Math 340: Lec 27 (Markov Chain Monte Carlo algorithms)

Asa Royal (ajr74)

April 23, 2024

1 Motivation for Markov Chain Monte Carlo Algorithms

Remark. We might want to sample a probability distribution

$$\pi(x) = \frac{f(x)}{c} \text{ for } x \in S$$

where we know $f(x)$ but cannot calculate $c = \sum_{x \in S} f(x)$ because the state space is so large.

To efficiently sample from the distribution π, we can try to generate a Markov chain that has π as its stationary distribution.

Remark. Examples of applications:

1. In Bayesian statistics, when we try to calculate $\mathbb{P}(Y = y|X = x)$, the normalizing denominator $\mathbb{P}(X = x)$ can be very expensive to calculate because it requires us to sum over all possible values of the random variable Y.

2. In cryptography, if we have a substitution cipher, we might create a mapping σ from the cipher alphabet to our normal alphabet. We could then decode an encrypted message using σ and measure how much the decrypted message mimics English letter patterns with some function $f(\sigma)$. But assuming the cipher alphabet has 26 letters, there are 26! possible σ mappings. So to normalize the score of any σ, we'd need to calculate all 26! $f(\sigma)$s. Expensive!!

2 Markov Chain Monte Carlo Algorithms

2.1 Metropolis-Hastings

Theorem 1 (Metropolis-Hastings). Objective: sample from $\pi(x) = \frac{f(x)}{c}$ using a proposal function $q(x, y)$. Metropolis-Hastings generates a Markov Chain X_n on S. Given $X_n = x$, M-H generates X_{n+1} as follows:

1. Propose a new state $y \in S$ according to the probability transition kernel $q(x, y)$

2. Accept or reject the Proposition
   y is accepted with probability

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

If we accept, $X_{n+1} = y$. Otherwise, $X_{n+1} = X_n = x$.

Remark. π is stationary for this Markov Chain, and with an appropriate kernel q, the chain is irreducible + aperiodic.

Example (Example of accept/reject stage of MH). Imagine we have

$$X_n = \sigma = (1, 3, \ldots, 7, 9, 12)y \qquad = \sigma' = (1, 12, \ldots, 7, 9, 3)$$

We check whether $f(\sigma') > f(\sigma)$. If so, we transition to $X_{n+1} = \sigma'$ with a decent probability.

2.2 Gibbs sampling

Theorem 2 (Gibbs sampling). Imagine we have a a graph (V, E) where the m vertices are pictures in an image recognititon dataset. Edges represent shared features between images. z represents the image of a label . We want to calculate $\pi = f(x)/c$, where $f(x)$ is some function involving the degree of a vertex. But there are so many edges and vertices that calculating c is impractical. Instead, we find π as follows:

1. Pick an index $i \in \{1, \ldots, m\}$ uniformly at random
2. Resample its label according to

$$\mathbb{P}(z_i = c) = \frac{f(z_1, \ldots, z_{i-1}, c, z_{i+1}, \ldots, z_m)}{\sum_{j=1}^{k} f(z_1, \ldots, z_{j-1}, c, z_{j+1}, \ldots, z_m)}$$

Basically, we try to identify the probabiliy that a vertex's label should be z_k given its neighbors have the labels they do.