Week 8: Monte Carlo (MC)

MATH-517 Statistical Computation and Visualization

Linda Mhalla

2023-11-10

Introduction

MC $\equiv$ repeated random sampling to mimic outcome of random process and produce numerical results such as

- generating draws from complicated distributions and/or domains
- integration
  - calculation of moments or confidence intervals
  - high-dimensional densities in Bayesian settings
- optimization
  - mode evaluation

Basic idea: If we can sample from a process or mimic its outcomes, we can learn a lot about it by doing statistics on the simulated samples (as opposed to analyzing the process itself)

MC methods $\equiv$ simulation-based statistical techniques/inference

Introduction

Gambling experiments have random outcomes – hence "Monte Carlo"

Method initially developed by Stanislaw Ulam and John von Neumann for the Manhattan Project (to estimate integrals)

Example

$(X_1, Y_1)^\top, \dots, (X_N, Y_N)^\top$ a sample from the standardized bivariate Gaussian distribution

$$\mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right)$$

- We want to test $H_0 : \rho = \rho_0$ against $H_1 : \rho \neq \rho_0$
- Statistic: $\hat{\rho} = N^{-1} \sum_{n=1}^{N} X_n Y_n$

Since the data generation process is fully determined under H_0, we can simulate data to approximate the sampling distribution and thus also the p-value

To test a hypothesis, we only need to simulate data under H_0

But how to draw samples from a specific distribution?

Section 1

Random Number Generation (RNG)

RNG

True randomness is hard to come by. Historically:

- dice, cards, coins
- physical processes
- census data, tables, etc.

Practical reasons not to use "truly" random numbers: debugging and reproducibility

John von Neumann: **pseudo-RNG**

- approximates the desired dist. for $N \to \infty$
- cannot be predicted
- pass a set of independence tests
- repeatability ($\Rightarrow$ reproducibility)
- long cycle (before it starts repeating) and fast sampling

Uniformity and independence tests needed to assess quality of pseudo-RNG

Cornerstone: Generating from $\mathcal{U}[0,1]$

Assume now we can generate numbers from the $\mathcal{U}[0,1]$ distribution

- e.g. the *linear congruential method*

$$X_n = (aX_{n-1} + c) \bmod m, \quad n = 1, 2, \dots,$$

  where a, c, m, and X_0 are cleverly chosen to fulfill the pseudo-RNG requirements, i.e., maximize period, speed, and "randomness"
  - X_0 is the seed
  - produces integers between 0 and $m - 1$
  - $U_n = X_n/m \dot{\sim} U(0,1)$

Bad example: $m = 2^{31}$, $a = 2^{16} + 3$, and $c = 0 \Rightarrow$ IBM's RANDU

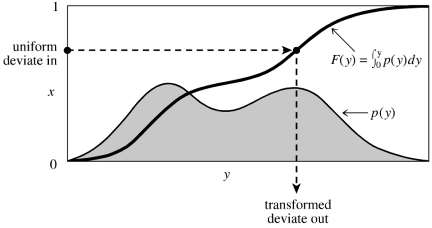
- now, better and much more complicated algorithms are available
  - every piece of software has its favorite pseudo-RNG
  - out of the scope of the course (see, e.g., shift-register generators or Wichmann–Hill generator)

Question: How do we generate from other distributions?

Transforms

Lemma. (*Inverse Transform.*)
Let $U \sim \mathcal{U}(0,1)$ and F be a distribution function and F^{-1} the quantile (or generalized inverse) function. Then $X = F^{-1}(U) \sim F$.

Proof: Simply
$P(X \leq x) = P\{F(X) \leq F(x)\} = P\{U \leq F(x)\} = F(x)$.



Transforms

The inverse transform method is general, but not almighty:

- distribution/quantile functions can be complicated/unknown
  - e.g. $\mathcal{N}(0,1)$

Often, simpler relationships can be used: diagram

- still, there is no arrow there between $\mathcal{U}(0,1)$ and $\mathcal{N}(0,1)$, generating $\mathcal{N}(0,1)$ is actually a bit tricky...

