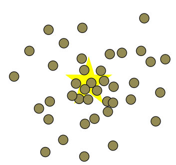
Stochastic Methods 1

- Definition and Use
- Monte Carlo methods

About Stochastic Optimization

Stochastic Optimization methods involve random variables. The actual word “stochastic” is derived from a Greek word meaning “aim” or “target”.

Suppose a small target, like a rock or a stick, is placed on a hillside. Many arrows are shot at it. Later the target is removed and the arrows are left.

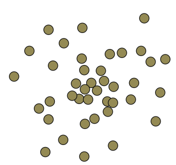


- The arrows’ positions are basically random.

About Stochastic Optimization

Stochastic Optimization methods involve random variables. The actual word “stochastic” is derived from a Greek word meaning “aim” or “target”.

Suppose a small target, like a rock or a stick, is placed on a hillside. Many arrows are shot at it. Later the target is removed and the arrows are left.



- The arrows’ positions are basically random.
- Nevertheless, you can still use their positions to estimate the location of the target.

About Stochastic Optimization

Like using randomly-positioned arrows to estimate the position of a target, stochastic methods have the goal of gaining information out of randomness.

About Stochastic Optimization

Stochastic Optimization is used when…

- the input or data is random or unpredictable (like how long it takes to travel somewhere)
- the input is prone to measurement error (like scientific data from imperfect instruments)
- the data or equations are known, but too complex to handle conventionally (like finding the probability of winning at solitaire)

About Stochastic Optimization

All stochastic processes involve a large amount of input or trials; in general, the more trials there are, the more accurate the results will be.

For example, let’s say you’re trying to find the probability that an actual coin flips “heads”. The best way to do this is by flipping the coin repeatedly and recording the results. This is considered a stochastic process because it involves repeated sampling of essentially random inputs.

About Stochastic Optimization

The reason you need a lot of data is that if you flipped a coin just once and it came up “heads”, you could reasonably conclude that it would always come up “heads”, 100% of the time.

It’s only over many coin flips that the real probability becomes more clear.

Convergence

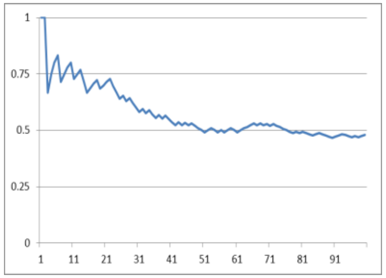
With a large enough sample, you will see a phenomenon called *convergence*.

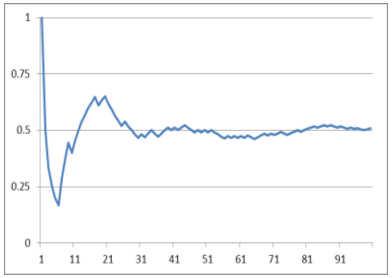
Let’s say you flipped a coin 100 times, and after each flip recorded the percent of “heads” out of the total number of flips.

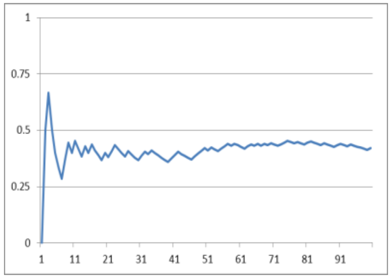
Your first flip would either give you 0% or 100%; your next, 0% or 50% or 100%; your next, 0%, 25%, 50%, 75%, or 100%, and so on.

Convergence

Here are some charts, made in Excel, showing percent heads over 100 “coin flips” (random numbers either 0 or 1)

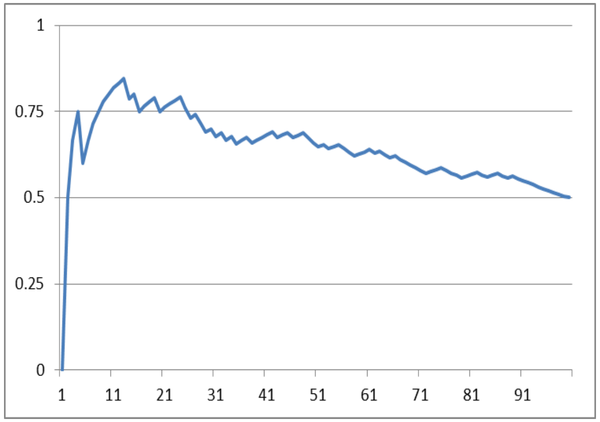






These charts all show convergence, though not all of them are converging to 0.5. In general, after a large number of samples, the line will flatten out at the “correct” answer. That is known as *convergence*.

