z is a normalizing factor that makes the probability function over all states sum to one. The evaluation of z requires a sum over an exponential set of values.

Exercise: What is the convergence time of the Metropolis-hasting algorithm?

Exercise: How would we generate two statistically independent values by the Metropolis-Hastings algorithm?

Gibbs sampling

Gibbs sampling is a Markov Chain Monte Carlo method to sample from a multivariate probability distribution. Let $p(x)$ be the target distribution with $x=(x_1,\cdots,x_n)$. At each step of Gibbs sampling for $x=(x_1,\cdots,x_n)$ only one of the x_i's is updated according to its posterior probability $p(x_i \mid x_1,\cdots,x_{i-1},x_{i+1},\cdots,x_n)$. (Posterior probability means the probability of the variable conditioned on the remaining variables being fixed.)
To generate samples for $x=(x_1,\cdots,x_n)$ with a target distribution $p(x)$, the Gibbs sampling algorithm repeats the following steps. One of the variables x_i is chosen to be updated. Its new value is chosen based on its posterior probability with the other variables fixed. There are two commonly used schemes to determine which x_i to update. One scheme is to choose x_i randomly, the other is to choose x_i by sequentially scanning from x_1 to x_n.

Next, we show that the stationary distribution obtained by Gibbs sampling process is indeed the target distribution $p(x)$. One can gain useful insight by viewing the above transition step of Gibbs sampling in the framework of Metropolis-Hastings algorithm. In Metropolis-Hastings algorithm, the acceptance rate of moving from state *x* to state *y* by a $q(x\to y)$ is given as

$$\rho(x,y)=\min\left(\tfrac{p(x)q(x\to y)}{p(y)q(y\to x)},1\right).$$

If we could choose the transition probability $q(x\to y)$ to be proportional to the target

distribution of the destination state *y*, that is, $q(x \to y) \propto p(y)$ and $q(y \to x) \propto p(x)$ the term $\frac{p(x)q(x \to y)}{p(y)q(y \to x)}$ conveniently cancels off, and the acceptance rate $\rho(x, y)$ is always 1 which is nice because the Gibbs sampling would always accept the new state *y* and the Markov chain would mix fast. However, finding such a $q(x \to y)$ is not feasible in general. To make the $q(x \to y)$ a proper transition probability, we need to normalize it over all destination states y

$$q(x \to y) = \frac{p(y)}{\sum_y p(y)}.$$

Since our random variable is $x = (x_1, \cdots, x_n)$, the summation over all possible states *y* in the denominator of $q(x \to y)$ is the summation over all possible configurations of $x = (x_1, \cdots, x_n)$. It is computationally infeasible, and in fact it is the very problem we want to solve by using the MCMC approach.

Note, however, that Gibbs sampling is special in this regard because we allow a transition from state *x* to state *y* only when state *y* differs from state *x* by at most one component. Let *m* be the index of such a component. Then, we can write state *x* as $(x_1 \cdots, x_m^*, \cdots . x_n)$ and state *y* as $(x_1 \cdots, x_m^{**}, \cdots . x_n)$. Since our destination state *y* is confined to an exponentially smaller space than the entire space, computing $q(x \to y)$ proportional to the destination state *y* becomes feasible. Indeed, $q(x \to y)$ becomes the posterior probability of x_m^{**} used in Gibbs sampling.

$$q(x \to y) \propto p(y) = p(x_1, \cdots, x_m^{**}, \cdots, x_n)$$

$$q(x \to y) = \frac{p(x_1, \cdots, x_m^{**}, \cdots, x_n)}{\sum_{x_m} p(x_1, \cdots, x_m, \cdots, x_n)}$$

$$= \frac{1}{n} p(x_m^{**} \mid x_1, \cdots, x_{m-1}, x_{m+1}, \cdots, x_n)$$

Thus, Gibbs sampling can be viewed as a Metropolis-Hastings algorithm with $q(x \to y)$ given above and with acceptance rate being always 1.

The proof is not complete until we consider how *m* is chosen. When *m* is chosen uniformly at random among all indices, $q(x \to y)$ is simply the previous form multiplied by $\frac{1}{n}$,

$$q(x \to y) = \frac{1}{n} p(x_m^{**} \mid x_1, \cdots, x_{m-1}, x_{m+1}, \cdots, x_n)$$

and the above argument of how $\rho(x, y) = 1$ is achieved holds for this new form of $q(x \to y)$ as well. Thus, when *m* is chosen randomly, Gibbs sampling is a Metropolis-Hastings algorithm with acceptance rate 1.

The more common practice in choosing *m* in Gibbs sampling is to scan from x_1 to x_n sequentially. In this case, the above argument does not hold. In fact, this Gibbs sampling approach does not satisfy the detailed balance equation (or time reversibility of a Markov chain) that Metropolis-Hastings algorithm is based on. Nonetheless, one can show that if one starts from a target distribution $p(x_1, \cdots, x_n)$ and applies the Gibbs sampling transition step for x_1 and then for x_2 and so on up to x_n, one obtains $p(x_1, \cdots, x_n)$ back.

Exercise: How would you integrate a multivariate polynomial distribution over some region?

Exercise: Why did we not just assign $p(x \to y) = p(y)$ and $p(y \to x) = p(x)$ to achieve the balance condition in the Metro-Hasting Algorithm?

Exercise: Construct the edge probability for a three state Markov chain so that the stationary probability is $\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right)$.

Exercise: Again consider the three state Markov chain with stationary probability is $\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right)$. In the Metro-Hasting Algorithm one might select $q(y \to x) = \frac{1}{n}$ for all x and y. This would mean all edges are equally likely and one might select an edge by randomly generating an integer between 1 and n. What is the expected probability that the edge would be accepted?

Exercise: Try Gibbs sampling on $p(x) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$.

What happens? How does the Metropolis Hasting Algorithm do?

Exercise: Consider p(x) given by $x = (x_1, \cdots, x_{100})$ and $p(0) = \frac{1}{2}$, $p(x) = \frac{1}{2^{100}} \quad x \neq 0$.

How does Gibbs sampling behave?

Add material on mixing time.