Monte Carlo Methods: Lecture 1 : Introduction

Nick Whiteley 2011

Course material originally by Adam Johansen and Ludger Evers 2007

Timetable

- 3 Hours each week: either 3 lectures (weeks 7,9,11) or 2 lectures + 1 computer practical (weeks 8,10,12)
- See the course website `http://www.maths.bris.ac.uk/~manpw/teaching/mcm` for teaching material to download, etc.

Unit assessment

Overall assessment

- 20% Coursework
- 80% Standard 1 1/2 hour examination

Assessment of the course work

5 problem sheets in total (2 mandatory questions each + optional ones)

- 3 on theory: T1 (week 8), T2 (week 10), and T3 (week 12)
- 2 on computer practicals: P1 (week 9), P2 (week 11)

Coursework mark based on the best *four* problem sheets.

1.1 & 1.3 Introduction

What is Monte Carlo?

What are Monte Carlo Methods?

One of many definitions

A Monte Carlo method consists of

- "representing the solution of a problem as a parameter of a hypothetical population, and
- using a random sequence of numbers to construct a sample of the population, from which statistical estimates of the parameter can be obtained."

(Halton, 1970)

Sometimes referred to as *stochastic simulation*.

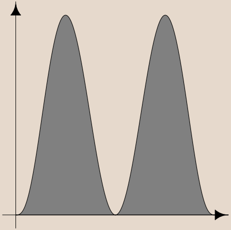
Examples of applications of Monte Carlo methods (1)

Numerical Integration

Objective is to estimate an integral

$$\int_{\mathcal{X}} f(\mathbf{x})\, d\mathbf{x},$$

which is analytically intractable.



Examples of applications of Monte Carlo methods (2a)

Bayesian statistics

- Data $\mathbf{y}_1, \ldots, \mathbf{y}_n$ and model $f(\mathbf{y}_i|\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is some parameter of interest.

  $\rightsquigarrow$ Likelihood $l(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta}) = \prod_{i=1}^{n} f(\mathbf{y}_i|\boldsymbol{\theta})$
- Frequentist estimate of $\boldsymbol{\theta}$ is the maximiser of $l(\mathbf{y}_1, \ldots, \mathbf{y}_n)$ ("maximum likelihood estimate").
- In the frequentist framework $\boldsymbol{\theta}$ is a parameter, not a random variable.

Examples of applications of Monte Carlo methods (2b)

Bayesian statistics (continued)

- In the Bayesian framework $\boldsymbol{\theta}$ is a random variable with prior distribution $f^{\text{prior}}(\boldsymbol{\theta})$. After observing $\mathbf{y}_1, \ldots, \mathbf{y}_n$ the posterior density of f is

$$
\begin{aligned}
f^{\text{post}}(\boldsymbol{\theta}) &= f(\boldsymbol{\theta}|\mathbf{y}_1, \ldots, \mathbf{y}_n) \\
&= \frac{f^{\text{prior}}(\boldsymbol{\theta}) l(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta})}{\int_\Theta f^{\text{prior}}(\boldsymbol{\vartheta}) l(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\vartheta})\, d\boldsymbol{\vartheta}} \\
&\propto f^{\text{prior}}(\boldsymbol{\theta}) l(\mathbf{y}_1, \ldots, \mathbf{y}_n|\boldsymbol{\theta})
\end{aligned}
$$

- For many complex models the integral in the denominator is hard to compute
  $\rightsquigarrow$ use of a Monte Carlo approximation

What you will learn in this lecture course

- Basic concepts: transformation, rejection, and reweighting.
- A brief reminder of important properties of Markov chains.
- Markov Chain Monte Carlo (MCMC) methods: Gibbs sampling and Metropolis-Hastings.
- Sequential Monte Carlo (SMC).

History of Monte Carlo methods

1733 Buffon's needle problem.

1812 Laplace suggests using Buffon's needle experiment to estimate π.

1946 ENIAC (Electronic Numerical Integrator And Computer) built.

1947 John von Neuman and Stanisław Ulam propose a computer simulation to solve the problem of neutron diffusion in fissionable material.

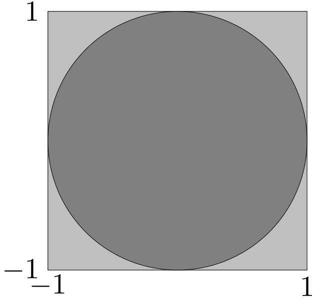
1949 Metropolis and Ulam publish their results in the *Journal of the American Statistical Association*.

