Lab13 Monte Carlo simulation (12.02.19) CS103 Fall 2019

author john k johnstone jkj at uab dot edu
course CS103 Fall 2019
license MIT
version Fall 2019

materials

- *lab13_19fa103_mc.pdf* (this document)
- *lab13_19fa103.py*
- pascalMC.py (an example of a Monte Carlo simulation)

purpose

goal Monte Carlo simulation of the Monty Hall problem

context

Recall the Monty Hall problem. The game show host Monty Hall shows you 3 doors: 2 doors contain nothing and 1 door contains a prize. Monty asks you to choose a door. You choose a door. Monty then opens one of the doors you didn't open, which you see does not contain the prize. (Monty always opens a non-prize door.) He now asks you whether you'd like to switch your door. Should you switch?

One approach is to analyze the probability formally. Another approach is to simulate the experiment a large number of times, using Monte Carlo simulation, and discover an approximation to the probability that way. Let's try the Monte Carlo approach. Then see how close to the perfect number it gets.

In a Monte Carlo simulation, you have an experiment (e.g., flip a coin, throw a pair of die 24 times, a single contestant on the Monty Hall show). Then you have a trial of several experiments, which computes a probability (e.g., several flips of a coin to determine probability of heads). A Monte Carlo simulation performs a batch of trials, each yielding a different answer for the probability, analyzes the quality of this set of answers (using standard deviation), then either returns the mean of the answers (if the quality is high) or runs a new larger batch of trials (if the quality is low). We will double the size of the batch every time the quality is too low.

My slides on Monte Carlo simulation (lectures 25-26), code (day26), and the Python documentation on random numbers (the random module) may be helpful. I have included pascalMC.py as a good example of a Monte Carlo simulation.

in-class exercise

- solution of trialMonty

exercises

Experiment

- Choose the prize door randomly and uniformly over [1,2,3].
- Choose your door randomly and uniformly over [1,2,3].
- Monty Hall chooses the door that is not the prize door and not your door.
- Depending on your strategy, you may or may not switch to the remaining door.

With the never-switch policy, you never switch.
With the always-switch policy, you always switch.
With the random-switch policy, you flip a fair coin to decide if you switch.
The experiment yields an answer: did you win the prize?

Use a batch of 100 trials as the first batch size.

Sets may be useful in implementing the experiment.

challenges

- compute π using Buffon's algorithm and Monte Carlo simulation
- compute the probability of a straight flush (poker hand) using Monte Carlo simulation

deliverables

A+: add one of the challenges
A: Monte Carlo simulation of Monty Hall
B: in-class exercise
C: attendance