

A6

Reinforcement Learning Study

2021-01-25

차 례

| | |
|--------------------------------------------------------|---|
| Implementation -basic | 2 |
| Implementation - The first-timer would write | 3 |
| Running estimate approach | 4 |
| Convergence trajectory | 5 |

Implementation -basic

```
import numpy as np
np.random.seed(1234)

MC_N = 10**3

x = np.random.uniform(0,1,size=MC_N)*2-1 # runif generates U(0,1)
y = np.random.uniform(0,1,size=MC_N)*2-1
t= np.sqrt(x**2+y**2)
pi_hat = 4*np.sum(t<=1) /MC_N

pi_hat

## 3.06
```

Implementation - The first-timer would write

```
np.random.seed(1234)

MC_N = 10**6
count = 0

for MC_i in range(MC_N):
    x_i = np.random.uniform(0,1)*2-1 # runif generates U(0,1)
    y_i = np.random.uniform(0,1)*2-1
    t_i= np.sqrt(x_i**2+y_i**2)

    if t_i <= 1 :
        count+=1

pi_hat = 4*count/MC_N
pi_hat

## 3.140204
```

Running estimate approach

```
import time

np.random.seed(1234)
beg_time = time.time() # to time
old_est = 0
n = 1
MC_N = 10**6

while True :
    x_i = np.random.uniform(0,1)*2-1
    y_i = np.random.uniform(0,1)*2-1
    t_i = np.sqrt(x_i**2+y_i**2)
    A_n = 4*(t_i<=1)
    new_est = ((n-1)/n)*old_est + (1/n)*A_n

    if n > MC_N:
        break

    n+=1
    old_est = new_est

print(new_est)

## 3.1402048597951002

end_time =time.time() # to time

print("Time difference of {}".format(end_time-beg_time)) # to time

## Time difference of 13.87293267250061
```

Convergence trajectory

```
import time

np.random.seed(1234)
beg_time = time.time() # to time
old_est = 0
n = 1
MC_N = 10**6
result = np.repeat(0, MC_N+1).astype('float') # to save

while True :
    x_i = np.random.uniform(0,1)*2-1
    y_i = np.random.uniform(0,1)*2-1
    t_i = np.sqrt(x_i**2+y_i**2)
    A_n = 4*(t_i<=1)
    new_est = ((n-1)/n)*old_est + (1/n)*A_n

    result[n] = new_est # to save

    if n >= MC_N:
        break

    n+=1
    old_est = new_est

print(new_est)
```

```
## 3.140203999999996
```

```
end_time = time.time() # to time
print("Time difference of {}".format(end_time-beg_time)) # to time
```

```
## Time difference of 14.106273889541626
```

Plot

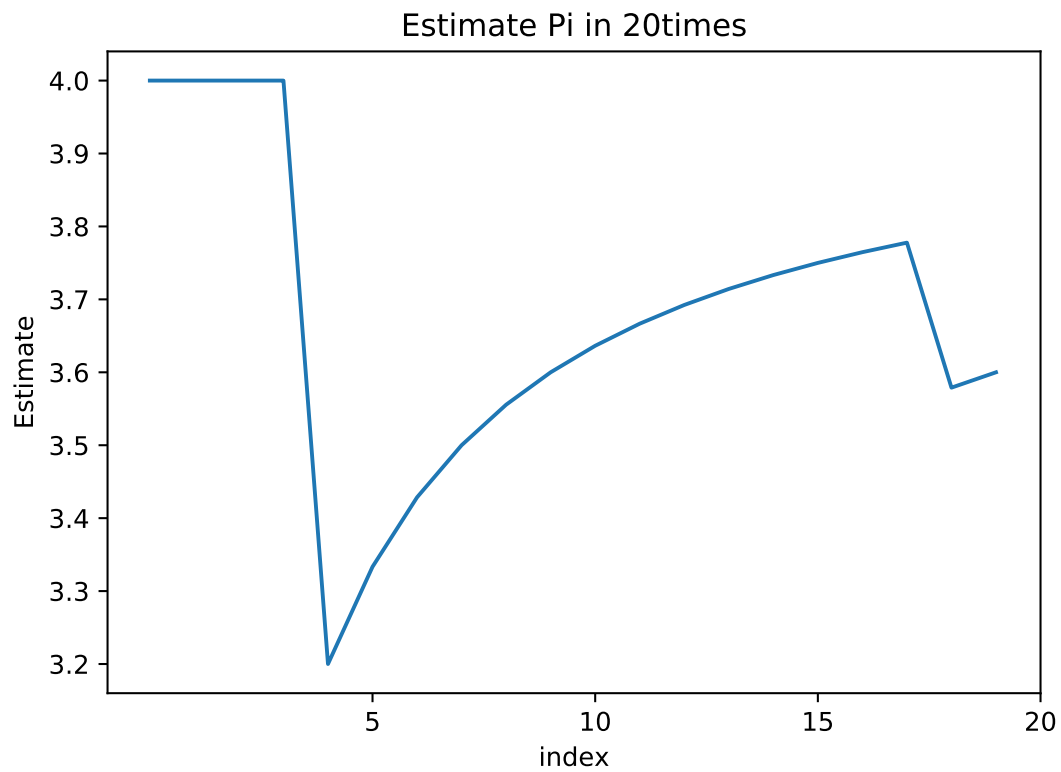
```
import matplotlib.pyplot as plt

plt.plot(result[1:21])
plt.xlabel('index')
```

```
plt.ylabel('Estimate')
plt.xticks([5,10,15,20])
```

```
## ([<matplotlib.axis.XTick object at 0x000000002B3732C8>, <matplotlib.axis.XTick object at 0x000000002B9C6CC8>]
```

```
plt.title("Estimate Pi in 20times")
```

