Monte Carlo Simulation

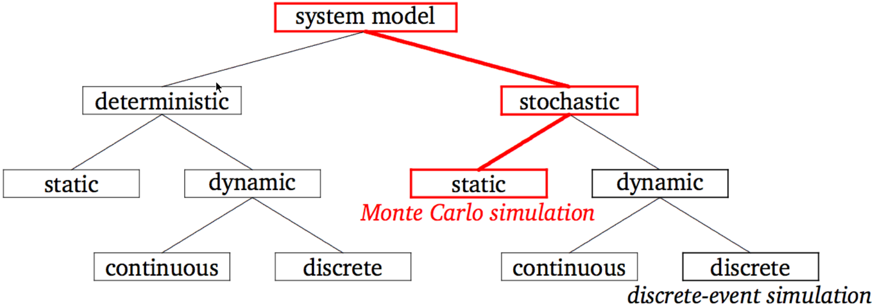
DCS 307: Simulation
Winter 2023

13 February 2023

Outline

- Overview of Monte Carlo Simulation
- Example 1: Galileo's Dice
- Example 2: Craps

Monte Carlo Simulation



Monte Carlo Simulation

- Monte Carlo model: stochastic & static
  - *stochastic*: (at least some of) state variables are random
  - *static*: time evolution is not important

- Based on the *frequency theory of probability*:

$$\Pr(\mathcal{A}) = \lim_{n \to \infty} \frac{n_a}{n}$$

- Relies on generation of good pseudo-random numbers

Two Approaches to Probability

- *Axiomatic* and *experimental* approaches are complementary
- Slight changes in assumptions can sink an axiomatic solution
- In other cases, an axiomatic solution is intractable

- Monte Carlo simulation can be used as an alternative in either case

Example 1: Galileo's Dice

- Three fair dice are rolled; which sum is more likely: 9 or 10?
- *Axiomatic*: $6^3 = 216$ possible outcomes

$$\Pr(X = 9) = \frac{25}{216} \cong 0.116$$

$$\Pr(X = 10) = \frac{27}{216} = \frac{1}{8} = 0.125$$

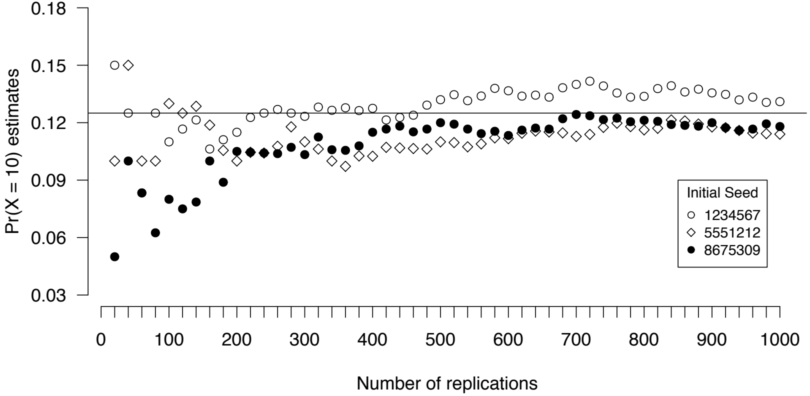
- *Experimental*: program `galileo` estimates prob. of sums 3–18

```
> probs <- galileo(nrep = 1000, seed = 8675309)
> probs[9]    # 9th entry is sum of 9
[1] 0.138
> probs[10]   # 10th entry is sum of 10
[1] 0.141
```

- Drawback: produces an *estimate*

Monte Carlo Simulation Results

- More replications does not guarantee a better estimate
- Frequency probability estimates converge slowly and somewhat erratically



Point Estimate Considerations

- How many significant digits should be reported?
- Solution: run the simulation multiple times
- One option: Use different initial seeds for each run
  Caveat: What effect does seed have on *warm-up*?

Example 2: Craps

- Roll a pair of fair dice and sum the up faces
- If 7 or 11, win immediately
- If 2, 3, or 12, lose immediately
- Otherwise, sum becomes "point"
  Roll until point is matched (win) or 7 (loss)

- What is $\Pr(\mathcal{A})$, the probability of winning at craps?

