Monte Carlo Methods

Advanced Statistical Inference

Simone Rossi

Introduction

We need to compute

$$\mathbb{E}_{p(\boldsymbol{x})}[f(\boldsymbol{x})] = \int f(\boldsymbol{x}) p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$$

where $p(\boldsymbol{x})$ is the probability density function of $\boldsymbol{x}$.

Monte Carlo methods are a class of computational algorithms that rely on repeated random sampling to obtain numerical results.

$$\mathbb{E}_{p(\boldsymbol{x})}[f(\boldsymbol{x})] \approx \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{x}_i)$$

where $\boldsymbol{x}_i \sim p(\boldsymbol{x})$.

Compute π with Monte Carlo

Area of a circle of radius r is πr^2 but also

$$I = \int_{-r}^{r} \int_{-r}^{r} \mathbb{I}(x^2 + y^2 \leq r^2) \mathrm{d}x \mathrm{d}y$$

where $\mathbb{I}$ is 1 if the condition is true and 0 otherwise. Then, $\pi = I/r^2$.

With Monte-Carlo

$$\begin{aligned} I &= (2r)^2 \int \int f(x, y) p(x) p(y) \mathrm{d}x \mathrm{d}y \\ &\approx (2r)^2 \frac{1}{N} \sum_{i=1}^{N} f(x_i, y_i) \end{aligned}$$

where $f(x, y) = \mathbb{I}(x^2 + y^2 \leq r^2)$ and $p(x) = p(y) = \mathcal{U}(-r, r)$.

Theoretical properties of Monte Carlo methods

- **Consistency**: As the number of samples N goes to infinity, the estimate converges to the true value.
- **Central Limit Theorem**: The estimate is normally distributed around the true value.

$$\left(\frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}_i) - \mathbb{E}_{p(\boldsymbol{x})}[f(\boldsymbol{x})]\right) \sim \mathcal{N}\left(0, \frac{\sigma^2}{N}\right)$$

where σ^2 is the variance of $f(\boldsymbol{x})$.

Note: This is independent of the dimensionality of $\boldsymbol{x}$.

! Important

How do we sample from $p(\boldsymbol{x})$ when $p(\boldsymbol{x})$ is not known? For example, when it's a posterior distribution $p(\boldsymbol{\theta} \mid \boldsymbol{y})$.

Sampling from simple distributions

Sampling from a Gaussian distribution

Univariate case

Suppose we want to sample from a (univariate) Gaussian distribution $p(x) = \mathcal{N}(x \mid \mu, \sigma^2)$.

Each sample can be obtained by

$$x_i = \mu + \sigma z_i$$

where $z_i \sim \mathcal{N}(0, 1)$.

Multivariate case

Suppose we want to sample from a multivariate Gaussian distribution $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

1. Compute the Cholesky decomposition of $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$.
2. Sample $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$.
3. Compute $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{L}z$.

Note: The Cholesky decomposition is only defined for positive-definite matrices. If $\boldsymbol{\Sigma}$ is not positive-definite, you will not be able to sample from the distribution.

Rejection sampling

Suppose we want to sample from a distribution $p(x)$ that we only know up to a normalizing constant $p(x) = \frac{p^*(x)}{Z}$, where $Z = \int p^*(x)\mathrm{d}x$.

In **rejection sampling**, we start from a distribution $q(x)$ such that $Cq(x) \geq p^*(x)$ for all x, with $C > 0$.

1. Sample $x \sim q(x)$.
2. Sample $u \sim \mathcal{U}(0, Cq(x))$.
3. If $u \leq p^*(x)$, accept x; otherwise, reject it and go back to step 1.

Rejection sampling algorithm

It's possible to prove that

$$\mathbb{P}(\text{accept}) \propto \frac{1}{C}$$

- We want to have high acceptance rate.
- We want to have C as small as possible.
- ... but large enough to satisfy $Cq(x) \geq p^*(x)$.

❗ Important

Note: In high dimensions (> 100), rejection sampling is not efficient. The volume of the region where $p^*(x)$ is large becomes very small compared to the volume of the region where $q(x)$ is large.
We need something more efficient.