COMP 480/580 — Probabilistic Algorithms and Data Structure Apr. 4, 2019

Lecture 21

Lecturer: Anshumali Shrivastava *Scribe By: Samantha Gilmore and Jingguang Yan*

1 Markov Chains

Recall our previous discussions of stationary Markov chains. Our goal can be defined as follows: given a large state space Ω, we want to sample $x \in \Omega$ such that the probability of sampling x is proportional to $w(x)$.

$$\pi(x) = \frac{w(x)}{Z} \qquad \text{or} \qquad \pi(x) \propto w(x)$$

Recall that Markov chains converge to a unique stationary distribution π if the following two conditions are met:

1. Irreducibility. A Markov chain is irreducible if every state can be reached from every other state. If we are given a Markov chain that is not irreducible and we wish to make it irreducible, we can simply reallocate the probabilities so that there is a very small probability of moving to a randomly sampled node.
2. Aperiodicity. A Markov chain is aperiodic if every state has a self-loop. By self-loop, we mean that there is a nonzero probability of remaining in that state.

By meeting these conditions, we ensure that there is a stationary distribution π for our Markov chain. If $P(x, y)$ is the stochastic matrix representing the transition probabilities of each state, then for a stationary distribution we have $\pi P(x, y) = \pi$.

2 Reversibility

A stochastic matrix $P(x, y)$ is reversible if

$$\pi(x)P(x, y) = \pi(y)P(y, x)$$

where $\pi(x)$ is stationary distribution.

Given a large state space Ω, we want to sample $x \in \Omega$, such that the probability of sampling x is proportional to $w(x)$. Our plan is to:

- Find a Markov chain that has the stationary distribution π, when $\pi(x) = \frac{\omega(x)}{z}$ or $\pi(x) \sim \omega(x)$.
- Since the distribution is stationary, simulating that Markov chain should be easy.

Our plan hinges on the following question. Given a stationary distribution π, can we construct a Markov chain $P(x, y)$ that will converge to π? Fortunately, this can be done using the Metropolis-Hastings algorithm.

3 Metropolis Processes

In Metropolis processes, we have a transition probability $k(x, y)$ that gives the probability of transitioning to state y when already in state x. We assume this transition probability is "easy" and it is not tied to the stationary distribution π. To illustrate the the concept of transition probability, we can think of graphs. If we are currently at state x in the graph, we can make the transition to any other state with probability defined by distribution $k(x, y)$.

We want to use $k(x, y)$ to find $P(x, y)$. Note, we cannot simply set $P(x, y)$ equal to $k(x, y)$ because we would not achieve our desired stationary distribution π. Instead, we will determine $P(x, y)$ by sampling $k(x, y)$ and accepting this with probability $A(x, y)$, so that:

$$P(x, y) = k(x, y) \times A(x, y)$$

Using the reversibility idea we have

$$\pi(x) \times k(x, y) = \pi(y) \times k(y, x) \times A(y, x)$$

We can rearrange this and see

$$\frac{A(x, y)}{A(y, x)} = \frac{w(y) \times k(y, x)}{w(x) \times k(x, y)}$$

We find that the value of our acceptance probability $A(x, y)$ turns out to be

$$A(x, y) = \min \left\{ 1, \frac{w(y) \times k(y, x)}{w(x) \times k(x, y)} \right\}$$

and

$$A(y, x) = \min \left\{ 1, \frac{w(x) \times k(x, y)}{w(y) \times k(y, x)} \right\}$$

This means that if our weight ratio is greater than 1, then we will always accept the sampled state $k(x, y)$ and if it is less than 1, we will accept with probability equal to our weight ratio. We can see that the success of this method clearly depends on the choice of $k(x, y)$. For example, if we simply take $k(x, y)$ as random, we will have a very bad Markov chain. Instead, we can use prior knowledge to get a better choice for $k(x, y)$ to solve our problem.

Algorithm 1: Metropolis Process

Start at x;
Sample $k(x, y)$;
if $k(x, y)$ *is accepted* **then**
    Move to state y;
else
    Stay at state x;
end

Note that $k(x, y)$ represents our neighborhood structure. A neighborhood contains all reachable states from x. Now we pose the question: How do we design $K(x, y)$?

We can use graph sampling to evaluate the likelihood of a structure. For example, we start with any graph, randomly add a node, delete an edge, or do something to that graph and continue to add and remove states in this way. Eventually we can do a random walk over the space and use this as our $k(x, y)$.

A Markov chain with stochastic matrix P is irredicible, aperiodic and has the same stationary distribution as P. That is, for any $0 < \alpha < 1$ we have that

$$P' = \alpha P + (1 - \alpha)I$$

$$\pi P' = \alpha\pi + (1 - \alpha)\pi = \pi$$

4 Rejection Sampling

Now we will contrast Metropolis processes with rejection sampling. We can view rejection sampling as an oversimplified Markov chain where the probability of transitioning is completely independent of the state. Recall that in rejection sampling we sample from a distribution $g(x)$ and accept the sample with probability

$$\frac{f(x)}{\mu \times g(x)} \qquad \text{where } f(x) < \mu \times g(x) \qquad \forall x, \mu \geq 1$$

Rejection sampling works very well in two or three dimensions, but is much less effective in higher dimensions. Recall that rejection sampling assumes we have a $g(x)$ which is similar to $f(x)$. Efficient sampling from a "close" distribution g is less likely as the dimensionality increases. Metropolis processes are also desirable because we can incorporate domain knowledge to make our approach more specific to a certain problem (e.g. robot movement).