Monte Carlo Methods Review

Lizi Chen

1 Monte Carlo Integration

Motivation: When we know the function, we can do numerical integration easily:

$$I = \int_a^b h(x)dx$$

But many time, we don't know the function. Thus, we need ways to approximate I. Monte Carlo methods are one of many ways to do this. (Other examples: Uniform Sampling, Importance Sampling, Sequential Monte Carlo a.k.a particle filter, and Mean Field Particle Methods.)

Law of Large Numbers (LLN) describes what happens when performing the <u>same</u> experiment many times:

Given an i.i.d. sequence of random variables $Y_1, Y_2, \cdots, Y_n$ with $\hat{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$ and $E(Y) = \mu$. Then for every $\epsilon > 0$,

$$P(|\hat{Y} - \mu| > \epsilon) \to 0, \text{ as } n \to \infty$$

Clarification: The **Law of Large Number** states that when sample size tends to infinity, the sample mean equals to population mean. The **Central limit Theorem** states that when sample size tends to infinity, the sample mean will be normally distributed.

For **Monte Carlo simulation**, this means that we can learn properties of a random variable (mean, variance, etc.) simply by simulating it over many trials.

Therefore, we can use Monte Carlo methods to approximate expectations by invoking LLN:

$$\begin{aligned} I &= \int_a^b h(x)dx \\ &= \int_a^b h(x) \cdot (b-a) \cdot \frac{1}{b-a} d(x) \\ \text{Let } f(x) = \frac{1}{b-a}&, \text{ and } w(x) = h(x) \cdot (b-a) \\ I &= \int_a^b w(x) \cdot f(x)dx \end{aligned}$$

, where$f(x)$ is the pdf of a $\mathcal{U}(a,b)$ random variable.

By LLN, if we randomly select N samples from $\mathcal{U}(a,b)$, we can estimate I as:

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} w(x_i) \tag{1}$$

, where

$$\frac{1}{N}\sum_{i=1}^{N} w(x_i) \text{ approximates to the expected value } \rightarrow \mathbb{E}(w(X)) = I$$

Example: In order to find the 'area' (see Figure.1) beneath the curve f within range $[a, b]$, we use Monte Carlo integration method.

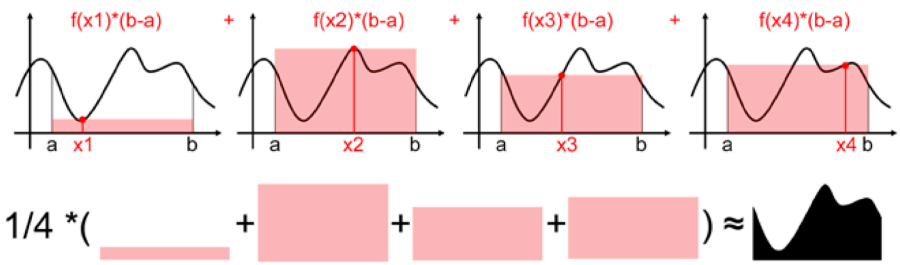

Figure 1: Monte Carlo Integration Example

As shown in Figure.1, we did four random samplings: x_1, x_2, x_3, x_4, each has corresponding $f(x_i)$ value: $f(x_1), f(x_2), f(x_3), f(x_4)$. The area of each is $A_i = f(x_i) * (b - 1)$. Thus, the expectation of the area is:

$$\mathbb{E}(Area) = \frac{1}{4}\Big(\sum_{i=1}^{4} f(x_i) * (b - a)\Big) \tag{2}$$

Compare Eq.2 with Eq.1, we can re-write Eq.2 to:

$$\mathbb{E}(Area) = \frac{1}{4}\Big(\sum_{i=1}^{4} \frac{f(x_i)}{\frac{1}{b-a}}\Big), \text{ where } pdf(x_i) = \frac{1}{b - a}$$

2 Sampling Methods

Two categories of sampling methods: independent sampling and dependent sampling.

- Independent Sampling: Rejection Sampling, Inverse Transform Sampling, Importance Sampling, etc.
- Dependent Sampling: Markov Chain Monte Carlo (MCMC), Sequential Monte Carlo (SMC), etc.

2.1 Inverse Transform Sampling

To generate sample numbers at random from any probability distribution, given its (assumed) cumulative distribution function.

