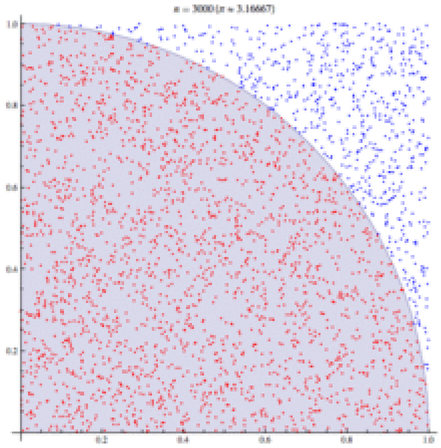
Introduction

--definition, history, and applications

Definition:

Monte Carlo methods (or **Monte Carlo experiments**) are a class of **computational algorithms** that rely on **repeated random sampling** to compute their results.



Typical Monte Carlo methods follow a particular pattern:

1. Define a domain of possible inputs.
2. Generate inputs randomly from a probability distribution over the domain.
3. Perform a deterministic computation on the inputs.
4. Aggregate the results.

Here’s a simple example as shown in figure:
Consider a circle inscribed in a unit square. Given that the circle and the square have a ratio of areas that is $\pi/4$, the value of π can be approximated using a Monte Carlo method:

1. Draw a square on the ground, then inscribe a circle within it.
2. Uniformly scatter some objects of uniform size (grains of rice or sand) over the square.
3. Count the number of objects inside the circle and the total number of objects.
4. The ratio of the two counts is an estimate of the ratio of the two areas, which is $\pi/4$. Multiply the result by 4 to estimate π.

To get an accurate approximation for π this procedure should have two other common properties of Monte Carlo methods. First, the inputs should **truly be random**. If grains are purposefully dropped into only the center of the circle, they will not be uniformly distributed, and so our approximation will be poor. Second, there should be **a large number of inputs**. The approximation will generally be poor if only a few grains are randomly dropped into the whole square. On average, the approximation improves as more grains are dropped.

History:

In 1930s, **Enrico Fermi** first experimented with the Monte Carlo method while studying neutron diffusion, but did not publish anything on it
In 1946, physicists at Los Alamos Scientific Laboratory were investigating radiation shielding and the distance that neutrons would likely travel through various materials which could not be solved with analytical calculations, **Stanisław Ulam** had the idea of using **random experiments**.
Later, He together with **John von Neumann** began to plan actual calculations. They gave the work a code name “Monte Carlo”, which is a reference to the Monte Carlo Casino in Monaco where Ulam's uncle would borrow money to gamble.
Monte Carlo methods were central to the simulations required for the Manhattan Project, though

severely limited by the computational tools at the time
In the 1950s, Monte Carlo methods were used at Los Alamos for early work relating to the development of the hydrogen bomb, and became popularized in the fields of physics, physical chemistry, and operations research.

Applications:

Physical Science

- In statistical physics, an alternative to computational molecular dynamics, and are used to compute statistical field theories of simple particle and polymer systems.
- In quantum physics, Monte Carlo methods solve the many-body problem for quantum systems.
- In experimental particle physics, Monte Carlo methods are used for designing detectors, understanding their behavior and comparing experimental data to theory.
- In astrophysics, they are used in such diverse manners as to model both the evolution of galaxies and the transmission of microwave radiation through a rough planetary surface.

Engineering:

Monte Carlo methods are widely used in engineering for sensitivity analysis and quantitative probabilistic analysis in process design.

Computational biology

Monte Carlo methods are used in computational biology, such for as Bayesian inference (贝叶斯推论) in phylogeny.

Applied statistics

Compare competing statistics for small samples under realistic data conditions.
Provide implementations of hypothesis tests that are more efficient than exact test

Games:

Monte Carlo methods have recently been incorporated in algorithms for playing games that have outperformed previous algorithms in games like **Go, Tantrix, and Battleship**.

Design and visuals

Monte Carlo methods have also proven efficient in solving coupled integral differential equations of radiation fields and energy transport, and thus these methods have been used to produce photo-realistic images of virtual 3D models.

Finance and business

Monte Carlo methods in finance are often used to calculate the value of companies, to evaluate investments in projects at a business unit or corporate level, or to evaluate financial derivatives.

Telecommunications

Monte Carlo methods are typically used to generate network users and their states so that performance is evaluated and, if results are not satisfactory, the network design goes through an optimization process.

Use in mathematics:

- Integration

Two problems encountered in deterministic numerical integration algorithms when the functions have many variables:

1st, the number of function evaluations needed increases rapidly with the number of dimensions.

2nd, the boundary of a multidimensional region may be very complicated, so it may not be feasible to reduce the problem to a series of nested one-dimensional integrals.

Monte Carlo methods provide a way out of this exponential increase in computation time. As long as the function in question is reasonably well-behaved, it can be estimated by randomly selecting points in 100-dimensional space, and taking some kind of average of the function values at these points.

More detailed info. at http://en.wikipedia.org/wiki/Monte_Carlo_integration

- Simulation-optimization

Refer to http://en.wikipedia.org/wiki/Stochastic_optimization

- Inverse Problems

The best-known importance sampling method, the Metropolis algorithm, can be generalized, and this gives a method that allows analysis of (possibly highly nonlinear) inverse problems with complex *a priori* information and data with an arbitrary noise distribution.

- Computational mathematics

Monte Carlo methods are useful in many areas of computational mathematics, where a "lucky choice" can find the correct result

e.g. Rabin's algorithm for primality testing:

For any *n* which is not prime, a random *x* has at least a 75% chance of proving that *n* is not prime. Hence, if *n* is not prime, but *x* says that it might be, we have observed at most a 1-in-4 event. If 10 different random *x* say that "*n* is probably prime" when it is not, we have observed a one-in-a-million event. In general a Monte Carlo algorithm of this kind produces one correct answer with a guarantee ***n* is composite, and *x* proves it so**, but another one without, but with a guarantee of not getting this answer when it is wrong **too often**—in this case at most 25% of the time.