1984 Geman & Geman publish their paper on the Gibbs sampler
From then onwards: continuously growing interest of statisticians in Monte Carlo methods.

1.2 Introductory examples

Example 1.1: Raindrop experiment for computing π (1)

- Consider "uniform rain" on the square $[-1,1] \times [-1,1]$, i.e. the two coordinates $X, Y \overset{\text{i.i.d.}}{\sim} \mathsf{U}[-1,1]$.
- Probability that a rain drop falls into the dark circle is



$$\mathbb{P}(\text{drop within circle}) = \frac{\text{area of the unit circle}}{\text{area of the square}} = \frac{\displaystyle\iint_{\{x^2+y^2\leq 1\}} 1 \, dxdy}{\displaystyle\iint_{\{-1\leq x,y\leq 1\}} 1 \, dxdy} = \frac{\pi}{2\cdot 2} = \frac{\pi}{4}.$$

Example 1.1: Raindrop experiment for computing π (2)

- If we know π, we can compute $\mathbb{P}(\text{drop within circle}) = \frac{\pi}{4}$.
- Consider n independent raindrops, then the number of rain drops Z_n falling in the dark circle is a binomial random variable:

$$Z_n \sim \mathsf{B}(n, \theta), \qquad \text{with } \theta := \mathbb{P}(\text{drop within circle}).$$

- We can estimate θ by

$$\hat{\theta}_n = \frac{Z_n}{n}.$$

- Thus we can estimate π by

$$\hat{\pi}_n = 4\hat{\theta}_n = 4 \cdot \frac{Z_n}{n}.$$

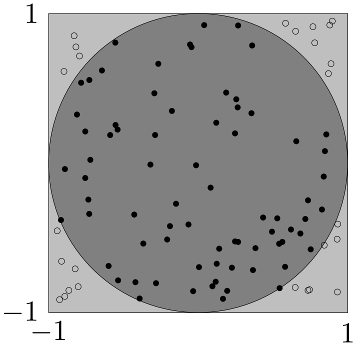
Example 1.1: Raindrop experiment for computing π (3)

- Result obtained for $n = 100$ raindrops: 77 points inside the dark circle.
- Resulting estimate of π is

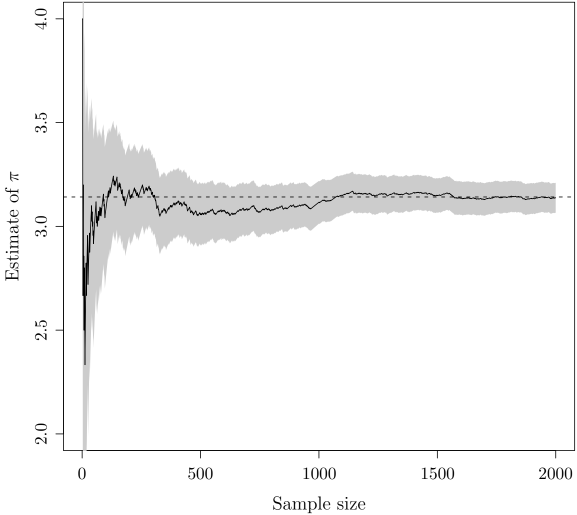
$$\hat{\pi} = \frac{4 \cdot Z_n}{n} = \frac{4 \cdot 77}{100} = 3.08,$$

(rather poor estimate)

- However: the *law or large numbers* guarantees that $\hat{\pi}_n = \frac{4 \cdot Z_n}{n} \to \pi$ almost surely for $n \to \infty$.



Example 1.1: Raindrop experiment for computing π (4)



Example 1.1: Raindrop experiment for computing π (5)

What can we say about the rate at which the sequence of estimates $\hat{\pi}_n$ converges to π? We can perform a simple calculation. Recall two things:

1. Chebyshev's inequality: For a real-valued random variable X, and any $\delta > 0$

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq \delta\right) \leq \frac{\mathbb{E}[(X - \mathbb{E}[X])^2]}{\delta^2}$$

2. The variance of the $\mathsf{B}(n, \theta)$ distribution is $n\theta(1-\theta)$

Example 1.1: Raindrop experiment for computing π (6)

Recall that $Z_n \sim \mathsf{B}(n, \theta)$, and $\hat{\theta}_n = \dfrac{Z_n}{n}$.