Steps:

1. Assume our samples are from a distribution $p(x)$, of which the CDF must be **strictly increasing**. (Note that in many real-life situation, we will not be able to know the pdf/pmf $p(x)$)
2. Integrate $p(x)$ to get its CDF $F_X(x)$.
3. Now, we let $Y = F_X(x)$, meaning that each Y points to an unique x, and since CDF ranges from 0 to 1, $Y \in [0, 1]$

4. Inverse $Y = F_X(x)$ to get $x = F_X^{-1}(Y)$.
5. Now if we can select a Y value from 0 to 1 each with the same probability, we can sample $F_X^{-1}(Y)$ value with uniformly, which is the value of x.
6. Thus, for each sampling iteration, we sample a value y_i from Uniform Distribution $Y \sim \mathcal{U}(0,1)$.
7. After n sampling iterations, we will have a list of uniformly distributed Y values: $[y_1, y_2, \cdots, y_n]$.
8. Substitute the Y values into $x = F_X^{-1}(Y)$, we will have a list of X values: $[x_1, x_2, \cdots, x_n]$, and these values will follow our previously mentioned $p(x)$ distribution.

2.2 Rejection Sampling

Problem:

- Wasteful to generate independent samples.
- Efficiency drops dramatically as dimension of the data increases.

2.3 Importance Sampling

To estimate properties of a particular distribution $h(x)$, while only having samples generated from a different distribution $g(x)$ than the distribution of interest.

Goal: $I = \int h(x)f(x)dx$, where f is some function and f is the probability density function of X that is difficult to sample from.

Now, specify a different pdf g as proposal distribution. We can have:

$$\begin{aligned} I &= \mathbb{E}_{\mathbf{f}}(h(X)) \\ &= \int h(x)\mathbf{f}(\mathbf{x})dx \\ &= \int h(x)\frac{\mathbf{f}(\mathbf{x})}{g(x)}d(x)dx \\ &= \int \frac{h(x)\cdot f(x)}{g(x)}g(x)dx \end{aligned}$$

Thus, we convert the sampling base on f to g:

$$I = \mathbb{E}_{\mathbf{f}}(h(X)) = \int \frac{h(x)\cdot f(x)}{g(x)}g(x)dx = \mathbb{E}_{\mathbf{g}}\Big[\frac{h(x)\cdot f(x)}{g(x)}\Big]$$

Hence, given an i.i.d. sample sequence $X_1, X_2, \cdots, X_n$ from g, our estimator of I becomes:

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N}\frac{h(x_i)\cdot f(x_i)}{g(x_i)}, \text{ which approximates to the actual } I = \mathbb{E}_{\mathbf{g}}\Big[\frac{h(x)\cdot f(x)}{g(x)}\Big]$$

Note: the choice of distribution $g(x)$ matters a lot.

The standard error of

$$SE(\hat{I}) \propto \mathbb{E}_{\mathbf{g}}\Big[\Big(\frac{h(x)\cdot f(x)}{g(x)}\Big)^2\Big]$$

, we want to have small $\frac{f}{g}$ rate. Thus, we want to select g that has similar ‘shape’ to f.

2.4 Reject-Accept Sampling

Take example from prob&stat course

todo

Another example: Gibbs Sampling (does not involve accept-reject)

2.5 MCMC Sampling

MCMC is a dependent sampling method (Video Example: Effective sample size: representing the cost of dependent sampling, Ben Lambert), meaning that the next value of the sampler depends on the current sampling value.

Dependent sampling is much less efficient than independent sampling (in low dimensional data). Note that independent sampling is used in Monte Carlo Integration.

verify

guideline: `https://zhuanlan.zhihu.com/p/24915332`

remove

2.5.1 Dependent Sampling: Random Walk Metropolis Algorithm (First order MCMC)

When using independent sampling methods in high dimensional data, the 'rejectiong' rate is $\to 100\%$.

2.5.2 Metropolis-Hasting Algorithm

2.5.3 Gibbs Sampling

2.6 Slice Sampling

3 Monte Carlo on Reinforcement Learning

强化学习: 蒙特卡洛方法介绍 https://zhuanlan.zhihu.com/p/37658363

4 More:

Monte Carlo theory, methods and examples, Art B. Owen, Stanford University, https://statweb.stanford.edu/~owen/mc/ Chapters:

- Variance reduction
- Importance sampling
- Advanced variance reduction
- Markov chain Monte Carlo
- Gibbs sampler
- Adaptive and accelerated MCMC