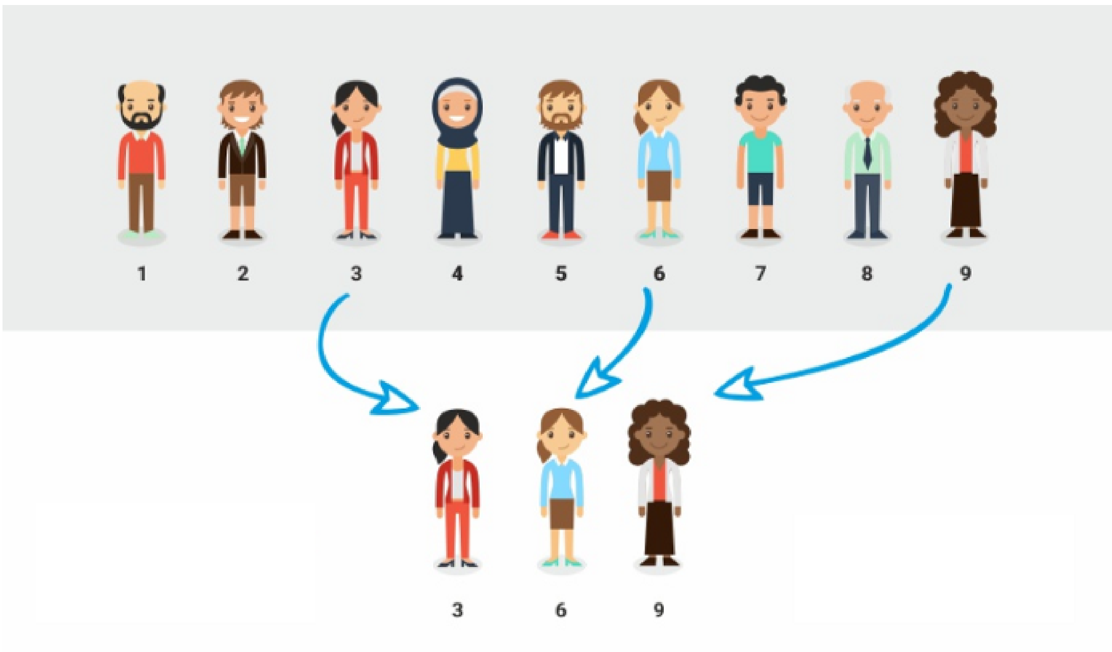
Monte Carlo Methods

MATH1900: Machine Learning

Location: http://people.sc.fsu.edu/~jburkardt/classes/ml_2019/monte_carlo/monte_carlo.pdf



Sampling a big data set.

A Monte Carlo method is a computational process that uses random sampling. This lab introduces a few examples, in which we estimate averages, areas, and integrals.

1 Example: Estimating statistics of a very large set

A surveys claims that 68% of the population are fans of the Peppa Pig show, and that each such viewer watches for an average of 23 minutes a day. These claims were not made by asking every person in the country, but rather, by selecting a random sample of the population, small enough to treat individually, but large enough to be representative. In machine learning, it may be the case that a dataset contains millions of entries, and that each query is rather expensive. If we only need a rough estimate for our result (say, just two digits of accuracy), then random sampling may give us an answer that is good enough.

For simplicity, we will assume that our data is simply a vector x of n real values, and that we want the minimum, average, and maximum. Instead of checking all n values, we will choose a much smaller number of values k. To carry out our task, we need to generate k distinct random indices i such that $0 \leq i < n$. We do this with `np.random.choice()`:

```
import numpy as np
n = 1000
mu = 23
sigma = 4
x = mu + sigma * np.random.randn ( n )
x_min = np.min ( x )
```

```
x_mean = np.mean ( x )
x_max = np.max ( x )

k = 5
i = np.random.choice ( n, k, replace = False )
xs = x[i]
xs_min = np.min ( xs )
xs_mean = np.mean ( xs )
xs_max = np.max ( xs )
```

2 Exercise: Estimating height and weight statistics

The file *hw_25000.txt* contains 25,000 records from the physical exams of a population. Each record consists of three values: an index (from 1 to 25,000), a height (inches), and a weight (pounds). Suppose we needed to estimate the minimum, mean, maximum and standard deviation of the height and weight of the entire population, but cannot afford to view all the records. Estimate these 4 statistical quantities for height and for weight, using random samples of size 10, 100, and 1,000. Compare with the exact values.

```
                   <-----Height----->                                  <-----Weight----->
     N     Min       Mean       Max        Std        Min       Mean       Max        Std
        ---------  ---------  ---------  ---------  ---------  ---------  ---------  ---------

    10  .........  .........  .........  .........  .........  .........  .........  .........
   100  .........  .........  .........  .........  .........  .........  .........  .........
 1,000  .........  .........  .........  .........  .........  .........  .........  .........
25,000  .........  .........  .........  .........  .........  .........  .........  .........
```