Transforms

Lemma. (*Box-Muler transform.*)
Let $U_1, U_2 \sim \mathcal{U}(0,1)$ be independent. Then

$$Z_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2) \quad \& \quad Z_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

are two independent standard Gaussian random variables

Again, software uses its favorite relationships

- e.g. `R` has tabulated F and F^{-1} for $\mathcal{N}(0,1)$ to a high precision and actually uses the inverse transform, because evaluating trigonometric functions is rather expensive (slow)
- `?rnorm` $\Rightarrow$ `rnorm`, `pnorm`, `qnorm`, `dnorm` help

Rejection Sampling

Setup: we know how to simulate from a *proposal* g, we want to simulate from a *target* f

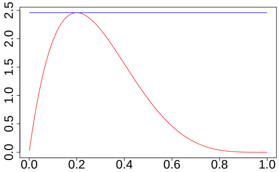
- let $supp(f) \subset supp(g)$, i.e.,$f(x) > 0 \Rightarrow g(x) > 0$
- let there be $c > 1$ such that $\forall x : \; f(x) \leq c\, g(x)$, i.e., $\sup_x \frac{f(x)}{g(x)} = c < \infty$

Algorithm: (to draw a single sample X from f)

1. Draw a proposal Y from g
2. Draw $U \sim \mathcal{U}(0, 1)$
3. If $U \leq \frac{1}{c} \frac{f(Y)}{g(Y)}$, accept $X = Y$ and stop, otherwise go back to 1

Example:

- $\mathcal{U}(0, 1)$ proposal
- $\mathcal{B}(2, 5)$ target
- $c \approx 2.5$



Rejection Sampling

Does the algorithm really sample from f?

$$P(X \leq x) = P\left\{ Y \leq x \,\middle|\, U \leq \underbrace{\frac{1}{c}\frac{f(Y)}{g(Y)}}_{=:t(Y)} \right\} = \frac{P\{Y \leq x \wedge U \leq t(Y)\}}{P\{U \leq t(Y)\}}$$

$$= \frac{\int_{-\infty}^{x} \int_{0}^{t(y)} du\, g(y) dy}{\int_{-\infty}^{+\infty} \int_{0}^{t(y)} du\, g(y) dy} = \frac{\int_{-\infty}^{x} t(y) g(y) dy}{\int_{-\infty}^{+\infty} t(y) g(y) dy} = \frac{\int_{-\infty}^{x} \frac{1}{c} f(y) dy}{\int_{-\infty}^{+\infty} \frac{1}{c} f(y) dy}$$

$$= \frac{\frac{1}{c} F(x)}{\frac{1}{c}} = F(x)$$

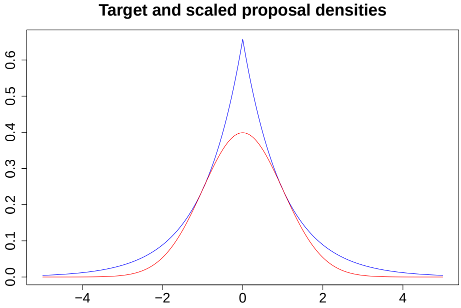
The rejection sampling algorithm above is again quite general, but it needs

- a good proposal g
  - with a similar shape than the target density, leading to
  - high acceptance probability $P\{U \leq t(Y)\} = 1/c$
- fast evaluation of f and g

Example: $\mathcal{N}(0,1)$ again

Goal: Simulate data from the standard Gaussian target using the doubly exponential proposal, i.e.,

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad \& \quad g(x) = \frac{\alpha}{2} e^{-\alpha|x|}, \text{ where } \alpha > 0, \quad x \in \mathbb{R}$$



Another way of obtaining $\mathcal{N}(0,1)$ from $\mathcal{U}(0,1)$:

$$\mathcal{U}(0,1) \longrightarrow Exp(1)$$
$$Exp(1) \longrightarrow DbExp(1)$$
$$DbExp(1) \longrightarrow N(0,1)$$

Note: $\alpha = 1$ minimizes the value of $c = \sup f(x)/g(x)$ and hence maximizes the acceptance probability

Section 2

Numerical Integration

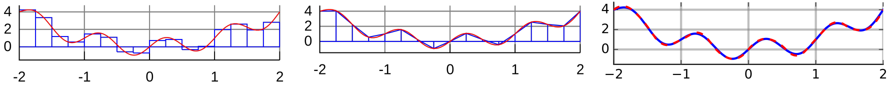
Deterministic Approaches

Goal: approximate $J = \int_a^b f(x)dx$

Quadrature method: evaluate the function on a grid

$$S_K = \{(b-a)/K\} \sum_{k=1}^{K} f(t_k)$$

- if f is nice (smooth), $S_K \to J$ for $K \to \infty$.
- corresponds to integrating a local constant interpolation of f
- local linear interpolation (*trapezoidal rule*) or local quadratic (*Simpson's rule*) are also well known



Naive Monte Carlo

We consider the more general integral

$$J = \int_{\mathcal{X}} m(x) f(x) dx = \mathbb{E}_f\{m(X)\} \quad \text{for} \quad X \sim f$$

