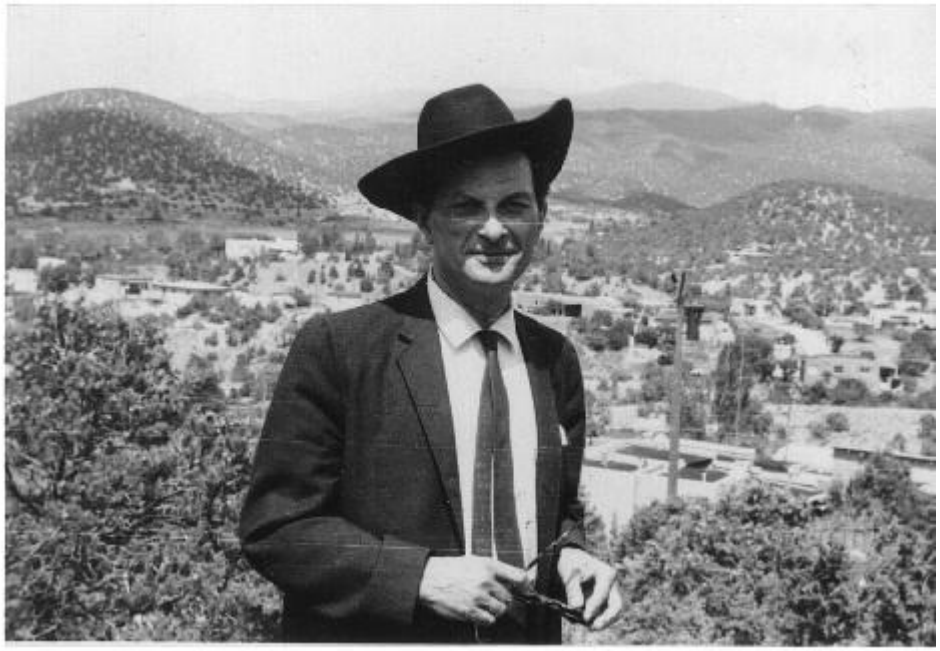
Lecture 13: Monte Carlo method and plotting

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

Monte Carlo method

Developed by Stanislaw Ulam



Monte Carlo Method

Key Idea:

- Use repeated simulation of a random process to compute a non-random quantity

You have already used it !

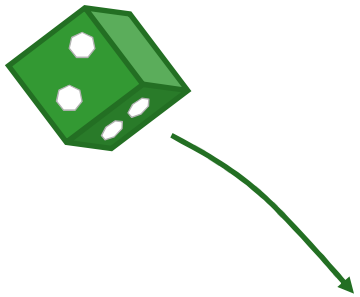
Example: Modeling a game of dice

A casino plays a game of dice. If the dice hits one, the casino wins. Otherwise the player wins.

How much should the house wager against a player who wagers $12?

This is a deterministic question

- What is the wager that will give an expected win of zero.



$$E_c = \frac{1}{6} * 12 - x * \frac{5}{6}$$

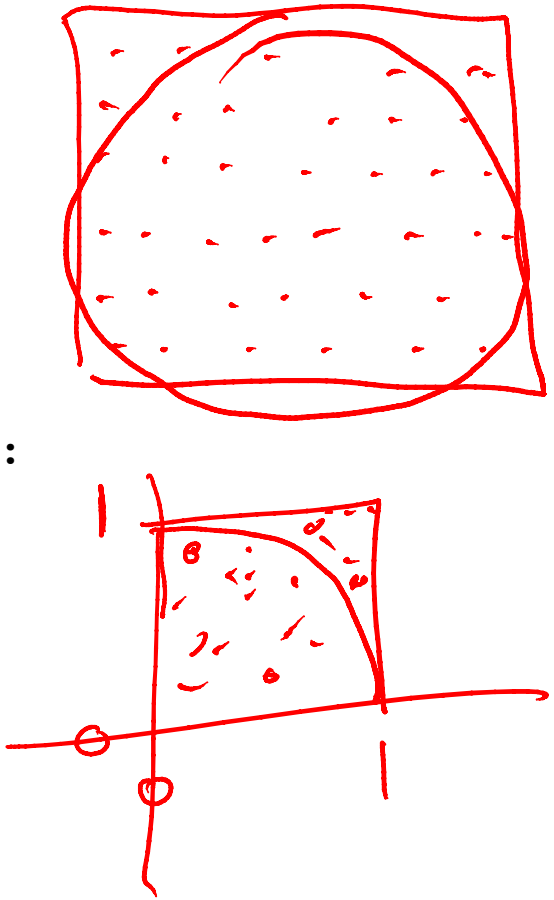
Monte Carlo Simulation

Code can be found here: http://bit.ly/12aEWdH

It's not just for probabilistic processes

Deterministic algorithms can sometimes have simpler Monte Carlo analogs

```
N=1000
NC=0.0
x=-1.0
for i in xrange(2*N):
    y=-1.0
    for j in xrange(2*N):
        if x*x+y*y<1:
            NC=NC+1
        y=y+1.0/N
    x=x+1.0/N
print NC/(N*N)
```



Monte Carlo Version

```
n = 5000000
count = 0
for t in xrange(0,n):
    x = rr.random()
    y = rr.random()
    if x*x + y*y < 1.0:
        count += 1

print 4* (float(count) / n)
```

Code with plots can be found here: http://bit.ly/XbdLeE

More about random

It's a Pseudo random number generator

- Given a seed x produces a sequence of numbers that 'look' random
  - random.seed(x) initializes the seed
- Given the same seed you will get the same sequence
  - Try it!

```
import random
random.seed(10)
for t in xrange(0, 5):
    print random.random()
```

  - If you run the code above many times, you will get the same sequence of numbers
  - If you change the seed you get a different sequence

Plotting

import pylab

Basic logic

- Create named figures
- Add lines and points to them
- show them on the screen or save them to a file

Main Plotting Routines

pylab.figure(*name*)

- Switch to a new figure
- (or to a previously used figure with that name)

pylab.plot(*lx, ly*)

- Plot a set of points
  - lx is a list of x coordinates
  - ly is a list of y coordinates
- alternative pylab.plot(*ly*)
  - lx is then equal to range(0, len(ly))

pylab.show()

- shows the plot in a window
- stops the execution until the window is closed

More on plot

You can use named parameters to pass properties

- pylab.plot(*lx, ly, porperty=val, ...*)
- Ex.
  - ```
    pylab.plot([1,2,3], [3,2,1], color='r',
    marker='*', markersize=20, linewidth=6.0)
    ```
    ```

More pylab properties

pylab.xlabel(*labelstring*)

- label of the x axis

pylab.ylabel(*labelstring*)

- label of the y axis

pylab.title(*titlestring*)

- title of the plot

pylab.axis([xlow, xhigh, ylow, yhigh])

- range for the x and y axis

To learn more

More documentation on available properties can be found here:

- http://matplotlib.org/users/pyplot_tutorial.html
- Note they import matplotlib.pyplot instead of pylab
  - for our purposes, they are the same thing.