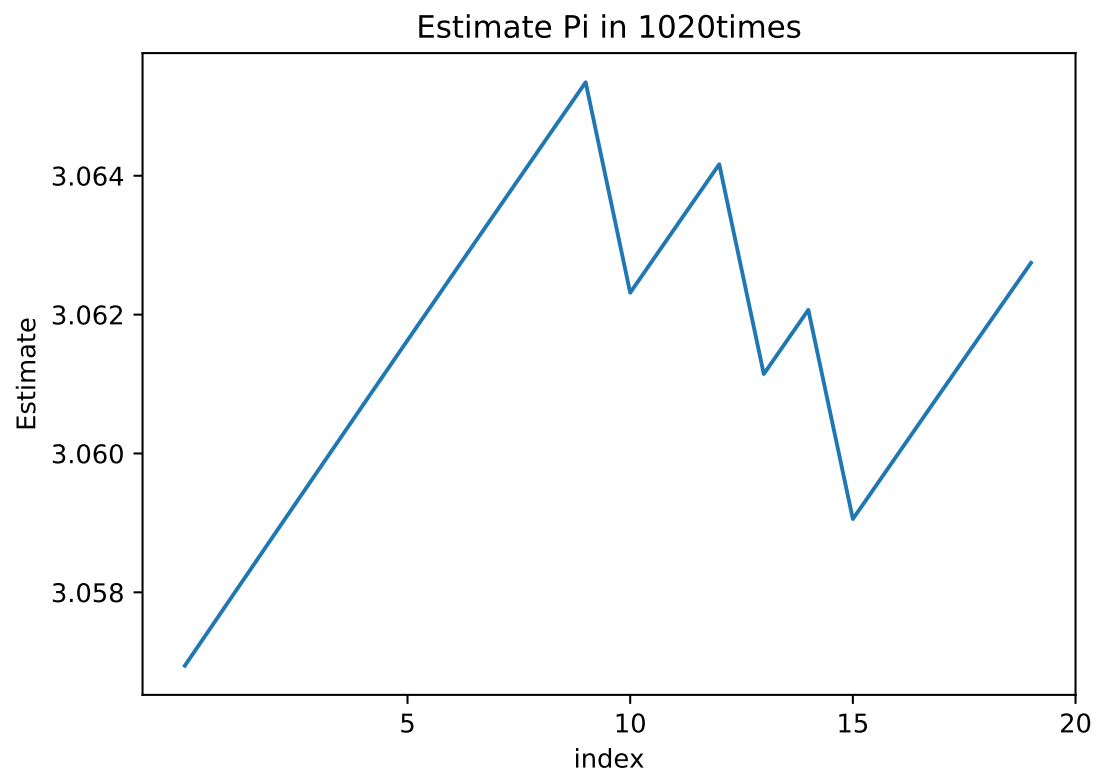


```
import matplotlib.pyplot as plt
```

```
plt.plot(result[1001:1021])
plt.xlabel('index')
plt.ylabel('Estimate')
plt.xticks([5,10,15,20])
```

```
## ([<matplotlib.axis.XTick object at 0x0000000034AAB108>, <matplotlib.axis.XTick object at 0x0000000034AA5248>]
```

```
plt.title("Estimate Pi in 1020times")
```



"Done "

```
## [1] "Done "
```