$\Rightarrow$ generate $X_1, \ldots, X_N \overset{i.i.d.}{\sim} f$ and approximate J by

$$\widehat{\bar{J}}_N = N^{-1} \sum_{n=1}^{N} m(X_n)$$

- unbiased and we get consistency due to SLLN
- monitoring convergence via CLT-based (approx.) confidence intervals:

$$\sqrt{N} \frac{\widehat{\bar{J}}_N - J}{v_N} \dot{\sim} \mathcal{N}(0,1), \quad \text{where} \quad v_N = \frac{1}{N-1} \sum_{n=1}^{N} \{m(X_n) - \widehat{\bar{J}}_N\}^2$$

- beware of **rare events**: if f has heavy tails, $\widehat{\bar{J}}_N$ can be a bad estimate and we need huge N to get small v_N

Importance Sampling

We often can not simulate directly from f and we require sophisticated approches. Rewrite

$$J := \int_{\mathcal{X}} m(x)f(x)dx = \int_{\mathcal{X}} m(x)\frac{f(x)}{g(x)}g(x)dx = \int_{\mathcal{X}} m(x)w(x)g(x)dx$$

with g a density whose support contains that of f and $w(x) \geq 0$ the **importance weighting function**

Thus, by sampling $X_1, \ldots, X_N \overset{\text{i.i.d.}}{\sim} g$, we can approximate J by

$$\widehat{J}_N := N^{-1} \sum_{n=1}^{N} m(X_n)w(X_n)$$

Idea:

- Use a simpler proposal distribution g from which we can generate
- Candidates generated from g fall within the domain of f
- Reweight the observations generated from it when taking the mean

Importance Sampling: Intuitive Explanation

Key: integrating f amounts to integrating f/g under sampling from g

$$J = \int_{\mathcal{X}} m(x) f(x) dx = \mathbb{E}_g\{m(X)w(X)\}$$

- when f is flat (all regions are equally important), use either the naive MC (with uniform sample) or deterministic approaches that need only small samples
- when f is not flat, using a "good" g allows us to encode which regions are important $\Rightarrow$ "importance sampling" (vs rejection sampling)

Of course, it is not always easy to find a "good" g which

- has a similar shape than f and
- from which we can easily sample

As we will see, when $\mathcal{X} = \mathbb{R}$, it is important to match the decay of the tails between the target and reference measures

Importance Sampling: Properties

- unbiased and the variance is given by

$$\begin{aligned}\mathrm{var}(\widehat{J}_N) &= \frac{1}{N}\left\{\int_{\mathcal{X}} m^2(x)\frac{f(x)}{g(x)}f(x)dx - J^2\right\}\\ &= N^{-1}\int_{\mathcal{X}}\left\{m(x)\frac{f(x)}{g(x)} - J\right\}m(x)f(x)dx,\end{aligned}$$

which is small if $g(x) = m(x)f(x)J^{-1}$ or $g(x) \propto m(x)f(x)$

$\Rightarrow$ good choices of g can yield huge improvements in efficiency

- approx. confidence intervals obtained by CLT

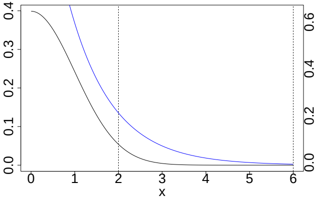
$$\sqrt{N}\frac{\widehat{J}_N - J}{v_N} \dot{\sim} \mathcal{N}(0,1),$$

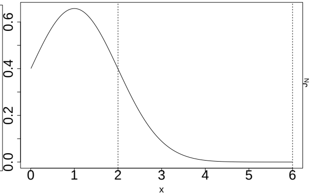
where $v_N = \frac{1}{N-1}\sum_{n=1}^{N}\{m(X_n)w(X_n) - \widehat{J}_N\}^2$

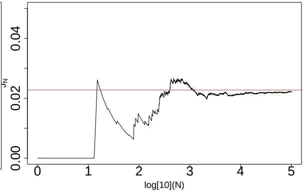
Examples

Task: Approximately calculate $P(2 < X < 6)$ for the target distribution $X \sim f$ using a reference g

Gaussian target, Exponential reference - densities (left), their ratio (middle), importance sampling error (right)



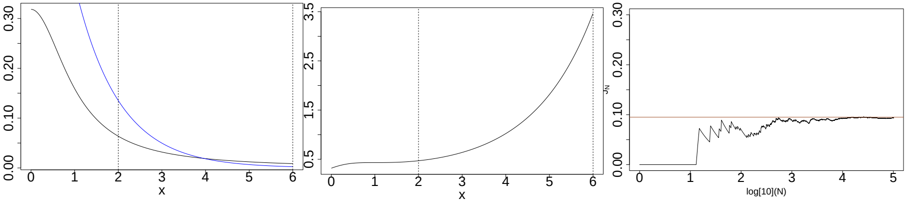




Examples

Approximately calculate $P(2 < X < 6)$ for the target distribution $X \sim f$ using a reference g

