

Exercise: approximate π with Metropolis MCMC

Modify the direct MC algorithm from last session to use a Markov chain instead of directly drawn (x,y) positions.

Start from an initial position within the unit square and update according to the rule below.

Choose the trial positions randomly and symmetrically around the current position in the Markov chain, e.g., within a small square or circle centered around the current position.

```
def approximate_pi_naive(n_sample):
    summation = 0
    for i in range(n_sample):
        x, y = np.random.rand(2)
        if x * x + y * y <= 1.0:
            summation += 1
    return 4.0 * summation / float(n_sample)
```

$$(x_n, y_n) = \begin{cases} (x_{\text{trial}}, y_{\text{trial}}), & x_{\text{trial}}, y_{\text{trial}} \in [0, 1] \\ (x_{n-1}, y_{n-1}), & \text{else} \end{cases}$$