Then as $\mathbb{E}[\hat{\theta}_n] = \theta$, we have, for any $\delta > 0$

$$\mathbb{P}\left(\left|\hat{\theta}_n - \theta\right| > \delta\right) \leq \frac{\mathbb{E}[(\hat{\theta}_n - \theta)^2]}{\delta^2} = \frac{\mathbb{E}[(Z_n - n\theta)^2]}{n^2\delta^2} = \frac{\theta(1-\theta)}{n\delta^2},$$

and therefore, for any $\lambda > 0$,

$$\begin{aligned}\mathbb{P}\left(\left|\hat{\theta}_n - \theta\right| \leq \lambda\sqrt{\frac{\theta(1-\theta)}{n}}\right) &= 1 - \mathbb{P}\left(\left|\hat{\theta}_n - \theta\right| > \lambda\sqrt{\frac{\theta(1-\theta)}{n}}\right) \\ &\geq 1 - \frac{1}{\lambda^2}.\end{aligned}$$

Example 1.1: Raindrop experiment for computing π (7)

From the previous bound, if we take, for example, $\lambda = 3$, then with probability greater than 0.888 $(1-(1/3)^2 = 8/9 \approx 0.8889)$ the event

$$\left|\hat{\theta}_n - \theta\right| \leq 3\sqrt{\frac{\theta(1-\theta)}{n}}$$

occurs. As $\theta \in [0,1]$, then $\theta(1-\theta) \leq 1/4$ and thus

$$\mathbb{P}\left(\left|\hat{\theta}_n - \theta\right| \leq \frac{3}{2}\frac{1}{\sqrt{n}}\right)$$
$$= \mathbb{P}\left(\hat{\theta}_n - \frac{3}{2}\frac{1}{\sqrt{n}} \leq \theta \leq \hat{\theta}_n + \frac{3}{2}\frac{1}{\sqrt{n}}\right) > 0.888.$$

Recalling that $\pi = 4\theta$, we obtain a confidence interval:

$$\mathbb{P}\left(4\hat{\theta}_n - \frac{6}{\sqrt{n}} \leq \pi \leq 4\hat{\theta}_n + \frac{6}{\sqrt{n}}\right) > 0.888.$$

Example 1.1: Raindrop experiment for computing π (8)

Recall the two core steps used in the example:

1. We have written the quantity of interest (in our case π) as an expectation:

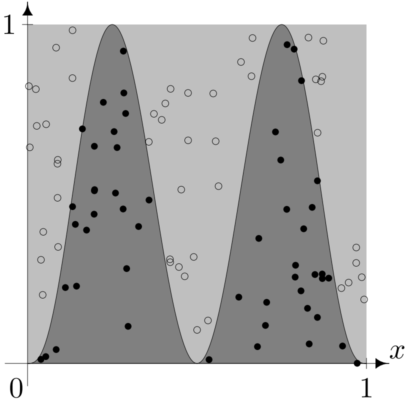
$$\pi = 4\mathbb{P}(\text{drop within circle}) = \mathbb{E}\left(4 \cdot \mathbb{I}_{\{\text{drop within circle}\}}\right)$$

2. We have replaced this algebraic representation of the quantity of interest by a sample approximation to it.
3. We will see this pattern throughout the course, in various situations and where we obtain the sample approximation by various means.

Generalisation to Monte Carlo Integration (cf. example 1.2)

$$
\begin{aligned}
&\int_0^1 f(x)\,dx \\
=\;& \int_0^1 \int_0^{f(x)} 1\;dt\;dx \\
=\;& \iint\limits_{\{(x,t):t\le f(x)\}} 1dt\;dx \\
=\;& \frac{\iint\limits_{\{(x,t):t\le f(x)\}} 1\;dt\;dx}{\iint\limits_{\{0\le x,t\le 1\}} 1dt\;dx}
\end{aligned}
$$

$f:\ [0,1] \to [0,1]$



Comparison of the speed of convergence

- Speed of convergence of Monte Carlo integration is $O_{\mathbb{P}}(n^{-1/2})$.
- Speed of convergence of numerical integration of a *one-dimensional* function by Riemann sums is $O(n^{-1})$.
- Does not compare favourably for one-dimensional problems.
- However:
  - Order of convergence of Monte Carlo integration is *independent* of the dimension.
  - Order of convergence of numerical integration techniqes like Riemann sums deteriorates with the dimension increasing.

$\rightsquigarrow$ Monte Carlo methods can be a good choice for high-dimensional integrals.

1.4 Pseudo-random numbers

First thoughts

- Philosophical paradox:
  - We need to reproduce randomness by a computer algorithm.
  - A computer algorithm is deterministic in nature.