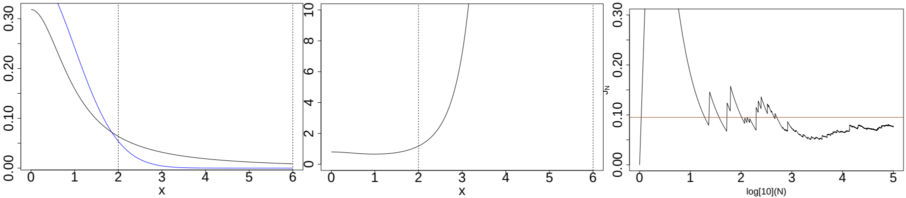
Cauchy target, Exponential reference - densities (left), their ratio (middle), importance sampling error (right)



Examples

Approximately calculate $P(2 < X < 6)$ for the target distribution $X \sim f$ using a reference g

Cauchy target, Gaussian reference - densities (left), their ratio (middle), importance sampling error (right)



- the tails of Cauchy and Gaussian distributions are too different $\Rightarrow$ importance sampling performs poorly
- if we can simulate from Gaussian, we can simulate directly from Cauchy: $Z_1, Z_2 \sim \mathcal{N}(0,1)$ independent $\Rightarrow Z_1/Z_2 \sim Cauchy(0,1)$

Variance Reduction

Accuracy of MC integration is assessed by the estimator's efficiency/variance (assuming efforts of simulation are similar)

There are ways to tweak the sampling scheme in order to reduce the variance

- importance sampling (we have seen above)
- antithetic variables (to follow)
- stratified sampling (to follow)
- quasi-random sampling and control variates (see the supplementary notes)
- many other techniques: latin hypercube sampling, ratio estimator, etc

Remark: When comparing several different estimators via simulations, the same simulated datasets should be used for all the estimators

Variance Reduction: Antithetic Variables

The method attempts to reduce variance by introducing negative dependence between pairs of replications

Given two i.i.d. samples $X_1, \dots, X_N \sim f$ and $Y_1, \dots, Y_N \sim f$, consider the estimator

$$\tilde{J}_N = \frac{1}{2N} \sum_{n=1}^{N} \{m(X_n) + m(Y_n)\} = \frac{1}{2}(\widehat{\bar{J}}_N^X + \widehat{\bar{J}}_N^Y)$$

Then,

$$\mathrm{var}(\tilde{J}_N) = \frac{1}{2}\mathrm{var}(\widehat{\bar{J}}_N)\{1 + \mathrm{corr}(\hat{\bar{\mathrm{J}}}_{\mathrm{N}}^{\mathrm{X}}, \hat{\bar{\mathrm{J}}}_{\mathrm{N}}^{\mathrm{Y}})\}$$

$\Rightarrow \tilde{J}_N$ is more efficient than the naive MC (with sample of size $2N$) if $m(X_n)$ and $m(Y_n)$ are **negatively correlated**

Basic result: if $g(u)$ is monotonic on $0 < u < 1$, then

$$\mathrm{corr}\{g(U), g(1-U)\} < 0$$

Hence $F^{-1}(U)$ and $F^{-1}(1-U)$ are negatively correlated variables with distribution F

Variance Reduction: Stratified Sampling

- Break sampling space into strata and sample appropriate number of observations in each
- Compute the naive MC estimator in each stratum and sum over all strata
- Method relies on conditional variance:

$$\mathrm{var}\{m(X)\} = \mathbb{E}[\mathrm{var}\{m(X)|I\}] + \mathrm{var}[E\{m(X) \mid I\}]$$

Thus,

$$\mathrm{var}\{m(X)\} \geq \mathbb{E}[\mathrm{var}\{m(X)|I\}] = \sum_{i=1}^{K} p_i \sigma_i^2$$

- Variance reduction substantial if I accounts for a large fraction of the variance of $m(X)$

References

Donald Knuth (1997, 3rd ed.) *The Art of Computer Programming*, vol. 2

Robert & Casella (2010) *Introducing Monte Carlo methods with R*

Feedback for the mini-project

- Good points
  - original datasets
  - going the extra mile to dig deeper in the data like proposing new distance metrics or creating new variables
  - nice introductions and good referencing
- Points to improve
  - captions missing!!!
  - code appearing in the text
  - bad sectioning of the text
  - missing introduction and/or conclusion
  - bad citations or missing references
  - remember to log-transform when needed ...

Assignment [5 %]

Go to Assignment 6 for details.