Craps: Axiomatic Solution

- Requires conditional probability
- Axiomatic solution: $\Pr(\mathcal{A}) = 244/495 \cong 0.493$
- Underlying mathematics must be changed if assumptions change

  E.g., unfair dice
- Axiomatic solution provides a nice consistency check for (easier) Monte Carlo simulation

Craps: Axiomatic Solution — More Details!

$$\Pr(\text{win}) = \sum_{x=2}^{12} \Pr(x)\Pr(\text{win} \mid x)$$

where $\Pr(x)$ is the probability of x on the first toss and $\Pr(\text{win} \mid x)$ is the probability of winning if the first toss is x. The $\Pr(x)$ probabilities are listed in exercise 2.2. To evaluate the $\Pr(\text{win} \mid x)$ probabilities first recognize that five of them are specified by the rules of the game:

$$\Pr(\text{win} \mid 7) = \Pr(\text{win} \mid 11) = 1$$

and

$$\Pr(\text{win} \mid 2) = \Pr(\text{win} \mid 3) = \Pr(\text{win} \mid 12) = 0.$$

To evaluate the other six conditional probabilities consider, for example, $\Pr(\text{win} \mid 4)$. This is the probability that a 4 will be rolled before a 7. Equivalently, this is the probability of drawing a red ball before a green ball when balls are drawn (with replacement) from an urn containing a $3/36$ proportion of red balls (the 4's), a $6/36$ proportion of green balls (the 7's) and a $27/36$ proportion of blue balls (the other possibilities). From example 6.8 this probability is $(3/36)/(3/36 + 6/36)$. The other five conditional probabilities are calculated in the same way with the result:

$$\begin{aligned}
\Pr(\text{win} \mid 4) = \Pr(\text{win} \mid 10) &= \frac{3/36}{3/36 + 6/36} = 3/9 \\
\Pr(\text{win} \mid 5) = \Pr(\text{win} \mid 9) &= \frac{4/36}{4/36 + 6/36} = 4/10 \\
\Pr(\text{win} \mid 6) = \Pr(\text{win} \mid 8) &= \frac{5/36}{5/36 + 6/36} = 5/11.
\end{aligned}$$

All that remains then is some arithmetic with the result

$$\Pr(\text{win}) = \sum_{x=2}^{12} \Pr(x)\Pr(\text{win} \mid x) = \frac{244}{495} \simeq 0.493.$$

Craps: Specification Model

Algorithm:

```
wins = 0
for ( i in 1:nrep ) {
    roll = sample( 1:6, 1 ) + sample( 1:6, 1 )
    if ( roll == 7 || roll == 11 ) {
        wins = wins + 1
    } else if ( roll != 2 && roll != 3 && roll != 12 ) {
        point = roll
        while ( TRUE ) {
            roll = sample( 1:6, 1 ) + sample( 1:6, 1 )
            if ( roll == point ) wins = wins + 1
            if ( roll == point || roll == 7 ) break
        }
    }
}
print( wins / nrep )
```

Craps: Computational Model

- For `nrep` $= 100$ and three different initial seeds

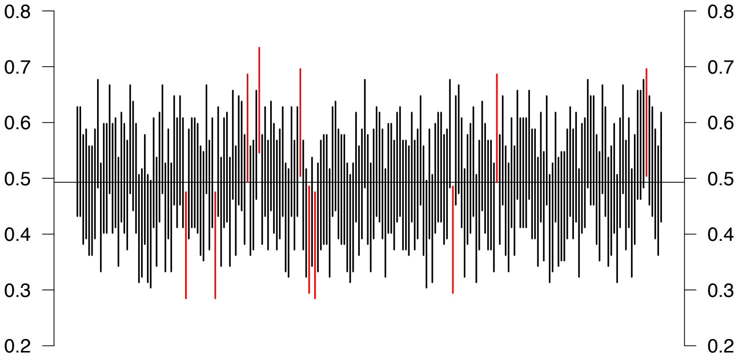
$$\hat{\Pr}(\mathcal{A}) = 0.53, 0.49, \text{ and } 0.46$$

```
> craps(nrep = 100, seed = 12345)
[1] 0.51

> craps(nrep = 100, seed = 8675309)
[1] 0.49

> craps(nrep = 100, seed = 37)
[1] 0.52
```

- These results are consistent with 0.493 (axiomatic solution)
- This (relatively) high probability is attractive to gamblers, yet ensures the house will win in the long run

Example (Lookahead)

- Program `craps` was replicated 200 times with `nrep` $= 100$
- For each, construct a 95% confidence interval estimate



(the left-most interval is 0.53 ± 0.10)

- Horizontal line is $\Pr(\mathcal{A}) \cong 0.493$ (axiomatic solution)

In-Class Work: Monte Carlo

1. Implement your own function for craps, but simulating <u>unfair</u> dice, where each die is loaded in the following way:
   - Each of 2, 3, 4, 5 are twice as likely as a 1.
   - A 6 is four times as likely as a 1.
2. Execute your function 100, 1000, and 10 000 times, each time storing the results into a vector.
3. Use your vector (of 100, then 1000, then 10 000) to provide a 95% confidence interval for the estimate of winning, e.g.:

```
t.test(estimates)
```