3 Example: Estimate the integral of the "humps" function

The mean value theorem for integrals says

$$\int_a^b f(x)\,dx = \bar{f} * (b-a)$$

where $\bar{f}$ is the average value of $f(x)$ over $[a,b]$. This gives us a way to estimate an integral by sampling. Simply choose a random sample of points x in $[a,b]$, average the values $f(x)$, and multiply by $(b-a)$. As the number of points used increases, we improve our estimate of $\bar{f}$ and hence of the integral.

```
def humps ( x )
  y = 1.0 / ( ( x - 0.3 )**2 + 0.01 ) + 1.0 / ( ( x - 0.9 )**2 + 0.04 ) - 6.0
  return y

import numpy as np

a = 0.0
b = 2.0
exact = 29.32621380439115

for logn in range ( 2, 6 ):
  n = 10**logn
  x = a + ( b - a ) * np.random.rand ( n )
  fx = humps ( x )
  estimate = ( b - a ) * np.mean ( fx )
  err = exact - estimate
  print ( 'n =', n, 'estimate =', estimate, 'error =', err )
```

4 Exercise: Estimate the integral of $\frac{1}{\sqrt{x}}$

What is the exact value of $\int_0^4 \frac{1}{\sqrt{x}} dx$?

Use a Monte Carlo method to estimate this quantity. Choose a value of n large enough that your error is less than 0.001.

```
Exact integral value = ..........
Estimated value      = ..........  using N = .......... samples
Error                = ..........
```

5 Example: Estimating the area of an ellipse

If an ellipse has the formula:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1-$$

then it lies inside the rectangle $-a \leq x \leq a, -b \leq y \leq b$. Hence, the area inside the ellipse must be no greater than $4 * a * b$. How could we estimate this area? If we randomly sample the surrounding rectangle, then it is reasonable, (and correct) to estimate the ellipse area by

$$\text{ellipse area} \approx \frac{\text{points inside ellipse}}{\text{points inside rectangle}} * \text{area of rectangle}$$

So we can generate n points inside the rectangle, and then we can determine if each point is inside the ellipse by checking if $\frac{x^2}{a^2} + \frac{y^2}{b^2} \leq 1$.

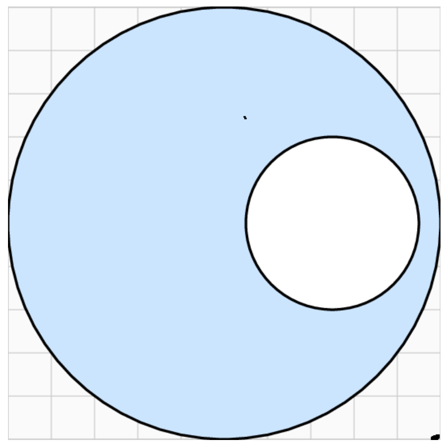
```
import numpy as np
a = 3
b = 2

n = 100
x = - a + 2 * a * np.random.rand ( n )
y = - b + 2 * b * np.random.rand ( n )
i = np.count_nonzero ( x**2/a**2 + y**2/b**2 <= 1 )
area_estimate = i / n * 4 * a * b
```

It turns out that this area is exactly $pi * a * b$. Can you modify this procedure so that it increases n until the error is less than 0.01?

6 Exercise: Estimating the area of a circle minus a circle

Let C_1 be the unit circle of radius 1 and center (0.0,0.0), and C_2 be the circle of radius 0.1 and center (0.5,0.0). We want to estimate the area of C_3, the shape formed by C_1 "minus" C_2, in other words, C_1 with a hole in it.



The C_3 region.

What are the areas of C_1, C_2, and C_3?

Use a Monte Carlo method to estimate the area of C_3. Try to use a value of n large enough that your error is less than 0.01. Note that, in your call to `np.count_nonzero ( )` you now need **two** conditions to be true in order that a point is inside C_1 and not inside C_2. You specify these, using an ampersand, as in:

```
i = np.count_nonzero {\tt{( condition 1 & condition 2 )}}.
```

Think for a moment about what condition 1 and condition 2 should be!

```
    exact C_1 area .........
    exact C_2 area .........
    exact C_3 area .........
estimated C_3 area .........  using N = .........
             error .........
```