ENGG7302: Advanced Computational Techniques in Engineering

Lecture 8: Markov Chain Monte Carlo

Mandar Gujrathi

School of ITEE
UQ

April 19, 2011

In this lecture..

- Monte Carlo Principle
- Uniform Sampling
- Importance Sampling
- Rejection Sampling
- Metropolis Hastings Algorithm

Monte Carlo

- We recently generated some numbers for our experiment in Matlab?

Monte Carlo

- We recently generated some numbers for our experiment in Matlab?
- Generates random numbers from a Normal distribution $\mathcal{N}(0,1)$
- randn, rand... functions in Matlab generate random numbers.

Monte Carlo Principle

- From a probability distribution function $P(\mathbf{x})$
- Draw i.i.d samples $\{\mathbf{x}^{(r)}\}_{r=1}^{R}$ from a given probability distribution function $P(\mathbf{x})$ defined on a high dimensional space
- Estimate the expectations of functions under these distributions, like mean and variance.

$$\Phi = E[\phi(x)] = \int \phi(\mathbf{x})P(\mathbf{x})d\mathbf{x}$$

Sampling

- Law of large numbers

$$\hat{\Phi} = \frac{1}{R}\sum_{r} \phi(\mathbf{x}^{(r)})$$

- As the number of samples increases the variance will decrease and the $E[\hat{\Phi}]$ will be equal to $E[\Phi]$

Why Sampling is hard

- We will assume that the density from which we wish to draw samples, P(x), can be evaluated, at least to within a multiplicative constant; that is, we can evaluate a function $P^*(x)$ such that

$$P(x) = \frac{P^*(x)}{Z}$$

- Hard to estimate the normalisation factor Z
- Hard to identify the correct places for sampling.
    - Might need to visit each and every point in 1 dimension.
- Process can be expensive for high dimensions.

Uniform Sampling

- From the law of large numbers if we make no. of samples (R) sufficiently large, we may be able to get an estimate of $P(\mathbf{x})$
- But how large should R be?
- In some cases it is estimated that R needs to be as large as 10^{150}!

Importance Sampling

- Target density $P(\mathbf{x})$
- Assume $P(\mathbf{x}) = \frac{P^*(\mathbf{x})}{Z}$
- Since $P(x)$ is too complicated to sample, assume we have a simpler density $Q(x)$ from which we can generate samples and evaluate to within a multiplicative constant Z_Q such that

$$Q(\mathbf{x}) = \frac{Q^*(\mathbf{x})}{Z_Q}$$

Importance Sampling

- Generate R samples from $Q(x)$
- Mapping them to $P(x)$ may appear some samples over represented and some under
- To avoid this, we introduce a weighting factor.

$$w_r = \frac{P^*(x^{(r)})}{Q^*(x^{(r)})}$$

- Therefore these weights need to be introduced when calculating the expectations of random variable.

Disadvantages

- For large dimensions, what would be the typical range of weights?
- Estimated that

$$\frac{w_r^{max}}{w_r^{med}} = \exp(\sqrt{2N})$$

- Hence for 1000 dimensions, largest weight is 10^{19} times greater than median weight.
- Some few large weights can be dominant and cause a problem while estimating the expectations.

Rejection Sampling

- Target density $P(x)$, Assume

$$P(x) = \frac{P^*(x)}{Z}$$

- Since P(x) would be too complicated assume simpler density $Q(x)$ and

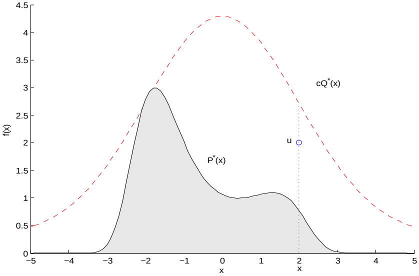
$$Q(x) = \frac{Q^*(x)}{Z_q}$$

- Lets also assume c is known such that

$$cQ^*(x) > P^*(x)$$

Rejection Sampling

- For a random no generated from $Q(x)$, evaluate $cQ^*(x)$ and generate a uniformly distributed variable $u \in [0, cQ^*(x)]$
- If $u > P^*(x)$ discard x else accept as a random variable.



Disadvantages: Rejection Sampling

- Works best if densities match i.e., $Q(x)$ is close to $P(x)$
- Else c has to be large.
- Now Acceptance rate (A),

$$A = \frac{\text{Volume under P(x)}}{\text{Volume under cQ(x)}}$$

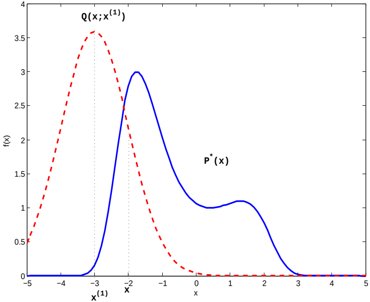
- If P and Q are both normalised

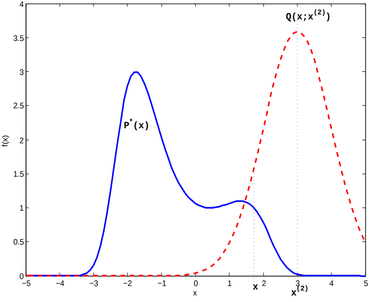
$$A \propto \frac{1}{c}$$

- Therefore as c increases, A decreases

Metropolis Hastings algorithm

- Makes use of a proposal density $Q(x)$ that depends on current state $x^{(t)}$
- $Q(x'; x^{(t)})$ centered around $x^{(t)}$ from which we can draw samples, $x^{'}$.





Metropolis Hastings algorithm

- Compute the likelihood ratio between the proposed sample and previous sample

$$a_1 = \frac{P^*(x')}{P^*(x^{(t)})}$$

- Calculate the ratio of proposal density in 2 directions

$$a_2 = \frac{Q(x^{(t)}; x')}{Q(x'; x^{(t)})}$$

- Get $a = a_1 a_2$

Metropolis Hastings algorithm

- If $a \geq 1$, the new state is accepted and set $x^{(t+1)} = x'$
- Else the new state is accepted with probability a and $x^{(t+1)} = x^{(t)}$
- Like rejection sampling, here rejected state is not discarded.
- If a Markov chain is started from a random initial value x_0 and after long iterations this initial state is forgotten. These samples, which are discarded, are known as burn-in.

MH algorithm

- The sampling methods can generate independent samples from a desired distribution, not related to each other.
- Metropolis method is an example of Markov Chain Monte Carlo
- In MH method, a sequence of states is generated each sample $x^{(t)}$ having a probability distribution which depends on the previous value, $x^{(t-1)}$
- It can be shown that as $t \to \infty$, for any positive Q the probability distribution of $x^{(t)}$ tends to $P(x)$. It may not mean that every point is selected.