$\rightsquigarrow$ "pseudo-random numbers"

- Pseudo-random number from $\mathrm{U}[0,1]$ will be our only "source of randomness".
- Other distributions can be derived from $\mathrm{U}[0,1]$ pseudo-random numbers using deterministic algorithms.

Characterisation of a pseudo-random number generator

- A pseudo-random number generator (RNG) should produce output for which the $\mathsf{U}[0,1]$ distribution is a suitable model.
- The pseudo-random numbers $X_1, X_2, \ldots$ should thus have the same *relevant* statistical properties as independent realisations of a $\mathsf{U}[0,1]$ random variable.
  - They should reproduce independence ("lack of predictability"): $X_1, \ldots, X_n$ should not contain any discernible information on the next value X_{n+1}.This property is often referred to as the lack of predictability.
  - The numbers generated should be spread out evenly across $[0,1]$.

A simple example

Algorithm 1.1: Congruential pseudo-random number generator

1. Choose $a, M \in \mathbb{N}$,$c \in \mathbb{N}_0$, and the initial value ("seed") $Z_0 \in \{1, \ldots M-1\}$.
2. For $i = 1, 2, \ldots$

   Set $Z_i = (aZ_{i-1} + c) \mod M$, and $X_i = Z_i/M$.

$Z_i \in \{0, 1, \ldots, M-1\}$, thus $X_i \in [0, 1)$.

Example 1.4

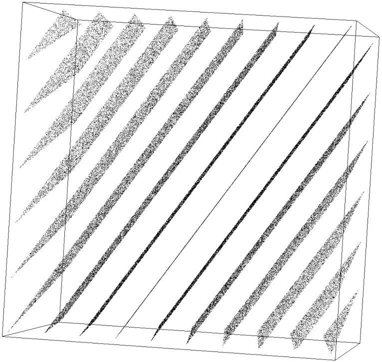
Cosider the choice of $a = 81$, $c = 35$, $M = 256$, and seed $Z_0 = 4$.

$$\begin{aligned}
Z_1 &= (81 \cdot 4 + 35) \mod 256 = 359 \mod 256 = 103 \\
Z_2 &= (81 \cdot 103 + 35) \mod 256 = 8378 \mod 256 = 186 \\
Z_3 &= (81 \cdot 186 + 35) \mod 256 = 15101 \mod 256 = 253 \\
&\ldots
\end{aligned}$$

The corresponding X_i are $X_1 = 103/256 = 0.4023438$,
$X_2 = 186/256 = 0.72656250$, $X_1 = 253/256 = 0.98828120$.

RANDU: A very poor choice of RNG

- Very popular in the 1970s (e.g. System/360, PDP-11).
- Linear congruential generator with $a = 2^{16} + 3$, $c = 0$, and $M = 2^{31}$.
- The numbers generated by RANDU lie on only 15 hyperplanes in the 3-dimensional unit cube!



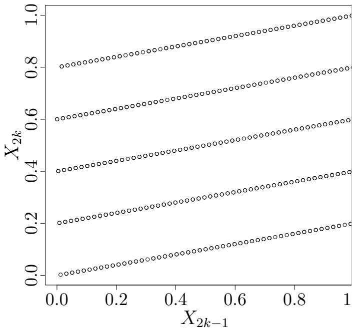
According to a salesperson at the time: "We guarantee that each number is random individually, but we don't guarantee that more than one of them is random."

The flaw on the linear congruential generator

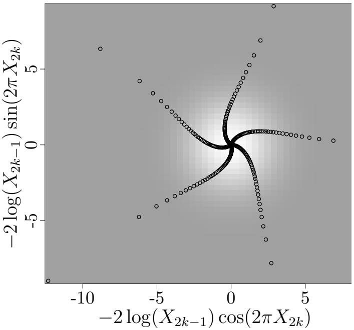
- "Crystalline" nature is a problem for every linear congurentrial generator.
- Sequence of generated values $X_1, X_2, \ldots$ viewed as points in an n-dimension cube lies on a finite, and often very small number of parallel hyperplanes.
- Marsaglia (1968): "the points [generated by a congruential generator] are about as randomly spaced in the unit n-cube as the atoms in a perfect crystal at absolute zero."
- The number of hyperplanes depends on the choice of a, c, and M.
- For these reasons do not use the linear congurential generator! Use more powerful generators (like e.g. the *Mersenne twister*, available in GNU R).

Another cautionary example

Linear congruential generator with $a = 1229$, $c = 1$, and $M = 2^{11}$.



Pairs of generated values (X_{2k-1}, X_{2k})



Transformed by Box-Muller method