Convergence



In contrast, this chart does not show convergence yet; we would need more data here.

We know this because the line is not flattening out.

About Stochastic Methods

The primary disadvantage of stochastic methods is that their accuracy is not very good, though it's usually close enough. For this reason they are typically not used when another method is feasible.

The other disadvantage of stochastic methods is without computer assistance, they are slow. This is why they have only become widely used in the computer era.

Monte Carlo Methods

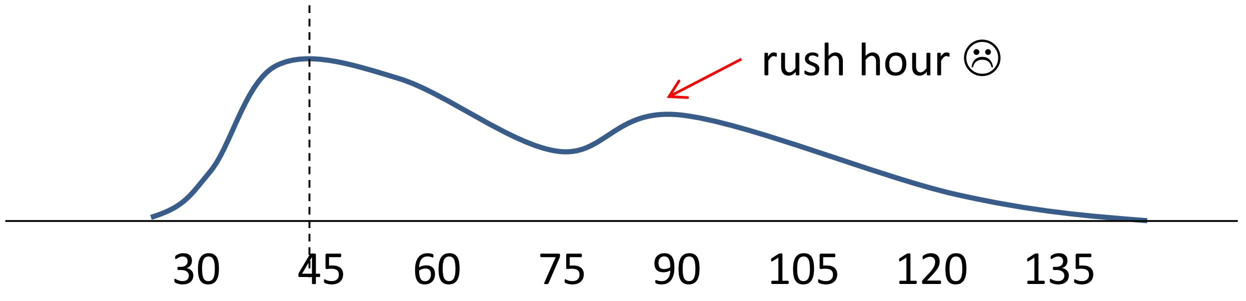
Monte Carlo methods, named after a famous gambling city, involve large numbers of computer simulations with randomly selected inputs.

Monte Carlo methods are used when the inputs involve a large amount of uncertainty, or when several factors depend on each other.

Monte Carlo: step 1

Define the domain – what are the limits on the possible inputs? Are there different probabilities?

For example, if an input was the amount of time it takes to drive 50 miles on the freeway, this variable would be limited to positive numbers below, say, 200 minutes, but more likely around 45 minutes, possibly with a curve like this:

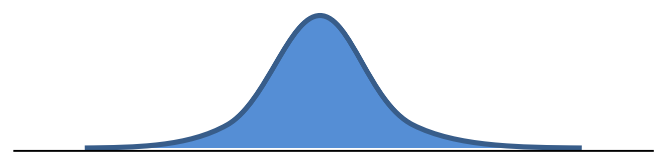


Monte Carlo: Step 2

Once you know what sort of domain limits and probability distribution your inputs will have, the next step is to generate random numbers to mimic each possible input. This usually involves a computerized random-number generator.

Julia can generate random numbers with several useful patterns. Here are two:

evenly distributed
`rand()`



normally distributed
`randn()`

Monte Carlo: Steps 3 and 4

Once you have your randomly-generated inputs, the next step is to perform whatever testing or math is required using those inputs, in effect running a simulation to see what outcome would result from those inputs.

Finally, you would record the results and repeat the experiment many, many times.

Practice Problem 1

1a. Using Julia, generate five random numbers between 0 and 1.

1b. Generate five random numbers between 40 and 45.

1c. Generate five random numbers from a normal distribution with mean 0 and standard deviation 1.

1d. Generate five random numbers from a normal distribution with mean 20 and standard deviation 4.2.

Practice Problem 2

You are shipping a product from A to B.

- The time it takes from production to the product leaving A is evenly distributed between 1 and 3 hours.
- The travel time from A to airport 1 is a normal distribution with mean = .8 and standard deviation .35 hours.
- The loading time onto the plane at airport 1 is evenly distributed between .5 and 2 hours.
- The flight time from airport 1 to airport 2 is a normal distribution with mean = 4 and standard deviation .1 hours.
- The unloading time at airport 2 is always .25 hours.
- The travel time from airport 2 to B is a normal distribution with mean = 3 and standard deviation = .8 hours.

Run one simulation of total travel time.

Practice Problem 3

Write a program that generates random numbers and finds the total travel time for the shipping scenario in problem 2.

Nest your program in a loop that will find the average travel time over *n* trials.

Calculate results for n = 20, 50, 100, 500.

Practice Problem 4

Using the same scenario, modify your code so that:

- if step 1 (production to leaving A) takes 2-3 hours, the next step (travel to airport 1) changes to mean 1.2 with standard deviation .6.
- if the total time before the last step is over 9 hours, the final step (delivery to B) changes to mean 5.1 and standard deviation 1.9.

Simulate 500 trials five times and estimate the average total travel time.